MODELING AND CONTROL OF A HYPER REDUNDANT MANIPULATOR

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ATİLLA BAYRAM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MECHANICAL ENGINEERING

FEBRUARY 2010

Approval of the thesis

MODELING AND CONTROL OF A HYPER REDUNDANT MANIPULATOR

submitted by **ATİLLA BAYRAM** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Suha Oral
Head of Department, **Mechanical Engineering**

Prof. Dr. M. Kemal Özgören
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. S. Kemal İder
Mechanical Engineering Dept., METU

Prof. Dr. M. Kemal Özgören
Supervisor, Mechanical Engineering Dept., METU

Prof. Dr. Aydan Erkmen
Electrical and Electronics Engineering Dept., METU

Prof. Dr. Reşit Soylu
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. Metin U. Salamcı
Mechanical Engineering Dept., Gazi Uni.

Date: 02.02.2010

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Atilla Bayram

Signature :

ABSTRACT

MODELING AND CONTROL OF A HYPER REDUNDANT MANIPULATOR

Bayram, Atilla

Ph.D., Department of Mechanical Engineering
Supervisor: Prof. Dr. M. Kemal Özgören

February 2010, 158 pages

The hyper redundant manipulators (HRMs) have excessively large degrees of freedom. As a special but practicable subset, the binary HRMs use binary (on-off) actuators with only two stable states such as pneumatic cylinders and solenoids. Such actuators are simple, cheap, and easy to control. Therefore, a binary HRM has been studied in this thesis. The thesis work covers the conceptual design of a spatial binary HRM together with its controlled motion simulations. The manipulator consists of many modules, each of which has the same constructive characteristics and consists of three submodules which are two cascaded variable geometry truss structures working in mutually orthogonal planes and a discrete twister. The manipulator is assumed to be powered with pneumatic on-off actuators. Because of the discrete nature of the binary actuators, a small but continuously actuated manipulator with six degrees of freedom is installed as the last module of the HRM in order to compensate the discretization errors.

To solve the inverse kinematics problem of the HRM, three methods have been presented. These are the spline fitting, the extended spline fitting, and the workspace filling methods. The spline fitting method is based on forcing the spine (i.e. the center line) of the manipulator to approximate a spatial reference spline which is specified as a desired curve. In the extended spline fitting method, the result found in the first method is improved by using a genetic algorithm. In the work space filling method, the workspace of the manipulator is filled randomly with a sufficiently large finite number of discrete configurational samples. If it is desired to have concentration on a particular region of the work space, then that region is filled by using a genetic algorithm. After the filling stage, the sample closest to the desired configuration is determined by a suitable search algorithm.

Finally, in order to simulate the motion of the HRM between two successive configurational steps, the equations of motions of the HRM are obtained in terms of the pressure forces generated by the binary pneumatic actuators. Then, the necessary simulations are carried out to demonstrate the performance of the HRM in some typical applications.

Keywords: Hyper redundant manipulator, binary actuation, spline, genetic algorithm.

ÖZ

BİR AŞIRI ARTIKSIL MANİPÜLATÖRÜN MODELLENMESİ VE KONTROLÜ

Bayram, Atilla

Doktora, Makina Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. M. Kemal Özgören

Şubat 2010, 158 sayfa

Aşırı artıksıl manipülatörler (AAM'ler) gereğinden çok fazla serbestlik derecesine sahip olan manipülatörlerdir. Bu manipülatörlerin özel fakat uygulanabilir olanları, iki konumlu eyleticilerle kontrol edilen ikili AAM'dir. İki konumlu eyleticilere örnek olarak solenoitler ve havalı silindirler gösterilebilir. Bu gibi eyleticiler basit ve ucuz olup hem de kolay kontrol edilebilirler. Bu yüzden, bu tezde ikili bir AAM üzerinde çalışılmıştır. Bu tez çalışması, ikili bir AAM'nin kavramsal tasarımı ile dinamik analizini ve hareket benzetimlerini kapsamaktadır. Bu manipülatör, aynı yapısal karakteristiğe sahip olan bir çok modülün üst üste eklenmesiyle oluşturulmuştur. Bu modüllerin her biri üç adet alt modülden oluşup bunlardan ikisi değişken geometrili kafes yapısında olup birbirine dik iki farklı düzlemde hareket etmektedirler. Üçüncü alt modül ise burulma hareketi yapmaktadır. Bu manipülatör için iki konumlu havalı eyleticilerin kullanıldığı varsayılmaktadır. Bu eyleticilerin ayrık yapısından kaynaklanan ayrıklaştırma hatalarını telafi etmek için manipülatörün en son modülüne küçük fakat sürekli eyletilen 6 serbestlik dereceli bir manipülatör eklenmiştir.

Bu aşırı artıksıl manipülatörün ters kinematik problemini çözmek için üç adet yöntem önerilmiştir. Bunlar eğriye uydurma, genişletilmiş eğriye uydurma ve çalışma uzayı doldurma yöntemleridir. Eğriye uydurma yöntemi, manipülatörün omurgasının (merkez çizgisinin) bir uzaysal referans eğrisine yaklaşmasına dayandırılmaktadır. Genişletilmiş eğriye uydurma yönteminde ise, birinci yöntemde bulunan sonucun bir genetik algoritma ile iyileştirilmesi sağlanmaktadır. Son yöntemde ise, manipülatörün çalışma uzayı rasgele bir şekilde ve yeterince fazla sayıda manipülatör biçimi örnekleri ile doldurulur. Eğer çalışma uzayının belli bir bölgesine odaklanmak istenilirse o zaman bu bölge bir genetik algoritmanın kullanımı ile doldurulmaktadır. Bundan sonra, istenilen biçime en yakın örnek, uygun bir arama algoritması ile tanımlanmaktadır.

Son olarak, AAM'nin hareketinin benzetimi iki ardışık manipülatör biçimi arasında yapılmıştır. Bunun için, manipülatörün hareket denklemleri, eyleticiler tarafından üretilen basınç kuvvetleri cinsinden elde edilmiştir. Daha sonra, AAM'nin performansını göstermek üzere bazı tipik uygulamalar için gerekli benzetimler gerçekleştirilmiştir.

Anahtar Kelimeler: Aşırı artıksıl manipülatör, ikili konumlu eyletilme, spline, genetik algoritma.

ACKNOWLEDGMENTS

First, I would like to acknowledge my supervisor, Prof. Dr. M. Kemal Özgören for his helpful suggestions, guidance and patience. Without his ideas and support this work would never have been possible.

I also wish to thank my Thesis Supervising Committee members, Prof. Dr. Aydan Erkmen and Prof. Dr. Kemal Ider for the hours of discussion and their valuable advices.

Finally, I thank my wife for all patience and encouragement during this study.

TABLE OF CONTENTS

ABSTRACT iv
ÖZ vi
ACKNOWLEDGMENTS viii
TABLE OF CONTENTS ix
LIST OF TABLES xii
LIST OF FIGURES xiii
LIST OF SYMBOLS xvi

CHAPTERS

1. INTRODUCTION 1
   - 1.1 Literature Survey 5
     - 1.1.1 Hyper Redundant Manipulators 5
     - 1.1.2 Binary Actuated Hyper Redundant Manipulators 6
     - 1.1.3 The Inverse Kinematics of Continuously Actuated Hyper Redundant Manipulator 12
     - 1.1.4 The Inverse Kinematics of Binary Hyper Redundant Manipulators 13
   - 1.2 The Contributions of This Thesis 15
   - 1.3 Organization of This Study 16
2. MODELING OF A HYPER REDUNDANT MANIPULATOR 17
3. FORWARD KINEMATICS OF THE HYPER REDUNDANT MANIPULATOR 24
   - 3.1 Introduction 24
   - 3.2 The Kinematics of the Submodules 25
   - 3.3 The Singularity Analysis 35
4. INVERSE KINEMATICS OF THE HYPER REDUNDANT MANIPULATOR 40

4.1 Introduction 40

4.2 The inverse Kinematics of the HRM 41

4.3 The Spline Fitting Method 44

  4.3.1 Splines 46

    4.3.1.1 Non-Uniform Rational Base Spline (NURBS) 48

    4.3.1.2 The Construction of Spline Parameters 52

  4.3.2 The Computer Algorithm 59

  4.3.3 The Intermediate Configurations for the Spline Fitting Method 64

4.4 The Extended Spline Fitting Method for the Inverse Kinematics of the HRM 68

  4.4.1 Genetic Algorithm 69

  4.4.2 Using Genetic Algorithm in the Extended Spline Fitting Method 73

  4.4.3 The Selection Process from the Configuration Set 75

4.5 The Workspace Filling Method 79

  4.5.1 Using Genetic Algorithm in the Workspace of the HRM 81

  4.5.2 The Selection Process from the Workspace 84

4.6 The Comparison of the Inverse Kinematics Methods 86

4.7 Summary and Results 92

5. DYNAMIC ANALYSIS OF THE HYPER REDUNDANT MANIPULATOR.......... 96

4.1 Introduction 96

5.2 The Virtual Elimination of the Excess Constraints for the Dynamic Analysis of the Hyper Redundant Manipulator 98

5.3 The Dynamic Analysis 103

  5.3.1 The Equations of Motion of the HRM 108

    5.3.1.1 The Elimination of the Constraint Forces 110

  5.3.2 Pneumatic Actuator Forces 112

  5.3.3 The Algorithm for the Computer Programming 116

5.4 COSMOSMOTION 123

5.5 The Comparison of the Joint Variables of Hyper Redundant Manipulator Found in COSMOSMOTION and the Computer Program in MATLAB …………………………………………….. 126

5.6 Summary and Results ..…………………………………... 128

6. THE APPLICATIONS OF THE HYPER REDUNDANT MANIPULATOR ……………………………………………….. 130

6.1 Applications ……………………………………………… 130

6.1.1 The Path Tracking …………………………………... 131

6.1.2 Grasping an Object …………………………………. 134

6.1.3 Wrapping an Object …..……………..…………… 136

7. CONCLUSION ………………………………………………… 138

REFERENCES …..……………………………………………………..… 144

APPENDICES

A. KINEMATICS OF THE FINE TUNING MANIPULATOR ……... 149

B. BEZIER CURVE AND ADDITIONAL FEATURES OF NURBS . 154

VITA ..………………………………………………………………….. 158

LIST OF TABLES

TABLES

Table 3.1 The Dimensions of Each Module ……………………………….. 34

Table 4.1 The Number of Possible Configurations Constructed Randomly and by Using the Genetic Algorithm for the Different Workspace Regions ……... 83

Table 4.2 Positional Errors with the Three Methods 91

Table 5.1 Applied Forces, Module Masses and Viscous Friction Coefficients ……………………………………………………………….…..... 119

Table 7.1 The Comparison of Three Inverse Kinematics Methods ………… 141

Table A.1 Denavite-Hartenberg Link Parameters for the FTM …………..... 149

LIST OF FIGURES

FIGURES

Figure 1.1 The Categorization of Hyper Redundant Manipulators ……. 2

Figure 1.2 The Elephant's Trunk Manipulator by Hannan ……..…….. 6

Figure 1.3 A 2D Binary VGT Manipulator ………………………….... 7

Figure 1.4 A 3D Binary Hyper Redundant Manipulator by Ebert-Uphoff ……………………………………………………………….. 8

Figure 1.5 Discretely Actuated Hyper Redundant (DAHR) Manipulator by Suthakorn ……………………………………...….. 9

Figure 1.6 A Prototype of the 3M-CaPaMan ...……………………….... 10

Figure 1.7 The Basic Design (a) and the Experimental Platform (b) of the BRAID 11

Figure 2.1 A Module of the HRM …..…………………………….... 18

Figure 2.2 The Submodule with The Variable Geometry Truss …..…... 19

Figure 2.3 The Submodule for Twisting Motion …………..……….. 19

Figure 2.4 The Hyper Redundant Manipulator ..…………………….... 21

Figure 2.5 Conversion of VGT Structure from Parallelogram to Trapezoidal Configurations …………………..………… 22

Figure 2.6 The Fine Tuning Manipulator Used in the HRM ……………. 23

Figure 3.1 The VGT Submodule ..…………………………………... 26

Figure 3.2 VGT-Mechanism Solutions ………………………………... 28

Figure 3.3 The Schematic View of the Rotational (Third) Submodule .. 31

Figure 3.4a The Different Manipulator Configurations with Planar Motion …………………………………………………… 34

Figure 3.4b The Different Manipulator Configurations with Spatial Motion …………………………………………………… 35

Figure 4.1 The Orientation Criterion for the Selection of the Configuration ……………………………………………. 42

Figure 4.2 The 8 Different Configurations of the VGT Submodule …... 45

Figure 4.3 The Contribution of the Basis Functions 48
Figure 4.4 The Control or de Boor Points and Polygon 49
Figure 4.5 The Spline Part for Different Values of w 51
Figure 4.6 Obstacle Avoidance 53
Figure 4.7 The Control Polygon 55
Figure 4.8 The Radius of Curveture for the Module 55
Figure 4.9 The Curvature and Torsion for the Spline 57
Figure 4.10 The Azimuth (ϕ), Elevation (θ) and Twist (ψ) Angles 58
Figure 4.11 The Flow Diagram of the Spline Fitting Algorithm 61
Figure 4.12 The Solution Configuration with the Spline Fitting Method . 62
Figure 4.13a The Configurations for Two-Based and One-Based Module Fitting 63
Figure 4.13b The Configuration for Two-Based Module and One-Based Module Fitting 63
Figure 4.14 The Construction of Intermediate Configuration 66
Figure 4.15 The Intermediate Configuration 67
Figure 4.16 The Trajectory Control of the Intermediate Configuration ... 68
Figure 4.17 Crossover and Mutation Operators 72
Figure 4.18 The Extended Spline Fitting Method 74
Figure 4.19 The Shape Criterion for the Selection of Configuration 77
Figure 4.20 The Trajectory Following Criterion 78
Figure 4.21 The Workspace with 100 000 Configurations 80
Figure 4.22a The Workspaces for Different Task Regions 82
Figure 4.22b The Top View of the Workspaces 83
Figure 4.23 The Possible Configurations and Related Trajectories 84
Figure 4.24 The Selected Configuration by Using the Workspace Filling Method 85
Figure 4.25 Avoiding the Moving Obstacles 86
Figure 4.26 The Planar and Linear Configurations 87
Figure 4.27 The Planar and Curve-Like Configurations 88
Figure 4.28 The Spatial Configurations 88
Figure 4.29 The Spatial Configurations 89
Figure 4.30 The Configurations for the Inconvenient Reference Curves . 89

Figure 5.1. The Module Joints of the Manipulator ..…………………… 99
Figure 5.2 The Assumption for the Joint Types of the Module ………. 102
Figure 5.3. The Free Body Diagrams of the 2nd and 3rd Links ………… 105
Figure 5.4 The Free Body Diagram of the Link 8 ………..…………. 106
Figure 5.5 Pneumatic Serial Circuit …………………………………. 113
Figure 5.6 The Piston Force versus Piston Velocity ...………………… 115
Figure 5.7 The Piston Force versus Piston Velocity for $V = 0 - 0.5 m/s$ 116
Figure 5.8 The Initial and Final Configurations for the Dynamic Analysis ……………………………………………………….. 119
Figure 5.9a The Joint Variables for the Module 1 ……………………… 120
Figure 5.9b The Joint Variables for the Module 4 ……………………… 120
Figure 5.9c The Joint Variables for the Module 6 ……………………… 121
Figure 5.9d The Joint Variables for the Module 9 ..……………………. 121
Figure 5.10 An Example of Non-Implemented Configuration Set ……. 122
Figure 5.11a The Joint Variables for the Module1 ..……………........... 123
Figure 5.11b The Joint Variables for the Module 3 …………………..... 123
Figure 5.12 The General View of SolidWorks with COSMOSMotion 124
Figure 5.13 General Process of Using COSMOSMotion ……………….. 125
Figure 5.14 The Joint Variables for the Module 1 ……………………… 127
Figure 5.15 The Joint Variables for the Module 2 ……………………… 127
Figure 6.1 The Selected Configurations for the Path Tracking ..……… 132
Figure 6.2 The Path Tracking Procedure ..…………………………… 133
Figure 6.3 The Trajectory of the HRM for the Path Tracking ..………. 133
Figure 6.4 Inconvenient (a) and Convenient (b) Target Positions for the FTM ……………………………………………………….. 134
Figure 6.5 The Schematic of Grasping an Object …………………….. 135
Figure 6.6 The Grasping an Object ..……………………..…………. 136
Figure 6.7 Wrapping an Object Graphically ..………………………… 137
Figure 6.8 Wrapping an Object in SolidWorks Environment ………… 137
Figure A.1 Schematic View of the Fine Tuning Manipulator and its Joint Variables and Dimensions ………………………………..… 150
Figure B.1 Bezier Cubic Curve and its Bezier Polygon $B_1B_2B_3B_4$ ……. 155

LIST OF SYMBOLS

$\vec{a}_{\mathrm{c}}$: Commanded Acceleration Vector

$\overline{a}_{c}^{(k)}$: Column Representation of $\vec{a}_{\mathrm{c}}$ in F_k (k=1, 2,....n)

$\hat{C}^{(a,b)}$: Transformation Matrix from *Fa* to *Fb*

Ci : Mass Center of Body i (i=1, 2)

d_0, d_1, d_2 : The Dimensions of the Module

$\mathcal{F}_0$: Earth-Fixed Reference Frame

$\mathcal{F}_{\mathrm{i}}$: Reference Frame of Platform *i* of the Manipulator

$\vec{g}$: Gravity Vector

$\overline{g}^{(b)}$: Column Representation of the Gravity Vector in *Fb*

$\hat{J}_{C_{\mathrm{i}}}^{(b)}$: Inertia Matrix of Body i expressed in *Fb* (i=1, 2,...)

$\mathrm{m_i}$: Mass of Body i (i=1, 2,..)

$\vec{u}_{\mathrm{j}}^{(\mathrm{k})}$: j'th Unit Vector of the reference frame *Fk* (j=1, 2, 3 and k=0,1,2,......)

$\vec{v}_{P/Q}$: Velocity Vector of Point P with respect to Point Q

$\alpha_{\mathrm{i}}, \beta_{\mathrm{i}}, \gamma_{\mathrm{i}}, \theta_{\mathrm{i}}, \phi_{\mathrm{i}}$: Manipulator Angles

$\vec{\omega}_{\mathrm{b/e}}$: Angular Velocity Vector of *Fb* with respect *Fe*

$\tilde{\omega}_{\mathrm{b/e}}^{(\mathrm{c})}$: Tilda Matrix of $\vec{\omega}_{\mathrm{b/e}}$ expressed in *Fc*

CHAPTER 1

INTRODUCTION

Most of the industrial robot manipulators are traditional ones with 6 or less than 6 degrees of freedom (DOF). A typical 6 DOF robot manipulator uses three actuated DOF for tip point positioning and the other three for hand orientation in order to reach any desired point in its workspace with arbitrary orientation. Adding more degrees of freedom to a manipulator may sometimes be necessary so as to provide enhanced manipulability and flexibility, to avoid singularity and obstacles and to optimize the specified performance index. Having more actuated degrees of freedom than are required to perform a specified task, a manipulator is called redundant. If the number of additional degrees of freedom is too many or infinite, such a manipulator is called hyper redundant manipulator. These types of manipulators, an alternative to traditional ones in some applications, have very large actuated degrees of freedom. Their designs may consist of flexible physical structures, a cascade of modules composed of redundantly actuated parallel platforms, or a large number of rigid links in series. Hyper redundant manipulators can be classified into three categories namely “articulated”, “continuum” and “cascade” as shown in Figure 1.1. An articulated manipulator is simply an open kinematics chain such as serpentine or snake-like. On the other hand, continuum type of manipulators can exhibit a large number of degrees of freedom, but, not all of these degrees of freedom are directly actuated, and have no rigid links and joints, and their actuation is distributed over the manipulator length. Trunk and tentacle-like devices belong to this category. The third group is the cascade manipulators which consist of modules stacked on top of each other in serial chain. Such modules are actuated in parallel, and could be variable geometry trusses, closed linkages with or without redundant actuation or combination of other closed loop mechanisms. Variable geometry trusses mostly used in this category are themselves parallel structures and each of them is a good example for demonstrating both kinematics and actuator redundancy.

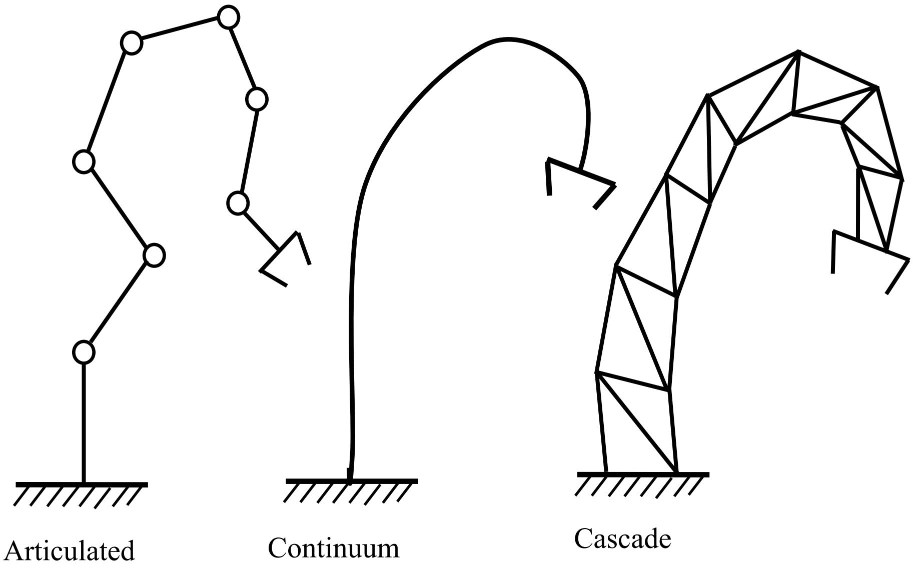


Figure 1.1 The Categorization of Hyper Redundant Manipulators

Hyper redundant manipulators (HRM) have been given a wide variety of names such as “swan’s neck”, “tentacle”, “snake-like”, “elephant trunk”, “tensor arm”, “highly articulated”, “active cord”, etc. Successfully implemented hyper redundant manipulators would offer great advantages in dexterity over non-redundant manipulators. For example, they have better performance for obstacle avoidance with many actuatable degrees of freedom, they can implement the given tasks though some of the actuators fail, and they are used in unknown environments. However, some drawbacks limit the use of HRMs. These are the lack of an efficient method for the inverse kinematics and motion planning, difficulty in mechanical design and some deficiencies in actuator technology. HRMs can be powered with continuous actuators such as tendons, DC motors or hydraulic cylinders, and also discrete or binary actuators.

In binary actuation, actuators have only two stable states, such as solenoids or pneumatic cylinders. This group is called "discretely actuated hyper redundant" or "binary hyper redundant" manipulator. Binary hyper redundant manipulators can be used in many applications such as pick-and-place, spot welding, repairing and inspection in dangerous and complex areas such as space satellites, nuclear powers, bridges. There are some advantages and disadvantages about this type of manipulators as follows.

Advantages of binary HRMs:

- a finite set of stable actuator states
- much cheaper actuators
- light weight
- high repeatability
- high capacity load to manipulator weight ratio
- no need for feedback
- less complexity in computer controlled interfacing
- allow tasks to be performed even when some actuators fail
- high ability in obstacle avoidance because of its hyper redundancy
- new forms of robot locomotion and grasping.

Disadvantages of binary HRMs:

- non-continuous (discrete) workspace
- need of more modules with less dimensions to improve its performance
- complicated structure and more moving parts
- need of a large amount of computations in inverse kinematics
- difficulty to control the trajectory between successive configurations.

In this thesis, a 3D hyper redundant manipulator actuated discretely is to be devised and controlled by the inverse kinematics methods and an additional continuously actuated manipulator. This manipulator consists of many modules stacked on top of each other in a long serial chain. Each module has three submodules. The first two are binary variable geometry trusses (VGT) and the last one is a discretely rotated submodule. The manipulator structure is very simple and powered with on-off pneumatic actuators, each of which is either fully extended or fully contracted. Because of the discrete nature of the binary actuators, for fine tuning, a continuously actuated manipulator is added as the last module of the HRM. To solve the inverse kinematics of this HRM, three methods are offered with respect to the given tasks. These are the spline fitting, the extended spline fitting and the workspace filling methods. The spline fitting method is based on a spatial reference curve, which is the non-uniform rational base spline (NURBS). The second method is the extended spline fitting method. It is the combination of the spline fitting method and a genetic algorithm. This method generates a solution set consisting of many possible configurations rather than one configuration. The best configuration is selected from this set in such a way that it supplies all the given criteria which are the minimum positional and orientational errors, the shape of the manipulator and the trajectory tracking. The third method, which is the workspace filling method, is a stochastic search method based on the workspace of the manipulator. Despite the whole number of the possible configurations, the workspace involving a finite number of the manipulator configurations is generated randomly in this method. If it is desired that to have concentration on a particular region, some workspaces with respect to the given tasks are filled by using a genetic algorithm. The control of the manipulator is based on the values of the joint variables, which are the pneumatic on-off actuator lengths represented by “0”s or “1”s, from the inverse kinematics analysis and the compensation by the additional continuously actuated part of the HRM. Then, the equations of motion of the HRM consisting of rigid bodies are developed with the dynamic analysis in terms of the pressure forces generated by the binary pneumatic actuators. This analysis is module-based and made recursively, which is useful in developing general purpose computer algorithms. As a result of the dynamic simulation, the real motion of the HRM is obtained and also it is investigated whether this motion upon the given actuator forces can come true or not. Finally, the necessary simulations are carried out to demonstrate the performance of the HRM in some typical applications.

1.1 Literature Survey

The literature survey consists of four subsections. Subsection 1.1.1 provides examples of hyper redundant manipulators with modeling and actuation. Subsection 1.1.2 presents the literature review on examples of binary hyper redundant manipulators. Subsection 1.1.3 discusses common approaches for the inverse kinematics of redundant and hyper redundant manipulators. Subsection 1.1.4 cites some methods for the inverse kinematics of binary hyper redundant manipulators. Section 1.2 is related to the contributions of this thesis.

1.1.1 Hyper Redundant Manipulators

Researchers have studied many types of hyper redundant manipulators. Most of them are continuously actuated manipulators which are powered with actuators such as DC motors or hydraulic cylinders. Although hyper redundant designs were presented previously in 1960's [1], Chirikjian and Burdick used firstly "hyper redundant" term in the robotics area [2]. In the literature, there are many types of hyper redundant manipulators with respect to their structure or actuation type. The ACM III (active cord mechanism) and The ACM VII (The Elastor) are two hyper redundant manipulators by Hirose [3]. Another hyper redundant manipulator studied by Hirose and his coworkers is The Float Arm [4]. It is actuated with a cable system called M-drive. One of the famous studies is a 30-DOF planar hyper redundant manipulator by Chirikjian and Burdick [5], [6] and [7]. This manipulator consists of 3-DOF variable geometry truss (VGT) structures and contains 10 identical modules. Each module has prismatic joints which are driven with DC servo motors and screws. As a continuum type of manipulator, The Elephant's Trunk manipulator as shown in Figure 1.2 is constructed and tested by Hannan and Walker [8]. The manipulator is composed of a serial connection of 16 two DOF joints and it is divided into four actuated parts. Each part is driven with a cable servo system. Moreover, many researchers have focused on serpentine robots belonging to the articulated category. For example, Howie Choset and his coworkers studied some types of serpentine robots at Carnegie Mellon University [9] and [10]. Kimura et. al. [11] introduced the hyper redundant manipulator arm test system for a part of the technology of "Orbital Maintenance System (OMS)" developed in Japan. This manipulator consists of seven 2-DOF modules and is designed for reorbiting and repairing satellites in space.

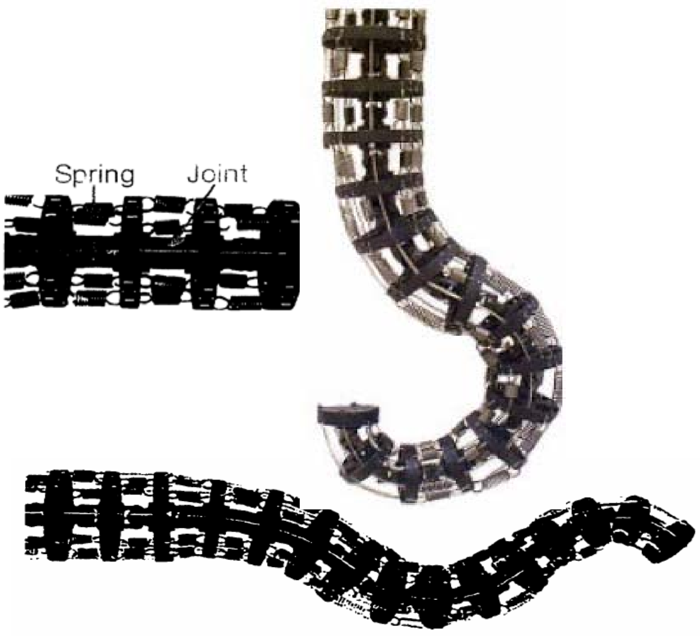


Figure 1.2 The Elephant's Trunk Manipulator by Hannan

Mark Yim and Chirikjian introduce reconfigurable robots and have used them in many applications [12], [13], [14]. Instead of designing modules for specified tasks, many copies of one simple module are built to implement various complex tasks. In reconfigurable robots, many modules are connected together to form a new system.

1.1.2 Binary Actuated Hyper Redundant Manipulators

Compared to the continuously actuated counterparts, binary manipulators need no feedback to control and no complex interfacing between computer and manipulator. Also, they are lightweight, robust and have higher repeatability. These manipulators generally use actuators with two stable states such as on-off pneumatic cylinders, solenoids or shape memory alloys (SMA). In the literature, many different types of discretely actuated hyper redundant manipulators are presented with their experimental studies. Some typical studies on them are presented briefly as follows.

- The Koliskor Study: In the former Soviet Union, Koliskor developed discretely actuated arms. His study reduces the complexity of the control system and the computer interfacing. [15].

- The planar Binary Robot Manipulator by Chirikjian [16]: Chirikjian is one of the famous researchers over the concept of hyper redundant manipulators. He introduced a binary actuated planar manipulator as shown in Figure 1.3. This manipulator includes five modules stacked on top of each other. Each module consists of 3-bit planar VGT structures and is actuated with pneumatic cylinders. The number of actuators used is $3 \times 5 = 15$ so that its end effecter can reach $2^{15} = 32768$ points in 2D space. This device was developed and tested in the Robot and Protein Kinematics Laboratory at Johns Hopkins University.

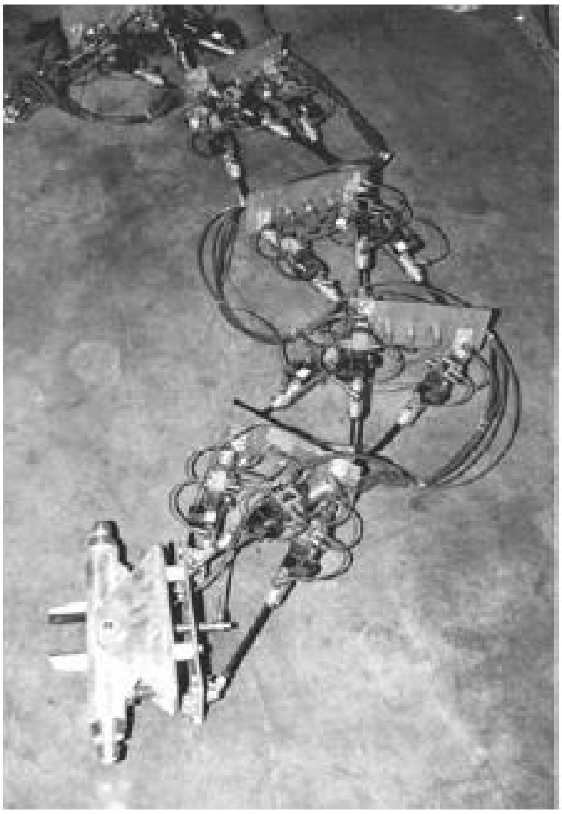


Figure 1.3 A 2D Binary VGT Manipulator

- The 3D Binary Manipulator by Ebert-Uphoff [17]: The device was also developed and tested in the same laboratory at Johns Hopkins University. Figure1.4 shows the 3D binary manipulator consisting of 6 modules each of which is influenced by the Stewart platform. It is attached vertically to the ceiling and has a 3D gripper at the end as an end effector. Each module consists of 6 binary pneumatic actuators. Therefore, the manipulator configurations can be $2^{6\times6}(\approx 68.7\ billion)$ different reachable states in 3D space. In the device, the fluid filled pneumatic cylinder is attached in parallel to each actuator to function as a dashpot. Thus, the speed of actuators and also vibrations are suppressed.

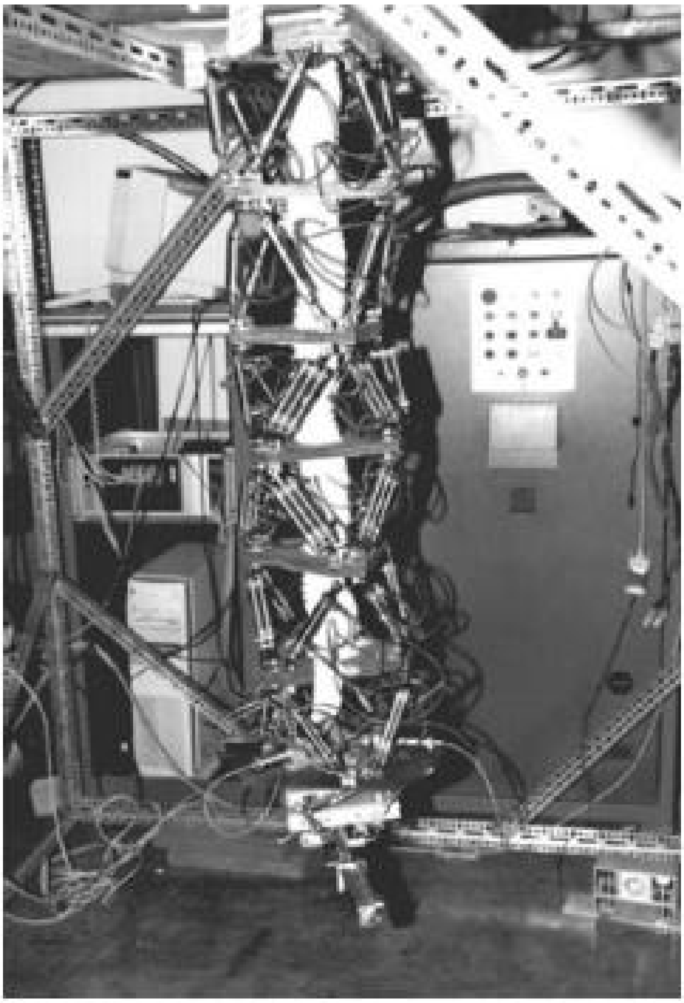


Figure 1.4 A 3D Binary Hyper Redundant Manipulator by Ebert-Uphoff

- Discretely Actuated Hyper Redundant (DAHR) Manipulator by Suthakorn [18], [19], [20]: The new manipulator design involves 3-bit binary VGT modules stacked on top of each other and discretely actuated rotating joint between each module. The actuators used in the prototype are the on-off pneumatic cylinders and the discrete rotary pneumatic actuators. The prototype as shown in Figure 1.5 consists of three modules, each has two parts; 3-bit binary VGT structure and a rotating joint with 16 steps. Therefore, the manipulator totally has $2^{3\times 3} \times 16^3 (\approx 2.1\, million)$ different configurations in 3D space. This was also studied at Johns Hopkins University.

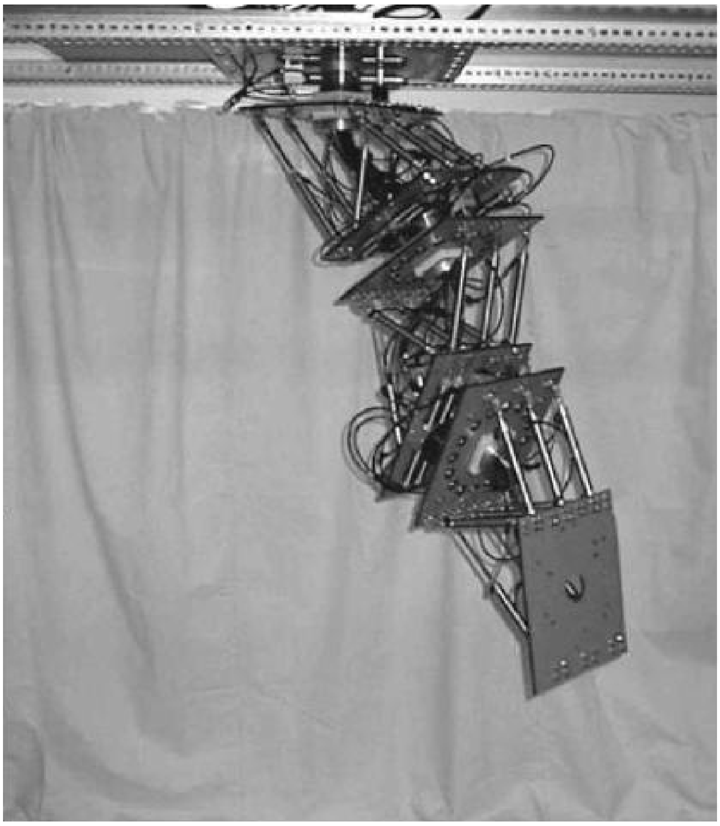


Figure 1.5 Discretely Actuated Hyper Redundant (DAHR) Manipulator by Suthakorn

- CaPaMan 3 (Cassino Parallel Manipulator Version 3) [21]: This manipulator is a discretely actuated spatial parallel manipulator and designed with binary pneumatic actuation. The manipulator consists of three modules with 3 DOF parallel manipulators. Each module has three legs composed of slider-crank mechanism. The actuation is made with this mechanism. The manipulator shown in Figure 1.6 was built and tested at LARM (Laboratory of Robotics and Mechatronics in Cassino University-Italy).

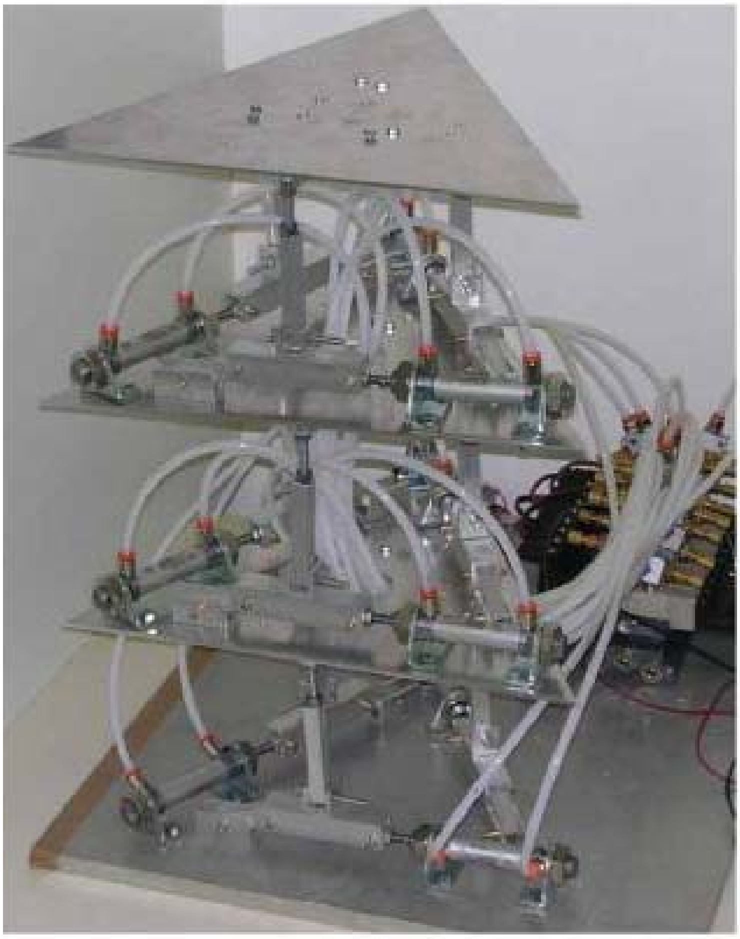
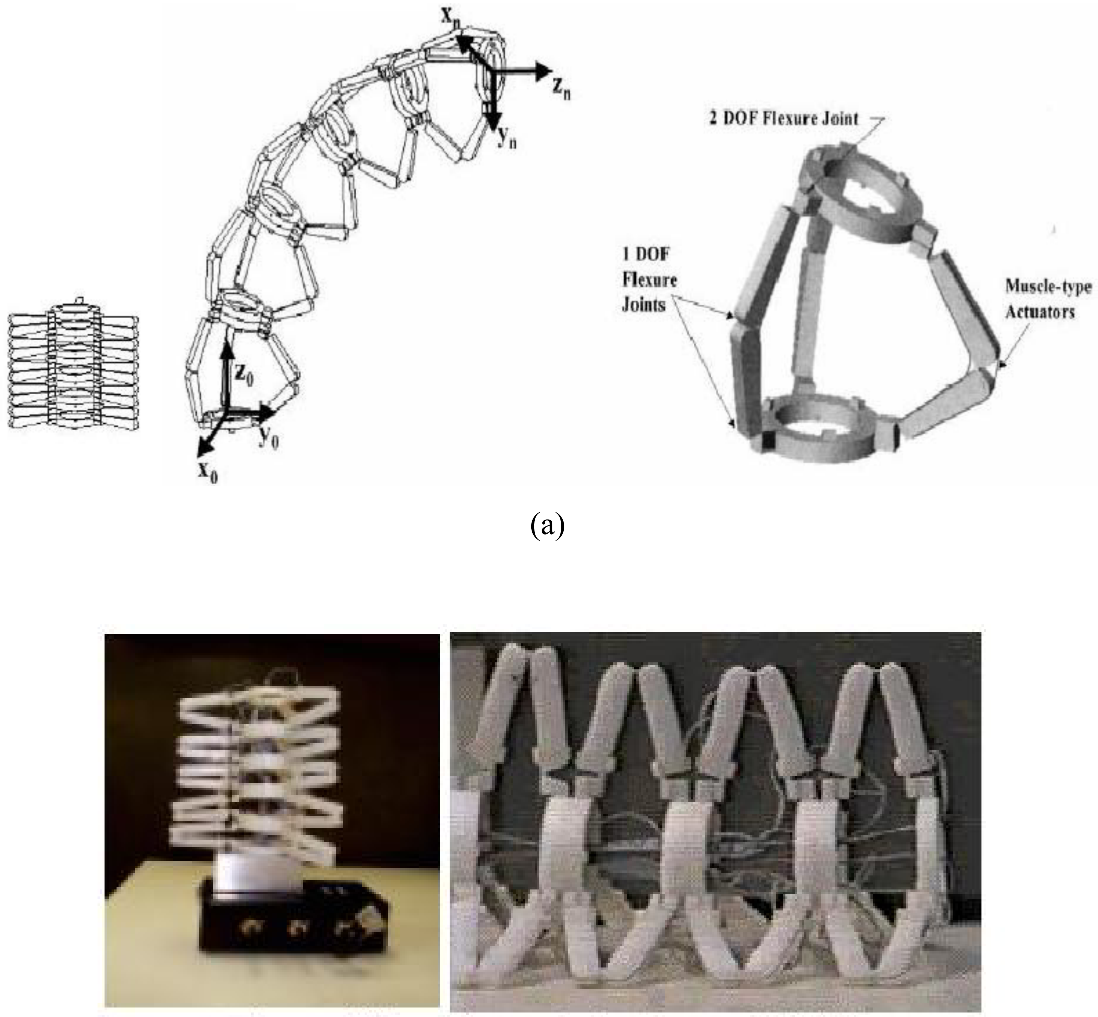


Figure 1.6 A Prototype of the 3M-CaPaMan

- The Binary Robotic Articulated Intelligent Device (BRAID) by Sujan and Dubowsky [22], [23] and [24]: Instead of using rigid joints and actuators, the hyper redundant deployable Binary Robotic Articulated Intelligent Device (BRAID) uses muscle type binary actuators and flexure linkages. This device has been simulated in manipulation and locomotion. Moreover, the prototype of this manipulator is developed and tested in Field and Space Robotics Lab. MIT, Cambridge. The

BRAID element is made of a serial chain of parallel stages. Each three DOF stage has three flexure based legs made of polyethylene, and muscle type binary actuator which is shape memory alloy. The experimental BRAID consists of five parallel stages so that the manipulator has totally 15 DOF and $2^{15} = 32768$ discrete configurations in 3D space. It is shown in Figure 1.7.



(a)

(b)

Figure 1.7 The Basic Design (a) and the Experimental Platform (b) of the BRAID

1.1.3 The Inverse Kinematics of Continuously Actuated Hyper Redundant Manipulators

The high degree of redundancy in HRMs causes difficulty in the calculation of workspaces and inverse kinematics, and the planning of their trajectories. Although some conventional methods such as pseudo-inverse or augmented Jakobian inverse kinematics methods can be efficient for redundant manipulators, because of the larger degrees of freedom, these methods are not efficient for hyper redundant manipulators. The methods studying inverse kinematics of HRMs are generally based on a spatial curve known as "backbone curve". A backbone curve is a piecewise continuous curve that captures the important macroscopic geometric features of a hyper redundant manipulator.

To efficiently solve the inverse kinematics problem, a "modal" approach is introduced. In this approach, a set of intrinsic backbone curve shape functions are used which are restricted to certain modal forms [25]. The problem reduces to determining the time varying backbone curve behavior. In [26], the backbone curve is modeled by cubic or quintic piecewise parametric spline segments. The concept of constant curvature sections and the use of differential geometry into a modified Denavite-Hartenberg procedure is another approach to determine the kinematics [27]. Again, the calculus of variations is used to develop differential equations whose solution is the optimal backbone curve shape [28]. In all approaches, the bending, twisting and extension are taken into consideration. The inverse kinematics of HRMs is almost based on a backbone curve, but, only the fitting algorithms may be different and construction of backbone curve is different. The criterion for inverse kinematics method is to configure the manipulator to adhere to the backbone curve, or discretely segmented morphologies fitted to the curve. Minimizing the elastic potential energy of the backbone curve which is the elastic cord [29] is another method for constructing backbone curve. With the inverse kinematics of HRMs, some studies have focused on the trajectory planning, obstacle avoidance and workspace generation [5], [30] and [31].

