UNIVERSITÉ DE MONTRÉAL

PLANIFICATION MULTI-OBJECTIF DE TRAJECTOIRES POUR MANIPULATEURS ROBOTIQUES PAR LAGRANGIEN AUGMENTÉ ET TECHNIQUES NEURO-FLOUES

AMAR KHOUKHI

DÉPARTEMENT DE GÉNIE MÉCANIQUE

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION

DU DIPLÔME DE PHILOSOPHIÉA DOCTOR

(GÉNIE MÉCANIQUE)

JUIN 2007

© Amar Khoukhi, 2007.

Library and Archives Canada

Published Heritage Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et Archives Canada

Direction du Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-35514-5
Our file Notre référence
ISBN: 978-0-494-35514-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

PLANIFICATION MULTI-OBJECTIF DE TRAJECTOIRES DES MANIPULATEURS ROBOTIQUES PAR LAGRANGIEN AUGMENTÉ ET TECHNIQUES NEURO-FLOUES

présentée par: KHOUKHI Amar

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

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

M. MUREITHI Njuki, Ph.D., Président

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

M. BALAZINSKI Marek, Ph.D., membre et codirecteur de recherche

M. BOUKAS El-Kébir, Ph.D., membre

M. SAAD Maarouf, Ph.D., membre

À la mémoire de mon père,

À toute ma famille pour leur grand support,

Et surtout à ma femme et mes deux enfants Rayan et Adam.

REMERCIEMENTS

Je voudrais avant tout remercier mon directeur de recherche Luc Baron, et mon codirecteur de recherche Marek Balazinski, pour m'avoir accordé leur confiance totale. Leurs qualités scientifiques et humaines m'ont été d'une précieuse aide.

Je remercie également, les distingués membres du jury qui ont accepté de lire et évaluer ma thèse.

Je remercie aussi le professeur Kudret Demirli de l'université Concordia pour sa collaboration et ses conseils qui ont aidé à la réalisation d'une partie de cette thèse.

Ma reconnaissance va au Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) pour son soutien financier.

Merci à tous mes amis, d'ici et d'ailleurs.

Enfin, un grand merci à toute ma famille; à la mémoire de mon père, à ma mère, mes frères et sœurs, à ma femme et mes deux bonheurs Rayan et Adam pour leur soutien et amour inconditionnels.

RÉSUMÉ

Dans cette thèse, on considère le problème de planification multi-objective de trajectoire des manipulateurs robotiques. D'abord, le problème de planification hors ligne est étudié. À partir des modèles cinématique et dynamique du robot, et ceux de l'espace de travail et de la tâche, le problème est formulé dans le cadre du calcul variationnel, comme un programme non linéaire sous contraintes. La technique du Lagrangien augmenté est appliquée à une représentation découplée de la dynamique du robot afin de résoudre le problème résultant de commande optimale. Cette étude est développée pour deux classes de manipulateurs robotiques; les robots sériels redondants et les robots parallèles. Les différences principales, par rapport au problème de planification de trajectoire, entre les robots sériels et parallèles sont revues et l'approche proposée tient compte de ces différences. Des simulations réalisées sur des modèles de ces robots montrent l'efficacité de cette approche comparée à celles développées jusqu'à date, notamment les approches utilisant les méthodes de pénalité ou celles basées uniquement sur la cinématique du robot.

La deuxième partie de cette thèse concerne le problème de planification en-ligne de trajectoire des robots manipulateurs. Pour surpasser la complexité élevée des approches conventionnelles de commande optimale en boucles ouvertes ou fermées, nous avons proposé d'utiliser un système d'inférence neuro-floue dirigé par les données (*Data-Driven Neuro-Fuzzy Inference System*). La ligne directrice de cette

approche est jalonnée en trois étapes: D'abord, diverses techniques sont utilisées pour générer hors ligne un ensemble de trajectoires couvrant suffisamment l'espace de travail du robot. Ensuite, ces données sont groupées par une technique de groupement de données appelée ''*Subtractive Clustering*''. Ceci nous permet par la même, l'initialisation des paramètres d'un réseau neuro-flou. Dans la 3ème étape, ces paramètres sont optimisés en entraînant le réseau en question. Une fois construit et optimisé, ce réseau est utilisé dans une phase de généralisation pour la planification en-ligne de trajectoire, avec une complexité moindre. Des études de simulation et des comparaisons ont été effectuées pour les différentes techniques et les résultats sont très encourageants.

Mots clés: Robots sériels (RSs), Machines cinématiques parallèles (MCPs), Planification multi-objective de trajectoire, Redondance cinématique, Lagrangien augmenté, Découplage, Projection, Groupement des données, Réseaux neuro-flous dirigés par les données.

ABSTRACT

The multi-objective trajectory-planning problem for robotic manipulators is considered in this thesis. In the first part, the offline-planning problem is formulated in a variation calculus framework. Optimised criteria are robot travelling time and electrical and kinetic energy, and a measure of manipulability to avoid singularities. The optimisation process is done under several constraints such as actuator limitations and passing through imposed poses. The resulting constrained non-linear and non-convex optimal control problem is solved using an augmented Lagrangian with decoupling technique. This approach has been implemented on two categories of robotic manipulators; a serial redundant manipulator and a parallel manipulator. The simulations gave very good results – as compared to only kinematic based planning and approaches based on penalty methods – in time and energy minimisation and constraints satisfaction.

The second part considers the online motion planning. Because of the limitations of conventional techniques, a data-driven neuro-fuzzy approach is developed. In a prior pre-processing step, a multi-objective trajectory planning is performed using an offline methodology to generate as many as needed trajectories covering the workspace and satisfying constraints related to robot kinematics and dynamics, task and workspace. The obtained dataset is partitioned using a subtractive clustering algorithm, initializing by the same token, for a data-driven neuro-fuzzy system parameters. Then, the neuro-fuzzy system is trained and optimized to capture

the dynamic multi-objective behaviour of the robot. This system allows online motion planning with lower time consumption in a generalization phase. Simulations on a 3 DOF planar serial redundant manipulator show the robustness and high generalization capabilities of the proposed system. These results show also its advantages – for time-energy minimization, redundancy resolution, and singularity avoidance – as compared to other approaches.

Keywords: Serial Robots (SRs), Parallel Kinematic Machines (PKMs), Multi-Objective Trajectory Planning, Augmented Lagrangian, Decoupling, Kinematic Redundancy Resolution, Subtractive Clustering, Data-Driven Neuro-Fuzzy Inference Systems.

TABLE DES MATIÈRES

DÉDICACE.....iv

REMERCIEMENTS.....v

RÉSUMÉ.....vi

ABSTRACT.....viii

TABLE DES MATIÈRES.....x

LISTE DES SYMBOLES ET DES NOTATIONS.....xvi

LISTE DES ABRÉVIATIONS.....xx

LISTE DES FIGURES.....xxii

LISTE DES TABLEAUX.....xxviii

LISTE DES ANNEXES.....xxix

CHAPITRE 1. INTRODUCTION 1

1. 1. Généralités.....1

1. 2. Position du problème.....3

1. 3. Motivations 4

1. 4. Autres impacts de la recherche proposée 4

CHAPITRE 2. REVUE DE LITTÉRATURE 6

2. 1. Planification hors ligne de trajectoires des manipulateurs robotiques.....6

2. 2. Planification en-ligne de trajectoires des manipulateurs robotiques.....12

CHAPITRE 3. ORGANISATION GÉNÉRALE DE LA THÈSE.....15

3. 1. Objectif général 15

3. 2. Objectifs spécifiques 15

3. 3. Recherche proposée.....16

3. 4. Planification hors ligne de trajectoire.....17

3. 4. 1. Formulation du problème. 17

A. Modèle du robot.....17

B. Modèle des contraintes.....19

C. Indice de performance.....22

3. 4. 2. Approche proposée : Lagrangien augmenté avec découplage.....25

A. Lagrangien augmenté.....25

B. Lagrangien augmenté avec découplage.....29

3. 5. Planification en-ligne de trajectoire 34

3. 5. 1. Limitations des approches conventionnelles.....34

3. 5. 2. Approche proposée : Réseaux neuro-flous dirigés par les données.....36

3. 6. Présentation des articles.....41

3. 7. Présentation de l'article 1 : Planification multi-objectif hors ligne de trajectoire des machines cinématiques parallèles 41

3. 8. Présentation de l'article 2 : Réseaux neuro-flous hiérarchiques pour la planification temps minimum en ligne de trajectoire des robots manipulateurs.....44

3. 9. Présentation de l'article 3 : Réseaux neuro-flous dirigés par les données pour la planification multi-objectif en ligne de trajectoires des manipulateurs redondants.....47

CHAPITRE 4. CONSTRAINED MULTI-OBJECTIVE TRAJECTORY PLANNING OF PARALLEL KINEMATIC MACHINES.....49

4. 1. Abstract 49

4. 2. Nomenclature.....50

4. 3. Introduction.....53

4. 4. Modelling.....56

4. 4. 1. Kinematic Model.....56

4. 4 2. Dynamic Model.....59

4. 5. Optimal Time-Energy Trajectory Planning Problem.....65

4. 5. 1. Constraints Modelling.....65

4. 5. 2. Performance Index.....68

4. 6. Non-linear Programming Formulation.....71

4. 6. 1. Augmented Lagrangian Approach.....71

4. 6. 2. Constrained Linear Decoupled Formulation.....73

4. 6. 3. Augmented Lagrangian for the Decoupled Formulation.....76

4. 6. 4. Implementation Issuses.....78

4. 7. Simulation Case Study.....81

4. 7. 1. Description of the 2-DOF Parallel Manipulator Case Study.....81

4. 7. 2. Kinematic and Dynamic Analysis.....82

4. 7. 3. Simulation Data and Scenario.....86

4. 7. 4. Sensitivity Analysis.....92

4. 8. Conclusions and Discussions.....93

4. 9. References.....96

CHAPITRE 5. HIERARCHICAL NEURO-FUZZY OPTIMAL TIME TRAJECTORY PLANNING FOR REDUNDANT MANIPULATORS.....103

5. 1. Abstract 103

5. 2. Introduction.....104

5. 2. 1. Related Works.....104

5. 2. 2. Approach Outline.....107

5. 3. Neuro-Fuzzy Inverse Kinematics.....109

5. 4. Off-Line Minimum-Time Trajectory Generation.....111

5. 4. 1. Cycloidal Initialization 111

5. 4. 2. Robot Dynamics and Constraints Models 112

5. 5. Non-Linear Programming Formulation 115

5. 5. 1. Performance Index.....115

5 5. 2. Augmented Lagrangian (AL).....115

5. 6. Neuro-Fuzzy Optimal-Time Controller (NEFOTC).....119

5. 6. 1. Rule Base and Membership Functions.....119

5. 6. 2. NeFOTC Learning Procedure 121

5. 6. 3. NeFOTC Architecture.....123

5. 7. Simulation results.....124

5. 7. 1. Training Protocol.....124

5. 7. 2. Simulation Results.....125

5. 8. Conclusion and Discussion.....133

5. 9. Appendix A: Minimum-Time Joint Space Trajectory Parameterization.....134

5. 10. Appendix B: Robot Kinematic and Dynamic Models Including Actuators...137

5. 11. References.....140

CHAPITRE 6. DATA-DRIVEN NEURO-FUZZY MULTI-OBJECTIVE TRAJECTORY PLANNING OF REDUNDANT ROBOTS.....145

6. 1. Abstract 145

6. 2. Introduction.....146

6. 2. 1. Related works.....146

6. 2. 2. Proposed Approach 149

6. 3. Off-line Dataset Building.....151

6. 3. 1. Kinematic Modelling.....151

6. 3. 2. Dynamic Modelling.....153

6. 3. 3. Constraints Modelling.....154

6. 3. 4. Performance Index.....156

6. 3. 5. Augmented Lagrangian with Decoupling (ALD).....156

6. 4. Data-Driven Neuro-Fuzzy Multi-Objective Planning (NeFuMOP) 160

6. 4. 1. NeFuMOP Structure and Architecture 160

A. Rule Base and Membership Functions.....160

B. NeFuMOP Architecture 162

6. 4. 2. NeFuMOP Learning Procedure.....164

A. Structure Identification.....165

B. Structure Optimization.....166

6. 5. Simulation Results.....167

6. 5. 1. Training Protocol.....167

6. 5. 2. Offline Planning Dataset.....169

6. 5. 3. Dataset Clustering.....170

6. 5. 4. NeFuMOP Performance.....171

6. 6. Conclusions.....177

6. 7. References.....179

CHAPITRE 7. DISCUSSION GÉNÉRALE 183

CHAPITRE 8. RECOMMENDATIONS 188

CONCLUSIONS 191

BIBLIOGRAPHIE.....194

ANNEXES.....205

LISTE DES SYMBOLES ET DES NOTATIONS

A : Repère attaché au centre de masse de l'effecteur du robot

B : Repère attaché au centre de masse de la base du robot

A_i , B_i : $i^{ème}$ points d'attache du vérin i de A à B

$\mathbf{p} = [x \ y \ z]^T$: Vecteur position de l'origine de A par rapport à B dans B

$\dot{\mathbf{p}} = [\dot{x} \ \dot{y} \ \dot{z}]^T$: Vecteur vitesse de l'origine de A par rapport à B dans B

${}^A\boldsymbol{a}_i$: Vecteur position constant du point d'attache A_i dans A

$\boldsymbol{e}_i$: Vecteur unitaire le long du $i^{ème}$ axe articulaire

${}^B\boldsymbol{R}_A$: Matrice de rotation du repère A par rapport au repère B

$x_1 = \mathbf{q}_E = [\mathbf{p}^T \ \varphi \ \theta \ \psi]^T$: Position et orientation du repère A dans B

$x_2 = \dot{q} = [\dot{\mathbf{p}}^T \ \boldsymbol{\omega}^T]^T$: Vitesse Cartésienne de l'effecteur du robot

$x = [x_1 \ x_2]^T$: État du manipulateur en temps continue

$x_k = [x_{1k} \ x_{2k}]^T$: État du manipulateur en temps discret

$l = [l_1, l_2, ..., l_6]^T$: Vecteur des longueurs des vérins

$\boldsymbol{i} = [i_1 \ i_2 \ ... \ i_6]^T$:Vecteur des courants électriques

$\boldsymbol{\tau}(t)$: Vecteur des couples moteurs dans l'espace Cartésien

$\mathbf{J}$: Matrice Jacobienne du manipulateur

$\mathbf{J}^{\perp}$: Inverse généralisée de $\mathbf{J}$

$\mathbf{J}_{\mathrm{D}}^{\perp}$: Inverse généralisée de $\mathbf{J}$ pondérée par la matrice d'inertie

$\boldsymbol{f}_{d_k}$: Approximation discrète du model dynamique du robot

$\boldsymbol{f}_{d_k}^{D}$: Approximation discrète du model dynamique découplé du robot

$\boldsymbol{M}_j(\boldsymbol{q}_E)$: Matrice d'inertie exprimée dans l'espace articulaire

$\boldsymbol{M}_c(\boldsymbol{q}_E)$: Matrice d'inertie exprimée dans l'espace Cartésien

$\boldsymbol{N}_j(\boldsymbol{q}_E,\dot{\boldsymbol{q}})$: Forces/couples de Coriolis et centrifuge dans l'espace articulaire

$\boldsymbol{N}_c(\boldsymbol{q}_E,\dot{\boldsymbol{q}})$: Forces/couples de Coriolis et centrifuge dans l'espace Cartésien

$\boldsymbol{G}_j(\boldsymbol{q}_E)$: Force de gravité dans l'espace articulaire

$\boldsymbol{G}_c(\boldsymbol{q}_E)$: Force de gravité dans l'espace Cartésien

$\mathbf{F}_{fric}$: Force des frottements

$\boldsymbol{M}_a$: Matrice d'inertie des actionneurs

M_a : Élément de $\boldsymbol{M}_a$

$\boldsymbol{V}_a$: Matrice d'amortissement visqueux

V_a : Élément de $\boldsymbol{V}_a$

$\boldsymbol{K}_a$: Matrice des gains des actionneurs

K_a : Élément de $\boldsymbol{K}_a$

$\boldsymbol{\tau}_m$: Vecteur des couples moteurs dans l'espace articulaire

$\boldsymbol{v}$: Vecteur des variables d'accélération dans l'espace articulaire

$\mathbf{C}$: L'ensemble définissant les limites des couples moteurs

h	: Période d'échantillonnage
H	: Domaine admissible des période d'échantillonnage
$\boldsymbol{U}$	: Facteur de pondération associé au critère d'énergie électrique
$\boldsymbol{Q}$	: Facteur de pondération associé au critère d'énergie cinétique
ι_1	: Facteur de pondération du critère temporel
$\boldsymbol{\delta}$	: Facteur de pondération du critère d'évitement des singularités
$\boldsymbol{\lambda}$	: Multiplicateur de Lagrange associés à l'état du robot
L_μ	: Fonction Lagrangien augmenté (AL)
L_μ^D	: Fonction Lagrangien augmenté pour le problème découplé
E_d	: Fonction objectif en temps-discret
E_d^D	: Fonction objectif en temps-discret pour le problème découplé
ϖ	: Fonction d'évitement des singularités
Ψ	: Fonction de pénalité associée aux contraintes d'égalités
Φ	: Fonction de pénalité associée aux contraintes d'inégalités
$\boldsymbol{\sigma}$	: Multiplicateurs de Lagrange associés aux contraintes d'égalités
$\boldsymbol{\rho}$	: Multiplicateurs de Lagrange des contraintes d'inégalités
$\boldsymbol{s}$	: Fonction générique définissants les contraintes d'égalités
$\boldsymbol{g}$	: Fonction générique définissants les contraintes d'inégalités
$\boldsymbol{\mu}_s$	: Coefficient de pénalité associée aux contraintes d'égalités
$\boldsymbol{\mu}_g$	: Coefficient de pénalité associée aux contraintes d'inégalités

Σ : L'ensemble définissant les limitations des variables articulaires

$\dot{\Sigma}$: L'ensemble définissant les limitations des vitesses angulaires

N : Nombre total de discrétisations de la trajectoire

T^* : Nombre total d'itérations externe du Lagrangien augmenté

w : Tolérances faisables pour la fonction objectif

η : Tolérances faisables pour les contraintes d'égalités

η_1 : Tolérances faisables pour les contraintes d'inégalités

w^* : Tolérance optimale pour la fonction objectif

η^* : Tolérance optimale pour les contraintes d'égalités

η_1^* : Tolérance optimale pour les contraintes d'inégalités

$\boldsymbol{\pi}^k$: $k^{ème}$ entrée de l'ensemble de données d'apprentissage

$\mathbf{o}$: Sortie calculée par le réseau neuro-flou

α_j : Valeur d'activation de la $j^{ème}$ règle floue

β_j : Valeur d'activation normalisée de la $j^{ème}$ règle floue

LISTE DES ABRÉVIATIONS

MCP : Machine cinématique parallèle

PGS : Plateforme de Gough-Stewart

RS : Robot sériel

MGD : Modèle géométrique directe

MGI : Modèle géométrique inverse

MDD : Modèle dynamique directe

MDI : Modèle dynamique inverse

DDL : Degré de liberté

CAO : Conception assistée par ordinateur

POT : Planification optimale de trajectoires

POT-MCP : Planification optimale de trajectoire des machines cinématiques parallèles

POT-RS : Planification optimale de trajectoire des robots sériels

LA : Lagrangien augmenté

LAD : Lagrangien augmenté avec découplage

LADP : Lagrangien augmenté avec découplage et projection

IA : Intelligence artificielle

PKM : PARALLEL KINEMATIC MACHINES, Machines inématique parallèle

GSP : GOUGH-STEWART PLATFORM, Plate forme de Gough-Stewart

IK : INVERSE KINEMATICS, Modèle géométrique inverse

ID : INVERSE DYNAMICS, Modèle dynamique inverse

FK : FORWARD KINEMATICS, Modèle géométrique directe

FD : FORWARD DYNAMICS, Modèle géométrique inverse

DOF : DEGREE OF FREEDON, Degré de liberté

NetIK : NEURO INVERSE KINEMATICS, Inverse cinématique neuronal

AL : AUGMENTED LAGRANGIAN, Lagrangien augmenté

ALD : AUGMENTED LAGRANGIAN WITH DECOUPLING, Lagrangien augmenté avec découplage

ALDP : AUGMENTED LAGRANGIAN WITH DECOUPLING AND PROJECTION, Lagrangien augmenté avec découplage et projection

AI : ARTIFICIAL INTELLIGENCE, Intelligence artificielle

ANFIS : ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM, Système adaptatif d'inférence neuro-flou

SC : SUBTRACTIVE CLUSTERING, Partitionnement des données par soustraction

NeFIK : NEURO-FUZZY INVERSE KINEMATICS, Inverse cinématique neuro-flou

NeFOTC : NEURO-FUZZY OPTIMAL TIME CONTROL, Commande en temps minimum neuro-flou

NeFuMOP : NEURO-FUZZY MULTI-OBJECTIVE PLANNING, Planification multi-objective neuro-floue

LISTE DES FIGURES

Figure 1. 1. Conception d'un système de programmation hors-ligne pour robot manipulateur.....5

Figure 2. 1. Exemple d'un robot sériel d'assemblage, Séries M16iB de Fanuc Robotics.....14

Figure 2. 2. Exemple d'un robot parallèle, Séries F200iB de Fanuc Robotics.....14

Figure 2. 3. Exemple d'un robot parallèle, Simulator de vol Séries 500 de CAE....14

Figure 3. 1. Géométrie d'un robot sériel planaire à 3-DDL 17

Figure 3. 2. Géométrie d'une MCP.....17

Figure 3. 3. Illustration des passages de l'effecteur par des poses imposées.....25

Figure 3. 4. Architecture et fonctionnement du Lagrangien augmenté.....33

Figure 3. 5. Utilisation du modèle cinématique directe (MCD) pour l'apprentissage de NeFIK 39

Figure 3. 6. Utilisation du modèle dynamique inverse (MDI) pour l'apprentissage du réseau NeFOTC.....40

Figure 3. 7. Topologie et architecture de NeFuMOP 40

Figure 3. 8. Architecture du système proposé de planification en-ligne neuro-flou 41

Figure 4. 1. Overall Off-line Programming Framework of PKMs.....58

Figure 4. 2. Geometry of a PKM.....58

Figure 4. 3. Illustration of EE passage through imposed poses: __ optimal path, ----feasible path, x_S Starting pose, x_T Target pose.....69

Figure 4. 4. A schematic representation of a planar parallel manipulator.....88

Figure 4. 5. Initial solution, trapezoidal velocity profile.....89

Figure 4. 6. Kinematic simulation results (a)-variations of *x, y* coordinates of the EE, and sampling periods (b)-variations of torques τ_1 , τ_2 and energy.....90

Figure 4. 7. Simulation results with the augmented Lagrangian (Minimum Time – Energy) (a)-variations of *x, y* coordinates of the EE, and sampling periods (b)- variations of torques τ_1 , τ_2 and energy.....90

Figure 4. 8. Augmented Lagrangian (Minimum Energy) (a)- variations of *x, y* coordinates of the EE, and sampling periods (b)- variations of torques τ_1 , τ_2 and energy.....91

Figure 4. 9. AL Minimum Time-Energy with imposed passage through Cartesian positions (0.0, -1.4), (0.4, -1.1), and (0.5, -1.0) (a) variations of *x, y* coordinates of the EE, and sampling periods(b)- variations of torques τ_1 , τ_2 and energy.....91

Figure 4. 10. Disturbed AL (Minimum Time-Energy) (modified mass of EE, m_{EE} =30.0 kg) (a)- variations of *x, y* coordinates of the EE, and sampling periods (b)- variations of torques τ_1 , τ_2 and energy.....92

Figure 4. 11. Flowchart for Augmented Lagrangian algorithm function and operation.....95

Figure 5. 1. Overall architecture and function of the proposed system for the online time optimal trajectory planning problem.....110

Figure 5. 2. Geometry of a 3-dof planar manipulator of joint angles q_j, $j = 1,2,3$ ...113

Figure 5. 3. The use of the Forward Kinematic (FK) to the learning of the Neuro-Fuzzy Inverse Kinematic (NeFIK) system.....113

Figure 5. 4. Architecture of the AL system for the minimum-time trajectory planning problem.....118

Figure 5. 5. Learning procedure of NeFOTC 121

Figure 5. 6. Architecture of the Tsukamoto neuro-fuzzy inference system in a simplified form with one output noted z.....126

Figure 5. 7. NeFIK performance: Root mean square error output with respect to learning epochs.....130

Figure 5. 8. NeFOTC learning error (RMS Error vs. Epochs).....130

Figure 5. 9. end-effector trajectory along axis x and y ___Cycloidal profile,*-*-*-Minimum-time AL trajectory 130

Figure 5. 10. Joint variations.....130

Figure 5. 11. Joint rates $\dot{q}_i$ variations 131

Figure 5. 12. Corresponding torques τ_i and sampling period h variations.....131

Figure 5. 13. End-Effector trajectory along axis x and y 132

Figure 5. 14. Energy and sampling period h instantaneous variations for the test trajectory.....132

Figure 6. 1. Overall proposed multi-objective planning system.....152

Figure 6. 2. Geometry of a 3-dof planar manipulator 155

Figure 6. 3. Learning of Neuro-Fuzzy Multi-Objective Planning (NeFuMOP) module with $\boldsymbol{X} = (x,\ y)$, $\dot{\boldsymbol{X}} = (\dot{x},\ \dot{y})$, $\boldsymbol{\pi}_d^k = \left[\boldsymbol{\tau}_{1k}, \boldsymbol{\tau}_{2k}, \boldsymbol{\tau}_{3k}, h_k\right]^T$ 166

Figure 6. 4. NeFuMOP architecture.....168

Figure 6. 5. Displacement along x-axis and y-axis of the EE, (___desired trajectory, _._._computed trajectory).....172

Figure 6. 6. Consumed energy and time vs. sampling points.....173

Figure 6. 7. Corresponding joint torques.....173

Figure 6. 8. (x, y) coordinate variations while satisfying the passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m) 174

Figure 6. 9. Associated consumed energy and traveling time to the trajectory with imposed passages.....174

Figure 6. 10. Instantaneous variations of joint angles q_j, $j = 1,2,3$ for the circle trajectory 174

Figure 6. 11. Instantaneous variations of consumed energy and sampling time for the circle trajectory.....174

Figure 6. 12. Membership functions defining y and $\dot{y}$ 175

Figure 6. 13. NeFuMOP performance, Learning error (RMS Error vs. Epochs) ---Training data, ***testing data, ____validation (with noisy 176

Figure 6. 14. $(x,\ y)$ coordinate instantaneous variations the EE 176

Figure 6. 15. Energy and sampling period h variations for the test trajectory.....177

Figure A. 1. Function and architecture of the ALD system for the multi-objective trajectory- planning problem.....227

Figure A. 2. Geometry of a 3-DOF planar manipulator of joint angles q_j, $j=1,2,3$ 228

Figure A. 3. EE trajectory along axis x and y displacement in meters versus sampling points, (___) Cycloidal profile, (*-*-*-*) Minimum-time including robot dynamics and constraints.....228

Figure A. 4. Joint q_i variations 228

Figure A. 5. Corresponding energy and sampling period h variations 229

Figure A. 6. (x, y) coordinate variations while satisfying the passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m) 229

Figure A. 7. Associated consumed energy and traveling time to the trajectory with imposed passages.....229

Figure A. 8. Energy and sampling period h instantaneous variations for the second Trajectory.....230

Figure A. 9. Instantaneous variations of joint angles q_j, $j=1,2,3$ for the circle trajectory.....230

Figure A. 10. Instantaneous variations of consumed energy and sampling time for the circle trajectory.....230

Figure A. 11. Instantaneous variations of consumed energy and sampling time for the first example (a) Trajectory with third link mass 2 Kg (b) Trajectory with increased third link mass by 1.5 Kg.....231

Figure A. 12. EE trajectory along axis x and y displacement for example 1 with modified third link mass.....231

Figure B. 1. Geometry of a PKM.....238

Figure B. 2. A geometric representation of the planar parallel manipulator.....245

Figure B. 3. Kinematic simulation results.....247

Figure B. 4. Simulation results using the augmented Lagrangian.....247

Figure C. 1. Illustration géométrique des conditions d'optimlité de Kuhn-Tucker..253

Figure C. 2. Interprétation géométrique de la méthode du Lagrangien augmenté de Rockafellar.....273

LISTE DES TABLEAUX

Table 4. 1. Limits of workspace, actuator torques and sampling periods.....89

Table 4. 2. Convergence history of minimum time-energy planning with augmented Lagrangian.....89

Table 5. 1. Kinematic and dynamic parameters of the manipulator.....126

Table 5. 2. Limits of workspace, actuator torques, joint rates and accelerations and sampling periods.....127

Table 5. 3. Performance of NeFOTC RMS Error for 2000 trajectory entries after 10 epochs training 129

Table 5. 4. Performance of NeFOTC, with disturbed testing data from Table 2, by an amount of 5% RMS Error after 20 epochs training.....132

Table 6. 1. Kinematic and dynamic parameters of the manipulator 172

Table 6. 2. Limits of workspace, actuator torques, joint rates and accelerations and sampling periods.....172

Table 6. 3. Performance of NeFuMOP.....175

Table 6. 4. Performance of NeFuMOP (with disturbed testing data).....176

Table A. 1. Kinematic and dynamic parameters of the manipulator 225

Table A. 2. Limits of workspace, actuator torques, joint rates and accelerations and sampling periods 225

LISTE DES ANNEXES

ANNEXE A. MULTI-OBJECTIVE TRAJECTORY PLANNING FOR REDUNDANT MANIPULATORS USING AUGMENTED LAGARANGIAN205

A. 1. Abstract.....205

A. 2. Résumé.....206

A. 3. Introduction.....207

A. 4. Modeling.....209

A. 4. 1. Kinematic Modeling.....209

A. 4. 2. Dynamic Modelling.....211

A. 4. 3. Constraints Modeling.....212

A. 5. Non-Linear Programming Formulation.....215

A. 5. 1. Augmented Lagrangian (AL)..... 216

A. 5. 2. Augmented Lagrangian with Decoupling (ALD).....218

A. 6. Simulation Results.....222

A. 7. Conclusion.....225

A. 8. References.....232

ANNEXE B. A DECOUPLED APPROACH TO OPTIMAL TIME ENERGY TRAJECTORY PLANNING OF PARALLEL KINEMATIC MACHINES...234

B. 1. Abstract.....234

B. 2. Introduction.....235

B. 3. Kinematic and Dynamic Models.....237

B. 4. Discrete-Time Operation Space Dynamic Model.....237

B. 5. Optimal Time-Energy Trajectory Planning Problem.....238

B. 6. Constrained Linear-Decoupled Formulation.....240

B. 7. Augmented Lagrangian for the Decoupled Formulation.....242

B. 8. Simulation Results.....244

B. 9. Conclusions.....246

B. 10. Bibliography.....247

ANNEXE C. RAPPELS SUR L'OPTIMISATION NON LINÉAIRE AVEC CONTRAINTES.....250

C. 1. Conditions nécessaires d'optimalité.....250

C. 1. 1. Qualification des contraintes 252

C. 1. 2. Conditions nécessaires d'optimalité (*Kuhn et Tucker 1951*).....252

C. 2. Conditions suffisantes d'optimalité (points-cols et fonction de Lagrange)....254

C. 3. Méthodes directes (primales) d'optimisation avec contraintes.....256

C. 4. Méthodes indirectes (ou duales).....256

C. 4. 1. Méthodes de pénalités.....256

C. 4. 2. Méthodes duales utilisant le Lagrangien ordinaire.....260

C. 4. 3. Méthodes duales utilisant le Lagrangien augmenté.....263

C. 5. Références 278

CHAPITRE 1

INTRODUCTION

1. 1. Généralités

Dans le contexte de concurrence et de compétition mondiales actuelles, et d'exigences de plus en plus croissantes en performance, flexibilité, agilité et réduction des coûts, les outils de planification de plus en plus rapides et précis deviennent prépondérants pour le succès des entreprises [1, 2]. La simulation sur ordinateur constitue un outil puissant et performant pour réduire les coûts en temps, énergie, matière première, et améliorer les conditions de sécurité de la machine et de l'opérateur humain [3]. En réduisant l'utilisation de prototypes physiques, l'assistance par ordinateur a permis une avancée considérable de la robotique, que ce soit au niveau conceptuel, ou bien au niveau opérationnel et exécution propre de la tâche. En effet, durant les deux dernières décennies, la robotique a connu des développements extraordinaires sur les plans matériel et logiciel. Les technologies logicielles de simulation et de programmation hors ligne, notamment les systèmes de commande et de conception assistée par ordinateur (CAO) ont été l'avant garde des progrès rapides des réalisations actuelles.

Un système de programmation des robots doit tenir compte de trois éléments principaux, à savoir, les modèles cinématiques et dynamiques du robot, ainsi que le modèle de l'espace de travail et les contraintes de la tâche. La Figure 1.1 montre un

diagramme conceptuel d'un système de programmation hors-ligne des manipulateurs robotiques. La modélisation cinématique traite des équations décrivant les variations des positions, vitesses et accélérations du robot, sans prendre compte les forces et les couples qui causent le mouvement. La modélisation dynamique permet de considérer en plus des équations cinématiques, les forces et couples qui sont à l'origine de ces mouvements. L'espace de travail est le domaine ou l'ensemble des poses accessibles par l'effecteur du robot sans singularités ou collisions avec les obstacles. Une tâche peut être définie par une pose de départ, une pose d'arrivée, et des poses intermédiaires [4-11]. Un problème important de programmation des robots est la planification optimale de trajectoire (POT). Ce problème peut être défini suivant différents niveaux de complexité. Le premier niveau est celui de planification de chemin. Il consiste à déterminer le chemin le plus court permettant le déplacement de l'effecteur d'une pose initiale à une pose finale. Ce problème a été classiquement identifié par le *problème du déménageur de piano*, et a reçu un grand intérêt des communautés robotiques et géométrie algorithmique [12, 13]. Le deuxième niveau inclue en plus, les informations sur la vitesse et l'accélération du mouvement tout en évitant les obstacles et interférences entre les segments du robot. Un troisième niveau de complexité est quand l'optimisation de la trajectoire incluse les forces et énergies nécessaires à son exécution. D'autre part, la POT peut être effectuée par deux principales façons; hors-ligne ou en-ligne. La planification en-ligne est appliquée quand le robot est en opération et requière en conséquence des algorithmes de faible complexité en temps de calcul [14-17]. La planification hors-ligne est réalisée à partir

de modèles du robot, de la tâche et de l'espace de travail, et ne nécessite pas d'équipement physique autre qu'un ordinateur. Dans cette thèse, on s'intéresse au développement de stratégies algorithmiques efficaces pour la planification multi-objectif hors-ligne et en-ligne de trajectoires des manipulateurs robotiques.

1. 2. Position du problème

Dans cette recherche, deux sous problèmes de POT sont étudiés. Le premier est celui de planification hors-ligne multi-objectif, avec deux applications, une sur les robots sériels (RSs) et l'autre sur les robots parallèles, (ou machines cinématiques parallèles, MCPs). Cette approche est basée sur le modèle dynamique incluant ceux des actionneurs du robot ainsi que les modèles de l'espace de travail et contraintes de la tâche [18-21]. Les critères d'optimisation sont le temps et l'énergie nécessaires à la réalisation de la tâche ainsi qu'un *indice de manipulabilité* pour éviter les singularités. Les techniques d'optimisation sous contraintes sont explorées pour résoudre le problème résultant de commande optimale non linéaire sous contraintes.

Le deuxième sous-problème est celui de la planification en-ligne de trajectoire. À cause de la grande complexité des modèles des robots, qui limite toute utilisation possible des techniques conventionnelles de commande optimale, plusieurs chercheurs en robotique ont introduit les techniques d'intelligence artificielle (IA) [22-25]. Dans ce cadre, nous avons conçu et réalisé une approche basée sur les systèmes neuro-flous dirigés par les données, de complexité moindre, en vue d'une mise en œuvre temps réel.

1. 3. Motivations

La motivation principale de cette recherche est la proposition d'un cadre unifié de résolution des problèmes de planification multi objectif hors-ligne et en-ligne de trajectoires des manipulateurs robotiques, pouvant inclure diverses sous-problèmes, tel que les modèles cinématiques et dynamiques du robot, des actionneurs, de l'espace de travail et des spécifications de la tâche. En deuxième lieu, concevoir des algorithmes adéquats et efficaces, ensuite réaliser des simulations ses techniques proposées sur des cas de manipulateurs parallèles et sériels.

1. 4. Autres impacts de la recherche proposée

La POT trouve des applications dans des domaines aussi large que les procédés de fabrication, intervention en milieu hostiles, exploration spatiale et sous marine. Les procédures de planification et d'optimisation de trajectoire vont sans doute contribuer à l'amélioration et augmenter les performances de ces systèmes robotiques. La planification optimale de trajectoire est depuis plus de trois décennies un axe dynamique de recherches dans le domaine le plus général de la robotique. De la robotique mobile, tel que les robots de type véhicule, robots marcheurs ou humanoïdes, à la robotique sérielle ou parallèle. Il va sans dire que la POT couvre bien des champs d'application aussi vastes que les véhicules automobiles, les domaines militaires, les transports aériens, maritimes, etc,...

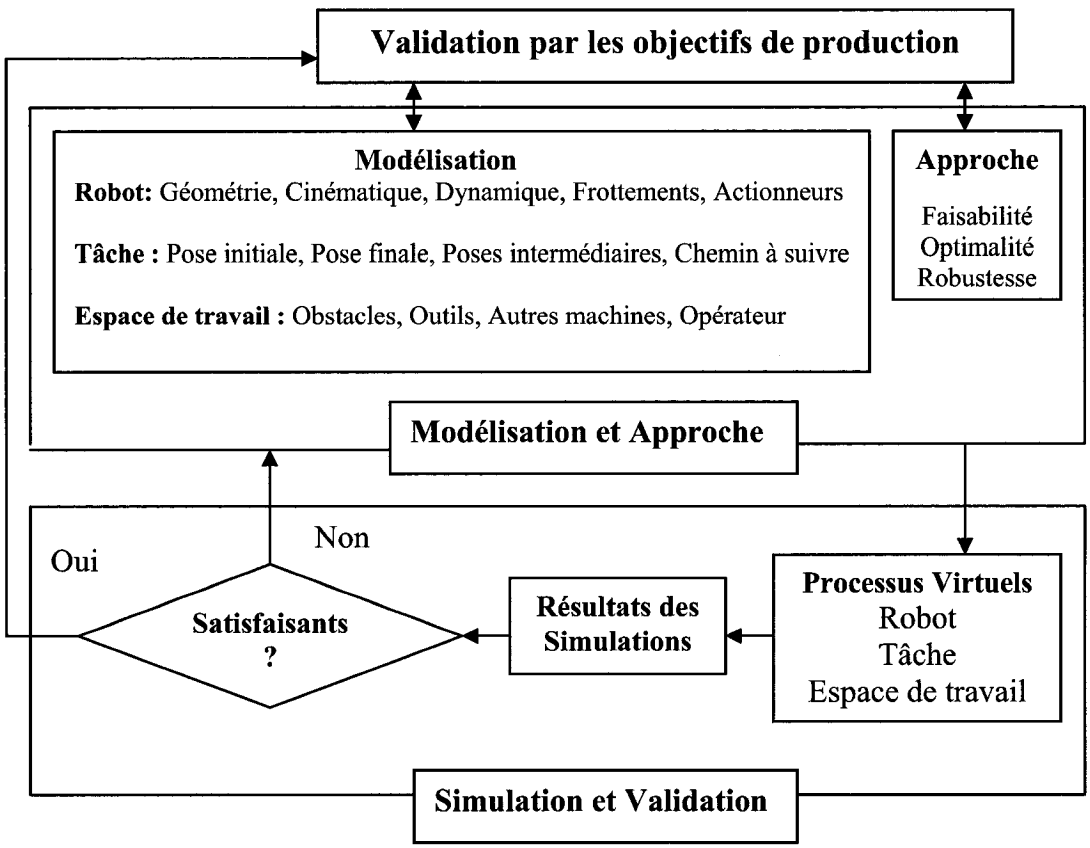


Fig. 1. 1. Conception d'un système de programmation hors-ligne pour robots manipulateus

CHAPITRE 2

REVUE DE LITTÉRATURE

Le problème de planification de mouvement a reçu un grand intérêt de la part de la communauté scientifique en général. En robotique, un nombre croissant des recherches a été enregistré dans les domaines de génération et d'optimisation de trajectoires des robots dans les deux modes de programmation hors-ligne et en-ligne.

2. 1. Planification hors-ligne de trajectoires des manipulateurs robotiques

On distingue deux écoles principales de POT. La première est celle de la CAO et géométrie algorithmique. Elle consiste à assimiler le robot, l'espace de travail et la tâche à une application Windows. On utilise alors des logiciels dédiés de CAO/CFAO, de graphisme, visualisation, et animation, ainsi que des utilitaires de modélisation et simulation pour obtenir des trajectoires de grandes précision de positionnement de l'effecteur dans l'espace tridimensionnel. Parmi les logiciels les plus répondus de cette catégorie, on cite CATIA Robotique, IGRID, RobotMaster de Mastercam, ROBOGUIDE de Fanuc Robotics, et ROBCAD [26-30]. Les résultats ainsi trouvés peuvent être améliorés par des commandes optimisant les forces, les énergies, et le temps d'exécution de la tâche, ou tout autre critère ayant un sens physique et contribuant à une meilleure performance du robot.

La deuxième école de pensée du problème de POT, qui est complémentaire à la première est celle de la commande optimale [14-17]. Notre projet de recherches s'inscrit dans le cadre de cette école. Dans ce cas, le problème consiste à déterminer la séquence des commandes optimales au sens d'un ou de plusieurs critères de performances permettant de déplacer l'effecteur du robot d'une pose initiale à une pose finale. Dans la littérature, le critère temps minimum a reçu un fort engouement de la communauté robotique. Ceci est justifié pour la raison évidente d'augmentation de la vitesse de production [31-39]. Globalement, il existe deux approches pour la POT basées sur la commande optimale. La première décompose le problème en deux sous-problèmes. D'abord, la trajectoire est calculée selon une forme paramétrée, telle qu'une forme de *Bésier, B-Spline, Cycloïdal* ou polynomiale. Ensuite, on résout le problème de commande en temps-minimum qui devient alors indépendant du nombre de degrés de liberté (DDL) du robot [31-35]. La deuxième approche opère en une seule étape et n'effectue pas de paramétrisation *a priori* de la trajectoire, mais plutôt calcule directement une trajectoire avec évitement des collisions, tout en optimisant un indice de performance. Parmi les travaux appartenant à cette catégorie on mentionne les références [14-15, 37, 38, 39]. Dans [8, 38], on démontre que la structure de la commande en temps-minimum sans contraintes est soit *bang-bang* soit *bang - arc singulier - bang.* Pour la commande en temps-minimum sous contraintes, il a été montré dans [34, 35] que la commande optimale est aussi de type *bang – arc singulier - bang.* Dans [39], il a été rapporté que toutes les trajectoires en temps-

minimum doivent être saturées au moins pour un actionneur, en tout temps. Cependant, l'inconvénient majeur de ces approches de commande en temps-minimum est la commutation permanente entre les valeurs extrêmes des couples moteurs. Une telle commande, de type tout ou rien, est de nature à accélérer la fatigue du mécanisme robotique. Un deuxième inconvénient est que la poursuite d'une trajectoire planifiée en temps minimum implique des vibrations importantes au nivaux des joints articulaires et des saturations permanentes des couples actionneurs [39]. Parmi les remèdes proposés à ce problème, l'approximation de la trajectoire par des segments linéaires, avec des choix appropriés des points de commutation [32]. Dans [15], les auteurs proposent d'inclure la dynamique du robot dans le processus de génération de trajectoires en utilisant une fonctionnelle coût pondérant le temps de cycle et l'énergie pour obtenir des trajectoires plus régulières comparées à celles obtenues en temps-minimum. Une autre approche de génération des trajectoires régulières a été proposée dans [5-8]. Cette dernière ne nécessite pas des variations élevées des couples moteurs aux points de commutation et permet une planification multi-objectif sous contraintes d'évitement de saturation des actionneurs, des butées articulaires, des poses singulières, des limitations de l'espace de travail, et d'autres contraintes spécifiques à la tâche. L'algorithme du Lagrangien augmenté a été utilisé dans cette approche [40, 41].

La revue de littérature rapportée ci-haut concerne notamment les robots sériels (ou RSs). Pour les robots à structure cinématique parallèle (ou MCPs), le problème de

POT est beaucoup plus difficile et les travaux publiés sont à un état beaucoup moins avancés, comparativement à leurs prédécesseurs sériels. Plusieurs raisons de ces difficultés ont été dénombrées. Une première raison réside dans les différences majeures entre ces deux types de machines [18, 42]. Ces différences sont d'abord au niveau de la structure géométrique. Pour les RSs, l'effecteur est connecté à la base par une seule chaîne mécanique articulée composée de segments actionnés par des actionneurs montés sur chacun des segments. Pour les MCPs, l'effecteur est connecté à la base par un ensemble de chaînes cinématiques articulées parallèles. Outre leur structure, l'autre raison de cette difficulté est le domaine d'application et d'utilisation de chacune de ces deux catégories de robots. Alors que les RSs sont généralement utilisés dans des applications tel que l'assemblage, la peinture et le soudage, les MCPs sont plutôt utilisés dans des applications qui nécessitent une grande précision de positionnement dans des espaces de travail très restreints. Les Figures 2.1 et 2.2 montrent des exemples de manipulateurs industriels sériel et parallèle. La Figure 2.3 montre un MCP de type simulateur de vol développé par CAE [43, 44]. Une troisième difficulté majeure du problème de POT-MCPs est due à la complexité de la transformation affine *"mappings"* ou entre l'espace articulaire actionné et l'espace Cartésien d'exécution propre de la tâche. L'évitement des poses singulières est une autre question fondamentale qui caractérise ces transformations pour les MCPs, car tout en étant la transformatrice de mouvements depuis l'espace articulaire vers l'espace Cartésien, la matrice Jacobienne est aussi la transformatrice des forces, qui

causent ces mouvements. À une pose singulière, cette matrice est mal conditionnée, avec un déterminant soit égale à zéro ou à l'infini ou encore à une valeur proche de ces valeurs limites. Physiquement, ceci correspond soit à des poses inatteignables à cause des limitations des déplacements articulaires, soit à des poses dans lesquelles un ou plusieurs segment du robot est bloqué [43-48]. Dans ces conditions, le robot devient incontrôlable (on dit aussi *irrésolvable*), et l'application de n'importe quel couple moteur ne sera à même d'emmener l'effecteur à la pose désirée. En outre, les MCPs possèdent deux types de singularités supplémentaires par rapport aux RSs. Tandis que les RSs ont des singularités appelées singularités aux bords, ou singularités aux frontières, les MCPs ont deux autres types de singularités issues de leurs structures géométriques en chaînes cinématiques parallèles [42-45]. Ce problème a donné ces deux dernières décennies à d'importants travaux de recherche en conception, en personnalisation, en spécialisation ainsi qu'en spécification [42-52]. Les performances dépendantes de l'architecture du robot, la forte non-linéarité et le couplage de sa dynamique, ainsi que la difficile question des singularités rendent le problème POT-MCP plus difficile comparativement à POT-RS. En effet, la plupart des critères rapportés jusqu'à date pour la POT-MCP sont essentiellement liés aux performances cinématiques, au bon conditionnement numérique de la Jacobienne, et au design de la structure du robot. Parmi ces critères, on distingue la *mesure de manipulabilité,* le *nombre condition* ou encore *l'indice de dextérité* du manipulateur [42-45]. Parmi les travaux publiés sur la POT-MCP, on mentionne l'approche par

groupement [19] *"Clustering"* pour isoler et éviter les poses singulières et les obstacles. Dans [20], les auteurs ont développé une technique d'évitement des singularités pour une plate forme de Gough-Stewart (PGS) par reconstruction d'une trajectoire pré-planifiée dans le voisinage d'une singularité. Dans [21], un algorithme de planification de trajectoire bien conditionnée entre deux points a été développé pour les PGSs. Dans [49], il a été montré qu'un changement de pose sans singularités est possible pour les MCPs. Un élément clé lié aux hautes performances des MCPs est sans doute l'inclusion de leurs dynamiques. Une approche variationnelle de POT-MCP entre deux points de l'espace de travail a été considérée dans [48]. Cette approche minimise un critère énergétique, tout en évitant les singularités. Elle est basée sur les méthodes de pénalité. Ces dernières ont cependant des inconvénients majeurs [40, 41]. Une autre question principale identifiée récemment pour le problème de POT-MCP, qui est de grande utilité industrielle, (dans un processus d'usinage, par exemple) est que pour une trajectoire pré-spécifiée de l'outil, le système de commande doit garantir l'exécution de la tâche dans l'espace de travail et cela pour une disposition donné de la plate forme (i.e., tel que la non violation des limitations physiques du robot). Ce problème a été considéré dans [51, 52], où des méthodologies de design incluant les limitations de l'espace de travail, des couples moteurs ont été introduites en utilisant des algorithmes itératifs d'optimisation. Dans ce contexte, la première partie de la présente thèse constitue une extension de ces travaux.

2. 2. Planification en-ligne de trajectoires des manipulateurs robotiques

Pour le problème de planification en-ligne de trajectoire, l'inconvénient majeur des approches conventionnelles est leurs grandes gourmandises en de temps de calcul pour obtenir une solution. Comme les tâches robotiques sont trop exigeantes en précision et en diversité, l'exécution de ces tâches peut être très difficile en raison des couplages et des non linéarités fortes des paramètres du modèle dynamique tel que les forces de Coriolis et centrifuges, les paramètres inertiels ainsi que les forces de frottements et gravitationnelles. L'implémentation d'une commande en temps-minimum pure avec des limitations sur les couples peut donner des résultats non satisfaisants. Par exemple, la poursuite directe d'une commande en temps-minimum implique des vibrations au niveau articulaire et dépassements des couples nominaux [16]. Pour les RSs, une commande en vitesse définissant une tâche secondaire a été utilisée dans [17], pour modifier en-ligne un profil de vitesse nominale en temps-minimum, tout en évitant la saturation des actionneurs. Toutefois, ces approches sont de nature en boucle ouverte, et ne sont pas nécessairement des bons choix pour les manipulateurs robotiques plutôt contrôlés en boucle fermée.

Plusieurs types de commande en boucles fermées ont été proposés pour les manipulateurs robotiques. Les commandes linéaires par retour d'état à gain fixe ont été largement implémentées. Cependant, il a été démontré que ce type de commande ne permet pas d'obtenir de bonnes performances pour des variations importantes de la charge, et des tâches à grandes vitesses et précisions. Des commandes plus

élaborées ont été développées récemment. Néanmoins, ces commandes font appel aux modèles détaillés de la dynamique et cinématique du robot. Évidement, ceci engendre une complexité considérable en temps de calcul [53-55].

Pour surpasser ces difficultés, plusieurs chercheurs en robotique ont utilisé les techniques d'intelligence artificielle (IA) qu'on appelle communément '*Soft Computing*' à savoir, les algorithmes génétiques, les réseaux de neurones et les systèmes flous. Les algorithmes génétiques sont en application croissante dans les domaines de planification de mouvement et des systèmes de commandes dynamiques des manipulateurs robotiques [9, 56-57]. Les réseaux de neurones et les systèmes flous constituent probablement les techniques les plus appliquées de l'intelligence artificielle à la POT des manipulateurs robotiques [22-25]. En outre, ces techniques ont apporté une valeur ajoutée à plusieurs domaines de l'ingénierie, depuis l'usinage, les télécommunications, la reconnaissance de la parole à la biomécanique [58-61].

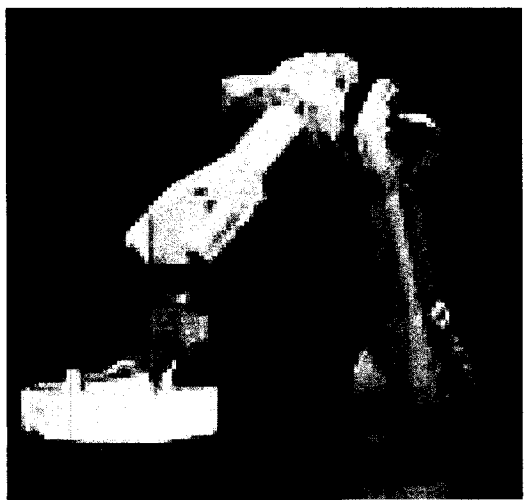


Fig. 2. 1. Exemple d'un robot sériel
Séries M16iB de Fanuc Robotic

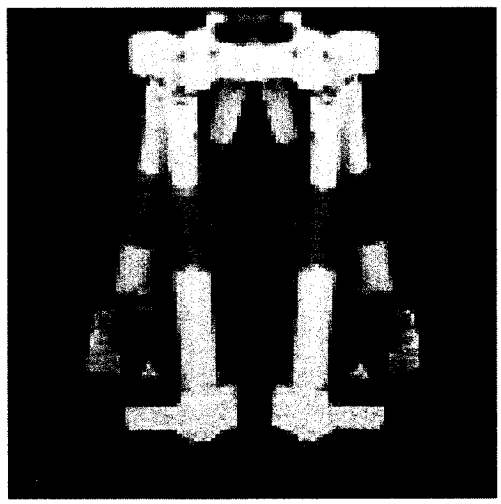


Fig. 2. 2. Exemple d'un robot parallèle
Séries F200iB de Fanuc Robotics
Courtoisie Fanuc obotics (http://www.fanucrobotics.com)

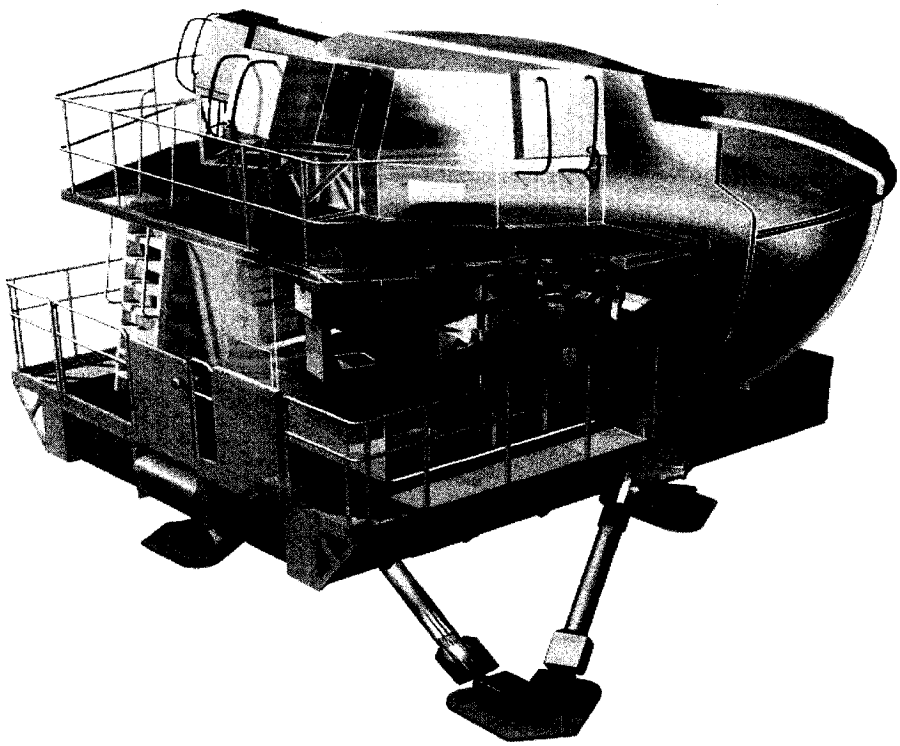


Fig. 2. 3. Exemple d'un robot parallèle, Simulator de vol Séries 500 de CAE
Courtoisie CAE (http://www.cae.com/www2004/Photo_Gallery/civil.shtml)

CHAPITRE 3

ORGANISATION GÉNÉRALE DE LA THÈSE

3. 1. Objectif général:

L'objectif principal de la présente recherche est le développement de systèmes et stratégies algorithmiques avancés de simulation et de planification multi-objectif de trajectoires des manipulateurs robotiques. Ces systèmes incluent les modèles cinématique et dynamique du robot, ainsi que les limitations de l'espace de travail et les contraintes de la tâche. Ceci permet d'améliorer les performances des systèmes robotiques en optimisant l'interaction du triplet : robot, tâche, et espace de travail. En effet, une programmation directe du robot physique peut s'avérer fastidieuse et coûteuse. Une manière d'économiser le temps et l'énergie est de simuler sur ordinateur les trajectoires du robot pour implémenter plus tard les meilleurs résultats sur le robot physique.

3. 2. Objectifs spécifiques:

Plus spécifiquement, on s'intéresse dans cette thèse au développement des stratégies algorithmiques appropriées pour chacun de ces sous-problèmes de POT.

Objectifs 1: Formulation et résolution du sous-problème de planification hors-ligne multi-objectif de trajectoire des robots, considérant les contraintes d'évitement des poses singulières, des limites des actionneurs, de l'espace de travail, et de passage par des poses imposées.

Objectifs 2: Utilisation du système de planification développé lors de l'objectif 1 pour réaliser un planificateur en-ligne multi-objectif de trajectoire des robots, avec une complexité raisonnable.

3. 3. Recherche proposée

D'abord, le problème de planification hors-ligne est introduit pour un système robotique de type général. Les chapitres 4 et 6 considèrent l'application de cette approche aux MCPs et aux RSs. Les Figure 3. 1 et 3. 2 montrent des schémas géométriques de manipulateurs de type RS et MCP. La deuxième partie de la recherche proposée traite du problème de planification en-ligne de trajectoires des manipulateurs robotiques redondants. Le choix des manipulateurs redondants est particulièrement difficile considérant les contraintes relatives aux effets dynamiques et exigences de la tâche, simultanément à celles de la redondance. Nous avons abordé ce problème par une technique de planification en-ligne basée sur les réseaux neuro-flous dirigés par les données. Les chapitres 5 et 7 développent l'application de cette approche aux RSs redondants.

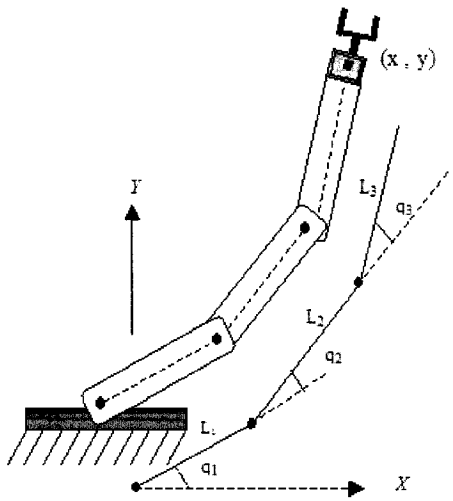


Fig. 3. 1. Géométrie d'un robot sériel planaire à 3 DDL (angles q_j, $j = 1,2,3$)

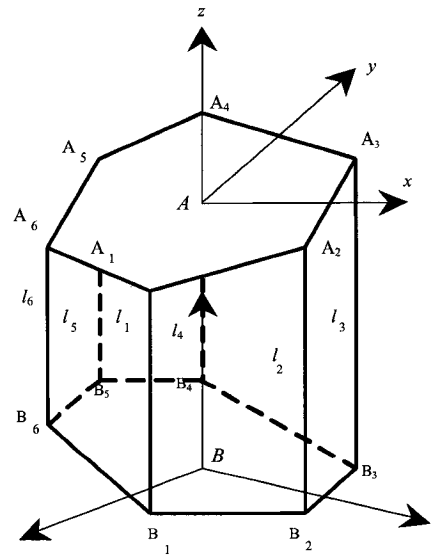


Fig. 3. 2. Géométrie d'une MCP

3. 4. Planification hors-ligne de trajectoire

3. 4. 1. Formulation du problème

A. Modèle du robot

Dans cette section, on considère le problème de planification multi-objectif de trajectoires pour un manipulateur robotique de type général. Ce problème peut être énoncé comme suit.

Étant donné le modèle dynamique incluant les paramètres inertiels et couples de Coriolis et centrifuges de l'effecteur du robot, des chaînes cinématiques, ainsi que le modèle des actionneurs [42, 46, 50]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \qquad (1)$$

Où $\boldsymbol{M}(\boldsymbol{q})$ décrit la matrice d'inertie, $\boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}})$ exprime les forces et couples de

Coriolis et centrifuge $\boldsymbol{q}$ désigne les positions articulaires du robot, $\dot{\boldsymbol{q}}$ les vitesses angulaires, $\boldsymbol{\tau}$ est le vecteur des couples moteurs.

Cette équation est d'abord mise sous forme discrète par l'approximation [16, 17]:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & h_k \boldsymbol{I}_{n\times n} \\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n} \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{M}^{-1}(\boldsymbol{x}_{1k}) \begin{bmatrix} \frac{h_k^2}{2} \boldsymbol{I}_{n\times n} \\ h_k \boldsymbol{I}_{n\times n} \end{bmatrix} \left[\boldsymbol{\tau}_k - \left[\boldsymbol{N}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \boldsymbol{G}(\boldsymbol{x}_{1k}) \right] \right] \quad (2)$$

Où $\boldsymbol{I}_{n\times n}$ désigne la matrice identité de dimension nxn, $\boldsymbol{O}_{n\times n}$ la matrice nulle de dimension nxn, h_k le pas d'échantillonnage, et $\boldsymbol{x}_k = \left[\boldsymbol{x}_{1k} \quad \boldsymbol{x}_{2k} \right]^T$ l'approximation discrète de l'état du robot. Pour une utilisation plus simple, on note le second membre de l'éq. (2) par $\boldsymbol{f}_{d_k}$ et on écrit l'eq. (2) sous la forme:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \quad (3)$$

Le problème consiste donc à générer une trajectoire pour tous les composants du manipulateur partant d'une pose initiale et arrivant à une pose finale spécifiée, tout en minimisant un indice de performance et satisfaisant diverses contraintes. Certaines de ces contraintes sont définies dans l'espace Cartésien, alors que d'autres sont définies dans l'espace articulaire.

B. Modèle des contraintes:

Contraintes sur le robot :

- Limitations sur les positions et vitesses articulaires:

Ces contraintes sont liées aux limitations des variables articulaires, qui sont souvent fournies par le fabricant. Elles sont exprimées comme suit :

$$\Sigma = \left\{ \boldsymbol{q}_k \in \Re^{nN} : \quad \boldsymbol{q}_{i\min} \le \boldsymbol{q}_{ik} \le \boldsymbol{q}_{i\max}, \quad i = 1,...,n, \quad k = 0,...,N \right\} \tag{4}$$

$$\dot{\Sigma} = \left\{ \dot{\boldsymbol{q}}_k \in \Re^{nN} : \quad \dot{\boldsymbol{q}}_{i\min} \le \dot{\boldsymbol{q}}_{ik} \le \dot{\boldsymbol{q}}_{i\max}, \quad i = 1,...,n, \quad k = 0,...,N \right\} \tag{5}$$

Dans les ensembles (4) et (5) ci-dessus, N désigne le nombre d'échantillonnage de la trajectoire et $\boldsymbol{q}_{\max}$ et $\boldsymbol{q}_{\min}$ sont respectivement les valeurs maximales et minimales des limitations des positions angulaires.

