Multi- Objective Trajectory Planning of Mobile Parallel Manipulator

BY

Mutaz "Mohammad Ali" Hamdan

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS ENGINEERING

May 2012

KING FAHD UNIVERSITY OF PETROLUM & MENERALS

DHAHRAN 31261, SAUDIARABIA

DENSHIP OF GRADUATE STUDIES

This thesis, written by MUTAZ "MOHAMMAD ALI" HAMDAN under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of MASTER OF SIENCE IN SYSTEMS ENGINEERING.

Thesis Committee

May 27, 2012

Dr. Amar Khoukhi (Thesis Advisor)

May 27, 12

Dr. Fouad Al-Sunni

M. Elsh.

Dr.Moustafa Elshafei

May 27, 12

Dr. Fouad Al-Sunni

Department Chairman

Dr. Salam Zummo

Dean of Graduate Studies

عمادة الدراسات العليا

DEANSHIP OF GRADUATE STUDIES

Date: 29/5/12

Dedicated to

My PARENTS, ADVISOR,

BROTHER and SISTERS

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor Dr. Amar Khoukhi, for his guidance, support, encouragement, and patience throughout my graduate study. Sincere thanks go to both the members of my advisors committee, Dr. Fouad Al-Sunni and Dr. Moustafa Elshafei for their constructive guidance and technical support.

Special thanks are due to my senior colleagues at the university, who were always there to provide thoughtful solutions to the various problems encountered in my research. I would also like to thank all my friends, for a wonderful company and good memories that will last a life time.

Finally, thanks are due to my parents, my sisters and brother as well as all my family members for their emotional and moral support throughout my academic career and also for their love, forbearance, encouragement and prayers.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS IV

TABLE OF CONTENTS V

LIST OF FIGURES VIII

THESIS ABSTRACT XI

ملخص الرسالة XII

Chapter 1 Introduction 1

Chapter 2 Literature Review 4

2.1 Trajectory planning of robotic systems 4

2.1.1 Trajectory planning of mobile manipulator 6

2.1.2 Parallel robots 8

2.1.3 Mobile Parallel robots 11

2.2 Off-line trajectory planning of robotic systems 13

2.2.1 Approaches to Off-line trajectory Planning 13

2.2.2 Off-line trajectory Planning Systems for Mobile Parallel Manipulator (MPM) 15

2.3 On line trajectory planning of robotic systems 16

2.3.1 Approaches to On-line trajectory planning through soft computing 16

2.3.2 On-line trajectory Planning Systems for MPM 17

Chapter 3 Kinematic Modeling 19

3.1 Architecture Design of the MPM 19

3.2 Kinematic Modeling 21

3.2.1 Architecture Description 21

3.2.2 Position Kinematics Analysis 22

3.2.3 Differential Kinematics Analysis 27

3.2.4 Kinematic Singularity Characterization 30

3.2.5 Redundancy Resolution through Joint Limits and Singularity Avoidance 31

Chapter 4 Kinematic Initialization 32

4.1 Optimal time trajectory parameterization 32

4.2 Neuro-fuzzy inverse kinematics 35

4.3 Dynamic trajectory generation 40

Chapter 5 Dynamic Modeling 41

5.1 Dynamic Model Analysis 42

5.1.1 Discrete Time Dynamic Model 48

5.2 Constraints Modeling 51

5.2.1 Robot Constraints 51

5.2.2 Task and Workspace Constraints 52

5.3 Model Validation 53

5.3.1 Model-based controller design 54

5.3.2 Simulation result for model validation 55

Chapter 6 Problem Formulation..... 60

6.1 Minimum Time Control Problem 61

6.2 Minimum Energy Control Problem 62

6.3 Redundancy Resolution and Singularity Avoidance Control Problem..... 62

6.4 The Objective Functional for the Considered Problem 63

Chapter 7 Offline Trajectory Planning 65

7.1 Augmented Lagrangian Approach 65

7.2 Constrained Linear-Decoupled Formulation 68

7.3 Augmented Lagrangian for the Decoupled Formulation 72

7.4 Implementation issues..... 74

7.4.1 Initial solution: 74

7.4.2 Search Direction Update 74

7.4.3 Overall Solution Procedure..... 76

7.5 Simulation and results for offline trajectory planning 77

7.6 Simulation and results for online trajectory planning..... 87

Chapter 8 Conclusions and Future Work..... 90

Appendix A: Maple Solution for Position Kinematic Analysis..... 92

References..... 93

Vita..... 105

LIST OF FIGURES

Figure 2.1: Schematic description of Shraga's mobile parallel mechanism [59] . 12

Figure 2.2: the mobile parallel manipulator studied by Yangmin Li, et al [6] 13

Figure 3.1: Schematic representation of a mobile parallel manipulator [6] 20

Figure 4.1: The use of the FK to the learning of the NeFIK module..... 35

Figure 4.2: NeFIK performance – root mean square error output with respect to learning epochs for derivatives of (q_1, q_2, q_3, q_4, q_5) 36

Figure 4.3: NeFIK performance – difference between real and estimated values of the MPR angles 37

Figure 4.4: variation of the EEF coordinates for the motion of the parallel mechanism alone..... 38

Figure 4.5: variation of the joint angles for the motion of the parallel mechanism alone 39

Figure 4.6: variation of the EEF coordinates for the motion of the MPR 39

Figure 4.7: variation of the EEF coordinates for the motion of the MPR 39

Figure 5.1: Block diagram of the model-based controller 54

Figure 5.2: Desired and controlled loci for the linear trajectory..... 56

Figure 5.3: Position and angular tracking error for the linear trajectory 57

Figure 5.4: Desired and controlled loci for the curve trajectory..... 57

Figure 5.5: Position and angular tracking error for the curve trajectory 58

Figure 5.6: variation of the condition number for unstable system 58

Figure 5.7: variation of the condition number for stable system 59

Figure 7.1: Flowchart for AL algorithm function and operation 75

Figure 7.2: Variations of the angels due to minimization of time 78

Figure 7.3: Variations of the end effector position due to minimization of time . 79

Figure 7.4: Variations of the end effector velocity due to minimization of time . 79

Figure 7.5: Variations of the torque due to minimization of time 80

Figure 7.6: Variations of the time steps due to minimization of time 80

Figure 7.7: Variations of the angels due to minimization of energy 81

Figure 7.8: Variations of the end effector position due to minimization of energy 82

Figure 7.9: Variations of the end effector velocity due to minimization of energy 82

Figure 7.10: Variations of the torque due to minimization of energy 83

Figure 7.11: Variations of the time step due to minimization of energy 83

Figure 7.12: Variations of the angels due to minimization of both time and energy 84

Figure 7.13: Variations of the end effector position due to minimization of both time and energy..... 85

Figure 7.14: Variations of the end effector velocity due to minimization of both time and energy..... 85

Figure 7.15: Variations of the torque due to minimization of both time and energy 86

Figure 7.16: Variations of the time step due to minimization of both time and energy..... 86

Figure 7.17: The use of the FK to the learning of the NeFIK module..... 88

Figure 7.18: NeFIK performance – root mean square error output with respect to learning epochs 88

Figure 7.19: NeFIK performance – difference between real and estimated values of the MPM values 89

THESIS ABSTRACT

NAME: Mutaz "Mohammad Ali" Hamdan

TITLE: Multi-Objective Trajectory Planning of Mobile Parallel Manipulator

MAJOR: Systems Engineering

DATE: June 2012

This thesis studies a new generation of robotic systems; Mobile Parallel Manipulator (MPM); these are composed of a multi-degree of freedom parallel platform carried by an autonomous wheeled mobile robot. Both position and differential kinematics problems are solved in details. The problems of kinematic singularity and redundancy are solved through joint limits avoidance and manipulability criteria. Furthermore, taking the MPM self motion into consideration due to its redundancy, the inverse kinematic is derived using hybrid neuro-fuzzy system called NeFIK. The discrete Augmented Lagrangean (AL) technique is then used to solve the highly nonlinear constrained multi-objective optimal control problem. Based on the result of the AL solution, an ANFIS based structure is built to solve for the online trajectory planning of the MPM. Result of both offline and online shows that the trajectory planning of MPM is derived with small acceptable error.

Master of Science Degree

King Fahd University of Petroleum and Minerals

Dhahran, Saudi Arabia

June 2012

ملخص الرسالة

الاسم : معتز "محمد علي" حمدان

العنوان: تخطيط مسار متعدد الأهداف للمناول المتوازي المتنقل

التخصص: هندسة النظم

التاريخ : رجب 1433هـ

في هذا البحث تم اشتقاق النموذجين الحركي والديناميكي للمناول المتوازي المتنقل والذي يتكون من روبوت متوازي متعدد درجة حرية الحركة ومنصة متنقلة ذاتية الحركة ذات عجلات. وتم حل مسألتي الوضع والحركة لهذا الهيكل الهجين بالتفصيل. أيضاً تم حل مسألة توصيف التفرد والتكرر من خلال وضع حدود على مفاصل الحركة وتجنب التفرد.

بالاضافة لما سبق وبأخذ بعين الاعتبار الحركة الذاتية للمناول المتوازي المتنقل والتي تحدث نتيجة التكرر، تم اشتقاق النموذج الحركي العكسي باستخدام نظام عصبي ضبابي مختلط. واستخدمت طريقة لاجرانج المتزامنة المنفصلة في حل مسألة التحكم الأمثل المقيدة ذات الدرجة غير الخطية العالية. وباستخدام نتائج حل المسألة بطريقة لاجرنج المتزامنة تم بناء نظام استدلال عصبي ضبابي تكيفي لحل مسألة تخطيط مسار الحركة. أظهرت نتائج المحاكاة التي تم الحصول عليها فعالية هاتين الطريقتين في تخطيط مسار الحركة للمناول المتوازي المتنقل وبدرجة خطأ قليلة متقبلة.

درجة الماجستير في العلوم

جامعة الملك فهد للبترول و المعادن

الظهران-المملكة العربية السعودية

رجب 1433هـ

Chapter 1

Introduction

Mobile robots have wide applications in such areas as automatic material handling in warehouse, transportation and health care in hospital, and exploration in hazardous environments. At the same time, the progress in the development of parallel robots (PRs) is accelerated since PRs possess many advantages over their serial counterparts in terms of high accuracy, velocity, stiffness, and payload capacity. However, the

major drawback of PRs is their limited workspace that restricts their wide variety of applications.

Up to now, several researchers have proposed parallel mobile robot using mobile joints between the legs and the fixed platform with each leg driven by an asynchronous unit, that allows the robot have a theoretically unlimited workspace in the horizontal plane [1], [2]. Nevertheless, the position accuracy of the robot needs to be improved due to odometric errors. In addition, a simple parallel mechanism mobile robot is presented in [3] by mounting a 4R or 5R closed kinematic chain on a crawler mechanism to perform such tasks as getting over a bump or going up to a high level, etc. And a combination of a mobile robot and a Stewart platform is proposed in [4], [5] for active acceleration compensation so as to transport objects smoothly.

In order to increase the effective workspace of parallel robots while maintaining their inherent advantages, the novel design of a mobile parallel manipulator (MPM) is proposed in [6] by adding a wheeled mobile platform to a parallel robot, which provides extra mobility to the robotic system and thus enlarges its reachable workspace extensively. Since in most cases, the mobile robot is subjected to nonholonomic constraints and the parallel robot introduces many complex kinematic constraints, the integration of a parallel robot and a mobile one induces a large number of challenging difficulties involving of how to decompose a given task into fine motions to be achieved by the parallel robot and the gross motions to be carried

out by the mobile robot, and how to establish the dynamic model of the hybrid system in a systematic way, etc.

Since a MPM possesses advantages of both a mobile robot and a parallel robot, it is a potential competitor in extensive applications where high accuracy operation, and high rigidity and payload capacity are required, such as an autonomous guidance vehicle, service robots and personal robots, underwater robots, and space robots, etc.

The remainder of this thesis is organized as follows. Chapter 2 proposed a literature review on the trajectory planning, parallel robots, and mobile parallel robots. The kinematic modeling is derived in Chapter 3 and the dynamic modeling described in Chapter 4. Then in Chapter 5, the problem of the thesis is formulated. The kinematic initialization solution is solved in Chapter 6, followed by off-line trajectory planning of the MPM in chapter 7. Finally, conclusions and suggestion for future work are shown Chapter 8.

Chapter 2

Literature Review

2.1 Trajectory planning of robotic systems

A problem of trajectory planning is an active field of the research so there is a vast literature treating this issue.

A new method for smooth trajectory planning of robot manipulators is developed by Gasparetto and Zanotto [7]. They worked out an objective function containing a term proportional to the integral of the squared jerk and the second term, proportional to

the total execution time. Saramago and Ceccarelli [8] formulated optimization problem physical constraints, input torque and force constraints and payload limits. They proposed the optimization of trajectory path planning taking into account robot actuating energy and grasping forces in manipulator gripper.

Minimum cost problem of manipulator motion is solved by Saramago and Steffen [9]. A multi-objective function is build using the optimal traveling time and the minimum mechanical energy of the actuators. Chettibi et al. [10] study the problem of minimum cost trajectory planning by transforming the optimal control problem via clamped cubic spline model of joint temporal evolutions into a non-linear constrained optimization problem by the SQP method (sequential quadratic programming). Using a genetic algorithm (GA)-enhanced optimization of the pose ruled [11] presented a unified approach to optimal pose trajectory planning for robot manipulators in cartesian space.

A trajectory motion planning in environments with obstacles are discussed by some research papers. Using the concept of APF (artificial potential fields) Agirrebeitia et al. [12] solving planning of mobile robot motion as well as high redundant multi-body systems. This strategy is valid for 2D and 3D environments, static or dynamic. Using algorithm capable of obtaining a sequence of feasible robot configurations between the given initial robot configuration and the goal robot configuration, Valero et al. [13] planning trajectory for industrial robots in workspaces with obstacles.

A method for optimal trajectory planning of robot manipulators in the operational space with moving obstacles is presented by Saramago and Steffen [14]. The algorithm regards the non-linear manipulator dynamics, actuator constraints, joint limits and obstacle avoidance.

2.1.1 Trajectory planning of mobile manipulator

Mobile manipulator contains mobile platform carrying a serial manipulator. A common approach in motion planning for this type is to conduct trajectory planning on the basis of a path generated by a path planner. A notable framework is the elastic strip method [15], which can deform a trajectory for a robot locally to avoid moving obstacles inside a collision-free "tunnel" that connects the initial and goal locations of the robot in a 3-D workspace. Such a "tunnel" is generated from a decomposition based path planning strategy [16]. Another approach is to conduct path and trajectory planning simultaneously. However, the offline algorithms takes the most effort in this category which is focused on assuming that the environment is completely known beforehand, i.e., static objects are known, and moving objects are known with known trajectories [17; 18; 19; 20]. As for dealing with unknown moving obstacles, only recently some methods were introduced for mobile robots [21; 22].

The coordination of the mobile base and the manipulator is the major issue of motion planning of mobile manipulators. This issue presents both challenges and opportunities since it involves redundancy resolution. There exists a lot of literature

addressing this issue from many aspects. Some researchers treat the manipulator and the mobile base together as a redundant robot in planning its path for place-to-place tasks [23; 24; 25]. Some focused on planning a sequence of "commutation configurations" for the mobile base when the robot was to perform a sequence of tasks [26; 27] subject to various constraints and optimization criteria. Others focused on coordinating the control of the mobile base and the manipulator in a contour-following task [28; 29] by trying to position the mobile base to maximize manipulability.

Most of the existing research assumes that the environments is known with the obstacles for a mobile manipulator, but a few local collision avoidance of unknown, moving obstacles online is considered. One method as in [30] used RRT as a local planner to update a roadmap originally generated by PRM to deal with moving obstacles. For contour-following tasks, an efficient method [31] allows the base to adjust its path to avoid a moving obstacle if possible while keeping the end-effector following a contour, such as a straight line. Another method [29] allowed the base to pause in order to let an unexpected obstacle pass while the arm continued its contour-following motion under an event-based control scheme. Other methods include one based on potential field [32] to avoid unknown obstacles and one based on a neuro-fuzzy controller [33] to modify the base motion locally to avoid a moving obstacle stably. There is also an online planner for the special purpose of planning the motions of two robot arms getting parts from a conveyer belt [34].

Zhijun Li and Weidong Chen proposed adaptive neuro-fuzzy (NF) control for coordinated multiple mobile manipulators for robust force/motion tracking on the constraint surface while it is in motion [35].

John Vannoy, Jing Xiao introduced a novel and general real-time adaptive motion planning (RAMP) approach suitable for planning trajectories of high-DOF or redundant robots such as mobile manipulators in dynamic environments with moving obstacles of unknown trajectories under various optimization criteria, such as minimizing energy and time and maximizing manipulability [36].

2.1.2 Parallel robots

One of the widely types of robotics research is the parallel robots, their design dates back to the work by Gough [37], who was behind the establishing of the basic principles of a manipulator in a closed loop structure. His machine was able to position and orientate an end-effector (EE), for testing tire wear and tear. After one decade, a platform manipulator for the use as an aircraft simulator is proposed by Stewart [38]. After that, extensive research efforts lead to the realization of several robots and machine tools with parallel kinematic structures [39].

Parallel machines have two basic advantages over conventional machines of serial kinematic structures. On one hand, the high structural stiffness and rigidity caused by the connection between the base and the EE is made with several kinematic chains. On the other hand, this structure, make it possible to mount all drives on or near the

base. Which allow 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 1/2 for parallel ones. Despite these advantages, PKMs are still rare in the industry.

The small workspace is one of the major reasons of this gap, also the complex transformations between joint and Cartesian space and singularities comparing to their serial counterparts. These issues lead to a lot amount of research in design and customization [39]. The under consideration of the dynamics of these machines is another reason which is identified [40].

Comparing to serial robots the architecture-dependent performance associated with the strong coupled nonlinear dynamics makes the trajectory planning and control system design for parallel robotics more difficult. There is a plentiful literature published for serial robots, on the topics of off-line and online programming, from both types: computational geometry and kinematics, and optimal control including robots dynamics [41; 42; 43; 44].

A relatively large amount of literature For PKMs is devoted to the computational kinematics and workspace optimization issues. For PKMs trajectory, the overwhelm criteria considered planning are essentially design-oriented. These include singularity avoidance and dexterity optimization [45; 46; 47; 48; 49]. The authors had developed a clustering scheme to isolate and avoid singularities and obstacles for a PKM path planning in [50]. In [51], a kinematic design and planning method had been described

for a four-bar planar manipulator mechanism. It had been shown that a motion planning with singularity-free pose change is possible for PKMs in [52].

Planning a singularity-free minimum-energy path between two end-points for Gough–Stewart platforms using variational approach is described in [53]. This method is based on a penalty optimization method. But as shown in [54]penalty methods, have several disadvantages. Using of PKMs in industry (in a machining process, for example) is one of the major and practical issue for off-line programming, the control system should guarantee the predetermined task completion within the workspace, for a given set up of the EE. This issue has been considered with design methodologies involving workspace limitations and actuator forces optimization using optimization techniques [55; 56].

Khoukhi et al presented a new approach to multi-objective dynamic trajectory planning of parallel kinematic machines (PKM) under task, workspace and manipulator constraints [57]. It minimizes electrical and kinetic energy, robot traveling time separating two sampling periods, and maximizes a measure of manipulability allowing singularity avoidance. The discrete augmented Lagrangean technique is used to solve the resulting strong nonlinear constrained optimal control problem.

2.1.3 Mobile Parallel robots

A literature survey on mobile parallel robots shows that the working on parallel mobile robot is a hot area of researcher, and the study of the topic is still open. Rene Graf and Rudiger Dillmann proposed the use of a Stewart platform mounted on a mobile platform to compensate the unwanted accelerations in the way that the Stewart platform generates anti-acceleration [4]. The necessary movement of the platform is calculated by a so called washout filter. Applications of this combination are either the transport of liquids in open boxes or medical transports, where the patients must not be affected by any acceleration.

M. W. Decker et al implemented and compared several different approaches for the motion planning of Gough-Stewart Platform mounted on mobile robot [58], they aimed to enhance the capabilities of transport vehicles so that they can carry delicate objects of various shapes and sizes without requiring extensive packaging to protect them.

Shraga Shoval and Moshe Shoham presented a novel design for a mobile robot [59], the kinematic of this robot is combines techniques of parallel mechanisms with conventional wheeled units. The robot consists of three legs, each driven by an asynchronous mechanism connected to the legs with a spherical joint. Each leg is also connected to an upper platform with a revolute joint, resulting in a mobile, six DOF, parallel mechanism.

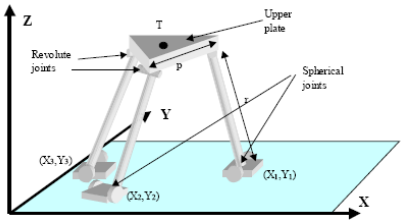


Figure 2.1: Schematic description of Shraga's mobile parallel mechanism [59]

The direct and inverse dynamic problems of [59] manipulator are solved by P. Ben Horin et al in [60]. It is shown that the Jacobian associated with the direct problem becomes identically singular when used to solve the inverse problem, and hence must be redefined; and that once redefined, it loses its standard structure and cannot be used to solve the direct problem. Three solution methods to the inverse problem are presented and are shown to lead to indistinguishable results.

T. Yamawaki et al proposed a self-reconfigurable parallel robot, which can be configured to 4R and 5R closed kinematic chains [61]. They proposed a parallel mechanism mobile robot by mounting it on a crawler mechanism. The combined mobile robot can gain some useful functionality from the advantage of its parallel mechanism other than just locomotion, such as getting over a hump by control of its center of gravity and carrying an object by making use of its shape. They analyzed the motions of the functionalities and verified them experimentally using the robots.

Yangmin Li, et al proposed a novel design and modeling of mobile parallel manipulator (MPM) [6]. This MPM composed of a three-wheeled nonholonomic

mobile platform and a 3-RRPaR translational parallel robot is designed and investigated in details. The position kinematics solutions are derived and the Jacobian matrix relating output velocities to the actuated joint rates is generated.

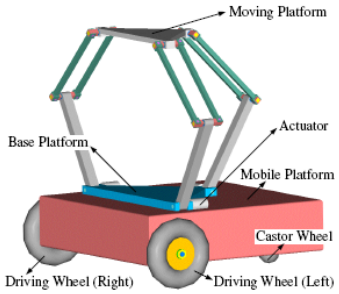


Figure 2.2: the mobile parallel manipulator studied by Yangmin Li, et al [6]

Huapeng Wu et al presented a novel mobile parallel robot, which is able to carry welding and machining processes from inside the international thermonuclear experimental reactor (ITER) vacuum vessel (VV) [62; 63; 64; 65]. The kinematic design of the robot has been optimized for ITER access.

2.2 Off-line trajectory planning of robotic systems

2.2.1 Approaches to Off-line trajectory Planning

A lot of researchers have been working over the last two decades on computational methods to generate optimal control for general manipulator robots for both offline and online programming. Motion planning for robots had been considered from two different points of view. First, from computational kinematics and CAD standpoints, it consists merely to assimilate the robot, workspace, and environment to that of a

Windows application using a dedicated CAD/CAM graphics-based interactive simulation system, with menus, toolbars and icons and implements advanced 3D modeling, drawing, and simulation tools to obtain as accurate positioning results as possible in 3D space. Examples of such software packages are CATIA-Robotics, IGRID, Robot Master, and ROBCAD.