In addition to the motion planning with mathematical tools, some manipulators especially locomotion manipulators called snake-like, use sensors for their motion controls [32]. In [33], a sensor-based method is used to control a hyper redundant manipulator. In this method, the full shape modification and partial shape modification work at different sampling rates and at different spatial resolutions. These two classes of sensor feedback control methods are responsible for the shape changes of the hyper redundant manipulator, and the obstacles and task constraints in the environment. These methods have been implemented on a-thirty degrees of freedom planar hyper redundant manipulator which has eleven ultrasonic distance measurement sensors and twenty infrared proximity sensors.

1.1.4 The Inverse Kinematics of Binary Hyper Redundant Manipulators

Compared to the continuously actuated manipulators, the binary manipulators have lower hardware cost. However, there exists a difficulty on the motion planning of binary hyper redundant manipulator since the number of configurations grows exponentially with the number of actuators. There is a tradeoff in the complexity of motion planning software. Position and trajectory control of a manipulator require the calculation of its inverse kinematics, but, there is not analytical solution for the inverse kinematics of binary hyper redundant manipulators. A number of efforts have been made to study the kinematics and control of hyper redundant manipulators with binary or discrete actuation. For the binary manipulator with few actuators, the brute force search or exhaustive search can be efficient since its workspace requires the computational effort of $O(2^n)$. A manipulator considered with larger number of actuators, for example, 50 binary actuators; it can reach 2^{50} distinct possible configurations. This makes the brute force search impractical. To avoid exponential growth in the configuration search space as the number of actuators grows, the combinatorial method is presented for computing the inverse kinematics and trajectory planning of binary manipulators [24], [34] and [35]. Changing only a small number of configurations at a time, the method reduces the size of search space considerably, and produces very smooth motion. Because of the discrete nature, the manipulators can only reach finite number of states. Therefore, the workspace and the distribution of the points inside the boundary of this workspace are very important tools for the inverse kinematics. The researchers studied many methods to generate workspace for this type of manipulators [31], [36], [37], [38] and [39]. The methods break up the workspaces into pixels/voxels in the planar/spatial case and, calculate how many end effector positions in each of these pixels/voxels are reached. In [36], after constructing the workspace of binary manipulator with the workspace density algorithm, the inverse kinematics method generate the solution satisfying the geometric constraints of the task, and in the second part, the strategies are discussed to resolve the remaining redundancy to avoid abrupt transitions between configurations. In [39], the concept of the mean of workspace density function is used to generate the workspace. The solution of inverse kinematics is based on minimization of the metric containing the orientational and positional quantities.

The backbone curve specifying the shape which the manipulator attains is used to solve the inverse kinematics of binary manipulators [40]. After solving the inverse kinematics for the continuously actuated counterpart of the binary HRM, the suitable configuration satisfying the given constraints is selected with respect to minimizing an error measure. The error measure includes the sum of the distances between the same characteristic points on the continuum model and the possible configuration of the binary manipulator. Along the same line, a continuous variable-based optimization method for the inverse kinematics method of binary manipulators was presented in [41]. In the optimization with gradient-based-search method, the distance between the end effector and the target is constrained, and the objective is to push the continuous design variables towards the permissible binary values.

Instead of finding actuator length as a joint variable, generally, on-off values represented by “0” for fully contracted positions and “1” for fully extended positions are searched. All possible configurations would be represented by bit-sequences such as 011001…..0110. This can in nature make genetic algorithms to be efficient for solving the inverse kinematics of binary HRMs [24], [42], [43] and [44]. Genetic algorithms are used as a minimization procedure. The problem is solved by minimizing the end effector position error for the given target position. These algorithms are executed to obtain inverse kinematics solution with a small error bound and density map of workspace, and also are used to generate the divided workspaces for higher density [45]. The details will be discussed in the following chapters of this thesis.

Concerning the binary HRMs, another important challenge is following continuous trajectories between two successive discrete configurations. The trajectory tracking problem is quite different for binary HRMs from that of continuously actuated ones. The subject is discussed in [34], [35], [42] and [45].

1.2 The Contributions of This Thesis

The following paragraphs sum up the contributions of this thesis.

A new 3D binary actuated hyper redundant manipulator with improved modular rigidity has been devised which may used in a variety of practical applications.

In addition to the binary actuation, for fine tuning, a continuously actuated 6 DOF manipulator (the fine tuning manipulator (FTM)) is added to the HRM as the last module. This additional module considerably improves the control accuracy and the applicability of the binary HRMs.

In this thesis, new efficient inverse kinematics methods, such as the *extended spline fitting method* and the *workspace filling method*, have been presented for binary HRMs. Unlike the other methods in the literature, which focus only on positional error or positional error together with trajectory tracking, the extended spline fitting method focuses on four criteria which are the positional error, orientational error, desired shape of the manipulator and trajectory tracking. On the other hand, the workspace filling method is particularly useful for successive multiple tasks and it provides quick configurational options to avoid moving or unexpected obstacles.

In this thesis, a simple approach is given for the dynamic analysis of the HRM, which by-passes the necessity of computing the structural reaction forces and moments. This approach decreases thus the computation burden and simulation time considerably.

1.3 Organization of This Study

This thesis consists of seven chapters. Chapter 1 gives an introduction and literature survey about hyper redundant manipulators. Chapter 2 presents the modeling of the hyper redundant manipulator studied in this thesis and gives some modifications to improve the performance of the manipulator. Chapter 3 studies the forward kinematics and singularity of the HRM. Chapter 4 discusses the three developed methods for solving the inverse kinematics of the HRM. In Chapter 5, dynamics analysis of the manipulator is performed and discussed. Chapter 6 involves some applications of the HRM. Finally, Chapter 7 presents conclusions and suggestions for future research.

CHAPTER 2

MODELING OF A HYPER REDUNDANT MANIPULATOR

This section contains a description of modeling and features of a 3-D binary actuated hyper redundant manipulator. In this model, the elements of a 2-D binary actuated variable geometry truss (VGT) manipulator and the elements of a mechanism with only rotational motion about one axis are joined.

At first, the detailed information about the structure of the HRM is given and the way it works is discussed. This manipulator consists of many modules stacked on top of each other in a long serial chain. A module of the HRM is shown in Figure 2.1. Each module of the HRM contains three submodules. The first and second of them are VGT structures, and the third one is a mechanism that makes only a single rotational motion about one axis. VGT manipulators are themselves parallel structures and this kind of structures are suited for demonstrating for both kinematics and actuator redundancy. Detailed information about VGTs can be found in [46]. As shown in Figure 2.2, the VGT-based submodules used in the HRM have only planar motion in their own reference frames. For example, assigning $\mathcal{F}_0$ as the reference frame attached to the first platform, the first one can only move in the $\vec{u}_2^{(0)} - \vec{u}_3^{(0)}$ plane. The second platform, which carries the reference frame $\mathcal{F}_1$, rotates about $\vec{u}_1^{(0)}$ axis and translates in the direction of $\vec{u}_2^{(0)}$ and $\vec{u}_3^{(0)}$ relative to the reference frame $\mathcal{F}_0$. Here, for the identification of the unit vectors $\vec{u}_i^{(k)}$, the subscripts 1,2,3 indicate the directions of x, y and z axes respectively. The superscript (k) indicates the reference frame number. The second submodule is placed on the top of the first submodule in such a way that its motion takes place only in the $\vec{u}_1^{(1)} - \vec{u}_3^{(1)}$ plane. The third platform, which carries the third reference frame $\mathcal{F}_2$, rotates about $\vec{u}_2^{(1)}$ and translates in the direction of $\vec{u}_1^{(1)}$ and $\vec{u}_3^{(1)}$ relative to the reference frame $\mathcal{F}_1$. Each module of the HRM involves three different reference frames. These are $\mathcal{F}_{3(\text{n-1})}$, $\mathcal{F}_{3(\text{n-1})+1}$ and $\mathcal{F}_{3(\text{n-1})+2}$ which are attached to the base or first platform, the second platform and the third platform of each module respectively. Here, n indicates the number of the discretely actuated modules.

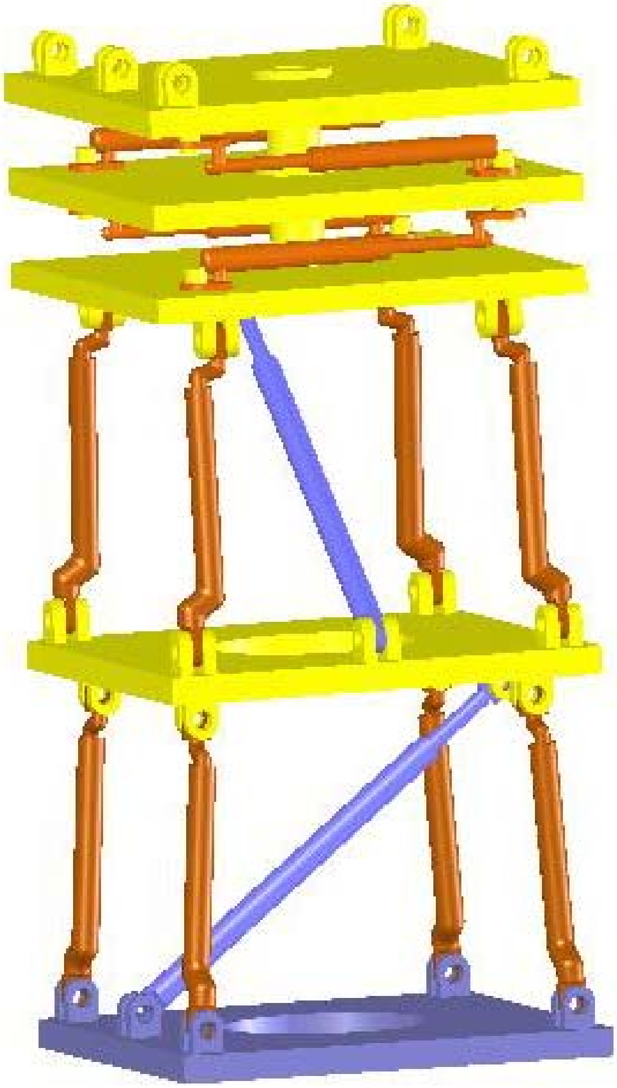


Figure 2.1 A Module of the HRM

In the last stage of construction of a module, the third submodule placed on top of the second submodule makes only rotational motion about the axis $\vec{u}_3^{(2)}$. In fact, this third submodule consists of two parts as shown in Figure 2.3. The first part supplies the rotational motion in CCW direction whereas the second one supplies the rotational motion in CW direction.

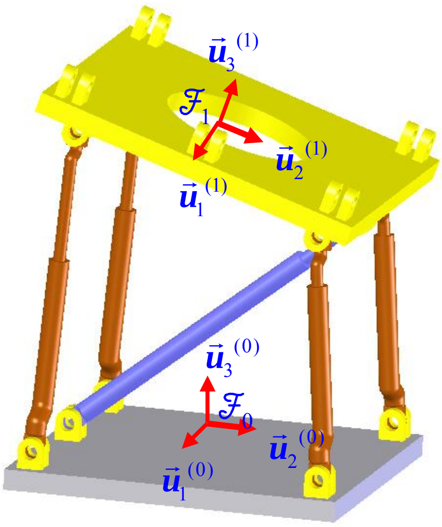


Figure 2.2 The Submodule with The Variable Geometry Truss

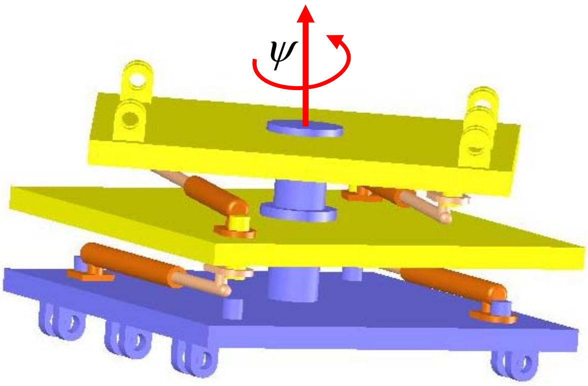


Figure 2.3 The Submodule for Twisting Motion

Because of binary actuation, such a design provides four angular configurations denoted by the binary set {00, 10, 01, 11} but they lead to only three distinct resultant angular positions denoted by the set $\{-\psi^*, 0, +\psi^*\}$ (here, ψ^* is the specified value for the twist motion). This is because both of the configurations 00 and 11 imply zero net rotation. This type of submodule is used to supply a twist motion for the HRM. It is assumed that all platforms are connected with rigid links and joints.

The HRM is powered with on-off pneumatic cylinders, each of which has either fully extended or fully contracted positions. In other words, the actuation type is binary in which the actuators have two stable states. This causes the challenge that the motion is very jerky. This problem can be eliminated with a damping element. In real systems, this damping effect is supplied with placing a dashpot in parallel to one or more pneumatic cylinders, but, in the modeling of the HRM, the pneumatic cylinders are assumed to have enough damping. The pneumatic actuators are controlled by solenoid valves. In pneumatic cylinders, the piston stops suddenly by crashing to the cylinder. In some studies in the literature, it is expressed that adding a rubber bumper to both end of the piston, the sudden collision can be eliminated. Besides, it is expressed also that this helps to avoid wear and tear, and high frequency transients, and also helps to ramp the speed up to steady state while the actuator is starting from its rest position, and allows for gradual deceleration at the other end. However, using a rubber bumper causes undesired deflection of the whole system. Especially, small deflections on the actuators towards the base of the manipulator may cause the larger errors for the tip point of hyper redundant manipulators.

All actuators on a VGT structure have the same dimensions whereas the actuators on the diagonal have different dimensions. The actuators are initially mounted in such a way that there exist not any rotations between the reference frames when all actuators are at off-positions, but, there are only offsets between the reference frames. In a module, the first and second submodules with VGT structure make planar motions with respect to their own reference frames, but, they are arranged orthogonally with respect to each other, and each of them 3 degrees of freedom. On the other hand, the third submodule has one DOF only. Thus, a module has 7 DOF in total for position and orientation of its uppermost platform in 3D space.

The conceptual design of the manipulator is based on two particular notions. The hyper redundant manipulator consists of many modules. The size of each module decreases with respect to the previous one. So, the manipulator shape becomes like a cone as shown in Figure 2.4. In addition to constructing the cone shape assembly, the structure of VGT based submodules is set up to be in the shape of a trapezoid rather than a parallelogram as shown in Figure 2.5. If the VGT submodule were constructed like a parallelogram, then it could be deformed easily because only the reaction moments at the bottom would hold the system. However, with the trapezoidal form, in addition to the reaction moments, the forces in the links support the system too. In the VGT submodules, the two identical nondiagonal actuators are symmetrically placed according to the motion planes such as the $\vec{u}_2^{(0)}-\vec{u}_3^{(0)}$ and $\vec{u}_1^{(1)}-\vec{u}_3^{(1)}$ planes as shown in Figure 2.2. Each pair makes the same motion.

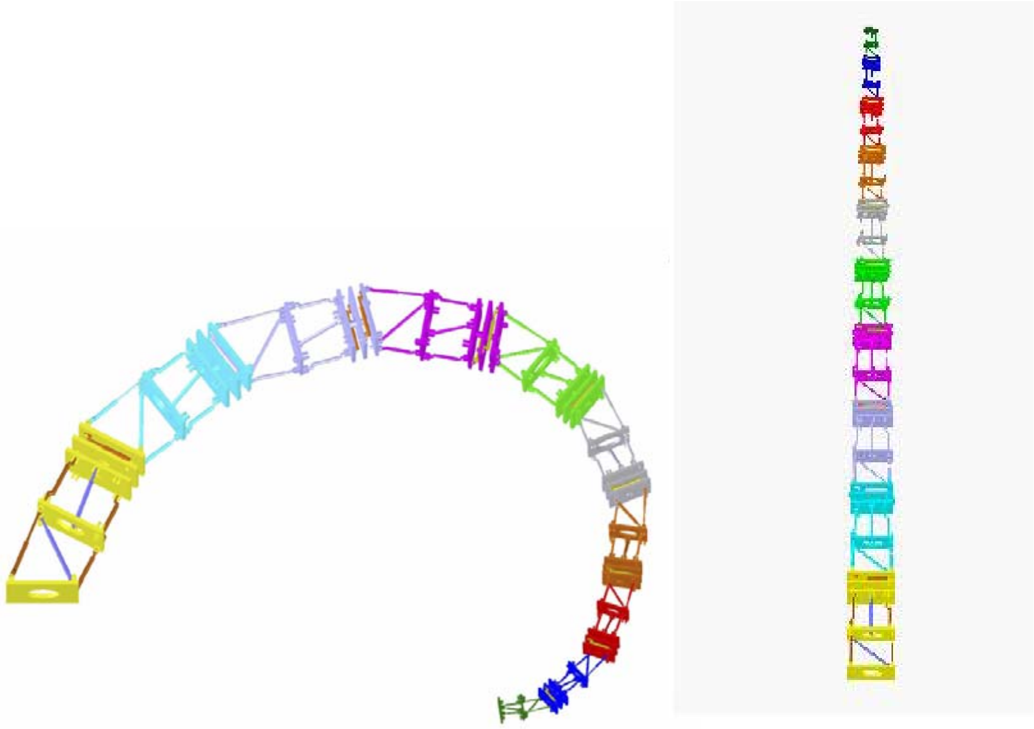


Figure 2.4 The Hyper Redundant Manipulator

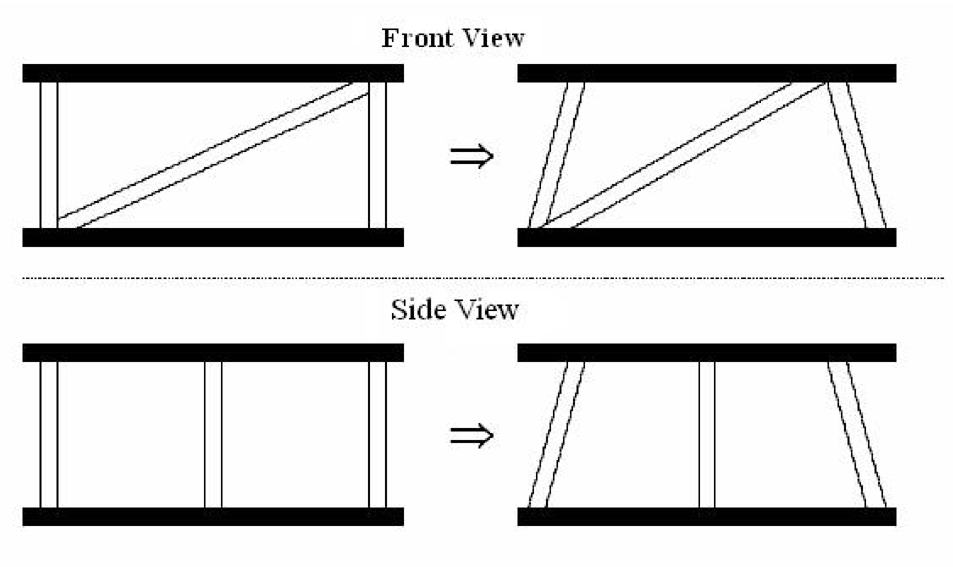


Figure 2.5 Conversion of VGT Structure from Parallelogram to Trapezoidal Configurations

The design notions mentioned above are taken into account for the following purposes.

- To improve rigidity,
- To obtain dynamic stability,
- To decrease the shear effects due to the lateral forces on the joints,
- To increase the payload capacity,
- To have more powerful actuators towards the base.

Due to binary actuation, the end effector of the HRM can reach a finite number of positions, i.e. the manipulator can not implement the given tasks (the desired positions and orientations) exactly. However, running at discrete states, the error between the desired pose and the real pose is most probably inevitable. To eliminate these errors, a continuously actuated manipulator is added as the last module of the HRM, which functions as a *fine tuning manipulator* (FTM) with 6 DOF as depicted in Figure 2.6. It is used not only for fine tuning but also for tracking the continuous trajectory between the discrete configurations.

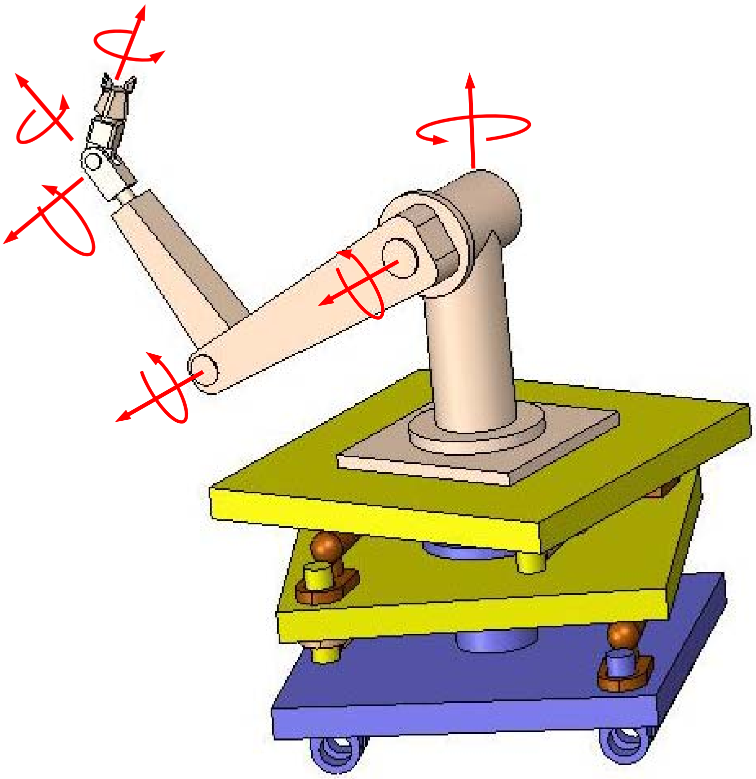


Figure 2.6 The Fine Tuning Manipulator (FTM) Used in the HRM

In this chapter, an introduction has been given about the hyper redundant manipulator. This involves the modeling, structure, actuation type and modifications of the HRM.

CHAPTER 3

FORWARD KINEMATICS OF THE HYPER REDUNDANT MANIPULATOR

3.1 Introduction

Forward kinematics is defined as a mapping from the joint space to the task space. In other words, forward kinematics finds the position vector of the tip point and the orientation matrix of the end effector of a manipulator in terms of the joint variables. Note that, here, the position vectors and transformation matrices of each module, the workspace for a certain number of possible configurations and also the inverse kinematics of the hyper redundant manipulator actuated binary are all based on the forward kinematics. All equations in the forward kinematics can be formulated in terms of the state of the actuators and dimensions.

Some terminology to be used in this thesis should be clarified. The configuration is used to express the stable positions of the HRM or its each module with respect to different fully expended or contracted positions of the actuators. For example, a sequence $[01011101]$ or $[01011................01100111]$ for the actuators represents a configuration of one module or the HRM respectively. State is used for actuators. For example, an actuator state is “0” if it is fully contracted with its minimum length, and it is “1”, if it is fully extended with its maximum length. Note that, the term state must be distinguished from the actual length of the actuator since the actuator can be at any length between its minimum and maximum length at any instant of time. Also, the configuration must be distinguished from the pose of the module or manipulator at any instant in time. The term pose is used to refer to the instantaneous position and orientation of the module or manipulator between two successive configurations. Because of the hyper redundancy, the different configurations belonging to the manipulator can reach the same pose.

For the binary HRM studied in this thesis, the calculations depend on the configuration sequences of the manipulator. These sequences are constructed from, in turn, “0”s or “1”s of each actuator state. For the sake of simplicity, they are made as module-based since all modules have the same characteristic structure, except the dimensions of each module.

$$\textit{An Example of Configuration Sequence} = [\underbrace{01100100}_{\text{module}1}\,\underbrace{11100001}_{\text{module}2}.............\underbrace{01000100}_{\text{module}10}]$$

To begin the forward kinematics, the kinematics equations for the three submodules are obtained and combined for each module. Then, the position vectors and transformation matrices from the module kinematics are evaluated for the HRM as in a serial robot manipulator. The kinematics equations for each submodule and module are derived in the following part.

3.2 The Kinematics of the Submodules

The first submodule is the VGT structure consisting of two platforms and three on-off pneumatic actuators between them. It has three degrees of freedom: two for positioning the center of the upper platform and one for the rotation of the upper platform. The first and second VGT submodules can be represented with Figure 3.1. This figure shows all the joint variables and the required parameters for the kinematics of the first submodule. Here, a Cartesian reference frame, $\mathscr{F}_0$, is located in the center of the base platform with the z_0 axis normal to the base. The moving reference frame, $\mathscr{F}_1$, is located in the center of the top platform. $\alpha_1, \beta_1, \gamma_1$ denote the passive joint angles between the related actuators and $+y_0$ axis of the base. The angle measured from the unit vector $\vec{u}_2^{(0)}$ to $\vec{u}_2^{(1)}$ is given by θ_{01}. This angle denotes the rotation of the upper platform about $\vec{u}_1^{(0)} = \vec{u}_1^{(1)}$. The position vector of $\mathcal{F}_1$ relative to $\mathcal{F}_0$ is given as $\vec{r}_{01}$. d_0, d_1 are constant design parameters. s_{11}, s_{12}, s_{13} are the joint variables which can be only minimum and maximum joint length. The values of these parameters change from module to module.

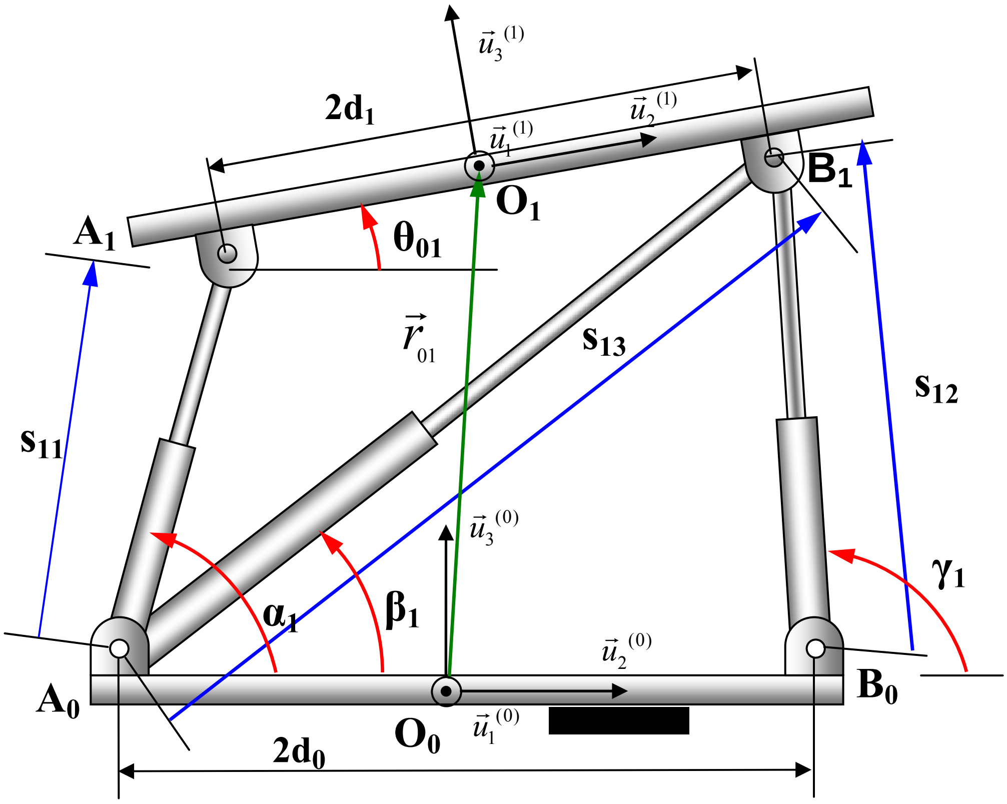


Figure 3.1 The VGT Submodule

By the help of the loop closure equations for the planar VGT, the solution of all the related variables is found in closed form in terms of the actuated joint variables s_{11}, s_{12}, s_{13} and the dimensions d_0, d_1. Using the above figure, the two independent loop closure equations (LCE) can be written as follows.

$$\overrightarrow{A_0B_0}+\overrightarrow{B_0B_1}=\overrightarrow{A_0B_1} \quad \text{(LCE 1)}$$

$$\overrightarrow{A_0A_1}+\overrightarrow{A_1B_1}=\overrightarrow{A_0B_1} \quad \text{(LCE 2)}$$

The scalar equations can be written from these loop closure equations as follows.

$$2d_0+s_{12}\cos(\gamma_1)=s_{13}\cos(\beta_1) \quad (3.1)$$

$$s_{12}\sin(\gamma_1)=s_{13}\sin(\beta_1) \quad (3.2)$$

$$s_{11}\cos(\alpha_1)+2d_1\cos(\theta_{01})=s_{13}\cos(\beta_1) \quad (3.3)$$

$$s_{11}\sin(\alpha_1)+2d_1\sin(\theta_{01})=s_{13}\sin(\beta_1) \quad (3.4)$$

In these equations, s_{11}, s_{12} and s_{13} are known parameters, whereas $\alpha_1, \beta_1, \gamma_1$ and θ_{01} are unknowns.

Taking the square of Equations (3.1) and (3.2), adding them side by side, the angle β_1 is eliminated and the angle γ_1 is obtained analytically.

$$\gamma_1=\text{atan2}\left(\sigma_1\sqrt{1-\xi_1^{\,2}},\xi_1\right) \quad (3.5)$$

Where $\cos(\gamma_1)=\dfrac{s_{13}^{\ 2}-s_{12}^{\ 2}-4d_0^{\ 2}}{4\,d_0\,s_{12}}=\xi_1$ and $\sin\left(\gamma_1\right)=\sigma_1\sqrt{1-\xi_1^{\,2}}$, $\sigma_1=\pm 1$. The configurations for $\sigma_1=+1$ and $\sigma_1=-1$ are shown in Figure 3.2.

From Equation (3.1), $\cos(\beta_1)=\dfrac{2d_0+s_{12}\cos(\gamma_1)}{s_{13}}=\lambda_1$ and from Equation (3.2), $\sin\left(\beta_1\right)=\dfrac{s_{12}\sin\left(\gamma_1\right)}{s_{13}}=\delta_1$.

$$\beta_1 = \operatorname{atan2}\left(\delta_1, \lambda_1\right) \tag{3.6}$$

After that, rearranging Equations (3.3) and (3.4), and to eliminate the angle α_1, take square of these arranged equations and add them side by side. The angle θ_{01} can be found analytically.

$$\theta_{01} = \beta_1 - \operatorname{atan2}\left(\sigma_2\sqrt{1-\kappa_1^{\,2}}, \kappa_1\right) \tag{3.7}$$

Where $\cos(\beta_1)\cos(\theta_{01}) + \sin(\beta_1)\sin(\theta_{01}) = \cos(\theta_{01} - \beta_1) = \dfrac{s_{13}^{\,2} + 4d_1^{\,2} - s_{11}^{\,2}}{4d_1 s_{13}} = \kappa_1$ and $\sin(\theta_{01} - \beta_1) = \sigma_2\sqrt{1-\kappa_1^{\,2}}$ $\sigma_2 = \pm 1$ (The configurations for $\sigma_2 = +1$ and $\sigma_2 = -1$ are shown in Figure 3.2)

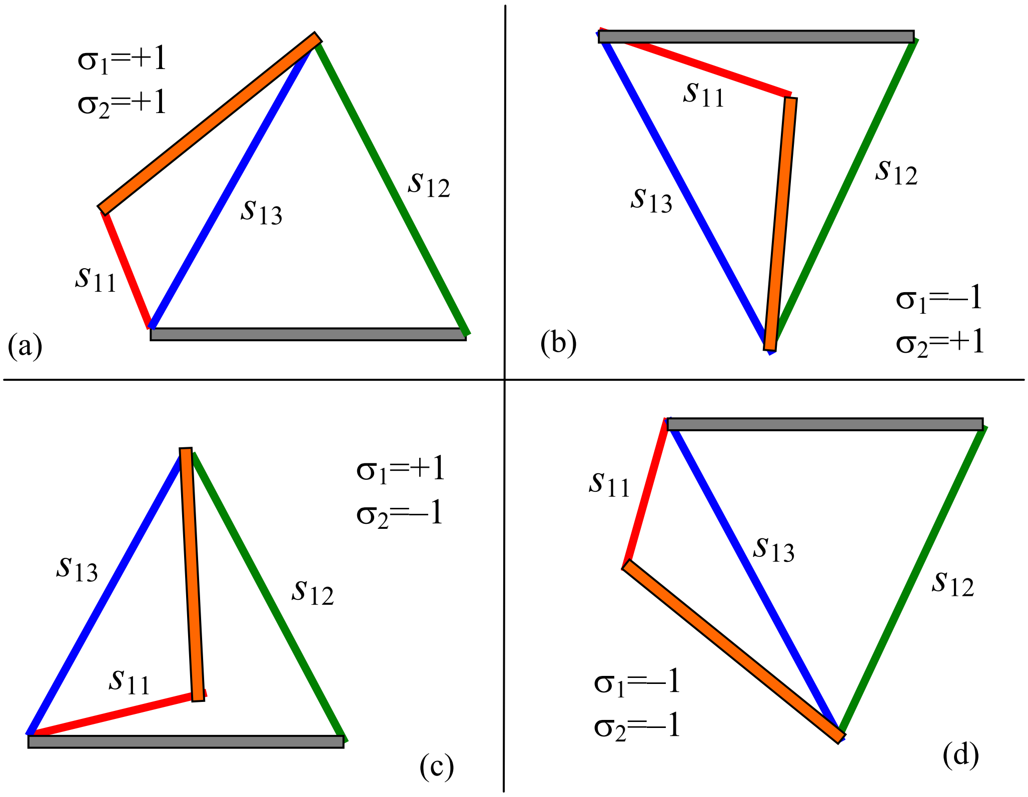


Figure 3.2 VGT-Mechanism Solutions

From Equation (3.3), $\cos(\alpha_1) = \frac{s_{13}\cos(\beta_1) - 2d_1\cos(\theta_{01})}{s_{11}} = n_1$ and from Equation (3.4), $\sin(\alpha_1) = \frac{s_{13}\sin(\beta_1) - 2d_1\sin(\theta_{01})}{s_{11}} = m_1$, the angle α_1 can be found uniquely as

$$\alpha_1 = \text{atan2}(m_1, n_1) \tag{3.8}$$

Note that, since the manipulator falls into positional singularity, the terms s_{11}, s_{12} and s_{13} should not be zero. In such a case the terms ξ_1, λ_1, δ_1 and κ_1 become indefinite. These conditions are already supplied because of the manipulator design because s_{11}, s_{12} and s_{13} are actuator lengths and between minimum and maximum lengths. That is, $s_{11} \neq 0$, $s_{12} \neq 0$ and $s_{13} \neq 0$.

Due to the inverse trigonometric terms in Equations (3.5) and (3.7), there are multiple solutions for the forward kinematics. Figure 3.2 shows the four possible configurations of a VGT submodule in a binary representation of $\left[s_{11}s_{12}s_{13}\right] = \left[010\right]$. The solutions in figures (a) and (d) can be referred to as open whereas, the other solutions, in which the extension of s_{11} crosses s_{12}, can be classified as closed. The only solution valid for this study is the solution "a" for $\sigma_1 = +1$ and $\sigma_2 = +1$.

After the values of the unknown joint variables are calculated, the position vector $\vec{r}_{01}$ can be written with respect to the base reference frame $\mathcal{F}_0$ as.

$$\vec{r}_{01} = \overset{\rightarrow}{O_0 A_0} + \overset{\rightarrow}{A_0 A_1} + \overset{\rightarrow}{A_1 O_1}$$

Since the first submodule moves in the $\vec{u}_2^{(0)} - \vec{u}_3^{(0)}$ plane, its first component becomes zero. Then $\vec{r}_{01}$ can be written in matrix form relative to $\mathcal{F}_0$ as.

$$\bar{r}_{01} = \begin{bmatrix} 0 \\ s_{11}\cos(\alpha_1) + d_1\cos(\theta_{01}) - d_0 \\ s_{11}\sin(\alpha_1) + d_1\sin(\theta_{01}) \end{bmatrix} \tag{3.9}$$

Applying the same procedure to the second VGT submodule, all variables related to this part can be found in terms of the actuated joint variables and dimensions. However, the only two differences are the subscript number and the motion plane. For example, the subscript "1" becomes "2" for the second submodule, and the motion plane becomes $\vec{u}_1^{(1)} - \vec{u}_3^{(1)}$ plane. For the second submodule, $\mathcal{F}_2$ reference frame is located in the center of top platform of the second VGT submodule. $\vec{r}_{12}$ indicates the position vector of the reference frame $\mathcal{F}_2$ relative to $\mathcal{F}_1$. After the replacement of $s_{11} \to s_{21}$, $s_{12} \to s_{22}$, $s_{13} \to s_{23}$, $\alpha_1 \to \alpha_2$, $\beta_1 \to \beta_2$, $\gamma_1 \to \gamma_2$ and $\theta_{01} \to \theta_{12}$, the unknown joint variables can be found, and $\vec{r}_{12}$ can be written in matrix form relative to $\mathcal{F}_1$.

$$\bar{r}_{12} = \begin{bmatrix} -s_{21}\cos(\alpha_2) - d_1\cos(\theta_{12}) + d_0 \\ 0 \\ s_{21}\sin(\alpha_2) + d_1\sin(\theta_{12}) \end{bmatrix} \tag{3.10}$$

Note that, there is no motion in the $\vec{u}_2^{(1)}$ direction, and the rotation of the top platform takes place about $\vec{u}_2^{(1)}$ axis.

Finally, the kinematics of the rotational submodule will be investigated in detail. As expressed before, this submodule makes rotational motion about only one axis relative to its reference frame. This part can be shown as a planar mechanism as shown in Figure 3.3.

Here, O_bK indicates the actuator length used in this submodule. O_b is a point on the top platform of the second VGT submodule to which an end point of the actuator is jointed to this point. O_t is the center point of the first platform for the rotating submodule. Taking the projection of these onto a plane, the equivalent mechanism can be obtained as shown in Figure 3.3. The rotation angle in terms of the joint variables and dimensions is determined according to this figure. This mechanism has one DOF and one independent loop closure equation. The LCE is as follows.

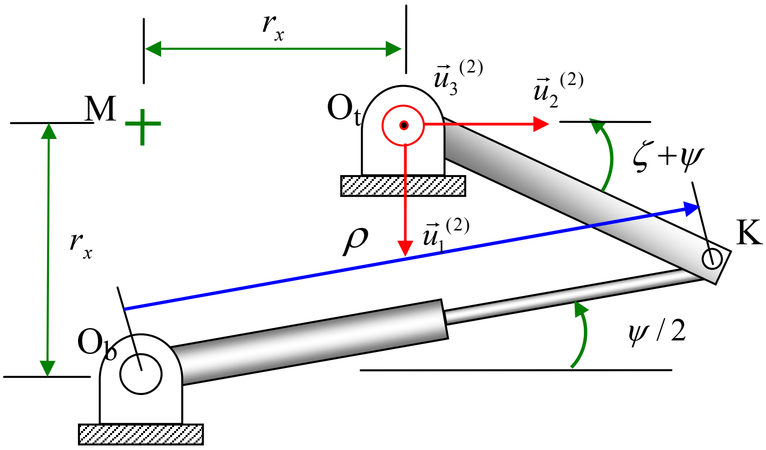


Figure 3.3 The Schematic View of the Rotational (Third) Submodule

$$\overrightarrow{O_bK} = \overrightarrow{O_bM} + \overrightarrow{MO_t} + \overrightarrow{O_tK}$$

Write the scalar equations from this LCE in the following.

$$\rho \cos(\frac{\psi}{2}) = r_x - \sqrt{2}\, r_x \cos(\zeta + \psi) \tag{3.11}$$

$$\rho \sin(\frac{\psi}{2}) = r_x + \sqrt{2}\, r_x \sin(\zeta + \psi) \tag{3.12}$$

Where ζ is a constant angle with $\zeta = \pi / 4$. Rearranging these equations, taking the square of them, and adding them side by side to eliminate $\zeta + \psi$, one obtains

$$\rho = 2r_x\left(\cos(\frac{\psi}{2}) + \sin(\frac{\psi}{2})\right) \tag{3.13}$$

The rotation angle of the third submodule can be found as follows.

$$\psi = \operatorname{atan2}\left(\chi, \sigma_3\sqrt{1-\chi^2}\right) \tag{3.14}$$

Where, $\sin(\psi) = \dfrac{\rho^2 - 4r_x^{\,2}}{4r_x^{\,2}} = \chi$ and $\cos(\psi) = \sigma_3\sqrt{1-\chi^2}$, $\sigma_3 = \pm 1$.

For $\sigma_3 = -1$, the other mechanism solution is the symmetry of the original mechanism with respect to O_bM .

Following these calculations, the module-based forward kinematics equations of the HRM can be written in a recursive manner up to its last platform, which is the base of the FTM. As for the kinematics of the FTM, it can be seen in Appendix A. The transformation matrix of the last platform and the position vector of its center point relative to the fixed base frame, $\mathcal{F}_0$, are denoted simply as $\vec{r}$ and $\hat{C}$. They are expressed as described below.

$$\vec{r} = \vec{r}_1 + \vec{r}_2 + \ldots\ldots\ldots\ldots + \vec{r}_n \tag{3.15}$$

$$\hat{C}^{(0,n)} = \hat{C}^{(0,1)}\hat{C}^{(1,2)}\ldots\ldots\ldots\ldots\ldots\hat{C}^{(n-2,n-1)}\hat{C}^{(n-1,n)} \tag{3.16}$$

Here, *n* is the number of discretely actuated modules, $\vec{r}_i$ is the position vector of center of the upper platform of the i^{th} module relative to the base frame and $\hat{C}^{(i-1,i)}$ is the transformation matrix from the reference frame of the i^{th} module to that of the $(i-1)^{th}$ module. The detailed expressions are given below for $\vec{r}_i$ and $\hat{C}^{(i-1,i)}$.

$$\vec{r}_i = \vec{r}_{01i} + \vec{r}_{12i} + h_i\vec{u}_3$$

In the matrix form,

$$\overline{r}_i^{(0)} = \hat{C}^{(0,i-1)}\left(\overline{r}_{1i}^{(i-1)} + \hat{C}^{(i-1,sub_{1i})}\,\overline{r}_{12i}^{(sub_{1i})} + h_i\hat{C}^{(i-1,sub_{1i})}\hat{C}^{(sub_{1i},sub_{2i})}\,\overline{u}_3^{(sub_{1i})}\right) \tag{3.17}$$

$\hat{C}^{(i-1,sub_{1i})}$: For the i^{th} module, the transformation matrix from the top frame of the first submodule to the reference frame located at the base of the module.

$\hat{C}^{(sub_{1i},sub_{2i})}$: The transformation matrix from the top frame of the second submodule to the reference frame located at the top of the first submodule.

$$\hat{C}^{(i-1,i)} = \hat{C}^{(i-1,sub_{1i})}\,\hat{C}^{(sub_{1i},sub_{21i})}\hat{C}^{(sub_{2i},rot_i)} \tag{3.18}$$

$\hat{C}^{(sub_{2i},rot_i)}$: Due to the rotation of the third submodule, the transformation matrix from the reference frame located at top of the i^{th} module to the reference frame located at the top of the second submodule.

The transformation matrices can be written as exponential form. More detailed information about this subject can be found in [47].

$$\hat{C}^{(i-1,i)} = e^{\tilde{u}_1\theta_{1i}}\,e^{\tilde{u}_2\theta_{2i}}e^{\tilde{u}_3(\psi_{1i}-\psi_{2i})}.$$

The calculation of the forward kinematics equations is very easy on the computer since all formulas are recursive. By using MATLAB computer program, the position vectors of each module for $2^3 \times 2^3 \times 3^1 = 192$ possible configurations are calculated with respect to the dimensions given in Table 3.1 and stored in a computer library. In Table 3.1, the dimensions d_0, d_1 and the maximum and minimum length of the joint actuators $\left(s_{ij}\right)_{min,max}$ and $\left(\rho\right)_{min,max}$ for each module are given according to Figure 3.1 and Figure 3.3. Given a configuration sequence, the position vectors and transformation matrices for all modules are calculated by using this library and they are combined with Equations (3.17) and (3.18) to obtain the position and orientation of the tip point of the HRM. For the different configuration sequences, the different manipulator configurations are shown in Figures 3.4a and 3.4b.

