

4-9-1992

Kinematic and dynamic analyses of general robots by applying the C-B notation-RaMIP (Robot and Mechanism Integrated Program)

Bernardo Donoso
Florida International University

DOI: 10.25148/etd.FI15101216

Follow this and additional works at: <https://digitalcommons.fiu.edu/etd>

 Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Donoso, Bernardo, "Kinematic and dynamic analyses of general robots by applying the C-B notation-RaMIP (Robot and Mechanism Integrated Program)" (1992). *FIU Electronic Theses and Dissertations*. 3079.
<https://digitalcommons.fiu.edu/etd/3079>

This work is brought to you for free and open access by the University Graduate School at FIU Digital Commons. It has been accepted for inclusion in FIU Electronic Theses and Dissertations by an authorized administrator of FIU Digital Commons. For more information, please contact dcc@fiu.edu.

ABSTRACT OF THE THESIS

Kinematic and Dynamic Analyses of General Robots

by Applying C-B Notation - RaMIP (Robot and
Mechanism Integrated Program)

by

Bernardo Donoso

Florida International University, 1992

Miami, Florida

Professor Ta Chung Yih, Major Professor

In this thesis, a new symbolic representation based on 4x4 homogeneous matrices, C-B (Cylindrical Coordinates - Bryant Angles) notation, has been applied to the kinematic and dynamic analyses of general robots, and a computer algorithm named RaMIP (Robot and Mechanism Integrated Program) has been developed on a Sun workstation for the design and analysis of robots and mechanisms. RaMIP can be used to model most industrial robots currently in use. It performs three-dimensional kinematic and dynamic analyses and takes advantage of the computational efficiency of C-B notation. The C-B notation allows the user to model an arbitrary mechanism consisting of any combination of revolute, prismatic and spherical joints. RaMIP has the capability of presenting results in the form of two- and three-dimensional plots of colored contours, as well as tables of numerical data. The algorithm is examined and tested by analyzing several commercial robots. Kinematic and dynamic results are computed and presented in two- and three-dimensional graphs and compared with known data to probe the validity and accuracy of RaMIP. It should be noticed that the efforts completed in this thesis present only the first step towards the implementation of a general purpose computer algorithm -RaMIP- for the automated design and analysis of open- and closed-chain mechanisms utilizing C-B notation.

FLORIDA INTERNATIONAL UNIVERSITY
Miami, Florida

Kinematic and Dynamic Analyses of General Robots
by Applying the C-B Notation - RaMIP (Robot and
Mechanism Integrated Program)

A thesis submitted in partial satisfaction of the
requirements for the degree of Master of Science
in Mechanical Engineering

by

Bernardo Donoso

1992

To Professors:

Dr. M. A. Ebadian

Dr. C. Levy

Dr. I. N. Tansel

Dr. T. C. Yih

This thesis, having been approved in respect to form and mechanical execution, is referred to you for judgement upon its substantial merit.

Dr. Gordon Hopkins
College of Engineering and Design

The thesis of Bernardo Donoso is Approved.

Dr. M. A. Ebadian (chairperson)

Dr. C. Levy

Dr. I. N. Tansel

Dr. T. C. Yih (Major Professor)

Date of Examination: April 9th 1992

Richard L. Campbell
Division of Graduate Studies

Florida International University, 1992

To my wife Margarita, to my children to be, and to my father and mother.

ACKNOWLEDGEMENTS

It is a great pleasure for me to have the opportunity to thank everyone who helped me during my graduate studies and this thesis work.

I would like to thank the members of the thesis committee, Dr. Cesar Levy, Dr. Ian Tansel, and Dr. M. A. Ebadian, for their support and help in reviewing the original manuscript. Special thanks to Dr. Ian Tansel for his interest in the present work, and to Dr. Cesar Levy for his dedication to the correction of this thesis.

I am deeply thankful to Dr. Ta Chung Yih for his invaluable support and guidance throughout the master's program. I would like to thank him for opening the doors to graduate studies for me, for the numerous opportunities for professional development he has given me, for providing the topic of this thesis and the central ideas for many other research projects, and for orchestrating the financial support necessary for my studies. His influence has redirected my engineering career. I am fortunate for having come in contact with Dr. T.C. Yih, and I look forward to collaborating with him in future research work.

I would like to express my sincere appreciation to the Department of Mechanical Engineering, Florida International University.

I owe a debt of gratitude to my wife Margarita, for her continuous encouragement, patience, and love.

TABLE OF CONTENTS

CHAPTER 1	1
1.1 Introduction	1
1.2 Background	1
1.3 Objectives and Significance	5
CHAPTER 2	7
COMPUTER MODELING AND KINEMATIC ANALYSIS	7
2.1 Introduction	7
2.2 Review of C-B notation	8
2.2.1 Cylindrical Coordinates and Bryant Angles Transformations ..	8
2.2.2 C-B Notation	9
A) Reference Frame	10
B) Principal Joint Parameters	11
C) Characteristic Matrices of Kinematic Pairs	12
i) Revolute Pair (R)	13
ii) Prismatic Pair (P)	13
iii) Spherical Pair (S)	13
2.3 Kinematics	15
2.3.1 Position Analysis	16
2.3.2 Separation of Variables in Matrix	16
2.3.3 Velocity Analysis	18
2.3.4 Acceleration Analysis	19
2.4 Kinematic Spaces	21
2.4.1 Graphic Representation	21
2.4.2 Numerical Examples	22

Example 2.1 - Cincinnati Milacron T3 robot (RRR/RRR) .	23
Example 2.2 - Bendix AA/CNC Industrial Robot (RRP/RRR)	29
Example 2.3 - Unimate 2000 Spherical Robot	36
Example 2.4 - IBM 7576 SCARA Robot (RRPR)	41
Example 2.5 - Space Shuttle RMS (Remote Manipulator System)	47
2.4.3 Verification of Results	53
CHAPTER 3	54
DYNAMIC ANALYSIS	54
3.1 Introduction	54
3.2 Static Analysis	56
3.2.1 Relative Joint Vectors	57
3.3 Dynamic Analysis	59
3.3.1 Newton-Euler Formulation	60
A) Revolute Pair	61
B) Prismatic Pair	62
C) Spherical Pair	63
3.3.2 General Matrix Notation	63
3.3.3 Numerical Examples	66
Example 3.1 - Cincinnati Milacron T3 Robot (RRR/RRR) .	66
Example 3.2 - Bendix AA/CNC Industrial Robot (RRP/RRR)	71
Example 3.3 - Unimate 2000 Spherical Robot (SP/RRR) . .	76
Example 3.4 - IBM 7576 SCARA Robot (RRPR)	81
Example 3.5 - Space Shuttle RMS Manipulator	

(RRR/RRR)	86
3.3.4 Discussion of Results	91
CHAPTER 4	93
RaMIP (<u>R</u> obot and <u>M</u> echanism <u>I</u> ntegrated <u>P</u> rogram)	93
4.1 Introduction	93
4.2 General Computational Procedures	94
4.2.1 Input Data	94
A) Kinematic Input Data	94
B) Dynamic Input Data	95
C) Data Structures and Subroutines	95
4.2.2 Algorithm Organization and Solution Procedures	101
4.2.3 Output of Results	108
CHAPTER 5	109
CONCLUSIONS AND FUTURE PERSPECTIVE	109
REFERENCES	112
APPENDIX	119
ROBOT KINEMATIC AND DYNAMIC DATA	119

LIST OF FIGURES

Figure 2.1 C-B (<u>C</u> ylindrical coordinates- <u>B</u> ryant angles) notation	10
Figure 2.2 Revolute Pair (R)	12
Figure 2.3 Prismatic Pair (P)	14
Figure 2.4 Spherical joint parameters	15
Figure 2.5a Cincinnati Milacron T3, XZ Workspace Projection	24
Figure 2.5b Cincinnati Milacron T3, YZ Workspace Projection	25
Figure 2.6a Cincinnati Milacron T3, XZ Velocity Space Projection	25
Figure 2.6b Cincinnati Milacron T3, YZ Velocity Space Projection	26
Figure 2.7a Cincinnati Milacron T3, XZ Acceleration Space Projection	26
Figure 2.7b Cincinnati Milacron T3, YZ Acceleration Space Projection	27
Figure 2.8 Cincinnati Milacron T3 Workspace, (viewp. (2,-0.5,0.3), sect. 30°-120°)	27
Figure 2.9 Cincinnati Milacron T3 Workspace, (viewp. (3,0.5,0) sect. 30°-120°)	28
Figure 2.10 Cincinnati Milacron T3 Average Vel. Shaded Workspace, (viewp. (2,- 0.75,0))	28
Figure 2.11 Cincinnati Milacron T3 Acceleration Space, (viewp. (1,1,0.3) section 30 to 120 deg.)	29
Figure 2.12a Bendix AA/CNC, XZ Workspace Projection	30
Figure 2.12b Bendix AA/CNC, YZ Workspace Projection	31
Figure 2.13a Bendix AA/CNC, XZ Velocity Space Projection	31
Figure 2.13b Bendix AA/CNC, YZ Velocity Space Projection	32
Figure 2.14a Bendix AA/CNC, XZ Acceleration Space Projection	32
Figure 2.14b Bendix AA/CNC, YZ Acceleration Space Projection	33
Figure 2.15 Bendix AA/CNC Workspace (viewpt. (.75,.75,.3) sect. 0°-95°)	33
Figure 2.16 Bendix AA/CNC Velocity Space (viewpt. (.75,.75,.3) sect. 0-95 deg.	34

Figure 2.17 Bendix AA/CNC Vz Shaded Workspace (viewpt. (.75,.75,.3) sect 0°-95°)	34
Figure 2.18 Bendix AA/CNC Acceleration Space (viewpt. (.75,.75,.3) sect. 0-95 deg. . .	35
Figure 2.19 Bendix AA/CNC Ay Shaded Workspace (viewpt. (.75,.75,.3) sect. 0°-95°)	35
Figure 2.20a Unimate 2000, XZ Workspace Projection	37
Figure 2.20b Unimate 2000, YZ Workspace Projection	37
Figure 2.21a Unimate 2000, XZ Velocity Space Projection	38
Figure 2.21b Unimate 2000, YZ Velocity Space Projection	38
Figure 2.22a Unimate 2000, XZ Acceleration Space Projection	39
Figure 2.22b Unimate 2000, YZ Acceleration Space Projection	39
Figure 2.23 Unimate 2000, Workspace (viewp. (1,-1,0.3) sect. 0°-104°)	40
Figure 2.24 Unimate 2000, Vy Shaded Workspace (viewp. (0.75,-1,0.4) sect. 0°-104°)	40
Figure 2.25 Unimate 2000, Acceleration Space (viewp. (1,-2,0.5) sect 0°-104°)	41
Figure 2.26 IBM 7576 Robot Workspace Plane Projections. a) XZ, b) YZ, c) XY	43
Figure 2.27a IBM 7576 Robot XZ Velocity Space Projection	44
Figure 2.27b IBM 7576 Robot YZ Velocity Space Projection	44
Figure 2.28a IBM 7576 Robot XZ Acceleration Space Projection	45
Figure 2.28b IBM 7576 Robot YZ Acceleration Space Projection	45
Figure 2.29 IBM 7576 Robot Workspace (viewp. (-1,-1,0.5) section 0°-120°)	46
Figure 2.30 IBM 7576 Robot Vy Shaded Workspace (viewp. (-1,-1,0.5) sect. 0°-120°)	46
Figure 2.31 IBM 7576 Robot Ay Shaded Workspace (viewp. (-1,-1,0.5) sect. 0°-	

120°)	47
Figure 2.32a RMS Manipulator XZ Workspace Projection	48
Figure 2.32b RMS Manipulator YZ Workspace Projection	49
Figure 2.33a RMS Manipulator XZ Velocity Space Projection	49
Figure 2.33b RMS Manipulator YZ Velocity Space Projection	50
Figure 2.34a RMS Manipulator XZ Acceleration Space Projection	50
Figure 2.34b RMS Manipulator YZ Acceleration Space Projection	51
Figure 2.35 RMS Manipulator Workspace (viewp. (1,-1,0.2) sect. 0°-180°)	51
Figure 2.36 RMS Manipulator Velocity Space (viewp. (1,1,0.6) sect. 0°-180°)	52
Figure 2.37 RMS Manipulator Acceleration Space (viewp. (1,-0.3,0.5) sect. 0°-180° . . .	52
Figure 2.38 RMS Manipulator A_y Shaded Workspace (viewp. (1,-0.3,0.5) sect. 0°- 90°)	53
Figure 3.1 Relative position vectors on the i-th link	58
Figure 3.2 Static free-body diagram of the i-th link	58
Figure 3.4a Cincinnati Milacron T3 robot, joint force x-component versus elbow joint θ_3 angle	67
Figure 3.4b Cincinnati Milacron T3 robot, joint force y-component versus elbow joint θ_3 angle	67
Figure 3.4c Cincinnati Milacron T3 robot, joint force z-component versus elbow joint θ_3 angle	68
Figure 3.4d Cincinnati Milacron T3 robot, joint resultant force (F_{res}) versus elbow joint angle θ_3 angle	68
Figure 3.5a Cincinnati Milacron T3 robot, joint moment x-component versus elbow joint θ_3 angle	69
Figure 3.5b Cincinnati Milacron T3 robot, joint moment y-component versus elbow joint θ_3 angle	69

Figure 3.5c Cincinnati Milacron T3 robot, joint moment z-component versus elbow joint θ_3 angle	70
Figure 3.5d Cincinnati Milacron T3 robot, joint resultant (M_{res}) moment versus elbow joint θ_3 angle	70
Figure 3.6a Bendix AA/CNC robot, joint force x-component versus elbow joint h_3 position	72
Figure 3.6b Bendix AA/CNC robot, joint force y-component versus elbow joint h_3 position	72
Figure 3.6c Bendix AA/CNC robot, joint force z-component versus elbow joint h_3 position	73
Figure 3.6d Bendix AA/CNC robot, joint resultant force (F_{res}) versus elbow joint h_3 position	73
Figure 3.7a Bendix AA/CNC robot, joint moment x-component versus elbow joint h_3 position	74
Figure 3.7b Bendix AA/CNC robot, joint moment y-component versus elbow joint h_3 position	74
Figure 3.7c Bendix AA/CNC robot, joint moment z-component versus elbow joint h_3 position	75
Figure 3.7d Bendix AA/CNC robot, joint resultant moment (M_{res}) versus elbow joint h_3 position	75
Figure 3.8a Unimate 2000 spherical robot, joint force x-component versus base joint ϕ_1 angle	77
Figure 3.8b Unimate 2000 spherical robot, joint force y-component versus base joint ϕ_1 angle	77
Figure 3.8c Unimate 2000 spherical robot, joint force z-component versus base joint ϕ_1 angle	78

Figure 3.8d Unimate 2000 spherical robot, joint resultant force (F_{res}) versus base joint ϕ_1 angle	78
Figure 3.9a Unimate 2000 spherical robot, joint moment x-component versus base joint ϕ_1 angle	79
Figure 3.9b Unimate 2000 spherical robot, joint moment y-component versus base joint ϕ_1 angle	79
Figure 3.9c Unimate 2000 spherical robot, joint moment z-component versus base joint ϕ_1 angle	80
Figure 3.9d Unimate 2000 spherical robot, joint resultant moment (M_{res}) versus base joint ϕ_1 angle	80
Figure 3.10a IBM 7576 SCARA robot, joint force x-component versus shoulder joint θ_2 angle	82
Figure 3.10b IBM 7576 SCARA robot, joint force y-component versus shoulder joint θ_2 angle	82
Figure 3.10c IBM 7576 SCARA robot, joint force z-component versus shoulder joint θ_2 angle	83
Figure 3.10d IBM 7576 SCARA robot, joint resultant force (F_{res}) versus shoulder joint θ_2 angle	83
Figure 3.11a IBM 7576 SCARA robot, joint moment x-component versus shoulder joint θ_2 angle	84
Figure 3.11b IBM 7576 SCARA robot, joint moment y-component versus shoulder joint θ_2 angle	84
Figure 3.11c IBM 7576 SCARA robot, joint moment z-component versus shoulder joint θ_2 angle	85
Figure 3.11d IBM 7576 SCARA robot, joint resultant moment (M_{res}) versus shoulder joint θ_2 angle	85

Figure 3.12a Space Shuttle RMS robot, joint force x-component versus shoulder joint θ_2 angle	87
Figure 3.12b Space Shuttle RMS robot, joint force y-component versus shoulder joint θ_2 angle	87
Figure 3.12c Space Shuttle RMS robot, joint force z-component versus shoulder joint θ_2 angle	88
Figure 3.12d Space Shuttle RMS robot, joint resultant force (F_{res}) versus shoulder joint θ_2 angle	88
Figure 3.13a Space Shuttle RMS robot, joint moment x-component versus shoulder joint θ_2 angle	89
Figure 3.13b Space Shuttle RMS robot, joint moment y-component versus shoulder joint θ_2 angle	89
Figure 3.13c Space Shuttle RMS robot, joint moment z-component versus shoulder joint θ_2 angle	90
Figure 3.13d Space Shuttle RMS robot, joint resultant moment (M_{res}) versus shoulder joint θ_2 angle	90
Main menu of RaMIP	102
Kinematics main menu	103
2-D graphics display	104
3-D graphics display	105
Dynamic Analysis	106
Graphics display of dynamic data	107

LIST OF TABLES

Table 2.1 Comparisons among D-H, S-U, and C-B notations	8
Table 2.2. Cincinnati Milacron T3 Curve Parameters	24
Table 2.3 Bendix AA/CNC Curve Parameters	30
Table 2.4 Unimate 2000 Curve Parameters	36
Table 2.5 IBM 7576 SCARA Robot Curve Parameters	42
Table 2.6 RMS Robot Curve Parameters	48
Table 4.1 Algorithm Data Structures	96
Table A.1a Kinematic Parameters for Cincinnati Milacron T3 Robot	119
Table A.1b Dynamic Parameters for Cincinnati Milacron T3 Robot	119
Table A.2a Kinematic Parameters for Bendix AA/CNC Robot	120
Table A.2b Dynamic Parameters for Bendix AA/CNC Robot	120
Table A.3a Kinematic Parameters for Unimate 2000 Spherical Robot	121
Table A.3b Dynamic Parameters for Unimate 2000 Spherical Robot	121
Table A.4a Kinematic Parameters for IBM 7576 (SCARA) Robot	122
Table A.4b Dynamic Parameters for IBM 7576 (SCARA) Robot	122
Table A.5a Kinematic Parameters for Space Shuttle (RMS) Robot	122
Table A.5b Dynamic Parameters for Space Shuttle (RMS) Robot	123

CHAPTER 1

1.1 Introduction

With the increasing needs for higher productivity, improved product quality, and lower cost, industry is turning toward computer-based automation. Unlike conventional manufacturing systems (fixed automation), characterized by mass production of non-changing products during the years after World War II, modern manufacturing systems require flexibility in order to meet frequent changes in product models and schedules. Contemporary industrial robots are the flexible reprogrammable machines that have come to fulfill such need. The first industrial robots were developed during the 50's, but it was not until the 60's, that industrial robotic automation took place. The first robots were developed for repetitive and tedious tasks such as pick and place, spot/arc welding, spray painting, and so on. Since then, robots have been adopted into the industrial mainstream by most industrial countries in the world. According to estimates by Kafrissen and Stephans (1984), in 1982, there was on the order of 6,000 robots and robotic devices in use in the American industry, 5,000 in Europe, and 30,000 in Japan. By 1985, the number of robots in the U. S. will increase to 20,000. By 1995, the U. S. automated manufacturer and the plastic processors will use around 200,000 industrial robots and robotic devices.

1.2 Background

As a consequence of the proliferation of computers and the need for improved, more efficient manufacturing processes, and faster and more powerful analytical methods, CAD/CAM (Computer Aided Design/Computer Aided Manufacturing) , CAE (Computer Assisted Engineering), computer simulation and modeling, and other computer techniques, have become indispensable tools in industry as well as in all fields of

science. In the field of mechanisms and robotics, the development of computer software packages capable of solving kinematic and/or dynamic problems of mechanical systems were started since 1950. Among them are, ADAMS (Automated Dynamic Analysis of Mechanical Systems) by Orlandea and Chace (1977), DADS (Dynamic Analysis and Design System) by Wehage and Haug (1982), IMP (Integrated Mechanisms Program) by Sheth and Uicker (1972), DRAM (Dynamic Response of Articulated Machinery) by Chace and Smith (1971), LINCAGES (LINKages Computer Analysis, and Graphically Enhanced Synthesis), DYMES (DYnamics of MEchanical Systems), DYNAPAC by Algor, and others. Besides these mechanisms programs, many FEA (finite Elements Analysis) codes such as ANSYS, MSC/NASTRAN, SDRC-IDEAS, have dynamic and modal analysis capabilities.

Kinematics is the science of motion which treats motion without regard to the forces which cause it. It is the fundamental science within which one studies the position, velocity, acceleration, and all higher order derivatives of the position variables. Kinematics is divided into direct and inverse kinematics. Direct kinematics studies the relative motions among the body segments (links) of the system. Basically, its final goal is to determine position, velocity, and acceleration, based on prescribed geometric specifications of the system. Inverse kinematics on the other hand, has as its objectives, to determine the system geometric parameters given function, trajectory, velocity, etc.

Research on the open-chain mechanism has been emphasized during the last 10 years due to its role in automated manufacturing. Roth (1975) first defined the performance evaluation of manipulators by the term "workspace". He studied the motion of manipulators in terms of points and sets of axes. Roth and Shimano (1977) applied an iterative method to describe the reachable workspace, and concluded that the normal distance between any two axes is maximized when the common normal between the them

simultaneously intersects each of the intermediate axis of rotation. Tsai and Soni (1981) solved the accessible region for planar two link robot arms in terms of equivalent area, and developed an algorithm to observe the workspace for n-R robots. Lee and Yang (1983) presented an analytical investigation of the characteristics and shape of workspace, and the existence of holes and voids in workspace. They also introduced the manipulator performance index, and found that for a given manipulator structure, the ratio of the volume of the workspace to the cube of its total link length is a constant. Kohli and Spanos (1985) developed a new method for the investigation of a manipulator workspace based on polynomial discriminants.

Algorithms for determining extreme distances between reference point and end effector of a robot arm were developed by Kumar and Waldron (1981), and by Duffy and Sugimoto (1981). Kumar and Waldron developed their algorithm by applying a wrench of zero pitch about an axis through the base point in the end effector. Sugimoto and Duffy developed an algorithm with a series of revolute or turning pairs, based on the theorem that all intermediate joint axes of a robot arm with an arbitrary number of joints, intersect an extreme distance line between an arbitrary base point and the center point of the hand. Recently, dexterity of robot manipulators were studied by Waldron (1985), Yang (1985), and Hunt and Davidson (1988). Analysis of parallel robots were also studied by Hunt (1983), Sugimoto (1986, 1989), and by Sugimoto and Hara (1989).

Dynamics is a field devoted to studying the forces required to cause motion. The dynamic analysis , also known as the kinetic analysis, is concerned with the position , velocity, and acceleration resulting from applied external forces and moments. The inverse dynamic analysis, usually referred to as the dynamic analysis in solid mechanics, has as its aim to determine the forces and moments required to produce a given set of positions, velocities, and accelerations. Essentially, dynamic equations are needed for the dynamic

control of manipulators. The objective of dynamic control is to maintain a proper dynamic response of a computer-based manipulator, so that the response is in accord with the prescribed objectives or goals (positions, speeds, etc) in what is basically a feedback process. There are essentially three approaches available to arrive at a set of governing coupled severely nonlinear differential equations describing the dynamic behavior of a manipulator: a) Bond Graph (Shahinpoor, 1987), b) Newton-Euler (Orin, McGhee, Vukobratovic, and Hartock, 1979, Townsend and Gupta, 1989), and c) Lagrange-Euler dynamic modeling (Li, 1988). In addition to these techniques, two recursive techniques have been developed. One is the Newton-Euler recursive approach, and the other is the Lagrangian recursive approach. These two techniques have the advantage of reducing drastically the number of computations, by the use of recurrence relationships for the velocities, accelerations, and generalized forces.

The already mentioned methods are also subdivided into two approaches. In the approach utilized by Orlandea (1977), and Wehage (1981), the configuration of the system is identified using a set of Cartesian coordinates that describe the positions and orientations of the bodies in the system. This approach essentially assumes rigid body links. The method has the advantages of its easier formulation of dynamic equations of motion, allowing easy additions of constraint equations and force functions, and only six degrees of freedom are needed to uniquely describe the body configuration. A second approach utilized to model multi-body systems with elastic links (Shabana and Wehage, 1983, Chang and Shabana 1990) presents more complications. Two sets of coordinate systems are used. A Cartesian coordinate system defines the position and orientation of a body reference, and a second coordinate system describes the elastic deformation of the body with respect to the first coordinate system. This approach is more complex in its formulation, and the relative coordinates and their time derivatives are not easily

available.

Luh, Walker, and Paul (1980), pointed out the importance of the inverse dynamics in control applications. By utilizing the Newton-Euler recursive formulation, the link velocities and accelerations were solved from the base to the end link by kinematic analysis, and then these were utilized to solve joint torques from the end link to the base of the manipulator recursively. A similar recursive approach using the Lagrangian formulation was used by Hollerbach (1980) in which 3 x 3 matrices instead of 4 x 4 homogeneous transformation matrices (Uicker, 1965) were used to increase computational efficiency. The Lagrangian recursive approach was utilized by Book (1984) to formulate the equations of motion of a flexible multilink manipulator. He used a mixed set of relative joint variables and the modal elastic degrees of freedom (d.o.f) to define the system configuration space.

1.3 Objectives and Significance

An open-chain mechanism, commonly known as a "robot" or "manipulator", is a mechanical system which consists of nearly rigid links interconnected with joints that allow relative motion of neighboring links. Link and joint geometry and relationships determine the way the mechanism behaves. The relationships and behavior of links and joints can be modeled by a symbolic notation such as C-B notation.

C-B notation (Yih, 1991) is an advanced 4x4 homogeneous matrix method developed to model and analyze spatial robots and mechanisms. This matrix notation was developed by applying Cylindrical coordinates and Bryant angles transformation matrices, thus, referred to as C-B notation. One of its advantages is that C-B notation defines absolute (with respect to a global coordinate system) robot joint positions, unlike the most commonly used D-H notation formulated by Denavit and Hartenberg (1955), which

provides only relative positions if two joint axes are not parallel or perpendicular to each other. A similar argument can be put forth against S-U notation formulated by Sheth and Uicker.

The objectives of this thesis are; first, utilize C-B (Cylindrical Coordinates - Bryant Angles) notation for the modeling and kinematic analysis of general open-chain mechanisms-robots. Second, combine the C-B notation and Newton-Euler formulation, into a matrix solution for the dynamic analysis of general open-chain mechanisms. Finally, develop a computer algorithm (RaMIP - Robot and Mechanism Integrated Program) capable of doing automated design and analysis of general mechanisms.

CHAPTER 2

COMPUTER MODELING AND KINEMATIC ANALYSIS

2.1 Introduction

Industrial robots generally consist of a mechanical unit (body, arm, wrist) to which an end effector (gripper, spot welder, drill) is fixed, a power source (hydraulic, electric, pneumatic), and a control unit to provide logical directions. The mechanical unit (body, arm, wrist) in a robot is what is usually called "robot" or "manipulator" which consists of a series of links and joints interconnected in a specific order to produce controlled movements in various directions. Computer modeling of a robot is accomplished by using a symbolic notation.

A symbolic notation permits a shorthand representation of a mechanical system, and contains the essential parameters for a complete description of the system. Based on the generalized notation, transformation matrices are developed to model link shapes and constraint motions of joints. The first symbolic notation and systematic approach for the modeling and analysis of mechanical systems was attempted by Denavit and Hartenberg (1955). They employed only four parameters (a , α , β , s) to describe the shape and joint characteristics of each link. The resulting D-H notation has been used extensively in the analysis of robot manipulators, as well as mechanisms. Due to its basic formulation, D-H notation only defined relative joint positions for certain mechanism geometries, thus introducing some complications to the analysis. A modified symbolism called S-U notation, was developed by Sheth and Uicker (1972). Its purpose was to improve and extend the use of D-H notation, and to determine the exact joint positions in space. S-U notation introduced six constant joint parameters (a , b , c , α , β , γ) plus a variable part that contains the same number of parameters as the degrees of freedom of the joint.

2.2 Review of C-B notation

Developed by Yih (1991), C-B notation is formulated based on homogeneous cylindrical coordinates and Bryant angles transformation matrices, and thus termed "C-B notation". Unlike D-H notation, C-B notation permits determination of exact joint positions in space. Also, it uses 5 parameters ($\theta, h, r, \alpha, \beta$) to describe the link shape and joint behavior, instead of six plus the joint degrees of freedom employed by S-U notation. As a consequence, mathematical complexity in computer modeling has been reduced and computational efficiency has been improved as well. A comparison among the D-H, S-U and C-B notations is listed in Table 2.1.

Table 2.1 Comparisons among D-H, S-U, and C-B notations

Notation	D-H	S-U	C-B
Means of transformations	Cartesian	Cartesian	Cylindrical + Bryant angles
Number of parameters	4 (a, s, α, β)	6+ (a, b, c, α, β, γ +joint d.o.f.)	5 ($\theta, h, r, \alpha, \beta$)
Reference frame defined	unique	non-unique	unique
Definition of joint position for concurrent (parallel or normal) joint axes	exact (6)*	exact (?)	exact (6)*
Definition of joint position for non-concurrent (3-D) joint axes	relative (N/A)*	exact (30+)	exact (22)

* Number of mathematical operations (+ & *) in shape matrix

2.2.1 Cylindrical Coordinates and Bryant Angles Transformations

The homogeneous cylindrical coordinate transformation matrix $T_c(\theta, h, r)$ is

derived as

$$T_c(\theta, h, r) = T_r(Z, \theta) T_t(Z, h) T_t(X, r) = \begin{bmatrix} c\theta & -s\theta & 0 & rc\theta \\ s\theta & c\theta & 0 & rs\theta \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$T_r(Z, \theta)$, $T_t(Z, h)$, and $T_t(X, r)$ are transformation matrices representing a rotation θ about the Z axis, a translation h along the Z axis, and a translation r along the X axis. Also, $s\theta = \sin\theta$ and $c\theta = \cos\theta$.

The Bryant angles convention considers rotations counterclockwise, in sequence, about the X, Y, and Z axes through angles α , β , and γ , respectively. Its homogeneous matrix $T_b(\alpha, \beta, \gamma)$ is thus

$$T_b(\alpha, \beta, \gamma) = T_r(X, \alpha) T_r(Y, \beta) T_r(Z, \gamma) = \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta & 0 \\ c\alpha s\gamma + s\alpha s\beta c\gamma & c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta & 0 \\ s\alpha s\gamma - c\alpha s\beta c\gamma & s\alpha c\gamma + c\alpha s\beta s\gamma & c\alpha c\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

where $T_r(X, \alpha)$, $T_r(Y, \beta)$, and $T_r(Z, \gamma)$ represent rotations about the X, Y, and Z axes respectively.

2.2.2 C-B Notation

The geometrical shape of a spatial linkage (Fig 2.1) can be described using the

transform notations of cylindrical coordinates and Bryant angles.

To transform the joint reference frame from $X_i-Y_i-Z_i$ to $X_{i+1}-Y_{i+1}-Z_{i+1}$, first, $X_i-Y_i-Z_i$ system is transformed to $\hat{e}_{x_i}-\hat{e}_{y_i}-\hat{e}_{z_i}$ in the sequence (θ_i, h_i, r_i) , which corresponds to the cylindrical coordinates transformations. Second, $\hat{e}_{x_i}-\hat{e}_{y_i}-\hat{e}_{z_i}$ is transformed to the resultant $X_{i+1}-Y_{i+1}-Z_{i+1}$ through Bryant angles $(\alpha_i, \beta_i, \gamma_i)$, with $\gamma_i = 0^\circ$.

The joint reference frame $(X_i-Y_i-Z_i)$ and principal joint parameters $(\theta_i, h_i, r_i, \alpha_i, \beta_i)$ for the i -th joint J_i are defined as follows.

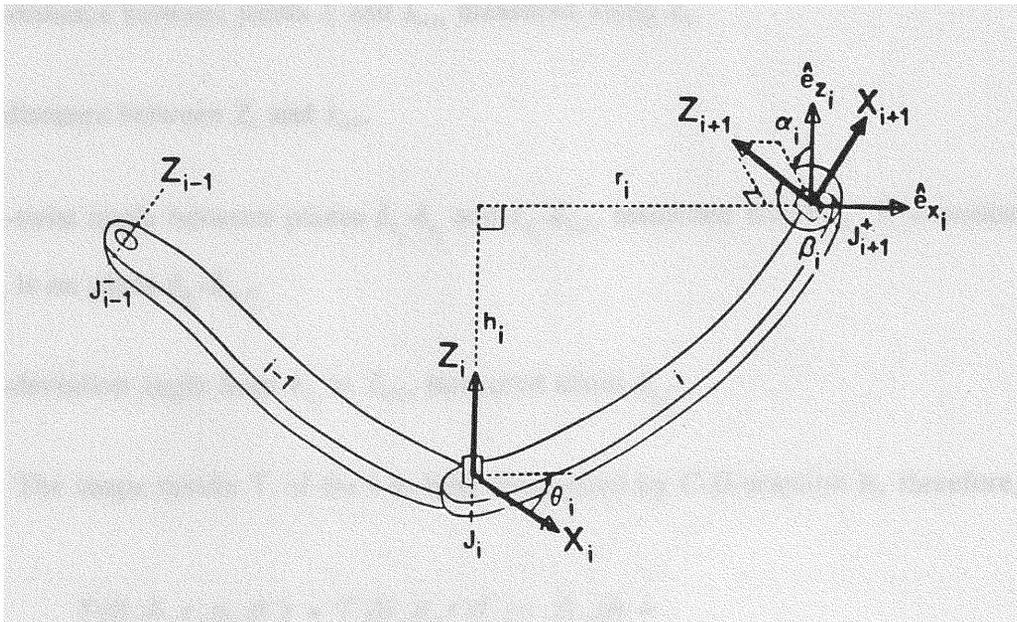


Figure 2.1 C-B (Cylindrical coordinates-Bryant angles) notation.

A) Reference Frame

The local origin of the i -th reference frame is chosen at the physical joint center of J_i , and:

Z_i -along the i th joint axis in the designated direction.

X_i -normal to Z_i in the plane Z_i - X_{i-1} . As Z_i is parallel to X_{i-1} , X_i and X_{i-1} are perpendicular to each other. Usually, X_i is chosen to be in the same direction as X_{i-1} .

Y_i -in the direction to form the right handed cartesian coordinate system.

B) Principal Joint Parameters

θ_i -rotation angle of the i th joint, measured about Z_i , as the angle between planes Z_i - X_{i-1} and Z_i - X_i .

h_i -distance between joints J_i and J_{i+1} , measured along Z_i .

r_i -distance between Z_i and J_{i+1} .

α_i -twist angle between planes \hat{e}_{x_i} - \hat{e}_{z_i} and \hat{e}_{x_i} - Z_{i+1} , measured about \hat{e}_{x_i} . It is noticed that X_{i+1} is on plane \hat{e}_{x_i} - Z_{i+1} .

β_i -deviation angle from \hat{e}_{z_i} to Z_{i+1} , measured about \hat{e}_{y_i} .

The shape matrix T_i of the i -th link, formulated by C-B notation is, therefore,

$$T_i(\theta_i, h_i, r_i, \alpha_i, \beta_i) = T_c(\theta_i, h_i, r_i)T_b(\alpha_i, \beta_i, 0) =$$

$$= \begin{bmatrix} c\theta_i c\beta_i - s\theta_i s\alpha_i s\beta_i & -s\theta_i c\alpha_i & c\theta_i s\beta_i + s\theta_i s\alpha_i c\beta_i & r_i c\theta_i \\ s\theta_i c\beta_i + c\theta_i s\alpha_i s\beta_i & c\theta_i c\alpha_i & s\theta_i s\beta_i - c\theta_i s\alpha_i c\beta_i & r_i s\theta_i \\ -c\alpha_i s\beta_i & s\alpha_i & c\alpha_i c\beta_i & h_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

in which the orientation and position of $X_{i+1}-Y_{i+1}-Z_{i+1}$, relative to $X_i-Y_i-Z_i$, can be expressed by the direction cosine matrix D_i and position vector P_i as

$$T_i(\theta, h, r, \alpha, \beta, \delta) = \begin{bmatrix} D_i & P_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

C) Characteristic Matrices of Kinematic Pairs

A characteristic matrix describes not only the shape of the link, but also its behavior constrained by kinematic pairs. The three kinematic pairs that have been implemented in the algorithm are: R (revolute), P (prismatic), and S (spherical) joints.

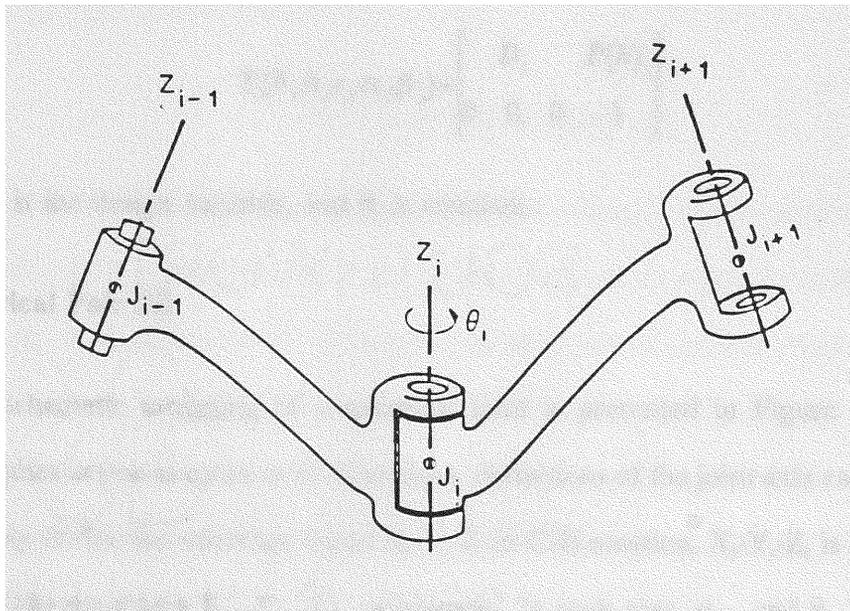


Figure 2.2 Revolute Pair (R)

i) Revolute Pair (R)

A revolute pair (Fig 2.2) is characterized by its axial rotation θ_i , whose characteristic matrix has the same expression as equation 2.3.

$$T_i(\theta_i, h_i, r_i, \alpha_i, \beta_i) = \begin{bmatrix} D(\theta)_i & P(\theta)_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

where θ_i is the only design variable.

ii) Prismatic Pair (P)

A prismatic pair allows linear motion along only one axis (Fig 2.3). Thus, its characteristic matrix is

$$T_i(\theta_i, h_i, r_i, \alpha_i, \beta_i) = \begin{bmatrix} D_i & P(h)_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

where h_i is the design variable, and θ_i is constant.

iii) Spherical Pair (S)

The schematic modeling of a spherical joint is presented in Figure 2.4. Since a spherical joint is free to move in all directions, definitions of the joint axis can be infinite. To uniquely define the reference frame X_i - Y_i - Z_i in C-B notation, X_i - Y_i - Z_i is chosen to be parallel to the preceding X_{i-1} - Y_{i-1} - Z_{i-1} coordinates. In such case, α_{i-1} , and β_{i-1} are equal to zero.

Referring to Figure 2.4, the relative link length a_i , between J_i and J_{i+1} remains constant at all times. However, parameters h_i and r_i vary with angle ϕ_i , measured from \vec{Z}_i

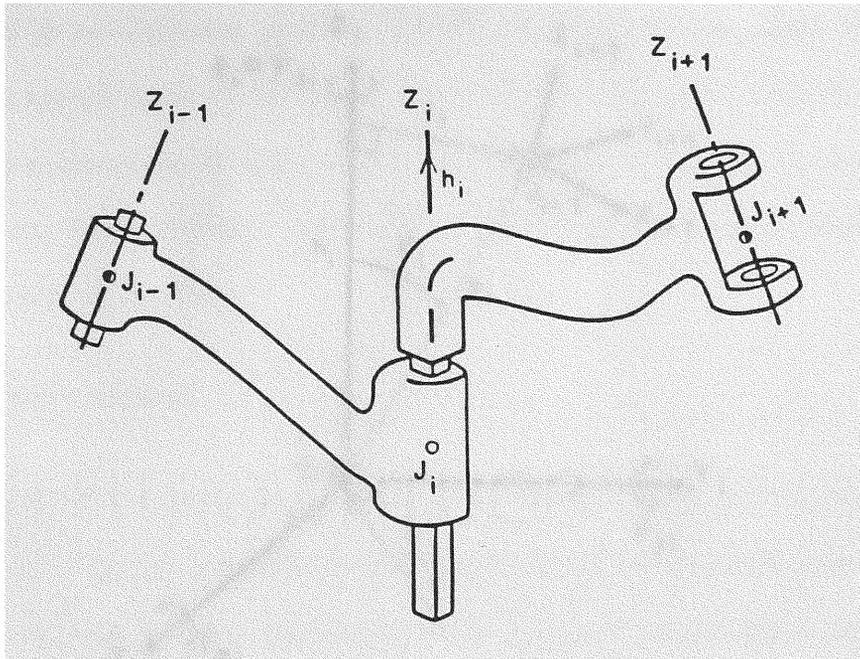


Figure 2.3 Prismatic Joint

to \vec{a}_i , as the link rotates. This yields,

$$h_i = a_i c \phi_i ; r_i = a_i s \phi_i \quad (2.7)$$

In order to define the angle ϕ_i , a spherical joint may have 3 degrees of freedom (ψ_x, ψ_y, ψ_z) which represent rotations about the cartesian coordinate axes. By definition, in C-B notation, ψ_z is equivalent to θ_i . Irrespective of the order of rotation about the X_i and Y_i axes,

$$T_{r_1}(X_P \psi_x) T_{r_2}(Y_P \psi_y) \text{ or } T_{r_1}(Y_P \psi_y) T_{r_2}(X_P \psi_x) = \begin{bmatrix} - & - & - & 0 \\ - & - & - & 0 \\ - & - & c\psi_x c\psi_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

where element $c\psi_x c\psi_y$ is the direction cosine between Z_i and a_i . Since the angle between Z_i and a_i was previously defined as ϕ_i , then

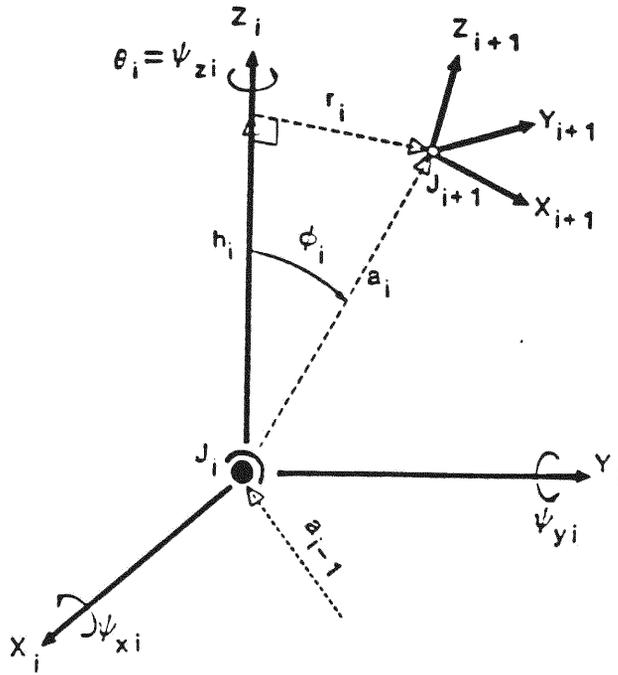


Figure 2.4 Spherical joint geometric parameters

$$\phi_i = c^{-1}(c\psi_x, c\psi_y) \quad (2.9)$$

The resulting characteristic matrix for the spherical joint can then be written as

$$T_i(\theta_i, h_i, r_i, \alpha_i, \beta_i) = T_i(\theta_i, a_i c \phi_i, a_i s \phi_i, \alpha_i, \beta_i) =$$

$$= \begin{bmatrix} D(\theta)_i & a_i s \phi_i c \theta_i \\ a_i s \phi_i s \theta_i & \\ a_i c \phi_i & \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} D(\theta)_i & P(\theta \cap \phi)_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

2.3 Kinematics

Kinematics is the science of motion which treats motion without regard to the forces which cause it. Kinematics is divided into direct and inverse kinematics. Direct kinematics studies the relative motions among the body segments (links) of the system. Basically, its

final goal is to determine position, velocity, and acceleration, based on prescribed geometric specifications of the system. Inverse kinematics, on the other hand, has as its objectives to determine the system geometric parameters with given function, trajectory, velocity, etc. Direct kinematic analysis is the topic of the following sections, beginning with the position equations.

2.3.1 Position Analysis

The general homogeneous characteristic matrix of a lower pair, T_i , was derived in equation 2.3. The relative orientation and position of the n-th reference frame (joint relative coordinate system), with respect to the universal coordinates, are obtained by a sequence of matrix multiplications. For an n joint open-chain mechanical system, its analytical position equation can be written as

$$H = \prod_{i=1}^n T_i = \begin{bmatrix} D & P \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.11)$$

where T_i is the characteristic matrix of the i-th joint. The resultant homogeneous matrix H contains the direction cosine matrix D and the position vector P to specify the orientation and position, respectively. These two components of H are the most essential pieces of information for the control of the manipulator. Also, H is the resultant system matrix for the position analysis of a specific reference point of the open-chain mechanical system. The reference point may indicate the end-effector, the gripper center, or any other point in the system.

