BHATT, VINAY DHIRAJLAL

A COMPUTER AIDED DESIGN APPROACH FOR OPTIMAL SYNTHESIS OF A HIGH SPEED, HIGH PRECISION PLANAR MANIPULATOR FOR PATH GENERATION AND PICK & PLACE APPLICATIONS

THE UNIVERSITY OF ARIZONA M.S. 1984

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark __√__.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background ✓
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _____
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages ✓
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Other _____

A COMPUTER AIDED DESIGN APPROACH FOR OPTIMAL SYNTHESIS OF A HIGH SPEED, HIGH PRECISION PLANAR MANIPULATOR FOR PATH GENERATION AND PICK & PLACE APPLICATIONS

by

Vinay Dhirajlal Bhatt

A thesis submitted to the faculty of the

AEROSPACE AND MECHANICAL ENGINEERING DEPARTMENT

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN MECHANICAL ENGINEERING

In the Graduate college

THE UNIVERSITY OF ARIZONA

1984

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: Vinay Dhirajlal Bhatt

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

DR. K. M. RAGSDELL
Professor of Mechanical Engineering

4/30/84
Date

to the science of iternal knowledge

ACKNOWLEDGEMENT

First of all, my sincere thanks are due to my major advisor, Dr. K. M. Ragsdell, for introducing me to the latest technologies of the modern scientific world, namely, Robotics, Computer Aided Design and Design Optimization. I am grateful for the resources made available to me and for his personal attention and encouragement throughout the research. I am also grateful to Dr. Terry Triffett, associate dean of engineering and the college of engineering for providing support for this research project. I take the opportunity to thank the members of my final examination committee, Dr. Roemer and Dr. Kamel for their valuable suggestions. The Computer Aided Engineering Center, managed by Mr. Thomas Cram, proved to be a stimulating research environment.

It is also acknowledged here that previous work on analysis of 'ROBOPT' carried out at Purdue University by Dr. Failahi, under the able guidance of Dr. Ragsdell, proved quite valuable.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS--Continued

LIST OF TABLES

ABSTRACT

ROBOTICS is the latest technological sophistication the modern scientific world has produced. Industrial robots are basically manipulators that can be easily programmed to perform a variety of manual tasks automatically. High speed, high precision robots are the next generation of robots. These robots deserve special attention from the designers due to special problems of inertia and vibrations. The approach adopted in this research displays how a computer aided design technique may be used to simplify optimal synthesis of a high speed, planar manipulator for a path generation application. This computer controlled robot may also be used for 'Pick & Place' applications. The robot motions are simulated on PS300, a high performance, dynamic display system. The optimized robot is named ROBOPT.

In today's competitive industry, a challenge for productivity and efficiency must be met with by intelligent application of computer aided design to robotics. A sincere effort is made towards this goal through the introduction of ROBOPT.

CHAPTER 1

INTRODUCTION

ROBOTICS is the latest technological sophistication the modern scientific world has produced. "INDUSTRIAL ROBOTS [19] in use today are basically manipulators that can be easily programmed to do a variety of manual tasks automatically". The motivation for most current industrial robotic development is: to increase productivity, reduce costs, overcome skilled labour shortages, provide flexibility in batch operations, improve product quality and to free humans from boring, repetitive tasks or operations in hostile environments.

1.1 General Robotics

An industrial robot primarily consists of (i) a manipulator, (ii) an end effector, (iii) a controller, (iv) a power supply and (v) sensors. MANIPULATORS are multi degrees of freedom open-loop kinematic chains. One end is either fixed or hinged to the support. The other end moves freely in space and holds an end effector. With regard to mechanisms theory, the manipulator consists of a series of links with either rotary (revolute) or translatory (prismatic) joints. The END EFFECTOR, often referred to as

(prismatic) joints. The END EFFECTOR, often referred to as the GRIPPER, holds the tool. The CONTROLLER is the main controlling unit with a microprocessor and closed loop feedback servoing. It processes all the commands directed to the robot and converts them into electrical pulses to be fed to the power unit for the required action. Feedback loops reduce positional, velocity and acceleration errors for accurate positioning and smooth motions. POWER UNITS are the actuators to apply movements to various links. Selecting an actuator for a joint is a state-of-art. Three kinds of actuators, mostly used, are pneumatic, hydraulic and electric. SENSORS vary from simple mechanical stops to sophisticated vision and tactile sensing devices.

Broadly speaking, industrial robots and manipulators may be subdivided into GENERAL-PURPOSE robots and SPECIAL-PURPOSE robots. In case of spatial robots with 6 degrees of freedom, only three degrees of freedom are required to position the end effector. The rest of the degrees of freedom, called yaw, pitch, and roll, are used for orientation of the gripper. These are general purpose robots. They are designed and produced to perform a large variety of tasks. The STANFORD MANIPULATOR is an example of a 6 degrees of freedom, general purpose robot [29]. Special-purpose robots are designed and produced for a limited range of specific tasks. They usually employ manipulators with limited degrees of freedom. ROBOPT, the

manipulator designed through the research presented in this thesis, is a specific application robot with 3 degrees of freedom associated with its three revolute joints. Constant gripper orientation is assumed. Figure 1 displays an artist's conception of ROBOPT. Figure 2 shows the graphical simulation of the STANFORD MANIPULATOR on PS300.

A novice reader in the field of robotics is directed to Engelberger [9], for introduction to robots, and Paul [29], for mathematics, programming and control of robots, to serve as good general references.

1.2 Computer Aided Design Applied To Robotics

One of the goals of Computer aided design (CAD) is to provide the engineer or designer with a means of conceptualizing and conveniently transforming ideas into form. In general, this conceptualization takes the form of either WIRE-FRAME MODELING on storage tube and vector refresh devices or SOLID MODELING on a raster-scan display. GRAPHICS [27] becomes more powerful when dynamic display capabilities are available. In this case, instead of trying to mentally visualize the movements of various parts, the designer may view them directly on the graphics terminal via simulation and animation.

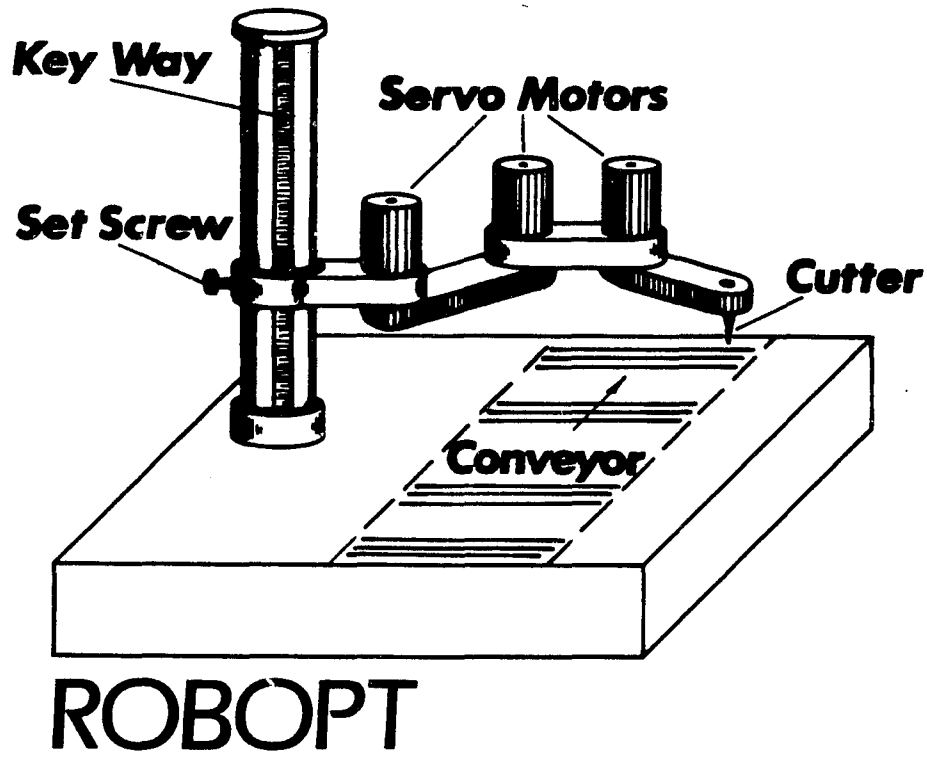


Figure 1 : An Artist's Conception of ROBOPT

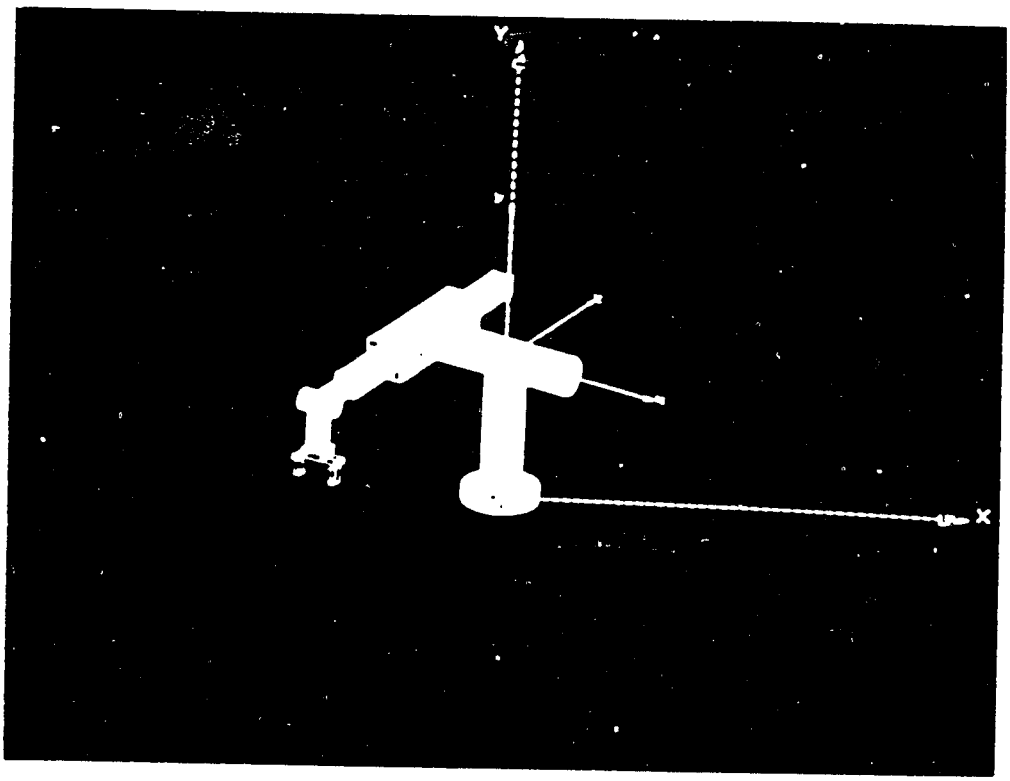


Figure 2 : Stanford Manipulator Model on PS300

According to a recent study [16] by the center for productivity of the National Science Foundation, "CAD/CAM has more potential to radically increase productivity than any development since electricity".

When computer aided design technique is applied to robotics, new robot designs can be analyzed and experimented with before investing time and expense in prototyping. Potential robotic applications can be modeled, ensuring that all required motions can be accomplished by the robot and that cycle time requirements can be met. An analysis of this sort yields, essentially, a verified robot type, tooling design, and shop floor layout, prior to the decision to invest.

ROBOTIC ANIMATION is also an excellent debugging and verification tool to assist in the development of offline robot programs. Real-time animation proves extremely helpful when relative motions and parts interferences are to be visualized. Finally, robotic animation in general provides an excellent robotics training tool and can be expected to be exploited for this purpose in the future. One such attempt of robotic animation in the past is due to KRETCH [24] at McDonnel Douglas, St. Louis, Missouri.

1.3 General Layout

The approach adopted in this research displays how a computer aided design (CAD) technique may be used to simplify synthesis of manipulators. A design optimization procedure offers flexibility for handling constraints developed during the design process and in different industrial situations. A relatively difficult problem of 'PATH GENERATION' is handled by a simple manipulator configuration. A simpler, still important application, called 'PICK & PLACE', is handled by solving a system of non-linear equations. This provides point to point control over robot motion. High speed, high precision robots have wide applications in industry but due to special problems of inertia and vibration, not many are available in the robotics market today. Two previous developments of high speed robots using nonlinear control theory are by European designers, Freund [15] and Vukobratovic [42].

Robotics is a multi-disciplinary field. In this thesis emphasis is given to manipulator dynamics and simulation. For an overview of "robotic art with regard to mechanisms theory", the reader is referred to a noteworthy article [37] by Dr. Bernard Roth of Stanford University.

The thesis text is divided in two parts. Initial chapters contain theoretical details of high speed robotics

and development of the design of ROBOPT. Later chapters include description of computer programming methods used, including programming for the PS300 dynamic display. However, cross references are given wherever necessary. After a brief introduction to the field of high speed robotics, the design problem is explained informally followed by the discription of design methodology, the ingenious design approach and selected configuration of ROBOPT. Chapter 4 contains the theory behind ROBOPT. Differential equations of motion are derived using Lagrangian dynamics. Kinematics and dynamics of joints, center of gravity and end effector point are also shown. The second section of this chapter deals with synthesis, the heart of the path generation problem. Chapter 5 explains specific applications of ROBOPT. Numerical methods used in developing the software are grouped together to form chapter 6. Programming pertaining to three dimensional dynamic display on the PS300 graphics system is elaborated in chapter 7 under the heading 'DYNAMIC SIMULATION'. The next chapter summarizes important aspects of interactive programming methods. The theme of the research is summarized at the end, including possible future extensions in the same line of approach.

Epitomizing, the goal is to contribute to the ever-expanding field of 'robot technology' by producing concrete results from an efficient interaction of CAD, robotics and design optimization.

CHAPTER 2

HIGH SPEED, HIGH PRECISION ROBOTICS

There is a need for a 'new generation of robots'. Here are some predictions regarding the 'new generation of robots' by Dr. Warren P. Seering of Massachusetts Institute of Technology. In a recently published article [39], Dr. Seering writes,

> What should be the design configuration of future robots? As the area develops, there will be many new, nonhumanoid configurations. Each will be designed to perform a generic set of tasks; there will be much less emphasis on 'general purpose' robots. The NEXT GENERATION OF ROBOTS WILL INCLUDE MANIPULATORS WITH FEWER THAN FIVE DEGREES-OF-FREEDOM. In a typical manufacturing station two or more units will appear. And their design configurations will be selected to match the types of tasks to be performed. Some will be light and fast and have large ranges of motions. Others will have more limited motions but will be capable of high speeds and high stiffness. Some will be more like programmable fixtures. Some, but not all, robots will be capable of withstanding large loads. While robot configurations for producing gross motion will change, robot end-effector design will change even more. One change will be an increase in the sensing capability of the end-effector. But more importantly, end effectors will be designed less as general purpose 'hands' and more as special purpose devices which depend on the robot superstructure primarily for gross positioning. Another difference will be that the NEXT GENERATION OF ROBOTS WILL MOVE FASTER more of the time. Current state-of-the-art involves a robot which descends upon an object at a reasonable speed, pauses to calculate, approaches slowly, pauses, grasps, pauses, retracts slowly, pauses, and slues at reasonable speed to a new location at which the sequence of slow motions and pauses is repeated. Some of this delays are required

> because of limitations in the design of the robot, controller, or fixturing. Others result from conservative programming practices. None can be tolerated if the robot is to use its workspace most productively. End-effector and workspace designs for future robots will be configured to allow continuous rapid robot motion for a large fraction of the time.
>
> Robotic manipulators will have a significant impact on manufacturing in the next five to ten years. And also one's imagination limits perception of technical innovations which will influence automated manufacturing within the productive lifetimes of today's graduating engineers.

ROBOPT is a step forward to the next generation of robots.

1.1 Dynamics Of High Speed Robotics

High Speed, High Precision (HSHP) ROBOTICS deserves special attention. " The demand for higher speed and greater precision has created problems for designers. Increasing velocity characteristics of machines using automated control, and high requirements placed upon their accuracy and reliability may determine special methods of study of dynamics" [4]. As speed increases, the old geometric relationships between the input and point to point or continuous path motion are no longer completely valid. Whether caused by dynamically induced or externally applied loads, the dynamic analysis of mechanisms is becoming a more important phase in design procedures.

A major dynamic problem related to high speed, high precision motion is internally induced by INERTIAL EFFECTS. In many instances, inertial coefficients are variables in position, velocity, and time. They are not constant, which may be assumed during low speed motion. Also, variation in the quivalent mass of the piece being translated will have negative effects. This is a problem especially with high speed mechanisms involved in loading and unloading or other production line applications [2]. Pin forces, axial acceleration, and thus bearing loads are further problems that must be considered from an inertial standpoint during high speed motions. Ignorance in this area will cause inaccuracies of position and fatigue, which could result in failure [25].

Another obvious high speed dynamic problem is VIBRATION. At low speeds, many mechanisms operate in a smooth manner, but as speed is increased, many small imbalances amplify to cause vibrational effects that render the mechanism inoperable. Further development of methods of balancing and equilibrating mechanisms is required, taking into account clearances and elasticity. One method involves finite element analysis, which considers both axial and lateral vibrations and conserves moment compatibility between elements. However, more development of this method is needed to analyze spatial mechanisms. Moreover, vibration

at high speeds is supplemented by impact forces created through quick decelerations. It was noted that wear and fatigue failure could be reduced considerably if the elements were considered slightly elastic, but this would add to inertial problems and increased bearing loads.

Increased speeds and loads also cause deflection and elastic deformation which results in poor repeatability and precision. In the past it was customary to over-design mechanisms to assure that failure would be avoided, which increased mass and inertial forces, producing additional linkage stress and bearing loads. In attempting to remedy the problem, KINETO-ELASTODYNAMICS was developed to study the motion of mechanisms consisting of elements which may deflect due to external loads or internal body forces. Previous forms of kineto-elastodynamics synthesis methods only eliminated deflections at specific points which would not be desirable for the analysis of continuous path applications. Presently, continuous motions have been solved by simply using more prescribed points due to the help of computers. "The design process is initialized by doing a kinematic synthesis assuming rigid links. Then, the kinematic analysis is done, proportioning the link cross sectional areas to counter the effects of deflection and elasticity. Using a simplex method and MINIMAL MASS, the DESIGN IS OPTIMIZED in accordance with prescribed

constraints" [38]. However, bearing loads are greatly increased when the elasticity of the mechanical element is considered.

Although the problems of high speed dynamics are critical when considering their effects on robotic motion, the fundamental aspect of robotic control is another major issue. Questions about creating system reliability, optimum parameters, and varying velocities and moments are all related to control. Yet, there lacks development of control theory for robotic manipulators.

Digital programming and the rapid development of microprocessors have assisted in creating an artificial intellect for robotics in their working environments. Moreover, digital programming and computation has made the design and synthesis of robots much easier. As the speed of motion increases, digital control will need to improve in order to insure that optimal design parameters are not violated. This improvement is not centered around actual digital computation. Rather the problem has focused mainly around the transmission of control, specifically on the aspect of robotic sensing. Both robot vision and tactile sensing capabilities of industrial robots today are unable to cope up with the required speed and reliability of high speed manipulators [3].

To achieve higher speeds, flexibility, accuracy, efficiency and dexterity that is desired in future robots, additional research in manipulator arms is needed. For large manipulators and precision work, joint and arm structural flexibilities can be a problem. Therefore stiff, but lightweight arms are desirable to minimize this problem and to aid in increasing speed. However as variable compliance in different directions is often desirable to provide the "give" needed for inserting parts and other assembly operations, unique design approaches may be required. Smaller, more reliable and efficient actuators and drive mechanisms, with high load to weight ratios, are needed for high speed, light weight robots of the future [19].

A standardized approach to performance measurement and evaluation is required to enable comparision of robots not only in such factors as working volume, load capacity and reliability, but also in aspects such as speed, smoothness and accuracy throughout their operating range.

CHAPTER 3

DESIGN PROBLEM

The current state of development of robotic systems and mechanical manipulators in USA is somewhat different than that observed abroad; especially in Japan and Germany. In those countries programmable mechanical manipulators/robotic systems are widely used for even simple but repetitive tasks, as well as in hostile environments (such as nuclear reactors, steel mills, etc.). There is a wide range of capability of available robotic systems; from the HARD AUTOMATION TYPE, thru MILDLY SMART DEVICES; to more sophisticated systems employing touch and vision feedback and powerful minicomputer based controllers. By far the largest number of currently used systems are in the 'mildly smart' device catagory. These systems tend to be reliable and of low first cost. On the other hand, US robotic system manufacturers have concentrated on the advanced systems, which typically are extremely expensive, highly versatile and programmable, and less reliable than desirable. There is a large growing market for less expensive, more specialized, yet programmable, 'mildly smart' mechanical manipulators. Keeping into view the above goals, the design problem statement follows.