Table 3.1 The Dimensions of Each Module (in meters)

Module Number	d_0	d_1	$(s_{i1})_{min}$ $(s_{i2})_{min}$	$(s_{i1})_{max}$ $(s_{i2})_{max}$	$(s_{i3})_{min}$	$(s_{i3})_{max}$	$(\rho)_{min}$	$(\rho)_{max}$
1	0.142	0.1278	0.21	0.342	0.3416	0.4354	0.217	0.277
2	0.13	0.117	0.2	0.318	0.3176	0.4024	0.192	0.245
3	0.12	0.108	0.185	0.291	0.2934	0.3695	0.177	0.226
4	0.1105	0.9945	0.148	0.244	0.2566	0.3217	0.159	0.203
5	0.0909	0.08145	0.125	0.213	0.2124	0.2736	0.156	0.199
6	0.08	0.072	0.112	0.190	0.1886	0.2432	0.130	0.169
7	0.07	0.063	0.105	0.175	0.1693	0.2197	0.112	0.143
8	0.058	0.0522	0.088	0.148	0.1409	0.1844	0.098	0.125
9	0.044	0.0396	0.084	0.132	0.1184	0.1562	0.076	0.097
10	0.036	0.0324	0.06	0.102	0.0909	0.1228	0.062	0.079

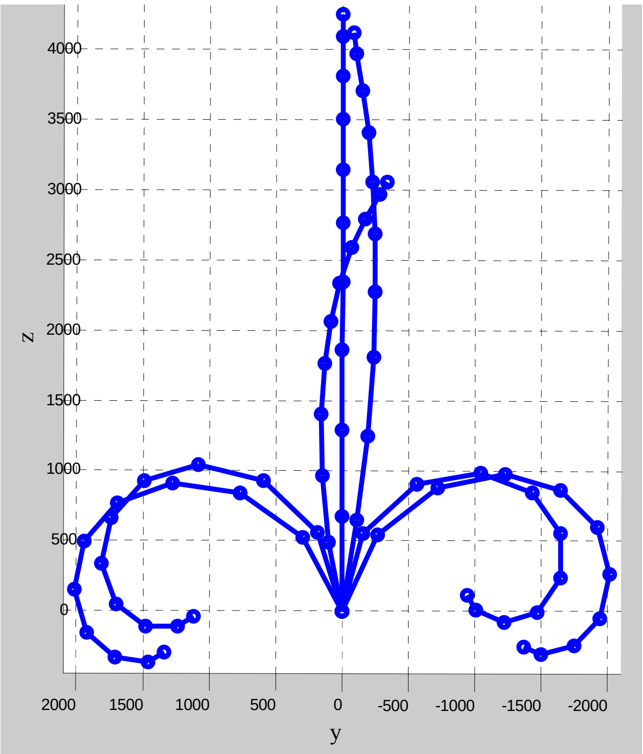


Figure 3.4a The Different Manipulator Configurations with Planar Motion

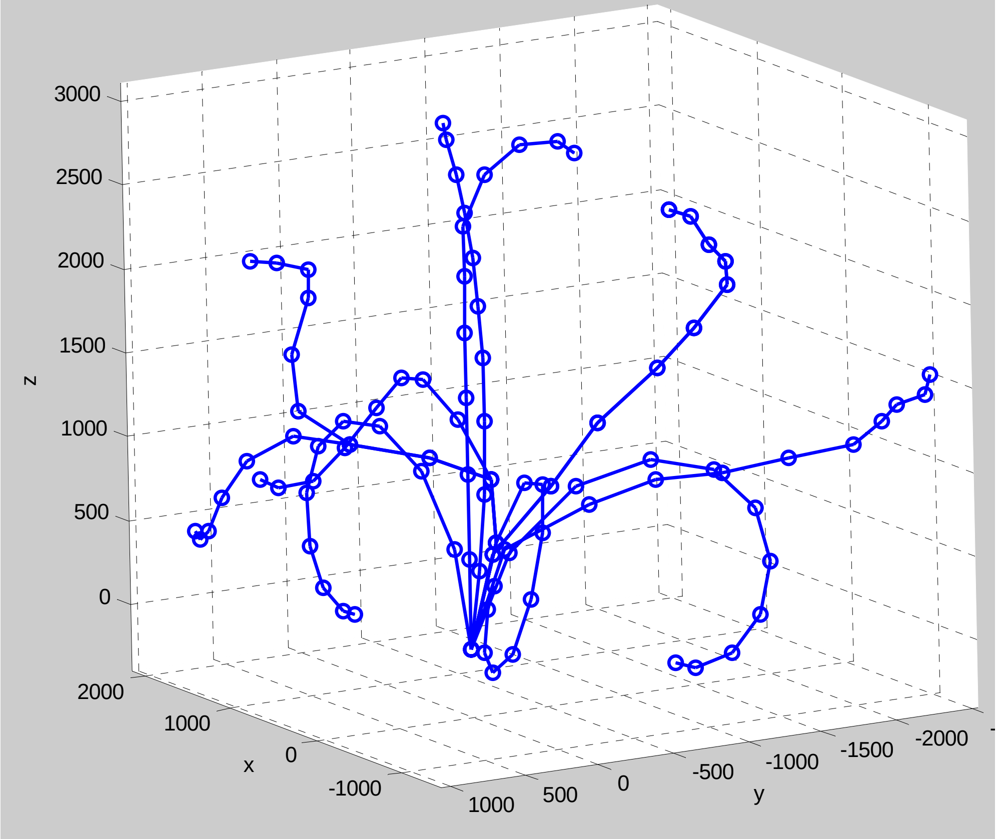


Figure 3.4b The Different Manipulator Configurations with Spatial Motion

3.3 The Singularity Analysis for the Submodules

Since the manipulator has so many links and it contains the symmetric constructions, the possibility of obtaining the singular positions is very high. It is necessary to solve the passive joint variables of the submodules for both the forward and inverse kinematics of the HRM. Therefore, the position and velocity singularity for the submodules will be investigated in this section.

First of all, let's solve analytically the passive joint variables from the inverse kinematics for the given actuated joint variables (s_{11}, s_{12} and s_{13}) of the submodule. Also, the position of the center of the top platform and the orientation of the top platform should be solved. The variables are shown in Figure 3.1.

According to Figure 3.1 and the first and second loop closure equations, the passive joint variables have been solved in Subsection (3.2). For this aim, Equations (3.5)-(3.8) can be written again as follows.

$$\gamma_1 = \operatorname{atan2}\left(\sigma_1\sqrt{1-\xi_1^2}, \xi_1\right)$$

where $\cos(\gamma_1) = \dfrac{s_{13}^2 - s_{12}^2 - 4d_0^2}{4\,d_0\,s_{12}} = \xi_1$ and $\sin\left(\gamma_1\right) = \sigma_1\sqrt{1-\xi_1^2}$.

$$\beta_1 = \operatorname{atan2}\left(\delta_1, \lambda_1\right)$$

where $\cos(\beta_1) = \dfrac{2d_0 + s_{12}\cos(\gamma_1)}{s_{13}} = \lambda_1$ and $\sin\left(\beta_1\right) = \dfrac{s_{12}\sin\left(\gamma_1\right)}{s_{13}} = \delta_1$.

$$\theta_{01} = \beta_1 - \operatorname{atan2}\left(\sigma_2\sqrt{1-\kappa_1^2}, \kappa_1\right)$$

where $\cos(\beta_1)\cos(\theta_{01}) + \sin(\beta_1)\sin(\theta_{01}) = \cos(\theta_{01} - \beta_1) = \dfrac{s_{13}^2 + 4d_1^2 - s_{11}^2}{4d_1 s_{13}} = \kappa_1$ and $\sin(\theta_{01} - \beta_1) = \sigma_2\sqrt{1-\kappa_1^2}$.

$$\alpha_1 = \operatorname{atan2}(m_1, n_1)$$

where

$$\cos(\alpha_1) = \frac{s_{13}\cos(\beta_1) - 2d_1\cos(\theta_{01})}{s_{11}} = n_1 \text{ and } \sin(\alpha_1) = \frac{s_{13}\sin(\beta_1) - 2d_1\sin(\theta_{01})}{s_{11}} = m_1 .$$

Note that, there are singularities in cases of $s_{11} = 0$, $s_{12} = 0$ and $s_{13} = 0$ but these singularities do not occur because these conditions are kinematically impossible for this submodule.

Using Figure 3.1, the position of the origin O_1 and the orientation of the top platform can be written as follows.

$$\bar{r}_{01} = \begin{bmatrix} 0 \\ s_{11}\cos(\alpha_1) + d_1\cos(\theta_{01}) - d_0 \\ s_{11}\sin(\alpha_1) + d_1\sin(\theta_{01}) \end{bmatrix} \text{ and } \hat{C}^{(0,1)} = e^{\tilde{u}_1\theta_{01}}.$$

Now, lets take a look at the velocity singularity for the given velocity of the actuated joint variables ($\dot{s}_{11}$, $\dot{s}_{12}$ and $\dot{s}_{13}$). For this aim, differentiating the scalar loop closure actuations and after some calculations, the angular velocities of the passive joint variables can be obtained in terms of the velocities of the actuated joint variables as follows.

$$\dot{\gamma}_1 = \frac{\dot{s}_{12}\cos(\gamma_1 - \beta_1) - \dot{s}_{13}}{s_{12}\sin(\gamma_1 - \beta_1)}$$

$$\dot{\beta}_1 = \frac{\dot{s}_{12} - \dot{s}_{13}\cos(\gamma_1 - \beta_1)}{s_{13}\sin(\gamma_1 - \beta_1)}$$

$$\dot{\alpha}_1 = \frac{1}{s_{11}\sin(\gamma_1 - \beta_1)\sin(\alpha_1 - \theta_{01})} \begin{pmatrix} \dot{s}_{13}\sin(\gamma_1 - \theta_{01}) + \dot{s}_{11}\cos(\alpha_1 - \theta_{01})\sin(\gamma_1 - \beta_1)\ldots \\ -\dot{s}_{12}\sin(\beta_1 - \theta_{01}) \end{pmatrix}$$

$$\dot{\theta}_{01} = \frac{1}{2d_1\sin(\gamma_1 - \beta_1)\sin(\alpha_1 - \theta_{01})}\left(\dot{s}_{13}\sin(\alpha_1 - \gamma_1) + \dot{s}_{11}\sin(\gamma_1 - \beta_1) - \dot{s}_{12}\sin(\alpha_1 - \beta_1)\right)$$

In these equations, the values making the denominators zero cause the velocity singularities at the following points.

- $\sin(\alpha_1 - \theta_{01}) = 0 \quad \Rightarrow \quad \alpha_1 = \theta_{01}$ or $\alpha_1 = \theta_{01} + \mathrm{n}\pi \quad \mathrm{n} = 1, 2....$
- $s_{11} = 0$
- $s_{12} = 0$
- $s_{13} = 0$
- $\sin(\gamma_1 - \beta_1) = 0 \quad \Rightarrow \quad \gamma_1 = \beta_1$ or $\gamma_1 = \beta_1 + \mathrm{n}\pi \quad \mathrm{n} = 1, 2...$

It is seen that none of above conditions are reached when Figure 3.1 is investigated.

These analyses show that neither the position singularity nor the velocity singularity at the first and second submodule occur for any given actuated joint variables.

For the third submodule which makes the only rotational motion, the passive joint variable can be obtained in terms of the actuated joint variable. In order to investigate the singularity of the third submodule, from Figure 3.3, Equations (3.13) and (3.14) can be rewritten as follows.

$$\psi = \operatorname{atan2}\left(\chi, \sigma_3\sqrt{1-\chi^2}\right)$$

where, $\sin(\psi) = \dfrac{\rho^2 - 4r_x^{\;2}}{4r_x^{\;2}} = \chi$ and $\cos(\psi) = \sigma_3\sqrt{1-\chi^2}$.

After differentiating Equation (3.13), the angular velocity of the passive joint can be obtained like this.

$$\dot{\psi} = \frac{\dot{\rho}}{r_x\left(\cos\dfrac{\psi}{2} - \sin\dfrac{\psi}{2}\right)}$$

For the third module, there is no positional singularity, but the condition of $\cos\dfrac{\psi}{2} - \sin\dfrac{\psi}{2} = 0$, i.e. $\psi = 90^0$ causes the velocity singularity. However, because of the construction of the third submodule, this value for ψ can not be reached. So, these analyses show that there are not any singular positions for the third submodule as well. In other words, the manipulator does not reach the singular positions with respect to the inverse kinematics analysis of the submodules. However, when the dynamic equations are solved, it is observed that the determinant of the system matrix $\left[\hat{A} \vdots \hat{B}\right]$ constructed from the equations of motion of the HRM is zero or closed to zero. This causes the system equations to be unsolvable or the results may not be correct. That is, apart from the kinematics singularities, there is another singularity that comes from the dynamic equations.

Briefly, this chapter presents the forward kinematics of the HRM to calculate the position vectors and transformation matrices in terms of the actuated joint variables. Then, the positional and velocity singularities are studied.

CHAPTER 4

INVERSE KINEMATICS OF THE HYPER REDUNDANT MANIPULATOR

4.1 Introduction

Inverse kinematics is a required study for the motion planning of manipulators. Given a position and orientation (simply expressed as pose) of the end effector, the required values belonging to the actuated joint variables of the manipulator are found by means of inverse kinematics. The trajectory planning in the configuration space is based on the result of the inverse kinematics study. It is always desired that manipulator can track the given reference curve. At the same time, the result of the motion planning determines also the shape of the manipulator for the given pose. Implementing all these studies successfully depends on the performance of the inverse kinematics methods to be used. Depending upon their degrees of freedom, there are three types of manipulators namely, normal (regular), deficient and redundant. These manipulators have 6 DOF, less than 6 DOF and more than 6 DOF respectively. If a manipulator has 6 or less than 6 DOF, then, its inverse kinematics solution can be found out analytically or numerically. But, in redundant manipulators, for example with up to 10, 11, or 12 DOF, an optimal solution is searched in the velocity level by using some methods like Pseudo-Inverse matrix method. However, these methods are not feasible for hyper redundant manipulators since there are many excessively variables to be solved and the dimension of matrix calculations becomes very large.

Inverse kinematics is a mapping from the task space to the joint configuration space for the hyper redundant manipulators. It is much more difficult for the binary hyper redundant manipulators compared to their continuously actuated counterparts because the actuators used in the binary manipulators do not have any intermediate state. On account of this, any analytical or numerical solution for the inverse kinematics is not considered. For the binary HRMs, solutions with minimum positional and orientational errors are expected from the inverse kinematics method. Besides, the shape of the manipulator and trajectory following are the other criteria to be implemented. All calculations should, of course, take short time. The methods presented in the literature try to reach the optimal solution. However, it can't be stated that the solution is the global optimum because of a great number of configurations. Nevertheless, the solution can approach a local optimum as closely as sufficient.

As mentioned in Chapter 1, a few methods have been presented in the literature for the inverse kinematics of binary hyper redundant manipulators. For the binary manipulator with 12-14 actuated degrees of freedom, the exhaustive search (brute force search) method is an applicable and effective method. In the exhaustive search method, all configurations are looked at to obtain the optimal solution. However, the number of configuration increases exponentially by increasing the DOF of the system, i.e. the number of actuators. Therefore, applying this method to the binary manipulators with very high DOF is not an efficient approach for the inverse kinematics. For example, the manipulator with 50 binary actuators has $2^{50} \approx 1.125 \times 10^{12}$ possible configurations. Searching all of them for optimal solution is impossible. Therefore, the binary manipulator with too many actuators needs more efficient methods.

4.2 The Inverse Kinematics of the HRM

In this study, three methods are presented to solve the inverse kinematics of the HRM. These methods try to select the convenient configuration from $\left(2^3 \times 2^3 \times 3^1\right)^{10} \approx 6.8 \times 10^{22}$ possible configurations for the manipulator with ten modules. In the result of the solution, the bit sequences constructed by 0's and 1's will be found. These bits represent the on-off pneumatic actuator lengths, "0" for fully contracted and "1" fully extended positions. Note that here, in all the kinematics studies, the hyper redundant manipulator covers only the discretely actuated part of the manipulator, and the continuously actuated part (the fine tuning manipulator (FTM)) of the manipulator is not included into the calculations. For the discretely actuated part of the HRM, the position and orientation of the target are taken as the reference position and orientation for the last platform of the HRM (i.e. the base of the FTM). However, the part of the HRM required to reach the given target is actually the end effector of the FTM. Therefore, there should be a certain distance allowed between the center point of the last platform and the target point (i.e. the point describing the target position) so that the FTM can function properly by having the target point within its workspace. In other words, this distance should have a lower and upper bound with respect to the workspace of the FTM. Here, the boundaries of the workspace of the FTM are approximated (for the sake of simplicity) as semi-spherical surfaces. This approximate workspace is a subset of the actual workspace of the FTM. It is depicted as shown in Figure 4.1.

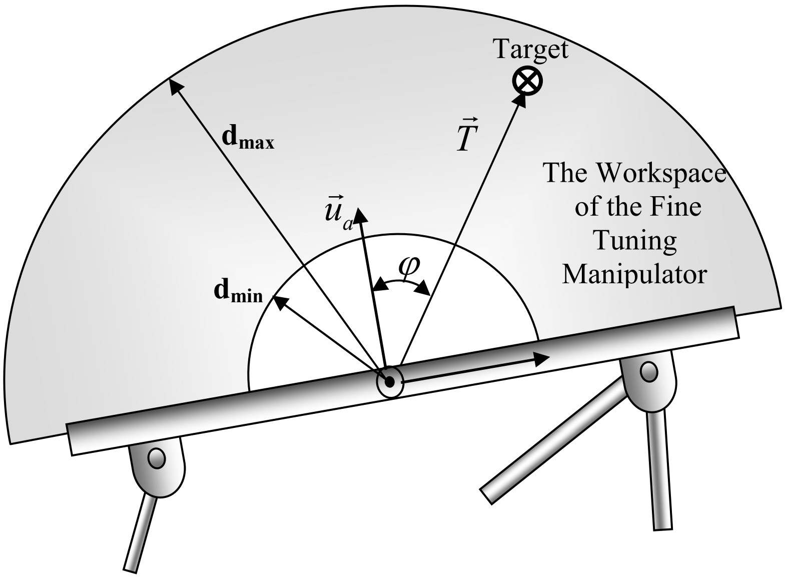

Target
$\mathbf{d}_{\mathbf{max}}$
$\vec{T}$
The Workspace of the Fine Tuning Manipulator
$\vec{u}_a$
φ
$\mathbf{d}_{\mathbf{min}}$

Figure 4.1 The Orientation Criterion for the Selection of the Configuration

The following criteria are taken into consideration for the selection of the convenient configurations.

Positional Error: The distance, d_{error}, between the tip point of the hyper redundant manipulator and the given task position should be in a certain threshold, i.e. $d_{\min} < d_{error} < d_{\max}$. The values of $d_{\min}$ and $d_{\max}$ are related to the workspace of the FTM for fine tuning as shown in Figure 4.1. $d_{\min}$ and $d_{\max}$ represent the lower and upper bounds of the hemi-sperical workspace respectively. Thus, this supplies the target point to be in the border of the compensation of the FTM.

Orientational Error: The orientational errors are also eliminated by the FTM. However, in the selection of the configuration, the top of the HRM (i.e. the base of the FTM) should be situated in such a way that the workspace of the FTM can involve the target point as shown in Figure 4.1 and it must be oriented as closely as desired in order to reduce the burden on the FTM.

Desired Manipulator Shape: In addition to satisfying the criteria of the positional and orientational errors concerning the last platform, the whole body of the HRM should also take a specified shape in 3D space in order to avoid the obstacles.

Trajectory Following: Between two successive configurations, binary manipulators can not follow a certain reference curve since there is not control over its actuators, i.e. the actuators are not located in intermediates states in a controllable way. They can not only exactly track a reference curves but also the deviations from the reference curve can be quite excessive. In this study, the exact trajectory tracking between the via points is not focused on, however, it is desired that the deviation is smaller, and the manipulator should not collide with the obstacles during the motion.

In this study, three inverse kinematics methods are presented in order to select the possibly better configurations to satisfy the above criteria in a reasonable way. These are *the spline fitting method*, *the extended spline fitting method* and *the workspace filling method*. The spline fitting method is based on the space reference curve and fitting each module to the curve parts segmented with respect to the average length of the related module. In the second method, the extended spline fitting method is the combination of the spline fitting method and a genetic algorithm. Finally, in the third method, the workspace filling method, the optimal configuration is selected from the workspace constructed randomly or by using a genetic algorithm. These methods are given in detail in the following subsections.

4.3 The Spline Fitting Method

In this method, the investigation of the inverse kinematics of the hyper redundant manipulator is based on the spline curve, which is a reference spatial curve. In order to construct a spline, some control points are needed in the workspace. These points are defined according to the task position that the end effector reaches and the shape of the manipulator to avoid the obstacles in the environment. The spline is formed by passing through or near these control points. After this construction, the aim is to fit the HRM to the spline as closely as possible.

The HRM contains many modules; each consists of two VGT submodules and one rotating submodule. The VGT submodule has 8 different configurations, as shown in Figure 4.2, because of three binary (on-off) actuators. In addition to this, the third module has a twist motion about only one axis. This part can be at only three angular positions $(-\psi^*, 0, +\psi^*)$. Hence, one module has totally $2^3 \times 2^3 \times 3^1 = 192$ possible configurations. The inverse kinematics is solved for the manipulator with ten modules. As mentioned earlier, the positions of the end point of each module are calculated for all 192 configurations by using the forward kinematics. The data, which is the positions and related bit sequences, is stored in a computer library.

To solve the inverse kinematics, matching between the spline and manipulator is needed. The spline defines the whole shape of the manipulator whereas we have only the information of each module and the target position. To match the spline and each module in a consistent way, the spline is divided into the number of the modules so that the length of each piece is proportional to the average length of corresponding module. This matching is implemented by minimizing a cost function given below.

$$
\begin{aligned}
f = \left|P-Q\right|^2 + L_1^2\left(\phi_M-\phi_s\right)^2 + L_2^2\left(\theta_M-\theta_s\right)^2 + L_3^2\left(\psi_M-\left(\psi_s+\kappa n\right)\right)^2 \\
+ L_4^2\left(\rho_M-\rho_s\right)^2 + L_5^2\left(\ell_M-\ell_S\right)^2
\end{aligned}
\tag{4.1}
$$

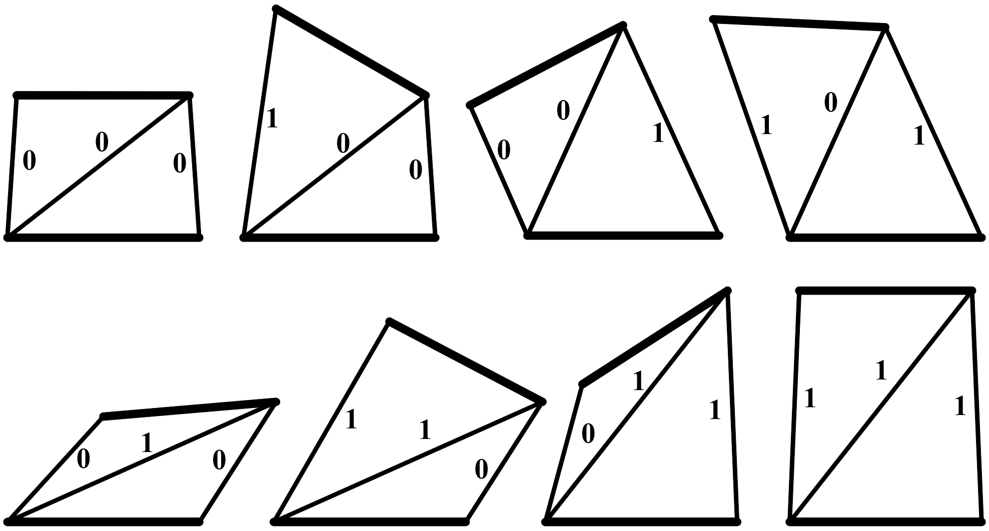


Figure 4.2 The 8 Different Configurations of the VGT Submodule

This cost function consists of the positional, orientational, radius of curvature and length errors. Here, Q and P denote the position of the end point of the spline segment and the n$^{\text{th}}$ module of the manipulator respectively. L_1, L_2, L_3, L_4 and L_5 are the gain constants of the azimuth angle (ϕ), the elevation angle (θ), the twist angle (ψ), the radius of curvature and the length respectively. The subscript "M" indicates the features of the HRM and "S" is used for that of the spline. The term $\psi_S + n\kappa$ represents the twist on the top of each spline. ψ_S comes from the spline calculation and κ is the twist value belonging to each spline part when the desired torsion on the top of the HRM is distributed over n spline parts. In this cost function, P, ϕ_M, θ_M, ψ_M, ρ_M, ℓ_M, L_1, L_2, L_3, L_4 and L_5 are the optimization parameters. The spline, of course, is very important tool for the inverse kinematics study. The detailed information about the spline is given in the following subsections.

In the spline fitting method, the HRM is fitted to the spline segments as closely as possible. The operation is carried out module by module and from the bottom to the top. The initial point of the first spline part and the first module are located in the center of the base platform. The cost function is calculated for 192 configurations by using the data stored in the computer library and calculating the values of ϕ_M, θ_M, ψ_M, ρ_M and ℓ_M for the related configurations. For the first module, the configuration minimizing the cost function is selected. The selected configuration involves the position of the end point of the module and its bit sequence like [00101100]. Using this information, the orientation of the first module is calculated and used in the selection of configuration for the second module. The same fitting procedure is applied for the second module. For this part, the configuration of the first module is fixed, and then the position vectors belonging to the 192 configurations of the second module are added to this fixed position vector. The cost function is calculated for the second module with respect to the frame at the top of fixed first module for the orientational and length errors. The position errors are already written with respect to the initial frame. Here, the orientation of the first module is taken into consideration. Again, according to the optimization of the cost function, the convenient configuration is chosen for the second module. This procedure continues until the last discrete module (10th module). After selecting all the configurations for 10 modules, the error between the given task pose and the real pose is compensated with the fine tuning manipulator (FTM). For this compensation, the calculation of the required joint variables for the FTM is given in Appendix A.

4.3.1 Splines

The term spline is named after the flexible strip used by ship builders to “interpolate” points of ship hulls. In general, the term "spline" is used to refer to a wide class of functions that are used in applications requiring data interpolation and/or smoothing [48]. Splines may be used for interpolation and/or smoothing of either one dimensional or multi-dimensional data. Spline functions for interpolation are normally determined as the minimizers of suitable measures of roughness (for example integral squared curvature) subject to the interpolation constraints.

Smoothing splines may be viewed as generalizations of interpolation splines where the functions are determined to minimize a weighted combination of the average squared approximation error over observed data and the roughness measure. For a number of meaningful definitions of the roughness measure, the spline functions are found to be finite dimensional in nature, which is the primary reason for their utility in computations and representation.

Splines are functions generally built of piecewise polynomials, which can to some extent overcome the drawbacks. These drawbacks are excessive oscillations that occur in case of many interpolation points, necessity of recomputation of the basis for additional knots and ill conditioned matrices in corresponding equations.

A piecewise function consisting of polynomial pieces of degree k is called a spline function provided that it is $(k-1)$ times continuously differentiable. As expressed before, the “pieces” may be described by different polynomials, resulting in, e.g., cubic splines, quintic splines, Hermite splines, but none of the above mathematical descriptions allow for geometric interpretation of the interpolating curve. More precisely the above descriptions do not include parameters which might indicate where in space the interpolating curve lies. That is why we considered interpolating curves with other, different basis, i.e. Bezier and NURBS curves. These curves contain intrinsic parameters which suit them to interactive design. More information about splines is given in [48], [49], and [50].

In this study, NURBS (Non-Uniform Rational Base Spline) is used as a reference curve since it contains parameters which suit the curve to interactive design, and is very efficient with its control polygon to construct the desired spatial curve with a broad range of geometric forms. Also, any local modification on the spline can be made easily by using the parameters of NURBS.

4.3.1.1 Non-Uniform Rational Base Spline (NURBS)

Non-Uniform Rational Base Spline (NURBS) curves are special cases of rational B (Base)-spline curves based on the B-spline basis functions $N_{i,k}$. Given a knot vector $T = (t_0, t_1, ..., t_{n+k})$, the associated normalized B-spline basis function $N_{i,k}$ of degree $(k-1)$ and of order k is defined to be

for $k = 1$ and $i = 0,1,....,n$

$$N_{i,1}(t) = \begin{cases} 1, & for \ \ t_i \le t \le t_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

for $k > 1$ and $i = 0,1,....,n$

$$N_{i,k}(t) = \frac{(t - t_i)}{(t_{i+k-1} - t_i)} N_{i,k-1}(t) + \frac{(t_{i+k} - t)}{(t_{i+k} - t_{i+1})} N_{i+1,k-1}(t) \tag{4.3}$$

The contribution of the basis function to the curve is shown in Figure 4.3.

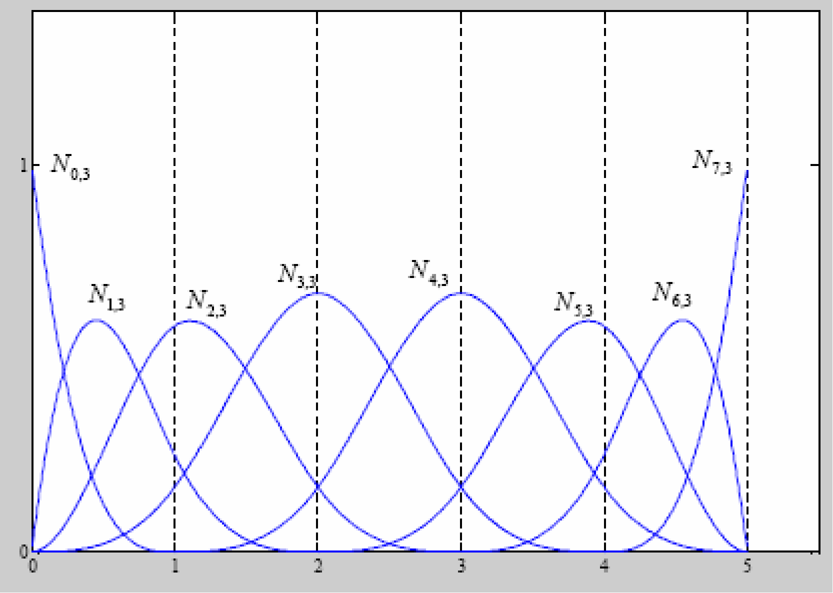


Figure 4.3 The Contribution of the Basis Functions

The knot vector, which specifies the distribution of the parametric variable along the curve, consists of a set of non-decreasing scalar values. A knot with certain value appears more than once in a knot vector. For example, $T = \left[\underbrace{0\ 0\ 0\ 0}_{k}\ t_{k+1},............,t_{n-k-1}\underbrace{1\ 1\ 1\ 1}_{k}\right]$ is used for the k^{th} order curve.

Suppose we are given the same knot vector *T* and points $B_i,\ i = 0,1,....,n$. Then a B-spline curve of degree (*k*-1) with knot vector can be defined as

$$B(t) = \sum_{i=0}^{n} B_i N_{i,k}(t), \quad n \geq k-1, t \in [t_{k-1}, t_{n+1}] \tag{4.4}$$

where the points B_i are called the control points or de Boor points and they form the control polygon or de Boor polygon as shown in Figure 4.4.

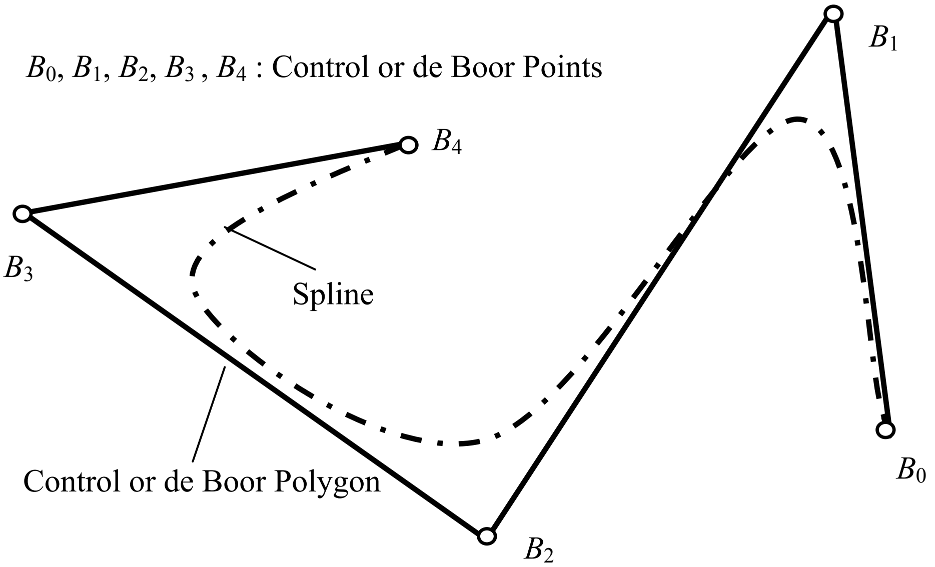

B_0, B_1, B_2, B_3, B_4 : Control or de Boor Points
B_1
B_4
B_3
Spline
B_0
Control or de Boor Polygon
B_2

Figure 4.4 The Control or de Boor Points and Polygon

The geometric interpretation of the de Boor polygon is the same as for Bezier polygon given in the Appendix B, but with the following additional features.

- B-spline functions are defined locally (This means that changing of one point in the de Boor points affects the corresponding B-spline curve only locally.),
- The number of de Boor points is not related to the degree of B-spline curve (One may insert additional de Boor points and keep the required curve degree).

A rational B-spline curve is described as the projection of a B-spline curve defined in four-dimensional homogenous space into three dimensional E^3 space. The fourth coordinate represents the weight associated with each control point. These weights are used to form a homogenous coordinate system. Thus, a point in 4D space can be represented by (wx, wy, wz, w). For a knot vector T and a set of de Boor points B_i, $i = 0,1,....,n$, given in homogenous coordinates, the rational B-spline curve form is described by

$$Q(t) = \frac{\sum_{i=0}^{n} B_i w_i N_{i,k}(t)}{\sum_{i=0}^{n} w_i N_{i,k}(t)}, \quad n \geq k-1, \; t \in [t_{k-1}, t_{n+1}] \tag{4.5}$$

Here, the parameter w_i is the homogenizing coordinate or the weight. Rational B-spline curves and basis functions are generalization of B-spline curves and, as such, they encompass all features of B-spline curves and B-spline basis functions as special cases. NURBS are rational B-spline curves with non-uniform knot vector, i.e. a knot vector whose values t_i are distributed evenly or non-evenly and have multiplicities. Due to their numerous parameters, NURBS curves seem to be a powerful tool for interpolants planning. These may be recognized as design parameters,

– Knot vector T, if we compare both equations above we see that the choice of the knot vector influences the basis functions $N_{i,k}$ and of course the NURBS curve.

– Curve degree. As expressed before, the change of the degree changes basis functions $N_{i,k}$ and thus change the form of the resulting NURBS curve.

– Number and multiplicity of de Boor points. We can change existing de Boor points, thus change the shape of the NURBS curve. This modifies the existing curve locally and we use this feature of NURBS to plan the interpolants to avoid boundary orientation poses.

– Weight values. Changing of the weights w_i influences the shape of NURBS curve, therefore they can be used as another design parameter. Figure 4.5 shows the effect of change of the weight values.

The detailed features of NURBS are given in the Appendix B.

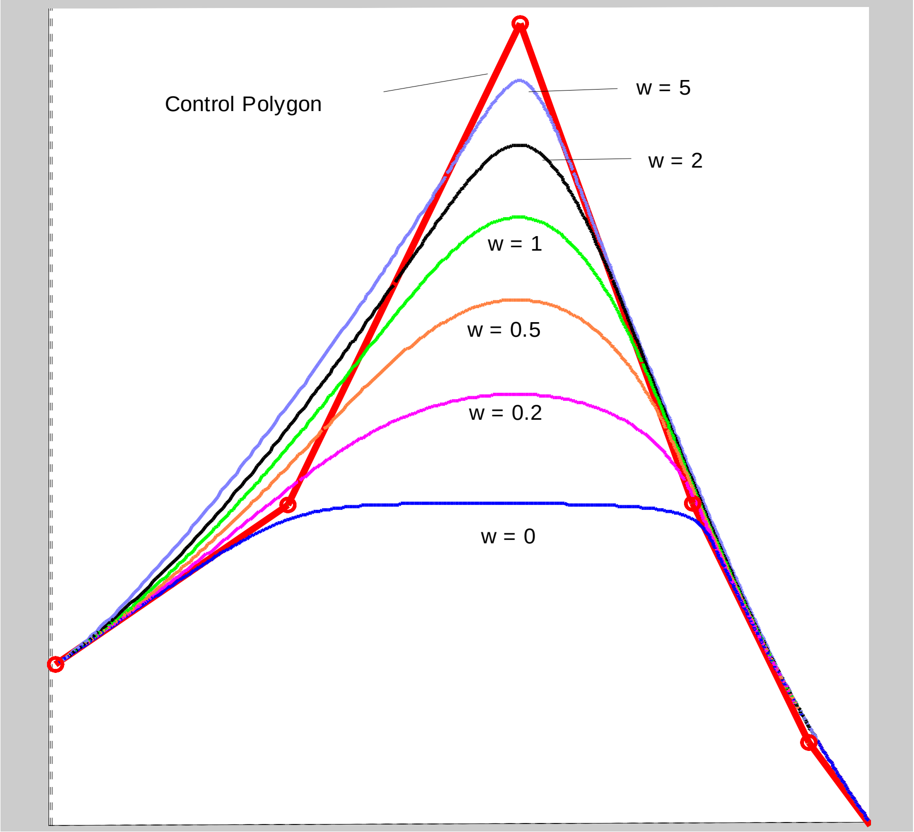


Figure 4.5 The Spline Part for Different Values of w

4.3.1.2 The Construction of Spline Parameters

For a given set of control points, a lot of splines can be drawn. In other words, the spline is not unique. Therefore, the spline is constructed by taking some various criteria into account for any set of control points so that it is subjected to a kind of shape modification. In this thesis, the control points defining the spline are selected by trial and error to supply the given task specifications. More appropriately, the positions of the control points can be defined according to some optimization criterion. However, such an optimization is held out of the thesis subject and it needs a further elaborate study. The shape of the spline should correspond to the motion capability of the hyper redundant manipulator, which is discretely actuated, since its inverse kinematics is investigated according to this spline. A spatial curve has some features such as arc length, radius of curvature, torsion, tangent and normal directions at any point on it. If these features are approximately met by the manipulator, i.e. by its modules, as a result of the inverse kinematics, the shape of the manipulator can be matched to the shape of the spline as closely as possible.

It is assumed that the dimensions of the HRM are assigned. The position and orientation of the end platform of each module or submodule are found from the forward analysis in terms of the system variables and the dimensions.

$$\overline{r} = \overline{r}(s_i, d_j) \tag{4.6}$$

$$\hat{C} = \hat{C}(s_i, d_j) \tag{4.7}$$

Here, s_i are the system variables defining the length of the actuators and d_j are the dimensions. According to these values, the end position, the length, the radius of curvature, the azimuth, elevation and twist angles of each module are obtained.

The external factors such as obstacles shown in Figure 4.6 should also be considered in the work space when constructing the splines. As for selecting the control points, the deviation of the manipulator from the spline should be considered. In this case, a sphere represents the obstacle in the workspace so that we can define it mathematically. Then, the algorithm can be setup for the manipulator to avoid the obstacles.

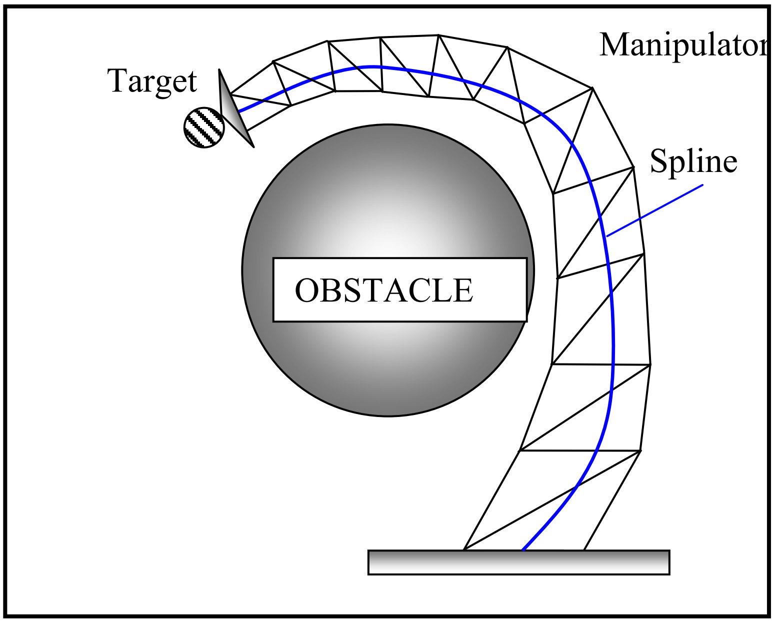


Figure 4.6 Obstacle Avoidance

The criteria of the spline optimization are as follows:

☐ For the curve $Q(t) = \left[x(t), y(t), z(t)\right]$ with the parameter *t*, the length of the spline should not be more than the maximum length of the manipulator or less than the minimum length of the manipulator. That is,

$$\ell_s > \ell_m^{\min} + \varepsilon$$

$$\ell_s < \ell_m^{\max} - \varepsilon$$

Where ε is the tolerance and ℓ_s is the length of spline defined as

$$\ell_s = \int_0^{t_{end}} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}\, dt \tag{4.8}$$

The approximated method for calculating ℓ_s in Equation (4.8) does not cause much error. Therefore, the following formula can be used

$$\ell_s = \sum_{i=1}^{n-1} \left(Q_{i+1}{}^2 - Q_i{}^2 \right)^{1/2} \tag{4.9}$$

Q_i are the points on the spline curve, and n is the number that the curve is evaluated.

☐ The control of the control polygon:

The control points construct a polygon, as shown in Figure 4.7, which is a reference to drive spline. If the angles between successive two lines in the control polygon are greater than any specified angle, this means that the spline supplies the requirements of the manipulator, otherwise, the number of iteration increases in selection of the spline, or, at least, it gives us any idea whether to change the position of the control points or not. The angle can be found as follows.

$$\theta_i = \cos^{-1}\left(\frac{< \overrightarrow{B_{i+1}B_i}, \overrightarrow{B_{i+2}B_{i+1}} >}{| \overrightarrow{B_{i+1}B_i} || \overrightarrow{B_{i+2}B_{i+1}} |} \right), \; i = 0.........n-2 \tag{4.10}$$

where B_i are the control points. Note that $\theta_i > \theta^*$ where θ^* is the specified angle.

☐ The radius of curvature of spline should be greater than that of the manipulator. Because of the discrete nature and physical restrictions, the manipulator does not allow all types of turnings with smaller curvatures. Therefore, sharp turnings of the manipulator are avoided. Since the success of the manipulator depends on the spline, the manipulator may keep track of the spline as closely as possible as far as the radius of curvature of the spline is greater than that of manipulator that is approximately calculated for one module.

$$\rho_s > \rho_m + \varepsilon \tag{4.11}$$

Where ε is the tolerance for the curvature.

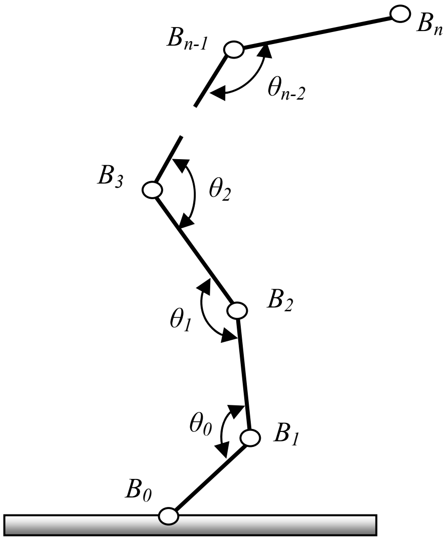


Figure 4.7 The Control Polygon

The geometric features of the module are given as follows. Here, P_1, P_2 and P_3 are three points on the manipulator placed on the base, intermediate and last platforms of the module as shown in Figure 4.8. These three points defines a plane in 3D space.

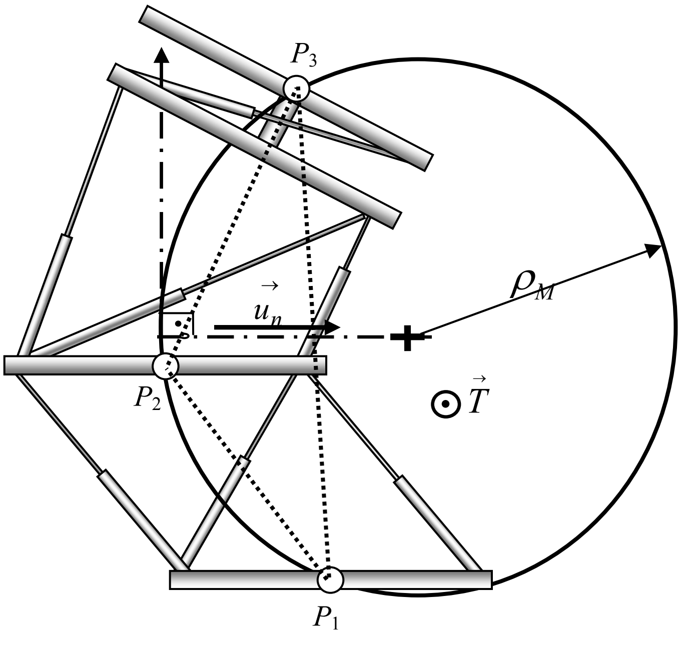


Figure 4.8 The Radius of Curveture for the Module

The normal vector $\vec{T}$ to the plane can be obtained by using the cross product of two vectors on the same plane,

$$\vec{T} = \overrightarrow{P_1P_3} \times \overrightarrow{P_1P_2}\,.$$

The vector directed on the normal of the circle is defined as $\vec{N} = \overrightarrow{P_1P_3} \times \vec{T}$. The unit normal vector $\vec{u}_n$ is also defined as $\vec{u}_n = \dfrac{\vec{N}}{|\vec{N}|}$. The magnitude of the radius of curvature ρ_M can be determined as

$$\rho_M = \frac{a\,b\,c}{4S} \tag{4.12}$$

where S is the area of the triangle and a, b, c are the lengths of sides of the triangle whose corner points are P_1, P_2, and P_3 . Note that, $S = \sqrt{u(u-a)(u-b)(u-c)}$, $u = \dfrac{a+b+c}{2}$.

The radius of curvature as shown in Figure 4.9 is given for the instantaneous value of t, which is the parameter of the spline. The equations below are used to carry out the magnitude of the radius of curvature.