On the other hand, from control systems standpoint, the problem consists in finding the sequence of optimal torques to achieve the displacement of the robot from a starting to an ending poses, while optimizing a cost function. One way of thought to the trajectory planning is that of making the analysis and planning over the phase space rather than the configuration space. The trajectory planning is solved by optimizing a performance index from a state-space representation and applying optimal control theory and variational calculus techniques using a system of differential equations [43]. From this, several criteria and constraints to satisfy in the course of the trajectory planning process by introducing dynamic parameters of the robot. Several works had been published, especially those dealing with minimum time path planning of serial manipulators. This is widely justified as to increase production by efficient use of the robot capacity, which is demonstrated by executing tasks as fast as possible. However, minimum time control is essentially of Bang-Bang type, which has several drawbacks [57]. Several other criteria had been proposed, such as minimum energy planning, minimum time-energy planning and obstacle avoidance.

2.2.2 Off-line trajectory Planning Systems for Mobile Parallel Manipulator (MPM)

In this research, an integrated off-line programming approach will be developed for MPM. A decoupling and linearizing approach to MPM multi-objective optimal control is introduced in order to handle some intractable computation issues within the non linear and non decoupled formulation. The multi-objective optimization procedure will performed within a proper balance between time and energy minimization, singularity avoidance, actuators, sampling periods, link lengths and workspace limitations, and task constraints. From a state space representation by a system of differential equations, the trajectory planning is formulated using a variational calculus framework. The resulting constrained nonlinear programming problem will solved using an augmented Lagrangian (AL) with decoupling technique.

AL algorithms have proven to be robust and powerful to cope with difficulties related to non-strictly convex constraints as compared to optimization methods employing only penalty. The decoupling technique is introduced in order to solve difficult computations, related mainly to the co-states, in the original coupled formulation. Another advantage of the proposed method is that one might introduce several criteria and constraints to satisfy in the trajectory planning process.

2.3 On line trajectory planning of robotic systems

2.3.1 Approaches to On-line trajectory planning through soft computing

Neuro-fuzzy systems represent a newly developed class of hybrid intelligent systems combining the main features of artificial neural networks with those of fuzzy logic systems. The main purpose in this issue is to overcome difficulties of applying fuzzy logic for systems represented by numerical knowledge (data sets), or conversely in applying neural networks for systems represented by linguistic information (fuzzy sets). As it known that neither fuzzy reasoning systems nor neural networks are by themselves capable of solving problems involving at the same time, both linguistic and numerical knowledge.

Using a set of simple "if-then" rules, fuzzy logic theory permits the accurate representation of a given system behavior, but it unable to processing knowledge stored in the form of numerical data. For this common type of system, "if-then" rules have to be extracted manually from the data sets, a process that becomes very tedious or even impossible to achieve for data sets with large numbers of patterns. But also the problem may become harder when the knowledge about the system is stored in both forms: linguistic (fuzzy sets) and numerical (data sets). This is the case for large-scale systems characterized by complex dynamics and ill-defined behavior.

On the other hand neural networks are universal approximators which have the ability learn virtually any (smooth) nonlinear mapping, and in the same time providing a high degree of accuracy. Neural networks are excellent classifiers and predictors. In spite of their versatility, neural networks have drawback which is the implicit representation of knowledge (known among researchers as the black box structure).

It was noticed that it is very difficult to quantitize the meaning of weights among the nodes of the network once the systems have been trained. As such, neural networks are not very clear in explaining their decision-making process. In addition, it is difficult to incorporate additional knowledge into the system without retraining it. It is even more difficult to extract from the data patterns linguistic representation of knowledge [66].

This leads us to find a way to overcome the limitations of both system representations (fuzzy and neural); researchers in the area have proposed incorporating fuzzy logic reasoning within a learning architecture of some sort. Neural networks have been shown to be excellent candidates for this task. Automating the generation of fuzzy rules using neural networks and optimizing the parameters of the fuzzy sets have been among the major objectives of several researchers in this field for recently.

2.3.2 On-line trajectory Planning Systems for MPM

Adaptive neuro-fuzzy network, called NeFOTC (Neuro-Fuzzy Optimal-Time Controller) in this research. It is based on a Tsukamoto fuzzy inference system will be

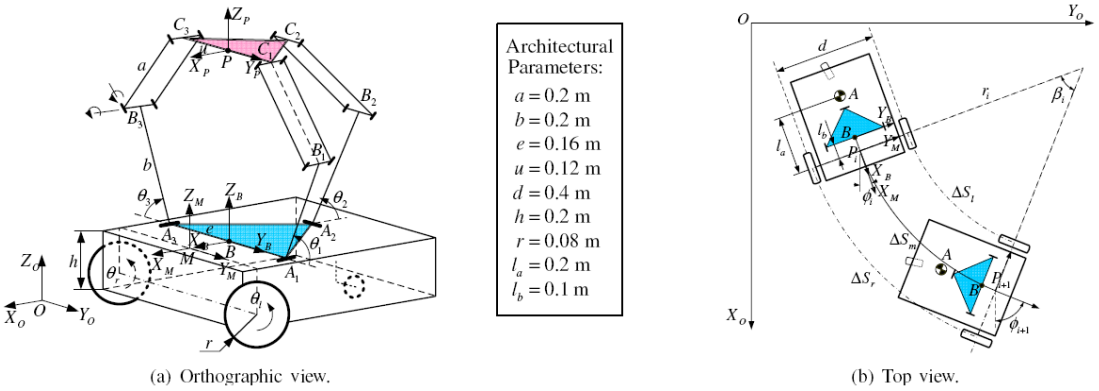
used to learn the premise parameters as well as the crisp outputs of the fuzzy rules. It starts with a subtractive clustering of input–output data and then the fuzzy inference parameters are learnt with a gradient back-propagation error function thereby giving the optimal time actuator torques. The Levenberg–Marquard version of the gradient back propagation algorithm is again used to learn premise and consequent parameters of the fuzzy rules.

Chapter 3

Kinematic Modeling

3.1 Architecture Design of the MPM

A mobile parallel manipulator (MPM) can be designed to have much different architecture. For the sake of this work, a 4-DOF MPM is chosen; it can be described as follows: A three-wheeled nonholonomic mobile robot with two fixed driving wheels and one castor wheel is chosen to construct the mobile platform. Additionally, a modified version of DELTA parallel robot with three translational DOF is selected to mount on the mobile platform [6].



(a) Orthographic view. (b) Top view.

Figure 3.1: Schematic representation of a mobile parallel manipulator [6]

Utilizing only revolute joints, the parallel robot is designed to have special arrangements of fixed motors, which result in a more compact structure with a larger reachable workspace than the original "DELTA robot". Moreover, the fixed actuators make it possible that the moving components of the parallel robot do not bear the load of actuators. This enables large powerful actuators to drive relatively small structures, facilitating the design of a robot with faster, stiffer, and stronger characteristics. Figure 3.1 illustrates a CAD model of the designed MPM, which possesses four DOF including three spatial translational DOF and one rotary DOF around the z axis.

With respect to the mechanism design, the selection of MPM architecture heavily depends on the task to be performed. Other types of mobile robots and parallel robots can also be employed to meet the requirement for a specified task performed by a MPM [6].

3.2 Kinematic Modeling

3.2.1 Architecture Description

In this research we use the architecture of [6] which consists of a three-wheeled nonholonomic mobile robot and a modified version of DELTA parallel robot (Fig.1). Figure 3.2 represents the schematic diagram of the designed MPR. The notation of R and Pa stands for the revolute and parallelogram joints, respectively [6].

Refers to Figure 3.3, the MPM is assumed just move on a plane. a fixed Cartesian frame (global frame) $O\{X_O, Y_O, Z_O\}$ is assigned on the plane of motion, a moving frame $M\{X_M, Y_M, Z_M\}$ on the mobile platform, a moving frame $B\{X_B, Y_B, Z_B\}$ at the centered point B of the base platform $\Delta A_1A_2A_3$, and another moving Cartesian frame $P\{X_P , Y_P , Z_P\}$ on the triangle moving platform $\Delta C_1C_2C_3$ at the centered point P.

Now, for frame M, the Y_M axis is along the coaxial-line of the two fixed wheels, X_M is perpendicular to Y_M and passes through the midpoint of the line segment connecting the two fixed wheel centers, and the Z_M axis is vertical to the mobile platform. In addition, the X_B and X_P axes are parallel to the X_M axis and the Y_B and Y_P axes are parallel to the Y_M axis, respectively [6].

In order to get a compact structure such as in parallel manipulator, both the base and moving platforms are designed to be isosceles right triangles described by parameters of e and u, respectively, i.e., $BA_i = e$ and $PC_i = u$, for i =1, 2, and 3. Also, the actuated variable of the ith limb is denoted by angle θ_i. The connecting joints between the

upper and lower links are denoted as B_i, and the lengths of upper and lower links for each limb are a and b respectively.

The plane of motion can be described as follows: the kinematics of the mobile platform can be consists of three parameters of coordinates of point M (x_m, y_m) and the heading angle (ϕ_m). Referring to Figure 3.2(b), let d be the distance between the two fixed wheels, l_b be the offset of the base platform of the parallel robot with respect to the origin of frame M, and l_a denote the offset of the mass center A of the mobile platform with respect to frame M. Additionally, $P^i(x_m^i\ ,\ y_m^i\ ,\ \phi_m^i)$ and $P^{i+1}(x_m^{i+1}\ ,\ y_m^{i+1}\ ,\ \phi_m^{i+1})$ represent coordinates of the mobile platform at time t^i and t^{i+1}respectively; β_i and r_i are the corresponding yaw angle and steering radius at time t_i; ΔS_l, ΔS_r, and ΔS_m denote the advance of the left wheel, right wheel, and the origin of frame M in the time interval ($\Delta t = t^{i+1} - t^i$) respectively.

3.2.2 Position Kinematics Analysis

Referring to figure 3.5, it can be shown that

$$\begin{cases} \dot{x}_m = \frac{rc\phi_m}{2}\left(\dot{\theta}_L + \dot{\theta}_r\right) \\ \dot{y}_m = \frac{rs\phi_m}{2}\left(\dot{\theta}_L + \dot{\theta}_r\right) \\ \dot{\phi}_m = \frac{r}{d}\left(\dot{\theta}_r - \dot{\theta}_L\right) \end{cases} \quad 3.1$$

With c stands for cosine, s stands for sine, and r is the radius of each driving wheel, also θ_L and θ_r denote respectively the rotating angles of the left and right driving wheels.

Now, the general coordinates of the mobile platform to be $\xi = [x_m \quad y_m \quad \phi_m \theta_L \quad \theta_r]^{\mathrm{T}}$. Solving eq. 3.1 for the nonholonomic constraints of the MPR, which can be written as:,

$$\mathbf{D}(\xi).\dot{\xi} = 0, \tag{3.2}$$

Where

$$\mathbf{D}(\xi) = \begin{bmatrix} c\phi_m & s\phi_m & -d/2 - r & 0 \\ c\phi_m & s\phi_m & d/2 \quad 0 & -r \\ s\phi_m & -c\phi_m & 0 \quad 0 & 0 \end{bmatrix} \tag{3.3}$$

The forward kinematics problem is very complex for a parallel robot, while the inverse kinematics problem is extremely straightforward in general [44]. In this subsection, the forward kinematics solution is generated for the designed MPM.

Assuming that linear (υ_m) and angular (ω_m) velocities of the mobile platform and the actuated inputs of the actuators (θ_i, i =1, 2, 3), the position (x, y, z) and orientation (ϕ) of the mobile platform are solved using the forward kinematics.

In this research we assume that there is no slip in the wheels of mobile platform on all directions. As $\Delta t \rightarrow 0$, the velocities during this time interval can be considered as a constant, and referring to Figure 3.6(b).

$$\left|P^i P^{i+1}\right| \approx \Delta S_m = v_m.\Delta t,$$

$$\Delta x_m^i = x_m^{i+1} - x_m^i \approx \left|P^i P^{i+1}\right|.c\phi_m^i = v_m.\Delta t.c\phi_m^i,$$

$$\Delta y_m^i = y_m^{i+1} - y_m^i \approx \left|P^i P^{i+1}\right|.s\phi_m^i = v_m.\Delta t.s\phi_m^i,$$

$$\beta \approx \Delta\phi_m^i = \phi^{i+1} - \phi^i = \omega_m.\Delta t. \qquad 3.4$$

Since eq. 3.4 applied in all the motion of the mobile platform, we can delete the superscript *i*. Thus,

$$\dot{x}_m = \lim_{\Delta t\to 0}\left(\frac{\Delta x_m}{\Delta t}\right) = v_m.c\phi_m,$$
$$\dot{y}_m = lim_{\Delta t\to 0}\left(\frac{\Delta y_m}{\Delta t}\right) = v_m.s\phi_m, \qquad 3.5$$
$$\dot{\phi}_m = lim_{\Delta t\to 0}\left(\frac{\Delta \phi_m}{\Delta t}\right) = \omega_m.$$

Integration of eq. 3.5, gives the posture of the mobile platform:

$$x_m(t) = \int_0^t \dot{x}_m.dt = \int_0^t v_m(t).c(\phi_m(t)).dt,$$
$$y_m(t) = \int_0^t \dot{y}_m.dt = \int_0^t v_m(t).s(\phi_m(t)).dt, \qquad \mathbf{3.6}$$
$$\phi_m(t) = \int_0^t \dot{\phi}_m.dt = \int_0^t \omega_m(t).dt.$$

Let us assume that the wheels of mobile platform have no slip in any direction. Let $\mathbf{P} = [x \quad y \quad z]^T$ and ${}^B\mathbf{P} = [x_p \quad y_p \quad z_p]^T$ are the vectors of point *P* in the fixed frame *O* and the moving frame *B*, respectively. Also, in frame B, let ${}^B\mathbf{e}_i = \overrightarrow{BA_i}$, ${}^B\mathbf{b}_i = \overrightarrow{A_iB_i}$ and ${}^B\mathbf{c}_i = \overrightarrow{PC_i}$. Referring to Figure 3.1 we obtain

$${}^B\mathbf{e}_i = [ec\varphi_i \quad es\varphi_i \quad 0]^T,\ {}^B\mathbf{c}_i = [uc\varphi_i \quad us\varphi_i \quad 0]^T,$$
$${}^B\mathbf{b}_i = [bc\theta_i c\varphi_i \quad bc\theta_i s\varphi_i \quad bs\theta_i]^T \qquad 3.7$$

Where $\varphi_i = \frac{i\pi}{2}$, for $i = 1, 2, \text{and } 3$

Since the distance between B_i and C_i is a constant a, we have

$$\left\| {}^{B}\mathbf{P} + {}^{B}\mathbf{c}_i - {}^{B}\mathbf{e}_i - {}^{B}\mathbf{b}_i \right\| = a \qquad 3.8$$

By substituting eq. 3.7 in eq. 3.8, we get

$$x_p^2 + (y_p + u - e - bc\theta_1)^2 + (z_p - bs\theta_1)^2 = a^2, \qquad 3.9$$

$$(x_p - u + e + bc\theta_2)^2 + y_p^2 + (z_p - bs\theta_2)^2 = a^2, \qquad 3.10$$

$$x_p^2 + (y_p - u + e + bc\theta_3)^2 + (z_p - bs\theta_3)^2 = a^2, \qquad 3.11$$

In order to solve these equations, first we used Maple12 (codes appear in appendix A) to solve (3.9) and (3.10) to get x_p and y_p as a function of z_p, we get the following result:

$$\begin{aligned} x_p = -\Big(b\big(&2bc\theta_1 c\theta_3 w - bc\theta_2 wc\theta_1 - bc\theta_2 wc\theta_3 - w^2 c\theta_1 - w^2 c\theta_3 + 2w^2 c\theta_2 \\ &+ 2z_p s\theta_2 w + z_p bs\theta_1 c\theta_3 - wz_p s\theta_1 - wz_p s\theta_3 + bz_p c\theta_1 s\theta_3 \\ &- bz_p c\theta_1 s\theta_2 - bz_p c\theta_3 s\theta_2\big)\Big) / (-bwc\theta_1 - bwc\theta_3 - 2bwc\theta_2 \\ &+ b^2 c\theta_1 c\theta_2 + b^2 c\theta_3 c\theta_2 + 2w^2) \end{aligned}$$

$$y_p = -b(wc\theta_1 - wc\theta_3 + z_p s\theta_1 - z_p s\theta_3)/(-2w + bc\theta_1 + bc\theta_3)$$

Where $w = u - e$.

Now substitute x_p and y_p in (3.11) and solve for z_p, then solving eq. 3.10 and eq. 3.11 leads to solutions for the forward kinematics of the parallel robot, i.e.

$$x_p = f_1 z_p + f_0, y_p = e_1 z_p + e_0, z_p = \frac{-h_1 + \sqrt{h_1^2 - 4h_2 h_0}}{2h_2} \qquad 3.12$$

Where $e_1 = -\frac{n_1}{n_2}, e_0 = -\frac{n_0}{n_2}$,

$$f_1 = \frac{m_2 n_1}{m_3 n_2} - \frac{m_1}{m_3}, \qquad f_0 = \frac{m_2 n_0}{m_3 n_2} - \frac{m_0}{m_3},$$

$$h_2 = 1 + e_1^2 + f_1^2,$$

$$h_1 = 2[e_1(e_0 + u - e - bc\theta_1) + f_1 f_0 - bs\theta_1],$$

$$h_0 = (e_0 + u - e - bc\theta_1)^2 + f_0^2 - a^2 + b^2 s^2 \theta_1,$$

$$m_3 = u - e - bc\theta_2, m_2 = u - e - bc\theta_1,$$

$$m_1 = b(s\theta_1 - s\theta_1),$$

$$m_0 = b(u - e)(c\theta_2 - c\theta_1),$$

$$n_2 = 2(u - e) - b(c\theta_1 + c\theta_3),$$

$$n_1 = b(s\theta_1 - s\theta_3),$$

$$n_0 = b(u - e)(c\theta_3 - c\theta_1),$$

The parallel robot has only a translation motion, so the rotation around the Z_O axis is the only factor to determine the orientation of the MPM, i.e., $\phi = \phi_m$. Referring to Fig. 3.1, one can derive the position of the mobile platform to be:

$$\mathbf{p} = \mathbf{b} + \mathbf{R}\,{}^{B}\mathbf{p} \qquad 3.13$$

With

$$\mathbf{b} = \overrightarrow{OB} = [x_m - l_b c\phi_m \quad y_m - l_b s\phi_m \quad h]^T, \tag{3.14}$$

And

$$\mathbf{R} = \begin{bmatrix} c\phi_m & -s\phi_m & 0 \\ s\phi_m & c\phi_m & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

Is the rotation matrix of the moving frame *B* regarding to the fixed frame *O*.

3.2.3 Differential Kinematics Analysis

Let the vector for the output velocities of the moving platform to be $\dot{\mathrm{x}} = [\dot{\mathrm{x}} \quad \dot{\mathrm{y}}\dot{\mathrm{z}} \quad \dot{\phi}]^{\mathrm{T}}$, and the vector of input joint rates is represented by $\dot{\mathrm{q}} = [\dot{\theta}_{\mathrm{l}} \quad \dot{\theta}_{\mathrm{r}}\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^{\mathrm{T}}$. Differentiating eq. 3.13 with respect to time, leads to

$$\dot{\mathbf{p}} = \dot{\mathbf{b}} + \dot{\mathbf{R}}\,{}^{\mathrm{B}}\mathbf{p} + \mathbf{R}\,{}^{\mathrm{B}}\dot{\mathbf{p}} \tag{3.16}$$

Also, let $\dot{\mathbf{q}}_{\mathrm{p}} = [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3]^T$ to be the vector of actuated joint rates for the parallel robot. Taking the derivative of both sides of (3.9)-(3.11) with respect to time and rewriting them into a matrix form, yields

$$\mathbf{A}\,{}^{\mathrm{B}}\dot{\mathbf{p}} = \mathbf{B}\dot{\mathbf{q}}_{\mathrm{p}} \tag{3.17}$$

The 3×3 forward and inverse Jacobean matrices **A** and **B** of the parallel robot can be written as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix} \quad 3.18$$

With $a_{11} = x_p,\ \ a_{12} = y_p + u - e - bc\theta_1,\ \ a_{13} = z_p - bs\theta_1$,

$a_{21} = x_p - u + e + bc\theta_2$, $a_{22} = y_p$, $a_{23} = z_p - bs\theta_2$,

$a_{31} = x_p$, $a_{32} = y_p - u + e + bc\theta_3,\ \ a_{33} = z_p - bs\theta_3$,

$b_{11} = -b\left[(y_p + u - e)s\theta_1 - z_p c\theta_1\right],$

$b_{22} = b\left[(x_p - u + e)s\theta_2 - z_p c\theta_2\right],$

$b_{33} = b[(y_p - u + e)s\theta_3 - z_p c\theta_3]$

It can be derived from eq. 3.17, that when the parallel robot is away from the singularity

$$^{B}\dot{\mathbf{p}} = \mathbf{J}_p \dot{\mathbf{q}}_p \quad 3.19$$

Where $\mathbf{J}_p = \mathbf{A}^{-1}\mathbf{B}$ is the Jacobian matrix of a 3-RRPaR parallel robot.

By substituting eq. 3.14, 3.15 and 3.19 into eq. 3.16 and consider of eq. 3.1, will give in

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} \quad 3.20$$

Define **J** 4×5 to be the Jacobean matrix of the MPM, that relating the output velocities to the actuated joint rates and it can be rewritten as shown in the following matrix

$$\mathbf{J} = \begin{bmatrix} J_{11} & J_{12} & & & \\ J_{21} & J_{22} & \mathbf{RA^{-1}B} & & \\ 0 & 0 & & & \\ -r/d & r/d & 0 & 0 & 0 \end{bmatrix} \tag{3.21}$$

With $J_{11} = \left(\frac{1}{2} + \frac{y_p}{d}\right) rc\phi_m - \frac{r}{d} s\phi_m (l_b - x_b)$,

$$J_{12} = \left(\frac{1}{2} - \frac{y_p}{d}\right) rc\phi_m + \frac{r}{d} s\phi_m (l_b - x_b),$$

$$J_{21} = \left(\frac{1}{2} + \frac{y_p}{d}\right) rs\phi_m + \frac{r}{d} c\phi_m (l_b - x_b),$$

$$J_{22} = \left(\frac{1}{2} - \frac{y_p}{d}\right) rs\phi_m + \frac{r}{d} c\phi_m (l_b - x_b).$$

Taking differentiation eq. 3.20 with respect to time, gives

$$\ddot{\mathbf{x}} = \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{J}\ddot{\mathbf{q}} \tag{3.22}$$

Also, solving eq. 3.20, leads to

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + (\mathbf{I}_{5x5} - \mathbf{J}^{\dagger}\mathbf{J})\dot{\mathbf{q}}_{\mathbf{s}} \tag{3.23}$$

Where $J^{\dagger} = J^{T}(JJ^{T})^{-1}$ is the generalized pseudo inverse of J, and $\dot{\mathbf{q}}_{\mathbf{s}} \in \Re^{5x1}$ is an arbitrary vector which can be chosen to achieve a secondary task, this will be shown clearly in section 3.2.5.