2.3.2 Separation of Variables in Matrix

Prior to the velocity and acceleration analyses, a procedure named separation of

variables in matrix must be performed. Referring to equations 2.1-3, T_i is composed of elementary transformation matrices

$$\begin{aligned} T_i(\theta_p, h_p, r_p, \alpha_p, \beta_p) &= T_{c_i}(\theta_p, h_p, r_p) T_{b_i}(\alpha_p, \beta_p, 0) \\ &= T_r(Z, \theta_p) T_t(X, h_p) T_r(X, r_p) T_r(X, \alpha_p) T_r(Y, \beta_p) \end{aligned} \quad (2.12)$$

One may regroup equation 2.12 so that each design variable in T_i is isolated into a single-variable matrix V . For instance, there are two variables to be considered in the case of a spherical joint. Assigning the number of variables $q = 2$ and $\xi_1 = \theta_i$, $\xi_2 = \phi_i$, and rearranging T_i into variable matrices $V_1(\xi_1 = \theta_i)$ and $V_2(\xi_2 = \phi_i)$, known $h_i = a_i c \phi_i$ and $r_i = a_i s \phi_i$,

$$\begin{aligned} T_i(\theta_p, h_p, r_p, \alpha_p, \beta_p) &= [T_r(Z, \theta_i)] [T_t(Z, a_i c \phi_i) T_t(X, a_i s \phi_i) T_r(X, \alpha_i) T_r(Y, \beta_i)] \\ &= V_1(\theta_i) V_2(\phi_i) \end{aligned} \quad (2.13)$$

the two design variables are individually isolated into the single-variable matrices V_1 and V_2 .

Apparently, for a joint that has only one design variable, such as a revolute joint, $T_i = V_i(\theta_i)$, where $q = 1$ and $\xi_1 = \theta_i$.

Therefore, T_i can be rewritten in the form

$$T_i = \prod_{j=1}^{q_i} V_j(\xi_j) \quad (2.14)$$

where q_i is the total number of degrees of freedom of the i th joint, and V_j is the corresponding j -th variable matrix which contains the variable ξ_j of the i th joint.

Substituting equation 2.14 into equation 2.13 yields

$$H = \prod_{i=1}^n T_i = \prod_{i=1}^n \left[\prod_{j=1}^{q_i} V_j(\xi_j) \right] \quad (2.15)$$

Also, the total number of design variables in the system is the sum of variables in each joint; let

$$m = \sum_{i=1}^n q_i \quad (2.16)$$

then equation 2.15 becomes

$$H = \prod_{i=1}^m V_i(\xi_i) \quad (2.17)$$

where index i indicates the i th design variable of the system counted sequentially from the first variable ξ_1 in T_1 to the last in T_n .

2.3.3 Velocity Analysis

The rate of change of joint variable ξ_i is

$$\dot{\xi}_i = d\xi_i/dt \quad (2.18)$$

Differentiating equation 2.17

$$\frac{dH}{dt} = \frac{d}{dt} \left\{ \prod_{i=1}^m V_i(\xi_i) \right\} = \frac{d}{dt} \{V_1 V_2 V_3 \dots V_m\} \quad (2.19)$$

Now, applying the chain rule,

$$\dot{H} = \frac{d\{ \}}{d\xi_1} \frac{d\xi_1}{dt} + \frac{d\{ \}}{d\xi_2} \frac{d\xi_2}{dt} + \dots + \frac{d\{ \}}{d\xi_m} \frac{d\xi_m}{dt} = \sum_{i=1}^m \frac{d\{ \}}{d\xi_i} \dot{\xi}_i \quad (2.20)$$

which yields,

$$\dot{H} = \sum_{i=1}^m \left(\prod_{j=1}^m M_j \right) \dot{\xi}_i \quad (2.21)$$

where

$$M_j = \begin{cases} V_j(\xi_j) & \text{if } i \neq j \\ V_j(\xi_j) = dV_j/d\xi_j & \text{if } i = j \end{cases} \quad (2.22)$$

at $i = j$,

$$K_i = \prod_{j=1}^m M_j \quad (2.23)$$

equation 2.21 is then formulated as

$$\dot{H} = \sum_{i=1}^m K_i \dot{\xi}_i \quad (2.24)$$

2.3.4 Acceleration Analysis

The relative accelerations between joints with constant time differential dt are expressed by,

$$\ddot{H} = d\dot{H}/dt \quad (2.25)$$

Substituting equation 2.24 into equation 2.25,

$$\ddot{H} = \frac{d\left(\sum_{i=1}^m K_i \dot{\xi}_i\right)}{dt} = \frac{d\left(\sum_{j=1}^m K_j\right)}{dt} \dot{\xi}_j + \sum_{i=1}^m K_i \left(\frac{d\dot{\xi}_i}{dt}\right) \quad (2.26)$$

which results in,

$$\begin{aligned} \ddot{H} &= \sum_{i=1}^m \frac{d\left(\sum_{j=1}^m K_j\right)}{d\xi_i} \dot{\xi}_j \frac{d\xi_i}{dt} + \sum_{i=1}^m K_i \ddot{\xi}_i \\ &= \sum_{i=1}^m \frac{d\left(\sum_{j=1}^m K_j\right)}{d\xi_i} \dot{\xi}_j \dot{\xi}_i + \sum_{i=1}^m K_i \ddot{\xi}_i \end{aligned} \quad (2.27)$$

Defining,

$$\sum_{i=1}^m \frac{d\left(\sum_{j=1}^m K_j\right)}{d\xi_i} \dot{\xi}_j = \sum_{i=1}^m \left(\sum_{j=1}^m \prod_{k=1}^m N_k \right) = \sum_{i=1}^m k_i \quad (2.28)$$

where

$$N_k = \begin{cases} V_k(\xi_k) & \text{if } k = j = i \\ V_k(\xi_k) & \text{if } k = i \text{ (} k \neq j \text{) or } k = j \text{ (} k \neq i \text{)} \\ V_k(\xi_k) & \text{if } k \neq i \text{ and } k \neq j \end{cases} \quad (2.29)$$

Finally, equation 2.27 becomes,

$$\ddot{H} = \sum_{i=1}^m (k_i \dot{\xi}_i + K_i \xi_i) \quad (2.30)$$

where the velocity $\dot{\xi}_i$ was previously obtained from the velocity analysis.

2.4 Kinematic Spaces

As mentioned in previous sections, kinematic analysis mainly consists of position (displacement), velocity, and acceleration analyses. The most important of the three being the position analysis. Roth and Shimano (1977) were first to describe the reachable space or volume of a robot manipulator. This was then standardized as the "workspace" of a manipulator.

The workspace of a robotic manipulator is defined as the set of all three-dimensional points that can be reached by a reference point located on the robotic hand. A restricted version of a robotic workspace is also defined as the set of all points that can be reached by a fixed orientation of the robotic hand. The study of robotic workspaces is important in arranging the associated flexible manufacturing cell of a robot, and assessing its efficiency in a manufacturing line. In general, the boundary of a robotic workspace is a complex surface composed of many surfaces, difficult to represent explicitly by geometrical equations.

Velocity and acceleration spaces are accomplished by determining velocities and accelerations over the surfaces which compose the workspace. Their applications are essential for the velocity/acceleration control of a robot following a specific contour.

2.4.1 Graphic Representation

The 3-dimensional representations of the kinematic spaces are generated by rotating

a planar projections of the corresponding kinematic space. The planar projections of the workspace are obtained through an interactive process, by fixing and moving specific joints. This procedure generates a collection of curves that describe the maximum and minimum reachable positions, along with other curves to show the outmost boundary of the workspace. All curves that determine the extreme positions are selected and all others discarded. To illustrate the 3-dimensional workspace (volume) this set of planar curves is then rotated about the joint axis of the base of the manipulator.

Once the planar workspace has been obtained, the velocity and acceleration spaces are obtained by determining velocities and accelerations along the curves which envelop the workspace. This procedure generates a set of 3-dimensional (non-planar) curves which is rotated about the joint axis of the base of the manipulator, generating the 3-dimensional velocity and acceleration spaces.

2.4.2 Numerical Examples

The RaMIP algorithm is utilized to analyze several robots. The following examples are selected for industrial robots with revolute, prismatic, and spherical joints. As mentioned earlier, each kinematic space is composed of a group of curves which represent the trajectory of the end-effector at its outmost reachable positions. Each curve is numbered so that they can be identified in each plot. The joint motion parameters used to describe each curve are listed in the tables presented with each example. The 3-dimensional kinematic spaces obtained are viewed from several different viewpoints for better clarity. In some cases, velocity and acceleration data are used to shade workspace sections for better clarity and identification, and a scale is shown next to the plot for reference. It is noted that the quality and clarity of the plots presented here are inferior in comparison with the original color plots shown on the computer screen.

Example 2.1 - Cincinnati Milacron T3 robot (RRR/RRR)

This robot is composed of six electric revolute joints, three for the base, shoulder, and elbow, and three for the wrist and hand. This robot was designed to bring the productivity of off-the-shelf robotic automation to simple tasks, such as machine tending and medium-duty materials handling. Its kinematic specifications modeled in C-B notation are listed Table A.1a in Appendix A. Figures 2.5-2.7 show the 2-dimensional workspace, velocity space, and acceleration space projected on the XZ and YZ planes, respectively. The kinematic curve parameters are given in Table 2.2. Figs. 2.8 and 2.9 show a sectional view, from 30 to 120 deg base rotation, of the 3-dimensional workspace from viewpoints located at (3,0.5,0.0), and at (2,-0.5,0.3), respectively. Fig. 2.10 shows the workspace shaded using resultant velocity values, by associating a color scale (or gray scale) to the resultant velocity data obtained. The advantage of this data displaying technique is based on the fact that a fourth quantity may be displayed in a 3-dimensional plot. This technique is often utilized (for instance in finite element software) to display temperature, pressure, stress distributions, and others over a certain computer generated model, or object. In the particular case of Fig. 2.10, it is possible to display the resultant velocity at the location in 3-d space to which it corresponds. Fig 2.11 on the other hand, shows the 3-dimensional acceleration space using acceleration coordinates. In Figure 2.11, no information about the location associated to each acceleration value is available or may be deduced by any means. This technique is also utilized in the examples that follow.

Table 2.2. Cincinnati Milacron T3 Curve Parameters

Base rotation $\theta_1 = 30^\circ$; $\theta_5 = 0^\circ$; $\theta_6 = \text{any angle}$

curve	1	2	3	4	5	6	7	8
θ_2°	0/90	0	0	0/90	90	90	90	90
θ_3°	0	-150/0	-150	-150	-150/0	0	-150/0	-150
θ_4°	0	0	-90/0	-90	90	0/90	90	-90/90

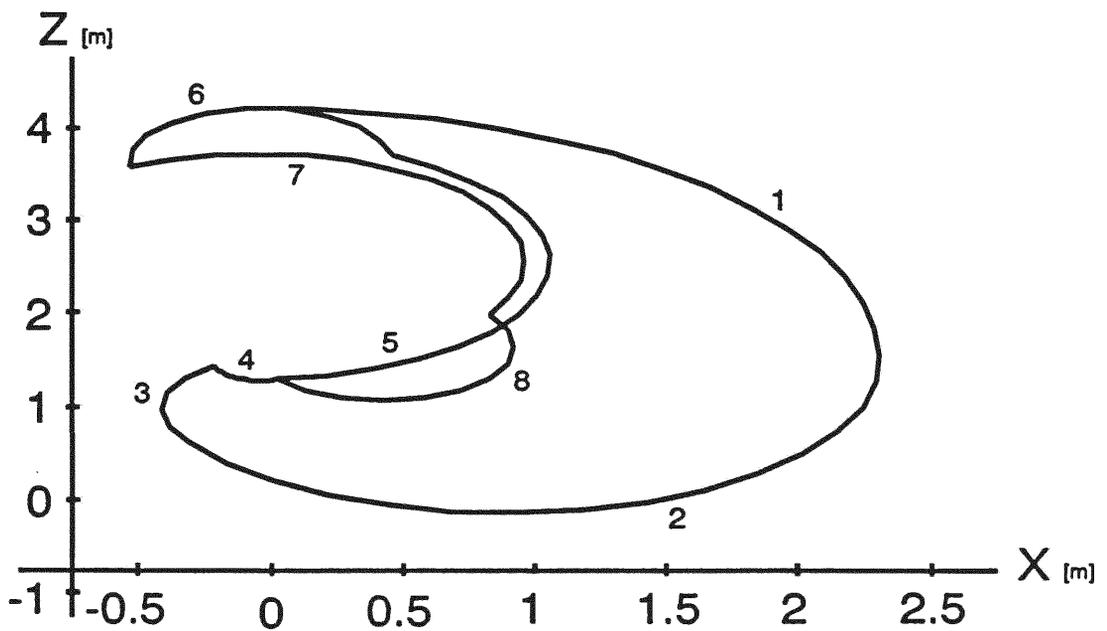


Figure 2.5a Cincinnati Milacron T3, XZ Workspace Projection.

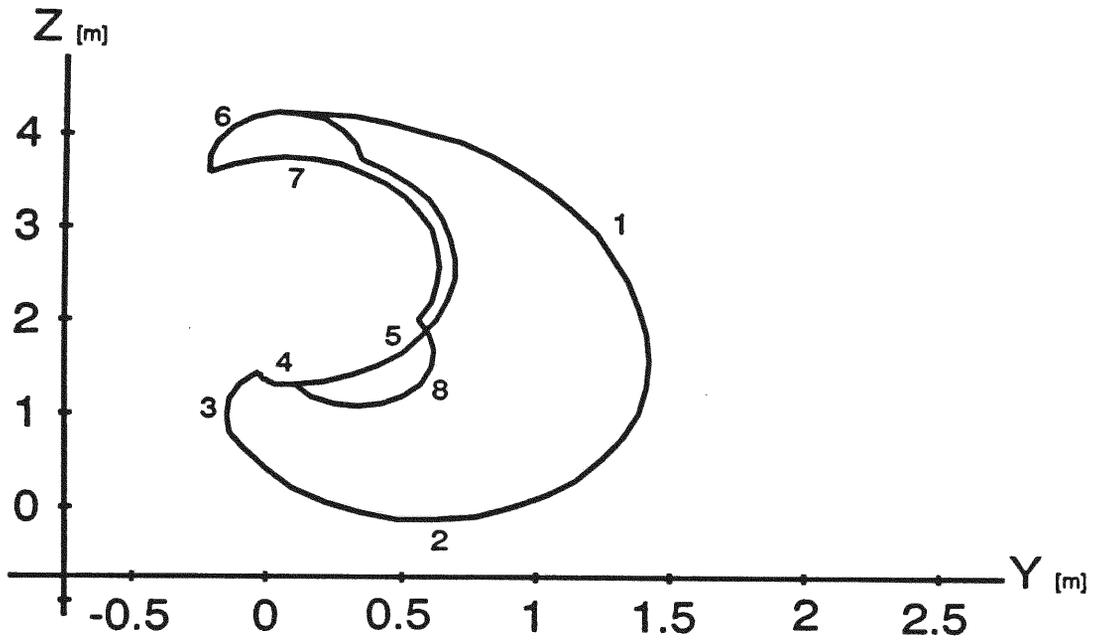


Figure 2.5b Cincinnati Milacron T3, YZ Workspace Projection.

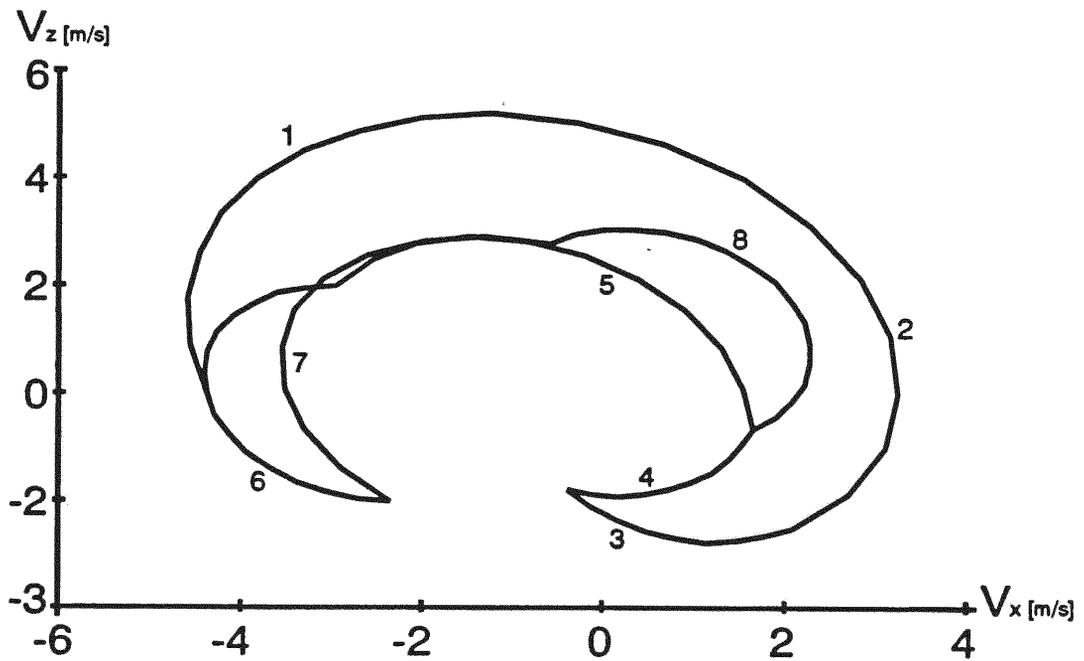


Figure 2.6a Cincinnati Milacron T3, XZ Velocity Space Projection.

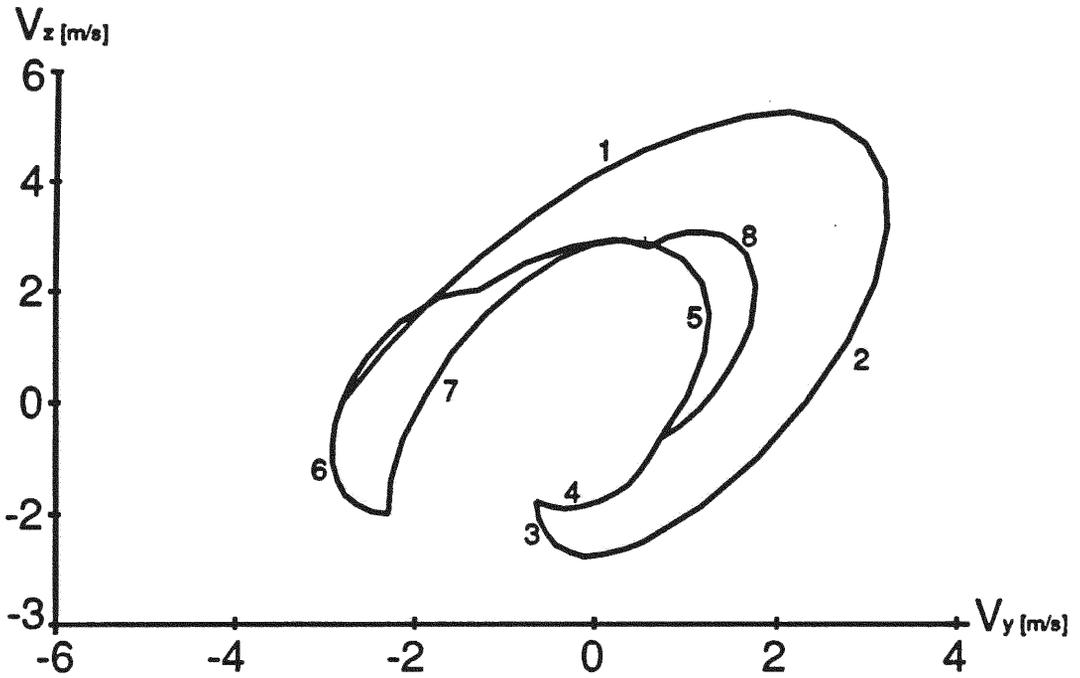


Figure 2.6b Cincinnati Milacron T3, YZ Velocity Space Projection.

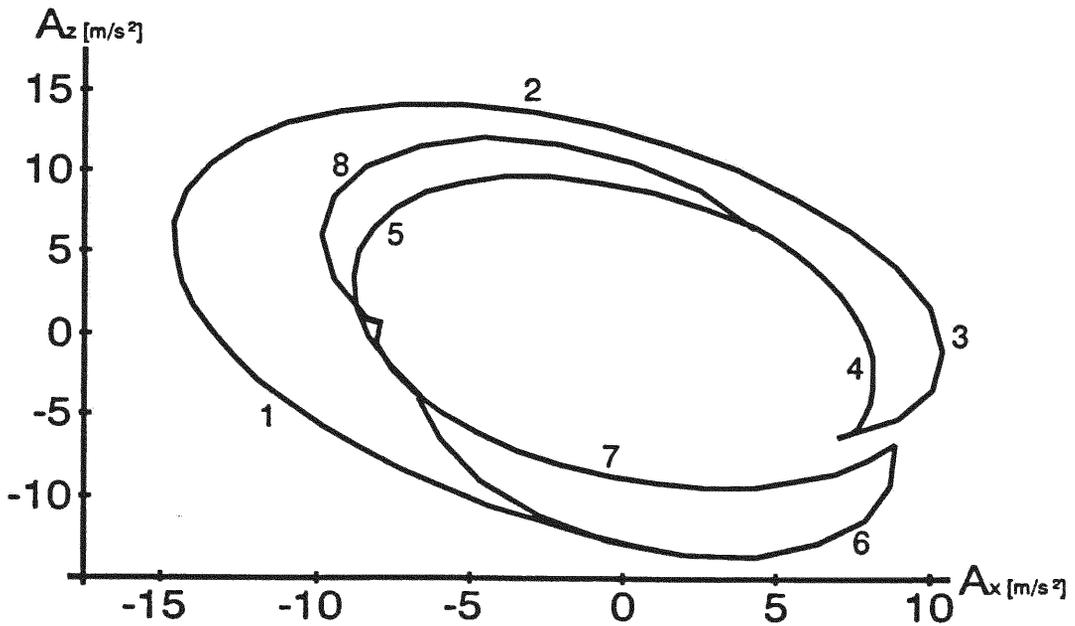


Figure 2.7a Cincinnati Milacron T3, XZ Acceleration Space Projection

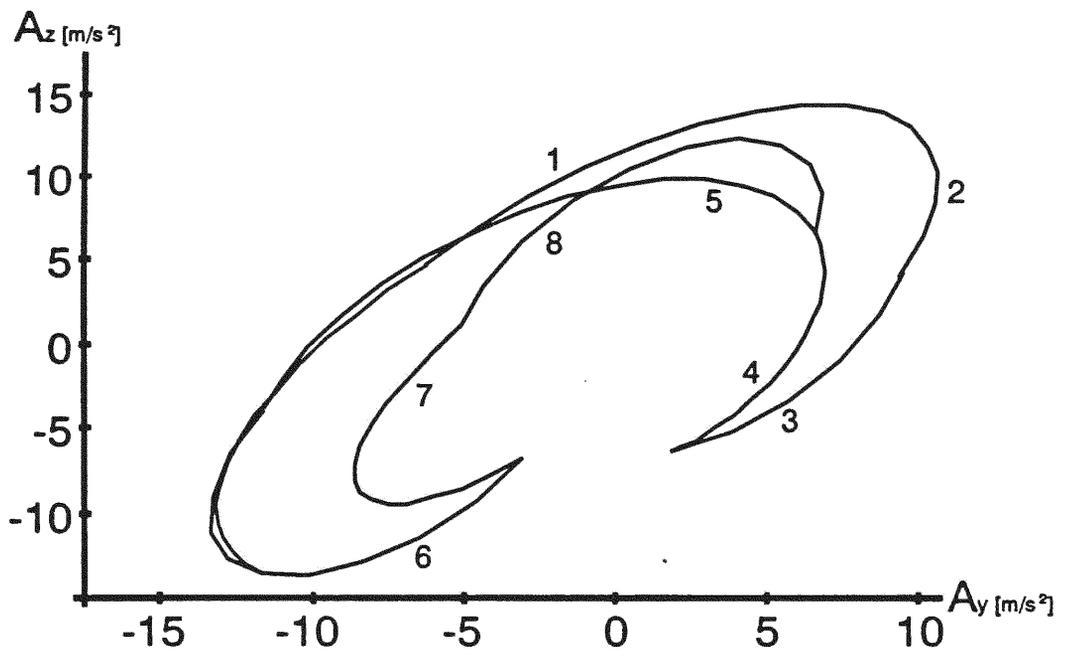


Figure 2.7b Cincinnati Milacron T3, YZ Acceleration Space Projection.

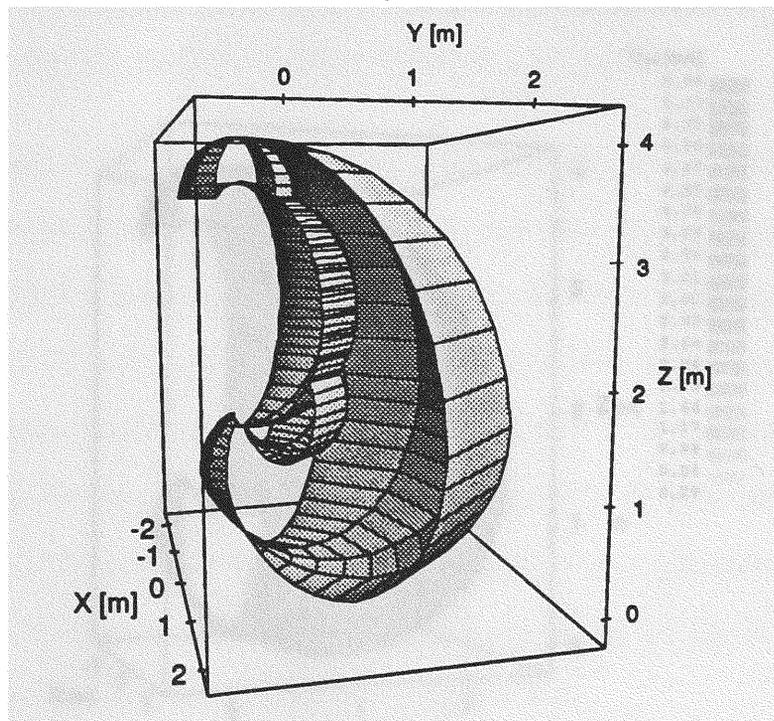


Figure 2.8 Cincinnati Milacron T3 Workspace, (viewp. (2,-0.5,0.3), sect. 30°-120°)

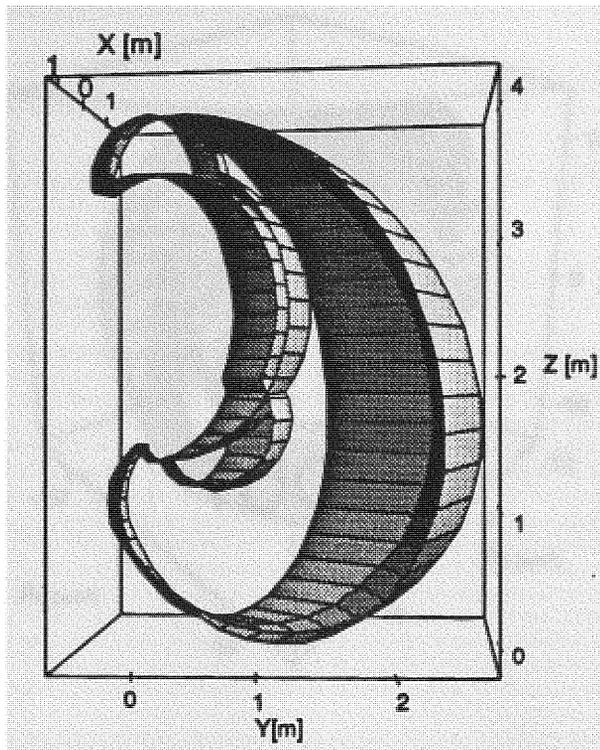


Figure 2.9 Cincinnati Milacron T3 Workspace, (viewp. (3,0.5,0) sect. 30°-120°)

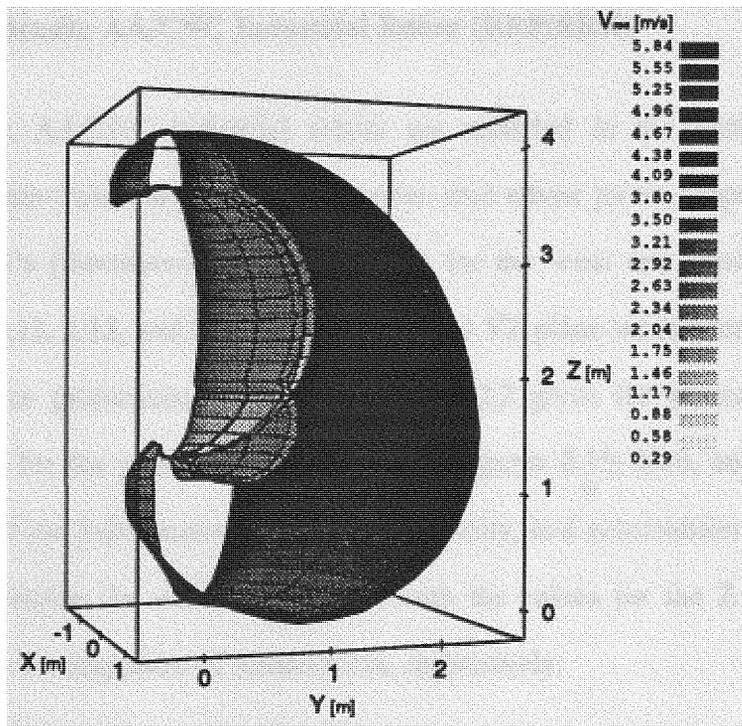


Figure 2.10 Cincinnati Milacron T3 Average Vel. Shaded Workspace, (viewp. (2,-0.75,0))

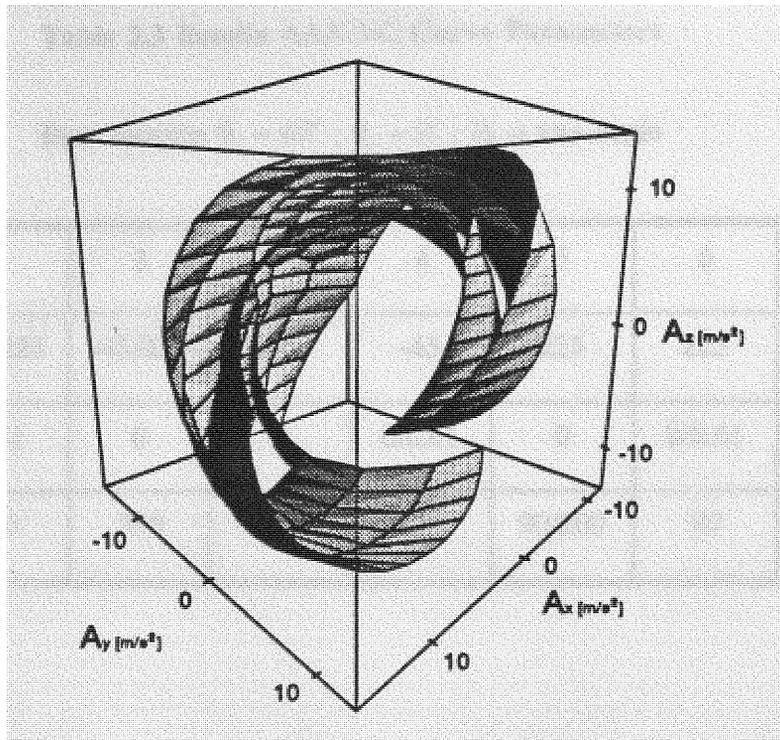


Figure 2.11 Cincinnati Milacron T3 Acceleration Space, (view. (1,1,0.3) section 30 to 120 deg.)

Example 2.2 - Bendix AA/CNC Industrial Robot (RRP/RRR)

The Bendix AA/CNC industrial robots are modeled in the pattern of R-R-P (Revolute-Revolute-Prismatic) for base, shoulder, and elbow joints, respectively; along with another 3R's (Revolute-Revolute-Revolute) for the wrist and hand joints (Table A.2a). Figures 2.12, 2.13, and 2.14 show the XZ and YZ plane workspace, velocity, and acceleration space projections, respectively. Table 2.2 gives the values for the joint parameters used for the generation of the curves. Figures 2.15, 2.16, and 2.18 show a sectional view of the 3-dimensional workspace, velocity, and acceleration spaces. Figure 2.17 and 2.19 display the workspace shaded with the values for the Z component of velocity and the Y component of acceleration, respectively.

Table 2.3 Bendix AA/CNC Curve Parameters

Base rotation $\theta_1 = 60^\circ$; $\theta_4 = 0^\circ$; $\theta_6 = \text{any angle}$

curve	1	2	3	4	5	6	7
θ_2°	-45/225	-45/225	-45	-45	225	225	225
h_3 m	0.61	0	0	0/0.61	0	0/0.61	0.61
θ_5°	-20	200	-20/200	-20	90/200	90	-20/90

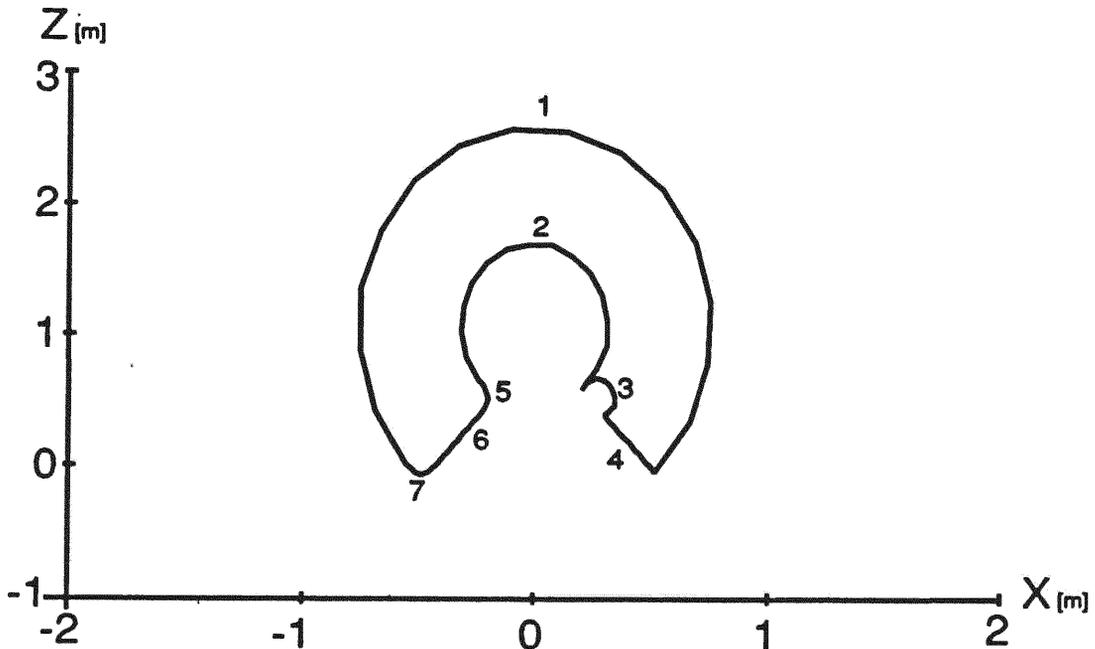


Figure 2.12a Bendix AA/CNC, XZ Workspace Projection

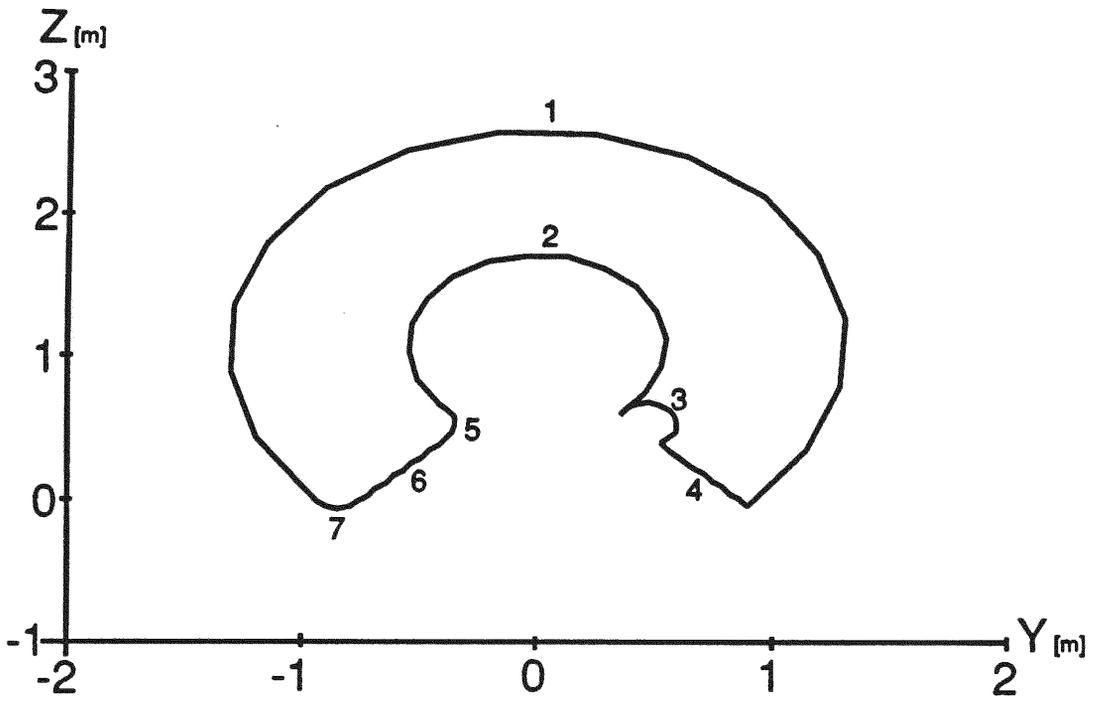


Figure 2.12b Bendix AA/CNC, YZ Workspace Projection

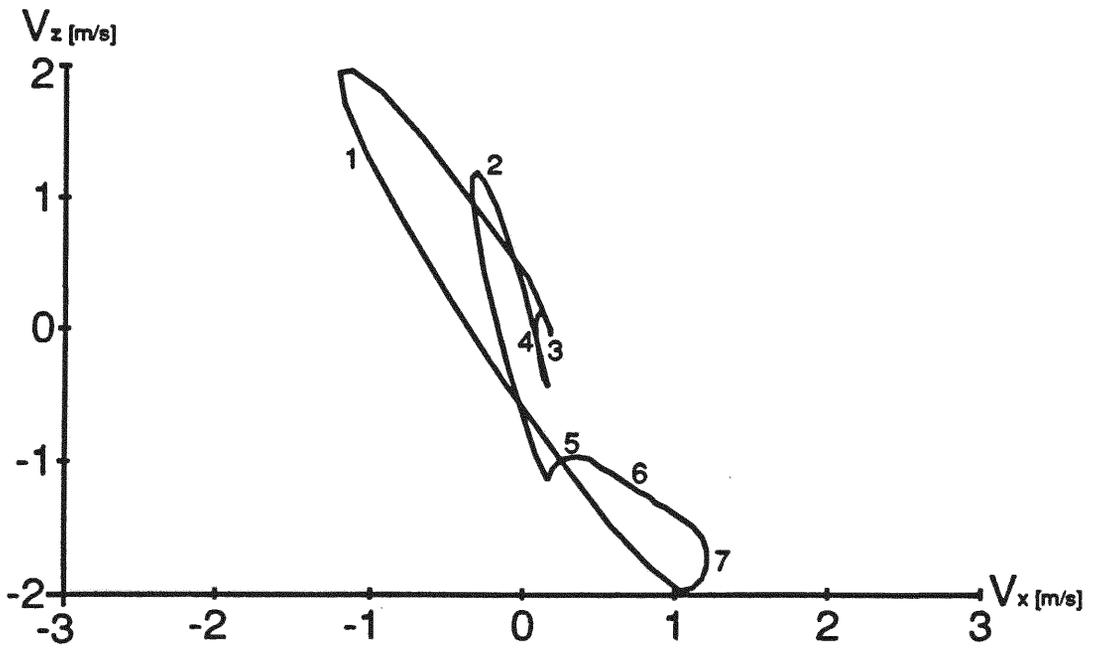


Figure 2.13a Bendix AA/CNC, XZ Velocity Space Projection

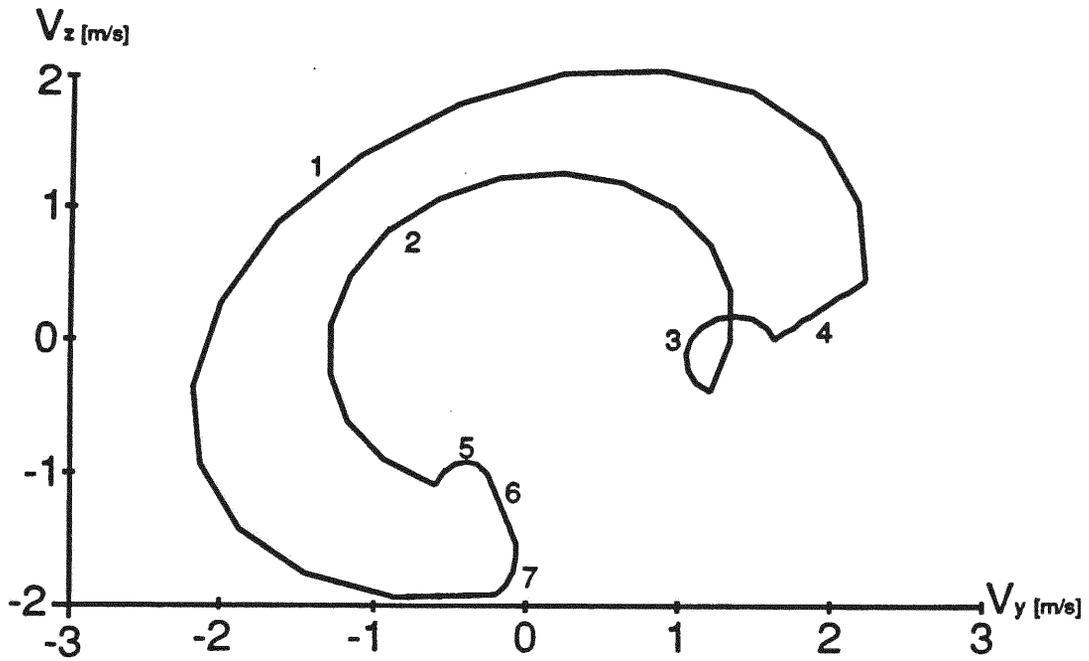


Figure 2.13b Bendix AA/CNC, YZ Velocity Space Projection

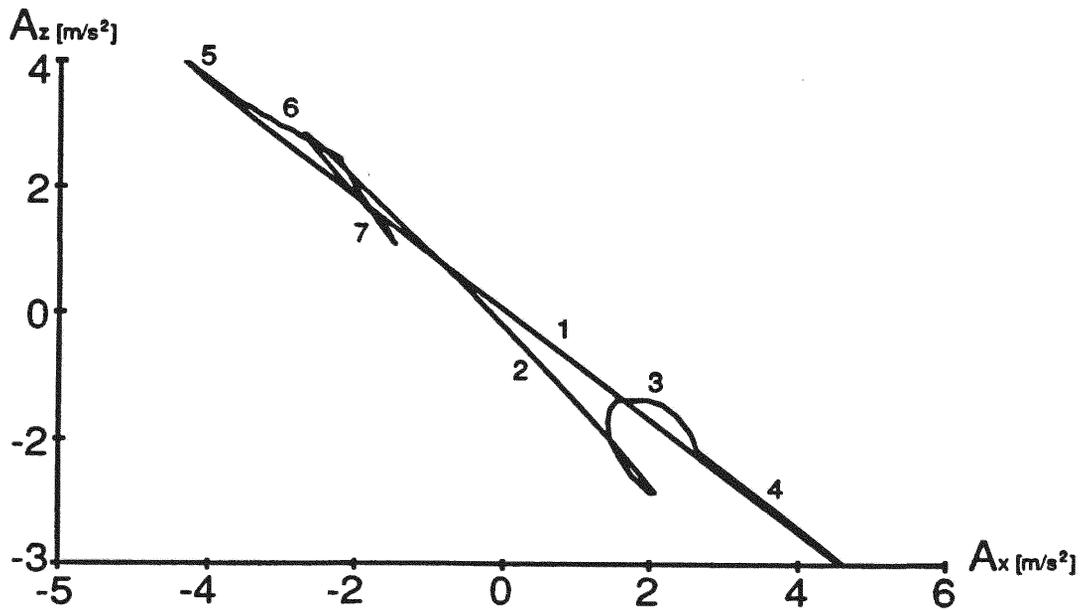


Figure 2.14a Bendix AA/CNC, XZ Acceleration Space Projection

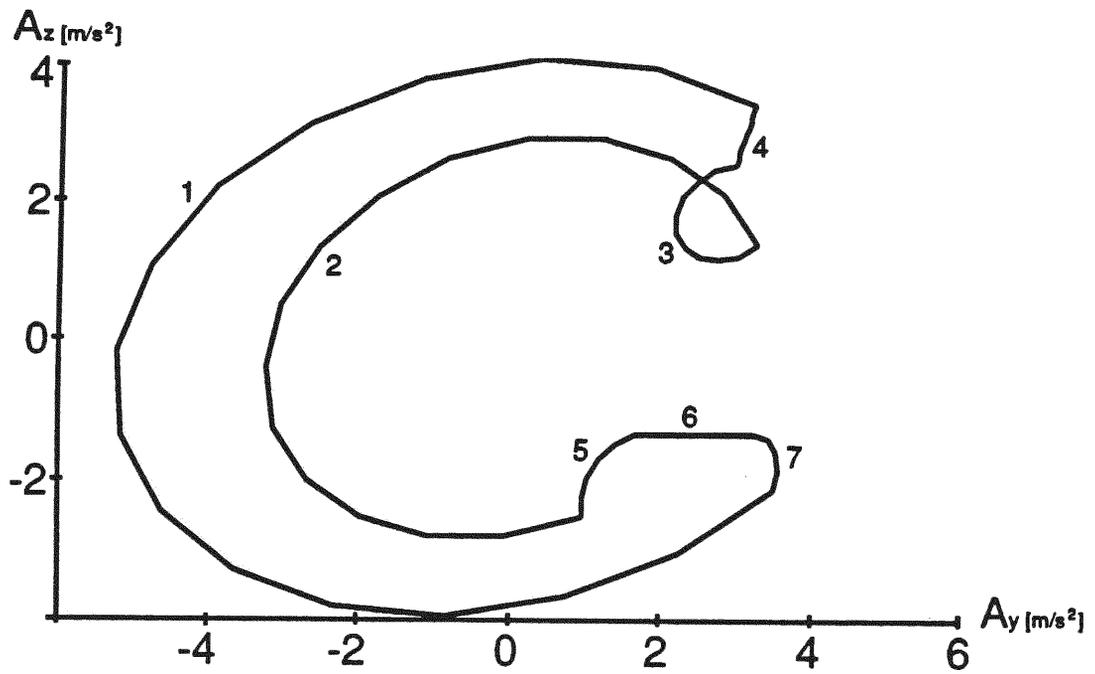


Figure 2.14b Bendix AA/CNC, YZ Acceleration Space Projection

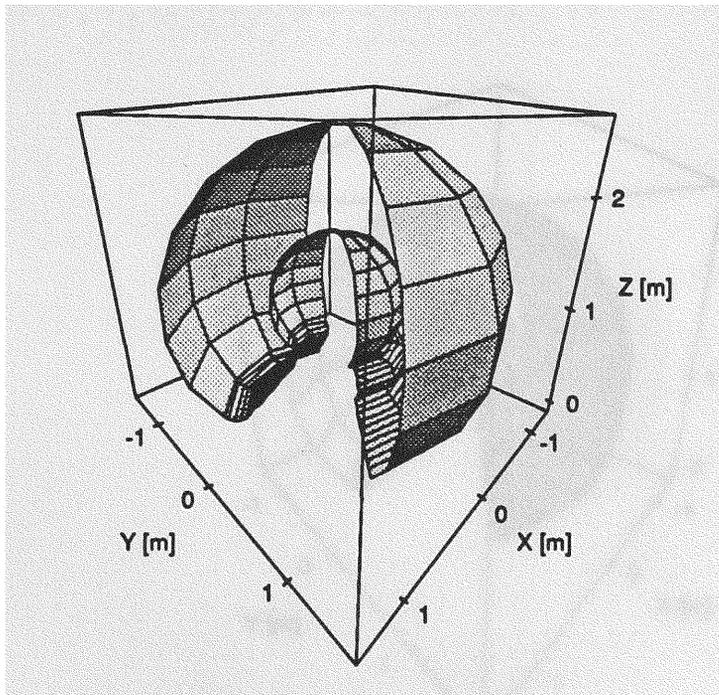


Figure 2.15 Bendix AA/CNC Workspace (viewpt. (.75,.75,.3) sect. 0°-95°)

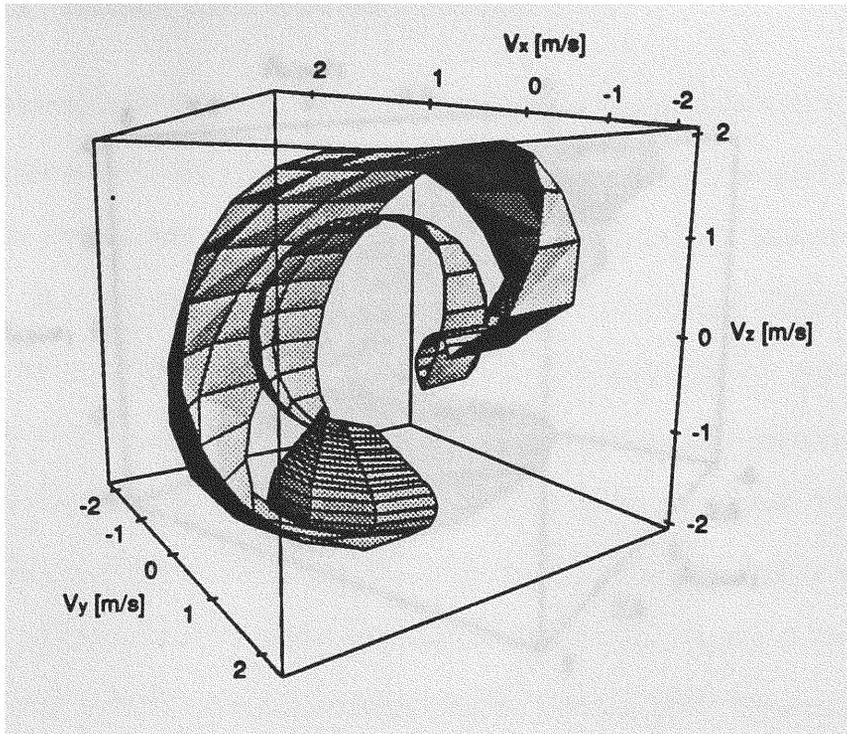


Figure 2.16 Bendix AA/CNC Velocity Space (viewpt. (.75,.75,.3) sect. 0-95 deg.