Let $\vec{Q} = \vec{Q}\left(x(t), y(t), z(t)\right)$ be a spline in 3D space. The magnitude of the radius of curvature for the spline is then

$$\rho_S = \frac{\left|\dot{\vec{Q}} \times \ddot{\vec{Q}}\right|}{\left|\dot{\vec{Q}}\right|^3} \tag{4.13}$$

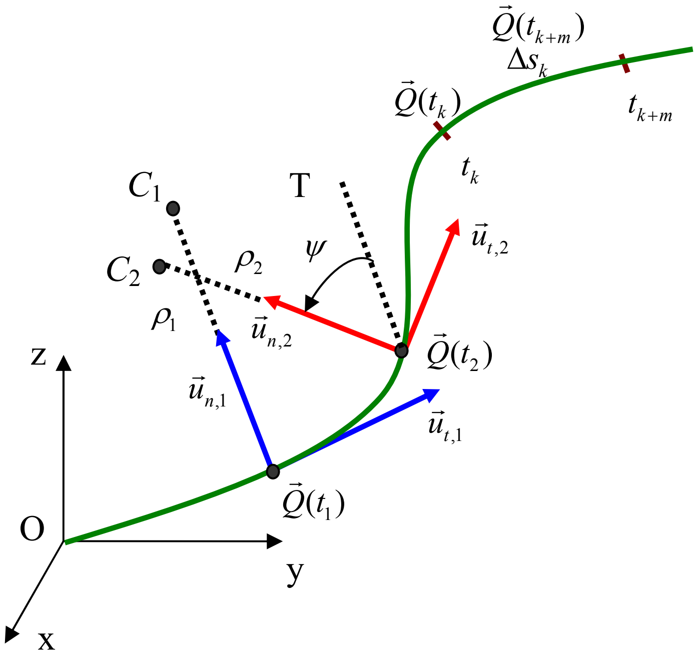


Figure 4.9. The Curvature and Torsion for the Spline

☐ The control of the azimuth and elevation and twist angles (Figure 4.10):

These angles show how much the module deviates from the reference axis for the spline fitting method. These angles are related to the motion capability of the manipulator. For each spline part, the azimuth and elevation angles should not exceed the specified values. That is, they should be in the ranges $-\theta^* < \theta < \theta^*$ and $-\phi^* < \phi < \phi^*$. According to the configurations of the module, θ^* and ϕ^* are determined by using the position vector from the base to the top of the module. The values of these angles can be calculated with some simple geometric calculations. For the values of the twist angle, the calculations are given as follows.

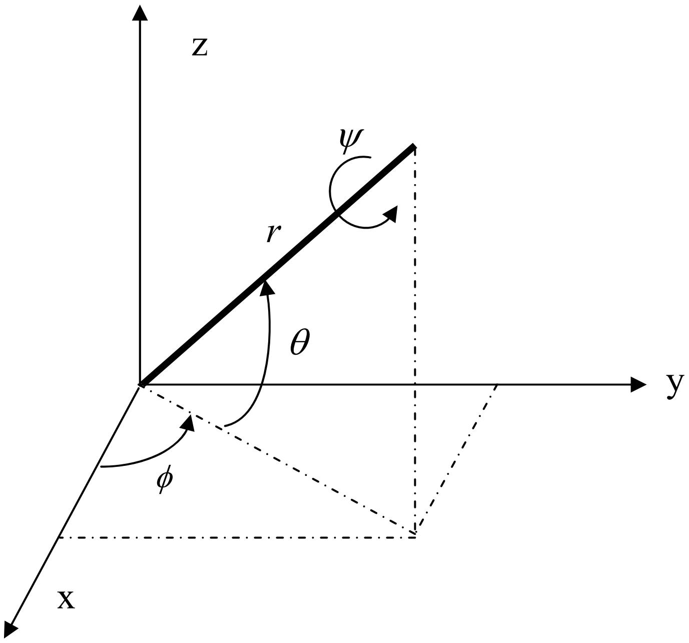


Figure 4.10 The Azimuth (ϕ), Elevation (θ) and Twist (ψ) Angles

The torsion: $\tau = \dfrac{\left(\dot{\vec{Q}} \times \ddot{\vec{Q}}\right) \bullet \dddot{\vec{Q}}}{\left|\dot{\vec{Q}} \times \ddot{\vec{Q}}\right|^2}$ (4.14)

Integrating both sides of $ds = \tau \, d\psi$, an expression between the arc length s and the twist angle ψ is obtained as

$$\psi = \frac{\Delta s}{\tau_{avg}} \tag{4.15}$$

$$\tau_{avg} = \frac{\tau(t_k) + \tau(t_{k+m})}{2} \tag{4.16}$$

$$\Delta s = \sum_{i=k}^{k+m} \left|\vec{Q}(t_{i+1}) - \vec{Q}(t_i)\right| \tag{4.17}$$

Here, Δs is the total arc length of the spline part (Equation (4.17)), and ψ is the twist angle of the final point of the spline part. And also, τ_{avg} is the average torsion of the spline part and calculated by using Equations (4.14) and (4.16). Since the manipulator is discretely actuated, the twist motion is calculated for the spline part between the initial and final point of this part.

4.3.2 The Computer Algorithm

The required computer program for the analysis of the HRM has been written in MATLAB based on Equations 4.1-4.17. The algorithm is given as follows and the flow chart for the spline fitting method is given in Figure 4.11.

☐ The position of the end platform of each module is calculated for 192 configurations with respect to the initial frame by using the forward kinematics. The positions, lengths, azimuth, elevation and twist angles belonging to each module together with the related bit sequences are saved in a data file.

☐ The control points in the workspace are assigned in such a way that the manipulator reaches the target position and avoids the obstacles in the environment. These control points construct the control polygon on which the spline depends. The number of control points is optional. In this study, 8 control points $P_0,............., P_7$ are selected as suitable for two-part cubic curve. P_0 is the starting point of the manipulator while P_7 is the target position (The error between the end effector and target position is eliminated by the continuously actuated manipulator).

☐ The angles between the successive control polygon lines are to be controlled to check whether the radius of curvature of the spline is greater than the specified one or not. This value can be taken as 70^0 with respect to the motion capability and construction of the manipulator. This condition decreases the number of iteration or, at least, gives information about which of the control points violate the curvature consistency.

☐ Passing the control polygon angle test, now, the spline is constructed. The NURBS curve is selected since it allows the curve to be reshaped interactively. This spline consisting of cubic rational polynomials has the following features:

- The degree is 3, ($p = 3$)
- The knot vector T is selected as uniform due to its simplicity; however, it could also be selected as non-uniform.

  $T = [0\ 0\ 0\ 0\ 0.2\ 0.4\ 0.6\ 0.8\ 1\ 1\ 1\ 1]$
- The weight vector W is selected initially as $W = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$.

The knots t_i increase from 0 to 1 by 0.001. For each t value, the interval is found to calculate the corresponding basis functions. According to the knot span found, the basis functions are calculated for t_i values. Finally, the curve $Q(t)$ is obtained by using the related basis functions, weighting factors and control points.

☐ The whole curve is obtained by repeating this procedure in a loop until $t = 1$.

☐ After that, the spline is divided into the number of the modules proportional in length to the average length of each module.

☐ If the radius of curvature calculated does not satisfy the specified one, the $Q(t)$ is reconstructed, reducing the weight factor w with respect to the unsatisfied part. When $w_i = 0.2$ or $\rho_i > \rho^*$, the loop ends, and the convenient spline is constructed.

☐ The spline arc length is controlled.

☐ The construction of the spline is completed.

☐ The spline is used to select the convenient configurations as expressed in detail above.

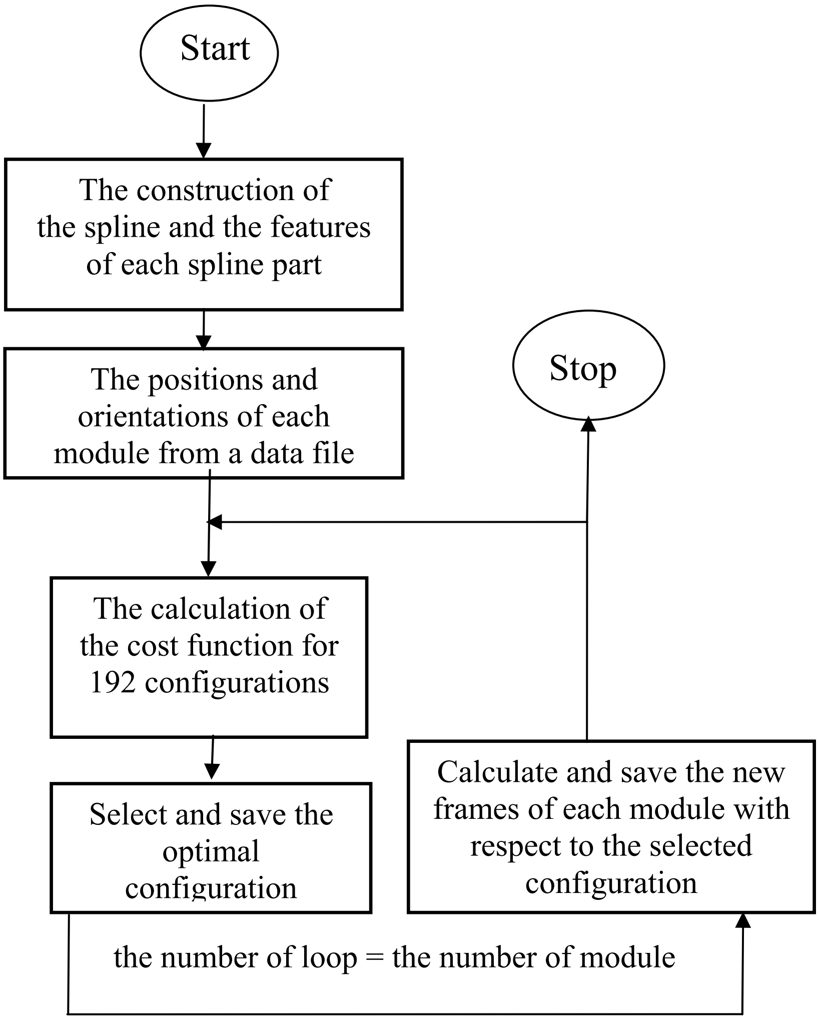

Start
The construction of the spline and the features of each spline part
The positions and orientations of each module from a data file
Stop
The calculation of the cost function for 192 configurations
Select and save the optimal configuration
Calculate and save the new frames of each module with respect to the selected configuration
the number of loop = the number of module

Figure 4.11 The Flow Chart of the Spline Fitting Algorithm

Using the spline fitting method, an example is given in Figures 4.12. In this figure, the control points, the control polygon, the spline and the selected configuration are defined in different line types, color and markers.

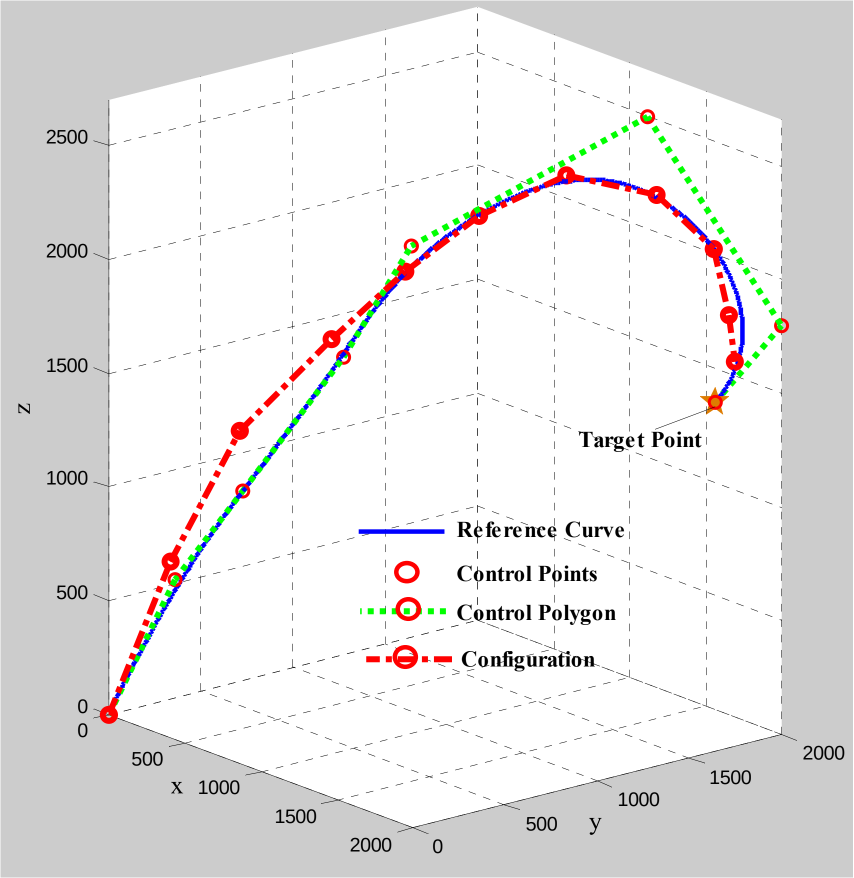


Figure 4.12 The Solution Configuration with the Spline Fitting Method

In this method, the selection of the configuration has been made module by module and in each selection one module has been used. So, $192\times10=1920$ iterations are run to resolve the HRM with ten modules. Using two modules and dividing the spline into five parts instead of ten is another alternative for implementing this method. This improves the success of the fitting operation since, instead of 192 possible configurations, $192\times192=36864$ possible configurations are searched for each fitting procedure. That is, the spline fitting method with two module-based has totally $36864\times5=184320$ iterations. The examples of this approach compared to the module-based method are given in Figures 4.13a-13b.

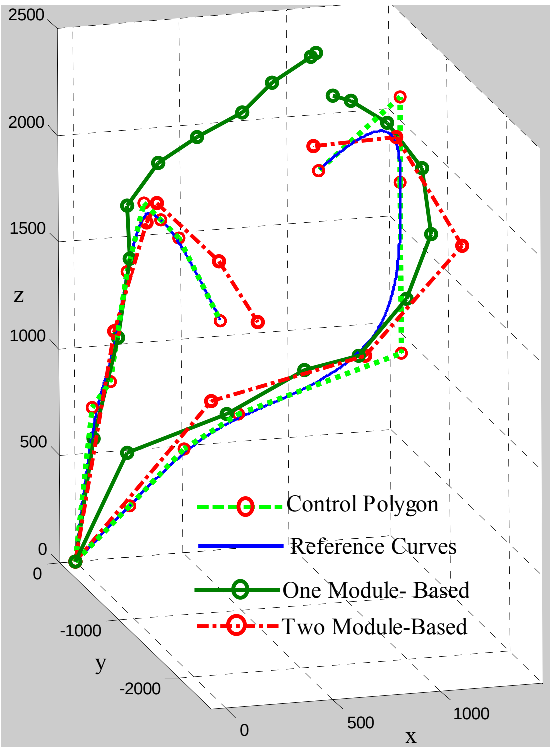


Figure 4.13a Two Module-Based and One Module-Based Fitting

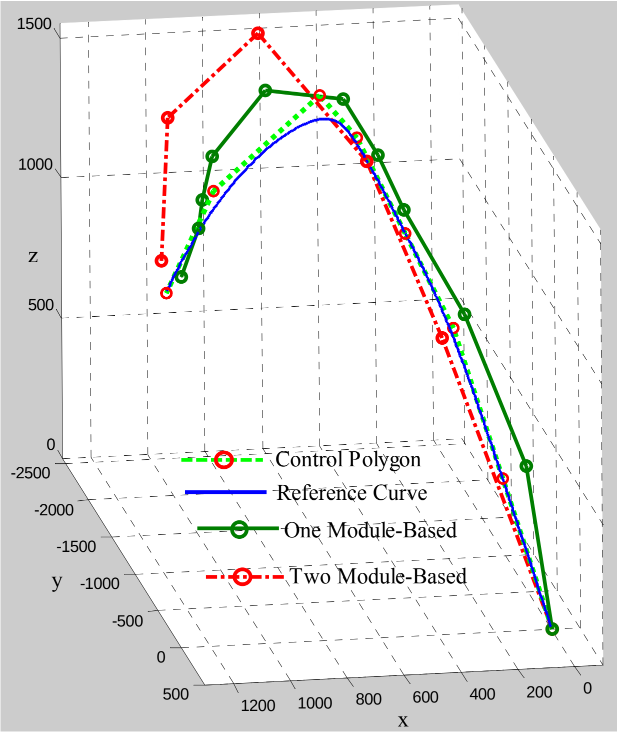


Figure 4.13b Two Module-Based and One Module-Based Fitting

4.3.3 The Intermediate Configurations for the Spline Fitting Method

The HRM may collide with the obstacles in its working space while it moves from an initial configuration to the final configuration. Therefore, according to the position and size of the obstacles, one or more than one intermediate configurations are needed to move safely. The NURBS can be used to construct the spline for the intermediate configurations if the workspace is complicated and therefore the manipulator needs some special shapes. Otherwise, a simpler version of spline such as a cubic spline can be used to select the intermediate configurations. In this study, a cubic spline, which is a polynomial curve $P(t)$ with four independent coefficients, is used. The problem here is to create a polynomial, $P(t)$, with an independent variable *t* that increases from 0.0 to 1.0. There are various approximations that can be used if we do not need exact equality at every point. However, for the purpose of understanding, we use the exact fit by a third order polynomial, $P(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$. Writing this equation in 3D spatial cubic spline, one obtains

$$\begin{bmatrix} P_x(t) \\ P_y(t) \\ P_z(t) \end{bmatrix} = \begin{bmatrix} c_{3x} \\ c_{3y} \\ c_{3z} \end{bmatrix} t^3 + \begin{bmatrix} c_{2x} \\ c_{2y} \\ c_{2z} \end{bmatrix} t^2 + \begin{bmatrix} c_{1x} \\ c_{1y} \\ c_{1z} \end{bmatrix} t + \begin{bmatrix} c_{0x} \\ c_{0y} \\ c_{0z} \end{bmatrix} \tag{4.18}$$

The boundary conditions and related coefficients are given such that:

- The spline starts with the origin.

$$\overline{P}(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} c_{0x} \\ c_{0y} \\ c_{0z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

- The slope of the spline at the origin is 0, i.e. the spline is oriented towards the z axis.

$$\overline{P}'(0) = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \Rightarrow \begin{bmatrix} c_{1x} \\ c_{1y} \\ c_{1z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}.$$

- The other boundary conditions are the final point and mid point of the spline. These are calculated for the intermediate configuration as follows.

$$\overline{P}(1/2) = \begin{bmatrix} c_{3x} \\ c_{3y} \\ c_{3z} \end{bmatrix} \times (1/2)^3 + \begin{bmatrix} c_{2x} \\ c_{2y} \\ c_{2z} \end{bmatrix} \times (1/2)^2 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (1/2)$$

$$\overline{P}(1) = \begin{bmatrix} c_{3x} \\ c_{3y} \\ c_{3z} \end{bmatrix} \times (1)^3 + \begin{bmatrix} c_{2x} \\ c_{2y} \\ c_{2z} \end{bmatrix} \times (1)^2 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times (1)$$

Noting that this is a set of simultaneous equations, define the constant matrix A from the equations above and writing the matrix vector equation.

$$C * A = X \tag{4.19}$$

where X is the column matrix formed by the boundary conditions, A is the system matrix and C is the coefficient matrix. Since X and A are both known, C can be directly computed as $C = X * A^{-1}$. For computational efficiency, the inverse of A is computed at most once and can be used for many different X vectors. Obtaining the C vector, this method is used to compute the intermediate values at the parameter t which are needed by our graphics application. Now using the method described above and considering a 3D point, we can compute the three C vectors, C_x, C_y and C_z. Thus, the three polynomials $P_x(t)$, $P_y(t)$ and $P_z(t)$ that interpolate and produce x, y and z for each value of t in the range 0.0 to 1.0 are obtained. The final and mid point of the spline for the boundary conditions is calculated according to Figure 4.14.

To find the end point of the spline for the intermediate configuration of the manipulator, we follow the procedure below.

- Determine a plane with the points P_s, P_f and O,
- Determine the normal vector of the plane, $\vec{N}$,

$$\vec{N} = \overrightarrow{OP_s} \times \overrightarrow{OP_f} \tag{4.20}$$

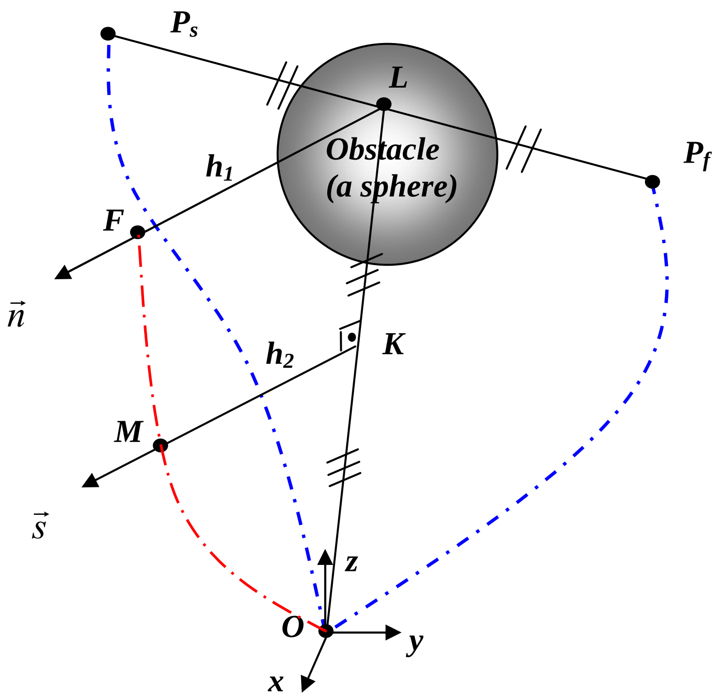


Figure 4.14 The Construction of Intermediate Configuration

- Using vector algebra, the new position of the end point of the spline (the point F) can be found. The coordinate of the end point of the spline can be found on the line whose direction is along the normal of the plane $P_sP_fO_s$ and passes through the point L.

$$\overrightarrow{OF} = \overrightarrow{OP_s} + \overrightarrow{P_sL} + h_1\vec{n} \tag{4.21}$$

where $\vec{n}$ is the unit normal vector, $\vec{n} = \dfrac{\vec{N}}{\left\|\vec{N}\right\|}$.

- Determine the L point, where $\overrightarrow{P_sL} = \dfrac{\overrightarrow{P_sP_f}}{k}$ and $k = \dfrac{\left|\overrightarrow{P_sP_f}\right|}{\left|\overrightarrow{P_sL}\right|}$.

Determine the mid point of the spline. The point M is found with respect to spline length, varying the distance h_2.

$$\vec{OM} = \frac{\vec{OL}}{2} + h_2 \vec{s}$$

Here, $\vec{s}$ is parallel to $\vec{n}$ and passes through the mid point M.

After this procedure, given the end points, P_s and P_f, of the initial and final configurations and the center point O_s of the sphere including the obstacle, the cubic spline to investigate the intermediate configurations can be constructed as shown in Figure 4.15. The values of the parameters h_1 and h_2 in this spline are adjusted with respect to two cases. The first one is that the manipulator should not collide with the obstacles in the motion, and secondly, the length of the spline should be consistent with the length of the manipulator. Although the manipulator is a discretely actuated manipulator, it goes from initial configuration to final one moving continuously. Therefore, we can see some intermediate states as shown in Figure 4.16. Thus, it can be seen that whether the manipulator touches or intersects the obstacle or not.

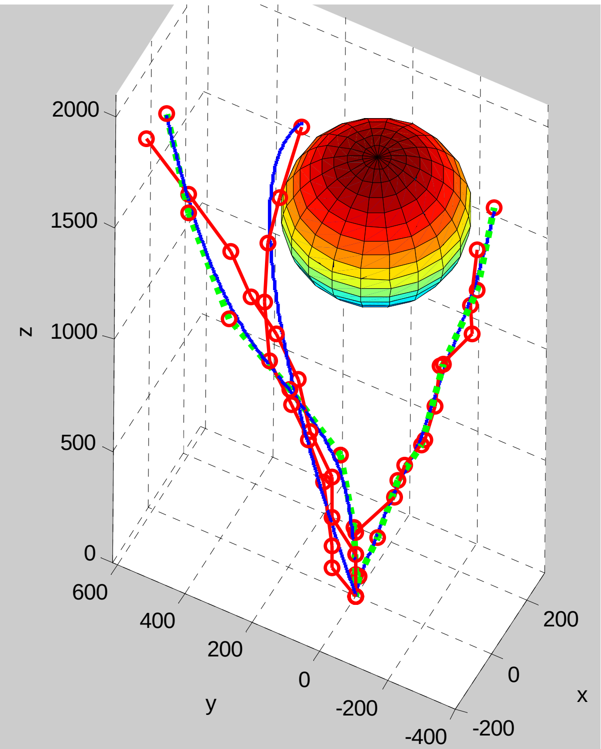


Figure 4.15 The Intermediate Configuration

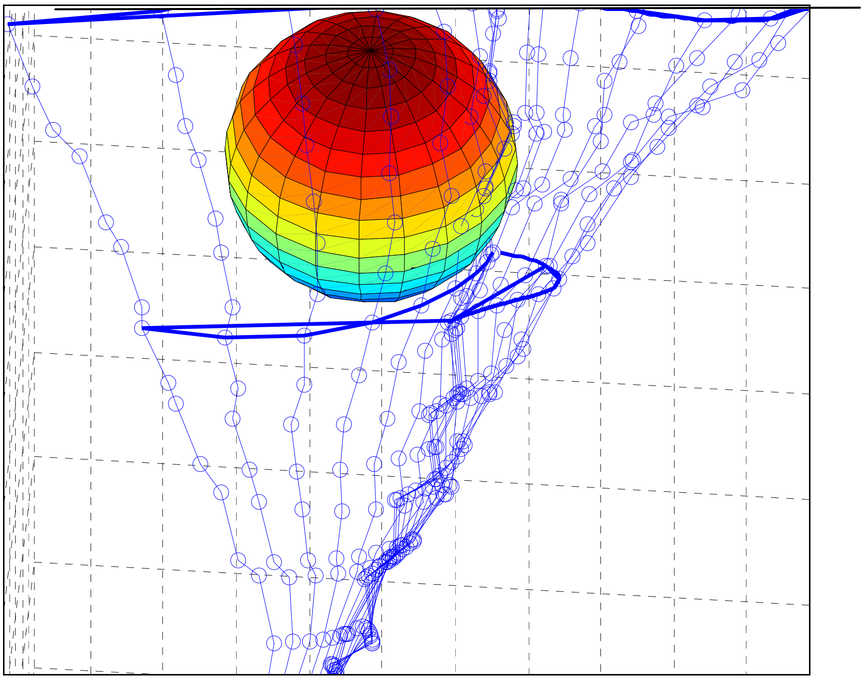


Figure 4.16 The Trajectory Control of the Intermediate Configuration

4.4 The Extended Spline Fitting Method for the Inverse Kinematics of the HRM

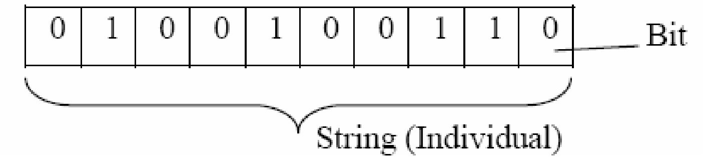
In the previous section, the inverse kinematics problem of the HRM has been examined by the spline fitting method in detail. This method gives only one configuration as a solution. Implementing the given criteria, which are the positional and orientational errors, the shape of the manipulator and the trajectory tracking, is very difficult by using only a single configuration. Also note that, in the first method, the orientation error and the trajectory following are not involved into the selection process. That is, we don’t have any control over these criteria. At the same time, the selection operation is carried out module by module and from the bottom to the top. In other words, there is not any flexibility to select the configurations of the intermediate modules. This causes the larger positional and orientational errors on the top of the manipulator and, even, the manipulator can excessively diverge from the given reference curve. On account of these reasons, the spline fitting method should be improved to fulfill the given tasks properly. Therefore, a second method, the extended spline fitting method, is presented. In this method, a genetic algorithm is used to solve the inverse kinematics problem effectively. Genetic algorithm is very suitable for the kinematics of binary hyper redundant manipulator since it uses binary coding system in its algorithm. One of the most important parameters of genetic algorithm is its initial population when to start to the optimization procedure, and the success in genetic algorithm depends in great extend on this given initial population. If the initial population is given far away from the solution, which is also a population, the desired solution may not be approached. Because of this, in this method, the solution configuration determined in the first method is assigned as an initial population for the genetic algorithm. This method generates a solution set with many possible configurations vicinity of the configuration found in the spline fitting method. This approach provides a control over the shape of the manipulator because the possible configurations heap up around the reference curve and they enable us to investigate the configuration with the optimal trajectory between two successive configurations due to the solution set. The best configuration selected by the extended spline fitting method does not only fulfill all required criteria but also it supplies the improvement in the positional errors.

4.4.1 Genetic Algorithm

A solution set is necessary for a better solution in the inverse kinematics of the HRM. For this aim, using genetic algorithm is an effective approach to implement the given tasks for the motion planning of the HRM. Genetic algorithm has been widely used to solve unusual of difficult optimization problems which are not easily solved by conventional optimization methods [51]. Genetic algorithm is a nondeterministic stochastic search algorithm based on the mechanism of natural selection and natural genetics. Although the solution space is very large, genetic algorithm approaches a better solution in a shorter time. Unlike the conventional search techniques, genetic algorithm starts with an initial set of random solutions called “population” satisfying the boundary conditions and/or system constraints of the problem.

In general, a genetic algorithm has five basic components.

(1) A genetic representation of potential solutions: A population consists of individuals that are chromosomes. Individual represents potential solutions in the optimization problem. The parameters of problem are encoded to be initial population, and each solution is also encoded as a finite length string. There are many types of solution representations such as the binary coding and permutation coding. The basic one is the binary coding consisting of binary digits “0” and “1”. Each bit in a string represents the characteristics of solution. A string, which is an individual, represents a potential solution, for example,



(2) A way to create a population: In genetic algorithm, instead of a single initial solution, an initial set of potential solutions is used in the solution of the problem. This provides the algorithm to be able to work without reaching a local optimum. Initial population is a very important component because if the initial population is closer to the solution, the genetic algorithm gives much better solutions in a shorter time.

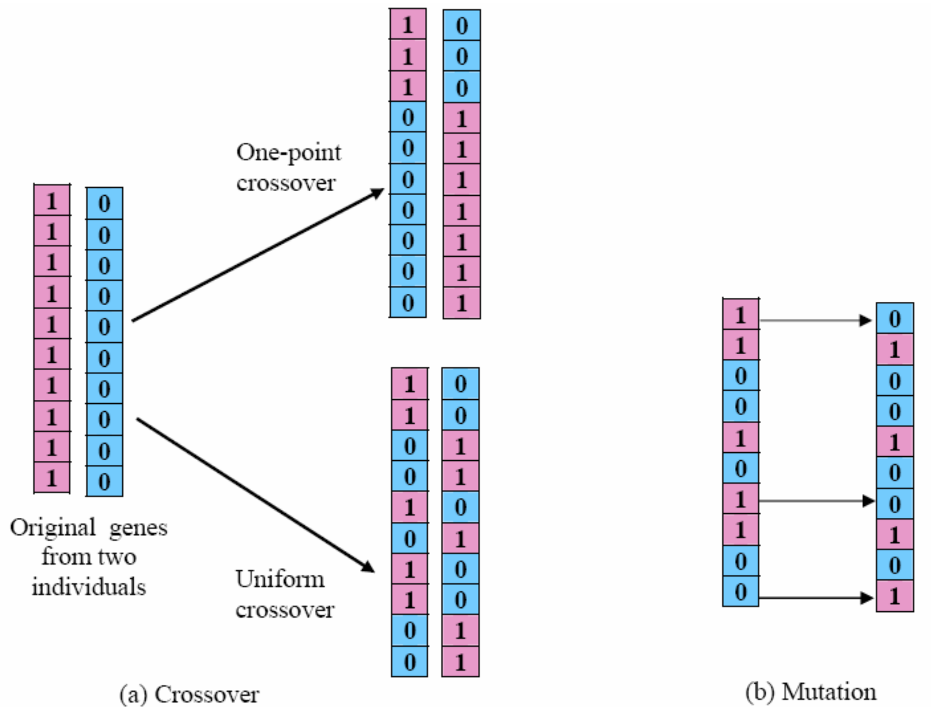
(3) An evaluation function: One common application of genetic algorithm is function optimization. Individuals in the current population are decoded and evaluated according to some predefined quality criteria that are one or more than one function. This function rates the solutions in terms of their fitness. Because of this, it is called as fitness function. Each individual is selected according to fitness value in which existing members of the current solution pool are replaced by newly created members. In genetic algorithm, only this part is peculiar to the problem.

(4) Genetic operators: Genetic operators are some operations applied on the current population. The aim of these operators is to produce new better generations and to expand the space of the searching algorithm. Genetic operators alter the genetic composition of offspring (next generation). These genetic operators are as follows:

- **Crossover:** Crossover is the main genetic operator. It takes two individuals and cuts their chromosome strings by using the random positions. After that, they have head and tail segments. The tail segments are swapped over to produce two new full length chromosomes where two new offspring inherit some genes each parent. A single point, two points or uniform crossover would be chosen for cut-points. This methods works well with bit string representation. An example of crossover is shown in Figure 4.17a. The normal probability of crossover varies between 0.6 and 1. The performance of genetic operators depends to a great extent on the performance of the crossover operator.

- **Mutation:** This operator flips some of the bits in an individual. It serves to introduce the diversity in the population. Mutation can occur at each bit position in a string with some very small probability that usually varies between 0.001 and 0.01. It changes an element from a binary string that is generated by the crossover and replaces a bit by digit “0” or “1” as shown in Figure 4.17b. In general, after a certain period, the chromosomes (individuals) become a homogenous structure in the population because of the effect of selection and crossover operations. This causes the chromosomes to stop their developments without reaching their optimum level. On account of this, mutation brings about the diversity of the chromosomes in the population.

- **Selection:** A selection is the process of determining a particular individual that is chosen for the reproduction and the number of offspring in which an individual will produce. It transforms the fitness values of individuals to the probability value for reproducing by the probability of reproduction according to the fitness values. The binary strings that have higher fitness values are more likely to be selected as parents. “Roulette Wheel”, “Tournament”, “Stochastic Universal Sampling” methods are generally used for selection.



(a) Crossover

(b) Mutation

Figure 4.17 Crossover (a) and Mutation (b) Operators

(5) Parameter Values: In implementing a genetic algorithm, the fifth decision to make is how to set the values for the various parameters. These parameters typically interact with one another nonlinearly, so they can not be optimized at a time [52]. Genetic algorithm uses population size, probability values of applied genetic operators, number of generation, number of individuals etc.

After some definitions, the work of the genetic algorithm can be summarized in the following steps.

Step 1: An initial population is assigned.

Step 2: Individuals are decoded and evaluated according to the fitness function. This function defines how much good individual is.

Step 3: A selection is made by mating individuals with respect to their fitness value.

Step 4: Old individuals are removed to create space for the new ones. Hence, the population with constant size is supplied.

Step 5: The success of the population is found again by evaluating the fitness values of all individuals.

Step 6: A great number of populations are constructed by repeatedly running the genetic algorithm. Go back to Step 3 if the time is not over.

Step 7: The best individual until that time is the result since, in the evaluation of the populations, the better individuals are saved.

To use a genetic algorithm in binary manipulators is very natural and convenient because the potential solutions for binary manipulators are already given in a binary manner. In many engineering problems, the conversion from the real world problem to a binary encoding can be very complex because of real number representation. However, this complexity does not exist in the inverse kinematics of binary manipulators. All joint variables constructing individual in the population can be represented by using digits “0” or “1”. Thus, all genetic operations can be performed on candidate solutions.

4.4.2 Using Genetic Algorithm in the Extended Spline Fitting Method

In this method, many configurations for the solution of the inverse kinematics of the HRM are generated by using the genetic algorithm in the vicinity of the configuration found in the spline fitting method. An example of this method is shown in Figure 4.18. In our genetic algorithm search, since the aim is to maximize the fitness values to select better individuals, the fitness function is set to $f = \dfrac{1}{\left|\overline{r}_i - \overline{r}_{target}\right|}$.

Despite choosing of one individual of the population from all the generations as a solution, many individuals from all regenerated populations which satisfy the constraint ($d^*_{\min} < \left|\overline{r}_i - \overline{r}_{target}\right| < d^*_{\max}$) are selected and stored. $\left|\overline{r}_i - \overline{r}_{target}\right|$ is the distance between the task position and the tip points of the HRM for the generated configurations in searching. Here, $d^*_{\min}$ and $d^*_{\max}$ are defined with respect to the workspace of the fine tuning manipulator as shown Figure 4.1.

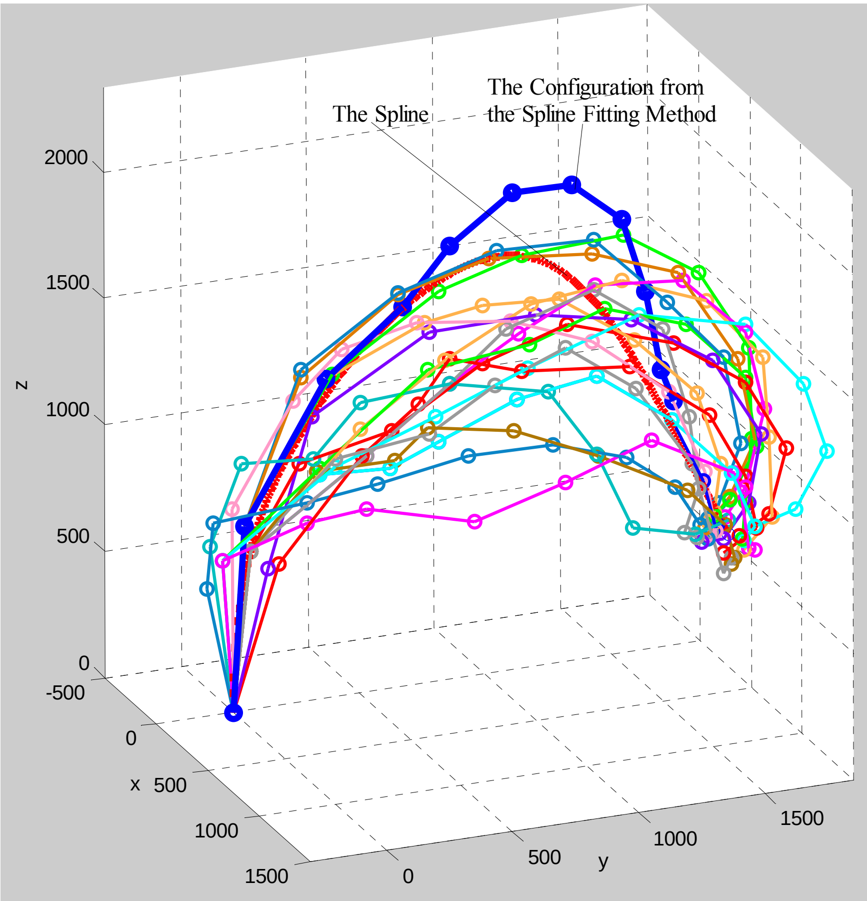


Figure 4.18 The Extended Spline Fitting Method

In genetic algorithm, the main algorithm design variables are the population size, the number of generations of evaluation, the mutation factor, the crossover ratio, the crossover method, the selection type and the scaling factor. In this study, many of these algorithm variables are heuristically based on general performance of the algorithm. In this method, the size of the populations, at the initial step, is two, where the first one comes from the spline fitting method and the second one is given by flipping one or more than one bits of the first one since a genetic algorithm uses populations with even numbers. However, at later steps, it takes the fixed assigned value, e.g., in this study, each population has one thousand individuals in each of the iterations. The parameter values used in this algorithm are given as follows.

The crossover type: uniform crossover

The probability of crossover: 0.9

The mutation probability (per bit): 0.01

The selection type: The Stochastic Universal Sampling method (In Roulette Wheel selection method, the wheel is span N times, where N is the number of individual in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. Rather than spin the roulette wheel N times to select N parents, the Stochastic Universal Sampling spins the wheel once, but with N equally spaced pointers, which are used to select N parents).

The sigma scaling coefficient: 1 (Sigma scaling is a method which keeps the selection pressure, i.e. the degree to which highly fit individuals are allowed many offspring, relatively constant over the course of the run rather than depending on the fitness variances in the population. The aim is to make the genetic algorithm less susceptible to premature convergence.)

The mask repository factor: 5 (Without changing the behavior of the genetic algorithm, it significantly improves the speed of the code. It is used in crossover and mutation operators to mask from a pregenerated repository of randomly generated bits.)

For more detail information about the concepts above, see [51] and [52].

4.4.3 The Selection Process from the Configuration Set

The selection process is made in turn with respect to the criteria given in the Subsection 4.2. This contains four successive operations. In each step, the inconvenient configurations are eliminated for the later operations so that the fewer configurations are searched. To solve the inverse kinematics with the extended spline fitting method, the order of the selection procedure of the configuration is given as follows.

1) The positional error, or the Euclidian norm, for the HRM is expressed as a distance between the tip point of the manipulator and the given target position. The configurations satisfying the given threshold, $d_{\min} < d_{error} < d_{\max}$ (shown as Figure 4.1), are chosen, and the other are eliminated. Here, $d_{\min} < \left\| P_i - Q_i \right\| < d_{\max} \quad i = 1 \ldots\ldots\ldots n_1$. After this process, the search space, $W(n_1)$ becomes $W(n_2)$, $n_2 << n_1$.

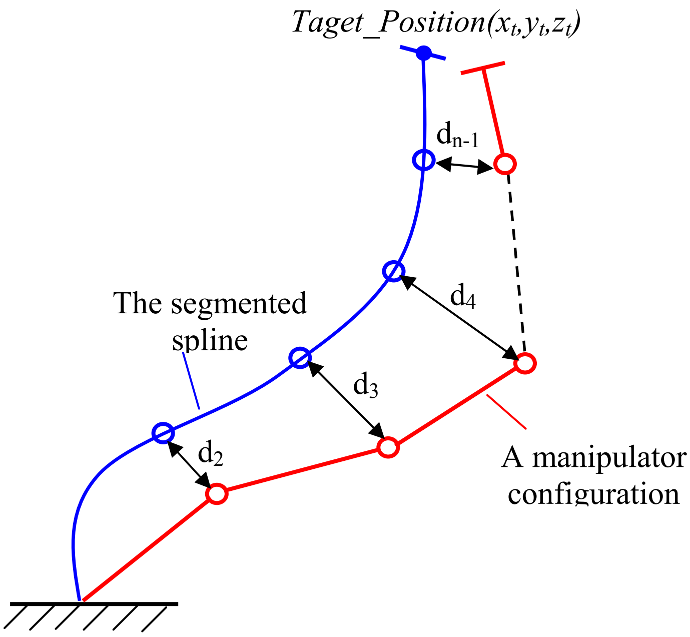
2) This step consists of the orientation of the manipulator. In this step, the configurations are selected in such a way that the burden on the fine tuning manipulator is reduced fairly. For this purpose, a cost function with the orientation angles of the possible configuration and the given orientation angles with respect to 323 Euler angle convention is minimized. The calculations are given in Appendix A for the Euler angles. As a result of this process, the search space, $W(n_2)$ becomes $W(n_3)$, $n_3 < n_2$.

$$Cost\ function = L_1\left(\phi_M - \phi_{given}\right)^2 + L_2\left(\theta_M - \theta_{given}\right)^2 + L_3\left(\psi_M - \psi_{given}\right)^2$$

3) The shape control of the manipulator is implemented in this stage. To avoid the obstacles and obtain the desired shape, the algorithm provides the convenient configurations so that the HRM can not collide with the obstacles represented by spheres. For an efficient algorithm, the reference curve is assigned such that it satisfies the requirements. Since the initial position of the configurations is already fixed and their final position is known, only intermediate positions should be defined for the shape of the manipulator. For this aim, the reference curve is divided into the segments like in the spline fitting method, and the selection is made with respect to a cost function. This cost function is a function of the total distance between the tip point of the intermediate modules and the related spline segments as shown in Figure 4.19. However, the cost function with too many parameters can not give successful results. In this application, the total distance belonging to third, fifth and seventh modules or only fifth module is taken for the cost function.

$$Cost\ function = d_2 + d_3 + \ldots\ldots\ldots + d_{n-1}$$

In the result of this stage, the set of convenient configurations is chosen for the later process. They are controlled graphically whether to collide with the obstacles or not. After this process, the search space, $W(n_3)$ becomes $W(n_4)$, $n_4 < n_3$.



Taget_Position(x_t,y_t,z_t)
d_{n-1}
d_4
d_3
d_2
The segmented spline
A manipulator configuration

Figure 4.19 The Shape Criterion for the Selection of Configuration

4) At the last process, the selection is made with the remaining configurations. The resulting configuration satisfies all the three criteria, and should have the smoothest motion between two successive configurations. Given a reference curve from the top point of the initial configuration to the target position for the next configuration, the configuration with the smoothest motion is defined from the other cost function defined according to Figure 4.20.

$$Cost\ function = d_1 + 2d_2 + d_3 = \|P_i(j_1) - Q(j_1)\| + 2\times\|P_i(j_2) - Q(j_2)\| + \|P_i(j_3) - Q(j_3)\|$$

Here, P_j and Q_j represent the possible trajectories and the reference curve respectively. These curves are parametric, and the parameters are taken as $j_1 = 20$, $j_2 = 50$ and $j_3 = 80$ with 100-step curves.