3.2.4 Kinematic Singularity Characterization

The robot Jacobian allows motion and force transformation from the actuators to the End Effector, so the forces demand at a given point on the trajectory needed to be continuously checked for possible violation of the preset limits as the robot moves close to singularity. The condition number of the Jacobian is used as a local performance index for evaluating the velocity, accuracy, and rigidity mapping characteristics between the joint variables and the moving platform. In this research a detailed characterization of robot singularities is given as follows.

From equation (3.17) it is clear that singularity in the MPM structure occurs in the following cases:

1st case: $|A| = 0, \text{and } |B| \neq 0$. This corresponds to a type-1 singularity.

2nd case: $|\mathbf{A}| \neq 0, \text{and } |\mathbf{B}| = 0$. This is a type-2 singularity.

3rd case: $|A| = 0, \text{and } |B| = 0$. This is a type-3 singularity for which both the determinant of A and B will equal to zero.

These cases are programmed to be calculated during running the simulation.

3.2.5 Redundancy Resolution through Joint Limits and Singularity Avoidance

To include a secondary task criterion by a performance index g($\mathbf{q}$), $\dot{\mathbf{q}}_s$ in eq. 3.22 is chosen to be $\dot{\mathbf{q}}_s = \pm k\nabla g(\mathbf{q})$, where k is a positive real number and $\nabla g(\mathbf{q})$ the gradient of g($\mathbf{q}$), with positive sign indicating that the criterion is to be maximized and a negative sign indicating minimization.

To avoid joint limits we chose $\dot{\mathbf{q}}_s$ as follows:

$$\dot{q}_{s1} = (\mathbf{q}_{max} - \mathbf{q})\mathbf{W}(\mathbf{q} - \mathbf{q}_{min}) \quad 3.24$$

Where: $\mathbf{W} = \frac{1}{2}(\mathbf{q}_{max} - \mathbf{q}_{min})$ 3.25

The related criterion to avoid the singularity is to maximize the manipulability, i.e. we choose $\dot{\mathbf{q}}_s$ as follows:

$$\dot{\mathbf{q}}_{s2} = \sqrt{\det(J J')}\ \ W_s \quad 3.26$$

Where:W_s is weight vector with appropriate dimension.

Now the formula of the augmented function to avoid singularity and joint limits is as follow:

$$\dot{\mathbf{q}}_s = \dot{\mathbf{q}}_{s1} + \dot{\mathbf{q}}_{s2} \quad 3.27$$

Chapter 4

Kinematic Initialization

4.1Optimal time trajectory parameterization

Considering the MPM, the task of the robot is to move its end-effector within a limited workspace and time interval. Also, each robot joint has to produce zero rates and accelerations at the ends of the interval of motion. A cycloidal function is chosen to achieve this purpose for modeling the trajectory time (t) from 0 to T with the normalized time *s* as

$$s = \frac{t}{T}, \quad 0 \le t \le T, \quad 0 \le s \le 1,$$

The cycloidal function is described as follows:

$$q(s) = s - \frac{1}{2\pi}\sin(2\pi s), \quad 4.1$$

Where the first and second derivatives obtained as

$$\dot{q}(s) = 1 - \cos(2\pi s), \quad 4.2$$

$$\ddot{q}(s) = 2\pi \sin(s) \quad 4.3$$

The cycloidal motion and its derivatives are defined within the range (-1, 1). With zero velocity and acceleration at the ends of the interval, i.e. $s = 0$ and $s = 1$, the initial and final joint values be detailed as $\boldsymbol{q}^{I}$ and $\boldsymbol{q}^{F}$.

The maximum velocity for the motion of joint j is attained at the center of the interval, i.e. $s = 0.5$, the maximum being $\dot{q}_{max} = \dot{q}(0.5) = 2$, so that,

$$(\dot{q}_j)_{max}(s) = \frac{2}{T}\left|q_j^F - q_j^I\right| \quad 4.4$$

In the same way, it can be shown that the acceleration of joint j allows its maximum and minimum values at $s = 0.25$ and $s = 0.75$, the maximum being $\ddot{q}_{max} = \ddot{q}(0.25) = \ddot{q}(0.75) = 2\pi$, and hence

$$(\ddot{q}_j)_{max}(s) = \frac{2\pi}{T^2}\left|q_j^F - q_j^I\right| \quad 4.5$$

And finally, the maximum jerk of joint j is achieved at $s = 0.0$, and $s = 1.0$, the maximum being

$\ddot{q}_{max} = \ddot{q}(0) = \ddot{q}(1) = 4\pi^2$ Thus,

$$\left(\ddot{q}_j\right)_{max}(s) = \frac{4\pi^2}{T^3}\left|q_j^F - q_j^I\right| \quad 4.6$$

The motion of the robot is constrained by the maximum joint velocity, accelerations and jerk which the motors produce, this can be interpreted as

$$\dot{q}_j \le \left(\dot{q}_j\right)_{motor}, \quad \ddot{q}_j \le \left(\ddot{q}_j\right)_{motor}, \text{and} \quad \dddot{q}_j \le \left(\dddot{q}_j\right)_{motor}, \quad 4.7$$

This means the strongest constraint among the $\left(\dot{q}_j\right)_{motor}$, $\left(\ddot{q}_j\right)_{motor}$ and $\left(\dddot{q}_j\right)_{motor}$ limits the minimum-time trajectory of joint *j*, which means that:

$$T_j = Max\left\{\frac{2\left|q_j^F - q_j^I\right|}{\left(\dot{q}_j\right)_{motor}}, \sqrt{\frac{2\pi}{\left(\ddot{q}_j\right)_{motor}}\left|q_j^F - q_j^I\right|}, \left(\frac{4\pi^2}{\left(\dddot{q}_j\right)_{motor}}\left|q_j^F - q_j^I\right|\right)^{1/3}\right\} \quad 4.8$$

Now, the overall minimum-time trajectory (for all the five joints together of the MPM) is written as

$$T_{Min} = Max\{T_1, T_2, T_3, T_4, T_5\} \quad 4.9$$

Thus, the resulting minimum-time trajectory characterized by joints position, velocity and acceleration is obtained as

$$\begin{aligned} \boldsymbol{q}(t) &= \boldsymbol{q}^{\mathbf{I}} + (\boldsymbol{q}^{\mathbf{F}} - \boldsymbol{q}^{\mathbf{I}})\left(\frac{t}{T_{Min}} - \frac{1}{2\pi}\sin\left(2\pi\frac{t}{T_{Min}}\right)\right) \\ \dot{\boldsymbol{q}}(t) &= \frac{\boldsymbol{q}^F - \boldsymbol{q}^I}{T_{Min}}\left(1 - cos\left(2\pi\frac{t}{T_{Min}}\right)\right) \\ \ddot{\boldsymbol{q}}(t) &= \frac{\boldsymbol{q}^{\mathbf{F}} - \boldsymbol{q}^{\mathbf{I}}}{T_{Min}^2}2\pi\sin\left(2\pi\frac{t}{T_{Min}}\right) \end{aligned} \quad 4.10$$

4.2 Neuro-fuzzy inverse kinematics

One of the most important challenges in robotics is the inverse kinematics problem which is the problem of finding the joint coordinates (q_1, q_2, q_3, q_4, q_5), from Cartesian coordinates (x, y, z, ϕ),Where the starting and ending Cartesian positions of the manipulator are specified in the workspace of the robot.

A neuro-fuzzy system called NeFIK is proposed to be used here for resolving the redundancy of the inverse kinematic problem. The NeFIK is going to be trained to produce joint position in a preferred configuration. The training dataset is generated with the forward kinematic (FK) equations of the manipulator described in chapter 3.

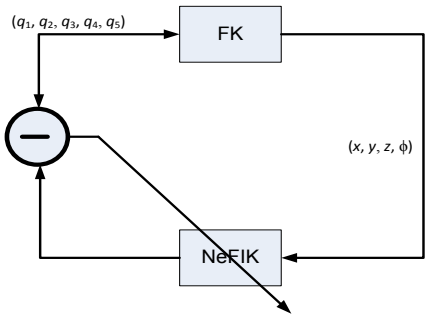


Figure 4.1: The use of the FK to the learning of the NeFIK module

A set of derivatives of (q_1, q_2, q_3, q_4, q_5) is used to construct the true derivatives (x, y, z, ϕ, x_n), and thus to get an error on which to apply the back-propagation algorithm. As mentioned here we add $\dot{x}_n$ related to $\dot{q}_s$to remove the redundancy of the system.

NeFIK is a multi-layer feed forward adaptive network. The first layer is a two input layer, characterizing the Cartesian position crisp values. The last layer is a three

output-layer characterizing the corresponding crisp joint values. NeFIK involves three hidden layers. The first one is the fuzzification layer, which transfers the crisp inputs to linguistic variables, through sigmoidal transfer functions. The second 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. 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 iteration as compared to standard back-propagation (e.g. gradient descent).

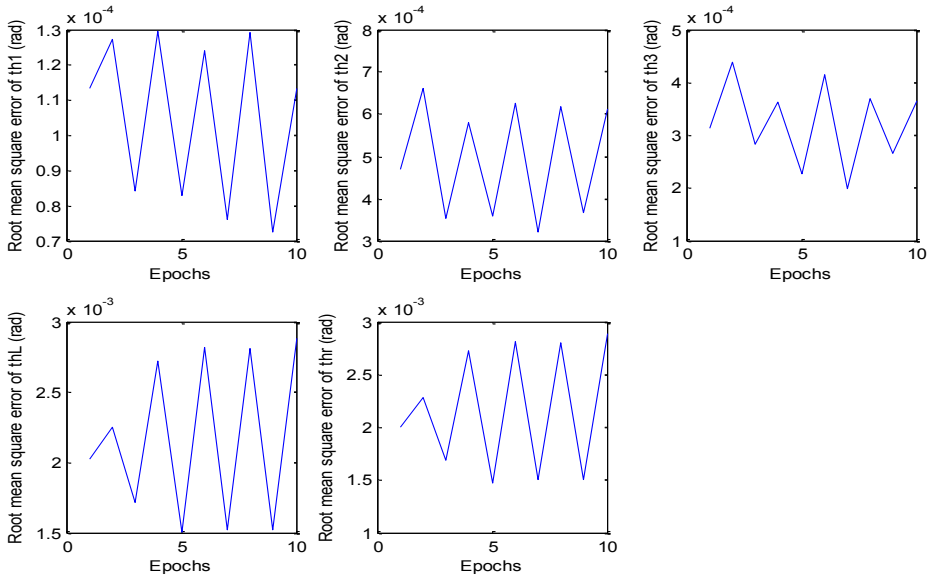


Figure 4.2: NeFIK performance – root mean square error output with respect to learning epochs for derivatives of (q_1, q_2, q_3, q_4, q_5)

To construct NeFIK, the forward kinematic equations are applied. Its learning is obtained through 400 samples, among which 320 are considered for training, whereas

testing and validation datasets, each of them is obtained using 10 entry samples. Fig. 4.2 shows the training performance for NeFIK, which is interesting as it reaches a very small root mean square error (RMSE), less than 10^{-3} in less than 10 epochs. It is noted that the configuration used for the learning is determined among infinitely many solutions that exist for each input.

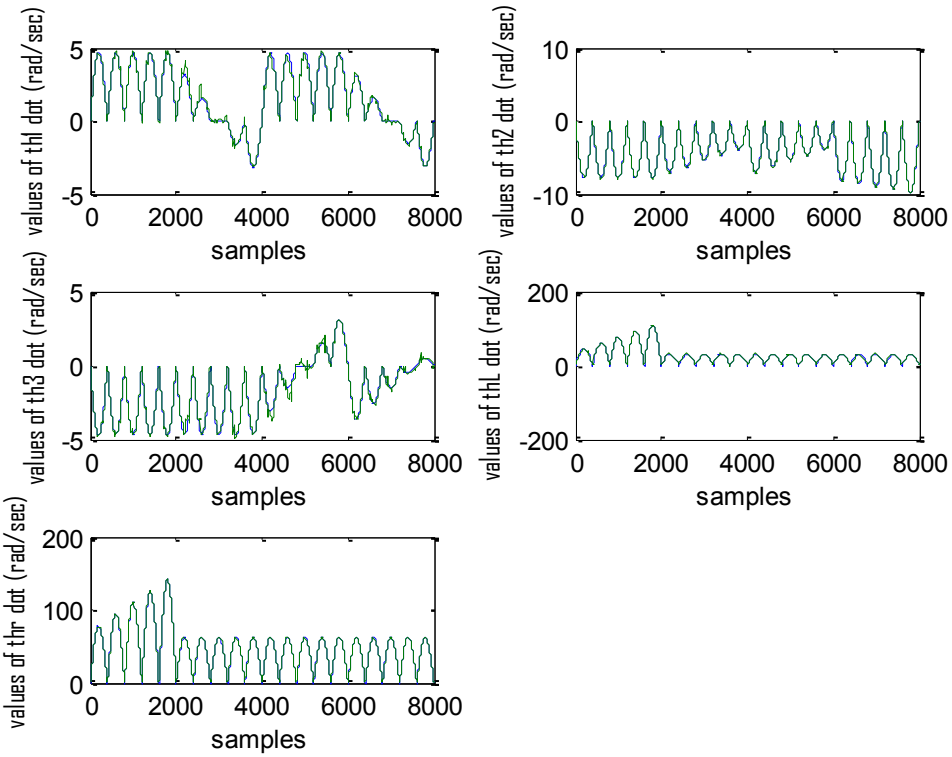


Figure 4.3: NeFIK performance – difference between real and estimated values of the MPR angles

Figure 4.3 shows the difference between the real and estimated values of the joint angles at the 8000 samples. The model derived by the NeFIK structure is used to

illustrate two trajectories motion of the MPR, in the first trajectory the motion is subjected to the parallel mechanism only (figure 4.4 and 4.5), and the combine motion of both the mobile and parallel structures are shown in figures 4.6 and 4.7.

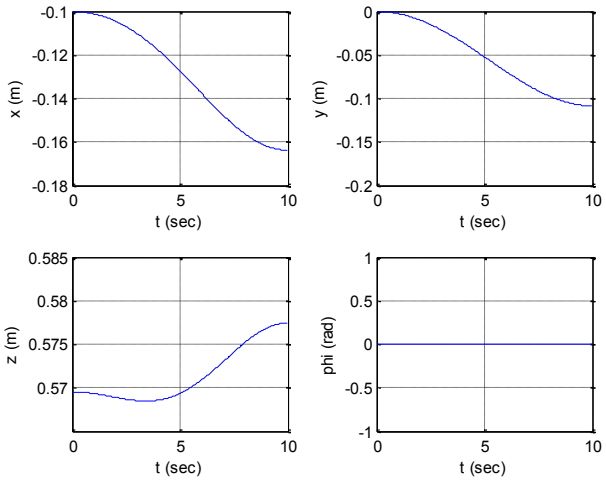


Figure 4.4: variation of the EEF coordinates for the motion of the parallel mechanism alone

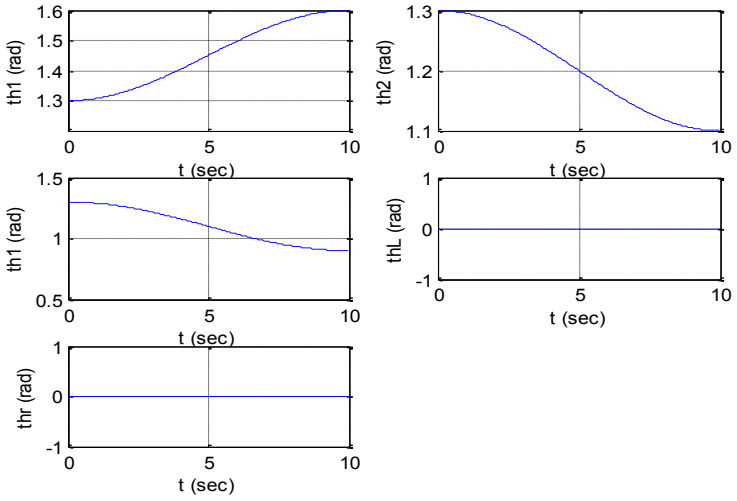


Figure 4.5: variation of the joint angles for the motion of the parallel mechanism alone

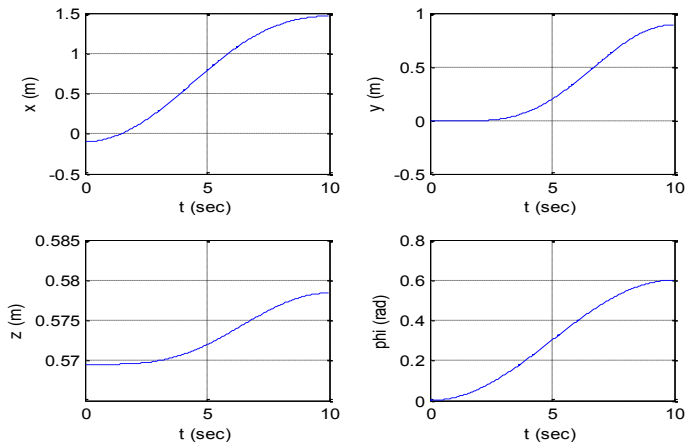


Figure 4.6: variation of the EEF coordinates for the motion of the MPR

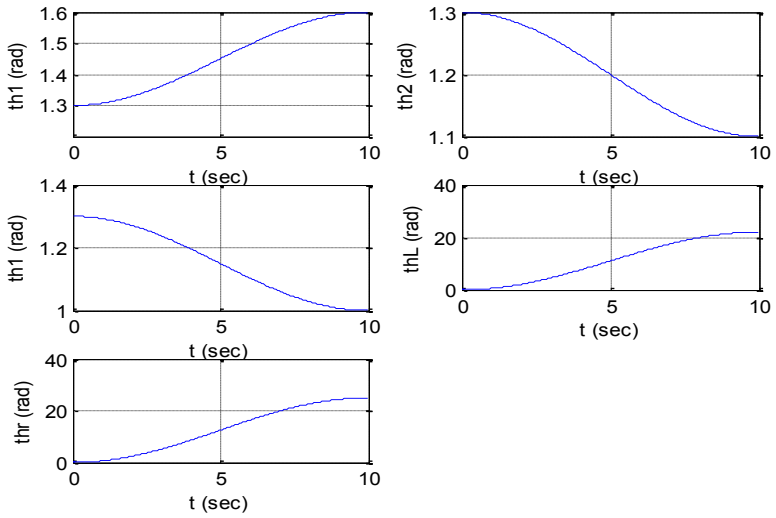


Figure 4.7: variation of the EEF coordinates for the motion of the MPR

4.3 Dynamic trajectory generation

The robot dynamic model is developed using a Lagrangian formalism, which includes actuators and friction models. This model allows closed-form expression of joint rates and accelerations characterizing the motion resulting from joint torques as in eq. (3.33). Now, using the minimum-time trajectory of Eq. (4.10) and the inverse dynamic solution of Eq. (4.33), one can write

$$\tau = \left[\frac{1}{T_{Min}^2}\left(2\pi \sin\left(2\pi \frac{t}{T_{Min}}\right)\right)\mathbf{D}(\mathbf{q}) + \frac{1}{T_{Min}}\left[\left(1 - \cos\left(2\pi \frac{t}{T_{Min}}\right)\right)(\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{F}v + \mathbf{F}c\ \mathrm{sgn}1 - \cos 2\pi t T\mathrm{Min}\mathbf{q}F - \mathbf{q}\mathrm{I}T \right. \right. \quad 4.11$$

Eq. (4.11) allows to compute the torques τ corresponding to the joint motion $(q, \dot{q}, \ddot{q})$ and then to project τ onto the admissible domain of torque limits (as provided by the manufacturer), i.e.

$$\tau_i = \mathrm{Max}(\mathrm{Min}(\tau_i, \tau_{i\,Max}), \tau_{i\,Min}) \quad 4.12$$

Chapter 5

Dynamic Modeling

In order to get the dynamic modeling of the hybrid MPM system the Lagrange method is used. This can be done by applying Lagrange equation to the mechanical systems with either holonomic or nonholonomic constraints, along with the equations of constraint and their first and second derivatives involved into the equations of motion to produce the number of equations that is equal to the number of unknowns.

Considering $\boldsymbol{\xi}$ which contains the variables of the mobile platform, Let the generalized coordinates to be $\boldsymbol{\zeta} = [\boldsymbol{\xi}^{T} \quad \theta_1 \quad \theta_2 \quad \theta_3 \quad x \quad y \quad z]^{T}$, notice that ζcontains all the variable of both the mobile platform and the parallel manipulator. In order to use the approach of Lagrangian equations for the derive the dynamic equations of the MPM, the kinetic and potential energies for all components of the manipulator must be expressed in terms of the chosen generalized coordinates and their derivatives. In this way we will get number of equations equal to the number of

the generalized coordinates (11 equations), later we will see how this helps to get the dynamic model.

5.1 Dynamic Model Analysis

Using the same simplification of [6] in this model while dealing with the mechanical structure. Concerning a 3-RRPaR parallel manipulator, the upper parallelogram links cause the complexity of the dynamic model. These connecting links can be made of light materials such as aluminum alloy, because of that the dynamic modeling can be simplified by the following hypotheses: The mass of each upper link is equally divided into two portions and placed at its two extremities, i.e., one half at its lower extremity (the end of lower link) and the other half at its upper extremity (moving platform). Thus, the rotational inertias of upper links can be neglected. Also, the castor of the mobile platform can be made to be very light, so its dynamics is neglected.

1) Constraint Equations: The mobile robot cannot move in the lateral direction, i.e., it satisfies the conditions of pure rolling and non-slipping. Then, the three constraints for the mobile platform can be represented by eq. 3.2.

Another three constraint equations for the MPM can be derive from eq. 3.8, i.e.

$$\Gamma_4 = x_p^2 + (y_p + u - e - bc\theta_1)^2 + (z_p - bs\theta_1)^2 - a^2,$$

$$\Gamma_5 = (x_p - u + e + bc\theta_2)^2 + y_p^2 + (z_p - bs\theta_2)^2 - a^2, \quad 5.1$$

$$\Gamma_6 = x_p^2 + (y_p - u + e + bc\theta_3)^2 + (z_p - bs\theta_3)^2 - a^2,$$

2) Dynamic Equations: because of moving on a horizontal plane, no change in the potential energy U_m of the mobile platform. While the kinetic energy can be calculated by:

$$T_m = \frac{1}{2}m_c\left(\dot{x}_m^2 + \dot{y}_m^2 + l_a^2\dot{\phi}_m^2\right) + \frac{1}{2}I_c\dot{\phi}_m^2 + \frac{1}{2}I_f\dot{\theta}_l^2 + \frac{1}{2}I_f\dot{\theta}_r^2 \quad 5.2$$

where m_c is the mass of cart including the mobile platform, the base platform and three actuators for the parallel robot, while without the two driving wheels and rotors of the two motors;

I_c is the moment of inertia of the mobile cart about a vertical axis through the mass center A;

I_f denotes the moment of inertia of each driving wheel and the motor rotor about the wheel axis.

The potential energy of the parallel manipulator is:

$$U_p = \left(m_p + 3m_a\right)z_p g + \sum_{i=1}^{3}\left(\frac{1}{2}m_b + m_a\right)bgs\theta_i \quad 5.3$$

Where, m_b, m_a, and m_p represent the mass of lower link, each connecting rod of upper link, and the moving platform, respectively.