3.1 Problem Statement

It is required to design a high speed, programmable mechanical manipulator. In particular, this manipulator must be versatile enough to perform two functions; a) path generation, and b) pick and place. In the path generation mode (see figure 3), the need to cut and/or bond in the plane is expected. That is, the fabric will move by the robot station at a constant speed and we wish to use the arm to trace out a prescribed path on the fabric. It may be assumed that there is no force feedback between the arm and the fabric. It is important for the path to be generated as accurately as possible and for V to be as high as possible. The path function is known but may change from day to day. In the pick & place mode (see figure 4), it is expected to use the robot as a part of a quality control system. That is, discrete products will come along belt 1 and be monitored, accepted or rejected. If a rejection signal is given to the arm, it must be able to locate, grasp, pick-up and discard the rejected item before it leaves its work area. Assume that the computer controlling the arm can also dynamically control the speed of the conveyor. Finally,

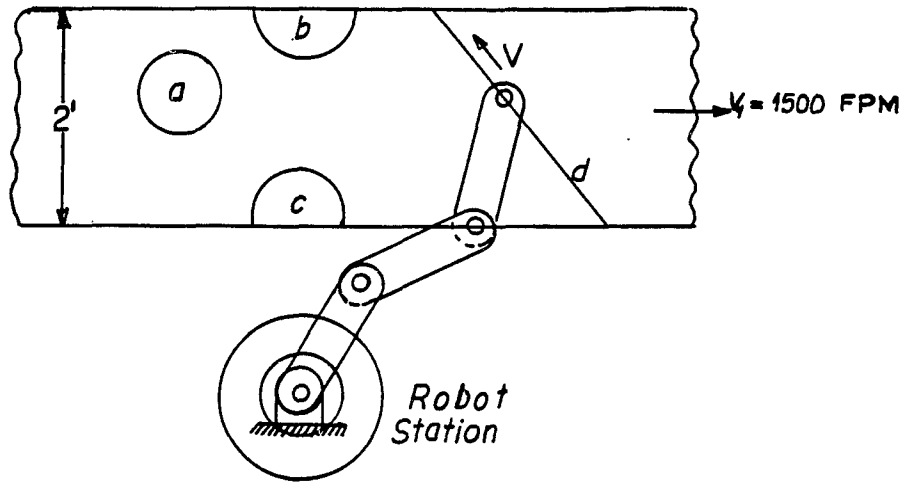


Figure 3 : Path Generation Application

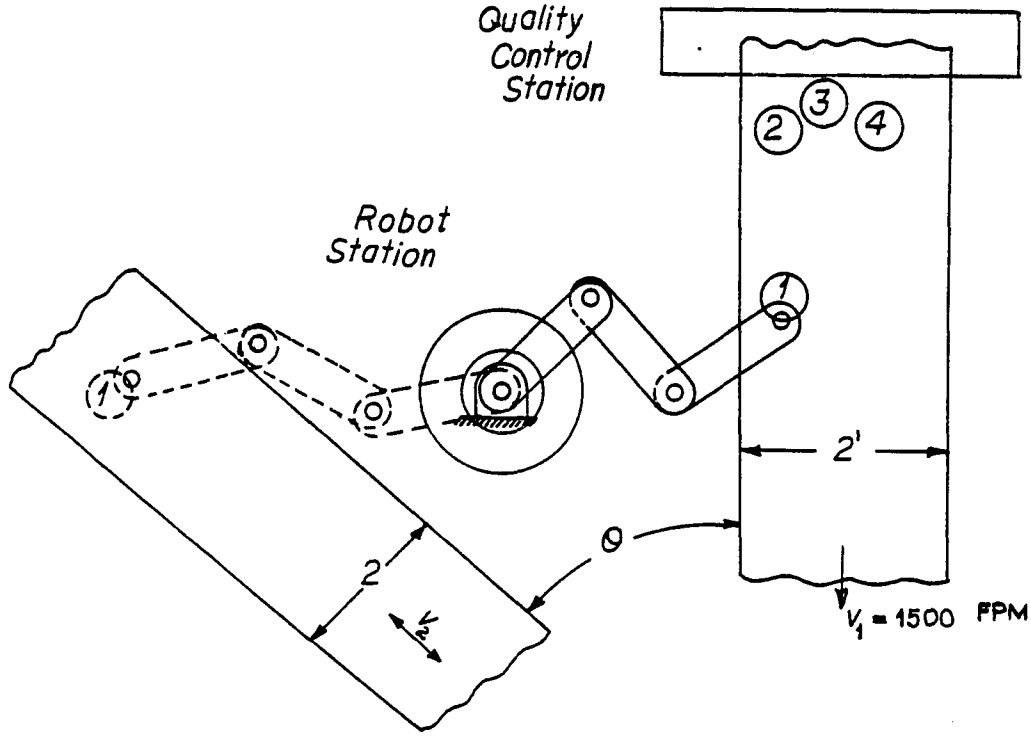


Figure 4 : Pick & Place Application

consider placing the rejects in a) a bin or b) on another conveyor moving at speed V_2 and at an angle θ to the product conveyor.

Develop a design strategy which should include a complete model construction and testing program for the development of the manipulator system, including the arm, control system, and computer hardware/software system. Estimate the actuator characteristics, especially, torque generation ability at the speeds specified. Also, examine in some detail arm dynamics.

3.2 Design Approach

What is a robot design? Designing a robot means to develop the robot model and try to analyze and synthesize the model under an environment of real time application. The first step is the selection of the manipulator configuration and the specification of design data. The second and most important step is deriving the differential equations of motion. Either the classical Lagrangian dynamics or the Newton-Euler method may be used for this purpose. The third stage is kinematic 'motion' analysis of the robot. It should primarily include position, velocity and acceleration characteristics of the centers of gravity of all links as well as that of the end effector point. Dynamic behavior of the manipulator components is analyzed as the fourth step.

Vibration analysis, stress analysis of various links and joints, and force analysis of the gripper are all included during this phase. Some general rules for the structural design of robots are depicted by Burckhardt and Helms [7].

While designing ROBOPT, RRR manipulator configuration (three Revolute joints) was selected in order to provide flexibility for handling complex path generation applications. Selecting a particular manipulator configuration is a qualitative aspect of the design process and no reasoning is 'complete' to prove the validity of such selection. Eventhough, some guide lines for the selection process are provided in the next section. ROBOPT synthesis process for the path generation application involves an ingenious design approach which is described now. Usually in a classical mechanism synthesis problem, link lengths are optimized. For example, in the well-known Tomas' coupler curve problem (a four bar linkage design problem for path generation), lengths of various links and link segments are selected as the design variables for the optimization problem. In the case of ROBOPT, this is not the case. Link lengths are assumed to be constant throughout the synthesis process. Refer to figure 5. To trace a path by the end effector point P, all that is required to be controlled is the torques (T_{12}, T_{23}, T_{34}) at joints J1, J2, and J3 at

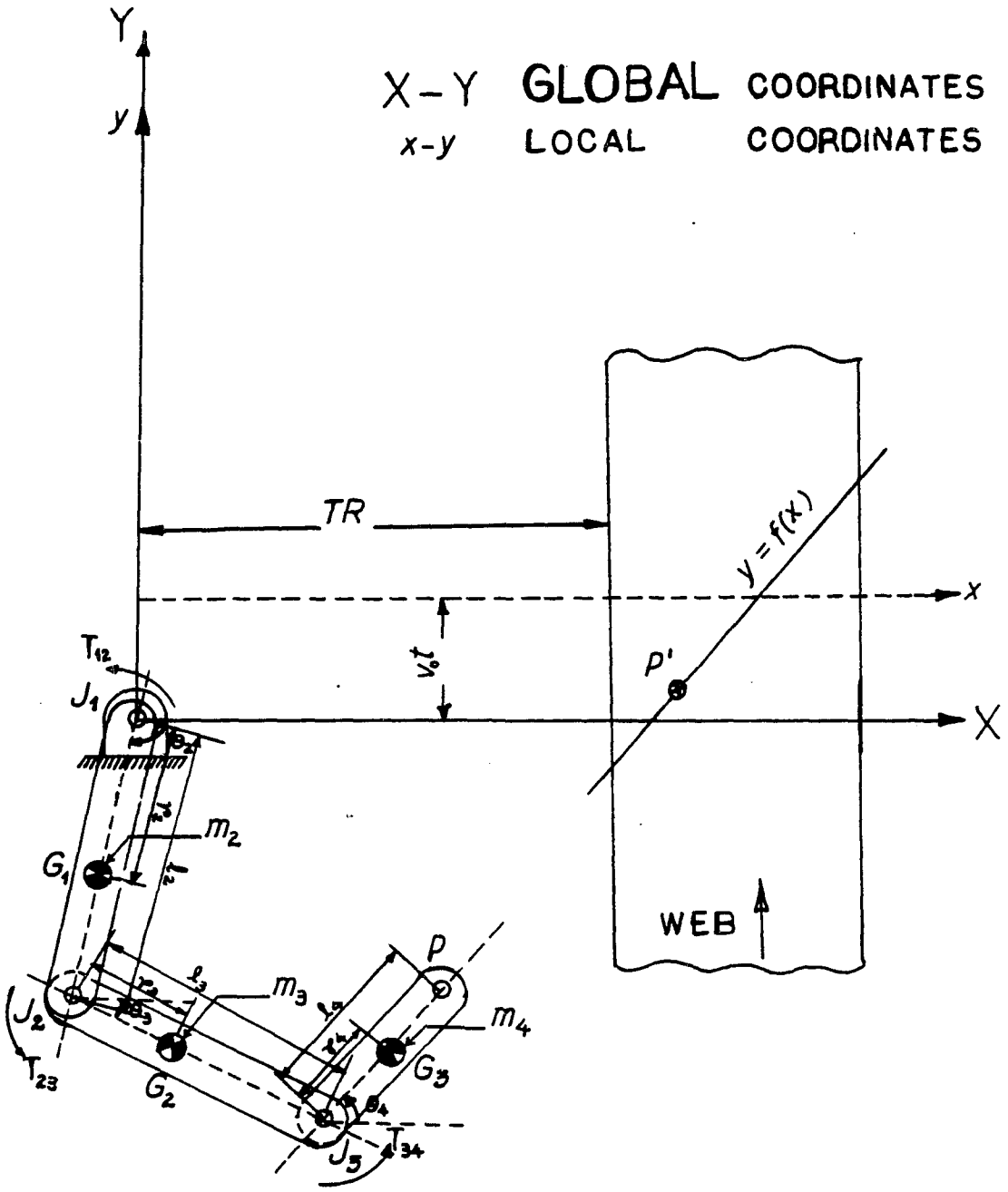


Figure 5 : Manipulator Arm

different instances of time. The torques as functions of time are called torque histories. If we supply these precalculated torque histories to all three servo motors, the required path will be generated. The torques are expressed as a Fourier series summation. The main three reasons for selecting these Fourier series summations are : (i) complex curves may be represented by a very few design variables, (ii) the Fourier series can also handle jump discontinuities in the torques, and (iii) the Fourier series convergence is fast. The constants of the Fourier series are selected as the design variables for the optimization problem. The best design variables are those that will minimize the objective function (minimization of the error of path generation). The optimized design variables offer optimum torque models. Thus, this is a TORQUE MODEL SYNTHESIS PROBLEM. The optimization procedure is described at length in the next chapter.

3.3 Manipulator Configuration Selection

' The arrangement or organization of links and joints, with a specification of their types, in a kinematic chain is called its KINEMATIC STRUCTURE '. In terms of manipulators, the kinamatic structure is called the configuration. The factors affecting manipulator configuration are : (1) Work-space , that is, total volume

swept. (2) Positional reachability, that is, interior and exterior approachability, possibility of wrap around an object, etc. (3) Controllability of orientation. For many robots the first axis is chosen as the gravity axis (vertical) because when the arm is rotated around this axis, the end effector orientation or attitude is not changed. Rotation around an axis other than vertical needs compensation of orientation when the position is changed. (4) Coordinate system selected. For a three link robot, this could be cartesian (all three translatory) , cylindrical (one rotary and two translatory joints) , polar (two rotary and one translatory joints) or prosthetic (all three rotary joints) (5) Special path generation requirements. For straight line continuous path generation, prismatic joints are better. For curvilinear paths, the revolute joint should be selected. (6) Actuator availability, suitability at a particular joint. (7) Links interferences. (8) Speed of actuation. (9) Load capacity.

For a three link spatial robot, the working area is calculated for different coordinate systems under the following assumptions. (i) All three links are of the same length ('L'). (ii) Translating link can move from 0. to 'L' length. (iii) Rotational joint can rotate through 360 degrees. (iv) Axes are arranged such as to obtain maximum

area in each construction. The results are summarized in table 1. A rotational joint is absolutely dominant compared to a translational one when work space is considered.

Kinematic complexity or 'dexterity' of a robot is governed by the number of rotational joints in the robot coordinate system, called RANK of the robot. KINEMATIC CLASSIFICATION of robots based on their DOF and RANK is tabulated as per table 2. Positional synthesis of a manipulator, that is, the solution for joint variables from the given position and orientation of the end effector, is a complex task. This solution is dependent on the rank of a robot. When homogeneous transformation equations are used for the robot analysis, each rotational degree of freedom makes the string of matrix multiplications and hence the final solution more complex. In case of cartesian robots, the differential equations of motion become more and more coupled. In general, when two or more, same degree of freedom robots are compared, the ascending order of their complexity is cartesian, cylindrical, polar, and prosthetic. Thus, kinematically, cartesian robots are the simplest robots, whereas, prosthetic arms with all rotational degrees of freedom are the complex most robots. ROBOPT is a three degrees of freedom robot with all three rotational coordinates. Hence, it is a rather complex robot with rank three. This configuration for ROBOPT is selected only due to

TABLE 1

Comparision of work-space for
Three Link Robots

TYPE OF JOINT / COORDINATE SYSTEM	ONE-DOF JOINT	TWO-DOF JOINT
Cartesian	$V = \ell^3 = (1.0)\ell^3$	—
Cylindrical	$V = 3\pi\ell^3 = (9.55)\ell^3$	—
Polar	$V = \frac{28}{3}\pi\ell^3 = (29.7)\ell^3$	$V = \frac{76}{3}\pi\ell^3 = (80.6)\ell^3$
Prosthetic	$V = 2\pi(\frac{8}{3}\pi + 3\sqrt{3})\ell^3 = (87.2)\ell^3$	$V = 36\pi\ell^3 = (114.6)\ell^3$

ℓ = *Link Lengths*

V = *Volume Swept*

TABLE 2

Kinematic Classification of Robots

DOF / RANK	1	2	3	4	5	6	7 and OVER
0	C			RR	RR	RR	RR
1	CY				RR	RR	RR
2	L	PR	P			RR	RR
3	L	L	ROBOT				RR
4	L	L	L				RR
5	L	L	L	L			RR
6	L	L	L	L	L		RR
7 and OVER	L	L	L	L	L	L	RR

C ⇨ Cartesian robot
CY ⇨ CYlindrical robot
P ⇨ Polar robot
PR ⇨ PRosthetic robot
RR ⇨ Redundant Robot
L ⇨ mechanically constrained Linkage

specific requirements.

Provided that all degrees of freedom of a robot are used effectively, there exists a definite relationship between the rank of a robot and the degrees of freedom of a robot for each type of coordinates. These relationships are shown in table 3. The kinematic classification in table 2 is based on these relationships.

A manipulator motion can be considered as either JOINT MOTION or as CARTESIAN MOTION. In joint motion calculations, motion variables are represented by joint coordinates. In cartesian motion, all position and orientation calculations are carried out in terms of a cartesian world coordinate system. Thus cartesian motion is natural to cartesian coordinates with the manipulator moving along straight lines and rotating about fixed axes in space. The extension of the cartesian motion to cylindrical, spherical, and other orthogonal coordinate system is simple. As in joint coordinate motion, trajectory segments are defined between positions described by homogeneous transform expressions. The difference between the cartesian motion and the joint motion is that, in the former, the motion is natural to the cartesian coordinate system, while in the latter the motion is linear in joint coordinates. The advantage of the cartesian motion as compared to the joint motion is that the motion between trajectory segment end

TABLE 3

Relationship between
Rank 'm' and DOF 'n'

DEXTERITY / ROBOT	RELATIONSHIP
Cartesian	$m = n-3$
Cylindrical	$m = n-2$
Polar	$m = n-1$
Prosthetic	$m = n$

m = Rank
n = DOF

points is well defined and thus is particularly suited to the initial and final trajectory segments. However, cartesian motion has a number of disadvantages. It involves the continuous evaluation of the manipulator set point and its subsequent transformation into joint coordinates. For motion in joint coordinates it is only necessary to interpolate, in joint coordinates, between segment end points. This involves roughly 1% of the computation necessary for cartesian motion. However, for functionally defined motions, such transformations are necessary. A second disadvantage is that cartesian motion breaks down whenever the manipulator becomes degenerate. Joint rates are unbounded in cartesian motion, becoming infinite as a manipulator moves into degeneracy [27, 31]. For example, in case of ROBOPT, then it is commanded to reach a point outside its workspace, the positional synthesis problem fails. In such a case, either the differential equations of motion are unsolvable or gives highly out of range joint variables.

ROBOPT is a planar manipulator. MINIMAL MANIPULATOR CONFIGURATION is assumed for this special-purpose robot. That is, only 3 degrees of freedom to position the end effector are used in design. Constant gripper orientation is assumed. One advantage of RRR configuration (three Revolute joints) of ROBOPT design is that it can trace any complex

shape within the work space. For high speed manipulations, light weight servo motors can be mounted right on the revolute joint axes. In path generation applications, the path is traced on the horizontal x-y plane. Z-axis, that is, height of the plane is adjustable in order to match conveyor plane.

Dr. R. Paul [30] reiterates in one of his articles, " Although the manipulator joint coordinates fully specify the position and orientation of the manipulator's end effector, they are unsuitable as a working coordinate system because generally they are 'not orthogonal' nor do they separate position from orientation ".

CHAPTER 4

DESIGN

In the last chapter we discussed the design problem, the design approach and the selected manipulator configuration for ROBOPT. Here, we expand the theory of ROBOPT in terms of both analysis and synthesis. When we consider the transformation of coordinates, the analysis problem deals with the direct problem, that is, given joint coordinates, find out the position and orientation of the end effector in the so called 'world' or base coordinate system. Whereas, Synthesis or the 'real design' phase solves the indirect problem. In the indirect problem the end effector position or a trace of positions is given and it is required to find out corresponding joint coordinates. Synthesis is the fifth and last stage of design and it is the most difficult stage. Section 4.1 describes theoretical derivations pertaining to ROBOPT analysis. Optimal synthesis technique is included in the next section.

4.1 Analysis

The use of Newton-Euler method to derive the equations of motion for a system of coupled rigid bodies can become very tedious. The task of deriving these equations is greatly simplified in many cases by the use of Lagrange's method. Separate freebody diagrams are required for each component of the system, and the forces of interaction have to be eliminated to arrive at the final set of equations. Use of the principle of virtual displacements eliminates the necessity of using interaction forces directly. Although the principle of virtual displacements can be extended to permit the derivation of equations of motion of MDOF systems, it is far simpler to use Lagrange's energy approach for this purpose. This permits the use of scalar quantities, work and kinetic energy, instead of the vector quantities, force and displacement. The only disadvantage of the Lagrangian formulation of differential equations of motion is its computational inefficiency. In Newton's method, the amount of computation increases linearly with the number of links whereas the conventional method based on the Lagrangian formulation increases as a quartic of the number of links [32]. We start with the derivation of the differential equations of motion using the classical approach of Lagrangian dynamics.

4.1.1 Differential Equations of Motion

The differential equations of motion can be formed by a set of Lagrange's equations,

$$\frac{d}{dt}\frac{dT}{d\dot{q}_i} - \frac{\partial T}{\partial q_i} + \frac{\partial V}{\partial q_i} = Q_i \qquad , i = 1,2,\ldots,N$$

where,

T = System Kinetic Energy

V = System Potential Energy

q_i = i-th generalized coordinate

$\dot{q}_i$ = Time derivative of i-th generalized coordinate

Q_i = Generalized force/torque assocoated with i-th coordinate

N = Number of degrees of freedom of the system

For a planar manipulator moving in a horizontal plane, the potential energy differential dV=0. Therefore, Lagrange's equations reduce to

$$\frac{d}{dt}\frac{dT}{d\dot{q}_i} - \frac{\partial T}{\partial q_i} = Q_i \qquad , i = 1,2,\ldots,N$$

ROBOPT is a three DOF manipulator with all three revolute joints. Hence q_i refers to angular displacements θ_i and Q_i referes to torques. Also, N=3. Support is considered to be component 1 of the system, hence, i=2,3 and 4 respectively for the three links of the robot.