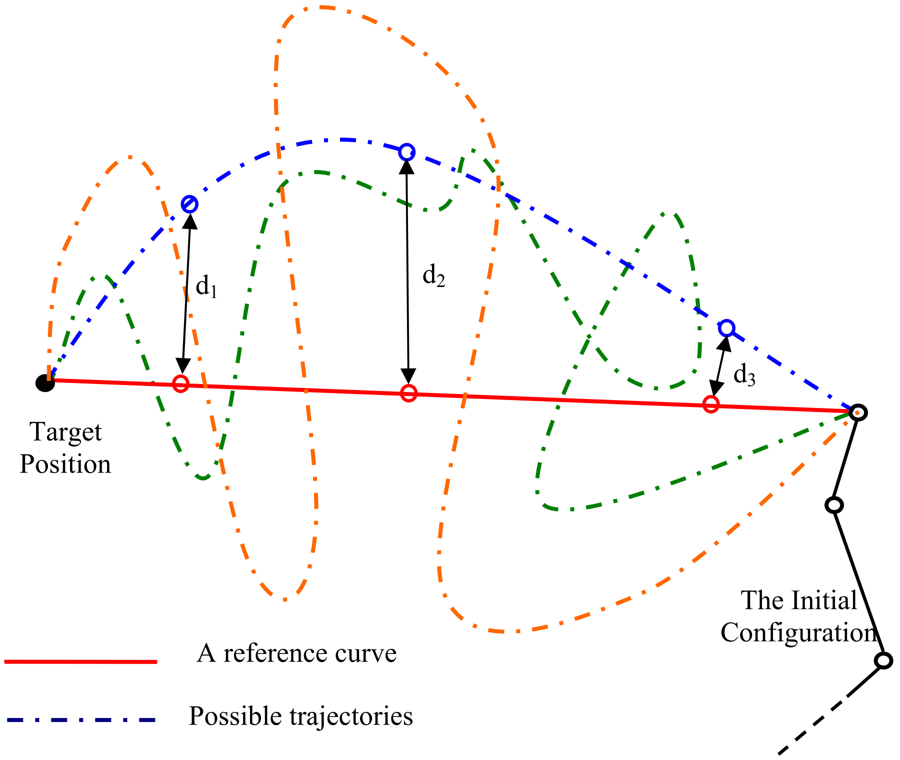


Figure 4.20 The Trajectory Following Criterion

At the end of all four stages, the optimal configuration is selected. This is not considered as globally optimal configuration, but it is the local optimal. For example, in Figure 4.18, the positional errors for different possible configurations vary between 1.2%-2.8% while this error is 17.05% for the configuration found in the first method. The other applications of this method are given compared to the other method as shown in Figure 4.26-4.30.

4.5 The Workspace Filling Method

The hyper redundant manipulator with ten modules has $\left(2^3 \times 2^3 \times 3^1\right)^{10} \approx 6.8 \times 10^{22}$ possible configurations. The calculation of position and orientation with the forward kinematics for all these configurations is impossible even though well developed and fast computers are used. The calculation for each configuration consists of not only the pose of the end-effector but also the poses of the tip point of every ten modules, which are based on many parameters in the loop. Because of this, instead of using this whole configuration set, the subset of this in a reasonable number is selected to construct a workspace. This workspace is represented with the tip point of the HRM. Here, the workspace involves only the discrete part of the HRM. These points are obtained with the forward kinematics for a finite number of configurations constructed by the related bit sequences. Each configuration sequence is a 1x80 row matrix including "0"s and "1"s such as $\left[01101100..........00110\right]_{1\times 80}$. In fact, there are 70 independent joints. However, each configuration sequence involves 80 bits because the submodule with twist motion is represented by two bits for one joint variable. At first, the sequences are constructed randomly. After that, the positions of the tip point of the HRM related to the initial reference frame for each configuration are calculated, and then the position vectors and related configurations are stored in a computer library.

Figure 4.21 shows the workspace with 100000 randomly selected configurations. This method is called "the workspace filling method" because it fills the workspace with the discrete configurations of the HRM. This operation is made by using the simple computer program written in MATLAB. The row matrix with 80 bits represents the actuator states for ten modules. The computer program generates simultaneously the number between 0 and 1 for each bit. The bit is taken as "0" if the number generated randomly is between 0 and 0.5, and otherwise it is taken as "1". After constructing the workspace, the inverse kinematics is solved by selecting the convenient configuration form this workspace.

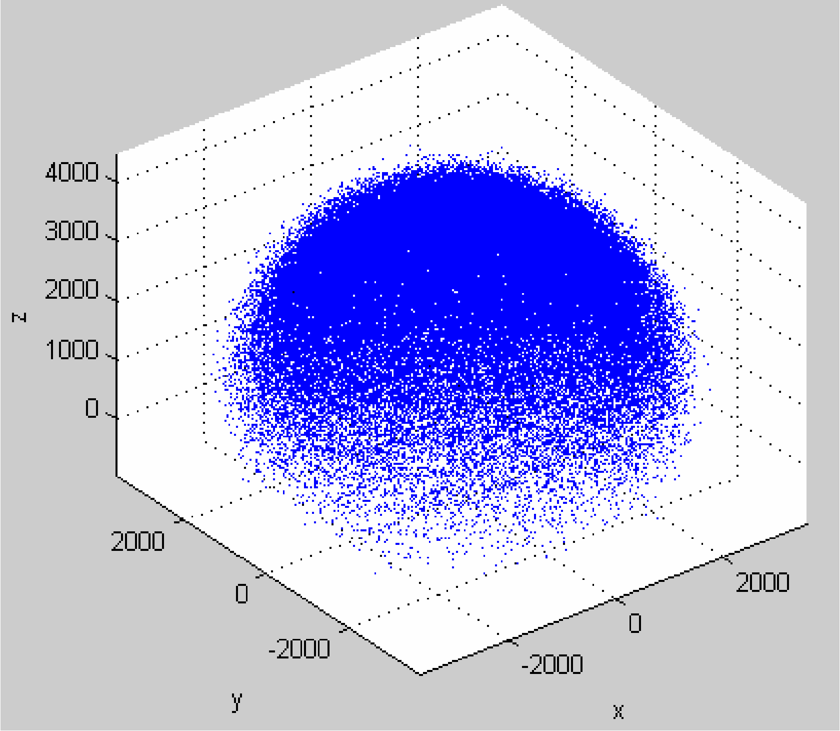


Figure 4.21 The Workspace with 100 000 Configurations

The workspace resembles almost like a sphere. It is noted that the number of mark points in some regions especially in the border of this workspace is rather small. Working on these regions can not give an efficient solution for the motion planning of the HRM. For example, the region represented by a sphere with its radius $r = 1000mm$ and centered at (0,0,0) involves only two mark points. That is, the manipulator has only two configurations whose end-effector is located at the inside of this sphere. In fact, this manipulator is expected to behave similar to its continuously actuated counterparts because of its $\left(2^3 \times 2^3 \times 3^1\right)^{10} \approx 6.8 \times 10^{22}$ possible configurations. More than 100000 configurations can be chosen to construct for this workspace. However, this does not guarantee enough number of configurations for any working region. In addition, the searching time becomes larger by the increased number in the same workspace. Therefore, the genetic algorithm will be used as another method to construct the efficient workspace. In addition, if it is desired to have concentration on a particular region of the workspace, then, that region is filled by using the genetic algorithm.

4.5.1 Using Genetic Algorithm in the Workspace of the HRM

New denser workspace is necessary for the efficient solutions in the inverse kinematics of the HRM. Instead of working on the whole workspace, one or more than one specified workspaces involving the solutions of possible configurations will be obtained with respect to the task regions. Here, the aim is to concentrate on some particular regions and to involve many more possible configurations in them. These workspaces are restricted by a constraint. This constraint is $\left|\overline{r}_i - \overline{r}_{target}\right| \leq r^*$, where $\overline{r}_i$ are the end points of the HRM, $\overline{r}_{target}$ is the origin points of the studied regions and r^* is the bounded value. In fact, this expression represents a sphere. The genetic algorithm provides enough number of configurations in the sphere. This method has two advantages. First, denser workspaces can be obtained and, secondly, the searching time becomes relatively smaller by searching the restricted workspaces.

To apply the genetic algorithm to the workspace generation for the inverse kinematics of the HRM, a computer program written in MATLAB implements the genetic algorithm for this study. In the simulations, the initial population has been selected from 100.000 randomly selected configurations that the library constructed in the previous study to approach better solutions since the better initial population selected the better generations yielded. In this selection, the size of the populations, at the initial step, changes with the initial selection from the randomly selected workspace with respect to the given constraint. This constraint collects the tip points of the HRM in a sphere with certain radius and origin point. However, at later steps, it takes the fixed assigned value expressed as in the previous method.

In this method, the fitness function is the same as one in the extended spline fitting method, i.e. $f = \dfrac{1}{\left|\overline{r}_i - \overline{r}_{target}\right|}$. Again, many individuals from all regenerated populations which satisfy the constraint ($\left|\overline{r}_i - \overline{r}_{target}\right| < r^*$) are selected and stored. This method starting with the possible solutions as an initial population increases the number of configurations in the workspaces to have concentration on some particular regions. Many workspaces can be obtained easily for different applications. The following figures (Figure 4.22a-b) show how successful the method is. Table 4.1 shows the number of possible configurations randomly constructed and by using the genetic algorithm for the different regions. Working on these restricted workspaces is useful for successive multiple tasks.

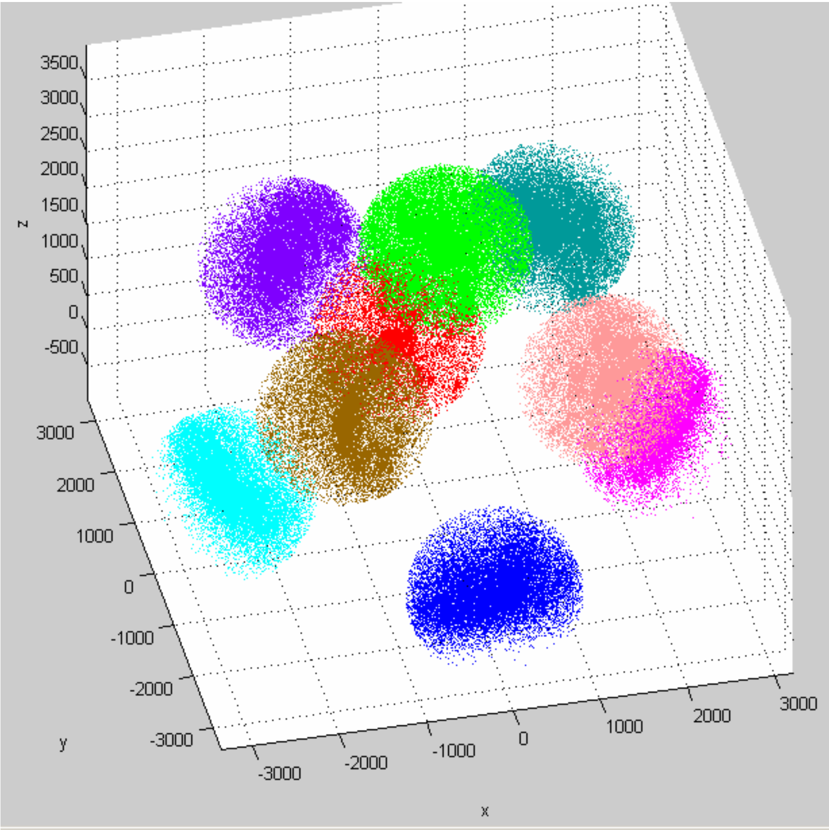


Figure 4.22a The Restricted Workspaces for Different Task Regions

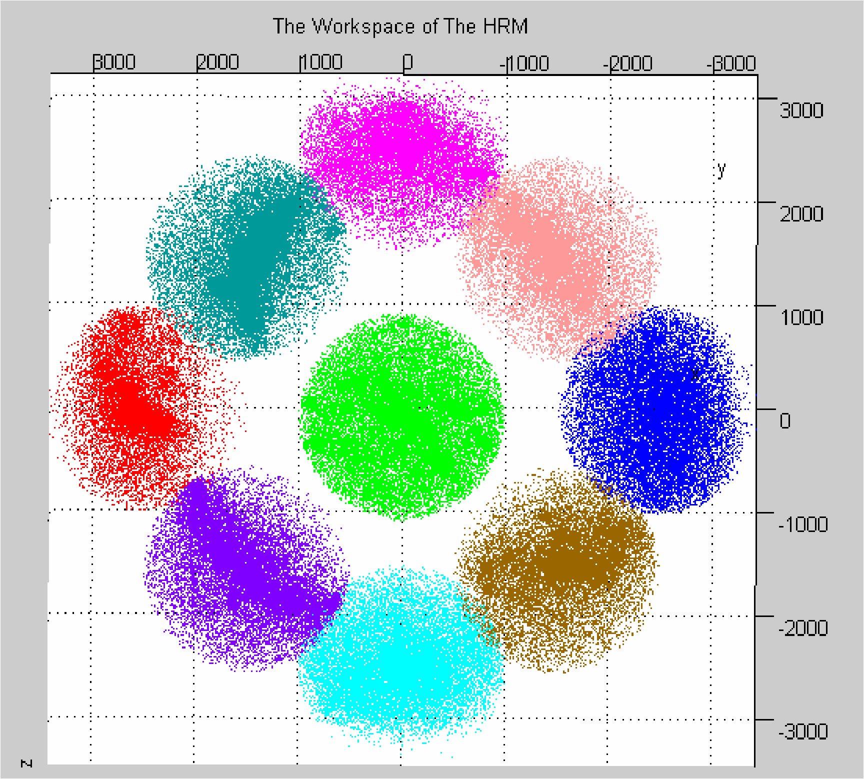


Figure 4.22b The Top View of the Restricted Workspaces

Table 4.1 The Number of Possible Configurations Constructed Randomly and by Using the Genetic Algorithm for the Different Workspace Regions

The coordinate of the center of the workspace	The number of the configurations randomly constructed	The number of the configurations with genetic algorithm	The total computation time (s)
(2500,0,1500)	489	1676	121.5
(-2500,0,1500)	410	2063	456
(0,2500,1500)	345	1405	237.5
(0,-2500,1500)	425	802	103
(0,0,3500)	2328	3811	179
(1500,1500,2000)	577	839	122.5
(1500,-1500,2000)	665	1617	832.5
(-1500,1500,2000)	472	1339	449.5
(-1500,-1500,2000)	564	1361	856.5
(0,0,0)	0, r =500mm 2, r =1000mm	1670, r = 500mm	219

4.5.2 The Selection Process from the Workspace

The selection process for the workspace filling method is made again, in turn, with respect to the given four criteria, which are the same as those in the extended spline fitting method. At the end of these four stages, the optimal configuration is selected. As an example, some possible solution configurations as shown in Figure 4.23 are given to select the best one. Figure 4.24 shows the best solution configuration together with the initial configuration and related trajectory.

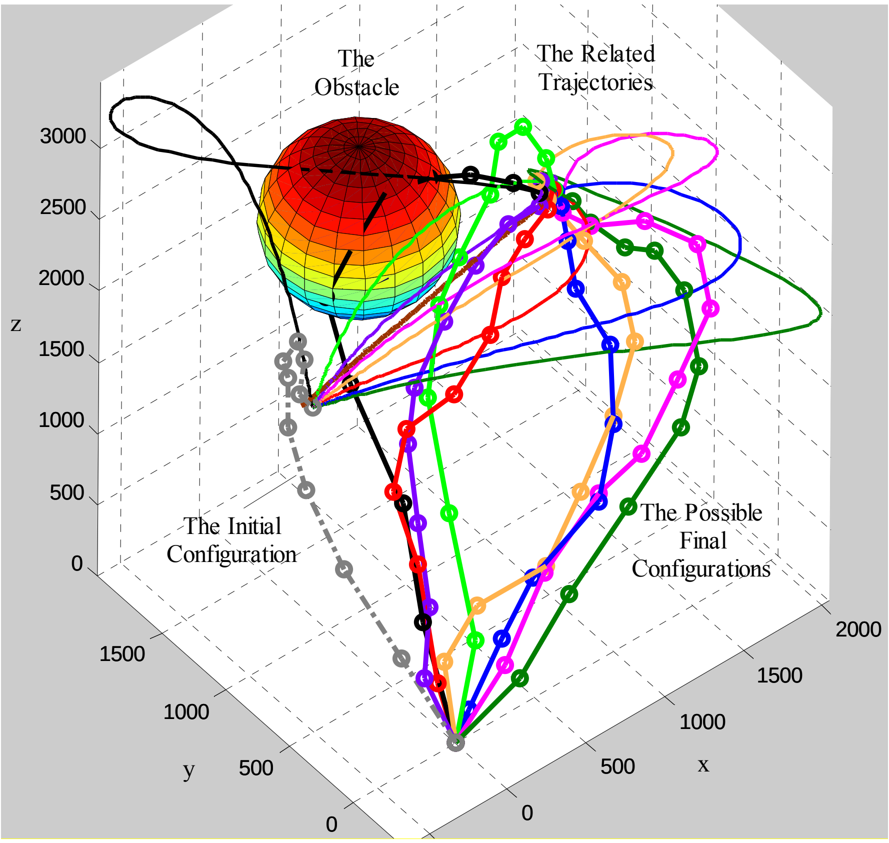


Figure 4.23 The Possible Configurations and Related Trajectories

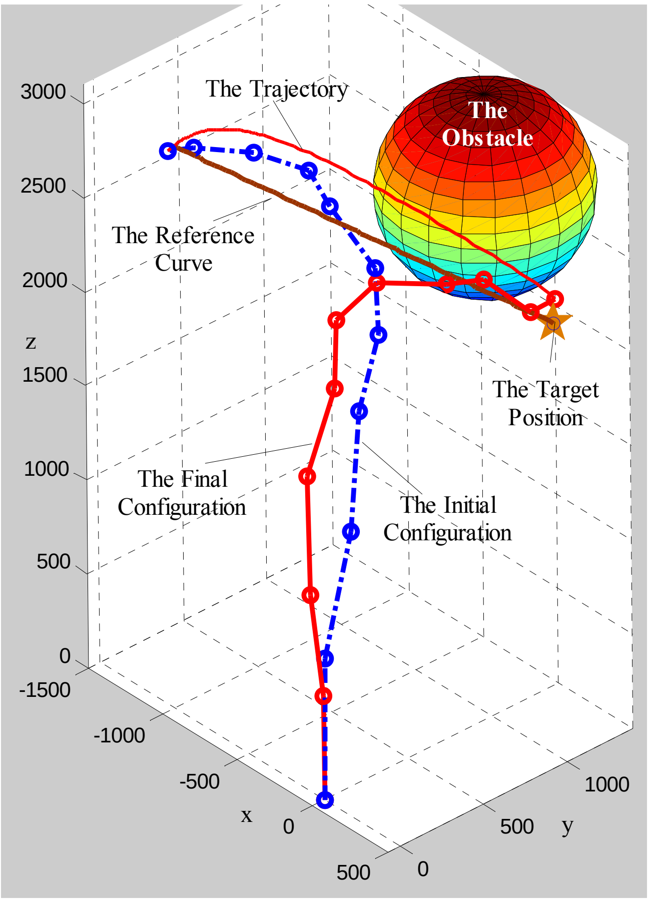


Figure 4.24 The Selected Configuration by Using the Workspace Filling Method

The workspace filling method can be used conveniently for avoiding moving or unexpectedly appearing obstacles because, even for a very close neighborhood of the desired end point position, its readily available configuration diversity is very numerous as shown in Figure 4.23 as an example. For a more specific example, illustrated in Figure 4.25, it is assumed that firstly the manipulator with the configuration 1 is selected to avoid the obstacle A. However, after a while, the manipulator with the configuration 1 runs into another obstacle (the obstacle B) as shown in the figure. In that case, by using the workspace filling method, the configuration of the manipulator is changed from the configuration 1 to the configuration 2 or 3 to avoid the obstacle A as well as the obstacle B by compromising very slightly in the desired position of the end point.

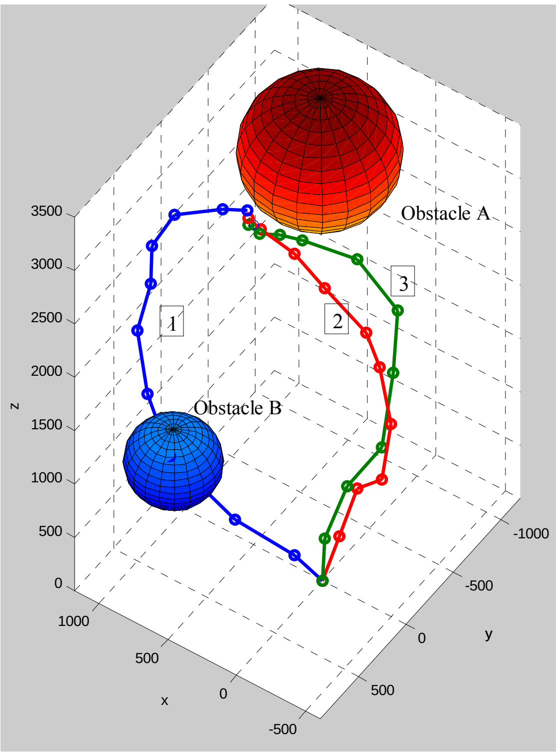


Figure 4.25 Avoiding the Moving Obstacles

After all these specifications, some examples about the workspace filling method compared to the other methods used in this thesis are given in Figures 4.26-4.30.

4.6 The Comparison of the Inverse Kinematics Methods

In this thesis, three different methods have been used for the inverse kinematics of the HRM. These are the spline fitting method, the extended spline fitting method, and the workspace filling method. In fact, the extended spline fitting method is not absolutely a different approach from the others because it is a combination of the spline fitting method and the genetic algorithm. This uses the advantages of both methods. In the solution of the inverse kinematics, selection of which method to be used is based on the given task. The positional and orientational error, the shape of the configuration and the trajectory following are the criteria for defining the selection of the method. To compare these methods, some special selected tasks are examined with respect to the positional error and the shape of the configuration. In order to see the behavior of these methods well, the examples have been selected with different features defined by reference curve. These curves are selected as line, parabolic or hyperbolic-like planar curves and spatial curves. In addition, the tasks, which are not convenient with the motion capability of the HRM, are given as another example. For the comparison of these three methods, 11 different tasks numbered from 1 to 11 are given and shown on the same figures according to their categories as shown in Figures 4.26-4.30.

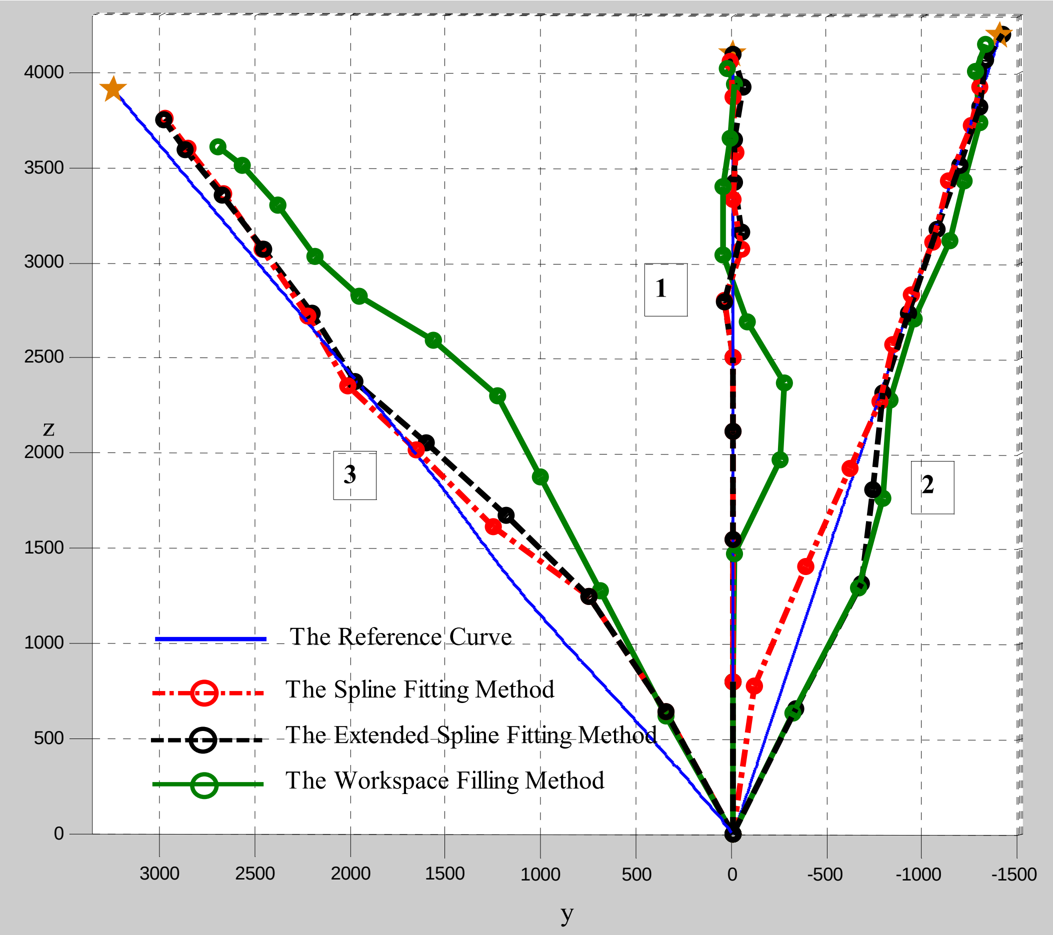


Figure 4.26 The Planar and Linear Configurations

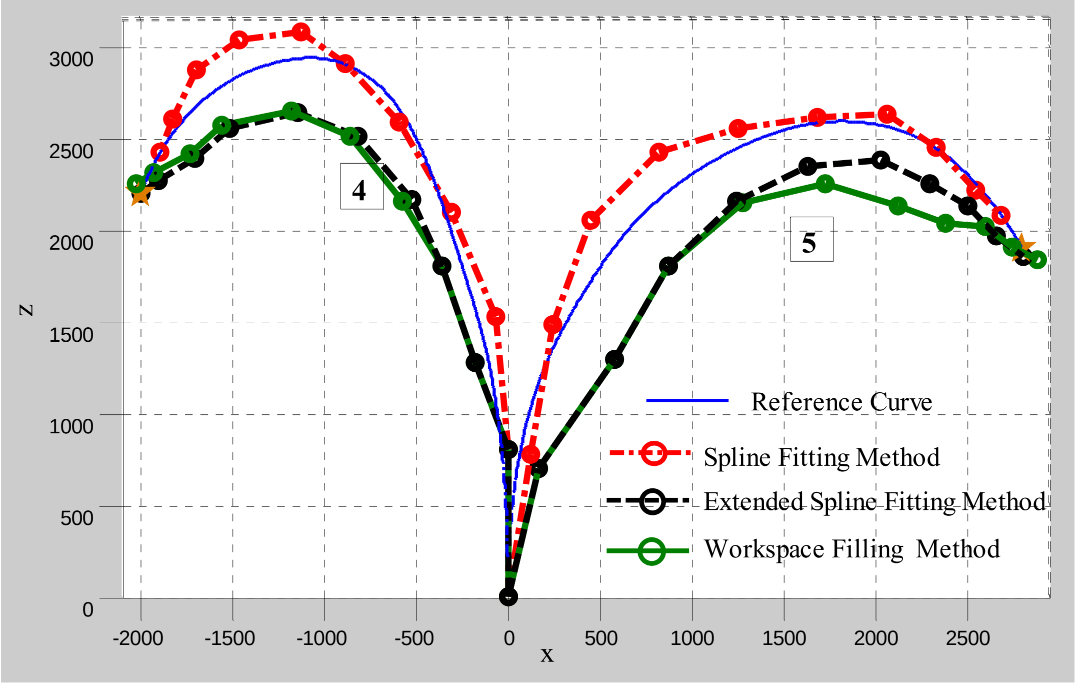


Figure 4.27 The Planar and Curve-Like Configurations

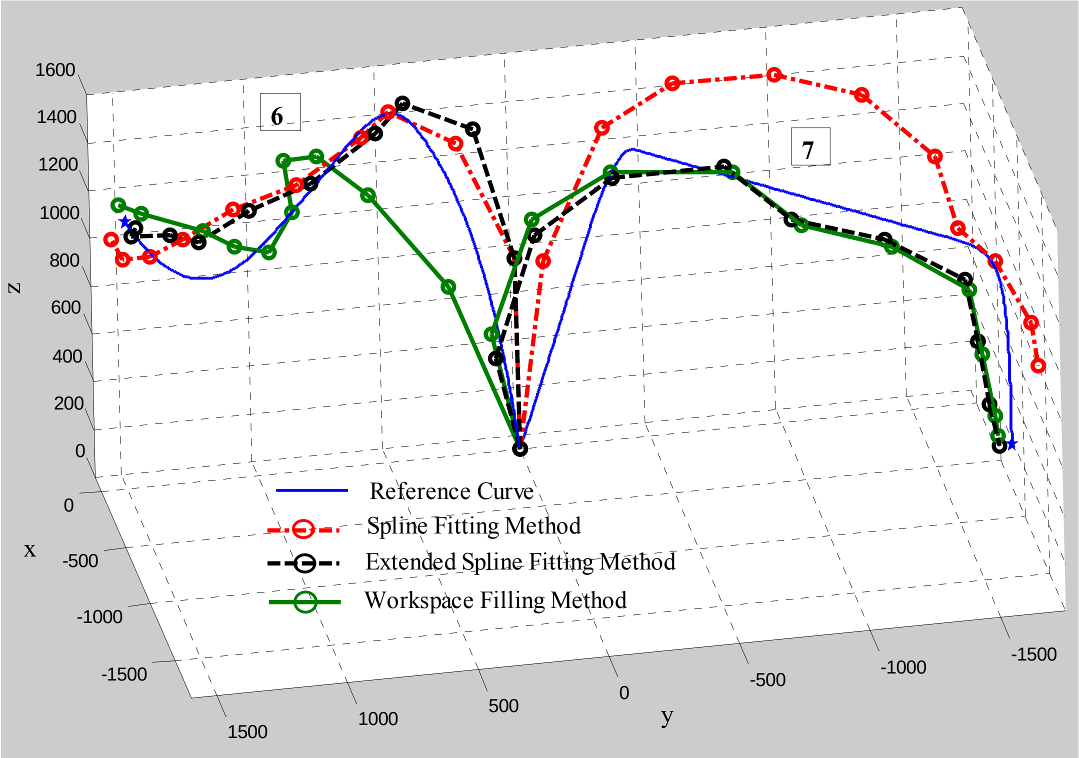


Figure 4.28 The Spatial Configurations

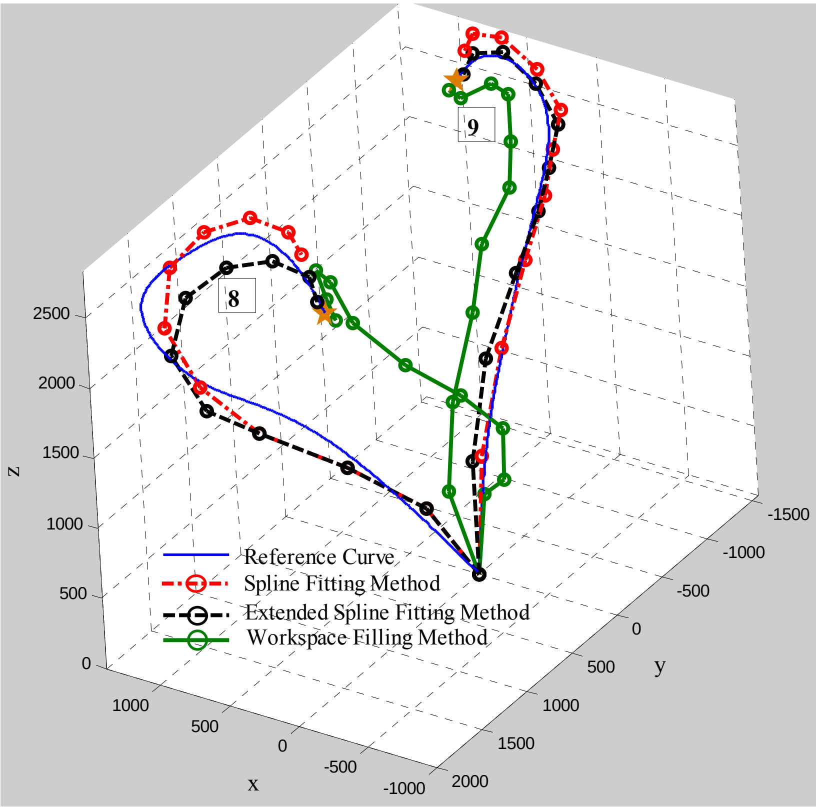


Figure 4.29 The Spatial Configurations

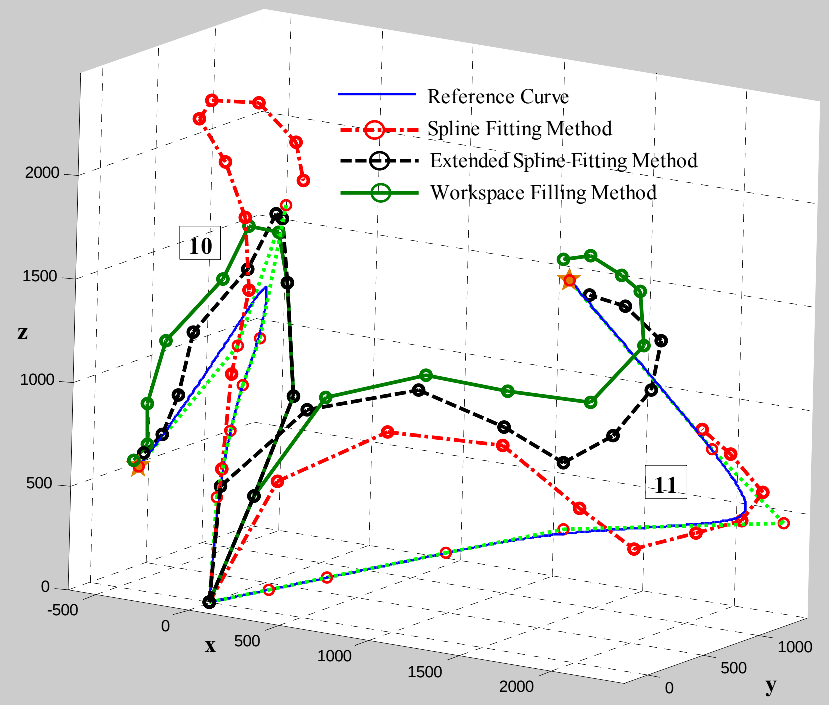


Figure 4.30 The Configurations for the Inconvenient Reference Curves

The evaluation of the solution configurations in the figures above is made with respect to the given reference curves, which are NURBS's. In Figure 4.26, the reference curves are lines on the y-z plane. Here, the configurations found with the spline fitting method and the extended spline fitting method can be fitted to the given reference curves in a better way whereas the configurations found with the workspace filling method display a little bit deviation from the reference curves, but, these amounts are small and in the acceptable limits of error. Figure 4.27 includes two curves on the x-z plane. The configurations found with all three methods can approach the given curves. However, the spline fitting method shows a better performance for this type of configurations for the shape of the configurations. In Figures 4.28 and 4.29, the tasks include spatial curves. Note that, the spline fitting method can fit the solution configurations to the initial parts of reference curves in a better way, but, the fitting procedure can not be implemented towards the last parts of the reference curves. These can cause the larger positional errors and inconvenient configuration shapes. For the workspace filling method, the configurations could excessively deviate from the reference curves although it could reach the desired target position sufficiently. In the spatial space tasks, it is apparently seen that the most successful results belong to the extended spline fitting method. Figure 4.30 is the most interesting of these examples. In this figure, the reference curves, which are inconvenient for the motion capability of the manipulator, are assigned intentionally to investigate the solutions with these methods. Note that none of them is not fitted to the given reference curves properly; even if the configurations by the spline fitting method deviates from the reference curves considerably and have larger positional errors. In fact, the workspace filling method and the extended spline fitting method could sufficiently approach the desired target positions, and although they deviate from the reference curves excessively, they try to resemble the reference curve as shown in the figure.

Another comparison subject is related to the positional error. The positional errors for 11 examples are given in Table 4.2 for three methods. Here, the positional error value is found by dividing the distance between the desired target position and the top position of the configuration by the average length of the HRM. It is expected that this error is less than 3.25% with respect to the workspace of the fine tuning manipulator (FTM) for the limits of $d^*_{\min} = 20mm$ and $d^*_{\max} = 140mm$.

Table 4.2 Positional Errors with the Three Inverse Kinematics Methods

The Configuration Number	The Spline Fitting Method (Error %)	The Extended Spline Fitting Method (Error %)	The Workspace Filling Method (Error %)
1	2.0	0.2	2.3
2	7.0	0.3	2.4
3	7.2	7.0	14.5
4	5.9	0.15	1.5
5	5.0	1.1	2.6
6	4.8	2.2	2.5
7	4.9	1.7	2.3
8	10.5	2.6	2.3
9	22.8	1.2	2.7
10	43.0	1.7	2.5
11	25.0	2.7	2.4

As seen on the above table, the solutions by the spline fitting method have the larger positional errors, even; these errors can exceed the limits which the continuously actuated manipulator can compensate. On the other hand, the extended spline fitting method and the workspace filling method ensure the condition of the positional error. According to the positional error, the extended spline fitting method gives better results in these three methods.

The trajectory following is the other comparison criterion. There is no control on the trajectory following with the spline fitting method for the specified reference curve since only a single configuration is obtained as a solution. However, the trajectory following is taken into consideration for the solutions with the extended spline fitting method and the workspace filling method because a set of configurations rather than a single configuration can be obtained by using these methods. In spite of this, as expressed earlier, the trajectory following cannot be implemented by the HRM because when the manipulator moves between two successive configurations, the larger deviations, which the FTM cannot compensate, take place.

The advantage of using the spline fitting method over the other methods is that it has relatively shorter run time for the inverse kinematics. In other methods, this time is much larger. In fact, there is not a certain run time for the extended spline fitting method and the workspace filling method since they have different number of configurations for each selection operation.

4.7 Summary and Results

In this chapter, the inverse kinematics of the HRM has been investigated with three methods. In these methods, the minimum positional and orientational error, the convenient shape of the manipulator and the trajectory following between successive configurations are the criteria for the solutions.

The first method is the spline fitting method. In this method, the manipulator is fitted to the given reference curve which is Non-Uniform Rational Base Spline (NURBS) defining the shape of the manipulator. This curve has also been used in the other methods as a reference curve. The fitting operation is module based, i.e. each module is fitted to the spline parts as closely as possible. For the obstacle avoidance, cubic spline, which is simpler, has been used for intermediate configuration as a reference curve.

The second method is the extended spline fitting method. This method is not, in fact, a different approach. It uses the spline fitting method and the genetic algorithm. The solution with this method is obtained directly by using genetic algorithm. In the genetic algorithm, the solution configuration from the spline fitting method is used as an initial population. Thus, better configurations can be obtained.

In the third method, the workspace filling method has been presented for the inverse kinematics. This method is based on the workspace of the manipulator. The workspace is filled with the certain number of configurations randomly selected from the whole workspace. Besides, some concentrated working regions according to the tasks are filled with many possible configurations by using the genetic algorithm. The selection of the configuration from the workspace has been implemented with respect to the given four criteria expressed above.

The figures above show that the manipulator can fit the spline as closely as possible for many times. The method is investigated for one module-based and two module-based fitting. Instead of one-module based fitting, two-module based fitting can decrease the errors, and the manipulator can approach the reference shape. However, in this case, the time for the library construction and the searching increases considerably because the number of calculation increases from O(n…) to O(n…^{2}). For example, the time constructing the library for two-module-based approach takes about 30 minutes whereas one-module based takes 0.6 seconds. Besides, there is also not any control on the position of the intermediate modules. The applications are run with the algorithm written in MATLAB in a computer which has an AMD ATLON 2000, 1.6 GHz CPU, 704 MB of RAM.

The success of the manipulator to fit the spline depends on the cost function in this algorithm. According to the figures, using the distance minimization alone in the cost function is more effective than using it with the other parameters. This is obvious because the optimization of the cost function with many parameters does not give an efficient solution. In addition, the control over the system parameters can not be supplied because of the discrete nature of the binary actuators, and the motion capability of the manipulator is restricted with its physical structure, e.g., it can reach finite discrete points and each module has approximately an orientation of the maximum angle of 30^0 in each direction. In this method, the positional errors are, in some cases, rather large to be eliminated by means of the FTM. In addition, the manipulator can not fit the reference curve appropriately. There are some reasons for this situation. The fitting process takes place from the bottom module to the top one.

The errors accumulate in each operation. The small errors for the modules towards the base cause the larger errors for the upper modules. Although each fitting operation tries to eliminate these errors, because of the nature of the binary actuation, the manipulator can deviate from the reference curve in an undesirable way. In addition to this, there are some reasons that can cause the failure of this method. The spline as a reference curve for the HRM is not divided optimally with respect to each related module. With this method, we can not have over the control of the trajectory following between successive configurations because the trajectory following is not taken into consideration.

The extended spline fitting method shows relatively better performance for the inverse kinematics. This method has minimum positional errors in these three methods. It obtains better results for the shape of the configurations because it uses the advantage of the spline fitting method and the genetic algorithm. From the Figures 4.26-4.30, the method can be fitted to the reference curve as closely as sufficient. Also, the trajectory following can be controlled by using this method because of the set of solution configurations.

In the workspace filling method, the optimum configurations can be reached to local optimum since the selection is made from the certain number of configurations fulfilling the error tolerance. In addition, this method permits us to select the configurations providing the better trajectory between successive configurations. However, it takes too much time to construct the workspaces as shown in Table 4.1, and the shape of the configurations selected sometimes can not be convenient for the given reference curve. The workspace is constructed by using two methods. The first one fills the workspace randomly with the configurations of the HRM. The second method constructs the working regions by using the genetic algorithm. The first approach generates the more different types of the configurations whereas the second one may consist of the similar configurations since the genetic algorithm forces to the configurations with the smallest fitting values.

The better running time is obtained in the spline fitting method. For example, the solution of the inverse kinematics with this method takes approximately 0.6 s for the module library and 0.2 s for the selection of configuration. Compared to the other methods, this time is much smaller. As seen from Table 4.1, the construction of the workspace and searching the possible configurations in the workspace filling method can take longer time. Note that the extended spline fitting method takes almost as much as the same time as in the workspace filling method.

CHAPTER 5

DYNAMIC ANALYSIS OF THE HYPER REDUNDANT MANIPULATOR

5. 1 Introduction

In this thesis, the dynamic analysis is carried out in order to get information about the behavior of the HRM under the effect of the actuator forces. For this aim, the accelerations are first determined by using the laws of motion. These accelerations are then integrated to determine the velocities and the positions of the links of the HRM.

The equations of motions of the hyper redundant manipulator consisting of rigid bodies are developed in this chapter in terms of reference coordinates that represent the translation of the reference points of the manipulator as well as their orientations with respect to the inertial reference frames. This representation will prove useful in developing general purpose computer algorithms for dynamic analysis of interconnected sets of rigid bodies, since, practically speaking, there is no limitation on the number of bodies or the types of forces and constraints that can be introduced to this formulation.

Until now, in the kinematics studies, it is assumed that all the actuators start and finish the motions simultaneously, and the characteristics of their motions are the same. However, the actual motion of the manipulator is different from this assumption. In a manipulator, after starting the motion, the motion of the actuators finish at different times and their motion characteristics can be considerably different from each other.

The dynamics of hyper redundant manipulators was first formulated by Chirikjian [53]. He used principles of continuum mechanics to approximately represent the dynamics of HRM, where the dynamics of the continuum is first formulated and then projected on to the actual physical structure. This modeling technique is only an approximation. In his other study, Chirikjian proposed an approximation for the binary pneumatic cylinders used in HRMs [40]. If the mass of the plunger of a pneumatic cylinder is *m*, then in absence of gravity the equation of motion for the cylinder is simply $m\frac{d^2x}{dt^2} = PA - c\frac{dx}{dt}$, where *P* is the pressure in the tube, *A* is the area of the plunger head, *c* is the viscous friction coefficient, and *x* is the amount of plunger that is displaced from a datum. In this approximation, it is assumed that an end point of the actuator is connected to a fixed point. For convenience, the variable $F = PA$, which is the force that the actuator is able to provide, will be used. More information can be obtained from this simple method. For instance, if the datum is chosen such that $x(0) = 0$, and viscosity and pressure are very high relative to the mass, then $m/c \to 0$ and the approximation for the displacement is obtained simply as $x(t) \approx (F/c)t$. An elementary analysis of the more complicated case of a whole manipulator composed of highly damped pneumatic actuators yields essentially the same result. That is, the manipulator dynamical equations are

$$\left[M(\overline{q})\right]\ddot{\overline{q}} + \left[C(\overline{q},\dot{\overline{q}})\right] + \left[G(\overline{q})\right] = \overline{f} - \left[\Delta\right]\dot{\overline{q}}$$

where $\overline{f}$ is the vector due to pressure, and $\left[\Delta\right]$ is the diagonal viscosity matrix. Under the assumption of slow motion, and very high pressure and viscosity, the left hand side of the equation of the manipulator dynamics can be essentially neglected. In this case, the linear behavior is simply obtained as

$$\overline{q}(t) = \overline{q}_0(t) + t.\left[\Delta\right]^{-1}\overline{f}$$

However, this result is a crude approximation and it involves many simplifying assumptions, such as neglecting the effect of gravity, etc. In this thesis, instead of using this approximate result, a realistic model is constructed to investigate the dynamic behavior of the hyper redundant manipulator.

In this chapter, the dynamic analysis of the HRM is made in a module-based manner. The equations of motion are written for each rigid body of the modules. The equations are arranged in a matrix form from the module-based equations written recursively. The unknowns are the piston accelerations of the actuators and the components of the structural reaction forces and moments at the joints. However, the system happens to be overconstrained and therefore the number of the mentioned equations turns out to be less than the number of the unknowns, majority of which are the reaction force and moment components. Nevertheless, after some virtual structural simplifications, these system equations become solvable, i.e. the number of equations conforms to the number of the unknowns with the justly reduced number of unknown reaction force and moment components.