The kinetic energy for the parallel manipulator consists of kinetic energy of the upper moving platform, the upper links, and the connecting rods. It is derived to be

$$T_p = \frac{1}{2}(m_p + 3m_a)(\dot{x}^2 + \dot{y}^2 + \dot{z}^2) + \frac{1}{2}I_p\dot{\phi}_m^2 + \frac{3}{2}(m_b + m_a)(\dot{x}_m^2 + \dot{y}_m^2 +$$

$$lb+e2\phi m2+i=131213mb+mab2\theta i \qquad 5.4$$

Where, I_p denotes the moment of inertia of the moving platform about a vertical axis through its mass center.

Thus, the Lagrange function for the MPM becomes

$$L = T_m + T_p - U_m - U_p \qquad 5.5$$

The constrained dynamics for the entire system of the MPM can be determined by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\zeta}_j}\right) - \frac{\partial L}{\partial \zeta_j} = Q_j + \sum_{i=1}^{6}\lambda_i\frac{\partial \Gamma_i}{\partial \zeta_j}, \quad (j = 1, 2, \dots, 11)$$

Or: $$Q_{\mathrm{j}} = \frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathrm{L}}{\partial \dot{\zeta}_{\mathrm{j}}}\right) - \frac{\partial \mathrm{L}}{\partial \zeta_{\mathrm{j}}} - \sum_{\mathrm{i}=1}^{6}\lambda_{\mathrm{i}}\frac{\partial \Gamma_{\mathrm{i}}}{\partial \zeta_{\mathrm{j}}}, \quad (j = 1, 2, \dots, 11) \qquad 5.6$$

Where $\boldsymbol{Q} = [0 \quad 0 \quad 0 \quad \tau_l \quad \tau_r \quad \tau_1 \quad \tau_2 \quad \tau_3 \quad 0 \quad 0 \quad 0]^{\mathrm{T}}$ are the generalized forces under the assumption of no external forces/torques exerted.

λ_{i} (i = 1, 2, …, 6) are Lagrange multipliers associated with the constraints eq. 3.2 and eq.3.22,

Now, the Lagrange multipliers can be calculated from the first set of linear equations of eq. 4.6 for j = 1, 2, 3, 9, 10, and 11.

Once the Lagrange multipliers are found, the actuated torques $\tau = [\,\tau_l \quad \tau_r \quad \tau_1 \quad \tau_2 \quad \tau_3]^T$ can be solved from the second set of equations of eq. 3.27 for j = 4, 5, 6, 7, and 8, which can be written into a matrix form:

$$\mathbf{H(\zeta)\ddot{q} + V(\zeta,\dot{\zeta})\dot{q} + G(\zeta) = \tau + C(\zeta)\lambda} \qquad 5.7$$

Here, Matlab is used to get complete expressions of eq. 5.7 as follows:

The dynamic parameters are: m_a = 0.2 kg, m_b = 0.5 kg, m_p = 0.8 kg, m_c = 7.5 kg, I_p = 0.00034 kg.m^2, I_c =0.13982 kg.m^2, and I_f = 0.00045 kg.m^2 [6].

Solving the augmented Lagrange equation gives:

$Q_i = \lambda_4\, g_{1j}(\zeta) + \lambda_5\, g_{2j}(\zeta) + \lambda_6\, g_{3j}(\zeta) + g_{4j}(\ddot{\zeta})$, where i=1, 2, 3, …, 11

$H(\zeta) = \mathrm{diag}([A_1,\ A_2,\ A_4, A_7\ , A_{10}\ ]) \in \Re^{5x5}$ is the symmetric and positive definite inertial matrix.

$A_1 = A_2 = 9/20000, \qquad A_4 = A_7 = A_{10} = 11/750$

$V(\zeta,\dot{\zeta}) \in \Re^{5x5}$ is the centripetal and Coriolis forces matrix, here equal to zero.

$G(\zeta) = [0;\ \ 0;\ \ A_5;\ \ A_8;\ \ A_{11}] \in \Re^{5x1}$ represents the vector of gravity forces,

$A_5 = 0.8829\ c\theta_1$, $A_8 = 0.8829\ c\theta_2$, $A_{11} = 0.8829\ c\theta_3$

$\lambda = [\lambda_1 \quad \lambda_2 \ldots \quad \lambda_6] \in \Re^{6x1}$ denotes the vector for Lagrange multipliers.

$$\lambda = \begin{bmatrix} f_1 & f_2 & f_3 \\ f_5 & f_6 & f_7 \\ f_9 & f_{10} & f_{11} \end{bmatrix}^{-1} \begin{bmatrix} f_4 \\ f_8 \\ f_{12} \end{bmatrix}, \text{ and,}$$

$f_1 = -0.4\ s\phi_m c\theta_1 - 0.08\ s\phi_m - 0.2\ c\phi_m + 2(x_m - x)$

$f_2 = -0.28\ c\phi_m - 0.4 c\phi_m c\theta_2 + 2(x_m - x)$

$f_3 = 0.08\ s\phi_m + 0.4\ s\phi_m c\theta_3 - 0.2\ c\phi_m + 2(x_m - x)$

$f_4 = 1.4\ \ddot{x}$

$f_5 = 2(y_m - y) - 0.2\ s\phi_m + 0.08\ c\phi_m + 0.4 c\phi_m c\theta_1$

$f_6 = 2(y_m - y) - 0.28\ s\phi_m - 0.4 s\phi_m c\theta_2$

$f_7 = 2(y_m - y) - 0.2\ s\phi_m - 0.08\ c\phi_m + -0.4 c\phi_m c\theta_3$

$f_8 = 1.4\ \ddot{y}$

$f_9 = -2z + 0.4 + 0.4\ s\theta_1$

$f_{10} = -2z + 0.4 + 0.4\ s\theta_2$

$f_{11} = -2z + 0.4 + 0.4\ s\theta_3$

$f_{12} = 1.4\,\ddot{z} + 13.734$

$\mathbf{C}(\zeta) = \begin{bmatrix} 000 & 0 & 0 & 0 \\ 000 & 0 & 0 & 0 \\ 000 & -A_3 & 0 & 0 \\ 000 & 0 & -A_6 & 0 \\ 000 & 0 & 0 & -A_9 \end{bmatrix} \in \Re^{5x6}$ is the parameter matrix for λ.

$A_3 = 0.4s\theta_1 s\phi_m(x - x_m) - 0.4\ s\theta_1 c\phi_m(y + y_m) + 0.016\ s\theta_1 + 0.4c\theta_1 z - 0.08c\theta_1$

$A_6 = 0.4s\theta_2 c\phi_m(x - x_m) + 0.4\ s\theta_2 s\phi_m(y - y_m) + 0.056\ s\theta_2 + 0.4c\theta_2 z - 0.08c\theta_2$

$A_9 = -0.4\ s\theta_3 s\phi_m(x - x_m) + 0.4\ s\theta_3 c\phi_m(y - y_m) + 0.016\ s\theta_3 + 0.4c\theta_3 z - 0.08c\theta_3$

Recalling eq. 3.20

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + (\mathbf{I_{5x5}} - \mathbf{J}^{\dagger}\mathbf{J})\dot{\mathbf{q}}_{\mathbf{s}}$$

Let $\mathbf{J_n} \in \Re^{5x1}$ be the normalized base of n($\mathbf{J}$) which is the null space of $\mathbf{J}$, then we have

$$\mathbf{J}\mathbf{J}_n = \mathbf{0}_{4x1}, \qquad \mathbf{J}_n^T\mathbf{J}_n = 1,$$

$$\mathbf{J}_n^T\mathbf{J}^{\dagger} = \mathbf{0}_{1x4}, \qquad \mathbf{J}_n\mathbf{J}_n^T = \mathbf{I}_{5x5} - \mathbf{J}^{\dagger}\mathbf{J} \qquad 5.8$$

By definition of $\mathbf{x}_n = \mathbf{J}_n^T\dot{\mathbf{q}}_s$ and taking eq. 3.29, eq. 3.30 and eq. 5.1 into consideration, it can be shown that

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{x}} + \mathbf{J}_n\dot{\mathbf{x}}_n$$

$$\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}\ddot{\mathbf{x}} - \mathbf{J}^{\dagger}\dot{\mathbf{J}}\mathbf{J}^{\dagger}\dot{\mathbf{x}} + \mathbf{J}_n\ddot{\mathbf{x}}_n - \mathbf{J}^{\dagger}\dot{\mathbf{J}}\mathbf{J}_n\dot{\mathbf{x}}_n \qquad 5.9$$

With the definition of $\mathbf{x}_E = [\mathbf{x}^T \quad \mathbf{x}_n{}^T]^T$, and $\mathbf{J}_E^\dagger = [\mathbf{J}^\dagger \quad \mathbf{J}_n]$ then substituting eq. 5.2 into eq. 3.28, we can get the derivation of the dynamic equations described in Cartesian space, which is described by the following equation:

$$\bar{\mathbf{H}}(\boldsymbol{\zeta})\ddot{\mathbf{x}}_\mathbf{E} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\dot{\mathbf{x}}_\mathbf{E} + \bar{\mathbf{G}}(\boldsymbol{\zeta}) = \bar{\boldsymbol{\tau}} \tag{5.10}$$

With $\bar{\mathbf{H}}(\zeta) = \mathbf{J}_E^{\dagger T}\mathbf{H}(\zeta)\mathbf{J}_E^\dagger$,

$$\bar{\mathbf{V}}(\zeta, \dot{\zeta}) = \mathbf{J}_E^{\dagger T}\left[\mathbf{V}(\zeta, \dot{\zeta}) - \mathbf{H}(\zeta)\mathbf{J}^{\dagger \mathrm{T}}\dot{\mathbf{J}}\right]\mathbf{J}_E^\dagger,$$

$$\bar{\mathbf{G}}(\zeta) = \mathbf{J}_E^{\dagger T}[\mathbf{G}(\zeta) - \mathbf{C}(\zeta)\lambda],$$

and $\bar{\tau} = \mathbf{J}_E^{\dagger T}\tau$.

5.1.1 Discrete Time Dynamic Model

From a state-space form of the continuous-time dynamic model of the MPM we obtain the approximate state space discrete-time model. By deleting the time index and the contact forces, from eq. 5.3 we obtain:

$$\ddot{\mathbf{x}}_\mathbf{E} = \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta})\bar{\boldsymbol{\tau}} - \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta})\left(\bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\dot{\mathbf{x}}_\mathbf{E} + \bar{\mathbf{G}}(\boldsymbol{\zeta})\right) \tag{5.11}$$

Let us use define the state $\boldsymbol{x}_1$, and its time derivative $\boldsymbol{x}_2$, such that $\boldsymbol{x}_1 = \mathbf{x}_E$ and $\boldsymbol{x}_2 = \dot{\mathbf{x}}_E$ i.e., $\boldsymbol{x} = [\boldsymbol{x}_1^T \quad \boldsymbol{x}_2^T]^T$, eq.5.8 is rewritten as

$$\bar{\mathbf{H}}(\boldsymbol{\zeta})\dot{\mathbf{x}}_\mathbf{2} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x}_\mathbf{2} + \bar{\mathbf{G}}(\boldsymbol{\zeta}) = \bar{\boldsymbol{\tau}} \tag{5.12}$$

Also, eq. 5.9 can be transformed to following form:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \mathbf{0}_{5x5} & \mathbf{I}_{5x5} \\ \mathbf{0}_{5x5} & \mathbf{0}_{5x5} \end{bmatrix} \boldsymbol{x} - \begin{bmatrix} \mathbf{0}_{5x1} \\ \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta}) \left(\bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) \boldsymbol{x_2} + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \right) \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{5x5} \\ \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta}) \end{bmatrix} \bar{\boldsymbol{\tau}} \qquad 5.13$$

Now, to obtain the discrete time dynamic model of the MPM, eq. 5.10 is expressed as following:

$$\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} - \mathbf{D}(\boldsymbol{x}) + \mathbf{B}(\boldsymbol{x})\bar{\boldsymbol{\tau}} \qquad 5.14$$

With

$$\mathbf{F} = \begin{bmatrix} \mathbf{0}_{5x5} & \mathbf{I}_{5x5} \\ \mathbf{0}_{5x5} & \mathbf{0}_{5x5} \end{bmatrix},$$

$$\mathbf{D}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_{5x1} \\ \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta}) \left(\bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}}) \boldsymbol{x_2} + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \right) \end{bmatrix},$$

$$\mathbf{B}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_{5x5} \\ \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta}) \end{bmatrix} \qquad 5.15$$

By defining the sampling period ash_k, such that $h_k < t < h_{k+1}$ and $\sum_{k=1}^{N} h_k = \mathrm{T}$, with being the total traveling time and the robot state is defined between two sampling points *k* and *k*+1 as

$$\boldsymbol{x}(\mathrm{t}) = \boldsymbol{x}(h_k), \qquad \text{for k} = 1, 2, \ldots, \mathrm{N}. \qquad 5.16$$

The discrete time model to modeleq.5.11is written as

$$\boldsymbol{x_{k+1}} = \boldsymbol{F_d}(h_k)\boldsymbol{x_k} - \boldsymbol{D_d}(\boldsymbol{x_k}, h_k) + \boldsymbol{B_d}(\boldsymbol{x_k}, h_k)\bar{\boldsymbol{\tau}}_{\boldsymbol{k}} \quad 5.17$$

With$\boldsymbol{F_d}, \boldsymbol{D_d}and\boldsymbol{B_d}$are the discrete equivalents to $\boldsymbol{F}, \boldsymbol{D},$ and $\boldsymbol{B}$ matrices, and described below.

$$\mathbf{F_d}(h_k) = \mathbf{F_d}(k+1, k) = \mathbf{e}^{\mathbf{Fh_k}} \cong \begin{bmatrix} \mathbf{I}_{5x5} & h_k\mathbf{I}_{5x5} \\ \mathbf{0}_{5x5} & \mathbf{I}_{5x5} \end{bmatrix}$$

$$\mathbf{D_d}(\boldsymbol{x_k}, h_k) = \int_0^{\mathbf{h_k}} \mathbf{F_d}(\boldsymbol{h_k} - \mathbf{t})\mathbf{G}(h_k - \mathbf{t})(\mathbf{D}(\mathbf{x_k}))\mathrm{dt}$$

$$\cong \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta}) \begin{bmatrix} \frac{h_k^2}{2}\mathbf{I}_{5x5} \\ h_k\mathbf{I}_{5x5} \end{bmatrix} (\bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\boldsymbol{x_2} + \bar{\mathbf{G}}(\boldsymbol{\zeta}))$$

$$\mathbf{B_d}(\boldsymbol{x_k}, h_k) = \int_0^{\mathbf{h_k}} \mathbf{F_d}(h_k - \mathbf{t})\mathbf{B}(\boldsymbol{x_k})\mathbf{dt} = \begin{bmatrix} \frac{h_k^2}{2}\mathbf{I_{5x5}} \\ h_k\mathbf{I_{5x5}} \end{bmatrix} \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta}) \quad 5.18$$

So, the discrete time state-space dynamic model of the MPM is rewritten in the final form:

$$\boldsymbol{x_{k+1}} = \begin{bmatrix} \mathbf{I}_{5x5} & h_k\mathbf{I}_{5x5} \\ \mathbf{0}_{5x5} & \mathbf{I}_{5x5} \end{bmatrix} \mathbf{x_k} - \begin{bmatrix} \frac{h_k^2}{2}\mathbf{I}_{5x5} \\ h_k\mathbf{I}_{5x5} \end{bmatrix} \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta})(\bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\boldsymbol{x_2} + \bar{\mathbf{G}}(\boldsymbol{\zeta}))$$

$$+ \begin{bmatrix} \frac{h_k^2}{2}\mathbf{I}_{5x5} \\ h_k\mathbf{I}_{5x5} \end{bmatrix} \bar{\mathbf{H}}^{-1}(\boldsymbol{\zeta})\boldsymbol{\tau_k} \quad 5.19$$

5.2 Constraints Modeling

The task of robotics simulation requires taking in the consideration many constraints, such as: the nominal values of kinematic and dynamic parameter, for example, the length of the link, velocities, accelerations, and also nominal torques which the actuators supported. These constraints are defined in joint space and in task space.

5.2.1 Robot Constraints

- Nonholonomic constraints: The mobile robot cannot move in the lateral direction, i.e., it satisfies the conditions of pure rolling and non-slipping. Then, the three constraints for the mobile platform can be represented by eq. 3.2.

- Dynamic state equations: These consists of eq. 5.16, which can be rewritten in the following formula:

$$\boldsymbol{x_{k+1}} = \mathbf{f_{d_k}}(\boldsymbol{x_k}, \boldsymbol{\tau_k}, \mathrm{h_k}) \qquad 5.20$$

- Limits on the intermediate lengths of links: expressed by eq. 3.22, from which the limits of the angles is found as: each angle of the parallel manipulator is between 0.65 and 1.65 radian, and for the mobile platform it is between -10 and 10 radian.
- Singularity avoidance: as described in section 3.2.4
- Torque limits:

Another major issue for trajectory planning is not violating the control torque limits. In this research we assume that the robot torques is belong to a bounded set $\mathbf{C} \subset \Re^N$ as shown in the following formula:

$$\mathbf{C} = \{\boldsymbol{\tau_k} \in \Re^{\mathrm{N}}, \quad \text{such that: } \boldsymbol{\tau_{min}} \leq \boldsymbol{\tau_k} \leq \boldsymbol{\tau_{max}}, \quad \mathrm{k} = 0, \dots, \mathrm{N}-1\} \qquad 5.21$$

- Sampling period limits:

Since the torque constraints bound indirectly the path traversal time, to achieve admissible solution to the optimal control problem the overall robot traveling time T should not be too small. Also in order to achieve system controllability, the sampling period must be smaller than the system smallest time mechanical constant between two control times. In this research time mechanical constant and limits of sampling periods are assumed to be available previously

Now, define H to be the sampling period:

$$\mathrm{H} = \{\mathrm{h_k} \in \Re^{+}, \quad \text{such that: } \mathrm{h_{min}} \leq \mathrm{h_k} \leq \mathrm{h_{max}}\} \qquad 5.22$$

5.2.2 Task and Workspace Constraints

Task and workspace constraints are basically geometric and kinematic, from which the size and shape of the manipulator workspace is determined. These constraints are expressed by imposing to the end effector (EE) to pass through a set of specified

poses. These constraints represent equality constraints and are written for simplicity as:

$$\mathbf{s}_1^l(\boldsymbol{x}) = \|\mathbf{p} - \mathbf{p}_l\| - T_{\text{PassThl}_\text{P}} = 0$$
$$\boldsymbol{s}_2^{\boldsymbol{l}}(\boldsymbol{x}) = \|\text{vect}(\mathbf{R}^\mathbf{T}\mathbf{R}_\mathbf{l})\| - T_{\text{PassThl}_\text{R}} = 0, \quad \text{l} = 1, \dots, \text{L} \qquad 5.23$$

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

$$\mathbf{g}_1(\boldsymbol{x}) = \boldsymbol{q}_{\text{Min}} - \boldsymbol{\Theta}(\boldsymbol{x}) \leq 0, \qquad \mathbf{g}_2(\boldsymbol{x}) = \boldsymbol{\Theta}(\boldsymbol{x}) - \boldsymbol{q}_{\text{Max}} \leq 0,$$
$$\mathbf{g}_3(\boldsymbol{\tau}) = \boldsymbol{\tau}_{Min} - \boldsymbol{\tau} \leq 0, \qquad \mathbf{g}_4(\boldsymbol{\tau}) = \boldsymbol{\tau} - \boldsymbol{\tau}_{Max} \leq 0, \qquad 5.24$$

Where $\boldsymbol{q}_{\text{Min/Max}}and\boldsymbol{\tau}_{\text{Min/Max}}$are for ($\theta_1$, θ_2, θ_3), all inequality constraints will be noted as $\mathbf{g}_j(\boldsymbol{x}, \boldsymbol{\tau}, h) \leq 0, j = 1, \dots, 4$, regardless if they depend only on state, control variables or both. Hence, we turn up with J = 12 inequality constraints, 2L equality constraints (imposed passages), and 6 equality constraints representing state dynamics equations.

5.3 Model Validation

To validate the effectiveness of the established dynamic model for the MPM, the dynamic control in task space is implemented by resorting to a model-based controller. Since the number of coordinates in task space is less than that in joint space, the proposed MPM possesses self motion with one degree of redundancy. In this research, a simple solution is presented to stabilize the redundant robotic system.

5.3.1 Model-based controller design

The desired trajectories, velocities and accelerations $(\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d)$ can be determined in advance, and the desired self motions $\mathbf{x}_{nd}, \dot{\mathbf{x}}_{nd}, \text{and} \ddot{\mathbf{x}}_{nd}$ can be selected so as to perform secondary tasks besides the one in task space. Here, the self motion is exploited to optimize the problem of minimizing $\{\dot{\mathbf{q}}^T, \dot{\mathbf{q}}\}$, subject to $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$.

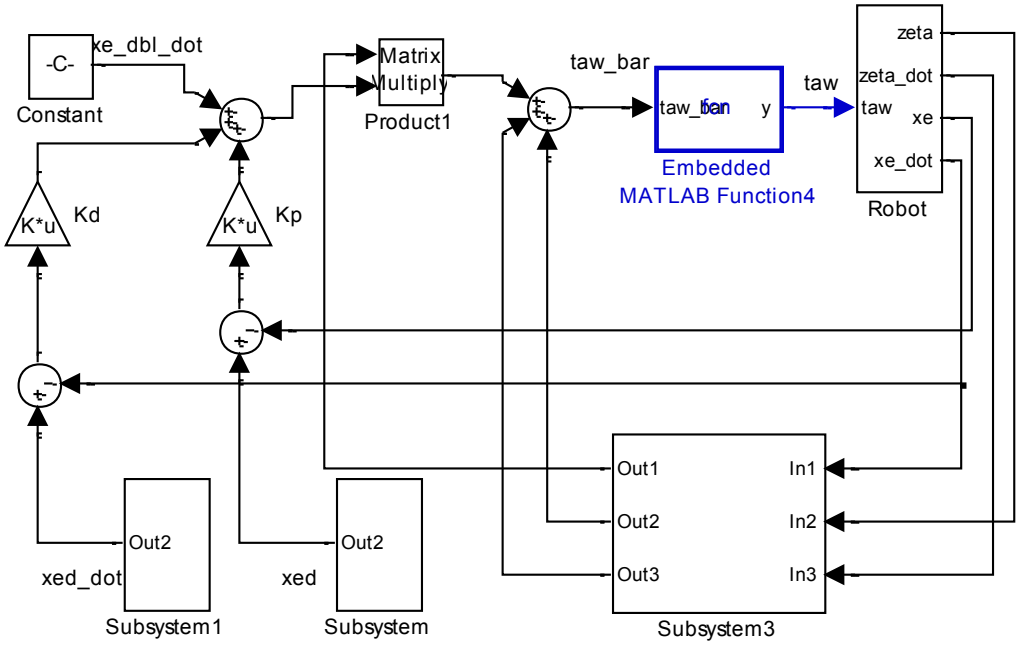


Figure 5.1: Block diagram of the model-based controller

Let $x_{Ed} = [x_d{}^T \quad x_{nd}{}^T]^T$, then the error system can be defined by

$$\mathbf{e} = \mathbf{x_{Ed}} - \mathbf{x_E} 5 \qquad 5.25$$

The adopted model-based controller is expressed in (5.36), and the control system block is illustrated in Fig. 5.1

$$\boldsymbol{\tau} = \mathbf{J}_E^T[\bar{\mathbf{H}}(\ddot{\mathbf{x}}_{Ed} + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e}) + \bar{\mathbf{V}}\dot{\mathbf{x}}_E + \bar{\mathbf{G}}] \quad 5.26$$

where $\mathbf{K}_D$ and $\mathbf{K}_P$ are positive definite constant gain matrices.