Consider a generic link (i) as shown in figure 6. Symbolically, three differential equations of motion corresponding to three degrees of freedom can be rewritten as :

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_2} - \frac{\partial T}{\partial \theta_2} = Q_2 \qquad \ldots(4.1)$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_3} - \frac{\partial T}{\partial \theta_3} = Q_3 \qquad \ldots(4.2)$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_4} - \frac{\partial T}{\partial \theta_4} = Q_4 \qquad \ldots(4.3)$$

Generalized torques Q_2, Q_3, and Q_4 can be obtained by applying the principle of virtual displacements.

$$\delta W = Q_2 \delta\theta_2 + Q_3 \delta\theta_3 + Q_4 \delta\theta_4 \qquad \ldots(4.4)$$

Also,

$$\delta W = T_{12}\delta\theta_2 + T_{23}\delta\phi_3 + T_{34}\delta\phi_4$$

$$= T_{12}\delta\theta_2 + T_{23}(\delta\theta_3 - \delta\theta_2) + T_{34}(\delta\theta_4 - \delta\theta_3)$$

$$\therefore \quad \delta W = (T_{12} - T_{23})\delta\theta_2 + (T_{23} - T_{34})\delta\theta_3 + (T_{34})\delta\theta_4 \qquad \ldots(4.5)$$

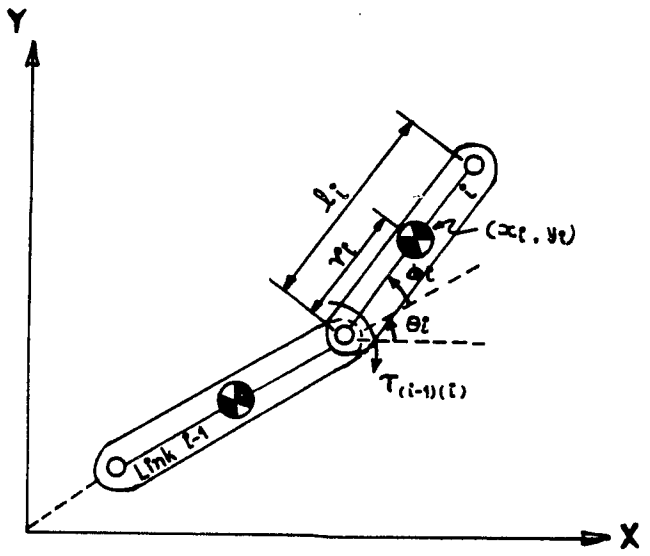
Comparing (4.4) and (4.5),

$$Q_2 = T_{12} - T_{23} \qquad \ldots(4.6)$$

$$Q_3 = T_{23} - T_{34} \qquad \ldots(4.7)$$

$$Q_4 = T_{34} \qquad \ldots(4.8)$$

Positions of centers of gravity of links are :



(x_i , y_i)	=	Coordinates of Center of Gravity
m_i	=	Mass of the Link
I_i	=	Moment of Inertia
ℓ_i	=	Length of the Link
r_i	=	Distance of C.G. of the Link
θ_i	=	Absolute Angle
ϕ_i	=	Relative Angle, $\phi_i = \theta_i - \theta_{i-1}$
$T_{(i-1)(i)}$	=	Torque acting on Link i

Figure 6 : Generic Link (i)

$$x_2 = r_2 \cos\theta_2 \qquad \ldots(4.9)$$

$$y_2 = r_2 \sin\theta_2 \qquad \ldots(4.10)$$

$$x_3 = L_2 \cos\theta_2 + r_3 \cos\theta_3 \qquad \ldots(4.11)$$

$$y_3 = L_2 \sin\theta_2 + r_3 \sin\theta_3 \qquad \ldots(4.12)$$

$$x_4 = L_2 \cos\theta_2 + L_3 \cos\theta_3 + r_4 \cos\theta_4 \qquad \ldots(4.13)$$

$$y_4 = L_2 \sin\theta_2 + L_3 \sin\theta_3 + r_4 \sin\theta_4 \qquad \ldots(4.14)$$

Differentiate (4.9) through (4.14) with respect to time to obtain velocity components

$$\dot{x}_2 = -r_2 \dot{\theta}_2 \sin\theta_2 \qquad \ldots(4.15)$$

$$\dot{y}_2 = r_2 \dot{\theta}_2 \cos\theta_2 \qquad \ldots(4.16)$$

$$\dot{x}_3 = -L_2 \dot{\theta}_2 \sin\theta_2 - r_3 \dot{\theta}_3 \sin\theta_3 \qquad \ldots(4.17)$$

$$\dot{y}_3 = L_2 \dot{\theta}_2 \cos\theta_2 + r_3 \dot{\theta}_3 \cos\theta_3 \qquad \ldots(4.18)$$

$$\dot{x}_4 = -L_2 \dot{\theta}_2 \sin\theta_2 - L_3 \dot{\theta}_3 \sin\theta_3 - r_4 \dot{\theta}_4 \sin\theta_4 \qquad \ldots(4.19)$$

$$\dot{y}_4 = L_2 \dot{\theta}_2 \cos\theta_2 + L_3 \dot{\theta}_3 \cos\theta_3 + r_4 \dot{\theta}_4 \cos\theta_4 \qquad \ldots(4.20)$$

The velocities of centers of mass are :

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

$$= (-r_2 \dot{\theta}_2 \sin\theta_2)^2 + (r_2 \dot{\theta}_2 \cos\theta_2)^2$$

$$= r_2^2 \dot{\theta}_2^2 \qquad \ldots(4.21)$$

$$v_3^2 = \dot{x}_3^2 + \dot{y}_3^2$$

$$= (-L_2 \dot{\theta}_2 \sin\theta_2 - r_3 \dot{\theta}_3 \sin\theta_3)^2 + (L_2 \dot{\theta}_2 \cos\theta_2 + r_3 \dot{\theta}_3 \cos\theta_3)^2$$

$$= L_2^2 \dot{\theta}_2^2 + r_3^2 \dot{\theta}_3^2 + 2 L_2 r_3 \dot{\theta}_2 \dot{\theta}_3 \cos(\theta_3 - \theta_2) \qquad \ldots(4.22)$$

$$v_4^2 = \dot{x}_4^2 + \dot{y}_4^2$$

$$= (-L_2 \dot{\theta}_2 \sin\theta_2 - L_3 \dot{\theta}_3 \sin\theta_3 - r_4 \dot{\theta}_4 \sin\theta_4)^2$$

$$+ (L_2 \dot{\theta}_2 \cos\theta_2 + L_3 \dot{\theta}_3 \cos\theta_3 + r_4 \dot{\theta}_4 \cos\theta_4)^2$$

$$= L_2^2\dot{\theta}_2^2 + L_3^2\dot{\theta}_3^2 + r_4^2\dot{\theta}_4^2 + 2L_2L_3\dot{\theta}_2\dot{\theta}_3\cos(\theta_3-\theta_2) + 2L_2r_4\dot{\theta}_2\dot{\theta}_4\cos(\theta_4-\theta_2) + 2L_3r_4\dot{\theta}_3\dot{\theta}_4\cos(\theta_4-\theta_3) \quad \ldots(4.23)$$

Total kinetic energy of the system is given by

$$T = (1/2)[m_2v_2^2 + I_2\dot{\theta}_2^2 + m_3v_3^2 + I_4\dot{\theta}_4^2 + m_4v_4^2 + I_3\dot{\theta}_3^2] \quad \ldots(4.24)$$

Substitute (4.21) through (4.23) into (4.24) to get

$$T = (1/2)\{m_2r_2^2\dot{\theta}_2^2 + I_2\dot{\theta}_2^2 + I_3\dot{\theta}_3^2 + I_4\dot{\theta}_4^2 + m_3[L_2^2\dot{\theta}_2^2 + r_3^2\dot{\theta}_3^2 + 2L_2r_3\dot{\theta}_2\dot{\theta}_3\cos(\theta_3-\theta_2)] + m_4[L_2^2\dot{\theta}_2^2 + L_3^2\dot{\theta}_3^2 + r_4^2\dot{\theta}_4^2 + 2L_2L_3\dot{\theta}_2\dot{\theta}_3\cos(\theta_3\theta_2) + 2L_2r_4\dot{\theta}_2\dot{\theta}_4\cos(\theta_4-\theta_2) + 2L_3r_4\dot{\theta}_3\dot{\theta}_4\cos(\theta_4-\theta_3)]\} \quad \ldots(4.25)$$

$$\frac{dT}{d\dot{\theta}_2} = m_2r_2^2\dot{\theta}_2 + I_2\dot{\theta}_2 + m_3L_2^2\dot{\theta}_2 + m_3L_2r_3\dot{\theta}_3\cos(\theta_3-\theta_2) + m_4L_2^2\dot{\theta}_2 + m_4L_2L_3\dot{\theta}_3\cos(\theta_3-\theta_2) + m_4L_2r_4\dot{\theta}_4\cos(\theta_4-\theta_2)$$

$$= [m_2r_2^2 + (m_3+m_4)L_2^2 + I_2]\dot{\theta}_2 + L_2(m_3r_3+m_4L_3)\dot{\theta}_3\cos(\theta_3-\theta_2) + m_4L_2r_4\dot{\theta}_4\cos(\theta_4-\theta_2)$$

$$\frac{d}{dt}\frac{dT}{d\dot{\theta}_2} = [m_2r_2^2 + (m_3+m_4)L_2^2 + I_2]\ddot{\theta}_2 + L_2(m_3r_3+m_4L_3)\ddot{\theta}_3\cos(\theta_3-\theta_2) - L_2(m_3r_3+m_4L_3)\dot{\theta}_3(\dot{\theta}_3-\dot{\theta}_2)\sin(\theta_3-\theta_2) + m_4L_2r_4\ddot{\theta}_4\cos(\theta_4-\theta_2) - m_4L_2r_4\dot{\theta}_4(\dot{\theta}_4-\dot{\theta}_2)\sin(\theta_4-\theta_2)$$

$$\frac{dT}{d\theta_2} = m_3L_2r_3\dot{\theta}_2\dot{\theta}_3\sin(\theta_3-\theta_2) + m_4L_2L_3\dot{\theta}_2\dot{\theta}_3\sin(\theta_3-\theta_2) + m_4L_2r_4\dot{\theta}_2\dot{\theta}_4\sin(\theta_4-\theta_2)$$

$$= L_2(m_3r_3+m_4L_3)\dot{\theta}_2\dot{\theta}_3\sin(\theta_3-\theta_2) + m\,L_2r_4\dot{\theta}_2\dot{\theta}_4\sin(\theta_4-\theta_2)$$

$$\left[\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_2} - \frac{\partial T}{\partial \theta_2}\right] = [m_2 r_2^2 + (m_3+m_4)L^2 + I_2]\ddot{\theta}_2 + m_4 L_2 r_4 \cos(\theta_4-\theta_2)\ddot{\theta}_4$$
$$+ L_2(m_3 r_3 + m_4 L_3)\cos(\theta_3-\theta_2)\ddot{\theta}_3$$
$$- L_2(m_3 r_3 + m_4 L_3)\sin(\theta_3-\theta_2)\dot{\theta}_3^2$$
$$- m_4 L_2 r_4 \sin(\theta_4-\theta_2)\dot{\theta}_4^2 \qquad \ldots(4.26)$$

Similarly, for θ_3 and θ_4 degrees of freedom,

$$\frac{dT}{d\theta_3} = m_3 r_3^2 \dot{\theta}_3 + m_3 L_3 r_3 \dot{\theta}_3 \cos(\theta_3-\theta_2) + I_3\dot{\theta}_3 + m_4 L_3^2 \dot{\theta}_3$$
$$+ m_4 L_2 L_3 \dot{\theta}_2 \cos(\theta_3-\theta_2) + m_4 L_3 r_4 \dot{\theta}_4 \cos(\theta_4-\theta_3)$$
$$= (m_3 r_3^2 + m_4 L_3^2 + I_3)\dot{\theta}_3 + L_2(m_3 r_3 + m_4 L_2)\dot{\theta}_2 \cos(\theta_3-\theta_2)$$
$$+ m_4 L_3 r_4 \dot{\theta}_4 \cos(\theta_4-\theta_3)$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_3} = (m_3 r_3^2 + m_4 L_3^2 + I_3)\ddot{\theta}_3 + L_2(m_3 r_3 + m_4 L_3)\cos(\theta_3-\theta_2)\ddot{\theta}_2$$
$$- L_2(m_3 r_3 + m_4 L_3)\dot{\theta}_2(\dot{\theta}_3-\dot{\theta}_2)\sin(\theta_3-\theta_2)$$
$$+ m_4 L_4 r_4 \cos(\theta_4-\theta_3)\ddot{\theta}_4 - m_4 L_3 r_4 \dot{\theta}_4(\dot{\theta}_4-\dot{\theta}_3)\sin(\theta_4-\theta_3)$$

$$\frac{\partial T}{\partial \theta_3} = -m_3 L_2 r_3 \dot{\theta}_2 \dot{\theta}_3 \sin(\theta_3-\theta_2) - m_4 L_2 L_3 \dot{\theta}_2 \dot{\theta}_2 \sin(\theta_3-\theta_2)$$
$$+ m_4 L_3 r_4 \dot{\theta}_4 \dot{\theta}_3 \sin(\theta_4-\theta_3)$$
$$= -L_2(m_3 r_3 + m_4 L_3)\dot{\theta}_2 \dot{\theta}_3 \sin(\theta_3-\theta_2) + m_4 L_3 r_4 \dot{\theta}_4 \dot{\theta}_3 \sin(\theta_4-\theta_3)$$

$\therefore$

$$\left[\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_3} - \frac{\partial T}{\partial \theta_3}\right] = (m_3 r_3^2 + m_4 L_3^2 + I_3)\ddot{\theta}_3 + L_2(m_3 r_3 + m_4 L_2)\cos(\theta_3-\theta_2)\ddot{\theta}_2$$
$$+ m_4 L_3 r_4 \cos(\theta_4-\theta_3)\ddot{\theta}_4 - m_4 L_3 r_4 \sin(\theta_4-\theta_3)\dot{\theta}_4^2$$
$$+ L_2(m_3 r_3 + m_4 L_3)\sin(\theta_3-\theta_2)\dot{\theta}_2^2 \qquad \ldots(4.27)$$

$$\frac{\partial T}{\partial \theta_4} = m_4 r_4 \dot{\theta}_4 + I_4 \dot{\theta}_4 + m_4 L_2 r_4 \dot{\theta}_2 \cos(\theta_4-\theta_2) + m_4 L_3 r_4 \dot{\theta}_3 \cos(\theta_4-\theta_3)$$

$$\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_4} = (m_4 r_4^2 + I_4)\ddot{\theta}_4 + m_4 L_2 r_4 \ddot{\theta}_2 \cos(\theta_4 - \theta_2)$$
$$- m_4 L_2 r_4 \dot{\theta}_2 (\dot{\theta}_4 - \dot{\theta}_2)\sin(\theta_4 - \theta_2) + m_4 L_3 r_4 \ddot{\theta}_3 \cos(\theta_4 - \theta_3)$$
$$- m_4 L_3 r_4 \dot{\theta}_3 (\dot{\theta}_4 - \dot{\theta}_3)\sin(\theta_4 - \theta_3)$$

$$\frac{\partial T}{\partial \theta_4} = -m_4 L_2 r_4 \dot{\theta}_2 \dot{\theta}_4 \sin(\theta_4 - \theta_2) - m_4 L_3 r_4 \dot{\theta}_3 \dot{\theta}_4 \sin(\theta_4 - \theta_3)$$

$$\left[\frac{d}{dt}\frac{\partial T}{\partial \dot{\theta}_4} - \frac{\partial T}{\partial \theta_4}\right] = (m_4 r_4^2 + I_4)\ddot{\theta}_4 + m_4 L_2 r_4 \cos(\theta_4 - \theta_2)\ddot{\theta}_2$$
$$+ m_4 L_3 r_4 \cos(\theta_4 - \theta_3)\ddot{\theta}_3 + m_4 L_2 r_4 \sin(\theta_4 - \theta_3)\dot{\theta}_2^2$$
$$+ m_4 L_3 r_4 \sin(\theta_4 - \theta_3)\dot{\theta}_3^2 \qquad \ldots(4.28)$$

Substitute (4.6) through (4.8) and (4.26) through (4.28) into Lagrange's three equations (4.1), (4.2), and (4.3)

$$[m_2 r_2^2 + (m_3 + m_4)L_2^2 + I_2]\ddot{\theta}_2 + L_2(m_3 r_3 + m_4 L_3)\cos(\theta_3 - \theta_2)\ddot{\theta}_3$$
$$+ m_4 L_2 r_4 \cos(\theta_4 - \theta_2)\ddot{\theta}_4 - L_2(m_3 r_3 + m_4 L_3)\sin(\theta_3 - \theta_2)\dot{\theta}_3^2$$
$$- m_4 L_2 r_4 \sin(\theta_4 - \theta_2)\dot{\theta}_4^2 = T_{12} - T_{23} \qquad \ldots(4.29)$$

$$[m_3 r_3^2 + m_4 L_3^2 + I_3]\ddot{\theta}_3 + L_2(m_3 r_3 + m_4 L_3)\cos(\theta_3 - \theta_2)\ddot{\theta}_2$$
$$+ m_4 L_2 r_4 \cos(\theta_4 - \theta_3)\ddot{\theta}_4 + L_2(m_3 r_3 + m_4 L_3)\sin(\theta_3 - \theta_2)\dot{\theta}_2^2$$
$$- m_4 L_3 r_4 \sin(\theta_4 - \theta_3)\dot{\theta}_4^2 = T_{23} - T_{34} \qquad \ldots(4.30)$$

$$[m_4 r_4^2 + I_4]\ddot{\theta}_4 + m_4 L_2 r_4 \cos(\theta_4 - \theta_2)\ddot{\theta}_2 + m_4 L_3 r_4 \cos(\theta_4 - \theta_2)\dot{\theta}_2^2$$
$$+ m_4 L_2 r_4 \sin(\theta_4 - \theta_2)\dot{\theta}_2^2 + m_4 L_3 r_4 \sin(\theta_4 - \theta_3)\dot{\theta}_3^2 = T_{34} \qquad \ldots(4.31)$$

Defining constants and rearranging,

Let,

$$a_1 = I_2 + m_2 r_2^2 + (m_3 + m_4)L_2^2$$
$$a_2 = (m_3 r_3 + m_4 L_3)L_2$$
$$a_3 = m_4 L_2 r_4$$

$$a_4 = I_3 + m_3 r_3^2 + m_4 L_3^2$$

$$a_5 = m_4 L_3 r_4$$

$$a_6 = I_4 + m_4 r_4^2$$

$$C_{32} = COS(\theta_3 - \theta_2)$$

$$C_{42} = COS(\theta_4 - \theta_2)$$

$$C_{43} = COS(\theta_4 - \theta_3)$$

$$S_{32} = SIN(\theta_3 - \theta_2)$$

$$S_{42} = SIN(\theta_4 - \theta_2)$$

$$S_{43} = SIN(\theta_4 - \theta_3)$$

Then,

ROBOPT differential equations of motion are :

$$\begin{cases} a_1\ddot{\theta}_2 + a_2 C_{32}\ddot{\theta}_3 + a_3 C_{42}\ddot{\theta}_4 = a_2 S_{32}\dot{\theta}_3^2 + a_3 S_{42}\dot{\theta}_4^2 + T_{12} - T_{23} & \ldots(4.32) \\ a_2 C_{32}\ddot{\theta}_2 + a_4\ddot{\theta}_3 + a_5 C_{43}\ddot{\theta}_4 = -a_2 S_{32}\dot{\theta}_2^2 + a_5 S_{43}\dot{\theta}_4^2 + T_{23} - T_{34} & \ldots(4.33) \\ a_3 C_{42}\ddot{\theta}_2 + a_5 C_{43}\ddot{\theta}_3 + a_6\ddot{\theta}_4 = -a_3 S_{42}\dot{\theta}_2^2 - a_3 S_{42}\dot{\theta}_2^2 + T_{34} & \ldots(4.34) \end{cases}$$

4.1.2 Kinematics

Kinematics is the study of motion. ROBOPT motion is characterized by position, velocity and acceleration analysis of its joints, centers of gravity of links and the end effector point. Equations are also included for relative motion of the end effector with respect to a conveyor belt moving at constant velocity V . A better visualization of overall setup and motions of ROBOPT linkages and conveyor belt could be obtained by referring to figure 5.

Joints . The position, velocity and acceleration of joints 1,2 and 3 are represented by PJ1,VJ1,AJ1 ; PJ2,VJ2,AJ2 and PJ3,VJ3 and AJ3 respectively. Also note that joint-1 is located at the origin of the world coordinate system.

Joint-1,

$$XJ1 = 0.0$$

$$YJ1 = 0.0$$

$$PJ1 = (XJ1, YJ1) = (0.0, 0.0) \qquad ...(4.35)$$

$$\dot{XJ1} = 0.0$$

$$\dot{YJ1} = 0.0$$

$$VJ1 = \sqrt{\dot{XJ1}^2 + \dot{YJ1}^2} = 0.0 \qquad ...(4.36)$$

$$\ddot{XJ1} = 0.0$$

$$\ddot{YJ1} = 0.0$$

$$AJ1 = \sqrt{\ddot{XJ1}^2 + \ddot{YJ1}^2} = 0.0 \qquad ...(4.37)$$

Joint-2,

$$XJ2 = L_2 \cos\theta_2$$

$$YJ2 = L_2 \sin\theta_2$$

$$PJ2 = (XJ2, YJ2) \qquad ...(4.38)$$

$$\dot{XJ2} = -L_2 \sin\theta_2 \dot{\theta}_2$$

$$\dot{YJ2} = L_2 \cos\theta_2 \dot{\theta}_2$$

$$VJ2 = \sqrt{\dot{XJ2}^2 + \dot{YJ2}^2} \qquad ...(4.39)$$

$$\ddot{XJ2} = -L_2 \cos\theta_2 \dot{\theta}_2^2 - L_2 \sin\theta_2 \ddot{\theta}_2$$

$$\ddot{YJ2} = -L_2 \sin\theta_2 \dot{\theta}_2^2 + L_2 \cos\theta_2 \ddot{\theta}_2$$

$$AJ2 = \sqrt{\ddot{XJ2}^2 + \ddot{YJ2}^2} \qquad ...(4.40)$$

Joint-3,

$$XJ3 = L_2 COS\theta_2 + L_3 COS\theta_3$$

$$YJ3 = L_2 SIN\theta_2 + L_3 SIN\theta_3$$

$$PJ3 = (XJ3, YJ3) \qquad ...(4.41)$$

$$\dot{XJ3} = -L_2 SIN\theta_2 \dot{\theta}_2 - L_3 SIN\theta_3 \dot{\theta}_3$$

$$\dot{YJ3} = L_2 COS\theta_2 \dot{\theta}_2 + L_3 COSO_3 \dot{\theta}_3$$

$$VJ3 = \sqrt{\dot{XJ3}^2 + \dot{YJ3}^2} \qquad ...(4.42)$$

$$\ddot{XJ3} = -L_2 COS\theta_2 \dot{\theta}_2^2 - L_2 SIN\theta_2 \ddot{\theta}_2 - L_3 COS\theta_3 \dot{\theta}_3^2 - L_3 SIN\theta_3 \ddot{\theta}_3$$

$$\ddot{YJ3} = -L_2 SIN\theta_2 \dot{\theta}_2^2 + L_2 COS\theta_2 \ddot{\theta}_2 - L_3 SIN\theta_3 \dot{\theta}_3^2 + L_3 COS\theta_3 \ddot{\theta}_3$$

$$AJ3 = \sqrt{\ddot{XJ3}^2 + \ddot{YJ3}^2} \qquad ...(4.43)$$

Centers of gravity of links. A similar analysis is carried out for the centers of gravity of three links. The distances of the C.G.'s are r_2, r_3, and r_4 for respective links. The position, velocity and acceleration are expressed by PG's, VG's and AG's respectively.