- Limitations des couples actionneurs:

La non violation des valeurs limites des couples moteurs est un autre élément principal à prendre en compte dans la planification de mouvement d'un robot. Les valeurs des forces exigées au niveau articulaire doivent être vérifiées constamment afin d'éviter les poses singulières, ou tout éventuel dépassement des couples nominaux. Pour ce faire, la procédure de planification de trajectoire doit être capable de déterminer de nouvelles valeurs d'actuation, aussitôt que celles-ci s'approchent des limites. Dans notre approche, on suppose que les couples actionneurs ont des valeurs nominales (fournies par le fabricant). Analytiquement, ceci est exprimé par un

ensemble admissible $\mathbf{C} \subset \Re^{nN}$ des couples moteurs donnés par:

$$\mathbf{C} = \left\{ \boldsymbol{\tau}_k \in \Re^{nN} : \quad \boldsymbol{\tau}^i_{\max} \leq \boldsymbol{\tau}^i_k \leq \boldsymbol{\tau}^i_{\max}, \quad i = 1,...,n, \quad k = 0,...,N-1 \right\} \tag{6}$$

- Limitations sur les périodes d'échantillonnage:

Les périodes d'échantillonnage sont définies tel que le temps total d'exécution de la tâche soit exprimé par $T = \sum_{k=0}^{N-1} h_k$, où h_k est le temps séparant deux poses discrètes successives k et $k+1$, $k = 0,...,N-1$. Si on assigne un temps total très petit pour un déplacement de l'effecteur du robot d'un point de départ à un point d'arrivée, il se peut qu'on ne trouve pas une commande admissible solution du POT, car les contraintes sur les couples bornent indirectement le temps nécessaire à une trajectoire [5-7].

En outre, les variables h_k doivent être plus petites que la constante de temps mécanique du système afin d'éviter qu'il soit incontrôlable entre deux instants de commande. Si on dispose des limites maximale et minimale $h_{\min}$ et $h_{\max}$ des périodes d'échantillonnage, on prendra alors les périodes d'échantillonnage comme des variables de commande qu'on supposera définies dans un domaine admissible H par :

$$\mathrm{H} = \left\{ h_k \in \Re^+ : \quad h_{\min} \leq h_k \leq h_{\max} \right\} \tag{7}$$

Contraintes de la tâche et de l'espace de travail:

Parmi ces contraintes, on distingue:

- Contraintes d'évitement d'obstacles

Ce problème constitue en lui-même un domaine de recherche à part entière. Plusieurs études ont été développées pour le résoudre dans le cadre des robots manipulateur ou mobiles. Dans cette thèse, cet aspect du problème de POT est réduit à celui d'imposer à l'effecteur du robot de passer à travers un ensemble de poses imposées. Ceci est exprimé par les contraintes suivantes :

- Contraintes de passage par des poses imposées :

Dans un système robotique industriel, l'utilisateur ou le planificateur de tâche fournie généralement une série de poses dans l'espace cartésien ou articulaire à travers lesquelles le manipulateur doit passer. Ces contraintes sont quantifiées par un ensemble de L poses prédéfinies ($\mathbf{p}_l$, $\boldsymbol{R}_l$) par lesquelles l'effecteur du robot doit passer, où $\mathbf{p}_l$ réfère à la $l^{\text{ième}}$ position Cartésienne imposée de l'effecteur, et $\boldsymbol{R}_l$ à l'orientation correspondante (Fig. 3.3). Celles-ci peuvent inclure les poses initiales et finales. Elles sont représentées par les contraintes égalités suivantes:

$$s_1^l(x) = T_{\text{PassTh}P_l} - \left\| \mathbf{p} - \mathbf{p}_l \right\| = 0 \tag{8}$$

$$s_2^l(x) = eps_l - \left\| \mathbf{vect}(\boldsymbol{R}^T \boldsymbol{R}_l) \right\| = 0 \quad l = 1,...,L \tag{9}$$

où $(\mathbf{p}, \boldsymbol{R})$ décrit la pose calculée de l'effecteur, **vect**(.) le vecteur axial de la matrice 3×3, $T_{\text{PassTh}P_l}$ mesure la tolérance de passage entre $\mathbf{p}$ et $\mathbf{p}_l$, et eps_l mesure la tolérance

de passage entre les orientations calculées et imposées $\boldsymbol{R}$ et $\boldsymbol{R}_l$, respectivement. Les contraintes inégalités mentionnées ci-dessus sont notées sous les formes suivantes :

$$\mathbf{g}_1(\boldsymbol{x}) = \boldsymbol{x}_{\min} - \boldsymbol{\Theta}(\boldsymbol{x}), \mathbf{g}_2(\boldsymbol{x}) = \boldsymbol{\Theta}(\boldsymbol{x}) - \boldsymbol{x}_{\max}, \; \boldsymbol{g}_3(\boldsymbol{\tau}) = \boldsymbol{\tau}_{\min} - \boldsymbol{\tau}, \; \boldsymbol{g}_4(\boldsymbol{\tau}) = \boldsymbol{\tau} - \boldsymbol{\tau}_{\max} \quad (10)$$

En plus, pour des raisons de simplicité d'écriture, toutes les contraintes inégalités seront écrites comme $\mathbf{g}_j(\boldsymbol{x}, \boldsymbol{\tau}) \leq 0, j = 1,...,\mathrm{J}$, indifféremment si elles dépendent ou non des variables d'état, de commande ou des deux. On aurait donc J contraintes inégalités, $2L$ contraintes égalités (passages imposés), et $2n$ contraintes égalités représentant les équations d'état, avec n composantes décrivant la pose et n autres composantes décrivant les vitesses cartésiennes et angulaires du robot.

C. Indice de performance:

Globalement, il est possible d'optimiser tout critère ayant un sens physique. Dans cette thèse on s'intéresse à une fonction objectif multicritères optimisant le temps et les énergies électriques et cinétiques nécessaires à l'exécution de la tâche, ainsi qu'une *mesure de manipulabilité* [62]. Cette dernière est introduite afin d'éviter les poses singulières. Comme nous l'avons mentionné plus haut, l'évitement des singularités est de première importance pour le bon succès d'un système de planification de trajectoire d'un robot. En se basant sur l'indice de performance relatant la *mesure de manipulabilité*, la fonction suivante d'évitement des singularités est définie [62] :

$$\varpi(\boldsymbol{x}_{1k}) = \frac{1}{\sqrt{\det(\boldsymbol{J}(\boldsymbol{x}_{1k})\boldsymbol{J}^T(\boldsymbol{x}_{1k}))}} \tag{11}$$

Pour le critère temps minimum, il existe deux façons d'effectuer la commande. La première consiste à appliquer une commande tout ou rien. L'optimisation est alors effectuée en déterminant le nombre optimum de commutation. La deuxième consiste à déterminer une valeur initiale faisable du pas d'échantillonnage, puis le faire varier pendant la procédure d'optimisation. Vu la forte non-linéarité de la dynamique et des contraintes du système en question, nous avons opté pour la deuxième. Ainsi, lors de l'initialisation, le pas d'échantillonnage est fixé à une valeur $h = \frac{t_f - t_0}{N}$, où t_0, t_f et N sont le temps initial, le temps final, et le nombre de discrétisation. En temps continu, le problème de commande optimale peut être énoncé comme suit :

Parmi toutes les commandes $(\boldsymbol{\tau}, h) \in \mathbf{C}\text{x}\mathrm{H}$ *qui permettent de ramener l'effecteur du robot d'un point initial à un point final, déterminer celles qui optimisent l'indice de performance multi-objectif:*

$$\underset{\substack{\boldsymbol{\tau}(t)\in \mathsf{C} \\ t_0, t_T \in \mathrm{H}}}{\mathrm{Min}} \; E = \int_{t_0}^{t_T} \left\{ \left[\boldsymbol{\tau}(t)\boldsymbol{U}\boldsymbol{\tau}^T(t) + \boldsymbol{\iota} + \frac{1}{2}\boldsymbol{x}_2(t)\boldsymbol{Q}\boldsymbol{x}_2^T(t) + \boldsymbol{\delta}\,\varpi(\boldsymbol{x}_1(t)) \right] dt \right\} \tag{12}$$

sous les contraintes (3)-(8)

Dans l'indice de performance (11), $\boldsymbol{\iota}$ et $\boldsymbol{\delta}$ sont des scalaires positifs pondérant le critère temps-minimum et la fonction d'évitement des singularités, alors que $\boldsymbol{U}$ et $\boldsymbol{Q}$ sont des matrices de dimension $n \times n$ symétriques et définies positives qui

servent de pondération des énergies électrique et cinétique. En temps discret, le problème est équivalent à déterminer parmi toute les commandes discrètes $(\tau_0,...,\tau_{N-1})$, $(h_0,...,h_{N-1})$ du domaine admissible et permettant de déplacer l'effecteur du robot de l'état initial $\boldsymbol{x}_0 = \boldsymbol{x}_S$ à l'état final $\boldsymbol{x}_N = \boldsymbol{x}_T$, celles qui rendent minimum l'indice de performance discret E_d définie par :

$$\underset{\substack{\tau \in \mathbf{C} \\ h \in \mathrm{H}}}{\mathrm{Min}}\, E_d = \left\{ \sum_{k=1}^{N} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \boldsymbol{\iota}_1 + \boldsymbol{x}_{2k} \boldsymbol{Q} \boldsymbol{x}_{2k}^T + \boldsymbol{\delta}\, \boldsymbol{\varpi}\,(\boldsymbol{x}_{1k})] h_k \right\} \tag{13}$$

sous :

$$\boldsymbol{x}_{k+1} = f_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k), \quad k = 0,..., N\text{-}1$$

$$\boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \leq 0\ , \quad j = 1,..,\mathrm{J}$$

$$\boldsymbol{s}_i(\boldsymbol{x}_k) = 0\ , \quad i = 1, ..., 2L$$

Dans (13) les contraintes égalités et inégalités $\boldsymbol{s}$ et $\boldsymbol{g}$ font référence aux contraintes (3) à (7) mentionnées ci-haut. Dans la section suivante, le problème résultant de commande optimale non linéaire et non convexe est résolue en appliquant le principe du maximum de Pontryagin [41]. La determination des étatsadjoints est rendue possible grace au découplage de la dynamique du robot. Enfin, la technique d'optimisation du Lagrangian augmenté en temps discret est utilisée pour la résolution numérique du problème.

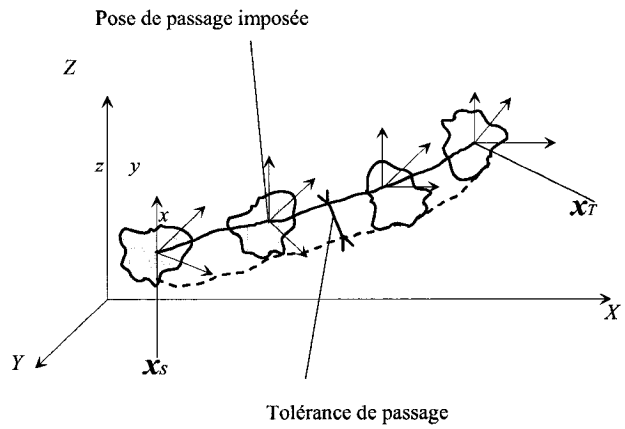


Fig. 3.3. Illustration du passage de l'effecteur par des poses imposées (positions et orientations)___ chemin optimal, --- chemin faisable,
xs pose de départ, , x_T pose d'arrivée

.

3. 4. 2. Approche proposée : Lagrangien augmenté avec découplage

A. Lagrangien augmenté (LA):

Une solution numérique itérative est plutôt utilisée lors de l'implémentation au lieu d'une solution analytique. Ceci est dû à la grande non linéarité et complexité du problème considéré. Dans le processus de résolution du problème posé, l'algorithme du Lagrangien augmenté (LA) est utilisé pour transformer le problème de commande sous contrainte en un problème de commande sans contraintes. Dans cette approche, le degré de pénalité de violation des contraintes est régulé par des coefficients de pénalité qu'on adapte à chaque itération. Cet algorithme est basé sur les méthodes de pénalité quadratique; mais réduit la possibilité de mauvais conditionnement des sous problèmes successifs générés par pénalisation en introduisant explicitement des multiplicateurs de Lagrange à chaque itération. Ceci permet d'accélérer la

convergence [40, 41, 63-67]. En outre, alors que le Lagrangien ordinaire est généralement non convexe en présence de contraintes non convexes (comme c'est le cas du problème considéré), il a été démontré que la fonction LA peut être convexifiée par un choix judicieux du coefficient de pénalité. Dans l'annexe C, on fournie un bref rappel sur l'optimisation non linèaire avec contraintes par Lagrangien augmenté et les conditions d'optimalité de Kuhn-Tuker. Cette procédure est décrite sur la figure 3.4, où les formules itératives de mise à jour des paramètres et variables sont mentionnées. La procédure part d'une trajectoire initiale faisable déterminée par un profil de vitesse trapézoïdale ou cycloïdal, une suite de positions, vitesses, et accélérations est déterminée. Les couples moteurs correspondants sont alors calculés à l'aide du modèle dynamique inverse (MDI). La première étape de la procédure d'optimisation consiste à calculer la suite des états adjoints de façon rétrograde. Ensuite, les valeurs des couples et périodes d'échantillonnage sont déterminés en utilisant les formules itératives du gradient de la fonction Lagrangien augmenté. Les couples moteurs et les périodes d'échantillonnage ainsi calculés sont introduits dans l'équation dynamique discrète (3) pour déterminer le nouvel état du robot. La trajectoire résultante est testée en vérifiant la satisfaction des contraintes égalité et inégalité par rapport à des tolérances de faisabilité. Si ce test n'est pas satisfait, une autre itération interne est effectuée et une nouvelle trajectoire est déterminée jusqu'à satisfaction de ce test. Quand le test de faisabilité est satisfait, un autre test de convergence est réalisé avec des tolérances plus serrées. Si les contraintes sont satisfaites par rapport à un premier niveau de tolérance, alors les valeurs actuelles de

pénalité sont bonnes en maintenant une trajectoire proche de l'optimum. Les paramètres de Lagrange sont mises à jour sans changer les coefficients de pénalité. Si les valeurs de pénalité ne satisfont pas ce test, alors elles sont réduites, tout en maintenant les paramètres de Lagrange inchangés. Dans les deux cas, les tolérances sont réduites pour forcer les itérations suivantes à fournir de meilleures trajectoires en termes de minimisation de la fonction objectif et de la satisfaction des contraintes. Pour plus de détails sur l'implémentation du Lagrangien augmenté, le lecteur est référé aux références [40, 41, 63-67].

La fonction LA transformant le problème d'optimisation sous contraintes en un problème sans contraintes peut s'écrire comme :

$$L_{\mu}(\boldsymbol{x}, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = E_d + \sum_{k=0}^{N-1} \left\{ \boldsymbol{\lambda}_{k+1}^T (\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)) \right\} + \tag{14}$$

$$\sum_{k=0}^{N-1} h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \Psi_{\mu_S}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^l(\boldsymbol{x}_k)) + \sum_{j=1}^{1} \Phi_{\mu_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)) \right] + \sum_{i=1}^{2} h_N \Psi_{\mu_S}(\boldsymbol{\sigma}_N^i, \boldsymbol{s}_i^L(\boldsymbol{x}_N))$$

où $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ est définie par l'équation discrète (3) à l'instant d'échantillonnage k, N est le nombre total d'échantillonnage, $\boldsymbol{\lambda} \in R^{2nN}$ désigne l'état adjoint, obtenue à partir des équations adjointes, $\boldsymbol{\sigma}, \boldsymbol{\rho}$ sont les multiplicateurs de Lagrange associés aux contraintes égalité et inégalité, $\boldsymbol{\mu}_S$ et $\boldsymbol{\mu}_g$ sont les coefficients de pénalité correspondant. Les fonctions de pénalité Ψ_{μ} et Φ_{μ} adoptées ici combinent les

méthodes de pénalité et les méthodes duales. Ceci permet la relaxation des contraintes inégalités aussitôt qu'elles sont satisfaites. Les expressions de ces fonctions sont données en annexe A, où on donne u aussi n bref rappel mathématique des techniques d'optimisation non linéaires sous contraintes

Les conditions d'optimalité de premier ordre de Karush-Kuhn-Tucker stipulent que pour qu'une trajectoire $(\boldsymbol{x}_0, \boldsymbol{\tau}_0, h_0),...,(\boldsymbol{x}_N, \boldsymbol{\tau}_{N-1}, h_{N-1})$ soit solution optimale du problème, il doit exister des multiplicateurs de Lagrange positifs $(\boldsymbol{\lambda}_k, \boldsymbol{\rho}_k)$, et des multiplicateurs de signes quelconques $\boldsymbol{\sigma}_k$, et des coefficients de pénalité positifs $\boldsymbol{\mu}_s$, $\boldsymbol{\mu}_g$ tel que :

$$\frac{\partial L_\mu}{\partial \boldsymbol{x}_k}=0, \frac{\partial L_\mu}{\partial \boldsymbol{\tau}_k}=0, \frac{\partial L_\mu}{\partial h_k}=0, \frac{\partial L_\mu}{\partial \boldsymbol{\lambda}_k}=0, \frac{\partial L_\mu}{\partial \boldsymbol{\rho}_k}=0, \frac{\partial L_\mu}{\partial \boldsymbol{\sigma}_k}=0,$$

$$\boldsymbol{\sigma}_k^T \boldsymbol{s}(\boldsymbol{x})=0, \boldsymbol{\rho}_k^T \boldsymbol{g}(\boldsymbol{x},\boldsymbol{\tau},h)=0, \text{ et } \boldsymbol{g}(\boldsymbol{x},\boldsymbol{\tau},h)\le 0 \qquad (15)$$

Le développement de ces conditions nous permet de déterminer les formules itératives pour résoudre le problème de commande optimale résultant, en ajustant les variables de commande, les multiplicateurs de Lagrange, ainsi que les coefficients de pénalité. La recherche linéaire de la solution optimale $(\boldsymbol{\tau}^*, h^*)$ est réalisée tout en satisfaisant les contraintes (solution faisables) et en optimisant la fonction objectif (solution utilisable) du domaine admissible des variables $\boldsymbol{\tau}$ et h. Cependant, dans l'équation (3), $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ contient l'inverse de la matrice d'inertie totale $\boldsymbol{M}^{-1}(\boldsymbol{x})$ de l'ensemble des éléments composant le manipulateur robotique, ainsi que celles des

forces de Coriolis et centrifuges associées $N(x_1, x_2)$. Les expressions décrivant ces matrices peuvent s'étaler sur plusieurs pages pour certains systèmes robotiques. (C'est le cas par exemple d'une PGS ou MCP de forme générale [7]). Le développement des conditions d'optimalité de premier ordre et le calcul des états adjoints λ_k, nécessitent la détermination de l'inverse de ces matrices, ainsi que des dérivées de ces inverses par rapport aux variables d'état. Ceci engendre une grande complexité même en utilisant le calcule symbolique.

B. Lagrangien augmenté avec découplage (LAD)*:*

La difficulté majeure d'évaluation des états adjoints mentionnée ci-dessus ne peut pas être résolue par une implantation directe du Lagrangien augmenté. Nous avons résolu ce problème en découplant la dynamique du robot, à l'instar des approches utilisées dans les systèmes de commande en boucles fermées [68].

Théorème:

Sous la condition d'inversibilité de la matrice d'inertie, la loi de commande définie par :

$$\boldsymbol{u} = \boldsymbol{M}(\boldsymbol{x}_1)\boldsymbol{v} + \boldsymbol{N}(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 + \boldsymbol{G}(\boldsymbol{x}_1) \tag{16}$$

permet au robot d'avoir un comportement linéaire découplé avec équation dynamique:

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{v} \tag{17}$$

où $\boldsymbol{v}$ est une entrée auxiliaire.

Ce résultat s'en suit simplement en substituant la loi de commande proposée (16) dans le modèle dynamique (2). On obtient

$$\boldsymbol{M}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 = \boldsymbol{M}(\boldsymbol{x}_1)\boldsymbol{v}$$

Comme $\boldsymbol{M}(\boldsymbol{x})$ est inversible, alors on a $\dot{\boldsymbol{x}}_2 = \boldsymbol{v}$

Ceci permet au robot d'avoir un comportement linéaire découplé décrit par l'équation discrète suivante:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_{dk}\boldsymbol{x}_k + \boldsymbol{B}_{dk}\boldsymbol{v}_k = \boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) , \tag{18}$$

avec $$\boldsymbol{F}_{dk} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & h_k\boldsymbol{I}_{n\times n} \\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n} \end{bmatrix} \text{ et } \boldsymbol{B}_{dk} = \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{n\times n} \\ h_k\boldsymbol{I}_{n\times n} \end{bmatrix}$$

Notons que cette formulation réduit énormément les calculs, en nous évitant de déterminer à chaque itération l'inverse de la matrice d'inertie et ses dérivées par rapport aux variables d'état. Ceci permet le calcul des états adjoints λ_k, avec une grande facilité. Cependant, dans cette formulation la non linéarité est transférée à l'indice de performance. En effet, dans la formulation découplée, le problème (13) devient celui de déterminer les périodes d'échantillonnage et accélérations articulaires $(h_0, h_2, ..., h_{N-1}), (\boldsymbol{v}_0, \boldsymbol{v}_2, ..., \boldsymbol{v}_{N-1})$, permettant à l'effecteur du robot de se déplacer d'un état initial à un état final, tout en minimisant la fonction coût E_d, exprimée pour le problème découplé comme suit :

$$E_d^D = \underset{\substack{v\in V \\ h_k\in H}}{\text{Min}} \left\{ \sum_{k=0}^{N-1} \left[\left[M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right]^T \right.\right.$$

$$\left.\left. U\left[M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] + l_1 + x_{2k}^T Q x_{2k} + \delta \varpi(x_{1k}) \right] h_k \right\} \quad (19)$$

sous la contrainte:

$$x_{k+1} = f_{d_k}^D(x_k, v_k, h_k), \quad k = 0,..., N\text{-}1$$

Ainsi que les autres contraintes (3)-(8), qui restent inchangées sauf pour les contraintes sur les couples moteurs, qui deviennent:

$$\tau_{\min} \le M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \le \tau_{\max} \quad (20)$$

Donc, les contraintes inégalité g_3 et g_4 peuvent être récrite comme:

$$g_3^D(x_k, v_k) = \tau_{\min} - \left[M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] \le 0 \quad (21)$$

$$g_4^D(x_k, v_k) = \left[M(x_{1k})v_k + N(x_{1k}, x_{2k})x_{2k} + G(x_{1k}) \right] - \tau_{\max} \le 0 \quad (22)$$

Le Lagrangien augmenté associé au problème découplé est donné par:

$$L_R(x, v, h, \lambda, \rho, \sigma) = E_d^D + \sum_{k=0}^{N-1} \left\{ \lambda_{k+1}^T (x_{k+1} - f_{d_k}^D(x_k, v_k, h_k)) \right\} +$$

$$\sum_{k=0}^{N-1} h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \Psi_{\mu_S}(\sigma_k^i, s_i^{Dl}(x_k)) + \sum_{j=1}^{J} \Phi_{\mu_g}(\rho_k^j, g_j^D(x_k, v_k)) \right] + \sum_{i=1}^{2} h_N \Psi_{\mu_S}(\sigma_N^i, s_i^{DL}(x_N)) \quad (23)$$

où la fonction $f_{d_k}^D(x_k, v_k, h_k)$ est définie par l'équation d'état (18) à l'instant discret k, les autres paramètres de (23) sont définis auparavant. De nouveau, l'application des conditions d'optimalité de premier ordre impliquent que pour qu'une trajectoire

$(x_0, v_0, h_0),...,(x_N, v_{N-1}, h_{N-1})$ soit une solution au problème (19), il doit exister des suites des multiplicateurs de Lagrange positifs (λ_k, ρ_k), des multiplicateurs de signes quelconques σ_k, et des coefficients de pénalité positifs (μ_s, μ_g) tel que les conditions (15) soient vérifiées pour le problème découplé. Les états adjoints λ_k et les variables de commande v_k, h_k sont aussi déterminés à partir des conditions nécessaires d'optimalité. Cette approche de modélisation et de résolution du problème de planification hors-ligne de trajectoire a fait l'objet de deux programmes de simulation: le premier pour réaliser la POT d'une MCP à deux DDL, le deuxième pour la POT d'un RS à trois DDL redondants par rapport à la tâche. Dans les simulations réalisées, l'emphase était mise sur la planification temps énergie minimum, sans singularités de trajectoire par Lagrangien augmenté. Les aspects cinématiques des robot sériels et parallèles étudiés, ont été considérés dans [57, 69]. Le programme de simulation a été codé sous Matlab [70]. Trois critères sont évalués dans nos simulations.

1) Évaluer l'impact des variations des paramètres dynamiques et la sensibilité du Lagrangien augmenté sur le comportement des trajectoires du robot.

2) A quelle précision versus temps de calcul le Lagrangien augmenté satisfait le passage par les poses imposées ?

3) Enfin, quel est l'impact du nombre d'itérations internes et externes sur les performances des trajectoires versus le temps machine de calcul ?

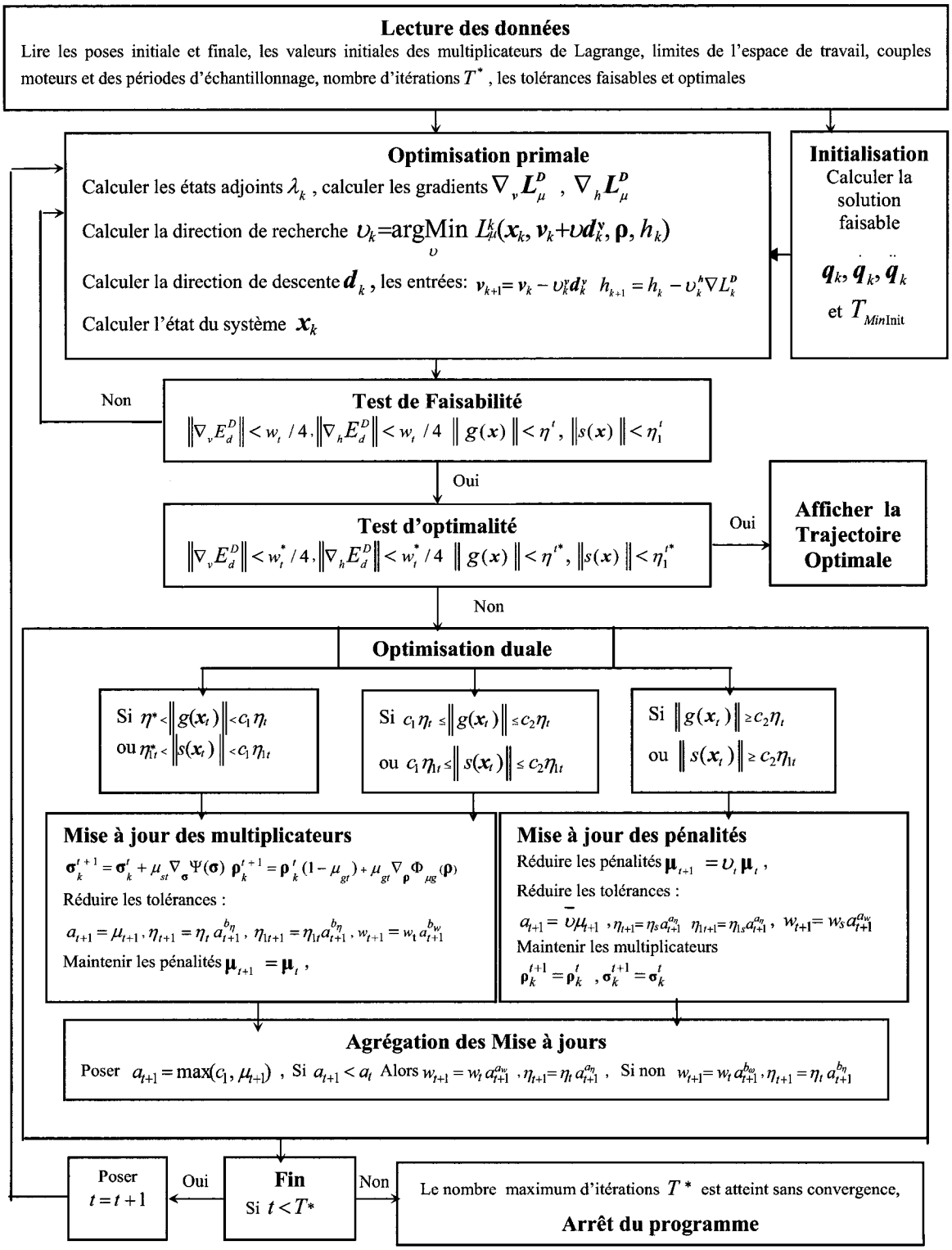


Fig. 3. 4. Architecture et fonctionnement de l'algorithme du Lagrangien augmenté

3. 5. Planification en-ligne de trajectoire

3. 5. 1. Limitations des approches conventionnelles

La deuxième partie de cette thèse traite de la planification en-ligne multi-objectif de trajectoire des manipulateurs robotiques. Bien que l'étude développée s'applique à des systèmes robotiques à n–DDL, nous l'avons implémenté pour un robot sériel planaire redondant à 3–DDL (Fig. 3.1).

Les performances des approches conventionnelles de commande des systèmes appliqués à la planification en-ligne de trajectoires des robots se sont avérées récemment très limitées. En effet, dans [17] on montre que la poursuite d'une trajectoire calculée en temps minimum entraîne des dépassements des valeurs nominales des couples, et des vibrations au niveau articulaire. En outre, les contrôleurs en temps minimum nécessitent un temps énorme de mise en œuvre et d'implémentation.

Enfin à cause de leur structure – en boucle ouverte – ces contrôleurs ne sont pas nécessairement les bons choix pour la planification en-ligne des trajectoires des robots, qui fonctionnent plutôt en boucle fermée. D'autre part, les commandes par retour d'état linéaire à gain fixe ont été largement utilisées pour la commande des systèmes mécaniques. Cependant, il a été démontré que ces commandes ont des limitations sérieuses, qui sont dues notamment à la complexité et fortes non linéarités, ainsi qu'aux couplages des transformations cinématiques et équations dynamiques régissant le mouvement du robot. Ces problèmes ont permis la découverte de

plusieurs schémas de commande non linéaires. Parmi ceux qui ont un grand succès en robotique, citons-en : la méthode '*Computed Torques*', la commande '*Backsteeping*', la commande par linéarisation par bouclage, la méthode de découplage et rejet des perturbations, et enfin les commandes robustes de type H_∞ [53-55, 71-73].

Enfin à cause de leur structure – en boucle ouverte – ces contrôleurs ne sont pas nécessairement les bons choix pour la planification en-ligne des trajectoires des robots, qui fonctionnent plutôt en boucle fermée. D'autre part, les commandes par retour d'état linéaire à gain fixe ont été largement utilisées pour la commande des systèmes mécaniques. Cependant, il a été démontré que ces commandes ont des limitations sérieuses, qui sont dues notamment à la complexité et fortes non linéarités, ainsi qu'aux couplages des transformations cinématiques et équations dynamiques régissant le mouvement du robot. Ces problèmes ont permis la découverte de plusieurs schémas de commande non linéaires. Parmi ceux qui ont un grand succès en robotique, citons-en : la méthode '*Computed Torques*', la commande '*Backsteeping*', la commande par linéarisation par bouclage, la méthode de découplage et rejet des perturbations, et enfin les commandes robustes de type H_∞ [53-55, 71-73].

Encore une fois, ces approches font appel aux modèles dynamiques et cinématiques détaillés du robot. Ces modèles sont d'une grande complexité pour une implémentation temps réel. Ces difficultés ont poussé plusieurs chercheurs à utiliser les techniques d'intelligence artificielle, notamment les systèmes flous [74-81].

3. 5. 2. Approche proposée : Réseaux neuro-flous dirigés par les données

Dans la modélisation par les systèmes flous, on distingue deux grandes classes : Les modèles subjectifs sont réalisés à partir d'interrogation d'experts du domaine. Les modèles objectifs sont construits à base de données entrées/sorties collectée à partir de mesures expérimentales ou à partir de résultats de simulation basés sur un modèle mathématique représentatif du problème. Dans notre étude, la modélisation objective est utilisée pour résoudre le problème de planification en-ligne multi-objectif de trajectoire. Ce modèle consiste en un système d'inférence neuro-flou dirigé par les données (*Data-Driven Neuro-Fuzzy Inference System*). La ligne directrice de cette approche consiste à générer par simulation des trajectoires optimisées au sens d'un ou plusieurs critères et satisfaisant diverses contraintes (Fig. 3.3). Ces trajectoires sont construites à partir de modèles représentatifs de la dynamique, cinématique, tâche et espace de travail du robot. Un ensemble de trajectoires suffisamment important est généré pour couvrir l'espace de travail. Ensuite, les résultats optimaux sont utilisés pour entraîner et optimiser un système d'inférence neuro-flou. Trois méthodes de génération de trajectoire ont été utilisées.

A. Première méthode

Dans la première, un réseau de neurone a été construit et optimisé pour apprendre l'inverse cinématique du manipulateur (Fig. 3. 5). Dans cette méthode, l'utilisateur introduit la position initiale et finale du point opérationnel de l'effecteur du robot, en coordonnées cartésiennes à deux dimensions [78]. Ensuite, une phase de

pré-traitement est effectuée, lors de laquelle une paramétrisation en temps-minimum de la trajectoire par un profil cycloïdal est déterminée dans l'espace articulaire. Les équations cinématiques des positions; vitesses et accélérations gouvernant le mouvement du robot sont déduites dans l'espace articulaire et sont ensuite remplacées dans le MDI du robot pour déterminer les couples moteurs correspondants. La trajectoire ainsi déterminée servira d'une solution initiale à un problème de commande en temps-minimum, incluant la redondance cinématique, la dynamique, et les limitations de l'espace de travail du robot. Le Lagrangien augmenté en temps discret est utilisé pour la résolution numérique du problème résultant de commande optimale non linéaire et non convexe. Ces deux procédures d'initialisation et d'optimisation sont répétées suffisamment pour couvrir la plus grande partie possible de l'espace de travail. Une fois l'ensemble des trajectoires en temps-minimum est construit, il est mis sous forme entrées/sorties avec comme entrées les positions et vitesses articulaires et comme sorties les couples moteurs associés.

Ces données sont normalisées en vue d'un deuxième pré-traitement. Ce dernier consiste à grouper ces données dans des partitions par application de l'algorithme de groupement de données appelé '*Subtractive Clustering*' [80, 81]. Par la même, cet algorithme permet l'initialisation des paramètres des fonctions d'appartenance qui vont modéliser les variables linguistiques entrées/sorties d'un réseau neuro-flou (Fig. 3. 6). Le nombre d'ensembles flous, donc de règles gouvernant le modèle flou objectif représentant le comportement des trajectoires du manipulateur en temps-minimum est aussi déterminé lors de cette initialisation. Le réseau neuro-flou qui est

ensuite entraîné, raffiné et optimisé est basé sur le mécanisme d'inférence de *Tsukamoto*. Après apprentissage, ce réseau est utilisé ultérieurement lors d'une phase de généralisation pour la planification temps-minimum en-ligne de trajectoire du manipulateur.

B. Deuxième méthode

La deuxième méthode est similaire à la première, sauf que le réseau de neurones utilisé pour la résolution de la cinématique inverse est remplacé par n réseau neuro-flou. On a remarqué une amélioration substantielle des performances du réseau neuro-flou comparées à celles du réseau de neurones [79].

C. Troisième méthode

Les limitations majeures des deux méthodes précédentes sont liées à la nature de la tâche considérée, coude haut ou coude bas. Ces limitations sont relaxées dans la troisième méthode, où la phase de pré-traitement est complètement réalisée hors-ligne par application de la technique LA développée dans la première partie de la thèse. LA Dans cette programmation hors ligne, tout l'espace de travail du robot est considéré. La redondance cinématique est résolue par la technique de la Jacobienne étendue. Un critère multi-objectifs incluant, le temps et l'énergie nécessaires à l'exécution de la tâche, ainsi qu'une *mesure de manipulabilité* afin d'éviter les poses singulières du robot est implanté. De plus, dans le processus d'optimisation plusieurs contraintes sont prises en compte. Les limitations des actionneurs, les débattements articulaires,

les périodes d'échantillonnages, ainsi que le passage de l'effecteur du robot par un ensemble de poses imposées. Un ensemble de 100 trajectoires est généré. Ensuite, à l'instar des deux approches précédentes, ces données sont normalisées, puis partitionnées en utilisant l'algorithme de groupement des données '*Subtractive Clustering*'. Cette procédure de groupement permet aussi l'initialisation des paramètres du système neuro-flou. Ces paramètres sont ensuite optimisés par un moteur d'inférence de type *Tsukamoto* (Fig. 3. 7). Les Figures 3.8 illustre le fonctionnement de la troisième méthode, ainsi que l'architecture du réseau proposé. De même que la deuxième méthode, ce réseau est utilisé après apprentissage pour une planification en-ligne de trajectoire.

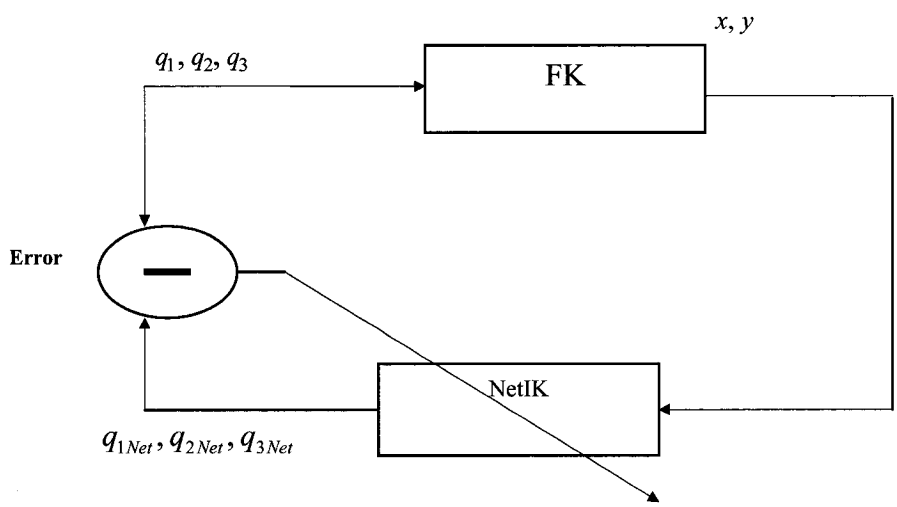


Fig. 3. 5. Utilisation du modèle cinématique directe (MCD) pour réaliser l'apprentissage de NetIK

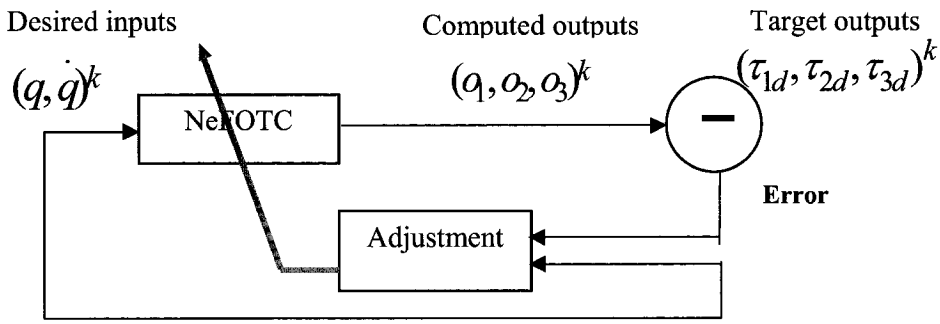


Fig. 3. 6. Utilisation du modèle dynamique inverse (MDI) pour l'apprentissage de NeFOTC

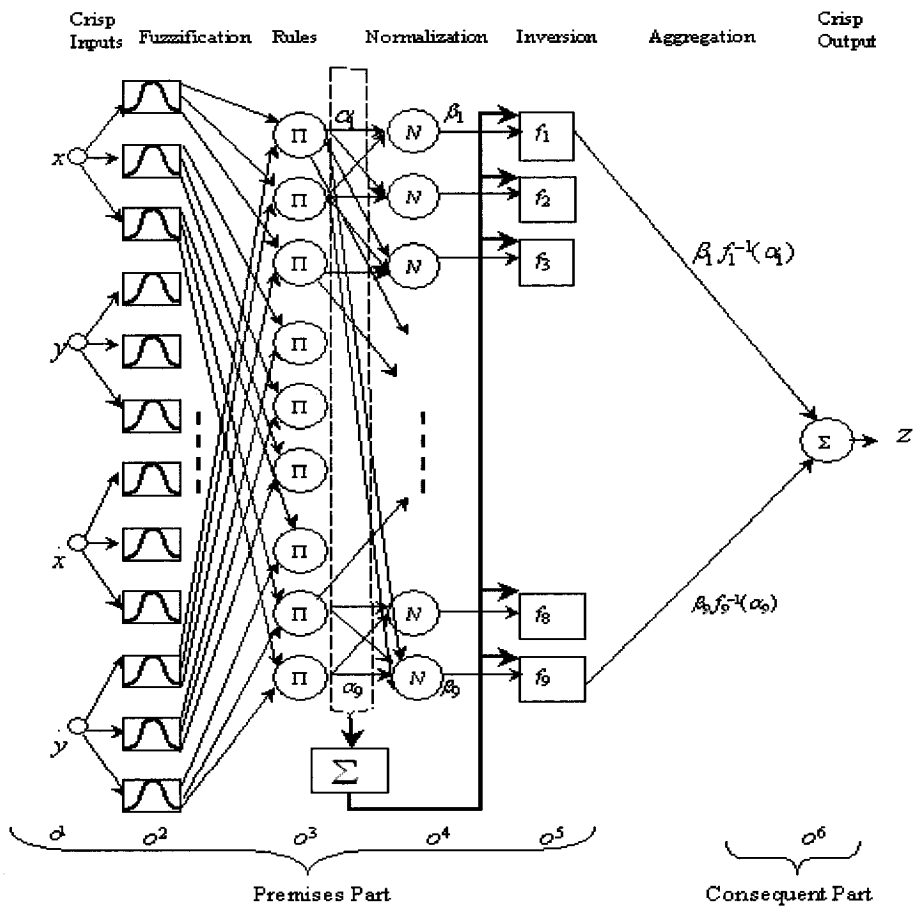


Fig. 3. 7. Topologie et architecture NeFuMOP

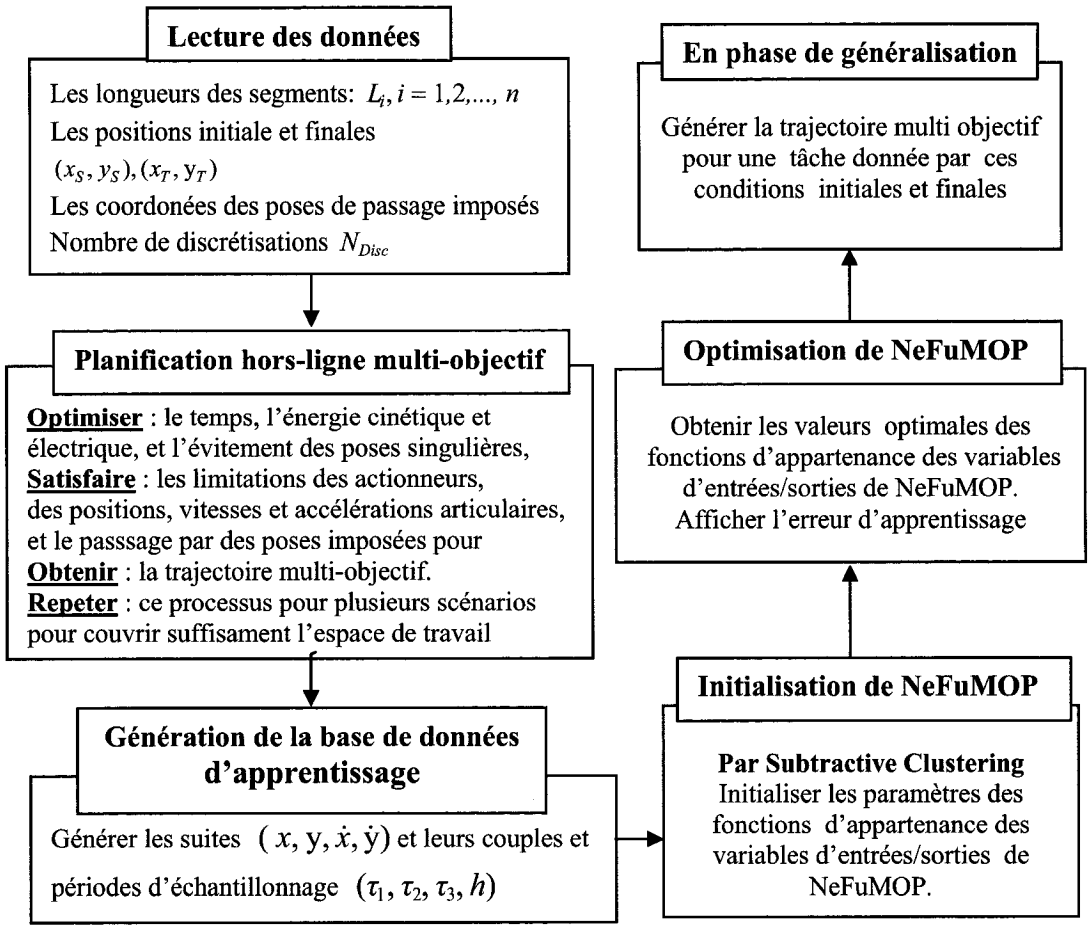


Fig. 3. 8. Architecture du système proposé de planification multi-objectif neuro-flou

3. 6. Présentation des articles

Cette thèse est présentée sous forme d'une thèse par articles. Les trois articles de revue réalisés font l'objet des chapitres 4 à 6. Dans la suite du présent chapitre, on va donner une brève présentation de chaque article.

3. 7. Présentation de l'article 1 : Planification multi-objectif hors-ligne de trajectoire des machines à cinématiques parallèles

À la suite de l'étude bibliographique sur le problème de POT-MCP, nous avons

constaté que ce problème n'a pas été suffisamment considéré dans la littérature, comparativement à son homologue POT-RS. Bien que plusieurs raisons, qu'on a mentionné au chapitre 2 on justifie cette situation, nous avons tenté d'aborder ce problème en le formulant dans le cadre du calcul variationnel, comme un problème de commande optimale non linéaire sous contraintes. Parce que la période d'échantillonnage est un élément fondamental, d'abord pour l'optimisation en temps-minimumle des trajectoires, ensuite pour permettre un meilleur suivi de ces trajectoires lors de la commande du robot physique, nous avons pris comme variables de commande les couples moteurs, ainsi que les périodes d'échantillonnage. L'indice de performance inclue donc le temps, les énergies cinétique et électrique nécessaire à l'exécution de la tâche, ainsi qu'une fonction d'évitement des poses singulières.

Différentes contraintes sont également introduites, telles que les limitations sur les couples actionneurs, l'espace de travail, les buttés articulaires, les périodes d'échantillonnage, ainsi que le passage par des poses (position et orientation) imposées. L'algorithme du Lagrangien augmenté en temps-discret a été utilisé pour surmonter les difficultés liées au conditionnement numérique du problème et aux non convexités des contraintes. Cet algorithme est implémenté sur une forme découplée de la dynamique du robot. Ce découplage a permis notamment de faciliter le calcul des états adjoints résultant de la mise en œuvre des conditions nécessaires d'optimalité de premier ordre. L'implémentation de cette approche a été réalisée sur un modèle d'une MCP à 2-DDL. L'étude cinématique de cette MCP a été rapportée dans [69]. Dans les simulations réalisées, l'emphase était mise sur la planification

temps énergie minimum, sans singularités de trajectoire par Lagrangien augmenté. Nous avons mis l'accent sur les éléments suivants : 1) Comparaison des trajectoires du robot pour différentes valeurs des pondérations ι, U, Q et δ du temps, des énergies électrique et cinétique, ainsi que de la fonction d'évitement des singularités. 2) Évaluation de l'impact des variations des paramètres dynamiques sur le comportement des trajectoires de la MCP par une étude de sensibilité de l'algorithme. 3) À quelle précision versus temps de calcul le Lagrangien augmenté satisfait le passage par les poses imposées ? et enfin, 4) Quel est l'impact du nombre d'itérations internes et externes, sur les performances de la MCP versus le temps machine de calcul ?

Ce travail a fait l'objet de travaux préliminaires communiqués dans des conférences internationales [6, 7]. Une extension de ces travaux a fait l'objet du premier article de revue soumis pour publication et accepté sous réserve d'une révision à *Robotics and Computer Integrated Manufacturing* [67]. Cet article est donné au chapitre 4.

La structure du chapitre 4 peut être résumée comme suit :

- Une introduction contenant une revue bibliographique ;
- Une présentation des modèles cinématique, dynamique et des contraintes relatives aux limitations des longueurs des vérins, des couples actionneurs, de l'espace de travail, et des exigences de la tâche;
- Une formulation du problème comme un programme nonlinèaire sous contraintes;

- Une présentation du Lagrangien augmenté, transformant le problème de commande optimale sous contraintes en un problème sans contraintes.
- Une présentation du Lagrangien augmenté implémenté sur un modèle dynamique découplé du robot;
- Une présentation des simulation et comparaisons avec les techniques basées uniquement sur la cinématique du robot.
- Une interprétation des résultats obtenus et une conclusion générale.

Cet article a été soumis et accepté pour publication sous réserve de quelques modifications dans a revue : Robotics and Computer Integrated Manufacturing

A. Khoukhi, L. Baron, M. Balazinski: "Constrained Multi-Objective Trajectory Planning of Parallel Kinematic Machines", *Robotics and Computer Integrated Manufacturing*, 2007.

3. 8. Présentation de l'article 2 : Réseaux neuro-flous hiérarchiques pour la planification temps-minimum en-ligne de trajectoire des robots manipulateurs

L'objectif de cette partie de la thèse consiste à déterminer un outil efficace de planification en-ligne de trajectoire pour manipulateurs robotiques. Une étude bibliographique nous a permis de connaître la difficulté du problème et les limites des approches proposées jusqu'à date. En effet, l'utilisation des techniques conventionnelles de commande des systèmes n'est sans doute pas la meilleure approche du problème. D'abord les approches simples par retour d'état à gain constant ne permettent pas une bonne poursuite en-ligne, à cause notamment des

fortes non linéarités des transformations cinématiques et des couplages dynamiques du système. Ceci a donné lieu à des techniques de commande avancées tel que les commande *Computed Torques,* H_∞ et *Backsteeping Predictive Control.* Cependant, ces techniques font appel aux modèles détaillés du robot. Ce qui engendre une grande complexité de calcul pour une implémentation temps réel.

Dans ce cadre nous avons suggéré d'utiliser les réseaux neuro-flous dirigés par les données. Les trajectoires d'apprentissage sont générées suivant un schéma de planification hors-ligne en temps-minimum sous contraintes des limites articulaires, des actionneurs, et périodes d'échantillonnage. Dans cet étude, l'indice de performance est le temps nécessaire à la réalisation de la trajectoire, sous contraintes de redondance cinématique et limitations des variables articulaires et couples moteurs. Ce travail a fait l'objet de travaux préliminaires ayant donné lieu à une communication publiée dans une conférence internationale [78]. Une extension de ce travail a été consacrée à un troisième article de revue, accepté pour publication sous réserve de quelques modifications, dans *Engineering Applications of Artificial Intelligence* [79]. Cet article fait l'objet du chapitre 6.

La structure du chapitre 6 peut être résumée comme suit :

- Une introduction contenant une revue bibliographique ;
- Une brève présentation de l'approhe proposée,
-Une présentation du modèle cinématique et du réseau NeFIK permettant de résoudre l'inverse cinématique du manipulateur redondant.

- Une présentation des modèles dynamique et des contraintes relatives aux limites articulaires, des couples moteurs, de l'espace de travail, et des exigences de la tâche;
- Une formulation du problème comme un programme non linéaire sous contraintes;
- Une solution initiale feasable basée sur un profil cycloïdal en temps-minimum est d'abord introduite. Cette solution satisfait aux conditions aux limites au niveau articulaire (positions, vitesses, accélérations et jerks angulaires)
- Une présentation du Lagrangien augmenté;
- Une présentation du réseau neuro-flou permettant d'achever l'apprentissage du comportement du robot réalisant la trajectoire en temps-minimum simulée par Lagrangien augmenté
- Une présentation des simulation et comparaisons avec les techniques basées uniquement sur la cinématique du robot.
- Une interprétation des résultats obtenus et une conclusion générale.

Cet article a été soumis et accepté pour publication dans la revue *Engineering Applications of Artificial Intelligence*

A. Khoukhi, K. Demirli, L. Baron, M. Balazinski: "Hierarchical Neuro-Fuzzy Optimal Time Trajectory Planning for Redundant Manipulators", *Engineering Applications of Artificial Intelligence,* 2007.

3. 9. Présentation de l'article 3 : Réseaux neuro-flous dirigés par les données pour la planification multi-objectif en-ligne de trajectoire des manipulateurs redondants

Le quatrième article est une extension de l'article précédent où l'indice de performance est généralisé à un critère multi-objectif pour inclure le temps et les énergies cinétiques et électriques nécessaires à l'exécution de la tâche, ainsi qu'une fonction d'évitement des poses singulières.

Dans cet article, l'espace de travail inclue l'ensemble des poses atteignables par l'effecteur du robot, ce qui constitue une deuxième généralisation de l'article précédent dans lequel seulement une configuration coude haut ou coude bas ont été considérées dans les simulations. Les trajectoires d'apprentissage sont réalisées suivant un schéma de planification hors-ligne optimisant un indice de performance multi-objectif, incluant le temps, les énergies électrique et cinétique nécessaires à l'exécution de la tâche, ainsi qu'une fonction d'évitement des configurations singulières. Cette optimisation est effectuée en satisfaisant les limites articulaires, des actionneurs, et des périodes d'échantillonnage. La redondance cinématique est résolue tout en évitant les limites articulaires. Dans cette étude, en plus du Lagrangien augmenté avec découplage de la dynamique, la contrainte d'état final est satisfaite à chaque itération du programme. Ceci est réalisé par une procédure de re-ajustement via l'algorithme du gradient projeté sur l'espace tangent à cette contrainte.

Cette étude a fait l'objet de travaux préliminaires ayant donné lieu à une communication soumise pour publication à une conférence internationale [85]. Une

extension de ce travail a permis la réalisation d'un quatrième article de revue de cette thèse, soumis pour publication dans *Control and Intelligent Systems* [86].

Cet article fait l'objet du chapitre 7.

La structure du chapitre 7 peut être résumée comme suit :

- Une revue bibliographique suivie d'une brève introduction de l'approche proposée;
- Une présentation des modèles cinématique, dynamique et des contraintes relatives à la redondance cinématique, aux limitations des couples actionneurs, de l'espace de travail, et des exigences de la tâche;
- Une formulation du problème comme un programme non-linéaire multi-objectif;
- Une présentation du Lagrangien augmenté en temps discret;
- Une présentation du Lagrangien augmenté implémenté sur un modèle dynamique découplé du robot;
- Une présentation du réseau neuro-flou permettant de réaliser l'apprentissage du comportement du robot correspondant aux trajectoires en temps-minimum simulées par Lagrangien augmenté.
- Une présentation des simulations et comparaisons avec les techniques basées uniquement sur la cinématique du robot, obtenus ainsi que des performances du réseau neuro-flou, et enfin une interprétation des résultats et une conclusion générale.

Cet article a été soumis pour publication à la revue *Control and Intelligent Systems.*

A.Khoukhi, K. Demirli, L. Baron, M. Balazinski: "Data-Driven Neuro-Fuzzy Multi-Objective Trajectory Planning of Redundant Manipulators", Submitted to *Control and Intelligent Systems*, 2007.

CHAPITRE 4

Constrained Multi-Objective Trajectory Planning of Parallel Kinematic Machines

Amar Khoukhi , Luc Baron and Marek Balazinski

Mechanical Engineering Department, École Polytechnique de Montréal,

C. P. 6079, Succ. CV, Montreal QC, Canada H3C 3A7

(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

4. 1. Abstract This paper presents a new approach to multi-objective dynamic trajectory planning of Parallel Kinematic Machines (PKM) under task, workspace and manipulator constraints. The robot kinematic and dynamic models, (including actuators) are first developed. Then the proposed trajectory planning system is introduced. It minimizes electrical and kinetic energy, robot traveling-time separating two sampling periods, and maximizes a measure of manipulability allowing singularity avoidance. Several technological constraints such as actuator, link length, and workspace limitations, and some task requirements, such as passing through imposed poses are simultaneously satisfied. The discrete augmented Lagrangian technique is used to solve the resulting non-linear and non-convex constrained optimal control problem. A decoupled formulation is proposed in order to cope with some difficulties arising from dynamic parameters computation. A systematic

implementation procedure is provided along with some numerical issues. Simulation results proving the effectiveness of the proposed approach are given and discussed.

Keywords: Parallel Kinematic Machines, Constrained Off-line Programming, Nonlinear Optimal Control, Time-Energy Trajectory Planning, Augmented Lagrangian, Decoupling.

4. 2. Nomenclature

B: Reference frame attached to the center of mass of the base

A: Reference frame attached to the center of mass of the end-effector (EE)

A_i , B_i : i^{th} attachment point of leg i on body A and B

$\boldsymbol{e}_i$: Unit vector along the ith joint axis

$\mathbf{p} = [x \;\; y \;\; z]^T$: Position vector of the origin of A relative to B in B

${}^A\boldsymbol{a}_i$: Constant position vector of attachment point A_i in frame A.

$\dot{\mathbf{p}} = [\dot{x} \;\; \dot{y} \;\; \dot{z}]^T$: Velocity vector of the origin of A relative to B

${}^B\boldsymbol{R}_A$: Rotation matrix of frame A with respect to frame B

$\boldsymbol{x}_1 = \mathbf{q} = [\mathbf{p}^T \;\; \varphi \;\; \theta \;\; \psi]^T$: Position and orientation of A in B

$\dot{\mathbf{q}}_E = [\dot{\mathbf{p}}^T \;\; \dot{\varphi} \;\; \dot{\theta} \;\; \dot{\psi}]^T$: Time derivatives of $\boldsymbol{x}_1(t)$

$\boldsymbol{x}_2 = \dot{\boldsymbol{q}} = [\,\dot{\mathbf{p}}^T \;\; \boldsymbol{\omega}^T\,]^T$: Cartesian and angular velocity of the EE

$\boldsymbol{x} = [\boldsymbol{x}_1 \;\; \boldsymbol{x}_2]^T$: Continuous-time state of the PKM

$\boldsymbol{x}_k = [\boldsymbol{x}_{1k} \;\; \boldsymbol{x}_{2k}]^T$: Discrete-time state of the PKM

$\boldsymbol{\tau}(t)$: Cartesian force/torques wrench

$\boldsymbol{i} = [i_1 \;\; i_2 \ldots i_6]^T$: Vector of electric currents

$\boldsymbol{l} = [l_1 \;\; l_2 \ldots l_6]^T$: Vector of the link lengths

$\boldsymbol{J}$: Jacobian matrix of the PKM

$\boldsymbol{M}_j(\boldsymbol{q})$, $\boldsymbol{M}_c(\boldsymbol{q})$: Inertia matrix expressed in joint and Cartesian space

$\boldsymbol{N}_j(\boldsymbol{q},\dot{\boldsymbol{q}})$, $\boldsymbol{N}_c(\boldsymbol{q},\dot{\boldsymbol{q}})$: Coriolis and centrifugal force/torque in joint and Cartesian space

$\boldsymbol{G}_j(\boldsymbol{q})$, $\boldsymbol{G}_c(\boldsymbol{q})$: Gravity force in joint and Cartesian space

$\boldsymbol{M}_a$, M_a : Actuator inertia matrix and its component

$\boldsymbol{V}_a$, V_a : Actuator viscous damping coefficient matrix and its component

$\boldsymbol{K}_a$, K_a : Actuator gain matrix and its component

$\boldsymbol{K}$: Control law gain matrix

$\boldsymbol{\tau}_m$: Joint torque vector produced by the DC motors

$\boldsymbol{F}$: 6-directional column of the generalized forces

p : Ballscrew pitch

n : Gear ratio

J_s , J_m : Ballscrew and motor mass moments of inertia

b_s , b_m : Ballscrew and motor viscous damping coefficients

$\boldsymbol{\lambda}$: Lagrange multipliers associated to state variables

$\boldsymbol{f}_{d_k}$: Approximated function of the discretized robot dynamic model

$\boldsymbol{f}_{d_k}^{D}$: Approximated function of the discretized robot decoupled dynamics

$(\boldsymbol{\rho}, \boldsymbol{\sigma})$: Lagrange multipliers related to inequality and equality constraints

(Ψ, Φ) : Penalty functions associated to inequality and equality constraints

$(\boldsymbol{\mu}_g, \boldsymbol{\mu}_S)$: Penalty coefficients associated to inequality and equality constraints

$(\boldsymbol{g}, \boldsymbol{s})$: Generic functions defining inequality and equality constraints

$\mathbf{C}$: Admissible domain of input torques

H : Admissible domain of sampling periods

$\boldsymbol{U}, \boldsymbol{Q}$: Weight factors associated to electric and kinetic energy criteria

$(\boldsymbol{\iota}_1, \boldsymbol{\delta})$: Weight factors for traveling time and singularity avoidance criteria

ϖ : Singularity avoidance function

N : Total number of discretizations of the trajectory

w_t, η_t, η_{t1} : Cost minimization, equality and inequality constraints feasible tolerances

w^*, η^*, η_t^* : Cost minimization, equality and inequality constraints optimal tolerances

4. 3. Introduction

The design of Parallel Kinematic Machines (PKMs) dates back to the pioneer work by Gough [1], who established the basic principles of a manipulator in a closed loop structure. The machine was able to position and orientate an end-effector (EE), such that to test tire wear and tear. A decade later, Stewart [2] proposed a platform manipulator for the use as an aircraft simulator. Since then, extensive research efforts lead to the realization of several robots and machine tools with parallel kinematic structures [3]. PKMs have two basic advantages over conventional machines of serial kinematic structures. First, the connection between the base and the EE is made with several kinematic chains. This results in high structural stiffness and rigidity. Second, with such structure, it is possible to mount all drives on or near the base. This results in large payloads capability and low inertia. Indeed, the ratio of payload to the robot load is usually about 1/10 for serial robots, while only ½ for parallel ones. Despite these advantages, PKMs are still rare in industry. Among the major reasons for this gap are the small workspace, complex transformations between joint and Cartesian space, and singularities as compared to their serial counterparts. These facts lead to a tremendous amount of research in PKMs design and customization [3]. Another reason recently identified is the under consideration of the dynamics of these machines [4]. The mentioned architecture-dependent performance associated with strong-coupled non-linear dynamics makes the trajectory planning and control system design for PKMs more difficult, as compared to serial machines. In fact, for serial robots, there is a plentiful literature published on the topics of offline and online

programming, from both stand points: computational geometry and kinematics, and optimal control including robots dynamics [5-8]. For PKMs, a relatively large amount of literature is devoted to the computational kinematics and workspace optimization issues. The overwhelm criteria considered for PKMs trajectory planning are essentially design-oriented. These include singularity avoidance and dexterity optimization [9-13]. In [14], the authors had developed a clustering scheme to isolate and avoid singularities and obstacles for a PKM path planning. A kinematic design and planning method had been described in [15] for a four-bar planar manipulator mechanism. Another related work was considered in [16], where it had been shown that a motion planning with singularity-free pose change is possible for PKMs. A variational approach is reported in [17] for planning a singularity-free minimum-energy path between two end-points for Gough-Stewart platforms. The approach used is based on penalty optimization methods. These methods transform the constrained problem in an unconstrained one by augmenting the objective function with additional functions that reflect the violation of the constraints. Penalty methods, however, have several drawbacks [18]. Another major issue for off-line programming and practical use of PKMs in industry (in a machining process, for example) is that for a prescribed tool path in the workspace, the control system should guarantee the prescribed task completion within the workspace, for a given set up of the EE (i.e., for which limitations on actuator lengths and physical dimensions are not violated). This problem has been recently considered in [19, 20], where design methodologies involving workspace and actuator force limitations were proposed, using iterative

optimization techniques.