Substituting (5.35) into (5.3), allows the generation of error equations

$$\ddot{\mathbf{e}} + \mathbf{K}_D\dot{\mathbf{e}} + \mathbf{K}_P\mathbf{e} = 0 \quad 5.27$$

5.3.2 Simulation result for model validation

The dynamic control algorithm is implemented in task space such that the moving platform can track a desired trajectory, and the simulations are carried out via Matlab and Simulink software.

Two desired trajectory is selected such that no kinematic singularity is encountered. A linear locus shown in Fig. 5.2 and a parabola like special locus shown in Fig. 5.4 is considered in this simulation. Regarding the heading angle, it is assigned as ϕ_d = 0 in the first curve and 0.1t in the second. The architectural parameters of the designed MPM are: a = 0.2 m, b = 0.2 m, e = 0.16 m, u = 0.12 m, d = 0.4 m, h = 0.2 m, r = 0.08 m, l_a = 0.2 m and l_b = 0.1. The dynamic parameters are: m_a = 0.2 kg, m_b = 0.5 kg, m_p = 0.8 kg, m_c = 7.5 kg, I_p = 0.00034 kg.m^2, I_c =0.13982 kg.m^2, and I_f = 0.00045 kg.m^2. In the simulation, all parameters are supposed to be accurate enough. And the

actuated joint angles are initialized to be at home position. Additionally, the simulation time interval is selected as 10 seconds, and the gain matrices are selected as $\mathbf{K}_D$ = diag {10} and $\mathbf{K}_P$ = diag {25} [6].

Figures 5.2 and 5.4 show both the desired locus and the controlled one. And the position tracking errors and the heading angular tracking errors are illustrated in Fig. 5.3 and Fig. 5.5. It can be observed that both the position and heading angular errors can be eliminated by the proposed model-based controller. Moreover, if proper gains are chosen, the initial errors can be decreased rapidly.

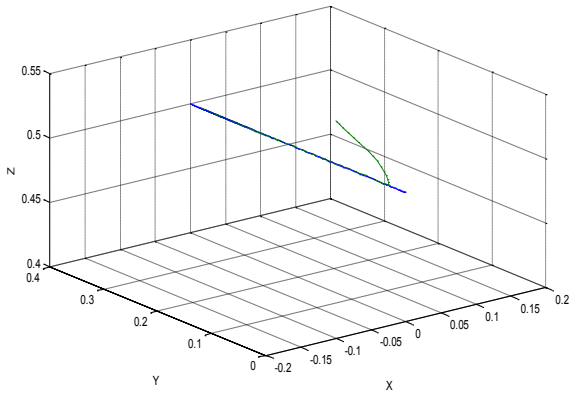


Figure 5.2: Desired and controlled loci for the linear trajectory

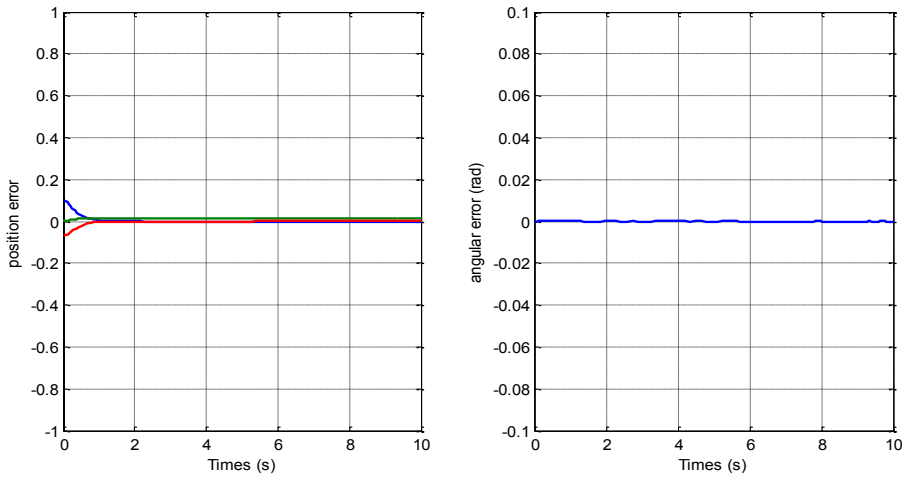


Figure 5.3: Position and angular tracking error for the linear trajectory

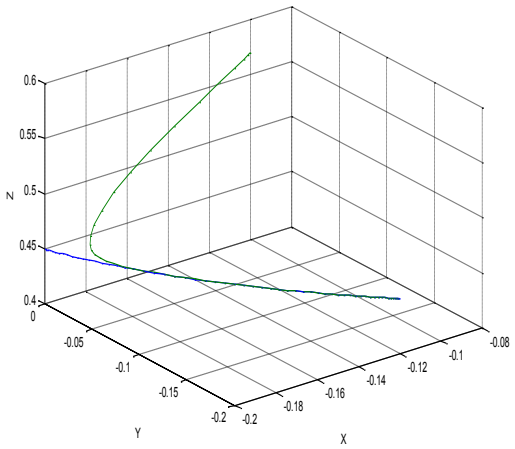


Figure 5.4: Desired and controlled loci for the curve trajectory

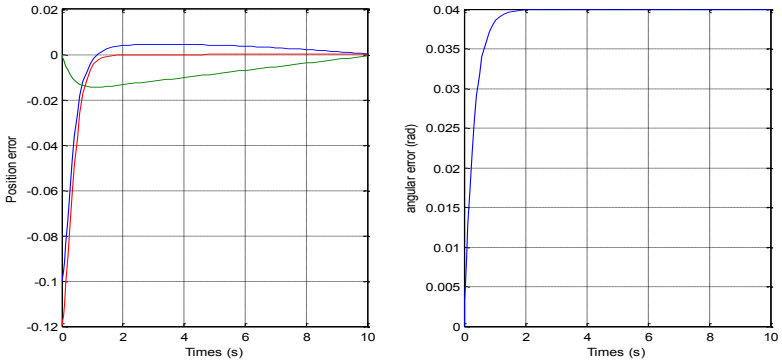


Figure 5.5: Position and angular tracking error for the curve trajectory

Two extra simulations are carried out to see the effect of choosing $\dot{q}_s$ in eq. 3.30, in the first $\dot{q}_s$ is put to equal to a vector of 0.001s and 0.0001 in the second, in each case we calculate the condition number of the Jacobean matrix J and plot it with time, clearly we find the value of $\dot{q}_s$ affect the condition number of J, in fig. 5.6 the condition number is increasing highly as time increase, while it becomes stable in fig. 5.7 around 1.8 which indicating good behavior.

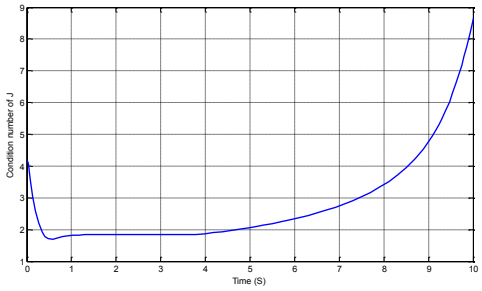


Figure 5.6: variation of the condition number for unstable system

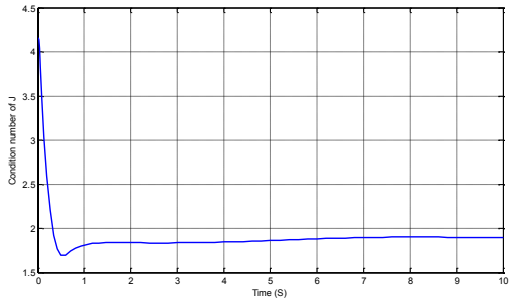


Figure 5.7: variation of the condition number for stable system

It should be noticed that by combining a mobile platform with a parallel robot, the problem of stability may occur since in some postures the external forces would cause the manipulator to topple. In addition, regarding the accurate navigation of the MPM, the odometric error containing both systematic and nonsystematic components should be taken into account for practical applications [6].

Chapter 6

Problem Formulation

In general, any cost function with a physical sense can be optimized, and in robotic several criteria have been implemented to obtain control optimization problem. The cost function can be defined according to task and planning objectives. The general objective function for a robot controlled in discrete time can be written in the following formula (P1):

$$E_d = F[\boldsymbol{x_N}] + \sum_{k=0}^{N-1} L[\boldsymbol{x_k}, \boldsymbol{\tau_k}] \quad 6.1$$

Where $F[\boldsymbol{x_N}]$ is a cost associated to the final state, whereas the second term the second term in the right-hand side of the equation is related to the instantaneous state

and control input variables (i.e. at time $t_k = kh$). The robot state and input vector $\boldsymbol{x}_k$ and $\boldsymbol{\tau}_k$ are related by the discrete dynamic model represented by eq.4.12.

6.1 Minimum Time Control Problem

The Minimum Time Control of robotic systems corresponds to $F = 0$, $L = 1$ in the mentioned criterion (P1) had been widely considered by several authors. This is of obvious interest considering production targets in industrial mass production process. But, the major disadvantage of this control method is its Bang-Bang character, which produces non smooth trajectories, which fastens the mechanical fatigue of the machine. The sampling periods are defined such that the overall robot traveling time is

$$T = \sum_{k=0}^{N-1} h_k \qquad 6.2$$

where h_k is the robot traveling time between two successive discrete configurations k and $k + 1$, $k = 0, \ldots, N\text{-}1$.

There are two basic approaches to the minimum time control problem:

1st Approach: in this approach we consider a fixed sampling period h and search for a minimum number N of discretizations of the trajectory. Which is equivalent to bring the robot from an initial configuration $\boldsymbol{x}_s$ to a final imposed one $\boldsymbol{x}_T$, within a minimum

number N of steps. For highly non linear and coupled mechanical systems like what we have with MPM it is impractical, even by using symbolic calculations.

2nd Approach: in this approach we consider a fixed number of discretizations N and vary the sampling time h_k. This is means that the robot moves from an initial configuration $\boldsymbol{x}_s$ to a final imposed one $\boldsymbol{x}_T$, within a fixed number of steps N while varying the sampling period's h_k.

6.2 Minimum Energy Control Problem

In this case while minimizing an electric energy cost, the robot moves from a starting point $\boldsymbol{x}_s$ to a target point $\boldsymbol{x}_T$, so, we obtain that

$$\mathrm{F} = 0 \text{ and} \mathrm{L} = \sum_{\mathrm{k}=0}^{\mathrm{N}-1} \boldsymbol{\tau}_{\mathbf{k}}^{\mathbf{T}} \mathrm{R} \boldsymbol{\tau}_{\mathbf{k}} \qquad 6.3$$

Using this criterion, or in general, using quadratic criterion, such as kinetic-energy criterion, ($L = \sum_{k=0}^{N-1} \mathbf{v}_k^T R \mathbf{v}_k$, $\mathbf{v}$ is the velocity vector), the obtained trajectory is smoother, as it away from discontinuous trajectories.

6.3 Redundancy Resolution and Singularity Avoidance Control Problem

Because of the redundancy robots, the Jacobian **J** is not a square matrix. The kinematic redundancy might be used to solve the inverse kinematics, by optimizing a secondary criterion. This was discussed previously in sections 3.2.4 and 3.2.5.

6.4 The Objective Functional for the Considered Problem

In this research the performance index considered, relates energy consumption, traveling time, and singularity avoidance. For time criterion, as shown in the previous section, there are two basic ways to perform optimization: The first one fixes the sampling period h and searches for a minimum number N of discretizations. The second one fixes the number of discretizations N and varies the sampling periods h_k. In this research, the number of sampling periods from an initial feasible kinematic solution is guessed. Then the sampling periods and the actuator torques are considered as control variables. In continuous-time, the constrained optimal control problem can be stated as follows:

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

$$E = \min_{\substack{\tau(t)\in C \\ t_o, t_T \in H}} \int_{t_o}^{t_T} \left\{ \left[\boldsymbol{\tau}(t)\mathbf{U}\boldsymbol{\tau}^{\mathbf{T}}(t) + t_1 + \frac{1}{2}\boldsymbol{x_2}(t)\mathbf{Q}\boldsymbol{x_2^T} + \delta\omega(\boldsymbol{x_1}(t)) \right] dt \right\} \qquad 6.4$$

Subject to constraints (5.14)-(5.21).

with $\mathbf{C}, H, \boldsymbol{U}, \boldsymbol{Q}, \mathbf{t}$ and $\boldsymbol{\delta}$ being, respectively, the set of admissible torques, the set of admissible sampling periods, electric energy, kinetic energy, and time weights, and a weight factor for singularity avoidance. The corresponding discrete-time optimal control problem consists of finding the optimal sequences $(\boldsymbol{\tau}_o, \boldsymbol{\tau}_2, \dots, \boldsymbol{\tau}_{N-1})$ and (h_o,

$h_2, \ldots, h_{N-1}$), allowing the robot to move from an initial state $\boldsymbol{x}_o = \boldsymbol{x}_S$ to a target state $\boldsymbol{x}_N = \boldsymbol{x}_T$, while minimizing the cost E_d:

$$\min_{\substack{\tau(t)\in C \\ t_o, t_T \in H}} \mathrm{E_d} = \left\{\sum_{k=0}^{N-1}\left[\boldsymbol{\tau_k}\mathbf{U}\boldsymbol{\tau_k^T} + \mathrm{t} + \boldsymbol{x_{2k}}\mathbf{Q}\boldsymbol{x_{2k}^T} + \delta\omega(\boldsymbol{x_{1k}}(\mathrm{t}))\right]\mathrm{h_k}\right\} \qquad 6.5$$

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

$$\mathbf{g}_j(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \leq 0,\ j = 1, \ldots, \mathrm{J},\ k = 0, \ldots, N-1$$

$$\boldsymbol{s}_j(\boldsymbol{x}_k) = 0, i = 1, \ldots, 2L,\ \ k = 0, \ldots, N$$

Chapter 7

Offline Trajectory Planning

7.1 Augmented Lagrangian Approach

For solving the stated Minimum Time-Energy Singularity-Free Trajectory Planning (MTE-SF-TP) n constrained on-linear control problem there are two basic approaches; which are: dynamic programming and variational calculus through the Maximum principle of Pontryagin. In the dynamic programming is used to find a

global optimal control. The optimal feedback control through Hamilton-Jacobi-Bellman partial differential equations (HJB-PDE) [67].

For linear-quadratic regulator problems, the HJB-PDE can be solved analytically or numerically by solving either an algebraic or dynamic matrix Riccati equation. For a general case, however, the PDE can be solved numerically for very small state dimension only [68].

Adding the inequality constraints on state and control variables makes the problem harder. In this research we propose to use the second approach [69] to solve this problem. The Augmented Lagrangean (AL) is used to solve the resulting non linear and non convex constrained optimal control problem.

Powell and Hestens originated independently the method of using the AL [70], [71]. The AL function transforms the constrained problem into a non-constrained one, where the degree of penalty for violating the constraints is regulated by penalty parameters. After that, several authors improved it [72; 73; 74; 75].

Moreover, AL might be convexified to some extent with a judicious choice of the penalty coefficients [76]. This procedure had been previously implemented by the first author on several cases of robotic systems [77]. The AL function transforming the constrained optimal control problem into an unconstrained one is written as:

$$L_{\mu}(\boldsymbol{x}, \boldsymbol{\tau}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) = \sum_{k=1}^{N} \left[\boldsymbol{\tau_k^T U \tau_k} + \boldsymbol{t} + \boldsymbol{x_{2k}^T Q x_{2k}} + \boldsymbol{\delta} \omega(\boldsymbol{x_{1k}}) \right] h_k$$

$$+\sum_{k=0}^{N-1}\left\{\lambda_{k+1}^{T}\left(\boldsymbol{x_{k+1}}-\boldsymbol{f}_{\boldsymbol{d_k}}(\boldsymbol{x_k},\boldsymbol{\tau_k},h_k)\right)\right\}+\sum_{k=0}^{N-1}h_k\left[\sum_{l=1}^{L-1}\sum_{i=1}^{2}\boldsymbol{\psi}_{\boldsymbol{\mu_s}}\left(\boldsymbol{\sigma_k^i},\boldsymbol{s_i^l}(\boldsymbol{x_k})\right)+\right.$$

$$\left.\sum_{j=1}^{J}\boldsymbol{\phi}_{\boldsymbol{\mu_g}}\left(\boldsymbol{\rho_k^j},\boldsymbol{g_j}(\boldsymbol{x_k},\boldsymbol{\tau_k},h_k)\right)\right]+\sum_{i=1}^{2}h_N\boldsymbol{\psi}_{\boldsymbol{\mu_s}}\left(\boldsymbol{\sigma_k^i},\boldsymbol{s_i^l}(\boldsymbol{x_k})\right) \quad 7.1$$

where the function $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k,\boldsymbol{\tau}_k,h_k)$ is defined by the discrete state eq. (5.25) at the sampling time k, N is the total sampling number, $\boldsymbol{\lambda}\in R^{12\mathrm{N}}$ designates the ajoint (or co-state) obtained from the adjunct equations associated to state equations, $\boldsymbol{\rho},\boldsymbol{\sigma}$ are Lagrange multipliers with appropriate dimensions, associated to equality and inequality constraints and $\boldsymbol{\mu}_g,\boldsymbol{\mu}_s$ 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:

$$\boldsymbol{\psi}_{\boldsymbol{\mu_s}}(\boldsymbol{a},\boldsymbol{b})=\left(\boldsymbol{a}+\frac{\boldsymbol{\mu_s}}{2}\boldsymbol{b}\right)^{\boldsymbol{T}}\boldsymbol{b},\text{and}\boldsymbol{\phi}_{\boldsymbol{\mu_g}}=\frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\mu_g}}\left\{\left\|\mathrm{Max}\left(0,\boldsymbol{a}+\boldsymbol{\mu_g}\boldsymbol{b}\right)\right\|^{\mathbf{2}}-\|\boldsymbol{a}\|^{\mathbf{2}}\right\} \quad 7.2$$

Where $\boldsymbol{a}$ and $\boldsymbol{b}$ refer respectively to Lagrange multipliers and the left hand side of equality and inequality constraints.

The requirements for the Karush-Kuhn-Tucker first order optimality necessary conditions that, there must exist some positive Lagrange multipliers $(\boldsymbol{\lambda}_k,\boldsymbol{\rho}_k)$, unrestricted sign multipliers $\boldsymbol{\sigma}_k$, and finite positive penalty coefficients $\left(\boldsymbol{\mu}_g,\boldsymbol{\mu}_s\right)$, for $\boldsymbol{x}_k,\boldsymbol{\tau}_k,h_k$, k=0, …, N to be solution to the problem, such that:

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

$$\boldsymbol{\rho}_{\boldsymbol{k}}^{\boldsymbol{T}} g(\boldsymbol{x}, \boldsymbol{\tau}, h) = 0, \quad \boldsymbol{\sigma}_{\boldsymbol{k}}^{\boldsymbol{T}} s(x) = 0, \quad g(\boldsymbol{x}, \boldsymbol{\tau}, h) \leq 0 \tag{7.3}$$

Applying of these conditions allows deriving the iterative formulas to solve the optimal control problem by adjusting control variables, Lagrange multipliers as well as penalty coefficients and tolerances. But, existing of the inverse of the total inertia matrix $\bar{\mathbf{H}}^{-1}$of the MPM in equation (5.24), $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$, including struts and actuators, as well as their Coriolis and centrifugal wrenches $\bar{\mathbf{V}}(\zeta, \dot{\zeta})$. These might very long to display contains. 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.

7.2 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 [78].

Theorem:

Provided that the inertia matrix is invertible, then the control law in the Cartesian space is defined as:

$$\mathbf{u} = \bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x_2} + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \tag{7.4}$$

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

$$\dot{\mathbf{x}}_{\mathbf{2}} = \boldsymbol{v} \tag{7.5}$$

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

This follows simply by substituting the proposed control law (7.4) into the dynamic model (4.12). One gets

$$\bar{\mathbf{H}}(\boldsymbol{\zeta})\dot{\mathbf{x}}_{\mathbf{2}} = \bar{\mathbf{H}}(\boldsymbol{\zeta})\boldsymbol{v} \tag{7.6}$$

Since $\bar{\mathbf{H}}(\zeta)$ 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:

$$\mathbf{x}_{\mathrm{k+1}} = \mathbf{F}_{\mathrm{dk}}\mathbf{x}_{\mathbf{k}} + \mathbf{B}(h_k)(\boldsymbol{v}_k) = \boldsymbol{f}_{\boldsymbol{d_k}}^{\boldsymbol{D}}(\mathbf{x}_k, \boldsymbol{v}_k, h_{\mathrm{k}}) \tag{7.7}$$

with

$$\boldsymbol{f}_{d_k}^{D}(\mathbf{x}_k, \boldsymbol{v}_k, h_k) = \begin{bmatrix} \mathbf{I}_{5\mathrm{x}5} & h_k\mathbf{I}_{5\mathrm{x}5} \\ \mathbf{0}_{5\mathrm{x}5} & \mathbf{I}_{5\mathrm{x}5} \end{bmatrix}\mathbf{x}_k - \begin{bmatrix} \frac{h_k^2}{2}\mathbf{I}_{5\mathrm{x}5} \\ h_k\mathbf{I}_{5\mathrm{x}5} \end{bmatrix}\boldsymbol{v}_k$$

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.