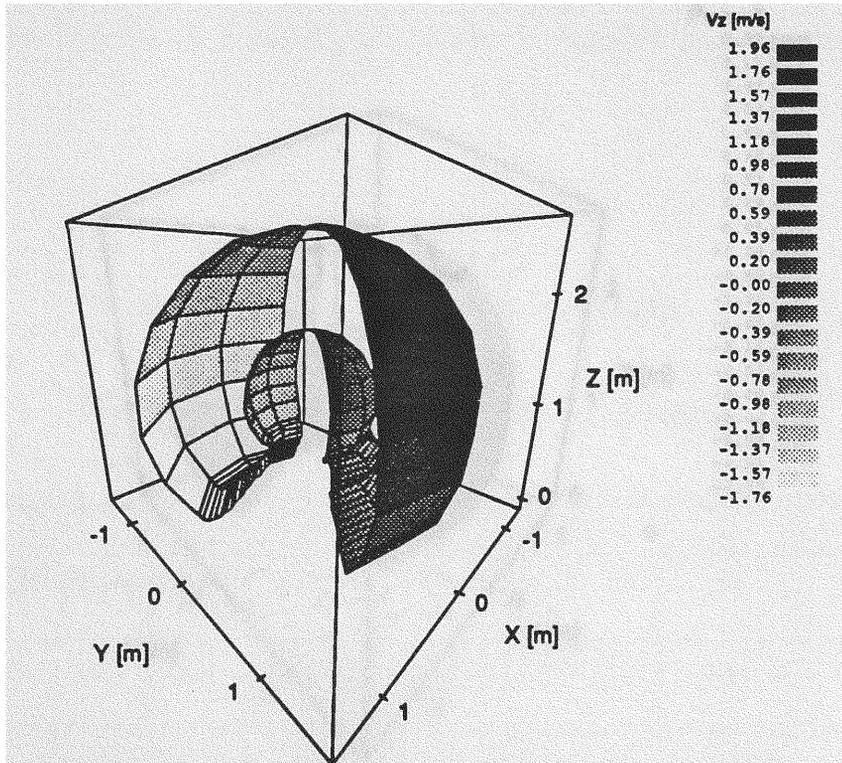


Figure 2.17 Bendix AA/CNC Vz Shaded Workspace (viewpt. (.75,.75,.3) sect 0°-95°)

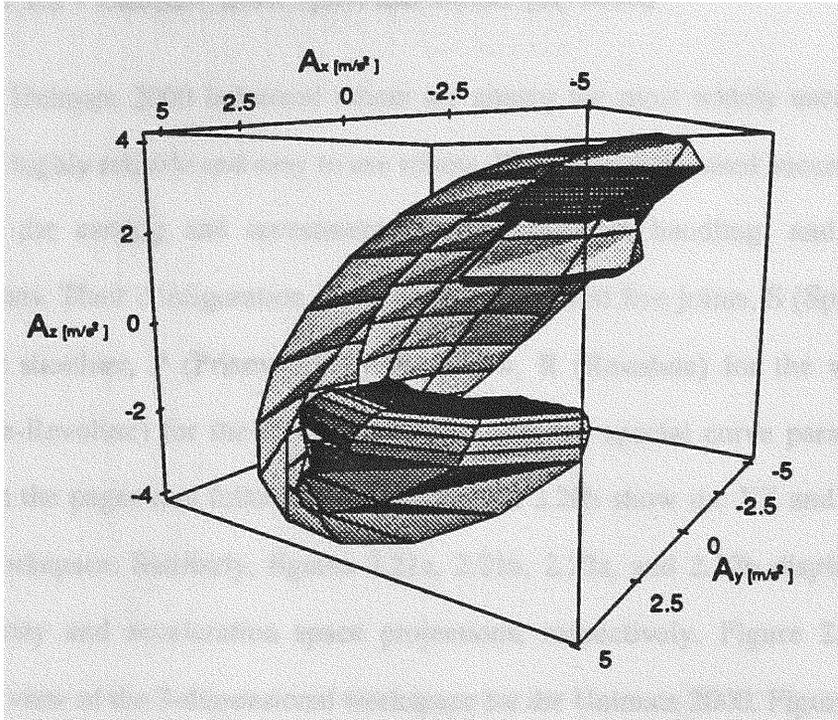


Figure 2.18 Bendix AA/CNC Acceleration Space (viewpt. (.75,.75,3) sect. 0-95 deg.

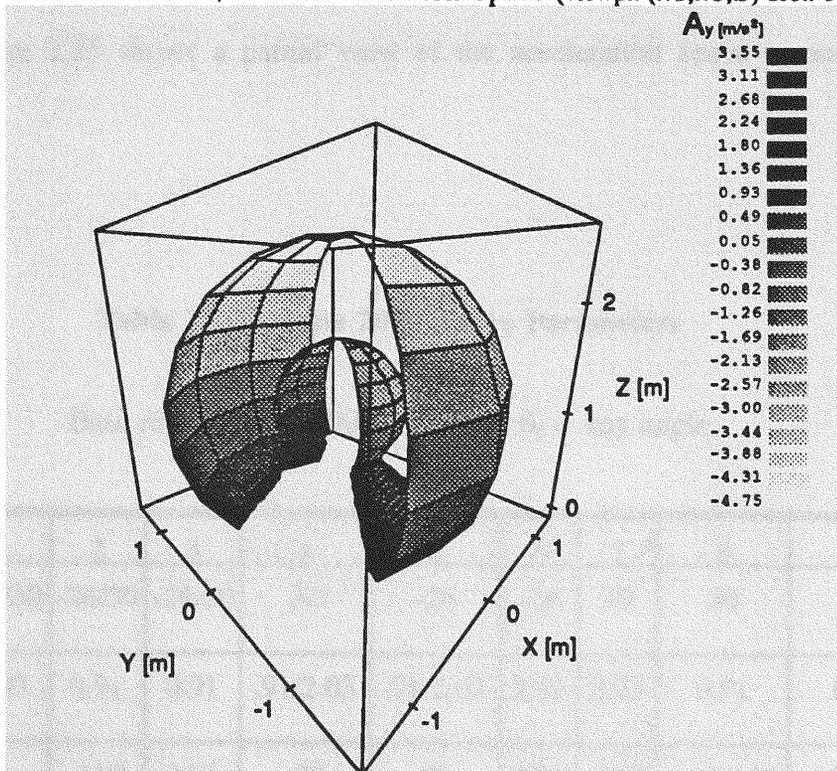


Figure 2.19 Bendix AA/CNC A_y Shaded Workspace (viewpt. (.75,.75,3) sect. 0°-95°)

Example 2.3 - Unimate 2000 Spherical Robot (SP/RRR)

The Unimate 2000 industrial robots are among the most widely used in the world. They are highly reliable and easy to use robots. They have been used successfully for spot welding, die casting and investment casting, materials handling, and several other applications. Their configuration, Table A.3a, consists of five joints, S (Spherical) for the base and shoulder, P (Prismatic) for the elbow, R (Revolute) for the wrist, and R-R (Revolute-Revolute) for the hand. Table 2.4 gives the spacial curve parameters for the figures in the pages that follow. Figure 2.20a and 2.20b show the XZ and YZ projection of the workspace. Similarly, figures 2.21a, 2.21b, 2.22a, and 2.22b display the XZ and YZ velocity and acceleration space projections, respectively. Figure 2.23 presents a sectional view of the 3-dimensional workspace for the Unimate 2000. Figure 2.24 displays a sectional view of the workspace shaded using the values of the Y component of velocity. Figure 2.25 shows a partial view of the acceleration space in acceleration coordinates.

Table 2.4 Unimate 2000 Curve Parameters

Base rotation $\theta_1 = 100^\circ$; $\theta_4 = 0^\circ$; $\theta_5 = \text{any angle}$

curve	1	2	3	4	5	6	7	8	9
ϕ_1°	-26/30	-26/30	-26/30	30	-26	-26	30	30	-26
h_2 m	2.03	0.91	0.91	.91/2.03	.91/2.03	2.03	2.03	0.91	0.91
θ_3°	0	-110	110	-88	88	0/88	-88/0	-110/110	-110/110

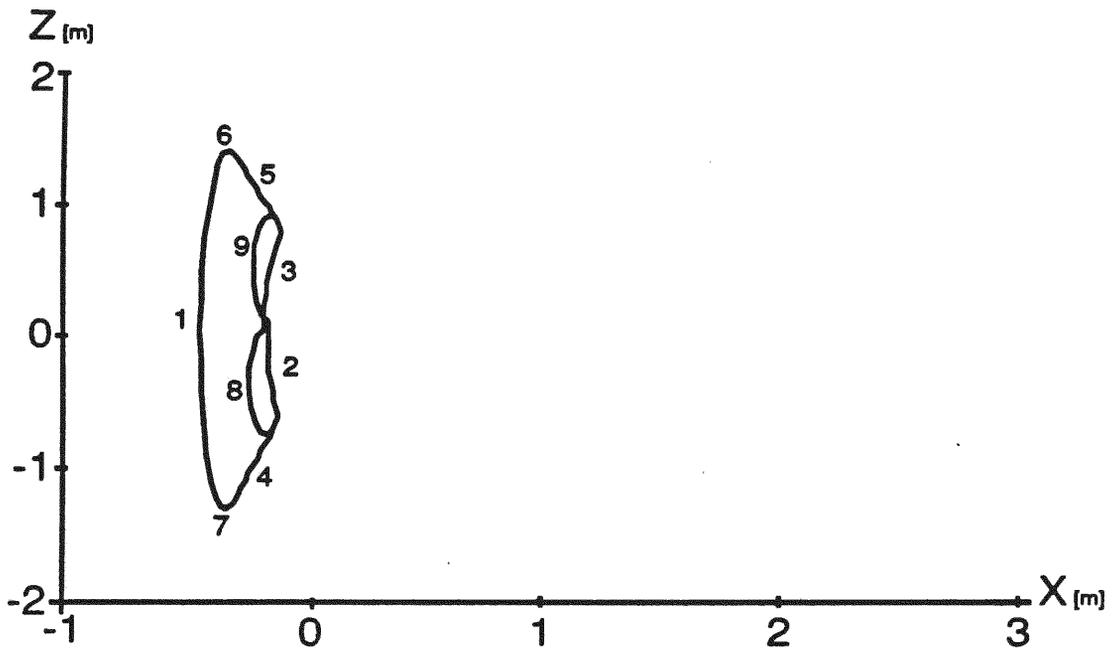


Figure 2.20a Unimate 2000, XZ Workspace Projection

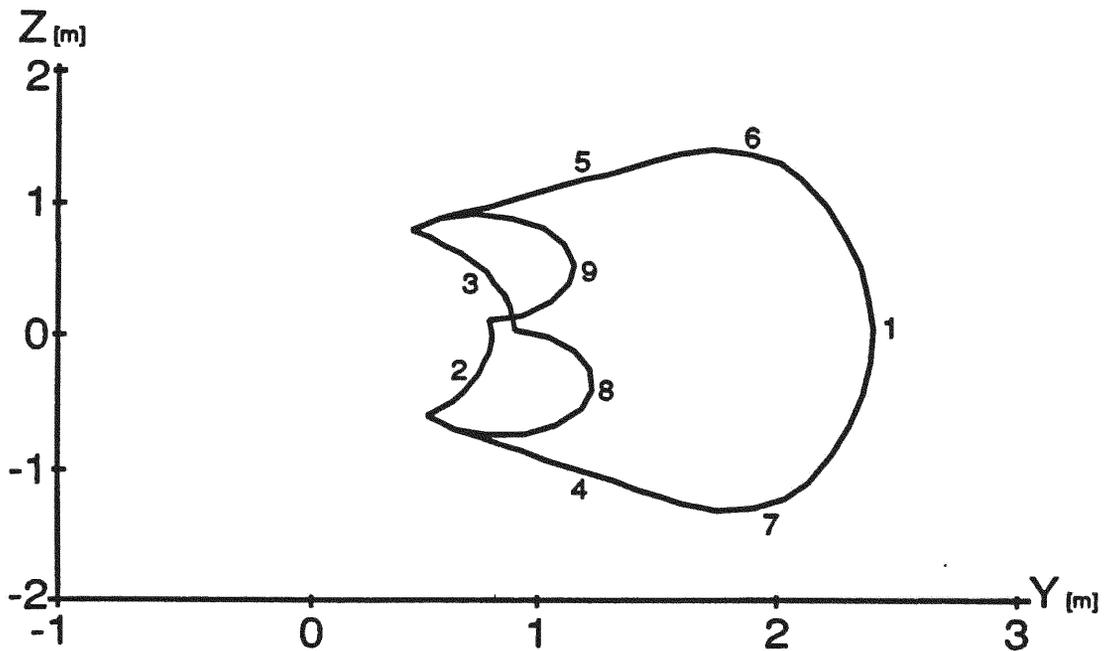


Figure 2.20b Unimate 2000, YZ Workspace Projection

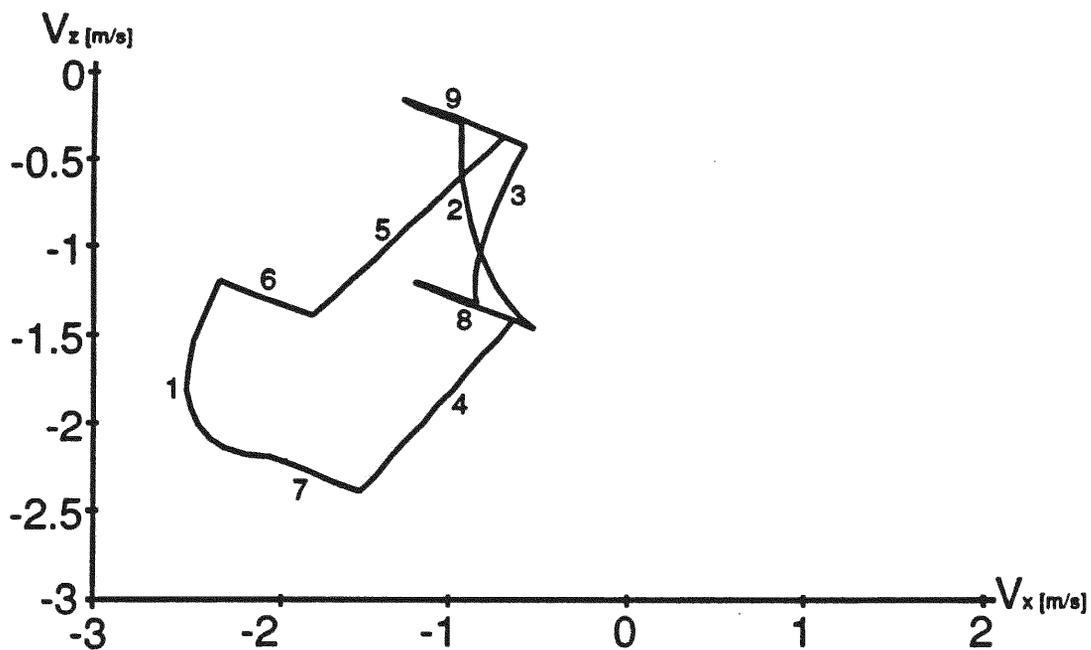


Figure 2.21a Unimate 2000, XZ Velocity Space Projection

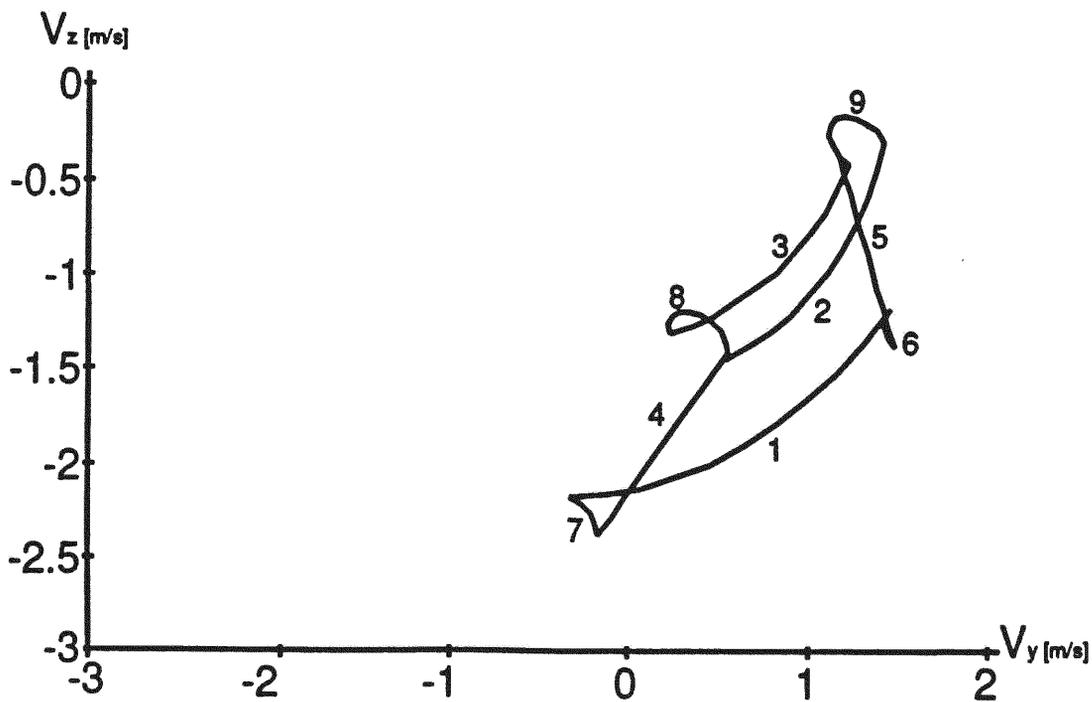


Figure 2.21b Unimate 2000, YZ Velocity Space Projection

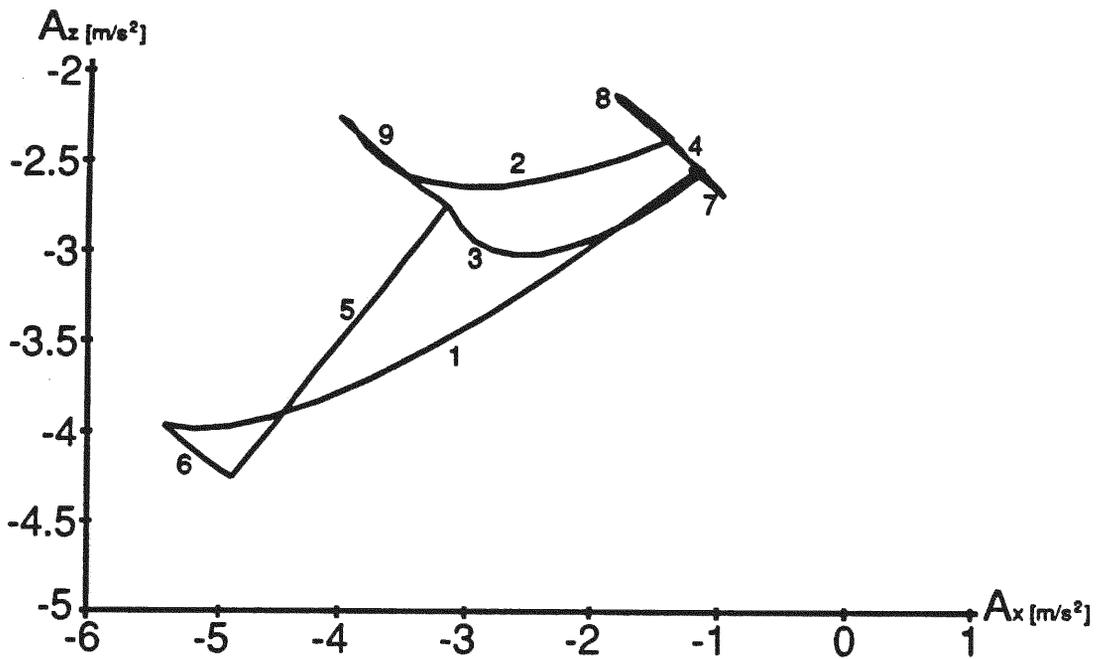


Figure 2.22a Unimate 2000, XZ Acceleration Space Projection

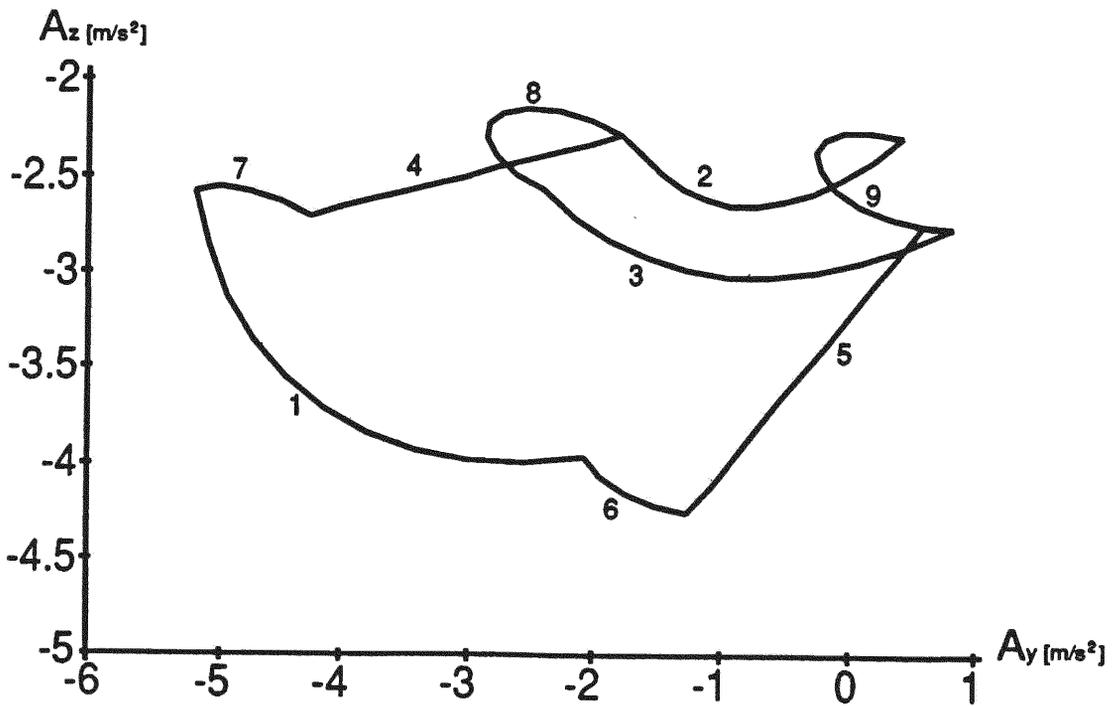


Figure 2.22b Unimate 2000, YZ Acceleration Space Projection

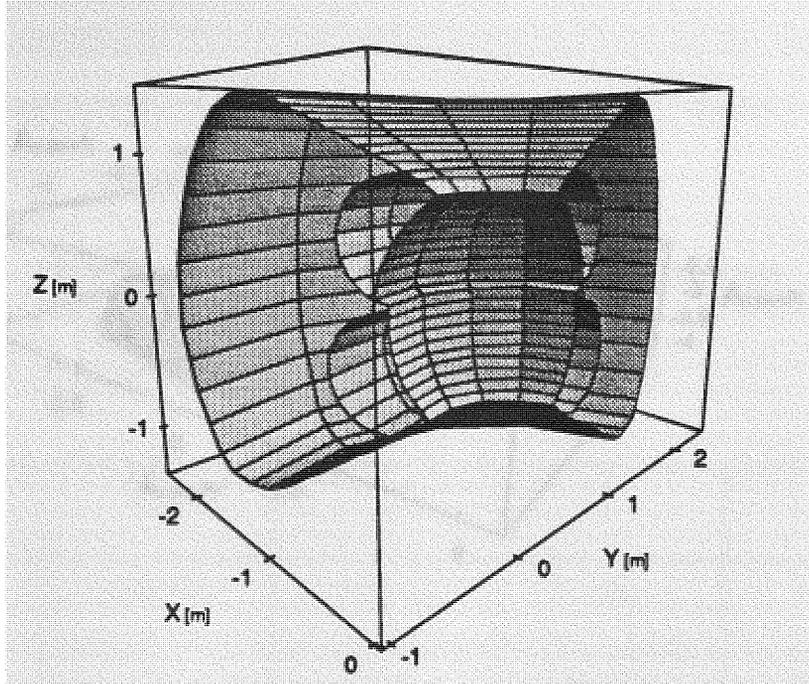


Figure 2.23 Unimate 2000, Workspace (viewp. (1,-1,0.3) sect. 0°-104°)

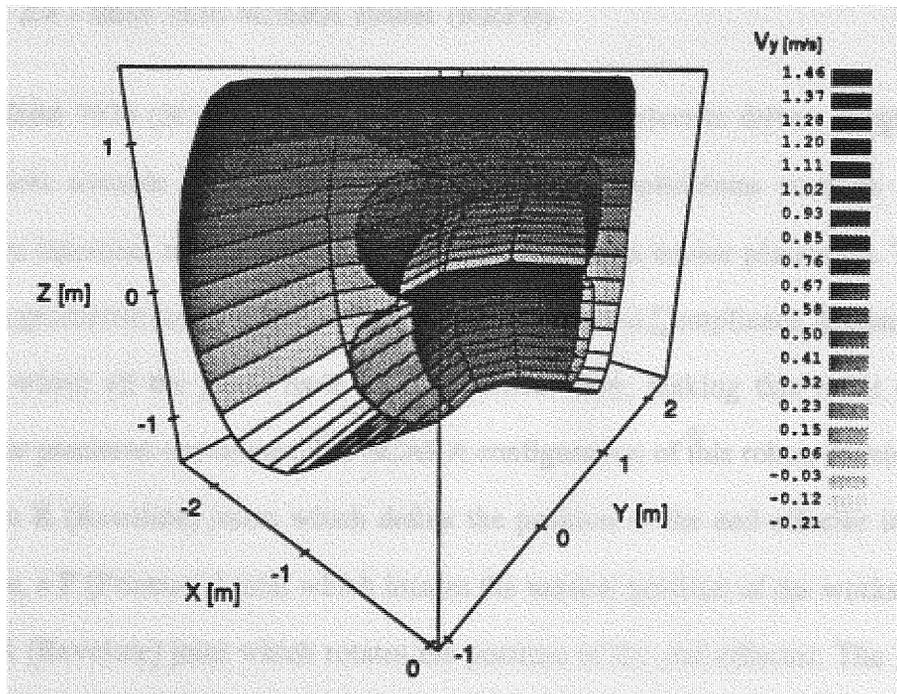


Figure 2.24 Unimate 2000, V_y Shaded Workspace (viewp. (0.75,-1,0.4) sect. 0°-104°)

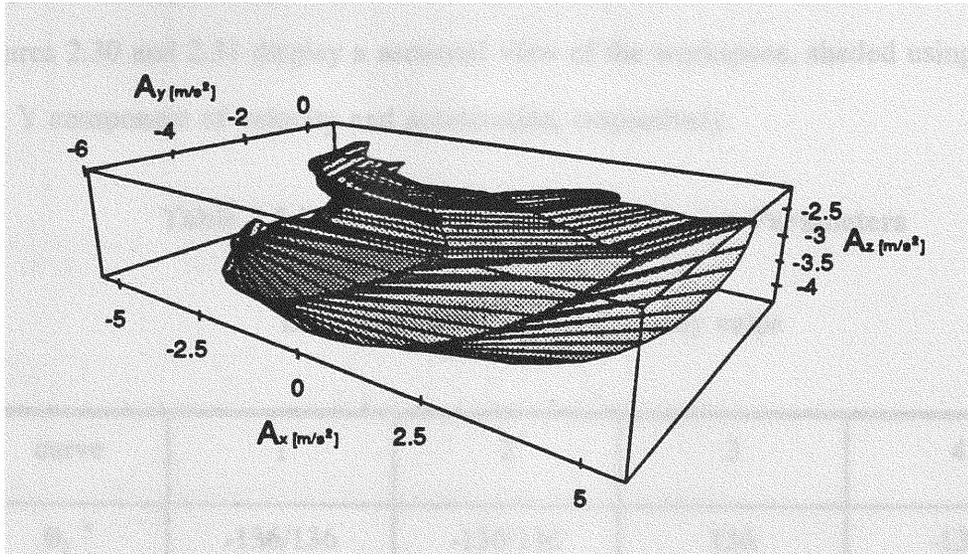


Figure 2.25 Unimate 2000, Acceleration Space (viewp. (1,-2,0.5) sect 0°-104°)

Example 2.4 - IBM 7576 SCARA Robot (RRPR)

The IBM 7576 (SCARA) manufacturing systems are electric driven, programmable manipulators suitable for a wide range of industrial applications such as electronic component insertion, testing, packaging, and surface-mount device placement. This robot presents a geometric configuration called SCARA (Selective Compliance Assembly Robot Arm), in which all the joints have parallel vertical axes, making this robot especially suitable for precision operations. The SCARA configuration of this robot, consists of four joints, two R (Revolute) joints which define the position of the end-effector in a planar workspace, a P (Prismatic) joint which locates the vertical position of the workspace, and a fourth R (Revolute) joint which rotates the position of the end-effector. The kinematic parameters for this robot are listed in Table A.4a in Appendix A. Table 2.4 gives the kinematic data utilized to generate all kinematic space curves. Figures 2.26a, 2.26b, and

2.26c show the XZ, YZ, and XY workspace projections for the IBM 7576 robot. Figures 2.27a, 2.27b, 2.28a, and 2.28b show the XZ and YZ velocity and acceleration space projections, respectively. Figure 2.29 shows a sectional view of the workspace, and figures 2.30 and 2.31 display a sectional view of the workspace, shaded using values of the Y component of velocity and acceleration, respectively.

Table 2.5 IBM 7576 SCARA Robot Curve Parameters

Base rotation $\theta_1 = 30^\circ$; $\theta_4 = \text{any value}$

curve	1	2	3	4
θ_2°	-136/136	-136/136	136	-136
h_3 m	0	0.25	0/0.25	0/0.25

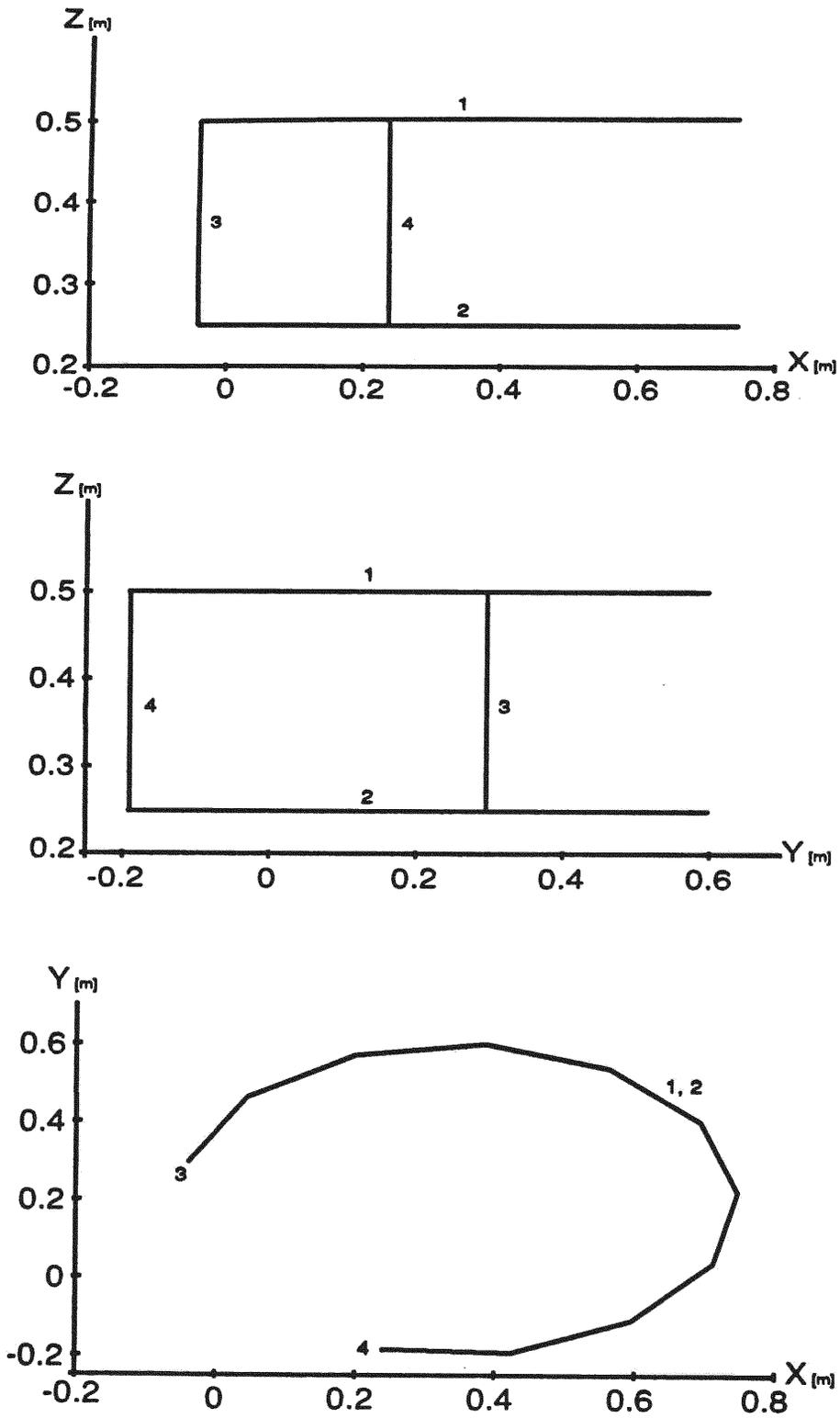


Figure 2.26 IBM 7576 Robot Workspace Plane Projections. a) XZ, b) YZ, c) XY

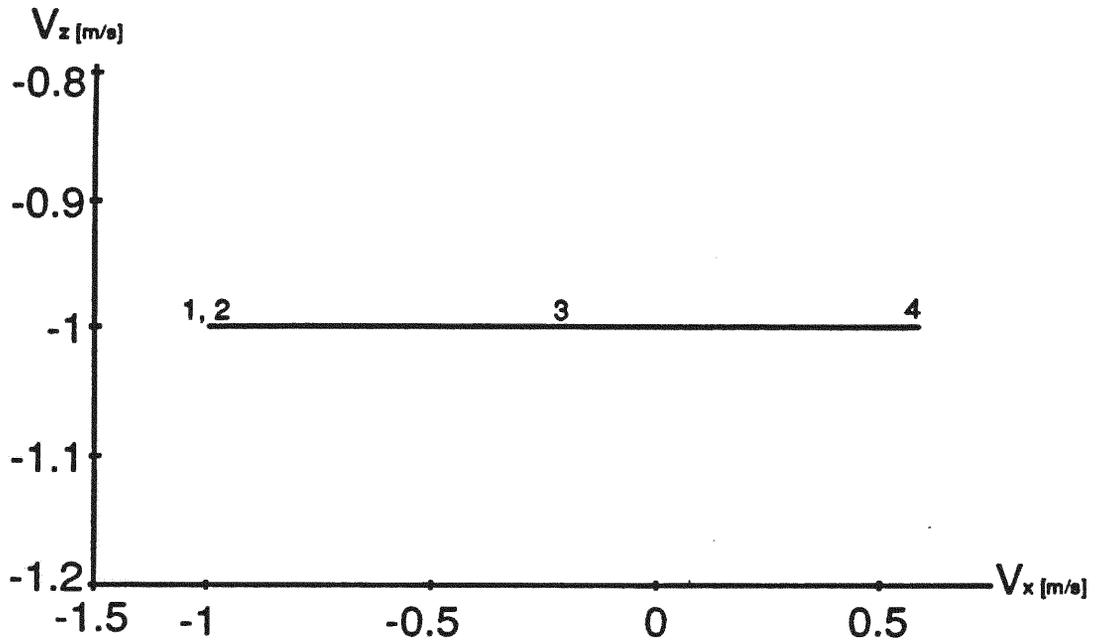


Figure 2.27a IBM 7576 Robot XZ Velocity Space Projection

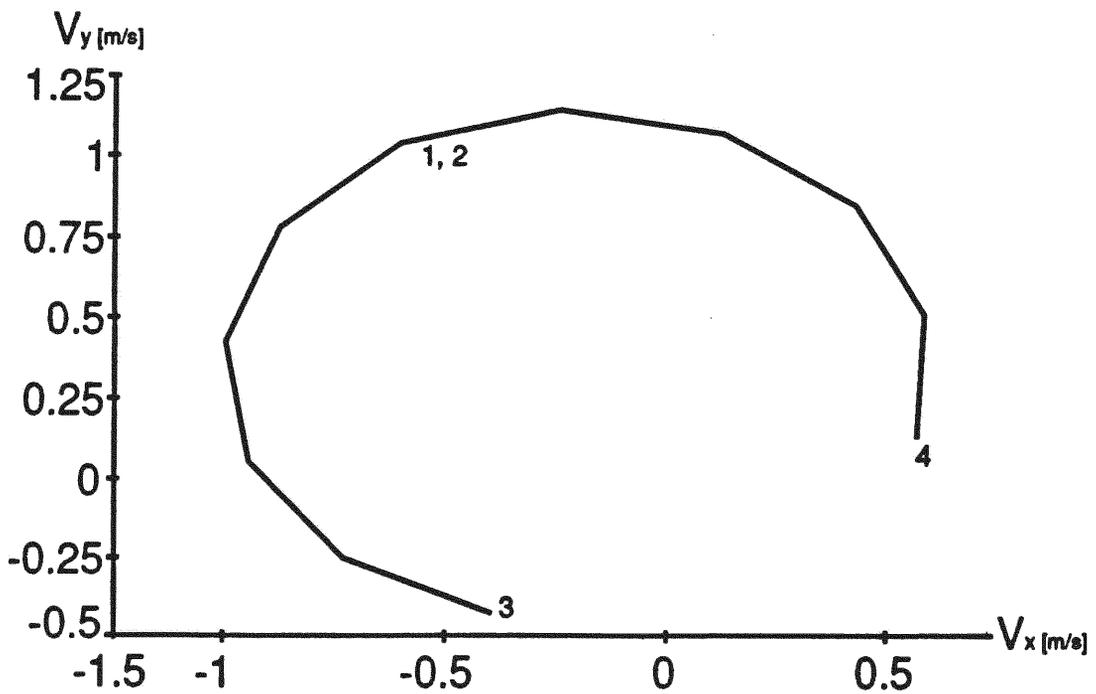


Figure 2.27b IBM 7576 Robot YZ Velocity Space Projection

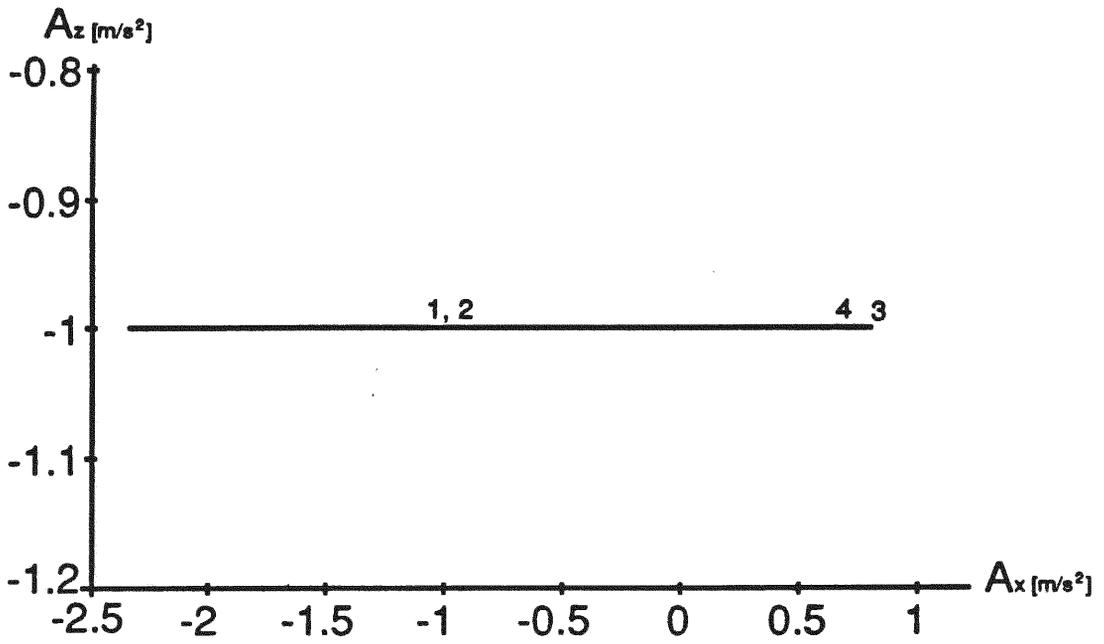


Figure 2.28a IBM 7576 Robot XZ Acceleration Space Projection

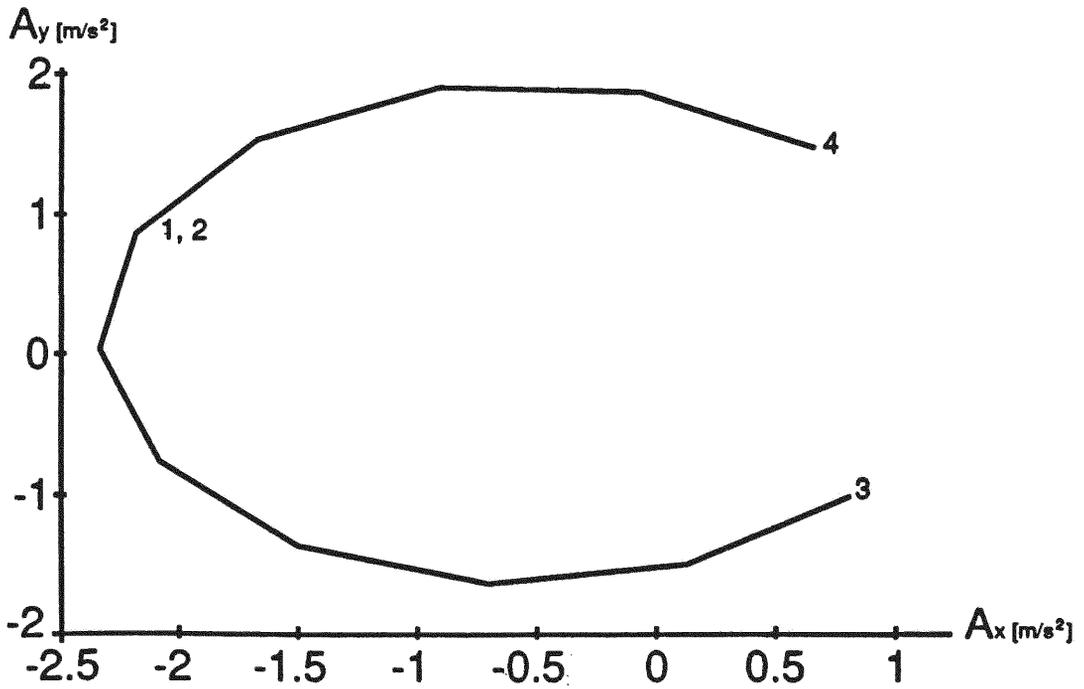


Figure 2.28b IBM 7576 Robot YZ Acceleration Space Projection

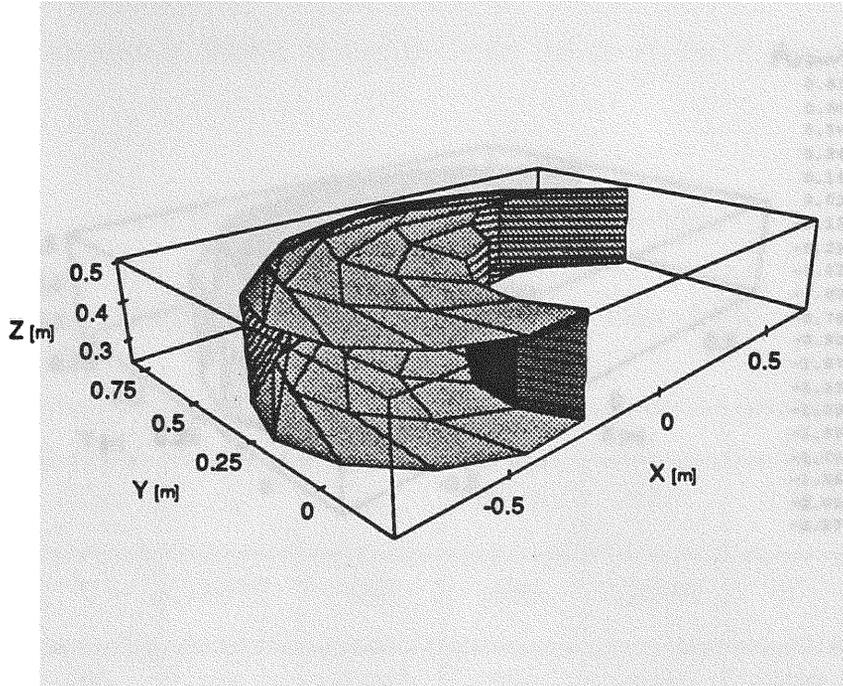


Figure 2.29 IBM 7576 Robot Workspace (viewp. (-1,-1,0.5) section 0°-120°)