Center of Gravity-1,

$$XG1 = r_2 COS\theta_2$$

$$YG1 = r_2 SIN\theta_2$$

$$PG1 = (XG1, YG1) \qquad ...(4.43)$$

$$\dot{XG1} = -r_2 SIN\theta_2 \dot{\theta}_2$$

$$\dot{YG1} = r_2 COS\theta_2 \dot{\theta}_2$$

$$VG1 = \sqrt{\dot{XG1}^2 + \dot{YG1}^2} \qquad ...(4.44)$$

$$\ddot{XG1} = -r_2 COS\theta_2 \dot{\theta}_2^2 - r_2 SIN\theta_2 \ddot{\theta}_2$$

$$\ddot{YG1} = -r_2 SIN\theta_2 \dot{\theta}_2^2 + r_2 COS\theta_2 \ddot{\theta}_2$$

$$AG1 = \sqrt{\ddot{XG1}^2 + \ddot{YG1}^2} \qquad ...(4.45)$$

Center of gravity-2,

$$XG2 = L_2 COS\theta_2 + r_3 COS\theta_3$$

$$YG2 = L_2 SIN\theta_2 + r_3 SIN\theta_3$$

$$PG2 = (XG2, YG2) \qquad ...(4.46)$$

$$\dot{XG2} = -L_2 SIN\theta_2 \dot{\theta}_2 - r_3 SIN\theta_3 \dot{\theta}_4$$

$$\dot{YG2} = L_2 COS\theta_2 \dot{\theta}_2 + r_3 COS\theta_3 \dot{\theta}_3$$

$$VG2 = \sqrt{\dot{XG2}^2 + \dot{YG2}^2} \qquad ...(4.47)$$

$$\ddot{XG2} = -L_2 COS\theta_2 \dot{\theta}_2^2 - L_2 SIN\theta_2 \ddot{\theta}_2 - r_3 COS\theta_3 \dot{\theta}_3^2 - r_3 SIN\theta_3 \ddot{\theta}_3$$

$$\ddot{YG2} = -L_2 SIN\theta_2 \dot{\theta}_2^2 + L_2 COS\theta_2 \ddot{\theta}_2 - r_3 SIN\theta_3 \dot{\theta}_3^2 + r_3 COS\theta_3 \ddot{\theta}_3$$

$$AG2 = \sqrt{\ddot{XG2}^2 + \ddot{YG2}^2} \qquad ...(4.48)$$

Center of gravity-3,

$$XG3 = L_2 COS\theta_2 + L_3 COS\theta_3 + r_4 COS\theta_4$$

$$YG3 = L_2 SIN\theta_2 + L_3 SIN\theta_3 + r_4 SIN\theta_4$$

$$PG3 = (XG3, YG3) \qquad ...(4.49)$$

$$\dot{XG3} = -L_2 SIN\theta_2 \dot{\theta}_2 - L_3 SIN\theta_3 \dot{\theta}_3 - r_4 SIN\theta_4 \dot{\theta}_4$$

$$\dot{YG3} = L_2 COS\theta_2 \dot{\theta}_2 + L_3 COS\theta_3 \dot{\theta}_3 + r_4 COS\theta_4 \dot{\theta}_4$$

$$VG3 = \sqrt{\dot{XG3}^2 + \dot{YG3}^2} \qquad ...(4.50)$$

$$\ddot{XG3} = -L_2 COS\theta_2 \dot{\theta}_2^2 - L_2 SIN\theta_2 \ddot{\theta}_2 - L_3 COS\theta_3 \dot{\theta}_3^2 - L\ SIN\theta_3 \ddot{\theta}_3 - r_4 COS\theta_4 \dot{\theta}_4^2 - r_4 SIN\theta_4 \ddot{\theta}_4$$

$$\ddot{YG3} = -L_2 SIN\theta_2 \dot{\theta}_2^2 + L_2 COS\theta_2 \ddot{\theta}_2 - L_3 SIN\theta_3 \dot{\theta}_3^2 + L_3 COS\theta_3 \ddot{\theta}_3 - r_4 SIN\theta_4 \dot{\theta}_4^2 + r_4 COS\theta_4 \ddot{\theta}_4$$

$$AG3 = \sqrt{\ddot{XG3}^2 + \ddot{YG3}^2} \qquad ...(4.51)$$

End effector. In this situation, the end effector of the manipulator is moving relative to a conveyor located at distance TR. Web is moving at constant velocity v in a plane perpendicular to horizontal (X) axis. Point P absolute

motion is represented by notations PP, VP, AP, whereas, its relative position, velocity and acceleration are represented by PPR, VPR, APR respectively in the world (global) coordinate system. 't' shows the time elapsed from the start of the motions.

$$XP = L_2 COS\theta_2 + L_3 COS\theta_3 + L_4 COS\theta_4$$

$$YP = L_2 SIN\theta_2 + L_3 SIN\theta_3 + L_4 SIN\theta_4$$

$$XPR = XP$$

$$YPR = YP - v_0 t$$

$$PP = (XP, YP) \qquad \ldots(4.52)$$

$$PPR = (XPR, YPR) \qquad \ldots(4.53)$$

$$\dot{XP} = -L_2 SIN\theta_2 \dot{\theta}_2 - L_3 SIN\theta_3 \dot{\theta}_3 - L_4 SIN\theta_4 \dot{\theta}_4$$

$$\dot{YP} = L_2 COS\theta_2 \dot{\theta}_2 + L_3 COS\theta_3 \dot{\theta}_3 + L_4 COS\theta_4 \dot{\theta}_4$$

$$\dot{XPR} = \dot{XP}$$

$$\dot{YPR} = \dot{YP} - v_0$$

$$VP = \sqrt{\dot{XP}^2 + \dot{YP}^2} \qquad \ldots(4.54)$$

$$VPR = \sqrt{\dot{XPR}^2 + \dot{YPR}^2} \qquad \ldots(4.55)$$

$$\ddot{XP} = -L_2 COS\theta_2 \dot{\theta}_2^2 - L_2 SIN\theta_2 \ddot{\theta}_2 - L_3 COS\theta_3 \dot{\theta}_3^2 - L_3 SIN\theta_3 \ddot{\theta}_3 - L_4 COS\theta_4 \dot{\theta}_4^2 - L_4 SIN\theta_4 \ddot{\theta}_4$$

$$\ddot{YP} = -L_2 SIN\theta_2 \dot{\theta}_2^2 + L_2 COS\theta_2 \ddot{\theta}_2 - L_2 SIN\theta_2 \dot{\theta}_3^2 + L_3 COS\theta_3 \ddot{\theta}_3 - L_4 SIN\theta_4 \dot{\theta}_4^2 + L_4 COS\theta_4 \ddot{\theta}_4$$

$$\ddot{XPR} = \ddot{XP}$$

$$\ddot{YPR} = \ddot{YP}$$

$$AP = \sqrt{\ddot{XP}^2 + \ddot{YP}^2} \qquad \ldots(4.56)$$

$$APR = AP \qquad \ldots(4.57)$$

4.1.3 Dynamics

Dynamics deals with the analysis of forces/torques acting on the system and those resulting from the motion. Torques T_{12}, T_{23} and T_{34} are acting on the system at three joints due to the presence of servo motors. Forces and torques acting on the system at the end effector are either zero or assumed to be known. Joint frictions are negligible. System force analysis is carried out with the help of freebody diagrams as shown in figure 7.

Figure 7(a) shows the freebody diagram of the first link. External torques T_{12} and T_{32} act on the body at joints J1 and J2 respectively. F12, F23 and F34 are the forces produced at the joints J1, J2 and J3. They are also known as pin forces. The components of all the forces acting in horizontal and vertical directions are expressed by superscripts x and y respectively. Inertial torque $I_{G1}\ddot{\theta}_2$ and force m_2 (AG1) act at the center of gravity G1, in a direction opposite to the motion.

Applying d'Alembert's principle,

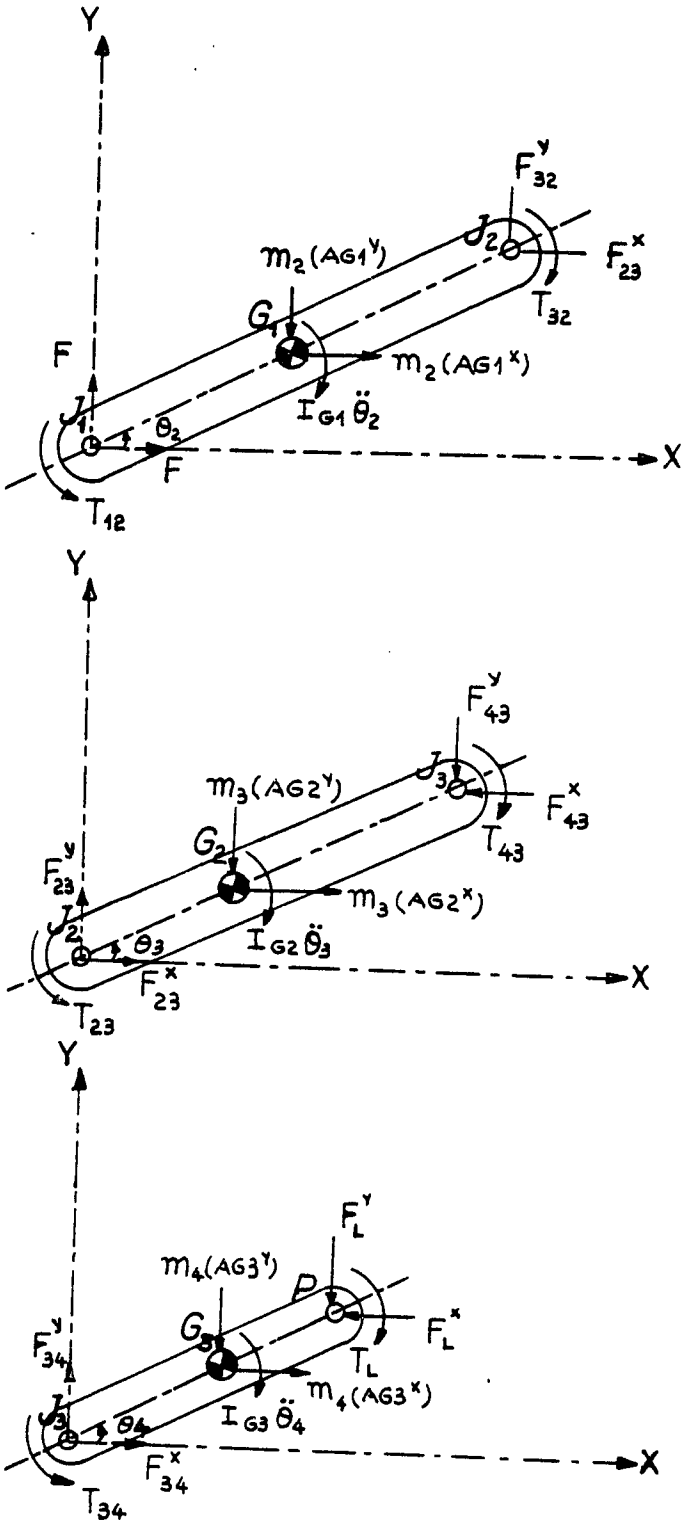


Figure 7 : Freebody Diagrams

Summation of moments about center of gravity,

$$\Sigma M_{G1} = 0$$

$$\longrightarrow T_{12} - T_{32} + F_{12}^{x} r_2 \sin\theta_2 - F_{12}^{y} r_2 \cos\theta_2 + F_{32}^{x} (L_2 - r_2) \sin\theta_2 - F_{32}^{y} (L_2 - r_2) \cos\theta_2 - I_{G1} \ddot{\theta}_2 = 0$$

$\therefore$

$$T_{12} = T_{32} - F_{12}^{x} r_2 \sin\theta_2 + F_{12}^{y} r_2 \cos\theta_2 - F_{32} (L_2 - r_2) \sin\theta_2 + F_{32}^{y} (L_2 - r_2) \cos\theta_2 + I_{G1} \ddot{\theta}_2 \quad \ldots(4.58)$$

Summation of forces in the horizontal direction,

$$\Sigma F^{x} = 0$$

$$\longrightarrow -F_{12}^{x} + F_{32}^{y} - m_2 (AG1^{x}) = 0$$

$$\therefore F_{12}^{x} = F_{32}^{x} - m_2 (AG1^{x}) \quad \ldots(4.59)$$

Summation of forces in the vertical direction,

$$\Sigma F^{y} = 0$$

$$\longrightarrow F_{12}^{y} - F_{32}^{y} - m_2 (AG1^{y}) = 0$$

$$\therefore F_{12}^{y} = F_{32}^{y} + m_2 (AG1^{y}) \quad \ldots(4.60)$$

Pin force at Joint-1,

$$F_{12} = \sqrt{F_{12}^{x\,2} + F_{12}^{y\,2}} \quad \ldots(4.61)$$

A similar analysis is carried out for the second and third links (See figures 7(b) and 7(c)).

$$\Sigma M_{G2} = 0$$

$$\longrightarrow T_{23} - T_{34} + F_{23}^{x} r_3 \sin\theta_3 - F_{23}^{y} r_3 \cos\theta_3 + F_{43}^{x} (L_3 - r_3) \sin\theta_3 - F_{43}^{y} (L_3 - r_3) \cos\theta_3 - I_{G2} \theta_3 = 0$$

$$\therefore T_{23} = T_{34} - F_{23}^{x} r_3 \sin\theta_3 + F_{23}^{y} r_3 \cos\theta_3 - F_{43}^{x} (L_3 - r_3) \sin\theta_3 + F_{43}^{y} (L_3 - r_3) \cos\theta_3 + I_{G2} \theta_3 \quad \ldots(4.62)$$

$$\Sigma F^{x} = 0$$

$$\longrightarrow F_{23}^{x} - F_{43}^{x} - m_3 (AG2^{x}) = 0$$

$$\therefore \quad F_{23}^{x} = F_{43}^{x} + m_3 (AG2^{x}) \qquad \ldots(4.63)$$

$$\Sigma F^{y} = 0$$

$$\longrightarrow \quad F_{23}^{y} - F_{43}^{y} - m_3 (AG2^{y}) = 0$$

$$\therefore \quad F_{23}^{y} = F_{43}^{y} + m_3 (AG2^{y}) \qquad \ldots(4.64)$$

Pin force at Joint-2,

$$F_{23} = \sqrt{F_{23}^{x\,2} + F_{23}^{y\,2}} \qquad \ldots(4.65)$$

$$\Sigma M_{G3} = 0$$

$$\longrightarrow \quad T_{34} + F_{34}^{x} r_4 \sin\theta_4 - F_{34}^{y} r_4 \cos\theta_4 + F_L^{x} (L_4 - r_4) \sin\theta_4$$
$$- F_L^{y} (L_4 - r_4) \cos\theta_4 - T_L - I_{G3} \ddot{\theta}_4 = 0$$

$$\therefore \quad T_{34} = -F_{34}^{x} r_4 \sin\theta_4 + F_{34}^{y} r_4 \cos\theta_4 - F_L^{x} (L_4 - r_4) \sin\theta_4$$
$$+ F_L^{y} (L_4 - r_4) \cos\theta_4 + T_L + I_{G3} \ddot{\theta}_4 = 0 \qquad \ldots(4.66)$$

$$\Sigma F^{x} = 0$$

$$\longrightarrow \quad -F_{34}^{x} + F_L^{x} - m_4 (AG4^{x}) = 0$$

$$\therefore \quad F_{34}^{x} = F_L^{x} - m_4 (AG3^{x}) \qquad \ldots(4.67)$$

$$\Sigma F^{y} = 0$$

$$\longrightarrow \quad F_{34}^{y} - F_L^{y} - m_4 (AG3^{y}) = 0$$

$$\therefore \quad F_{34}^{y} = F_L^{y} + m_4 (AG3^{y}) \qquad \ldots(4.68)$$

Pin force at Joint-3,

$$F_{34} = \sqrt{F_{34}^{x\,2} + F_{34}^{y\,2}} \qquad \ldots(4.69)$$

End effector loading (T_L, F_L^x, F_L^y) is zero, if arm moves freely in space. If not, it is assumed to be known. Noting that $F_{32}^x = -F_{23}^x$, $F_{32}^y = -F_{23}^y$, $F_3^x = -F_{34}^x$ and $F_{43}^y = -F_{34}^y$, equations (4.51) through (4.68) are self contained and are solvable for pin forces (F_{12}, F_{23}, F_{34}) and joint torques (T_{12}, T_{23}, T_{34}) for the given angular accelerations ($\ddot{\theta}_2$, $\ddot{\theta}_3$, $\ddot{\theta}_4$).

4.2 Synthesis

In the process of design, one usually makes a preliminary geometric layout, which is subjected to a KINEMATIC ANALYSIS in order to determine whether the displacements, velocities and accelerations are suitable for the intended purpose. KINEMATIC SYNTHESIS is a systematic approach to compute the major kinematic design variables (link lengths, fixed angles, pivot locations, etc.) [28]. Three usual problems handled by synthesis are, namely, (i) rigid body guidance, (ii) function generation, and (iii) path generation. A RIGID BODY GUIDANCE mechanism is used to guide a rigid body through a series of prescribed positions in space. A mechanism that creates an output motion that is a specified function of the input motion is known as a FUNCTION GENERATION mechanism. A PATH GENERATION mechanism will guide a point on a rigid body through a series of points on a specified path in space. This is the end effector point in case of a manipulator. In this thesis, ROBOPT synthesis is carried out for path generation application. Only some of the synthesis problems are solvable to yield 'exact' solutions. In case of highly nonlinear systems or design problems subjected to tight constraints, no exact solution can be found. In such cases,

the 'best approaximation' is sought. Approaximate synthesis problem is solved by optimal synthesis technique.

4.2.1 Optimal synthesis technique

OPTIMAL SYNTHESIS means synthesis using optimization. This method is well suited to handle path generation problem. The difference between the specified (desired) and generated motion is known as STRUCTURAL ERROR [40]. In precision point synthesis, the objective is to reduce the structural error to zero at all precision points. Freudenstein has discussed a method of systematic respacing of precision points such that the maximum structural error between each successive pair of precision points is equalized. Chebyshev spacing is used as an initial approaximation to the optimal precision point spacing. This leads to an optimum for a precision point synthesis.

A different approach to the problem is to establish a function of the structural error integrated over the full range of motion of the mechanism or the manipulator. The objective function now reduces to a minimization of this integrated structural error. A specific set of parameters that would lead to a minimum of the error function would be intuitively selected as design variables. The results obtained in this way are often superior in terms of the maximum structural error computed over the full range of

motion, even though there may be fewer points of zero error than for a comparable precision point synthesis. This is design point synthesis method. A design point is defined as any position of the mechanism or the manipulator end effector where the motion is specified. There is no requirement that the structural error be zero at all design points, only that thc sum of the squares of the structural errors, used as an approaximation of the integrated error function, should be a minimum. The generated path then tends to approach the desired path closely over the full range of the motion.

A major advantage of optimal synthesis technique is the flexibility with which the design problem can be reformulated. In real time industrial situations, the design problems are prone to change drastically. Applying aadditional constraints will solve these problems. Refer to Suh and Radcliffe [40] for the details of optimal synthesis techniques.

4.2.2 Optimization problem formulation

ROBOPT is the robot designed using the optimal synthesis technique to generate a desired path at a desired constant, high velocity with maximum possible precision.

As discussed earlier, when ROBOPT end effector traces a path, torque histories, that is torques as a function of time, are created for all three motors. In the playback mode, we just need to supply back those torque histories to repeat the required path. One possible way is to find out all three torques corresponding to each sample point of the required path and then interpolate these torques to find torque histories.

The approach adopted here involves simpler and still versatile models of the torques. The torque, T, is expressed as the summation of the first n terms of the Fourier series with an argument as time, t.

$$T = a_0 + \sum_{i=1}^{n-1} a_i \mathrm{COS}(b_i t) + \sum_{i=1}^{n-1} c_i \mathrm{SIN}(b_i t)$$

where, $a_0, ((a_i, b_i, c_i), i=1, n-1)$, are the constants of the Fourier series. Integer n controls the complexity or the refinement of the function. These constants are selected as the DESIGN VARIABLES for the optimization problem.

Integrated structural error function , involving positional and velocity errors, is selected as the OBJECTIVE FUNCTION.

Minimize the error function,

$$F(x) = \sum_{i=1}^{m} \{ (XPD_i - XPG_i)^2 + (YPD_i - YPG_i)^2 + (SPD_i - SPG_i)^2 \}$$

Where,

XPD_i, YPD_i, SPD_i are the x,y coordinates and the constant speed of sample points on the desired

curve.

XPG_i, YPG_i, SPG_i are the x,y coordinates and the speed of corresponding points generated by the end effector. Their formulae are included in KINEMATICS section.

m is the number of segments of the desired and generated curves and controls the smoothness of the curves.