In this context, we consider a new integrated multi-objective dynamic trajectory planning system for PKMs. Part of this work has been presented in [21-22]. The proposed approach considers PKM's dynamics, including actuators models as well as task and workspace requirements, as a unique entity. It can be encapsulated into two levels (see Fig. 1): the modeling and approach level, and the simulation and testing level [7, 23]. The former consists of selecting the appropriate models and control approaches in order to optimize an overall performance of the robot-task-workspace interactions. The second level is devoted to coding, testing, and validation. Criteria to be optimized in this study are traveling time, energy necessary for a task execution and a measure of manipulability. The optimization procedure is performed within a proper balance between time and energy minimization, singularity avoidance, actuators, sampling periods, link lengths and workspace limitations, and task constraints satisfaction. From a state space representation by a system of differential equations in the phase plane, the trajectory planning is formulated within a variational calculus framework. The resulting constrained non-linear programming problem is solved using an augmented Lagrangian (AL) technique, implemented on a decoupled dynamics. AL algorithms have proven to be robust and powerful to cope with difficulties related to none strictly convex constraints [24-27] as compared to optimization methods employing only penalty. The decoupling technique is introduced in order to solve some difficult computations in the original non-linear and coupled formulation. Another advantage of the proposed method is that one might

introduce several criteria and constraints to satisfy in the trajectory planning process.

Section 2 introduces the kinematic and dynamic models using Euler-Lagrange formulation, and gives the associated discrete-time state space model. In section 3, the constrained nonlinear optimal control problem is formulated. In section 4, the augmented Lagrangian with decoupling technique is developed to solve the resulting linear and decoupled optimal control problem. In section 5, an implementation on a case study 2-DOF planar PKM is provided. Finally, section 6 concludes this work.

4. 4. Modeling

4. 4. 1. Kinematic Model

The PKM shown in Fig. 2 represents a full 6-Degrees-Of-Freedom (DOF) motion of its EE from its articulated motion of its six leg lengths. The *pose*, i.e., position and orientation, of the EE, namely, $\mathbf{q}=[\mathbf{p}^T \ \varphi \ \theta \ \psi]^T$ can be expressed in frame B by $\mathbf{p}$ the position vector of the origin of frame A relative to frame B, and the set of Euler angles (ϕ, θ, ψ) defining the orientation of the EE in frame B, namely, ${}^B\boldsymbol{R}_A$, as used to uniquely determine the EE orientation. This rotation matrix is given as

$$ {}^B\boldsymbol{R}_A = (\boldsymbol{R}_z(\varphi),\ \boldsymbol{R}_y(\theta),\ \boldsymbol{R}_z(\psi)) = \begin{bmatrix} c\psi c\phi - c\theta s\phi s\psi & -s\psi c\phi - c\theta s\phi c\psi & s\theta s\phi \\ c\psi s\phi + c\theta c\phi s\psi & -s\psi s\phi + c\theta c\phi c\psi & -s\theta c\phi \\ s\psi s\phi & c\psi s\theta & c\theta \end{bmatrix}, \qquad (1) $$

where c and s stand for the cosine and sine functions, respectively. Clearly, the orientation of A is described with respect to B by a rotation matrix ${}^B\boldsymbol{R}_A = [\mathrm{r}_1 \ \mathrm{r}_2 \ \mathrm{r}_3]$, where r_1, r_2 and r_3 are, respectively, 3×1 unit vectors along the axes of A

The velocity of the EE, namely, $\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix}$, can be obtained as a function of the time derivative of $\mathbf{q}$, i.e.,

$$\dot{\boldsymbol{q}} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3x3} & O_{3x3} \\ O_{3x3} & \mathbf{R}(\varphi, \theta, \psi) \end{bmatrix} \dot{\mathbf{q}} \tag{2}$$

where the transformation between the angular velocity $\boldsymbol{\omega}$ and the time derivatives of the Euler angles $\dot{\varphi}$, $\dot{\theta}$, and $\dot{\psi}$ is given as [28, 29]:

$$\boldsymbol{R}(\varphi, \vartheta, \psi) = \begin{bmatrix} 0 & c\varphi & s\varphi c\theta \\ 0 & s\varphi & -c\varphi s\theta \\ 1 & 0 & c\theta \end{bmatrix} \tag{3}$$

There are two basic problems in kinematic modeling; the *forward kinematics* is the determination of the EE motion from a given motion of the leg lengths, while the *inverse kinematics* is the determination of the leg lengths motion from a given EE motion. Below these two kinematic problems are formulated at the velocity level.

4. 4. 1. 1. Inverse Rate Kinematics

The closure of each kinematic loop passing through the origin of A and B, and through the six attachment points B_i on the base and the hip attachment points A_i on the EE is given as

$$\boldsymbol{a}_i = \mathbf{p} + {}^{B}\boldsymbol{R}_{A}\,{}^{A}\boldsymbol{a}_i, \quad i=1,...,6, \tag{4}$$

where ${}^{A}\boldsymbol{a}_i$ is the constant position vector of A_i in A. By differentiating eq. (4) with

respect to time, projecting along the joint axis and grouping in a matrix form, one

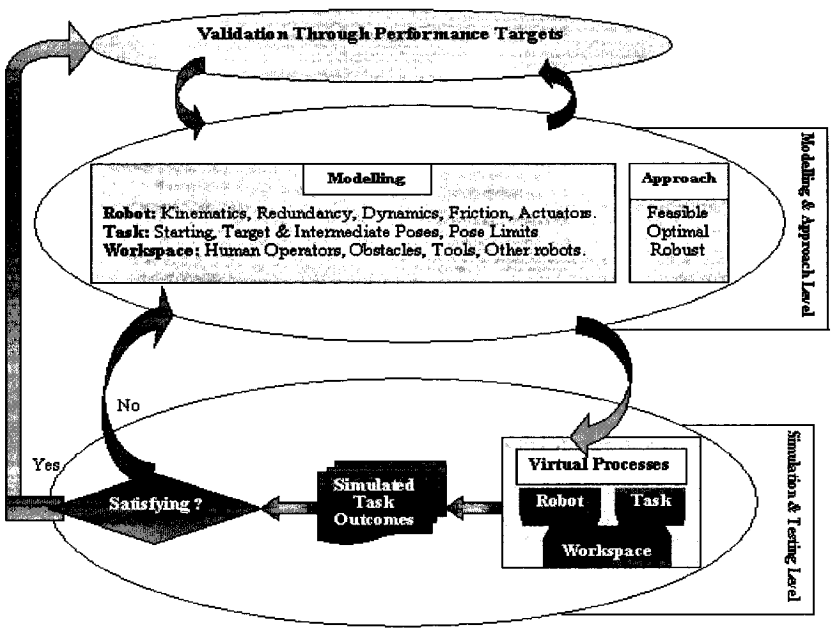


Fig. 4. 1. Overall Off-line Programming Framework of PKMs

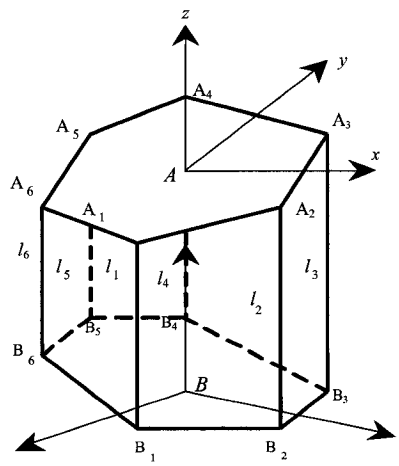


Fig. 4. 2. Geometry of a PKM

gets the inverse kinematic model as:

$$\dot{\boldsymbol{l}} = \boldsymbol{J}^{-1}\,\dot{\boldsymbol{q}} \tag{5}$$

where $\dot{\boldsymbol{l}}$ is the actuated leg length velocity, $\boldsymbol{J}^{-1}$ is the inverse Jacobian matrix given as:

$$\boldsymbol{J}^{-1} = \begin{pmatrix} \boldsymbol{e}_1^T & ({}^B\boldsymbol{R}_A\ {}^A\boldsymbol{a}_1 \times \boldsymbol{e}_1)^T \\ \vdots & \vdots \\ \boldsymbol{e}_6^T & ({}^B\boldsymbol{R}_A\ {}^A\boldsymbol{a}_6 \times \boldsymbol{e}_6)^T \end{pmatrix} \tag{6}$$

and $\boldsymbol{e}_i$ is a unit vector along the ith joint axis.

4. 4. 1. 2. Forward Displacement Model

Unlike the inverse kinematic problem, the forward kinematics is more challenging for general PKMs. The number of solutions depends on the number of configurations the mechanism can be assembled into, for a given set of link lengths. Equation (5) representing the inverse kinematic solution cannot be inverted to find $\boldsymbol{q}$ for a given $\boldsymbol{l}$, because $\boldsymbol{q}$ does not explicitly occur in (5). Numerical methods are generally used to solve the forward kinematic problem [28, 29]. In this paper, a Newton method is used [26].

4. 4. 2. Dynamic Model

Likewise to kinematic modeling, there are two basic problems in dynamic modeling [28, 29]: the *forward and inverse dynamics*. The latter consists to find the

joint force/torque from a given EE motion. The former is to find the EE motion from a given joint input force/torque and initial position, velocity and acceleration conditions. For optimal control and trajectory planning purposes, the dynamic equations are derived using Euler-Lagrange formalism. In Cartesian space, the inverse dynamic model is given a canonical form as:

$$\boldsymbol{\tau} = \boldsymbol{M}_c(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{N}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}_c(\boldsymbol{q}) \tag{7}$$

where $\boldsymbol{M}_c(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{N}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $\boldsymbol{G}_c(\boldsymbol{q})$ are the Coriolis and centrifugal, and gravitational forces, respectively.

4. 4. 2. 1. Actuators Model

It has been shown that actuator dynamics are significant and cannot be neglected for simulation and control [28]. The dynamic equations of actuators yield:

$$\boldsymbol{M}_a\ddot{\boldsymbol{l}} + \boldsymbol{V}_a\dot{\boldsymbol{l}} + \boldsymbol{K}_a\boldsymbol{F} = \boldsymbol{\tau}_m \tag{8}$$

where

$$\boldsymbol{M}_a = \mathrm{M}_a\boldsymbol{I}_{6\times6} = \frac{2\pi}{np}(J_s + n^2J_m)\boldsymbol{I}_{6\times6} \tag{9}$$

$$\boldsymbol{V}_a = \mathrm{V}_a\boldsymbol{I}_{6\times6} = \frac{2\pi}{np}(b_s + n^2b_m)\boldsymbol{I}_{6\times6} \tag{10}$$

$$\boldsymbol{K}_a = \mathrm{K}_a\boldsymbol{I}_{6\times6} = \frac{p}{2\pi n}\boldsymbol{I}_{6\times6} \tag{11}$$

where the various parameters are described in the nomenclature.

In order to included actuator's dynamics, written in joint space, it is necessary to transform the motion eq. (7) into Cartesian space, using eq. (5) as:

$$\boldsymbol{M}_j(\boldsymbol{q})\ddot{\boldsymbol{l}} + \boldsymbol{N}_j(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}_j(\boldsymbol{q}) = \boldsymbol{F} \tag{12}$$

where

$$\boldsymbol{M}_j(\boldsymbol{q}) = \boldsymbol{J}^T\boldsymbol{M}_c(\boldsymbol{q})\boldsymbol{J}\,, \tag{13}$$

$$\boldsymbol{N}_j(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{J}^T\boldsymbol{N}_c(\boldsymbol{q},\dot{\boldsymbol{q}}) - \boldsymbol{J}^T\boldsymbol{M}_c(\boldsymbol{q})\boldsymbol{J}\frac{d(\boldsymbol{J}^{-1})}{dt} \text{ and} \tag{14}$$

$$\boldsymbol{G}_j(\boldsymbol{q}) = \boldsymbol{J}^T\boldsymbol{G}_c(\boldsymbol{q}) \tag{15}$$

with $\boldsymbol{J}$ being the Jacobian, and $\boldsymbol{F}$ the 6-directional column of the generalized forces. The combination of eqs (8) and (12) gives the dynamic model including the actuators in joint space as

$$\overline{\boldsymbol{M}}_j(\boldsymbol{q})\ddot{\boldsymbol{l}} + \overline{\boldsymbol{N}}_j(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \overline{\boldsymbol{G}}_j(\boldsymbol{q}) = \boldsymbol{\tau}_m \tag{16}$$

with

$$\overline{\boldsymbol{M}}_j(\boldsymbol{q}) = \boldsymbol{K}_a\boldsymbol{J}^T\boldsymbol{M}_c(\boldsymbol{q})\boldsymbol{J} + \boldsymbol{M}_a\,, \tag{17}$$

$$\overline{\boldsymbol{N}}_j(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{K}_a\boldsymbol{J}^T\boldsymbol{N}_c(\boldsymbol{q},\dot{\boldsymbol{q}}) + (\boldsymbol{V}_a - \boldsymbol{K}_a\boldsymbol{J}^T\boldsymbol{M}_c(\boldsymbol{q})\boldsymbol{J}\frac{d(\boldsymbol{J}^{-1})}{dt}\boldsymbol{J})\boldsymbol{J}^{-1} \text{ and} \tag{18}$$

$$\overline{\boldsymbol{G}}_j(\boldsymbol{q}) = \boldsymbol{K}_a\boldsymbol{J}^T\boldsymbol{G}_c(\boldsymbol{q}) \tag{19}$$

The bars over uppercase boldface letters are used to include both manipulator and actuator elements. Going back to Cartesian space, the overall PKM dynamic model is given as

$$\overline{\boldsymbol{M}}_c(\boldsymbol{q})\ddot{\boldsymbol{q}} + \overline{\boldsymbol{N}}_c(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \overline{\boldsymbol{G}}_c(\boldsymbol{q}) = \boldsymbol{\tau}_m \tag{20}$$

with

$$\overline{\boldsymbol{M}}_c(\boldsymbol{q}) = \boldsymbol{K}_a \boldsymbol{J}^T \boldsymbol{M}_c(\boldsymbol{q}) + \boldsymbol{M}_a \boldsymbol{J}^{-1}, \tag{21}$$

$$\overline{\boldsymbol{N}}_c(\boldsymbol{q},\dot{\boldsymbol{q}}) = \boldsymbol{K}_a \boldsymbol{J}^T \boldsymbol{N}_c(\boldsymbol{q},\dot{\boldsymbol{q}}) + (\boldsymbol{V}_a\ \boldsymbol{J}^{-1} + \boldsymbol{M}_a^T(\boldsymbol{q})\frac{d(\boldsymbol{J}^{-1})}{dt}) \text{ and} \tag{22}$$

$$\overline{\boldsymbol{G}}_c(\boldsymbol{q}) = \boldsymbol{K}_a \boldsymbol{J}^T \boldsymbol{G}_c(\boldsymbol{q}) \tag{23}$$

It is noteworthy that in the proposed approach one might include contact effort models. Among such models, there are friction and other application-specific forces. Such inclusion is very useful in many practical cases as deflashing and screwing, as it allows avoiding actuator saturation and improving the trajectory planning performance.

4. 4. 2. 2. Discrete-Time Dynamic Model

The approximate state-space discrete-time model of the PKM is deducted from a state-space form of the continuous-time dynamic model. Without loss of generality and for the sake of writing simplicity, the time index and the contact forces are omitted. So, one might write eq. (20) as:

$$\ddot{\boldsymbol{q}} = \overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{q})\boldsymbol{\tau}_m - \overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{q})\left[\overline{\boldsymbol{N}}_c(\boldsymbol{q},\dot{\boldsymbol{q}})\,\dot{\boldsymbol{q}} + \overline{\boldsymbol{G}}_c(\boldsymbol{q})\right] \tag{24}$$

By using state $\boldsymbol{x}_1$, and its time derivative $\boldsymbol{x}_2$, i.e., $\boldsymbol{x} = [\boldsymbol{x}_1^T, \boldsymbol{x}_2^T]^T$ (defined in the nomenclature), eq. (24) is transformed as:

$$\overline{\boldsymbol{M}}_c(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 + \overline{\boldsymbol{N}}_c(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{x}_2 + \overline{\boldsymbol{G}}_c(\boldsymbol{x}_1) = \boldsymbol{\tau}_m \tag{25}$$

In turns, one might rewrite eq. (25) as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{O}_{6\times6} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \boldsymbol{0}_{6\times1} \\ \overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_1)\left[\overline{\boldsymbol{N}}_c(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{x}_2 + \overline{\boldsymbol{G}}_c(\boldsymbol{x}_1)\right] \end{bmatrix} + \begin{bmatrix} \boldsymbol{O}_{6\times6} \\ \overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_1) \end{bmatrix} \boldsymbol{\tau}_m \tag{26}$$

with

$$\overline{\boldsymbol{M}}_c(\boldsymbol{x}_1) = \boldsymbol{K}_a\boldsymbol{J}^T(\boldsymbol{x}_1)\boldsymbol{M}_c(\boldsymbol{x}_1) + \boldsymbol{M}_a\boldsymbol{J}^{-1}(\boldsymbol{x}_1) \tag{27}$$

and

$$\overline{\boldsymbol{N}}_c(\boldsymbol{x}_1,\boldsymbol{x}_2) = \boldsymbol{K}_a\boldsymbol{J}^T(\boldsymbol{x}_1) + (\boldsymbol{V}_a\ \boldsymbol{J}^{-1}(\boldsymbol{x}_1) + \boldsymbol{M}_a\frac{d(\boldsymbol{J}^{-1}(\boldsymbol{x}_1))}{dt}) \tag{28}$$

In order to derive the discrete-time dynamic model of the robot, eq. (25) is written in the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}\boldsymbol{x} - \boldsymbol{D}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{\tau} \tag{29}$$

with

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{O}_{6\times6} \end{bmatrix}, \boldsymbol{D}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{O}_{6\times1} \\ \overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x})\left[\overline{\boldsymbol{N}}_c(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{x}_2 + \overline{\boldsymbol{G}}_c(\boldsymbol{x})\right] \end{bmatrix}, \text{ and}$$

$$\boldsymbol{B}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{O}_{6\times6} \\ \overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}) \end{bmatrix} \tag{30}$$

Now, let's define the sampling period h_k, such that $h_k \le t \le h_{k+1}$, and $\sum_{k=1}^{N} h_k = \mathrm{T}$, with T being the total traveling time and the robot state being assumed invariable between two sampling points k and k+1, and defined as:

$$\boldsymbol{x}(t)=\boldsymbol{x}(h_k),\ \text{for}\ k=0,2,...,N-1,\quad h_k\le t\le h_{k+1} \tag{31}$$

The equivalent discrete-time model to (26) is given as [8]:

$$\boldsymbol{x}_{k+1}=\boldsymbol{F}_d(h_k)\boldsymbol{x}_k-\boldsymbol{D}_d(\boldsymbol{x}_k,h_k)+\boldsymbol{B}_d(\boldsymbol{x}_k,h_k)\boldsymbol{\tau}_k \tag{32}$$

where $\boldsymbol{F}_d, \boldsymbol{D}_d, \boldsymbol{B}_d$ are the discrete equivalents to $\boldsymbol{F}, \boldsymbol{D}, \boldsymbol{B}$ matrices. The relationships between these pair of matrices are developed through a first order approximation as:

$$\boldsymbol{F}_d(h_k)=\boldsymbol{F}_d(k+1,k)=e^{\boldsymbol{F}h_k}=\boldsymbol{I}+\boldsymbol{F}h_k=\begin{bmatrix}\boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6}\\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6}\end{bmatrix} \tag{33}$$

$$\boldsymbol{D}_d(\boldsymbol{x}_k,h_k)=\int_0^{h_k}\boldsymbol{F}_d(h_k-t)\boldsymbol{G}(h_k-t)\,(\boldsymbol{D}(\boldsymbol{x}_k))dt$$

$$=\overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_{1k})\begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{6\times6}\\ h_k\boldsymbol{I}_{6\times6}\end{bmatrix}\left[\overline{\boldsymbol{N}}_c(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k}+\overline{\boldsymbol{G}}_c(\boldsymbol{x}_{1k})\right] \tag{34}$$

$$\boldsymbol{B}_d(\boldsymbol{x}_k,h_k)=\int_0^{h_k}\boldsymbol{F}_d(h_k-t)\boldsymbol{B}(\boldsymbol{x}_k)dt=\begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{6\times6}\\ h_k\boldsymbol{I}_{6\times6}\end{bmatrix}\overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_{1k}) \tag{35}$$

Hence, the discrete-time state-space dynamic model of the PKM is developed with a second order of accuracy for the position and one order for the velocity, to finally be written in the following form:

$$\boldsymbol{x}_{k+1}=\begin{bmatrix}\boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6}\\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6}\end{bmatrix}\boldsymbol{x}_k-\overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_{1k})\begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{6\times6}\\ h_k\boldsymbol{I}_{6\times6}\end{bmatrix}\left[\overline{\boldsymbol{N}}_c(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k}+\overline{\boldsymbol{G}}_c(\boldsymbol{x}_{1k})-\boldsymbol{\tau}_k\right] \tag{36}$$

4. 5. Optimal Time-Energy Trajectory Planning Problem

4. 5. 1. Constraints Modeling

Simulating a robotics task requires taking into account several constraints; structural and geometric constraints, kinematic and dynamic parameter nominal values, such as limits on link lengths, velocities, accelerations, and nominal torques supported by the actuators. Some of these constraints are defined in joint space, while others are in task space.

4. 5. 1. 1. Robots Constraints

- Dynamic state equations: These consists of eq. (36), which is rewritten for later easy use as:

$$\boldsymbol{x}_{k+1}=\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \tag{37}$$

- Link intermediate length limits:

$$\boldsymbol{l}_{\min} \le \boldsymbol{l}_k \le \boldsymbol{l}_{\max}, \text{ with } k=0,2,...,N, \text{ and } \boldsymbol{l}_{\max}=\boldsymbol{\Theta}_{\max}(\boldsymbol{x}) \tag{38}$$

- Singularity avoidance:

Singularities are particular poses in which the robot becomes uncontrollable. Therefore, they are crucial for a successful trajectory planning system. The conditions characterizing singularities are difficult to find analytically for a general PKM, since an analytical expression for the determinant of $\boldsymbol{J}^{-1}$ is not available. Several studies had been dealt with the problem and many singularity avoidance algorithms were

proposed [9-13, 16, 17, 31-33]. A common kinematic performance index related to singularity avoidance is the manipulability measure [27]. Accordingly, by defining the manipulability measure as

$$w(\boldsymbol{x}_{1k}) = \sqrt{\det(\boldsymbol{J}(\boldsymbol{x}_{1k})\boldsymbol{J}^T(\boldsymbol{x}_{1k}))} \tag{39}$$

The following singularity avoidance function can be used:

$$\varpi(\boldsymbol{x}_{1k}) = \frac{1}{\sqrt{\det(\boldsymbol{J}(\boldsymbol{x}_{1k})\boldsymbol{J}^T(\boldsymbol{x}_{1k}))}} \tag{40}$$

- Torque limits:

Non-violation of control torque limits is another major issue for trajectory planning. The required leg forces must continuously be checked for possible violation of the limits as the manipulator moves close to a singular pose. As soon as any leg actuator crosses its limit, the optimal planning procedure has to determine an alternate leg actuation strategy leading to another path on which the actuator forces/torques would be constrained within the limits. In this paper, the robot torques is assumed to belong to a compact and bounded set

$\mathbf{C} \subset \Re^{6N}$, expressed as:

$$\mathbf{C} = \left\{ \boldsymbol{\tau}_k \in \Re^{6N}, \text{such that}: \boldsymbol{\tau}_{\min} \le \boldsymbol{\tau}_k \le \boldsymbol{\tau}_{\max} \ , \ k = 0,...,N-1 \right\} \tag{41}$$

- Sampling period limits:

If the overall robot traveling time T is too small, there may be no admissible solution to the optimal control problem, since the torque constraints bound indirectly the path traversal time. On the other hand, the sampling period h_k must be smaller than the system smallest time mechanical constant in order to prevent the system from being uncontrollable between two control times. In this paper, a tradeoff is made through variation of the sampling period within an admissible domain H defined as:

$$\mathrm{H} = \left\{ h_k \in \Re^+, \text{ such that: } h_{\min} \le h_k \le h_{\max} \right\} \tag{42}$$

4. 5. 1. 2. Task and Workspace Constraints

Task and workspace constraints are basically geometric and kinematic, and allow the determination of the size and shape of the manipulator workspace, which defines the set of poses that can be reached by the EE without singularity or link interference [19, 20]. These constraints are expressed by imposing the EE to pass through a set of specified poses (Fig. 4. 3). These poses are quantified by a set of L pairs ($\mathbf{p}_l$, $\boldsymbol{R}_l$) with $\mathbf{p}_l$ referring to the Cartesian position, and $\boldsymbol{R}_l$ to the orientation of the lth imposed pose on the EE, such that:

$$\left\| \mathbf{p} - \mathbf{p}_l \right\| - \mathrm{T}_{\mathrm{PassThl}_p} = 0, \tag{43}$$

and

$$\left\| \mathbf{vect}(\boldsymbol{R}^T \boldsymbol{R}_l) \right\| - \mathrm{T}_{\mathrm{PassThl}_R} = 0 \tag{44}$$

where $(\mathbf{p}, \boldsymbol{R})$ describes the current computed pose of the EE, while **vect**(.) is the axial vector of its 3×3 matrix argument, and measures the absolute value of the angle of rotation between $\boldsymbol{R}$ and $\boldsymbol{R}_l$. These constraints represent equality constraints and are written for simplicity as:

$$s_1^l(x) = \left\| \mathbf{p} - \mathbf{p}_l \right\| - T_{PassTh_{lp}} = 0, \; s_2^l(x) = \left\| \mathbf{vect}(\boldsymbol{R}^T \boldsymbol{R}_l) \right\| - T_{PassTh_{l_R}} = 0, \; l = 1,...,L \quad (45)$$

The above inequality constraints are written in the following simplified forms:

$$\mathbf{g}_1(x) = l_{min} - \Theta(x) \le 0, \mathbf{g}_2(x) = \Theta(x) - l_{max} \le 0,$$

$$g_3(\tau) = \tau_{min} - \tau \le 0, g_4(\tau) = \tau - \tau_{max} \le 0 \quad (46)$$

For the sake of development simplicity, all inequality constraints will be noted as $\mathbf{g}_j(x, \tau, h) \le 0$, j=1,...,4, regardless if they depend only on state, control variables or both. Hence, we turn up with J = 24 inequality constraints, $2L$ equality constraints (imposed passages), and 12 equality constraints representing state dynamics equations.

4. 5. 2. Performance Index

In general, it is possible to optimize any cost function that has a physical sense. It can be specified according to task and performance targets. The performance index

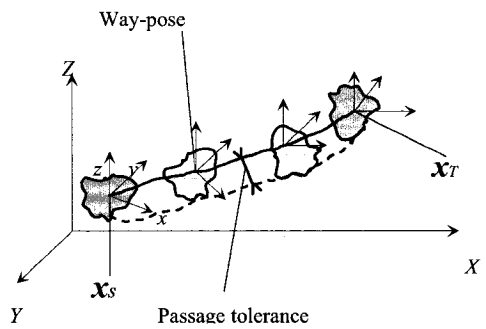


Fig. 4. 3. Illustration of EE passage through imposed poses (positions and orientations) _____ optimal path, ---- feasible path, xs Starting pose, , x_T Target pose,

considered in this paper, relates simultaneously energy consumption, traveling time, and singularity avoidance. For energy criterion, both electric and kinetic energies are optimized. For time criterion, there are two basic ways to perform optimization: the first one assumes a fixed sampling period h and searches for a minimum number N of discretisations. This is equivalent to bring the robot from an initial pose x_S to a final pose x_T, within a minimum number N of discretizations. For the unconstrained case, the time optimal control is basically bang-bang with singularities occurring at the vicinity of the switching function. In serial robots literature, there are several publications following this approach [30]. For strongly non-linear and coupled mechanical systems (like for the PKM at hand), this is simply impractical, even by using symbolic calculation. The second approach fixes the number of discretisations N and varies the sampling periods h_k. This is equivalent to bring the robot from an initial pose x_S to a final pose x_T, within a fixed number of steps N while varying

(minimizing) the sampling periods. In this paper, the number of sampling periods is guessed from an initial feasible kinematic solution. Then the sampling periods and the actuator torques are taken as control variables. For singularity avoidance, it is included through maximization of eq. (40). In continuous-time, the constrained optimal control problem can be stated as follows:

Among all admissible control sequences $\boldsymbol{\tau}(t) \in \mathsf{C}$ *and* $h \in \mathrm{H}$, *that allow the robot to move from an initial state* $\boldsymbol{x}(t_0)=\boldsymbol{x}_S$ *to a final state* $\boldsymbol{x}(t_T)=\boldsymbol{x}_T$, *find those which minimize the cost function* E *:*

$$E = \min_{\substack{\boldsymbol{\tau}(t)\in\mathsf{C} \\ t_0,\, t_T \in \mathrm{H}}} \int_{t_0}^{t_T} \left\{ \left[\boldsymbol{\tau}(t)\boldsymbol{U}\boldsymbol{\tau}^T(t) + \mathfrak{l}_1 + \frac{1}{2}\boldsymbol{x}_2(t)\boldsymbol{Q}\boldsymbol{x}_2^T(t) + \boldsymbol{\delta}\,\varpi(\boldsymbol{x}_1(t)) \right] dt \right\} \quad (47)$$

Subject to constraints (37), (38), (41), (42), (43) and (44).

In the optimization criterion (47), C and H represent the set of admissible torques and the admissible set of sampling periods and $\boldsymbol{U}, \boldsymbol{Q}$, $\mathfrak{l}_1$ and $\boldsymbol{\delta}$ are, respectively, the weight factors associated to the electric energy, kinetic energy, traveling time and singularity avoidance. The corresponding discrete-time optimal control problem consists of finding the optimal sequences $(\boldsymbol{\tau}_0, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_{N-1})$ and $(h_0, h_2, ..., h_{N-1})$, allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a target state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost E_d:

$$\underset{\substack{\tau \in C \\ h \in H}}{\text{Min}} E_d = \left\{ \sum_{k=1}^{N} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \boldsymbol{\iota}_1 + \boldsymbol{x}_{2k} \boldsymbol{Q} \boldsymbol{x}_{2k}^T + \boldsymbol{\delta}\, \boldsymbol{\varpi}\, (\boldsymbol{x}_{1k})] h_k \right\} \tag{48}$$

Subject to:

$$\boldsymbol{x}_{k+1} = f_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k), \quad k = 0,..., N\text{-}1$$

$$\boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \le 0 \ , \ j=1,..,4, \ k = 0,..., N\text{-}1$$

$$\boldsymbol{s}_i(\boldsymbol{x}_k) = 0 , \quad i = 1, ..., 2L , \quad k = 0,..., N$$

4. 6. Nonlinear Programming Formulation

4. 6. 1. Augmented Lagrangian Approach

In the course of solving the constrained non-linear multi-objective optimal control problem in eq. (48), the Augmented Lagrangian (AL) function is used to transform it into a non-constrained one, where the degree of penalty for violating the constraints is regulated by penalty parameters. This method was originated independently by Powell and Hestens [34, 35], and subsequently improved by several authors [24-26]. It basically relies on quadratic penalty methods, but reduces the possibility of ill conditioning of the sub-problems that are generated with penalization by introducing explicit Lagrange multipliers estimates at each step into the function to be minimized, which results in a super linearly convergence iterates. Furthermore, while the ordinary Lagrangian is generally non convex (in the presence of non convex constraints like for the considered problem), AL might be convexified to some extent with a judicious choice of the penalty coefficients [26]. An outline of AL implementation procedure for the case at hand is given at the end of section 5, and a flowchart diagram implementation appears in the appendix. The AL function

transforming the constrained optimal control problem to an unconstrained one is written as:

$$L_{\mu}(\boldsymbol{x}, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = \sum_{k=1}^{N}\left[\boldsymbol{\tau}_k^T \boldsymbol{U} \boldsymbol{\tau}_k + \boldsymbol{\iota}_1 + \boldsymbol{x}_{2k}^T \boldsymbol{Q} \boldsymbol{x}_{2k} + \boldsymbol{\delta}\, \varpi\,(\boldsymbol{x}_{1k})\right] h_k + \sum_{k=0}^{N-1}\left\{\boldsymbol{\lambda}_{k+1}^T (\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)\right\} +$$

$$\sum_{k=0}^{N-1} h_k \left[\sum_{l=1}^{L-1}\sum_{i=1}^{2} \Psi_{\mu_S}(\sigma_k^i, s_i^l(\boldsymbol{x}_k)) + \sum_{j=1}^{J} \Phi_{\mu_g}(\rho_k^j, \boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k))\right] + \sum_{i=1}^{2} h_N\, \Psi_{\mu_S}(\sigma_N^i, s_i^L(\boldsymbol{x}_N)) \quad (49)$$

where the function $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ is defined by the discrete state eq. (37) at the sampling time k, N is the sampling number, $\boldsymbol{\lambda} \in R^{12N}$ designates the adjoint (or co-state) obtained from the adjunct equations associated to state equations, $\boldsymbol{\sigma}, \boldsymbol{\rho}$ are Lagrange multipliers with appropriate dimensions, associated to equality and inequality constraints and $\boldsymbol{\mu}_S$, $\boldsymbol{\mu}_g$ are the corresponding penalty coefficients. The penalty functions adopted here combine penalty and dual methods. This allows relaxation of the inequality constraints as soon as they are satisfied. Typically, these penalty functions are defined by:

$$\Psi_{\mu_s}(\boldsymbol{a}, \boldsymbol{b}) = (\boldsymbol{a} + \frac{\boldsymbol{\mu}_s}{2}\boldsymbol{b})^T \boldsymbol{b} \text{ and } \Phi_{\mu_g}(\boldsymbol{a}, \boldsymbol{b}) = \frac{1}{2\boldsymbol{\mu}_g}\left\{\left\|\text{Max}(0, \boldsymbol{a} + \boldsymbol{\mu}_g\, \boldsymbol{b})\right\|^2 - \left\|\boldsymbol{a}\right\|^2\right\} \quad (50)$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ refer respectively to Lagrange multipliers and the left hand side of equality and inequality constraints. The Karush-Kuhn-Tucker first order optimality necessary conditions require that for $(\boldsymbol{x}_0, \boldsymbol{\tau}_1, h_1, \boldsymbol{x}_1, \boldsymbol{\tau}_2, h_2, ..., \boldsymbol{x}_N, \boldsymbol{\tau}_{N-1}, h_{N-1})$ to be

solution to the problem, there must exist some positive Lagrange multipliers $(\boldsymbol{\lambda}_k, \boldsymbol{\rho}_k)$, unrestricted sign multipliers $\boldsymbol{\sigma}_k$, and finite positive penalty coefficients $(\boldsymbol{\mu}_g, \boldsymbol{\mu}_s)$ such that:

$$\frac{\partial L_\mu}{\partial \boldsymbol{x}} = 0, \frac{\partial L_\mu}{\partial \boldsymbol{\tau}} = 0, \frac{\partial L_\mu}{\partial h} = 0, \frac{\partial L_\mu}{\partial \boldsymbol{\lambda}} = 0, \frac{\partial L_\mu}{\partial \boldsymbol{\rho}} = 0, \frac{\partial L_\mu}{\partial \boldsymbol{\sigma}} = 0, \text{ and}$$

$$\boldsymbol{\rho}_k^T g(\boldsymbol{x}, \boldsymbol{\tau}, h) = 0, \boldsymbol{\sigma}_k^T s(\boldsymbol{x}) = 0, g(\boldsymbol{x}, \boldsymbol{\tau}, h) \le 0 \tag{51}$$

The development of these conditions enables us to derive the iterative formulas to solve the optimal control problem by adjusting control variables, Lagrange multipliers as well as penalty coefficients. However, in equation (37), $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ contains the inverse of the total inertia matrix $\bar{\boldsymbol{M}}_c^{-1}(\boldsymbol{x})$ of the PKM, including struts and actuators, as well as their Coriolis and centrifugal wrenches $\bar{\boldsymbol{N}}_c(\boldsymbol{x}_1, \boldsymbol{x}_2)$. These might take several pages long to display. In developing the first necessary optimality conditions and computing the co-states $\boldsymbol{\lambda}_k$, one has to determine the inverse of the mentioned inertia matrix and its derivatives with respect to state variables. This results in an intractable complexity even by using symbolic calculation.

4. 6. 2. Constrained Linear-Decoupled Formulation

The major computational difficulty mentioned earlier cannot be solved by performing with the original non-linear formulation. Instead, it is solved using a linear-decoupled formulation [36].

Theorem:

Under the invertibility condition of the inertia matrix, the control law defined in the Cartesian space as

$$\boldsymbol{u} = \bar{\boldsymbol{M}}_c(\boldsymbol{x}_1)\boldsymbol{v} + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_1) \tag{52}$$

allows the robot to have a linear and decoupled behavior with a dynamic equation:

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{v} \tag{53}$$

where $\boldsymbol{v}$ is an auxiliary input

This follows simply by substituting the proposed control law (52) into the dynamic model (20). One gets

$$\bar{\boldsymbol{M}}_c(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 = \bar{\boldsymbol{M}}_c(\boldsymbol{x}_1)\boldsymbol{v}$$

Since $\bar{\boldsymbol{M}}_c(\boldsymbol{x})$ is invertible, it follows that $\dot{\boldsymbol{x}}_2 = \boldsymbol{v}$

This brings the robot to have the decoupled and linear behavior described by the following linear dynamic equation written in discrete form as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_{dk}\boldsymbol{x}_k + \boldsymbol{B}(h_k)(\boldsymbol{v}_k) = \boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \tag{54}$$

with $$\boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) = \begin{bmatrix} \boldsymbol{I}_{6\times6} & h_k\boldsymbol{I}_{6\times6} \\ \boldsymbol{O}_{6\times6} & \boldsymbol{I}_{6\times6} \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{6\times6} \\ h_k\boldsymbol{I}_{6\times6} \end{bmatrix} \boldsymbol{v}_k$$

Notice that this formulation reduces drastically the computations, by alleviating us the calculation at each iteration of the inertia matrix inverse and its derivatives with

respect to state variables, which results in ease calculation of the co-states. The non-linearity is however transferred to the objective function.

The decoupled formulation transforms the discrete optimal control problem into finding optimal sequences of sampling periods and acceleration inputs $(h_0, h_2, ..., h_{N-1})$, $(\boldsymbol{v}_0, \boldsymbol{v}_2, ..., \boldsymbol{v}_{N-1})$, allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a final state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost function:

$$E_d^D = \underset{\substack{v \in \mathbf{V} \\ h_k \in \mathbf{H}}}{\text{Min}} \left\{ \sum_{k=0}^{N-1} \left[\left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k}) \boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right]^T \boldsymbol{U} \right. \right.$$

$$\left. \left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k}) \boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right] + t_1 + \boldsymbol{x}_{2k}^T \boldsymbol{Q} \boldsymbol{x}_{2k} + \boldsymbol{\delta} \boldsymbol{\varpi}(\boldsymbol{x}_{1k}) \right] h_k \right\} \quad (55)$$

and satisfying the above mentioned constraints, which mainly remain the same, except for actuator torques, which become

$$\boldsymbol{\tau}_{\min} \le \bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k}) \boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \le \boldsymbol{\tau}_{\max} \quad (56)$$

Moreover, inequality constraints $\boldsymbol{g}_3$ and $\boldsymbol{g}_4$ can be rewritten as:

$$\boldsymbol{g}_3^D(\boldsymbol{x}_k, \boldsymbol{v}_k) = \boldsymbol{\tau}_{\min} - \left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k}) \boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right] \le 0 \quad (57)$$

$$\boldsymbol{g}_4^D(\boldsymbol{x}_k, \boldsymbol{v}_k) = \left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k}) \boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right] - \boldsymbol{\tau}_{\max} \le 0 \quad (58)$$

Similarly to the non-decoupled case, the decoupled problem might be written in the following form:

$$\underset{\substack{v\in V\\ h\in H}}{\text{Min}}\ E_d^D$$

Subject to:

$$\begin{aligned} & \boldsymbol{x}_{k+1}=f_{d_k}^D(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k), \quad k=0,\ldots, N\text{-}1 \\ & \boldsymbol{g}_j^D(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)\le \boldsymbol{0}, \quad j\in\{1,2,\ldots,\mathrm{J}\} \\ & \boldsymbol{s}_i^D(\boldsymbol{x}_k)=0, \quad i=1,\ldots,2L, \quad k=0,\ldots,N \end{aligned} \tag{P}$$

4. 6. 3. Augmented Lagrangian for the Decoupled Formulation

Now, *mutatis mutandis,* the augmented Lagrangian associated to the decoupled formulation (P) is (after removing bars and *c* indexes for writing simplicity):

$$\begin{aligned} L_R(\boldsymbol{x}, \boldsymbol{v}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = \sum_{k=0}^{N-1}\Big\{ \Big[& \big[\boldsymbol{M}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \boldsymbol{N}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \boldsymbol{G}(\boldsymbol{x}_{1k})\big]^T \boldsymbol{U} \\ & \big[\boldsymbol{M}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \boldsymbol{N}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \boldsymbol{G}(\boldsymbol{x}_{1k})\big] + \boldsymbol{\iota}_1 + \boldsymbol{x}_{2k}^T \boldsymbol{Q}\boldsymbol{x}_{2k} + \boldsymbol{\delta}\,\varpi(\boldsymbol{x}_{1k}) \Big] h_k \Big\} + \\ & \sum_{k=0}^{N-1}\Big\{\boldsymbol{\lambda}_{k+1}^T(\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}^D(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k))\Big\} + \sum_{k=0}^{N-1} h_k \Big[\sum_{l=1}^{L-1}\sum_{i=1}^{2} \Psi_{\boldsymbol{\mu}_S}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^{Dl}(\boldsymbol{x}_k)) + \\ & \sum_{j=1}^{\mathrm{J}} \Phi_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^D(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k))\Big] + \sum_{i=1}^{2} h_N\, \Psi_{\boldsymbol{\mu}_S}(\boldsymbol{\sigma}_N^i, \boldsymbol{s}_i^{DL}(\boldsymbol{x}_N)) \end{aligned} \tag{59}$$

where the function $\boldsymbol{f}_{d_k}^D(\boldsymbol{x}_k,\boldsymbol{\tau}_k,h_k)$ is defined by eq. (54) at time k, N is the total sampling number, other parameters appearing in (59) are defined above.

Again, the development of the first order Karush-Kuhn-Tucker optimality

necessary conditions require that for $(\boldsymbol{x}_0, \boldsymbol{v}_1, h_1, \boldsymbol{x}_1, \boldsymbol{v}_2, h_2, ..., \boldsymbol{x}_N, \boldsymbol{v}_{N-1}, h_{N-1})$ to be solution to the problem (P), there must exist some positive Lagrange multipliers $(\boldsymbol{\lambda}_k, \boldsymbol{\rho}_k)$, unrestricted sign multipliers $\boldsymbol{\sigma}_k$, and finite positive penalty coefficients $\boldsymbol{\mu}=(\boldsymbol{\mu}_g, \boldsymbol{\mu}_s)$ such that equations (51) are satisfied for the decoupled formulation.

The co-states $\boldsymbol{\lambda}_k$ are determined by backward integration of the adjunct state equation yielding for $k = N-1,...,0$:

$$\boldsymbol{\lambda}_{k-1} = -2h_k \frac{\partial \left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right]}{\partial \boldsymbol{x}_k} \boldsymbol{U}$$

$$\left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right] - 2\boldsymbol{Q}\boldsymbol{x}_{2k}h_k - \boldsymbol{\delta}\nabla_{\boldsymbol{x}_{1k}} \boldsymbol{\varpi}(\boldsymbol{x}_{1k}) -$$

$$\boldsymbol{F}_d^T \boldsymbol{\lambda}_k - h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \nabla_{\boldsymbol{x}_k} \Psi_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^{Dl}(\boldsymbol{x}_k)) \right] - h_k \left[\sum_{j=1}^{J} \nabla_{\boldsymbol{x}_k} \Phi_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^D(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)) \right] \qquad (60)$$

The gradient of the Lagrangian with respect to sampling period variables is

$$\nabla_{hk} L_{\boldsymbol{\mu}}^D = \left[\left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_k)\boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right]^T \boldsymbol{U} \left[\bar{\boldsymbol{M}}_c(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \bar{\boldsymbol{N}}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \bar{\boldsymbol{G}}_c(\boldsymbol{x}_{1k}) \right] \right.$$

$$\left. + \boldsymbol{x}_{2k}^T \boldsymbol{Q}\boldsymbol{x}_{2k} + \boldsymbol{\iota}_1 + \boldsymbol{\delta}\boldsymbol{\varpi}(\boldsymbol{x}_{1k}) \right] + \sum_{l=1}^{L-1} \sum_{i=1}^{2} \Psi_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^{Dl}(\boldsymbol{x}_k)) + \sum_{j=1}^{J} \Phi_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^D(\boldsymbol{x}_k, \boldsymbol{v}_k)) \qquad (61)$$

The gradient of the Lagrangian with respect to acceleration variables is

$$\nabla_{v_k} L_{\mu}^{D} = 2\bar{M}_C(x_{1k})U^T\left[\bar{M}_C(x_{1k})v_k + \bar{N}_C(x_{1k}, x_{2k})x_{2k} + \bar{G}_C(x_{1k})\right]h_k + Z_k^T\lambda_k +$$

$$h_k\left[\sum_{j=1}^{J}\nabla_{v_k}\Phi_{\mu_g}(\rho_k^j, g_j^D(x_k, v_k, h_k))\right] \tag{62}$$

where $$Z_k = \begin{bmatrix} I_{6\times6} & h_k I_{6\times6} \\ O_{6\times6} & I_{6\times6} \end{bmatrix}[x_k] + \begin{bmatrix} \frac{h_k^2}{2} I_{6\times6} \\ h_k I_{6\times6} \end{bmatrix}[v_k], \quad k = 0,1,...,N-1$$

The development of various related expressions are quite long and not given here. They are detailed in [37].

4. 6. 4. Implementation Issues

4. 6. 4. 1. Initial Solution

To fasten convergence of AL — although it converges even if it starts from an unfeasible solution — a kinematic-feasible solution is defined. It is based on a velocity profile. This solution is divided into three zones, acceleration zone with duration T_1. In this zone, the actuators are assumed to supply an initial force to accelerate the EE until the maximum velocity is reached. Then a constant velocity zone of duration T_2 is achieved. Finally a deceleration zone of duration $T_3 = T_1$ finishes the cycle. The initial time discretisation is assumed an equidistant grid for convenience, i.e.

$$h_k = t_{k+1} - t_k = \frac{t_f - t_0}{N}, \; k = 0,1,...,N-1 \tag{63}$$

4. 6. 4. 2. Search Direction Update

Because the considered problem is of large-scale type, to solve for the minimization step at the primal level of AL, a limited-memory Quasi-Newton-like method is used at each iteration of the optimization process. This method allows the computing of an approximate Hessian matrix by using only the first derivative information, and without need to storing of this approximated Hessian matrix. It performs the second order BFGS (Broyden-Fletcher-Goldfarb-Shano) search technique. It is briefly outlined bellow. For more details, the reader is referred to [18]. At the $(k+1)^{th}$ iteration, set $\boldsymbol{\alpha}_k = \boldsymbol{\nu}_{k+1} - \boldsymbol{\nu}_k$ as the update of the control variable $\boldsymbol{\nu}$, $\boldsymbol{\beta}_k = \nabla_{\nu} L_{k+1} - \nabla_{\nu} L_k$ the update of the gradient and $\mathbf{H}_k^{-1}$ the approximation of the inverse of the Hessian. The inverse of the approximate Hessian $\mathbf{H}_k^{-1}$ can be obtained using the BFGS update formula:

$$\mathbf{H}_k^{-1} = \mathbf{V}_k \mathbf{H}_k^{-1} \mathbf{V}_k^T + \frac{\boldsymbol{\alpha}_k \boldsymbol{\alpha}_k^T}{\boldsymbol{\beta}_k^T \boldsymbol{\alpha}_k}, \text{ with } \mathbf{V}_k = \mathbf{I} - (\boldsymbol{\beta}_k \boldsymbol{\alpha}_k^T)/(\boldsymbol{\beta}_k^T \boldsymbol{\alpha}_k) \tag{64}$$

The following pseudo-code describes the BFGS two-loop iterative procedure used to compute the search direction $\mathbf{H}_k^{-1} \nabla_{\nu} L_k$ efficiently by using the last m pairs of $(\boldsymbol{\alpha}_k, \boldsymbol{\beta}_k)$:

$\mathbf{s} \leftarrow \nabla_{\nu} L_k$
For $i = k-1, k-2, \ldots, k-m$
$\gamma_i \leftarrow \boldsymbol{\alpha}_i^T \mathbf{s} / \boldsymbol{\beta}_i^T \boldsymbol{\alpha}_i$;
$\mathbf{s} \leftarrow \mathbf{s} - \gamma_i \boldsymbol{\beta}_i$;
End (for) (65)
$\mathbf{r} \leftarrow (\mathbf{H}_k^0)^{-1} \mathbf{s}$;
For $i = k-m, k-m, \ldots, k-1$
$\delta_i \leftarrow \boldsymbol{\beta}_i^T \mathbf{r} / \boldsymbol{\beta}_i^T \boldsymbol{\alpha}_i$;

$\mathbf{r} \leftarrow \mathbf{r} + (\gamma_i - \delta_i)\alpha_i$;
End (for)
Stop with result $\mathbf{H}_k^{-1}\nabla_v L_k = \mathbf{r}$
where $(\mathbf{H}_k^0)^{-1}$ is the initial approximation of the inverse of the Hessian matrix. One can set it as: $(\mathbf{H}_k^0)^{-1} = \varsigma_k \mathbf{I}$, with $\mathbf{I}$ is the identity matrix of appropriate dimension, and $\varsigma_k = (\alpha_{k-1}^T \beta_{k-1})/(\beta_{k-1}^T \beta_{k-1})$.

4. 6. 4. 3. Overall Solution Procedure

A systematic procedure flowchart for the augmented Lagrangian implementation appears in the appendix. In this procedure, after selecting robot parameters, task definition, (such as starting, intermediate and final poses), workspace limitations and simulation parameters (block 1), the kinematic unit defines a feasible solution satisfying initial and final poses (block 2). Then the inner optimization loop (block 3) solves for the AL minimization with respect to sampling periods and actuator torques control variables to give the robot dynamic state. This state is then tested within a singularity test unit. If singular, the state is recalculated by going back to the inner optimization loop. If non-singular, a feasibility test is performed. The feasibility is done by testing the norms of all equality and inequality constraints against given tolerances. If the feasibility test fails, restart inner optimization unit. Otherwise, if the feasibility test succeeds, i.e., the current values of penalty are good in maintaining near-feasibility of iterates, a convergence test is made against optimal tolerances. If convergence holds, display optimal results and end the program. If non-convergence,

go further to the dual part of AL (at block 4), to test for constraints satisfaction and update multipliers, penalty and tolerance parameters. If the constraints are satisfied with respect to a first tolerance level (judged as good, though not optimal), then the multipliers are updated without decreasing penalty. If the constraints are violated with respect to a second tolerance level, then one multiplier values and decreases penalty to ensure that the next sub-problem will place more emphasis on reducing the constraints violations. In both cases the tolerances are decreased to force the subsequent primal iterates to be increasingly accurate solutions of the primal problem.

4. 7. Simulation Case Study

4. 7. 1. Description of the 2-DOF Parallel Manipulator Case Study

A simulation program has been implemented using Matlab [38] to test the proposed multi-objective trajectory planning approach on a PKM case study reported in [39]. Preliminary results are encouraging. This PKM consists of two degrees of freedom planar parallel manipulator. The robot kinematic and dynamic models considered have been developed in [37]. A schematic of the manipulator is depicted in Figure 4. 4, where the base is labeled 1 and the EE is labeled 2. The EE is connected to the base by two identical legs. Each leg consists of a planar four-bar parallelogram: links 2, 3, 4, and 5 for the first leg and links 2, 6, 7, and 8 for the second leg. Prismatic actuators actuate the link 3 and 8, respectively. Motions of the EE are achieved by combination of movements of links 3 and 8 that can be transmitted to the EE by the

system of the two parallelograms. Due to its structure, the manipulator can position a rigid body in a 2D space with a constant orientation.

4. 7. 2. Kinematic and Dynamic Analysis

As illustrated in Figure 4. 4, a reference frame $A:(O', x', y')$ is attached to the EE, and a reference frame $B:(O, x, y)$ is attached to the robot base, where O' is the origin of frame A and O the origin of B. To characterize the planar four-bar parallelogram, the chains A_1B_1 and A_2B_2 are considered as shown in Figure 4. 4. Vectors A_{iA} and A_{iB} $(i=1,2)$ define the positions of points A_i in frames A and B respectively. Vectors b_{iB} $(i=1,2)$ define the position of B_i points in frame B. The geometric parameters of the manipulator are $A_iB_i = L$ $(i=1,2)$, $A_1A_2 = 2r$ and $B_1B_2 = 2R$

The position of point O' in the fixed frame B is defined by the vector $(x, y)^T$. The kinematic equations of this manipulator are given by:

$$\boldsymbol{J}_I \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \boldsymbol{J}_x \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \tag{66}$$

where $\boldsymbol{J}_I$ and $\boldsymbol{J}_x$ are respectively, the 2×2 inverse and forward Jacobian matrices of the manipulator, which can be expressed as

$$\boldsymbol{J}_I = \begin{bmatrix} y - y_1 & 0 \\ 0 & y - y_2 \end{bmatrix}, \qquad \boldsymbol{J}_x = \begin{bmatrix} r + x + R & y - y_1 \\ x - r + R & y - y_2 \end{bmatrix} \tag{67}$$

If $\boldsymbol{J}_I$ is non-singular, the Jacobian matrix of the manipulator can be obtained as

$$\boldsymbol{J} = \boldsymbol{J}_I^{-1}\boldsymbol{J}_x = \begin{bmatrix} (r + x - R)/(y - y_1) & 1 \\ (x - r + R)/(y - y_2) & 1 \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \tag{68}$$

Accordingly, it is clear that singularity occurs when one of the following cases holds [12, 13]:

1st case: $|\boldsymbol{J}_I| = 0$ and $|\boldsymbol{J}_x| \neq 0$. This case is known as the type one singularity, and corresponds to the situation where $y=y_1$ or $y=y_2$, i.e., the first or the second leg is parallel to the x-axis.

2nd case: $|\boldsymbol{J}_I| \neq 0$ and $|\boldsymbol{J}_x| = 0$. This case is known as the type two singularity. It corresponds to the pose where four bars of the parallelogram in one of the two legs are parallel to each other. It is analytically expressed by the equality $x+r=R$ for the first leg when x is positive, and $x+R=r$ for the second leg when x is negative.

3rd case: $|\boldsymbol{J}_I| = 0$ and $|\boldsymbol{J}_x| = 0$. This corresponds to the type three singularity for which the two legs are both parallel to the x-axis. This is mainly a design issue as it is characterized by a geometric parameters condition given by:

$$L+r=R \tag{69}$$

The robot dynamic model in the task space of the PKM is obtained through Lagrange formalism as follows:

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix}=\begin{bmatrix}(m_p+\frac{4}{3}m_l)+2(m_s+\frac{m_l}{3})(J_{11}^2+J_{21}^2) & (m_s+\frac{2}{3}m_l)(J_{11}+J_{21})\\ (m_s+\frac{2}{3}m_l)(J_{11}+J_{21}) & m_p+2m_s+\frac{8}{3}m_l\end{bmatrix}\begin{bmatrix}\ddot{x}\\ \ddot{y}\end{bmatrix}+$$

$$\begin{bmatrix}L^2\left[\frac{\dot{x}^2}{\sqrt{\dot{x}^2+\dot{y}^2}}(m_s+\frac{2}{3}m_l)-\sqrt{\dot{x}^2+\dot{y}^2}\right]\left[\frac{J_{11}\sqrt{J_{11}}}{(x+r-R)^{3/2}}+\frac{J_{21}\sqrt{J_{21}}}{(x-r+R)^{3/2}}\right]\\ L^2\left[\dot{x}^2(m_s+\frac{2}{3}m_l)+\frac{\dot{x}\dot{y}}{\sqrt{\dot{x}^2+\dot{y}^2}}\right]\left[\frac{J_{11}\sqrt{J_{11}}}{(x+r-R)^{3/2}}+\frac{J_{21}\sqrt{J_{21}}}{(x-r+R)^{3/2}}\right]\end{bmatrix}+ \quad (70)$$

$$\begin{bmatrix}\left[\frac{\dot{y}(\ddot{x}\dot{y}-(\dot{x}\ddot{y})}{(\dot{x}^2+\dot{y}^2)^{3/2}}+g(m_s+\frac{m_l}{2})\right](J_{11}+J_{21})\\ \left[\frac{\ddot{y}\dot{x}^2-\ddot{x}\dot{y}\dot{x})}{(\dot{x}^2+\dot{y}^2)^{3/2}}(J_{11}+J_{21})+g(m_p+m_l)\right]\end{bmatrix}$$

More details on the derivation of the dynamic model might be found in [37]. Following the streamline developed in previous sections, a discrete-time state-space model associated to of the state eq. (70) is:

$$\boldsymbol{x}_{k+1}=\begin{bmatrix}\boldsymbol{I}_{2\times2} & h_k\boldsymbol{I}_{2\times2}\\ \boldsymbol{O}_{2\times2} & I_{2\times2}\end{bmatrix}\boldsymbol{x}_k-\overline{\boldsymbol{M}}_c^{-1}(\boldsymbol{x}_{1k})\begin{bmatrix}\frac{h_k^2}{2}\boldsymbol{I}_{2\times2}\\ h_k\boldsymbol{I}_{2\times2}\end{bmatrix}\left[\overline{\boldsymbol{N}}_c(\boldsymbol{x}_{1k},\boldsymbol{x}_{2k})\boldsymbol{x}_{2k}+\overline{\boldsymbol{G}}_c(\boldsymbol{x}_{1k})-\boldsymbol{\tau}_k\right] \quad (71)$$

The optimal control problem consists to minimize criterion (47) subject to dynamic eq. (71) and equality and inequality constraints (37-44), and the following specific constraints.

- Workspace limitations:

$$x_{Min} \le x_k \le x_{Max},\ y_{Min} \le y_k \le y_{Max} \text{ for } k = 0,2,...,N \tag{72}$$

- Singularity avoidance:

In our considered case study, the first type singularity constraint might be expressed by:

$$\left|(y_k - y_{1k})(y_k - y_{2k})\right| \ge \varepsilon_1 , \tag{73}$$

whereas the second it is given as:

$$\left|(x_k + \mathrm{sgn}(x_k)(r-R))\right| \ge \varepsilon_2 \tag{74}$$

ε_1 and ε_2 represent small positive tolerances.

The third singularity type concerns mainly the geometric parameters $L, r,$ and R. These parameters are chosen at the design level, such that the equality $L+r=R$ does not hold. The required passage poses is reduced to positioning ones, insofar a constant orientation is assumed during task execution. Typically, one might have:

$$s(x) = \left\| \mathbf{p} - \mathbf{p}_l \right\| - \mathrm{T}_{\mathrm{PassTh}l_p} = 0 \tag{75}$$

The augmented Lagrangian and the associated decoupled formulation are obtained

along with various gradients. These calculations are quite long. The reader is referred to [37] for further details.

4. 7. 3. Simulation Data and Scenario

The following numeric values are used: The EE mass is m_{EE}=200.0 kg, that of each leg is m_l = 570.5 kg, and that of the slider is m_s =100 kg. The platform radius is $r = 0.075m$, $R = 1.2030\,m$ and the strut length $L = 1.9725\,m$. Table. 1 shows the limits of the workspace, actuator torques and sampling periods. For AL, the following parameter values had been taken ω_s=0.5, η_s=0.5, α_w=α_η=0.4, β_w=β_η=0.4, w_0= η_0= η_{10}=10^{-2}, w^*= η^*= η_1^*=10^{-5}, γ_1=0.25, γ_2=1.2, υ=0.01, $\bar{\upsilon}$=0.3. The initial Lagrange multipliers $\boldsymbol{\rho}_0$, $\boldsymbol{\sigma}_0$ components are set to zero. The singularity weight is $\delta = 1$. The maximum value for $\boldsymbol{\delta}_{Max}$=$10^{42}$, and the minimum value for $\boldsymbol{\delta}_{Min}$=$10^{-42}$. The scenario consists of a straight line trajectory from an initial Cartesian state position $x_0 = -0.7$, $y_0 = -0.1$ to a final position $x_T = 0.7$, $y_T = -1.6$ (in meters). The initial and final linear and angular velocities are equal to zero.

The maximum velocity is 0.2m/sec and maximum acceleration is 2 m/sec^2. The maximum allocated time for this trajectory is 10 sec. In the presented simulations, the focus is on time-energy constrained trajectory planning by the augmented Lagrangian, more kinematic related performance evaluation and design for a similar case study might be found in [37, 39]. Typically, four objectives are considered:

- Compare robot trajectories for different values of the weights $\boldsymbol{U}$, $\boldsymbol{Q}$, ι_1 and $\boldsymbol{\delta}$.
- Assess the effects of the dynamic parameters changes on the augmented Lagrangian sensitivity and on the behavior of the PKM.
- At which precision vs. time consumption, the augmented Lagrangian achieves passage satisfaction through imposed poses?
- To what extent the number of inner and outer iterations of AL impacts PKM performance *vs.* CPU Time?