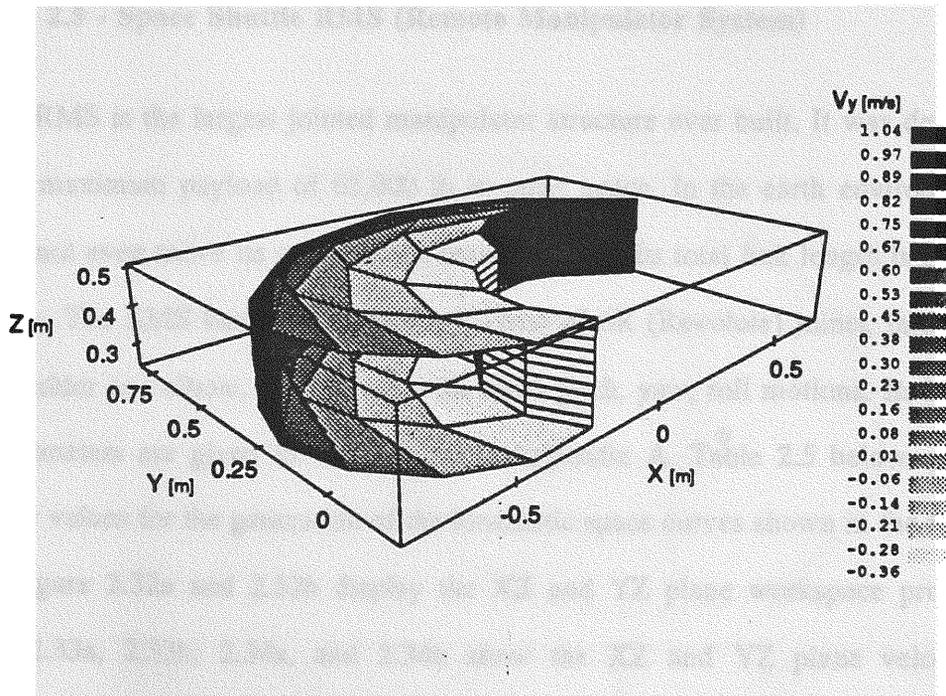


Figure 2.30 IBM 7576 Robot V_y Shaded Workspace (viewp. (-1,-1,0.5) sect. 0°-120°)

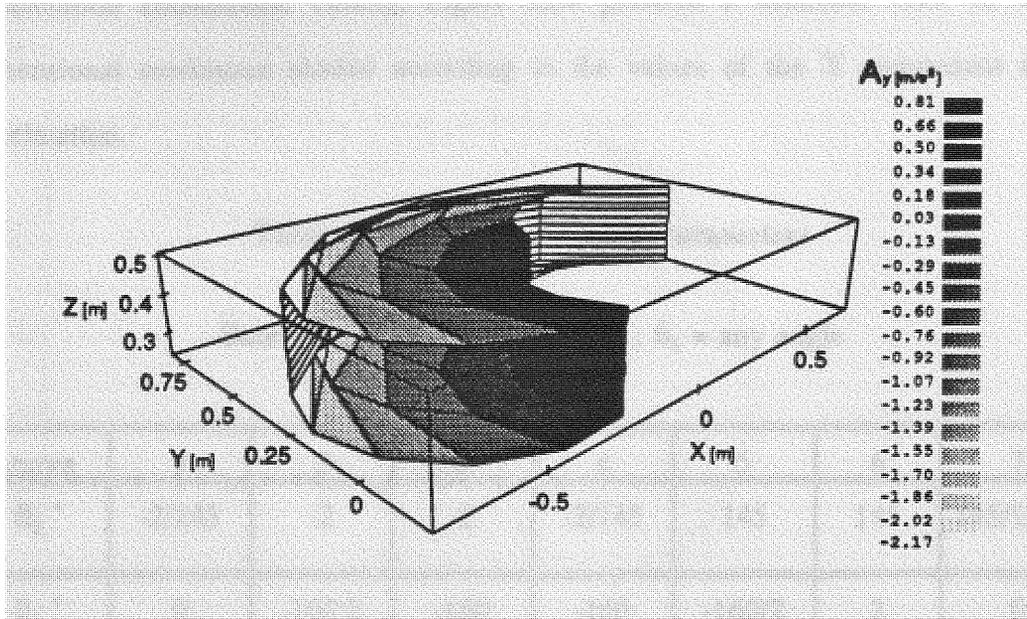


Figure 2.31 IBM 7576 Robot A_y Shaded Workspace (viewp. (-1,-1,0.5) sect. 0° - 120°)

Example 2.5 - Space Shuttle RMS (Remote Manipulator System)

The RMS is the largest jointed manipulator structure ever built. It was designed to handle a maximum payload of 65,000 lb in outer space. In the earth environment the RMS cannot even move its own linkage mass (905 lb). Its total link length is more than 15 meters. The RMS basic configuration consist of 6R (Revolute) joints, three for the base, shoulder and elbow, and three for the wrist pitch, yaw, roll motions. Its kinematic joint parameters are given in Table A.5a in Appendix A. Table 2.5 below, gives the parameter values for the generation of the kinematic space curves shown in the following pages. Figure 2.32a and 2.32b display the XZ and YZ plane workspace projections. Figures 2.33a, 2.33b, 2.34a, and 2.34b show the XZ and YZ plane velocity and acceleration space projections, respectively. Figure 2.35 presents a sectional view of the 3-dimensional workspace. Figure 2.36 shows a view of the velocity space plotted in

velocity coordinates. Figure 2.37 gives a representation of the acceleration space in acceleration coordinates. Finally, Figure 2.38 presents a sectional view of the 3-dimensional workspace shaded according to the values of the Y component of the acceleration.

Table 2.6 RMS Robot Curve Parameters

Base rotation $\theta_1 = 30^\circ$; $\theta_5 = 0^\circ$; $\theta_6 = \text{any angle}$

curve	1	2	3	4	5	6	7
θ_2°	-2/145	-2	-2	-2/145	145	145	145/125.4
θ_3°	0	-160/2	-160	-160	-160/2	2	0
θ_4°	0	0	-120/0	-120	-120	120/0	2

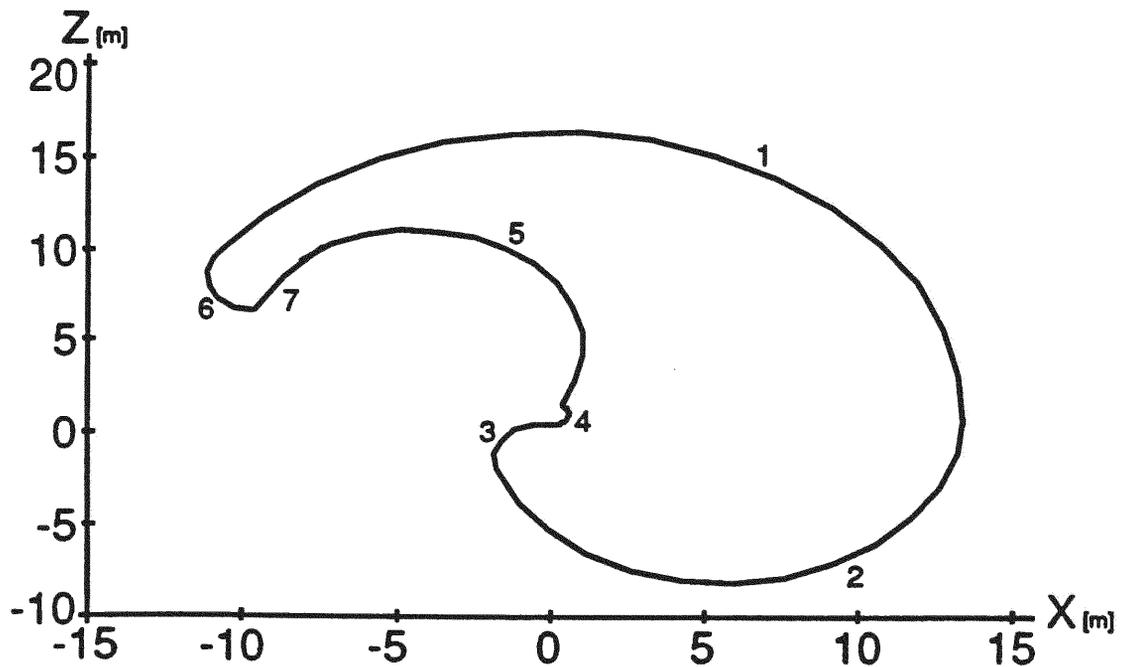


Figure 2.32a RMS Manipulator XZ Workspace Projection

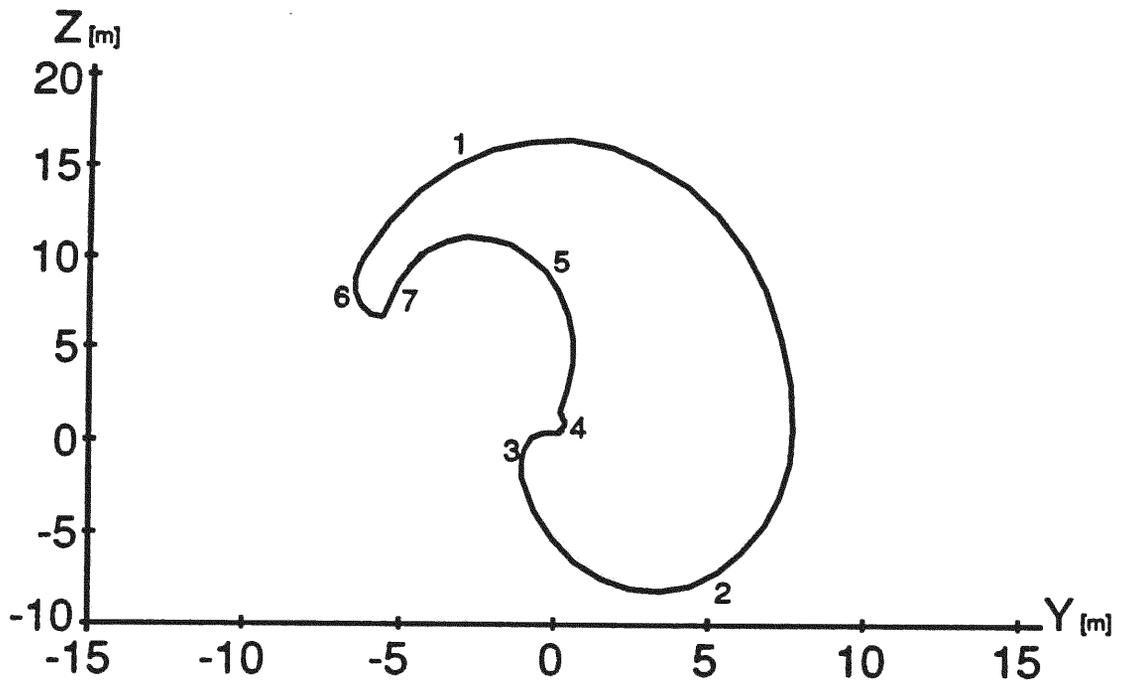


Figure 2.32b RMS Manipulator YZ Workspace Projection

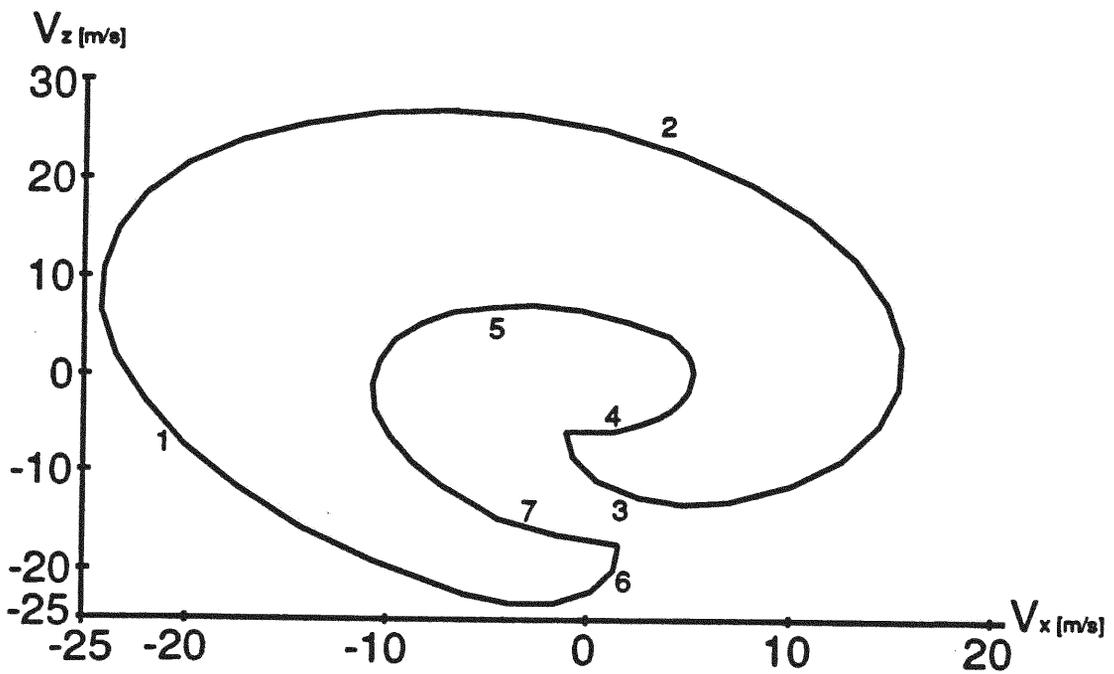


Figure 2.33a RMS Manipulator XZ Velocity Space Projection

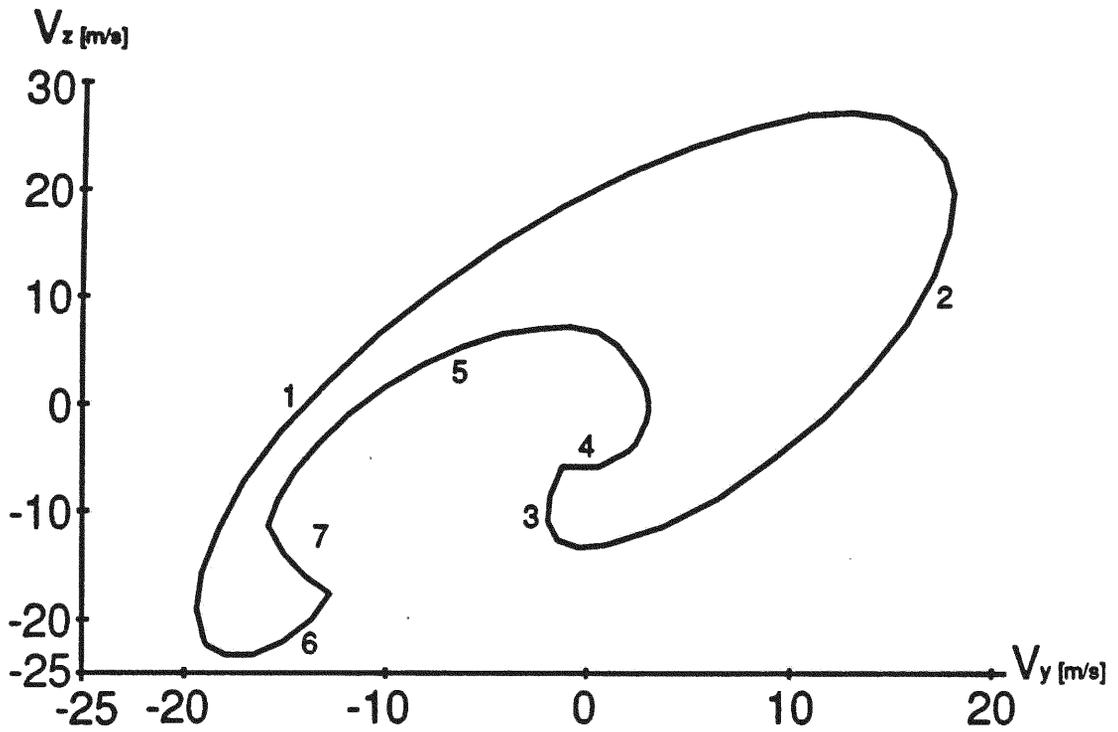


Figure 2.33b RMS Manipulator YZ Velocity Space Projection

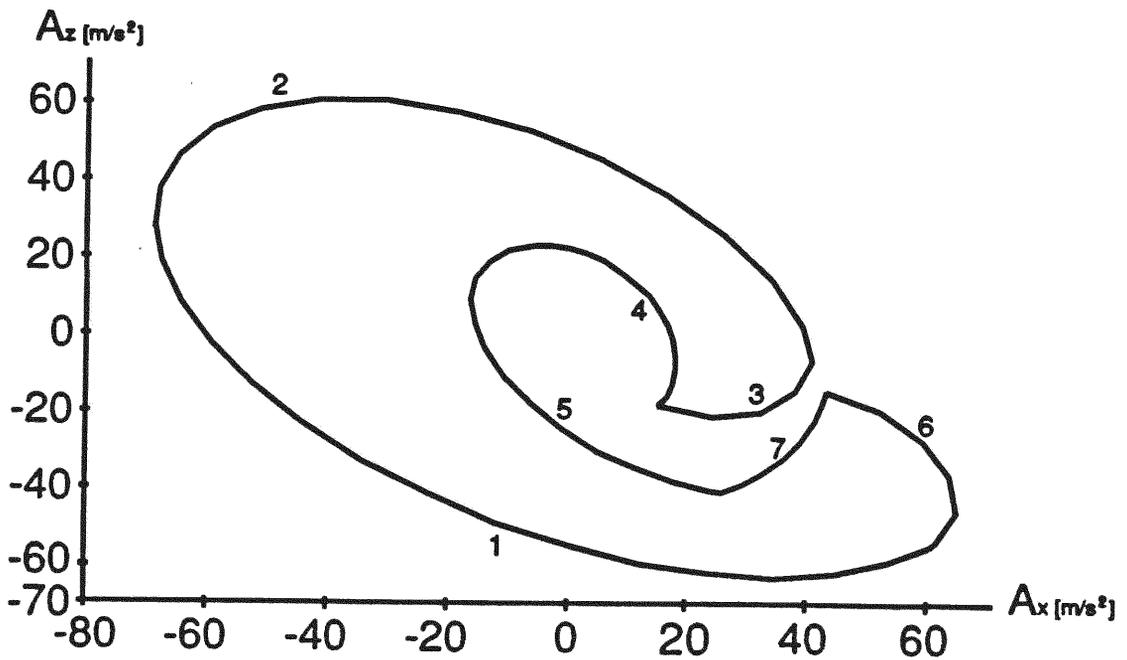


Figure 2.34a RMS Manipulator XZ Acceleration Space Projection

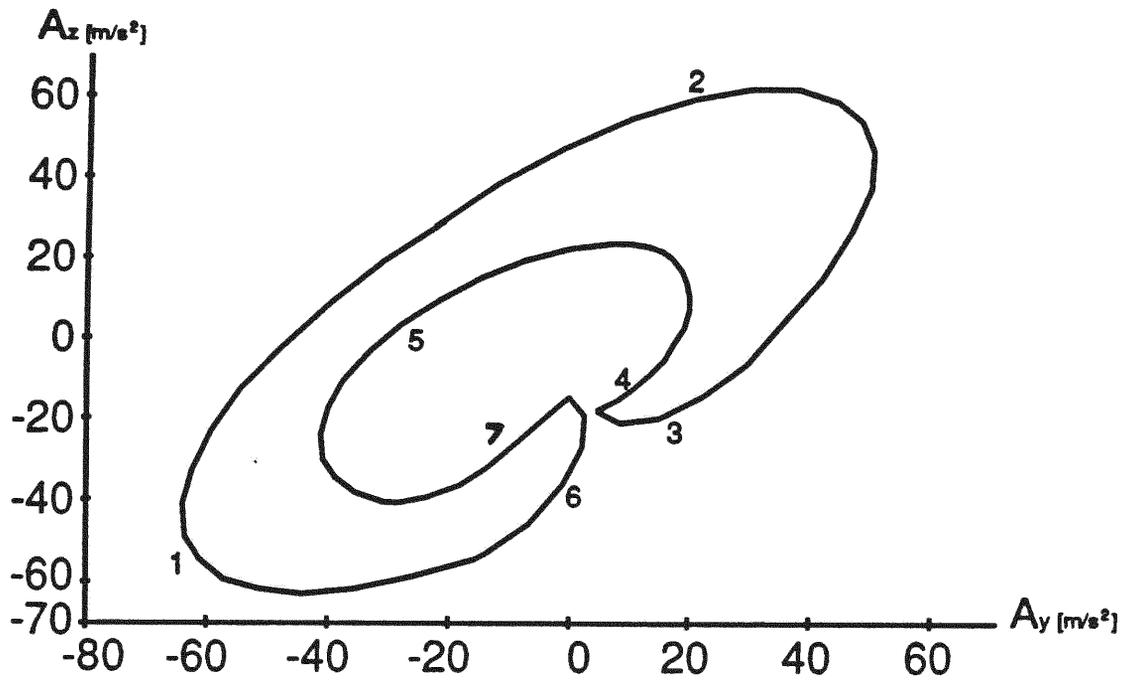


Figure 2.34b RMS Manipulator YZ Acceleration Space Projection

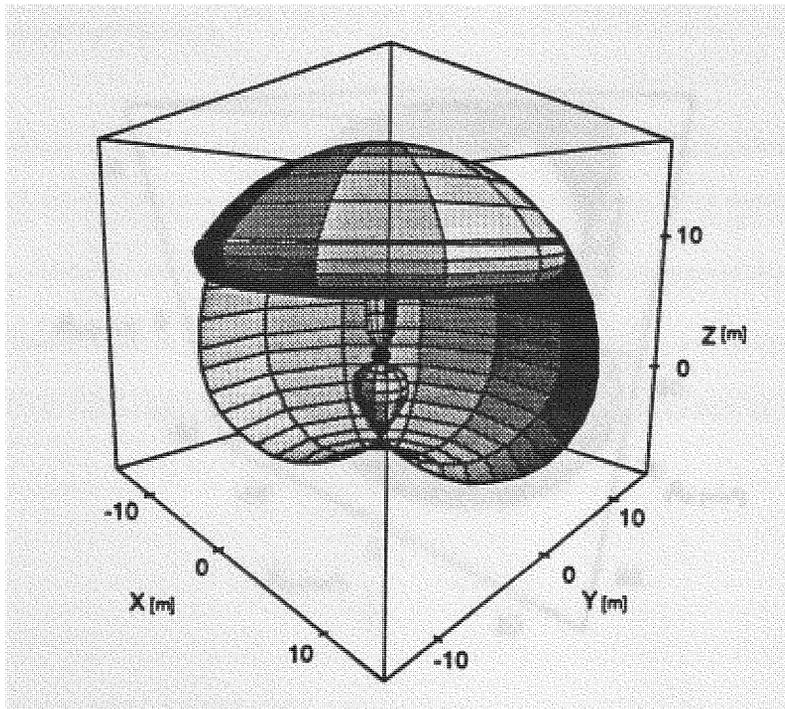


Figure 2.35 RMS Manipulator Workspace (viewp. (1,-1,0.2) sect. 0°-180°)

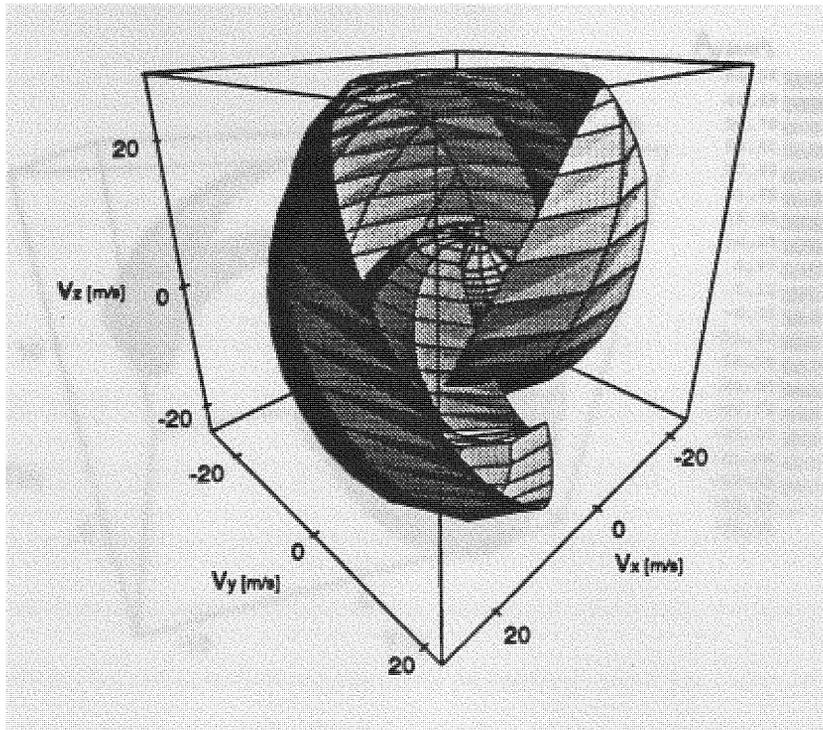


Figure 2.36 RMS Manipulator Velocity Space (viewp. (1,1,0.6) sect. 0° - 180°)

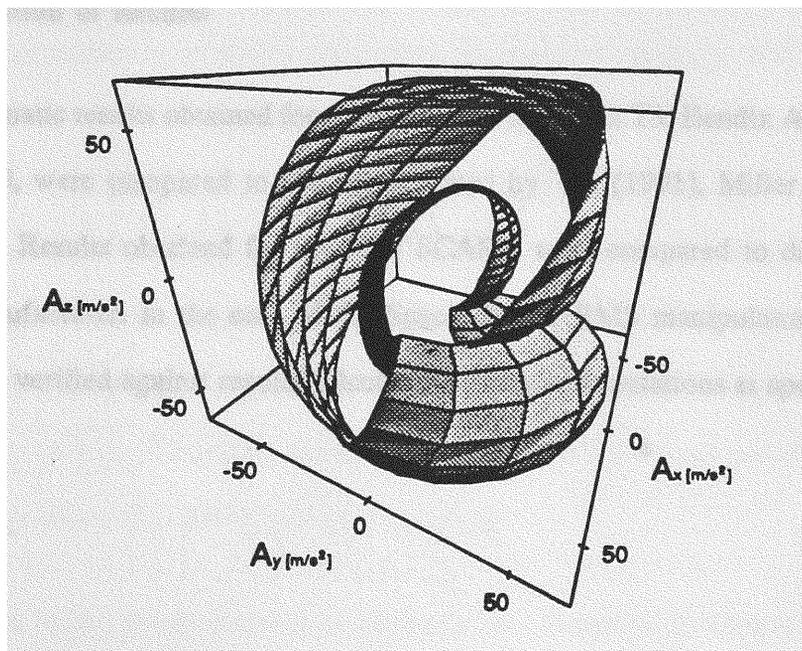


Figure 2.37 RMS Manipulator Acceleration Space (viewp. (1,-0.3,0.5) sect. 0° - 180°)

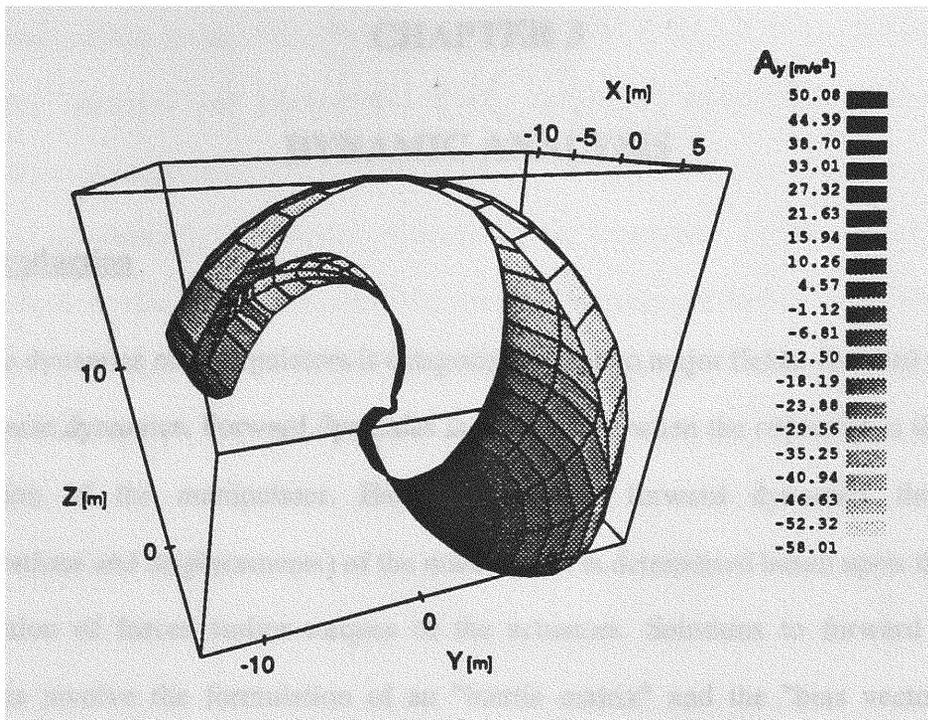


Figure 2.38 RMS Manipulator A_y Shaded Workspace (viewp. (1,-0.3,0.5) sect. 0°-90°)

2.4.3 Verification of Results

The kinematic results obtained for the Cincinnati Milacron T3, Bendix AA/CNC, and Unimate 2000, were compared to results presented by Yih (1991), Miller (1988), and other authors. Results obtained for the IBM SCARA were compared to data available from the manufacturer. In the case of the Space Shuttle RMS manipulator, the results obtained were verified against results calculated by hand computations at specific points.

CHAPTER 3

DYNAMIC ANALYSIS

3.1 Introduction

The dynamics of manipulators is categorized into two major fields: forward dynamics and inverse dynamics. Forward dynamics is emphasized when the objective is the motion simulation of the manipulator. Basically, through forward dynamics the motion (accelerations and displacements) of the manipulator is determined based upon the known information of forces and/or torques of the actuators. Solutions to forward dynamic problems involve the formulation of an "inertia matrix" and the "bias vector" which contains the terms of gravity, Coriolis and centripetal accelerations, and joint displacements. Inverse dynamic analysis determines the joint forces and moments required to produce the motions specified by a given set of positions, velocities, and accelerations. If the inverse dynamics is modelled accurately, the position, velocity, and acceleration information calculated or measured by sensors can be used to compute and control the forces and torques of the joint actuator required for a robot to follow a certain trajectory during acceleration and deceleration. There are three approaches available to arrive at a set of governing differential equations describing the dynamic behavior of a manipulator: 1) Bond Graph method (Shahninpoor, 1987; Samanta, 1990), 2) Newton-Euler dynamic approach (Orin, McGhee, Vukobratovic and Hartoch, 1979; Townsend and Gupta, 1989), and 3) Lagrange-Euler (Li, 1988) dynamic modeling.

In addition to the above mentioned methods, two recursive techniques have been developed. One is the Newton-Euler recursive approach (Luh, Walker and Paul, 1980; Bae, Hwang and Haug, 1991), and the other is the Lagrangian recursive approach

(Hollerbach, 1980; Book, 1984; Wang and Kohli, 1985). Silver (1980), and Unda, Garcia de Jalon, Losantos and Enparantza, (1987) presented comparative studies of the equivalence of Lagrangian and Newton-Euler dynamics. These two techniques have the advantage of reducing the number of computations drastically, by the use of recurrent relationships for the velocities, accelerations, and generalized forces. Luh, Walker and Paul (1979) pointed out the importance of inverse dynamics in control applications utilizing the Newton-Euler recursive formulation. In their approach, the link velocities and accelerations were solved from the base to the end link by kinematic analysis, and then utilized to solve joint torques from the end link to the base of the manipulator recursively. A similar recursive approach using the Lagrangian formulation was developed by Hollerbach (1980) in which 3x3 matrices instead of 4x4 homogeneous transformation matrices (Uicker, 1965) were used to increase computational efficiency. Book (1984) utilized the Lagrangian recursive approach to formulate the equations of motion of a flexible multilink manipulator, mixing a set of relative joint variables and the modal elastic degrees of freedom to define the system configuration space.

In this chapter, the matrix notation for the analysis of dynamic joint forces and moments for an n-link open-chain manipulator is derived. The n-link manipulator can be composed of any combination of the kinematic revolute, prismatic, and spherical joints. This formulation is developed based on the Newton-Euler dynamic equations under the quasi-static equilibrium condition; that is, the problem has been isolated from its time dependency. In other words, the problem is treated and solved for one specific position at a particular instant, and all the time dependent dynamic parameters needed for the solution are obtained in advance either through direct input or kinematic analysis. Also, link inertias are known, and displacements, velocities, and accelerations of each joint and link center of gravity (C.G.) are pre-calculated kinematically.

3.2 Static Analysis

In static analysis, the forces due to motion (inertia forces) are equal to zero since there is no motion (accelerations). For a body under static equilibrium, the sum of all forces and moments acting on any point in the body must be equal to zero. This condition is expressed by,

$$\sum \vec{F} = 0 \quad (3.1)$$

$$\sum \vec{M} = 0 \quad (3.2)$$

Equations (3.1) and (3.2) can be written in terms of X, Y, and Z components such that

$$\begin{aligned} \sum F_x &= 0 \\ \sum F_y &= 0 \end{aligned} \quad (3.3)$$

$$\sum F_z = 0$$

$$\begin{aligned} \sum M_x &= 0 \\ \sum M_y &= 0 \end{aligned} \quad (3.4)$$

$$\sum M_z = 0$$

The above equations describe the general case of three dimensional static equilibrium.

For the case of a robotic manipulator, the manipulator arm can be analyzed link by link

individually. Based on the free body diagram for link i shown in Fig. 3.2, given the position vectors definition of Fig. 3.1, and writing vectors in terms of their components, the conditions for static equilibrium are

$$\begin{aligned}\sum F_x &= F_{x_i} - F_{x_{i+1}} = 0 \\ \sum F_y &= F_{y_i} - F_{y_{i+1}} = 0 \\ \sum F_z &= F_{z_i} - F_{z_{i+1}} - m_i g = 0\end{aligned}\quad (3.5)$$

and,

$$\begin{aligned}\sum M_x &= M_{x_i} + F_{z_i} * r_{y_i} - F_{y_i} * r_{z_i} - F_{z_{i+1}} * r'_{y_i} + F_{y_{i+1}} * r'_{z_i} - M_{x_{i+1}} = 0 \\ \sum M_y &= M_{y_i} + F_{x_i} * r_{z_i} - F_{z_i} * r_{x_i} - F_{x_{i+1}} * r'_{z_i} + F_{z_{i+1}} * r'_{x_i} - M_{y_{i+1}} = 0 \\ \sum M_z &= M_{z_i} + F_{y_i} * r_{z_i} - F_{x_i} * r_{y_i} - F_{y_{i+1}} * r'_{x_i} + F_{x_{i+1}} * r'_{y_i} - M_{z_{i+1}} = 0\end{aligned}\quad (3.6)$$

The following section 3.2.1 contains the formulations explaining the relationships between relative and absolute vectors.

3.2.1 Relative Joint Vectors

From Fig. 3.1, \vec{R} vectors represent absolute position vectors, and \vec{r} vectors the relative position vectors. Based on Fig. 3.1, the following relationships can be established

$$\vec{R}_{i+1} = \vec{R}_i + \vec{r}_{i,i+1}$$

or,

$$\vec{r}_{i,i+1} = \vec{R}_{i+1} - \vec{R}_i \quad (3.7)$$

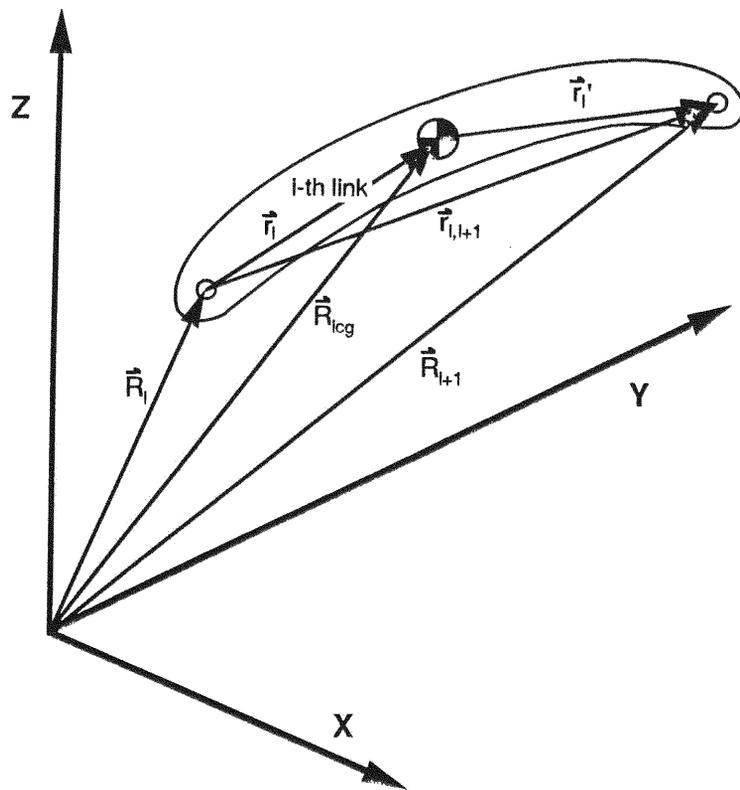


Figure 3.1 Relative position vectors on the i -th link

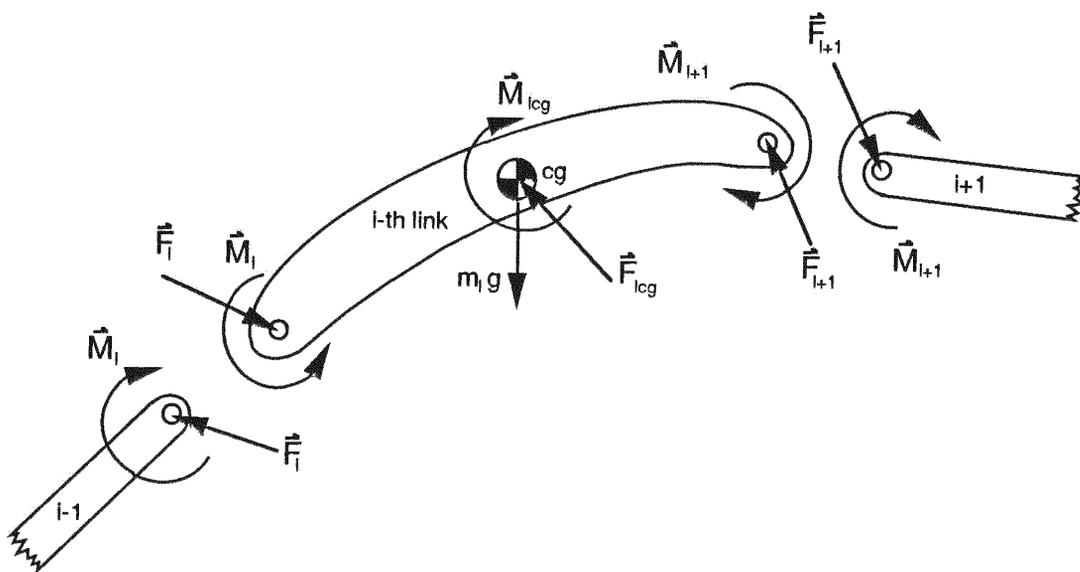


Figure 3.2 Static free-body diagram of the i -th link

Now, provided that the C.G. of the i -th link is on the line of joints i and $i+1$, the parameter μ ($0 \leq \mu \leq 1$) is defined such that,

$$\vec{r}_i = \mu \vec{r}_{i,i+1}$$

and,

$$\vec{r}'_i = (1-\mu)\vec{r}_{i,i+1} \quad (3.8)$$

Therefore the relative position vector of the i -th link C.G. is given by,

$$\vec{r}_{i_{cg}} = \vec{r}_i = \mu(\vec{R}_{i+1} - \vec{R}_i) \quad (3.9)$$

Following a similar argument for the velocity and acceleration, the following equations are derived.

$$\begin{aligned} \vec{v}_{i_{cg}} &= \mu \vec{v}_i = \mu(\vec{V}_{i+1} - \vec{V}_i) \\ \vec{a}_{i_{cg}} &= \mu \vec{a}_i = \mu(\vec{A}_{i+1} - \vec{A}_i) \end{aligned} \quad (3.10)$$

Analytical solutions for the joint forces and moments for a complete manipulator structure may be calculated by solving sets of linear equations obtained from equations (3.5) and (3.6), provided the required information is available (forces and moments at end effector, position vectors, and so on).

3.3 Dynamic Analysis

The procedure for the dynamic analysis to be used here is based on the Newton and Euler dynamic equations. This procedure is basically a recursive method, in which joint

forces and moments are calculated recursively from the end-effector backward to the base of the robotic manipulator. Initially, joint positions, velocities, and accelerations must be calculated in order to determine the joint forces and torques required to cause such motions. Based on D'Alembert principle, the dynamic equations are derived from the static analysis.

3.3.1 Newton-Euler Formulation

Considering that a robot is composed of a series of rigid links connected by joints, a center of mass for each link can be defined. For a rigid body system with such characteristics, Newton's equation of motion states that the force \vec{F} and acceleration \vec{a} are related by the total mass of the body

$$\vec{F}_{cg} = m\vec{a}_{cg} \quad (3.11)$$

In a similar way, Euler's equation of motion describes the rotational motion and acceleration of a rigid body. This relationship relates torque, angular velocity and acceleration, and mass moment of inertia such that,

$$\vec{M}_{cg} = I_{cg}\vec{\alpha}_{cg} + \vec{\omega}_{cg} \times I_{cg}\vec{\omega}_{cg} \quad (3.12)$$

where \vec{M}_{cg} is the moment of the body about its center of gravity, $\vec{\omega}_{cg}$ the angular velocity, $\vec{\alpha}_{cg}$ the angular acceleration, and I_{cg} the body mass moment of inertia. Subscript cg denotes the center of gravity of the body in question.