The CONSTRAINTS vary according to industrial situations. They could be either geometric or physical in nature. A typical set of constraints would be :

(1) $\theta_{2_{min}} < \theta_2 < \theta_{2_{max}}$ (bound on joint-1 motion)

(2) $\theta_{3_{min}} < \theta_3 < \theta_{3_{max}}$ (bound on joint-2 motion)

(3) $\theta_{4_{min}} < \theta_4 < \theta_{4_{max}}$ (bound on joint-3 motion)

(4) $T_{12_{min}} < T_{12} < T_{12_{max}}$ (servo motor capacity bounds)

(5) $T_{23_{min}} < T_{23} < T_{23_{max}}$ (servo motor capacity bounds)

(6) $T_{34_{min}} < T_{34} < T_{34_{max}}$ (servo motor capacity bounds)

(7) $(XPG^2+YPG^2) \leq (L_2+L_3+L_4)^2$ (work-space bound)

According to special needs, some equality constraints also may arise. A major requirement for this approach is the need for a powerful nonlinear software package as most of the problems would involve design variables in the range of 12 to 24 with 6 to 10 inequality constraints and possible 1 or 2 equality constraints. One more fact to notice is that at first sight, it appears that,

the design variables are not involved in the objective function. In fact, the involvement is indirect. The design variables control the torques which in turn control the generated positions and velocities. Also, in such formulations, the initial guess of the design variables plays an important role. The optimization problem would converge by different amounts by starting at different initial points. For reasonable convergence, the initial guess should not be too close to the solution or too far away from it. A simple analysis software package, ROB, is created to find a reasonable starting point to be fed to the main software package, ROBIAS, which involves the optimization procedure.

4.2.3 Actuators and the torque models

There are three major ways today of powering the actuators that move the manipulator joints: pneumatic, hydraulic and electrical. The PNEUMATIC actuators are simple and inexpensive but "non-rigid" and is best used for pick & place robots whose trajectory motions are controlled by mechanical stops. HYDRAULIC actuation (currently the most popular system of the three) is capable of high power for a given size, but suffers from leackage problems as well as requires pumps, storage tanks and other accompanying paraphenalia. ELECTRO-MECHANICAL actuation is clean but does

paraphenalia. ELECTRO-MECHANICAL actuation is clean but does not offer good power to weight ratios. STEPPER MOTORS are inexpensive but sometimes lose pulses and therefore accuracy. D.C. SERVO MOTORS are reliable and offer good controllability, but can be expensive [19].

High speed, high precision robot manipulators require smaller, more reliable and efficient actuators and drive mechanisms with high load to weight ratios. Hence three small d.c. servo motors are selected for three revolute joints of ROBOPT. These motors are mounted right on the pin axes of the joints.

As described in the previous section, the torque models are selected as the summation of first n terms of the Fourier series. For the present state of formulation of the optimization problem, it is found that including only the first sine and cosine terms of the fourier series provides a good torque model and avoids complexity for the optimization problem. The torque models for the three motors are :

$$\begin{cases} T_{12} = X(1)+X(2)*COS(X(3)*t)+X(4)*SIN(X(3)*t) \\ T_{23} = X(5)+X(6)*COS(X(7)*t)+X(8)*SIN(X(7)*t) \\ T_{34} = X(9)+X(10)*COS(x(11)*t)+X(12)*SIN(X(11)*t) \end{cases}$$

where, x(1) thru x(12) are the design variables for the optimization problem. Boykin and Diaz [6] provide a detailed technological assessment of robotic actuators available today.

CHAPTER 5

APPLICATIONS

Robots have wide applications in the industry. They have already been tested and proven reliable for a large variety of operations in various industries like electronics and electrical machinery, automobile, plastic moulding, metal working, steel making, and chemical, etc.

According to the kind of motion generated by the robot controller, industrial robots can be divided into three main catagories. (i) pick & place robots, (ii) point-to-point control robots, and (iii) continuous path robots.

PICK & PLACE ROBOTS are also referred to as "limited sequence" or "fixed stop" robots. As the names suggest, these robots employ the simplest controllers to control the manipulator end point positions along each of the axes. The controllers are often non-servo (open loop) devices that rely on sequencers and mechanical stops. So these robots have no provision for trajectory control between the end points. Eventhough these are the simplest kind of robots, they have maximum applications in a number of important industrial operations like die-casting, investment casting, forging, foundry, press working, machine tool loading,

assembling, heat treatment, glass manufacturing, and many more.

POINT-TO-POINT ROBOTS are directed by a programmable controller that memorizes a sequence of arm and end effector positions. The robot moves in a series of steps from one memorized point to another under servo control, using internal joint sensors for feedback. Because of the servos, trajectory control between the memorized points is possible and relatively smoother motions can be achieved.

CONTINUOUS PATH ROBOTS do not depend on a series of intermediate points to generate a trajectory but duplicate during the playback mode the continuous motions recorded during the teaching mode. PATH GENERATION ROBOTS fall under this catagory. These robots are used for seam welding (arc welding), spray painting, deburring along a prescribed contour line, laser or flame cutting, conveyor tracking and other processes requiring smooth continuous motions. For further discussion regarding these robots, the reader is directed to Paul [29], Ardayfio and Pottinger [1], and Heer [21].

5.1 Pick & Place

ROBOPT is to perform two main tasks. The pick & place and the path generation. The simpler of the two, the pick & place operation could be performed by commanding the robot to move from the current position to a desired position. A set of non-linear equations are solved in joint coordinates and corresponding joint torques are supplied to the actuators to move the end effector to the desired position. These robots could be much conveniently operated if a MOVE(x,y,z) macro command, written in the system language of the robot controllong computer, is transmitted to the robot to actuate the joint motors. In order to avoid jerks between pick and place, interpolation in joint coordinates or directly in joint torques could be carried out to supply intermediate points of the motion. A resolved rate positional control method using interpolation criteria was suggested by Whitney [42], as early as 1972.

5.2 Path Generation

There are basically three approaches to a path generation problem. The simplest way is the 'teach-playback' capability. For continuous motion applications, it requires a large amount of memory to memorize all intermediate positions of the trajectory between its end points. They can not be operated "on-hand" as they require tedious teaching

lesson. The second method is an extension to the method described for pick & place operation. It requires the input of a number of MOVE commands corresponding to the trajectory sample points and then using the interpolation technique, robot could be operated almost in real-time. Again, this is an "exact" method. The third method uses the optimal synthesis technique and requires a pre-programming. Although it requires a reasonable amount of time to run a single program using optimization, this method can handle complex curves to be generated within tight industrial constraints. This flexibility surges this method ahead of the two previous methods.

A major path generation application of the high speed, high precision ROBOPT arises in the textile industry when it is required to cut and/or bond a path on the fabric moving on a conveyor belt. Laser reflectors at the robot base and end effector are required for this cutting, to avoid gripper and hence inertia problems.

Two interesting surveys of contemporary manipulator devices along with their applications are cited by Robotics Age [36] and The Indutrial Robots [22] magazines.

CHAPTER 6

NUMERICAL METHODS

Numerical methods are essential in solving mathematical systems by using a digital computer. This science is well developed and many algorithms are available. In developing an analysis and synthesis software package for ROBOPT, numerical methods were used for different purposes like: (i) solving simultaneous higher order differential equations, (ii) solving a system of linear equations, (iii) solving a system of nonlinear equations, (iv) numerical integration, (v) numerical differentiation, and (vi) interpolation. Whenever possible, different kinds of algorithms were tested and the one found best for the application was used. Particularly, experience with interpolation was found to be interesting. Gerald [18] is strongly recommended as a general reference to this area.

6.1 Numerical Solution Of Simultaneous, Higher Order Differential Equations

The Adams-Moulton predictor corrector procedure with the Runge-Kutta starter is used. Subroutine RKAM [23,33] is designed to compute the numerical solution of a system of N, (1<N<10), simultaneous first order differential equations over a specified interval with given initial conditions.

An n-th order differential equation of the form

$$\frac{d^n y}{dx^n} = f\left(x, y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \ldots, \frac{d^{n-1} y}{dx^{n-1}}\right)$$

with initial values $y, \frac{dy}{dx}, \frac{d^2 y}{dx^2}, \ldots, \frac{d^{n-1} y}{dx^{n-1}}$

can be transformed into a system of n simultaneous first order differential equations. This can be accomplished by assigning,

$$y_1 = y \ ;$$

$$y_2 = \frac{dy}{dx} \ ;$$

$$y_3 = \frac{d^2 y}{dx^2} \ ;$$

$$\vdots$$

$$y_n = \frac{d^{n-1} y}{dx^{n-1}}$$

Using this technique, ROBOPT system of three simultaneous second order differential equations is transformed into a system of six simultaneous first order differential equations.

The second order differential equations are :

$$\begin{cases} a_1\ddot{\theta}_2 + a_2C_{32}\ddot{\theta}_3 + a_3C_{42}\ddot{\theta}_4 = a_2S_{32}\dot{\theta}_3^2 + a_3S_{42}\dot{\theta}_4^2 + T12 - T23 & \ldots(6.1) \\ a_2C_{32}\ddot{\theta}_2 + a_4\ddot{\theta}_3 + a_5C_{43}\ddot{\theta}_4 = -a_2S_{32}\dot{\theta}_2^2 + a_5S_{43}\dot{\theta}_4^2 + T23 - T34 & \ldots(6.2) \\ a_3C_{42}\ddot{\theta}_2 + a_5C_{43}\ddot{\theta}_3 + a_6\ddot{\theta}_4 = -a_3S_{42}\dot{\theta}_2^2 - a_5S_{43}\dot{\theta}_3^2 + T34 & \ldots(6.3) \end{cases}$$

Let,

$$\theta 2 = \dot{\theta}_2 \quad \ldots(6.4)$$

$$\dot{\theta} 2 = \ddot{\theta}_2 \quad \ldots(6.5)$$

$$\theta 3 = \dot{\theta}_3 \quad \ldots(6.6)$$

$$\dot{\theta} 3 = \ddot{\theta}_3 \quad \ldots(6.7)$$

$$\theta 4 = \dot{\theta}_4 \quad \ldots(6.8)$$

$$\dot{\theta} 4 = \ddot{\theta}_4 \quad \ldots(6.9)$$

substituting (6.4) thru (6.9) into (6.1) thru (6.3),

$$\begin{cases} a_1\dot{\theta}2 + a_2C_{32}\dot{\theta}3 + a_3C_{42}\dot{\theta}4 = a_2S_{32}\theta 3^2 + a_3S_{42}\theta 4^2 + T12 - T23 & \ldots(6.10) \\ a_2C_{32}\dot{\theta}2 + a_4\dot{\theta}3 + a_5C_{43}\dot{\theta}4 = -a_2S_{32}\theta 2^2 + a_5S_{43}\theta 4^2 + T23 - T34 & \ldots(6.11) \\ a_3C_{42}\dot{\theta}2 + a_5C_{43}\dot{\theta}3 + a_6\dot{\theta}4 = -a_3S_{42}\theta 2^2 - a_5S_{43}\theta 3^2 + T34 & \ldots(6.12) \end{cases}$$

This linear system can be solved for $\dot{\theta}2=\ddot{\theta}_2$, $\dot{\theta}3=\ddot{\theta}_3$ and $\dot{\theta}4=\ddot{\theta}_4$. The system of 6 simultaneous first order differential equations is:

$$\begin{cases} F(1) = \theta_2 & \ldots(6.13) \\ F(2) = \theta_3 & \ldots(6.14) \\ F(3) = \theta_4 & \ldots(6.15) \\ F(4) = \dot{\theta}_2 & \ldots(6.16) \\ F(5) = \dot{\theta}_3 & \ldots(6.17) \\ F(6) = \dot{\theta}_4 & \ldots(6.18) \end{cases}$$

Submitting the system of equations F(1) thru' F(6) to RKAM, differential equations are solved for $\theta_2, \theta_3, \theta_4, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4$, given, range of independent variable (time) and initial conditions for the dependent variables.

6.2 Numerical Solution Of A System Of Linear Equations

Standard Gaussian elimination procedure is used to solve a system of 3 linear equations in 3 unknowns. Subroutine GAUSS [33] accepts a coefficient matrix and a right hand side constant vector and returns the solution vector. In the analysis package for ROBOPT, differential equations are solved for finding accelerations by Gaussian elimination.

6.3 Numerical Solution Of A System Of Nonlinear Equations

For solving a system of nonlinear equations, the Newton-Ralphson method [18] is a well-acclaimed procedure.It is interesting to observe that basically Newton's method, as applied to a set of nonlinear equations, reduces the problem to solving a set of linear equations in order to determine the values that improve the accuracy of the estimates.

Newton's method has the advantage of converging quadratically, at least when near a root, but is expensive in terms of function evaluations. For n simultaneous equations, the number of function evaluations is $(n^2 + n)$. Hence it is rarely applied to large systems. It is good strategy to reduce the number of equations as much as possible by solving for one variable in terms of the others and eliminating that one by substituting for it in the other equations.

The subroutine that solves a nonlinear system is named NLSYS. The partial derivatives of the functions are estimated by difference quotients when a variable is perturbed by an amount equal to delta (delta is added). This is done for each variable in each function. The system of linear equations is solved by subroutine GAUSS, employing Gaussian elimination techniques. Various debugging messages are also provided.

While employing Newton's method, every care must be taken in selecting the initial guess to avoid divergence of the algorithm.

In case of ROBOPT, the use of subroutine NLSYS to solve a system of 2 nonlinear equations proved to be quite effective. Given the position of the end effector, (XPG,YPG), and constant value of one of the joint variables, say θ_2, system of 2 nonlinear equations can be solved to yield the values of the other two joint variables, θ_3 ,θ_4 .

$$\begin{cases} XPG=L_2COS(0.0)+L_3COS(\theta_3)+L_4COS(\theta_4) & ...(6.19) \\ YPG=L_2SIN(0.0)+L_3SIN(\theta_3)+L_4SIN(\theta_4) & ...(6.20) \end{cases}$$

Given a position or a trace of positions of the end effector, (XPG,YPG), and initial guesses for variables θ_3 ,θ_4 , Newton's method finds a root by an iterative procedure. It should be noted that for a system of 2 equations, 2 roots are possible. Joint variables for path generation application may be obtained by submitting different (XPG,YPG) positions and keeping the initial guess the same. Using this method, the path generation application was successfully tested to generate a variety of paths including horizontal straight lines, vertical straight lines, ellipses and circles.

6.4 Numerical Integration

The arc length of a parametric curve is obtained by integrating the arc segments over initial and final angles. Basically, the trapezoidal rule is used in subroutine SIMP to calculate Simpson's one-third rule value of integration. The subroutine SIMP calls routine TRAP twice with different step sizes for this purpose [20]. The subroutine SIMP was tested to find perimeter of a unit circle yielding high numerical accuracy. It was then used to calculate the value of the arc length of the given curve.

6.5 Numerical Differentiation

In finding the arc length, gradient calculations are required. The first order derivatives were calculated by forward differencing. In double precision application of the subroutine RGRAD, a perturbation value of 1.D-8 was used for high precision.

6.6 Inverse Interpolation

Inverse interpolation [18] is the technique of finding the value of argument for given function value from a set of descrete data. Usually in practical problems, the function values are unevenly spaced. Due to this, only Lagrangiam and cubic spline methods can handle such inverse interpolations. In ROBOPT application, equal arc length

segments were obtained by finding approapriate time intervals through inverse interpolation. The number of data pairs used for interpolation were 20. Both simple and then two segment Lagrangian methods failed to give even reasonable inverse interpolation because of the dexterity of the input data pairs (for generated curve, the values were highly unevenly spaced). Also, Lagrangian method is computationally inefficient when the number of points increases. On the other hand, cubic spline method was found very accurate and efficient for the present application. After a brief introduction to the cubic spline theory, a variety of its applications are explained to conclude the chapter.

6.6.1 The cubic spline theory

A cubic spline divides the entire interval into subintervals and then fits the data within each of these subintervals. The cubic for the i-th interval, which lies between the points (x_i, y_i) and (x_i, y_{i+1}) is of the form:

$$y = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i \qquad \ldots(6.21)$$

Since, the cubic spline fits through the end points of the of the interval,

$$y_i = a_i(x_i-x_i)^3 + b_i(x_i-x_i)^2 + c_i(x_i-x_i) + d_i$$

$$\therefore \quad d_i = y_i \qquad \ldots(6.22)$$

$$y_{i+1} = a_i(x_{i+1}-x_i)^3 + b_i(x_{i+1}-x_i)^2 + c_i(x_{i+1}-x_i) + d_i$$

$$\therefore \quad y_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i h_i + d_i \qquad \ldots(6.23)$$

where,

$$h_i = x_{i+1} - x_i \qquad \ldots(6.24)$$

The slopes (y') and curvature (y'') are obtained by differentiating y with respect to x successively.

$$y' = 3a_i(x-x_i)^2 + 2b_i(x-x_i) + c_i \qquad \ldots(6.25)$$

$$y'' = 6a_i(x-x_i) + 2b_i \qquad \ldots(6.26)$$

Let,

S_i represent the second derivative at point (x_i, y_i)

S_{i+1} represent the second derivative at point (x_{i+1}, y_{i+1})

from equations (6.25) and (6.26),

$$S_i = 6a_i(x_i-x_i) + 2b_i$$

or,

$$b_i = S_i/2 \qquad \ldots(6.27)$$

$$S_{i+1} = 6a_i(x_{i+1}-x_i) + 2b_i$$

or,

$$a_i = (S_{i+1}-S_i)/6h_i \qquad \ldots(6.28)$$

substitute (6.22),(6.27) and (6.28) in (6.23),

$$y_{i+1} = \frac{S_{i+1}-S_i}{6h_i} h_i^3 + \frac{S_i}{2} h_i^2 + (C_i)h_i + (y_i)$$

$$\therefore \quad C_i = \frac{y_{i+1}-y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \qquad \ldots(6.29)$$

Now, invoke the condition that the slopes of two cubics that join at (x_i, y_i) are the same.

from (6.26),

$$y_i' = 3a_i(x_i - x)^2 + 2b_i(x_i - x_i) + c_i$$

Also,

$$y_i' = 3a_{i-1}(x_i - x_{i-1}) + 2b_{i-1}(x_i - x_{i-1}) + c_{i-1}$$

therefore,

$$(h_{i-1})S_{i-1} + (2h_{i-1}+2h_i)S_i + (h_i)S_{i+1} = 6\,\frac{y_{i+1}-y_i}{h_i} - \frac{y_i - y_{i-1}}{h_i} \quad \ldots(6.30)$$

where, $i=2 \longrightarrow n-1$ yields n-2 equations in n unknowns (S_n).

<u>end conditions</u> of the curve, S_1 and S_n, are assumed.

$$\begin{cases} S_1 = 0.,\ S_n = 0. & \text{(Linear)} \\ S_1 = S_2,\ S_n = S_{n-1} & \text{(Parabolic)} \\ S_1 = \dfrac{(h_1+h_2)S_2 - h_1 S_2}{h_2}, \quad S_n = \dfrac{(h_{n-2}+h_{n-1})S_{n-1} - (h_{n-1})S_{n-2}}{h_{n-2}} & \text{(Extrapolation)} \end{cases}$$

Now, tridiagonal system of n-2 equations in n-2 unknowns is solved by Gaussian Elimination to find values of S_1 thru' S_n .

Hence,

For the given value of independent variable x, bounded by the interval $[x(I)<x<x(I+1)]$, the value of dependent variable y is found by interpolation by the following equations provided that $((x_i, y_i), i=1\text{-}n)$ data points and second derivatives at these points are known.

$$y = a_i(x-x_i)^3 + b_i(x-x_i)^2 + c_i(x-x_i) + d_i \quad \ldots(6.31)$$

$$a_i = \frac{S_{i+1} - S_i}{6h_i} \qquad \ldots(6.32)$$

$$b_i = S_i/2 \qquad \ldots(6.33)$$

$$c_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i S_i + h_i S_{i+1}}{6} \qquad \ldots(6.34)$$

$$d_i = y_i \qquad \ldots(6.35)$$

6.6.2 Cubic spline applications

The cubic spline theory is particularly important for its applications to discrete, unevenly spaced data points. The following are the major uses.(1) Direct interpolation, (2) Extrapolation, (3) Inverse interpolation, (4) First and second derivatives, (5) Numerical integration, (6) To find zero's of a function, (7) For extension to B-Spline theory to draw smooth curves.

(1) Direct interpolation,

(2) Extrapolation,

(3) Inverse Interpolation,

(4) First and second derivatives,

(5) Numerical integration,

(6) Zero's of a function,

(7) Extension to B-spline theory
to draw smooth curves.

CHAPTER 7

DYNAMIC SIMULATION

A picture is worth a thousand words. The visualization becomes much more powerful when the full range of motions of all the parts of the system is displayed on a graphics terminal. For such a dynamic display, the graphics system used is PS300 [10-14], a high performance, 3-dimensional, vector refresh system, designed and developed by Evans & Sutherland. E & S systems have been used in a variety of applications, including CAD, visualization of molecular and mathematical models, visual simulations, etc.

7.1 PS300

PS300 has the capability to define and display two and three dimensional pictures that may be rotated, translated and scaled dynamically while maintaining an orthographic or true perspective view. Depth cueing and intensity variations make it an efficient three dimensional system.

7.1.1 Graphics concept

PS300 provides high performance graphics. An 8Kx8K addressible, 19 inch diagonal, HIGH RESOLUTION, VECTOR REFRESH display is used. For the display to appear flicker-free, REFRESH RATE, the number of pictures drawn per second, used, is as high as 60 Hz. UPDATE RATE, the number of different pictures drawn per second, is kept greater than 10 so that SMOOTH MOTIONS appear on the screen. As the system time-lag is less than a few seconds, it works as an 'interactive' device. Modeling data space coordinates are automatically converted to screen coordinates. Depth cueing and perspective transformations create an illusion of three dimensional objects. Using system clock functions, various transformations can be carried out to offer dynamic display capability.

Various interactive devices available for data manipulation are keyboard, data tablet, function buttons, control dials, joy sticks, light pen, track ball, mouse and touch-sensitive CRT's. An optional output device is selected from one of the CRT's or phosphors like Direct View Storage Tube (DVST), Raster Scan terminal (dot matrix), Calligraphic Refresh (Vector Refresh) and Calligraphic Shadow Mask (CSM) color monitor. Other highlights of PS300 are its data structuring, function networking, local actions and distributed graphics capabilities.