To start, Figure 4. 5 shows the velocity profile used to initialize the augmented Lagrangian. Figures 4. 6 and 4. 7 show the simulation results for both initial kinematic and augmented Lagrangian solutions. In part (a) of these figures, the first plot from the top shows the displacement along x-axis of the end-effector point of operation. The second plot shows the displacement along y-axis of the end-effector point of operation. The third shows the instantaneous values of consumed time to achieve the trajectory. In part (b), the first and second plots from the top show the instantaneous variations of joint torques, while the third one shows the instantaneous values of the consumed energy. It is noteworthy that although the initial solution is kinematically feasible, when the corresponding torques is computed considering the dynamic model and forces, one gets shortly torque values outside the admissible domain resulting in high values for the energy cost. With the augmented Lagrangian, however, with four inner and 7 outer iterations, the variations of the energy consumption are increasing smoothly and monotonously. Figure 4. 8 displays the

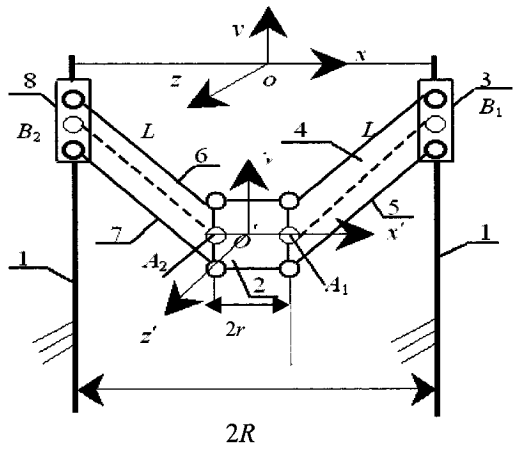


Fig. 4. 4. A schematic representation of a planar parallel manipulator

simulation outcomes for the only energy criterion (i.e. the time weight is set to zero, so the sampling period is kept constant). One gets a 34% faster trajectory with time-energy criterion (Fig. 4. 7), as compared to a trajectory computed with only minimum energy criterion (Fig. 4. 8). As for imposed passages through pre-specified poses, the same scenario as above is simulated, while constraining the EE to pass through the following positions: (0.0, -1.4), (0.4, -1.1), (0.5, -1.0), all in meters. Figure 4. 9 shows the trajectory corresponding to passage through imposed poses. With 10 dual iterations and 7 primal ones, one gets a precision of 7.10^{-4}, which confirms the well-known constraints satisfaction performance of AL for constrained optimization problems as compared to its counterparts like penalty methods. Furthermore, we observe that the proposed variational approach has not only been successful in finding singularity-free trajectory, but also the obtained trajectories are optimal in time and energy minimization. To analyze with respect to AL parameters, Table 4. 2. shows comparison of results for different simulation parameters of AL, where NDisc

is the number of discretisations, NPrimal is the number of inner optimization loops, NDual is the number of outer optimization loops, $t_T=\sum_{k=1}^{N} h_k$ is the total traveling time, Energy = $\sum_{k=1}^{N}[(\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \boldsymbol{x}_{2k} \boldsymbol{Q} \boldsymbol{x}_{2k}^T)]$ is the consumed electric and kinetic energy, and APEq and APIneq for Achieved Precision for Equality and Inequality constraints satisfaction, respectively. The values shown for the total traveling time t_T, Energy, and AP correspond to those computed for the last outer iteration.

Table 4. 1. Limits of workspace, actuator torques and sampling periods

Parameter	x-coordinate (m)	y-coordinate (m)	τ_1 (N)	τ_2 (N)	h (sec)
max	0.8	-0.720	550	700	0.7
min	-0.8	-1.720	-550	-700	0.005

Table 4. 2. Convergence history of Minimum Time-Energy Planning with Augmented Lagrangian

NDisc	NPrimal	NDual	CPU(sec)	t_T (sec)	Energy(J)	APEq	APIneq
10	4	7	12.62	9.50	6874.37	3.10^{-3}	3.10^{-3}
20	4	7	29.25	8.01	5156.28	10^{-3}	10^{-3}
20	5	10	35.54	7.31	4910.39	5.10^{-4}	4.10^{-4}
30	5	10	69.83	6.07	4010.23	2.10^{-5}	3.10^{-5}

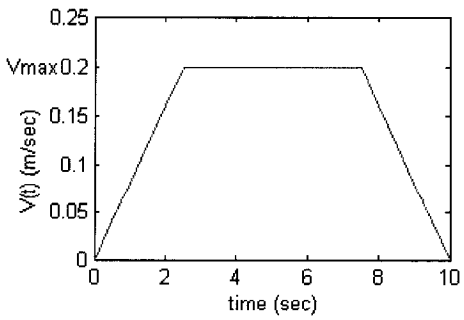
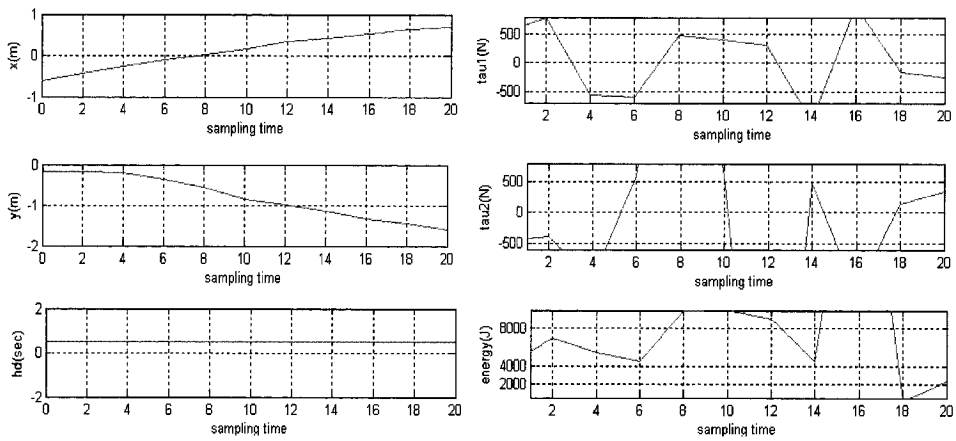
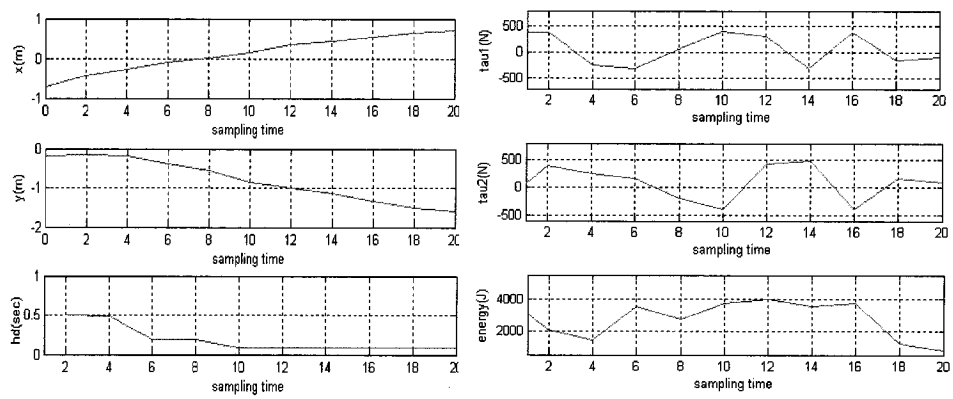


Fig. 4. 5. Initial solution, trapezoidal velocity profile



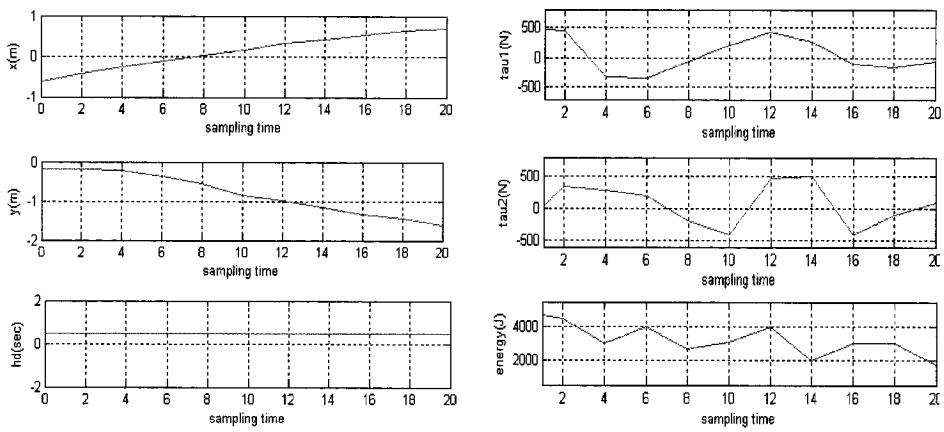
(a) (b)

Fig. 4. 6. Kinematic simulation results
(a)- Variations of x, y, coordinates of the EE, and sampling periods
(b)- Variations of torques τ_1 , τ_2 , and energy



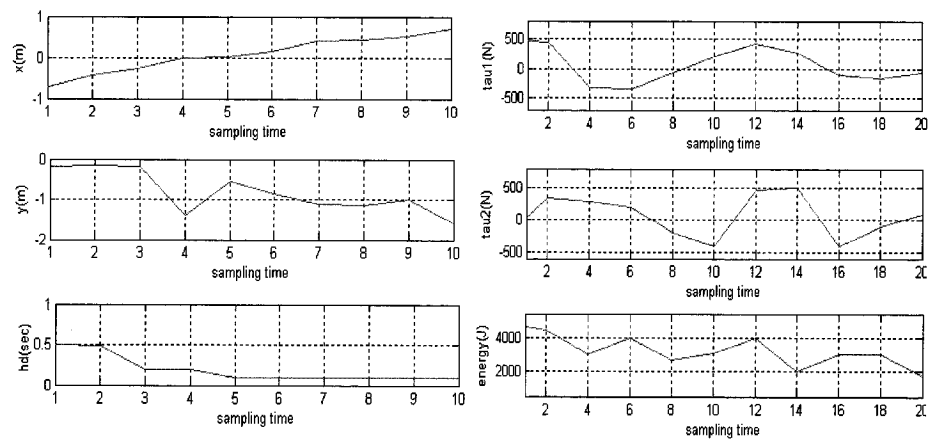
(a) (b)

Fig. 4. 7. Simulation results with the augmented Lagrangian
(Minimum Time-Energy)
(a)- Variations of x, y, coordinates of the EE, and sampling periods
(b)- Variations of torques τ_1 , τ_2 , and energy



(a) (b)

Fig. 4. 8. Augmented Lagrangian (Minimum Energy)
(a)- Variations of x, y, coordinates of the EE, and fixed sampling periods
(b)- Variations of torques τ_1 , τ_2 and energy



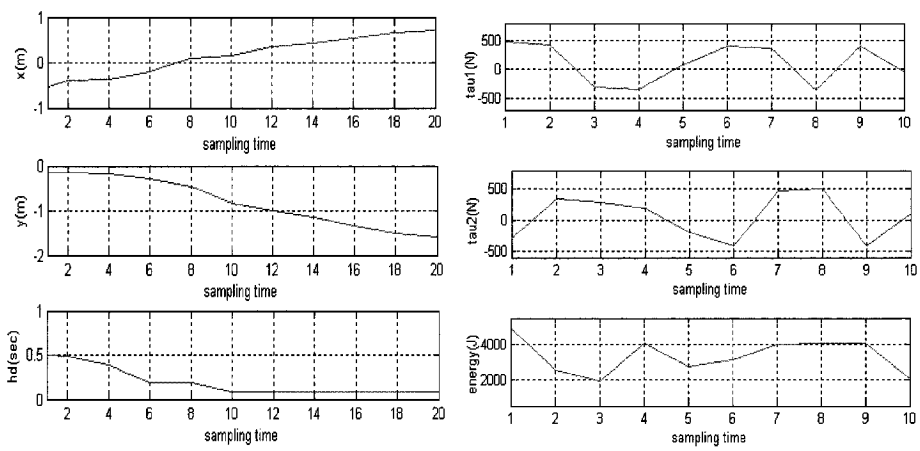
(a) (b)

Fig. 4. 9. AL Minimum Time-Energy with imposed passage through Cartesian positions (0.0, -1.4), (0.4, -1.1), and (0.5, -1.0)
(a) Variations of x, y, coordinates of the EE, and sampling periods
(b)- Variations of torques τ_1 , τ_2 and energy

4. 7. 4. Sensitivity Analysis

The optimal time-energy control considered so far is dependent on the values of the dynamic parameters of the PKM. As PKMs are strongly non-linear and coupled mechanical systems, several of these parameters such as inertial parameters are known only approximately or may change. So, a sensitivity analysis [40] is necessary to assess how robust the proposed approach to the parameter changes is.

This is performed through varying the value of the EE mass. On Figure 10, it is shown the AL simulation with modified EE mass as m_{EE}=300.0kg. One notices that the needed actuator torques and necessary energy and time to achieve the same task are higher, especially at the beginning.



(a) (b)

Fig. 4. 10. AL, Minimum Time-Energy, Modified mass of EE, m_{EE}=30.0 kg
(a)- Variations of x, y, coordinates of the EE, and sampling periods
(b)- Variations of torques τ_1 , τ_2 and energy

4. 8. Conclusions and Discussions

The basic contribution of this paper is the formulation and resolution of the trajectory-planning problem of parallel kinematic machines using a variational calculus framework. This is performed by considering robot kinematic and dynamic models, while optimizing time and energy necessary to achieve the trajectory, avoiding singularities, and satisfying several constraints related to the robot, task and workspace. The robot dynamic model includes the EE, struts and actuators models. The augmented Lagrangian algorithm is used to solve the resulting non-linear and non convex optimal control problem. This optimization technique is used along with a decoupled and linearized formulation of the original problem, permitting the ultimate benefit of easing the computation of the co-states and other variables necessary to perform optimization. Although it is task and algorithm parameter settings dependent, the computational time is drastically reduced when the decoupled and linearized formulation is used. It has been shown that the proposed approach performs better in optimizing traveling time and actuator torques than kinematic only based schemes. Furthermore, the proposed trajectory planning is robust to dynamic parameters changes. This in fact is due to the ability of the Augmented Lagrangian to cope with numerical ill-conditioning problems, as compared to other optimization techniques like penalty methods. Moreover, a major advantage of this approach is that one can introduce any type of constraints related to the robot, task or environment, like obstacles or link interference avoidance, by deriving the corresponding constraint expressions and adding them naturally in the Lagrangian in

order to have them included in the trajectory planning system. These issues are now being incorporated in an ongoing work.

Acknowledgment

The authors gratefully thank the Natural Science and Engineering Research Council of Canada (NSERC) for supporting this work under grants ES D3-317622, RGPIN-203618, RGPIN-105518 and STPGP-269579.

Appendix: Augmented Lagrangian Algorithm

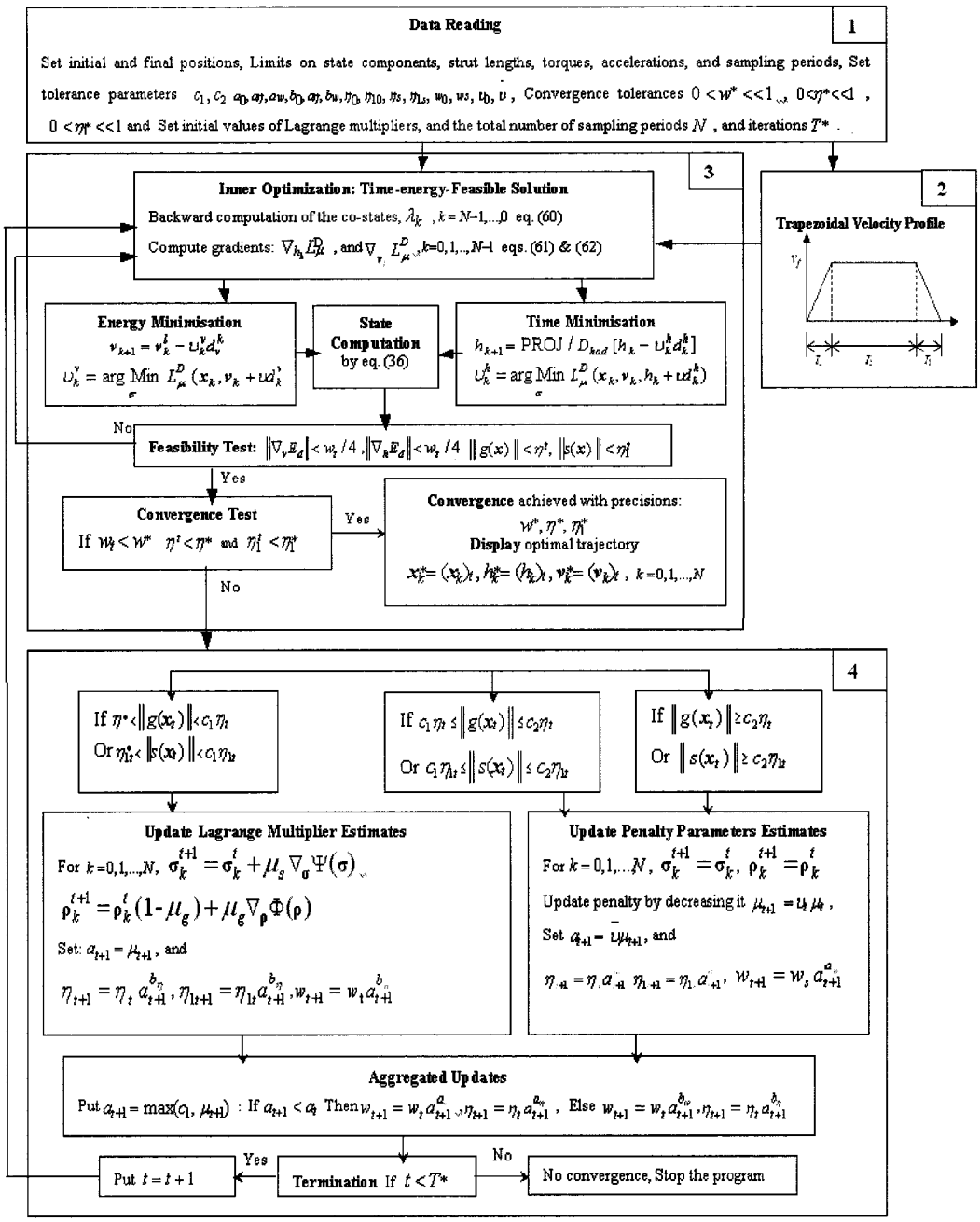


Fig. 4. 11. Flowchart for Augmented Lagrangian algorithm function and operation

4. 9. References

[1] V. E. Gough, Contribution to discussion of papers on research in automobile stability, control and type performance, in *Proc. Auto Div. Inst. Mech. Eng. Part D, J. Automob. Eng.* p 392-395, 1956.

[2] D. Stewart, A platform with six degree-of freedom, in *Proc. Inst. Mech. Eng.*, v 180, p 371-386, 1965.

[3] J.P. Merlet, *Parallel Robots*, Kluwer Academic Publisher: Dordrecht, the Netherlands, 2000.

[4] I.T. Pietsch, M. Krefft, O.T. Becker, C.C. Bier, J. Hesselbach, How to reach the dynamic limits of parallel robots? An autonomous control approach, *IEEE Trans on Aut. Sc. and Eng.* v 2, n 4, p 369-380, 2005.

[5] C. Ahrikencheikh and A. Seireg, *Optimized-motion planning*, John Wiley & Sons, 1994.

[6] A. Khoukhi and A. Ghoul, On the Maximum dynamic stress search for a robot manipulator, *Robotica*, v 22, n 5, p 513-522, 2004.

[7] A. Khoukhi, An optimal time-energy control design for a prototype educational robot, *Robotica,* v 20, n 6, p 661-671, 2002.

[8] M. Vukobratovic, D. Stokic, Is dynamic control needed in robotics system? and if so to what extent? *Int'l J. of Rob. Res.* v 2, n 2, p 18-35, 1983.

[9] B. Dasgupta, T.S. Mruthyunjaya, The Stewart platform manipulator, a review, *Mechanism and Machine Theory,* v 35, n 1, p 15-40, 2000.

[10] L. Baron, J. Angeles, The kinematic decoupling of parallel manipulators under joints-sensor redundancy, *IEEE Trans. on Robotics and Automation*, v 16, n 1, p 12-19, 2000.

[11] L. Baron, J. Angeles, The direct kinematics of parallel manipulators under joints-sensor data, *IEEE Trans. on Robotics and Automation*, v 16, n 6, p 644-651, 2000.

[12] O. Ma, J. Angeles, Architecture singularities of parallel manipulators, *Int'l Jou. of Robotics and Automation,* v 7, n 1, p 23-29, 1992.

[13] C. Gosselin, Parallel computational algorithms for the kinematics and dynamics of planar and spatial parallel manipulators, *ASME, J. of Dyn. Sys, Meas. and Cont.*, v 118, n 1, p 22-28, 1996

[14] D. A. Kumar, C. I. Ming, Y. S. Huat and Y. Guilin, Workspace generation and planning singularity-free path for parallel manipulators, *Mechanism and Machine Theory*, v 40, n 7, p 776-805, 2005.

[15] V. Mermertas, Optimal design of manipulator with four-bar mechanism, *Mechanism and Machine Theory*, v 39, n 5, p 545-554, 2004.

[16] S. Bhattacharya, H. Hatwal, A. Ghosh, Comparison of an exact and approximate method of singularity avoidance in platform type parallel manipulators, *Mech. Mach. Theory* v 33, n 7, p 965-974, 1998.

[17] S. Sen, B. Dasgupta, A. K. Mallik, A Variational Approach for singularity-free path-planning of parallel manipulators, *Mec. and Machine Theory,* v 38, n 11, p 1165-1183, 2003.

[18] E. Polak, *Optimization, algorithms and consistent approximation*, Springer-

Verlag, New York, 1997.

[19] M Hay, J. A. Snyman, Methodologies for the optimal design of parallel manipulators, *Int'l J. for Numerical Methods in Engineering*, v 59, n 1, p 131-152, 2004.

[20] M Hay, J. A. Snyman, A multi-level optimization methodology for determining the dexterous workspaces of planar parallel manipulator, *Structural and Multidisciplinary Optimization*, v 30, n 6, p 422-427, 2005.

[21] A. Khoukhi, L. Baron, M. Balazinski, Programmation dynamique temps-énergie minimum des robots parallèles par Lagrangien augmenté, *CCToMM Symposium on Mechanics, Machines and Mechatronics,* Canadian Space Agency, Saint Hubert, Canada, p 147-148, May 26-27, 2005.

[22] A. Khoukhi, L. Baron, M. Balazinski, A decoupled approach to time-energy trajectory planning of parallel kinematic machines, Proceedings of 16th CISM-IFFToMM Robotics and Manipulators Symposium Warsaw, Poland, June 20-24, p 179-186, 2006.

[23] A. Khoukhi, Dynamic modelling and optimal time-energy off-line programming for mobile robots, a cybernetic problem, *Kybernetes,* v 31, n 5-6, p 731-735, 2002.

[24] T. Rockafellar, Lagrange multipliers and optimality, *SIAM Review*, v 35, n 2, p 183-238, 1993.

[25] D. P. Bertsekas, *Non-linear programming*, Athena Scientific, 1995.

[26] J.B.Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms*, Springer Verlag, Berlin, New York, 1993.

[27] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization,* Addison-Wesley, 1991

[28] J. Angeles, *Fundamentals of robotic mechanical systems, theory, methods and algorithms.* Second Ed., Springer Verlag, 2003.

[29] K. Harib, K. Srinivasan, Kinematic and dynamic analysis of stewart platform-based machine tool structures, *Robotica,* v 21, n 5, p 541-551, 2003.

[30] H. Jurgen, Mc. John, Determination of minimum-time maneuvers for a robotic manipulator using numerical optimization methods, *Mechanics of Structures and Machines*, v 27, n 2, p 185-201, 1999.

[31] H. Yu, J. Wang, G. Duan and L. Sun, Approach for singularity avoidance path planning of parallel robots, *Proc. of the Eleventh World Congress in Mechanism and Machine Science*, p 1961 -1965, 2004.

[32] D. N. Nenchev, S. Bhattacharya, and M. Uchiyama, Dynamic analysis of parallel manipulators under singularity-consistent parameterization, *Robotica,* v 15, n 4, p 375-384, 1997.

[33] C. Innocenti and V. Parenti-Castelli. Singularity-free evolution from one pose to another in serial and fully-parallel manipulators, *ASME J. of Mechanical Design,* v 120, n 1, p 73-79, 1998.

[34] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization (R. Fletcher, Ed.), Academic Press London, p 283-298, 1969.

[35] M. R. Hestens, Multiplier and gradient methods, *J. Opt. Theory and App.* v 4, p 303-320, 1969.

[36] A. Isidori, *Non linear control systems,* Springer; 3rd Edition, London, UK, 1995.

[37] A. Khoukhi, L. Baron, M. Balazinski, *Constrained multi-objective off-line programming of parallel kinematic machines*, Technical Report, *CDT-P2877-05,* 70 pages, École Polytechnique of Montreal, 2007.

[38] Available: www.mathworks.com/products/matlab/

[39] X. J. Liu, Q. M. Wang, J. Wang, Kinematics, Dynamics and dimentional synthesis of a novel 2- dof translational manipulator, *J. of Intelligent and Robotic Systems,* v 41, p 205-224, 2004.

[40] Andrea Saltelli , K. Chan , E. M. Scott (Eds), *Sensitivity analysis*, John Wiley & Sons, 2000.

Figure Captions:

Figure 1. Overall Off-line Programming Framework of PKMs

Figure 2. Geometry of a PKM

Figure 3. Illustration of EE passage through imposed poses (positions and orientations)

Figure 4. A schematic representation of the planar parallel manipulator

Figure 5. Initial solution, velocity profile

Figure 6. Kinematic simulation results

Figure 7. Simulation results with the augmented Lagrangian (Minimum Time-Energy)

Figure 8. Augmented Lagrangian (Minimum-Energy)

Figure 9. AL Minimum Time-Energy with imposed passage through Cartesian positions

Figure 10. Disturbed AL (Minimum Time-Energy) (modified mass of EE m_{EE}=300.0 kg,)

Figure 11. Flowchart for Augmented Lagrangian algorithm function and operation

List of Tables:

Table 1. Limits of the workspace, actuator torques and sampling periods

Table 2. Convergence history of the Minimum Time-Energy Planning with Augmented Lagrangian

Vitae

DR. AMAR KHOUKHI

Dr. Amar Khoukhi received his Advanced Diploma of University Studies in Operation Research from University of Paris-IX Dauphine, France in 1985, a Master Degree in Software Engineering from Superior School of Telecommunications Engineers, France in 1991, and a PhD degree in 2007 in robotics, from École Polytechnique, Montreal, Canada.

Over the last several years, Dr. Khoukhi was an associate research and lecturer at Control Systems and Automation Dept., Superior School of Electronics Engineers, Paris, France. From 1998 to 2001 he was a faculty member at the Computer Science Dept., University of Algiers, Algeria. From 2004 to 2007 he held an NSERC Post-Graduate Scholarship and worked as a lecturer and associate research in Mechanical Engineering Dept. at École Polytechnique of Montreal. His research interests include robot trajectory planning, applied optimization, and intelligent systems.

DR. LUC BARON

Dr. Luc Baron received his B.Eng. and M.A. Sc. degrees from École Polytechnique, Montreal, Canada, in 1983 and 1985, respectively, and his Ph.D. degree from McGill University, Montreal, Canada, in 1997, all in mechanical engineering.

From 1986 to 1993, he held cross appointments as faculty lecturer at École Polytechnique, the engineering faculty of University of Montreal, and as Research Engineer at Walsh Automation Inc. Since 1997, Dr. Baron is Professor in the Department of Mechanical Engineering at École Polytechnique. His current research interests are in the field of robotics and industrial automation, kinematics, synthesis and control of manipulators. Dr. Baron is a member of IEEE, an Associate Member of ASME and a registered engineer in Quebec.

DR. MAREK BALAZINSKI

Dr Balazinski is a specialist in the area of manufacturing, fuzzy logic and genetic algorithms. He has obtained his bachelor, master and doctoral degrees from the Technical University of Carcow in Poland. Over the last several years, he has focused on metal cutting of hard and difficult to machine materials and on application of fuzzy logic and genetic algorithms in manufacturing.

Dr Balazinski is currently professor and head of the Manufacturing Section in the Department of Mechanical Engineering at École Polytechnique of Montréal. Over the last ten years, he has published over 100 research papers and he is an active member of CIRP (The International Academy for Production Engineering).

CHAPITRE 5

HIERARCHICAL NEURO-FUZZY OPTIMAL TIME TRAJECTORY PLANNING FOR REDUNDANT MANIPULATORS

Amar KHOUKHI, Luc BARON, Marek BALAZINSKI
Department of Mechanical Engineering,
École Polytechnique de Montréal,
C. P. 6079, Succ. CV, Montreal, QC, Canada H3C 3A7
(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

and

Kudret DEMIRLI
Department of Mechanical and Industrial Engineering
Concordia University,
1455 de Maisonneuve Blvd. W., EV4.173, Montreal, QC, Canada H3G 1M8
demirli@encs.concordia.ca

5. 1 Abstract - In this paper, the problem of minimum-time trajectory planning is studied for a three degrees-of-freedom (3-DOF) planar redundant manipulator using a hierarchical hybrid neuro-fuzzy system[1]. A first pre-processing step involves two components. The first component is NeFIK (for Neuro-Fuzzy Inverse Kinematics),

[1] Preliminary results of this work are published in the Proceedings of NAFIPS'06, June 3-6, 2006, Concordia University, Montreal, Quebec, Canada.

a neuro-fuzzy network designed to learn and solve the inverse kinematics problem.The second one is an optimal time control problem involving robot dynamics and constraints. This latter is solved using an Augmented Lagrangian (AL) technique. These pre-processing steps allow generating a large dataset of minimum-time singularity-free trajectories, covering mostly of the robot workspace. Upon this dataset, an objective neuro-fuzzy inference system is built to learn and capture the robot minimum time dynamic behavior. Once this system is trained and optimized, it is used in a generalization phase to achieve online planning. Simulation results showing the effectiveness of the proposal are presented and presented.

Key words: Redundant Manipulators, Trajectory Planning, Minimum-Time Control, Augmented Lagrangian, Subtractive Clustering, Hierarchical Neuro-Fuzzy Systems.

5. 2. Introduction

5. 2. 1. Related Works

A great advantage of robots is their ability and flexibility to rearrange themselves for new tasks. However, utilization of robot's flexibility presupposes effective motion planning. This is aimed at generating trajectories for a specific task, according to a set of desired performance criteria. The task is usually specified in terms of end-effector (EE) motion, which results in a geometric path to be followed according to a given time law. Many researchers had addressed this problem and several algorithms and performance criteria were proposed [1-4]. The most popular

performance criterion is the traveling time minimization. This is due for the obvious reason of production targets. However, the major drawback of this control criterion is its bang-bang character as it produces non-smooth trajectories. Moreover, tracking a planned minimum-time trajectory had shown to lead to large joint vibrations, and overshoot of the nominal torque limits [5, 6]. Furthermore, as robot tasks are demanding in precision and diversity, these tasks execution might be difficult to achieve because of the strong non-linearities of the robot dynamics involving components such as Coriolis and centrifugal wrenches, inertia parameters, and friction forces [2]. The coupling between robot's axes makes even more difficult the design of a trajectory planning or a control system of the robot [2-4]. As a matter of fact, conventional fixed gain, linear feedback controllers are not capable of effectively controlling the movements of multi-joint robot manipulators under different distance, velocity, and load requirements. Through the use of non-linear feedback, approaches like computed torque and robust $H_{Infnity}$ methods provide better compensation for the dynamic interactions present in various robot motions [7, 8]. But, these approaches require complete, non-linear dynamic models describing the robot manipulator which are difficult to be accurately modeled and implemented in real-time.

To overcome these difficulties, several researchers have used soft computing techniques. Genetic algorithms are increasingly applied to motion planning and dynamic control of robotic systems [9-11]. Neural networks and fuzzy control perhaps constitute the most applied artificial intelligence fields in control systems of

industrial manipulators and mobile robots. Compared to conventional controllers, soft computing based-controllers are capable of tackling highly complex dynamics without explicit model dependence and identification. Moreover, these artificial intelligence techniques have added a new dimension to many engineering fields to study, and have effectively been applied to everything from voice and image recognition to toasters and automobile transmissions [12-22].

Neural networks are well known for their learning abilities [13, 22]. They are robust and fault tolerant. However, their lack of interpretability and non use of linguistic knowledge capabilities, make fuzzy systems a very suitable complements. Indeed, fuzzy systems are successful in using human skills and thinking processes within a machine, by acquiring their knowledge from an expert who encodes his knowledge in a series of IF/THEN rules [20]. However, the problem arises when a system has many inputs and outputs, obtaining a rule base for such a large system is often too difficult. This dilemma opened the way to hybrid neuro-fuzzy systems, which are widely recognized as a remedy for the above mentioned gaps [12-21].

Recently, several methods had been considered for trajectory design and robot motion planning using neuro-fuzzy techniques [14-17]. Some of theme involves the use of a conventional control algorithm, e.g., PD or PID in controlling a multi-joint robot manipulator [15, 23]. The conventional control brings the system close to the desired state and the neuro-fuzzy learning mechanism then compensates for the remaining error. Other systems use only learning algorithms to execute the control. These learning controllers are computationally efficient after training and had been

shown to have very good generalization capabilities, provided that a large enough dataset points representing the system's behaviour is given [13-17].

5. 2. 2. Approach Outline

In this context, this paper presents a multi-level neuro-fuzzy system to minimum-time trajectory planning of a 3-DOF planar redundant manipulator. Recently, in [24], a data-driven neuro-fuzzy system has been considered to online motion planning of a mobile robot, where the dataset was generated from real time measurements. In this paper, the dataset is generated from the outcomes of an offline planning. The best possible robot trajectories can be obtained through a constrained optimal control of full, non-linear robot dynamic and kinematic models as well as workspace limitations and task requirements. Simulated data provide reliable training and comparison for the neuro-fuzzy system. Furthermore, they allow testing of specific parameter's impact on the overall performance, by changing just the specified parameter, something, which is not possible with a physical robot, (because they are never two identical situations in the real world!!). The proposed system is composed of five steps as illustrated in Fig. 5.1.

It starts by reading data related to link lengths and masses, initial and final points, and limits on actuator torques, joint angles, rates, accelerations and jerks, as well as task requirements, workspace limitations and simulation parameters. The second step is NeFIK (standing for Neuro-Fuzzy Inverse Kinematics), a neuro-fuzzy network built to learn and solve the Inverse Kinematics Problem (IKP). IKP function

consists of transforming Cartesian coordinates (given within the task space) into joint coordinates (given in the actuation joint space) of the manipulator. It is trained using forward kinematics of the robot. This results in converting the initial and target Cartesian coordinates into joint coordinates. The third step is an initialization unit defining a feasible solution satisfying boundary constraints in joint angles, velocities, accelerations and jerks. A cycloidal profile is used to this end (see Appendix A). The obtained solution is put into the robot's inverse dynamic model to get the corresponding torques. In a fourth step, the generated near minimum-time trajectory outcomes are used to initialize a minimum time control problem under limits on EE poses, as well as constraints on actuator torques. This problem is discretized and an augmented Lagrangian (AL) technique is used to solve it. These offline pre-processing steps are done as many times as possible to cover mostly of the robot workspace. This yields to an input/output dataset, where the inputs are the three joint angles and rates, whereas the outputs are the corresponding actuator torques. The fifth step implements a subtractive clustering algorithm to partition the so-generated dataset into clusters. By the same token, the generated clusters allow initialisation of membership function parameters and number of rules of a Tsukamoto neuro-fuzzy system, named NeFOTC (for Neuro-Fuzzy Optimal Time Controller). The last step consists of optimizing NeFOTC in order to learn and capture the dynamic minimum-time behaviour of the robot.

The whole purpose of the proposed approach is that from a pure but representative model and offline minimum-time trajectory planning outcomes, a

neuro-fuzzy system can be built. Once the neuro-fuzzy structure (rules number and premise and consequence membership function parameters) is identified and optimized, it might be used in a generalization phase to achieve online planning with reasonable time delays.

In section 5.3, the inverse kinematic problem is considered using the neuro-fuzzy system NeFIK. Section 5.4 presents the constrained minimum-time control problem and trajectory generation. In section 5.5, the neuro-fuzzy system NeFOTC is developed. Section 5.6 presents simulation results, and finally, section 5.7 concludes the paper.

5. 3. Neuro-Fuzzy Inverse Kinematics

Although this study can be applied to a general n-Degrees-Of-Freedom (DOF) redundant robot, in this paper, it is only applied to a 3-DOF serial planar redundant manipulator, as shown in Fig. 5. 2. The problem of finding the joint coordinates (q_1, q_2, q_3), from Cartesian coordinates (x, y) is known as the inverse kinematics problem. The initial and final Cartesian positions of the manipulator are specified in the task space of the robot.

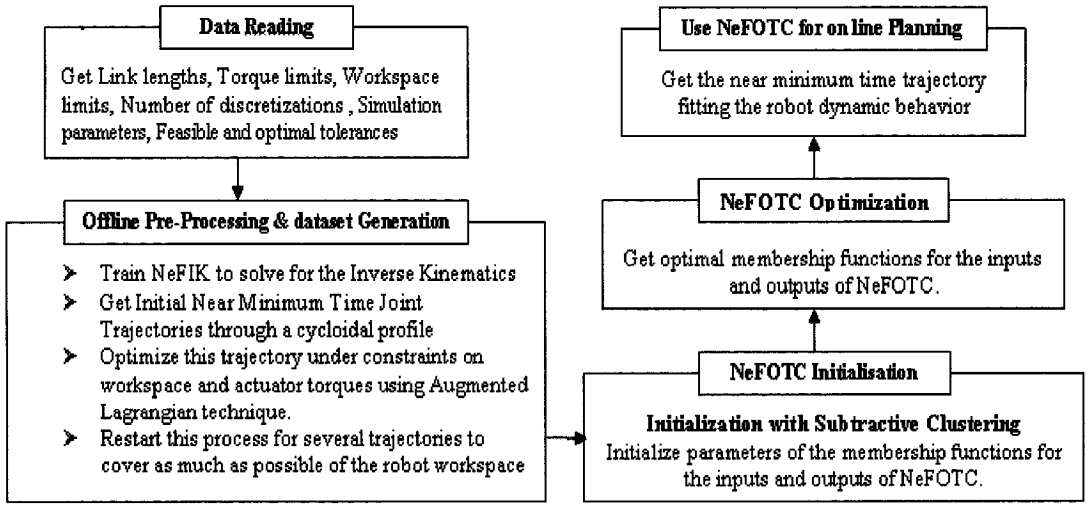


Fig. 5. 1. Overall architecture of the proposed optimal-time trajectory planning system

The detailed kinematic model is provided in Appendix B. The kinematic redundancy is solved using a neuro-fuzzy system called NeFIK. The training dataset is obtained using Forward Kinematics (FK) model of the manipulator given by (Fig. 5.3):

$$x = L_1\cos(q_1) + L_2\cos(q_1 + q_2) + L_3\cos(q_1 + q_2 + q_3)$$

$$y = L_1\sin(q_1) + L_2\sin(q_1 + q_2) + L_3\sin(q_1 + q_2 + q_3), \quad (1)$$

where L_i, is the length of link i of the robot. A set of joint angles (q_1, q_2, q_3) is used to construct the true (x, y), and thus to get an error on which to apply the back-propagation algorithm. NeFIK is a multi-layer feedforward adaptive neuro-fuzzy network. The first layer is a two input layer, characterizing the crisp Cartesian position values. The last layer is a three output-layer characterizing the corresponding

crisp joint angles. NeFIK involves three hidden layers. The second one is the fuzzification layer, which transfers the crisp inputs to linguistic variables, through sigmoidal transfer functions. The third layer performs fuzzy rules, which applies the product t-norm to produce the firing strengths of each rule. It is followed by a normalization layer. The training rule option is the Levenberg-Marquard version of the gradient back-propagation algorithm. This choice allows speeding up the learning process substantially with less iterations as compared to standard back-propagation (e.g. gradient descent). This algorithm for multi-layer networks can be thought of as a generalization of Least Mean Square (LMS) algorithm; since both use a mean square error performance index.

5. 4. Off-line minimum-time trajectory generation

In this section, the minimum-time trajectories that will serve for building the data-driven neuro-fuzzy system are generated, by implementing an AL technique, according to the architecture shown in Fig. 5. 4.

5. 4. 1. Cycloidal Initialization

To fasten convergence of the optimization procedure, an initial solution is defined through a minimum- time trajectory parameterization. This solution consists of a kinematic feasible solution satisfying boundary constraints in joint angles, velocities, accelerations and jerks. It is obtained through a cycloidal profile (see

Appendix A). This choice is justified as it allows a near-minimum time smooth continuous trajectory as compared to a trapezoidal profile.

5. 4. 2. Robot Dynamics and Constraint Models

A. Dynamic Model

The robot dynamics is developed using a Lagrangian formalism, which includes actuators and friction models [25]. This model allows closed-form expression of joint rates and accelerations characterizing the motion resulting from joint torques. It can be expressed as:

$$\mathbf{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) + \mathbf{F}_{fric}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{2}$$

where $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$ is the joint torques vector produced by the joint actuators, $\boldsymbol{q} = [q_1, q_2, q_3]^T$, $\dot{\boldsymbol{q}} = [\dot{q}_1, \dot{q}_2, \dot{q}_3]^T$, $\ddot{\boldsymbol{q}} = [\ddot{q}_1, \ddot{q}_2, \ddot{q}_3]^T$ are vectors describing joint positions, rates and accelerations. $\mathbf{D}(\boldsymbol{q})$ is the 3×3 inertia matrix of the manipulator, $\mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the 3×1 Coriolis and centrifugal wrench, $\mathbf{G}(\boldsymbol{q})$ is the 3×1 gravitational wrench, and $\mathbf{F}_{fric}(\dot{\boldsymbol{q}})$ is the 3×1 viscous and Coulomb (or dry) friction force.

The linear combination of the two friction models constitutes the joint friction force $\mathbf{F}_{fricl}(\dot{\boldsymbol{q}})$, expressed as:

$$\mathbf{F}_{fricl} = \mathbf{F}_V \dot{\boldsymbol{q}} + \mathbf{F}_C \operatorname{sgn}(\dot{\boldsymbol{q}}) \tag{3}$$

$\mathbf{F}_V$ and $\mathbf{F}_C$ are, respectively, the viscous and Coulomb friction coefficients. The derivation of the robot kinematic and dynamic models is presented in Appendix B.

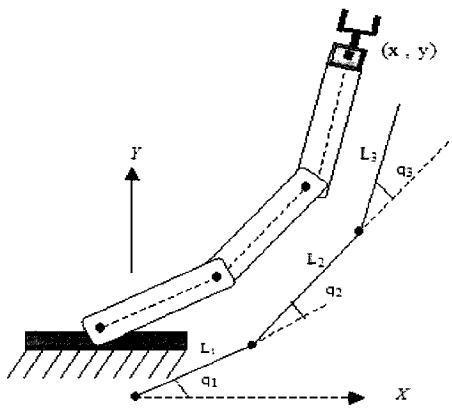


Fig. 5. 2. Geometry of a 3-dof planar manipulator of joint angles q_j, $j = 1,2,3$

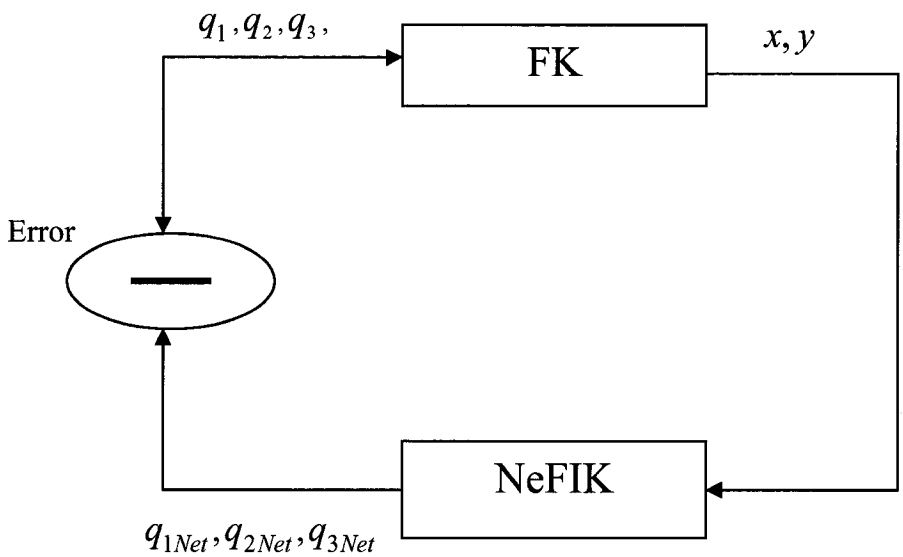


Fig. 5. 3. The use of the forward kinematic (FK) to the learning of the Neuro-Fuzzy Inverse Kinematic (NeFIK) system

In a discrete form, the inverse dynamic model (2) is approximated as [4, 26-29]:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{I}_{3\times3} & h_k \boldsymbol{I}_{3\times3} \\ \boldsymbol{O}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \boldsymbol{x}_k - \mathbf{D}^{-1}(\boldsymbol{x}_{1k}) \begin{bmatrix} \frac{h_k^2}{2} \boldsymbol{I}_{3\times3} \\ h_k \boldsymbol{I}_{3\times3} \end{bmatrix} \left[\mathbf{V}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \mathbf{G}(\boldsymbol{x}_{1k}) + \mathbf{F}_{fric}(\boldsymbol{x}_{2k}) \right]$$

$$+ \mathbf{D}^{-1}(\boldsymbol{x}_{1k}) \begin{bmatrix} \frac{h_k^2}{2} \boldsymbol{I}_{3\times3} \\ h_k \boldsymbol{I}_{3\times3} \end{bmatrix} \tag{4}$$

where $\boldsymbol{x}_k = (\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)^T = (\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})^T$ is a 6-dimensional directional column representing the robot state, composed of joint angles and angular velocity. Equation (4) is re-written for simplicity as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k \ (\boldsymbol{x}_{k,} \boldsymbol{\tau}_k, h_k) \tag{5}$$

B. Associated Constraints

In addition to eq. (5), the following constraints are considered:

- Boundary conditions: These concern the starting and final states $\boldsymbol{x}_s$ and $\boldsymbol{x}_T$

$$\boldsymbol{x}_0 = \boldsymbol{x}_S, \boldsymbol{x}_N = \boldsymbol{x}_T \tag{6}$$

- Admissible domain of the sampling periods

$$\mathrm{H} = \left\{ h_k \in \Re^+, \text{ such that } h_{\mathrm{Min}} \le h_k \le h_{\mathrm{Max}}, \ k = 0,...,N-1 \right\} \tag{7}$$

- Admissible domain of the actuator torques

$$\mathbf{C} = \left\{ \boldsymbol{\tau}_k \in \Re^3, \text{ such that } \boldsymbol{\tau}_{\mathrm{Min}} \le \boldsymbol{\tau}_k \le \boldsymbol{\tau}_{\mathrm{Max}}, \ k = 0,...,N-1 \right\} \tag{8}$$

5. 5. Non-Linear Programming Formulation

5. 5. 1. Performance Index

Although it has been widely used by several authors [1-4, 26-29], time optimal control of manipulator robots still has a major drawback, namely, its bang-bang character, producing non-smooth trajectories, which in turn, fastens the mechanical fatigue of the machine. Moreover, it had been shown that tracking a planned minimum-time trajectory could lead to large joint vibrations, and overshoot of the nominal torque limits [5, 6]. Conversely, one of the ultimate objectives of offline planning is to build dataset trajectories, to be later used for online planning and control. Therefore, these trajectories must be as smooth as possible. On the other hand, the sampling periods might be defined such that the overall traveling time is $T = \sum_{k=0}^{N-1} h_k$, where h_k is the EE traveling time between two successive discrete poses k and $k+1$, $k = 0,...,N-1$. The variables h_k are first guessed according to the minimum time cycloidal profile. Then they are taken as control variables, along with the torque inputs in the optimal control procedure.

5. 5. 2. Augmented Lagrangian (AL)

In discrete-time, the optimal control problem consists of finding the optimal sequences $(\boldsymbol{\tau}_0, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_{N-1})$ and $(h_0, h_2, ..., h_{N-1})$, allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a target state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost function E_d:

$$\underset{\substack{\tau \in C \\ h \in H}}{\text{Min}} \left\{ E_d = \left[\sum_{k=0}^{N-1} h_k \right] \right\} \tag{9}$$

Subject to constraints (5)-(8).

An AL technique is used to solve the resulting constrained non-linear optimal control problem [26-29, 30, 31]. This technique basically relies on quadratic penalty methods, but reduces the possibility of ill conditioning of the sub-problems that are generated with penalization by introducing explicit Lagrange multiplier estimates at each step into the function to be minimized. The AL function is written as:

$$L_{\mu}(\boldsymbol{x}, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = \sum_{k=1}^{N} h_k + \sum_{k=0}^{N-1} \boldsymbol{\lambda}_{k+1}^{T} \left[\boldsymbol{x}_{k+1} - \boldsymbol{f}_k (\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \right] +$$

$$\sum_{k=0}^{N} h_k \left\{ \sum_{j=1}^{2} \left[\Phi_{\mu_g} (\boldsymbol{\rho}_k^j, \boldsymbol{g}_j (\boldsymbol{x}_k, \boldsymbol{\tau}_k)) + \Psi_{\mu_s} (\boldsymbol{\sigma}_k^j, \boldsymbol{s}_j (\boldsymbol{x}_k)) \right] \right\} \tag{10}$$

In eq. (10), the function $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$ is defined by the discrete state eq. (5) at the sampling point k, N is the total sampling number. $\boldsymbol{\lambda} \in R^{6N}$ designates the co-states obtained from the adjunct equations. $\boldsymbol{\sigma}$ and $\boldsymbol{\rho}$ are Lagrange multipliers with appropriate dimensions, related to equality and inequality constraints, noted respectively as $\boldsymbol{s}$, and $\boldsymbol{g}$, and $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_g$ are the associated penalty coefficients. The adopted penalty functions combine penalty and dual methods allowing relaxation of inequality constraints as soon as they are satisfied. These are given by:

$$\Psi_{\mu_s}(\boldsymbol{a},\boldsymbol{b})=(\boldsymbol{a}+\frac{\boldsymbol{\mu}_s}{2}\boldsymbol{b})^T\boldsymbol{b},\ \Phi_{\mu_g}(\boldsymbol{a},\boldsymbol{b})=\frac{1}{2\boldsymbol{\mu}_g}\left\{\left\|\text{Max}(0,\boldsymbol{a}+\boldsymbol{\mu}_g\boldsymbol{b})\right\|^2-\left\|\boldsymbol{a}\right\|^2\right\} \tag{11}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ refer to Lagrange multipliers and left hand side of equality and inequality constraints. The Karush-Kuhn-Tucker first order optimality conditions state that for a trajectory $(\boldsymbol{x}_0,\boldsymbol{\tau}_0,h_0,...,\boldsymbol{x}_N,\boldsymbol{\tau}_{N-1},h_{N-1})$, to be optimal solution to the problem, there must exists some positive Lagrange multipliers $(\boldsymbol{\lambda}_k,\boldsymbol{\rho}_k)$, unrestricted sign multipliers $\boldsymbol{\sigma}_k$, and finite positive penalty coefficients $\boldsymbol{\mu}_S$ and $\boldsymbol{\mu}_g$ such that:

$$\frac{\partial L_\mu}{\partial \boldsymbol{x}_k}=0,\frac{\partial L_\mu}{\partial \boldsymbol{\tau}_k}=0,\ \frac{\partial L_\mu}{\partial h_k}=0,\ \frac{\partial L_\mu}{\partial \boldsymbol{\lambda}_k}=0,\frac{\partial L_\mu}{\partial \boldsymbol{\rho}_k}=0,\ \frac{\partial L_\mu}{\partial \boldsymbol{\sigma}_k}=0,$$

$$\boldsymbol{\sigma}_k^T\boldsymbol{s}(\boldsymbol{x})=0,\boldsymbol{\rho}_k^T\boldsymbol{g}(\boldsymbol{x},\boldsymbol{\tau},h)=0 \text{ and } \boldsymbol{g}(\boldsymbol{x},\boldsymbol{\tau},h)\le 0 \tag{12}$$

The development of conditions (12) enables one to derive the iterative formulas to solve the optimal control problem by updating control variables, Lagrange multipliers and penalty coefficients. In evaluating the co-states, the function $\text{arctg}(c_i\,\boldsymbol{x}_{2k}^i/(\pi/2))$, i=1, 2, 3, is used to approximate the sign function present in the friction term of the dynamic model, where c_i is a suitable constant. In AL procedure (Fig. 5. 4), one first has to select a set of data related to robot, task definition and workspace limitations, as well as AL parameters. Then a kinematic unit defines a feasible initial solution, through a cycloidal profile, as described in sub-section 6.3.1. In an inner optimization loop, one first computes the co-states

backwardly, then solves for AL minimization with respect to sampling periods and actuator torque control variables. This allows computing the robot state, which is tested against feasibility tolerances. The test is done on the norms of all equality and inequality constraints. If the feasibility test fails, restart the inner optimization unit. Otherwise, if the feasibility test succeeds, i.e., the current values of penalty are good in maintaining near-feasibility of iterates, and a convergence test is made against optimal tolerances. If convergence holds, display optimal results and end the program. If non convergence, go further to the dual part of AL to test for constraints satisfaction and update multipliers, penalty and tolerance parameters to force the subsequent primal iterates to be increasingly accurate solutions of the primal problem.

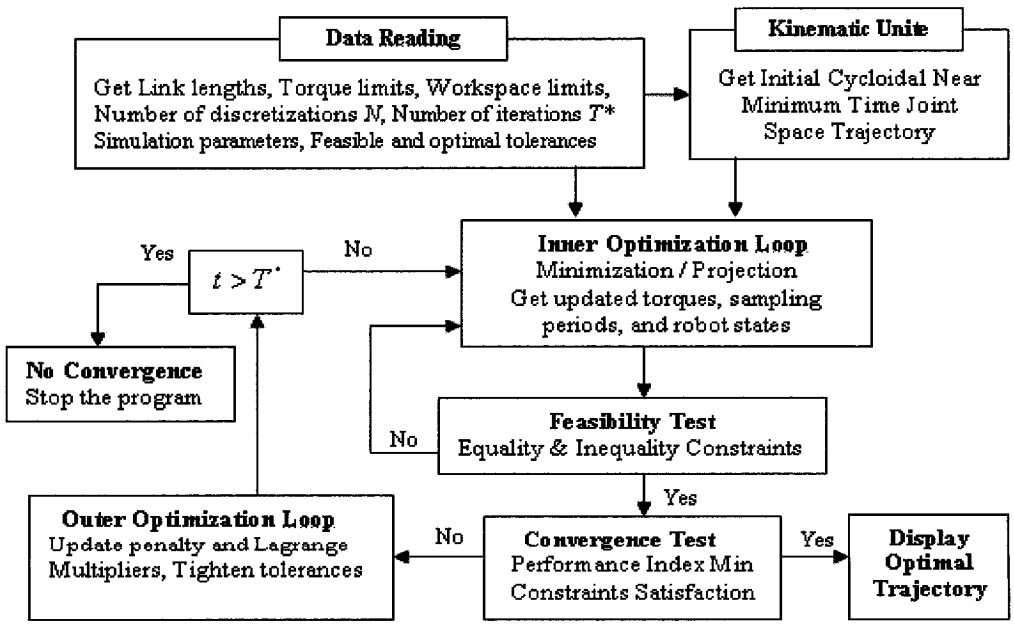


Fig. 5. 4 Architecture of the AL system for the minimum-time trajectory planning problem.

5. 6. Neuro-Fuzzy Optimal-Time Controller (NeFOTC)

In this section, the neuro-fuzzy system NeFOTC (for Neuro-Fuzzy Optimal-Time Controller) is developed. This system is an objective adaptive neuro-fuzzy network, based on Tsukamoto fuzzy inference. It starts with a subtractive clustering of input-output data. Then the inference parameters are learnt and optimized. Figure 5. 5 shows a schematic representation of NeFOTC learning procedure.

5. 6. 1 Rule Base and Membership Functions

Let's define the network inputs by $\boldsymbol{x} = [q_1, q_2, q_3, \dot{q}_1, \dot{q}_2, \dot{q}_3]^T = [x_1, x_2, x_3, x_4, x_5, x_6]^T$, and the outputs by the torques vector $\boldsymbol{\tau} = [\tau_1, \tau_2, \tau_3]^T$. Consider the following dataset $[(\boldsymbol{x}_1, \boldsymbol{\tau}_1),..., (\boldsymbol{x}_K, \boldsymbol{\tau}_K)]$. In NeFOTC, fuzzy rules are defined according to the following format:

R^1 : If (x_1 is A_1^1) and (x_2 is A_2^1) and...and(x_6 is A_6^1)Then ($o_1 = c_1^1$)and ($o_2 = c_2^1$)and ($o_3 = c_3^1$)

also

...

R^j :If (x_1 is A_1^j)and (x_2 is A_2^j) and...and(x_6 is A_6^j)Then($o_1 = c_1^j$)and($o_2 = c_2^j$)and ($o_3 = c_3^j$)

...

also

R^N : If(x_1 is A_1^N)and(x_2 is A_2^N) and...and(x_6 is A_6^N)Then ($o_1 = c_1^N$)and($o_2 = c_2^N$)and($o_3 = c_3^N$)

where x_i is the input to node *i*. A_i^j is the associated linguistic term. The membership functions $\mu_{A_i^j}$ defining the antecedent parts A_i^j are Gaussian expressed as:

$$\mu_{A_i^j}(x_i) = \exp\{-\frac{(x_i - a_i^j)^2}{2(b_i^j)^2}\} \tag{13}$$

where a_i^j and b_i^j are the mean and standard deviation of the jth membership function associated with the input variable x_i and o_k are fuzzy sets representing the consequent part o_k^j of the jth rule. These are obtained as:

$$c_k^j = f^{-1}(\mu_{A_i^j}(x_i)),\ f:\Re \to [0,1] \tag{14}$$

f being a continuous and strong monotone function. For such a function, there always exists an inverse function f^{-1} . A typical choice of such a function would be a logistic function defined as:

$$f(z) = \frac{1}{1+e^{-(\rho z-\sigma)}} \tag{15}$$

where ρ and σ are real numbers affecting the position of the inflection point of *f*, and its slope. This system might be thought of as a generalization of 0-order Takagi Sugeno inference, and is known as Tsukamoto inference [21].

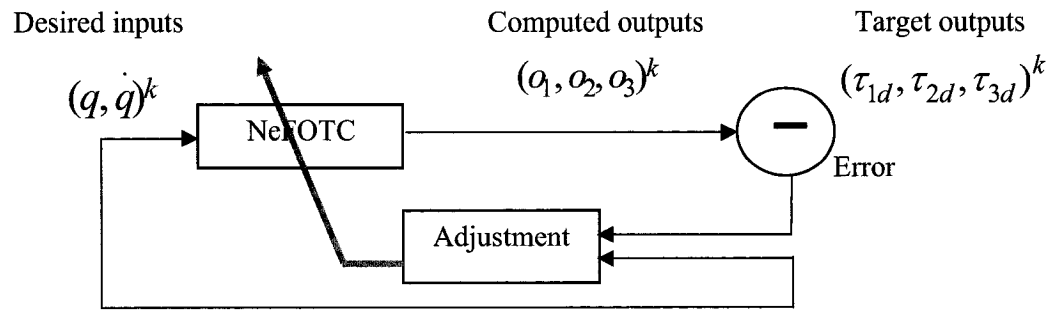


Fig. 5. 5. Learning procedure of NeFOTC

5. 6. 2. NeFOTC Learning Procedure

The learning of NeFOTC is performed in two phases: The first phase is data pre-processing for structure identification; the second phase is network training and optimization.

A. Data partitioning and structure identification

The structure identification phase is to find proper input space fuzzy partitions and rules with two objectives: 1) minimize the number of rules generated, and 2) minimize the number of fuzzy sets on the universe of discourse for each input variable. The definition of membership functions used for fuzzification and defuzzification of input and output variables plays an important role in the ability of a neuro-fuzzy controller to learn and generalize [21]. To initialize the premise membership function parameters a_i^j and b_i^j in the antecedent part and the conditional

parameters ρ_k^j, σ_k^j in the consequent part of a fuzzy rule R^j, a clustering method is applied. With clustered training points, the network will be very accurate while dealing with points near the cluster and should breed to the robot familiarity with its workspace. Another advantage is that it reduces the problem of combinatorial explosion of rules when the input data has a high dimension (as for the case at hand).

For instance, in our case of 6-input network, if each input variable is represented by three fuzzy sets, one gets 729 rules. Worse, if a more representative model with five fuzzy sets for each input variable is taken; one ends up to 15625 rules. It is clearly impractical to implement a fuzzy system with thousands of rules. This phenomenon has been identified in the literature as the *curse of dimensionality problem* [20]. Assume that we don't have a clear idea on how many clusters there should be for our input-output dataset so generated from the pre-processing steps. To find the number of clusters and the cluster centers in the dataset, the subtractive clustering method introduced in [22], and extended in [32] is implemented. This approach provides a fast; one-pass algorithm to take input-output training data to generate the objective fuzzy inference model of the robot dynamic behaviour. The Matlab function [33] *subclust* is used to implement this method.

B. NeFOTC Training and Optimization

The optimization phase is a supervised iterative algorithm. At this level, membership function parameters of the antecedent and consequence parts of fuzzy

rules are adjusted to minimize an error function aiming at capturing the robot dynamic behaviour, while achieving a minimum-time trajectory, and satisfying constraints. This error function is:

$$E = \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{o}^k - \boldsymbol{\tau}^k\right\|^2 = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{N}\sum_{l=1}^{3}(\mathbf{o}_{jl}^k(a_i, b_i, \rho_i, \sigma_i) - \boldsymbol{\tau}_{jl}^k)^2 \qquad (16)$$

while $\mathbf{o}^k$ is the computed output associated to the kth training dataset entry, $\boldsymbol{\tau}_l^k$ is the kth real (or desired) output corresponding to the lth joint torque, and N is the number of rules.