For the i-th link, equation (3.11) can be written in terms of its x, y, and z components

as,

$$F_{x_{i\text{cg}}} = m_i a_{x_{i\text{cg}}} \quad (3.13)$$

where $a_{x_{i\text{cg}}}$, $a_{y_{i\text{cg}}}$, and $a_{z_{i\text{cg}}}$ represent the relative acceleration of the i -th link C.G. with respect to the i -th joint, and $F_{x_{i\text{cg}}}$, $F_{y_{i\text{cg}}}$, and $F_{z_{i\text{cg}}}$ are the components of the force (inertial) about the C.G. of the i -th link relative to the i -th joint.

Similarly, Euler's equation (3.12) can be expanded into its components and the final relationships are given by the expressions in (3.14). In these expressions, $M_{x_{i\text{cg}}}$, $M_{y_{i\text{cg}}}$, and $M_{z_{i\text{cg}}}$ are the components of the moments about the i -th link C.G. relative to the i -th joint.

$$\begin{aligned} M_{x_{i\text{cg}}} &= I_{xx_{i\text{cg}}} \alpha_{x_i} + \omega_{y_i} \omega_{z_i} (I_{zz_{i\text{cg}}} - I_{yy_{i\text{cg}}}) + I_{xy_{i\text{cg}}} (\omega_{z_i} \omega_{x_i} - \alpha_{y_i}) \\ &\quad - I_{xz_{i\text{cg}}} (\alpha_{z_i} + \omega_{y_i} \omega_{x_i}) - I_{yz_{i\text{cg}}} (\omega_{y_i}^2 - \omega_{z_i}^2) \\ M_{y_{i\text{cg}}} &= I_{yy_{i\text{cg}}} \alpha_{y_i} + \omega_{z_i} \omega_{x_i} (I_{xx_{i\text{cg}}} - I_{zz_{i\text{cg}}}) + I_{yz_{i\text{cg}}} (\omega_{x_i} \omega_{y_i} - \alpha_{z_i}) \\ &\quad - I_{xy_{i\text{cg}}} (\alpha_{x_i} + \omega_{z_i} \omega_{y_i}) - I_{xz_{i\text{cg}}} (\omega_{z_i}^2 - \omega_{x_i}^2) \\ M_{z_{i\text{cg}}} &= I_{zz_{i\text{cg}}} \alpha_{z_i} + \omega_{x_i} \omega_{y_i} (I_{yy_{i\text{cg}}} - I_{xx_{i\text{cg}}}) + I_{xz_{i\text{cg}}} (\omega_{y_i} \omega_{z_i} - \alpha_{x_i}) \\ &\quad - I_{yz_{i\text{cg}}} (\alpha_{y_i} + \omega_{x_i} \omega_{z_i}) - I_{xy_{i\text{cg}}} (\omega_{x_i}^2 - \omega_{y_i}^2) \end{aligned} \quad (3.14)$$

where all angular velocity and acceleration components are relative to the i -th joint.

In the following, Newton-Euler equations for the kinematic revolute, prismatic, and spherical pairs are derived from equations (3.13) and (3.14).

A) Revolute Pair

A revolute joint has only one degree of freedom. This single degree of freedom is the relative rotation of the i -th joint with respect to the $(i-1)$ th joint about the local z -axis.

Consequently, ω_x , ω_y , α_x , and α_y equal to zero, and the Newton-Euler equations become,

$$\begin{aligned} F_{x_{i_{cg}}} &= m_i a_{x_{i_{cg}}} \\ F_{y_{i_{cg}}} &= m_i a_{y_{i_{cg}}} \\ F_{z_{i_{cg}}} &= m_i a_{z_{i_{cg}}} \end{aligned} \quad (3.15)$$

$$\begin{aligned} M_{x_{i_{cg}}} &= I_{xz_{i_{cg}}} \alpha_{z_i} - I_{yz_{i_{cg}}} \omega_{z_i}^2 \\ M_{y_{i_{cg}}} &= I_{yz_{i_{cg}}} \alpha_{z_i} - I_{xz_{i_{cg}}} \omega_{z_i}^2 \\ M_{z_{i_{cg}}} &= I_{zz_{i_{cg}}} \alpha_{z_i} \end{aligned} \quad (3.16)$$

B) Prismatic Pair

A prismatic joint is constrained to a linear motion along its joint axis, say z-axis. It has no rotational degrees of freedom, therefore all relative angular velocities and accelerations are equal to zero. The Newton and Euler equations can be simplified as

$$\begin{aligned} F_{x_{i_{cg}}} &= m_i a_{x_{i_{cg}}} \\ F_{y_{i_{cg}}} &= m_i a_{y_{i_{cg}}} \\ F_{z_{i_{cg}}} &= m_i a_{z_{i_{cg}}} \end{aligned} \quad (3.17)$$

where $a_{x_{i_{cg}}}$, $a_{y_{i_{cg}}}$, and $a_{z_{i_{cg}}}$ represent the three components of linear acceleration of the i-th link C.G. relative to the i-th joint. Also,

$$\begin{aligned} M_{x_{i_{cg}}} &= 0 \\ M_{y_{i_{cg}}} &= 0 \\ M_{z_{i_{cg}}} &= 0 \end{aligned} \quad (3.18)$$

C) Spherical Pair

A spherical pair has three degrees of freedom, each rotates about one cartesian coordinate axis indicated by $(\psi_{x_i}, \psi_{y_i}, \psi_{z_i})$. Based on the C-B notation (Yih, 1991), design variable ψ_{z_i} is equivalent to θ_i . Also, parameters ψ_{x_i} and ψ_{y_i} are initially implied by design variables h_i and r_i , which in turn are defined as functions of ϕ_i solely. As a result, the original three degrees of freedom $(\psi_{x_i}, \psi_{y_i}, \psi_{z_i})$ are reduced to only two (θ_i, ϕ_i) . Based on the above mentioned concepts, angular velocities and accelerations about the x- and y- axes are replaced by $\dot{\phi}_i$ and $\ddot{\phi}_i$. Then, the Newton and Euler dynamic equations become,

$$\begin{aligned} F_{x_{i,cs}} &= m_i a_{x_{i,cs}} \\ F_{y_{i,cs}} &= m_i a_{y_{i,cs}} \\ F_{z_{i,cs}} &= m_i a_{z_{i,cs}} \end{aligned} \quad (3.19)$$

and,

$$\begin{aligned} M_{x_{i,cs}} &= I_{yy_{i,cs}} \ddot{\phi}_i + I_{xz_{i,cs}} \alpha_{z_i} + \dot{\phi}_i (I_{yz_{i,cs}} \dot{\phi}_i + I_{zz_{i,cs}} \omega_{z_i}) \\ &\quad - \omega_{z_i} (I_{yy_{i,cs}} \dot{\phi}_i + I_{yz_{i,cs}} \omega_{z_i}) \\ M_{y_{i,cs}} &= I_{yy_{i,cs}} \ddot{\phi}_i + I_{yz_{i,cs}} \alpha_{z_i} - \omega_{z_i} (I_{xy_{i,cs}} \dot{\phi}_i + I_{xz_{i,cs}} \omega_{z_i}) \\ M_{x_{i,cs}} &= I_{yz_{i,cs}} \ddot{\phi}_i + I_{zz_{i,cs}} \alpha_{z_i} - \dot{\phi}_i (I_{xy_{i,cs}} \dot{\phi}_i + I_{xz_{i,cs}} \omega_{z_i}) \end{aligned} \quad (3.20)$$

3.3.2 General Matrix Notation

Consider the position vectors of the i-th link shown in Fig. 3.1, and the free body diagram shown in Fig. 3.2. Based on D'Alembert principle, the following dynamic equilibrium equations can be formulated

$$\begin{aligned}
F_{x_{i_{cg}}} &= F_{x_i} - F_{x_{i+1}} = m_i a_{x_{i_{cg}}} \\
F_{y_{i_{cg}}} &= F_{y_i} - F_{y_{i+1}} = m_i a_{y_{i_{cg}}} \\
F_{z_{i_{cg}}} &= F_{z_i} - F_{z_{i+1}} - m_i g = m_i a_{z_{i_{cg}}}
\end{aligned} \tag{3.21}$$

and,

$$\begin{aligned}
M_{x_{i_{cg}}} &= -F_{y_i} * r_{z_i} + F_{z_i} * r_{y_i} + M_{x_i} + F_{y_{i+1}} * r'_{z_i} - F_{z_{i+1}} * r'_{y_i} - M_{x_{i+1}} \\
M_{y_{i_{cg}}} &= F_{x_i} * r_{z_i} - F_{z_i} * r_{x_i} + M_{y_i} - F_{x_{i+1}} * r'_{z_i} + F_{z_{i+1}} * r'_{x_i} - M_{y_{i+1}} \\
M_{z_{i_{cg}}} &= -F_{x_i} * r_{y_i} + F_{y_i} * r_{x_i} + M_{z_i} + F_{x_{i+1}} * r'_{y_i} - F_{y_{i+1}} * r'_{x_i} - M_{z_{i+1}}
\end{aligned} \tag{3.22}$$

where $F_{x_{i_{cg}}}$, $F_{y_{i_{cg}}}$, and $F_{z_{i_{cg}}}$ are given in equations (3.15), (3.17) and (3.19), and $M_{x_{i_{cg}}}$, $M_{y_{i_{cg}}}$, and $M_{z_{i_{cg}}}$ are expressed by equations (3.16), (3.18) and (3.20) for a revolute, prismatic, and spherical pair, respectively.

Furthermore, equations (3.21) and (3.22) can be rearranged into a general matrix form for any n-link robot manipulator. The resulting matrix expression is given in equation (3.24). The left hand side of the resulting expressions are grouped into a column vector [b] of length 6n, containing the relative forces and moments about the C.G. of each link. Terms M_{x_L} , M_{y_L} , and M_{z_L} are the components of the moment due to the external load. The right hand side of the equation is a relative displacement coefficient matrix [A] of order 6nx6n, and the unknown joint forces and moments column vector [x] of length 6n.

Once Newton-Euler forces and moments column vector [b] and displacement coefficient matrix [A] are formed, the dynamic solutions for joint forces and moments can be calculated by the product of A^{-1} and [b],

$$[x] = [A]^{-1}[b] \tag{3.23}$$

3.3.3 Numerical Examples

In this section, the general matrix method explained previously is applied to determine the joint forces and moments needed to move the manipulator's arm according to specific (relative) joint position, velocity and acceleration. Five robots are analyzed utilizing the developed RaMIP (Robot and Mechanism Integrated Program). For each one of these robots, joint forces and moments are calculated at several positions during the motion of one of the joints. That is, the robot is actuated by the motion of one joint while all others remain fixed. The results are presented in terms of forces and moment components versus the moving joint variable.

Example 3.1 - Cincinnati Milacron T3 Robot (RRR/RRR)

The kinematic and dynamic data of this robot is presented in Tables A.1a and A.1b. Figures 3.4a - 3.5d presented next, resulted from the dynamic analysis of the six-jointed Cincinnati Milacron T-3 robot. Joint forces and moments are calculated at several points in the motion of elbow joint 3, while all others remain fixed at specific positions (see section C.1 in appendix C). In this example, the manipulator carries a 25 Kg payload at its end effector. From the figures, it is noticed that the forces and moments at joints 4, 5, and 6 are smaller when compared to the same quantities at joints 1, 2, and 3. This result is expected, as forces and moments tend to add up towards the base of the manipulator arm.

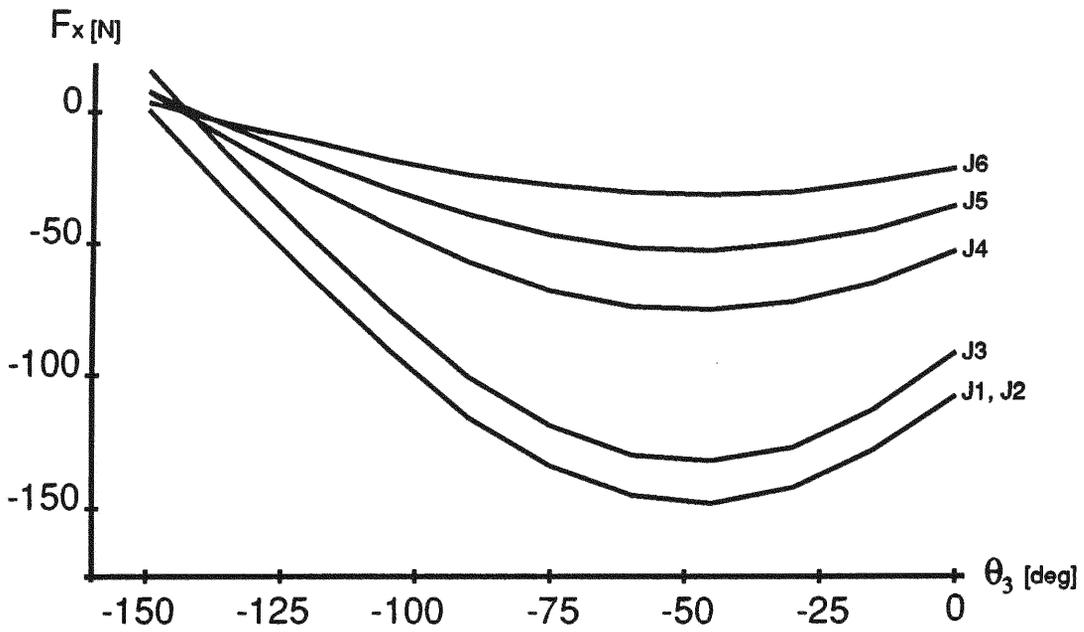


Figure 3.4a Cincinnati Milacron T3 robot, joint force x-component versus elbow joint θ_3 angle

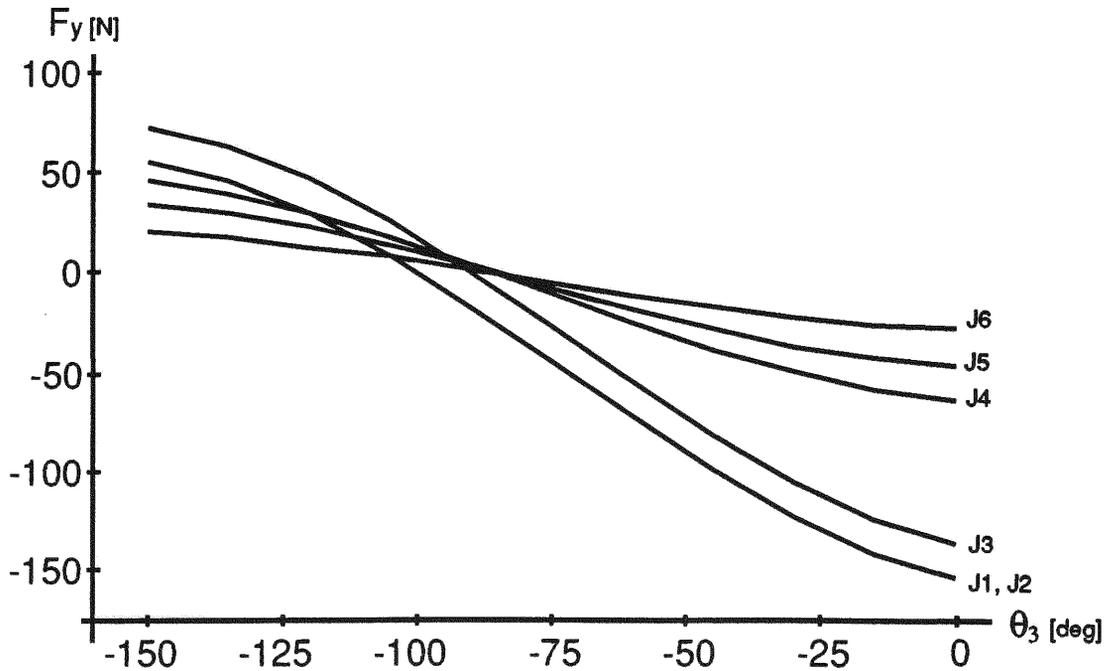


Figure 3.4b Cincinnati Milacron T3 robot, joint force y-component versus elbow joint θ_3 angle

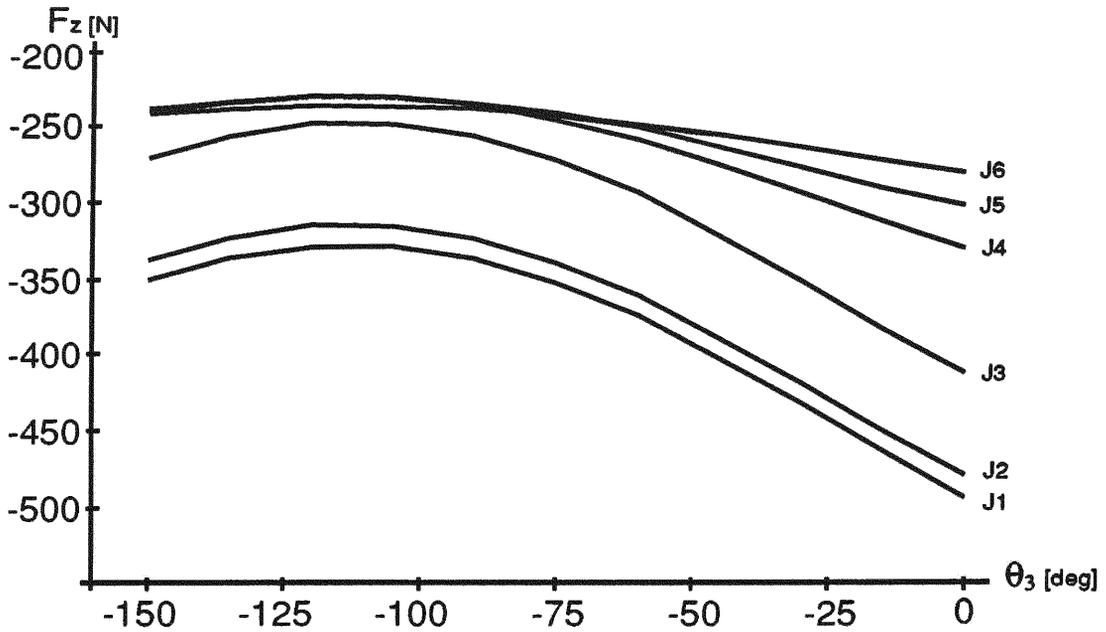


Figure 3.4c Cincinnati Milacron T3 robot, joint force z-component versus elbow joint θ_3 angle

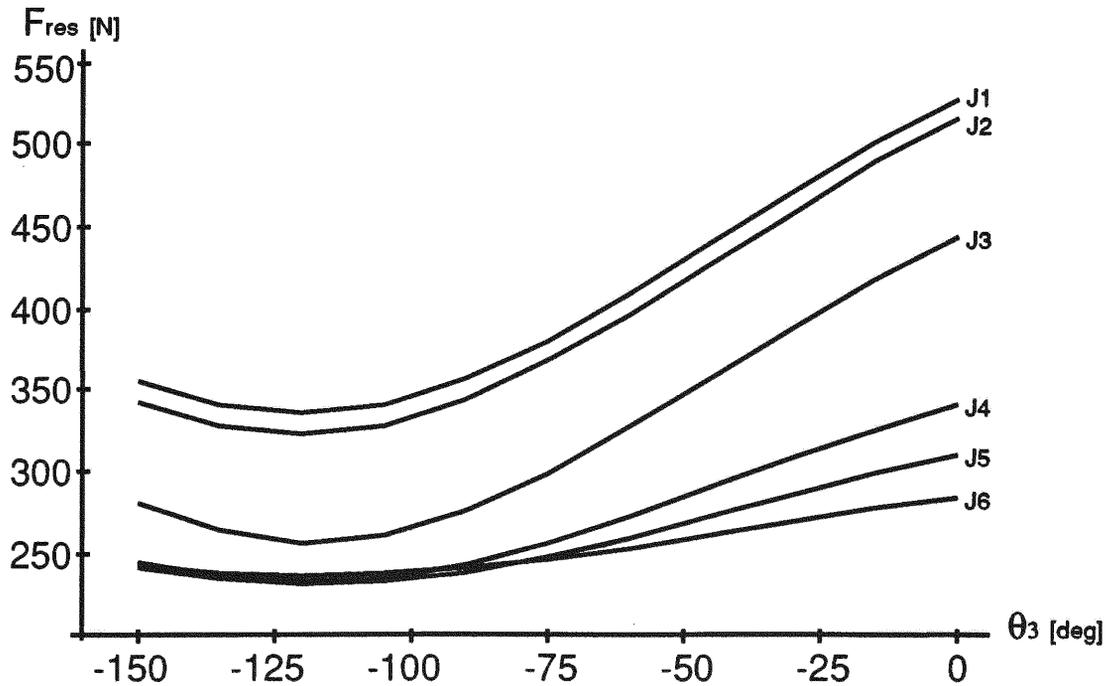


Figure 3.4d Cincinnati Milacron T3 robot, joint resultant force (F_{res}) versus elbow joint angle θ_3

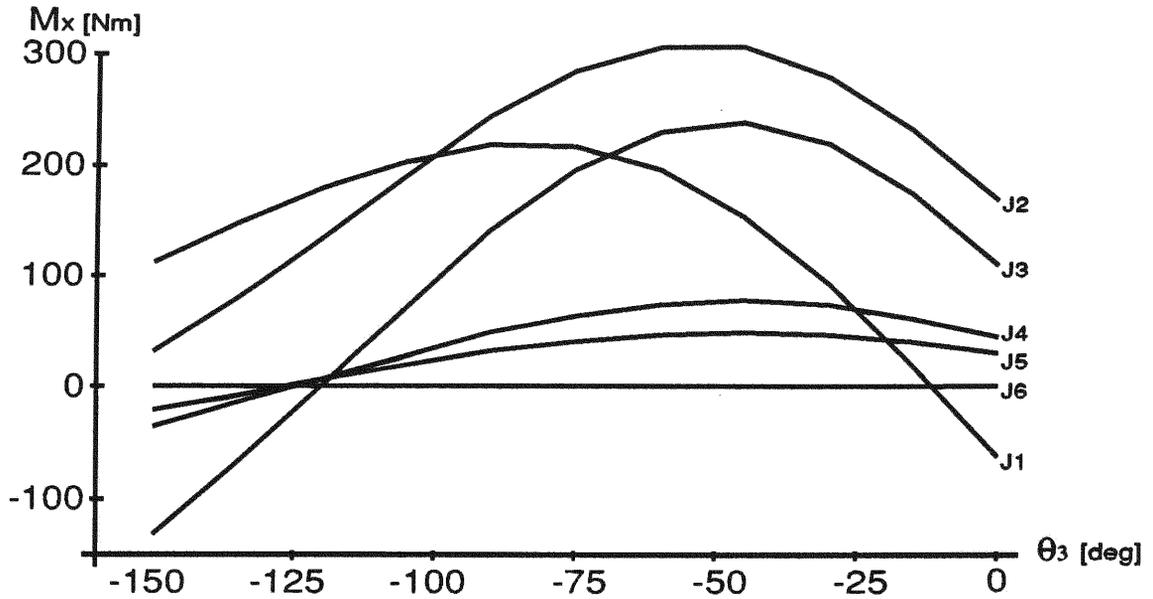


Figure 3.5a Cincinnati Milacron T3 robot, joint moment x-component versus elbow joint θ_3 angle

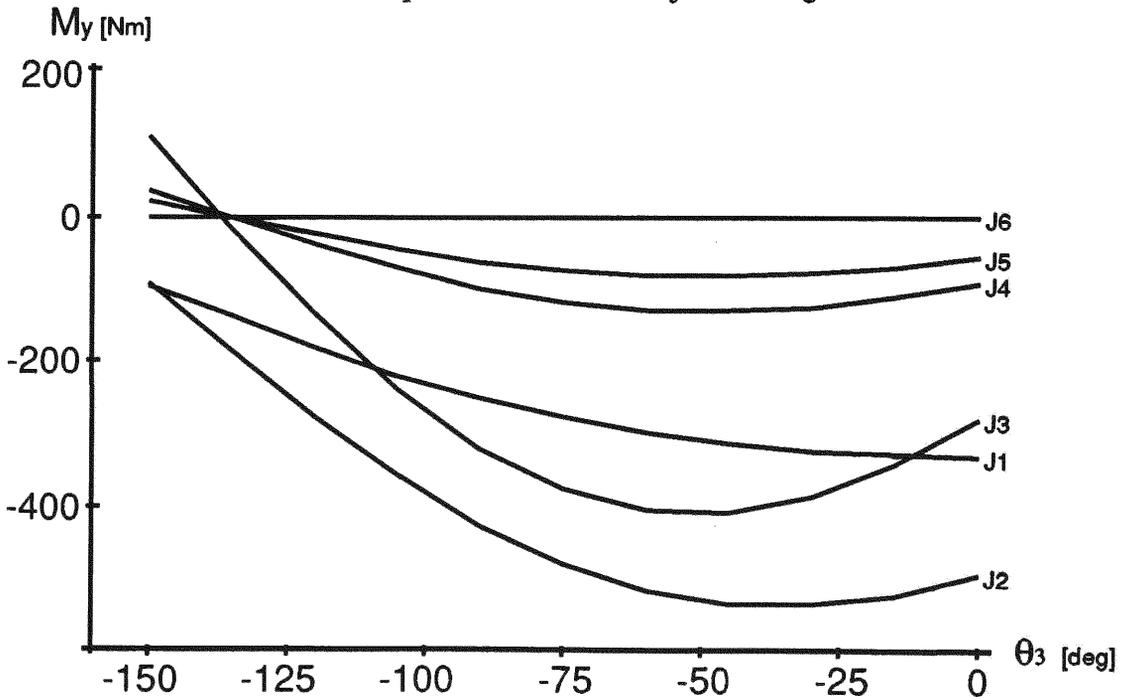


Figure 3.5b Cincinnati Milacron T3 robot, joint moment y-component versus elbow joint θ_3 angle

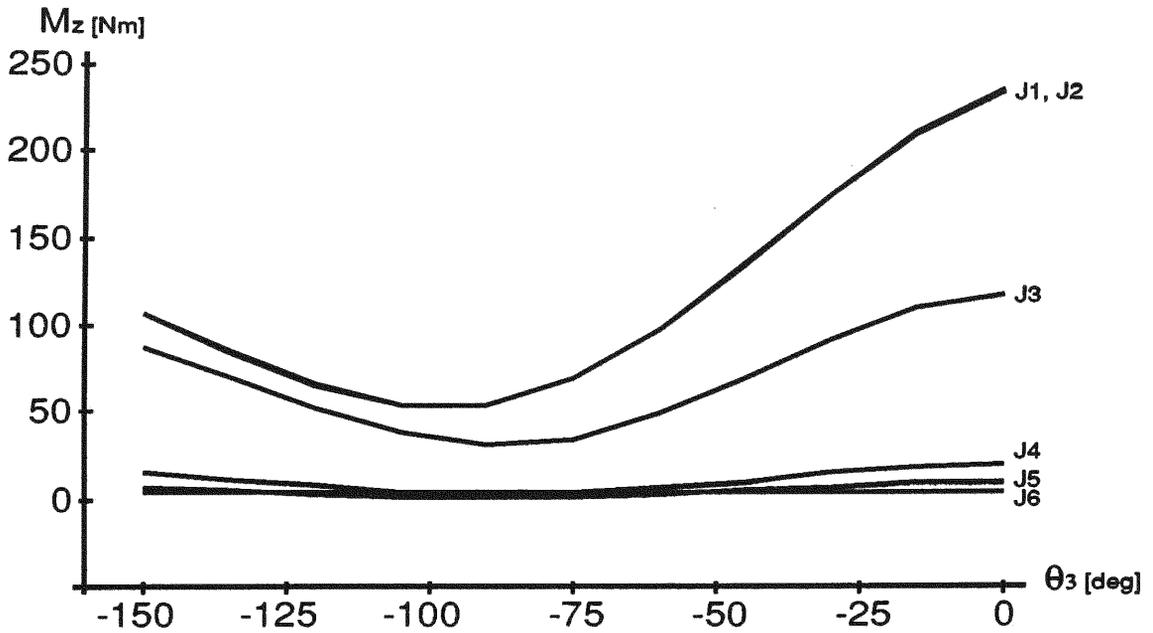


Figure 3.5c Cincinnati Milacron T3 robot, joint moment z-component versus elbow joint θ_3 angle

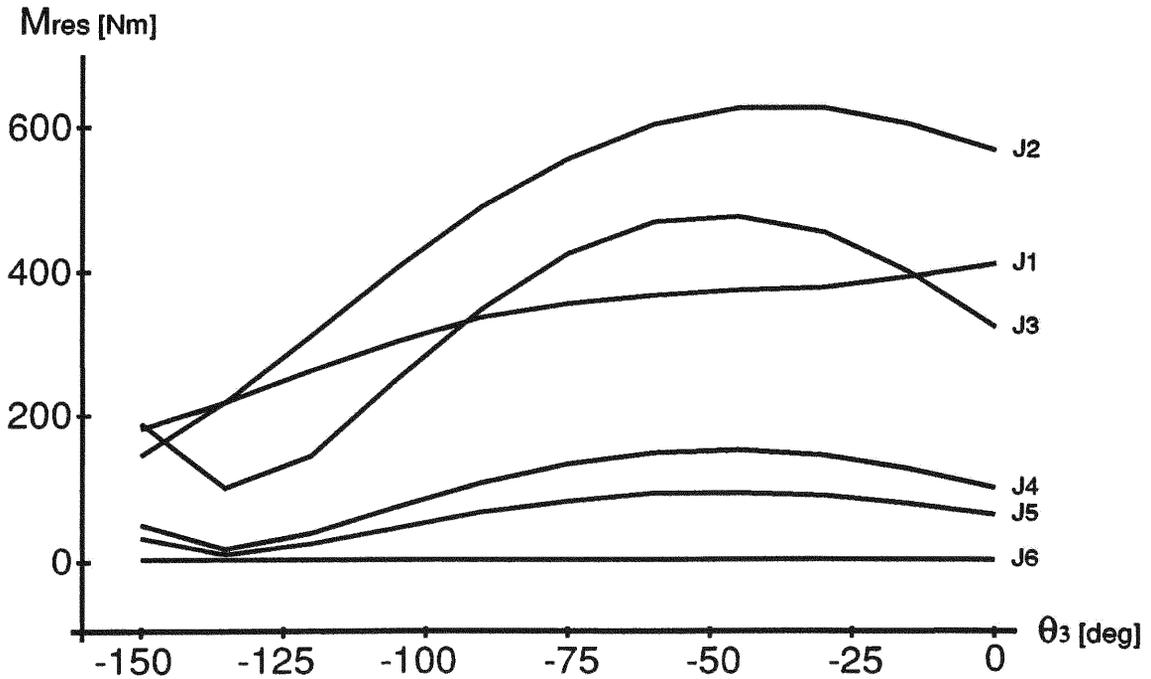


Figure 3.5d Cincinnati Milacron T3 robot, joint resultant (M_{res}) moment versus elbow joint θ_3 angle

Example 3.2 - Bendix AA/CNC Industrial Robot (RRP/RRR)

The geometry of this manipulator consists of one prismatic and five revolute joints. All the kinematic and dynamic data for this robot are included in Tables A.2a and A.2b in Appendix A. In this example the robot carries a 25 Kg load at its end effector. In order to show the generality of the matrix method, prismatic elbow joint 3 is allowed to move throughout its entire motion span. In Figs. 3.6a - 3.7d, the output obtained characterizes the dynamic response resulting from the motion of a prismatic joint. The linear motion of this prismatic joint generates a nearly linear variation of joint forces and moments.

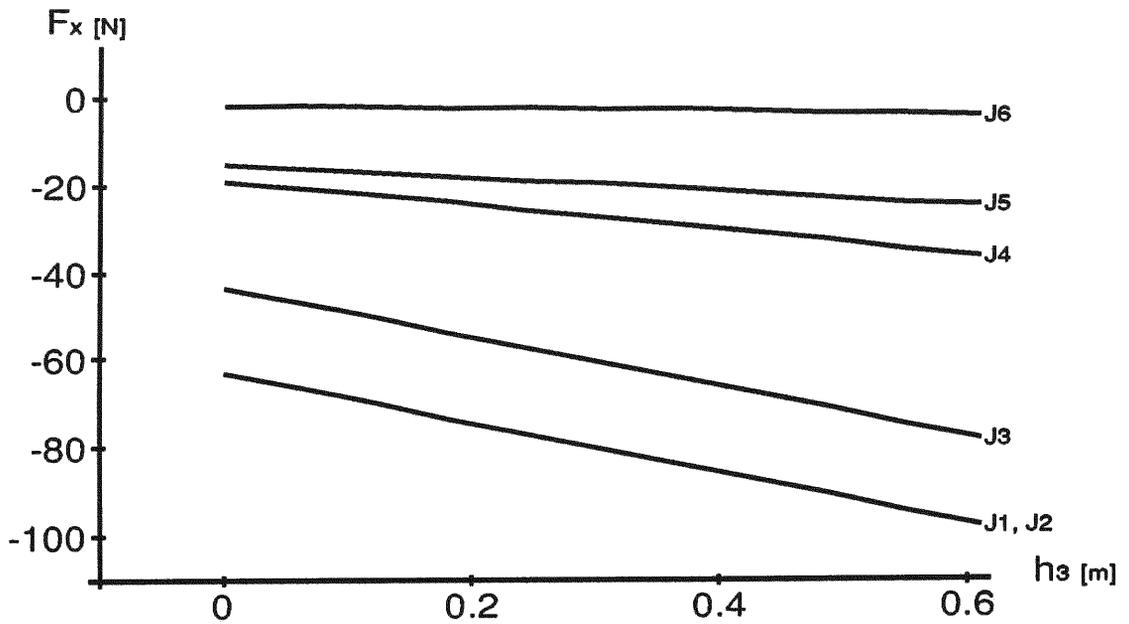


Figure 3.6a Bendix AA/CNC robot, joint force x-component versus elbow joint h3 position

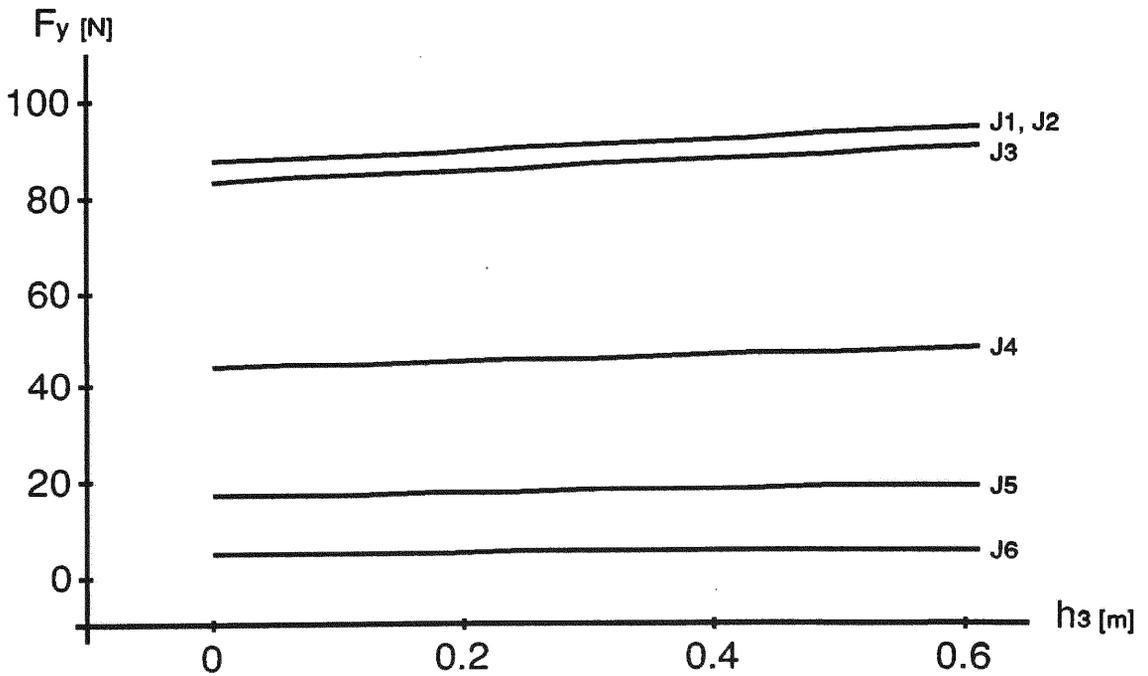


Figure 3.6b Bendix AA/CNC robot, joint force y-component versus elbow joint h3 position

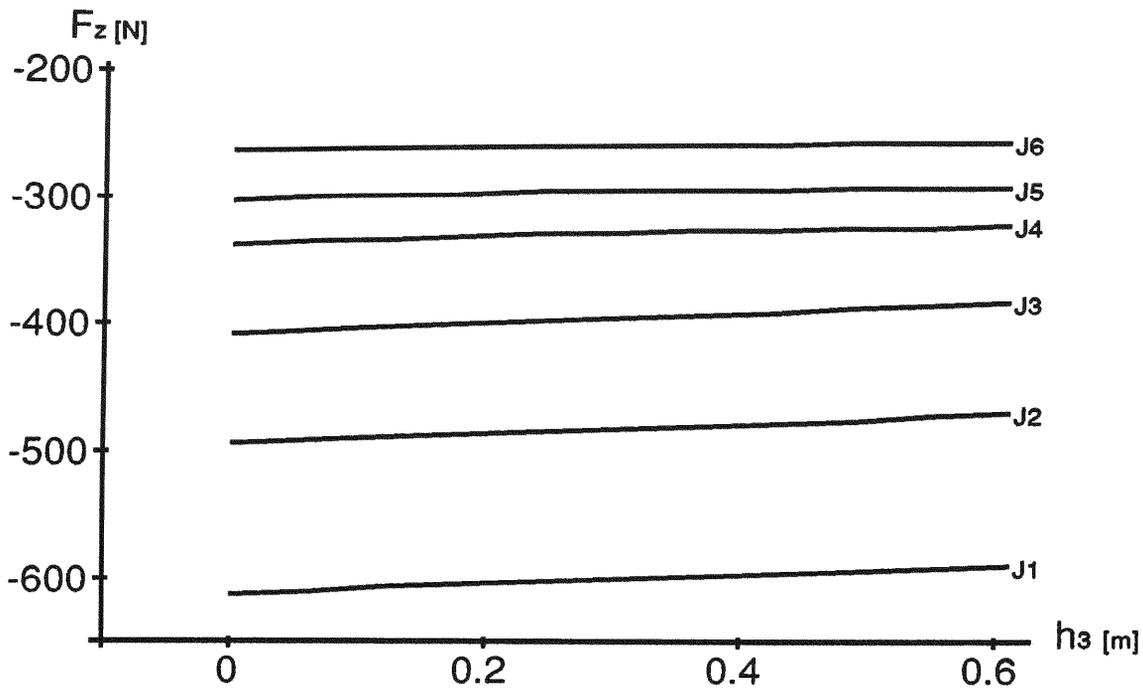


Figure 3.6c Bendix AA/CNC robot, joint force z-component versus elbow joint h_3 angle

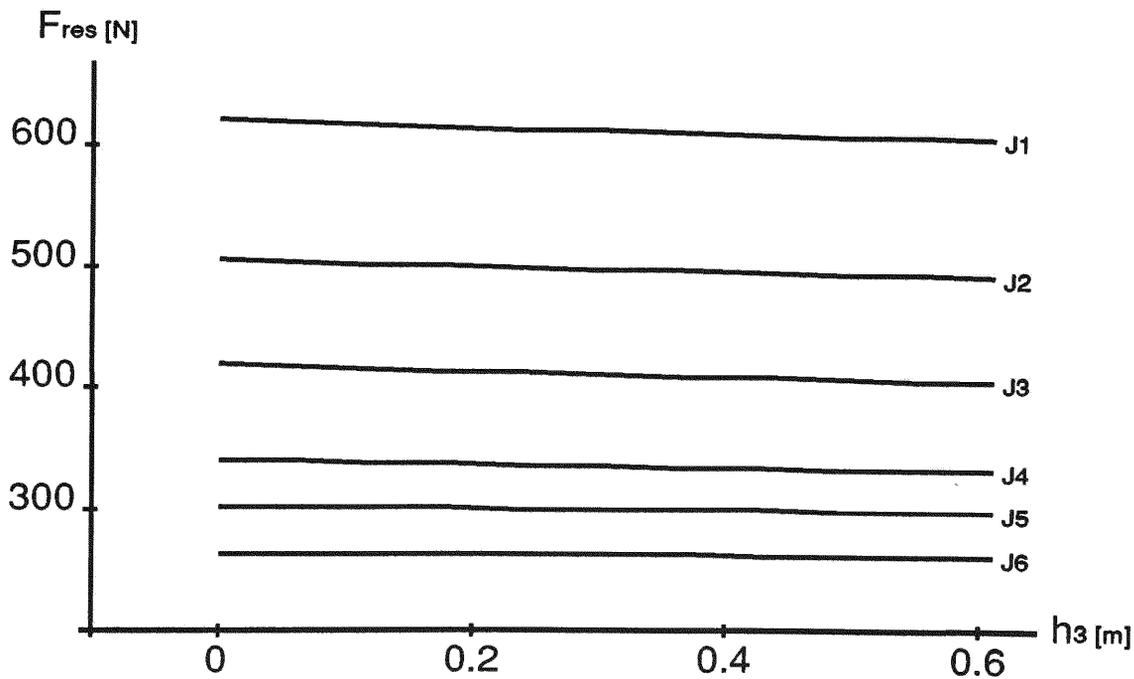


Figure 3.6d Bendix AA/CNC robot, joint resultant force (F_{res}) versus elbow joint h_3 position

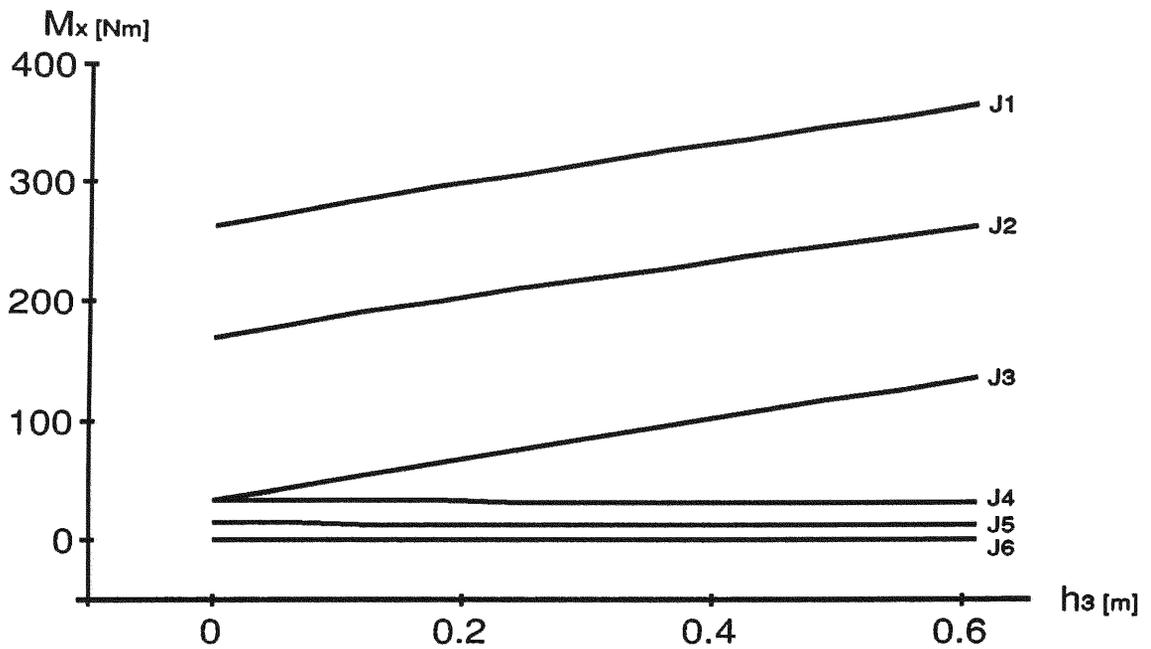


Figure 3.7a Bendix AA/CNC robot, joint moment x-component versus elbow joint h3 angle

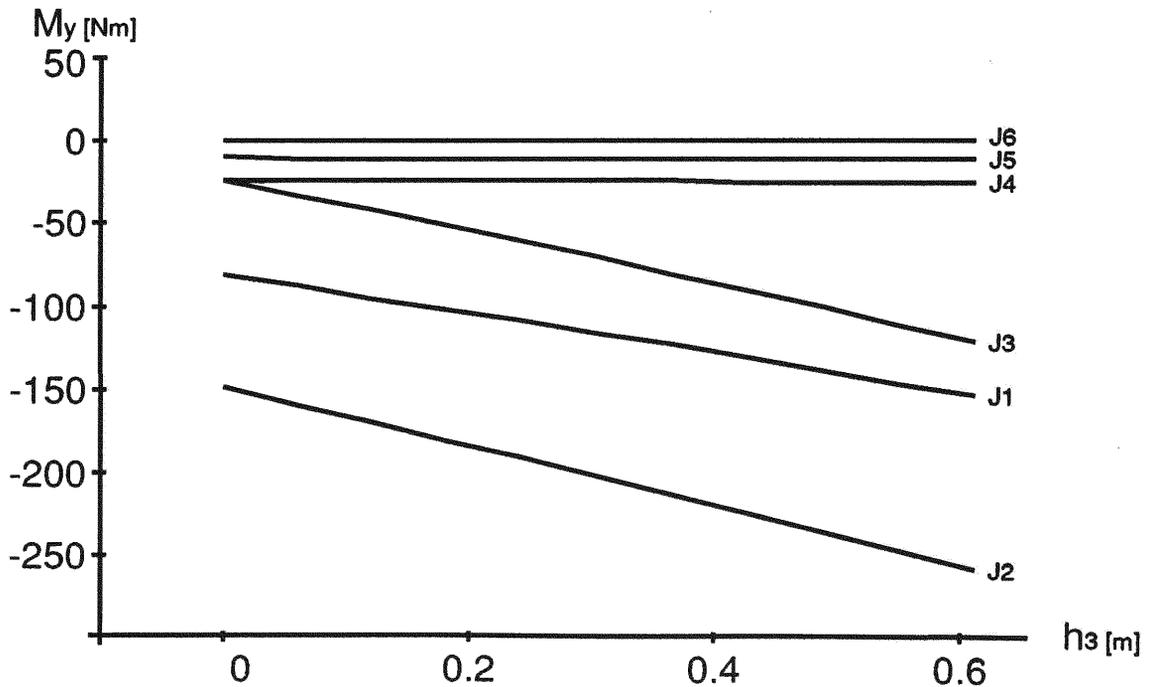


Figure 3.7b Bendix AA/CNC robot, joint moment y-component versus elbow joint h3 position