One problem of this formula which is the Euler's method is less of accuracy, in order to improve the accuracy, and because the MPM structure contains highly nonlinear equations as shown in the previous chapters, we use the Adams-Bashforth Formula given by the following general formula:

$$y_{i+1} = y_i + \frac{h}{2}(3f_i - f_{i-1}) + \frac{5}{12}h^3\acute{f}_i \quad 7.8$$

Now, applying Adams-Bashforth Formula eq. 7.7 into the dynamic equation 7.7 will gives:

$$x_{1_{k+1}} = x_{1_k} + 1.5h_k x_{2_k} + 0.5h_k x_{2_{k-1}} + \frac{5h_k^3}{12} v_{2_k} \quad 7.9$$

$$x_{2_{k+1}} = x_{2_k} + 1.5h_k v_{2_k} + 0.5h_k v_{2_{k-1}} + \frac{5h_k^3}{12} \dot{v}_{2_k} \quad 7.10$$

Since it is difficult to get the derivative of $\boldsymbol{v}_k$, To improve the accuracy the following formulas from numerical differentiation methods are used:

$$\dot{y}_1 = (y_2 - y_1)/h_1 \quad 7.11$$

$$\dot{y}_2 = (y_3 - y_1)/2h_2 \quad 7.12$$

$$\dot{y}_k = (-y_{k+2} + 8y_{k+1} - 8y_{k-1} + y_{k-2})/12h_k \quad 7.13$$

$$\dot{y}_{N-1} = (y_N - y_{N-2})/2h_2 \quad 7.14$$

$$\dot{y}_N = (y_N - y_{N-1})/h_2 \quad 7.15$$

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

$$\mathrm{E}_{\mathrm{d}}^{\mathrm{D}} = \mathrm{Min}_{\substack{\mathrm{v}\in\mathrm{V} \\ \mathrm{h_k}}} \left\{ \sum_{\mathrm{k}=0}^{\mathrm{N}-1} \left[\left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \right]^{\mathrm{T}} \mathbf{U} \left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \right. \right. \right.$$

$$\left. \left. \left. \bar{\mathbf{G}}(\boldsymbol{\zeta}) \right] + \mathbf{t} + \mathbf{x}_{2k}^{\mathrm{T}} \mathbf{Q} \mathbf{x}_{2\mathrm{k}} + \boldsymbol{\delta\omega}(\mathbf{x}_{1\mathrm{k}}) \right] \mathrm{h_k} \right\} \qquad 7.16$$

under the above mentioned constraints, which remain the same, except actuator torques, which become:

$$\boldsymbol{\tau}_{\min} \leq \bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \leq \boldsymbol{\tau}_{\max} \qquad 7.17$$

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

$$\mathbf{g}_3^{\mathrm{D}}(\mathbf{x}_{\mathrm{k}}, \mathbf{v}_{\mathrm{k}}) = \boldsymbol{\tau}_{\min} - \left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \right] \leq 0 \qquad 7.18$$

$$\mathbf{g}_{\mathbf{3}}^{\mathbf{D}}(\mathbf{x}_{\mathrm{k}}, \mathbf{v}_{\mathrm{k}}) = \left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta}, \dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta}) \right] - \boldsymbol{\tau}_{\max} \leq 0 \qquad 7.19$$

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

$$\underset{\substack{v\in\mathrm{V} \\ h_k}}{\mathrm{Min}} E_d^D$$

subject to $\quad \boldsymbol{x}_{k+1} = \boldsymbol{f}_{d_k}^D(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k), \quad k = 0, \dots, N-1$

$$\boldsymbol{g}_j^D(\boldsymbol{x}_k, \boldsymbol{v}_k, h_k) \leq 0, \quad j \in \{1, 2, \dots, \mathrm{J}\}$$

$$\mathbf{s}_{\mathrm{j}}^{\mathrm{D}}(\mathbf{x}_{\mathrm{k}}) = 0, \quad \mathrm{i} \in \{1, 2, \dots, \mathrm{I}\}, \quad \mathrm{k} = 0, \dots, \mathrm{N} \qquad 7.20$$

7.3 Augmented Lagrangian for the Decoupled Formulation

Now, the augmented Lagrangian associated to the decoupled formulation (P)

$$L_{\boldsymbol{\mu}}^{\mathrm{D}}(\boldsymbol{x}, \boldsymbol{v}, h, \boldsymbol{\lambda}, \boldsymbol{\rho}, \boldsymbol{\sigma}) =$$

$$\sum_{k=0}^{N-1} \left\{ \left[\left[\bar{\mathbf{H}}(\zeta)\boldsymbol{v} + \bar{\mathbf{V}}(\zeta, \dot{\zeta})\boldsymbol{x}_2 + \bar{\mathbf{G}}(\zeta) \right]^T U \left[\bar{\mathbf{H}}(\zeta)\boldsymbol{v} + \bar{\mathbf{V}}(\zeta, \dot{\zeta})\boldsymbol{x}_2 + \bar{\mathbf{G}}(\zeta) \right] + \mathbf{t} \right. \right.$$

$$\left. \left. + \boldsymbol{x}_{2k}^T \boldsymbol{Q} \boldsymbol{x}_{2k} + \boldsymbol{\delta}\omega(\boldsymbol{x}_{1k}) \right] h_k \right\} + \sum_{k=0}^{N-1} \left\{ \boldsymbol{\lambda}_{k+1}^T \left(\boldsymbol{x}_{k+1} - \boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \right) \right\}$$

$$+ \sum_{k=0}^{N-1} h_k \left[\sum_{l=1}^{L-1} \sum_{i=1}^{2} \psi_{\boldsymbol{\mu}_s} \left(\boldsymbol{\sigma}_k^i, \boldsymbol{s}_i^{Dl}(\boldsymbol{x}_k) \right) + \sum_{j=1}^{J} \phi_{\boldsymbol{\mu}_g} \left(\boldsymbol{\rho}_k^j, \boldsymbol{g}_j^D(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k) \right) \right]$$

$$+ \sum_{\mathrm{i}=1}^{2} \mathrm{h_N} \boldsymbol{\psi}_{\mu_\mathrm{s}} \left(\boldsymbol{\sigma}_{\mathrm{N}}^{\mathrm{i}}, \mathbf{s}_{\mathrm{i}}^{\mathrm{Dl}}(\mathbf{x}_{\mathrm{k}}) \right) \qquad 7.21$$

where the function $\boldsymbol{f}_{d_k}(\boldsymbol{x}_k, \boldsymbol{\tau}_k, h_k)$is defined by eq. (7.7) at time *k*, *N* is the total sampling number, other parameters appearing in (7.15) are defined above.

The first order Karush-Kuhn-Tucker optimality necessary conditions require that for $\boldsymbol{x}_k, \boldsymbol{v}_k, h_k, \quad k = 0, \ldots, N$ 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 (7.3) are satisfied for the decoupled formulation.

The co-states $\boldsymbol{\lambda}_k$ are determined by backward integration of the adjunct state equation yielding:

$$\lambda_{k-1} = -2h_k \frac{\partial[\bar{\mathbf{H}}(\boldsymbol{\zeta})\boldsymbol{v}+\bar{\mathbf{V}}(\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\mathbf{x}_2+\bar{\mathbf{G}}(\boldsymbol{\zeta})]}{\partial x_k} \mathbf{U}\left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta})\right] -$$

$$2\mathbf{Q}\mathbf{x}_{2k}h_k - \delta\nabla_{x_{1k}}\omega(\mathbf{x}_{1k}) - \mathbf{F}_d^T\lambda_k - h_k\left[\sum_{l=1}^{L-1}\sum_{i=1}^{2}\nabla_{x_k}\psi_{\mu_s}\left(\boldsymbol{\sigma}_k^i, s_i^{Dl}(\mathbf{x}_k)\right)\right] -$$

$$h_k\left[\sum_{j=1}^{J}\nabla_{x_k}\phi_{\mu_g}\left(\boldsymbol{\rho}_k^j, \mathbf{g}_j^D(\mathbf{x}_k, \boldsymbol{v}_k, h_k)\right)\right] \qquad 7.22$$

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

$$\nabla_{hk}L_\mu^D = \left[\left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\boldsymbol{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta})\right]^{\mathbf{T}} \mathbf{U}\left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\boldsymbol{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \right.\right.$$

$$\left.\left.\bar{\mathbf{G}}(\boldsymbol{\zeta})\right]\mathbf{x}_{2k}^T\mathbf{Q}\mathbf{x}_{2k} + \mathbf{t} + \delta\omega(\mathbf{x}_{1k})\right] + \sum_{l=1}^{L-1}\sum_{i=1}^{2}\psi_{\mu_s}\left(\boldsymbol{\sigma}_k^i, s_i^{Dl}(\mathbf{x}_k)\right) +$$

$$\sum_{j=1}^{J}\phi_{\mu_g}\left(\rho_k^j, g_j^D(\mathbf{x}_k, \boldsymbol{v}_k)\right) \qquad 7.23$$

The gradient of the Lagrangian with respect to acceleration variables is

$$\nabla_{vk}L_\mu^D = 2\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{U}^T\left[\bar{\mathbf{H}}(\boldsymbol{\zeta})\boldsymbol{v} + \bar{\mathbf{V}}(\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\mathbf{x}_2 + \bar{\mathbf{G}}(\boldsymbol{\zeta})\right]h_k + \mathbf{Z}_k^T\lambda_k$$

$$+ h_k\left[\sum_{j=1}^{J}\nabla_{v_k}\boldsymbol{\phi}_{\mu_g}\left(\boldsymbol{\rho}_k^j, \mathbf{g}_j^D(\mathbf{x}_k, \boldsymbol{v}_k, h_k)\right)\right] \qquad 7.24$$

where $\boldsymbol{Z}_k = \begin{bmatrix}\mathbf{I}_{5x5} & h_k\mathbf{I}_{5x5} \\ \mathbf{0}_{5x5} & \mathbf{I}_{5x5}\end{bmatrix}\boldsymbol{x}_k - \begin{bmatrix}\frac{h_k^2}{2}\mathbf{I}_{5x5} \\ h_k\mathbf{I}_{5x5}\end{bmatrix}\boldsymbol{v}_k, \qquad k = 0, 2, \ldots, N\text{-}1$

In the previous equations $\frac{\partial[\bar{\mathbf{H}}(\boldsymbol{\zeta})\mathbf{v}+\bar{\mathbf{V}}(\boldsymbol{\zeta},\dot{\boldsymbol{\zeta}})\mathbf{x}_2+\bar{\mathbf{G}}(\boldsymbol{\zeta})]}{\partial\mathbf{x}_k}$, $\nabla_{\mathbf{x}_{1k}}\boldsymbol{\omega}(\mathbf{x}_{1k})$, $\nabla_{\mathbf{x}_k}\boldsymbol{\phi}_{\mu_g}$, and $\nabla_{\mathbf{v}_k}\boldsymbol{\phi}_{\mu_g}$ are calculated using numerical differentiation formulas in equation 7.11 – 7.15.

7.4 Implementation issues

7.4.1 Initial solution:

To fasten convergence of AL algorithm ─ although it converges even if it starts from an unfeasible solution ─ a kinematic-feasible solution is defined. It is based on a optimal time trajectory parameterization. The initial time discretizations is assumed an equidistant grid for convenience, i.e.

$$h_k = t_{k+1} - t_k = \frac{t_f - t_0}{N}, k = 1, 2, \dots, N-1 \qquad 7.25$$

Upon this parameterized minimum time trajectory, a model predictive planning is built in order to achieve a good initial solution for the AL.

At the calculation of the inertia matrix and Coriolis and centrifugal dynamics components, we can use the approach developed initially for serial robots by Walker and Orin and based on the application of Newton-Euler model of the robot dynamics. This method is straightforwardly general is able to the case of MPM robots.

7.4.2 Search Direction Update

A limited-memory quasi-Newton-like method is used at each iteration of the optimization process to solve for the minimization step at the primal level of AL, because of the fact that the considered problem is of large scale type.

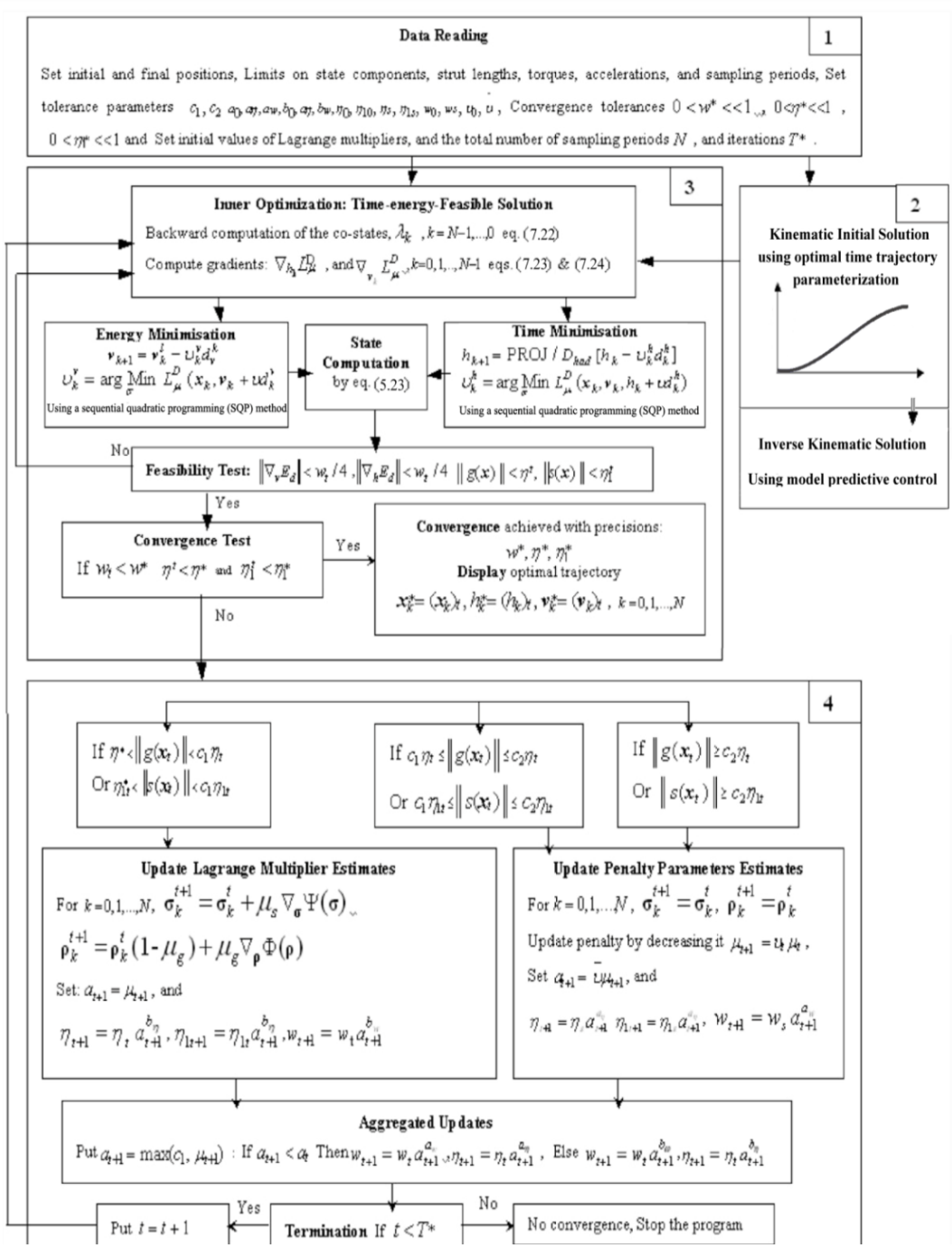


Figure 7.1: Flowchart for AL algorithm function and operation

7.4.3 Overall Solution Procedure

In this research a systematic procedure is used for solving the augmented Lagrangian implementation, see fig 7.1 above. In this procedure, the first step is selecting robot parameters, task definition, (such as starting, intermediate and final poses), workspace limitations and simulation parameters. Then, the kinematic unit defines a feasible solution satisfying initial and final poses. After that, the inner optimization loop solves for the ALD minimization with respect to sampling periods and acceleration control variables to give the MPM dynamic state.

In the following step, this state is tested within against feasibility tolerances. 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 ALD, 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 keep multipliers unchanged and decrease 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.

7.5 Simulation and results for offline trajectory planning

The algorithm described in the previous section is build using Matlab. The following simulation figures show different scenarios of minimizing time, energy, and both together.

In the following cases the initial values of thetas are as follows:

$\Theta_L = 0;$

$\Theta_r = 0;$

$\Theta_1 = 1.3;$

$\Theta_2 = 0.8;$

$\Theta_3 = 1.4;$

And the target values of thetas are:

$\Theta_L = 1;$

$\Theta_r = 0.8;$

$\Theta_1 = 1.1;$

$\Theta_2 = 1.2;$

$\Theta_3 = 1.1;$

Case 1: Minimizing Time

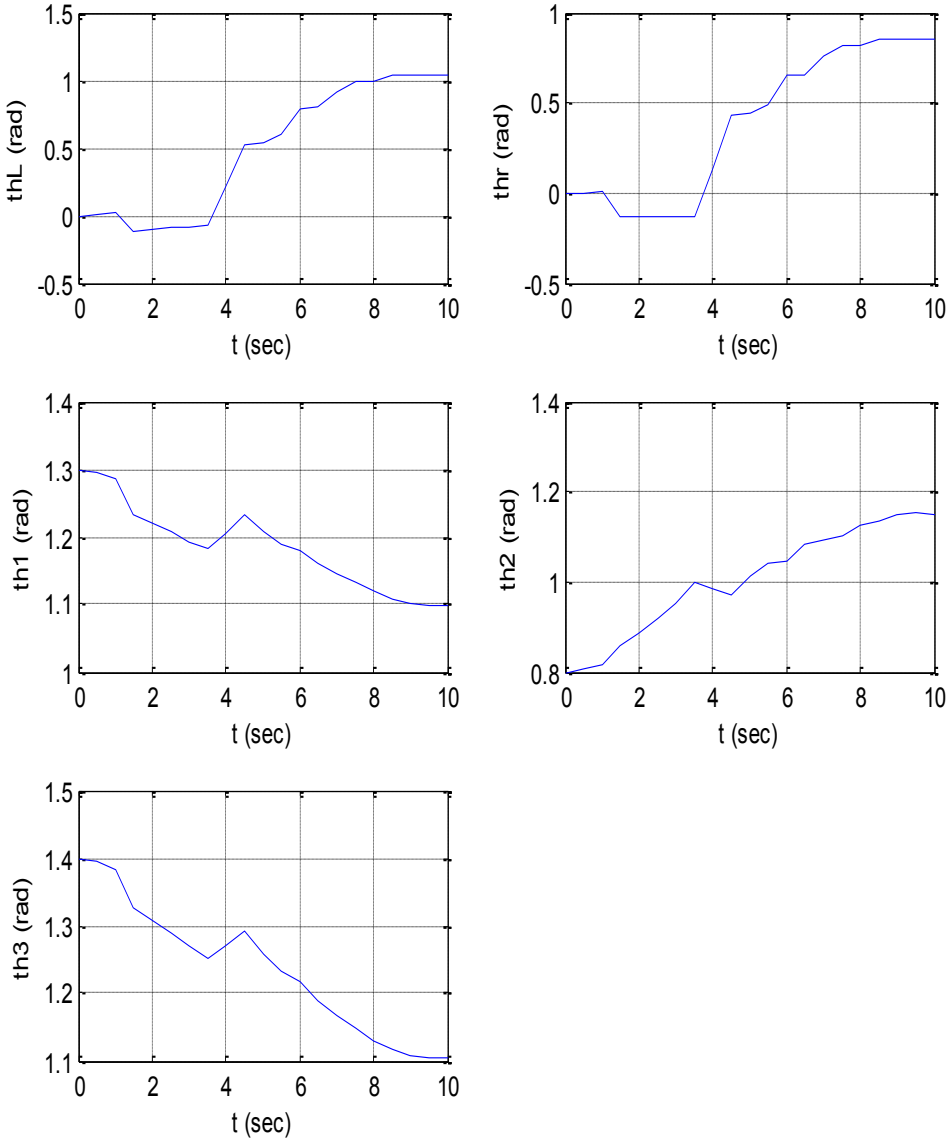


Figure 7.2: Variations of the angels due to minimization of time

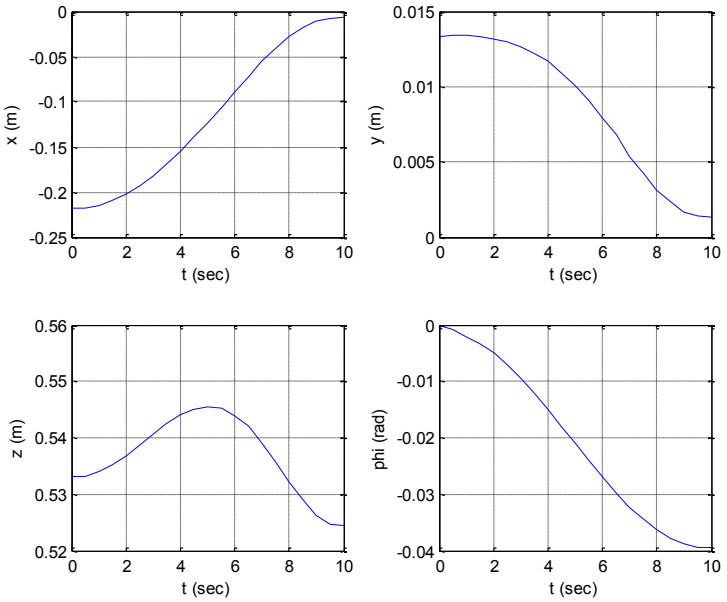


Figure 7.3: Variations of the end effector position due to minimization of time

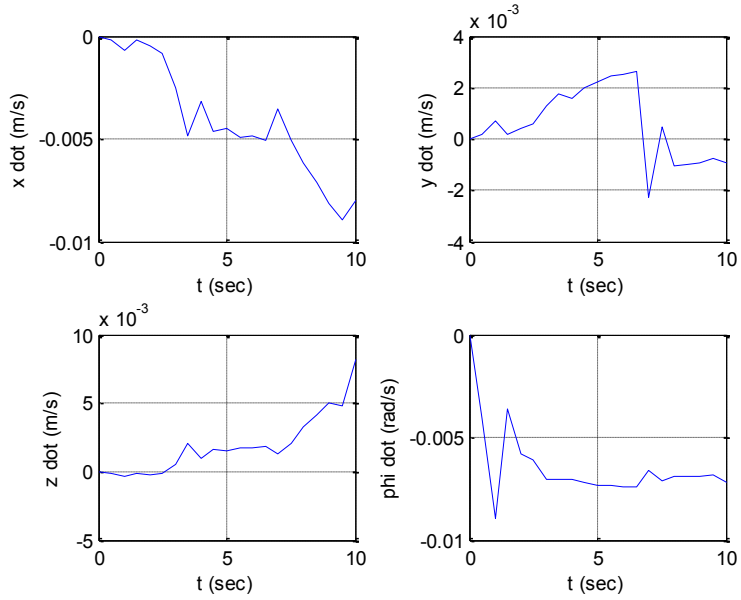


Figure 7.4: Variations of the end effector velocity due to minimization of time

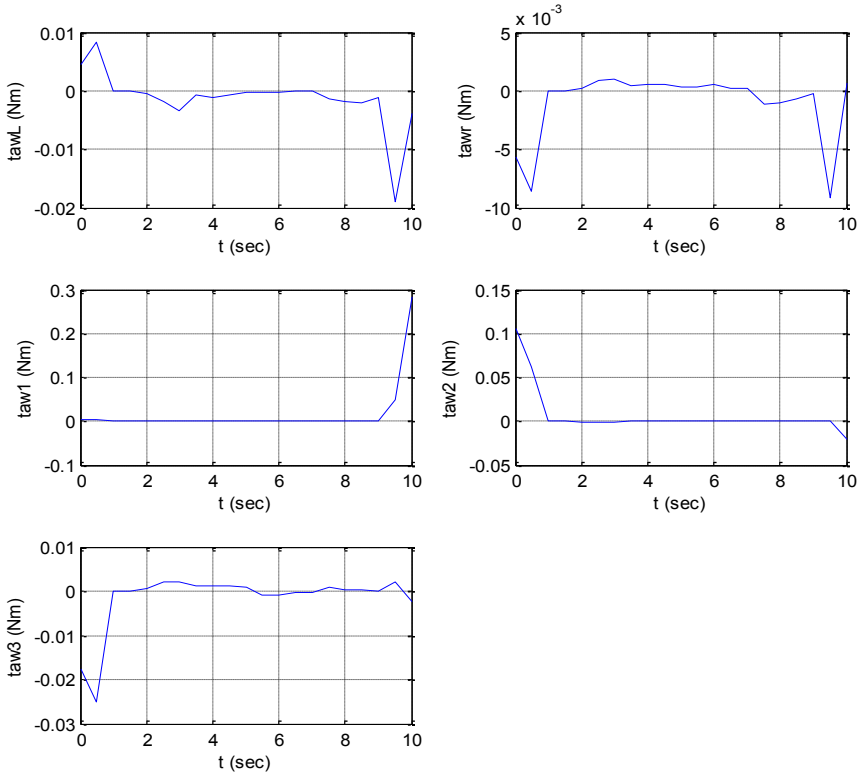


Figure 7.5: Variations of the torque due to minimization of time

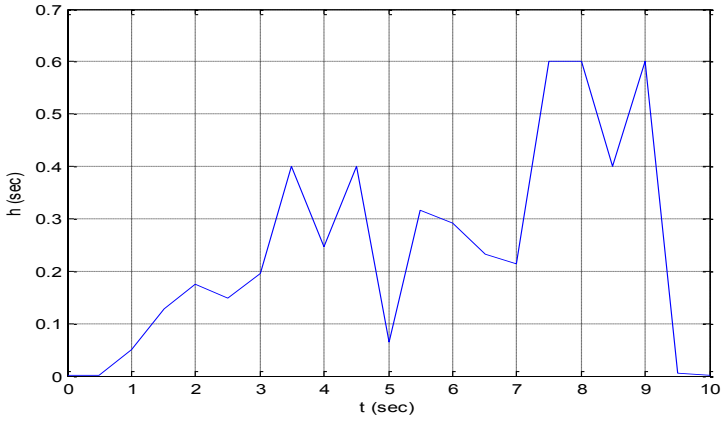


Figure 7.6: Variations of the time steps due to minimization of time

Case 2: Minimizing Energy

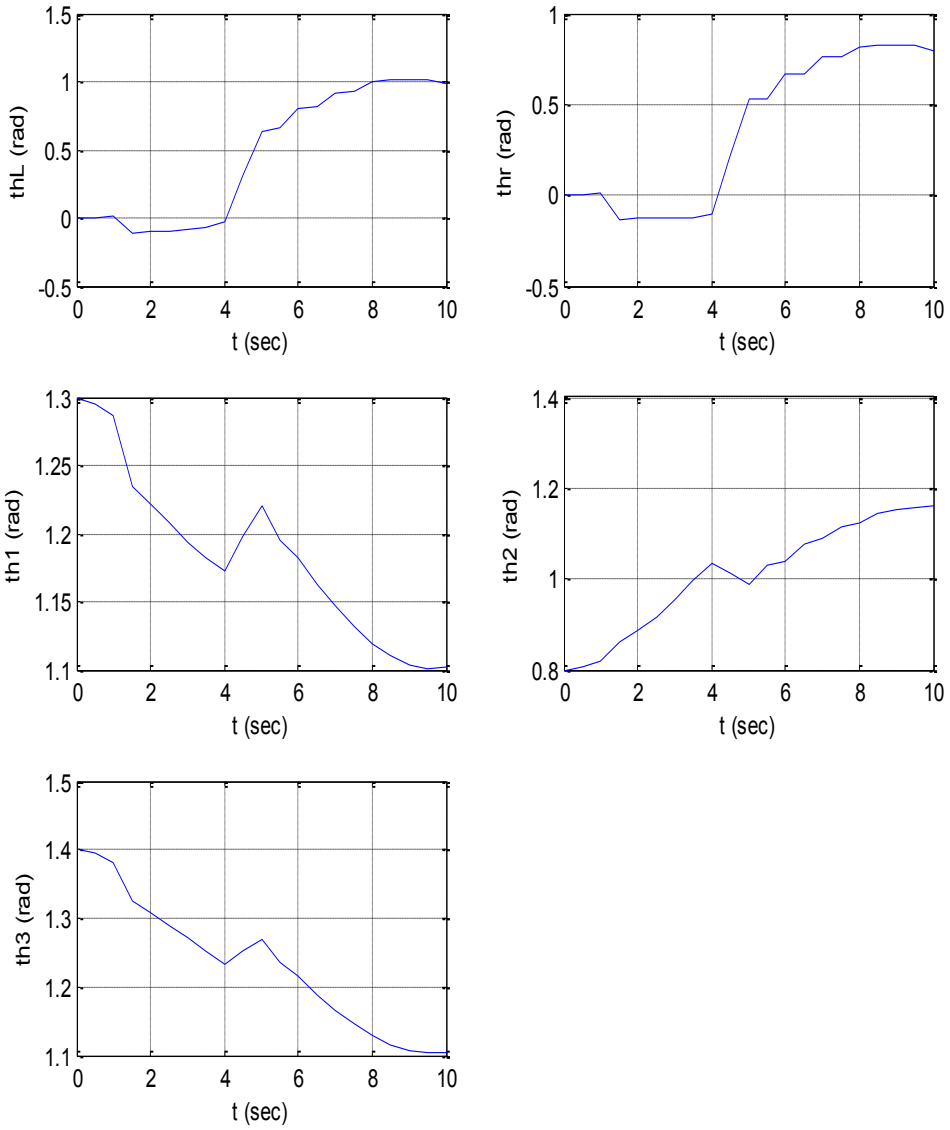


Figure 7.7: Variations of the angels due to minimization of energy

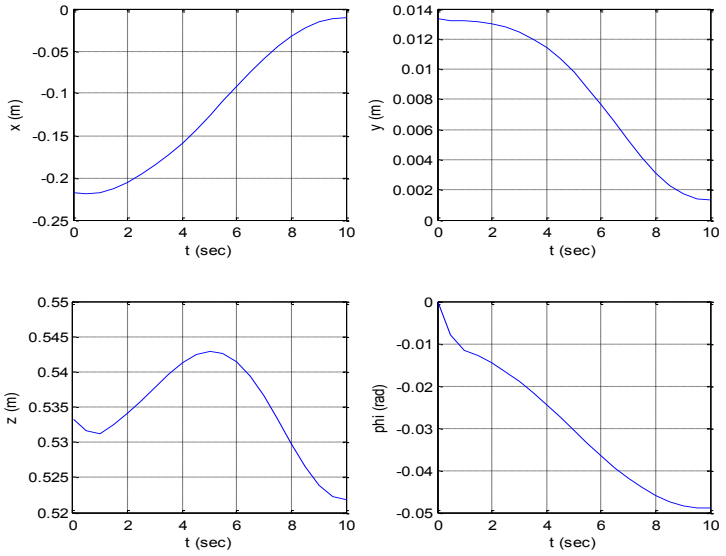


Figure 7.8: Variations of the end effector position due to minimization of energy

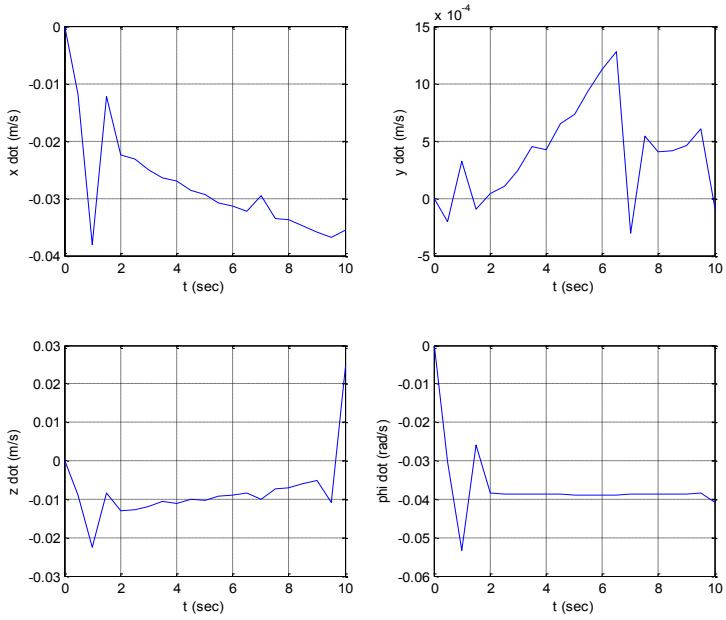


Figure 7.9: Variations of the end effector velocity due to minimization of energy

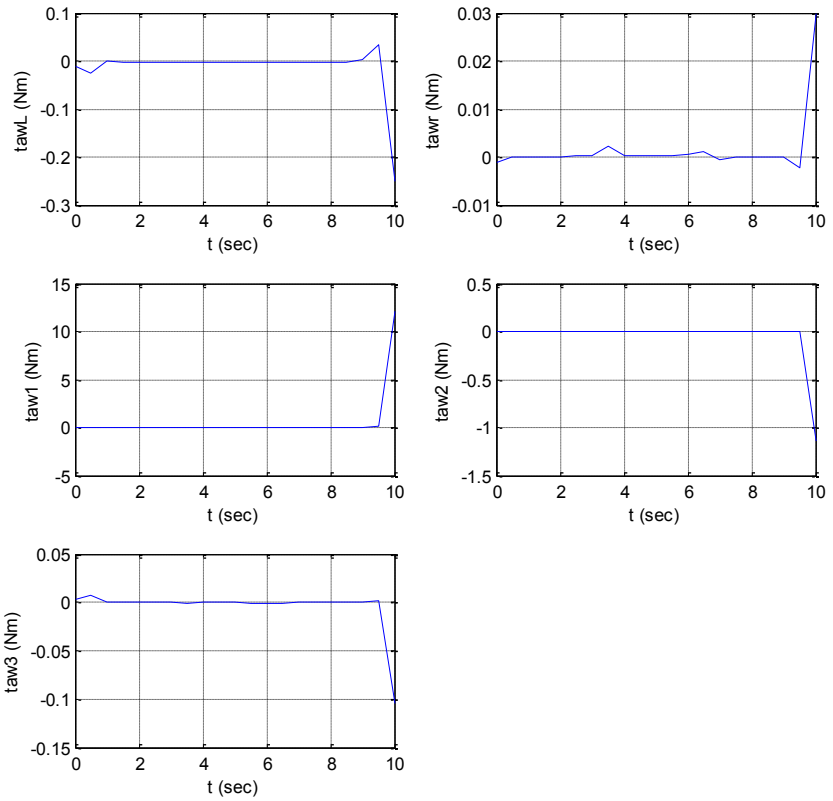


Figure 7.10: Variations of the torque due to minimization of energy

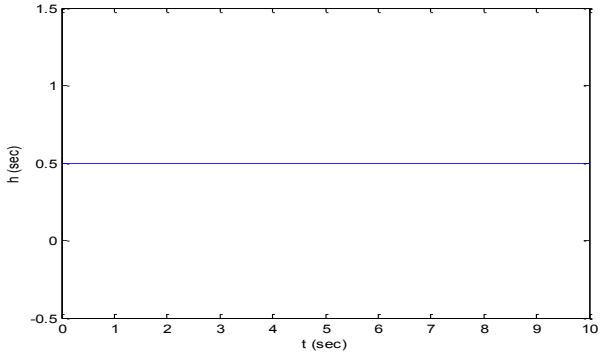


Figure 7.11: Variations of the time step due to minimization of energy

Case 3: Minimizing time and energy (scale 1:1)

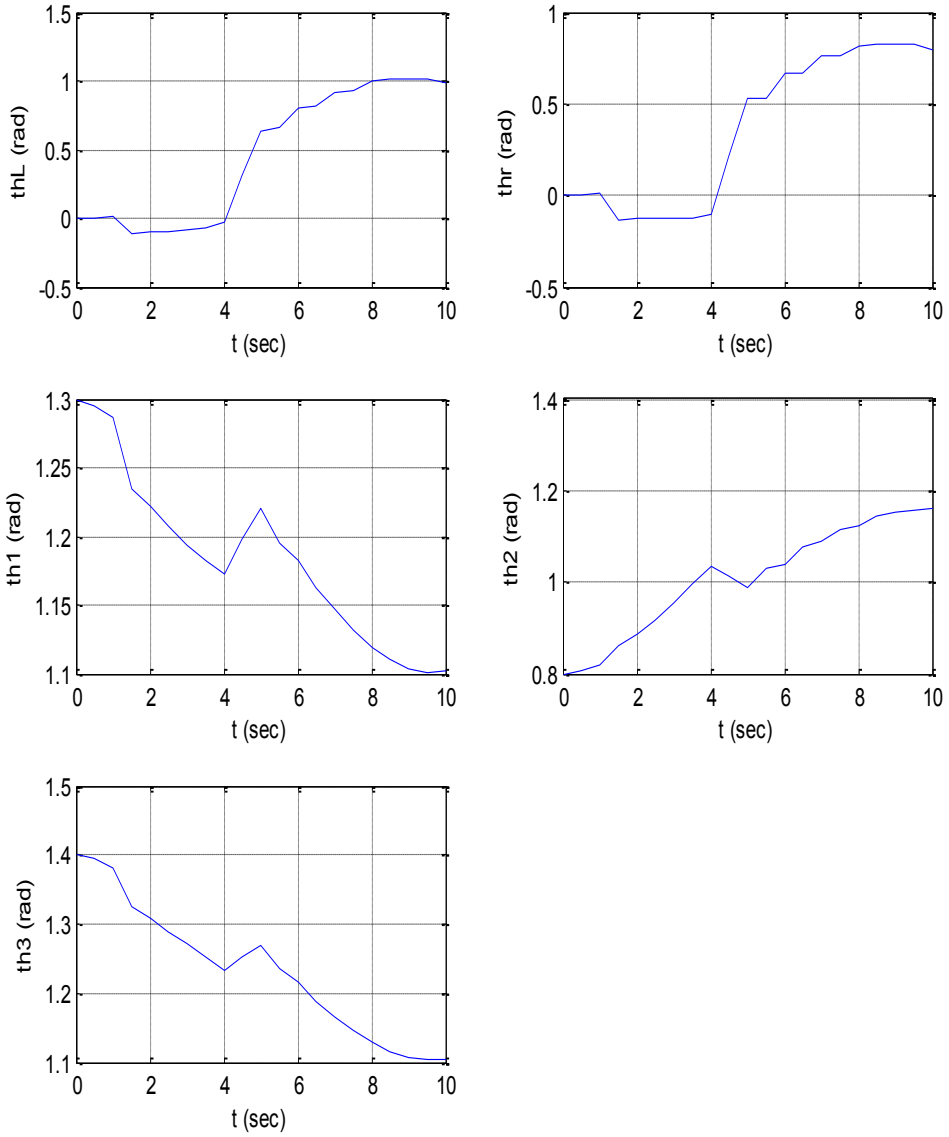


Figure 7.12: Variations of the angels due to minimization of both time and energy

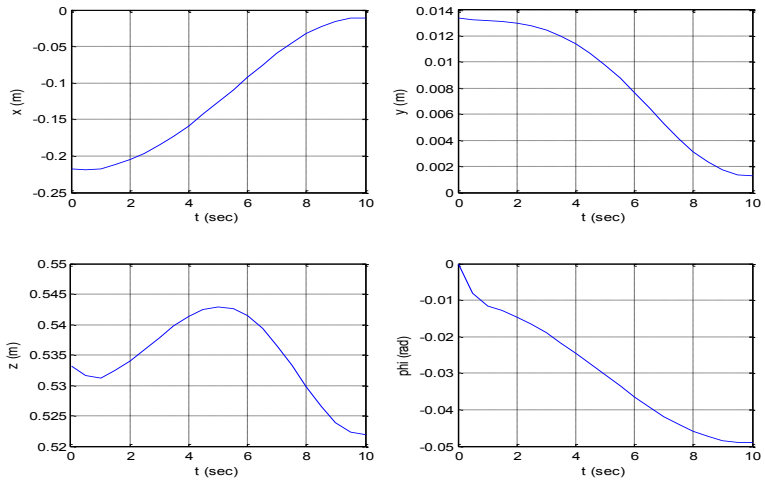


Figure 7.13: Variations of the end effector position due to minimization of both time and energy

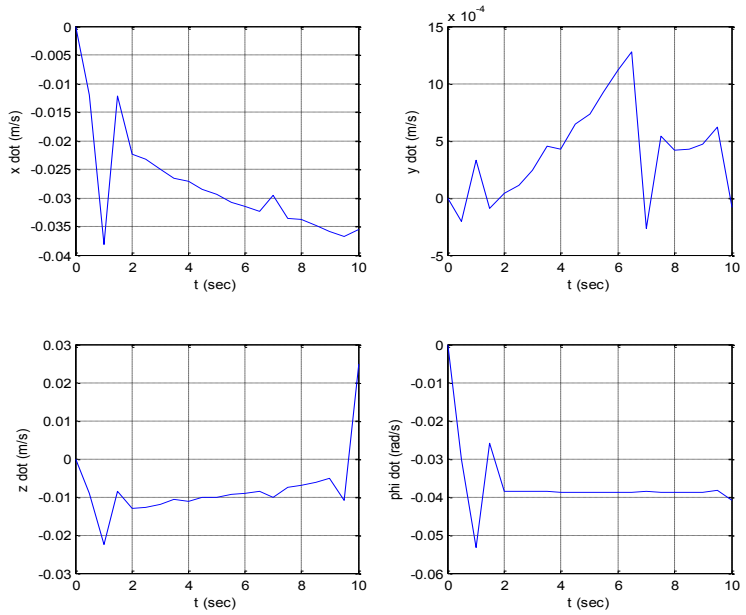


Figure 7.14: Variations of the end effector velocity due to minimization of both time and energy

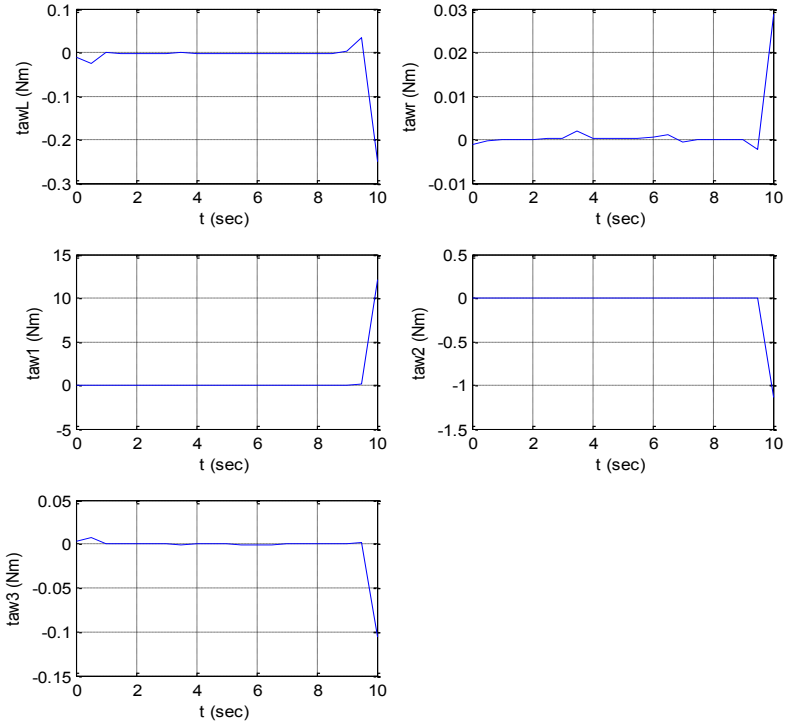


Figure 7.15: Variations of the torque due to minimization of both time and energy

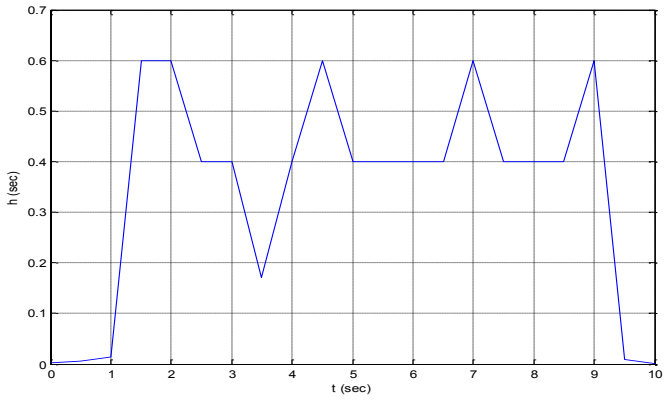


Figure 7.16: Variations of the time step due to minimization of both time and energy

The figures above include the results of the minimization of time alone, energy alone, and both time and energy. The figures show the variation of angles from start position to the end position, also the variation of the position of the end effector and the variation of its velocity, also it show the variation of torque during the interval, and the variation of time steps along the path.

All the previous figures show that the minimization of both h and *v* gives result closed to the desired values with small and acceptable error. Moreover, the figures of thetas show differences between the desired values and the achieved one, which are very closed to the target points.

7.6 Simulation and results for online trajectory planning

In this section, ANFIS is used to construct an online trajectory planning as shown in Fig. 7.17, the result of the offline trajectory planning is used to run 50 different trajectories, each one contains 21 points along the trajectory, this gives 1050 samples, among which 950 are considered for training, whereas testing and validation datasets, each of them is obtained using 100 entry samples.

Figure 7.18 shows the training performance for AL-ANFIS, which is interesting as it reaches very small root mean square error (RMSE), less than 0.1 in less than 10 epochs. It is noted that the configuration used for the learning is determined among infinitely many solutions that exist for each input.

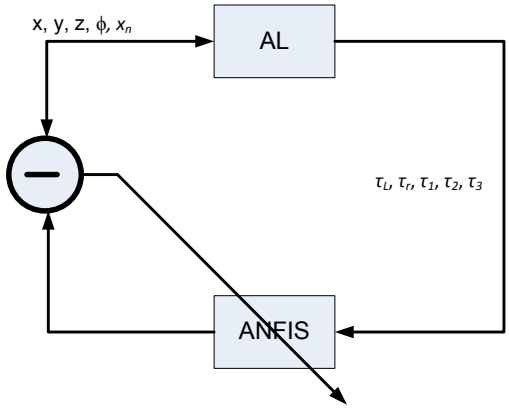


Figure 7.17: AL-ANFIS, The use of AL solution for the learning of ANFIS module

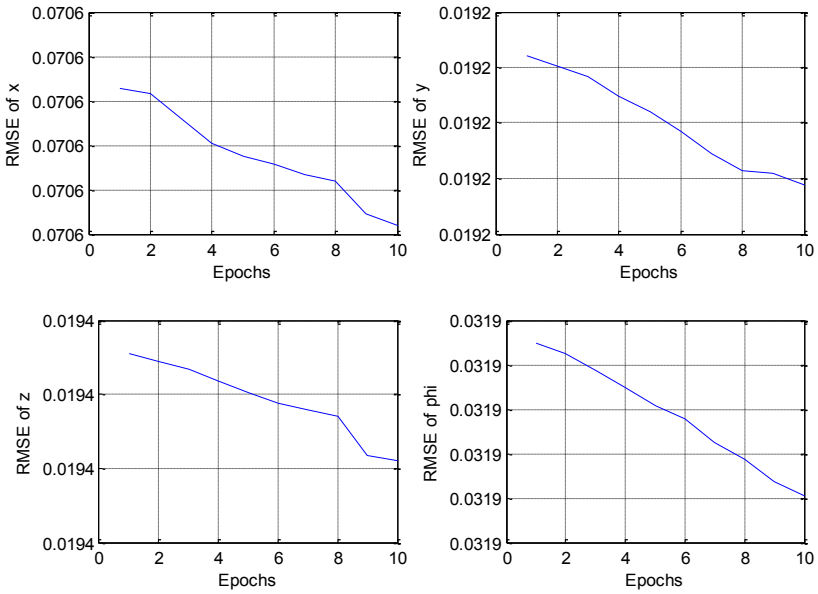


Figure 7.18: AL-ANFIS performance – root mean square error output with respect to learning epochs

Figure 7.19 shows the difference between the real and estimated values of the joint angles at the 1050 samples. It is believe that a better fine tuning of the ANFIS parameters will improve in the accuracy of the matching between ANFIS outcomes and the AL provided results. This is being undertaken in an ongoing work.