5. 6. 3. NeFOTC Architecture

Likewise to NeFIK, NeFOTC is made up of six layers as shown in Fig. 5. 6. The function of each layer is discussed below. The first layer is a six input layer, characterizing the crisp joint angles and rates. The second layer performs the fuzzification of the crisp inputs into linguistic variables, through Gaussian transfer functions. The third one is the rule layer, which applies the product t-norm to produce the firing strengths of each rule. This is followed by a normalization layer, at which each node calculates the ratio of a rule's firing strength to the sum of all rules firing strengths. The fifth layer performs the inversion. The last layer does the aggregation and deffuzification, where the output is obtained as the summation of all incoming signals. The used training rule option is again the Levenberg-Marquard version of the gradient back-propagation algorithm.

5. 7. Simulation results

5. 7. 1. Training Protocol

The robot Kinematic and dynamic parameters are given in Tables 6. 1 and 6. 2. The forward kinematics and inverse dynamics are used to generate offline optimal trajectories. As the robot interpolates between sampling points to build the entire path from a starting to an ending points, it might pass through or near singular poses. These poses are also referred to as boundary conditions. For instance, the development of the robot full Jacobian – including the end-effector orientation – yields to singular poses described by $q_2 = 0$, or $q_2 = 180$ degrees. Physically, this corresponds to the elbow being either completely extended or flexed. Such discontinuities and singular poses might be avoided by incorporating them as constraints, including robot dynamics within the offline minimum time trajectory generation. In our preliminary simulations, this indeterminacy is overcome by considering the following assumptions: First, the training data are reduced using either elbow up or elbow down poses. Elbow up is used to generate our training dataset. A similar dataset could be built for elbow-down pose. Note that the elbow up pose used for the learning is determined among infinitely many solutions that exist for each input. Second, because the order in which data points are presented to the network affects the speed and quality of the learning convergence, the training points are equally distributed throughout the workspace and sequentially introduced

5. 7. 2. Simulation Results

One hundred trajectories had been achieved from different initial to final 2-dimentional positions for elbow-up EE poses. Each trajectory is sampled through 20 discretization points. One turns up with 2000 data points to perform the training dataset. For each trajectory, first NeFIK is applied. Second, the initialization unite defines the near-minimum time solution in the joint space using the cycloidal algorithm. Third, the offline optimization procedure gives the optimal time trajectory associated to the given starting and ending points.

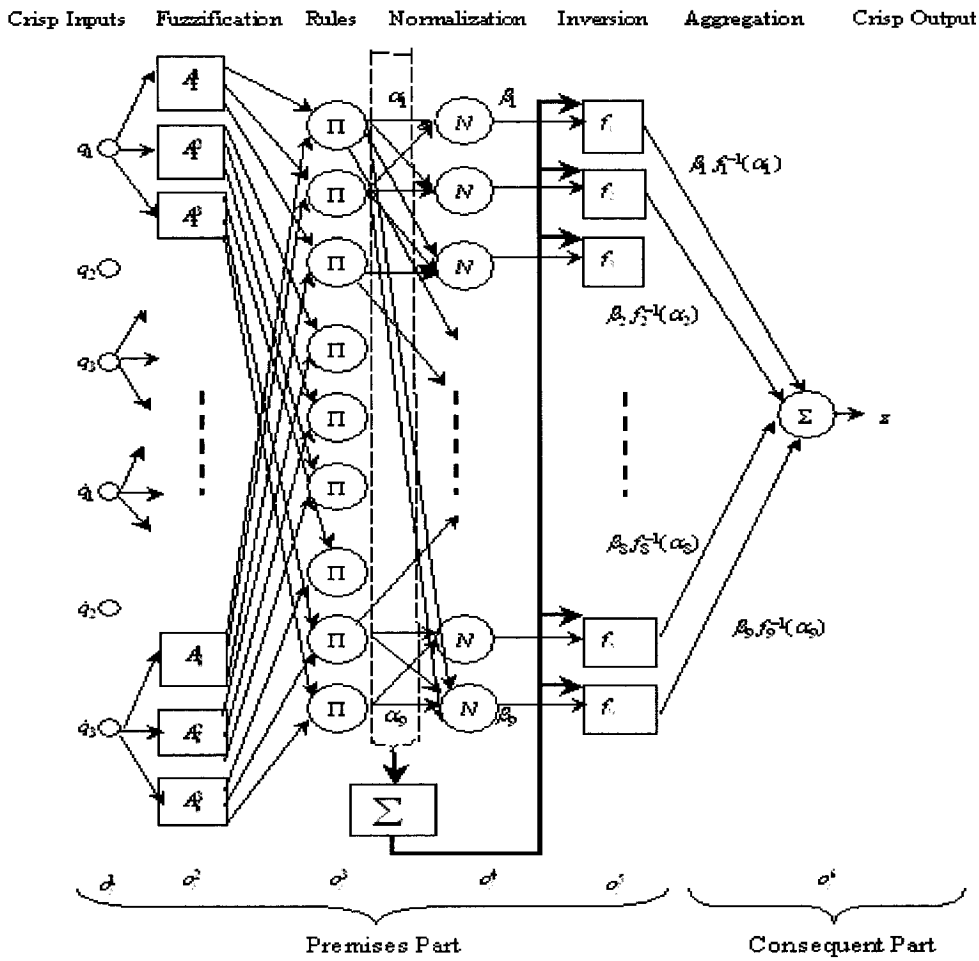


Fig. 5. 6. Architecture of the Tsukamoto neuro-fuzzy inference system in a simplified form with one output noted *z*.

Table 5. 1. Kinematic and dynamic parameters of the manipulator

Link I	Mass (kg)	Inertia I_i(kg m^2)	Length L_i(m)	Fv (Nm Sec/rad)	Fc (Nm)
1	10	9.23	1.0	0.7	0.3
2	8	7.5	0.8	0.6	0.5
3	4	5.21	0.45	0.5	0.6

Table 5. 2. Limits of workspace, actuator torques, joint rates and accelerations and sampling periods

Parameter	x (m)	y (m)	τ_1 (Nm)	τ_2 (Nm)	τ_3 (Nm)	$\dot{q}_{i\text{Motor}}$ (rad/s)	$\ddot{q}_{i\text{Motor}}$ (rad/s^2)	h (sec)
Max	3.0	3.0	40	25	20	2	3	0.1
Min	-1.0	0.0	-40	-25	-20	-2	-3	0.001

On all obtained trajectories a subtractive clustering is performed with the following parameters: The upper acceptance threshold for a data point to be a cluster center is $\bar{\varepsilon} = 1.0$. The lower rejection ratio is $\underline{\varepsilon} = 0.8$. The cluster radius defining the neighbourhood ranges (or zone of association of a cluster centre in the data space) is $r_a = 0.9$, and the squash parameter is $\eta = 0.5$. A six-rule Tsukamoto model is identified for NeFIK, and a 9-rules network characterizes NeFOTC. One notices the rules number is greater for NeFOTC as compared to NeFIK. This might stem from the fact that NeFOTC is larger, involving more variables and parameters. Figure 6.7 shows the training performance for NeFIK, which is very good as it reaches a root mean square error (RMSE) of 10^{-4} units in less than 10 epochs. Table 6.3 and Fig. 5.8 show the performance of NeFOTC. One notices the interesting performance of NeFOTC, getting an error of order of 10^{-3} in less than 10 training epochs.

A scenario consists of moving the robot EE from the starting position (1.95, 0.82) (m), corresponding to the joint position (0°, 30°, 30°) in a straight line to the final Cartesian position (1.35, 1.3) (m), without considering the orientation. Hence, this task is performed with a serial 3-DOF planar manipulator that is redundant with

respect to the given task. Figure 5. 9 shows the position variations for the minimum-time trajectory generated based on cycloidal parameterization and that generated including robot dynamics and constraints. Although the Cartesian minimum-time trajectory is a straight line, it is no longer the case for its corresponding dynamic-based trajectory. This bias is due to the non-linearity of the dynamic model, considering gravity forces for example. Figure 5. 10 displays the corresponding joint angle variations. The joint velocities are zero at initial and final positions and the end-effector travels through a distance of 0.75 m in 1.4sec. Figure 5. 11 shows joint rates for this trajectory. Figure 5. 12 shows the torques and time variations with respect to sampling points. One notices that although kinematically feasible, the cycloidal parameterized trajectory associated torques – as computed from the robot inverse dynamics – are fairly high and exceed nominal values.

In order to assess the generalization capabilities of the proposed approach, two tests had been performed. The first one is made on a trajectory that had been learnt by the offline AL technique, which consists to move the robot tip along a straight line, starting from the Cartesian coordinates (0.2, 0.9) (m) corresponding to the joint position (0^0, 15^0, 20^0) to the ending position (0.6, 0.3) (m). Both NeFIK and NeFOTC were used to achieve the inverse kinematics for the former and get the near minimum time trajectory for the latter. The actuator torques and sampling period variations can be obtained from the outcomes $\mathbf{o}^k$ of NeFOTC. The corresponding trajectory and consumed energy are shown in Figs. 6.13 and 6. 14. These figures show the very

good generalization capabilities of NeFOTC. A second test is with respect to changes of some values of entry data. The testing data from Table 6. 3 were disturbed by adding an amount of 5% for each data point. Table 6. 4 shows the high robustness capability of NeFOTC.

As for time consumption, NeFIK required 84 seconds to learn the inverse kinematic solution, whereas NeFOTC required 117 sec to capture the robot dynamic behaviour associated to the minimum-time trajectory. The CPU time might be greater if the overall workspace has been considered for both NeFIK, and NeFOTC. Nonetheless, the reduced workspace assumption to elbow up or down is quite restrictive. This assumption is relaxed in an ongoing work, by including a resolution redundancy function in the offline planning procedure.

Table 5. 3. Performance of NeFOTC

RMS Error for 2000 trajectory entries after 10 epochs training

2000 data entries	RMS Error
Training data (90%)	0.003771
Testing data (10%)	0.0012857

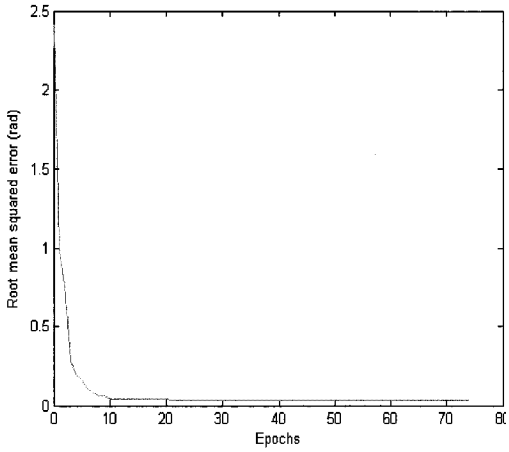


Fig. 5.7. NeFIK performance: RMS Error with respect to learning epochs

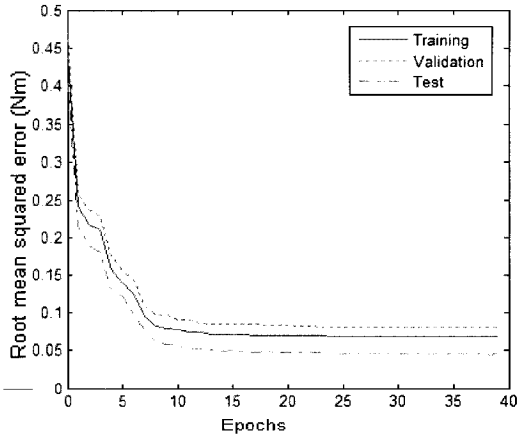


Fig. 5.8. NeFOTC performance learning error (RMS error vs. Epochs)

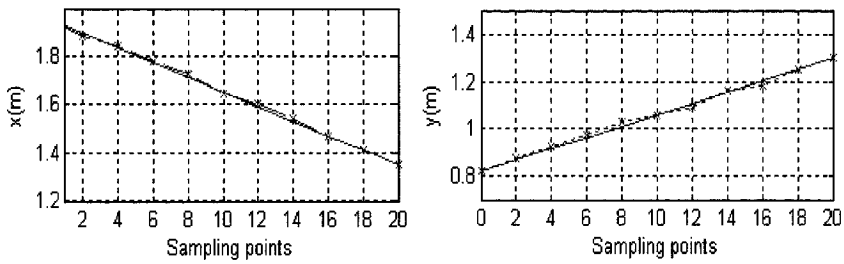


Fig. 5. 9. end-effector trajectory along axis x and y
___Cycloidal profile, *-*-*-Minimum-time AL trajectory

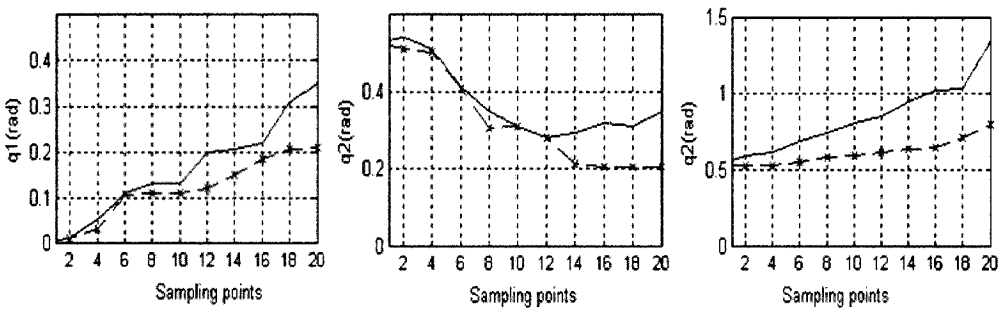


Fig. 5. 10. Joint variations
(___Cycloidal profile, *-*-*- ALD trajectory)

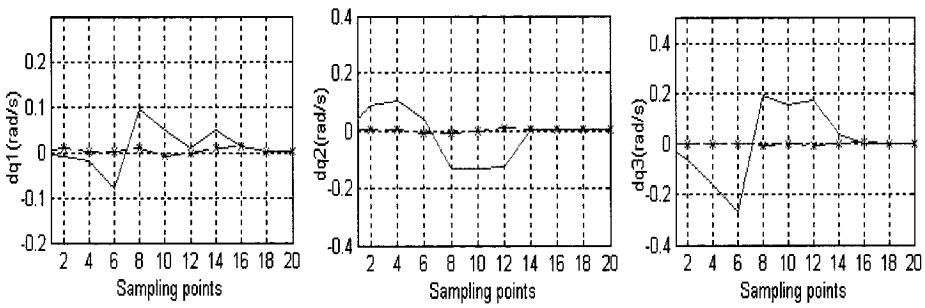


Fig. 5.11. Joint rates $\dot{q}_i$ variations
(___Cycloidal profile, *-*-*- ALD trajectory)

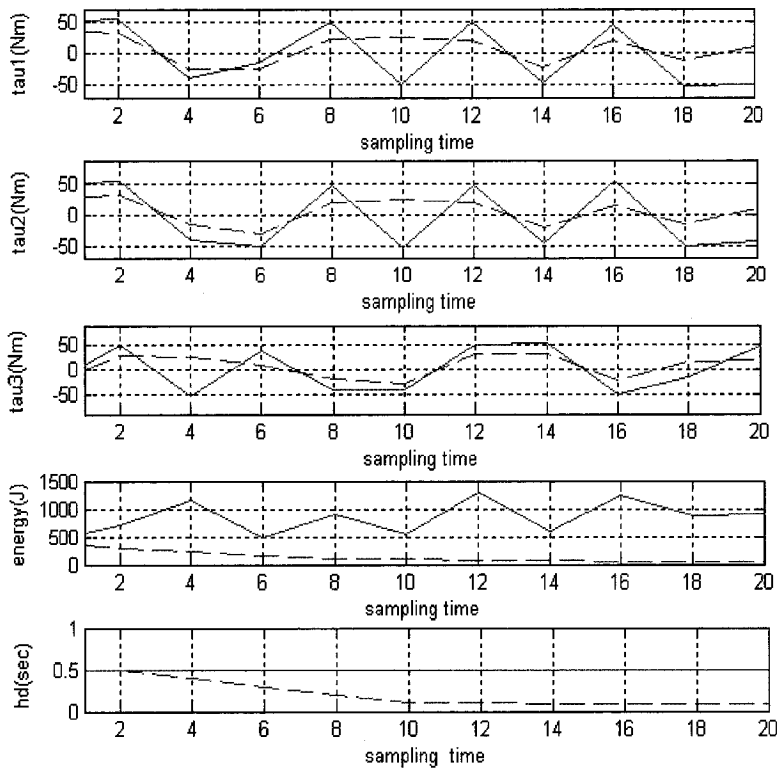


Fig.5.12. Corresponding torques τ_i electric energy and sampling period h instantaneous variations
(___Cycloidal profile, *-*-*- ALD trajectory)

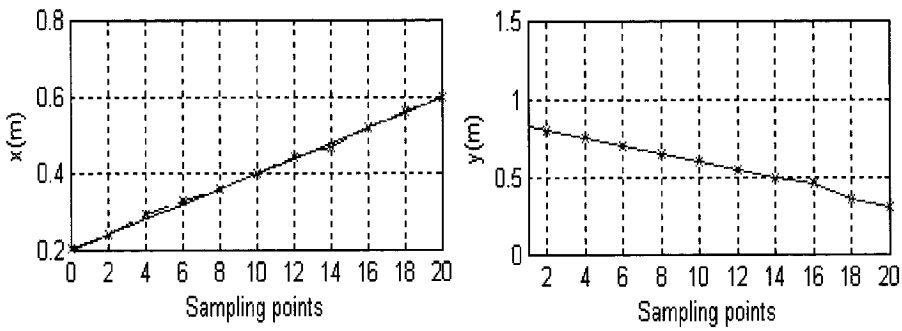


Fig. 5. 13. End-effector trajectory along axis x and y
(___Cycloidal profile, *-*-*- ALD trajectory)

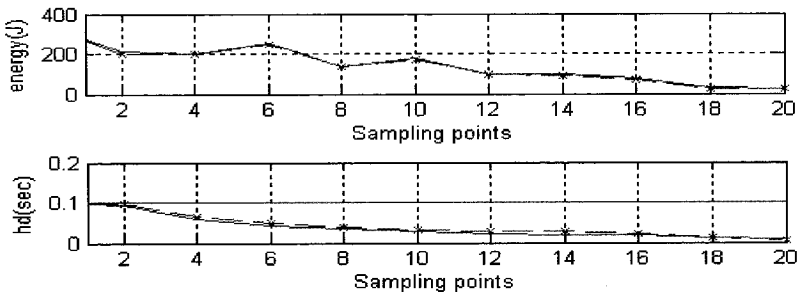


Fig. 5.14. Energy and sampling period h instantaneous variations for the test trajectory
(___Cycloidal profile, *-*-*- ALD trajectory)

Table 5. 4. Performance of NeFOTC, with disturbed testing data, by an amount of 5% RMS Error after 20 epochs training

2000 data entries	RMS Error (after 20 epochs)
Training data (90%)	0.003771
Testing data (10%)	0.0012857
Validation (noised data) (10%)	0.0024150

5. 8. Conclusion and Discussion

In fuzzy systems literature, there are several studies considering online robot motion control and planning problems. Some of these studies had been reported and discussed. The main contribution of this paper is the design of a new system to online optimal time motion planning. This system involves joint angles, joint rates, accelerations, jerks, as well as actuator torques and workspace limitations. The training dataset is performed offline. A first neuro-fuzzy system is trained to solve the inverse kinematics problem. Then, four pre-processing steps allow setting up the minimum time trajectory planning, based on a constrained dynamic model. This is followed by a data-driven neuro-fuzzy system, which is trained to represent and capture the robot constrained dynamic behavior associated to the minimum time trajectory problem.

Once the neuro-fuzzy model (rules number and premise and consequence membership function parameters) is learnt and optimized, a minimum time trajectory capturing the robot dynamics is obtained for a given started and target points within elbow-up EE pose. This is provided without the need to re-do all the calculations that would be necessary with a conventional optimal time control. Simulation results show that the proposed system has very good learning and generalization capabilities of the difficult non linear dynamics. This makes it possible for the development of an online optimal time motion planning system for manipulator robots. As for short trends, more simulations and experimental tests on a physical robot are being undertaken.

Another perspective is to develop further the proposed approach to involve multi-objective online motion planning.

Acknowledgment

The authors gratefully thank the anonymous reviewers for the valuable comments and the Natural Science and Engineering Research Council of Canada (NSERC) for supporting this work under grants ES D3-317622, RGPIN-203618, RGPIN-105518, RGPIN-18898 and STPGP-269579.

5. 9. Appendix A: Minimum-Time Joint Space Trajectory Parameterization

Let us consider a 3-DOF planar serial manipulator operating in a two dimensional positioning space as shown in Fig. 5. 2. The robot task is to position its EE without considering its orientation within a limited workspace and time interval. Moreover, each robot joint is to produce zero rates and accelerations at the ends of the task. For that purpose, a cycloidal function is chosen to model the trajectory time t from 0 to an unknown final travel time T with the normalized time scale s as:

$$s = \frac{t}{T}, \quad 0 \le t \le T, \quad 0 \le s \le 1, \tag{A1}$$

where the cycloidal function is given by:

$$q(s) = s - \frac{1}{2\pi}\sin(2\pi s) \tag{A2}$$

with its first and second derivatives being given as:

$$\dot{q}(s) = 1 - \cos(2\pi s) \tag{A3}$$

$$\ddot{q}(s) = 2\pi \sin(2\pi s) \tag{A4}$$

It is noteworthy that the cycloidal motion and its derivatives are defined within the range (-1, 1). It has zero velocity and acceleration at the ends of the interval. The initial and final joint angles are given as $\boldsymbol{q}^I$ and $\boldsymbol{q}^F$. The motion of joint j attains its maximum velocity at the center of the interval, i.e., $s = 0.5$, the maximum being $\dot{q}_{\max} = \dot{q}(0.5) = 2$, and hence

$$\dot{q}_{j\,\mathrm{Max}} = \frac{2}{T}\left| q_j^F - q_j^I \right| \tag{A5}$$

Similarly, the acceleration of joint j attains its maximum and minimum values at $s = 0.25$, and $s = 0.75$, the

maximum being $\ddot{q}_{\mathrm{Max}} = \ddot{q}(0.25) = \ddot{q}(0.75) = 2\pi$

Hence

$$(\ddot{q}_j)_{\mathrm{Max}} = \frac{2\pi}{T^2}\left| q_j^F - q_j^I \right| \tag{A6}$$

Finally, the jerk of joint j attains its maximum at $s = 0.0$, and $s = 1.0$, the maximum being

$$\dddot{q}_{\mathrm{Max}} = \dddot{q}(0) = \dddot{q}(1) = 4\pi^2$$

Thus

$$(\ddot{q}_j)_{\text{Max}} = \frac{4\pi^2}{T^3}\left|q_j^F - q_j^I\right| \tag{A7}$$

This motion is constrained by the maximum joint velocities, accelerations and jerks delivered by the motors, i.e.

$$\dot{q}_j \le \dot{q}_{j\text{Motor}},\ \ddot{q}_j \le \ddot{q}_{j\text{Motor}}, \text{ and } \dddot{q}_j \le \dddot{q}_{j\text{Motor}} \tag{A8}$$

Hence, the minimum-time trajectory of joint j is limited by the strongest constraint among the $\dot{q}_{j\text{Motor}}$, $\ddot{q}_{j\text{Motor}}$ and $\dddot{q}_{j\text{Motor}}$. It is given by:

$$T_j = \text{Max}\left\{\frac{2\left|q_j^F - q_j^I\right|}{\dot{q}_{j\text{Motor}}},\ \sqrt{\frac{2\pi}{\ddot{q}_{j\text{Motor}}}\left|q_j^F - q_j^I\right|},\ \left(\frac{4\pi^2\left|q_j^F - q_j^I\right|}{\dddot{q}_{j\text{Motor}}}\right)^{1/3}\right\} \tag{A9}$$

The overall minimum-time trajectory (for all of the three joints together) is therefore obtained by:

$$T_{\text{Min}} = \text{Max}\left\{T_1, T_2, T_3\right\} \tag{A10}$$

The resulting minimum-time trajectory characterized by joints position, velocity and acceleration is given as:

$$\begin{aligned}
\boldsymbol{q}(t) &= \boldsymbol{q}^I + (\boldsymbol{q}^F - \boldsymbol{q}^I)\left(\frac{t}{T_{\text{Min}}} - \frac{1}{2\pi}\sin\left(2\pi\frac{t}{T_{\text{Min}}}\right)\right) \\
\dot{\boldsymbol{q}}(t) &= \frac{\boldsymbol{q}^F - \boldsymbol{q}^I}{T_{\text{Min}}}\left(1 - \cos\left(2\pi\frac{t}{T_{\text{Min}}}\right)\right) \\
\ddot{\boldsymbol{q}}(t) &= \frac{\boldsymbol{q}^F - \boldsymbol{q}^I}{T_{\text{Min}}^2}2\pi\sin\left(2\pi\frac{t}{T_{\text{Min}}}\right)
\end{aligned} \tag{A11}$$

5. 10. Appendix B: Robot kinematic and dynamic models including actuator and friction models

The robot forward kinematic model at the velocity level expresses the Cartesian motion rates as function of joint rates. Following equation (1), one has:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} \tag{B1}$$

with $\mathbf{J}$ represents the robot Jacobian given by:

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \end{bmatrix} \tag{B2}$$

where

$$J_{11} = -L_1 \sin(q_1) - L_2 \sin(q_1 + q_2) - L_3 \sin(q_1 + q_2 + q_3)$$

$$J_{21} = L_1 \cos(q_1) + L_2 \cos(q_1 + q_2) + L_3 \cos(q_1 + q_2 + q_3)$$

$$J_{12} = -L_2 \sin(q_1 + q_2) - L_3 \sin(q_1 + q_2 + q_3)$$

$$J_{22} = L_2 \cos(q_1 + q_2) + L_3 \cos(q_1 + q_2 + q_3)$$

$$J_{13} = -L_3 \sin(q_1 + q_2 + q_3)$$

$$J_{23} = L_3 \cos(q_1 + q_2 + q_3)$$

The robot inverse dynamic model is developed using Lagrange formalism as [2, 3]:

$$\mathbf{D}_R(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{V}_R(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}_R(\boldsymbol{q}) + \mathbf{F}_{fric}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{B3}$$

where the components of the symmetric inertia matrix $\mathbf{D}_R$ are respectively :

$$D_{11} = I_1 + I_2 + I_3 + \frac{1}{4}m_1 L_1^2 + m_2(L_1^2 + \frac{1}{4}L_2^2 + L_1 L_2 \cos(q_2))$$

$$+ m_3(L_1^2 + L_2^1 + \frac{1}{4}L_3^2 + 2L_1 L_2 \cos(q_2) + L_1 L_3 \cos(q_2 + q_3) + L_2 L_3 \cos(q_3)$$

$$D_{12} = I_1 + I_3 + \frac{1}{2}[m_2(\frac{1}{2}L_2^2 + L_1 L_2 \cos q_2) + L_1 L_2 \cos(q_2) + L_1 L_3 \cos(q_2 + q_3)$$

$$+ 2L_2 L_3 \cos(q_3))]$$

$$D_{13} = I_3 + \frac{1}{2}m_3(\frac{1}{2}L_3^2 + L_1 L_3 \cos(q_2 + q_3) + 2L_1 L_3 \cos(q_3)$$

$$D_{22} = I_2 + I_3 + \frac{1}{4}m_2 L_2^2 + m_3(L_2^2 + \frac{1}{4}L_3^2 + L_2 L_3 \cos(q_3))$$

$$D_{23} = I_3 + \frac{1}{2}m_3(\frac{1}{2}L_3^2 + L_2 L_3 \cos(q_3))$$

$$D_{33} = I_3 + \frac{1}{4}m_3 L_3^2$$

The components of Coriolis and centrifugal wrenches $\mathbf{V}_R$ are:

$$C_1 = -[m_2 L_1 L_2 \sin q_2 + m_3 L_1(2L_2 \sin q_2 + L_3 \sin(q_2 + q_3))]\dot{q}_1 \dot{q}_2$$

$$- m_3 L_3(L_1 \sin(q_2 + q_3) + L_2 \sin q_3)\dot{q}_1 \dot{q}_3$$

$$-\frac{1}{2}[m_2 L_1 L_2 \sin q_2 + m_3 L_1(2L_2 \sin q_2 + L_3 \sin(q_2 + q_3)]\dot{q}_1^2$$

$$- m_3 L_3(L_1 \sin(q_2 + q_3) + L_2 \sin q_3)\dot{q}_1 \dot{q}_3 \quad -\frac{1}{2}m_3 L_3(L_1 \sin(q_2 + q_3) + L_2 \sin q_3)\dot{q}_3^2$$

$$C_2 = -\frac{1}{2}[m_2 L_1 L_2 \sin q_2 + m_3 L_1(2L_2 \sin q_2 + L_3 \sin(q_2 + q_3))]\dot{q}_1 \dot{q}_2$$

$$-\frac{1}{2}m_3L_3(L_1\sin(q_2+q_3)+L_2\sin(q_3)\dot{q}_1\dot{q}_3-m_3L_2L_3\dot{q}_2\dot{q}_3-\frac{1}{2}m_3L_2L_3\sin(q_3)\dot{q}_3^2$$

$$C_3=-\frac{1}{2}m_3L_1L_3\sin(q_2+q_3)\dot{q}_1\dot{q}_3-\frac{1}{2}m_3L_3(L_1\sin(q_2+q_3)+L_2\sin(q_3)\dot{q}_2\dot{q}_3$$

$$-\frac{1}{2}m_3L_2L_3\sin(q_3)\dot{q}_2\dot{q}_3$$

$$\beta_1=0$$

$$\beta_2=-[m_2L_1L_2\sin q_2+m_3L_1(2L_1L_2\sin q_2+L_1L_3\sin(q_1+q_2))]\dot{q}_1^2-$$

$$\left[m_2L_1L_2\sin q_2+m_3(2L_1L_2\sin(q_2)+L_1L_3\sin(q_2+q_3))\right]\dot{q}_1\dot{q}_3-m_3L_1L_3\sin(q_2+q_3)\dot{q}_1\dot{q}_3$$

$$\beta_3=-m_3(L_1L_3\sin(q_2+q_3)+L_2L_3\sin(q_3)\dot{q}_1^2-m_3(L_1L_3\sin(q_2+q_3)+2L_2L_3\sin(q_3))\dot{q}_1\dot{q}_2$$

$$-m_3(L_1L_3\sin(q_2+q_3)+L_2L_3\sin(q_3)\dot{q}_1\dot{q}_3-m_3L_2L_3\sin q_3\ \dot{q}_3^2-m_3L_2L_3\sin q_3\dot{q}_2\dot{q}_3$$

On the other hand, each motor *i* is subjected to wrenches defined by:

$$\mathbf{D}_i(\boldsymbol{q})\ddot{\boldsymbol{q}}+\mathbf{V}_i(\boldsymbol{q},\dot{\boldsymbol{q}})+\mathbf{K}_a f_i=\boldsymbol{\tau}_{mi} \tag{B4}$$

Where τ_{mi} and f_i are the torque produced by the DC motor, and the force actuating the link *i* of the robot. In a matrix form, one has:

$$\mathbf{D}_a(\boldsymbol{q})\ddot{\boldsymbol{q}}+\mathbf{V}_a(\boldsymbol{q},\dot{\boldsymbol{q}})+\mathbf{K}_a f=\boldsymbol{\tau}_m \tag{B5}$$

With $$\mathbf{D}_a=\mathrm{D}_a\boldsymbol{I}_{3\times3}=\frac{2\pi}{np}(J_s+n^2J_m)\boldsymbol{I}_{3\times3},$$

$$\mathbf{V}_a=\mathrm{V}_a\boldsymbol{I}_{3\times3}=\frac{2\pi}{np}(b_s+n^2b_m)\boldsymbol{I}_{3\times3}, \text{ and} \tag{B6}$$

$$\mathbf{K}_a=\mathrm{K}_a\boldsymbol{I}_{3\times3}=\frac{p}{2\pi n}\boldsymbol{I}_{3\times3}$$

while $\mathbf{D}_a, \mathbf{V}_a, \mathbf{K}_a$ are respectively, the actuators inertia matrix, the actuator viscous damping matrix, and actuators gain matrix, $\mathrm{D}_a, \mathrm{V}_a, \mathrm{K}_a$ represent their respective elements. f is the 3-vector of the input forces in the joint space. Now, lets define the following generalized inertia matrix and Coriolis and centrifugal wrenches.

$$\mathbf{D}(\boldsymbol{q}) = \mathbf{D}_R(\boldsymbol{q}) + \mathbf{D}_a(\boldsymbol{q}),$$

$$\mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{V}_R(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \mathbf{V}_a(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{B7}$$

Then the dynamic model of the robot including friction and actuator models is thereby obtained by grouping eqs. (B3) to (B7) yielding:

$$\mathbf{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) + \mathbf{F}_{fric}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} \tag{B8}$$

This equation corresponds to the robot inverse dynamics including actuators and friction models given by eq. (2).

5. 11. References

[1] D. Tesar, "Where is the Field of Robotics Going?", *ICRAM '99*, Istanbul Turkey, pp. 24-26, 1999.

[2] C. Ahrikencheikh and A. Seireg: "*Optimized-Motion Planning*", John Wiley &Sons, 1994.

[3] M. Vukobratovic, D. Stokic: "Is Dynamic Control Needed in Robotics System? and If So to What Extent? ", *Int'l Jou. of Robotics Research* 2(2): 18-35, 1983.

[4] A. Khoukhi: "An Optimal Time-Energy Control Design for a Prototype Educational Robot", *Robotica* 20: 661-671, 2002.

[5] J. Kieffer, A. J.Cahill, and M. R. James: "Robust and Accurate Time-Optimal Path-Tracking Control for Robot Manipulators", *IEEE Trans. Robot. Automat.* vol. 13, pp.880-890, Dec. 1997.

[6] O. Dahl: "Path-Constrained Robot Control with Limited Torques-Experimental Evaluation", *IEEE Trans. Robot. Automat.*, vol. 10, pp.658-669, Oct. 1994.

[7] J. J. E. Slotine, and W. Li: "On The Adaptive Control of Robot Manipulators", *Int. J. Robotics Research,* v 6, 3, 1987.

[8] Z. Yu, H. Chen, P. Y. Woo: "Advanced Gain Scheduled H $_{infinity}$ Controller for Robotic Manipulators", *Robotica,* v 20, n 5, p 537-544, September/October, 2002.

[9] L. Baron: "A Genetic Algorithm for Computing the Real Solutions of the Inverse Kinematics of Serial Manipulators", *Tenth World Congress on the Theory of Machines and Mechanisms,* v 3, pp. 1134-1139, Oulu, Finlande, 20-24 June 1999.

[10] M. A. C. Gill, A. Y. Zomaya: "*Obstacle Avoidance in Multi-Robot Systems: Experiments in Parallel Genetic Algorithms*", World Scientific Series in Robotics and Intelligent Systems , v 20, 1998.

[11] A. Khoukhi, A. Ghoul: "On the Maximum Dynamic Stress Search Problem for Robot Manipulators", *Robotica*, 22, 5, pp: 513-522, 2004.

[12] J. K. Son, H. Y. Sik, P. C. Kug: "Neuro-Fuzzy Control of a Robot Manipulator for a Trajectory Design", *IEEE Int'l Workshop on Robot and Human Communication,* 1997.

[13] H. Bersini, L. Gonzales, E. Decossaux: "*Hoplfield Net Generation and Encoding of Trajectories in Constrained Environments*", In Neural Networks in Robotics, Kluwer Academic Publishers 1993.

[14] A. Khoukhi, K. Demirli, L. Baron, M. Balazinski: "Fuzzy-Neuro Optimal Time-Energy Control of a Three Degrees of Freedom Manipulators", *NAFIPS'06 (North American Fuzzy Information Processing Society),* Concordia University, Montreal, Qc, Canada, June 3-6, 2006.

[15] H. Shao, K. Nonami, T. Wojtara, R. Yuasa, S. Amano, D. Waterman: "Neuro-Fuzzy Position Control of Demining Tele-Operation System Based on RNN Modeling", *Robotics and Computer Integrated Manufacturing,* 22, pp: 25-32, 2006.

[16] Y. L. Sun, M. J. Er: "Hybrid Fuzzy Control of Robotics Systems", *IEEE Trans. On Fuzzy Systems* vol. 12, No. 6, pp: 755-765, December 2004.

[17] W. S. Lin, C. H. Tsai, and C. H. Wang: "Robust Neuro-Fuzzy Model-Following Control of Robot Manipulators", *Proceedings of the IEEE Int'l Conference on Control Applications,* Trieste, Italy, September, 1998.

[18] A. Khoukhi, K. Benfreha: "Application des Réseaux Neuro-Flous Multicouches à la Détection de Collision Outils-Pieces en Tournage Simulé", (In French), *Transactions of the Canadian Society of Mechanical Engineering,* vol. 29 (3-4), pp: 431-444, January, 2004.

[19] A. Khoukhi, A. Moualek:"Genetic Agents, A New Class of Quasi-Biologic Algorithms", *International Journal of Hybrid Systems,* v 2, 4, pp: 383-399, 2002.

[20] L. X. Wang: "*A course in Fuzzy Systems and Control*", Prentice Hall, 1997.

[21] J. S. R. Jang, C. T. Sun, E. Mizutani: "*Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*", Prentice-Hall Engineering, 614 pages, 1996.

[22] S. L. Chiu: "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent and Fuzzy Systems;* vol. 2, 267-278, 1994.

[23] V. Tipsowanpron, S. Intajag, K. Witheephanich, N. Koetasm-eng, S. Samiamg: "Neuro-fuzzy controller design for industrial process controls", *Proc. of the SICE Annual Conference, SICE Annual Conference 2004*, p 547-552, 2004.

[24] M. Al-Khatib and J. J. Saade: "An Efficient Data-Driven Fuzzy Approach to the Motion Planning Problem of a Mobile Robot", *Fuzzy Sets and Systems,* v 134, no. 1, 16, pp: 65-82, 2003.

[25] J. J. Craig: "*Introduction to Robotics: Mechanics and Control*", 3rd Ed, Addison-Wesley, 2005.

[26] A. Khoukhi, L. Baron, M. Balazinski: "A Decoupled Approach to Optimal Time Energy Trajectory Planning of Parallel Kinematic Machines", *Romansy'06, 16th CISM-IFFToMM Symposium on Robot Design, Dynamics and Control,* Warsaw, Poland, June 20-24, 2006.

[27] A. Khoukhi, L. Baron, M. Balazinski: "Programmation Dynamique Time-Énergie Minimum des Robots Parallèles par Lagrangien Augmenté", (In French),

CCToMM M^3, Canadian Space Agency, Saint Hubert QC, Canada, pp:147-148, May 26-27, 2005.

[28] A. Khoukhi, L. Baron, M. Balazinski: "Constrained Multi-Objective Trajectory Planning of Parallel Kinematic Machines", Submitted to *Robotics and Computer Integrated Manufacturing*, 2006.

[29] A. Khoukhi, L. Baron, M. Balazinski: "Multi-Objective Trajectory Planning for Redundant Manipulators", Accepted at the *Fifth Int. Conference on Industrial Automation,* Montreal, Canada, June 11-13, 2007.

[30] D. P. Bertsekas: "*Non Linear Programming*", Athena Scientific, 1995.

[31] T. Rockafellar: "Lagrange Multipliers and optimality", *SIAM Review 35*, pp: 183-238, 1993.

[32] K. Demirli, S. X. Cheng, P. Muthukumaran: "Subtractive Clustering Based Modeling of Job Sequencing with Parametric Search", *Fuzzy Sets and Systems;* vol. 137, pp: 235-270, 2003.

[33] Available: www.mathworks.com/products/matlab/

CHAPITRE 6

DATA-DRIVEN NEURO-FUZZY MULTI-OBJECTIVE TRAJECTORY PLANNING OF REDUNDANT MANIPULATORS

Amar KHOUKHI, Luc BARON, Marek BALAZINSKI

Department of Mechanical Engineering, École Polytechnique de Montréal,

C. P. 6079, Succ. CV, Montreal, QC, Canada H3C 3A7

(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

and

Kudret DEMIRLI

Department of Mechanical and Industrial Engineering, Concordia University,

1455 de Maisonneuve Blvd. W., EV4.173, Montreal, QC, Canada H3G 1M8

demirli@encs.concordia.ca

6. 1. Abstract: In this paper, the problem of multi-objective trajectory planning is studied for redundant planar serial manipulators using a data-driven hybrid neuro-fuzzy system. This system has been developed in two major steps. First, an offline planning is performed to generate a large dataset of multi-criteria trajectories, covering mostly of the robot workspace. The later is based on a full kinematic and

dynamic model of the robot, which minimizes time and energy, while avoiding singularities, and limits on joint angles, rates, accelerations, jerks and torques. An augmented Lagrangian technique is implemented on a decoupled form of the robot dynamics in order to solve the resulting non-linear constrained optimal control problem. Second, the outcomes of this first step are used to build a data-driven neuro-fuzzy inference system to learn and capture the desired dynamic behavior of the manipulator. Once this system is trained and optimized, it is used to achieve the online planning. Simulation results on a 3- degrees of freedom manipulator show the effectiveness of the proposed approach.

Key words: Redundant Manipulators, Multi-Objective Trajectory Planning, Optimal Control, Augmented Lagrangian, Subtractive Clustering, Data-Driven Neuro-Fuzzy Systems.

6. 2. Introduction

6. 2. 1. Related Works

Robot optimal motion planning is a dynamic research area aiming at generating optimal trajectories for a specific task, according to a set of desired performance criteria. A feasible planning approach could be satisfactory if required target performance is not too tight. Otherwise, or when the assigned trajectory is infeasible due to actuation limits or passing through singular poses, the motion planning algorithm must be able to generate torques that allow achieving performance criteria,

while avoiding singularities and satisfying other task-related constraints [1-5]. Many researchers had addressed this problem [2-6]. The most popular performance criterion is traveling time minimization. This is due for the obvious reason of production targets. However, the major drawback of this control criterion is its bang-bang character as it produces non-smooth trajectories. Moreover, tracking a planned minimum-time trajectory had shown to lead to large joint vibrations, and overshoot of the nominal torque limits [6]. Another online motion control algorithm problem is that the desired trajectory may cause saturation of the speed and/or torques delivered by the joint actuators in the vicinity of singularities. This might also occur in any other region of the workspace, due to nonlinear kinematic transformations between task and joint spaces. This problem is more challenging when the planning is made under kinematic redundancy conditions.

Furthermore, as robot tasks are demanding in precision and diversity, the coupling between robot's axes makes even harder the design of a trajectory planning or a control system of the robot. As a matter of fact, conventional fixed gain, linear feedback controllers are not capable of effectively controlling the movements of multi-joint robot manipulators under different distance, velocity and load requirements. Through the use of non-linear feedback, approaches like computed torque and H_∞ methods provide better compensation for the dynamic interactions present in various robot motions [7, 8]. However, these approaches require complete, non-linear dynamic models describing the robot manipulator which are difficult to be accurately modeled and implemented in real-time. Because of these difficulties,

several researchers have seen soft computing techniques as attractive alternatives. For example, genetic algorithms are increasingly applied to motion planning and dynamic control of robotic systems [9-11]. Neural networks and fuzzy control perhaps constitute the most applied artificial intelligence fields in motion control systems of industrial manipulators and mobile robots [12-14]. Compared to conventional controllers, soft computing based-controllers are capable of tackling highly complex dynamics without explicit model dependence and identification. Moreover, these artificial intelligence techniques have added a new dimension to many engineering fields to study, and have effectively been applied to everything from voice and image recognition to automobile transmission and breaking systems [15-19].

Several methods had been considered for trajectory design and robot motion planning using neuro-fuzzy techniques [12-14]. Some of theme involves the use of a conventional control algorithm, e.g., PD or PID in controlling a multi-joint robot manipulator. The conventional control brings the system close to the desired state and the neuro-fuzzy learning mechanism then compensates for the remaining error. Other systems use only learning algorithms to execute the control. These learning controllers are computationally efficient after training and had been shown to have very good generalization capabilities, provided that a large enough dataset points representing the system's behaviour is given [13, 14]. Within this category of works, a data-driven neuro-fuzzy system has been introduced recently for mobile robots online motion planning [20], where the dataset was generated form real time measurements.

6. 2. 2. Proposed Approach

This paper develops a data-driven neuro-fuzzy system to multi-objective trajectory planning of an *n*- degrees of freedom (*n*-DOF) planar serial manipulator. The best possible manipulator trajectories can be obtained by a constrained optimal control of the robot non linear kinematic and dynamic models, while optimizing a multi-criteria performance index and satisfying several constraints relating robot, workspace and task interactions. Furthermore, simulated data provide reliable training and comparison for the neuro-fuzzy system and allow testing of a specific parameter's impact on the overall performance, by changing only the value of the suitable parameter. The present work is an extension of a recent work [21], where a data-driven neuro-fuzzy system had been built based on robot simulation outcomes. The main limitations in [21] were due essentially to the inverse kinematic solution, as one had to limit the system's task to either elbow up or elbow down configuration. In the present work, the redundancy is exploited to solve the inverse kinematics, while satisfying a secondary task of avoiding joint limits. Two main parts compose the proposed approach:

The first one is a dataset building from an offline multi-objective trajectory planning. This part contains three steps. First, an initialization unit defines a feasible solution satisfying boundary constraints in joint angles, velocities, accelerations and jerks. This solution is determined through a cycloidal profile [21, 22]. The obtained trajectory is put into the robot's inverse dynamic model to get the corresponding torques. Second, the generated trajectory outcomes are used to initialize for a multi-

objective optimal control problem under limits on EE poses, as well as other constraints such as actuator torque, sampling period limits, and passing through imposed poses. Third, the resulting optimal control problem is discretized and an augmented Lagrangian technique is used to solve it. These steps are done as many times as possible to cover mostly of the robot workspace. The generated trajectories are gathered in an input/output dataset. The inputs are the EE positions and velocities, and the outputs are the corresponding joint actuator torques and sampling periods.

The second major part of the proposed approach is an adaptive neuro-fuzzy system, named NeFuMOP (for Neuro-Fuzzy Multi-Objective Planning). It starts with a subtractive clustering of the so-generated dataset into clusters. By the same token, this allows initialisation of membership function parameters and number of rules of a neuro-fuzzy system. The last step is to optimize NeFuMOP in order to learn and capture the dynamic multi-objective behaviour of the robot. Figure 6. 1 shows the overall proposed scheme for online trajectory planning. The whole purpose of the proposed approach is that from a pure but representative model and offline trajectory planning outcomes, a neuro-fuzzy system is built. Once the neuro-fuzzy structure (rules number and premise and consequence membership function parameters) is identified and optimized, it is used in a generalization phase to achieve online planning with a reasonable computational complexity. In section II, the multi-objective offline planning and dataset building procedures are considered. In section III, the data-driven neuro-fuzzy system NeFuMOP is developed. Section IV presents simulation results, and finally, section V concludes the paper.

6. 3. Offline dataset building

In this section the multi-objective offline trajectory planning is developed. We consider an *n*-degrees-of-freedom (DOF) planar serial manipulator operating in a two dimensional positioning space. The degree of redundancy of the manipulator is then *n-2.* All the joints of the manipulator are actuated. The robot task is to position its EE without considering its orientation within a limited workspace and time interval. Figure 6. 2 shows an example of a 3 DOF, which serves as a case study to implement our approach.

6. 3. 1. Kinematic Modelling

The forward kinematics for position displacement might be given for an *n*-DOF planar manipulator as:

$$x(t) = \sum_{i=1}^{n} L_i \cos(q_1(t) + ... + q_i(t)) \tag{1}$$

$$y(t) = \sum_{i=1}^{n} L_i \sin(q_1(t) + ... + q_i(t)) \tag{2}$$

At the velocity level, the forward kinematics might be expressed as [4] :

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{q}}, \tag{3}$$

where $\boldsymbol{q}(t) = [q_1(t), q_2(t), ..., q_n(t)]^T$ is the joint position vector, $\mathbf{x}(t) = [x(t), y(t)]^T$ is the position vector of the EE, and $\mathbf{J}$ is the robot's 2x*n* Jacobian matrix. Since $\mathbf{J}$ is row rectangular, the inverse kinematics is redundant. This redundancy can be used to solve the inverse kinematics as [5]

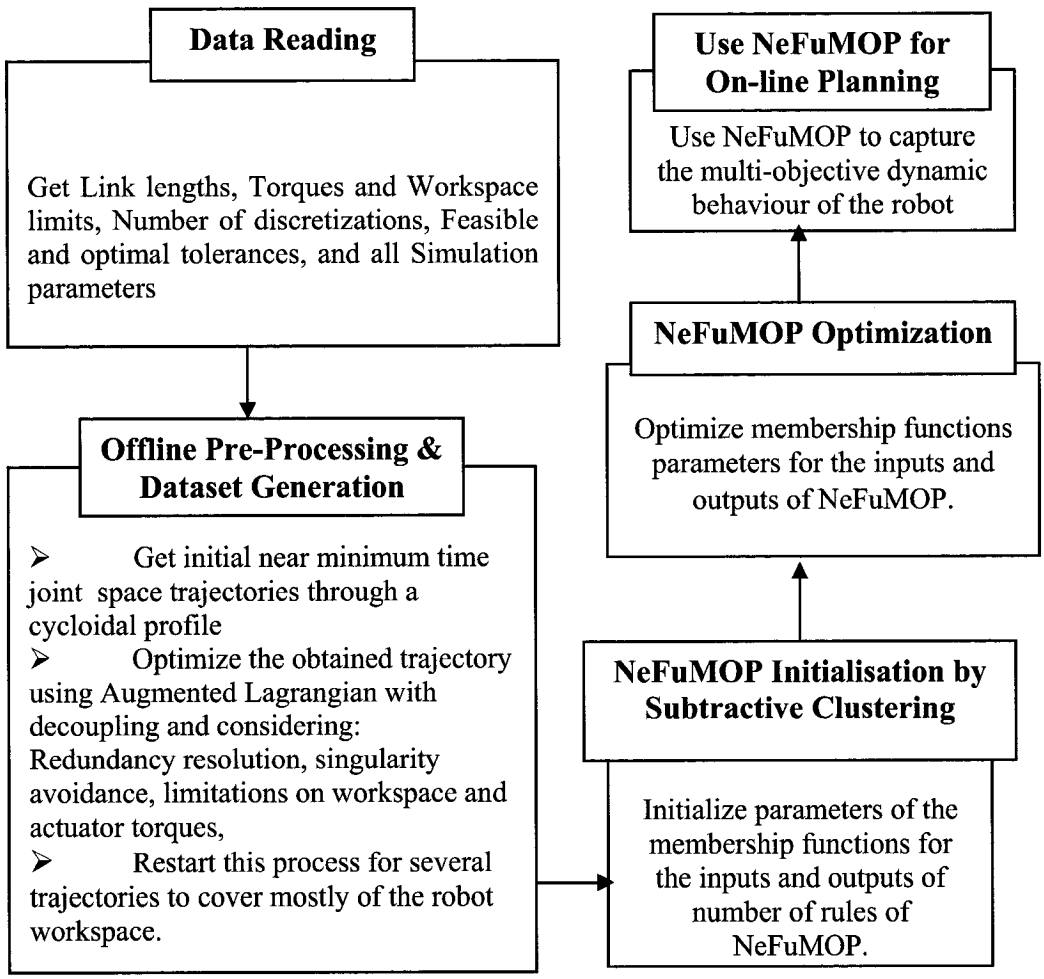


Fig. 6. 1. Overall proposed multi-objective planning system

$$\dot{q} = \mathbf{J}^{\perp}\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{\perp}\mathbf{J})\mathbf{z} \tag{4}$$

where $\mathbf{J}^{\perp}$ is a generalized inverse of $\mathbf{J}$. The pseudo-inverse solution in the first term of eq. (4) is known to minimize the two-norm of the joint velocity vector. The second term is called the homogeneous or null-space solution. It has been shown that solving

(4) using a pseudo-inverse of **J** sometimes leads the robot to singularities [23]. A weighted pseudo-inverse by the inertia matrix is used here, allowing dynamic consistency, i.e,

$$\mathbf{J}_{\mathbf{D}}^{\#}=\mathbf{D}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{D}^{-1}\mathbf{J}^{T})^{-1} \tag{5}$$

In including joint limits avoidance as a secondary task criterion in redundancy resolution, **z** can be as

$$\mathbf{z}=-\mathbf{W}(\boldsymbol{q}-\bar{\boldsymbol{q}}) \tag{6}$$

with

$$\bar{\boldsymbol{q}}=\frac{1}{2}(\boldsymbol{q}_{\mathrm{Max}}+\boldsymbol{q}_{\mathrm{Min}}), \tag{7}$$

and **W** is a weighting matrix to scale the magnitude of the manipulator response to joint displacement. A typical choice for this matrix is

$$\mathbf{W}=\mathrm{diag}\,(\boldsymbol{q}_{\mathrm{Max}}-\boldsymbol{q}_{\mathrm{Min}}) \tag{8}$$

6. 3. 2. Dynamic Modelling

The robot dynamic model is developed with Lagrangian formalism, including actuators model [1, 2]

$$\mathbf{D}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\mathbf{V}(\boldsymbol{q},\dot{\boldsymbol{q}})\ddot{\boldsymbol{q}}+\mathbf{G}(\boldsymbol{q})=\boldsymbol{\tau} \tag{9}$$

where $\boldsymbol{\tau}$ is the torques vector produced by joint actuators, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ describe joint angles, rates and accelerations, $\mathbf{D}(\boldsymbol{q})$ is the inertia matrix, $\mathbf{V}(\boldsymbol{q},\dot{\boldsymbol{q}})$ represents Coriolis and centrifugal wrenches, and $\mathbf{G}(\boldsymbol{q})$ is the gravity force vector.

The discrete-time model might be approximated as [2]

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_{dk}\boldsymbol{x}_k + \boldsymbol{B}_{dk}\left[\boldsymbol{\tau}_k - \left[\mathbf{V}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \mathbf{G}(\boldsymbol{x}_{1k})\right]\right] \tag{10}$$

where

$$\boldsymbol{F}_{dk} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & h_k \boldsymbol{I}_{n\times n} \\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n} \end{bmatrix}, \quad \boldsymbol{B}_{dk} = \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{n\times n} \\ h_k \boldsymbol{I}_{n\times n} \end{bmatrix} \tag{11}$$

and $\boldsymbol{x}_k=(\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)^T=(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})^T$ is the $2n$-dimensional robot state. Equation (10) is written for simplicity as

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k\ (\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \tag{12}$$

6. 3. 3. Constraints Modeling

In addition to eq. (10), one has the following constraints:

- Sampling period limits $\quad h_{\text{Min}} \le h_k \le h_{\text{Max}}$ (13)
- Boundary conditions $\quad \boldsymbol{x}_0 = \boldsymbol{x}_S,\ \boldsymbol{x}_N = \boldsymbol{x}_T$ (14)
- Actuator torque limits $\quad \boldsymbol{\tau}_{\text{Min}} \le \boldsymbol{\tau}_k \le \boldsymbol{\tau}_{\text{Max}}$ (15)
- Redundancy resolution while avoiding joint limits: This is done using eqs. (4)-(5) as

$$\boldsymbol{x}_{2k} = \mathbf{J}_{\text{Dk}}^{\perp}(\boldsymbol{x}_{1k})\dot{\mathbf{x}}_k + \left[\mathbf{I} - \mathbf{J}_{\text{Dk}}^{\perp}(\boldsymbol{x}_{1k})\mathbf{J}_k(\boldsymbol{x}_{1k})\right]\mathbf{z}_k(\boldsymbol{x}_{1k}) \tag{16}$$

with $\quad \mathbf{z}_k = -\mathbf{W}\ (\boldsymbol{x}_{1k} - \bar{\boldsymbol{x}}_{1k})$, and

$$\mathbf{W} = \mathrm{diag}(\bar{\boldsymbol{x}}_1) = \mathrm{diag}(\frac{\boldsymbol{x}_{1\mathrm{Max}} - \boldsymbol{x}_{1\mathrm{Min}}}{2}) \tag{17}$$

- Imposed passages through intermediate EE poses

$$\left\| \mathbf{p} - \mathbf{p}_l \right\| - \mathrm{T}_{\mathrm{PassTh}lp} = 0, l = 1,...,L, \tag{18}$$

where $\mathbf{p}$ is the current position of the EE, $\mathbf{p}_l$ is the lth passage point, L is the number of imposed passages and $\mathrm{T}_{\mathrm{PassTh}lp}$ is the passage tolerance. All equality constraints are written as $\boldsymbol{s}_i(\boldsymbol{x}) = 0$, $i = 1,...,\mathrm{I}$ and inequality constraints as $\mathbf{g}_j(\boldsymbol{x}, \boldsymbol{\tau}) \le 0$, $j = 1,...,\mathrm{J}$, regardless if they depend only on state, control inputs or both.

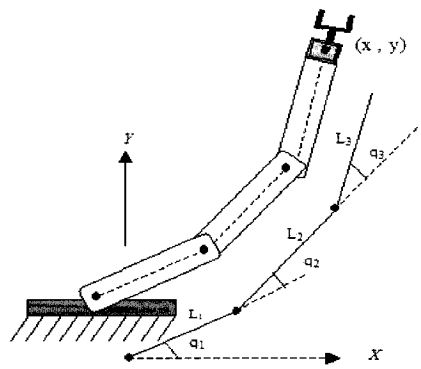


Fig. 6. 2. Geometry of a 3-dof planar manipulator

6. 3. 4. Performance Index

In discrete-time, the constrained optimal control problem consists of finding the optimal sequences $(\boldsymbol{\tau}_0, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_{N-1})$ and $(h_0, h_2, ..., h_{N-1})$, allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a target state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost E_d

$$\underset{\substack{\tau \in \mathrm{C} \\ h \in \mathrm{H}}}{\mathrm{Min}} \left\{ E_d = \left[\sum_{k=1}^{N} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \boldsymbol{\iota} + \boldsymbol{\delta}\, \varpi(\boldsymbol{x}_{1k})] h_k \right] \right\} \tag{19}$$

where $\mathrm{C}, \mathrm{H}, \boldsymbol{U}$, $\boldsymbol{\iota}$ and $\boldsymbol{\delta}$ are, respectively, the set of admissible torques, admissible sampling periods, electric energy weight, a scalar time weight, and a weight factor for singularity avoidance, h_k is the EE traveling time between two successive discrete poses k and $k+1$, $k = 1, ..., N$, while the overall traveling time being $T = \sum_{k=1}^{N} h_k$, and ϖ is a singularity avoidance function defined as [5]:

$$\varpi(\boldsymbol{x}_{1k}) = \frac{1}{\sqrt{\det(\mathbf{J}(\boldsymbol{x}_{1k})\mathbf{J}^T(\boldsymbol{x}_{1k}))}} \tag{20}$$

The optimization is done under constraints (12)-(18).

6. 3. 5. Augmented Lagrangian with Decoupling (ALD)

The multi-objective non-linear optimal control problem (19) is solved using an augmented Lagrangian (AL) technique transforming the constrained problem into a

non-constrained one, where the degree of penalty for violating the constraints is regulated by penalty parameters [24-26]. It basically relies on quadratic penalty methods, but reduces the possibility of ill conditioning of the sub-problems that are generated with penalization by introducing explicit Lagrange multipliers estimates at each step in the function to be minimized. In developing the first order optimality conditions and computing the co-states λ_k, an inverse of the inertia matrix and its derivatives with respect to state variables are required, resulting in huge calculations. For this reason, the AL is implemented on a decoupled dynamics of the robot [27].

Theorem: Under the invertibility condition of the inertia matrix, the control law defined as

$$\boldsymbol{u} = \mathbf{D}(\boldsymbol{x}_1)\boldsymbol{v} + \mathbf{V}(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{x}_2 + \mathbf{G}(\boldsymbol{x}_1) \tag{21}$$

allows the robot to have a linear and decoupled behaviour with a dynamic equation:

$$\dot{\boldsymbol{x}}_2 = \boldsymbol{v} \tag{22}$$

where $\boldsymbol{v}$ is an auxiliary input.

This can be demonstrated by substituting the proposed control law (21) into the dynamic model (9), one gets $\mathbf{D}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 = \mathbf{D}(\boldsymbol{x}_1)\boldsymbol{v}$. Since $\mathbf{D}$ is invertible, it follows that $\dot{\boldsymbol{x}}_2 = \boldsymbol{v}$.

This gives the robot a decoupled and linear behavior described by the following linear discrete-time equation:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_{dk}\boldsymbol{x}_k + \boldsymbol{B}_{dk}\boldsymbol{v}_k = \boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \tag{23}$$

with $\boldsymbol{F}_{dk}$ and $\boldsymbol{B}_{dk}$ being given by eq. (11)

Notice that while this dramatically eases the calculation of the co-states, the non-linearity is transferred to the objective function. The decoupled problem consists then of finding the optimal sequences of sampling periods and accelerations minimizing the augmented Lagrangian (ALD) function

$$L_{\mu}^{\mathrm{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k, \boldsymbol{\lambda}_k, \boldsymbol{\rho}_k, \boldsymbol{\sigma}_k) = \sum_{k=0}^{N-1} h_k \Bigg[\iota + \delta\varpi(\boldsymbol{x}_{1k}) +$$

$$\left[\mathbf{D}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \mathbf{V}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) + \mathbf{G}(\boldsymbol{x}_{1k})\right]^T \boldsymbol{U}\left[\mathbf{D}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \mathbf{V}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) + \mathbf{G}(\boldsymbol{x}_{1k})\right]\Bigg] +$$

$$\sum_{k=0}^{N} \boldsymbol{\lambda}_{k+1}^T \left[\boldsymbol{x}_{k+1} - \boldsymbol{f}_k^{\boldsymbol{D}}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)\right] + \sum_{k=0}^{N} h_k \left[\sum_{j=1}^{2} \Phi_{\boldsymbol{\mu}_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{v}_k)) + \Psi_{\boldsymbol{\mu}_s}(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i(\boldsymbol{x}_k))\right] \quad (24)$$

where $\boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k)$ is defined by the decoupled state eq. (23), N is the total sampling number, $\boldsymbol{\lambda} \in R^{2nN}$ designates the co-states, $\boldsymbol{\sigma}$ and $\boldsymbol{\rho}$ are Lagrange multipliers with appropriate dimensions, associated to equality and inequality constraints and $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_g$ are the penalty coefficients. The used penalty functions combine penalty and dual methods allowing relaxation of inequality constraints as soon as they are satisfied. These are given by

$$\Psi_{\mu_s}(\boldsymbol{a},\boldsymbol{b}) = (\boldsymbol{a} + \frac{\boldsymbol{\mu_s}}{2}\boldsymbol{b})^T\boldsymbol{b}\,, \text{ and } \Phi_{\mu_g}(\boldsymbol{a},\boldsymbol{b}) = \frac{1}{2\boldsymbol{\mu}_g}\left\{\left\|\text{Max}(0, \boldsymbol{a} + \boldsymbol{\mu}_g\,\boldsymbol{b})\right\|^2 - \left\|\boldsymbol{a}\right\|^2\right\} \tag{25}$$

The Karush-Kuhn-Tucker first order optimality conditions state that for a trajectory $(\boldsymbol{x}_0, \boldsymbol{v}_0, h_0, ..., \boldsymbol{x}_N, \boldsymbol{v}_{N-1}, h_{N-1})$, to be optimal solution to the problem, there must exists some positive Lagrange multipliers $(\boldsymbol{\lambda}_k, \boldsymbol{\rho}_k)$, unrestricted sign multipliers $\boldsymbol{\sigma}_k$ and finite positive penalty coefficients $\boldsymbol{\mu} = (\boldsymbol{\mu}_S, \boldsymbol{\mu}_g)$ such that:

$$\frac{\partial L_\mu^D}{\partial \boldsymbol{x}_k} = 0, \frac{\partial L_\mu^D}{\partial \boldsymbol{v}_k} = 0, \frac{\partial L_\mu^D}{\partial h_k} = 0, \frac{\partial L_\mu^D}{\partial \boldsymbol{\lambda}_k} = 0, \frac{\partial L_\mu^D}{\partial \boldsymbol{\rho}_k} = 0, \frac{\partial L_\mu^D}{\partial \boldsymbol{\sigma}_k} = 0,$$

$$\boldsymbol{\sigma}_k^T \boldsymbol{s}(\boldsymbol{x}) = 0\,,\ \boldsymbol{\rho}_k^T \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{v}, h) = 0 \text{ and } \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{v}, h) \le 0 \tag{26}$$

The development of the above conditions enables one to derive the iterative formulas to solve the optimal control problem by updating control variables, Lagrange multipliers and penalty coefficients. These expressions are quite long and are not detailed here. In implementing ALD procedure, after initialization through the cycloidal profile [21], an inner optimization loop solves for the AL minimization with respect to sampling periods and acceleration variables. One first computes the co-states backwardly and updates control inputs and states. Then, all equality and inequality constraints are tested against feasibility tolerances. If non-feasible, go back to the inner optimization unit. Else, if feasible, do a convergence test for cost minimization and constraints satisfaction against optimal tolerances. If convergence

holds, display optimal trajectory and end the program. If non-convergence, go further to the dual part of AL to update Lagrange multipliers, penalty and tolerance parameters to force the subsequent primal iterates to be increasingly accurate solutions to the primal problem. More details on ALD implementation can be found in [24-26].