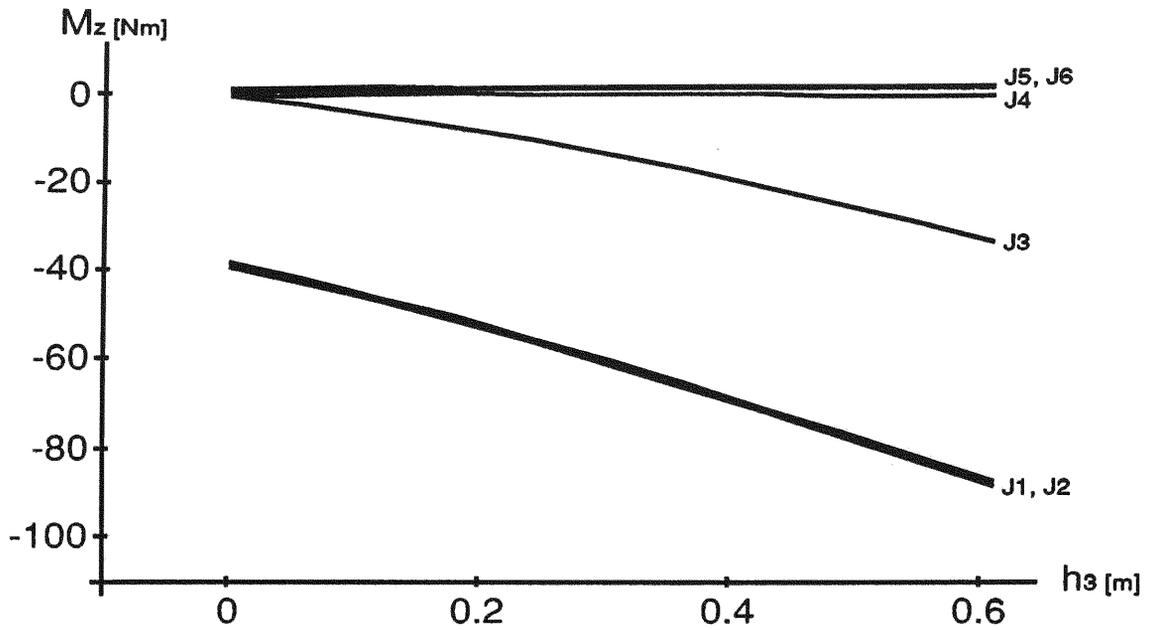


Figure 3.7d Bendix AA/CNC robot, joint moment z-component versus elbow joint h3 position

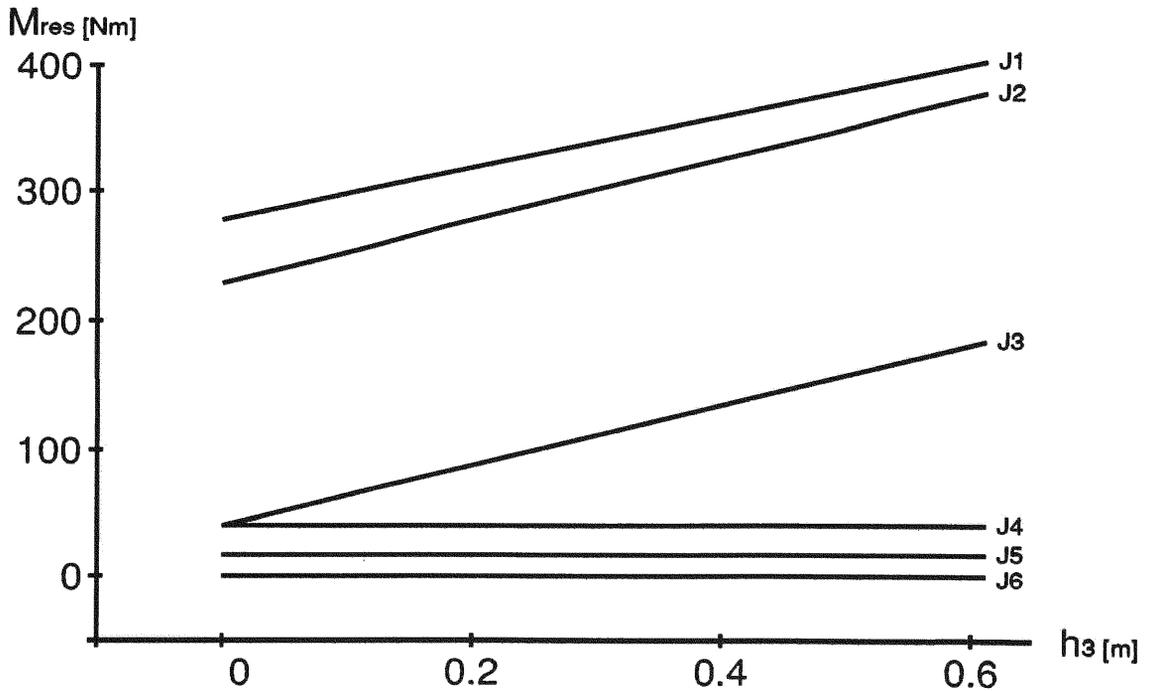


Figure 3.7d Bendix AA/CNC robot, joint resultant moment (M_{res}) versus elbow joint h3 position

Example 3.3 - Unimate 2000 Spherical Robot (SP/RRR)

The Unimate 2000 geometry involves a spherical joint at the base of the robot. The kinematic and dynamic data for the Unimate 2000 is given in Tables A.3a and A.3b in Appendix A. In the previous examples, the analysis was carried out allowing the motion of either a revolute or a prismatic joint. In this example, the robot manipulates a 25 Kg payload, and spherical joint 1 (base joint) is selected as the moveable joint. Comparing the shapes of the dynamic response curves (Figures 3.8a - 3.9d) to those in example 3.1 for Cincinnati Milacron T3 robot, the response of a spherical joint is basically similar to that obtained from a revolute joint, since both types of joints have similar types of motion (rotational motion).

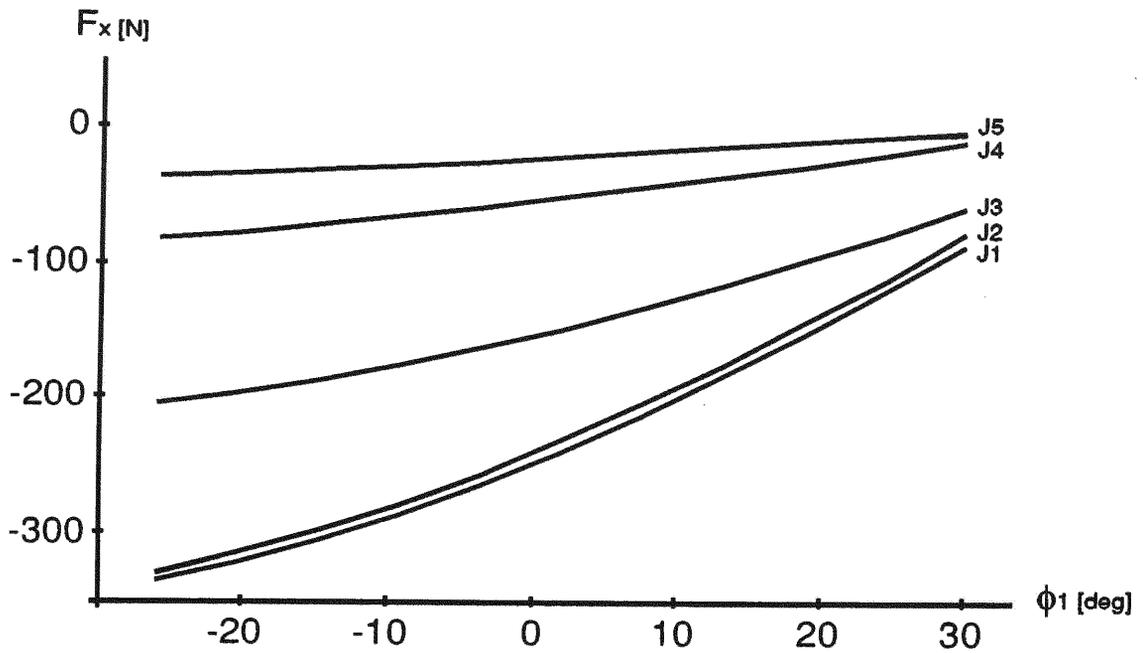


Figure 3.8a Unimate 2000 spherical robot, joint force x-component versus base joint ϕ_1 angle

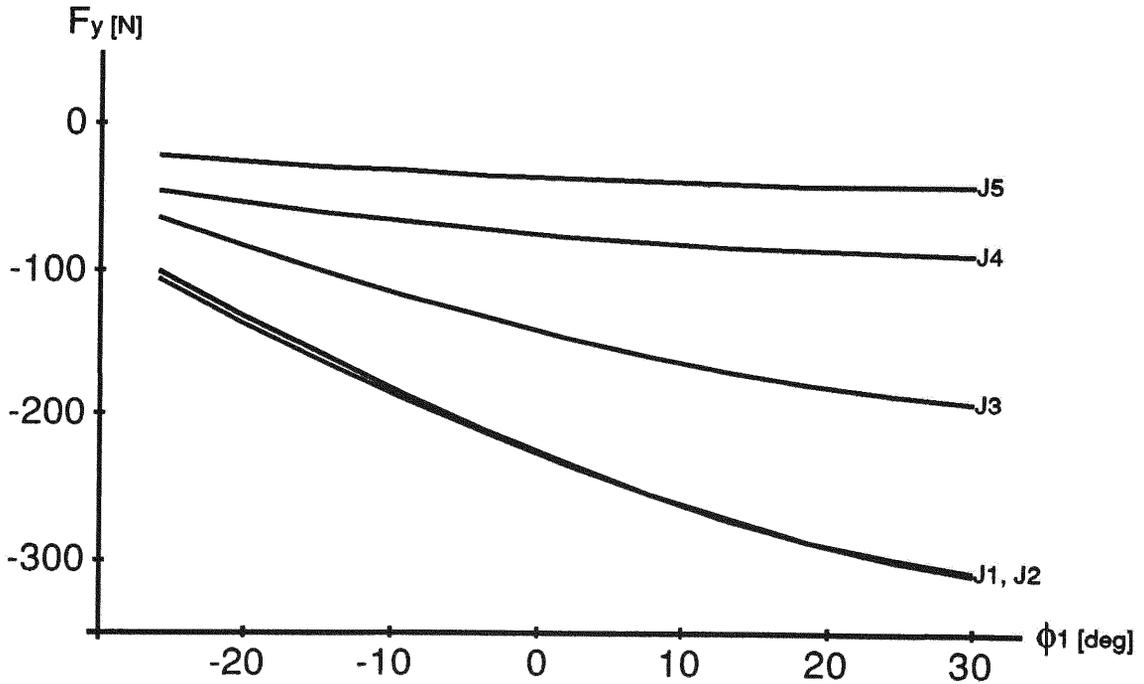


Figure 3.8b Unimate 2000 spherical robot, joint force y-component versus base joint ϕ_1 angle

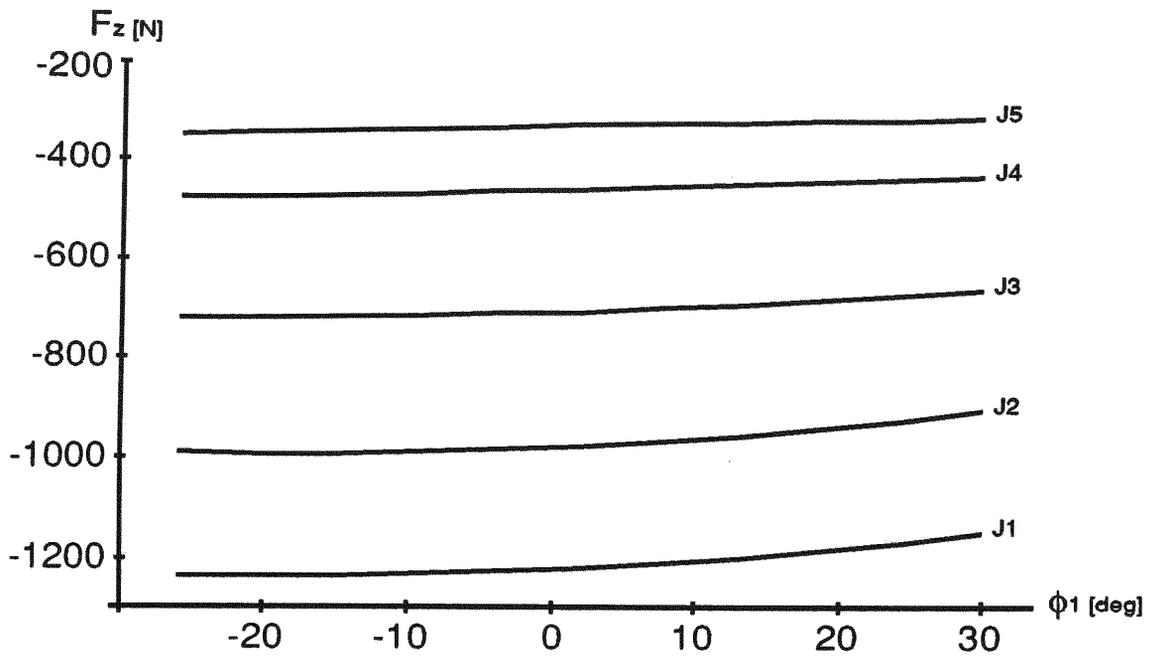


Figure 3.8c Unimate 2000 spherical robot, joint force z-component versus base joint ϕ_1 angle

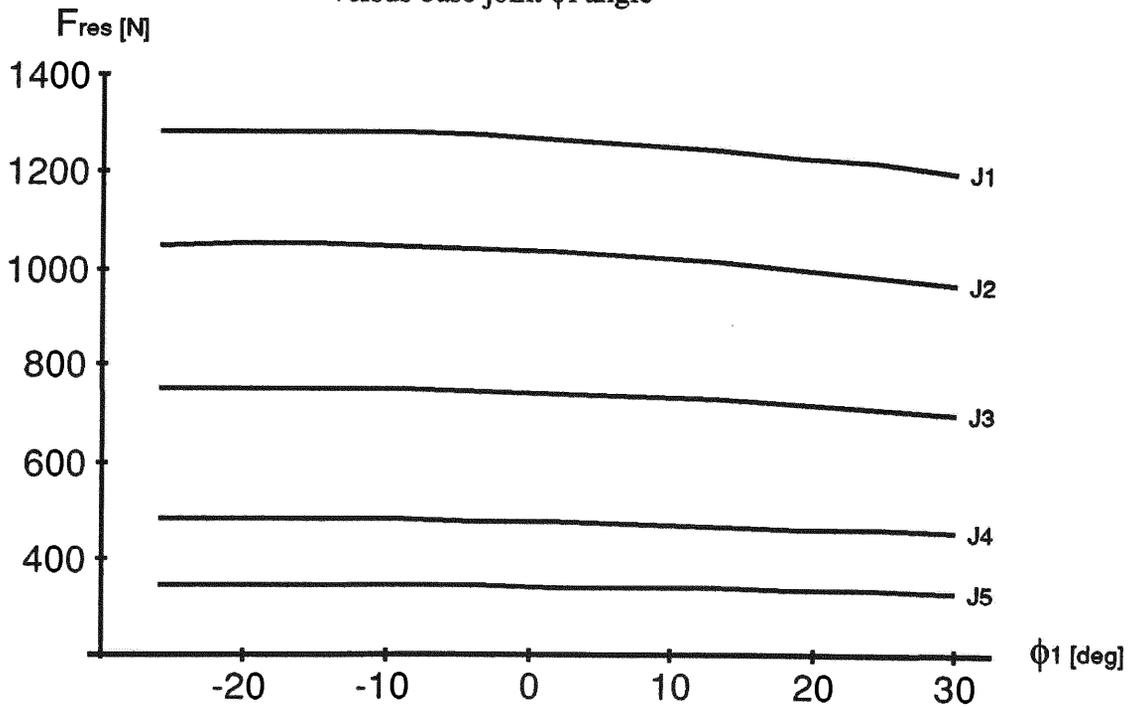


Figure 3.8d Unimate 2000 spherical robot, joint resultant force (F_{res}) versus base joint ϕ_1 angle

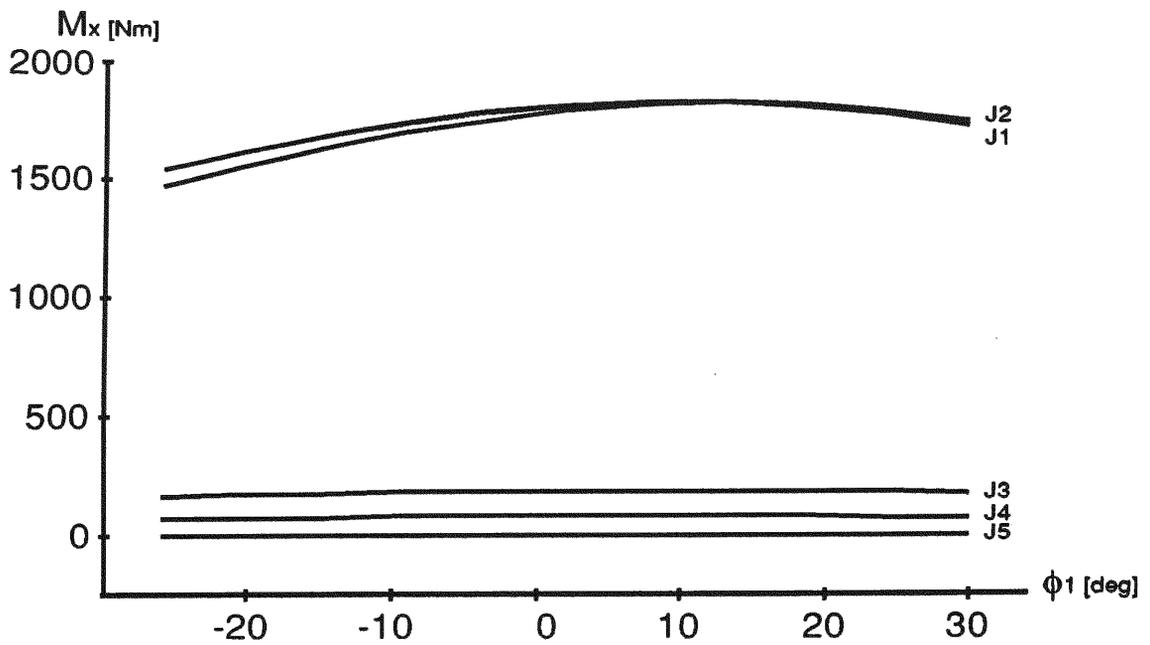


Figure 3.9a Unimate 2000 spherical robot, joint moment x-component versus base joint ϕ_1 angle

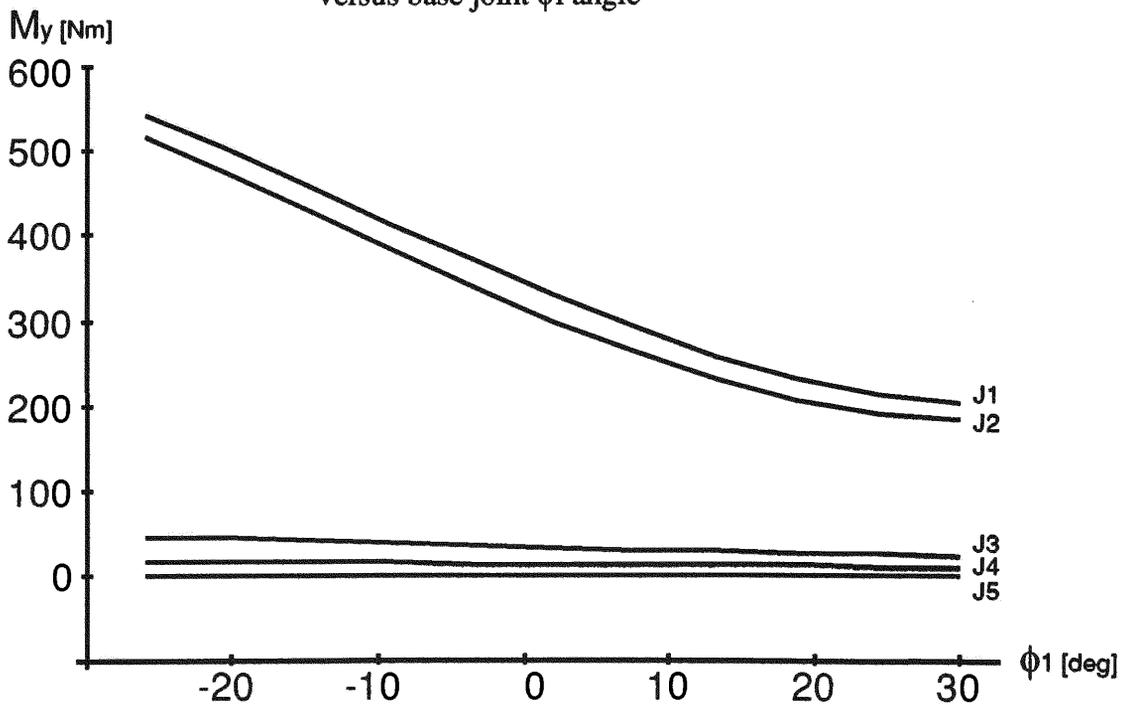


Figure 3.9b Unimate 2000 spherical robot, joint moment y-component versus base joint ϕ_1 angle

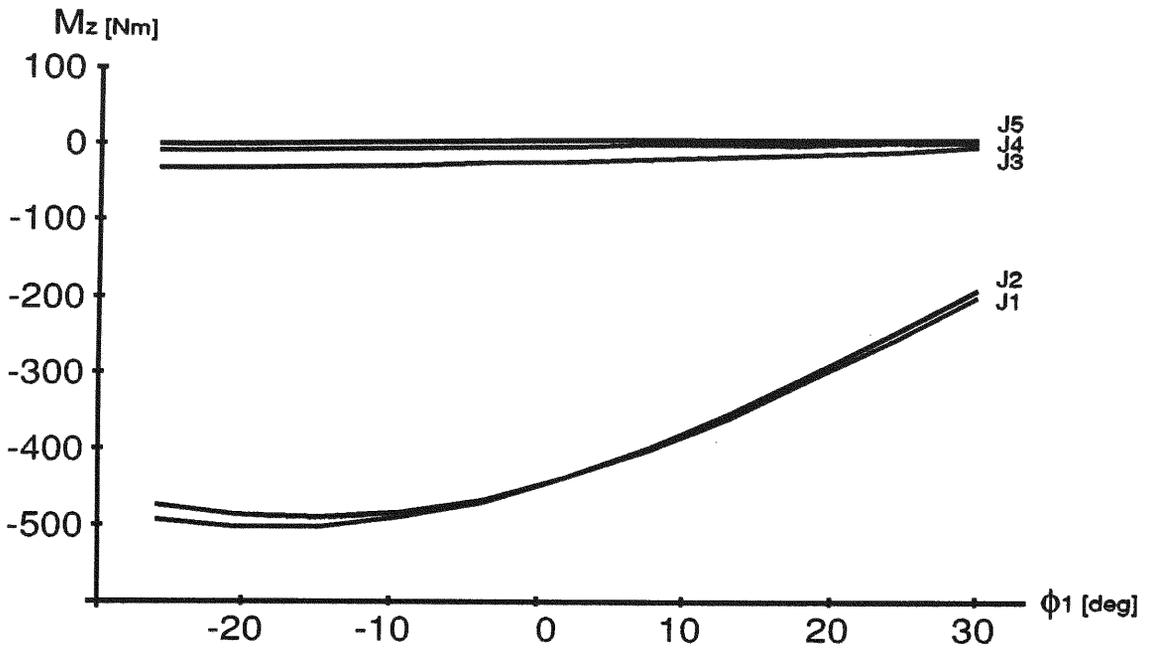


Figure 3.9c Unimate 2000 spherical robot, joint moment z-component versus base joint ϕ_1 angle

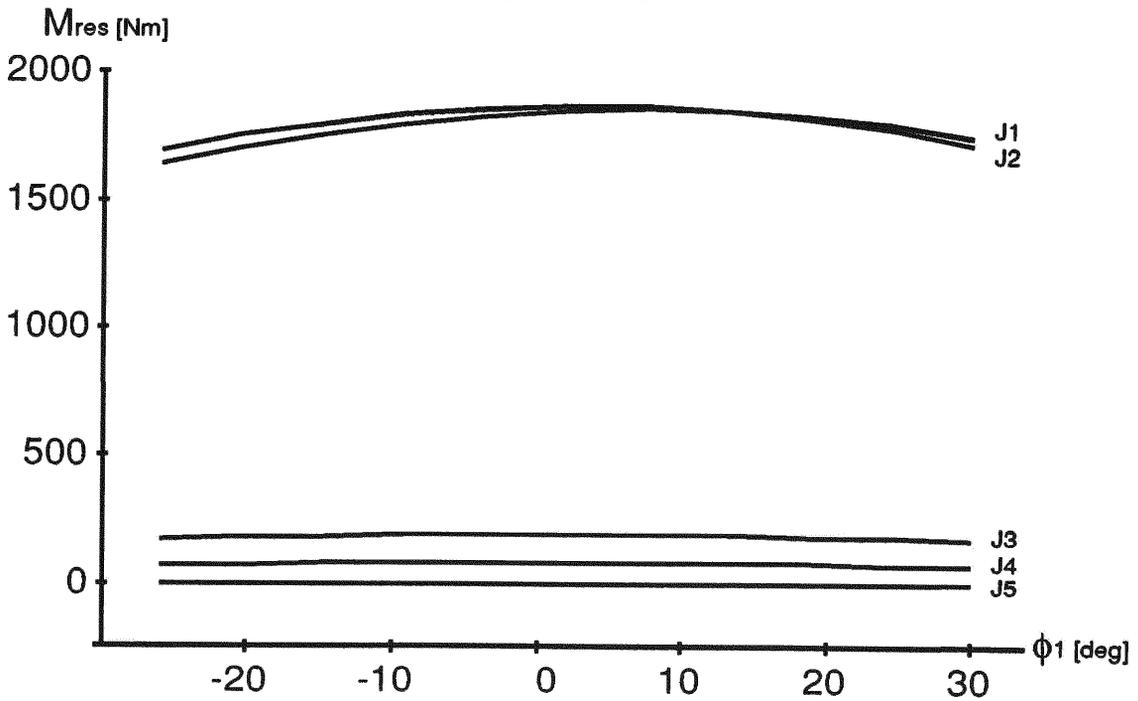


Figure 3.9d Unimate 2000 spherical robot, joint resultant moment (M_{res}) versus base joint ϕ_1 angle

Example 3.4 - IBM 7576 SCARA Robot (RRPR)

The SCARA configuration has four vertical joint axes. The kinematic and dynamic data for this robot are given in Tables A.4a and A.4b in appendix A. This configuration consists of three revolute joints and one prismatic joint. Because of the orientation of the joint axes of motion, some peculiar results can be expected from the dynamic analysis. The following figures display the joint forces and moments resulted from the motion of revolute shoulder joint θ_2 (i.e. curve 1 in Figures 2.26 a, b, and c). Due to the SCARA configuration, the x and y components of the joint forces are due to inertia effects only, and the z component results mostly from static forces (weight) and in a minimal part by the acceleration of prismatic joint 3. Also, the x and y components of the joint moment are mostly generated by the weight, and the z component of joint moment is due mainly to the inertia x and y components of force.

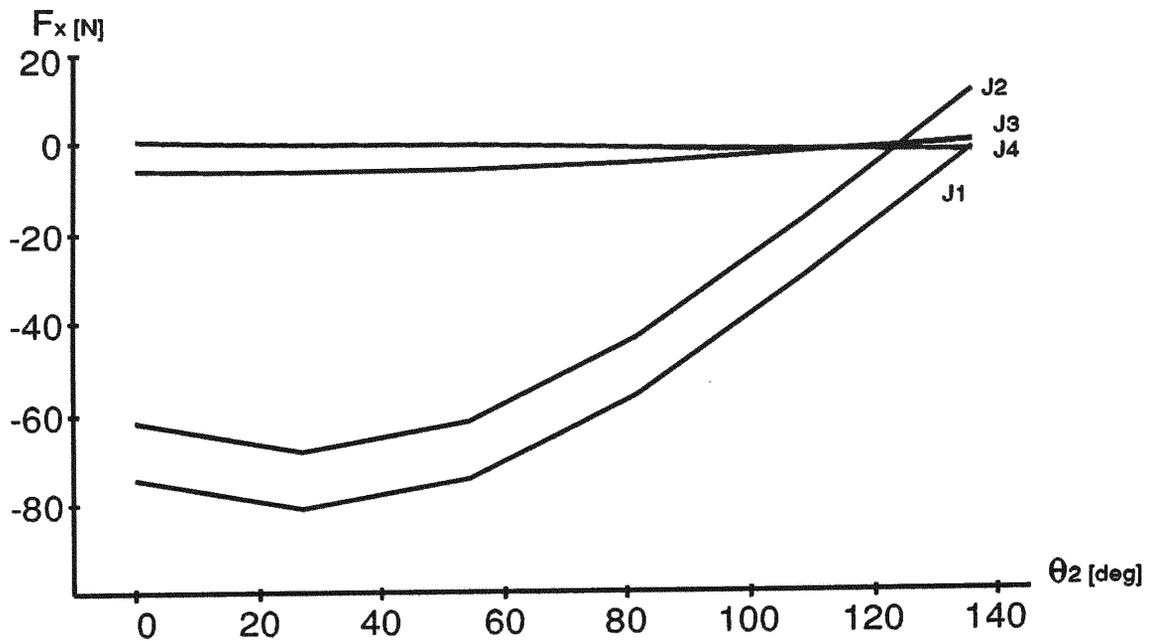


Figure 3.10a IBM 7576 SCARA robot, joint force x-component versus shoulder joint θ_2 angle

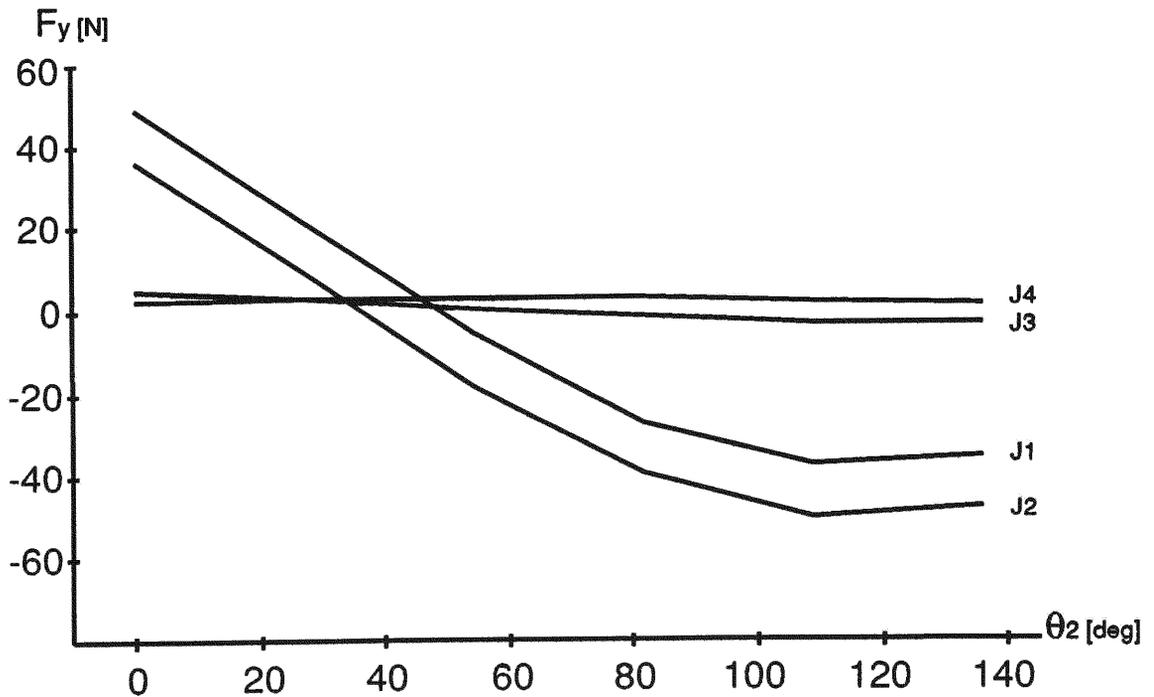


Figure 3.10b IBM 7576 SCARA robot, joint force y-component versus shoulder joint θ_2 angle

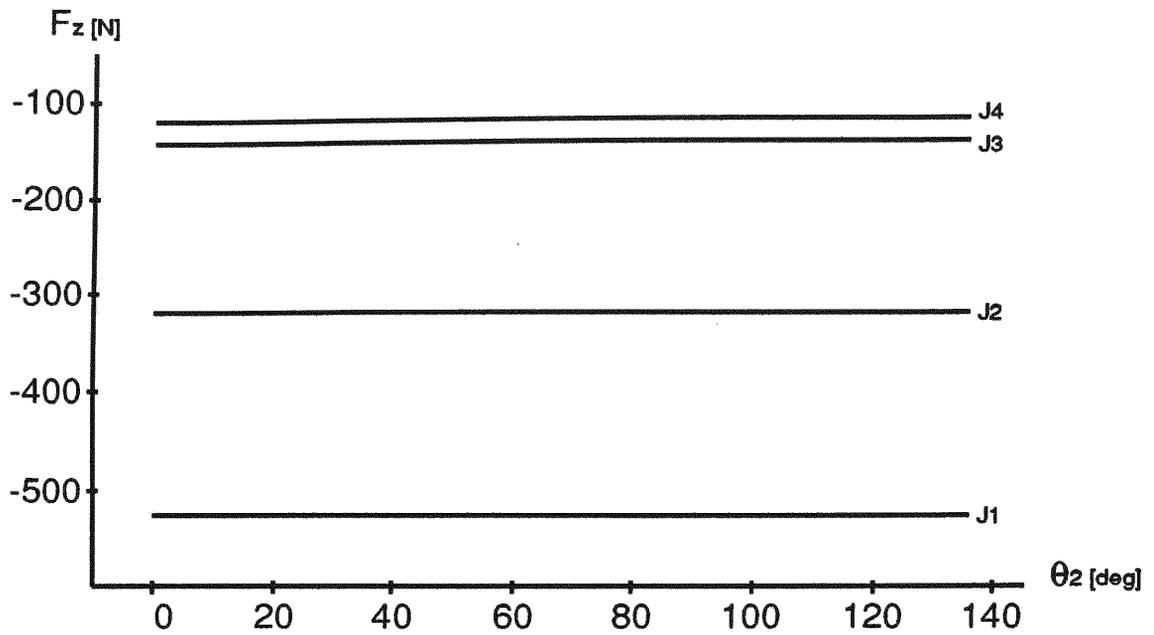


Figure 3.10c IBM 7576 SCARA robot, joint force z-component versus shoulder joint θ_2 angle

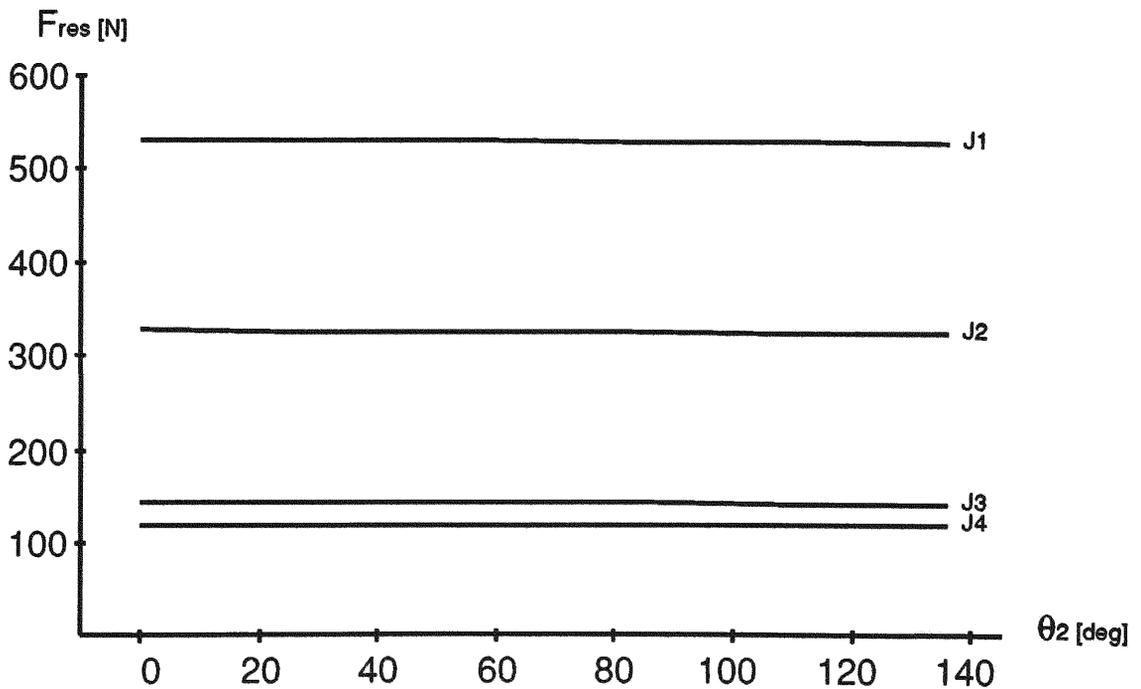


Figure 3.10d IBM 7576 SCARA robot, joint resultant force (F_{res}) versus shoulder joint θ_2 angle

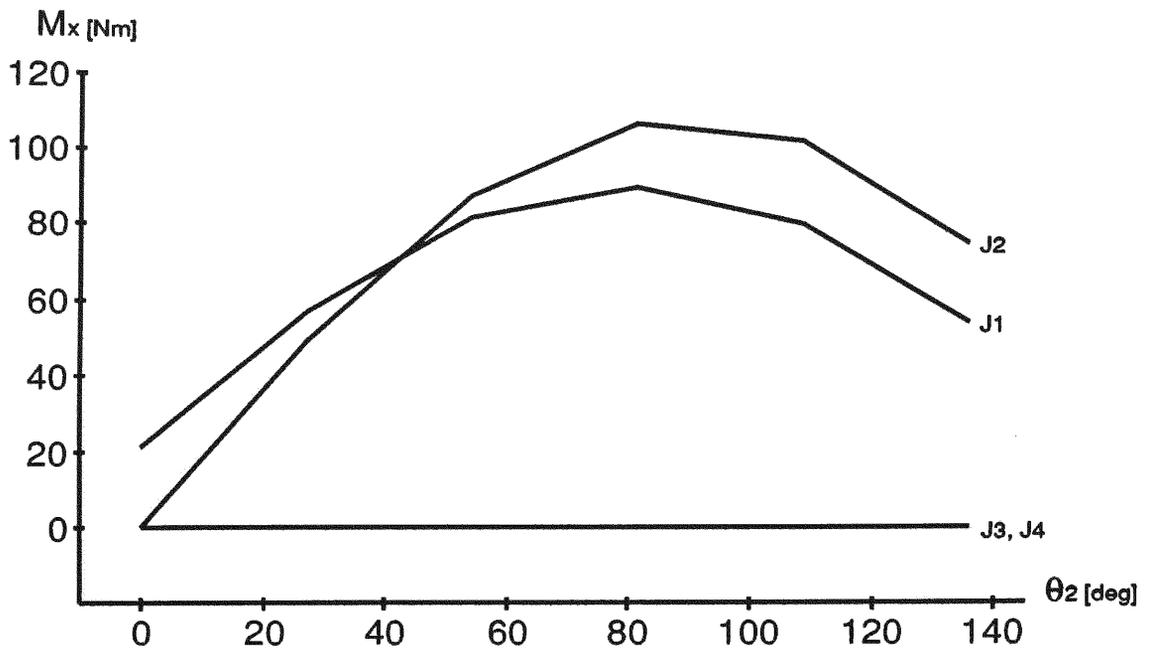


Figure 3.11a IBM 7576 SCARA robot, joint moment x-component versus shoulder joint θ_2 angle

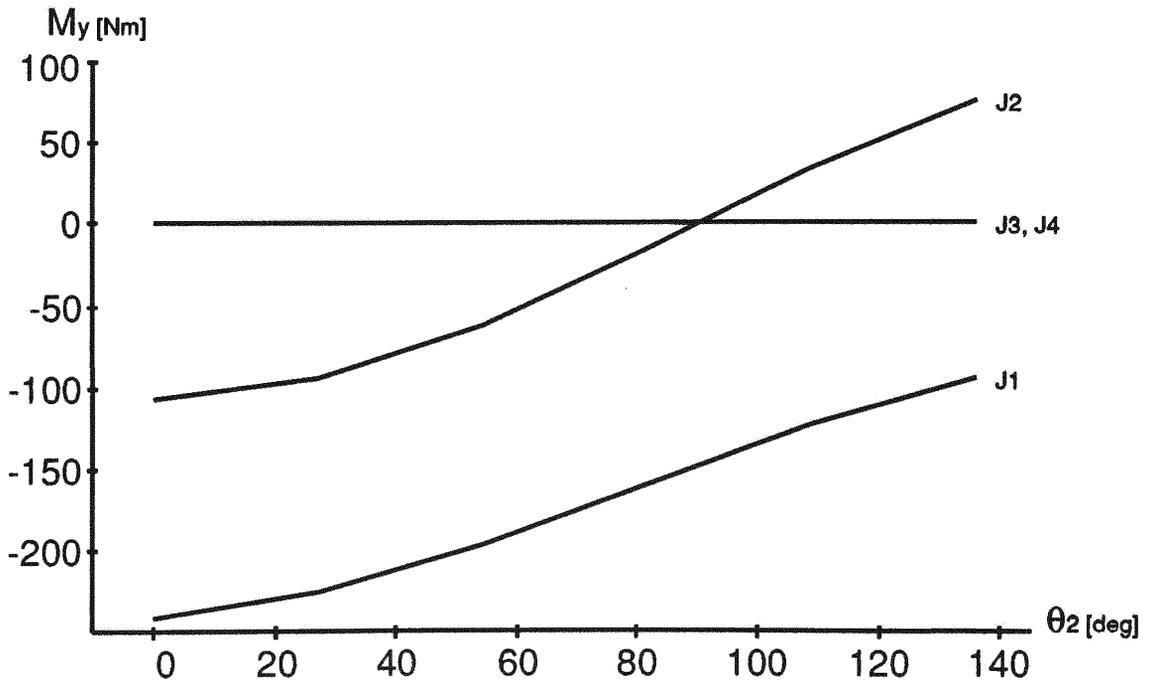


Figure 3.11b IBM 7576 SCARA robot, joint moment y-component versus shoulder joint θ_2 angle

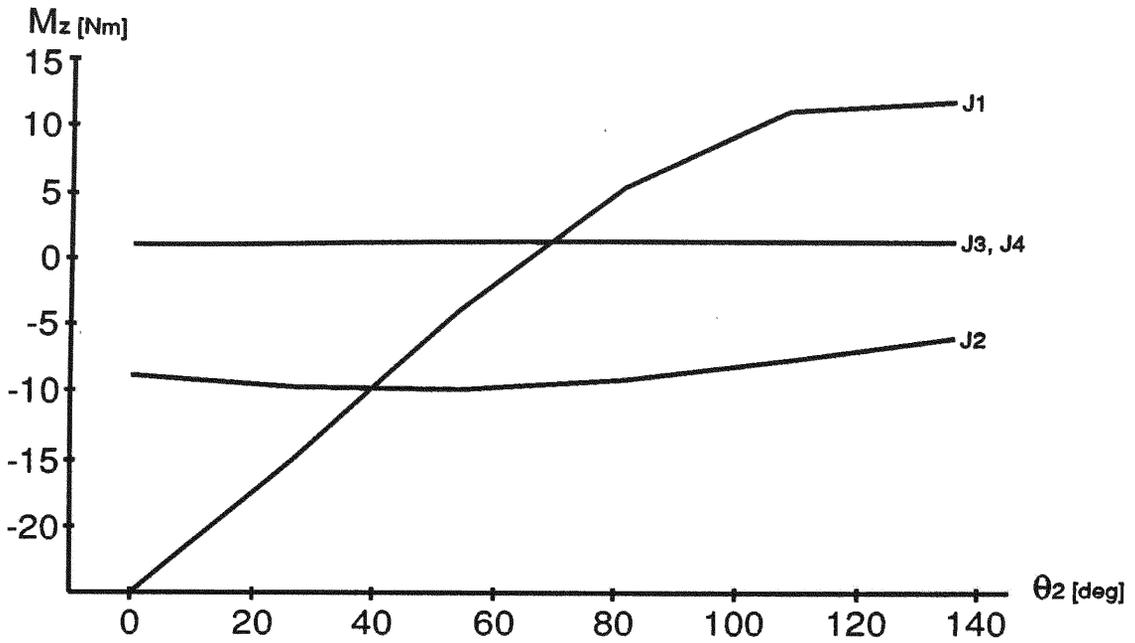


Figure 3.11c IBM 7576 SCARA robot, joint moment z-component versus shoulder joint θ_2 angle

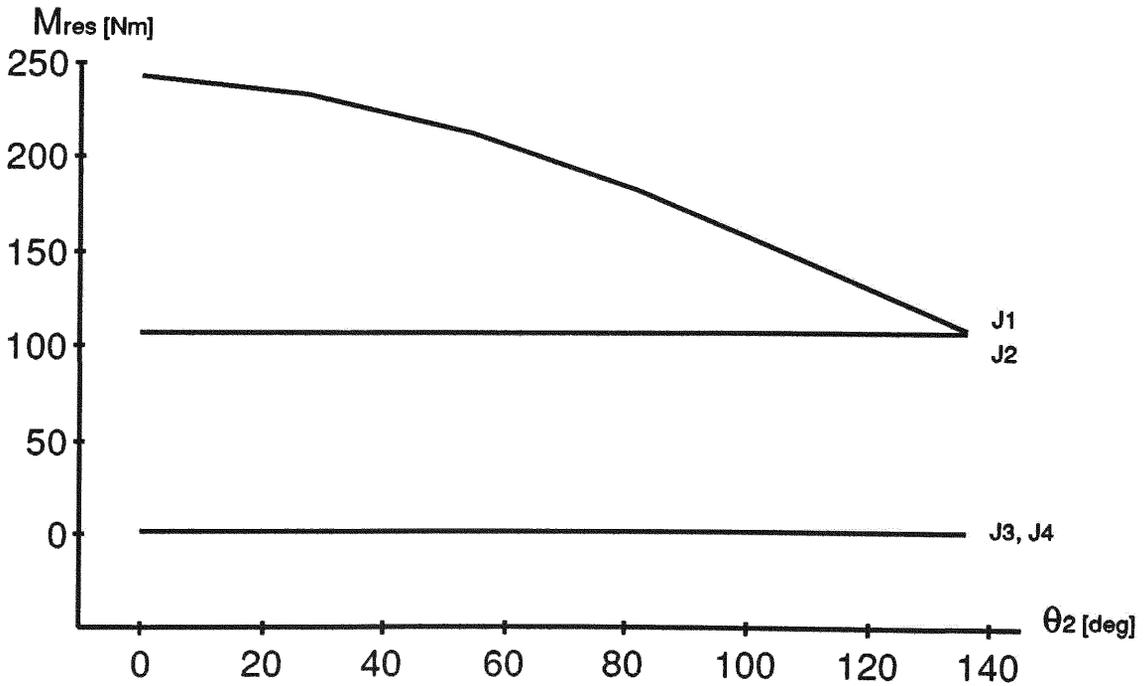


Figure 3.11d IBM 7576 SCARA robot, joint resultant moment (M_{res}) versus shoulder joint θ_2 angle

Example 3.5 - Space Shuttle RMS Manipulator (RRR/RRR)

All kinematic and dynamic data for the RMS manipulator are included in Tables A.5a and A.5b in Appendix A. In this example, the acceleration of gravity was set to zero, since this manipulator was originally designed to work in outer space. As a result of this, only inertia joint forces and moments result from the motion. Because of the large dimensions of the links of this robot, even small joint angular velocities and accelerations will cause large velocities and accelerations, and therefore large inertia forces and moments develop. In the analysis, revolute shoulder joint θ_2 is allowed to move throughout its entire range of motion. The manipulator carries a 100 Kg mass at its end effector. This manipulator is basically similar to the Cincinnati Milacron T-3 robot, except for its weight and link dimensions, and its working environment conditions. The figures shown display curves of similar shapes as the curves obtained from the Cincinnati Milacron T3 robot.