5.2 The Virtual Elimination of the Excess Constraints for the Dynamic Analysis of the Hyper Redundant Manipulator

With the dynamic analysis, the independent joint variables, which are the lengths of the actuators, and the reaction force and moment components between the rigid members of the manipulator are determined. However, the real aim is to find the independent joint variables. Although the system is binary actuated, for the two successive configurations of the HRM, the actuators moves continuously between the two states under the action of the pressure forces, which are specified depending on the on or off states of the actuators. To solve an equation system, the number of equations should be equal to the number of unknowns, and also the equations should be independent to get a unique solution. In this study, due to the modules with the same characteristics, the equations obtained for any module of the HRM are similar for the other modules too. Because of this, all the virtual simplifications and derivations are carried out for a typical module of the HRM as shown in Figure 5.1, and then, they are adapted to the other modules recursively to construct the equations of motion of the whole HRM.

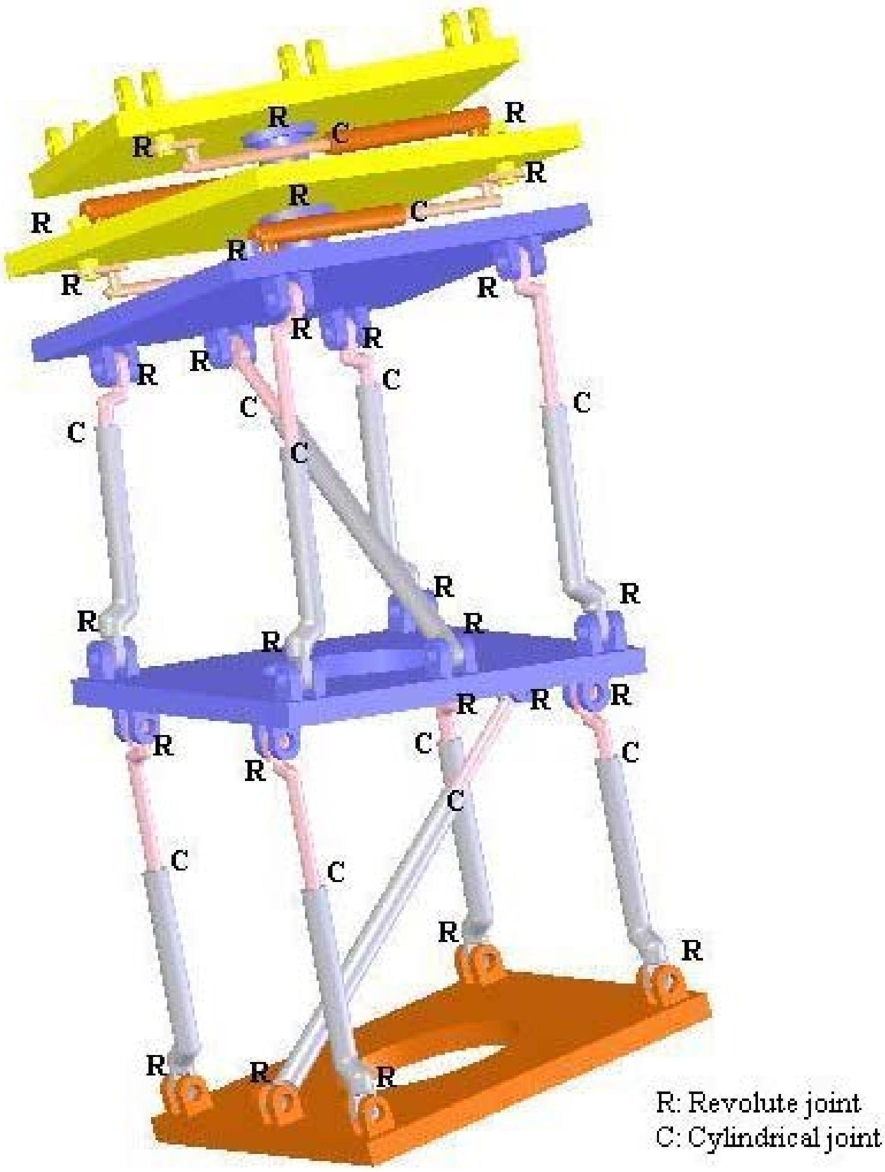


Figure 5.1 The Module Joints of the Manipulator

The following formula shows the relation between the number of equations and the number of unknowns for any mechanism.

$$\lambda(n-1) = N + F \tag{5.1}$$

where

$\lambda(n-1)$ is the number of equations for n links including the base,

$\lambda = 6$ for spatial mechanisms,

N is the number of unknown reaction force and moment components,

$N = 5 \times N_R + 4 \times N_C + 4 \times N_U$

N_R, N_C and N_U are the number of revolute, cylindrical and universal joints respectively,

and

F is the degree of freedom (DOF) for the module, which is equal to the unknown accelerations of the independent joint variables associated with that module.

For the module as shown in Figure 5.1, the actuators are connected to the platforms with revolute joints, and the piston-cylinder joints for the actuators are cylindrical. The module involves 33 links, 30 revolute (R) joints and 14 cylindrical (C) joints. The revolute joints allow only one relative rotational motion between two bodies and, thus, they impose 5 constraints, which lead to 3 reaction force and 2 reaction moment components. As for the cylindrical joints, they allow only one rotational and only one translational relative motion and, thus, they impose 4 constraints, which lead to 2 reaction force and 2 reaction moment components.

If the dynamics of the system is expressed by means of the Newton-Euler equations, then, many reaction force and moment components appear as unknowns in the equations. For each module with $N_R = 30$ revolute joints and $N_C = 14$ cylindrical joints, the number of these unknowns is $N = 5(30) + 4(14) = 206$. On the other hand, the degrees of freedom of each module is $F = 7$. However, there are 8 joint variables; 6 for the first and second VGT submodules and 2 for the third submodule. The two joint variables (ψ_1, ψ_2) for the third submodule define, in fact, only one rotational motion (ψ). Therefore, one additional kinematics constraint ($\psi = \psi_1 + \psi_2$)

is added to the system. So, for each module, the number of unknowns is $206+7=213$ whereas the number of equations is $\lambda(n-1)=6(33-1)=192$. The number of equations is not equal to the number of unknowns. This shows that the system is over-constrained. In other words, a unique solution can not be obtained. As a result, $213-192=21$ unknown reaction force and moment components should be treated as if absent for the module shown in the Figure 5.1 by virtually simplifying its structure. Of course, this must be done without violating the rigidity assumption of the links. For this aim, the joint types of the module are virtually changed to obtain a solvable system without changing the motion characteristic of the module. Here, such a virtual simplification can be made since we are not actually interested in the reaction force and moment components. As a result of this simplification, the unknown reaction force and moment components arise according to the new configuration.

For the simplification, there are many alternatives with respect to the joint configurations. That is, the simplification for the module is not unique. So, the alternative shown in Figure 5.2 is selected. Thus, the extra links causing the system to be over-constrained are removed from the model. To repeat, these simplifications are considered only for solving the equations of motion of the HRM and do not involve any structural modification on the original model. These simplifications do not change the form of the motions of the first and second submodules but they convert the last submodule into a 3-position actuator with the displacement set $\{-\psi^*,0,+\psi^*\}$.

According to the new considered model, the module shown in Figure 5.2 involves 18 links, 7 revolute joints, 12 cylindrical joints and 3 universal joints. Note that a universal joint allows only two relative rotational motions between the connected bodies and it imposes 4 constraints that create 3 reaction force components and 1 reaction moment component.

The number of unknown reaction force and moment components is $7\times5+12\times4+3\times4=95$ ($N_R=7$, $N_C=12$ and $N_U=3$) and the degrees of freedom is 7, i.e. there are 7 independent joint variables. As a result, there are totally $95+7=102$ unknowns while there are $(18-1)\times6=102$ equations. Thus, the number of unknowns becomes equal to the number of equations, and the system turns into a solvable system.

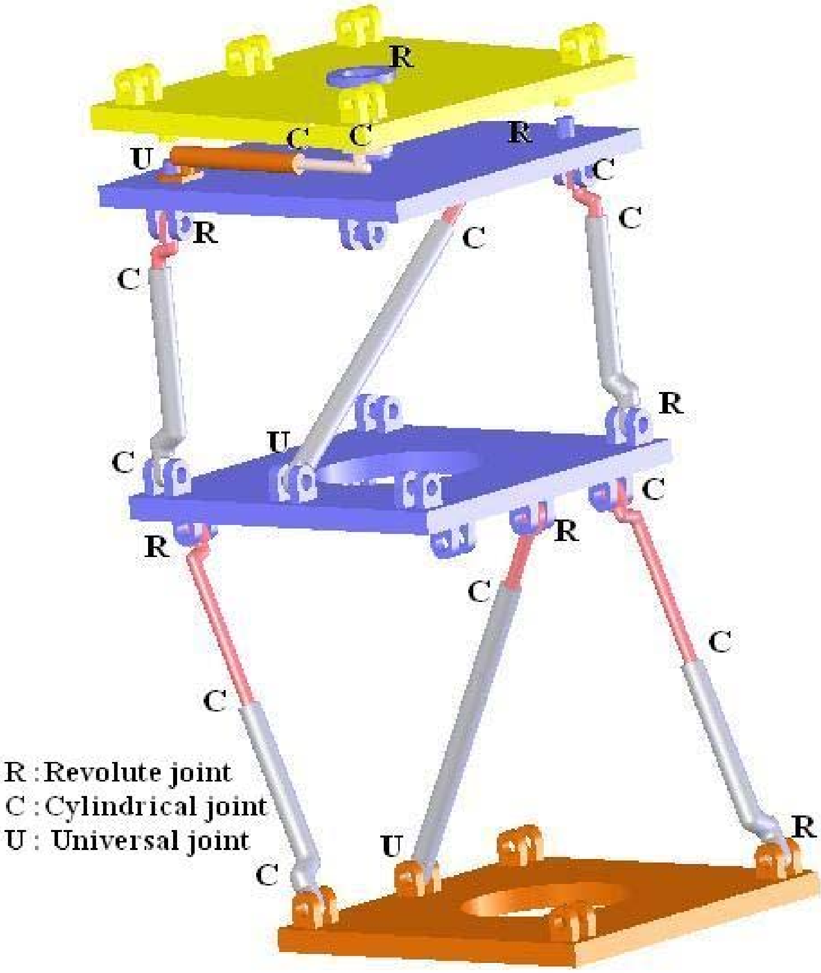


Figure 5.2 The Assumption for the Joint Types of the Module

5.3 The Dynamic Analysis

The most efficient methods for representing manipulator dynamics in the literature require serial computations proportional to degrees of freedom. The manipulator dynamics problem is generally formulated by using a technique based on Lagrangian mechanics, iterative Newton-Euler formulation or Newton-Euler method. Lagrangian mechanics results in equations of motion that can be written compactly as

$$\hat{M}(\overline{q})\ddot{\overline{q}} + \overline{C}(\overline{q},\dot{\overline{q}}) + \overline{G}(\overline{q}) = \overline{F} \tag{5.2}$$

Here, $\overline{q}$ is the vector of the generalized coordinates, i.e. the actuator lengths s_{11}, s_{12}, s_{13}, etc, $\hat{M}(\overline{q})$ is the mass matrix, $\overline{C}(\overline{q},\dot{\overline{q}})$ and $\overline{G}(\overline{q})$ are vectors containing the velocity and gravity dependent terms, and $\overline{F}$ is a vector containing the actuating forces.

This manipulator is actuated with on-off pneumatic actuators which are either fully extended or fully contracted in the initial and terminal positions of the HRM. While the HRM is in motion between the initial and terminal positions, some of the actuators move from the extended state to the contracted state, some of them move in the opposite sense, and the rest of them remain unchanged. The actuator motions are obtained by applying pneumatic forces in the given direction. The motion of an actuator from one state to the other one, whenever commanded, is of course possible if the applied pneumatic force is sufficiently large. Whether such motions of the actuators can come true or not, can be tested by means of the dynamic analysis of the manipulator. If the required motion of any of the actuators does not come true, then the manipulator fails performing the desired task.

To analyze the dynamic ability of the system, at first, the dynamic equations for each module are established by considering the free body diagram of each link as shown in Figures 5.3 and 5.4. By applying the Newton-Euler Method, the equations are written in matrix form separately for the 17 rigid links belonging to that module with the presence of the given forces of the actuators.

The Newton-Euler equations are written as follows.
The force equations are

$$\sum \bar{F}_i = m_i \bar{a}_i \tag{5.3}$$

The moment equations are

$$\sum \bar{M}_i = \hat{J}_i \bar{\alpha}_i + \tilde{\omega}_i \hat{J}_i \bar{\omega}_i \tag{5.4}$$

In these equations, $\bar{a}_i$, $\bar{\alpha}_i$ and $\bar{\omega}_i$ are the linear acceleration, angular acceleration and angular velocity of each link respectively. They are generated recursively as shown below.

$$\vec{\omega}_{i/0} = \vec{\omega}_{k/0} + \vec{\omega}_{i/k} \tag{5.5}$$

$$\vec{v}_{i/0} = \vec{v}_{k/0} + \vec{v}_{i/k} + \vec{\omega}_{k/0} \times \vec{r}_{i/k} \tag{5.6}$$

$$\vec{\alpha}_{i/0} = \vec{\alpha}_{k/0} + \vec{\alpha}_{i/k} + \vec{\omega}_{k/0} \times \vec{\omega}_{i/k} \tag{5.7}$$

$$\vec{a}_{i/0} = \vec{a}_{k/0} + \vec{a}_{i/k} + 2\vec{\omega}_{k/0} \times \vec{v}_{i/k} + \vec{\omega}_{k/0} \times \left(\vec{\omega}_{k/0} \times \vec{r}_{i/k}\right) + \vec{\alpha}_{k/0} \times \vec{r}_{i/k} \tag{5.8}$$

Here, $\vec{\omega}_{i/j}$ and $\vec{\alpha}_{i/j}$ are the angular velocity and acceleration of the body i relative to the reference frame j, which is fixed to body j. $\vec{r}_{i/j}$, $\vec{v}_{i/j}$ and $\vec{a}_{i/j}$ are the position, velocity and acceleration of the center of gravity of the body i relative to the reference frame j.

The Newton-Euler equations for body 2 are written as shown below first as vector equations and then as matrix equations expressed in the first reference frame attached to the first link.

$$\vec{F}_{12} + \vec{F}_{32} + m_2 g \vec{u}_3^{(0)} + (F_s - F)\vec{u}_3^{(2)} = m_2 \vec{a}_2$$

$$\overline{F}_{12}^{(1)} + \overline{F}_{32}^{(1)} + m_2 g \hat{C}^{(1,0)} \overline{u}_3 \;\; + (c\dot{s}_1 - F)\hat{C}^{(1,2)} \overline{u}_3{}^{(2)} \;\; = m_2 \overline{a}_2^{(1)}$$

$$\vec{M}_{12} + \vec{M}_{32} + \vec{\ell}_{G2} \times \vec{F}_{12} + \vec{r}_{G2} \times \vec{F}_{32} + d\,\vec{u}_1{}^{(2)} \times (F_s - F)\vec{u}_3{}^{(2)} = \overset{\vee}{J}_2 \cdot \vec{\alpha}_2 + \vec{\omega}_2 \times \overset{\vee}{J}_2 \cdot \vec{\omega}_2$$
$$\overline{M}_{12}^{(1)} + \overline{M}_{32}^{(1)} + \tilde{\ell}_{G2}^{(1)} \overline{F}_{12}^{(1)} + \tilde{r}_{G2}^{(1)} \overline{F}_{32}^{(1)} - d\,(c\dot{s}_1 - F)\hat{C}^{(1,2)} \overline{u}_2{}^{(2)} = \hat{J}_2{}^{(1)} \overline{\alpha}_2^{(1)} + \tilde{\omega}_2^{(1)} \hat{J}_2{}^{(1)} \overline{\omega}_2^{(1)}$$

Now, the Newton-Euler equations can be written for body 3 also as above.

$$\vec{F}_{23} + \vec{F}_{12-3} + m_3 g \vec{u}_3^{(0)} + (F - F_s)\vec{u}_3{}^{(2)} \;\; = m_3 \vec{a}_3$$

$$\overline{F}_{12-3}^{(1)} - \overline{F}_{32}^{(1)} + m_3 g \hat{C}^{(1,0)} \overline{u}_3 \;\; + (F - c\dot{s}_1)\hat{C}^{(1,2)} \overline{u}_3{}^{(2)} \;\; = m_3 \overline{a}_3^{(1)}$$

$$\vec{M}_{23} + \vec{M}_{12-3} + \vec{\ell}_{G3} \times \vec{F}_{12-3} + \vec{r}_{G3} \times \vec{F}_{23} + d\,\vec{u}_1{}^{(2)} \times (F - F_s)\vec{u}_3{}^{(2)}$$
$$= \overset{\vee}{J}_3 \cdot \vec{\alpha}_3 + \vec{\omega}_3 \times \overset{\vee}{J}_3 \cdot \vec{\omega}_3$$
$$\overline{M}_{12-3}^{(1)} - \overline{M}_{32}^{(1)} + \tilde{\ell}_{G3}^{(1)} \overline{F}_{12-3}^{(1)} - \tilde{r}_{G3}^{(1)} \overline{F}_{32}^{(1)} - d\,(c\dot{s}_1 - F)\hat{C}^{(1,2)} \overline{u}_2{}^{(2)}$$
$$= \hat{J}_3{}^{(1)} \overline{\alpha}_2^{(1)} + \tilde{\omega}_2^{(1)} \hat{J}_3{}^{(1)} \overline{\omega}_2^{(1)}$$

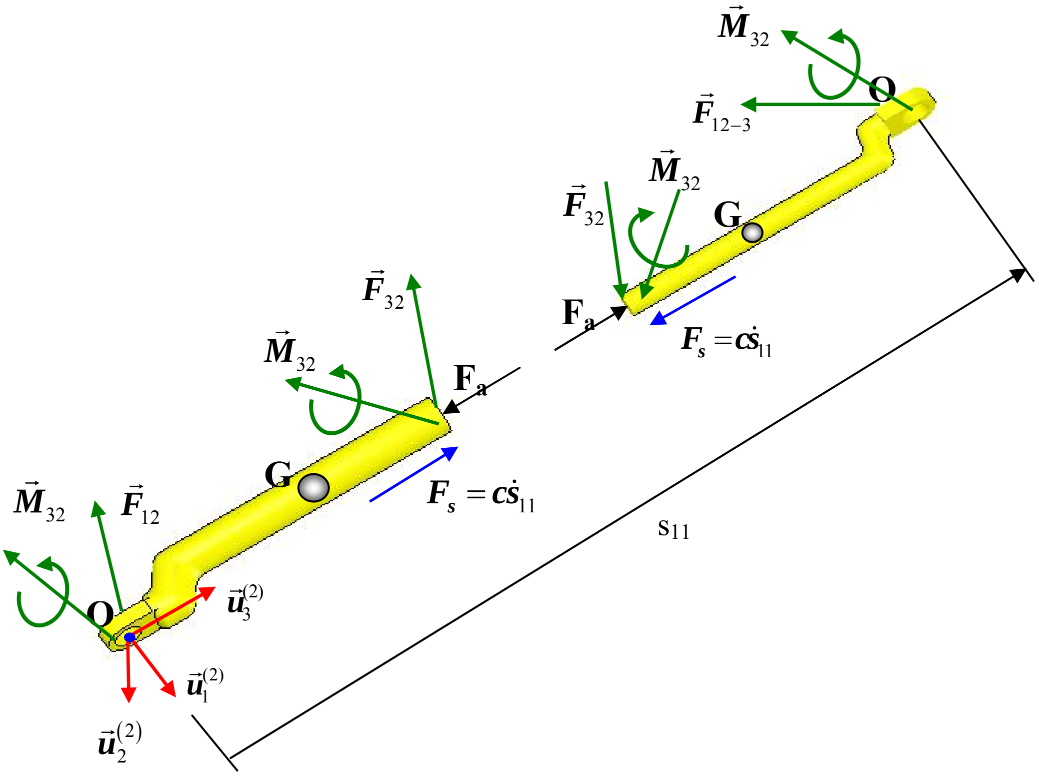


Figure 5.3. The Free Body Diagrams of the 2nd and 3rd Links

The vector and matrix equations for body 8 (the second platform) are written with respect to the reference frame attached to its own mass center as shown in Figure 5.4.

$$\vec{F}_{3-8}+\vec{F}_{5-8}+\vec{F}_{7-8}+\vec{F}_{9-8}+\vec{F}_{11-8}+\vec{F}_{13-8}+m_8 g\vec{u}_3^{(0)}=m_8\vec{a}_8$$

$$-\hat{C}^{(8,2)}\overline{F}_{8-3}^{(2)}-\hat{C}^{(8,4)}\overline{F}_{8-5}^{(4)}-\hat{C}^{(8,6)}\overline{F}_{8-7}^{(6)}+\overline{F}_{9-8}^{(8)}+\overline{F}_{11-8}^{(8)}+\overline{F}_{13-8}^{(8)}+m_8 g\hat{C}^{(8,0)}\overline{u}_3 \;=m_8\overline{a}_8^{(8)}$$

$$\vec{M}_{3-8}+\vec{M}_{5-8}+\vec{M}_{7-8}+\vec{M}_{9-8}+\vec{M}_{11-8}+\vec{M}_{13-8}+\vec{r}_{3-8}\times\vec{F}_{3-8}+\vec{r}_{5-8}\times\vec{F}_{5-8}+\vec{r}_{7-8}\times\vec{F}_{7-8}$$
$$+\vec{r}_{9-8}\times\vec{F}_{9-8}+\vec{r}_{11-8}\times\vec{F}_{11-8}+\vec{r}_{13-8}\times\vec{F}_{13-8}=\overset{\vee}{J}_8\cdot\vec{\alpha}_8+\vec{\omega}_8\times\overset{\vee}{J}_8\cdot\vec{\omega}_8$$

$$-\hat{C}^{(8,2)}\overline{M}_{8-3}^{(2)}-\hat{C}^{(8,4)}\overline{M}_{8-5}^{(4)}-\hat{C}^{(8,6)}\overline{M}_{8-7}^{(6)}+\overline{M}_{9-8}^{(8)}+\overline{M}_{11-8}^{(8)}+\overline{M}_{13-8}^{(8)}$$
$$-\tilde{r}_{3-8}^{(8)}\hat{C}^{(8,2)}\overline{F}_{8-3}^{(2)}-\tilde{r}_{5-8}^{(8)}\hat{C}^{(8,4)}\overline{F}_{8-5}^{(4)}-\tilde{r}_{7-8}^{(8)}\hat{C}^{(8,6)}\overline{F}_{8-7}^{(6)}+\tilde{r}_{9-8}^{(8)}\overline{F}_{9-8}^{(8)}+\tilde{r}_{11-8}^{(8)}\overline{F}_{11-8}^{(8)}$$
$$+\tilde{r}_{13-8}^{(8)}\overline{F}_{13-8}^{(8)}=\hat{J}_8^{(8)}\overline{\alpha}_8^{(8)}+\tilde{\omega}_8^{(8)}\hat{J}_8^{(8)}\overline{\omega}_8^{(8)}$$

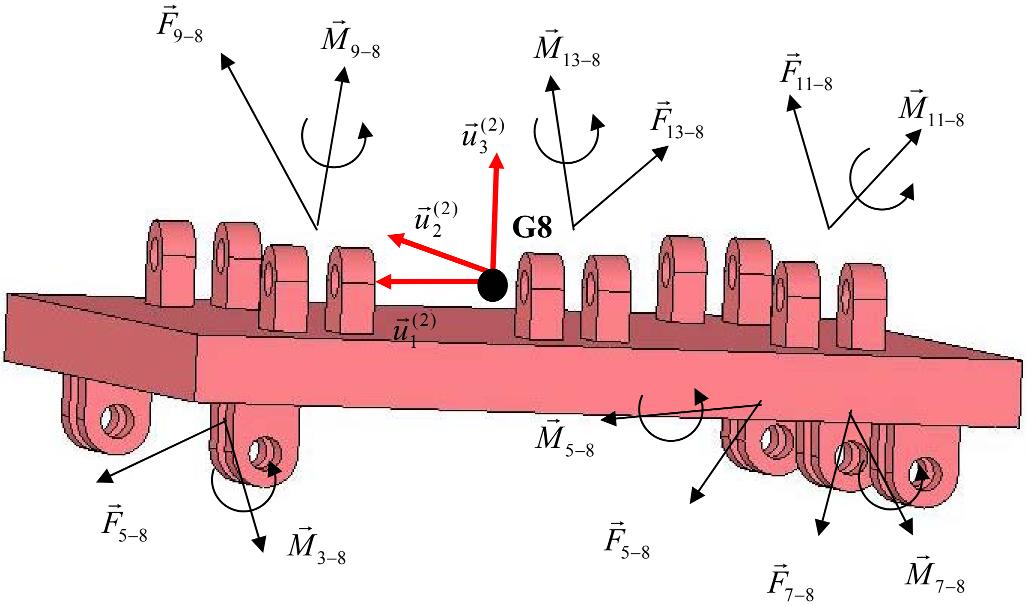


Figure 5.4 The Free Body Diagram of the Link 8

This procedure is applied until the 18th link, which is the last link of the model.

The vector and matrix equations for body 18 (the last link of the model) is written with respect to the 18th reference frame.

$$\vec{F}_{15-18} + \vec{F}_{17-18} + \vec{F}_{19-18} + \vec{F}_{21-18} + \vec{F}_{23-18} + m_{18} g \vec{u}_3^{(0)} = m_{18} \vec{a}_{18}$$

$$-\hat{C}^{(18,15)} \overline{F}_{18-15}^{(15)} - \hat{C}^{(18,16)} \overline{F}_{18-17}^{(16)} - \overline{F}_{19-18}^{(18)} - \overline{F}_{21-18}^{(18)} - \overline{F}_{23-13}^{(18)} + m_{18} g \hat{C}^{(18,0)} \overline{u}_3 \ = m_{18} \overline{a}_{18}^{(18)}$$

$$\vec{M}_{15-18} + \vec{M}_{17-18} + \vec{M}_{19-18} + \vec{M}_{21-18} + \vec{M}_{23-18} + \vec{r}_{15-18} \times \vec{F}_{15-18} + \vec{r}_{17-18} \times \vec{F}_{17-18} + \vec{r}_{19-18} \times \vec{F}_{19-18}$$
$$+ \vec{r}_{21-18} \times \vec{F}_{21-18} + \vec{r}_{23-18} \times \vec{F}_{23-18} = \check{J}_{12} \cdot \vec{\alpha}_{12} + \vec{\omega}_{12} \times \check{J}_{12} \cdot \vec{\omega}_{12}$$
$$-\hat{C}^{(18,15)} \overline{M}_{18-15}^{(15)} - \hat{C}^{(18,16)} \overline{M}_{18-17}^{(16)} - \overline{M}_{18-19}^{(18)} - \overline{M}_{18-21}^{(18)} - \overline{M}_{18-23}^{(18)} - \tilde{r}_{15-18}^{(18)} \hat{C}^{(18,15)} \overline{F}_{18-15}^{(15)}$$
$$-\tilde{r}_{17-18}^{(18)} \hat{C}^{(18,16)} \overline{F}_{18-17}^{(16)} - \tilde{r}_{19-18}^{(18)} \overline{F}_{18-19}^{(18)} - \tilde{r}_{21-18}^{(18)} \overline{F}_{18-21}^{(18)} - \tilde{r}_{23-18}^{(18)} \overline{F}_{18-23}^{(18)}$$
$$= \hat{J}_{18}{}^{(18)} \overline{\alpha}_{18}^{(18)} + \tilde{\omega}_{18}^{(18)} \hat{J}_{18}{}^{(18)} \overline{\omega}_{18}^{(18)}$$

where,

$\overline{\omega}_i = \overline{\omega}_{i/0}$ and $\overline{\alpha}_i = \overline{\alpha}_{i/0}$

$\vec{r}_{i-j}$: the position vector from the connection point between the body i and the body j to the center of gravity of the body j.

$\tilde{r}_{i-j}{}^{(k)}$: the tilde matrix of $\vec{r}_{i-j}$ with respect to the reference frame k.

$\vec{F}_{i-j}$, $\vec{M}_{i-j}$: the force/moment on the body j applied by the body i.

$\overline{F}_{i-j}{}^{(k)}$, $\overline{M}_{i-j}{}^{(k)}$: the matrix representation of $\vec{F}_{i-j} / \vec{M}_{i-j}$ with respect to the reference frame k.

$\hat{C}^{(i,j)}$: the component transformation matrix from the reference frame j to the reference frame i.

$\overset{\vee}{J}_k$ and $\hat{J}_k^{(i)}$: the inertia tensor of the body k and the corresponding inertia matrix of the body k written relative to the reference frame i.

These equations of motion as a reference are used to write the equations of motion of the HRM.

5.3.1 The Equations of Motion of the HRM

The equations of motion are at first written by applying the Newton-Euler equations to the links of the reference module as shown in Figure 5.2. These equations constitute a reference set of equations. By using this set, the equations of motions for all modules of the HRM can be obtained set by set according to their dimensions, applied forces, inertias etc. Taking the orientations and positions of the modules into account, all equation sets can be combined in matrix form as follows.

$$\hat{M}(\overline{q})\begin{bmatrix}\overline{a}_C \\ \overline{\alpha}\end{bmatrix}+\hat{B}(\overline{q})\overline{R}+\hat{D}(\overline{q},\dot{\overline{q}})+\hat{G}(\overline{q})=\overline{F} \tag{5.9}$$

where

$\overline{a}_c$: the acceleration vector of the link centers of gravity

$\overline{\alpha}$: the angular acceleration vector of the links

$\hat{M}(\overline{q})=\begin{bmatrix}[m_i] & [\hat{0}] \\ [\hat{0}] & [\hat{J}_i(\overline{q})]\end{bmatrix}$: the mass matrix,

$\hat{B}$: the reaction force and moment component matrix

$\overline{R}$: the unknown reaction force and moment component vector

$\overline{D}$: the dissipative force vector

$\overline{G}$: the gravity force vector

$\overline{F}$: the applied force vector

In Equation (5.9), the acceleration vector $\overline{a}_c$ and the angular acceleration vector $\overline{\alpha}$ involve the independent and dependent joint variables. These acceleration vectors can be written in terms of the independent joint variable accelerations and the velocity terms involving the dependent and independent joint velocity and position terms. That is,

$$\begin{bmatrix} \overline{a}_C \\ \overline{\alpha} \end{bmatrix} = \hat{H}\ddot{\overline{q}} + \overline{E} \tag{5.10}$$

where

$\overline{q}$ is the independent joint variable vector

$\overline{E}$ is the vector involving the velocity and position terms.

$\hat{H}(\overline{q})$ is the Jacobian matrix.

Substituting Equation (5.10) into Equation (5.9), Equation (5.9) can be written in terms of only independent joint variable accelerations as follows.

$$\hat{M}(\overline{q})\left(\hat{H}\ddot{\overline{q}} + \overline{E}\right) + \hat{B}(\overline{q})\overline{R} + \hat{D}(\overline{q},\dot{\overline{q}}) + \hat{G}(\overline{q}) = \overline{F}$$

This equation can be rewritten in a simpler form.

$$\hat{M}\hat{H}\ddot{\overline{q}} + \hat{B}\overline{R} = \overline{Q} \tag{5.11}$$

where,

$$\overline{Q} = \overline{F} - \hat{M}\overline{E} - \hat{D} - \hat{G}$$

This equation can also be written as,

$$\left[\hat{M}\hat{H} \vdots \hat{B}\right]\begin{bmatrix} \ddot{\overline{q}} \\ \overline{R} \end{bmatrix} = \overline{Q}$$

Solution of this equation for $\ddot{\overline{q}}$ and $\overline{R}$ is

$$\begin{bmatrix} \ddot{q} \\ \overline{R} \end{bmatrix} = \begin{bmatrix} \hat{M}\hat{H} \vdots \hat{B} \end{bmatrix}^{-1} \overline{Q}$$

The dimension of the matrix $\hat{L} = \begin{bmatrix} \hat{M}\hat{H} \vdots \hat{B} \end{bmatrix}$ for the ten-module manipulator considered here is 1020 by 1020. The inversion of such a huge matrix is very difficult and the solution time for the manipulator motion simulation takes a very long time because the inverted matrix is needed at each time step during the numerical integration of the motion equations. On the other hand, what cause the matrix $\hat{L}$ to grow in size are the unknown reaction force and moment components. For example, there are 950 unknown reaction force and moment components while there are only 70 independent joint variables. If the reaction force and moment components are eliminated, the dimension of the matrix to be inverted becomes much smaller compared to $\hat{L}$.

5.3.1.1 The Elimination of the Constraint Forces

The constraint forces (i.e. the reaction force and moment components) are eliminated from the dynamic equations by expressing these equations in terms of the independent joint variables.

To extract the independent joint variables from Equation (5.11), the equation is premultiplied side-by-side by the matrix $\hat{H}^T$. Then, Equation (5.11) becomes

$$\hat{H}^T \hat{M} \hat{H} \ddot{q} + \hat{H}^T \hat{B} \overline{R} = \hat{H}^T \overline{Q} \tag{5.12}$$

This operation is called “embedding technique” or “reduction of matrix dimension” in the literature [54], [55] and [56]. Here, it must be noted that

$$\hat{H}^T \hat{B} \overline{R} = \overline{0} \tag{5.13}$$

This equation is based on the "virtual work method" and it states the renown fact that the constraint forces do not do any work [54].

Owing to Equation (5.13), the system differential equation reduces to

$$\hat{H}^T \hat{M} \hat{H} \ddot{\bar{q}} = \hat{H}^T \bar{Q} \tag{5.14}$$

Here, $\hat{M}' = \hat{H}^T \hat{M} \hat{H}$ and $\bar{Q}' = \hat{H}^T \bar{Q}$ are the generalized mass matrix and the vector of generalized forces associated with the independent coordinates. Noting that $\hat{M}' = \hat{H}^T \hat{M} \hat{H}$ has a much smaller size compared to $\hat{L}$. With the embedding technique described above, the dimension of the system matrix to be inverted is reduced form 1020 by 1020 to 70 by 70. Although the dimension of the other matrices is larger, the operation on these matrices is arithmetic, summation and multiplication of two matrices. This is very easier rather than calculating the inversion of the matrix.

Equation (5.14) can be solved for the independent accelerations easily as follows.

$$\ddot{\bar{q}} = \left(\hat{H}^T \hat{M} \hat{H} \right)^{-1} \hat{H}^T \bar{Q} \tag{5.15}$$

The accelerations determined above can then be integrated forward in time in order to determine the independent coordinates and velocities.

Using the Euler's approach to solve the differential equations numerically with a time step h, the solution can be expressed as follows.

$$\bar{q}(t+h) = \bar{q}(t) + \bar{v}(t)h + \ddot{\bar{q}}(t)\frac{h^2}{2}$$

$$\bar{v}(t+h) = \bar{v}(t) + \ddot{\bar{q}}(t)h$$

Here, the components of $\bar{q}$ and $\bar{v}$ are the lengths and speeds of the actuators.

In the numerical solution of the equations of motion of the HRM, when the pistons of the actuators touch the ends of their cylinders, the joint variables of the actuators take specified values, which are the minimum or maximum actuator lengths. To handle such cases, saturation limits are employed in the computer program prepared to integrate the differential equations.

If one wants to find the constraint forces, then Equation (5.11) is rearranged to obtain them as follows.

$$\bar{R} = \left(\hat{B}^T \hat{B}\right)^{-1} \left\{\hat{B}^T \left(\bar{Q} - \hat{M}\hat{H}\ddot{\bar{q}}\right)\right\} \tag{5.16}$$

From Equation (5.15) and (5.16), all the unknowns of the HRM can be obtained for the given actuator forces between the initial and final configurations of the HRM.

5.3.2 Pneumatic Actuator Forces

In this study, on-off pneumatic cylinders have been used as actuators. Due to the compressibility of air, the presence of friction and the non-linearity of valves, pneumatic actuators exhibit highly non-linear characteristics. Modeling of pneumatic actuator systems is based on the standard orifice theory. In this modeling, the equations are derived for the compressible air, the ideal gas with one-dimensional adiabatic, isentropic flow. Also, it is assumed that the supply pressure is constant. The relations and information about pneumatic systems are given in [57] in detail.

According to Figure 5.5, the expressions with the given assumptions can be written as follows.

$$\frac{T_0}{T} = 1 + \frac{k-1}{2} M^2 \tag{5.17}$$

$$\frac{P_0}{P} = \left(1 + \frac{k-1}{2} M^2\right)^{\frac{k}{k-1}} \tag{5.18}$$

Mass flow rate passing through unit area is that

$$\frac{\omega}{A} = \sqrt{\frac{k}{R}} \frac{P_0}{\sqrt{T_0}} \frac{M}{\left(1 + \frac{k-1}{2} M^2\right)^{\frac{k+1}{2(k-1)}}} \tag{5.19}$$

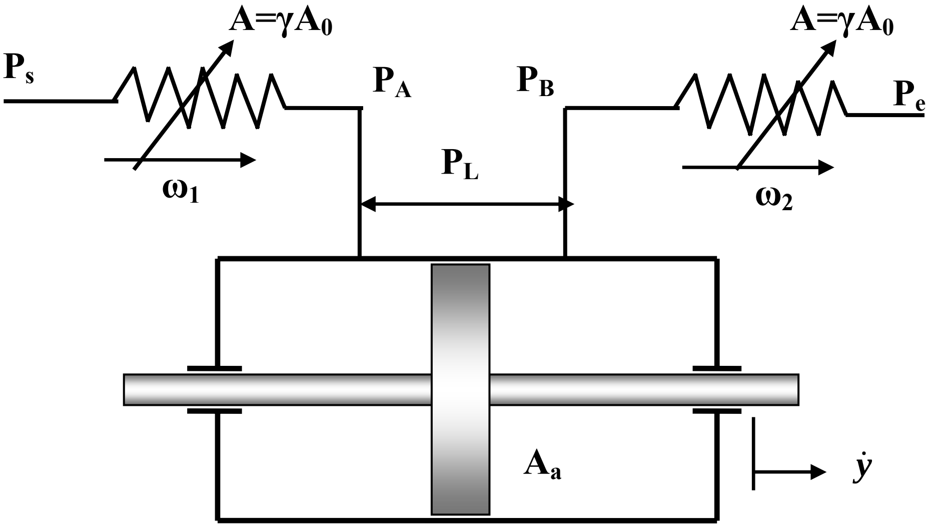


Figure 5.5 Pneumatic Serial Circuit

where,

ω : mass flow rate, *kg/s*

A : cross sectional area, m^2

k : specific heat ratio, c_p / c_v ($c_p / c_v = 1.4$ for air)

R : idea gas constant ($R = 287.0$, m^2/s^2-K for air)

T : absolute gas temperature, *K*

T_0 : state temperature, *K*

P : gas pressure, *Pa*

P_0 : state pressure, Pa

M : Mach number

Solving equation (5.18) for M, and substituting it into equation (5.19), the mass flow rate passing through unit area can also be written as,

$$\frac{\omega}{A} = \sqrt{\frac{2k}{R(k-1)}} \frac{P_0}{\sqrt{T_0}} \left(\frac{P}{P_0}\right)^{\frac{1}{k}} \sqrt{1 - \left(\frac{P}{P_0}\right)^{\frac{k-1}{k}}} \tag{5.20}$$

For the compressible flow,

$$\omega = C_d A \sqrt{\frac{2k}{R(k-1)}} \frac{P_0}{\sqrt{T_0}} \left(\frac{P}{P_0}\right)^{\frac{1}{k}} \sqrt{1 - \left(\frac{P}{P_0}\right)^{\frac{k-1}{k}}} \tag{5.21}$$

According to Figure 5.5, the piston velocity is given as

$$\dot{y} = \frac{\omega_1}{\rho_A A_a} \tag{5.22}$$

or

$$\dot{y} = \frac{\omega_2}{\rho_B A_a} \tag{5.23}$$

When Equation (5.21) is substituted into these two velocity equations and the ideal gas equation ($P = \rho RT$) is used, the working pressures P_A and P_B are obtained as follows in terms of the piston velocity.

For part A,

$$\dot{y} = \frac{RT}{P_A A_a} C_d \gamma A_0 \sqrt{\frac{2k}{R(k-1)}} \frac{P_S}{\sqrt{T_S}} \left(\frac{P_A}{P_S}\right)^{\frac{1}{k}} \sqrt{1 - \left(\frac{P_A}{P_S}\right)^{\frac{k-1}{k}}} \tag{5.24}$$

For part B,

$$\dot{y} = \frac{RT}{P_B A_a} C_d \gamma A_0 \sqrt{\frac{2k}{R(k-1)}} \frac{P_B}{\sqrt{T_B}} \left(\frac{P_e}{P_B}\right)^{\frac{1}{k}} \sqrt{1 - \left(\frac{P_e}{P_B}\right)^{\frac{k-1}{k}}} \quad (5.25)$$

The load pressure and the piston force depending on the velocity can be obtained from the following formulas respectively.

$P_L = P_A - P_B$ and $F = A_a P_L$

For the given specifications, the piston force versus piston velocity is shown in Figure 5.6 and 5.7. The system specification is as follows.

$T_A = T_B = T_S \simeq 20^0 C = 293K$ (environment temperature with small change)

$\gamma = 1$ (full opening of valve)

$A_a = 1250 mm^2 = 1.25x10^{-3} m^2$ (cross sectional area of the piston)

$P_S = 5.10^5 N/m^2$ (source pressure)

$P_e = 10^5 N/m^2$ (environment pressure)

$A_0 = 10^{-5} m^2$ (cross sectional area of the valve)

$C_d = 0.85$ (discharge coefficient for orifices)

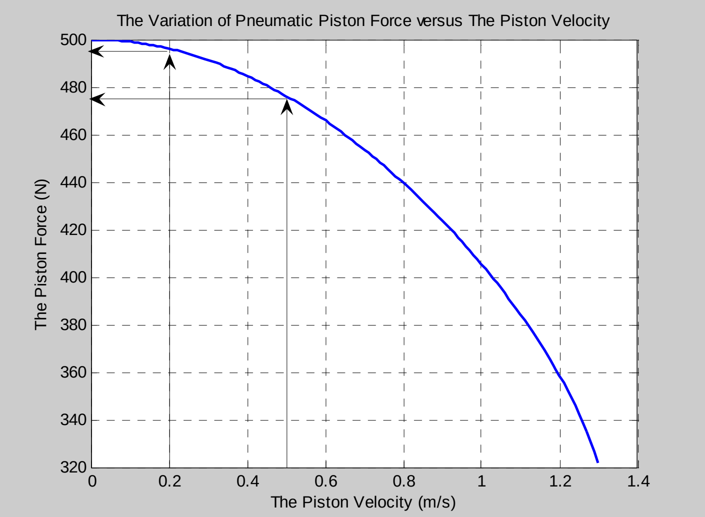


Figure 5.6 The Piston Force versus Piston Velocity

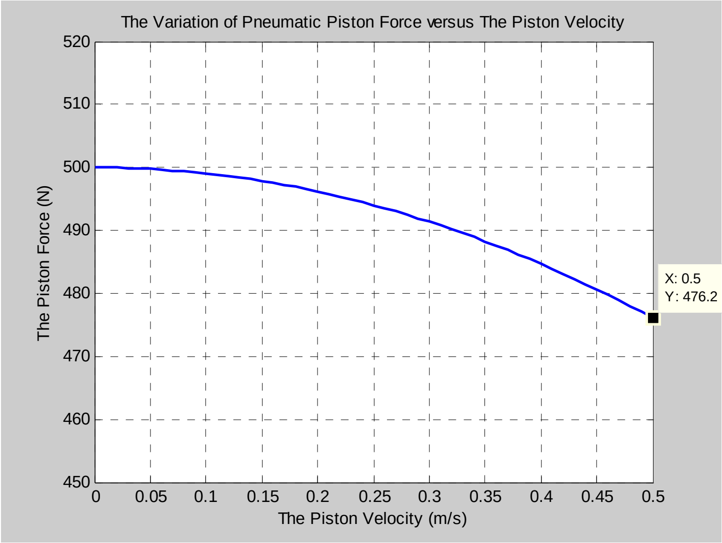


Figure 5.7 The Piston Force versus Piston Velocity for $v = 0 - 0.5m/s$

Due to the physical restrictions, actuators generally have a maximum velocity limit. In this study, this limit is assumed to be $v = 0.5m/s$ as a case study. Note that the relationship between the piston force of the pneumatic actuator and the piston velocity is almost linear for this velocity limit.

According to the figures, the push and pull forces of pistons can be taken as constant in the simulations since the change in piston force is negligibly small in the range of the piston speeds. For example, for the velocity limitation of $v = 0.5m/s$, taking a constant force value during the operations causes a maximum error of 4.8%.

5.3.3 The Algorithm for the Computer Program

In this dynamic problem, the independent joint variables which are the lengths of the actuators used in the HRM will be solved. This solution will be made numerically by using the algorithm for the computer program as follows.

- The masses, inertias and dimensions are taken from 3D solid drawing software module by module.