6. 4. Data-Driven Neuro-Fuzzy Multi-Objective Online Planning (NeFuMOP)

The outcomes of the offline trajectory planning system developed in the previous section are used to obtain an input/output dataset on which to build a neuro-fuzzy system, called NeFuMOP (standing for Neuro-Fuzzy Multi-Objective Online Planner). This neuro-fuzzy controller is trained to capture the multi-objective dynamic trajectory behaviour and generate the associated actuator torques and sampling periods. Schematic representations of NeFuMOP operation and architecture are depicted in Figs. 6. 3 and 6. 4.

6. 4. 1. NeFuMOP Structure and Architecture

A. Rule Base and Membership Functions

NeFuMOP is a data-driven neuro-fuzzy system based on Tsukamoto fuzzy inference mechanism. In this mode of reasoning, the consequence linguistic terms are supposed to have a continuous strong monotone membership function. The multiple input multiple output (MIMO) objective fuzzy model consists of four inputs and four outputs. The inputs are the discrete Cartesian 2-dimentional positions and velocities

as $\mathbf{x}_d = \left[x_i,\ y_i,\ \dot{x}_i, \dot{y}_i \right]\ i = 0,...,N$. The outputs consist of the joint torques and sampling periods given as $\boldsymbol{\pi}_d = \left[\boldsymbol{\pi}_1, \boldsymbol{\pi}_2,..., \boldsymbol{\pi}_N\right]^T$, $\boldsymbol{\pi}_i = \left[\boldsymbol{\tau}_i, h_i\right]^T$, with $\boldsymbol{\tau}_i$ being the vector of joint torques and h_i the sampling period at the discrete time *i*. Consider the following input/output dataset entries $\left[(\mathbf{x}_d^1, \boldsymbol{\pi}_d^1),..., (\mathbf{x}_d^K, \boldsymbol{\pi}_d^K)\right]$. The fuzzy rules are defined in the following format:

R^1 :If (x_1 is A_1^1) and (x_2 is A_2^1) and ...and (x_4 is A_4^1) Then ($o_1 = c_1^1$) and ($o_2 = c_2^1$) ...and ($o_4 = c_4^1$)

Also

...

R^j: If (x_1 is A_1^j) and (x_2 is A_2^j) and...and (x_4 is A_4^j) Then ($o_1 = c_1^j$) and ($o_2 = c_2^j$)...and ($o_4 = c_4^j$)

Also

R^J :If (x_1 is A_1^J) and (x_2 is A_2^J) and...and (x_4 is A_4^J) Then ($o_1 = c_1^J$) and ($o_2 = c_2^J$)...and ($o_4 = c_4^J$)

where x_l is the input to node *l*, A_l^j is the associated linguistic term. The membership functions $\mu_{A_l^j}$ defining the fuzzy sets A_l^j are Gaussian functions given as:

$$\mu_{A_l^j}(x_l) = \exp\frac{-1}{2}\left\{\frac{(x_l - a_l)^2}{(b_l)^2}\right\},\quad l=1,\ldots,4,\ j=1,\ldots,J \tag{27}$$

where a_i^j and b_i^j are the mean and standard deviation of the jth membership function of the input variable x_i, and c_t^j are fuzzy sets defining the consequence of the jth rule, such that:

$$c_t^j = f^{-1}(\mu_{A_i^j}(x_i)), f: \Re \to [0,1],\ t = 1,...,4,\ j=1,\ldots,J \tag{28}$$

f being a continuous and strong monotone function, such that it has an inverse function. A typical choice of this function is:

$$f(z) = \frac{1}{1+e^{-(\rho z-\sigma)}}, \tag{29}$$

where ρ and σ are real numbers affecting respectively the position and slope of the inflection point of f.

B. NeFuMOP Architecture

NeFuMOP is made of six layers (Fig. 6. 4). The function of each layer is explained below:

In the first layer, each node corresponds to one input variable. They are defined as:

$$o^1(x_i) = x_i,\ \ i = 1,...,4 \tag{30}$$

The role of these nodes is only to pass the crisp input signals to the next layer.

The second layer does the fuzzification of the input data into fuzzy numbers. Each node is shown as a square, and the Gaussian membership functions defined in equation (27) specifies the degree to which the given x_i satisfies the quantifier A_i^j,

$$o_j^2 = \mu_{A_i^j}(x_i), \quad l=1,\ldots,4, \quad j=1,\ldots,J \tag{31}$$

The third layer associates a rule to each node, namely, Π, representing the pre-condition of one fuzzy logic rule. Therefore, each node multiplies the incoming signals (Product operator is chosen for t-norm). Each node output α_j of this layer represents a rules firing strength, which relates layer 3 to layer 4,

$$o_j^3 = \alpha_j, \text{ with } \alpha_j = \prod_{l=1}^{4} \mu_{A_l^j}(x_l), \tag{32}$$

The fourth layer does the normalization, and is labelled N. The jth node calculates the ratio of the jth rule's firing strength to the sum of all rules firing strengths:

$$o_j^4 = \beta_j \text{ where } \beta_j = \frac{\alpha_j}{\sum_{j=1}^{J} \alpha_j}, \quad j=1,\ldots,J \tag{33}$$

The fifth layer does the inversion. Every node in this layer is a square with an outcome given by:

$$o_j^5 = \beta_j f_j^{-1}(\alpha_j), \quad j = 1,..., J \tag{34}$$

where β_j are the normalized firing strengths from layer 4 associated with the jth rule, and f_j^{-1} is the inverse function describing the consequent part.

The sixth and last layer acts as an aggregator. The nodes of this layer are labelled Σ. At each node, a system output is obtained as the summation of all incoming signals,

$$o^6 = z = \sum_{j=1}^{J} \beta_j f_j^{-1}(\alpha_j) \tag{35}$$

This inference mechanism is known as Tsukamoto reasoning [16-18].

6. 4. 2. NeFuMOP Learning Procedure

The learning scheme of the objective fuzzy model NeFuMOP is composed of two steps: structure identification where the rules number and membership function parameters are initialized, and structure optimization, where these parameters are optimized.

A. Structure Identification

At this phase the structure parameters are initialized through partitioning the input/output dataset into clusters. The purpose of clustering is to identify natural grouping of data from the generated large data set to produce a concise representation of the robot's behavior. This results in initial rules that are more tailored to the input data. That, with clustered training points, the network is very accurate while dealing with points near the cluster, and should breed to the objective fuzzy model of the robot familiarity with its workspace. Assume that we don't have a clear idea on how many clusters there should be for the generated offline planning dataset. To find the number of clusters and the cluster centers, we apply the subtractive clustering algorithm [28, 29, 30]. This approach provides a fast; one-pass algorithm to take input-output training data to generate an objective fuzzy inference system that captures the robot dynamic behaviour. It starts with a normalization of all values in the dataset to fit in a hypercube unit. Each cluster center may be translated into a fuzzy rule for identifying a class. The cluster radius indicates the range of its influence. Specifying a small cluster radius yields many small clusters in the data, thus resulting in many rules, and *vice versa*. This procedure allows initialization of the premise membership function parameters a_i , b_i and the consequent parameters ρ_i, σ_i of a fuzzy rule R^j . The Matlab function *subclust* is used to initialize and identify these parameters.

B. Structure Optimization

The outcomes of structure identification are treated as approximate values for the membership function parameters. To optimize these parameters, a hybrid algorithm combining a gradient descent and least square estimate (LSE) is used as discussed in [17]. The Levenberg-Marquard version of the gradient back-propagation learning rule is used to fine-tune the parameters in the hidden layer and the least square method is used to optimize the output layer parameters. The cost function considered minimizes the error between the computed outputs by the network and the desired ones relating the robot dynamic behaviour. It is written as:

$$E = \frac{1}{2}\sum_{k=1}^{K}\left\|\mathbf{o}^k - \boldsymbol{\pi}^k\right\|^2 = \frac{1}{2}\sum_{k=1}^{K}\sum_{j=1}^{J}\sum_{l=1}^{4}(\mathbf{o}_{jl}^k(a_l, b_l, \rho_l, \sigma_l) - \boldsymbol{\pi}_{jl}^k)^2 \tag{36}$$

where $\mathbf{o}^k$ is the computed output from the fuzzy system, $\boldsymbol{\pi}_d^k$ is the kth desired output associated to the kth training and dataset entry J is the number of rules.

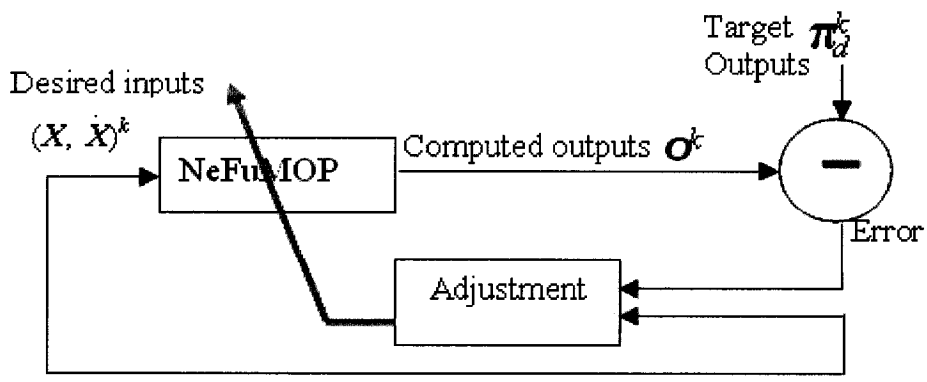


Fig. 6. 3. Learning of the Neuro-Fuzzy Multi-Objective Planning (NeFuMOP)

6. 5. Simulation Results

The considered robot is a 3 revolute serial redundant planar manipulator moving on the vertical plan shown in Figure 6. 2. The robot numerical values are given in Tables 6. 1 and 6. 2.

6. 5. 1. Training Protocol

The general training protocol is as follows: In order for the neuro-fuzzy system to learn mostly of the robot workspace and gets high generalization capabilities, the training dataset is made of many trajectories from different EE starting and ending Cartesian positions. This is made possible through the implementation of ALD technique to solve the non-linear and non-convex optimal control problem.

Because the order in which the points are presented to the network affects the speed and quality of learning convergence, and since the robot interpolates between sampling points to build the entire path from a starting to an ending points, this coverage is made while ensuring singularity avoidance and redundancy resolution. Moreover, the training points are equally distributed throughout the workspace. One hundred trajectories had been achieved from different initial to final 2-dimentional Cartesian positions of the EE. Each trajectory is sampled through 20 discretization points. Hence, one turns up with 2000 data points to perform the training of NeFuMOP, with 90% used for training and 10% for testing.

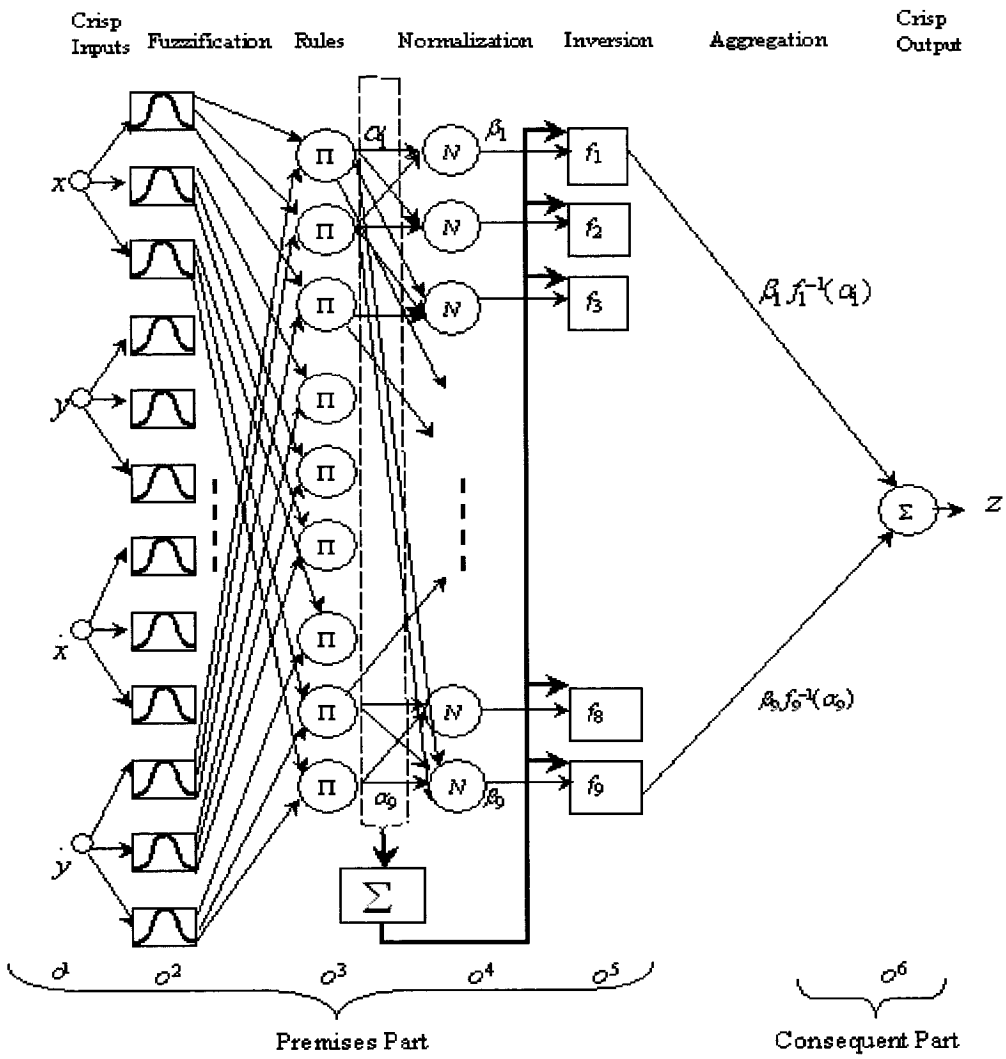


Fig. 6. 4. NeFuMOP architecture

6. 5. 2. Offline Planning Dataset

In the offline trajectory generation, the simulation objectives are to: 1) minimize travelling time, instantaneous kinetic and electric energy and maximizes a singularity avoidance function, during the motion; 2) satisfy several constraints related to limits on joint angles, rates, accelerations and torques; and 3) resolve redundancy. The program is coded in Matlab. All trajectories are performed by taking each weight equals to unity in the performance index.

A task example is to move the tip of the manipulator along a straight line from (1.95, 0.83) (m) corresponding to joint position (0°, 30°, 30°) to the final Cartesian position (1.35, 1.35) (m). The maximum allowed time for this trajectory is 3 sec. The initial pose values are chosen to satisfy a secondary goal, which consists of avoiding joint limits. Figure 6. 5 shows the displacements along x and y axes. The slight bias between the desired straight line and the computed path using ALD is due to the non-linearity of the dynamic model, considering for example gravity effects. In Fig. 6. 6 (a), the plots show the instantaneous variations of consumed time and energy necessary for both cycloidal and ALD trajectories. The consumed time for ALD after 5 outer and 4 inner iterations, is dramatically less than the initial trajectory done with the cycloidal profile, an increase of 40% for the cycloidal profile with $t_{\text{Cycloidal}} = 2\,\text{sec}$ and $t_{\text{Multi-objective}} = 1.175\,\text{sec}$. Moreover, while the cycloidal profile allows a feasible path, the associated energy, computed from the robot inverse dynamics are fairly high and exceed the nominal values. Figure 6. 6 (b) illustrates a good robustness to parameter

changes by increasing the value of the third link mass by an amount of 1.75 (Kg). Figure 6. 7 plots the instantaneous variations of joint torques. Figure 6. 8 shows the (x, y) coordinate variations while satisfying passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m), (in the Figure, the cycloidal trajectory does not consider this constraint). Figure 6. 9 shows the consumed energy and traveling time for the trajectory with passage points.

The second trajectory is a circle with a center (-0.045, 0.75) (m) and radius 0.14 (m). The joint angle variations are shown in Fig 6. 10, and the consumed energy and traveling time in Fig. 6. 11. On the whole, the computation time is quite long to find an optimal solution. It took about 13 minutes on a Pentium III, 996 MHz, to get the performances of Figs 6. 5 and 6. 6 (a). This is due mainly to the high nonlinear robot dynamics and constraints. However, this is not too jeopardizing, as the purpose at this level is offline programming.

6. 5. 3. Dataset Clustering

A subtractive clustering algorithm partitions the so-generated input-output dataset. After several trials, the following subtractive clustering parameters had been set: The upper acceptance threshold for a data point to be a cluster center is 1.0. The lower rejection ratio is 0.8. The cluster radius is 0.9, and the squash parameter is 0.5. Figure 6. 12 depicts the membership functions defining the Cartesian x and y position coordinates and velocities, before and after NeFuMOP training.

6. 5. 4. NeFuMOP Performance

The learning results show good performance with an error of order of 10^{-3} in less than 10 training epochs, as illustrated in Fig. 6. 13 and Table 6. 3. NeFuMOP required 213 seconds to learn from the overall 2000 data points the specified trajectory, and capture the robot dynamic behaviour associated to the multi-objective planned trajectories. To test the learning and generalization and robustness abilities of NeFuMOP, a preliminary test is performed with arbitrary small modifications (0.5%) of each of the 10% testing data points. Table 6. 4 shows the learning performance achieved by NeFuMOP, with the modified data getting an error in an order of 10^{-2} after 40 training epochs, which highlights the good generalization capabilities of NeFuMOP.

In order to assess the generalization and robustness capabilities of the proposed system, two tests had been performed. The first one is made on a trajectory that had been learnt by the ALD offline technique, which consists to move the robot tip along a straight line, starting from the Cartesian coordinates (0.2, 2) (m) corresponding to the joint position (0^0, 15^0, 20^0) to the ending position (0.7, 1.3) (m). The ALD technique was used to achieve the offline planning. The actuator torques and sampling period variations can be obtained from the outcomes $\mathbf{o}^k$ of NeFuMOP. The corresponding trajectory and consumed energy are shown in Figs. 6. 14 and 6. 15. These figures show the very good generalization capabilities of NeFuMOP.

Table 6. 1. Kinematic and dynamic parameters of the manipulator

Link I	Mass (kg)	Inertia I_i(kg m^2)	Length L_i(m)
1	10	9.23	1.0
2	8	7.5	0.8
3	4	5.21	0.45

Table 6. 2. Limits of workspace, actuator torques, joint rates, accelerations and sampling periods

Parameter	x (m)	y (m)	τ_1 (Nm)	τ_2 (Nm)	τ_3 (Nm)	$\dot{q}_{iMotor}$ (rad/s)	$\ddot{q}_{iMotor}$ (rad/s^2)	h (sec)
Max	0.9	-1.720	40	25	20	2	3	0.1
Min	-0.9	0.710	-40	-25	-20	-2	-3	0.005

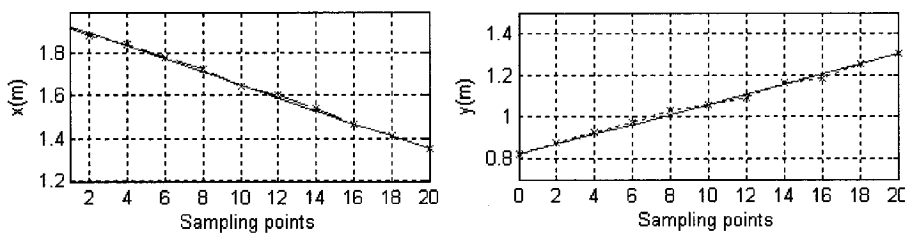
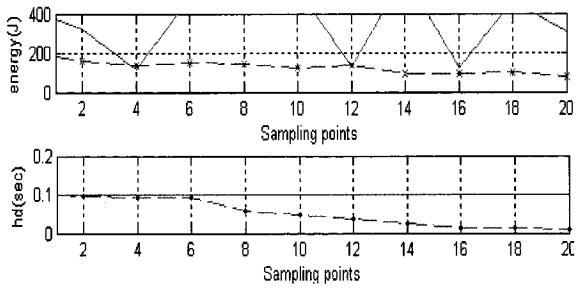
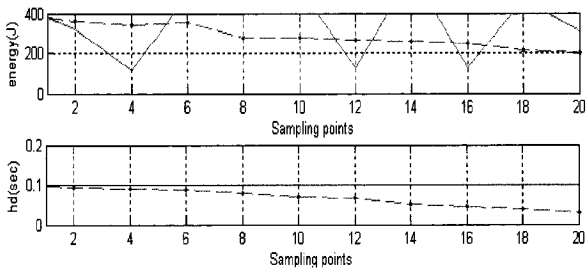


Fig. 6. 5. Displacement along x-axis and y-axis of the EE, (__desired trajectory, ._._computed trajectory)



(a) Trajectory with third link mass 2 Kg



(b) Trajectory with a third link mass increased by 1.5 Kg
Fig. 6. 6. Consumed energy and time vs. sampling points
(__Cycloidal profile trajectory, ._._ALD trajectory)

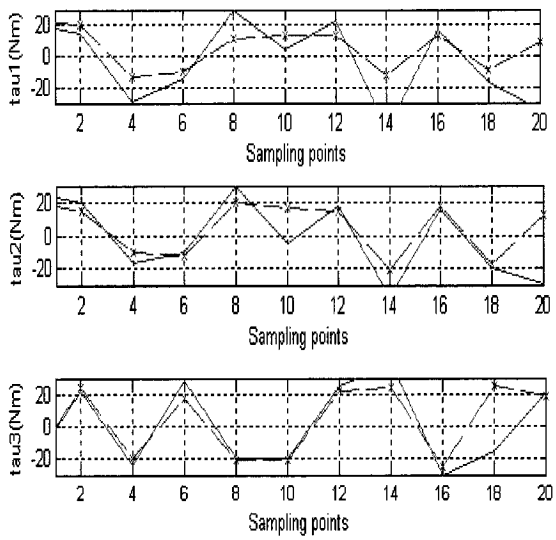


Fig. 6. 7. Corresponding joint torques
(__Cycloidal profile trajectory, ._._ALD trajectory)

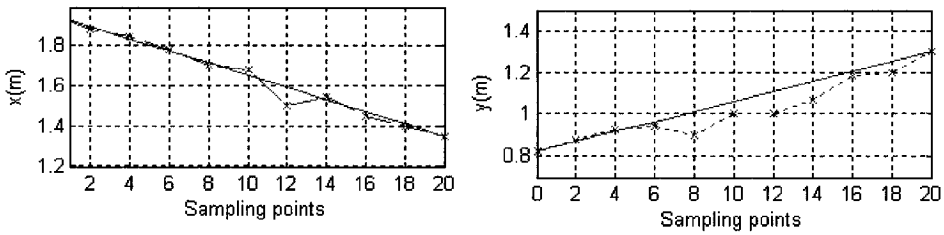


Fig. 6. 8. Instantaneous variations of (x, y) coordinate while satisfying the passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m)
(__Cycloidal profile trajectory, ._._ALD trajectory)

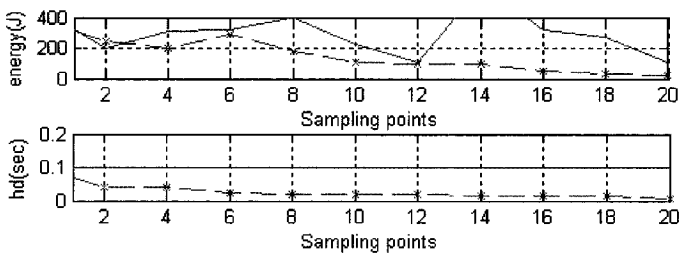


Fig. 6. 9. Consumed energy and traveling time for the trajectory with imposed passages
(__Cycloidal profile trajectory, ._._ALD trajectory)

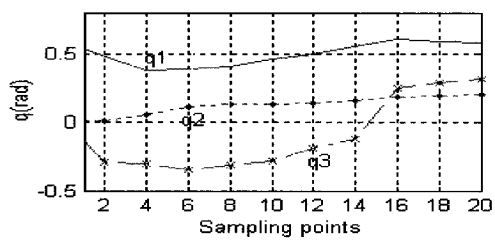


Fig. 6. 10. Instantaneous variations of joint angles q_j, $j = 1,2,3$ for the circle trajectory
(__Cycloidal profile trajectory, ._._ALD trajectory)

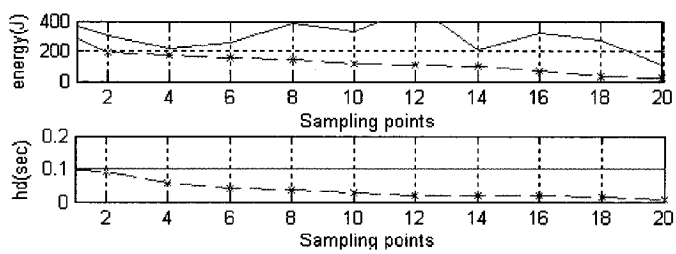
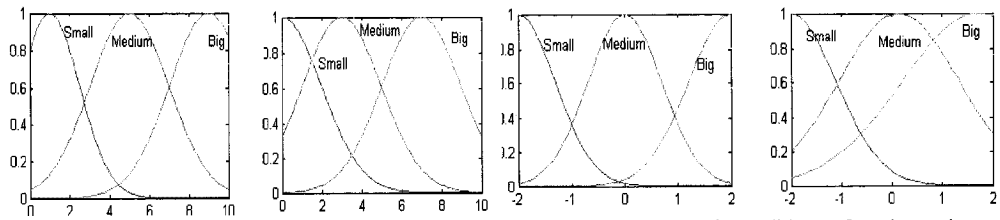
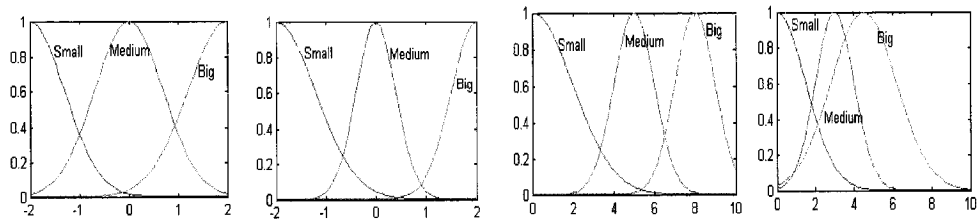


Fig. 6. 11. Instantanuous energy and sampling time variations (Circle trajectory)
(__Cycloidal profile trajectory, ._._ALD trajectory)



(a) x before learning (b) x after learning (a) y before learning (b) y after learning



(a) $\dot{x}$ before learning (b) $\dot{x}$ after learning (a) $\dot{y}$ before learning (b) $\dot{y}$ after learning

Fig. 6. 12. Membership functions defining x, $\dot{x}$, y and $\dot{y}$

Table 6. 3. NEFUMOP Performance

2000 Data entries	RMS Error (after 20 epochs)
Training data (90%)	0.004571
Testing data (10%)	0.0002392

TABLE 6. 4 NEFUMOP Performance (with disturbed testing data)

2000 Data entries	RMS Error (after 40 epochs)
Training data (90%)	0.00591
Testing data (10%)	0.005013857
Validation data (10%)	0.006213857

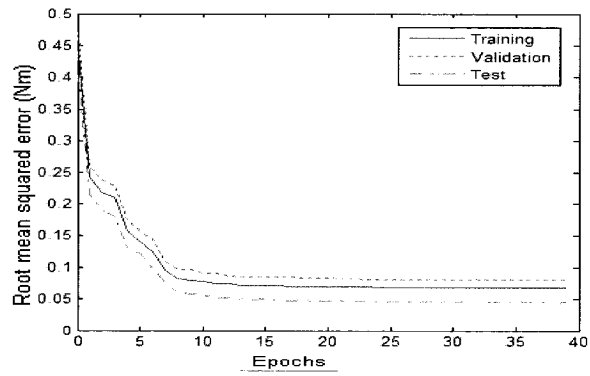


Fig. 6. 13 NeFUMOP performance (RMS Error vs. Epochs)

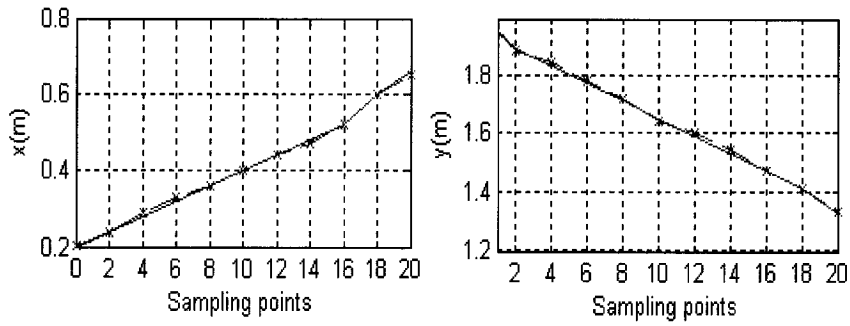


Fig. 6. 14. Instantaneous variations of (x, y) coordinate variations the EE, (__desired trajectory, ._._computed trajectory)

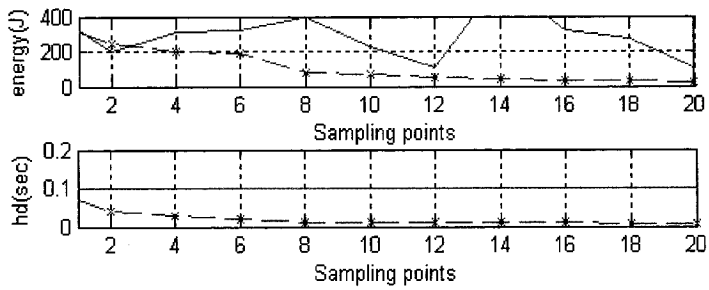


Fig. 6. 15. Energy and sampling period h variations for the test trajectory (__Cycloidal profile trajectory, ._._ALD trajectory)

6. 6. Conclusions

The main contribution of this paper is a systematic design approach to multi-objective online motion planning. This system is built upon the outcomes of an offline planning based on full kinematic and dynamic models of a redundant serial planar manipulator. The offline planning involves also joint angles, rates, accelerations, and jerks, as well as workspace and actuator torques limitations, and optimizes travelling time and energy necessary to achieve a trajectory, while solving redundancy and avoiding singularities. According to simulation results, the proposed offline planning system is effective and robust in solving the non-convex and non-linear constrained motion-planning problem. The multi-objective planning produced singularity-free smooth trajectories, making them very suitable to design optimal trajectories for online planning.

The second contribution is an objective neuro-fuzzy model built to achieve online multi-objective motion planning using the previous offline generated data. A subtractive clustering method was first used to initialize the structure parameters of an

objective neuro-fuzzy model. Then the neuro-fuzzy system is trained to represent and capture the robot constrained multi-objective planning behavior. The main advantages of this approach is that once the neuro-fuzzy model is learnt and optimized, then a multi-objective trajectory capturing the robot dynamic behavior might be obtained for a given starting and ending task points, without the need to re-do all the calculations that would be necessary with a conventional optimal control.

In our implementation, the cluster center and other subtractive clustering parameters are chosen on a trial-error basis. This might be improved by selecting these parameters based on an intelligent search like genetic or simulated annealing algorithms. Another ongoing work consists of including static obstacle avoidance and other optimization criteria within the proposed neuro-fuzzy multi-objective trajectory planning system. Another perspective consists of doing more tests and validation experiments are also planed on a physical robot.

Acknowledgment

The authors would like to gratefully thank the Natural Science and Engineering Research Council of Canada (NSERC) for supporting this work under grants ES D3-317622, RGPIN-203618, RGPIN-105518, RGPIN-18898 and STPGP-269579.

6. 7. References

[1] M. Vukobratovic, D. Stokic: "Is Dynamic Control Needed in Robotics System? and If So to What Extent? ", *Int'l Jou. of Robotics Research* 2(2): 18-35, 1983.

[2] A. Khoukhi: "An Optimal Time-Energy Control Design for a Prototype Educational Robot", *Robotica* 20 (6), p 661-671, 2002.

[3] A. Khoukhi, L. Baron, M. Balazinski: "A Decoupled Approach to Optimal Time Energy Trajectory Planning of Parallel Kinematic Machines", *Romansy'06, 16th CISM-IFFToMM Symposium on Robot Design, Dynamics and Control,* Warsaw, Poland, June 20-24, 2006.

[4] J. J. Craig: "*Introduction to Robotics Mechanics and Control*", Prentice Hall, , NJ, 2005.

[5] Y. Nakamura, Advanced Robotics: "*Redundancy and Optimization*", Addison-Wesley, 1991

[6] J. Kieffer, A. J.Cahill, and M. R. James: "Robust and Accurate Time-Optimal Path-Tracking Control for Robot Manipulators", *IEEE Trans. Robot. Aut.* 13, p 880-890, Dec.1997.

[7] J. J. E. Slotine, and W. Li: "On the Adaptive Control of Robot Manipulators," *Int. J. Rob. Research,* 6 (3), 1987.

[8] Z. Yu, H. Chen, P. Y. Woo: "Advanced gain scheduled $H_{infinity}$ controller for robotic manipulators, *Robotica*, 20 (5), p 537-544, September/October, 2002.

[9] M. A. C. Gill, A. Y. Zomaya: "*Obstacle Avoidance in Multi-Robot Systems: Experiments in Parallel Genetic Algorithms*", World Sc. Series in Robotics and Intelligent Systems , 20, 1998.

[10] L. Baron: "A Genetic Algorithm for Computing the Real Solutions of the Inverse Kinematics of Serial Manipulators", *Tenth World Congress on the Theory of Machines and Mechanisms,* Vol. 3, pp. 1134-1139, Oulu, Finland, 20-24 June 1999.

[11] A. Khoukhi, A. Ghoul: "On the Maximum Dynamic Stress Search Problem for Robot Manipulators", *Robotica*, 22 (5), pp: 513-522, 2004.

[12] Y. L. Sun, M. J. Er: "Hybrid Fuzzy Control of Robotics Systems", *IEEE Trans. On Fuzzy Systems* 12 (6), pp: 755-765, December 2004

[13] W. S. Lin, C. H. Tsai, and C. H. Wang: "Robust Neuro-Fuzzy Model-Following Control of Robot Manipulators", *Proceedings of the IEEE Int'l Conference on Control Applications,* Trieste, Italy, September, 1998.

[14] J. K. Son, H. Y. Sik, P. C. Kug: "Neuro-Fuzzy Control of a Robot Manipulator for a Trajectory Design", *IEEE Int'l Workshop on Robot and Human Communication*, 1997.

[15] A. Khoukhi, K. Benfreha: "Application of Multi-Layer Neuro-Fuzzy Networks to Collision Detection in a Simulated Manufacturing", *Transactions of the Canadian Society of Mechanical Engineering*, 29 (3-4), pp: 431-444, January 2004.

[16] A. Behnam, A. Nader, N. Detlef (Eds.): "*Intelligent Systems and Soft Computing, Prospects, Tools and Applications*", XVII, 359 p. Springer Verlag, NY, 2000.

[17] J. S. R. Jang, C. T. Sun, E. Mizutani: "*Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*", Prentice-Hall Engineering, 614 pages, 1996.

[18] L. X. Wang: "*A course in Fuzzy Systems and Control*", Prentice Hall, 1997.

[19] S. Achiche, L. Baron, and M. Balazinski, : "*Real/Binary-Like Coded Versus Binary Coded Genetic Algorithms to Automatically Generate Fuzzy Knowledge Bases: A comparative study*, Engineering Applications of Artificial Intelligence, 17 (4), pp. 313-325, June 2004.

[20] M. Al-Khatib and J. J. Saade: "An Efficient Data-Driven Fuzzy Approach to the Motion Planning Problem of a Mobile Robot", *Fuzzy Sets and Systems,* 134 (1), 16, p 65-82, 2003

[21] A. Khoukhi, K. Demirli, L. Baron, M. Balazinski: "Hierarchical Neuro-Fuzzy Near Optimal Time Trajectory Planning for Redundant Manipulators", Accepted, *Engineering Applications of Artificial Intelligence*, 2007

[22] A. Khoukhi, L. Baron, M. Balazinski: "Multi-Objective Trajectory Planning for Redundant Manipulators by Augmented Lagarangian", Accepted, *CCToMM, Sym. on Mechanics, Machines and Mechatronics,* May 31 to June 1 2007, Canadian Space Agency, Montreal, Canada.

[23] C. R. Carignan: "Trajectory Optimization for Kinematically Redundant Arms", *J. of Robotic Systems*, 8(2) p: 221 - 248, 1991.

[24] D. P. Bertsekas: "*Non linear programming*", Athena Scientific, 1995.

[25] J.B.Hiriart-Urruty and C. Lemaréchal: "*Convex analysis and minimization algorithms*", Springer Verlag, Berlin, New York, 1993.

[26] T. Rockafellar: "Lagrange multipliers and optimality", *SIAM Review*, 35(2), p 183-238, 1993.

[27] A. Isidori: "*Non linear control systems*", Springer; 3rd Edition, London, UK, 1995.

[28] A. Khoukhi, K. Demirli, L. Baron, M. Balazinski: "Fuzzy-Neuro Optimal Time-Energy Control of a Three Degrees of Freedom Manipulators", *NAFIPS'06 (North American Fuzzy Information Processing Society)*, Concordia University, Montreal, QC, Canada, June 3-6, 2006.

[29] K. Demirli, S. X. Cheng, P. Muthukumaran, "Subtractive Clustering Based Modeling of Job Sequencing with Parametric Search", *Fuzzy Sets and Systems;* 137, p 235-270, 2003.

[30] S. L. Chiu: "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent and Fuzzy Systems*; 2, 267-278, 1994.

CHAPITRE 7

DISCUSSION GÉNÉRALE

Dans cette thèse, nous avons étudié le problème de planification multi-objective de trajectoire des manipulateurs robotiques.

Pour la planification hors-ligne, l'objectif était de développer une méthode capable de répondre au besoin d'un modèle complet du système pour une meilleure précision dans l'exécution de la tâche. Cette méthode doit donc être capable de tenir compte des fortes non linéarités des transformations cinématiques et couplages dynamiques de la structure robotique considérée. En premier lieu, cet objectif est traité en formulant le problème dans le cadre du calcul variationnel. Le choix de l'utilisation de la technique du Lagrangien augmenté se basait sur le fait que le problème à résoudre est un problème de grande taille contenant plusieurs objectifs contradictoires (tel que minimisation du temps et de l'énergie nécessaires à l'exécution de la tâche). Aussi, le Lagrangien augmenté (ALD) permet une meilleure prise en compte de contraintes non convexes (telle que les contraintes relatives aux équations cinématiques et dynamiques du système), comparativement aux autres techniques proposées dans la littérature. Une implémentation directe de cette méthode d'optimisation nous a conduit à des calculs très complexes, non solutionables même par utilisation d'un logiciel de calcul symbolique. Ce problème apparaît notamment

lors du calcul des états adjoints. Ceci nous a emmené à implémenter l'algorithme sur une dynamique découplée du robot, transférant ainsi la non linéarité à la fonction objective et rendant facile le calcul des états adjoints. D'autre part, les deux types de systèmes robotiques sur lesquels cette approche a été implémentée sont des systèmes fortement non linéaires, non convexes sujettes à plusieurs contraintes sur les variables d'état et de commande; à savoir les manipulateurs sériels planaires redondants et les machines cinématiques parallèles.

Les performances de l'approche ALD ont été comparées à celles des approches utilisées jusqu'à date, notamment les méthodes de planification de trajectoires basées uniquement sur le modèle cinématique du robot ou les méthodes de planification monocritères. L'ALD est plus performant, à la fois pour les manipulateurs parallèles et les manipulateurs sériels redondants. Les comparaisons ont montré que ces performances sont encore plus évidentes lorsqu'on considère des contraintes supplémentaires issues d'exigences spécifiques de la tâche, telle que le passage par des poses imposées lors du mouvement du robot.

Ceci a été bien illustré aux chapitres 4 et 5. En effet, dans ces deux chapitres, on a montré que ALD permettait d'obtenir des trajectoires plus régulières, minimisant une fonctionnelle multicritères relatant le temps et l'énergie nécessaires à l'exécution de la tâche, ainsi qu'une fonction d'évitement des configurations singulières. En outre, comme les systèmes robotiques sont des systèmes mécaniques dans lesquels les paramètres dynamiques, tel que par exemple les paramètres inertiels, sont susceptibles d'une grande variation lors de l'exécution d'une tâche, une étude de

sensibilité de l'ALD a été effectuée pour évaluer la robustesse de cette technique en variant la masse de l'effecteur du robot, tout en maintenant les même conditions et paramètres de simulation. Les résultats ont montré une très bonne robustesse de l'ALD, en atteignant l'état final avec une précision de 10^{-4}(m).

Néanmoins, le temps de calcul est assez grand, de l'ordre de 10 minutes pour la trajectoire et la précision mentionnée ci-dessus. Cependant, ceci demeure acceptable, car l'objectif à ce niveau la planification hors-ligne.

Comparées aux autres stratégies de planification de trajectoire, par exemple, les méthodes de commande en temps minimum, cette technique est fortement souhaitable, car elle permet de réaliser des trajectoires régulières, répondants à plusieurs objectifs de production, et satisfaisant diverses contraintes. Elle est particulièrement souhaitable comme trajectoire de référence pour la synthèse d'un contrôleur par retour d'état de type, PD ou PID.

Une autre utilisation possible des résultats de POT ci-dessus, est la génération d'une base de données qui va servir à l'apprentissage d'un réseau d'inférence neuro-flou, dirigé par les données (*Data-Driven Neuro-Fuzzy Inference Systems*), pour la POT en-ligne des manipulateurs robotiques. Ceci constitue la deuxième partie de la thèse.

En effet, comme nous l'avons mentionné dans notre étude bibliographique, le problème de POT en-ligne, est d'une grande complexité, et les approches conventionnelles de commande optimale en boucles ouvertes ou en boucles fermées

ne sont pas à même de le résoudre, à cause notamment des fortes non-linéarités des modèles cinématiques et dynamiques, ainsi que des contraintes associées.

Pour surmonter ces difficultés, nous avons proposé au chapitre 6 d'utiliser les systèmes d'inférence neuro-flous dirigés par les données.

En premier lieu, cet objectif est résolu partiellement, en restreignant l'espace des configurations du manipulateur à une pose coude haut, ou coude bas. D'abord, un réseau de neurones (NetIK) a été implémenté pour apprendre et résoudre le problème cinématique inverse du robot dans l'espace de travail restreint. Bien que des performances acceptables ont été obtenue, le temps de calcul pour réaliser l'apprentissage était assez grand, comparé à la deuxième approche implémentant un réseau neuro-flou (NeFIK). Une fois le modèle cinématique inverse est résolu, la deuxième étape est celle de génération d'un ensemble suffisamment grand de trajectoires en temps minimum. Ces trajectoires permettent le déplacement de l'effecteur du robot de différents points de départ vers différents points d'arrivée, balayant le plus possible de l'espace de travail (configuration coude haut ou coude bas). L'ensemble des trajectoires ainsi générées est partitionné par la méthode du '*Subtractive Clutsering*' pour initialiser la structure du réseau neuro-flou NeFOTC, qu'on va ensuite optimiser pour apprendre le comportement des trajectoires en temps minimum du robot. Les performances du réseau développé sont bien intéressantes, considérant l'erreur des moindre carrées d'apprentissage du réseau, qui est de l'ordre de 10^{-3} (m) sur les données tests.

Cependant, deux limitations sérieuses sont à mentionner pour cette approche. D'abord, elle est limitée à une configuration coude haut ou coude bas du robot. L'autre limitation est relative à la fonctionnelle coût, qui considère uniquement la minimisation du temps d'exécution de la tâche. Pour relaxer ces limitations, au chapitre 7, nous avons résolu d'abord le problème de génération des trajectoires hors-ligne, optimisant une fonctionnelle multi objective comprenant le temps, l'énergie ainsi qu'une fonction d'évitement des configurations singulières. La technique ALD développée au chapitres 4 et 5 est alors utilisée pour la construction de la base de données d'apprentissage du réseau neuro-flou (NeFUMOP). La procédure d'implémentation est la même que celle du chapitre 6. La redondance cinématique est prise en compte dans la partie hors ligne de la présente approche. De très bonnes performances du réseau NeFUMOP ont été réalisées avec une erreur d'apprentissage du même ordre (10^{-2} m) de grandeur que le réseau NeFOTC du chapitre 6; d'abord sur les données test, qui constituent 10% de l'ensemble de la base de données d'apprentissage. Le deuxième test sur une trajectoire que nous avons réalisé d'abord par ALD, et puis introduite au réseau pour tester ces capacités de généralisation. Les résultats confirment bien ces performances soit au niveau de la précision atteinte, soit en niveau de la minimisation de la fonction objective.

CHAPITRE 8

RECOMMENDATIONS

Dans ce chapitre, on va proposer quelques idées et recommandations, pour la poursuite de la recherche initiée dans cette thèse.

Bien que les deux approches développées pour les planification multi-objectives hors-ligne et en-ligne de trajectoire de manipulateurs robotiques ont donné de bons résultats, on pense qu'il y a moyen d'amélioration et d'extension de ces approches sur deux volets. Le premier est sur le plan algorithmique. Le deuxième est sur le plan robotique.

Sur le plan algorithmique, dans l'implémentation du Lagrangien augmenté, nous avons utilisé un algorithme de gradient pour la mise à jour des périodes d'échantillonnage. Une hybridation du Lagrangien augmenté avec un algorithme de programmation en nombre entier [87] serait fort utile pour déterminer un meilleur mécanisme d'adaptation du pas d'échantillonnage.

La deuxième extension sur le plan algorithmique est relative aux modèles et hypothèses du problème de POT hors-ligne considérés. En effet, l'hypothèse d'un modèle déterministe de la dynamique et de la cinématique du système robotique, incluant son environnement est quelque peu restrictive pour en être réellement représentative du système réel. L'introduction des aléas dans les modèles du robot et

de son environnement nous paraît bien souhaitable. Néanmoins, les couplages dynamiques et les fortes non linéarités des transformations cinématiques et des contraintes associées font en sorte que les outils disponibles présentement dans la littérature ne sont à même de traiter ce problème. Par exemple l'application du filtre de Kalman étendue [88] ou de la commande de type LQG (Linéaire Quadratique Gaussien) introduit des linéarisations et approximations sévères. Ceci résulte en une grande perte de l'information et de représentativité par rapport au problème initial.

Par rapport à la planification en-ligne de trajectoire, nous suggérons deux extensions possibles. D'abord, des données mesurées par des capteurs incorporés sur le robot à partir de trajectoires expérimentales peuvent être ajoutées pour enrichir la base de données actuelle. Ce qui permettra d'améliorer d'avantage les performances de la méthode proposée. La deuxième extension est par rapport à la procédure de groupement des données et d'initialisation des paramètres du système d'inférence flou. Les paramètres de simulation de cette procédure ont été choisis manuellement en effectuant plusieurs essais. Un ajustement de ces paramètres au cours de la procédure de groupement par exemple en utilisant un algorithme génétique ou un algorithme de recuit simulé pourrait améliorer significativement les performances de cette procédure.

Sur le plan application robotique; plusieurs extensions du présent travail sont à souligner. D'abord il serait fort intéressant d'implémenter ces deux approches de planification sur des robots physiques et puis les intégrer aux systèmes de contrôle de ces robots.

Une autre perspective souhaitable serai d'inclure une fonction d'évitement d'obstacles tridimensionnels et une procédure d'évitement des interférences entre les segments du robot dans les système de POT hors-ligne et ligne développés.

CONCLUSIONS

En somme, les contributions principales de la présente thèse peuvent être énumérées comme suit :

- L'utilisation du calcul variationnel pour formuler le problème de planification de trajectoire des manipulateurs robotiques est un choix adéquat.
- L'aspect multi-objectif de la présente approche a permis une personnalisation de cet outil de planification en fonction des spécifications de la tâche à planifier. Ceci est rendu possible grâce au choix du cœfficient de pondération attribué à chaque critère.
- La technique du Lagrangien augmenté en temps discret implémentée sur une dynamique découplée du système a permis de résoudre le programme non linéaire, non convexe sous contraintes résultant avec un meilleur conditionnement numérique.
- La formulation du problème de POT comme un problème de commande optimale où les périodes d'échantillonnage et couples moteurs sont les variables indépendantes de commande, constitue une nouvelle approche pour la commande en temps miniumum et commande multi-critère des manipulateurs robotiques. Cette approche permet aussi la prise en compte de toutes contraintes, par inclusion de son expression dans le Lagrangien augmenté.

- Deux implémentations de cette approche ont été considérées sur deux cas de structures robotiques; les manipulateurs parallèles et les manipulateurs sériels redondants. Ces machines sont particulièrement complexes, à cause notamment des problèmes des singularités pour la première et de la redondance cinématique pour la deuxième Les résultats de simulation ont montré l'efficacité et la robustesse de cette approche, considérant l'optimalité du critère et la satisfaction des contraintes, comparée aux autres approches, tel que les méthodes de pénalité, les méthodes de commande en temps minimum ou encore les méthodes basées uniquement sur la cinématique du robot.

La deuxième partie de la thèse a porté sur la planification en-ligne de trajectoires des manipulateurs robotiques.

- Nous avons montré que l'approche préconisée des systèmes adaptatifs d'inférence neuro-flou dirigés par les données était une bonne alternative aux approches conventionnelles, pourvu qu'un ensemble important couvrant suffisamment l'espace de travail du robot soit fournie.
- L'utilisation de l'algorithme de groupement de données '*Subtractive Clustering*' permet de partitionner l'ensemble des trajectoires générées et initialiser au même temps les paramètres d'un réseau adaptatif d'inférence neuro-flou. Après apprentissage et optimisation, ce réseau est utilisé en phase de généralisation pour une implémentation temps réel. Cette approche hybride s'est avérée bien efficace, notamment en termes de réduction du nombre de règles floues, compte tenu du

nombre de variables d'entrées/sorties du système et de la dimension de la base de données d'apprentissage.

Deux méthodes de conception de réseaux neuro-flous ont été élaborées :

- La première conception consiste en deux contrôleurs neuro-flous hiérarchiques pour la planification temps minimum de trajectoire des robots sériels redondants. Le premier implémente un réseau neuronal pour résoudre l'inverse cinématique. Le deuxième le réalise par un réseau neuro-flou. Nous avons remarqué que pour les mêmes performances, le temps machine que nécessitait le réseau neuronal était environ 5 fois plus grand que pour le réseau neuro-flou.
- La deuxième conception implémente un réseau neuro-flou pour la planification en-ligne multi-objective de trajectoire. Lors d'une première étape, le programme développé dans la première partie de la thèse est utilisé pour générer hors-ligne suffisamment de trajectoires multi-objective, couvrant l'espace de travail du manipulateur et satisfaisant plusieurs contraintes, relatives aux performances des interactions du trio robot, tâche, et espace de travail. La redondance cinématique est résolue, en évitant les limites articulaires. Les critères d'optimisation portent sur le temps et l'énergie nécessaires à la réalisation de tâche. Sur les trajectoires ainsi générées, le réseau neuro-flou entraîné et optimisé, a montré de bons résultats de généralisation à la planification en-ligne multi-objective de trajectoires.

BIBLIOGRAPHIE

[1] D. Tesar, "Where is the Field of Robotics Going?", ICRAM '99, Istanbul Turkey (1999), p 24-26.

[2] G.S. Sukhatme, M.J. Mataric: "Robots: Intelligence, Versatility, Adaptativity", Communications of the ACM, vol. 45, no. 3, Marsh 2002, p 30-63.

[3] M. Shoubik, , "Simulation of the Industry and the Firm", American Economic Review L, No. 5, p 908-919, 1960.

[4] C. Ahrikencheikh and A. Seireg, "*Optimized-Motion Planning*", John Wiley &Sons, 1994.

[5] A. Khoukhi, "An Optimal Time-energy Control Design for a Prototype Educational Robot". *Robotica* 20 p 661-671, 2002.

[6] A. Khoukhi, L. Baron, M. Balazinski, "Programmation Dynamique Time-Énergie Minimum des Robots Parallèles par Lagrangien Augmenté", *CCToMM SM3, May 26-27,* Canadian Space Agency, Saint Hubert QC, Canada p 147-148, 2005

[7] A. Khoukhi, L. Baron, M. Balazinski, "A Decoupled Approach to Time-Energy Trajectory Planning of Parallel Kinematic Machines", *CISM Courses and Lectures No. 487, Romansy 16 Robot Design, Dynamics and Control, Proceedings of 16th CISM-IFFToMM Symp.* Warsaw Poland, June 20-24, p 179-186, SpringerWien N. Y, 2006

[8] A. Khoukhi, Y. Hamam, "AFMTC Design for Robot Dynamic Control » *First*

International Conference on Optimal Design and Analysis of Experiments, Neuchâtel, Switzerland, 8-12 July 1989.

[9] A. Khoukhi, A. Ghoul, "On the Maximum Dynamic Stress Search Problem for Robot Manipulators", *Robotica*, 22, 5, p 513-522, 2004.

[10] U. Nehmzow, "*Mobile Robotics: A Practical Introduction*", Springer-Verlag, London, 2000.

[11] J.C. Latombe, "*Robot Motion Planning*", Kluwer Academic Publishers, Norwell, Mass. 1991.

[12] J.H. Reif, "Complexity of the Mover's Problem and Generalizations", *Proc. of the 20th IEEE Symposium on Foundations of Computer Science, IEEE,* New York, p 421-427, 1979

[13] J. Canny, "*The Complexity of Robot Motion Planning*", MIT Press, Cambridge, Mass. 1988.

[14] D. Constantinescu and E. A. Croft, "Smooth and Time-Optimal Trajectory Planning for Industrial Robots Along Specified Paths" *J. Robot. Syst.,* vol. 17, p 223-249, 2000.

[15] P.Pledel and Y. Bestaoui, "Actuator Constraints in Point to Point Motion Planning of Manipulators", *Proc. IEEE Conf. Decision and Control,* p 1009-1010, 1995.

[16] O. Dahl, "Path-Constrained Robot Control with Limited Torques-Experimental Evaluation", *IEEE Trans. Robot. Automat.*, vol. 10, p 658-669, Oct.1994.

[17] J. Kieffer, A. J.Cahill, and M. R. James, "Robust and Accurate Time-Optimal

Path-Tracking Control for Robot Manipulators", *IEEE Trans. Robot. Automat.* vol. 13, p 880-890, Dec.1997.

[18] K. J. Waldron & K. H. Hunt, "Series-Parallel Dualities in Actively Coordinated Mechanisms", *International Journal of Robotics Research,* v 10, n 5, p 473-480, Oct. 1991.

[19] D. A. Kumar, C. I. Ming, Y. S. Huat and Y. Guilin, "Workspace generation and planning singularity-free path for parallel manipulators", *Mechanism and Machine Theory,* v 40, n 7, p 776-805, June, 2005

[20] S. Bhattacharya, H. Hatwal, A. Ghosh, "Comparison of an Exact and Approximate Method of Singularity Avoidance in Platform Type Parallel Manipulators". *Mech. Mach. Theory* 33 (7) p 965-974, 1998.

[21] B. Dasgupta, T.S. Mruthyunjaya, "The Stewart Platform Manipulator, A Review", *Mechanism and Machine* Vol. 35, No. 1, p 15-40, 2000.

[22] J. K. Son, H. Y. Sik, P. C. Kug, "Neuro-Fuzzy Control of a Robot Manipulator for a Trajectory Design", *IEEE Int'l Workshop on Robot and Human Communication,* 1997.

[23] H. Shao, K. Nonami, T. Wojtara, R. Yuasa, S. Amano, D. Waterman, "Neuro-Fuzzy Position Control of Demining Tele-Operation System Based on RNN Modeling", *Rob. and Computer Integrated Manuf.*, 22, p 25-32, 2006.

[24] Y. L. Sun, M. J. Er, "Hybrid Fuzzy Control of Robotics Systems", *IEEE Trans. on Fuzzy Sys.* vol. 12, No. 6, p 755-765, 2004.

[25] W. S. Lin, C. H. Tsai, and C. H. Wang, "Robust Neuro-Fuzzy Model-Following Control of Robot Manipulators", *Proc. of the IEEE Int'l Conference on Control Applications*, Trieste, Italy, September, 1998.

[26] Available: www.3ds.com/products-solutions/brands/CATIA

[27] Available: http://www.igrid2005.org/

[28] Available: http://www.robotmaster.ca /

[29] Available: www.its-robotics.be/RobCAD

[30] Available: http://www.fanucrobotics.com/24,1374,Simulation+Software.html

[31] M. E. Kahn and B. Roth, "The Near-Minimum-Time Control of Open-Loop Articulated Kinematic Chains", *J. Dyn. Syst., Meas., Contr.,* vol. 93, p 164-172, Sept.1971.

[32] B. Cao, G. I. Dodds, and G. W. Irwin, "Time-Optimal and Smooth Constrained Path Planning for Robot Manipulators", *Proc. IEEE Int. Conf. Robotics and Automation,* p 1853-1858, 1994.

[33] Z. Shiller and W. Lu, "Computation of Path Constrained Time Optimal Motion with Dynamic Singularities", *Trans. ASME,* vol. 114, p 34-41, 1992.

[34] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-Optimal Control of Robotic Manipulators Along Specified Paths" *Int. J. Robot. Res.,* vol. 4, p 3-17, Fall.1985.

[35] K. G. Shin and N. D. McKay, "Minimum-Time Control of Robotic Manipulator with Geometric Path Constraints", *IEEE Trans. Robot. Automat. Contr.,* vol. AC-30, p 531-541, June 1985.

[36] G. Sahar and J. M. Hollerbach, "Planning of Minimum-Time Trajectories for Robot Arms", *Int. J. Robot. Res.,* vol. 5, p 90-100, Fall 1986.

[37] H. Jurgen, Mc. John, "Determination of Minimum-Time Maneuvers for A Robotic Manipulator Using Numerical Optimization Methods", *Mechanics of Structures and Machines*, v 27, n 2, p 185-201, May, 1999

[38] H. P. Geering, L.Guzzella, S. A. R. Hepner, and C. H. Onder, "Time-Optimal Motions of Robots in Assembly Tasks", *IEEE Trans. Robot. Automat., Contr.,* vol. AC-31, p 512-518, June1986.

[39] Y. Chen and A. A. Desrochers, "Structure of Minimum-Time Control Law for Robotic Manipulators with Constrained Paths", *Proc. IEEE Int. Conf. Robotics and Automation,* p 971-976, 1989.

[40] T. Rockafellar, "Lagrange Multipliers and Optimality", *SIAM Review* 35, p 183-238, 1993

[41] D. P. Bertsekas, "*Non Linear Programming*" , Athena Scientific, 1995.

[42] J.P. Merlet, "*Parallel Robots*", Kluwer Academic Publisher: Dordrecht, the Netherlands, 2000

[43] L. Baron, J. Angeles, "The Kinematic Decoupling of Parallel Manipulators Under Joints-Sensor Redundancy", *IEEE Transactions on Robotics and Automation,* vol. 16, no 1 , p 12-19, Dec. 2000.

[44] L. Baron, J. Angeles, "The Direct Kinematics of Parallel Manipulators Under Joints-Sensor Data", *IEEE Trans. on Robotics and Automation,* vol. 16, no 6, p 644-651, 2000.

[45] C. Gosselin, "Parallel Computational Algorithms for the Kinematics and Dynamics of Plannar and Spatial Parallel Manipulators", *ASME, J. of Dynamic Systems, Measurements and Control,* vol. 118 no. 1, p 22-28, 1996

[46] K. Harib, K. Srinivasan, "Kinematic and Dynamic Analysis of Stewart Platform-Based Machine Tool Structures", *Robotica*, vol. 21, p 541-551, 2003

[47] I.T. Pietsch, M. Krefft, O.T. Becker, C.C. Bier, J. Hesselbach, How to reach the dynamic limits of parallel robots? An autonomous control approach, *IEEE Trans on Aut. Sc. and Eng.,* Vol. 2, No 4, p 369-380, 2005.

[48] S. Sen, B. Dasgupta, A. K. Mallik, "A Variational Approach for Singularity-Free Path-Planning of Parallel Manipulators", *Mec. and Machine Theory* 38: p 1165-1183, 2003.

[49] C. Innocenti and V. Parenti-Castelli, "Singularity-Free Evolution From One Configuration to Another in Serial and Fully-Parallel Manipulators", *Journal of Mechanical Design, Transactions of the ASME* 120 (1), p 73-79, 1998.

[50] J. Angeles, "*Fundamentals of Robotic Mechanical Systems, Theory, Methods and Algorithms*", 2nd Ed. Springer Verlag. 2000

[51] M Hay, J. A. Snyman, methodologies for the optimal design of parallel manipulators, *Inl J. for Numerical Methods in Engineering*, Vol. 59, No. 1, p 131-152, 2004.