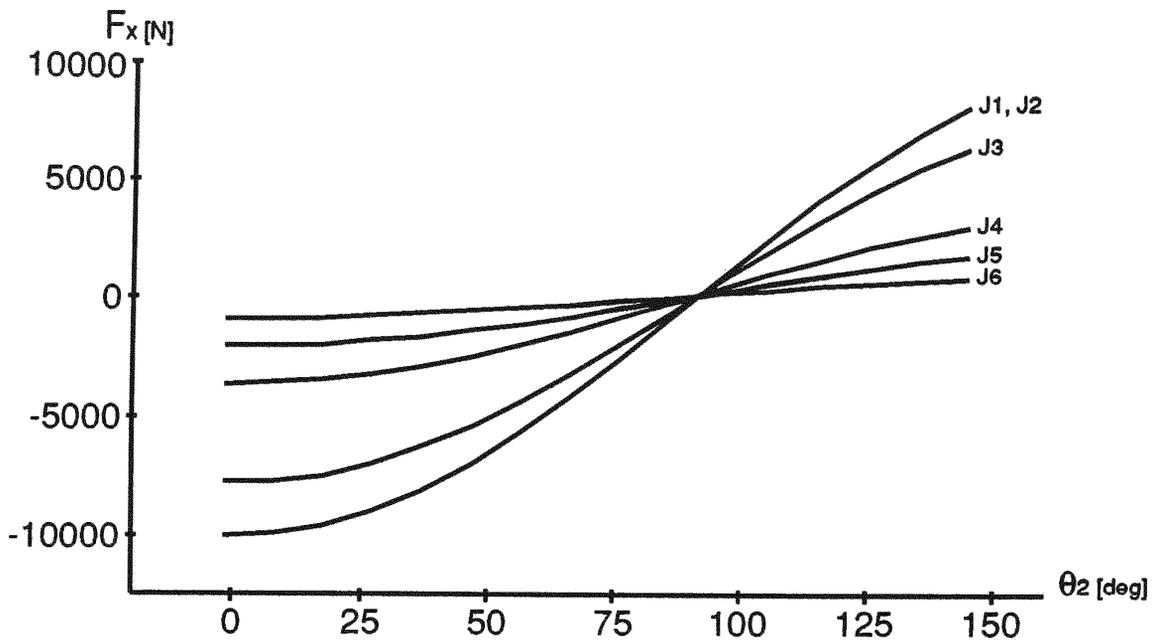


Figure 3.12a Space Shuttle RMS robot, joint force x-component versus shoulder joint θ_2 angle

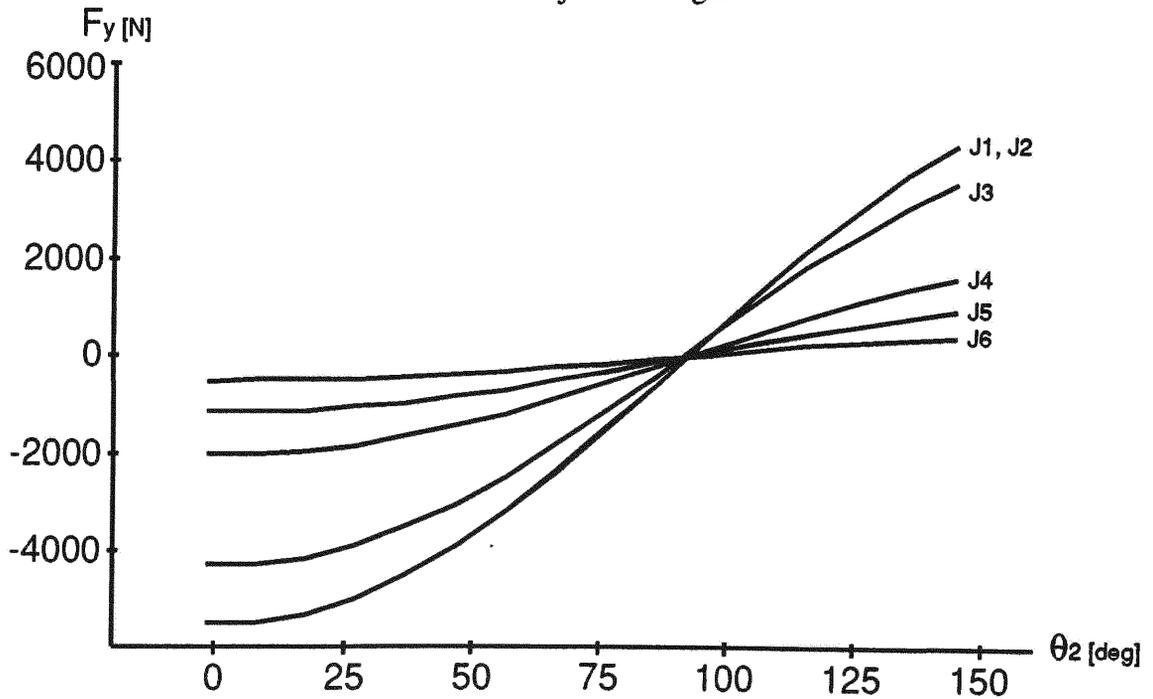


Figure 3.12b Space Shuttle RMS robot, joint force y-component versus shoulder joint θ_2 angle

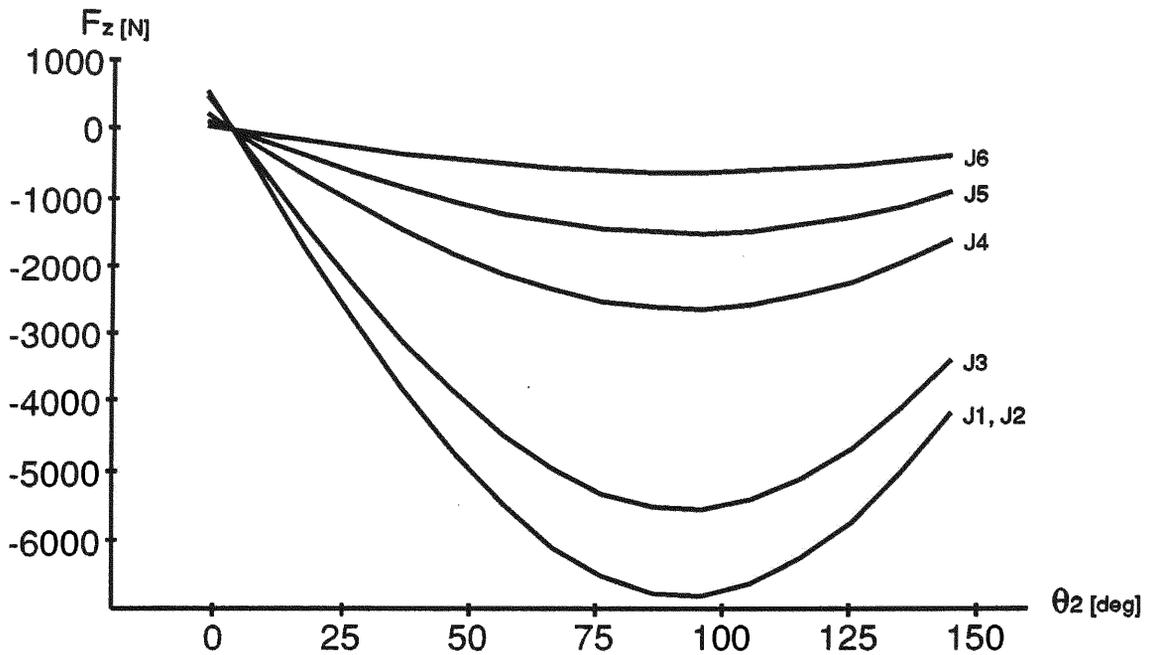


Figure 3.12c Space Shuttle RMS robot, joint force z-component versus shoulder joint θ_2 angle

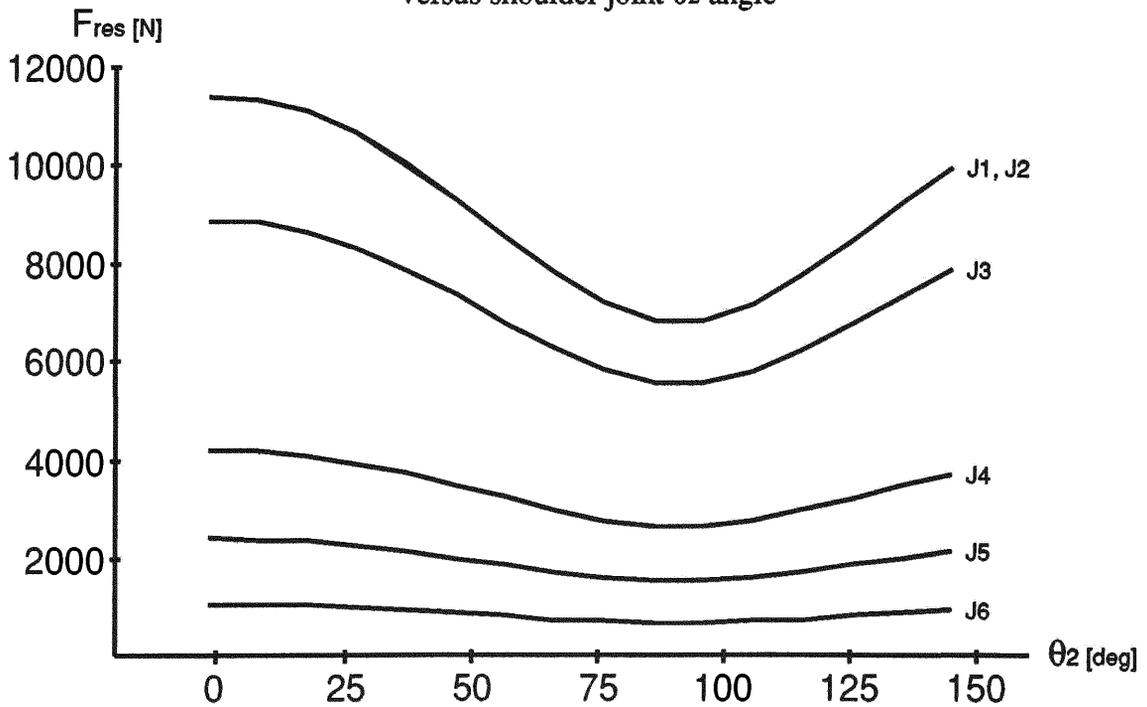


Figure 3.12d Space Shuttle RMS robot, joint resultant force (F_{res}) versus shoulder joint θ_2 angle

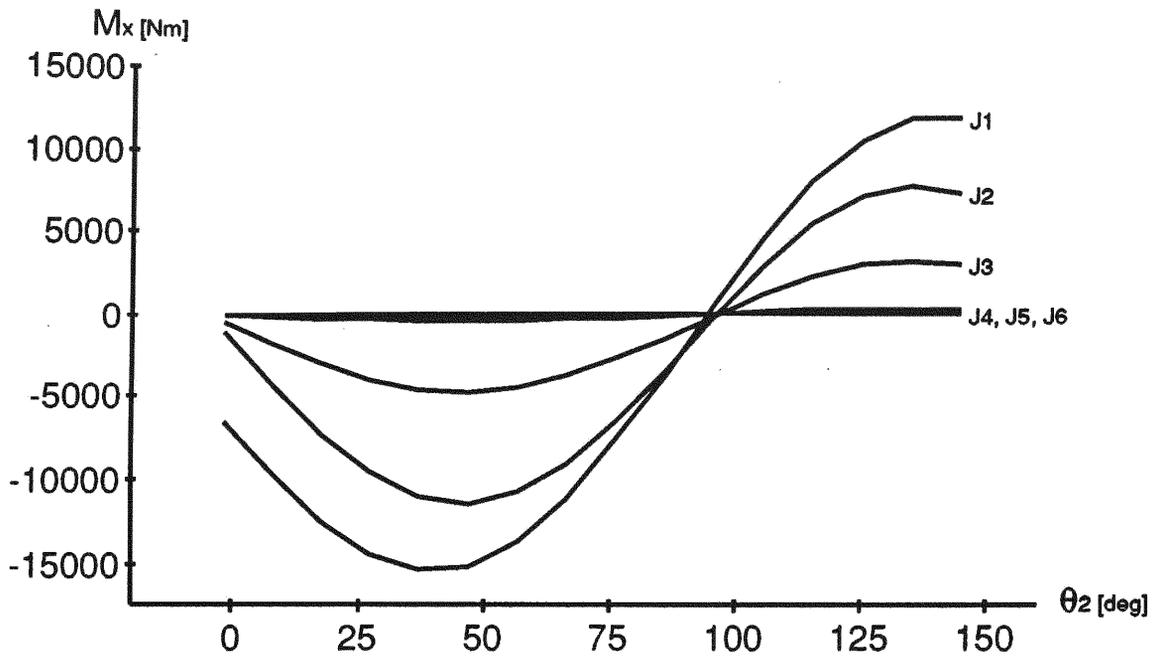


Figure 3.13a Space Shuttle RMS robot, joint moment x-component versus shoulder joint θ_2 angle

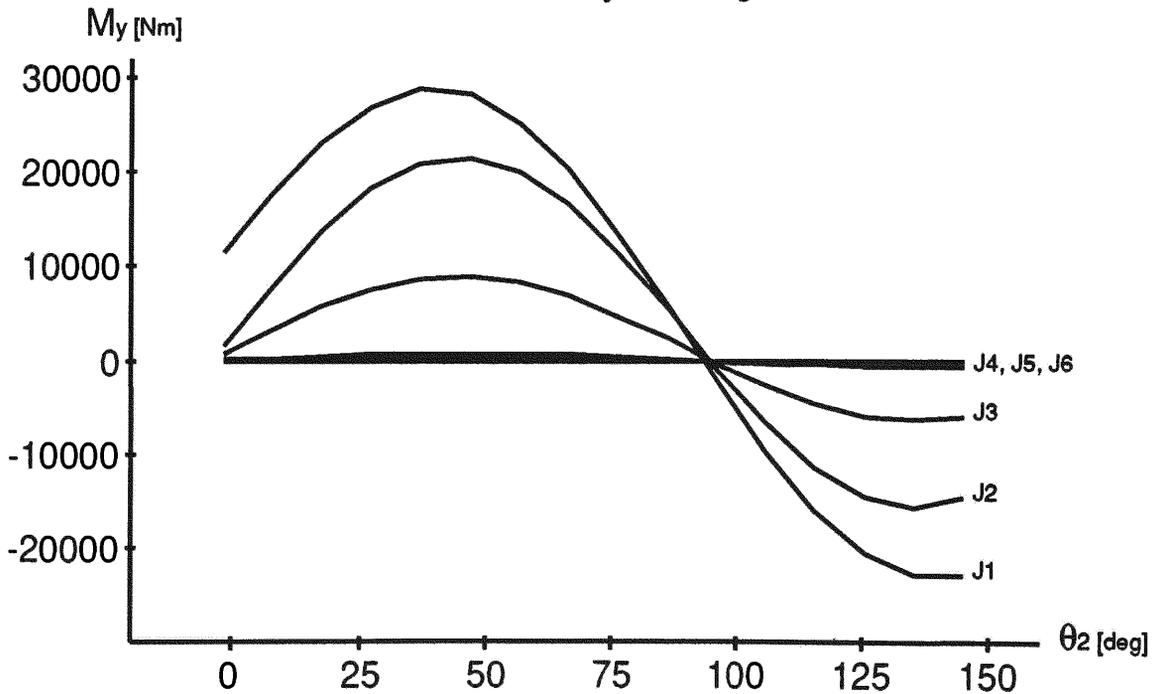


Figure 3.13b Space Shuttle RMS robot, joint moment y-component versus shoulder joint θ_2 angle

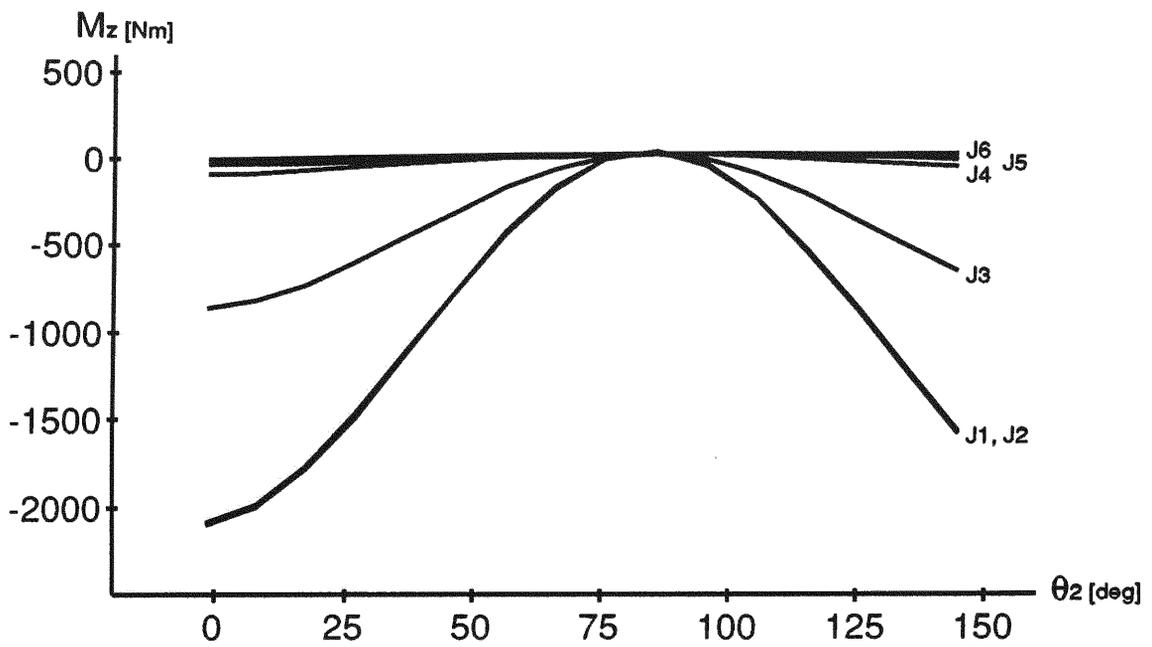


Figure 3.13c Space Shuttle RMS robot, joint moment z-component versus shoulder joint θ_2 angle

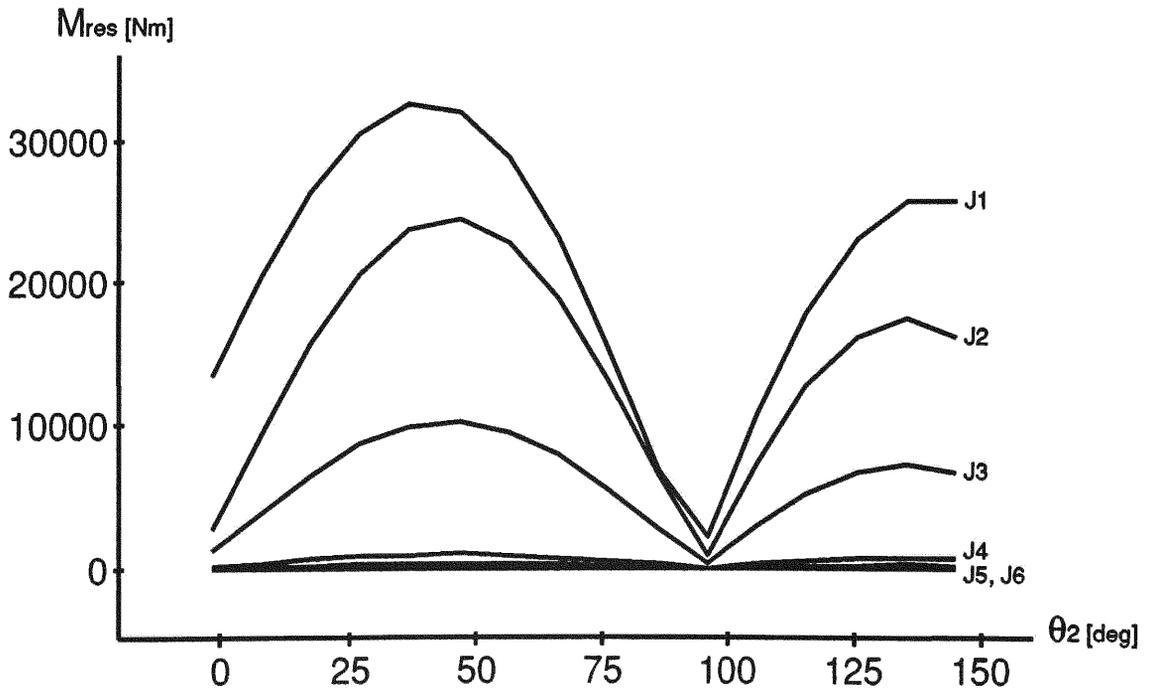


Figure 3.13d Space Shuttle RMS robot, joint resultant moment (M_{res}) versus shoulder joint θ_2 angle

3.3.4 Discussion of Results

The results presented for the Cincinnati Milacron T3, Bendix AA/CNC, and Unimate 2000 were compared to those obtained by Yih (1987), and by Chang and Shabana (1990). Also, all dynamic results obtained for all five robots were analyzed and verified by hand calculations assuming static (zero joint velocities and accelerations) and dynamic equilibrium conditions.

Dynamic joint forces and moments were determined under a condition of quasi-static equilibrium. These joint forces and moments are the required input forces and moments necessary to drive the manipulator's links given the prescribed kinematic conditions (joint positions, velocities, and accelerations). Robotic links were assumed to be rigid. The following conclusions are drawn based on these considerations.

In general, the magnitude of the joint forces and moments appear to be larger towards the base joint of the robot. This is expected, as the forces and moments add up towards the base joint, which is responsible for supporting the whole linkage system. Therefore, an obvious conclusion is that; larger joint actuators are required near the base joint.

The joint force and moment curves display special features depending on the type of joint which generates the motion. For the case of a revolute joint, the joint driving forces and moments vary non-linearly against the motion of the joint. This means that the joint driving forces and moments must be controlled in a similar fashion (non-linearly) in order to produce such motion. The motion of a prismatic joint is generated by an almost linear variation of joint force and moment according to Figs. 3.6a - 3.7d. Also this type of motion results in a nearly linear force and moment response on the non-moving joints. For the case of a spherical joint, the joint forces and moments response is similar (non-linear) to that caused by the motion of a revolute joint. This may be explained by the fact

that both type joints have a somewhat similar type of motion (revolving). The difference being that one moves about one axis, while the other (spherical) rotates about three axes.

The IBM SCARA and the RMS manipulators present special dynamic joint force and moment results. Due to the particular configuration of the SCARA robot (all joint axes aligned vertically to the ground) the X and Y components of the joint forces result from inertial effects alone. The Z component of the joint force is mostly due to static forces (weight), and in a small part, it results from the motion (acceleration) of prismatic joint 3, therefore, it is almost constant as displayed in Fig 3.10. The X and Y components of the joint moments for the IBM SCARA are mainly due to weight forces, and the Z component results from the X and Y components of force (inertial). The RMS manipulator was designed to work in a zero gravity environment, thus, the actuator forces and moments required to drive the robot must work against the inertia of the robot and payload mass. These are displayed in Figs. 3.12a - 3.13d.

CHAPTER 4

RaMIP (Robot and Mechanism Integrated Program)

4.1 Introduction

The application of C-B notation for the kinematic and dynamic analyses of open-chain mechanisms has led to the development of a computer algorithm named RaMIP (Robot and Mechanism Integrated Program). RaMIP is a general purpose computer program written in C language and implemented to run on a SUN platform. It integrates advanced 2- and 3-dimensional graphical capabilities for the presentation and analysis of results. It permits modeling of robotic manipulators with any combination of revolute, prismatic, and/or spherical joints (up to 30 d.o.f.). Currently, all procedures are organized through the usage of selection menus and keyboard input. In the future, RaMIP will include menu selection using mouse. Moreover, RaMIP is will be further expanded to include kinematic and dynamic analyses of closed-chain mechanisms, Lagrangian dynamics, and inverse kinematics as well.

Utilizing the multi-tasking feature of UNIX, RaMIP exchanges information with *Mathematica* for the generation of graphic displays. The multi-tasking feature allows more than one process (running program) to be processed at the same time. Utilizing this feature, RaMIP makes calls to *Mathematica* from within itself, giving *Mathematica* the necessary instructions (*Mathematica* graphics language files) for the generation of graphics. In this manner all the 2- and 3-dimensional graphics capabilities of *Mathematica* are utilized, without going into major programming complications (though substantial programming effort is involved during the process).

4.2 General Computational Procedures

The program is organized to allow the user to move from one step to another by the use of menus. The menus provide complete mobility for the user to move either forward or backward in the solution procedure. Figures 4.1 through 4.6 present the flow charts which reveal the program's organization and analysis procedures. The first step in the analysis involves the input of all kinematic and/or dynamic data pertaining the manipulator under study. The second step contains the kinematic and/or dynamic solution through an interactive procedure in which the desired solution is obtained from a series of solution procedures. The third step consists of displaying, saving, and printing the results based on the analytical data.

4.2.1 Input Data

The first step in the analysis involves the input of all necessary data pertaining the open-chain mechanism to be analyzed. For this reason, it is suggested that all input data needed for the analysis should be prepared in advance (see appendix A). The data may be stored and read from a data file, or it may be entered using the keyboard. All input data typed using the keyboard are written and stored automatically to a data file for later use.

A) Kinematic Input Data

The data necessary for the kinematic analysis include:

- a) number of joints,
- b) type of each joint (R, P, or S),
- c) joint and link parameters:

- Revolute - ($\alpha, \beta, h, r, \theta_{\min}, \theta_{\max}, \theta_{\text{interval}}, d\theta/dt, d^2\theta/dt^2$)
- Prismatic - ($\alpha, \beta, \theta, r, h_{\min}, h_{\max}, h_{\text{interval}}, dh/dt, d^2h/dt^2$)
- Spherical - ($\alpha, \beta, a, \phi_{\min}, \phi_{\max}, \phi_{\text{interval}}, \theta_{\min}, \theta_{\max}, \theta_{\text{interval}}, d\phi/dt, d^2\phi/dt^2, d\theta/dt, d^2\theta/dt^2$)

B) Dynamic Input Data

The data needed for the dynamic analysis include all kinematic data as explained previously, along with the following data (in order):

- a) Number of links of the manipulator
- b) Acceleration of gravity (MKS or Ft-lb-sec)
- c) Mass of each link
- d) Parameter μ for each link (to locate the C.G. of the link)
- e) Inertia tensor components for each link ($I_{xx}, I_{yy}, I_{zz}, I_{xy}, I_{xz}, I_{yz}$)
- f) Mass of Payload

Note: same as for the kinematic input, all data may be entered using the keyboard or by reading a data file. If the data is entered through the keyboard, it is automatically saved to a data file.

C) Data Structures and Subroutines

The data structures used by RaMIP include integer, single and double precision floating point variables and arrays, character variables and strings (character arrays), and

a set of fixed value parameters. Table 4.1 describes these parameters.

Table 4.1 Algorithm Data Structures

Parameter	Type	Description
A[6xn][6xn]	float (g)	Dynamics coefficient matrix A from $Ax = b$
accelvalue[curve#][point#]	float (g)	Holds shading values for 3-D acceleration graphics in floating point form
A_inv[6xn][6xn]	float (g)	A^{-1} matrix
analysistype	int (g)	Indicates type of analysis: displacement=0, velocity=1, acceleration=2
aval[curve#][point#]	int (g)	Holds shading values for 3-D acceleration shaded graphics in integer form
b[6xn]	float (g)	Dynamics column vector b from $Ax = b$, holds forces and moments about the each link C.G.
blue[level]	int (g)	Contains blue scale shading
brianmat[4][4]	float (g)	Bryant angles matrix
brimat[4][4]	float (l)	Bryant angles matrix
charma[joint#][4][4]	float (g)	Characteristic acceleration matrix
charmd[joint#][4][4]	float (g)	Characteristic displacement matrix
charm[v][joint#][4][4]	float (g)	Characteristic velocity matrix
chma[joint#][4][4]	float (l)	Characteristic acceleration matrix
chmd[joint#][4][4]	float (l)	Characteristic displacement matrix
chmv[joint#][4][4]	float (l)	Characteristic velocity matrix
cgaccel[link#][3]	float (g)	Holds x, y, z acceleration of each link C.G.
cgpos[link#][3]	float (g)	Holds x, y, z position of each link C.G.
cnumb	int (g)	Kinematic curve # index variable
defdynamicfile[80]	char (l)	Holds default name for dynamics input data file
filedef[80]	char (l)	Holds default name for kinematic input data file
gravity	float (g)	Acceleration of gravity
green[20]	int (g)	Contains green scale shading
i	int (l)	index variable
j	int (l)	index variable
jointaccel[joint#][3]	float (g)	Holds x, y, z acceleration of each joint
jointdata[joint#][15]	doub (g)	Holds all basic kinematic information

jointpos[joint#][3]	float (g)	Holds x, y, z position of each joint
jtype[joint#]	char (g)	Holds type of joint (R, P, S)
k	int (g)	index variable
kinfile[80]	char (g)	Holds default name for kinematic results file
kinfile1[80]	char (g)	Holds user entered name for kinematic results file
link_info[link#][8]	float (g)	Holds all link dynamic information
mass	float (g)	mass of payload
matres[9][4][4]	float (g)	matrix which holds resultant displacement, velocity, and acceleration matrices
nj	int (g)	number of joints
nlinks	int (g)	number of links
options[400]	char (g)	Holds all options for the 2-D and 3-D graphics
outfile[80]	char (g)	Holds default name for dynamic results file
outfile1[80]	char (g)	Holds user entered name for dynamic results file
plane[2]	char (g)	Holds the plane to be displayed in 2-D (i.e. xz, yz, xy,...)
plotpoints[curve#][9][point#]	float (g)	Holds position, velocity, and acceleration values (end-effector) for all kinematic curves
plotpoints1[curve#][9][point#]	float (g)	Holds intermediate values for position, velocity, and acceleration of end-effector
plotpoints2[curve#][9][point#]	float (g)	Holds intermediate values for position, velocity, and acceleration of end-effector
pnumb[curve#]	int (g)	number of points for each kinematic curve
pnumb1[curve#]	int (l)	number of points for each kinematic curve
red[20]	int (g)	Contains red scale shading
reljoaccel[joint#][2][3]	float (g)	Holds relative joint acceleration (acceleration of joint i+1 w/r to joint i)
rejopos[joint#][2][3]	float (g)	Holds relative joint positions (position of joint i+1 w/r to joint i)
scalearray[20]	float (g)	array which holds scaled values of velocity or acceleration utilized for shading 3-D graphics
strg[150]	char (g)	Arbitrary string array used to compose strings

varij[20]	int (g)	Contains a record of the joint # that has moved
varj	int (g)	joint # which moves (moveable joint)
velvalue[curve#][point#]	float (g)	Holds shading values for 3-D velocity shaded graphics in floating point form
vval[curve#][point#]	int (g)	Holds shading values for 3-D velocity shaded graphics in integer form
X[point#][90]	float (g)	Resultant dynamic force and moment column vector at different points.
x_coord[point#]	float (g)	Keeps track of values of moveable joint parameter (i.e. θ , h , ϕ)

- g: global ; l: local

RaMIP consists of 54 functions including the main function (main program). Next, a basic introduction of each of the functions is stated briefly.

- **addmat()** : adds the contents of one 3-D matrix to another 3-D matrix
- **brian_angles_mat()** : computes the bryant angles matrix of a given joint
- **clean_graphics_files()** : opens new data files in which mathematica graphics commands will be written.
- **constant_data()** : lets the user input all the constant data for all non-moving joints
- **curve_analysis()** : performs a dynamic analysis along the path generated by the motion of one joint
- **curve_calculation()** : combines several functions to calculate new 2-D curves
- **delete_bad_curves()** : lets user select the 3-D curves which are to be included in the final graphics (i.e. for the workspace) and deletes the others.
- **determax()** : determines maximum and minimum values of velocity and acceleration for use in 3-D graphics shading
- **dyna_graphics()** : writes dynamic results graphics files ready for mathematica 2-D plotting
- **dynamic_analysis()** : this is the main dynamic function which organizes and combines several functions to do all dynamic analyses and generate output graphics, etc...
- **dynamic_info_from_file()** : reads in all the necessary dynamic input information from a file
- **dynamic_info_to_file()** : writes all dynamic input data to a file for use later

- **dyna_results()** : organizes several functions to generate, modify, and save dynamic graphics and data results
- **fixed_joint_pos_mat_mult()** : multiplies position matrices around the moveable joint
- **formatted_points_to_file()** : write 3-D kinematic curve points to a data file
- **from_file()** : reads in all necessary kinematic input data from a file
- **funchmata()** : computes the characteristic acceleration matrix
- **funchmatd()** : computes the characteristic displacement matrix
- **funchmatv()** : computes the characteristic velocity matrix
- **input_data()** : lets user input all necessary robot kinematic information
- **input_dynamic_data()** : lets user input all necessary robot dynamic information
- **input_forces_and_moments_matrix()** : generates the relative forces and moments column vector b ($Ax = b$)
- **input_move_joint()** : lets user specify moveable joint and limit its range of motion
- **kinematic_analysis()** : this is the main kinematic function which organizes and combines several functions to do all dynamic analyses and generate output graphics, etc...
- **main()** : main program function. Organizes all kinematic, dynamic, graphics functions, and so on...
- **mat_mult()** : computes the product if $A^{-1}*b = x$, obtaining the joint forces and moments
- **matmult23()** : multiplies a 2-D array matrix by a 3-D array matrix
- **matmult33()** : multiplies a 3-D array matrix by another 3-D array matrix
- **matrix_inversion()** : computes matrix A^{-1} by inverting matrix A
- **mathematica_3d_points_to_file()** : CREATES 3-D position, velocity, and acceleration graphics files ready for input to mathematica
- **move_variable_joint()** : moves the selected moveable joint in steps, and combines other functions to make all kinematic calculations at each step
- **out_table_results()** : saves formatted dynamic results to a user named data file
- **p_analysis_joint_location()** : lets user fix the position of each one of the robot's joints, for the dynamic analysis at a specific robot location (not following a curve)
- **plot_2D_curves()** : combines several functions to generate 2-D kinematic graphic files and interfaces with mathematica to generate and/or print the plots
- **plotpoints_to_file()** : saves unformatted 3-D kinematic points to data file
- **point_analysis()** : combines several functions to perform a dynamic analysis with the robot fixed at a specific position (all joints fixed)
- **pointdeterstore()** : the most complex function in the algorithm. It combines several

functions to determine and store the end-effector position, velocity, and acceleration, through a series a complicated matrix multiplications

- **pos_accel_calculations()** : similar to function `pointdeterstore()` but it computes the position and acceleration of each individual joint not just the end-effector's position and acceleration
- **print_dyna_plot()** : it allows the user to print dynamic results plots to a printer (Postscript)
- **relative_and_cg_vectors()** : calculates all relative C.G. position and acceleration vectors based on the joint positions and accelerations
- **rel_pos_matrix_A()** : generates coefficient matrix A based on the relative C.G. position vectors
- **rotate_base_joint()** : rotates all kinematic curves about the manipulators base axis to generate the workspace, velocity, and acceleration space (this is done this way in order to save cpu time)
- **save_dyna_plot()** : saves dynamic results plots to a file in postscript or mathematica graphics language
- **save_3D_graphics()** : saves kinematic 3-D graphics to a file in postscript or mathematica graphics languages
- **select_Show_options()** : lets the user select different options in order to customize (modify) 2-D plots
- **select_Show_3D_options()** : lets the user select different options for the purpose of customizing (modify) 3-D plots
- **shading_values()** : computes gray scale and color shading values for the 3-D velocity and/or acceleration shaded graphics
- **store_points()** : stores kinematic points in a three dimensional array (plotpoints) for plotting, and others
- **to_file()** : save robot's kinematic data to a file for use later
- **transform()** : transforms (rotates) 3-D kinematic curve points about the base joint axis and calls `mathematica_3d_points_to_file()` which generates a mathematica 3-D graphics file ready for 3-D plotting
- **twoD_curves_to_file()** : generates mathematica graphics file ready for 2-D plotting
- **_2D_graphics()** : organizes several functions to generate, save, modify, and print 2-D kinematic space plots
- **_3D_graphics()** : similar to `_2Dgraphics()` function but for 3-D graphics

4.2.2 Algorithm Organization and Solution Procedures

Once all kinematic and/or dynamic initial data have been input into the program, the user must provide the specific information pertaining to the type of analysis to be accomplished. For the case of the kinematic analysis, the second step consists of an interactive process by which the total solution is built part by part, by moving one joint at a time and obtaining the results for that particular motion. For the dynamic analysis, the solution is obtained in a single solution procedure. For both type of analyses, each run consists of selecting one of the manipulator's joints as the moveable parameter (except for the dynamic solution at a point), and then obtaining a solution for that motion. For example, in the kinematic analysis one would obtain the trajectory, velocity, and acceleration of the end-effector at several points through the selected joint motion. In the dynamic analysis the solution would give the forces and moments at each one of the robot's joints for each trajectory point (each point has an associated velocity and acceleration used in this analysis). The final result for the kinematic analysis would be, for instance, the collection of kinematic trajectory curves described by the manipulator's end-effector which represent the robot's workspace, and for the dynamic analysis, the forces and moments at each one of the joints at each point of the motion of the manipulator's arm.