- From the inverse kinematics, the on-off information belonging to the actuators is taken and, the initial and final positions of the actuators and the required angles are calculated for each module.
- The numeric iterations come true step by step for each time value in the loop.
- In order to obtain the whole dynamic system equations, the required sub-matrices and the sub-vectors are obtained for each module. These are the acceleration matrix $\left[\hat{a}_i\right]$, the constraint force matrix involving the reaction forces and moment components $\left[\hat{R}_i\right]$, the mass matrix $\left[\hat{M}_i\right]$, the gravity vector $\left[\bar{G}_i\right]$, the dissipative force vector $\left[\bar{D}_i\right]$, the applied force vector $\left[\bar{F}_i\right]$, the inertial force vectors $\left[J_{M_i}\right]$, $\left[\hat{M}_i\bar{E}_i\right]$. They are written by using the dimensions, and the initial positional and velocity values of each module for $i = 1, 2, ...10$.
- After that, the upper matrices are obtained by combining these sub-matrices as follows.

$$\hat{a} = \begin{bmatrix} [\hat{a}_1] & & & [\hat{0}] \\ [\hat{a}_1^* \quad \hat{a}_2] & & & [\hat{0}] \\ [\hat{a}_1^* \quad \hat{a}_2^* \quad \hat{a}_3] & & & [\hat{0}] \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ [\hat{a}_1^* \quad \hat{a}_2^* .... & & & \hat{a}_{10}] \end{bmatrix} \qquad \hat{R} = \begin{bmatrix} [\hat{R}_1] & & & [\hat{0}] \\ & [\hat{R}_2] & & \\ & & \ddots & \\ [\hat{0}] & & & [\hat{R}_{10}] \end{bmatrix}$$

$$\hat{M} = \begin{bmatrix} [\hat{M}_1] & & & [\hat{0}] \\ & [\hat{M}_2] & & \\ & & \ddots & \\ [\hat{0}] & & & [\hat{M}_{10}] \end{bmatrix} \quad \bar{D} = \begin{bmatrix} \bar{D}_1 \\ \bar{D}_2 \\ \vdots \\ \bar{D}_{10} \end{bmatrix} \quad \bar{F} = \begin{bmatrix} \bar{F}_1 \\ \bar{F}_2 \\ \vdots \\ \bar{F}_{10} \end{bmatrix} \quad \bar{G} = \begin{bmatrix} \bar{G}_1 \\ \bar{G}_2 \\ \vdots \\ \bar{G}_{10} \end{bmatrix}$$

Since the last link of each module is the first link of the following module, the Newton-Euler equations for the last link are written out of the sub-matrix $\left[\hat{R}_i\right]$ and, it is added to the upper matrix $\left[\hat{R}\right]$.

- To construct system matrix, make the matrix calculations between the acceleration coefficient matrix and the mass matrix. By inverting the system matrix, calculate the independent accelerations of the actuated joint variables using the Euler approach. Then, the velocity and also the joint variables are calculated by integrating these accelerations.

- The values of the joint variables are controlled for the maximum and minimum length of the actuators. That is, when each joint variable reach these values, it is accepted that it finishes its motion unless there is any motion in the reverse direction. Each value is saved and, then, the algorithm passes to the next step.

- After calculating all the joint variables, using these values and Equation (5.16) from the whole dynamic equation set, the constraint forces are calculated.

According to this algorithm, the dynamics solution of the manipulator is given in the motion between the configurations shown in Figure 5.8. The plots belonging to the joint variables versus time for some modules as an example are depicted in Figures 5.9a-5.9d. In this simulation, the system specifications are given in Table 5.1. These values are approximately specified with respect to the mass, inertia and dimensions of each module.

Table 5.1 Applied Forces, Module Masses and Viscous Friction Coefficients

Module Number	Applied Force (N)	The Total Mass of Each Module (kg)	Viscous Friction Coefficient (Ns/m)
1	1200	11.5	$c = 500$ for the diagonal and off-diagonal actuators. $c = 1500$ for the actuators in the rotating submodule.
2	1000	9.6	
3	900	8.3	
4	800	7.2	
5	700	6.4	
6	600	5.4	
7	500	4.3	
8	240	3.5	
9	120	2.8	
10	60	2.2	

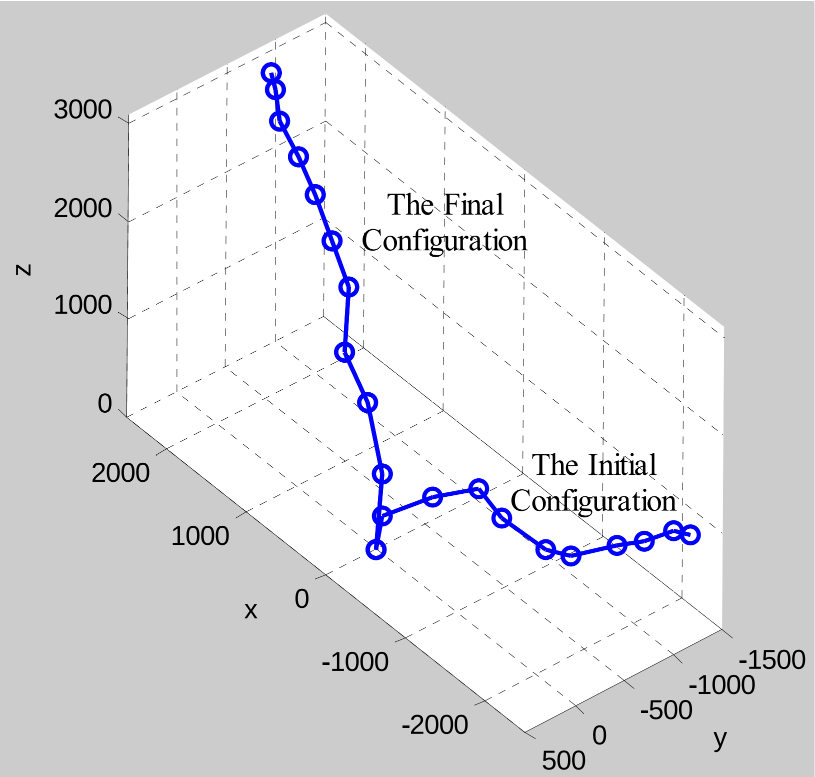


Figure 5.8 The Initial and Final Configurations for the Dynamic Analysis

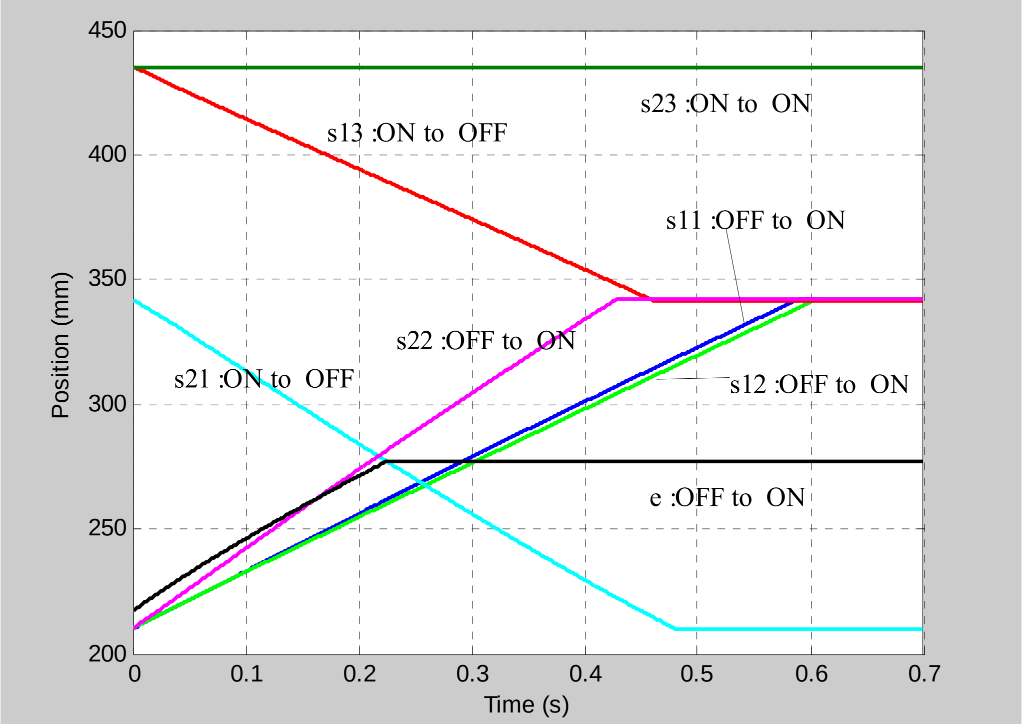


Figure 5.9a The Joint Variables for the Module 1

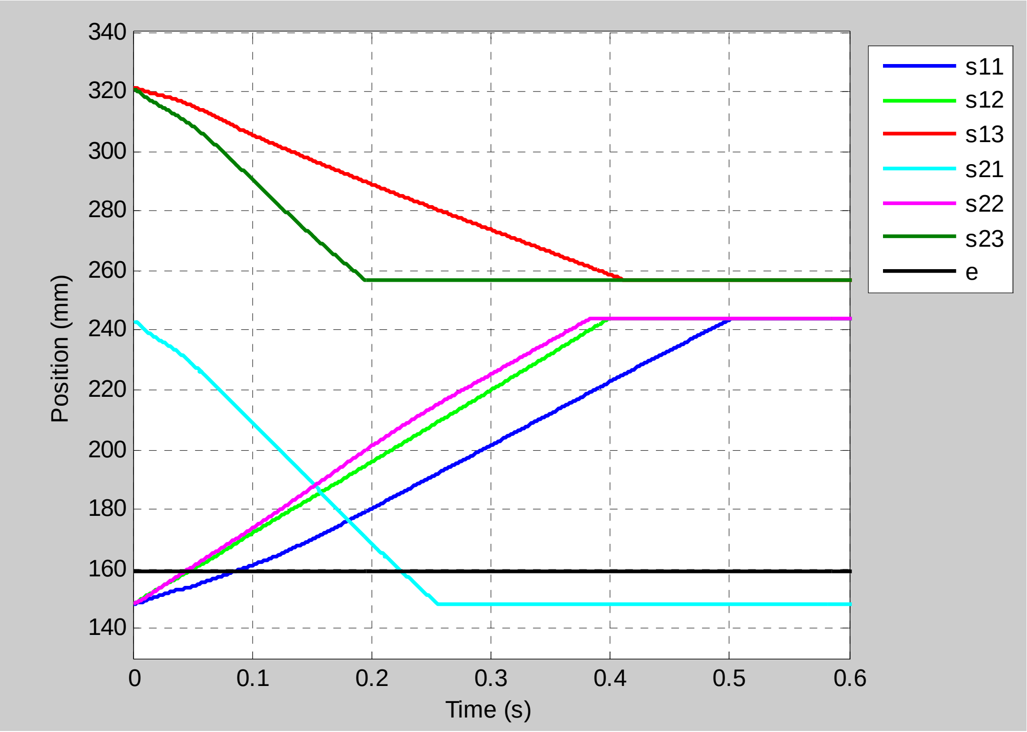


Figure 5.9b The Joint Variables for the Module 4

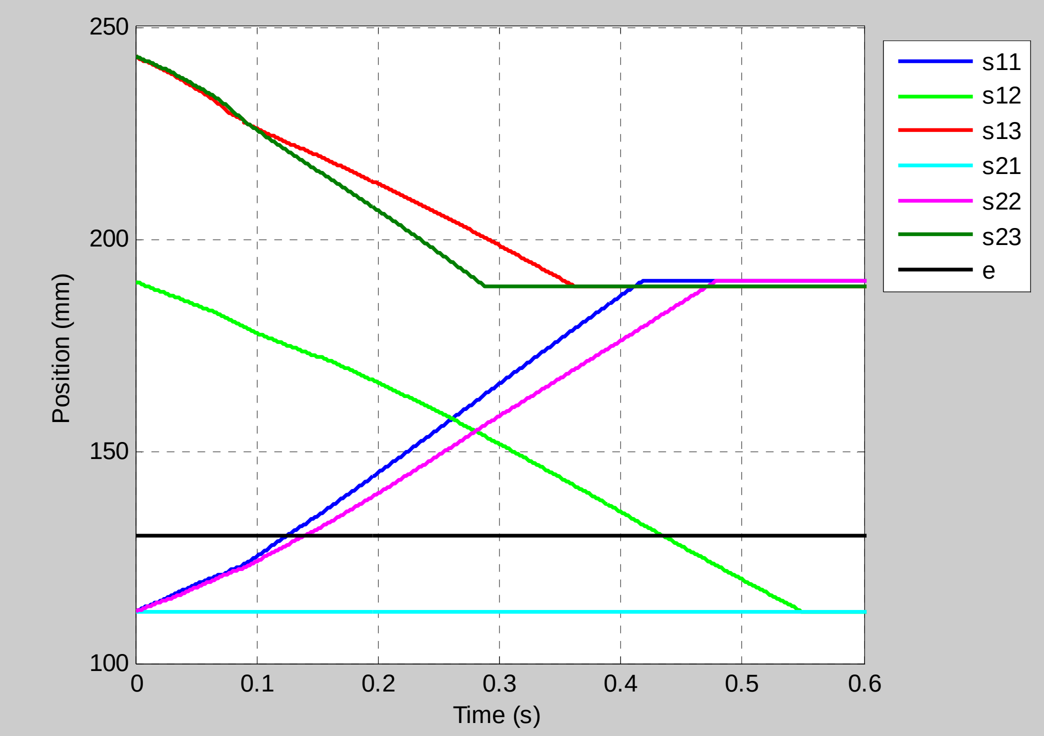


Figure 5.9c The Joint Variables for the Module 6

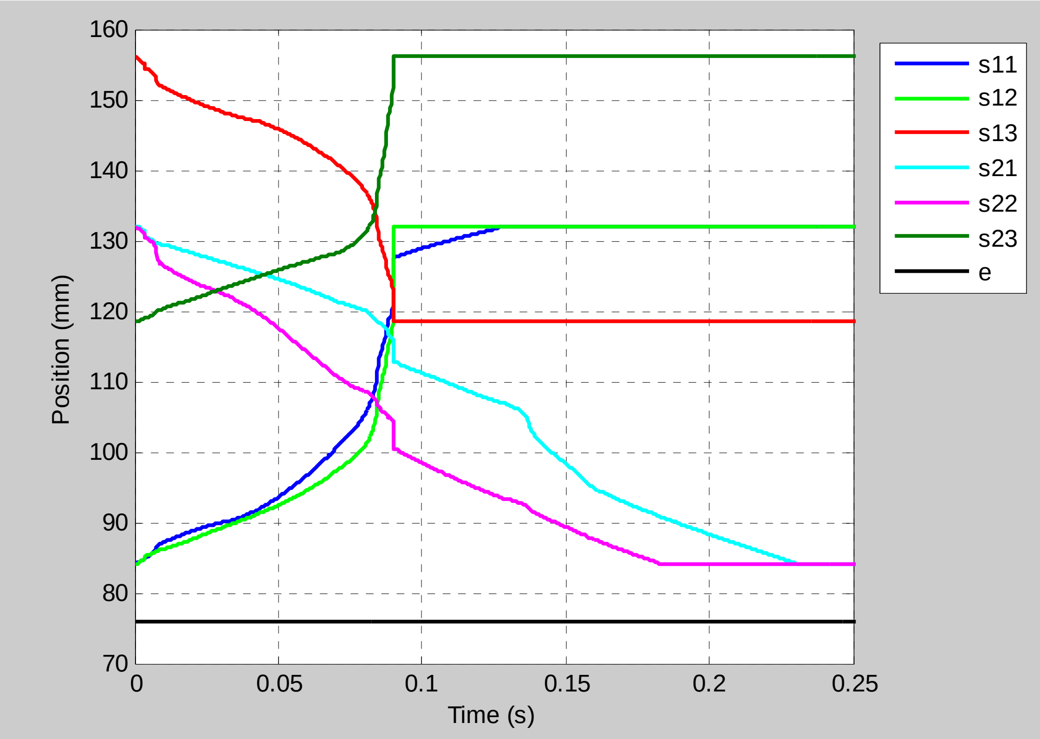


Figure 5.9d The Joint Variables for the Module 9

In some applications, some actuators can not reach the given stable states when many actuators of the HRM are in motion and especially the HRM is forced to take the configuration with both the fully extended and twisted motions. For example, for the initial and final configurations as shown in Figure 5.10, the actuators represented with the s_{12} and s_{22} joint variables for the module 1 in Figure 5.11a can not complete their motion. s_{12} and s_{22} have to reach the contracted stable state whereas they are in some intermediate states. Also, for the module 3 as shown in Figure 5.11b, s_{12} has to remain in the contracted stable state whereas it is again in an intermediate state.

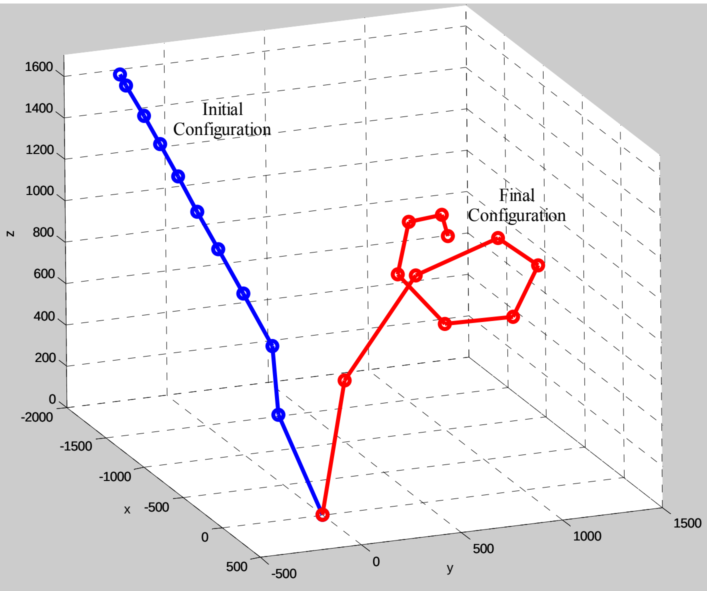


Figure 5.10 An Example of Non-Implemented Configuration Set

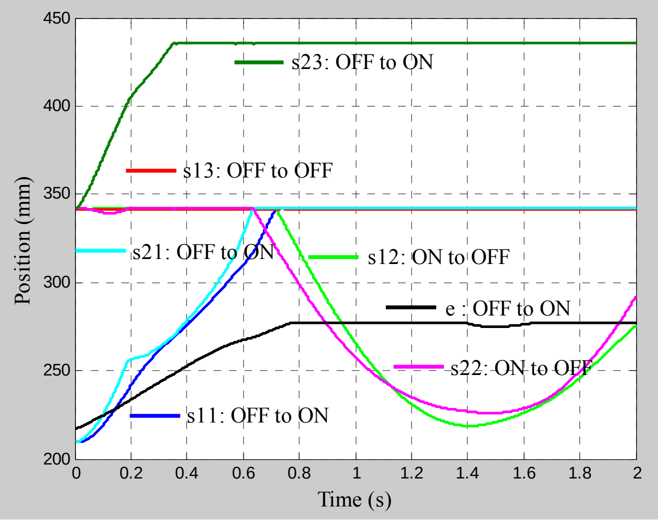


Figure 5.11a The Joint Variables for the Module1

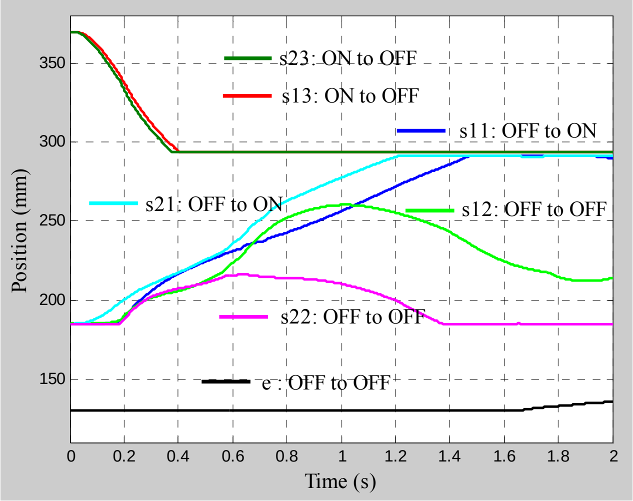


Figure 5.11b The Joint Variables for the Module 3

5.4 COSMOSMOTION

COSMOSMotion is a computer software tool that supports engineers to analyze and design mechanisms. COSMOSMotion is a module of the SolidWorks product family developed by SolidWorks Corporation. This software supports users to create virtual mechanisms. The analysis and simulation capabilities in CosmosMotion employ simulation engine, ADAMS/Solver, which solves the equations of motion for the mechanism. Typical simulation problems, including statics (equilibrium configuration) and motion (kinematics and dynamics), are supported. All mechanisms can be simulated and visualized as shown in Figure 5.12.

COSMOSMotion creates a kinematics analysis model; e.g., using a motion driver to drive the mechanism; and then carries out dynamic analyses. In dynamic analysis, position, velocity, and acceleration results are identical to those of kinematics analysis. However, the inertia of the bodies will be taken into account for analysis; therefore, reaction forces are calculated between bodies.

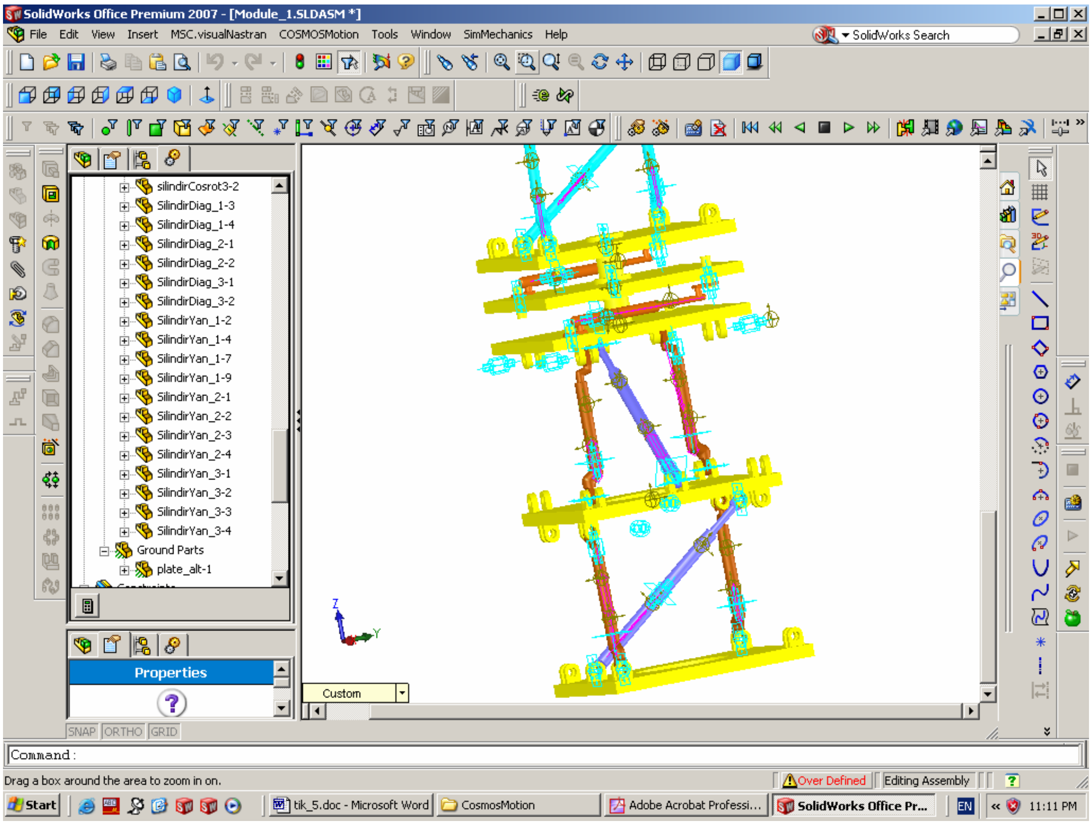


Figure 5.12 The General View of SolidWorks with COSMOSMotion

The overall process of using COSMOSMotion for analyzing a mechanism consists of three main steps: model generation, analysis (or simulation), and result visualization (or post-processing), as illustrated in Figure 5.13. The analysis or simulation capabilities in COSMOSMotion employ simulation engine, ADAMS/Solver, which solves the equations of motion for the mechanism. ADAMS/Solver calculates the position, velocity, acceleration, and reaction forces acting on each moving part in the mechanism.

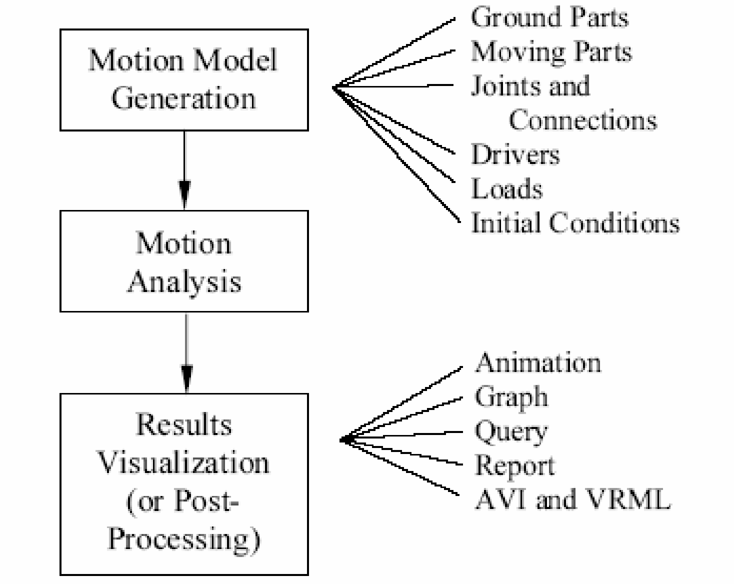


Figure 5.13 General Process of Using COSMOSMotion

ADAMS/Solver simulation engine has some features as follows:

- ADAMS/Solver uses the system of Euler-Lagrange equations of motion. This system, composed of algebraic and second order differential equations, requires sophisticated numerical methods for their solutions.
- ADAMS/Solver solves all equations simultaneously.
- ADAMS/Solver arbitrarily determines which constraints are redundant, deletes them from the set of equations, and provides a set of results that characterize the motion and forces in the system.
- ADAMS/Solver can simulate an overconstrained model only if the redundant constraints are consistent. Redundant constraints are consistent if a solution

satisfying the set of independent constraint equations also satisfies the set of redundant constraint equations.

- If a singularity takes place in the initial position, ADAMS/Solver removes it by making very small motions. It uses the information at the previous iteration to pass singular positions when mechanism is in motion.

The solid model of all the links of the manipulator and their assemblies are made in SolidWorks software. The applied forces, the dissipative forces and the gravity forces are modeled in COSMOSMotion and then, the dynamic equations are solved.

5.5 The Comparison of the Joint Variables of Hyper Redundant Manipulator Found in COSMOSMOTION and the Computer Program in MATLAB

The computer program to solve the unknowns of the dynamic equation set has been written in MATLAB with respect to the algorithm expressed above. It consists of a main program and many sub programs. In this part, the solution of the independent joint variables of the HRM will be found out for the different initial and final positions of the manipulator. Some assumptions have been made in this application. The applied forces on the actuators are taken as fixed due to the pneumatic calculations and the physical velocity restriction. It is supposed that the dissipative force type is viscous friction on the actuators and, the dissipative forces on the passive joints are assumed to be negligible. In addition to these, in the simulations, the applied forces are assumed to be large enough to perform the motions and the viscous friction coefficient is taken as larger because these cover the realistic requirements of such a system.

After obtaining the dynamic equations of HRM with Newton-Euler method, the joint variables, which are the lengths of the pneumatic cylinders, have been solved in the computer program written in MATLAB. For visualization and justification, the hyper redundant manipulator consisting of the first two of ten modules has been constructed in SolidWorks CAD software and the dynamic solution of this part has been investigated to find out the joint variables in CosmosMotion. For example, the simulations for both methods have been run under the same conditions. The results are shown in Figures 5.14 and 5.15.

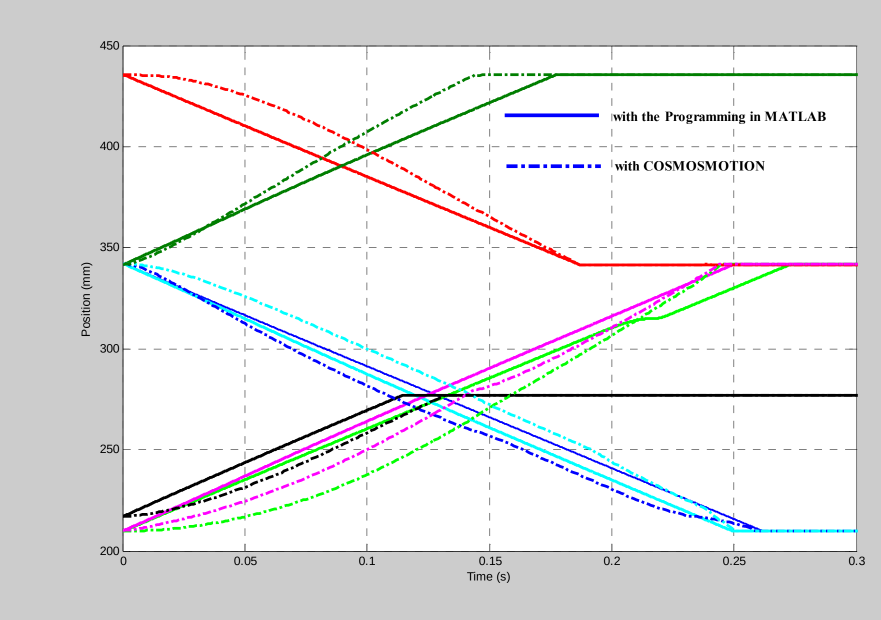


Figure 5.14 The Joint Variables for the Module 1

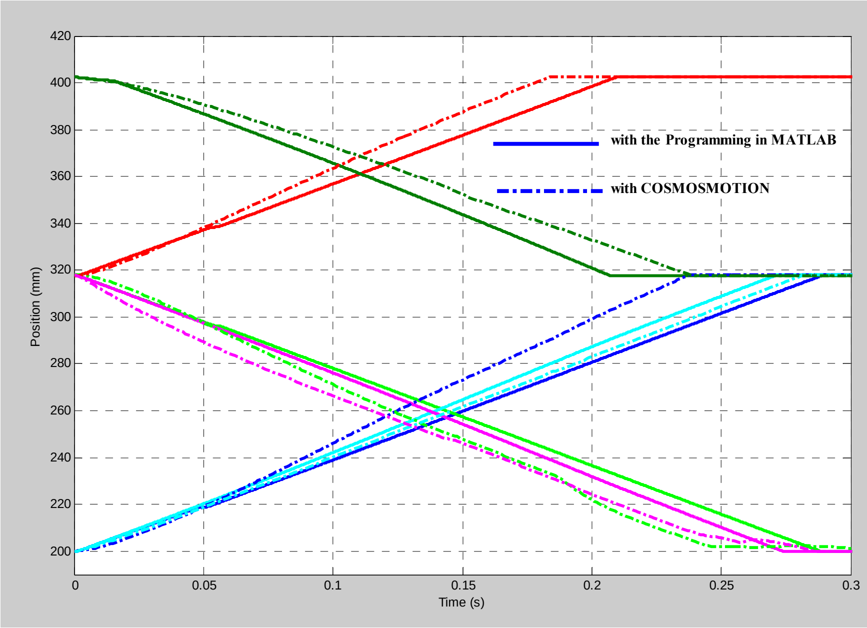


Figure 5.15 The Joint Variables for the Module 2

As shown in the figures above, the variations of the joint variables obtained with both methods are almost the same. Nevertheless, some errors appear between some joint variables. This contradiction can depend on some reasons. The first one, when the CosmosMotion encounters a redundancy in solving the dynamics equations, the software removes the redundant equations. However, in order to eliminate the redundancy for solving the dynamic equation by the computer program, some assumptions and simplifications have been made and, during the simulation we don't interfere the simulation. This operation is made by changing virtually the joint types for only the conformity of the equation set. Another reason to cause this contradiction may be that the solution technique is different for these two methods. The CosmosMotion uses iterative approach to solve the joint variables while the computer program written in MATLAB uses matrix inversion method for the solution.

5.6 Summary and Results

In this chapter, the dynamics analysis of the hyper redundant manipulator has been considered in detail by using Newton-Euler Laws. At first, the equations of motion are obtained for a single module. Then, these equation set is applied to construct the equations of the motion for the whole system because the manipulator is module-based, i.e. all modules have same constructive characteristics. These system equations are written in matrix form. The dimensions of the system matrix are rather large to solve the system unknowns, which are the independent joint variables and the constraint forces. Especially, the constraint forces (reaction moment and force components) increase the dimension. This causes too much time loss to solve the equations. It is known that, according to the virtual work principles, the constraint forces are not involved in the equations of the motion since the virtual work of them equals to zero. Using embedding technique, the dimension of the system matrix is diminished. The reaction forces and moment components can be then found by using the calculated independent joint variables.

The solid model of the hyper redundant manipulator is designed in SolidWorks computer program and, the features of each part of the manipulator are given in the same program. After obtaining the set of inertial and dimensional features, the dynamic equations are solved with the algorithm written in the MATLAB computer program. At the same time, the results for the same features are obtained by using the CosmosMotion dynamic simulation software. The two groups of the results are compared with each other on the same figures. The plots show that the two methods give approximately the same results. However, there are some differences for the independent joint variables between the results with the MATLAB algorithm and the one with CosmosMotion. These differences can be originated from firstly the solution technique and secondly the behavior of the CAD software when to encounter the redundancy in the simulation.

According to the figures, the independent joint variables change linearly with time. This conforms to the simple modeling of the pneumatic actuator for hyper redundant manipulators by Chirikjian [40]. However, the actuators can open and close at the different times.

In the simulations, it is assumed that the actuator forces are larger enough and the coefficient of the viscous friction is large. This assumption causes the actuators to make almost linear motion.

CHAPTER 6

THE APPLICATIONS OF THE HYPER REDUNDANT MANIPULATOR

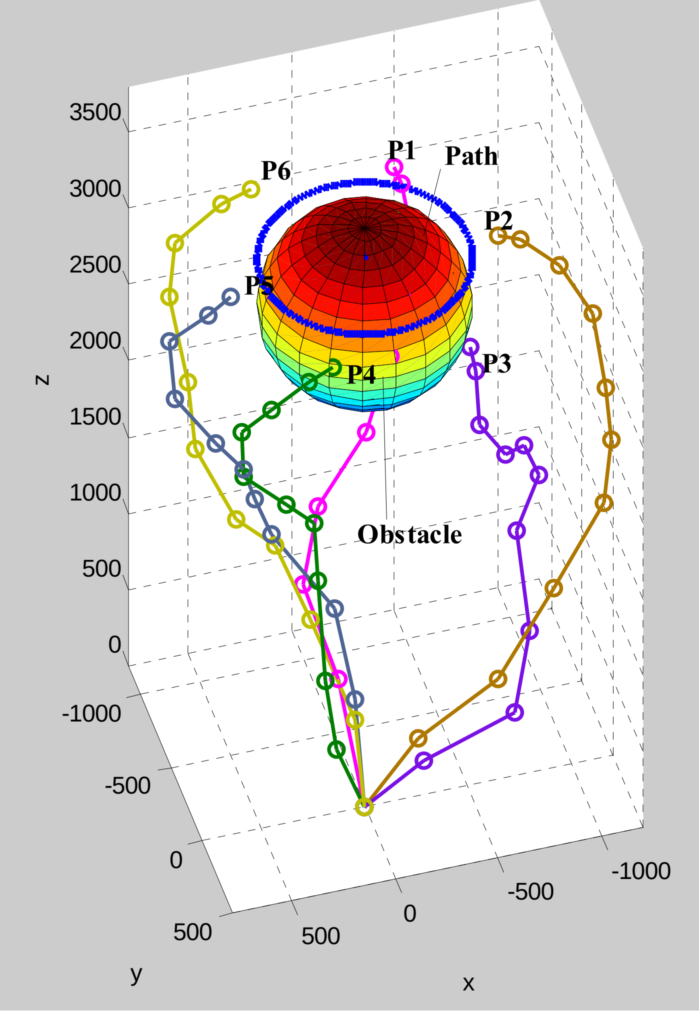
6.1 Applications

Binary hyper redundant manipulators can be used in many applications such as pick-and-place, spot welding, repairing and inspection in dangerous and complex areas such as space satellites, nuclear powers, bridges. Compared to their continuously actuated counterparts, they have many advantages such as much cheaper, light weight, no need for feedback, less complexity in computer controlled interfacing, allowing tasks to be performed even when some actuators fail etc. In spite of these advantages, they have a discrete workspace and can reach a finite set of points in this workspace. In addition to this, binary hyper redundant manipulators can not follow a given reference trajectory when they move from a configuration to another one since the actuators used in binary hyper redundant manipulators have either fully open or fully closed position. This does not allow to have any control in intermediate positions, and the manipulators can deviate excessively from a given reference trajectory. As a result, these manipulators can not be used in the applications based on a certain trajectory except for some specific cases.

In this chapter, some applications of the hyper redundant manipulator are given. These applications show the capability of the manipulator and its applicability in different fields. The first application is the path tracking of the manipulator and, the second one is grasping and wrapping an object.

6.1.1 The Path Tracking

As expressed earlier, one of the fields of binary hyper redundant manipulators is spot welding. In this study, according to the inverse kinematics, the manipulator can reach any given point with very small error in its workspace. This provides the manipulator to make spot welding easily. However, if the manipulator is required to follow the continuous path exactly, then, this process can not be performed due to the reasons expressed above. Despite working in the discrete space, the last module of the HRM is taken as a continuous manipulator to fine tuning so that it can follow any continuous path in its workspace. The 6 DOF fine tuning manipulator (FTM) has been selected for the continuously actuated manipulator. As an example, the circular path is to be tracked. The application of the path tracking can be applied for continuously welding of object as well as cleaning, repairing. For this path tracking, at first, the discrete target points are specified as shown in Figure 6.1. According to these selected discrete points, the convenient configurations are found by using the workspace filling method (or the extended spline fitting method) in such a way that the manipulator can avoid the obstacles and its trajectory is as close as possible to the given circular path. The last module traces the circular path while the discrete parts of the HRM are passing through these discrete target points. However, in this motion, the trajectory of the discrete part, especially the trajectories of the last module, get out of the space in which the continuously actuated part of the HRM can reach. This makes the path tracking impossible for the specified dimensions of the continuously actuated part (the FTM). For this reason, in this application, we use another procedure for the path tracking. First, the circular reference path is divided into some parts such that the initial and final points of each part should be in the workspace of the continuously actuated manipulator. Secondly, the discrete modules are moved from the initial configuration to the next configuration and stop, then, the FTM goes to the initial point of the circular path and follows the path until the end point of the segmented part. After that, the same procedure is applied for the next steps till completing the whole path tracking work. This procedure is shown in Figure 6.2.



The Figure 6.1 The Selected Configurations for the Path Tracking

Note that the motion between two successive configurations is continuous although the system is discrete. The values of the joint variables, the length of actuators, between two successive configurations have been taken from the data files found with the inverse kinematics and dynamics analysis. These files are created by the offline solution of the manipulator by using the computer programs written in MATLAB. Then, the solutions are used in the online path tracking problem. In Figure 6.3, the trajectory of the last discrete module of the HRM, the circular trajectory to be tracked and the initial configuration are shown. According to this figure, it is obvious that the fine tuning manipulator could not follow the reference circular curve for the HRM in motion.

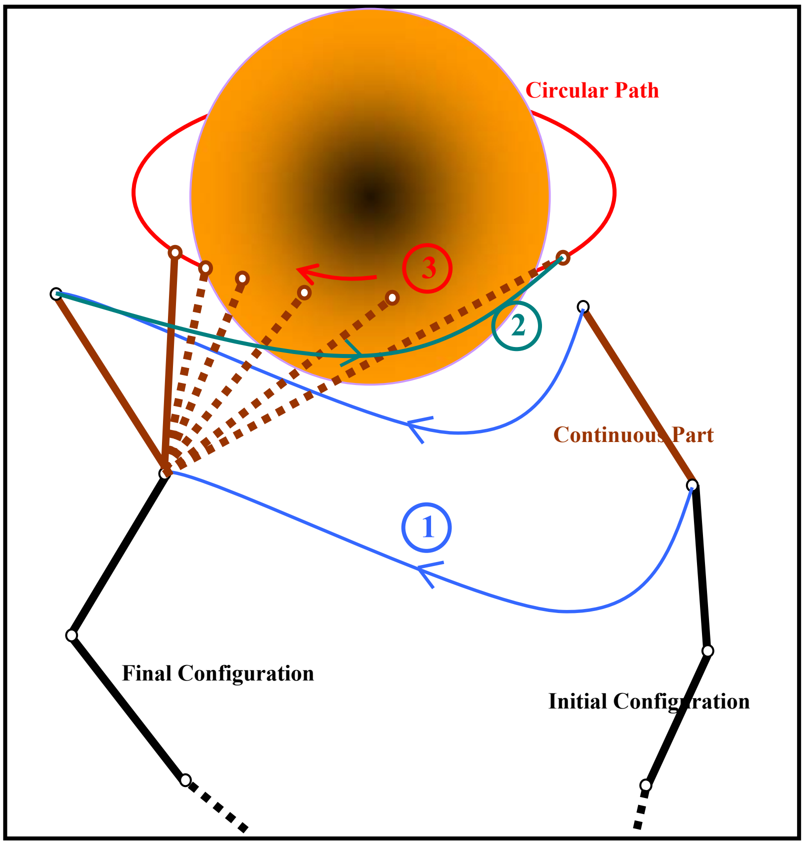


Figure 6.2 The Path Tracking Procedure

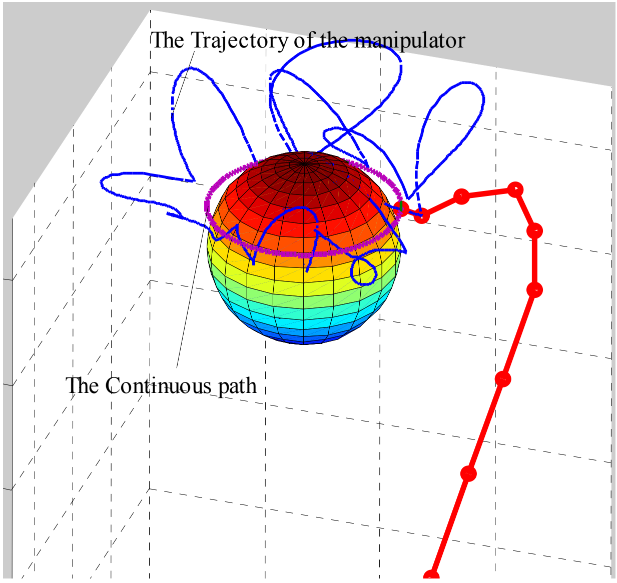


Figure 6.3 The Trajectory of the HRM for the Path Tracking

In the applications, the selection of the configuration is very important. The configurations should not collide with the obstacles and should have smaller deviation from the reference curve. Also, the distance of the last point between two successive configurations should be in the workspace of the FTM. In addition to these criteria, the orientational error of the top platform of the HRM is another criterion for the selection of configuration. Although the target position is in the workspace of the FTM, any configuration as shown in Figure 6.4(a) increases the burden on the FTM considerably. On the other hand, Figure 6.4(b) shows a convenient target position for the FTM.

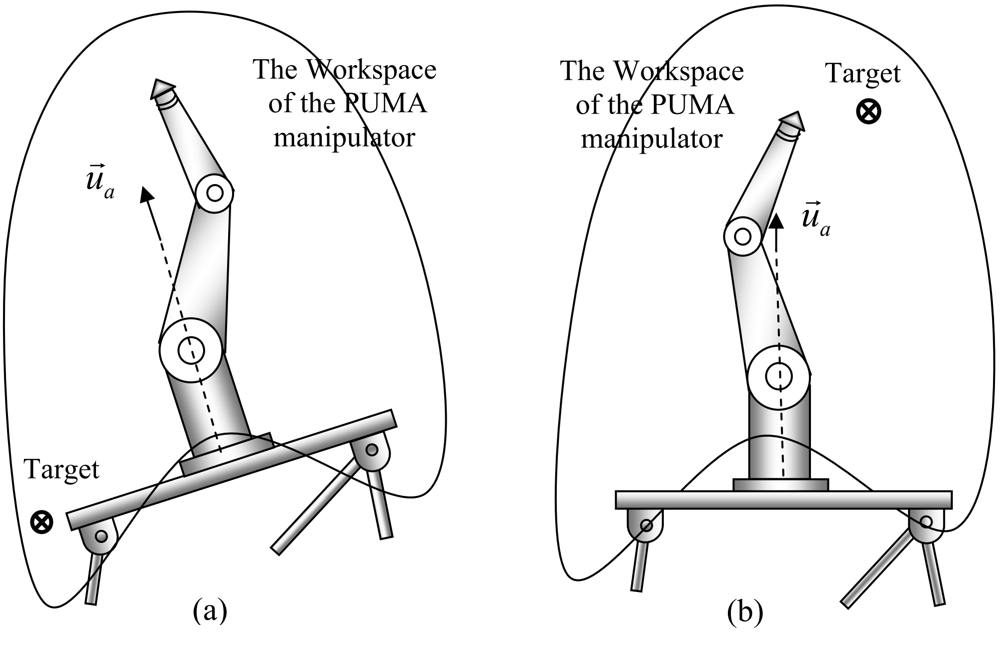

The Workspace of the PUMA manipulator
$\vec{u}_a$
Target
⊗
(a)
The Workspace of the PUMA manipulator
Target
⊗
$\vec{u}_a$
(b)

Figure 6.4 Inconvenient (a) and Convenient (b) Target Positions for the FTM

6.1.2 Grasping an Object

In this application, the manipulator grasps an object, surrounding it, and moves it from one position to another one. The last discrete 4 modules of the manipulator makes grasping and the fine tuning manipulator (FTM) presses the object at a convenient point to hold it stably as the remaining modules carries the object to the desired place. It is assumed that the manipulator provides enough contact force and there is no possibility of rotation of the object during grasping and carrying. Figure 6.5 and 6.6 shows this application. In this application, the shape of the manipulator can be defined by the inverse kinematics. However, the motion from any configuration to this grasping configuration should be defined by the spline fitting method or heuristically. Not only the change of on-off positions of the actuators, but also the sequence of the change of actuator is important for efficient process. Here, the grasping is made by the last four modules and the continuously actuated manipulator. Therefore, the actuator states can be specified for different objects in such a way that the maximum contact points and surface are obtained to move the object in a stable manner. In this application, any analysis for the grasping has not been made and also only a cylindrical object as a sample is selected.