[52] M Hay, J. A. Snyman, A multi-level optimization methodology for determining the dexterous workspaces of planar parallel manipulator, *Structural and Multidisciplinary Optimization*, Vol. 30, No. 6, p 422-427, 2005.

[53] P. R. Ouyang, W. J. Zhang, "Comparsion of PD-based controllers for robotic manipulators, *Proceedings of the ASME Design Engineering Technical Conference and Computers and Information in Engineering Conference 2004: Volume 2: 28th Biennial Mechanisms and Robotics Conference*, p 23-31, 2004.

[54] J. J. E. Slotine, and W. Li "On The Adaptive Control of Robot Manipulators," *Int. J. Robotics Research,* v 6, 3, 1987.

[55] D. Nganga Kuya, M. Saad, L.Lamarche, "Commande adaptative hybride SFB pour les systèmes robotiques basée sur la passivité", *Transactions de la Société Canadienne de génie mécanique*, v 26, n 2, p 163-177, 2002.

[56] M. A. C. Gill, A. Y. Zomaya, "*Obstacle Avoidance in Multi-Robot Systems: Experiments in Parallel Genetic Algorithms*", World Scientific Series in Robotics and Intelligent Systems , v. 20, 1998

[57] L. Baron, "A Genetic Algorithm for Computing the Real Solutions of the Inverse Kinematics of Serial Manipulators", *3rd World Congress on the Theory of Machines and Mechanisms,* vol. 3, p 1134-1139, Oulu, Finlande, 20-24 June 1999.

[58] S. Achiche, L. Baron, M. Balazinski, M. Benaoudia, "Online prediction of pulp brightness using fuzzy logic models", *Engineering Applications of Artificial Intelligence*, v 20, n 1, p 25-36, February, 2007.

[59] A. Khoukhi, K. Benfreha: "Application des réseaux neuro-flous multi-couches à la détection de collisions outils-pièces en tournage simulé", *Trans. de la Société Canadienne de Génie Mécanique, SCGM,* v 28.(3-4), p 431-443 2004

[60] C. Andrzej: "Applications of knowledge technologies to sound and vision engineering", *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v 4062 LNAI, *Rough Sets and Knowledge Technology - First International Conference, RSKT 2006, Proceedings*, p 57, 2006.

[61] Y. Hua. J. M. Zurada, W. Karwowski, W. S. Marras: "A hybrid neuro-fuzzy approach for spinal force evaluation in manual materials handling tasks, *Lecture Notes in Computer Science*, v 3612, n PART III, *Advances in Natural Computation: First International Conference, ICNC 2005. Proceedings*, 2005, p 1216-1225

[62] Y. Nakamura: "*Advanced Robotics: Redundancy and Optimization*", Addison-Wesley, 1991

[63] M. J. D. Powell, "A Method for Nonlinear Constraints in Minimization Problems", in *Optimization*, R. Fletcher, Ed., Academic Press London, p 283-298, 1969.

[64] M. R. Hestens, "Multiplier and Gradient Methods", *J. Optimization Theory and Applications* 4, 303-320, 1969.

[65] N. Gould, D. Orban, A. Sartenear, and P. Toint, "Super Linear Convergence of Primal-Dual interior Points Algorithms for Non Linear Programming", in *SIAM J. on Opt, Math Prog.* B, v 11, 4, p 974-1002, 2001

[66] J.B. Hiriart-Urruty and C. Lemaréchal: "*Convex Analysis and Minimization Algorithms*", Springer, N.Y, 1993.

[67] A. Khoukhi, L. Baron, M. Balazinski: "Constrained Multi-Objective Trajectory Planning of Parallel Kinematic Machines", Accepté, *Robotics and Computer Integrated Manufacturing*, 2007

[68] A. Isidori, Non Linear Control Systems, Springer; 3rd Edition, , London, UK, 1995

[69] X. J. Liu, Q. M. Wang, J. Wang, "Kinematics, Dynamics and Dimentional Synthesis of a Novel 2- DoF Translational Manipulator", *Journal of Intelligent and Robotic Systems,* 41, p 205-224, 2004.

[70] Available: www.mathworks.com/products/matlab/

[71] F. Mnif, E. K. Boukas, M. Saad, Robust control for constrained robot manipulators, *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, v 121, n 1, p 129-133, 1999.

[72] S. P. Chan, "Decoupling control of robot manipulator with perturbation feedforward compensation", *IEE Conference Publication*, v 2, n 332, p 1111-1116, 1991.

[73] Z. Yu, H. Chen, P. Y. Woo, "Advanced gain scheduled H infinity controller for robotic manipulators, *Robotica*, v 20, n 5, p 537-544, September/October, 2002.

[74] L. X. Wang, "*A course in Fuzzy Systems and Control*", Prentice Hall, 1997.

[75] H. Bersini, L. Gonzales, E. Decossaux, "Hoplfield Net Generation and Encoding of Trajectories in Constrained Environments", In *Neural Networks in Robotics,* Kluwer Academic Publishers 1993.

[76] J. S. R. Jang, C. T. Sun, E. Mizutani, "*Neuro-Fuzzy and Soft Computing, A Computational Approach to Learning and Machine Intelligence*", Prentice-Hall Engineering, 614 pages, 1996

[77] M. Al-Khatib and J. J. Saade, "An Efficient Data-Driven Fuzzy Approach to the Motion Planning Problem of a Mobile Robot", *Fuzzy Sets and Systems,* vol.134, no. 1, 16, p 65-82, 2003

[78] A. Khoukhi, K. Demirli, L. Baron, M. Balazinski, "Fuzzy-Neuro Optimal Time-Energy Control of a Three Degrees of Freedom Manipulators", *NAFIPS'06 (North American Fuzzy Information Processing Society),* Concordia University, Montreal, Qc, Canada, June 3-6, 2006.

[79] A. Khoukhi, K. Demirli, L. Baron, M. Balazinski, "Hierarchical Hybrid Neuro-Fuzzy System to Optimal Time-Energy Trajectory Planning of Redundant Manipulators", en revision avec *Engineering Applications of Artificial Intelligence EAAI* 2007.

[80] S. L. Chiu, "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent and Fuzzy Systems*, vol. 2, 267-278, 1994.

[81] K. Demirli, S. X. Cheng, P. Muthukumaran, "Subtractive Clustering Based Modeling of Job Sequencing with Parametric Search", *Fuzzy Sets and Systems;* vol. 137, p 235-270, 2003.

[82] A. Khoukhi, L. Baron, M. Balazinski: "Multi-Objective Trajectory Planning for Redundant Manipulators", accepté à la 5ème *Conférence Internationale sur l'Automatisation Industrielle,* ÉTS, Montréal, Canada, 11-13 juin 2007

[83] A. Khoukhi, L. Baron, M. Balazinski: "An Augmented Lagrangian with Projection Technique Multi-Objective Trajectory Planning of Redundant Manipulators", *CCToMM, Symposium on Mechanics, Machines and Mechatronics,* 31 may-1 juin 2007, at the Canadian Space Agency (CSA) , Saint Hubert (Montréal), QC, Canada.

[84] A. Khoukhi, L. Baron, M. Balazinski: "Multi-Objective Trajectory Planning for Redundant Manipulators Using Augmented Lagarangian", soumis à *Trans. of the Canadian Society of Mechanical Engineering,* 2007

[85] A. Khoukhi, K. Demirli, M. Balazinski, L. Baron: "A Data-Driven Neuro-Fuzzy System for Online Motion Planning of Redundant Manipulators", Accepté 2007 *IEEE Int. Conf. on Sys. Man, and Cyb.* Oct.7-10, Montréal, Canada.

[86] A. Khoukhi, K. Demirli, L. Baron, M. Balazinski: "Data-Driven Neuro-Fuzzy Multi-Objective Trajectory Planning of Redundant Manipulators", soumis à *Control and Intelligent System,* 2007.

[87] E. Polak: "*Optimization, Algorithms and Consistent Approximation*", Springer, N.Y, 1997.

[88] A. J. Kerner: "The Convergence of the Extended Kalman Filter", *Directions In Mathematical Systems Theory & optimization,* A. Rautzer & C. I. Byrnes Eds. Springer, p 1274-182, 2003.

ANNEXE A

Ce travail a été publié à la conférence CCToMM'2007, sur la Mécanique, les Machines et la Mécatronique, qui a eu lieu les 31 mai et 1 juin 2007 à l'Agence Spatiale Canadienne, Saint Hubert. Il est aussi soumis pour publication à la revue Transactions de la Société Canadienne de Génie Mécanique (SCGM)

MULTI-OBJECTIVE TRAJECTORY PLANNING FOR REDUNDANT MANIPULATORS USING AUGMENTED LAGRANGIAN

Amar KHOUKHI, Luc BARON, Marek BALAZINSKI

Department of Mechanical Engineering,

École Polytechnique de Montréal,

C. P. 6079, Succ. CV, Montreal, QC, Canada H3C 3A7

(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

A. 1. Abstract In this paper, a multi-objective trajectory planning system is developed for redundant manipulators. This system involves kinematic redundancy resolution, as well as robot dynamics, including actuators model. The kinematic redundancy is taken into account through a secondary criterion of joint limits avoidance. The optimization procedure is performed subject to limitations on actuator

torques and workspace, while passing through imposed poses. The Augmented Lagrangian with decoupling (ALD) technique is used to solve the resulting constrained non convex and nonlinear optimal control problem. Furthermore, the final state constraint is solved using a gradient projection. Simulations on a three degrees of freedom planar redundant serial manipulator show the effectiveness of the proposed system.

Key words: Redundant Manipulators, Trajectory Planning, Augmented Lagrangian, Decoupling, Projection

PLANIFICATION MULTI-OBJECTIVE DE TRAJECTOIRE DES MANIPULATEURS REDONDANTS PAR LAGRANGIEN AUGMENTÉ

A. 2. Résumé Dans cet article, nous considérons le problème de planification multi-objective de trajectoire des manipulateurs sériels redondants. À partir des modèles cinématique et dynamique du robot, et ceux de l'espace de travail et de la tâche, le problème est formulé dans le cadre du calcul variationnel, comme un programme non linéaire sous contraintes. La redondance cinématique est résolue en appliquant une version modifiée de l'algorithme de la Jacobienne augmentée. La technique du Lagrangien augmenté est ensuite appliquée à une représentation découplée de la dynamique du robot afin de résoudre le problème résultant de commande optimale.

Des études de simulations montrent l'efficacité de cette approche comparée à celles développées jusqu'à date, notamment les approches utilisant les méthodes de pénalité ou celles basées uniquement sur la cinématique du robot.

Mots Clés : Manipulateurs Redondants, Planification de trajectoire, Lagrangien augmenté, Découplage, Projection

A. 3. Introduction

A great advantage of robots is their ability and flexibility to rearrange themselves for new tasks. Utilization of robot's flexibility presupposes effective motion planning. This is aimed at generating trajectories for a specific task, according to a set of desired performance criteria [1, 2]. The task is usually specified in terms of a motion of the end-effector (EE), which results in a geometric path to be followed with a given time law. Moreover, the robot arm is actuated at the joints, thus requiring control actions to be performed by the joint servos. A feasible planning could be satisfactory if the required target performance is not too tight. Otherwise, one must use an optimal (or near optimal) approach, and then test how it is robust to changes of the dynamic parameters [3, 4]. One of the motion control algorithm problems is that the desired trajectory may cause saturation of the speed and/or torques delivered by the joint actuators in the vicinity of singularities. This might occur in many regions of the workspace, due to nonlinear kinematic transformations between task and joint spaces. Furthermore, when the assigned trajectory results are

unfeasible due to actuation limits or passing through singular poses, the motion planning system must still generates torques that allow achieving the performance criteria, while avoiding singularities and satisfying other task-related constraints. Several studies had considered this problem [6-9]. Some of them dealt with kinematic redundancy resolution, while others included robot dynamics and force optimization [3-5].

In this paper, the multi-objective trajectory-planning problem is formulated based on kinematic and dynamic, including actuators models. The cost functional involves time optimization through sampling period variations and electric energy as well as a measure of manipulability. The resulting constrained non-convex and nonlinear optimal control problem is solved using AL technique on a decoupled form of the robot dynamics. The advantage of using such a technique – as compared to other optimization methods like penalty methods – is its ability to deal with non convexities (due mainly to the strong non linear character of the system's constraints) and ill conditioning that may occur during the iterative resolution process. Furthermore, in many applications, such as pick and place or assembly tasks, the final state attainability constraint is a primary issue. This constraint is achieved through a gradient projection algorithm [10-11]. After problem modelling and formulation, simulation results on a three degrees of freedom planar redundant serial manipulator show very encouraging results as compared to other approaches, namely, minimum-time control and kinematic-based methods. In section 5.4, the kinematic and dynamic models are considered as well as other associated constraints. In section 5.5, the

augmented Lagrangian with decoupling and projection gradient technique is developed. Section 5.6 provides simulation results, and section 5.7 concludes this paper.

A. 4. Modeling

A. 4. 1. Kinematic modeling

The forward kinematics problem deals with the determination of the EE motion from a given motion of the joints [12]. At the velocity level, it is expressed in a vector form as:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\boldsymbol{q}} \tag{1}$$

where $\boldsymbol{q}(t) = [q_1(t), q_2(t), ..., q_n(t)]^T$ is the n-dimensional joint angles vector, $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_m(t)]^T$ is the m-dimensional position vector of the EE and $\mathbf{J}$ is the mxn robot's Jacobian [12]. Although this study is applicable to a general n-degrees-of-freedom (DOF) serial manipulator with m-DOF at the Cartesian EE level, it is implemented on a two dimensional positioning robot.

The inverse kinematics problem is the determination of the joints motion from a given EE motion. Because $\mathbf{J}$ is not a square matrix, a kinematic redundancy holds for this inverse kinematics problem. A first approach to utilize redundancy to solve the inverse kinematics was proposed by Liégeois [7]. In his approach, a gradient projection is used to devise a general solution expressed as:

$$\dot{\boldsymbol{q}} = \mathbf{J}^{\perp}\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^{\perp}\mathbf{J})\mathbf{z} \tag{2}$$

where $\mathbf{J}^{\perp}$ is a generalized inverse of $\mathbf{J}$. The pseudo-inverse solution in the first term of eq. (2) is known to minimize the two-norm of the joint velocity vector, whereas the second term is called the homogeneous or null-space solution. The later does not contribute to the EE motion yet determines the minimum-norm solution. However, it has been shown that rather than driving the robot away from singularities at very high demands in joint velocities, this solution sometimes leads the robot to singularities [8]. A weighted pseudo-inverse by the inertia matrix is used here. This allows dynamic consistency compared to traditional pseudo-inverse. It is given by:

$$\mathbf{J}_{\mathbf{D}}^{\perp}=\mathbf{D}^{-1}\mathbf{J}^{T}(\mathbf{J}\mathbf{D}^{-1}\mathbf{J}^{T})^{-1} \tag{3}$$

In order to include a secondary task criterion within a performance index $r(\boldsymbol{q})$, $\mathbf{z}$ is chosen to be

$$\mathbf{z} = \pm b\nabla r(\boldsymbol{q}) \tag{4}$$

where b is a positive real number and $\nabla r(\boldsymbol{q})$ the gradient of $r(\boldsymbol{q})$. A positive sign in eq. (4) indicates that the criterion is to be maximized. A negative sign indicates minimization. Including joint limits avoidance constraint through redundancy

resolution might be performed by choosing $r(\boldsymbol{q}) = \frac{1}{2}(\boldsymbol{q} - \bar{\boldsymbol{q}})^T \mathbf{W}(\boldsymbol{q} - \bar{\boldsymbol{q}})$ with $\bar{\boldsymbol{q}}$ being chosen such that it ensures joint limits avoidance. For example,

$$\bar{\boldsymbol{q}} = \frac{1}{2}(\boldsymbol{q}_{\max} + \boldsymbol{q}_{\min}) \tag{5}$$

yields

$$\mathbf{z} = -\mathbf{W}(\boldsymbol{q} - \bar{\boldsymbol{q}}), \tag{6}$$

where $\mathbf{W}$ is a positive weighting matrix to scale the magnitude of the manipulator response to joint displacement. A typical choice for this matrix is $\mathbf{W} = \text{diag}\,(\boldsymbol{q}_{\max} - \boldsymbol{q}_{\min})$

A. 4. 2. Dynamic modelling

The robot dynamic model is developed using a Lagrangian formalism, which includes actuators model. This model allows closed-form expression of joint rates and accelerations characterizing the motion resulting from joint torques. It can be expressed in continuous-time as:

$$\mathbf{D}(\boldsymbol{q})\,\ddot{\boldsymbol{q}} + \mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \mathbf{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{7}$$

where $\boldsymbol{\tau}$ is the $n \times 1$ joint torques vector produced by the joint actuators, $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}}$ are vectors describing joint positions, rates and accelerations, $\mathbf{D}(\boldsymbol{q})$ is the

$n \times n$ manipulator inertia matrix, $\mathbf{V}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the $n \times n$ matrix representing Coriolis and centrifugal wrenches, and $\mathbf{G}(\boldsymbol{q})$ is the $n \times 1$ vector representing gravity forces [12]. Now, following references [2, 3], the discrete-time dynamic model can be approximated as:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & h_k \boldsymbol{I}_{n\times n} \\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n} \end{bmatrix} \boldsymbol{x}_k + \mathbf{D}^{-1}(\boldsymbol{x}_{1k}) \begin{bmatrix} \frac{h_k^2}{2} \boldsymbol{I}_{n\times n} \\ h_k \boldsymbol{I}_{n\times n} \end{bmatrix} \left[\left[\boldsymbol{\tau}_k - \mathbf{V}_c(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k}) \boldsymbol{x}_{2k} + \mathbf{G}(\boldsymbol{x}_{1k}) \right] \right] \quad (8)$$

where $\boldsymbol{x}_k = (\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)^T = (\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})^T$ is the $2n$-dimensional robot state and $\boldsymbol{I}$ is the $n \times n$ identity matrix. Equation (8) is written for simplicity as:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_{dk}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \quad (9)$$

A. 4. 3. Constraints modeling

In addition to eqs. (2)-(9), the following constraints are considered:

- Boundary conditions: These concern the starting and final states $\boldsymbol{x}_s$ and $\boldsymbol{x}_T$

$$\boldsymbol{x}_0 = \boldsymbol{x}_S, \ \boldsymbol{x}_N = \boldsymbol{x}_T \quad (10)$$

- Redundancy resolution ensuring joint limits avoidance: This is achieved using eqs. (2) and (3)

$$\boldsymbol{x}_{2k} = \mathbf{J}_{Dk}^{\perp}(\boldsymbol{x}_{1k}) \dot{\mathbf{x}}_k + (\mathbf{I} - \mathbf{J}_{Dk}^{\perp}(\boldsymbol{x}_{1k}) \mathbf{J}_k(\boldsymbol{x}_{1k})) \mathbf{z}_k(\boldsymbol{x}_{1k}) \quad (11)$$

with $$\mathbf{z}_k = -\mathbf{W}\ (\boldsymbol{x}_{1k} - \bar{\boldsymbol{x}}_{1k}),\ \mathbf{W} = \mathrm{diag}(\frac{\boldsymbol{x}_{1\max} - \boldsymbol{x}_{1\min}}{2}) \tag{12}$$

where $\bar{\boldsymbol{x}}_{1k}$, $\boldsymbol{x}_{1\max}$ and $\boldsymbol{x}_{1\min}$ refer, respectively, to discrete values of joint angles $\bar{\boldsymbol{q}}$, $\boldsymbol{q}_{\max}$ and $\boldsymbol{q}_{\min}$

- Admissible domain of the sampling periods: If the overall robot traveling time $T = \sum_{k=0}^{N-1} h_k$ is too small, there may be no admissible solution to the optimal control problem, since the torque constraints bound indirectly the path traversal time. On the other hand, the sampling period h_k must be smaller than the system smallest time constant in order to prevent the system from being uncontrollable between two control times. The determination of the system smallest time constant and the maximum value of a sampling period is out of the scoop of this paper. It is assumed that they are available from a prior design are provided by the robot manufacturer. In this paper, a trade-off is made through variation of the sampling period within an admissible domain H defined as:

$$h_{\min} \le h_k \le h_{\max}\ ,\ k = 0,\ldots, N-1 \tag{13}$$

- Admissible domain of the actuator torques: Non-violation of control torque limits is another major issue for trajectory planning. The required leg forces must

continuously be checked for possible violation of the limits as the manipulator moves close to a singular pose. As soon as any leg actuator crosses its limit, the optimal planning procedure has to determine an alternate leg actuation strategy leading to another path on which the actuator forces would be constrained within the limits. In this paper, robot torques are assumed to belong to a compact and bounded set $\mathbf{C} \subset \Re^{6N}$, expressed as:

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau}_k \leq \boldsymbol{\tau}_{\max} \ , \ k = 0,..., N-1 \tag{14}$$

- Imposed passage through intermediate EE poses: This consists of a set of two-dimensional point positions defined as:

$$\left\| \mathbf{p} - \mathbf{p}_l \right\| - \mathrm{T}_{\mathrm{PassTh}lp} = 0, \ l = 1,...,L \tag{15}$$

where p is the current position vector of the EE, $\mathbf{p}_l$ is the lth passage point and L is the number of imposed points and $\mathrm{T}_{\mathrm{PassTh}lp}$ is the passage tolerance. For the sake of simplicity, all equality and inequality constraints are written as $s_i(\boldsymbol{x}) = 0 \quad i = 1,...,\mathrm{I}$ and $\mathbf{g}_j(\boldsymbol{x}, \boldsymbol{\tau}) \leq 0 \quad j = 1,...,\mathrm{J}$, regardless if they depend only on state, control inputs or both, where I and J denote the numbers of equality and inequality constraints.

A. 5. Non-Linear Programming Formulation

Several criteria had been proposed, such as traveling time, consumed energy, obstacle avoidance and manipulability measure [2-6]. The most popular is traveling time for the obvious reason of production targets. However, the major drawback of this control criterion is its bang-bang character, producing non-smooth trajectories exceeding joint speed and/or acceleration limits achievable by the actuators [4]. To deal with this problem, some authors introduced a virtual time augmenting the actual time in an effort to track the desired trajectory. In this paper, a multi-objective planning strategy is implemented.

The discrete-time constrained optimal control problem consists of finding the optimal sequences $(\boldsymbol{\tau}_0, \boldsymbol{\tau}_2, ..., \boldsymbol{\tau}_{N-1})$ and $(h_0, h_2, ..., h_{N-1})$ allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a target state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the multi-objective performance index E_d including energy, traveling time, and a singularity avoidance function. It is expressed as:

$$\underset{\substack{\tau_k \in \mathsf{C} \\ h_k \in \mathrm{H}}}{\mathrm{Min}} \left\{ E_d = \left[\sum_{k=0}^{N-1} [\boldsymbol{\tau}_k \boldsymbol{U} \boldsymbol{\tau}_k^T + \iota + \delta \varpi (\boldsymbol{x}_{1k})] h_k \right] \right\} \tag{16}$$

where $\mathsf{C}, \mathrm{H}, \boldsymbol{U}$, are, respectively, the set of admissible torques, the set of admissible sampling periods, and electric energy matrix weight, ι and δ are positive scalars

that weight sampling time and the singularity avoidance function ϖ . This singularity avoidance function is defined as:

$$\varpi(\boldsymbol{x}_{1k}) = \frac{1}{\sqrt{\det(\mathbf{J}(\boldsymbol{x}_{1k})\mathbf{J}^{T}(\boldsymbol{x}_{1k}))}} \tag{17}$$

This optimization problem is performed subject to constraints (9)-(15).

A. 5. 1. Augmented Lagrangian (AL)

The problem (16) is a multi-objective non-linear and non-convex optimal control problem. It is solved using an Augmented Lagrangian technique, which transforms the constrained problem into a non-constrained one. The degree of penalty for violating the constraints is regulated by penalty parameters. This method basically relies on quadratic penalty methods, but reduces the possibility of ill conditioning of the sub-problems that are generated with penalization by introducing explicit Lagrange multipliers estimates at each step into the function to be minimized [10, 11]. The AL function is written as:

$$L_{\mu}(\boldsymbol{x}, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = E_d + \sum_{k=0}^{N-1} \boldsymbol{\lambda}_{k+1}^{T} \left[\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \nu_k, h_k) \right] +$$

$$\sum_{k=0}^{N-1} h_k \left\{ \sum_{j=1}^{2} \left[\Phi_{\mu_g}(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k)) + \Psi_{\mu_s}(\boldsymbol{\sigma}_k^j, \boldsymbol{s}_j(\boldsymbol{x}_k)) \right] \right\} \tag{18}$$

In eq. (18), the function $f_{d_k}(x_k,\tau_k,h_k)$ is defined by the discrete state eq. (9) at the sampling point k, N is the total sampling number $\lambda \in R^{6N}$ designates the co-states obtained from the adjunct equations, σ and ρ are Lagrange multipliers with appropriate dimensions, associated to equality and inequality constraints and μ_s and μ_g are the corresponding penalty coefficients. The adopted penalty functions combine penalty and dual methods allowing relaxation of inequality constraints as soon as they are satisfied. These are given by:

$$\Psi_{\mu_s}(a,b)=(a+\frac{\mu_s}{2}b)^T b,\ \Phi_{\mu_g}(a,b)=\frac{1}{2\mu_g}\left\{\left\|\mathrm{Max}(0,a+\mu_g b)\right\|^2-\left\|a\right\|^2\right\} \tag{19}$$

where a and b refer to Lagrange multipliers and left hand side of equality and inequality constraints. The Karush-Kuhn-Tucker first order optimality conditions [11] state that for a trajectory $(x_0,\tau_0,h_0,...,x_N,\tau_{N-1},h_{N-1})$ to be optimal solution to the problem, there must exists some positive Lagrange multipliers (λ_k,ρ_k), unrestricted sign multipliers σ_k and positive penalty coefficients μ_s and μ_g such that:

$$\frac{\partial L_\mu}{\partial x_k}=0,\frac{\partial L_\mu}{\partial \tau_k}=0,\ \frac{\partial L_\mu}{\partial h_k}=0,\ \frac{\partial L_\mu}{\partial \lambda_k}=0,\frac{\partial L_\mu}{\partial \rho_k}=0,\ \frac{\partial L_\mu}{\partial \sigma_k}=0,$$

$$\sigma_k^T s(x)=0,\ \rho_k^T g(x,\tau,h)=0 \quad \text{and} \quad g(x,\tau,h)\le 0 \tag{20}$$

The development of the above conditions enables one to derive the iterative formulas to solve the optimal control problem by updating control variables, Lagrange multipliers and penalty coefficients. However, in eq. (9), $f_{d_k}(x_k, \tau_k, h_k)$ contains the inverse of the inertia matrix $\mathbf{D}^{-1}(x_1)$ and Coriolis and centrifugal wrenches $\mathbf{V}(x_1, x_2)$. These might be very cumbersome to express. In developing the first order optimality conditions and computing the co-states λ_k, an inverse of the mentioned inertia matrix and its derivatives with respect to state variables must be computed, resulting in huge calculations.

A. 5. 2. Augmented Lagrangian with decoupling (ALD)

The computational difficulty mentioned beforehand is solved using a linear-decoupled formulation [13].

Theorem: Under the invertibility condition of the inertia matrix, the control law defined as

$$u = \mathbf{D}(x_1)v + \mathbf{V}(x_1, x_2)x_2 + \mathbf{G}(x_1) \tag{21}$$

allows the robot to have a linear and decoupled behavior with a dynamic equation:

$$\dot{x}_2 = v \tag{22}$$

where v is an auxiliary input.

This can be demonstrated by substituting the proposed control law given by eq. (21)

into the dynamic model eq. (7). One gets $\mathbf{D}(\boldsymbol{x}_1)\dot{\boldsymbol{x}}_2 = \mathbf{D}(\boldsymbol{x}_1)\boldsymbol{v}$, since $\mathbf{D}(\boldsymbol{x}_1)$ is invertible, it follows that $\dot{\boldsymbol{x}}_2 = \boldsymbol{v}$

This gives the robot a decoupled linear behavior approximated by the following discrete linear equation:

$$\boldsymbol{x}_{k+1} = \mathbf{F}_{dk}\boldsymbol{x}_k + \mathbf{B}_{dk}\boldsymbol{v}_k = \boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \tag{23}$$

with

$$\mathbf{F}_{dk} = \begin{bmatrix} \boldsymbol{I}_{n\times n} & h_k\boldsymbol{I}_{n\times n} \\ \boldsymbol{O}_{n\times n} & \boldsymbol{I}_{n\times n} \end{bmatrix}, \quad \mathbf{B}_{dk} = \begin{bmatrix} \frac{h_k^2}{2}\boldsymbol{I}_{n\times n} \\ h_k\boldsymbol{I}_{n\times n} \end{bmatrix} \tag{24}$$

Notice that while this dramatically eases the calculation of the co-states. The non-linearity is transferred to the objective function. The decoupled problem consists then of finding the optimal sequences of sampling periods and accelerations $(h_0, h_2, ..., h_{N-1}), (\boldsymbol{v}_0, \boldsymbol{v}_2, ..., \boldsymbol{v}_{N-1})$, allowing the robot to move from an initial state $\boldsymbol{x}_0 = \boldsymbol{x}_S$ to a final state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost function E_d^D, expressed as:

$$\underset{\substack{v\in\mathbf{V} \\ h_k\in\mathbf{H}}}{\mathrm{Min}}\left\{ E_d^D = \sum_{k=0}^{N-1}\Big[\left[\mathbf{D}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \mathbf{V}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \mathbf{G}(\boldsymbol{x}_{1k})\right]^T \boldsymbol{U}\left[\mathbf{D}(\boldsymbol{x}_{1k})\boldsymbol{v}_k + \mathbf{V}(\boldsymbol{x}_{1k}, \boldsymbol{x}_{2k})\boldsymbol{x}_{2k} + \mathbf{G}(\boldsymbol{x}_{1k})\right] + \mathfrak{l} + \delta\varpi(\boldsymbol{x}_{1k}) \Big] h_k \right\} \tag{25}$$

Subject to the decoupled dynamic state eq. (22), and the above-mentioned constraints. The augmented Lagrangian function (ALD) associated to the decoupled problem (25) is

$$L_{\mu}^{D}(\boldsymbol{x},\boldsymbol{v},h,\boldsymbol{\lambda},\boldsymbol{\rho},\sigma)=E_{d}^{D}+\sum_{k=0}^{N-1}\boldsymbol{\lambda}_{k+1}^{T}\left[\ \boldsymbol{x}_{k+1}-\boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k,\boldsymbol{v}_k,h_k)\ \right]$$

$$+\sum_{k=0}^{N-1}h_k\left\{\sum_{j=1}^{2}\left[\Phi_{\mu_g}(\boldsymbol{\rho}_k^j,\boldsymbol{g}_j(\boldsymbol{x}_k,\boldsymbol{\tau}_k))+\Psi_{\mu_s}(\boldsymbol{\sigma}_k^j,\boldsymbol{s}_j(\boldsymbol{x}_k))\right]\right\} \tag{26}$$

with the function $\boldsymbol{f}_{d_k}^{D}(\boldsymbol{x}_k,\boldsymbol{v}_k,h_k)$ being defined by the state of eq (23), and other parameters appearing in eq. (26) are defined above. Again, the development of the first order optimality conditions allows deriving iterative formulas for control and state variables, Lagrange multipliers as well as penalty coefficients. These expressions are quite long and are not detailed here. The final state constraint $\boldsymbol{x}_N=\boldsymbol{x}_T$ is not included within the penalty procedure, as we have to satisfy it at each iteration. A re-adjustment is performed with an orthogonal projection on the tangent space of this constraint, through the application of the descent direction

$$\boldsymbol{d}\ =-\mathbf{P}_{v}\nabla_{v}L_{\mu}^{D} \tag{27}$$

with $\mathbf{P}_v$ being defined as

$$\mathbf{P}_{v}=\mathbf{I}_{d}-\mathbf{Q}_{v}^{T}(\mathbf{Q}_{v}\,\mathbf{Q}_{v}^{T})^{-1}\,\mathbf{Q}_{v} \tag{28}$$

where $\mathbf{I}_d$ is an identity matrix with appropriate dimension and $\mathbf{Q}_v$ is the projection matrix on the tangent space of the final state constraint. The re-adjustment process allows satisfying target attainability with any given ε- precision [11]. If other imposed states are to be satisfied at each iteration, then the re-adjustment procedure must be extended to these constraints. Figure A. 1 depicts a flowchart diagram of ALD function and architecture for the multi-objective trajectory-planning problem. In this procedure, one has to select robot parameters, task definition, (such as starting, intermediate and target positions), workspace limitations and simulation parameters (block 1). Then a kinematic unit (block 2) defines a feasible initial solution satisfying boundary constraints in joint angles, velocities, accelerations and jerks. This solution is defined through a cycloidal profile. This profile has been chosen as it allows a near-minimum time smooth continuous trajectory as compared to a trapezoidal profile [14]. Then an inner optimization loop (block 3) solves for the AL minimization with respect to sampling periods and acceleration control variables. One first computes the gradients of the Lagrangian, the co-states backwardly and the projection matrix and operator. Then a steepest descent is calculated and tested against a suitable tolerance (block 4). If non-satisfied, one computes new search directions and updates sampling time and acceleration inputs. Then go back to inner optimization loop to update gradients and direction descent. When satisfied, one goes further to test other equality and inequality constraints against feasibility tolerances. If non-feasible, go back to the inner optimization unit. Else, if feasible, do a

convergence test for cost minimization and constraints satisfaction against optimal tolerances. If convergence holds, display optimal trajectory and end the program. Otherwise, go further to the dual part of AL to test for constraints satisfaction and update penalty and tolerance parameters. If the constraints are not violated with respect to first order optimal tolerances then the multipliers are updated without decreasing penalty. If they are violated, decrease penalty while keeping unchanged Lagrange multipliers, to ensure that the next sub-problem will place more emphasis on reducing the constraints violations. In both cases the tolerances are decreased to force the subsequent primal iterates to be increasingly accurate solutions for the primal problem.

A. 6. Simulation Results

A three revolute (n=3-DOF) serial manipulator moving on the vertical plan with a 2-DOF task is considered (Fig.A. 2). The robot kinematic and dynamic parameters are given in Tables 5. 1 and 5. 2. The simulation objectives are to: 1) minimize travelling time and instantaneous energy during the motion; 2) resolve the redundancy and avoid singularities; 3) satisfy several constraints related to limits of joint angles, rates, accelerations and torques. The first example is to move the tip of the manipulator along a straight line from the starting position (1.95, 0.82) (m), corresponding to joint angles (0°, 30°, 30°), to the final position (1.35, 1.3) (m), without considering the orientation. Hence, this task is performed with a serial 3-DOF

planar manipulator that is redundant with respect to the given task. The joint velocities are zero at the starting and ending positions and the EE travels through a distance of 0.75 m. The multi-objective trajectory is performed by taking each weight in the performance index equals to unity. The initial pose values are chosen to satisfy a secondary goal, which consists of avoiding joint limits. Figure A. 3 shows the (x, y) position variations for the cycloidal minimum-time trajectory and those generated based on ALD approach including robot dynamics, kinematics and constraints. Although the cycloidal minimum-time trajectory is a straight line, it is slightly disturbed in the case of its ALD counterpart. The bias between the two paths is due to the non-linearity of the dynamic model, considering for example the gravity effects. Figure A. 4 displays the corresponding joint angle variations. Figure A. 5 shows the associated instantaneous variations of the consumed electrical and kinetic energy and sampling periods. One notices the significant and monotonous diminishing of the consumed energy with the ALD as compared to the cycloidal one. As for traveling time, after 4 outer and 4 inner ALD iterations, one notices a dramatic reduction compared to the initial cycloidal trajectory, with $t_{\text{Cycloidal}} = 2\,\text{sec}$ and $t_{\text{Multi-objective}} = 1.237\,\text{sec}$, an increase of 45% for the initial solution. On the other hand, although one gets a feasible path with the cycloidal profile, the associated torques and necessary energy, computed from the robot inverse dynamics are fairly high and exceed the nominal values. The same trajectory has been performed with three imposed passage points for the robot EE. Figure A. 6 shows the (x, y)

coordinate variations while satisfying passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m), (in this Figure, the cycloidal trajectory does not consider this constraint). Figure A. 7 shows consumed energy and traveling time for the trajectory with passages. The second trajectory is a second straight line, starting from the joint position (0^0, 15^0, 20^0) corresponding to the Cartesian coordinates (0.2, 0.9) (m) to the ending position (0.64, 0.45) (m). The consumed energy and sampling period variations are shown in Fig. 5. 8. The third trajectory is a circle with a center (-0.045, 0.75) (m) and radius 0.14 (m). The joint angle variations are shown in Fig 5. 9, and the consumed energy and traveling time are shown in Fig. 5. 10. In order to assess the sensitivity of the proposed multi-objective trajectory planning to dynamic parameter changes, the mass of the third link was increased by 1.5 Kg. Figs. A. 11 (a) and (b) show respectively, the variations of sampling time and consumed energy for the original and modified third link mass trajectories. Although the necessary energy and traveling time to perform the modified link mass task grow greater, the achieved performance is good, as it reaches the position with a precision of order 10^{-3} (m). This highlights the ALD good robustness to parameters changes as illustrated in Fig A. 12. On the whole, the computation time is quite long. It took about 9 minutes on a Pentium III, 996 MHz, to simulate and get the performances of the first straight line trajectory. This is due mainly to the high nonlinear robot dynamics and projection matrix and operator calculations to satisfy the final state constraint at each iteration.

Table A. 1. Kinematic and dynamic parameters of the manipulator

Link I	Mass (kg)	Inertia I_i(kg m^2)	Length L_i(m)
1	8	9.23	1.0
2	4	7.5	0.8
3	2	5.21	0.45

Table A. 2. Limits of workspace, actuator torques, joint rates and accelerations and sampling periods

Parameter	x (m)	y (m)	τ_1 (Nm)	τ_2 (Nm)	τ_3 (Nm)	$\dot{q}_{iMotor}$ (rad/s)	$\ddot{q}_{iMotor}$ (rad/s^2)	h (sec)
Max	2.5	2.5	40	25	20	2	3	0.1
Min	0.4	0.20	-40	-25	-20	-2	-3	0.001

A. 7. Conclusion

In this paper, a method for computing a multi-objective trajectory planning is developed. This method includes joint angles, rates, accelerations, jerks, workspace, and actuator torques limitations. It is solved through an Augmented Lagrangian on a decoupled form of the robot dynamics. According to preliminary simulation results, this approach is effective in solving the non-convex and non-linear constrained motion-planning problem. The trajectories are smooth and singularity free, a capability which makes them very suitable for use as reference inputs to a feedback like PID position controller or as training datasets on which to build an objective data-driven neuro-fuzzy system for on-line planning and control. As a short trend for

this work, it is to use the outcomes of such a multi-objective trajectory planning to build a data-driven neuro-fuzzy control system for on-line planning with reasonable computational time.

Acknowledgments

The authors gratefully thank the anonymous reviewers for the valuable comments and the Natural Science and Engineering Research Council of Canada (NSERC) for supporting this work under grants ES D3-317622, RGPIN-203618, RGPIN-105518, and STPGP-269579.

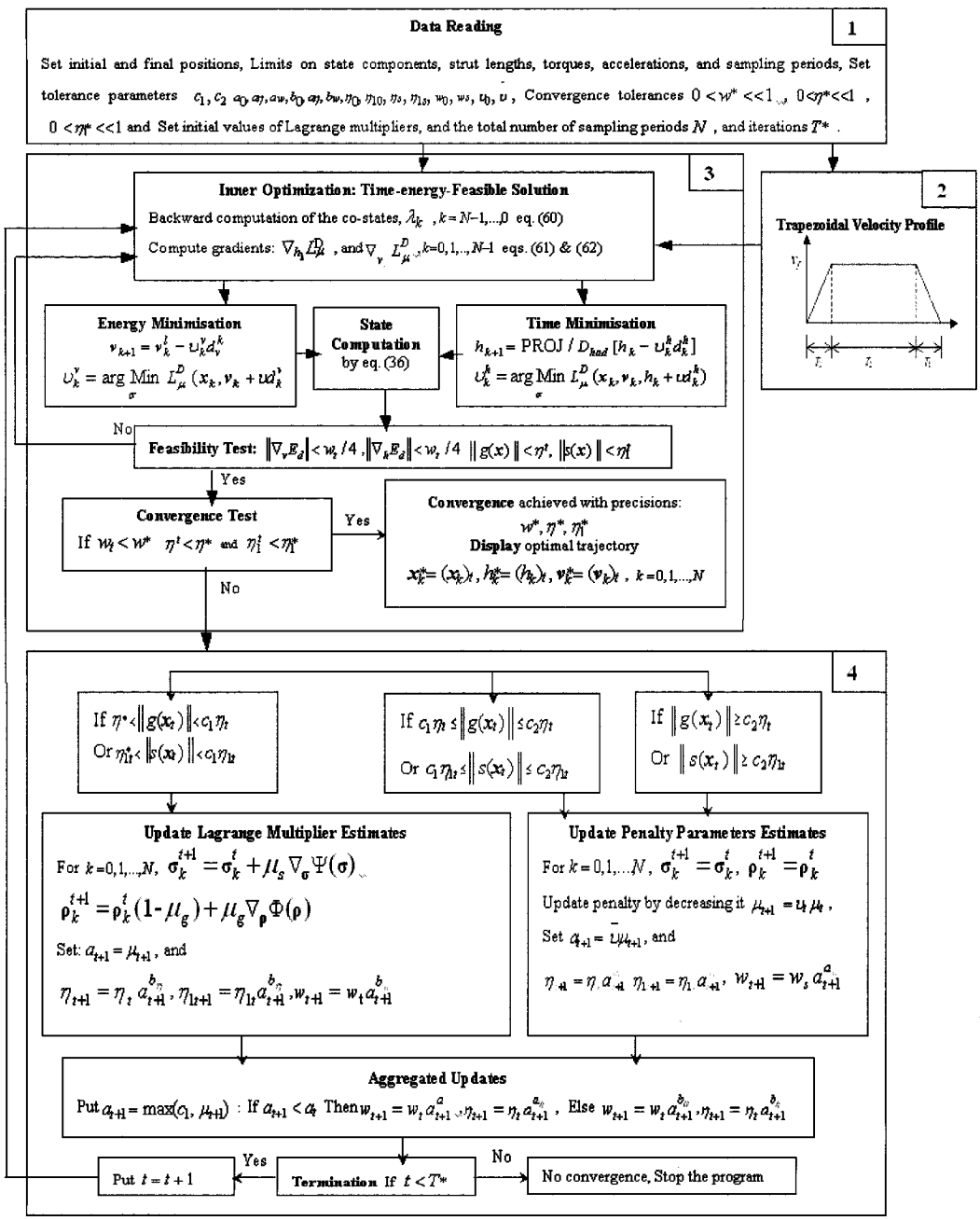


Figure A. 1. Flowchart for the augmented Lagrangian Technique for the trajectory- planning problem

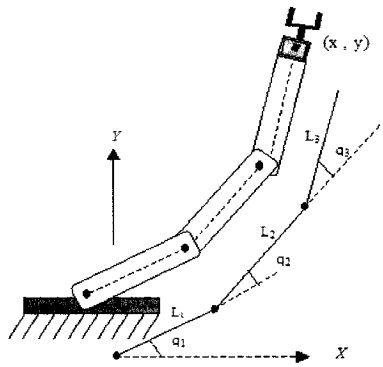


Fig. A. 2. Geometry of a 3-DOF planar manipulator of joint angles q_j, $j = 1,2,3$

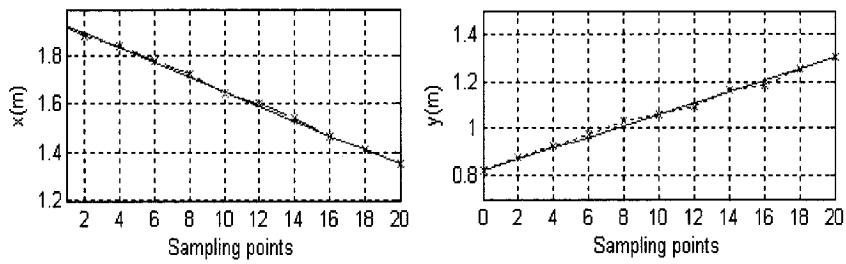


Fig. A. 3. EE trajectory along axis x and y displacement in meters versus sampling points, (___) Cycloidal profile, (*-*-*-*) Minimum-time including robot dynamics and constraints

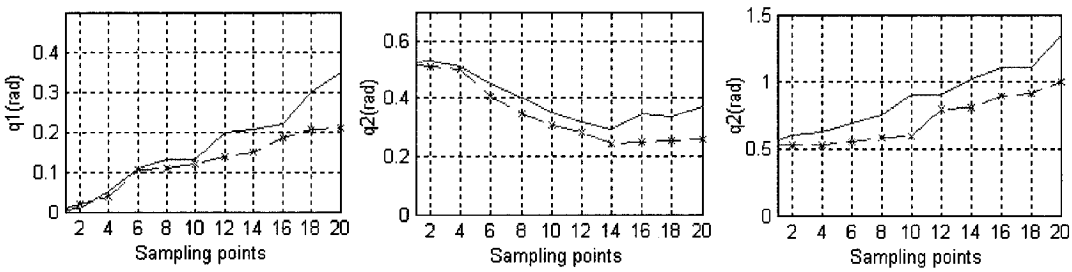


Fig.A. 4. Joint q_i variations
(___Cycloidal profile, *-*-*- ALD trajectory)

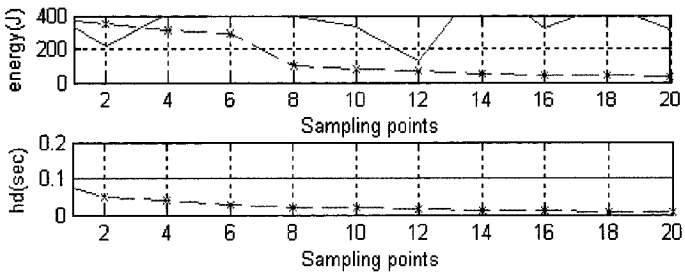


Fig.A. 5. Corresponding energy and sampling period *h* variations
(___Cycloidal profile, *-*-*- ALD trajectory)

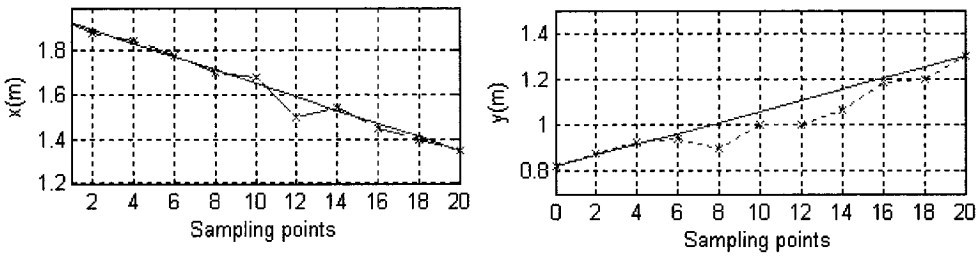


Fig. A. 6. (*x, y*) coordinate variations while satisfying the passage through imposed points (1.7, 0.9), (1.5, 1.0), (1.4, 1.2) (m)
(___Cycloidal profile, *-*-*- ALD trajectory)

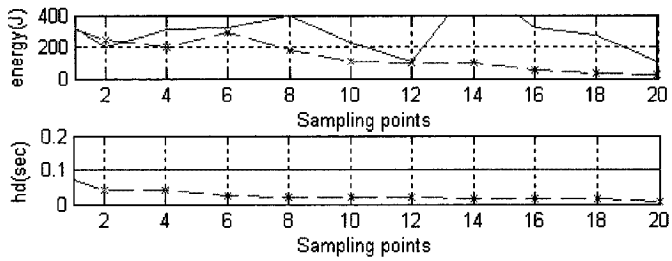


Fig. A. 7 Associated consumed energy and traveling time to the trajectory with imposed passages
(___Cycloidal profile, *-*-*- ALD trajectory)

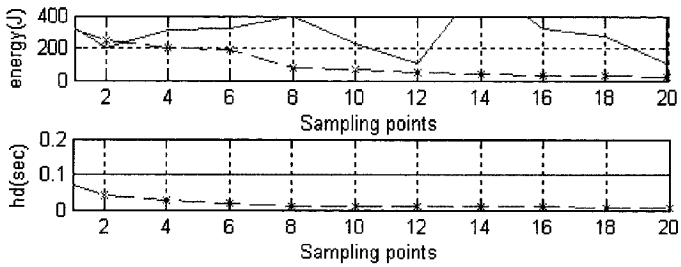


Fig. A. 8. Energy and sampling period *h* instantaneous variations for the second trajectory
(___Cycloidal profile, *-*-*- ALD trajectory)

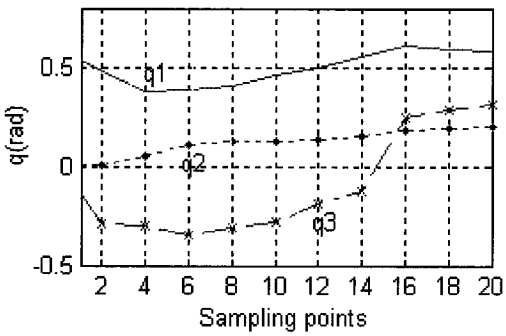


Fig. A. 9. Instantaneous variations of joint angles q_j, $j = 1,2,3$ for the circle trajectory

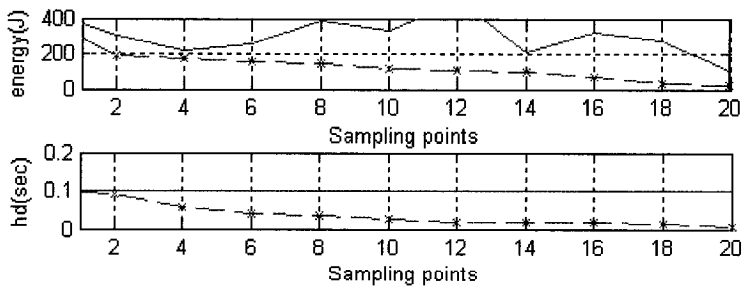
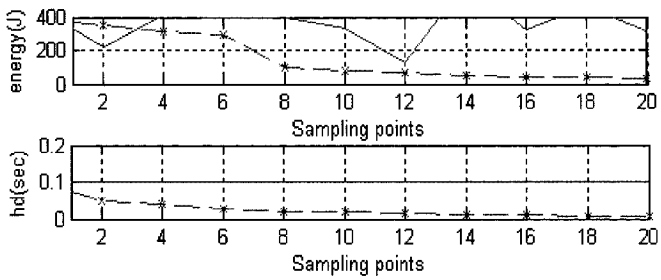
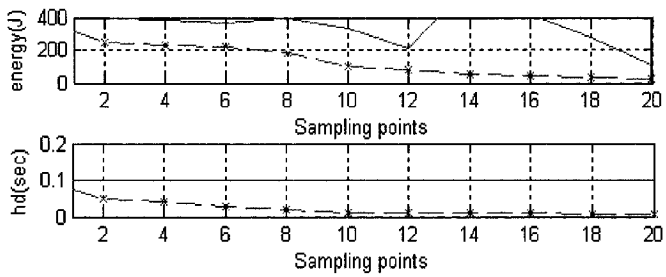


Fig. A. 10. Instantaneous variations of consumed energy and sampling time for the circle trajectory
(___Cycloidal profile, *-*-*- ALD trajectory)



(a)



(b)

Fig. A. 11. Instantaneous variations of consumed energy and sampling time for the first example (a)Trajectory with third link mass 2 Kg (b) Trajectory with increased third link mass by 1.5 Kg (___Cycloidal profile, *-*-*- ALD trajectory)

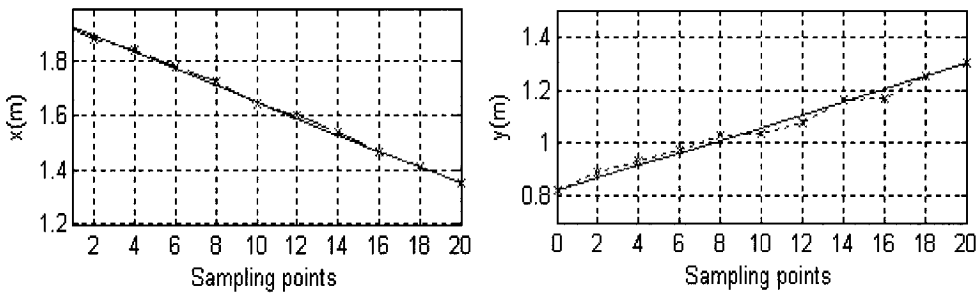


Fig. A. 12. EE trajectory along axis x and y displacement for example 1 with modified third link mass (___Cycloidal profile, *-*-*- ALD trajectory)

A. 8. References

[1] C. Ahrikencheikh and A. Seireg: "*Optimized-Motion Planning*", John Wiley & Sons, 1994.

[2] A. Khoukhi: "An Optimal Time-Energy Control Design for a Prototype Educational Robot ", *Robotica* v. 20, p 661-671, 2002.

[3] A. Khoukhi, L. Baron, M. Balazinski: "A Decoupled Approach to Time-Energy Trajectory Planning of Parallel Kinematic Machines", *Proc. of 16th CISM-IFFToMM Symposium, Robot Design, Dynamics and Control, ROMANSY'2006,* p 179-186, Warsaw, Poland, June 20-24, 2006.

[4] H. Jurgen, Mc. John: "Determination of Minimum-Time Maneuvers for a Robotic Manipulator Using Numerical Optimization Methods", *Mech. of Struct. and Mach.*, 27, (2), p 185-201, 1999.

[5] O. Dahl: "Path-Constrained Robot Control with Limited Torques-Experimental Evaluation", *IEEE Trans. Robot. Automat.*, v. 10, p 658-669, Oct.1994.

[6] Y. Nakamura: "*Advanced Robotics: Redundancy and Optimization*", Addison-Wesley, 1991.

[7] A. Liegeois: "Automatic Supervisory Control of Configurations and Behavior of Multibody Mechanisms", *IEEE Trans. Syst., Man, Cybern.*, v SMC-7, p 868–871, Dec. 1977.

[8] C. R. Carignan: "Trajectory Optimization for Kinematically Redundant Arms", *Journal of Robotic Systems*, 8(2), p 221 - 248, 1991.

[9] S. Chiaverini: "Singularity-Robust Task-Priority Redundancy Resolution for Real-Time Kinematic Control of Robot Manipulators", *IEEE Trans. On Robotics & Automation,* 13, (3), p 398 – 410, 1997.

[10] T. Rockafellar: "Lagrange Multipliers and optimality", *SIAM Review* 35, p 183-238, 1993.

[11] E. Polak: "*Optimization, Algorithms and Consistent Approximation*", Springer, N.Y, 1997.

[12] J. J. Craig: "*Introduction to Robotics Mechanics and Control*", Pearson Prentice Hall, Pearson Education, Inc., Upper Saddle River, NJ 07458, 2005.

[13] A. Isidori: "*Non linear control systems*", Springer; 3rd Edition, London, UK, 1995.

[14] A. Khoukhi, K. Demirli, L. Baron and M. Balazinski: "Hierarchical Neuro-Fuzzy Near-Minimum Time Trajectory Planning of a Serial Planar Robot", in revision, *Engineering Applications of Artificial Intelligence,* 2007.

ANNEXE B

Cet article a été présenté à la conférence: ROMANSY 16 Robot Design, Dynamics and Control, Proceedings of 16th CISM-IFFToMM Symposium, Warsaw, Poland, June 20-24, 2006. Il a fait également l'objet d'un chapitre du livre: CISM Courses and Lectures No. 487, T. Zielinska, C. Zielinski Eds, p 179-186, Springer Wien New York, 2006, *pages: 179-186.*

A DECOUPLED APPROACH TO OPTIMAL TIME ENERGY TRAJECTORY PLANNING OF PARALLEL KINEMATIC MACHINES

Amar KHOUKHI, Luc BARON, Marek BALAZINSKI
Department of Mechanical Engineering,
École Polytechnique de Montréal,
C. P. 6079, Succ. CV, Montreal, QC, Canada H3C 3A7
(amar.khoukhi, luc.baron, marek.balazinski)@polymtl.ca

B. 1. Abstract. This paper deals with a new optimal dynamic trajectory planning approach for Parallel Kinematic Machines (PKM) under task, workspace, and manipulator constraints. From kinematic, and dynamic models based on Euler-Lagrange formalism, we develop a multi-objective trajectory planning, optimizing electrical energy and robot traveling time and satisfying several technological constraints such as task requirements, actuator, link length, and workspace

limitations. A new decoupled formulation is considered in order to cope with the difficulty of dynamic parameters computation. A systematic implementation procedure is then given along with simulation results.

B. 2. Introduction

The path planning problem of parallel manipulators is receiving an increasing interest by the robotics community. Bhattacharya *et al.* (1998)[1] had developed a scheme for avoiding singularities of PKM by restructuring a pre-planned path in the vicinity of a singularity. In (Sen *et al.*, 2003)[2], a variational approach is reported for planning a singularity-free minimum-energy path between two end-points for Gough-Stewart platforms. These trajectory planners are however essentially kinematic based schemes, while others consider only static effects. For weakly coupled mechanical structures where static forces dominate over inertial ones, and tasks requiring low velocities, such kinematic approaches like the inverse rate control algorithm (Craig, 2005)[3] can assure satisfactory results. But for strongly non-linear, coupled mechanical structures, and for high velocity and precision tasks, a proved fact for serial manipulators is that these approaches fail to give acceptable results (Vukobratovic *et al.,* 1985)[4], (Khoukhi, 2002)[5]. When the linear speed goes over 1.5m/s at the end-effector, good performance can not be achieved without taking into

account of the Coriolis and centrifugal forces. So, how do these dynamic parameters affect the performance of parallel robots?

In this paper, an integrated optimal control system is designed for PKM dynamic trajectory planning. From a state space representation by a system of differential equations, the trajectory planning is solved with an optimization procedure by applying optimal control techniques (Khouhki *et al.,* 2005)[6]. The robot traveling time separating two sampling periods is optimized as well as the actuator torques. This multi-objective aspect allows achieving smooth trajectories satisfying several technological constraints such as actuator, link length, task, and workspace limitations. The resulting constrained multi-objective optimal control problem is solved through the augmented Lagrangean optimization technique (Polak, 1997)[7]. A new decoupling approach is introduced here in order to handle some inextricable computation issues arising in the original non-decoupled formulation. Section 2 introduces the kinematic and dynamic models of the PKM. Section 3, gives the discrete time dynamic model. In section 4, the optimal control problem is formulated in a non-linear programming framework. Section 5 develops the new decoupled formulation to the control problem, and in section 6 the Augmented Lagrangean technique is used to solve the resulting linear and decoupled problem. In section 7, preliminary simulation results are reported. Finally, section 8 concludes this paper.

B. 3. Kinematic and Dynamic Models

The kinematic and dynamic models are briefly introduced here. The reader is referred to (Baron *et al.*, 2000)[8], and (Harib *et al.,* 1997)[9] for the details. From Fig. B.1, the closure equations are:

$$a_i = x + {}^{W}R_P \, {}^{P}a_i \tag{1}$$

while the inverse rate kinematic model is given by:

$$\dot{l} = J^{-1}\dot{q} \tag{2}$$

where ${}^{W}R_P$ is the rotation matrix describing the orientation of the frame ***P*** with respect to the base frame ***W***, J is the robot Jacobian, l represents the joint coordinates and $q(t) = (x(t), y(t), z(t), \varphi(t), \theta(t), \psi(t))$ constitutes the Cartesian coordinates of ***P***, and the Euler angles describing the orientation of the platform with respect to the frame ***W***.

The dynamic model of the PKM is derived using Euler-Lagrange formalism (Angeles 2003)[10], (Harib *et al.,* 1997)[9]. In the operation space this dynamic equation is given in a canonical form as:

B. 4. Discrete Time Operation Space Dynamic Model

To develop the discrete-time operation space dynamic model, we introduce the variables:

$$X(t) = (X_1(t), X_2(t))^T \text{ , with } X_1(t) = q(t) \text{ , } X_2(t) = \dot{q}(t) \tag{4}$$

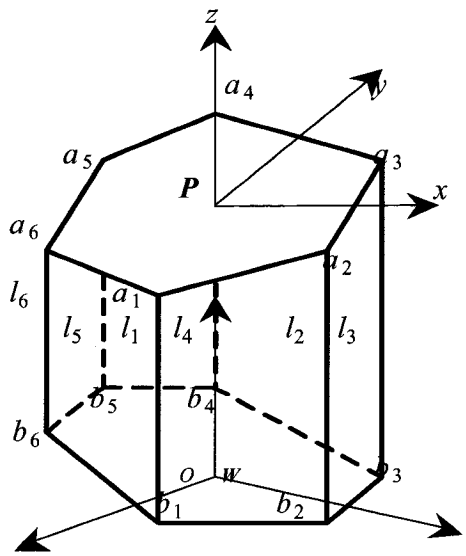


Fig. B.1. Geometry of a PKM

$$\tau = M(q)\ddot{q} + N(q,\dot{q}) + G(q) \tag{3}$$

Hence, the discrete-time state-space dynamic model can be written in the following form (Khoukhi, 2002)[5]:

$$[X_{k+1}] = \begin{bmatrix} I_{6\times6} & h_k I_{6\times6} \\ O_{6\times6} & I_{6\times6} \end{bmatrix} [X_k] - \begin{bmatrix} \frac{h_k^2}{2} I_{6\times6} \\ h_k I_{6\times6} \end{bmatrix} \left[M^{-1}(X_{1k}) \left[N(X_{1k}, X_{2k}) + G(X_{1k}) \right] \right] + \begin{bmatrix} \frac{h_k^2}{2} I_{6\times6} \\ h_k I_{6\times6} \end{bmatrix} \left[M^{-1}(X_{1k}) \right] [\tau]_k \tag{5}$$

B. 5. Optimal Time-Energy Trajectory Planning Problem

The performance index relates time and energy consumption. The approach used to solve for time minimization consists to fix the number of discretisations N

and vary the sampling periods h_k. The discrete time optimal control problem consists in finding the optimal sequences $((\tau_0^1, \tau_0^2,..., \tau_0^6), (\tau_1^1, \tau_1^2,..., \tau_1^6),..., (\tau_{N1}^1, \tau_{N1}^2,..., \tau_{N1}^6))$ and $(h_0, h_1,..., h_{N-1})$ allowing the robot to move from an initial state $X_0 = X_S$ to a given final state $X_N = X_T$, while minimizing the cost:

$$C_d = \underset{\substack{\tau \in \Re^{6\times N} \\ h \in \Re^{N+}}}{\text{Min}} \left\{ \sum_{k=0}^{N-1} [(\tau_k R \tau_k^T + \iota + X_{2k} Q X_2^T) h_k] \right\} \tag{6}$$

where $R, Q,$ and ι are respectively electric, and kinetic energy and time level-headnesses.