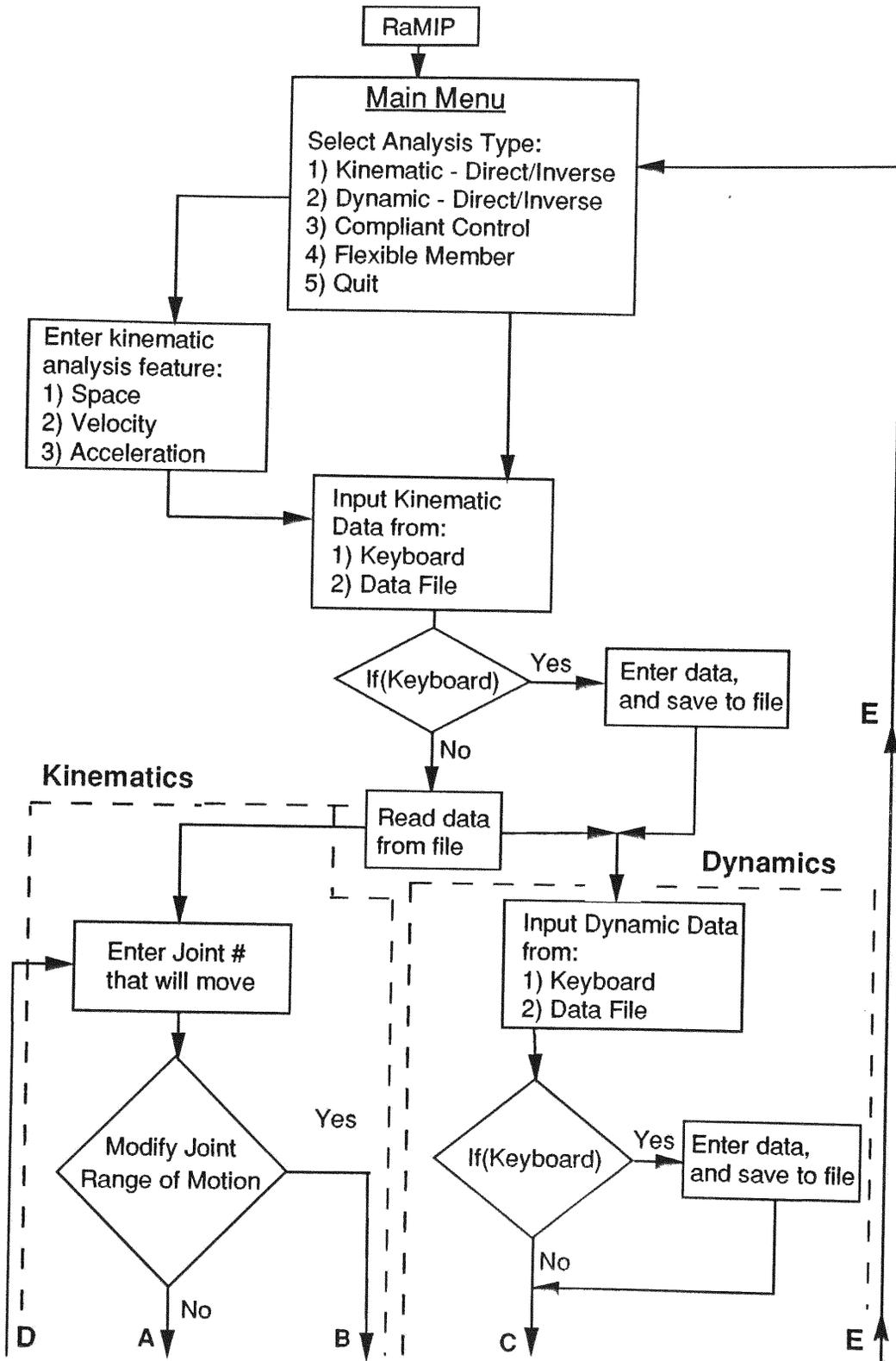


Figure 4.1 Main program

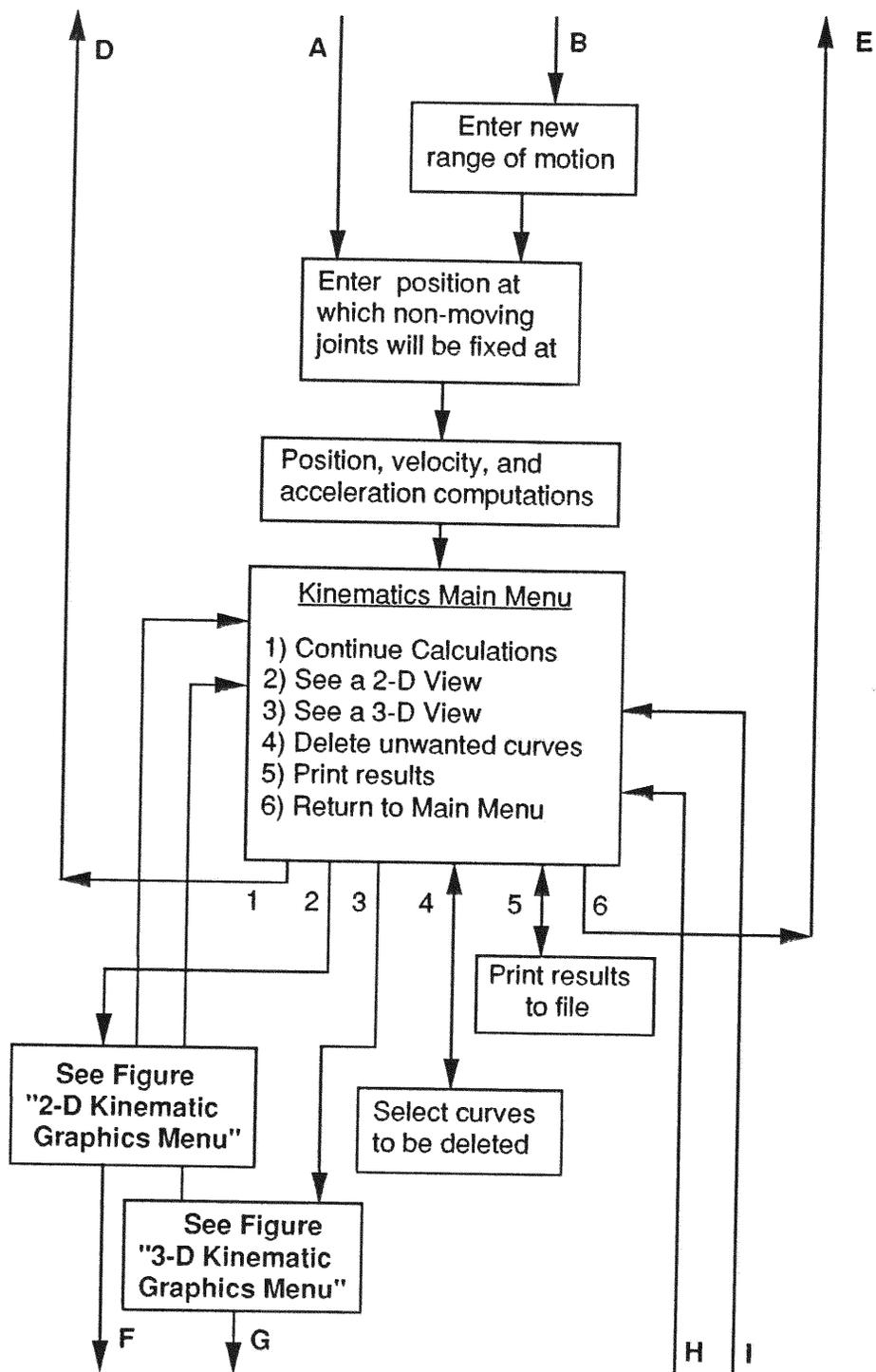


Figure 4.2 Kinematics main menu

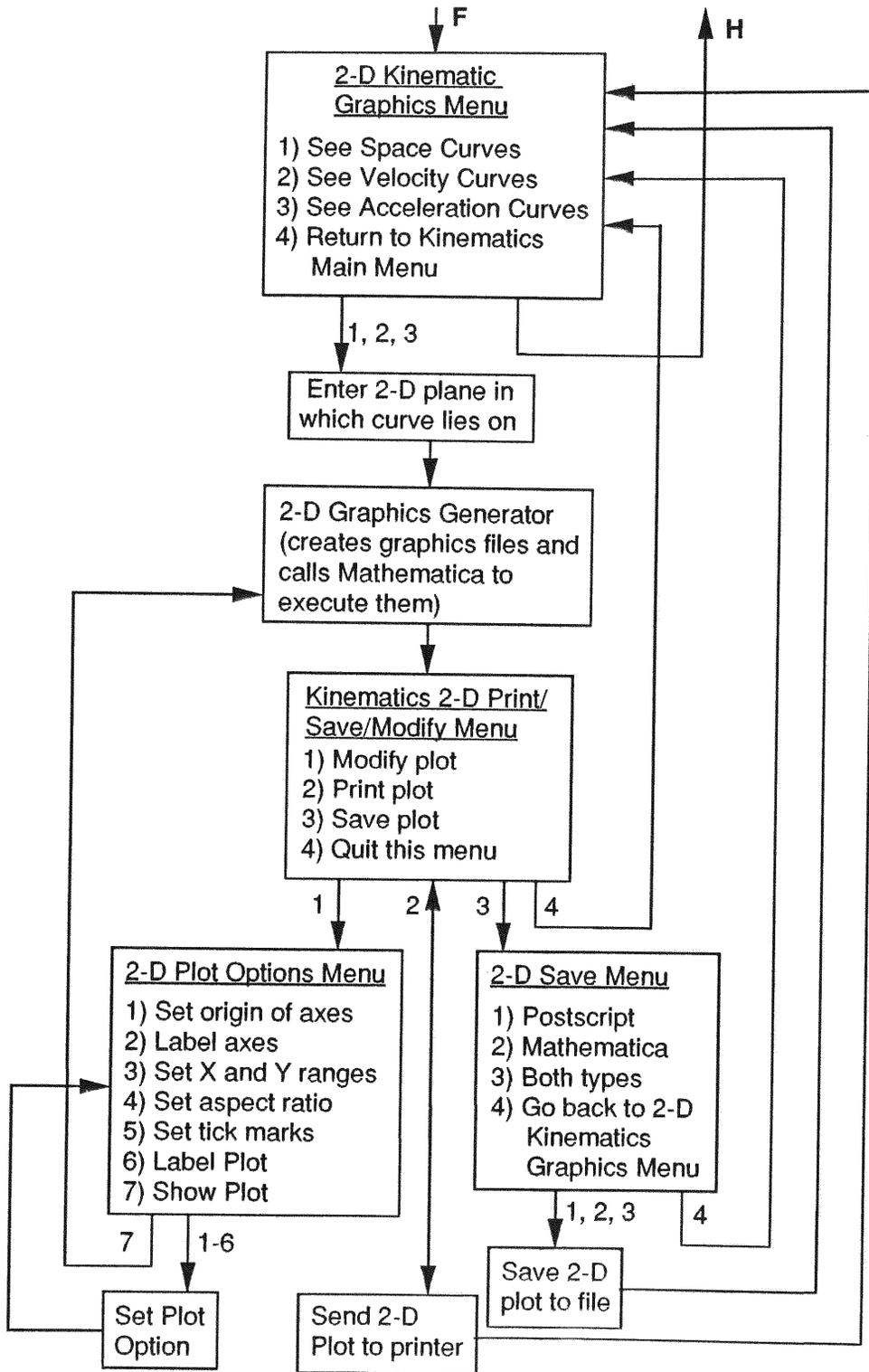


Figure 4.3 2-D kinematic graphics

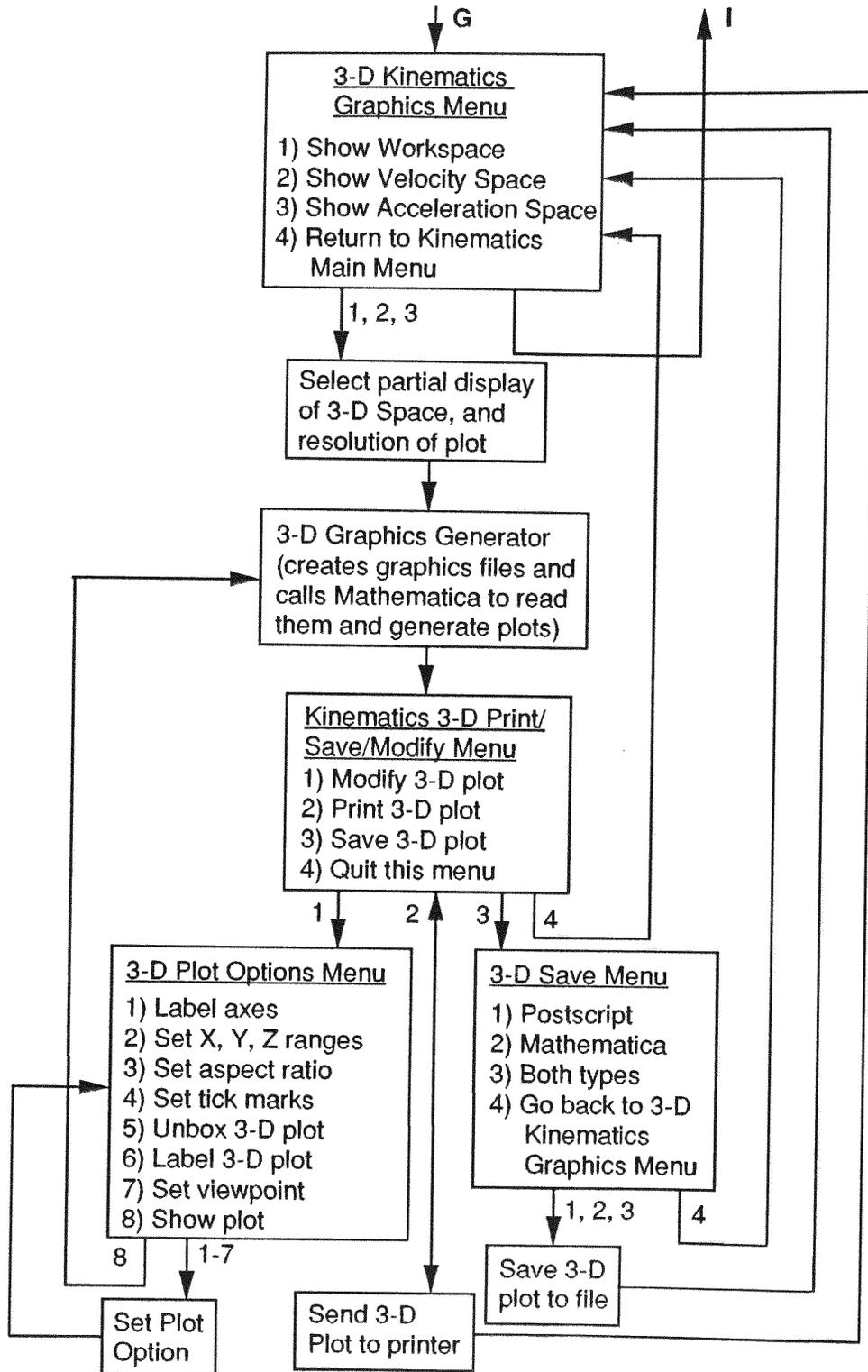


Figure 4.4 3-D kinematic graphics

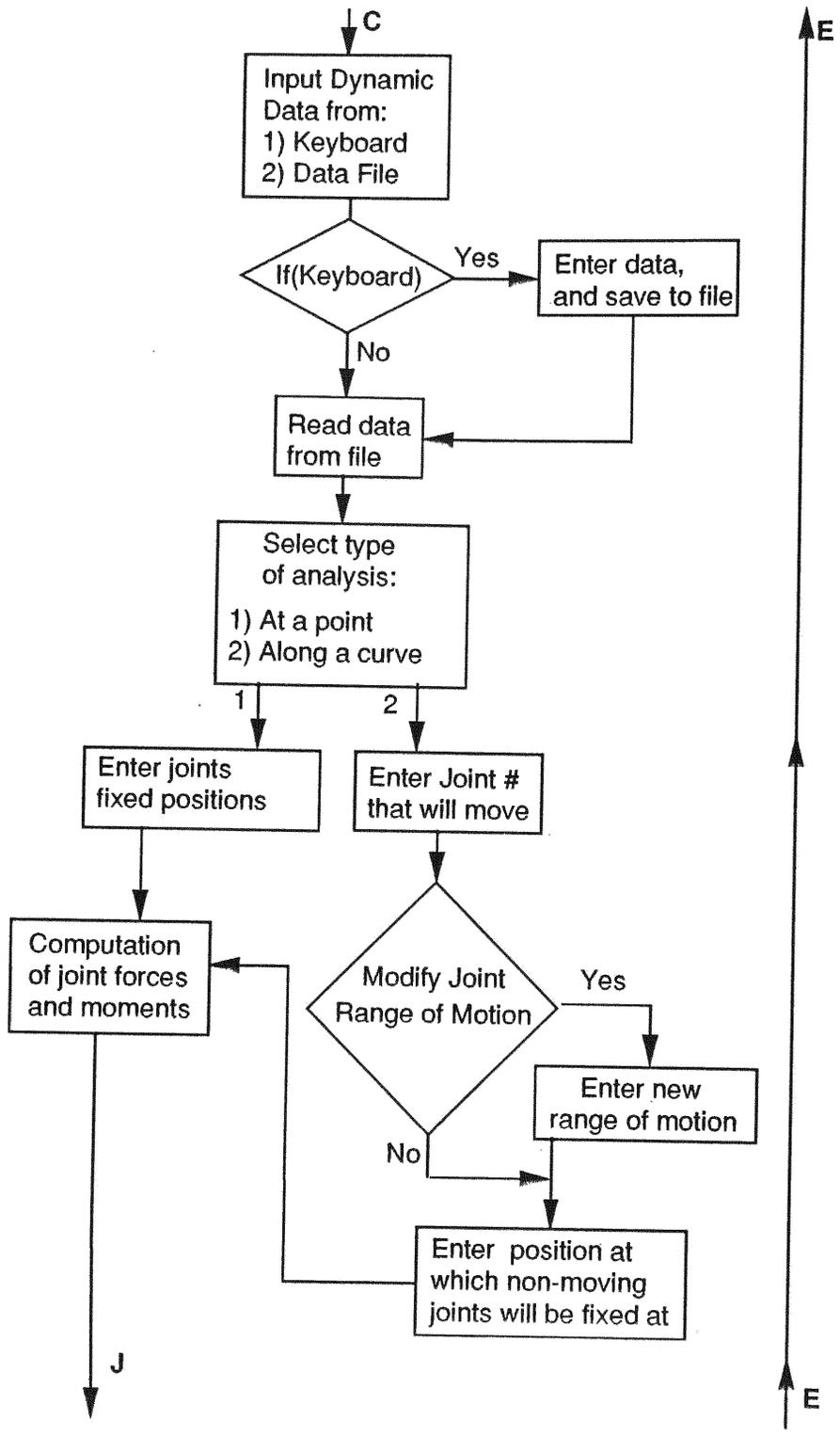


Figure 4.5 Dynamic solution procedure

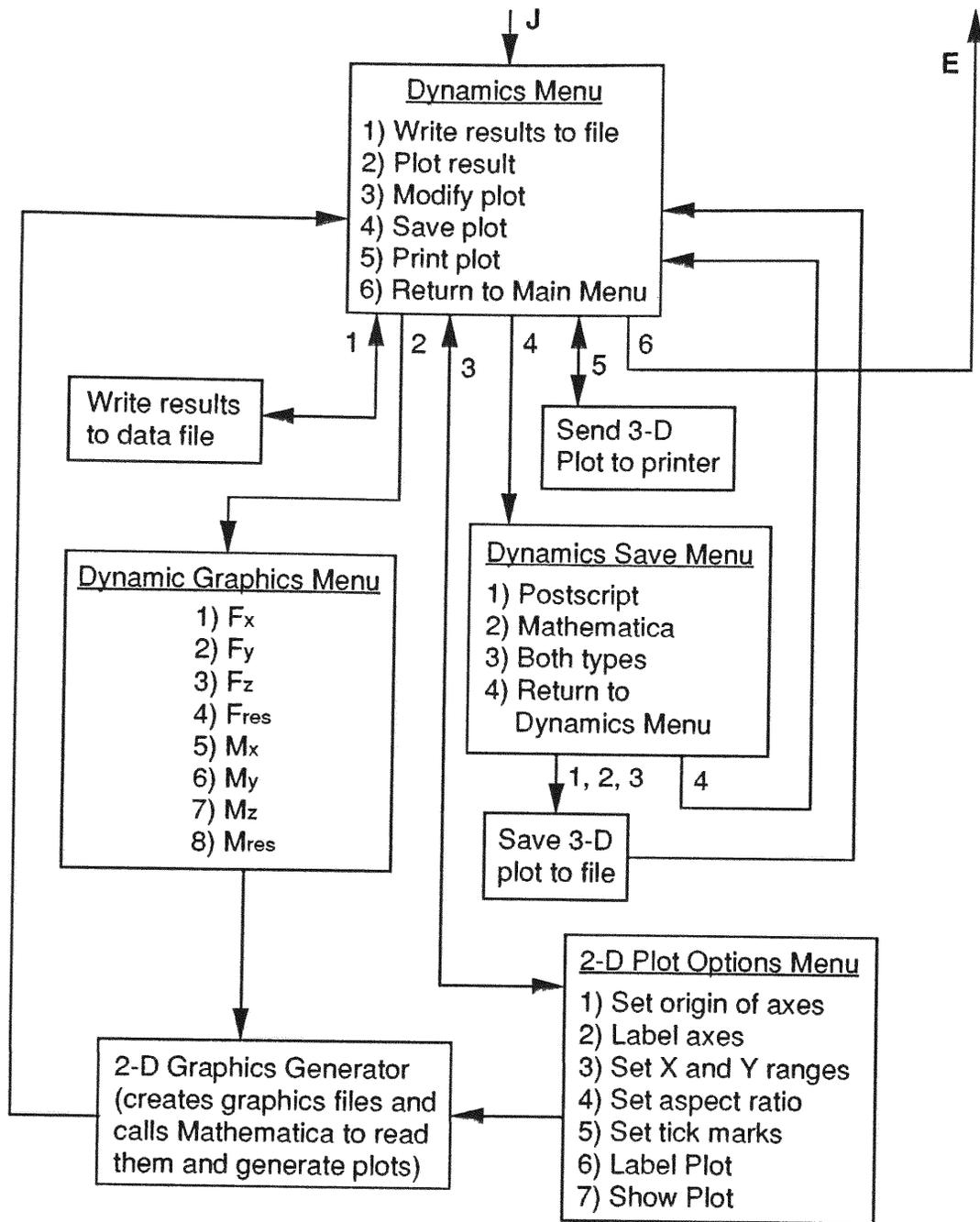


Figure 4.6 Dynamic results presentation

4.2.3 Output of Results

The third step of the analysis procedure consists of displaying, saving, and printing results. The algorithm is capable of displaying 2- and 3-dimensional graphics on the computer's monitor, and also, it can generate hard copies of the graphics by sending them to a postscript output device (such as a postscript printer). Also, the graphics may be saved to a graphics file in postscript and/or mathematica graphics language for its regeneration on the computer's screen and in hard copy form. Besides graphics, numerical output is also available in both hard copy and data files form. All this output processes have been integrated within the algorithm, without any need of external programs or procedures.

CHAPTER 5

CONCLUSIONS AND FUTURE PERSPECTIVE

Systematic approaches in the form of symbolic notations for the analysis of spatial linkage systems (mechanisms and robots) using 4x4 homogeneous matrices were originally developed for the display of three dimensional computer images. A symbolic notation provides a general shorthand representation of a mechanical system and contains the essential parameters for the complete description of the system. In this thesis, a new symbolic representation, C-B notation, was successfully utilized for the three dimensional kinematic analysis of any n-link open-chain mechanism. Also, the C-B notation was combined with the Newton-Euler formulation to derive a general matrix solution for the dynamic analysis of open-chain mechanisms. A computer algorithm, RaMIP (Robot and Mechanism Integrated Program), capable of designing and analyzing both open-chain and close-chain mechanisms, was initiated utilizing the methods mentioned before. Finally, several existing robots were analyzed using RaMIP and the corresponding kinematic and dynamic results were presented and compared with known data.

First, the C-B notation was successfully applied to model any n-link open-chain mechanism consisting of an arbitrary combination of revolute, prismatic, or spherical joints. Five robots (Cincinnati Milacron T3, Bendix AA/CNC, Unimate 2000, IBM 7576 SCARA, Space Shuttle RMS) were modeled using the C-B notation, and the kinematic analysis resulted in the determination of 3-dimensional positions, velocities, and accelerations of each joint and link C.G. The corresponding kinematic spaces were

generated graphically and displayed as 3-dimensional images. The results from the kinematic analysis were employed later in the dynamic analysis.

In chapter 3, the general dynamics matrix notation for a rigid n-link open-chain manipulator was derived. Dynamic joint forces and moments for any n-link robot were calculated based on the Newton-Euler equations of motion under the consideration of quasi-static equilibrium. The displacements, velocities, and accelerations of each joint and link C.G. were pre-determined in kinematics, and the resultant external forces and moments applied to the mass center were pre-calculated. Subsequently, the forces and moments column vector $[b]$, and displacement coefficient matrix $[A]$ were formed, and the dynamic solutions for joint forces and moments was obtained from the product of the inverse of displacement coefficient matrix $[A]$ ($[A]^{-1}$), and the forces and moments column vector $[b]$. The derived matrix notation provides a simple means of performing dynamic analysis for open-chain mechanisms. The resulting matrix notation was tested for five robots, and the joint forces and moments produced by varying the geometry (joint angle/position) of such robots was presented.

Chapter 4 describes RaMIP (Robot and Mechanism Integrated Program), the algorithm which resulted from the work accomplished in this thesis, has proven to be effective and versatile. Its analytical approach renders it general enough for the analysis of many types of robotic manipulators with the most common geometries found in industrial applications. Kinematic and dynamic solutions may be obtained in a matter of a few seconds. Also, RaMIP includes special functions which allow shortening CPU time for processes. For instance, 3-dimensional graphics may take longer than just a few

seconds for their generation on the computer screen. In order to shorten this time, RaMIP allows changing the resolution of 3-dimensional graphics and also, the selection of partial representations of 3-dimensional graphics.

RaMIP was developed to make it as user friendly as possible, using menus to move forward and backward in the analysis procedures. With respect to this, its development has not yet been finished. In the near future, a new mouse and menu graphical interface will be developed to make the analytical task even simpler. At the present time, RaMIP is capable of doing forward kinematic and inverse dynamic analyses. Further development in this aspect is currently in the process to include manipulator inverse kinematics and mechanism design. The next step planned involves the development of forward dynamics modeling, and the inclusion of close-chain and combined open- and closed-chain mechanism analysis.

REFERENCES

- Argrawal, O. P., and Shabana, A. A. (1985), "Dynamic Analysis of Multibody Systems Using Component Modes," *Computer & Structures*, 21(6):1303-1312.
- Armstrong, B., Khatlib, O., and Burdik, J. (1986), "The Explicit Dynamic Model and Inertia Parameters of the PUMA 560 Arm," *Proc. IEEE Intern. Conf. on Robotics and Automation*, San Francisco, 510-518.
- Bacquet, M. (1985), "Computing of Robot's Workspace," *Proceedings of 85 International Conference on Advanced Robotics*, Japan.
- Bae, D. S., Hwang, R. S., and Haug, E. J. (1991), "A Recursive Formulation for Real-Time Dynamic Simulation of Mechanical Systems," *J. of Mechanical Design*, 113(2):158-166.
- Bejczy, A. K. (1974), *Robot Arm Dynamics and Control*, NASA Technical Memo, 33-669, A JPL Report, or NASA-CR-136935.
- Book, W. J., (1984), "Recursive Lagrangian Dynamics of Flexible Manipulator Arms," *International Journal of Robotics Research*, 3(3):87-101.
- Chase, M. A. (1963), "Vector Analysis of Linkages," *ASME Journal of Engineering for Industry*, 85:289-297.
- Chace, M. A. and Smith, D. A. (1971), "DAMN - Digital Computer Program for the Dynamic Analysis of Generalized Mechanical Systems," *SAE Paper No. 710244*.
- Chang, C. W., and Shabana, A. A. (1990), "Spatial Dynamics of Deformable Multibody Systems With Variable Kinematic Structure: Part 2-Velocity Transformation," *J. of Mechanical Design*, 112(2):160-167.
- Cwiakala, M., and Lee, T. W. (1985), "Generation and Evaluation of a Manipulator Workspace Based on Optimum Path Search," *ASME J. of Mechanisms, Transmissions, and Automation in Design*, 107(2):245-255.
- Davidson, J. K., and Hunt, K. H. (1988), "Robot Workspace of a Tool Plane: Part 1-A-Ruled Surface and Other Geometry," *ASME 86-DET-126*.
- Davidson, J. K. (1989), "A Survey of One Class of 7-Joined Serially Connected Robots: Type-Synthesis to Obtain Controllably Dexterous Workspace," *ASME J. of Mech., Trans., and Automat in Des.*, 111(2):163-175.
- Denavit, J., and Hartenberg, R. S. (1955) "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *ASME Trans. J. Appl. Mech.* 77:215-221.
- Denavit, J. (1956), "Description and Displacement Analysis of Mechanisms Based on (2 X 2) Dual Matrices," *Ph.D. Thesis, Field of Mechanical Engineering, Northwestern University, Evanston, Il.*
- Doo-Yearn, J., and Haug, E. J. (1989), "Workspace Analysis of Multibody Mechanical

- Systems Using Continuation Methods," *J. of Mechanisms, Transmissions, and Automation in Design*, 111(4):581-589.
- Dubetz, M. W., Kuhl, J. G., and Haug, E. J. (1988), "A Network Implementation of Real-Time dynamic Simulation with Interactive Animated Graphics," *Advances in Design Automation*, S. Rao, ed., ASME DE-14:519-524.
 - Duffy, J. (1980), *Analysis of Mechanisms and Robot Manipulators*, Wiley, New York.
 - Duffy, J., and Sugimoto, K. (1981), "Determination of Extreme Distances of a Robot Hand Part 1: A General Theory," *ASME J. of Mech., Trans., and Automat. in Des.*, 103(3):631-636.
 - Duffy, J., and Sugimoto, K. (1981), "Determination of Extreme Distances of a Robot Hand Part 2: Robot arms with Special Geometry," *ASME J. of Mech., Trans., and Automat. in Des.*, 103(3):631-636.
 - Erdman, A. G., and Sandor, G. N. (1984), *Mechanism Design: Analysis and Synthesis*, Vol. 1, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
 - Erdman, A. G. (1985), "Computer-Aided Design of Mechanisms: 1984 and Beyond," *Mechanism and Machine Theory*, 20(4):245-249.
 - Featherstone, R. (1983), "The Calculation of Robot Dynamics Using Articulated-Body Inertias," *International Journal of Robotics Research*, 2(1):13-30.
 - Garcia de Jalon, J., Unda, J., Avello, A., and Jimenez, J. M. (1987), "Dynamic Analysis of Three-Dimensional Mechanisms in "Natural" Coordinates," *J. of Mechanisms, Transmissions, and Automation in Design*, 109(4):460-465.
 - Hara, A., and Sugimoto, K. (1989), "Synthesis of Parallel Manipulators," *ASME J. of Mech., Trans., and Automat. in Des.*, 111(1):34-39.
 - Hartenberg, R. S. and Denavit J. (1963), "Analysis of Spatial Linkages by Matrix Method," NSF Report, Northwestern University, September.
 - Hartenberg, R. S. and Denavit, J.(1964), *Kinematic Synthesis of Linkages*, McGraw-Hill, New York, 1964.
 - Hollerbach, J. M. (1980), "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamic Formulation Complexity," *IEEE Trans. on Systems, Man and Cybernetics*, SMC-10:730-736.
 - Hostetter, G. H., Santina, M. S., and D'Carpio-Montalvo, P. (1991), *Analytical, Numerical, and Computational Methods for Science and Engineering*, Prentice-Hall, Englewood Cliffs, New Jersey.
 - Hunt, K. H. (1983), "Structural Kinematics of In-Parallel-Actuated Robot Arms," *ASME J. of Mech., Trans., and Automat. in Des.*, 105(4):705-712.
 - Huston, R. L., Passerello, C. E. and, Harlow M. W. (1978), "Dynamics of Multi-Rigid Body Systems," *ASME Journal of Applied Mechanics*, 45(4):889-894.
 - Hwang, R. S., Bae, D. S., Kuhl, J. G., and Haug, E. J. (1990), "Parallel Processing for

- Real Time Dynamic System Simulation," *Journal of Mechanical Design*, in Press.
- Kafrissen, E., and Stephans, M. (1984), *Industrial Robots and Robotics*, Reston Publishing Company, Inc., Reston, Virginia.
 - Kohli, D. and Soni, A. H. (1975), "Kinematic Analysis of Spatial Mechanisms Via Successive Screw Displacements," *ASME Journal of Engineering for Industry*, 97(2):739-747.
 - Kohli, D. and Spanos, J. (1985), "Workspace Analysis of Mechanical Manipulators Using Polynomial Discriminants," *ASME J. of Mech., Trans., and Automat. in Des.*, 107(2): 209-215.
 - Kumar, A., and Waldron, K. J. (1981), "The Workspace of a Mechanical Manipulator," *ASME Journal of Mechanical Design*, 103:665-672.
 - Lee, C. S. G. (1982), "Robot Arm Kinematics, Dynamics, and Control," *Computer, IEEE Trans.*, Dec.:62-80.
 - Lee, T. W., and Yang, D. C. H. (1983), "On the Evaluation of Manipulator Workspace," *ASME J. of Mech., Trans., and Automat. in Des.*, 105(1):70-77.
 - Li, C. J. (1988), "A New Method of Dynamics for Robot Manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-18, 105-114.
 - Li, C. J., Hemami, A., and Sankar, T. S. (1990), "Effect of Inexact Inertial Parameters on Robot Manipulator System Performance," *Journal of Robotic Systems*, 7(5):703-720.
 - Lin, Y. J., and Song, S. M. (1990), "A Comparative Study of Inverse Dynamics of Manipulators With Closed-Chain Geometry," *Journal of Robotic Systems*, 10(3):507-534.
 - Luh, J. Y. S., Walker, M. W., and Paul, R. P. C. (1980), "On-Line Computational Scheme for Mechanical Manipulators," *ASME Journal of Dynamic Systems, Measurement, and Control*, 102(2):69-76.
 - Mabie, H. H., and Reinholtz, C. F. (1987), *Mechanisms and Dynamics of Machinery*, Fourth Edition, John Wiley & Sons, Inc, New York, New York.
 - Nakamura, S. (1991), *Applied Numerical Methods with Software*, Prentice-Hall, Englewood Cliffs, New Jersey.
 - Miller, R. (1988), *Fundamentals of Industrial Robots and Robotics*, PWS-Kent Publishing Company, Boston, Ma.
 - Nikravesh, P.E., Wehage, R. A., and Kwon, O. K. (1984), "Euler Parameters in Computational Kinematics and Dynamics, Parts I and II," *ASME Papers No. 84-DET-112 and 84-DET-113*.
 - Nikravesh, P. E. (1988), *Computer-Aided Analysis of Mechanical Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.
 - Orin, D. E., McGhee, R. B., Vukobratovic, M., and Hartoch, G.(1979), "Kinematic and Kinetic Analysis of Open-Chain Linkages Utilizing Newton-Euler Methods,"

- Mathematical Biosciences, 43:107-130.
- Orlandea, N., Chace, M. A. and Callahan, D. A. (1977), "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems," ASME Journal of Engineering for Industry, 99:773-784.
 - Paul, B.(1976), Analytical Dynamics of Mechanisms - A Computer Oriented Overview, Mechanism and Machine Theory, 10(6):481-507.
 - Paul, B. (1979), *Kinematics and Dynamics of Planar Machinery*, Prentice-Hall, Englewood Cliffs, NJ.
 - Paul, R. P. (1982), *Robot Manipulators: Mathematics, Programming, and Control*, The MIT Press, Cambridge, MA.
 - Pennock, G. P. (1983), "Kinematics and Dynamics of Multi-Rigid-Body Open-Chain Systems: Application to Robot Manipulators," Ph.D. Dissertation, University of California, Davis.
 - Pieper, D. L. and Roth, B. (1969), "The Kinematics of Manipulators Under Computer Control," Proceedings II-International Congress on the Theory of Machines and Mechanisms, 2:159-168.
 - Rastegar, J., and Dervai, P. (1987), "Methods to Determine Workspace, its Subspaces with Different Number of Configurations, and All the Possible Configurations of a Manipulator," Mechanisms and Machine Theory, 22(4):343-350.
 - Rastegar, J. (1988), "Workspace Analysis of 4R Manipulators With Various Degrees of Dexterity," J. of Mechanisms, Transmissions, and Automation in Design, 110(1):42-47.
 - Rooney, J. and Duffy, J. (1977), "On the Closures of Spatial Mechanisms," ASME Paper No. 72-Mech-77.
 - Roth, B. (1975), "Performance Evaluation of Manipulators from a Kinematic Viewpoint," National Bureau of Standards Workshop on Performance Evaluation of Manipulators.
 - Roth, B. and Shimano, B. E. (1977), "Range of Motion of Manipulators," Third CISM-IFTToMM Int. Symposium on Theory and Practice of Robot and Manipulators.
 - Samanta, B. (1990), "Dynamics of Flexible Multibody Systems Using Bond Graphs and Lagrange Multipliers," ASME J. of Mechanical Design, 112(1):30-35.
 - Schiehlen, W. O. (1984), "Dynamics of Complex Multibody Systems," SM Archives, 9:159-195.
 - Selfridge, R. G. (1983), "The Reachable Work Area of a Manipulator," Mechanisms and Machine Theory, 15:131.
 - Shahinpoor, M. (1983). *Bond Graph Dynamic Modeling of Robotic Manipulators*, Recent Developmental Applied Mathematics Series 1, I. G. Tadjbakhsh and F. F. Ling (Eds.) RPI Press, Troy, NY, 176-186.
 - Sheth, P. N. (1968), "Improved Iterative Techniques for the 4 x 4 Matrix Method of

- Kinematic Analysis," M. S. Thesis, University of Wisconsin, Madison, Wi.
- Sheth, P. N. and Uicker, J. J. (1972), "IMP (Integrated Mechanism Program): A Computer-Aided Design Analysis System for Mechanisms and Linkages," ASME Journal of Engineering for Industry, 94:454-464.
 - Sheth, P. M., Hodges, T. M., and Uicker, J. J. Jr. (1990), "Matrix Analysis Method for Direct and Multiple Contact Multibody Systems," J. of Mech. Design, 112(2):145-152.
 - Silver, W. M. (1982), "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators," The International Journal of Robotics Research, 1(1):118-128.
 - Snyder, W. E. (1985), *Industrial Robots: Computer Interfacing and Control*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
 - Song, S. M., and Lin, Y. J. (1989), "A Computationally Efficient Method for Manipulator Dynamic Analysis," Journal of Robotic Systems, 6(2):175-189.
 - Soni, A. H., and Tsai, Y. C. (1983), "An Algorithm for the Workspace of a General n-R robot," ASME J. of Mech., Trans., and Automat. in Des., 105(1):52-57.
 - Sugimoto, K. (1986), "Kinematic and Dynamic Analysis of Parallel Manipulators by Means of Motor Algebra," ASME paper, 86-DET-139.
 - Sugimoto, K. (1989), "Computational Scheme for Dynamic Analysis of Parallel Manipulators," ASME J. of Mech., Trans., and Automat. in Des., 111(1):29-33.
 - Suh, C. H., and Kang, H. Y. (1989) "A Dynamic Analysis of a Spatial Mechanism With a Passive Degree of Freedom," J. of Mech., Trans., and Automat. in Des., 111(2):238-242.
 - Sunada, W. H., and Dubowsky, S. (1983), "On the Dynamic Analysis and Behavior of Industrial Robotic Manipulators with Elastic Members," ASME J. of Mech. Trans. and Automation in Design, 105(1):42-52.
 - Townsend, M. A., and Gupta, S. (1989), "Automated Modeling and Rapid Solution of Robot Dynamics Using the Symbolic Polynomial Technique," J. of Mechanisms, Transmissions, and Automation in Design, 111(4):537-544.
 - Thomas, M., and Tesar, D. (1982), "Dynamics of Manipulators," ASME Journal of Dynamic Systems, Measurement, and Control, 1982.
 - Tsai, Y. C., and Soni, A. H. (1981), "Accessible Region and Synthesis of Robot Arms," ASME Journal of Mechanical Design, 103(4):803-811.
 - Tsai, L. W., and Morgan, A. P. (1985), "Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods," ASME J. Mech., Trans, and Automat. in Des., 107(2):189-200.
 - Uicker, J. J. (1965), On the Dynamic Analysis of Spatial Linkages Using 4x4 Matrices, Ph.D. Thesis, Department of Mechanical Engineering and Astronautical Sciences, Northwestern University.
 - Uicker, J. J. (1969), "Dynamic Behavior of Spatial Linkages: Part 1 - Exact Equations

- of Motion; Part 2 - Small Oscillations About Equilibrium," J. of Eng. Ind. Trans. ASME, 91(series B): 251-265.
- Unda, J. Garcia de Jalon, J., Losantos, F., and Enparantza, R. (1987), "A Comparative Study on Some Different Formulations of the Dynamic Equations of Constrained Mechanical Systems," J. of Mechanisms Transmissions and Automation in Design, 109(4):466-474.
 - Vijaykumar, R., Tsai, M. J., and Waldron, K. J. (1985), "Geometric Optimization of Manipulator Structures for Working Volume and Dexterity," ASME J. of Mech., Trans., and Automat. in Des., 110:42-47.
 - Walker, M. W., and Orin, D. E. (1982), "Efficient Dynamic Computer Simulation of Robotics Mechanism," ASME Journal of Dynamic Systems, Measurement, and Control, 104:205-211.
 - Wang, T., and Kohli, D. (1985), "Closed and Expanded Form of Manipulator Dynamics Using Lagrangian Approach," J. of Mech., Trans., and Automat. in Des., 107(2):223-225.
 - Wehage, R. A., and Haug, E. J. (1982), "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," ASME Journal of Mechanical design, 104(1):247-255.
 - Wolfram, S. (1988), "*Mathematica: A System for Doing Mathematics by Computer*", Addison-Wesley Publishing Company.
 - Yang, A. T. and Freudenstein, F. (1964), "Application of Dual Numbers Quaternion Algebra to the Analysis of Spatial Mechanisms," ASME Journal of Applied Mechanics, 86:300-308.
 - Yang, A. T. (1969), "Displacement Analysis of Spatial Five-Link Mechanisms Using (3 x 3) Matrices with Dual-Number Elements," ASME Journal of Engineering for Industry, 91(1):152-157.
 - Yang, D. C. H., and Lee, T. W. (1983), "On the Workspace of Mechanical Manipulators," ASME Journal of Mechanisms, Transmissions, and Automation in Design, 105(1):62-70.
 - Yang, D. C. H., Lai, Z. C. (1985) "On the dexterity of robotic Manipulators, Service Angle," ASME J. of Mech., Trans., and Automat. in Des., 107(2):262-270 (1985).
 - Yang, D. C. H., Lin, E. Y., and Cheng, S. Y. (1990), "Primary Workspace of Industrial Robots With Roll-Pitch-Yaw Wrists," J. of Mechanical Design, 112(3):347-353.
 - Yih, T. C., and Youm, Y. (1986), "The Kinematic Spaces of Planar n-R Open-Loop System with Rotating Base," ASME paper, 86-DET-98.
 - Yih, T.C., (1987) "Unified Analysis of Mechanical Systems by the Spherical-Euler Transformation Method, a Computer-Aided Mechanical System Design," Ph.D. Thesis, Field of Mechanical Engineering, The Catholic University of America, Washington D.C..

- Yih, T. C., and Youm, Y. (1989), "Analysis of Spatial Open-Loop System by Means of Direction Cosine Transformation Matrices," ASME J. of Mech., Trans., and Automat. in Des., 111(4):508-512.
- Yih, T. C., and Donoso, B. (1989), "On the 3-Dimensional Workspace of Robots by Applying C-B Notation," Proceedings of National Conference on Applied Mechanisms and Robotics, Cincinnati, Ohio, November 5-8, pp. 20-25.
- Yih, T. C. (1991), "A New Method for the Modeling of Lower Pairs and its Application to the Kinematic Spaces of Spatial Robots," Journal of Robotic Systems, 8(4):415-442.

APPENDIX

ROBOT KINEMATIC AND DYNAMIC DATA

The following tables contain all kinematic joint parameters (in C-B notation convention) for each one of the robots used as examples for kinematic and dynamic analysis.

Table A.1a Kinematic Parameters for Cincinnati Milacron T3 Robot

Joint	θ [deg]	h [m]	r [m]	α [deg]	β [deg]	variable	position [deg]
1(R)	-120/120	1.5	0	90	0	θ_1	30
2(R)	0/90	0	1.067	0	0	θ_2	45
3(R)	-150/0	0	1.067	0	0	θ_3	variable
4(R)	-90/90	0	0.205	270	0	θ_4	0
5(R)	-90/90	0	0.369	0	90	θ_5	0
6(R)	-135/135	0	0	0	0	θ_6	0

Table A.1b Dynamic Parameters for Cincinnati Milacron T3 Robot

Link No.	m [Kg]	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{xz}
1	1.4	2.00	2.00	0.634	0.0	0.0	0.0
2	6.8	2.94	58.8	58.8	0.0	0.0	0.0
3	6.8	2.94	58.8	58.8	0.0	0.0	0.0
4	1.8	0.226	6.22	6.22	0.0	0.0	0.0
5	1.4	2.00	0.634	2.00	0.0	0.0	0.0
6	2.6	1.02	3.5	3.50	0.0	0.0	0.0

Joint velocities and accelerations are assigned unity. Payload = 25 Kg.

Table A.2a Kinematic Parameters for Bendix AA/CNC Robot

Joint	θ [deg]	h [m]	r [m]	α [deg]	β [deg]	variable	position [deg]
1(R)	-95/95	1.067	0	90	0	θ_1	60
2(R)	-45/225	0	0.659	90	90	θ_2	-45
3(P)	0	0/0.61	0	0	0	h_3	variable
4(R)	-95/95	0.109	0	90	0	θ_4	0
5(R)	-20/200	0	0.146	90	90	θ_5	-20
6(R)	0/360	0	0	0	0	θ_6	arbitrary

Table A.2b Dynamic Parameters for Bendix AA/CNC Robot

Link No.	m [Kg]	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{xz}
1	12.0	1.00	1.00	1.00	0.0	0.0	0.0
2	10.0	1.00	1.00	1.00	0.0	0.0	0.0
3	9.0	1.00	1.00	1.00	0.0	0.0	0.0
4	6.0	1.00	1.00	1.00	0.0	0.0	0.0
5	5.0	1.00	1.00	1.00	0.0	0.0	0.0
6	2.5	1.00	1.00	1.00	0.0	0.0	0.0

Joint velocities and accelerations are assigned unity. Payload = 25 Kg.

I_{xx} , I_{yy} , and I_{zz} are assigned unity for generality purposes

Table A.3a Kinematic Parameters for Unimate 2000 Spherical Robot

Joint	θ [deg]	h [m]	r [m]	α [deg]	β [deg]	variable	position
1(S)	-104/104	**	**	0	90	θ_1, ϕ_1	$\theta_1=100$ $\phi_1=var.$
2(P)	90	0.91/2.03	0	0	270	h_2	2.03
3(R)	-110/110	0	0.2	270	0	θ_3	0
4(R)	-100/100	0	0.2	0	90	θ_4	0
5(R)	0/360	0	0	0	0	θ_5	0

** $h = a \cos\phi$ and $r = a \sin\phi$; $a = 0.12$ m and $-26^\circ \leq \phi \leq 30^\circ$

Table A.3b Dynamic Parameters for Unimate 2000 Spherical Robot

Link No.	m [Kg]	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{xz}
1	25.0	1.00	1.00	1.00	0.0	0.0	0.0
2	18.0	1.00	1.00	1.00	0.0	0.0	0.0
3	15.0	1.00	1.00	1.00	0.0	0.0	0.0
4	10.0	1.00	1.00	1.00	0.0	0.0	0.0
5	7.0	1.00	1.00	1.00	0.0	0.0	0.0

Joint velocities and accelerations are assigned unity. Payload = 25 Kg.

I_{xx} , I_{yy} , and I_{zz} are assigned unity for generality purposes

Table A.4a Kinematic Parameters for IBM 7576 (SCARA) Robot

Joint	θ	h(m)	r(m)	α	β	variable
1(R)	-120/120	0.5	0.4	0	0	θ_1
2(R)	-136/136	0	0.4	0	0	θ_2
3(P)	0	0/0.25	0	0	180	h_3
4(R)	-3600/3600	0	0	0	0	θ_4

Joint velocities and accelerations are assigned unity. Payload = 25 Kg.

Table A.4b Dynamic Parameters for IBM 7576 (SCARA) Robot

Link No.	m [Kg]	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{xz}
1	21.0	1.00	1.00	1.00	0.0	0.0	0.0
2	18.0	1.00	1.00	1.00	0.0	0.0	0.0
3	2.0	1.00	1.00	1.00	0.0	0.0	0.0
4	2.0	1.00	1.00	1.00	0.0	0.0	0.0

I_{xx} , I_{yy} , and I_{zz} are assigned unity for generality purposes

Table A.5a Kinematic Parameters for Space Shuttle (RMS) Robot

Joint	θ	h(m)	r(m)	α	β	variable
1(R)	-2/145	1.0	0.10	90	0	θ_1
2(R)	-180/180	0	6.3767	0	0	θ_2
3(R)	-160/2	0	7.0599	0	0	θ_3
4(R)	-120/120	0	0.5	270	0	θ_4
5(R)	-120/120	0	0.5	0	90	θ_5
6(R)	-447/447	0	.8796	0	0	θ_6

Joint velocities and accelerations are assigned unity. Payload = 25 Kg.

Table A.5b Dynamic Parameters for Space Shuttle (RMS) Robot

Link No.	m [Kg]	I_{xx}	I_{yy}	I_{zz}	I_{xy}	I_{yz}	I_{xz}
1	50.0	1.00	1.00	1.00	0.0	0.0	0.0
2	140.0	1.00	1.00	1.00	0.0	0.0	0.0
3	110.0	1.00	1.00	1.00	0.0	0.0	0.0
4	45.0	1.00	1.00	1.00	0.0	0.0	0.0
5	30.0	1.00	1.00	1.00	0.0	0.0	0.0
6	25.0	1.00	1.00	1.00	0.0	0.0	0.0

I_{xx} , I_{yy} , and I_{zz} are assigned unity for generality purposes