7.1.2 PS300 GRAPHICS LANGUAGE

For dynamic display or transformation of objects, a user defined 'structured' program in PS300 GRAPHICS LANGUAGE [5, 10, 11] is to be provided. Basically, any PS300 program is a combination of display data structures, one or more vector_lists of the objects to be displayed and a function network for desired transformations of the objects. The display data structure mainly contains viewing, modeling transformations and object instancing commands. A PS300 vector_list defines the shape of the object to be displayed by a series of world coordinate locations with associated Position or Line attributes. A function network links inputs from an interactive device to transformation commands in a display data structure. Also, various arithmatic and logical operations are carried out within such networks.

7.1.3 Local actions

The ability to create, incrementally modify, and delete hierarchial data structures at the directions of host computer without the need to retransmit the entire model is unique to PS300. The LOCAL ACTIONS are performed on PS300 using collections of functions that are linked together by the user to form data-driven networks. One end of the network is connected to an element of a display data structure and the other end to the port associated with an

interactive device or the host. These local actions usually determine how the operations of an interactive device affect visual presentation.

7.1.4 Distributed graphics

The DISTRIBUTED GRAPHICS processing capability lets the host computer partition an application into graphical and non-graphical portions, and to distribute the graphical portion of the application processing task to PS300. The advantages of the distributed graphics are threefold. (i) PS300 performs locally the processing and data management tasks needed to define, display and interact with graphical data without burdening the host computer. Then, whenever a result of a graphical process impacts the host application, that result can be communicated to the host at a relatively low bandwidth, thereby greatly reducing the frequency and magnitude of host communications and minimizing graphical processing tasks in the host. (ii) In many applications, such as computer-Aided-Design (CAD), non-graphical portion represents most of the tasks. Nonetheless, designers who interact with the image being viewed to build, modify, or examine a mathematical model, spend most of their time with graphical portion. This interaction may continue, for a significant period of time, and only occassionally during that period does the non-graphical portion in the host need to be appraised of the graphical portion locally, the total

processing requirements of the host computer can be greatly reduced so that more designers can share the application program before increasing host processing capabilities. (iii) In other applications, such as visual simulations, the graphical portion represents the bulk of the total application. Most of the transformation required for this application can be performed locally in PS300.

7.1.5 PS300 architecture

PS300 is a data-driven, interactive computer graphics system. It consists of several general and special purpose processors and subsystems that are interfaced by means of a general data bus in conjunction with a mass memory system. A Motorola MC68000 microprocessor is dedicated to control these activities. PS300 hardware component architecture is shown in figure 8 [10]. The main components are : (i) A Graphics Control Processor (GCP), incorporating the microprocessor; (ii) Mass Memory (MM), containing Random Access Memory (RAM) upto 4MB, and (iii) A Display Processor (DP), including an arithmatic control processor, a pipeline subsystem and a line generator. Display station incorporates the vector refresh CTR,

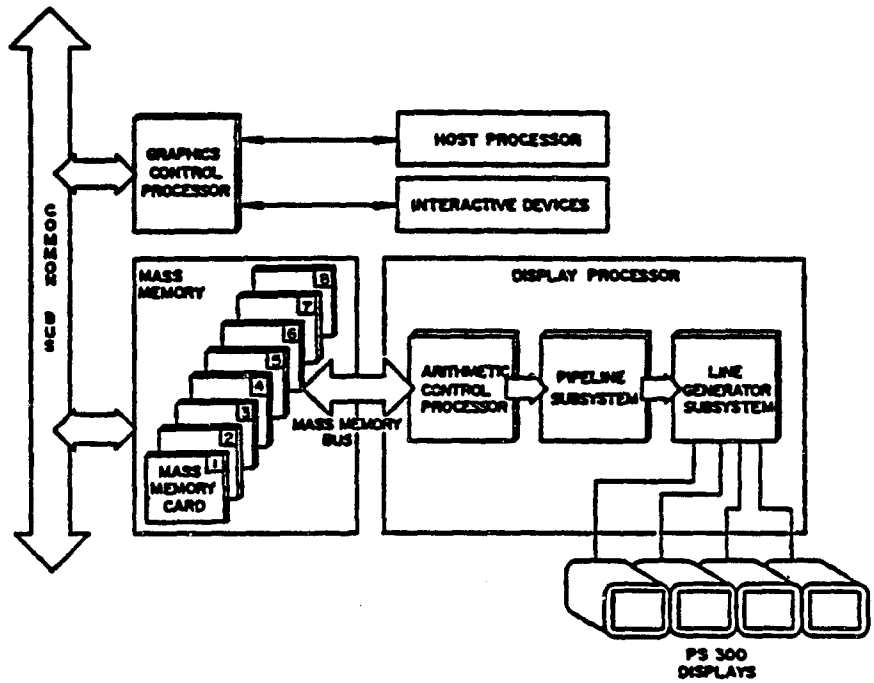


Figure 8 : PS300 System Architecture

a keyboard with a LED display, eight dial control unit and a 200 units per inch resolution data tablet.

7.2 Frame-By-Frame Animation

Frame-by-frame animation [12] is a technique of using the system defined clock functions (F:CLCSECONDS, F:CLFRAMES, or F:CLTICKS) to cycle through a series of previously calculated, user defined frames. Typically, each frame would consist of different transformations applied to the same object. The modulo function (F:MODC) allows for the animation to recycle indefinitely. Offcourse, this recycling could be stopped by sending a 'false' message at an appropriate port in the network.

The frame-by-frame animation method is used to display various positions of the links of the manipulator arm as corresponding frames and then recycling thru' these frames to display the full range of motions of the arm. Also, the same technique is used to display a moving conveyor by applying cyclic translations to the 'strips' on the belt to create an illusion of continuous motion. The function network diagram is shown in figure 9. Based on this function network, a typical PS300 animation program would be:

```
TIMER:=F:CLCSECONDS ;
MODULO:=F:MODC ;
TRIGGER1:=F:SWITCHC ;
TRIGGER2:=F:SWITCHC ;

SEND FIX(1) TO <2>TIMER ;
```

```
SEND FALSE TO <3>TIMER ;
SEND FIX(1) TO <4>TIMER ;
SEND FIX(0) TO <5>TIMER ;
CONNECT FKEYS<1>:<1>TRIGGER1 ;
CONNECT FKEYS<1>:<1>TRIGGER2 ;
SEND TRUE TO <2>TRIGGER1 ;
SEND FALSE TO <2>TRIGGER2 ;
SEND 'STARTANI' TO <1>FLABEL1 ;
SEND 'STOPANI' TO <1>FLABEL2 ;
CONNECT TRIGGER1<1>:<6>TIMER ;    { start animation }
CONNECT TRIGGER2<2>:<6>TIMER ;    { stop animation }
CONNECT TIMER<2>:<5>TIMER ;
CONNECT TIMER<2>:<1>MODULO ;
CONNECT MODULO<1>:<1>ARM.LEVEL ;
SEND FIX(4) TO <1>TIMER ;         { animation speed }
SEND FIX(21) TO <2>MODULO ;       { number of frames }

ARM:=BEGIN_STRUCTURE
LEV:=SET LEVEL_OF_DETAIL TO 0 ;
        IF LEVEL=0 THEN POSITION0 ;
        IF LEVEL=1 THEN POSITION1 ;
        :
        :
        IF LEVEL=20 THEN POSITION20 ;
END_STRUCTURE ;
```

POSITION0 through POSITION20 are the static frames of the motion. The motion of the arm looks continuous when POSITION0 through POSITION20 are displayed sequentially at high rate. A similar network is applied for the conveyor motion. The speed of animation can be changed dynamically by sending an integer to input <1> of TIMER. Due to these capabilities, PS300 is called a dynamic display system.

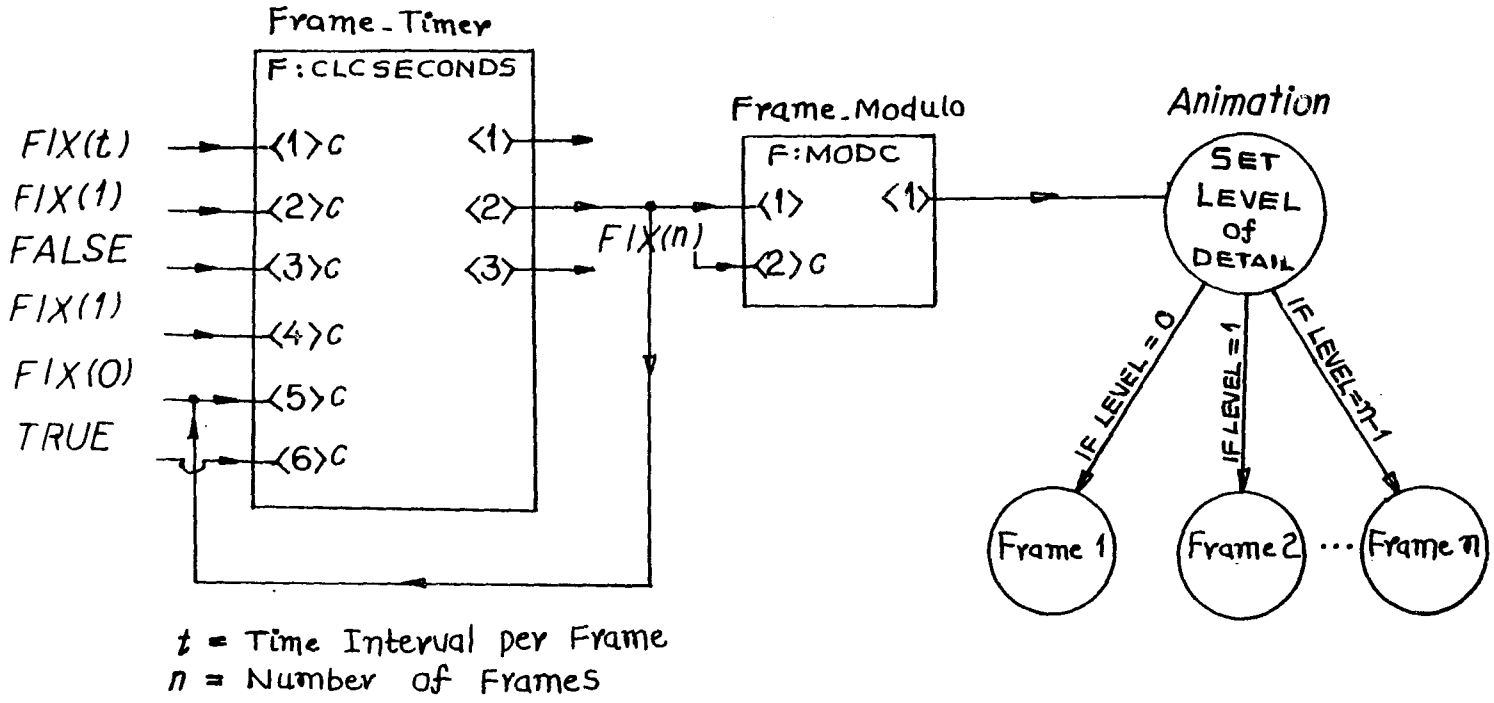


Figure 9 : Frame-By-Frame Animation

7.3 Data Tablet

The PS300 data tablet is a high resolution input device. It is used mainly for menu selection by the use of a 'pick' function and vector input by using a 'tabletin' function. For interactive use of the PS300 data tablet, program SKETCHPAD was developed which enables the user to input his curve by a grid banding technique, send the associated vector_list over to the host and also store the same in a program opened file for redisplay of the curve. Figure 10 shows an overview of the function networks used in SKETCHPAD. Program SKETCHPAD contains the following elements:

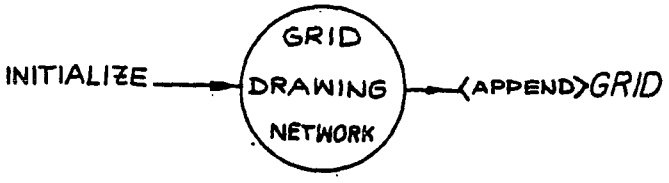
(i) grid drawing network
(ii) grid banding network for curve input
(iii) curve clear/delete network
(iv) dynamic cursor network
(v) host message network

7.3.1 Grid drawing network

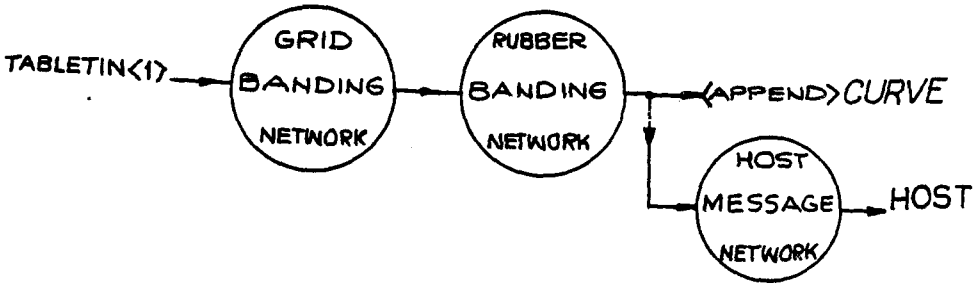
The GRID DRAWING NETWORK [10] defines and displays a 'grid' and is initialized by sending a count to <1>N where N

GRID: = VEC 0,0 ;
CURVE: =VEC 0,0 ;

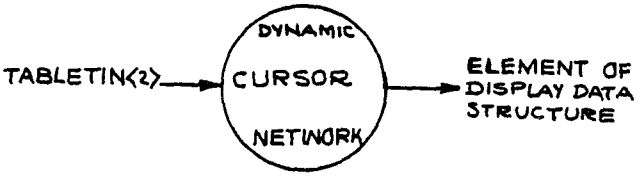
(a)



(b)



(c)



(d)

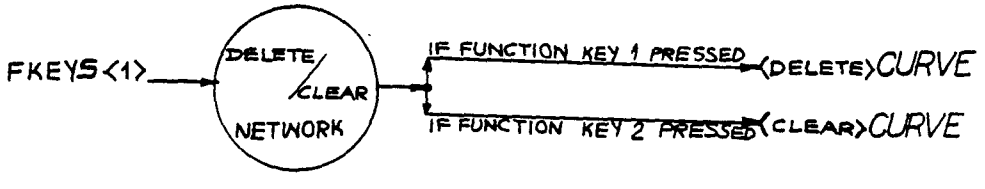


Figure 10 : Program 'Sketchpad' Function Networks

is the name given to F:NOP function and is used as a variable. The count defines the dimensions of the grid squares by setting how many lines are drawn in each quadrant of the screen. A count of 50 causes 50 grid lines in x and in y to be drawn from the center to the edge of the full screen viewport. That is, a grid with 100 vertical lines and 100 horizontal lines is created. Each fifth line in the grid is displayed at a higher intensity, providing a 'grid paper' effect. The viewport selected is -1:1 units in both horizontal and vertical directions. For this count and the size of viewport, minimum dimension of the unit in both x and y directions would be .02. The grid could be refined by sending a higher count.

7.3.2 Grid banding network

The GRID BANDING NETWORK [12] is used for the user curve input. When a rubber banding is performed to discrete points on a grid by the process of 'normalization', grid banding is generated. That is, 2D coordinate points selected from the data tablet are constrained to fall on intersecting points of the user defined grid lines. Also, by sending a high count to <1>N, the grid becomes refined and gives 'rubber-band' sketches. User inputs his curve by selecting discrete points from the

data tablet grid. A selected point can be either starting or ending position of a line segment. A combination of position or line vectors defines the curve.

In RUBBER-BANDING technique [12], the stylus on the tablet is pressed and released once to fix the first position. Moving the stylus around on the tablet now will create a rubber-band line from the initial position to the current position of the cursor. Pressing and releasing the stylus again will fix this line segment, and a new rubber line will start from the last point to the next point pressed down and so on. To break this continuous line and start a new series of rubber_band segments, the stylus must be moved away from the tablet surface for a short time (15 centiseconds). This will cause the current rubber band line to disappear; a new one will start as soon as a new position is selected.

7.3.3 DELETE/CLEAR network

The delete/clear network enables the user correct any error in his curve input or start over again for a new curve. By pressing the function key with LED display 'clear', the whole curve will disappear whereas by pressing the function with LED display 'delete', only the last vector will be removed.

7.3.4 Dynamic cursor

A DYNAMIC CURSOR [12] is used for curve input. The cursor is dispayed as a 'pen' when it is in proximity to the tablet surface. But, when a vector is selected by pressing the cursor down, the cursor changes its shape to a 'cross' (X) sign. This gives confirmation to the user whether correct points have been selected. A conditional bit is used to select the shape of the cursor. When the conditional_bit is 'off', the cursor shape is a pen and when the bit is set 'on', a cross shape is displayed for the cursor. A PS300 conditional_bit is turned 'on' if fed a true message, and 'off' if fed a false message. The result is then passed on to an 'if' statement which tests the status of the bit and performs the desired action. The structure that selects the shape of the dynamic cursor is:

```
DISPLAY CURSOR ;
CURSOR:=BEGIN_S
        UP_DOWN:=SET CONDITIONAL_BIT 1 OFF ;
        IF CONDITIONAL_BIT 1 IS OFF THEN CURSOR_UP ;
        IF CONDITIONAL_BIT 1 IS ON THEN CURSOR_DOWN ;
        END_S ;
```

CURSOR_UP and CURSOR_DOWN are the predefined vector_lists for the optional shapes of the dynamic cursor.

7.3.5 Host message network

The selected vectors are sent to the host computer by a HOST_MESSAGE network [14]. HOST_MESSAGE and HOST_MESSAGEB are the system defined function names. The position/line vectors selected by grid banding are connected to a PRINT function. The output from the PRINT function is connected to the input <1>HOST_MESSAGE. The network has to be initialized each time a message is sent to the host by sending a fixed point 0 to input <1>HOST_MESSAGEB. The message sent to the host is read as a character string in the FORTRAN program, stored in an opened file called TAB.BAK, and coordinates are extracted from this string by a subroutine named TABREAD.

7.4 Conditional Display

In addition to the networks for frame-by-frame animation and SKETCHPAD, other major work carried out on PS300 involves the conditional display of various slides. By pressing different function keys of PS300, different slides may be selected for viewing. A SLIDE is simply defined, here, as a display occupying the full window. SLIDE-1 (Figure 11) contains the design data and curves for input and output characteristics of ROBOPT in different viewports. This slide is the default display.

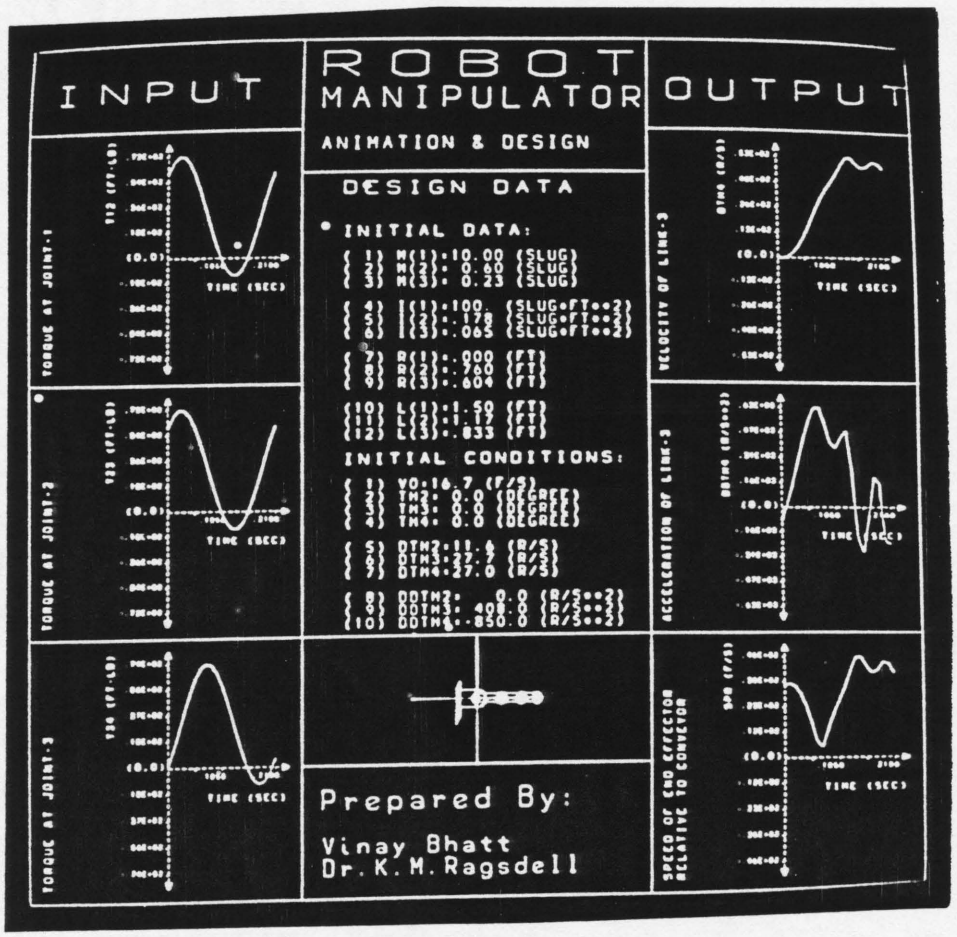


Figure 11 : PS300 SLIDE-1

SLIDE-2 (Figure 12) dispays the animation of ROBOPT. Also, an analog clock is displayed in one corner of the slide to measure 'real' time of motions. The desired and generated curves can be separately displayed by viewing SLIDE-3 (Figure 13). By turning one slide 'on' the others are automatically turned 'off'. Although, SLIDE-2 and SLIDE-3 can be superimposed. The structure for conditional display is :

```
DISPLAY ROBOPT ;
ROBOPT:=BEGIN_STRUCTURE
        C_B_1:=SET CONDITIONAL_BIT 1 ON ;
        C_B_2:=SET CONDITIONAL_BIT 2 OFF ;
        C_B_3:=SET CONDITIONAL_BIT 3 OFF ;
        IF CONDITIONAL_BIT 1 IS ON THEN SLIDE-1 ;
        IF CONDITIONAL_BIT 2 IS ON THEN SLIDE-2 ;
        IF CONDITIONAL_BIT 3 IS ON THEN SLIDE-3 ;
        END_STRUCTURE ;
```

A function network may be written to change the ON/OFF status of conditional_bit 1,2 and 3. PS300 function KEYS are used for this purpose.