This minimization is made subject to dynamic eq. (5) and next equality and inequality constraints:

- Initial and target state attainability:

$$X_0 = X_S, \; X_N = X_T \tag{7}$$

- Link intermediate length limits:

$$l_{Min}^i \le l_k^i \le l_{Max}^i \; i = 1,...,6, \text{ and } k = 1,2,...,N, \text{ with } l_{iMax}^i = \sqrt{L_{iMax} \cdot L_{iMax}} = \Theta_{iMax}(X) \tag{8}$$

- Torque limits:

The robot torques is assumed to belong to a compact and bounded set $D_{\tau ad} \subset R^{2N}$:

$$D_{\tau_{ad}} = \left\{ \tau_{k,i}, \text{ such that : } \tau_{\min,i} \le \tau_{k,i} \le \tau_{\max,i} \; , i \in \{1,2,...,6\}, k = 0,...,N-1 \right\} \tag{9}$$

- Sampling period limits:

If the total traveling time T is too small, there may be no admissible solution to the optimal control problem. On the other hand, h_k must be smaller than the system smallest time constant in order to prevent the system from being uncontrollable between two control times. A tradeoff is made through bounding the sampling periods within an admissible domain such that:

$$D_{had} = \left\{ h_k \text{, such that: } h_{\min} \le h_k \le h_{\max} \right\} \tag{10}$$

B. 6. Constrained Linear-Decoupled Formulation

A major computational difficulty in the right hand side of equation (5) is that it contains the inverse of the total inertia matrix of the moving platform and the struts as well as their Coriolis and centrifugal wrenches. These matrices might take several pages long to display. By performing with the original non linear formulation, when one computes the co-states λ_k associated to the dynamic states X_k, he or she will have to perform the derivatives of the inverse of this total inertia matrix with respect to state components. The difficulty of these calculations has motivated a linear-decoupled formulation similar to that used in decoupled non linear control systems (Isidori 1995)[11]. This formulation reduces drastically the computations, by alleviating us performing the mentioned hard calculations.

Theorem:

Under the invertibility condition of the inertia matrix, the control law defined as:

$$\tau = M(q)v + N(q, \dot{q}) + G(q) \tag{11}$$

allows the robot to have a linear and decoupled behavior as:

$$\ddot{q} = v \tag{12}$$

where v is an auxiliary input.

Proof: See for instance (Isidori 1995)[11].

This brings the robot to have the decoupled and linear behavior described by the following simple dynamic equation written in discrete form as:

$$X_{k+1} = A(h_k)X_k + B(h_k)v_k = f_{d_k}^{D}(X_k, v_k, h_k) \tag{13}$$

with $$f_{d_k}^{D}(X_k, v_k, h_k) = \begin{bmatrix} I_{6\times6} & h_k I_{6\times6} \\ O_{6\times6} & I_{6\times6} \end{bmatrix} [X_k] + \begin{bmatrix} \frac{h_k^2}{2} I_{6\times6} \\ h_k I_{6\times6} \end{bmatrix} [v_k]_{6\times1}$$

Note that the control inputs for the decoupled problem become the acceleration and sampling period variables.

B. 7. Augmented Lagrangian for the Decoupled Formulation

The augmented Lagrangian associated to the decoupled formulation is written as:

$$L_{\mu}^{D}(X, v, h, \lambda, \alpha, \beta, \gamma, \delta, \rho, \zeta, \varsigma) = \sum_{k=0}^{N-1}\Big\{[[(M(X_{1k})v_k + N(X_{1k}, X_{2k}) + G(X_{1k}))^T R$$

$$(M(X_{1k})v_k + N(X_{1k}, X_{2k}) + G(X_{1k}))] + X_{2k}^T Q X_{2k} + \iota]h_k\Big\} + \sum_{k=0}^{N-1}\Big\{\lambda_{k+1}^T (X_{k+1} - f_{d_k}^{D}(X_k, v_k, h_k))\Big\} +$$

$$\sum_{k=0}^{N-1} h_k\Big\{\Phi_{\mu}(\alpha_k, X_k - X_{Max}) + \Phi_{\mu}(\beta_k, X_{Min} - X_k)\Big\} + \sum_{k=0}^{N-1} h_k\Big\{\Phi_{\mu}(\delta_k, \tau_{Min} - (M(X_{1k})v_k + N(X_{1k}, X_{2k}) + G(X_{1k}))\Big\} +$$

$$\sum_{k=0}^{N-1} h_k\Big\{\Phi_{\mu}(\gamma_k, (M(X_{1k})v_k + N(X_{1k}, X_{2k}) + G(X_{1k}) - \tau_{Max}^{j}))\Big\} +$$

$$\sum_{k=0}^{N-1} h_k\Big\{\Phi_{\mu}(\rho_k, \Theta(X_k) - l_{Max}) + \Phi_{\mu}(\zeta_k, l_{Min} - \Theta(X_k))\Big\} + h_N\{\Psi_{\mu}(\varsigma_N, X_{1N} - X_{1T})\} \tag{14}$$

where the function $f_{d_k}^{D}(X_k, \tau_k, h_k)$ is defined by the state equation (13), N is the total number of sampling points, $\alpha, \beta \in \Re^{12N+}$, $\gamma, \delta, \rho, \zeta \in \Re^{6N+}$, $\varsigma \in \Re^{N+}$ are the Lagrange multipliers. The vector $\lambda \in R^{12N}$ designates the co-states. Ψ_{μ} and Φ_{μ} are penalty functions associated to equality and inequality constraints. For the control sequences, $v_0, \ldots, v_{N-1}$, $h_0, \ldots, h_{N-1}$ and the corresponding trajectory states $X_0, \ldots, X_N$ to be optimal solution to the problem, there must exist some positive Lagrange multipliers

$$P_k = (\alpha_k, \beta_k, \gamma_k, \delta_k, \rho_k, \zeta_k, \varsigma_k),\ k = 0, \ldots, N, \tag{15}$$

and a finite positive penalty coefficient μ satisfying the first order Karush-Kuhn-Tucker (KKT) optimality necessary conditions. The development of these conditions and various related expressions and gradients are quite long and not given here. For more details, the reader is referred to (Khoukhi *et al.,* 2006)[12].

A systematic procedure for the augmented Lagrangian implementation appears in the Appendix, where notations $\Xi(X)$ and $\mathrm{I}(X)$ had been generically used to define equality and inequality constraints. Also, the Lagrange multipliers are divided into two sets *P1* and *P2* corresponding respectively to equality and inequality constraints. In this procedure, after selecting robot parameters, task definition, (such as starting and target poses), workspace limitations and simulation parameters, the kinematic unit defines a feasible solution satisfying initial and target poses. Then the inner optimization including energy and time minimization units give a dynamic solution, from which the states are computed, and tested. If feasible, a further convergence test is made. If convergence holds, then the program stops and displays the optimal solution. Otherwise, if the feasibility test fails, restart inner optimization unit. Else, if the feasibility test succeeds, go further to the dual part of the augmented Lagrangian and test for constraints satisfaction and update penalty and tolerance parameters. Then re-iterate until the number of permitted iterations is reached.

B. 8. Simulation Results

A simulation program has been implemented in Matlab to test the proposed optimal trajectory planning approach on a case study parallel robot reported in (Liu *et al* 2004)[13], and preliminary results are encouraging. This robot consists of two degrees of freedom planar parallel manipulator. A schematic of the manipulator is given in Figure 2. The following numeric values of the parameters are used: The number of sampling points is 10. The radius is $r=75mm$, $R=1203.0mm$ and $L=1972.5mm$. The mass of the platform is $m_p(kg)$=200, that of each leg is $m_l(kg)$=575, and that of the slider is $m_s(kg)$=60. The maximum velocity is $V_{\max}$=200 (mm/s). The workspace is defined as: ($x\in[-800mm, 800mm]$ and $y\in[-1718.14mm,-718.14mm]$). As for the augmented Lagrangian, the following parameter values had been taken ω_s=0.5, η_s=0.5, $\alpha_w=\alpha_\eta$=0.4, γ_1=0.25, γ_2=1.2, $w^*=10^{-4}$, υ=0.01, $\eta^*=\eta_1^*=10^{-3}$, $\beta_w=\beta_\eta$=0.4, υ=0.01, $\bar{\upsilon}$=0.3.

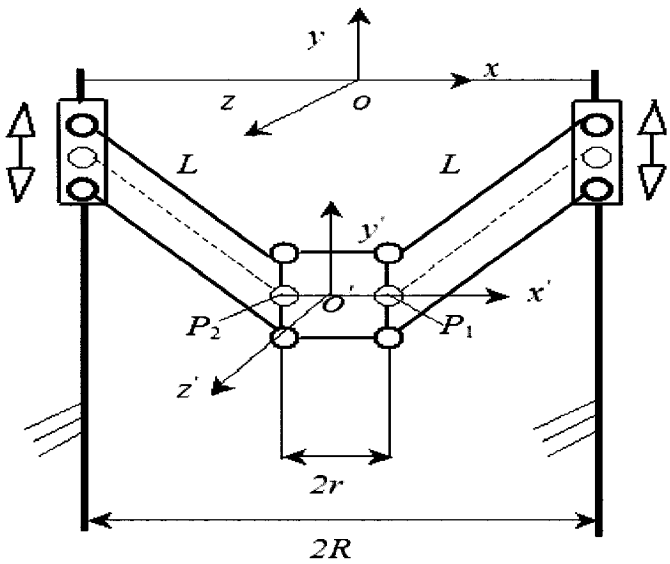


Fig. B. 2. A geometric representation of the planar parallel manipulator

The time and energy level-headedness are $R = \mathrm{I}_{6\times 6}$ and $t = 1$. The initial Lagrange multipliers $P_0^T = (\alpha_0, \beta_0, \gamma_0, \delta_0, \rho_0, \varsigma_0, \upsilon_0)$ are set to zero. The trajectory consists of a straight line from an initial state position x_0=–0.48, y_0=–1.48 to a target position x_F=0.15, y_F=–0.40 (all in meters). Note that at this level of simulation, the constraints (8) mentioned in section 4 is not considered. Figures 3 and 4 show the simulation results for both initial kinematic solution and augmented Lagrangian solution. The first plots from the top shows the 2-dimension linear displacement of the platform. The second plots show the consumed time to achieve the trajectory. The third and fourth plots show the instantaneous variations of the joint torques. While the last ones show the consumed energy. One notes that although the initial solution is

cinematically feasible, when the corresponding torques is computed considering the dynamic model and forces, one gets several times torque values outside the admissible domain resulting in high values for the cost function. However, with the augmented lagrangian, the variations of the energy consumption are increasing smoothly and monotonously. Finally, in terms of CPU processing time, for seven dual iterations and four inner iterations performed to find a minimum time-energy trajectory, with a precision of $0.5x10^{-3}$ in target attainability, the total CPU time is of order of 75sec.

B. 9. Conclusions

In this paper, a new design of optimal multi-objective trajectory planning is presented. This design involves several criteria and constraints related to a global performance index integrating the interactions of the robot kinematics, dynamics, task, and environment requirements. A decoupling formulation had been also considered in order to tackle several inextricable computational issues. The augmented Lagrangian algorithm had been used to solve the resulting non linear and non convex optimal control problem. A major advantage of this approach is that one can introduce any type of constraints related to robot, task or environment, like actuators and friction models, obstacles and singularity avoidance by deriving the corresponding constraint expressions and adding them naturally in the Lagrangian in order to have them included in the trajectory planning system. Some of these topics are being developed in an ongoing work.

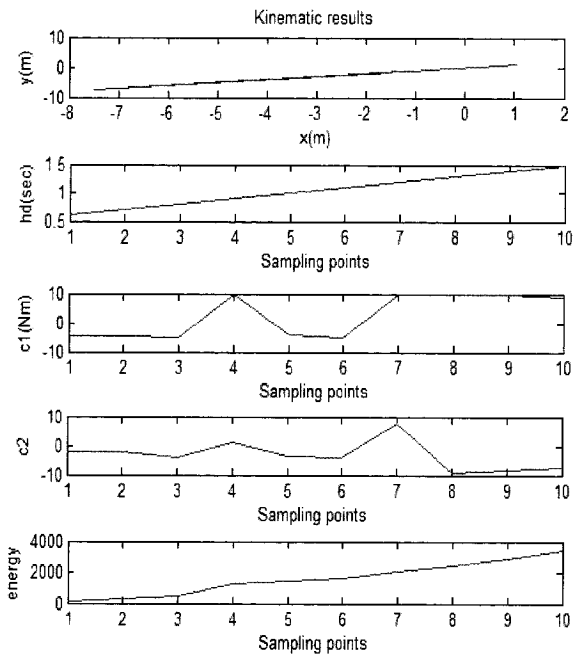


Fig. B.3. kinematic simulation results

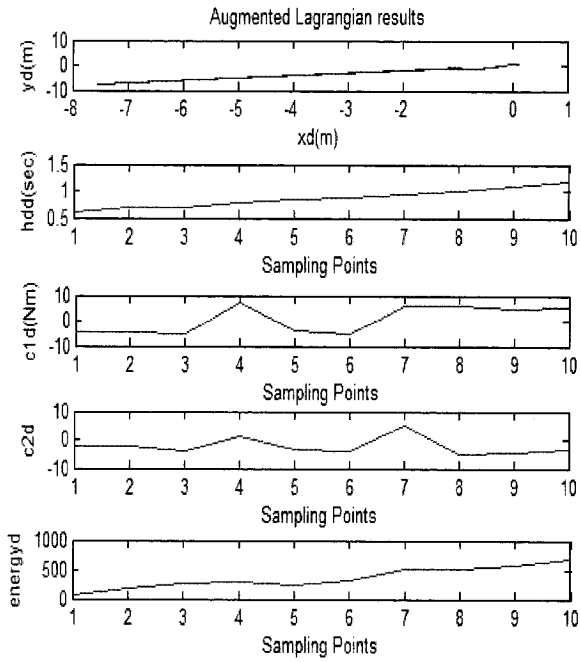


Fig.B. 4. Simulation results using the augmented Lagrangian

Acknowledgment: The authors would like to gratefully thank the Natural Science and Engineering Research Council of Canada for supporting the present work.

B. 10. Bibliography

[1] S. Bhattacharya, H. Hatwal, A. Ghosh (1998). Comparison of an Exact and Approximate Method of Singularity Avoidance in Platform Type Parallel Manipulators. Mech. Mach. Theory 33 (7): 965-974.

[2] S. Sen, B. Dasgupta, A. K. Mallik. (2003). A Variational Approach for Singularity-Free Path-Planning of Parallel Manipulators. Mec. and Machine Theory 38: 1165-1183.

[3] J. J. Craig (2005). Introduction to Robotics: Mechanics and Control. 3rd Ed, Addison-Wesley.

[4] M. Vukobratovic, D. Stokic, (1983). Is Dynamic Control Needed in Robotics System? and If So to What Extent? ", Int'l Jou. Of Robotics Research 2(2): 18-35.

[5] A. Khoukhi (2002). An Optimal Time-energy Control Design for a Prototype Educational Robot. Robotica 20: 661-671.

[6] A. Khoukhi, L. Baron, M. Balazinski (2005). Programmation Dynamique Time-Énergie Minimum des Robots Parallèles par Lagrangien Augmenté. CCToMM SM3, May 26-27, Canadian Space Agency, Saint Hubert QC, Canada, : 147-148.

[7] E. Polak. (1997). Optimization, Algorithms and Consistent Approximation. Springer-Verlag, New York.

[8] L. Baron and J. Angeles (2000). The Kinematic Decoupling of Parallel Manipulators Using Joint-Sensor Data, *IEEE Transactions on Robotics and Automation* 16(6): 644-651.

[9] K. Harib, K. Srinivasan. (1997). Kinematic and Dynamic Analysis of Stewart Platform-Based Machine Tool Structures. Robotica 21: 541-551.

[10] J. Angeles (2003). Fundamentals of Robotic Mechanical Systems, Theory, Methods and Algorithms. Second Ed., Springer Verlag.

[11] A. Isidori. Non Linear Control Systems, Springer; 3 edition

[12] A. Khoukhi, L. Baron, M. Balazinski (2006). Multi-Objective Off-line Programming of Parallel Kinematic Machines, Submitted to Robotics and Computer-Integrated Manufacturing.

[13] X. J. Liu, Q. M. Wang, J. Wang (2004). Kinematics, Dynamics and Dimentional Synthesis of a Novel 2- DoF Translational Manipulator", Journal of Intelligent and Robotic Systems, 41, 205-224, 2004.

ANNEXE C

RAPPELS SUR L'OPTIMISATION NON LINÉAIRE AVEC CONTRAINTES

Considérons le problème d'optimisation non linéaire avec contraintes suivant :

$$(P): \quad \begin{cases} \min f(x) \\ \text{avec} \\ g_i(x) \le 0 \;\; i \in I = \{1,2,...,m\} \\ x \in \Re^n \end{cases}$$

Où toutes les fonctions f et $g_i \,(i \in I)$ sont supposées continues et différentiables, et l'ensemble des solutions $X = \left\{x \in \Re^n / g_i(x) \le 0, \forall i \in I\right\}$ est supposé non vide. Nous examinons dans cet annexe les conditions nécessaires et suffisantes d'optimalité de ce problème, et passerons en revues les méthodes indirectes (ou duales) de sa résolution.

C. 1. Conditions nécessaires d'optimalité

Les conditions nécessaires d'optimalité du problème (P) sont étroitement liées avec l'hypothése de qualification des contraintes (QC) [1-4]. Nous rappelons dans ce paragraphe d'abord la notion de (QC) et ensuite les conditions necessaires d'optimalité de Kuhn et Tucker.

On dira que $x^0 \in X$ est un optimum local de (P) si $f(x)$ ne peut décroître lorsque x décrit un arc de courbe Γ (suffisamment régulier) partant de x^0 et contenu dans

l'ensemble X des solutions. Un tel arc de courbe Γ est dit admissible en x^0 et sera défini par une fonction $\phi(\Re^+ \to \Re^n)$ continûment différentiable du paramètre $\theta \geq 0$: $\phi(\theta) = \left[\phi_1(\theta), \phi_2(\theta), ..., \phi_n(\theta)\right]$ telle que :

(a) $\phi(0) = x^0$,

(b) pour $\theta > 0$ suffisamment petit, $\phi(\theta) \in X$

On appellera direction admissible en x^0 tout vecteur

$$y = \frac{d\phi}{d\theta}(0) = \left[\frac{d\phi_1}{d\theta}(0), \frac{d\phi_2}{d\theta}(0), ..., \frac{d\phi_n}{d\theta}(0)\right]^T \tag{1}$$

tangent à un arc de courbe $\phi(\theta)$ admissible en x^0, et on notera C_{ad} le cône formé par l'ensemble des directions admissible en x^0, I^0 l'ensemble des indices des contraintes saturées x^0, c'est-à-dire

$$I^0 = \left\{ i \in I / g_i(x^0) = 0 \right\}, \tag{2}$$

et on définie le cône G défini par:

$$G = \left\{ y / \nabla g_i^T(x^0) \cdot y \leq 0 \quad (\forall i \in I^0) \right\} \tag{3}$$

Une condition nécessaire pour que $y \in \Re^n$ vérifie $y \in C_{ad}$ est:

$\nabla g_i^T(x^0) \cdot y \leq 0 \quad \forall i \in I^0$, autrement dit : $C_{ad} \subset G$. Cependant, toutes les directions y vérifiant cette relation ne sont pas nécessairement des directions admissibles.

L'hypothése de qualification des contraintes consiste donc à supposer que les conditions $\nabla g_i^T(x^0)\cdot y \le 0 \quad (\forall i \in I^0)$ sont nécessaires et suffisantes pour que y soit une direction admissible en x^0 (ou limite d'une suite convergente de directions admissibles en x^0), soit $cl(C_{ad}) = G$, avec cl designe la fermeture de l'ensemble C_{ad}.

C. 1. 1. Qualification des contraintes

La vérification directe de l'hypothése (QC) est difficile en pratique. Les résultats suivants fournissent les conditions suffisantes pour que (QC) soit réalisée :

Pour que (QC) soit vérifiée en tout point $x \in X$, il suffit que l'une des conditions (a) ou (b) soit réalisé :

(a) Toutes les fonctions g_i sont linéaires;

(b) Toutes les fonctions g_i sont convexes et X a un intérieur non vide.

Pour que (QC) soit vérifiée en un point $x^0 \in X$, il suffit que

(c) Les gradients $\nabla g_i(x^0)\,(i \in I^0)$ des contraintes saturées en x^0 sont linéairement indépendants (pour g_i linéaire ou non linéaire).

C. 1. 2. Conditions nécessaires d'optimalité (*Kuhn et Tucker 1951*, [1-4]):

Sous l'hypothése de qualification des contraintes, une condition nécessaire pour que

x^0 soit un optimum local du problème (P) est qu'il existe des nombres $\lambda_i \geq 0 \ \ (i \in I)$ appelés multiplicateurs de Kuhn et Tucker, tel que :

$$\nabla f(x^0) + \sum_{i \in I} \lambda_i \nabla g_i(x^0) = 0 \tag{4}$$

$$\lambda_i g_i(x^0) = 0 \quad (i \in I) \tag{5}$$

Une interprétation géométrique de ce résultat est illustrée sur la Figure C. 1. Dans le cas où il existe aussi des contraintes égalités :

$$h_l(x) = 0, \qquad l \in L = \{1,2,...,p\} \tag{6}$$

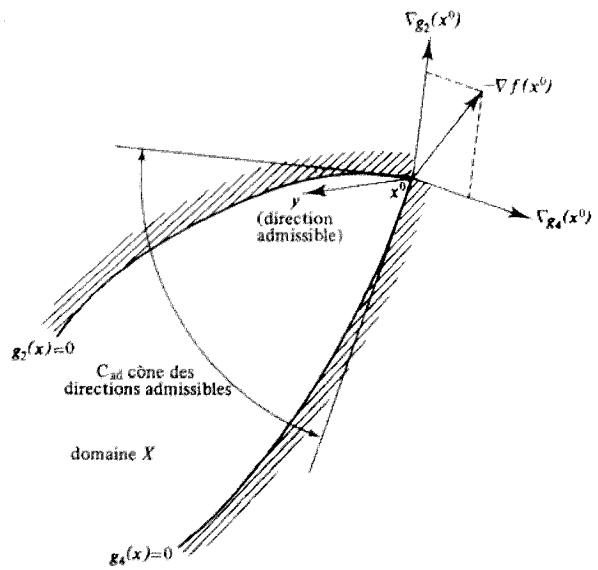


Figure C. 1. Illustration géométrique des conditions d'optimlité de Kuhn-Tucker [2]. Sous (QC) des contraintes, l'ensemble des directions admissibles forme un cone C_{ad}, intersection des $|I^0|$ demi espaces d'équations $\nabla g_i(x^0).y \leq 0, \ \ (\forall i \in I^0)$. Pour que x^0 soit un optimum local de f il faut que le vecteur $-\nabla f(x^0)$ fasse un angle obtus avec chaque direction admissible y. On peut vérifier géométriquement que $-\nabla f(x^0)$ doit s'exprimer comme une combinaison linéaire à coefficient λ_i positifs des $\nabla g_i(x^0)$, pour $i \in I^0$.

En recrivant l'hypothése de qualification de contraintes : $cl(C_{ad}) = G_1$ avec

$$G_1 = \left\{ y / y \in \Re^n : \nabla g_i^T(x^0) \cdot y \leq 0 \, (i \in I^0) : \nabla h_l^T(x^0) \cdot y = 0 \, (l \in L) \right\} \quad (7)$$

et

$$I^0 = \left\{ i / g_i \, (x^0) = 0 \right\} \quad (8)$$

Une condition nécessaire pour que x^0 soit un optimum local du problème est alors qu'il existe des nombres $\lambda_i \geq 0 \, (i \in I)$ et $\mu_l \, (l \in L)$ (μ_l de signe quelconque) tels que :

$$\nabla f(x^0) + \sum_{i \in I} \lambda_i \nabla g_i \, (x^0) + \sum_{l \in L} \mu_l \nabla h_l \, (x^0) = 0 \quad (9)$$

$$\lambda_i g_i \, (x^0) = 0 \quad (i \in I) \quad (10)$$

C. 2. Conditions suffisantes d'optimalité (points-cols et fonction de Lagrange)

D'abord, comme pour les conditions nécessaires, les principales conditions suffisantes connues pour que (QC) soit vérifiée pour le problème avec contraintes égalités et inégalités sont:

(a) Les fonctions $g_i \, (i \in I)$ sont convexes, les fonctions $h_l \, (l \in L)$ sont linéaires et X est non vide;

(b) En $x^0 \in X$ les gradients $\nabla g_i(x^0)$ $(\forall i \in I^0)$ et $\nabla h_l(x^0)(\forall l \in L^0)$ sont linéairement indépendants.

Considérons maintenant les conditions suffisantes d'optimalité du problème (P). Si l'on associe à chaque contrainte $i \in I$ un nombre réel $\lambda \geq 0$ appelé multiplicateur de Lagrange, la fonction de Lagrange associée au problème (P) peut être définie par :

$$L(x,\lambda) = f(x) + \sum_{i \in I} \lambda_i g_i(x) \tag{11}$$

On dit que $(\bar{x},\bar{\lambda})$ est un point-col de $L(x,\lambda)$ si :

$$L(\bar{x},\bar{\lambda}) \leq L(x,\bar{\lambda}) \quad \forall x \in X \tag{12}$$

$$L(\bar{x},\lambda) \leq L(\bar{x},\bar{\lambda}) \quad \forall \lambda \geq 0 \tag{13}$$

C'est-à-dire, si $\bar{x}$ minimise $L(x,\bar{\lambda})$ sur X et $\bar{\lambda}$ maximise $L(\bar{x},\lambda)$ sur $\Re^+$

Soit $\bar{x} \in X$ et $\bar{\lambda} \geq 0$: on démontre que $(\bar{x},\bar{\lambda})$ est un point-col pour $L(x,\lambda)$ si et seulement si :

$$\begin{cases} L(\bar{x},\bar{\lambda}) = \min\limits_{x \in X} L(x,\bar{\lambda}) \\ g_i(\bar{x}) \leq 0 \qquad (\forall i \in I) \\ \bar{\lambda}_i g_i(\bar{x}) = 0 \qquad (\forall i \in I) \end{cases} \tag{14}$$

et si $(\bar{x},\bar{\lambda})$ est un point-col de $L(x,\lambda)$ alors $\bar{x}$ est un optimum global du (P).

Ce résultat est très général et peut s'appliquer à n'importe quel programme mathématique convexe ou non convexe, f et g_i différentiables ou non, l'ensemble X continu ou discret, fini ou dénombrable. Cependant, pour les problèmes non convexes, il peut ne pas exister de point-col.

C. 3. Méthodes directes (primales) d'optimisation avec contraintes

Les méthodes primales d'optimisation non linéaires opérent directement sur le problème donné. Elles engendrent une séquence de solutions satisfaisant les contraintes en assurant une décroissance monotone de la fonction à minimiser. Parmis ces méthodes, on distingue les méthodes de programmation séquentielles quadratiques (SQP) et ses pluasieurs variantes (programmation quadratiques itératives, les méthodes à métrique variables). Ces algorithmes présentent un avantage de pouvoir fournir une solution approchée qui satisfait les contraintes lors de l'interruption du processus itératif. Cependant, il est généralement difficile d'obtenir une convergence globale de ces méthodes, et la mise au point d'un tel algorithme est parfois délicate [2-4].

C. 4. Méthodes indirectes (ou duales) :

C. 4. 1. Méthodes de pénalités

Nous rappelons ici les méthodes de pénalités dont les idées sont utilisées dans les méthodes duales. Le principe général des méthodes de pénalités est simple : pour le

problème de la minimisation d'une fonction $f(x)$ sous les contraintes $g_i(x) \leq 0, (i \in I = \{1,2,...,m\})$, on définit une fonction de pénalisation de la façon suivante :

$$\forall x: \qquad H(x) = \sum_{i=1}^{m} h(g_i(x)) \tag{15}$$

avec

$$h(g_i(x)) = \begin{cases} 0 & g_i(x) \leq 0 \\ +\infty & g_i(x) > 0 \end{cases} \tag{16}$$

et on résout le problème sans contrainte :

$$\min_{x \in \Re^n} \phi(x) = f(x) + H(x) \tag{17}$$

Il est évident que l'optimum de $\phi(x)$ ne peut pas être atteint en un point n'appartenant pas à l'ensemble de solution X ($\phi(x) = +\infty$ dans ce cas). D'autre part, pour tout $x \in X, H(x) = 0$ et par conséquent $\forall x \in X, \phi(x) = f(x)$. Cependant, la fonction de pénalisation H est, par construction, discontinue, et il en est de même de ϕ. On ne peut appliquer directement aucune des méthodes d'optimisation sans contrainte décrite précédemment.

Pour éviter cette difficulté, la méthode dite des pénalités extérieures remplace la fonction de pénalisation H par :

$$\forall x: \qquad H(x)=\sum_{i=1}^{m}[\max\{0,g_i(x)\}]^2 \tag{48}$$

et on résout à chaque itération le problème sans contrainte :

$$\min_{x\in\Re^n}\phi(x,r)=f(x)+rH(x) \tag{49}$$

Où $r>0$ est appelé le coefficient de pénalité. On voit que lorsque r est choisi suffisamment grand, le point $\bar{x}(r)$ obtenu par minimisation de $\phi(x,r)$ est proche de l'ensemble des solutions X parce que $H(\bar{x}(r))$ doit être suffisamment faible. Pour éviter des difficultés numériques, on peut commencer par choisir un coefficient de pénalité r de valeur pas trop élevée, et puis augmenter r itérativement pour obtenir une solution.

Dans la méthode précédente, l'optimum x^* est approché de l'extérieur de l'ensemble des solutions X. Dans la méthode dite de pénalité intérieures, l'optimum est approché par l'intérieur, c'est-à-dire qu'on définit une fonction de pénalisation intérieure par :

$$B(x)=-\sum_{i=1}^{m}g_i(x) \tag{50}$$

Cette fonction vérifie :

(a) $B(x)\geq 0\,.\ \forall x\in \text{int}(X)$

(a) $B(x)\to+\infty$ lorsque x tend vers la frontière de X ;

(b) Les fonctions $g_i\ (i=1,...,m)$ continues, et $B(x)$ continue sur l'intérieur de X

Considérons maintenant le problème sans contrainte :

$$\min_{x \in f(x)} \psi(x,t) = f(x) + tB(x) \tag{51}$$

où $t > 0$ est le coefficient de pénalité. Supposons que X est fermé, qu'il a un intérieur non vide, que tout $x \in X$ est limite d'une suite de points appartenant à l'intérieur de X, que f est une fonction continue, et que l'une des deux conditions suivantes est vérifiée :

(a) $f(x) \to +\infty$ quand $\|x\| \to +\infty$

(b) X est borné

Alors, lorsque le coefficient de pénalité t tend vers 0, on démontre que la suite de point $\bar{x}(t)$ obtenue sur le minimum de $\psi(x,t)$ admet au moins un point d'accumulation et tout point d'accumulation de la suite $\bar{x}(t)$ est un optimum (global) du problème de minimisation de $f(x)$, et que la quantité $t.B(\bar{x}(t))$ tend vers 0.

En supposant que l'on connaisse une solution de départ x^0 appartenant à l'intérieur de l'ensemble des solutions X, on peut donc utiliser une méthode d'optimisation sans contrainte et diminuer itérativement le coefficient de pénalité pour trouver une solution du problème.

Les méthodes de pénalités sont en général des méthodes simples et efficaces pour obtenir rapidement de bonnes solutions approchées d'un problème d'optimisation avec contraintes. Elles fournissent de bonnes approximations des multiplicateurs de

Kuhn et Tucker. Cependant, elles ne permettent généralement pas d'obtenir une précision élevée à cause du mauvais conditionnement des fonctions φ et ψ lorsque les coefficients de pénalités sont très grands (r) ou très petits (t).

C. 4. 2. Méthodes duales utilisant le Lagrangien ordinaire

Considérons encore le problème :

$$(P): \quad \begin{cases} \min f(x) \\ \text{avec} \\ g_i(x) \le 0 \qquad i \in I = \{1,2,...,m\} \\ x \in \Re^n \end{cases}$$

La fonction de Lagrange classique est définie de la façon suivante :

$$L(x,\lambda) = f(x) + \sum_{i \in I} \lambda_i g_i(x) \tag{52}$$

Nous avons vu que le problème (P) peut être résolu si nous pouvons déterminer un point col de la fonction de Lagrange, c'est-à-dire un couple $(\bar{x},\bar{\lambda})$ vérifiant:

$$\begin{cases} L(\bar{x},\bar{\lambda}) = \min_{x \in X} L(x,\bar{\lambda}) \\ g_i(\bar{x}) \le 0 \quad (\forall i \in I) \\ \bar{\lambda}_i g_i(\bar{x}) = 0 \quad (\forall i \in I) \end{cases} \tag{53}$$

Définissons maintenant pour $\lambda \ge 0$ une fonction $\omega(\lambda)$ dite une fonction duale :

$$\omega(\lambda) = \min_{x \in \Re^n} \{L(x,\lambda)\} \tag{54}$$

et notons le problème dual de (P) comme suit:

$$(D): \quad \begin{cases} \max\limits_{\lambda} \omega(\lambda) = \max\limits_{\lambda} \{\min\limits_{x \in X} L(x, \lambda)\} \\ \lambda \in \Re^{m+} \end{cases}$$

Il a été démontré que la recherche d'un point-col du problème (P), lorsqu'il existe, peut se faire précisément en résolvant le problème dual (D). Plus exactement, on a les propriétés suivantes du problème dual (D):

1. Pour tout $\lambda \in \Re^{m+}$, la valeur de la fonction dual $\omega(\lambda)$ est un minorant de l'optimum absolu $f(x)$ de (P), si $\omega(\lambda^*)$ est la valeur optimale du problème dual:

$$\forall \lambda \geq 0 : \omega(\lambda) \leq \omega(\lambda^*) \leq f(x^*) \tag{55}$$

2. La fonction duale ω est une fonction concave de λ, quelque soient les natures (convexité, continuité, différentiabilité) des fonctions f et g_i.

Théoréme de la dualité :

(a) Si le problème (P) admet un point-col (x^*, λ^*) alors, on a :

$$\max(D) = \omega(\lambda^*) = f(x^*) = \min(P) \tag{56}$$

Autrement dit, la valeur optimale du problème primal (P) est égale à la valeur optimale du problème dual (D).

(b) Réciproquement, s'il existe x^* solution de (P) et $\lambda^* \geq 0$ tels que : $\omega(\lambda^*) = f(x^*)$ alors (P) admet un point-col et (x^*, λ^*) est un tel point-col.

3. Pour $\overline{\lambda} \in \Re^{m+}$ soit :

$$Y(\overline{\lambda}) = \left\{ y \in \Re^n \Big/ f(y) + \sum_i \overline{\lambda}_i g_i(y) = \omega(\overline{\lambda}) \right\} \tag{57}$$

Alors, pour tout $y \in Y(\overline{\lambda})$, $g(y)$ est un gradient (ou sous-gradient) de ω en $\overline{\lambda}$.

4. Soit $\overline{\lambda} \geq 0$ une solution optimale du problème dual. Alors :

(a) Si le problème (P) admet un point-col, il existe x^* solution de (P) tel que (x^*, λ) soit un point-col (x^* est alors solution optimale de (P))

(b) Si le problème (P) admet un point-col et si $L(x, \overline{\lambda})$ admet un minimum unique en x, soit $\overline{x} \in \Re^n$, alors nécessairement $(\overline{x}, \overline{\lambda})$ est un point-col et $\overline{x}$ est solution optimale de (P).

(c) Si on suppose que ω est différentiable en $\overline{\lambda}$ et si $\overline{x}$ est (l'unique) minimum en x de $L(x, \overline{\lambda})$ alors (P) admet un point-col et $(\overline{x}, \overline{\lambda})$ est un tel point-col.

5. Pour $\lambda \geq 0$ quelconque, notons $\overline{x}(\lambda) \in Y(\lambda)$ un minimum en x de la fonction $L(x, \lambda)$. Alors $\overline{x}(\lambda)$ est optimum globale du problème perturbé :

$$(P'): \quad \begin{cases} \min f(x) \\ \text{sous les contraintes} \\ g_i(x) \le g_i(x(\lambda)) \quad i \in I = \{1,2,...,m\} \\ x \in \Re^n \end{cases}$$

Par conséquent, lorsqu'il existe un point-col (en particulier dans le cas convexe) et que le minimum en x de $L(x,\lambda^*)$ est unique pour $\lambda = \lambda^*$ optimum du dual, la résolution du problème dual permet d'obtenir une solution optimale du problème primal [2-4].

Dans le cas où il n'existe pas de point-col, la fonction duale ω n'est pas différentiable à l'optimum, et la résolution du problème dual permet d'obtenir d'une part, des solutions approchées du problème primale (P) et d'autre part, des minorants de la valeur optimale du problème primal .

Dans le cas où les fonctions f et g_i ne sont pas convexes, la fonction de Lagrange $L(x,\lambda)$ n'est pas convexe en x pour λ donné. Il est donc difficile d'obtenir le minimum absolu en x de $L(x,\lambda)$ et on doit se contenter d'un minimum local $\bar{x}$ ou un point-col local.

C. 4. 3. Méthodes duales utilisant le Lagrangien augmenté

Nous avons vu que les méthodes de pénalités présentent l'inconvénient d'avoir une fonction male conditionnée sur laquelle les méthodes de gradients ont une convergence lente. Les méthodes utilisant la dualité du Lagrangien classique ne

s'appliquent efficacement qu'au cas des fonctions convexes. Dans le cas des méthodes duales, si les variables duales λ convergent vers l'optimum λ^* du problème dual, les solutions du problème primal convergent vers une solution optimale si la fonction de Lagrange $L(x,\lambda)$ a un minimum unique en x dans le voisinage de λ^*.

Par opposition aux méthodes précédentes, les méthodes duales utilisant un Lagrangien augmenté permettent de surmonter en partie les difficultés ci-dessus pour les problèmes non convexes. Leur intérêt fondamental est l'extension des conditions nécessaires et suffisantes d'optimalité classiques à certaines classes de problèmes non convexes. D'autre part, elles peuvent être considérées comme une amélioration des méthodes de pénalités en évitant d'avoir à utiliser des coefficients de pénalité trop grands.

Nous introduisons d'abord la théorie des Lagrangiens généralisés [2, 5]. Considérons encore le problème (P) de minimiser $f(x)$ sous les contraintes $g_i(x) \le 0\,(i=1,...,m)$ et $x \in \Re^n$. On peut étendre la définition de la fonction de Lagrange ordinaire et la fonction objectif à $x \in \Re^n$ et $\lambda \in \Re^m$ de la façon suivante :

$$L(x,\lambda) = \begin{cases} f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) & \lambda \ge 0 \\ -\infty & \lambda < 0 \end{cases} \tag{58}$$

et

$$\overline{f}(x) = \begin{cases} f(x) & \text{si } g_i(x) \le 0,\ (\forall i \in I) \\ +\infty & \text{sinon} \end{cases} \tag{59}$$

On peut écrire alors la relation :

$$\forall x \in \Re^n : \max_{\lambda \in \Re^m} L(x,\lambda) = \overline{f}(x) \tag{60}$$

Puisque la minimisation de $\overline{f}(x)$ sur $\Re^n$ est équivalente avec la résolution de (P), on se ramène donc à la résolution du problème (appelé problème primal) :

$$\min_{x \in \Re^n} \max_{\lambda \in \Re^m} \{L(x,\lambda)\} \tag{61}$$

Posons maintenant en toute généralité un problème d'optimisation sous la forme :

$$(\overline{P}) \qquad \min_{x \in X} \overline{f}(x)$$

Où X est un espace vectoriel réel, généralement $X \equiv \Re^n$. On définit un Lagrangien généralisé $\overline{L}$ de $(\overline{P})$ comme suit :

Étant donné un espace vectoriel réel et une fonction réelle $\overline{L}(x,\lambda)$ sur $X \times \Lambda$, on dit que $\overline{L}(x,\lambda)$ est une représentation lagrangienne de $(\overline{P})$ si et seulement si :

(a) $\overline{L}(x,\lambda)$ est, pour tout x, une fonction concave fermée de λ ;

(b) $\forall x \in X, \overline{f}(x) = \max_{\lambda \in \Lambda} \overline{L}(x,\lambda)$

Les éléments de Λ sont appelés des multiplicateurs de Lagrange généralisés.

Pour chaque $\lambda \in \Lambda$ fixé, on a les relations :

$$\overline{f}(x) \geq \overline{L}(x,\lambda) \qquad (\forall x) \tag{62}$$

et

$$\min_{x \in X} \overline{f}(x) \geq \min_{x \in X} \{\overline{L}(x,\lambda)\} = \overline{\omega}(\lambda) \tag{63}$$

Où la fonction duale généralisée $\overline{\omega}(\lambda)$ est définie par :

$$\overline{\omega}(\lambda) = \min_{x \in X} \{\overline{L}(x, \lambda)\} \tag{64}$$

On retrouve donc la propriété de minorant de la fonction duale. D'autre part, $\overline{\omega}(\lambda)$ est le minimum d'une collection de fonctions concaves. On en déduit que $\overline{\omega}(\lambda)$ est une fonction concave de λ. On dit aussi que λ^* est un multiplicateur de point-col pour $(\overline{P})$ relativement à la fonction de Lagrange $\overline{L}$ si :

$$\min_{x \in X} \overline{f}(x) = \min_{x \in X} \{\overline{L}(x, \lambda^*)\} = \overline{\omega}(\lambda^*) \tag{65}$$

La détermination d'un multiplicateur de point-col λ^* peut se faire en résolvant le problème dual généralisé :

$$\max_{\lambda \in \Lambda} \overline{\omega}(\lambda) \tag{66}$$

Autrement dit, l'existence d'un point-col est une condition nécessaire et suffisante pour que l'on puisse permuter les opérateurs min et max.

La caractérisation des conditions d'existence de points-col peut s'effectuer également avec une représentation perturbationnelle généralisée de $(\overline{P})$:

Etant donné un espace vectoriel réel Y, une fonction réelle $\overline{F}(x, y)$ sur $X \times Y$ est appelée une représentation perturbationnelle de $(\overline{P})$ si et seulement si :

(a) $\overline{F}(x, y)$ est, pour tout x une fonction convexe fermée de y

(b) $\forall x \in X, \quad \overline{F}(x,0) = \overline{F}(x)$,

Le problème perturbé correspondant à un vecteur $y \in Y$ quelconque est par définition :

$$(\overline{P}) \qquad \min_{x \in X} \overline{F}(x, y)$$

Par exemple, on peut choisir pour le problème (P) : $\min f(x), g_i(x) \leq 0, x \in \Re^n$, la représentation perturbationnelle, correpondant à la représentation lagrangienne ordinaire, peut être donnée de la façon suivante :

$$F(x,y) = \begin{cases} f(x) & \text{si } g_i(x) \leq y_i, \ (\forall i \in I) \\ +\infty & \text{sinon} \end{cases} \tag{67}$$

Rockafeller dans [5] a prouvé une relation très étroite entre les représentations lagrangiennes et représentations perturbationnelles d'un même problème $(\overline{P})$, qui peut s'énoncer comme suit :

Théorème (*Rockafeller 1974*) :

Soient Y et Λ deux espaces vectoriels réels duaux l'un de l'autre (le produit de dualité est noté .) , alors :

1. Si $\overline{F}(x, y)$ est une représentation perturbationnelle de $(\overline{P})$ (avec $\overline{F}$ convexe fermé en y, alors on peut en déduire une représentation lagrangienne $\overline{L}(x, \lambda)$ par :

$$\overline{L}(x, \lambda) = \min_{y \in Y} \{\overline{F}(x, y) + \lambda . y\} \tag{68}$$

2. Inversement, si $\overline{L}(x, \lambda)$ est une représentation lagrangienne de $(\overline{P})$ (avec $\overline{L}$ concave fermée en λ) alors on peut en déduire une représentation perturbationnelle $\overline{F}(x, y)$ par :

$$\overline{F}(x,y) = \max_{\lambda \in \Lambda} \{\overline{L}(x,\lambda) - \lambda.y\} \tag{69}$$

On peut généraliser alors la notion de fonction de perturbation de la façon suivante : Étant donnée une représentation perturbationnelle $\overline{F}(x,y)$ d'un problème $(\overline{P})$, on appelle fonction de perturbation généralisée la fonction $\overline{\Phi}$ définie par :

$$\forall y \in Y : \quad \overline{\Phi}(y) = \min(\overline{P_y}) = \min_{x \in X} \overline{F}(x,y) \tag{70}$$

On remarque que pour

$$y = 0 \tag{71}$$

on a :

$$\overline{\Phi}(0) = \min(\overline{P}). \tag{72}$$

De plus, à partir des fonctions

$$\overline{\Phi}(y) = \min_{x \in X} \overline{F}(x,y) \text{ et } \overline{\omega}(\lambda) = \min_{x \in X} \overline{L}(x,\lambda) \tag{73}$$

et la relation :

$$\overline{L}(x,\lambda) = \min_{y \in Y} \{\overline{F}(x,y) + \lambda.y\} \tag{74}$$

On déduit immédiatement que $\overline{\Phi}(y)$ et $\overline{\omega}(\lambda)$ sont liées par :

$$\overline{\omega}(\lambda) = \min_{y \in Y} \{\overline{\Phi}(y) + \lambda.y\} \leq \overline{\Phi}(0) = \min(\overline{P}) \tag{75}$$

Ceci permet de déduire qu'une condition nécessaire et suffisante pour que λ^0 soit un multiplicateur de point-col est que:

$$\forall y \in Y : \quad \overline{\Phi}(y) \geq \overline{\Phi}(0) - \lambda^0.y \tag{76}$$

Autrement dit, le graphe de la fonction de perturbation généralisée $\overline{\Phi}(y)$ doit également se situer au dessus d'un hyperplan d'appui en $y = 0$ du graphe de $\overline{\Phi}(y)$.

On peut remarquer que la condition précédente prend la même forme que dans le cas des Lagrangiens classiques. Cependant, grâce à une construction plus judicieuse du Lagrangien généralisé, la condition ci-dessus peut être beaucoup moins stricte. Par conséquent, le Lagrangien généralisé permet d'élargir considérablement la classe des problèmes possédant un point-col et qui peuvent être résolu par les algorithmes précédent de résolution du problème dual. Nous donnons dans ce qui suit quelques exemples de Lagrangien augmentés.

Dans le cas des contraintes d'égalité. Les travaux de Hestenes [6] et, indépendamment, ceux de Powell [7] ont proposé de résoudre une séquence de problème sans contrainte de la forme :

$$\min_x L(x,\lambda,r) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{i=1}^{m} [g_i(x)]^2 \qquad (77)$$

Où λ est de signe quelconque.

On peut voir que cette méthode présente une amélioration directe des méthodes de pénalité en évitant d'avoir à utiliser des coefficients de pénalité trop grands. En effet, à l'optimum (x^*, λ^*), le gradient en x^* de la fonction $L(x, \lambda^*, r)$ est nul pour toutes les valeurs de r (car $g_i(x) = 0$). Au contraire, le gradient de la fonction de pénalité extérieure :

$$f(x)+r\sum_i[g_i(x)]^2 \tag{78}$$

en x^* est égale à $\nabla f(x^*)$ (puisque $g_i(x^*)=0$) donc généralement différent de zéro, quelle que soit la valeur de r. C'est pourquoi r doit être choisi très grand pour approcher suffisamment x^*, avec l'inconvénient que la fonction de pénalité devient très mal conditionnée.

Rockafellar [5] a généralisé la fonction de Hestenes et Powell au cas des contraintes inégalités. L'obtention du Lagrangien augmenté de Rockafellar peut s'interpréter de la façon suivante :

On introduit pour les contraintes inégalités des variables d'écart $s_i \geq 0$ et on récrit le problème sous la forme :

$$\begin{cases} \min f(x) \\ g_i(x)+s_i=0 \\ s_i \geq 0 \\ x \in \Re^n \end{cases} \tag{79}$$

On résout alors à chaque étape le problème :

$$\min_{x\in\Re^n, s\geq 0}\{f(x)+\sum_i \lambda_i(g_i(x)+s_i)+r\sum_i[g_i(x)+s_i]^2\} \tag{80}$$

Où λ est de signe quelconque. On remarque que la minimisation suivant $s \geq 0$, pour x fixé, peut se faire explicitement pour chaque s_i séparément. En supposant $r > 0$ le minimum sur s_i est atteint :

$$s_i = \begin{cases} -g_i(x) - \dfrac{\lambda_i}{2r} & \text{si} \quad g_i(x) \leq -\dfrac{\lambda_i}{2r} \\ 0 & \text{si} \quad g_i(x) \geq -\dfrac{\lambda_i}{2r} \end{cases} \tag{81}$$

Le problème revient alors à minimiser suivant x à chaque étape la fonction :

$$L(x,\lambda,r) = f(x) + \sum_{i=1}^{m} G(g_i(x),\lambda_i,r) \tag{82}$$

Où

$$G(g_i(x),\lambda_i,r) = \begin{cases} -\dfrac{\lambda_i}{4r} & \text{si} \quad r>0 \text{ et } g_i(x) \leq -\dfrac{\lambda_i}{2r} \\ \lambda_i g_i(x) + r[g_i(x)]^2 & \text{si} \quad r>0 \text{ et } g_i(x) \geq -\dfrac{\lambda_i}{2r} \\ \lambda_i g_i(x) & \text{si} \quad r=0 \text{ et } \lambda_i \geq 0 \\ -\infty & \text{si} \quad r=0 \text{ et } \lambda_i < 0 \end{cases} \tag{83}$$

Le Lagrangien augmenté de Rockafellar correspond à la représentation perturbationnelle :

$$\hat{F}(x,\hat{y}) = \begin{cases} f(x) & \text{si } g_i(x) \leq y_i, \ (\forall i \in I) \text{ et } \sum_i y_i^2 \leq s \\ +\infty & \text{sinon} \end{cases} \tag{84}$$

où $\hat{y} = (y_1, y_2, \ldots, y_m, s) = (y,s) \in \hat{Y} \equiv \Re^{m+1}$

La fonction duale généralisée devient :

$$\hat{\omega}(\hat{\lambda}) = \hat{\omega}(\lambda, r) = \min_{x \in \Re^n} \hat{L}(x, \lambda, r), \tag{85}$$

et le problème dual augmenté associé est :

$$\max_{(\lambda, r) \in \Re^{m+1}} \hat{\omega}(\lambda, r) \tag{86}$$

Rappelons que la fonction de perturbation ordinaire est définie par :

$$\Phi(y) = \min(P_y) = \min\{f(x)/g(x) \leq y\} \tag{87}$$

On a maintenant la fonction de perturbation généralisée :

$$\hat{\Phi}(\hat{y}) = \min_x \hat{F}(x, \hat{y}) = \min_x \hat{F}(x, y, s) \tag{88}$$

D'après la définition de $\hat{F}$, on a :

$$\hat{\Phi}(\hat{y}) = \hat{\Phi}(y, s) = \begin{cases} \Phi(y) & \text{si } \|y\|^2 \leq s \\ \infty & \text{sinon} \end{cases} \tag{89}$$

On peut interpréter $\hat{\omega}(\hat{\lambda})$ en terme de $\Phi(y)$:

$$\hat{\omega}(\hat{\lambda}) = \hat{\omega}(\lambda, r) = \min_{y,s}\{\hat{\Phi}(y, s) + \lambda.y + r.s\} = \min_{y,s,\|y\|^2 \leq s} \{\Phi(y) + \lambda.y + r.s\} \tag{90}$$

Comme $r > 0$, s doit être minimal et vaut donc $\|y\|^2$. On a donc :

$$\hat{\omega}(\hat{\lambda}) = \hat{\omega}(\lambda, r) = \min_y\{\Phi(y) + \lambda.y + r\|y\|^2\} \tag{91}$$

En supposons y unidimensionnel [2], on a représenté à la Figure C.2 la fonction de perturbation ordinaire

$$z = \Phi(y) \tag{92}$$

Soit y^0 la valeur de y pour laquelle le minimum de $\hat{\omega}(\hat{\lambda})$ est obtenu. On a :

$$\hat{\omega}(\lambda,r) = \Phi(y^0) + \lambda.y^0 + r\left\|y^0\right\|^2, \tag{93}$$

$$\Phi(y) \geq \hat{\omega}(\lambda.r) - \lambda.y - r\left\|y\right\|^2 \tag{94}$$

et pour $y = y^0$

$$\Phi(y^0) = \hat{\omega}(\lambda.r) - \lambda.y^0 - r\left\|y^0\right\|^2 \tag{95}$$

Ceci montre que la valeur de la fonction duale $\hat{\omega}(\lambda.r)$ est l'ordonnée de l'intersection avec l'axe $y = 0$ d'une parabole d'équation :

$$z = z^0 - \lambda.y - r\left\|y^0\right\|^2 \qquad r > 0 \tag{96}$$

Située partout en dessous de la fonction $\Phi(y)$ et tangente en y^0 au graphe de $\Phi(y)$.

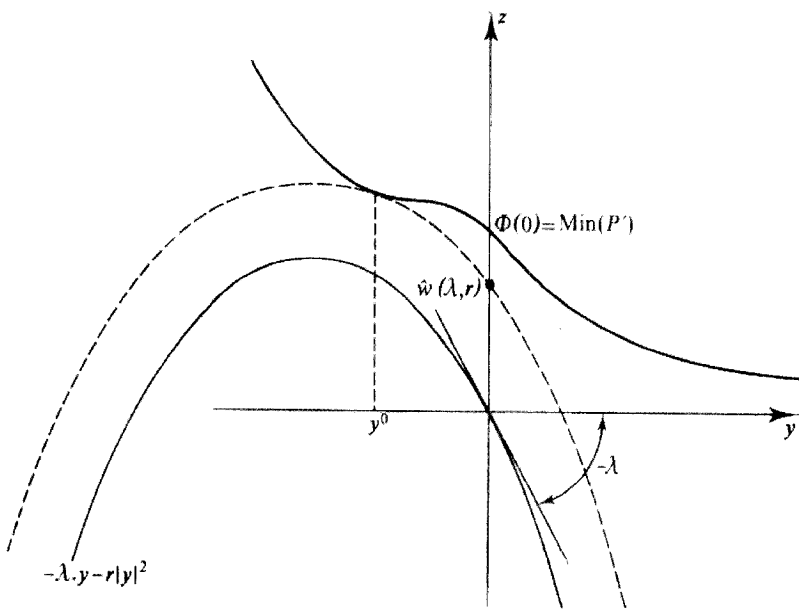


Figure C. 2 : Interprétation géométrique de la méthode du Lagrangien augmenté de Rockafellar

La condition d'existence d'un point-col tel que : $\hat{\omega}(\lambda.r) = \Phi(0) = \min(P)$ est alors équivalente à la condition : La fonction de perturbation ordinaire Φ admet au point $(y = 0, \Phi(0))$ une fonction de support quadratique concave.

On voit que cette condition peut être rempli par une classe de fonctions $\Phi(y)$ beaucoup plus vaste que les fonctions convexes. Les méthodes des Lagrangien augmentés permettent ainsi d'étendre les conditions nécessaires et suffisantes d'optimalité classiques à certaines classes des problèmes non convexes.

Cependant, il demeure difficile de trouver le minimum absolu de $\overline{L}(x,\lambda)$, fonction généralement non convexe en x. C'est pourquoi on doit toujours se contenté d'un minimum local sur éventuellement un point-col local.

Une autre méthode du Lagrangien augmenté a été décrite dans [8]. Dans le cas des contraintes d'égalité $(\theta(x) = 0)$. Elle est identique à la fonction de Hestenes et Powell.

Soit :

$$L_c(x,p) = f(x) + p.\theta(x) + c\left\|\theta(x)\right\|^2 / 2 \qquad (97)$$

Où p est le multiplicateur de Lagrange de signe quelconque et $c > 0$ est le coefficient de pénalité (ou d'augmentation).

Dans le cas des contraintes d'inégalité $(\theta(x) \le 0)$, la fonction de Lagrangien généralisé est sous la forme suivante :

$$L_c(x,p) = f(x) + \{\left\|\max(0, p + c.\theta(x)\right\|^2 - \left\|p\right\|^2 / 2c \qquad (98)$$

Où $p > 0$ est le multiplicateur de Lagrange et $c > 0$ est le coefficient de pénalité.

Nous remarquons d'abord que la formule ci-dessus, est équivalente à la formule de Rockafellar. En fait, elle a été déduite de la façon suivante. La dualisation des contraintes d'inégalité est une généralisation de celle des contraintes d'égalité en introduisant des variables d'écart ζ :

$$L_c(x,p) = \min_{\zeta \le 0} f(x) + p.(\theta(x) - \zeta) + c\|\theta(x) - \zeta\|^2 / 2 \qquad (99)$$

Pour résoudre le problème ci-dessus, on introduit un multiplicateur $q \ge 0$ et définit une fonction de Lagrangien classique :

$$L(x,p,\zeta,q) = f(x) + p.(\theta(x) - \zeta) + c\|\theta(x) - \zeta\|^2 / 2 + q.\zeta \qquad (100)$$

On en déduit que :

$$L_c(x,p) = \max_{q \ge 0} \min_{\zeta \le 0} l(x,p,\zeta,q) \qquad (101)$$

On obtient par des calculs simples une solution explicite

$$\zeta = (p + c\theta - q)/c \qquad (102)$$

et la relation suivante:

$$L_c(x,p) = \max_{q \ge 0} \{ f(x) + q.\theta - \|p - q\|^2 / 2c \qquad (103)$$

La dérivée de l'expression précédente par rapport à q s'annule lorsque $q = p + c.\theta$, on a donc :

$$L_c(x,p) = \begin{cases} f(x) + p.\theta + \frac{c}{2}\|\theta\|^2 & \text{si} \quad p + c.\theta \geq 0 \\ f(x) - \frac{p^2}{2c} & \text{si} \quad p + c.\theta \leq 0 \end{cases} \tag{104}$$

Soit :

$$L_c(x,p) = f(x) + \{\|\max(0, p + c.\theta(x)\|^2 - \|p\|^2 / 2c \tag{105}$$

En posant $p = \lambda$, et $c = 2r > 0$, la formule ci-dessus s'identifie avec le Lagrangien augmenté de Rockefeller, pour lequel nous avons déjà évoqué des propriétés intéressantes. De plus, Cohen et Zhu [8] ont énoncé le résultat important :

Pour c suffisamment grand tout problème non convexe admet au moins un point-col.

Il existe d'autres types de Lagrangiens augmentés basés sur l'idée de combinaison du Lagrangien classique avec des fonctions de pénalités. Par exemple, le Lagrangien augmenté présenté dans [9] combine à la fois les méthodes de dualité et les méthodes de pénalités intérieures et de pénalités extérieures. Il utilise la fonction :

$$L(x,\lambda,t) = f(x) + \sum_{i=1}^{m} \left\{ \begin{array}{ll} t[g_i(x)]^2 + \lambda_i g_i(x) & \text{si} \quad g_i(x) \geq 0 \\ \dfrac{\lambda_i^2 g_i(x)}{\lambda_i - t g_i(x)} & \text{si} \quad g_i(x) < 0 \end{array} \right\} \tag{106}$$

avec $\lambda \geq 0$, $t > 0$

Citons aussi l'article de Gould *et al.* [10], et les références incluses contiennent des versions encore plus élaborées du Lagrangien augmenté.

En ce qui concerne les méthodes numériques utilisant la dualité lagrangienne, on peut citer les algorithmes d'Uzawa et de Arrow-Hurwiez [8]. L'algorithme d'Uzawa utilise une méthode de gradient classique (de sous-gradient dans le cas non différentiable) pour résoudre le problème dual. A chaque étape la fonction de Lagrange est minimisée. Dans ses grandes lignes, la méthode est la suivante :

Algorithme d' Uzawa

(a) Partir d'un point $\lambda^0 \geq 0$

(b) A l'itération t on est au point λ^t . Calculer

$$\omega(\lambda^t) = \min_{x \in \Re^n} \{f(x) + \lambda^t . g(x)\} = f(x^t) + \lambda^t . g(x^t) \qquad (107)$$

(c) $g(x^t) = [g_i(x^t)]_{i \in I}$ est un sous-gradient de au point λ^t (le gradient dans le cas différentiable).

(d) Définir λ^{t+1} par :

$$\lambda^{t+1} = \text{Proj}_{\Re^{m+}} \{\lambda^t + p_t . g(x^t)\} \qquad (108)$$

Où p_t est le pas de déplacement à l'étape t.

Si test d'arrêt vérifié : FIN Sinon faire $t \leftarrow t+1$ et retourner en (b).

La méthode de Arrow-Hurwiez est très voisine de la précédente. La différence réside dans le fait qu'à chaque itération t, on ne recherche pas le minimum en x de la fonction de Lagrange $L(x, \lambda^t)$ mais on se contente d'un déplacement suivant un pas

prédéterminé dans la direction de descente $-\nabla_x L(x^{t-1}, \lambda^t)$. La méthode consiste ainsi à se déplacer alternativement dans l'espace des variables primales et dans l'espace des variables duales avec des pas prédéterminés.

C. 5. Références :

[1] A E. Bryson & Y C Ho, "*Applied optimal control: optimization, estimation, and control*", Waltham, MA: Blaisdell, Stanford CA, 1969. p 481

[2] M. Minoux, "*Programmation mathématique*", Tome 1, Dunod, 1983.

[3] D. P. Bertsekas, "*Non Linear Programming*", Athena Scientific, 1995.

[4] E. Polak: "*Optimization, Algorithms and Consistent Approximation*", Springer, N.Y, 1997.

[5] T. Rockafellar, "Lagrange Multipliers and Optimality", *SIAM Review* 35, p 183-238, 1993

[6] M. R. Hestens, "Multiplier and Gradient Methods", *J. Optimization Theory and Applications* 4, 303-320, 1969.

[7] M. J. D. Powell, "A Method for Nonlinear Constraints in Minimization Problems", in *Optimization*, R. Fletcher, Ed., Academic Press London, p 283-298, 1969.

[8] H. Nakayama, H. Sayama and Y. Sawaragi, "A generalized Lagrangian function and multiplier method", J. Optim. Theory Appl. 17, pp: 211-227, 1975

[9] G. Cohen, , and D. L. Zhu, "*Decomposition-Coordination Methods in Large-Scale Optimization Problems*: *The Non-differentiable Case and the Use of Augmented Lagrangians*", Advances in Large-Scale Systems: Theory and Applications, Edited by J. B. Cruz, JAI Press, Greenwich, Connecticut, Vol. 1, pp.203–266, 1984.

[10] N. Gould, D. Orban, A. Sartenear, and P. Toint, "Super Linear Convergence of Primal-Dual interior Points Algorithms for Non Linear Programming", in *SIAM J. on Opt, Math Prog.* B, v 11, 4, p 974-1002, 2001.