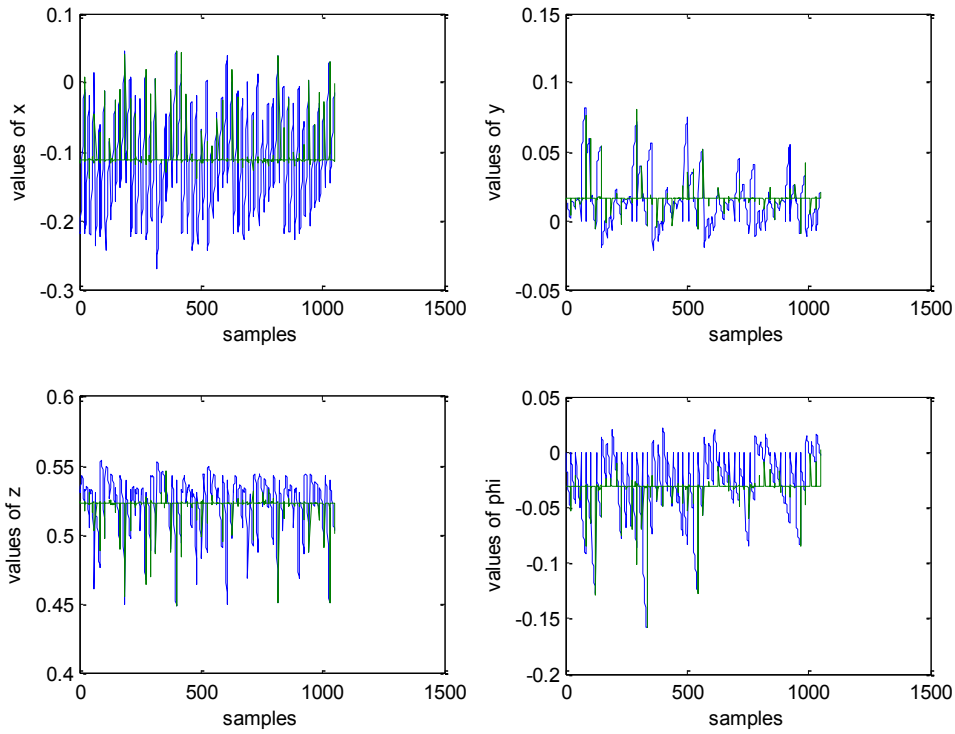


Figure 7.19: AL-ANFIS performance – difference between real and estimated values of the MPM values

Chapter 8

Conclusions and Future Work

In this work, the problem of kinematic, dynamic modeling and motion planning of mobile parallel manipulators is considered. This relatively new generation of machines combines the large space of mobile robots and high accuracy and payload of the parallel machines; this allows wide application of these machines. Comparison shows complexity of the result hybrid structure which contains a high level of nonlinearities. According to their complexity, the forward and inverse kinematics

models of mobile parallel manipulator are difficult to derive. In this research, the forward and inverse kinematic models of a mobile parallel manipulator (MPM) are derived. An MPR composed of a three-wheels non-holonomic mobile platform and a 3-RRPaR translational parallel robot is used for this purpose.

The position and differential kinematic solutions are derived and the Jacobian matrix relating output velocities to the actuated joint rates is generated. By resorting to the Neuro-fuzzy structure, the inverse kinematic is obtained using ANFIS. Moreover, joint limit and singularity avoidance is achieved taking the advantage of the minimum time cycloidal parameterization and the additional factor. The dynamic modeling for the MPM is derived. And since it possesses self motion with one degree of redundancy, the dynamic control in task space is carried out by utilizing a model-based controller, and validate the effectiveness of the derived models is validated by the simulation results. The minimum time energy optimal control of the MPM is then solved using an augmented Lagrangian technique. Upon this solution a dataset of trajectories is built and used to train an ANFIS system. Simulation results of both parts are encouraging.

As a future trend of this work, two main recommendations are to optimize the ANFIS structure to achieve better online planning accuracy. The second perspective consists of including obstacle avoidance for both offline and online planning.

Appendix A: Maple Solution for Position Kinematic Analysis

```
> e1:=2*w*y + 2*w*x - 2*b*(y+w)*cos(th1) - 2*b*(x-
w)*cos(th2)- 2*z*b*(sin(th1)-sin(th2))=0;
```

$$e1 := 2\,w\,y + 2\,w\,x - 2\,b\,(y + w)\cos(th1) - 2\,b\,(x - w)\cos(th2) - 2\,z\,b\,(\sin(th1) - \sin(th2)) = 0$$

```
> e2:=4*w*y - 2*b*(y+w)*cos(th1) - 2*b*(y-w)*cos(th3) -
2*z*b*(sin(th1)-sin(th3))=0;
```

$$e2 := 4\,w\,y - 2\,b\,(y + w)\cos(th1) - 2\,b\,(y - w)\cos(th3) - 2\,z\,b\,(\sin(th1) - \sin(th3)) = 0$$

```
> e3:=x^2 + (y+u-e-b*cos(th1))^2 +(z-b*sin(th1))^2 =a^2;
```

$$e3 := x^2 + (y + u - e - b\cos(th1))^2 + (z - b\sin(th1))^2 = a^2$$

```
> solve({e1,e2},{y,x});
```

$$\left\{x = -\left(b\left(-b\cos(th3)\,w\cos(th2) + 2\cos(th2)\,w^2 - \cos(th1)\,w^2 + z\,b\sin(th1)\cos(th3) + 2\,z\sin(th2)\,w - z\,b\sin(th2)\cos(th1) - z\,b\sin(th2)\cos(th3) - w\,z\sin(th1) - w\,z\sin(th3) + 2\,b\cos(th1)\cos(th3)\,w + b\cos(th1)\,z\sin(th3) - \cos(th3)\,w^2 - b\cos(th1)\,w\cos(th2)\right)\right) \Big/ \left(-b\cos(th1)\,w - b\cos(th3)\,w - 2\,b\cos(th2)\,w + b^2\cos(th2)\cos(th1) + b^2\cos(th2)\cos(th3) + 2\,w^2\right), y = \frac{b\,(-\cos(th1)\,w + \cos(th3)\,w - z\sin(th1) + z\sin(th3))}{-2\,w + b\cos(th1) + b\cos(th3)}\right\}$$

References

[1] Tahmasebi, L. W. Tsai and F., "Synthesis and analysis of a new class of six-DOF parallel mini-manipulators." J. Robot. Syst., 1993, Issue 5, Vol. 10, pp. 561–580.

[2] Shoham, S. Shoval and M., "Sensory redundant parallel mobile mechanism." 2001. Proc. of IEEE Int. Conf. on Robotics and Automation. pp. 2273–2278.

[3] T. Yamawaki, T. Omata, and O. Mori., "4R and 5R parallel mechanism mobile robots ." 2004. Proc. of IEEE Int. Conf. on Robotics and Automation. pp. 3684–3689.

[4] Dillmann, R. Graf and R., "Active acceleration compensation using a stewart-platform on a mobile robot." 1997. Proc. of 2nd Euromicro Workshop on Advanced Mobile Robots. pp. 59–64.

[5] M. W. Decker, A. X. Dang, and I. Ebert-Uphoff., "Motion planning for active acceleration compensation." 2002. Proc. of IEEE Int. Conf. on Robotics and Automation , 2001. pp. 1257–1264.

[6] Yangmin Li, Qingsong Xu, and Yugang Liu., "Novel Design and Modeling of a Mobile Parallel Manipulator." Orlando, Florida : s.n., May 2006. Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 1135–1140.

[7] A. Gasparetto, V. Zanotto., "A new method for smooth trajectory planning of robot manipulators." Mechanism and Machine Theory, 2007, Vol. 42, pp. 455–471.

[8] S.F.P. Saramago, M. Ceccarelli., "Effect of basic numerical parameters on a path planning of robots taking into account actuating energy." Mechanism and Machine Theory, 2004, Vol. 39, pp. 247–260.

[9] S.F.P. Saramago, V. Steffen jr., "Optimization of the trajectory planning of robot manipulators taking into account the dynamics of the system." Mechanism and Machine Theory, 1998, Vol. 33, pp. 883–894.

[10] T. Chettibi, H.E. Lehtihet, M. Haddad, S. Hanchi., "Minimum cost trajectory planning for industrial robots." European Journal of Mechanics A/Solids, 2004, Vol. 23, pp. 703–715.

[11] Zha, Xuan F., "Optimal pose trajectory planning for robot manipulators." Mechanism and Machine Theory, 2002, Vol. 37, pp. 1063–1086.

[12] J. Agirrebeitia, R. Avilés, I.F. de Bustos, G. Ajuria., "A new APF strategy for path planning in environments with obstacles." Mechanism and Machine Theory, 2005, Vol. 40, pp. 645–658.

[13] F. Valero, V. Mata, A. Besa., "Trajectory planning in workspaces with obstacles taking into account the dynamic robot behavior." Mechanism and Machine Theory, 2006, Vol. 41, pp. 525–536.

[14] S.F.P. Saramago, V. Steffen Jr., "Optimal trajectory planning of robot manipulators in the presence of moving obstacles." Mechanism and Machine Theory, 2000, Vol. 35, pp. 1079–1094.

[15] Khatib, O. Brock and O., "Elastic strips: A framework for motion generation in human environments." Int. J. Robot. Res., 2002, Issue 12, Vol. 21, pp. 1031–1052.

[16] Kavraki, O. Brock and L. E., "Decomposition-based motion planning: A framework for real-time motion planning in high-dimensional configuration spaces." 2001. IEEE Int. Conf. Robot. Autom. pp. 1469–1474.

[17] Shiller, P. Fiorini and Z., "Time optimal trajectory planning in dynamic environments." 1996. IEEE Int. Conf. Robot. Autom. pp. 1553–1558.

[18] Kuffner, S. M. LaValle and J. J., "Randomized kinodynamic planning." Int. J. Robot. Res., May 2001, Issue 5, Vol. 20, pp. 378–400.

[19] Overmars, J. van den Berg and M., "Roadmap-based motion planning in dynamic environments." IEEE Trans. Robot., Oct 2005, Issue 5, Vol. 21, pp. 885–897.

[20] C. Ferrari, E. Pagello, M. Voltolina, J. Ota, and T. Arai., "Varying paths and motion profiles in multiple robot motion planning." Monterey, CA : s.n., Jul.1997. in Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom. pp. 186–193.

[21] D. Vasquez, F. Large, T. Fraichard, and C. Laugier., "High-speed autonomous with motion prediction for unknown moving obstacles." 2004. Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. pp. 82–87.

[22] J. van den Berg, D. Ferguson, and J. Kuffner., "Anytime path planning and replanning in dynamic environments." May, 2006. Proc. IEEE Int. Conf. Robot. Autom. pp. 1243–1248.

[23] W. Carriker, P. Khosla, and B. Krogh., "Path planning for mobile manipulators for multiple task execution." IEEE Trans. Robot. Autom., Jun. 1991, Issue 3, Vol. 7, pp. 403–408.

[24] Seraji, H., "A unified approach to motion control of mobile manipulators." Int. J. Robot. Res., 1998, Issue 2, Vol. 17, pp. 107–118.

[25] Kyriakopoulos, H. Tanner and K., "Nonholonomic motion planning for mobile manipulators." Apr. 2000. Proc. IEEE Int. Conf. Robot. Autom. Vol. 2, pp. 1233–1238.

[26] F. Pin, J. Culioli, and D. Reister., "Using minimax approaches to plan optimal task commutation configurations for combined mobile platformmanipulator systems." IEEE Trans. Robot. Autom., Feb. 1994, Issue 1, Vol. 10, pp. 44–54.

[27] D. H. Shin, B. S. Hamner, S. Singh, and M. Hwangbo., "Motion planning for a mobile manipulator with imprecise locomotion." Oct., 2003. Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. Vol. 1, pp. 847–853.

[28] Yun, Y. Yamamoto and X., "Coordinating locomotion and manipulation of a mobile manipulator." IEEE Trans. Autom. Control, Jun. 1994, Issue 6, Vol. 39, pp. 1326–1332.

[29] Xi, J. Tan and N., "Unified model approach for planning and control of mobile manipulators." 2001. Proc. IEEE Int. Conf. Robot. Autom. Vol. 3, pp. 3145–3152.

[30] Simeon, L. Jaillet and T., "A PRM-based motion planner for dynamically changing environments." 2004. Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. Vol. 2, pp. 1606–1611.

[31] O. Brock, O. Khatib, and S. Viji., "Task-consistent obstacle avoidance and motion behavior for mobile manipulation." May 2002. Proc. IEEE Int. Conf. Robot. Autom. Vol. 1, pp. 388–393.

[32] P. O¨ gren, N. Egerstedt, and X. Hu., "Reactive mobile manipulation using dynamic trajectory tracking." Apr. 2000. Proc. IEEE Int. Conf. Robot. Autom. Vol. 4, pp. 3473–3478.

[33] J. Mbede, S. Ma, Y. Toure, V. Graefe, and L. Zhang., "Robust neuro-fuzzy navigation of mobile manipulator among dynamic obstacles." May 2004. Proc. IEEE Int. Conf. Robot. Autom. Vol. 5, pp. 5051–5057.

[34] Latombe, T.-Y. Li and J.-C., "On-line manipulation planning for two robot arms in a dynamic environment." May 1995. Proc. IEEE Int. Conf. Robot. Autom. Vol. 1, pp. 1048–1055.

[35] Zhijun Li, Weidong Chen., "Adaptive neural-fuzzy control of uncertain constrained multiple coordinated nonholonomic mobile manipulators." 2008. Engineering Applications of Artificial Intelligence 21. pp. 985–1000.

[36] John Vannoy, Jing Xiao., "Real-Time Adaptive Motion Planning (RAMP) of Mobile Manipulators in Dynamic Environments With Unforeseen Changes." IEEE TRANSACTIONS ON ROBOTICS, OCTOBER 2008, Issue 5, Vol. 24.

[37] VE, Gough., "Contribution to discussion of papers on research in automobile stability, control and type performance." 1956. Auto Div Inst Mech Eng Part D (J Automob Eng). pp. 392–395.

[38] D, Stewart., "A platform with six degree-of-freedom." 1965. Proc Inst Mech Eng. Vol. 180, pp. 371–386.

[39] JP, Merlet., *Parallel robots.* s.l. : Kluwer Academic Publisher, 2000.

[40] Pietsch IT, Krefft M, Becker OT, Bier CC, Hesselbach J., "How to reach the dynamic limits of parallel robots? An autonomous control approach." IEEE Trans Autom Sci Eng, 2005, Issue 4, Vol. 2, pp. 369–80.

[41] Ahrikencheikh C, Seireg A., *Optimized-motion planning.* New York : Wiley, 1994.

[42] Khoukhi A, Ghoul A., "Maximum dynamic stress search for a robot manipulator." Robotica, 2004, Issue 5, Vol. 22, pp. 513–22.

[43] A, Khoukhi., "An optimal time–energy control design for a prototype educational robot." Robotica, 2002, Issue 6, Vol. 20, pp. 661–71.

[44] Vukobratovic M, Stotic D., "Is dynamic control needed in robotics system? And if so to what extent?" Int J Robotics Res, 1985, Issue 2, Vol. 2, pp. 18–35.

[45] Baron L, Angeles J., "The kinematic decoupling of parallel manipulators under joints-sensor redundancy." IEEE Trans Robotics Autom, 2000, Issue 1, Vol. 16, pp. 12–9.

[46] Baron L, Angeles J., "The direct kinematics of parallel manipulators under joints-sensor data." IEEE Trans Robotics Autom, 2000, Issue 6, Vol. 16, pp. 644–51.

[47] Dasgupta B, Mruthyunjaya TS., "The stewart platform manipulator, a review." Mech Mach Theory, 2000, Issue 1, Vol. 35, pp. 15-40.

[48] Ma O, Angeles J., "Architecture singularities of parallel manipulators." Int J Robotics Autom, 1992, Issue 1, Vol. 7, pp. 23–9.

[49] C, Gosselin., "Parallel computational algorithms for the kinematics and dynamics of plannar and spatial parallel manipulators." ASME, J Dyn Syst, Meas Control, 1996, Issue 1, Vol. 118, pp. 22–8.

[50] Kumar DA, Ming CI, Huat YS, Guilin Y., "Workspace generation and planning singularity-free path for parallel manipulators." Mech Mach Theory, 2005, Issue 7, Vol. 40, pp. 776–805.

[51] V, Mermertas., "Optimal design of manipulator with four-bar mechanism." Mech Mach Theory, 2004, Issue 5, Vol. 39, pp. 545–54.

[52] Bhattacharya S, Hatwal H, Ghosh A., "Comparison of an exact and approximate method of singularity avoidance in platform-type parallel manipulators." Mech Mach Theory, 1998, Issue 7, Vol. 33, pp. 965–74.

[53] Sen S, Dasgupta B, Mallik AK., "A variational approach for singularity-free path planning of parallel manipulators." Mech Mach Theory, 2003, Issue 11, Vol. 38, pp. 1165–83.

[54] E, Polak., *Optimization, algorithms and consistent approximation.* New York : Springer, 1997.

[55] Hay M, Snyman JA., "Methodologies for the optimal design of parallel manipulators." Inl J Numer Methods Eng, 2004, Issue 1, Vol. 59, pp. 131–52.

[56] Hay M, Snyman JA., "A multi-level optimization methodology for determining the dexterous workspaces of planar parallel manipulator." Struct Multi- disciplinary Optim, 2005, Issue 6, Vol. 30, pp. 422–7.

[57] Amar Khoukhi, Luc Baron, Marek Balazinski., "Constrained multi-objective trajectory planning of parallel kinematic machines." Robotics and Computer-Integrated Manufacturing, 2009, Vol. 25, pp. 756–769.

[58] Michael W. Decker, Anh X. Dang, and Imme Ebert-Uphoff., "Motion planning for active acceleration compensation." 2001. Proc. of IEEE Int. Conf. on Robotics and Automation. pp. 1257–1264.

[59] Shoham, Shraga Shoval and Moshe., "Sensory Redundant Parallel Mobile Mechanism." Seoul, Korea : s.n., May 2001. Proceedings of the 2001 IEEE International Conference on Robotics & Automation. pp. 21-26.

[60] P. Ben Horin, S. Djerassi, M. Shoham, R. Ben Horin., "Dynamics of a six degrees-of-freedom parallel robot actuated by three two-wheel carts." Multibody Syst Dyn, 2006, Vol. 16, pp. 105–121.

[61] T. Yamawaki, T. Omata, and O. Mori., "4R and 5R parallel mechanism mobile robots." 2004. Proc. of IEEE Int. Conf. on Robotics and Automation. pp. 3684–3689.

[62] Huapeng Wu, Heikki Handroos, Pekka Pessi, Jussi Hopia., "A Mobile Hybrid Parallel Robot with Redundant Kinematic Structure." Kunming, China : s.n., December 2006. Proceedings of the 2006 IEEE International Conference on Robotics and Biomimetics, 1157 - 1162.

[63] Huapeng Wu, Pekka Pessi, Heikki Handroos., "Assembling and Repairing for ITER Vacuum Vessel with Mobile Parallel Robot." proceeding of Fusion Engineering, 2007. SOFE 2007. 2007 IEEE 22nd Symposium, pp.1-4.

[64] Huapeng Wu, Pekka Pessi, Heikki Handroos., "A Hybrid Parallel Robot for the Assembling of ITER." Robotics, Automation and Mechatronics, 2008 IEEE Conference, Lappeenranta, 21-24 Sept. 2008, pp. 640 - 644 .

[65] Pekka Pessi, Huapeng Wu, Heikki Handroos, Lawrence Jones., "A mobile robot with parallel kinematics to meet the requirements for assembling and machining the ITER vacuum vessel." Fusion Engineering and Design, 2007, Vol. 82, pp. 2047–2054.

[66] Amar Khoukhi, Luc Barona, Marek Balazinskia, Kudret Demirli., "A hierarchical neuro-fuzzy system to near optimal-time trajectory planning of redundant manipulators." Engineering Applications of Artificial Intelligence, 2008, Vol. 21, pp. 974–984.

[67] Lions, P. L., *Generalized Solutions of Hamilton-Jacobi Equations.* s.l. : Pittman, 1982.

[68] Bellman, P. E., *Dynamic Programming.* s.l. : Princeton University Press, 1957.

[69] L. S. Pontryagin, V. G. Boltyanski, R. V. Gamkrelidge, E. F. Miscenko., *the Mathematical Theory of Optimal Processes.* s.l. : Wiley, 1962.

[70] Powell, M. J. D., "A Method for Nonlinear Constraints in Minimization Problems." *Optim(R. Fletcher, ed.).* pp. 283-298.

[71] Hestens, M. R., "Multiplier and Gradient Methods." Optimization Theory and Application, 1969, Vol. 4, pp. 303-320.

[72] Lemaréchal, J.B.Hiriart-Urruty and C., *Convex Analysis and Minimization Algorithms.* Berlin, New York : SpringeVerlag, 1993.

[73] G. Cohen, D. L. Zhu., "Decomposition-Coordination Methods in Large Scale Optimization Problems: The non Differential Case and the Use of Augmented Lagrangean." Advances in Large Scale Systems, 1984, Issue 2, Vol. 1.

[74] N. Gould, D. Orban, A. Sartenear, and P. Toint., "Super Linear Convergence of Primal-Dual interior Points Algorithms for Non Linear Programming." SIAM Jour on Optimization. Mathematical Programming, B, 2001, Issue 4, Vol. 11, pp. 974-1002.

[75] , *www.cse.clrs.ac.uk/Activity/LANCELOT.* [Online]

[76] Rockafellar, T., "Lagrange Multipliers and optimality." SIAM Review, 1993, Vol. 35, pp. 183-238.

[77] Shoubik, M., "Simulation of the Industry and the Firm." American Economic Review L, 1960, Issue 5, pp. 908-919.

[78] Claude, D., "Decoupling for Non Linear System." Syst. Cont. Letters, 1982, pp. 242-248.

[79] Dillmann, Rene Graf and Rudiger., "Active acceleration compensation using a stewart-platform on a mobile robot." 1997. Proc. of 2nd Euromicro Workshop on Advanced Mobile Robots. pp. 59–64.

Vita

NAME: Mutaz "Mohammad Ali" Hamdan

BIRTH: 13th November 1982

EDUCATION: Bachelor of engineering degree in Mechanical Eng. Mechatronics Engineering Branch

University of Palestine Polytechnic University (PPU), www.ppu.edu

Hebron, West-bank, Palestine

MASTER OF SCIENCE IN Systems Engineering

Systems Engineering Department, King Fahd University of Petroleum & Minerals

Dhahran, K.S.A.

Current Address: King Fahd University of Petroleum & Minerals, Dhahran, K.S.A.

Permanent Address: Hebron, West-bank, Palestine

CONTACT: eng_mutaz@yahoo.com

Mobile: +966 56 550 50 42