7.5 The PS300 Experience

Working with PS300 helped ROBOPT design to a large extent. Particularly, in the optimal synthesis for path generation, the user gains 'insight' in selection of the initial shape of his curve and design parameters by viewing the slides. The user may also modify his design quite comfortably by the display of his final curves. It is

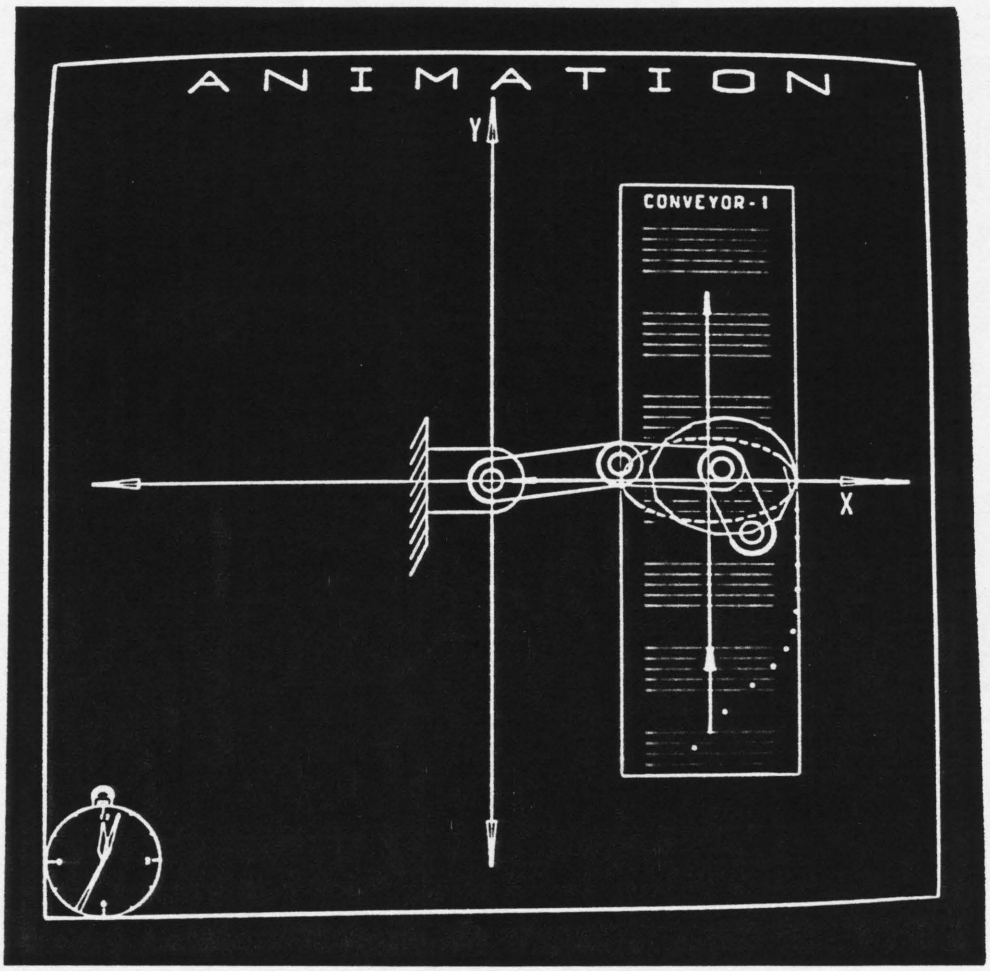


Figure 12 : PS300 SLIDE-2

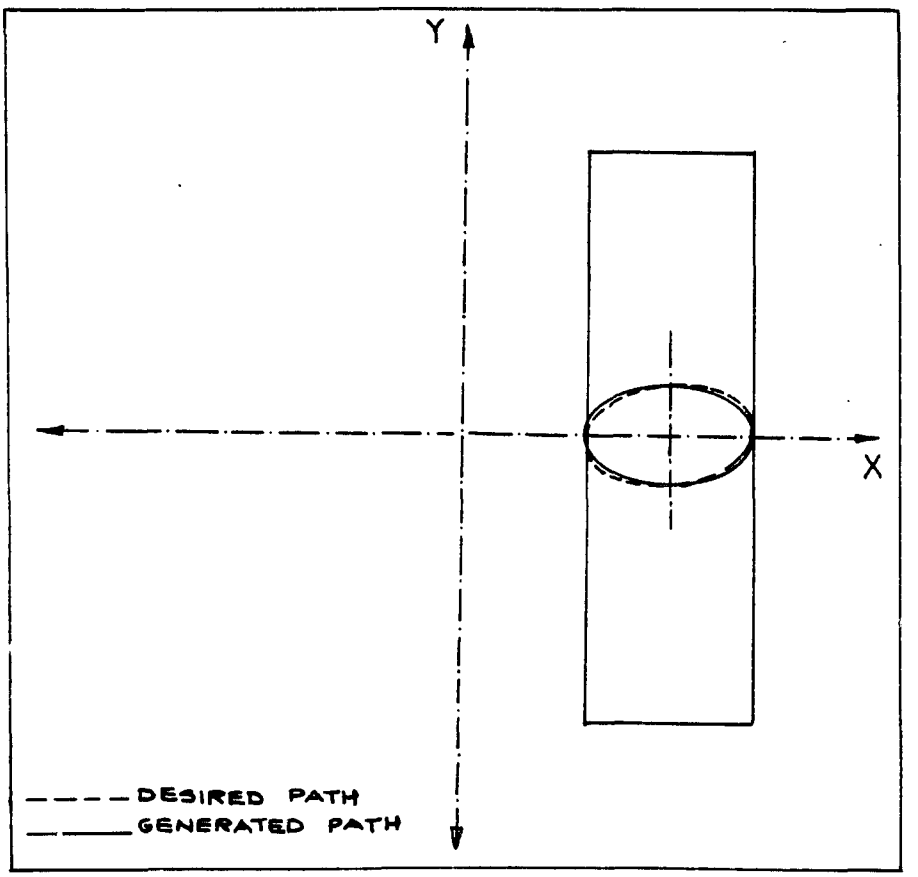


Figure 13 : PS300 SLIDE-3

acknowledged here that throughout the research, PS300 proved extremely helpful in improving the process of synthesis and in coming up with creative ideas.

CHAPTER 8

PROGRAMMING

This chapter describes the programming aspects of the research. Once the initial stage of mathematical derivations was over, software progressed simultaneously along with other theoretical enhancements. The details pertaining to PS300 programming and the use of various numerical methods have already been described in the previous two chapters. Here, we discuss only structuring and computer aided design part of the software package.

The FORTRAN part of the programming is carried out on a Data General, 32 bit, MV/10000 computer, using FORTRAN 77 compiler [8]. The program uses double precision mode of compilation to achieve high computational accuracy in the numerical methods used. The graphics part involves the use of the PS300 GRAPHICS LANGUAGE, the structured language detailed in chapter 7. Care has been taken in minimizing the requirements for storage and CPU time of execution and in producing a user-friendly, interactive graphics package.

8.1 Structure Of Robopt Program

The overall structure of the ROBOPT program is illustrated by means of a block diagram as shown in figure 14. The figure gives an idea about the CAD approach and also shows various phases of the program. For the input of 'desired curve' by the user, an option is provided for the medium to be used. Either the PS300 data tablet or a system generated parametric equations function routine may be used. For the input of design data , both for analysis and synthesis, a data file, a subroutine called INPUT and answers to terminal prompts are required. Once the optimization package receives the initial guesses for the design variables, the objective function is minimized and the optimal design variables are found. Finally, the full analysis is carried out for the known optimal actuator models. This analysis results are stored in an OUTPUT file. The manipulator motion is simulated on PS300 and depending upon the decision of the user, a redesign or a termination call is performed. CPU time for optimization is (T2-T1) and that for the final analysis is computed as (T3-T2). The total time required to run through one design cycle is (T3-T1+T4) where T4 is the CPU time for the interactive use of the graphics work station.

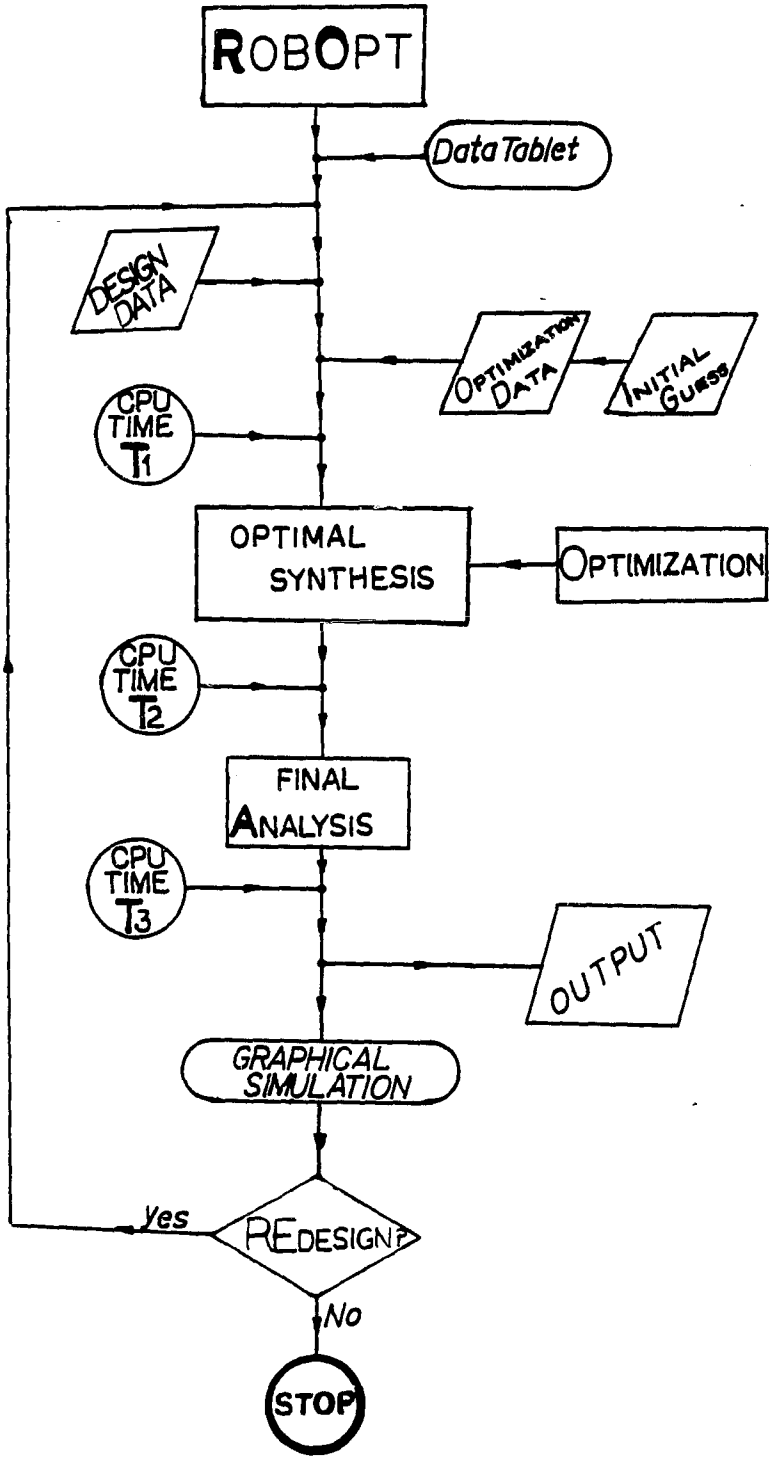


Figure 14 : ROBOPT program Block Diagram

8.2 Development Phases

The four main stages of software development according to chronological order are : (1) Analysis, (2) Graphical Simulation, (3) Synthesis, and (4) Final CAD approach.

8.2.1 Programming for analysis

This part of the program deals with the analysis of ROBOPT. First of all, the differential equations of motion are solved numerically when the design data and torque models are made known. Then position, velocity and acceleration analysis is carried out for the centers of gravity, joints and the end-effector point. The analysis results are stored in a file named OUTPUT. A main file ROB.F77 is created to execute the analysis package separately. This becomes particularly important in finding a suitable initial guess for the optimization package. The analysis package is used partially throughout the optimization phase but the whole package is used only after the process of optimization is over to yield the final analysis results. There are 15 routines in this package including one data file. Figure 15 contains the tree of subroutines in the analysis package.

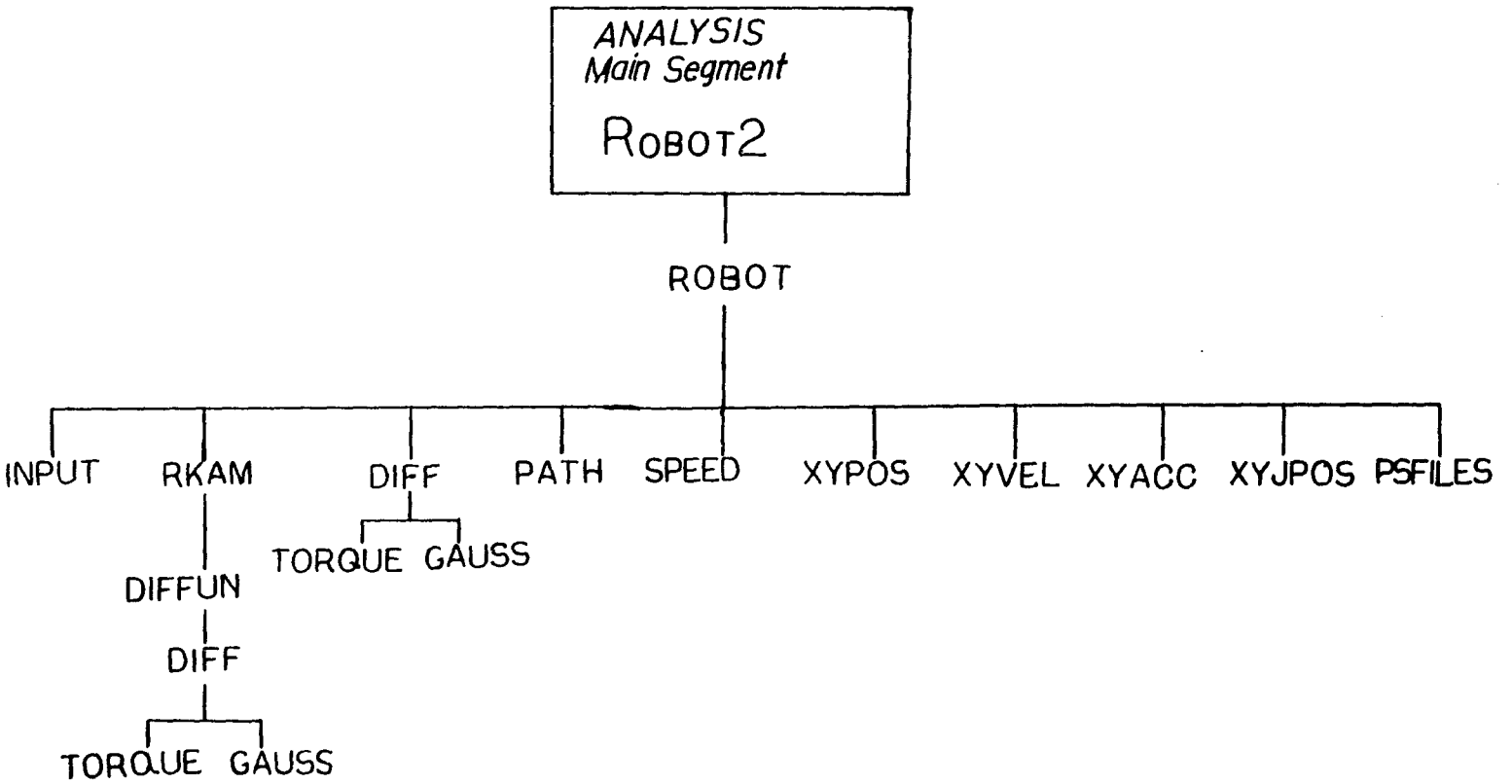


Figure 15 : Tree of Analysis Package Routines

8.2.2 Programming for graphical simulation

There are three kinds of files to generate graphical simulation. (i) FORTRAN files to generate PS300 files, (ii) PS300 files generated in the PS300 GRAPHICS LANGUAGE by the FORTRAN files of type (i), and (iii), permanent PS300 files in the PS300 GRAPHICS LANGUAGE created by using a system text editor.

The files of the first catagory are the subroutines written in FORTRAN 77. PSFILES is the subroutine that generates IO.PS, AA.PS, PR.PS, PSS.PS and STR.ERR.PS, the output files of the second catagory for PS300. The data within these files is problem dependent and hence they have to be generated each time a new problem is solved. The data from this files is used mainly for displaying INPUT, OUTPUT characteristic curves and frame-by-frame animation. The PSS.PS file contains data for the number of frames in animation and speed of animation that could be changed dynamically by sending different values.

Subroutine PSSEND opens a unit at a PS300 port and accepts a file name as an argument. This file could be either the one generated by the FORTRAN program or the parmanent file already existing in the user's directory. In short, this routine sends information to PS300 from the program files for display. Program TAB is written for an interactive use of the data tablet. TABREAD and ITOF are

supplementary routines. The rest of the PS300 files of catagory 3, with an extension .PS are permanent files for the display and transformation structures. Overall, 18 fortran and PS300 GRAPHICS LANGUAGE files are required for the graphical simulation.

8.2.3 Synthesis phase

The optimal synthesis technique (figure 16) needs a nonlinear programming package which in turn requires an objective function routine and a constraint routine to be supplied. The objective is to minimize the integrated structural error. Figure 17 shows a digrametic representation of the structural error concept. What points on the desired and the generated curves should be matched to find the errors? For this, equal arc length segment points on both the desired and the generated curves are used to find errors E(1) thru' E(N). The integrated structural error is then found by the sum of the squares of these errors and the errors due to difference in speed at these points. The following steps are involved in calculating the integrated structural error.

For the desired curve,

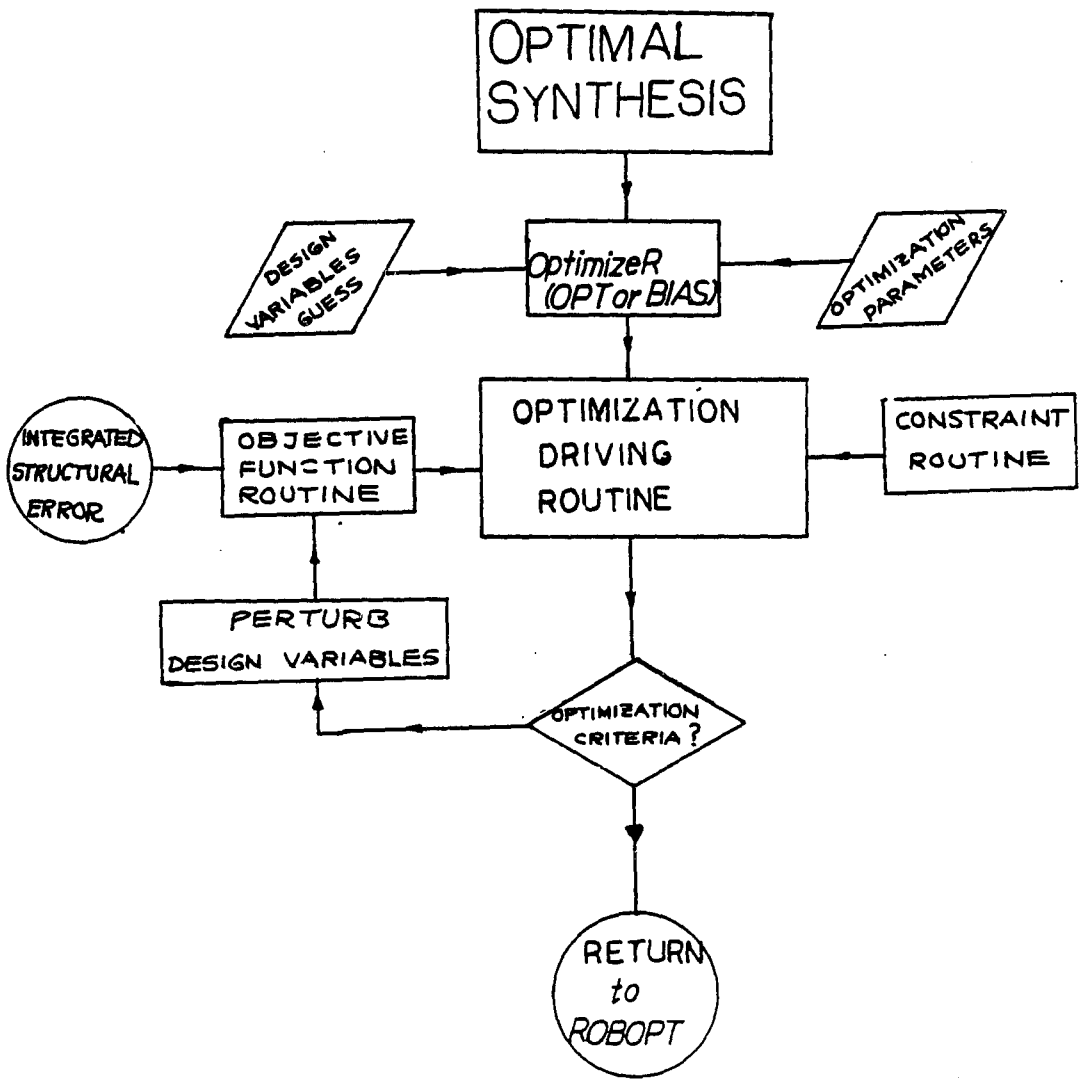


Figure 16 : Flow Chart of Optimal Synthesis Technique

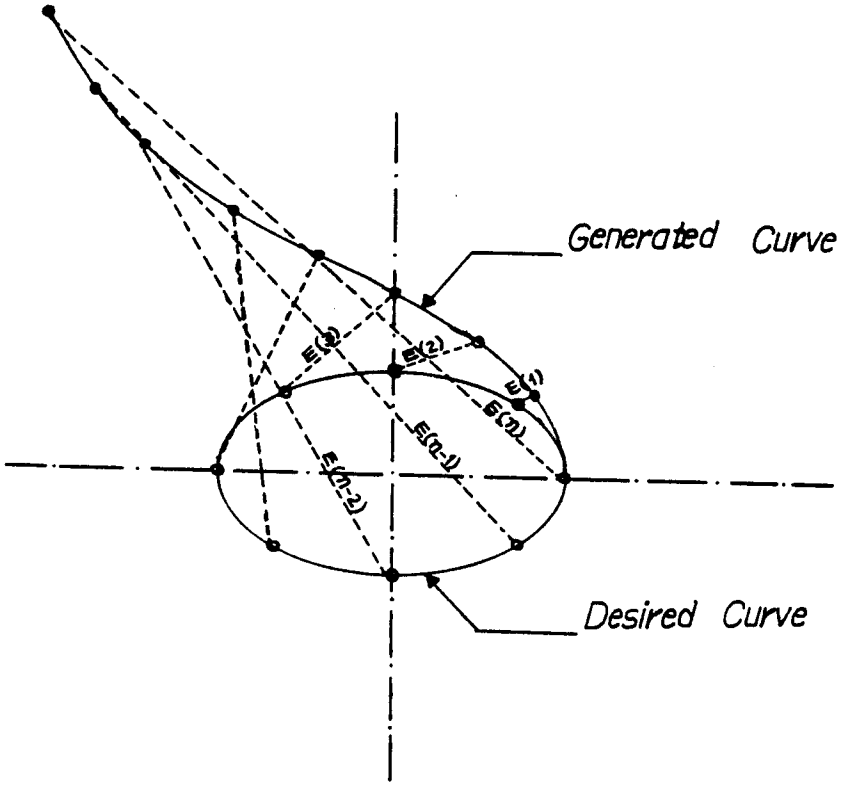


Figure 17 : Structural Error Concept

(i) Calculate the data points for interpolation (equal theta⟶arc length), that is, divide the curve into equal angular displacement segments and find out corresponding arc lengths and also the total arc length of the curve.