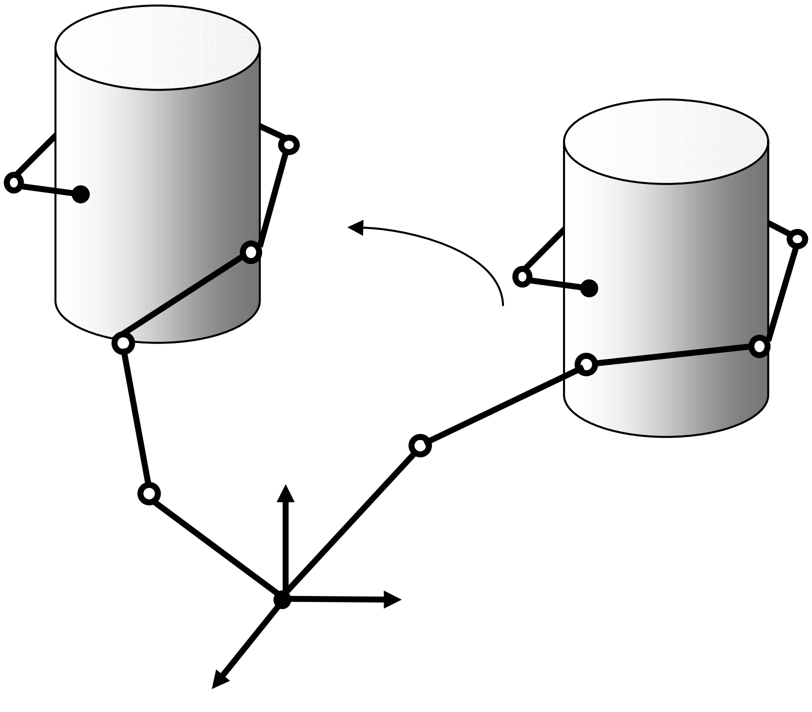


Figure 6.5 The Schematic of Grasping an Object

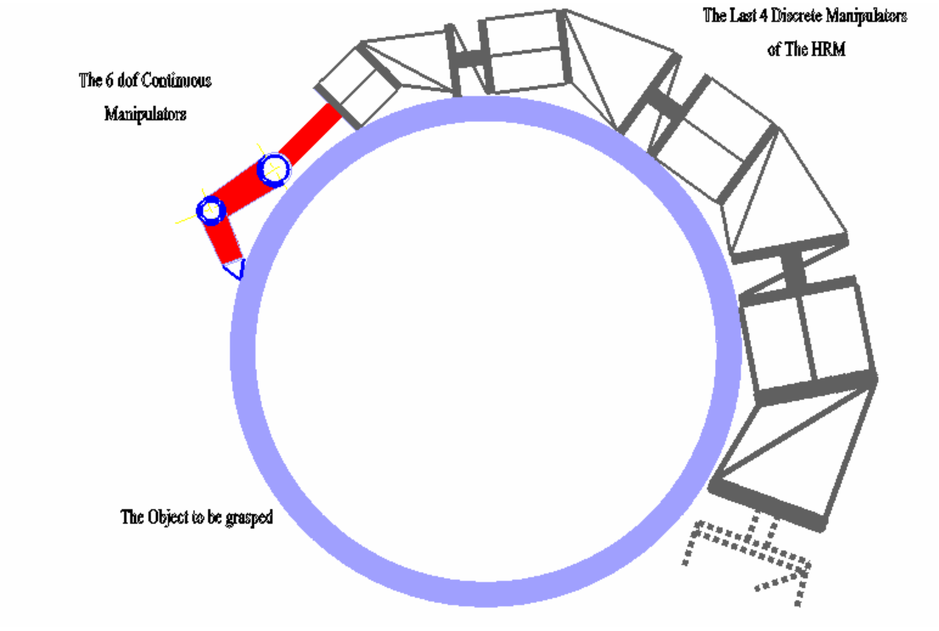

The Last 4 Discrete Manipulators of The HRM
The 6 dof Continuous Manipulators
The Object to be grasped

Figure 6.6 The Grasping an Object

6.1.3 Wrapping an Object

In the previous study, it is shown that the manipulator could grip a cylindrical object and move it from a position to another one. In addition, the HRM can wrap around a cylindrical object in 3D space as shown in Figures 6.7 and 6.8. However, it can not completely wrap around the object. In this type of applications, for full wrapping an object, it is required that many modules should be used, dimensions must be smaller and motion of each module must be larger. Like the grapping operation, in this application, the shape of the manipulator can be defined from the inverse kinematics and the sequence of the action of the actuator should be specified.

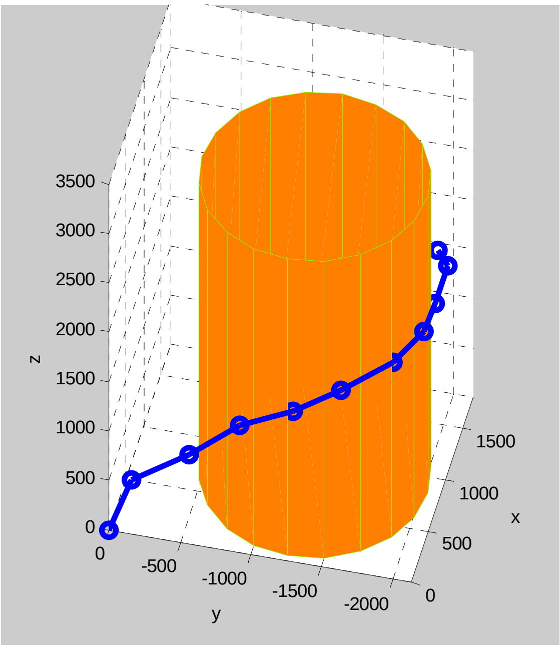


Figure 6.7 Wrapping an Object Graphically

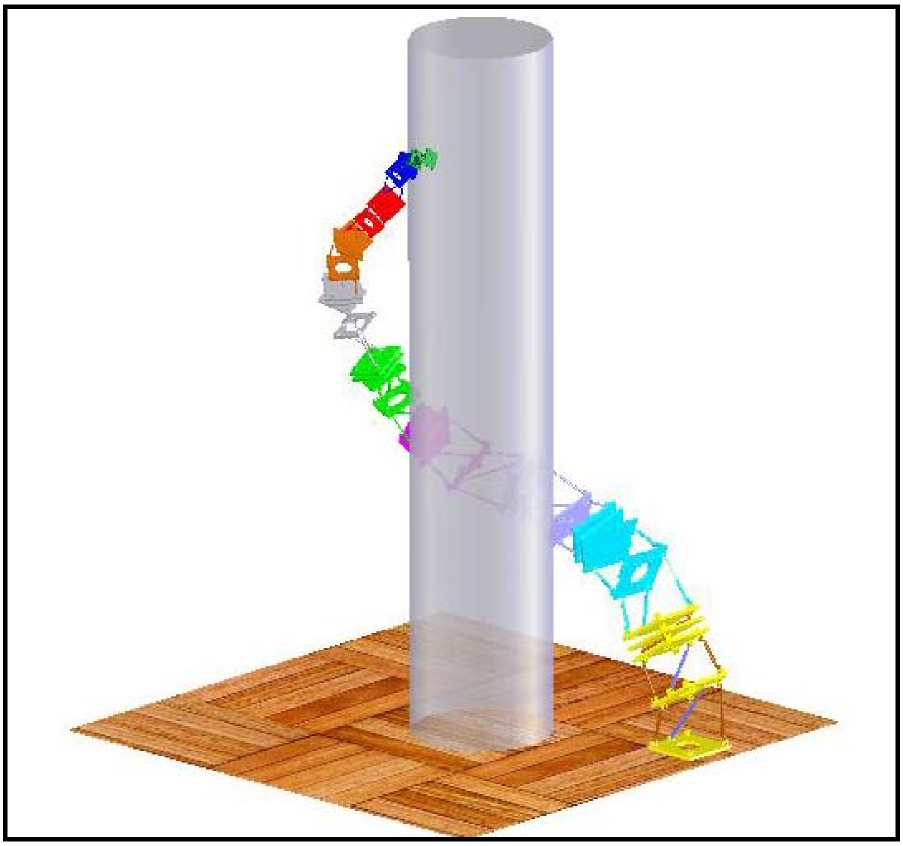


Figure 6.8 Wrapping an Object in SolidWorks Environment

CHAPTER 7

CONCLUSION

In this study, a hyper redundant manipulator driven by binary actuators has been devised and controlled by the inverse kinematics methods. It also has an additional continuously actuated manipulator at its tip for fine tuning. In the first part (Chapter 2), the manipulator has been devised and modified to improve its rigidity and performance with respect to the structure and dimensions of the modules. In the second part (Chapters 3 & 4), the kinematics of the hyper redundant manipulator (HRM) has been studied and, the algorithms have been developed to control the manipulator. In Chapter 5, the dynamic analysis has been carried out to create the equations of motion for the HRM. In the last part, the applications of the manipulator such as the path tracking, grasping an object have been studied.

Note that the hyper redundant manipulators have some disadvantages such as insufficient rigidity and powerful actuation system. The modifications on the HRM, which the structure of the variable geometry truss (VGT) submodules is converted from parallelogram to trapezoidal, and the manipulator constructed like a cone, improve these disadvantages. Furthermore, these modifications reduce the lateral forces on the actuators and joints.

The most important disadvantage about the discretely actuated hyper redundant manipulators is that their inverse kinematics and control of trajectories based on the inverse kinematics are rather difficult to implement the given tasks. In this study, the inverse kinematics of the HRM with ten modules has been studied for

$\left(2^3 \times 2^3 \times 3^1\right)^{10} \simeq 6.8 \times 10^{22}$ possible configurations. For such a huge system, three approaches have been offered for the inverse kinematics. The first approach is the spline fitting method based on a reference curve which is a spline. With this method, the manipulator is fitted to the spline part divided into the number of the modules. This method needs only a little bit memory and the operation for the library construction and selection of the configuration take very short time since this method is module-based and the searching process is based only on the number of the module configurations. Using this method, the manipulator can fit fairly well into the given spline on condition that the spline confirms with the motion capability of the HRM. However, the results show that the positional errors can be larger although the manipulator can fit well into the reference curve. Even, these errors can be out of the limits which the continuously actuated manipulator can compensate. This is possible because the searching process is module-based and made from the base to the top. This procedure accumulates the larger errors on the top of the HRM. In some cases, especially, for the splines which do not fit the motion capabilities of the manipulator, this method would give the inconvenient configurations, which deviate excessively from the reference curve.

To eliminate the drawbacks of the first method, the extended spline method has been offered. This method is the combination of the spline fitting method and the genetic algorithm. The extended spline fitting method has supplied a quite improvement on this first method as seen from Table 4.2 and Figure 4.26-30.

In another approach for the inverse kinematics, the workspace filling method based on the workspace of the manipulator has been offered. Instead of the whole workspace of the manipulator involving $\left(2^3 \times 2^3 \times 3^1\right)^{10} \simeq 6.8 \times 10^{22}$ configurations, a workspace which is the subset of this whole workspace and consists of a finite number of configurations is created randomly or by using the genetic algorithm. The selection process is made with respect to some criteria which are the minimum positional and orientational errors, the shape of the manipulator and the trajectory tracking. The selected configurations can provide the desired criteria as shown in

Chapter 4. However, a problem related to this method is appeared. Although the other criteria are supplied, one does not have a control over the shape of the manipulator since the set of the configurations is selected randomly from the workspace created.

The comparison of these three methods is made according to Figure 4.26-30 and Table 4.2. The results point out that the extended spline fitting and the workspace filling methods give more feasible solutions than ones with the spline fitting method in terms of the positional errors. In addition, the extended spline fitting and the workspace filling methods contain the minimum orientational errors whereas this is not involved in the spline fitting method except for the torsion. When considered the shape of the manipulator, all three methods, especially the extended spline fitting method, can provide the convenient solutions. The other criterion for the comparison is the trajectory tracking. The extended spline fitting and the workspace filling methods can select the configurations with better trajectories since these two methods provide multiple solutions for the inverse kinematics and involve a set of the solution configurations instead of one configuration. In the spline fitting method, this criterion is excluded from the searching since the method does not have any relation about the trajectory between the successive configurations and selects only one configuration. However, the control of the trajectory tracking with all these methods does not seem to be feasible. The trajectory of the manipulator can deviate excessively from the given reference curve. In fact, although it is claimed in the literature that the combinatorial method [34] gives feasible solutions for the trajectory tracing, this study was given for the planar manipulators and the solution configuration was searched on the neighboring of the preceding configuration, i.e. the method tries to find the solution by changing the state of only a few actuators. This is not sufficient for the 3D space and changing the state of actuators towards the base of the manipulator can cause larger deviations at the top of the manipulator.

For the workspace filling method, the workspace is divided some restricted parts for different tasks and, also with this method, the larger configuration diversity is obtained. Because of these, in addition to four criteria expressed above, the workspace filling method is useful for the multiple tasks and avoiding the moving or unexpectedly appearing obstacles. For the task of avoiding the obstacles, the HRM has already a set of solution configuration with respect to its tip point. Thus, by compromising the tip point accuracy by a small amount, a different configuration from this set can be selected in such a way that the manipulator can not collide with the obstacles.

In this thesis, the configurations found with these three inverse kinematics methods are not necessarily globally optimum. However, the configurations found by the extended spline and the workspace filling methods are local optimum. These methods can be applied to all binary actuated hyper redundant planar, spatial, parallel manipulators. The use of these methods is the task dependent. If the obstacles in the environment are definite, the extended spline method is preferred to select better configurations. Otherwise, the workspace filling method is more useful for adaptable configurations. As for the spline fitting method, it is better for reducing the computation time. Table 7.1 summarizes the comparison of these three inverse kinematics methods with respect to several criteria.

Table 7.1 The Comparison of Three Inverse Kinematics Methods

	Spline Fitting Method	Extended Spline Fitting Method	Workspace Filling Method
Positional Error	✗	✓	✓
Orientation Error	✗	✓	✓
Shape	✓	✓	✗
Trajectory	✗	✓	✓
Time	✓	✗	✗
Multiple Tasks	✗	✓	✓
Moving Obstacles	✗	✗	✓

Finally, with the dynamic analysis, the equations of motion have been written in terms of the pneumatic forces generated by the binary actuators and the real motion of the manipulator has been obtained by using the solution of these equations. The results show that the real motion of the manipulator between two successive configurations has a very non-smooth trajectory with respect to the given reference curve.

In some cases, the manipulator can not reach the desired configurations. The discrete nature of the manipulator causes this problem. The manipulator could take the desired shape and follow exactly the given reference trajectory if it were actuated continuously. So, it is desirable to have a binary hyper redundant manipulator to approximate the motion capability of a continuously actuated counterpart as much as possible. For this aim, the dimensions of the modules, i.e. the dimensions of their actuators, should be diminished and the stroke of the actuators with respect to their nominal length should be increased. However, these are some theoretical considerations. Here, there appears a problem. In fact, designing such actuators with much smaller dimensions is rather difficult in practice. In particular, when these actuators are used towards the base of the manipulators, to actuate such a huge system becomes almost impossible.

The future works can be focused on the following subjects.

- As expressed earlier, in order to implement the given task correctly the structure of the discretely actuated hyper redundant manipulators should be approached to the continuously actuated counterparts. Therefore, a study will be focused on devising new binary small but rigid actuators. These should provide larger power and have very small dimensions and larger strokes relative to their nominal lengths.

- In order to define a spline efficiently for inverse kinematics of hyper redundant manipulators, the position of the control points is a very important subject. For the further study, according to the given task and motion capability of HRMs, the positions and number of these points can be obtained by an appropriate optimization study.

- The environment in which binary HRMs work may involve moving or unexpectedly appearing obstacles. In such an environment, the control of binary HRMs is rather difficult in real time by using the ordinary or extended spline fitting methods because of the long computation times they need. However, the workspace filling method generates a larger configuration diversity readily. Therefore, it can be useful to adapt the configuration of the HRM quickly to the changes in the environment. As a future work, an efficient algorithm can be developed for this purpose.

- An investigation related to this binary hyper redundant manipulator can be made for different realistic applications. Also, defining the tasks, which the HRM with the present actuators can make, is another subject to be studied for the future work. That is, the working limit of the manipulator needs to be defined.

- The task of trajectory tracking with the discretely actuated hyper redundant manipulators is still a dilemma. Different methods for solving this problem can be studied for a discrete workspace with a very large number of configurations.

REFERENCES

[1] Anderson V.C., and Horn R.C., "*Tensor Arm Manipulator Design*" ASME paper, 67-DE-57, 1967.

[2] Chirikjian, G.S., and Burdik, J.W., "*An Obstacle Avoidance Algorithm for Hyper-Redundant Manipulators*", Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, May, 13-18, 1990.

[3] Hirose, H., "*Biologically Inspired Robots (Snake-like Locomotor and Manipulator)*", Oxford University Press, 1993

[4] Hirose, H. and Chu, R., "*Development of a Lightweight Torque Limiting M-Drive Actuator for Hyper-Redundant Manipulator Float Arm*", in Proc. IEEE International Conference on Robotics and Automation, pp2831-2836, 1999.

[5] Chirikjian, G.S., "*Theory and Applications of Hyper-Redundant Robotic Mechanisms*", PhD Thesis, Department of Applied Mechanics, California Institute of Technology, Pasadena, CA (1992).

[6] Chirikjian, G.S., and Burdik, J.W., "*Design and Experiments with a 30 DOF Robot*", ISRAM'93, November 1993, pp. 113-119.

[7] Chirikjian, G.S., and Burdik, J.W., "*A Hyper Redundant Manipulator*", IEEE Robotics and Automation Magazine, December 1994, pp. 22-29.

[8] Hannan, M.W. and Walker, I. D., "*Analysis and Experiments with an Elephant's Trunk Robot*", Advanced Robotics 15, 847–858, 2001.

[9] Hickman, F., Henning, W. and Choset, H., "*Motion Planning for Serpentine Robots*", ASCE Space and Robotics, Albuquerque, New Mexico, 1998.

[10] Carnegie Melon University, Howie Choset's Serpentine Robots, http://www.cs.cmu.edu/~biorobotics/serpentine/serpentine.html, 01/02/2009.

[11] Kimura, S., Takahashi, M., Okuyama, T., Tsuchiya, S., and Suzuki, Y., "*A Fault-tolerant Control Algorithm for Space Hyper-Redundant Manipulators Standing on Decentralized Autonomous Architecture*", IEEE Trans. Syst. Man and Cebern., Vol. 28, No: 4, July 1998.

[12] Yim, M., "*A Reconfigurable Modular Robot with Many Modes of Locomotion*", Proceedings of the 1993 JSME International Conference on Advanced Mechaton, Tokyo, August 1993, pp. 282-288.

[13] Yim, M., Shen, W., Salemi, B., Rus, Mall, D., Lipson, M. H., Klavins, E. and Chirikjian, G. S., "*Modular Self-Reconfigurable Robot Systems: Challenges and Opportunities for the Future*", IEEE RAM. 14(1):43-52 Mar. 2007

[14] Chirikjian, G.S. and Pamecha, A., "*Bounds for Self-Reconfiguration of Metamorphic Robots*", Proceedings of the 1996 IEEE ICRA, 1996.

[15] A. Koliskor, "*The 1-Coordinate Approach to the Industrial Robots Design, in Information Control Problems in Manufacturing Technology*", Proc. 5th IFAC/IFIP/ IMACS/ IFORS Conf., Suzdal, USSR, pp. 225– 232, 1986.

[16] Chirikjian, G.S., "*A Binary Paradigm for Robotic Manipulators*", Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, May 1994, pp. 3063-3069.

[17] Ebert-Uphoff, I, "*On the Development of Discretely-Actuated Hybrid-Serial-Parallel Manipulators*", PhD thesis, Johns Hopkins University, 1997.

[18] Suthakorn J. "*Paradigm for Service Robotics*", PhD thesis, Johns Hopkins University, 1997.

[19] Suthakorn J., "*Binary Hyper-Redundant Robotic Manipulator Concept*", TENCON 2004. 2004 IEEE Region 10 Conference, Vol. 4, pp. 625- 628, 21-24 November 2004.

[20] Suthakorn J. and Chirikjian, G.S., "*Design and Implementation of a New Discretely Actuated Manipulator*", International Symposium on Experimental Robotics 2000, 7th ISER 2000, Waikiki, Hawaii, USA, December 11-13, 2000.

[21] Ottaviano,E., Ceccarelli, M., Di Giorgio, J. and Varone, M., "*Design and Evaluation of a Discretely Actuated Multi-Module Parallel Manipulator*", Proceedings of the Institution of Mechanical Engineers, Journal of Mechanical Engineering Science, Vol. 220, No: 4, 2006.

[22] Lichter M.D. Sujan V.A. Dubowsky S., "*Experimental Demonstration of a New Design Paradigm in Space Robotics*", In 7th International Symposium on Experimental Robotics (ISER), pp. 10-13, Honolulu, Hawaii, December 2000.

[23] Sujan V.A., Lichter M.D. and Dubowsky S., "*Lightweight hyper-redundant binary elements for planetary exploration robots*", In IEEE/ASME

Conference on Advanced Intelligent Mechatronics (AIM), Como, Italy, July 2001.

[24] Sujan V.A. and Dubowsky S., "*Design of a Lightweight Hyper-Redundant Deployable Binary Manipulator*", ASME Journal of Mechanical Design, 126/29-39, January 2004.

[25] Chirikjian, G. S. and Burdick, J.W., "*A Modal Approach to Hyper-Redundant Manipulator Kinematics*", IEEE Transactions on Robotics and Automation 10(3):343–354, 1994.

[26] Zanganeh, K.E. and Angeles, J., "*The Inverse Kinematics of Hyper Redundant Manipulators Using Splines*", Proceedings of IEEE International Conference on Robotics and Automation, pp. 2797-2802, Nagoya, May 1995.

[27] Hannan, M.W. and Walker, I.D., "*Kinematics and the Implementation of an Elephant's Trunk Manipulator and Other Continuum Style Robots*", Journal of Robotic Systems 20(2), 45-63, Wiley Periodicals, Inc., 2003.

[28] Chirikjian, G.S. and Burdick, J.W., "*Kinematically Optimal Hyper-Redundant Manipulator Configurations*", Proc. IEEE Conf. on Robotics and Automation, Nagoya, pp. 794–780, 1995.

[29] Zanganeh, K.E., Lee, R.S.K and Hughes, P.C., "*A Discrete Model for the Configuration Control of Hyper Redundant Manipulators*", Proceedings of the 1997 IEEE International Conference on Robotics and Automation, pp. 167-172, Albuquerque, New Mexico, April 1997.

[30] Agirrebeitia, J, Aviles, R., De Bustos, I.F. and Ajuria, G., "*A Method for the Study of Position in Highly Redundant Multibody Systems in Environments with Obstacles*", IEEE Transactions on Robotics and Automation, Vol. 18, No. 2, April 2002.

[31] Wang, Y. and Chirikjian, G.S., "*Workspace Generation of Hyper-Redundant Manipulators as a Diffusion Process on SE(N)*", IEEE Transactions on Robotics and Automation, Vol. 20, No. 3, June 2004.

[32] Gevher, M., Erkmen, A.M. and Erkmen, I., "*Sensor Based Online Path Planning for Serpentine Robots*", Proceedings of the 2001 IEEE International Conference on Robotics and Automation, Seoul, Korea, May 21-26, 2001.

[33] Takanashi, N, Choset, H. and Burclick, J.W., "*Simulated and Experimental Results of Dual Resolution Sensor Based Planning for Hyper-redundant Manipulators*", Proceedings of the 1993 IEEWRSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 2630,1993.

[34] Lees D.S. and Chirikjian G.S., "*A Combinatorial Approach to Trajectory Planning for Binary Manipulators*", In IEEE International Conference on Robotics and Automation, pp. 2749-2754, Minneapolis, MN, 1996.

[35] Lees D.S. and Chirikjian G.S., "*An Efficient Trajectory Planning Method for Binary Manipulators*", In ASME Mechanisms Conference, 96-DETC/MECH-1, Vol. 161, 1996.

[36] Ebert-Uphoff, I. and Chirikjian G.S., "*Inverse Kinematics of Discretely Actuated Hyper-Redundant Manipulators Using Workspace Densities*", In IEEE International Conference on Robotics and Automation, pp. 139-145, 1996.

[37] Ebert-Uphoff I. and Chirikjian G.S., "*Efficient Workspace Generation for Binary Manipulators with Many Actuators*" Journal of Robotic Systems, 12:383-400, June 1995.

[38] Chirikjian G.S. and Ebert-Uphoff I., "*Numerical Convolution on the Euclidean Group with Applications to Workspace Generation*", IEEE Transactions on Robotics and Automation, 14(1):123-136, 1998.

[39] Suthakorn J. and Chirikjian G.S., "*A New Inverse Kinematics Algorithm for Binary Manipulators with Many Actuators*", Advanced Robotics, Vol. 15, No. 2, pp. 225– 244, 2001.

[40] Chirikjian, G.S., "*Inverse Kinematics of Binary Manipulators Using a Continuum Model*", Journal of Intelligent and Robotic Systems 19: 5–22, Kluwer Academic Publishers, Netherlands, 1997.

[41] Kim, Y.Y., Jang, G.W. and Nam, S.J., "*Inverse Kinematics of Binary Manipulators by Using the Continuous-Variable-Based Optimization Method*", IEEE Transactions on Robotics, Vol. 22, No. 1, February 2006.

[42] Lichter, M.D., Sujan, V.A. and Dubowsky, S., "*Computational Issues in the Planning of Binary Robots*", Proceedings of the 2002 IEEE International Conference on Robotics and Automation, Washington, DC, May 2002.

[43] Ryu, G. and Rhee, I., "*A Kinematic Analysis of Binary Robot Manipulator Using Genetic Algorithms*", International Journal of the Korean Society of Precision Engineering, Vol. 2, No. 1, April 2001.

[44] Scofano, F.D.S, Meggiolaro, M.A. and Sujan, V.A., "*Inverse Kinematics of a Binary Flexible Manipulator using Genetic Algorithms*", 18th International Congress of Mechanical Engineering, Ouro Preto, MG, November 6-11, 2005.

[45] Bayram, A. and Özgören, M.K., "*Aşırı Artıksıl Bir Manipülatörün Modellenmesi ve Kontrolü*", 14th Ulusal Makina Teorisi Sempozyumu, Güzelyurt, KKTC, 2-4 Temmuz 2009.

[46] Farrow, W.C., "*Motion Model for a Variable Geometry Truss (VGT) Manipulator*", PhD Thesis, Marquette University, Wisconsin, 1994.

[47] Özgören, M. K., "*Kinematic Analysis of Spatial Mechanical Systems Using Exponential Rotation Matrices*", Journal of Mechanical Design, ASME, Vol. 129, pp. 1144-1152, November 2007.

[48] Piegl, L. and Tiller, W., "*The NURBS Book*", Springer-Verlag Berlin Heidelberg, Germany, Second Edition, 1997.

[49] Emery, D. J., "*Minimal Energy Splines*", PhD Thesis, The Faculty of the Graduate School of The University of Maryland, 1989.

[50] Cebula, A.J., "*Use of NURBS to Avoid Boundary Orientation Poses in Serial Manipulators*", PhD Thesis, Department of Mechanical Engineering, McGill University, Montreal, Canada, August 2003.

[51] Goldberg, D., "*Genetic Algorithms in Search, Optimization, and Machine Learning*", Addison-Wesley, Reading, MA, 1989.

[52] Melanie, M., "*An Introduction to Genetic Algorithms*", A Bradford Book, The MIT Press, Massachusetts, 1999.

[53] Chirikjian, G. S., *"Hyper Redundant Manipulator Dynamics: a Continuum Approximation*", Advanced Robotics, Vol. 9, No. 3, pp. 217-243 (1995), VSP and Robotic Society of Japan, 1995.

[54] Shabana, A.A., "*Computational Dynamics*", A Wiley-Interscience Publication, Second Edition, New York, 2001.

[55] Boutin, B. A. and Misra, A. K., "*Dynamics and Control of Variable Geometry Truss Structures*", Acta Astronautica Vol. 45, No. 12, pp. 717-728, Elsevier Science, Great Britain, 1999.

[56] Spanos, P.D., Berka, R. B., and Tratskas, P., "*Multisegment Large Space Robot: Concept and Design*", Journal of Aerospace Engineering, pp. 123-132, October, 2000.

[57] Ercan, Y., "*Akışkan Gücü Kontrol Teorisi*", Gazi Üniversitesi Yayınları, Yayın No: 206, Ankara, 1995.

APPENDIX A

KINEMATICS OF THE FINE TUNING MANIPULATOR (FTM)

The fine tuning manipulator (FTM) has been used as a continuously actuated part of the HRM for fine tuning. The kinematics analysis of this manipulator has been made according to Figure A.1 and the Denavite-Hartenberg parameters as shown in Table A.1.

The transformation matrix from the base frame of the FTM to the end effector is given in exponential form with respect to Table A.1 as follows.

$$\hat{C} = \hat{C}^{(0,6)} = \mathrm{e}^{\tilde{u}_3\theta_1}\mathrm{e}^{\tilde{u}_2\theta_{23}}\mathrm{e}^{\tilde{u}_3\theta_4}\mathrm{e}^{\tilde{u}_2\theta_5}\mathrm{e}^{\tilde{u}_3\theta_6} \tag{A.1}$$

where $\theta_{23} = \theta_2 + \theta_3$

Table A.1 Denavite-Hartenberg Link Parameters for the FTM

Link	Variable	α	a	d
1	θ_1	$-\pi/2$	0	d_1
2	θ_2	0	a_2	d_2
3	θ_3	$\pi/2$	0	0
4	θ_4	$-\pi/2$	0	d_4
5	θ_5	$\pi/2$	0	0
6	θ_6	0	0	0

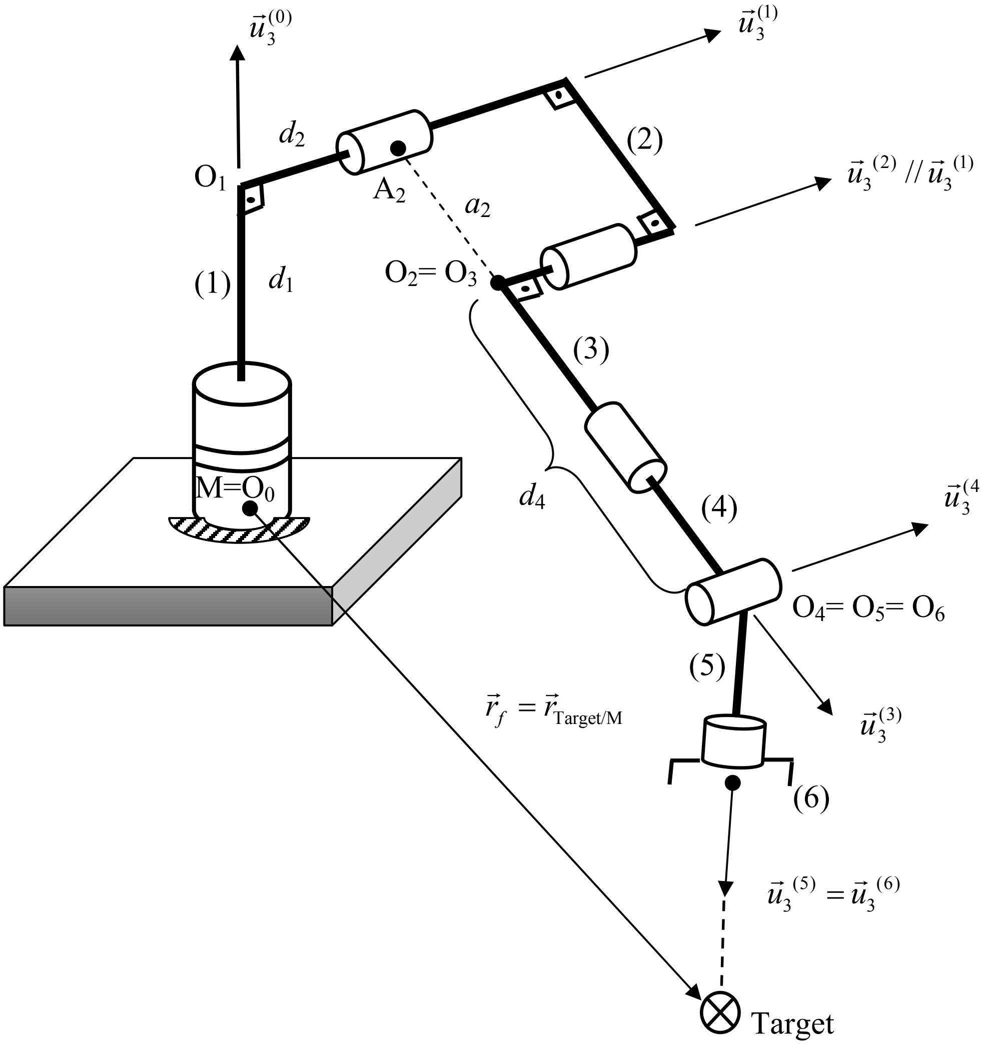


Figure A.1 Schematic View of the Fine Tuning Manipulator and its Joint Variables and Dimensions

The position vector from the base frame of the FTM to the tip point of this manipulator is given in the following.

$$\vec{r}_f = d_1\vec{u}_3^{(0)} + d_2\vec{u}_3^{(1)} + a_2\vec{u}_1^{(2)} + d_4\vec{u}_3^{(3)} \tag{A.2}$$

$$\overline{r}_f = d_1\overline{u}_3^{(0)} + \hat{C}^{(0,1)}\overline{r}_f^{\,*}$$

where $\overline{r}_f^{\,*} = \begin{bmatrix} a_2\cos\theta_2 + d_4\sin\theta_{23} \\ a_2\sin\theta_2 - d_4\cos\theta_{23} \\ d_2 \end{bmatrix}$

The given orientation matrix is $\hat{C}^* = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$.

The translational vector is $\overline{r}_f = \begin{bmatrix} r_{f1} \\ r_{f2} \\ r_{f3} \end{bmatrix}$

From the positional and orientational equations, all the joint variables can be found, in turn, as follows.

$$\theta_1 = \text{atan}\left(-2t_1, 1 - t_1^{\,2}\right) \tag{A.3}$$

where, $t_1 = \dfrac{r_{f1} + \sigma_1\sqrt{r_{f2}^{\,2} - d_2^{\,2} + r_{f2}^{\,2}}}{d_2 + r_{f2}}$, $\sigma_1 = \pm 1$

$$\theta_2 = \text{atan}\left(-2t_2, 1 - t_2^{\,2}\right) \tag{A.4}$$

where, $t_2 = \dfrac{2a_2\left(r_{f3} - d_1\right) + \sigma_2\sqrt{4a_2^{\,2}\left(r_{f3} - d_1\right)^2 - k_2^{\,2} + 4a_2^{\,2}B_1^{\,2}}}{k_2 + 2a_2B_1}$, $\sigma_2 = \pm 1$

$B_1 = r_{f1}\cos\theta_1 + r_{f2}\sin\theta_1$

$$\theta_3 = \text{atan}\left(\zeta_1, \zeta_2\right) - \theta_2 \tag{A.5}$$

where, $\sin\theta_{23} = \dfrac{B_1 - a_2\cos\theta_2}{d_4} = \zeta_1$, $\cos\theta_{23} = \dfrac{r_{f3} - d_1 + a_2\sin\theta_2}{d_4} = \zeta_2$

$$\theta_5 = \text{atan}\left(\sigma_5\sqrt{1 - c_{33}^{*\,2}}, c_{33}^*\right) \tag{A.6}$$

where, $\sigma_5 = \pm 1$ and $\hat{C}^{**} = \mathrm{e}^{-\tilde{u}_2(\theta_2+\theta_3)}\mathrm{e}^{-\tilde{u}_3\theta_1}\hat{C}^{*} = \begin{bmatrix} c_{11}^{*} & c_{12}^{*} & c_{13}^{*} \\ c_{21}^{*} & c_{22}^{*} & c_{23}^{*} \\ c_{31}^{*} & c_{32}^{*} & c_{33}^{*} \end{bmatrix}$

$$\theta_4 = \operatorname{atan}\left(\sigma_5 c_{23}^{*}, \sigma_5 c_{13}^{*}\right) \tag{A.7}$$

$$\theta_6 = \operatorname{atan}\left(\sigma_5 c_{32}^{*}, -\sigma_5 c_{31}^{*}\right) \tag{A.8}$$

$\vec{r}_{\text{Target/M}} = \vec{r}_{\text{Target/0}} - \vec{r}_{\text{M/0}}$: the given positional vector

The given orientation matrix $\hat{C}^{*}$ can be found with respect to 323 Euler angle convention as follows.

$$\hat{C}^{*} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} c\phi\,\mathrm{c}\theta\,\mathrm{c}\psi - s\phi\, s\psi & -c\phi\,\mathrm{c}\theta\, s\psi - s\phi\, c\psi & c\phi\, s\theta \\ s\phi\,\mathrm{c}\theta\,\mathrm{c}\psi + c\phi\, s\psi & -s\phi\,\mathrm{c}\theta s\psi + c\phi\, c\psi & s\phi\, s\theta \\ -s\theta\, c\psi & s\theta\, s\psi & \mathrm{c}\theta \end{bmatrix} \tag{A.9}$$

$$\theta = \operatorname{atan2}\left(\sigma\sqrt{1-c_{33}^{\;2}}, c_{33}\right),\ \sigma = \pm 1 \tag{A.10}$$

$$\phi = \operatorname{atan2}\left(c_{23}, c_{13}\right) \tag{A.11}$$

Where, $\cos\phi = \dfrac{c_{13}}{\sin\theta}$, $\sin\phi = \dfrac{c_{23}}{\sin\theta}$

$$\psi = \operatorname{atan2}\left(c_{32}, -c_{31}\right) \tag{A.12}$$

where, $\cos\psi = \dfrac{c_{31}}{\sin\theta}, \sin\psi = \dfrac{c_{32}}{\sin\theta}$

Equations A.1-12 calculate the required joint variables to compensate the positional and orientational errors of the HRM. The given orientation matrix $\hat{C}^{*}$ can be expressed in terms of the orientation of the last platform of the HRM (included only discrete part of the manipulator) and the FTM in the following.

$$\hat{C}^{*} = \hat{C}^{(0,Target)} = \hat{C}^{(0,HRM)}\hat{C}^{(HRM,Target)} \tag{A.13}$$

Here, $\hat{C}^{(0,Target)}$: the given orientation matrix,

$\hat{C}^{(0,HRM)}$: the orientation matrix of the last platform of the HRM with respect to the initial reference frame,

$\hat{C}^{(HRM,Target)}$: the orientation matrix of the FTM with respect to the last platform of the HRM.

If it is desired that the FTM does minimum work or no work, all joint variables of the FTM can be assigned zero. Thus, the orientation matrix of the FTM becomes an identity matrix. So, the orientation matrix of the discrete part of the HRM becomes equal to the given orientation matrix.

$$\hat{C}^{*} = \hat{C}^{(0,Target)} = \hat{C}^{(0,HRM)}\hat{I} = \hat{C}^{(0,HRM)}$$

APPENDIX B

BEZIER CURVE AND ADDITIONAL FEATURES OF NURBS

There are many types of splines. Here, Bezier curves and NURBS is given with their features.

B.1 Bezier Curve

The basis functions used for Bezier curves are Bernstein polynomials of degree *k*, defined by

$$B_i^k = \binom{k}{i} t^i (1-t)^{k-i}, \qquad i = 0,......k .$$

With the aid of Bernstein polynomials, the definition of a Bezier curve *b(t)* is given as $b(t) = \sum B_i B_i^k(t)$.

The vectors B_i are called Bezier points and they form a Bezier polygon. Figure B.1 shows, for parameter interval $0 \le t \le 1$, an example of a cubic Bezier with corresponding Bezier points and polygon.

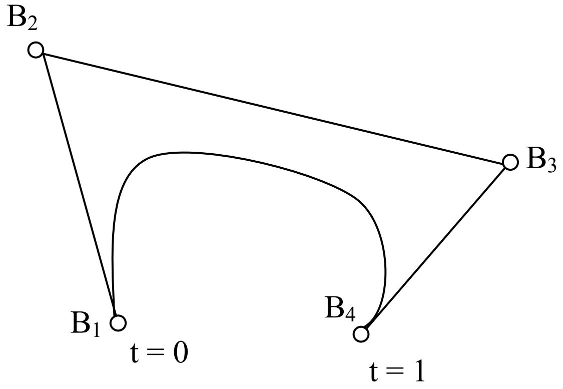


Figure B.1 Bezier Cubic Curve and its Bezier Polygon $B_1B_2B_3B_4$

Analyzing Figure B.1 and additionally the Bezier curve's first and second derivative we notice that:

- The first Bezier curve point coincides with Bezier point B_0 and the last Bezier curve point coincides with Bezier point B_k,
- The Bezier polygon sides B_0B_1 and $B_{k-1}B_k$ are tangent to the Bezier curve,
- The degree k of the Bezier curve is one less than the number of Bezier points k+1,
- The Bezier curve is contained entirely with in convex hull of its Bezier polygon.

Curves consisting of just one polynomial or rational segment are often inadequate. Their shortcomings are:

- a high degree is required in order to satisfy a large number of constraints; e.g., $(n-1)$-degree is needed to pass a polynomial Bezier curve through n data points. However, high degree curves are inefficient to process and are numerically unstable;
- a high degree is required to accurately fit some complex shapes;
- single-segment curves are not well-suited to interactive shape design; although Bezier curve can be shaped by means of their control points (and weights), the control is not sufficiently local.

The solution is to use curves which are piecewise polynomial, or piecewise rational

B.2 Additional Features of NURBS

NURBS is a piecewise rational splines. Here, some features of this spline are presented in the following.

- $N_{i,0}(t)$, which equals to zero everywhere except on the half-open interval $[t_i, t_{i+1})$, is a unit step function;
- For $k > 0$, $N_{i,k}(t)$ is a linear combination of two $(k-1)$-degree basis functions;
- The calculation of a set of basis functions needs specification of a knot vector, T, and the degree, k;
- The $N_{i,k}(t)$ are piecewise polynomials on the entire real line only between $[t_0, t_m]$;
- The half-open interval, $[t_i, t_{i+1})$, is called the i^{th} knot span. It can have zero length, since knots need not be distinct;
- In any given knot span, $[t_j, t_{j+1})$, at most, $k+1$ of the $N_{i,k}(t)$ are non zero, i.e. the functions $N_{j-k,k}, \ldots\ldots\ldots\ldots, N_{j,k}$ are non-zero;
- For an arbitrary knot span $[t_i, t_{i+1})$, $\sum_{i=0}^{n} N_{i,k}(t) = 1$ for all $t \in [0,1]$;

The $R_{ik}(t)$ are the rational basis functions on $t \in [0,1]$. The $R_{i,k}(t)$ have the following properties derived NURBS equation and the corresponding properties of the $N_{i,k}(t)$:

1- $R_{i,k}(t) \geq 0$ for all *i*, *k* and $t \in [0,1]$;

2- $\sum_{i=0}^{n} R_{i,k}(t) = 1$ for all $t \in [0,1]$;

3- $R_{0,k}(0) = R_{n,k}(1) = 1$;

4- For $k > 0$, all $R_{i,k}(t)$ reach exactly one maximum on the interval $t \in [0,1]$.

5- $R_{i,k}(t)=0$ for $t \notin \left(t_i, t_{i+k+1}\right)$. Like the basis functions, in any given knot span, at most, $k+1$ of the $R_{i,k}(t)$ are nonzero ($R_{i\text{-}k,k}(t)$..........$R_{i,k}(t)$ are nonzero in $\left[t_i, t_{i+1}\right)$);

6- At a knot, $R_{i,k}(t)$ is $k-r$ times continously differentiable, where r is the multiplicity of the knot;

7- If w_i are constant for all *i,* then $R_{i,k}(t)=N_{i,k}(t)$ for all *i*; i.e. the $N_{i,k}(t)$ are special cases of the $R_{i,k}(t)$.

8- Properties (1)-(4) yield the following important geometric characteristics of NURBS curves:

8- From (3), $Q(0)=P_0$ and $Q(1)=P_n$;

9- From (1), (2) and (5), if $t \in \left[t_i, t_{i+1}\right)$, then $Q(t)$ lies within the convex hull of the control points P_{i-k},..........,P_i;

10- $Q(t)$ is infinitely differentiable on the interior of knot spans and is $k-r$ times differentiable at a knot of multiplicity *r*;

11- As special cases, A NURBS curve with no interior knots is a rational Bezier curve, since the $N_{i,k}(t)$ reduce to the $B_{i,n}(t)$ (Bernstain function for Bezier curve). This, means that NURBS curves contain nonrational B-Spline and rational and nonrational Bezier curves;

12- From (5), the control point P_i affects only that portion of the curve on the interval $t \in \left[t_i, t_{i+k+1}\right)$ if the control point P_i is moved or the weight w_i is changed;

Property (12) is very important for interactive shape design. Using NURBS curves, both control point movement and weight modification can be used to attain local shape control. Figure 4.5 shows the effect of modifying a single weight. Qualitatively the effect is: assume $t \in \left[t_i, t_{i+k+1}\right)$; then w_i increases (decreases), the point $Q(t)$ moves closer to (farther from) P_i, and hence the curve is pulled towards (pushed away from) P_i. Furthermore, the movement of $Q(t)$ for fixed *t* is along a straight line.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Bayram, Atilla
Nationality: Turkish (TC)
Date and Place of Birth: 20 February 1973 , Van
Marital Status: Married
Phone: +90 312 210 52 78
Fax: +90 312 210 25 39
email: abayram@metu.edu.tr

EDUCATION

Degree	**Institution**	**Year of Graduation**
MS	Yüzüncü Yıl University Mechanical Engineering	2002
BS	Gazi University Mechanical Engineering	1995
High School	Aydınlıkevler High School, Ankara	1991

WORK EXPERIENCE

Year	**Place**	**Enrollment**
2002- Present	METU-Mechanical Engineering	Research Assistant
2000-2002	YYU-Mechanical Engineering	Research Assistant
1998-2000	YYU Erciş MYO	Instructor

PUBLICATIONS

1. Bayram, A. and Özgören, M.K., "Aşırı Artıksıl Bir Manipülatörün Modellenmesi ve Kontrolü", 14th Ulusal Makina Teorisi Sempozyumu, Güzelyurt, KKTC, 2-4 Temmuz 2009.