(ii) Carry out the inverse interpolation (Equal arc length segments⟶theta), that is, once the total arc length of the curve is known, divide it into equal arc length segments and find corresponding angular displacements.

(iii) Find the coordinates of 'desired points' corresponding to the angular displacements calculated by step (ii).

For the generated curve,

(iv) Find analysis final time 'TF' such that the arc length of the generated curve is almost equal to that of the desired curve.

(v) Find the data points for interpolation (equal time⟶arc length), that is, divide the time period 0.⟶TF into equal time segments and find out the arc length at each time segment by simple analysis.

(vi) Carry out the inverse interpolation (Equal arc length⟶time), that is, divide the curve into equal arc length segments and find out corresponding time periods.

(vii) Calculate coordinates of 'generated points' and speed of path generation at the time intervals calculated by step (vi).

For both the curves,

(viii) For the 'desired points' calculated in step (iii) and the 'generated points' calculated by step (vii), find out the integrated structural error.

Steps (i) thru' (iii) for the desired curve are calculated only once. The rest of the steps, (iv) thru' (viii), are cycled through, at each step of optimization that involves a change in design variables. This is required because the generated curve changes its shape and total arc length whenever there is a change in the design variables.

The optimization procedure tries to minimize the error function by an appropriate change in the design variables staying within the limits of the variable bounds and also satisfying the constraints. When the optimization convergence criteria is met, control is returned to the main program ROBOPT.

The optimal synthesis problem was executed by using both OPT [17], a generalized reduced gradient code, and BIAS [35], a penalty function method code. Better results were obtained by using BIAS for an unconstrained problem.

Reklaitis, Ravindran and Ragsdell [34] is an excellent reference for the study of design optimization (nonlinear programming) methods.

8.2.4 The Final CAD approach

Some of the design data is acquired through a user friendly interactive coding so that during the REDESIGN phase, this data may be varied to improve results. There is an option for the user to input his desired curve. Some design curves, like, circles, ellipses, straight lines, etc., can be represented by simple parametric equations. In such a case, the user may want to input his 'desired curve' by parametric equations in function routines. Else, the user shall use the PS300 data tablet to input his curve. The user also has to answer a few questions pertaining to the optimization package used. Finally, after looking at the simulation, there is again an option to redesign or to terminate the programme. The CAD approach can be enhanced drastically by providing more options. An improved CAD approach is suggested for further research in figure 18.

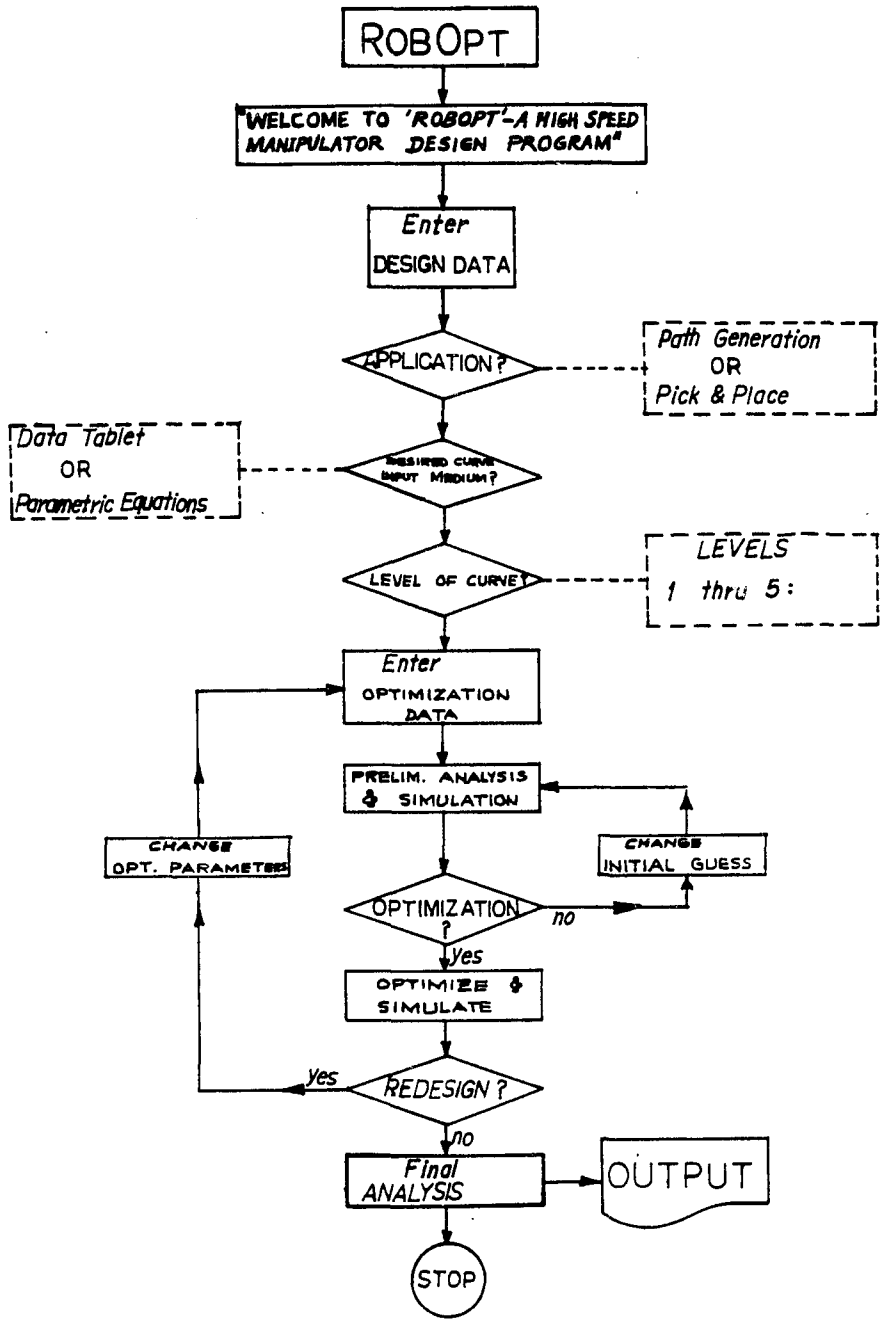


Figure 18 : Suggested Improved CAD Approach

8.3 Results & Conclusions

The ROBOPT software has been tested to generate a variety of curves including horizontal ellipses, vertical ellipses, circles, horizontal straight lines and vertical straight lines. It is realized that there are five levels of difficulty of the generated curves. (i) An absolute curve (conveyor static) generated at arbitrary speeds, (ii) a curve generated relative to the high speed conveyor at arbitrary speeds, (iii) an absolute curve generated at constant speed, (iv) a relative curve generated at constant high speed and a curve of level (iv) along with tight industrial constraints. It should be noted here that little 'insight' is required in selecting the initial guess for design variables as in this optimization problem formulation, amount of convergence is sensitive to the initial guess. Also, the design space has multiple solutions while executing an unconstrained problem. For example, if the initial guess is 12*10., the optimization will try to refine the fourier series around this guess, say , in a pattern 5., 8.4, 10., 12., 12.4, 10., 20., 8., 8., 12., 3., 0. If the initial guess is 12*20.,then the optimization will converge to a solution such as 10., 16.8, 20., 24.5, 24., 20., 40., 16., 16., 24., 6., 0. This is due to the fact that the angular displacements of the first two links of the arm

are proportional to the difference in torques (see the differential equations of motions). A rigorous testing of the five levels of curves is required.

The Results of a rather complex 'ellipse' curve are presented here now. The path generation problem was solved by using 'BIAS' [35] library. When the initial guess of 8*10., 4*5. was supplied, the optimization converged from a structural error of 15.37 down to .000771. The optimum design variables found were 4.15, 10.33, 16.57, 5.94, 8.08, 12.51, 17.10, 8.36, 6.68, 5.09, 3.21, and 7.80. The CPU time for optimization of this problem was 711 seconds. Thus, the optimum torque models to generate the 'desired ellipse' are:

$$\begin{cases} T_{12} = 4.15 + 10.33*COS(16.57*t) + 5.94*SIN(16.57*t) \\ T_{23} = 8.08 + 12.51*COS(17.10*t) + 8.36*SIN(17.10*t) \\ T_{34} = 6.68 + 5.09*COS(3.21*t) + 7.80*SIN(3.21*t) \end{cases}$$

The computer printouts for this 'ellipse' path generation problem follows. Figure 19 contains the design data (ROBOPT specifications). Resulting optimal torque histories and final structural errors are included in tables 4 and 5 respectively. Figures 20 through 22 shows the final results of other interesting curves produced through the path generation application. Figure 23 illustrates the concept of 'pick & place' application. Given the link lengths (L_1 ,L_2 ,L_3), constant value of one of the angles (say, θ_1) and the user desired position (XP,YP), two nonlinear equations in two unknowns (θ_2 and θ_3) are solved by using the

Newton-Ralphson method. As an optional, more refined approach, a system of three non-linear equations may be solved for all three joint variables (θ_1, θ_2, θ_3), given an additional data for the desired constant speed of path generation. Using the former approach (Method-1), the Newton-Ralphson iterations to reach the desired position (2.5, 0.0) are tabulated as shown in table 6. F(I)'s show the positional errors. Finally, some interesting results of a 'horizontal straight line' path generation via the above mentioned 'pick & place' concept are depicted in table 7. All the desired positions are achieved by providing the same initial guess for the Newton-Ralphson method. It should be noted here that with these values for the joint variables, optimal torque histories could be obtained by using the analysis package described earlier. An average execution time for a path generation problem was observed to be around 600 cpu seconds for a 20 segment curve, whereas, that for the pick & place application was .05 cpu seconds. ROBOPT was satisfactorily tested for both the applications with 20 test problems.

```
I N P U T   D A T A

MASS OF THE LINKS ARE
---------------------
         M( 1)=   .250E+00  (SLUG)
         M( 2)=   .500E+00  (SLUG)
         M( 3)=   .230E+00  (SLUG)
THE MOMENT OF INERTIA OF THE LINKS ARE
--------------------------------------
         I( 1)=   .100E+00  (SLUG*FT**2)
         I( 2)=   .178E+00  (SLUG*FT**2)
         I( 3)=   .650E-01  (SLUG*FT**2)
THE CENTERS OF MASS ARE LOCATED AT
----------------------------------
         R( 1)=   .860E+00  (FT)
         R( 2)=   .760E+00  (FT)
         R( 3)=   .604E+00  (FT)
THE LINK LENGTH ARE
-------------------
         L( 1)=   .150E+01  (FT)
         L( 2)=   .117E+01  (FT)
         L( 3)=   .833E+00  (FT)
THE INITIAL CONDITIONS ARE
--------------------------
         THETA( 2)=   .000E+00  (DEG)
         THETA( 3)=   .000E+00  (DEG)
         THETA( 4)=   .000E+00  (DEG)
         THETA DOT( 2)=   .000E+00  (RAD/SEC)
         THETA DOT( 3)=   .000E+00  (RAD/SEC)
         THETA DOT( 4)=   .000E+00  (RAD/SEC)
THE WEB VELOCITY IS
-------------------
         V0=   .000E+00(FT/SEC)
```

Figure 19 : Computer Printout of the DESIGN DATA submitted to 'ROBOPT' program

TABLE 4

OPTIMAL TORQUE HISTORIES

(through final analysis)

TORQUE HISTORIES:			
TIME	T12	T23	T34
.000E+00	.145E+02	.206E+02	.118E+02
.839E-01	.118E+02	.181E+02	.137E+02
.117E+00	.595E+01	.104E+02	.143E+02
.142E+00	.116E+01	.419E+01	.147E+02
.162E+00	-.245E+01	-.483E+00	.150E+02
.179E+00	-.501E+01	-.375E+01	.152E+02
.195E+00	-.667E+01	-.580E+01	.154E+02
.209E+00	-.755E+01	-.681E+01	.155E+02
.223E+00	-.776E+01	-.691E+01	.157E+02
.237E+00	-.736E+01	-.620E+01	.158E+02
.250E+00	-.641E+01	-.478E+01	.158E+02
.263E+00	-.504E+01	-.281E+01	.159E+02
.274E+00	-.340E+01	-.525E+00	.159E+02
.285E+00	-.164E+01	.189E+01	.160E+02
.295E+00	.190E+00	.436E+01	.160E+02
.305E+00	.205E+01	.682E+01	.160E+02
.315E+00	.392E+01	.928E+01	.160E+02
.324E+00	.579E+01	.117E+02	.160E+02
.334E+00	.771E+01	.141E+02	.160E+02
.345E+00	.971E+01	.166E+02	.159E+02
.357E+00	.116E+02	.189E+02	.159E+02

TABLE 5

Final Structural Errors

S T R U C T U R A L E R R O R :				
TIME	XPU	XPG	YPU	YPG
.0000	3.5000	3.5000	.0000	.0000
.0839	3.4498	3.4482	.2370	.2353
.1171	3.3133	3.3115	.4378	.4368
.1416	3.1189	3.1189	.5838	.5862
.1616	2.8917	2.8927	.6711	.6771
.1791	2.6500	2.6508	.7000	.7075
.1949	2.4083	2.4086	.6711	.6775
.2095	2.1811	2.1815	.5838	.5879
.2234	1.9867	1.9869	.4378	.4402
.2369	1.8502	1.8480	.2370	.2386
.2501	1.8000	1.7931	.0000	-.0002
.2627	1.8502	1.8403	-.2370	-.2413
.2743	1.9867	1.9780	-.4378	-.4449
.2852	2.1811	2.1758	-.5838	-.5902
.2953	2.4083	2.4061	-.6711	-.6744
.3050	2.6500	2.6500	-.7000	-.7004
.3146	2.8917	2.8936	-.6711	-.6708
.3241	3.1189	3.1228	-.5838	-.5864
.3340	3.3133	3.3219	-.4378	-.4419
.3450	3.4498	3.4511	-.2371	-.2272
.3567	3.5000	3.5000	.0000	.0000
STRUCTURAL ERROR FUNCTION =				.000771

TIME = Time Instance in Seconds

(XPU, YPU)= 'Desired' Position

(XPG, YPG)= 'Generated' Position

DESIRED PATH :

3.500000 ,	.000000
3.455951 ,	.278115
3.328116 ,	.529006
3.129007 ,	.728115
2.878116 ,	.855951
2.600001 ,	.900000
2.321886 ,	.855951
2.070995 ,	.728116
1.871886 ,	.529008
1.744050 ,	.278117
1.700000 ,	.000002

GENERATED PATH :

3.499990 ,	.000000
3.449990 ,	.286886
3.330277 ,	.526473
3.130113 ,	.734957
2.880967 ,	.862654
2.598324 ,	.904864
2.317096 ,	.859541
2.061753 ,	.732730
1.860256 ,	.529980
1.747916 ,	.266340
1.700000 ,	.000002

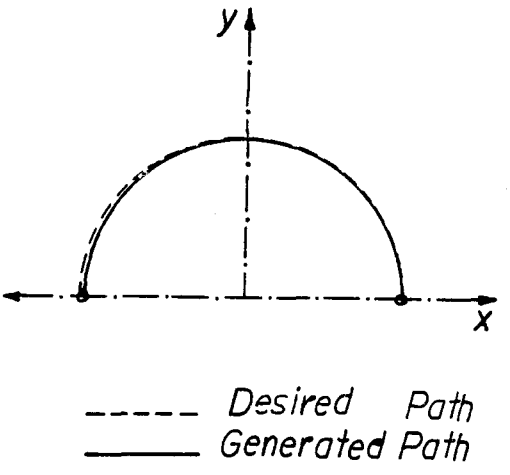


Figure 20 : 'Half Cicle' thru Path Generation

DESIRED PATH :

3.500000 ,	.000000
3.304011 ,	.000000
3.101267 ,	.000000
2.899644 ,	.000000
2.699747 ,	.000000
2.500289 ,	.000000
2.299851 ,	.000000
2.100015 ,	.000000
1.900022 ,	.000000
1.699987 ,	.000000
1.500003 ,	.000000

GENERATED PATH :

3.499990 ,	.000000
3.298307 ,	-.022306
3.116492 ,	-.008463
2.907211 ,	-.002890
2.707997 ,	-.001768
2.510114 ,	.000538
2.309893 ,	.005659
2.110988 ,	.010263
1.912300 ,	.006205
1.714676 ,	-.016999
1.500003 ,	.000000

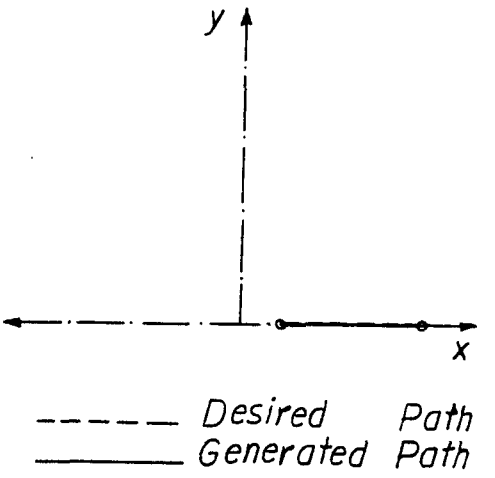


Figure 21 : 'Horizontal Straight Line' thru Path Generation

DESIRED PATH :

3.500000 ,	.000000
3.408575 ,	.267960
3.205966 ,	.470526
2.960816 ,	.620348
2.696886 ,	.734365
2.423051 ,	.822185
2.143380 ,	.889286
1.860072 ,	.938959
1.574490 ,	.973239
1.287564 ,	.993363
1.000003 ,	1.000000

GENERATED PATH :

3.499990 ,	.000000
3.387492 ,	.269121
3.192525 ,	.457768
2.947686 ,	.618533
2.692330 ,	.741520
2.420474 ,	.838042
2.142062 ,	.907669
1.858668 ,	.952524
1.572256 ,	.973776
1.285891 ,	.972204
1.000003 ,	1.000000

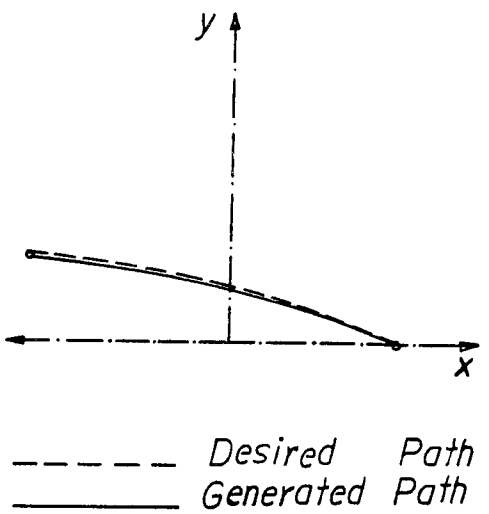
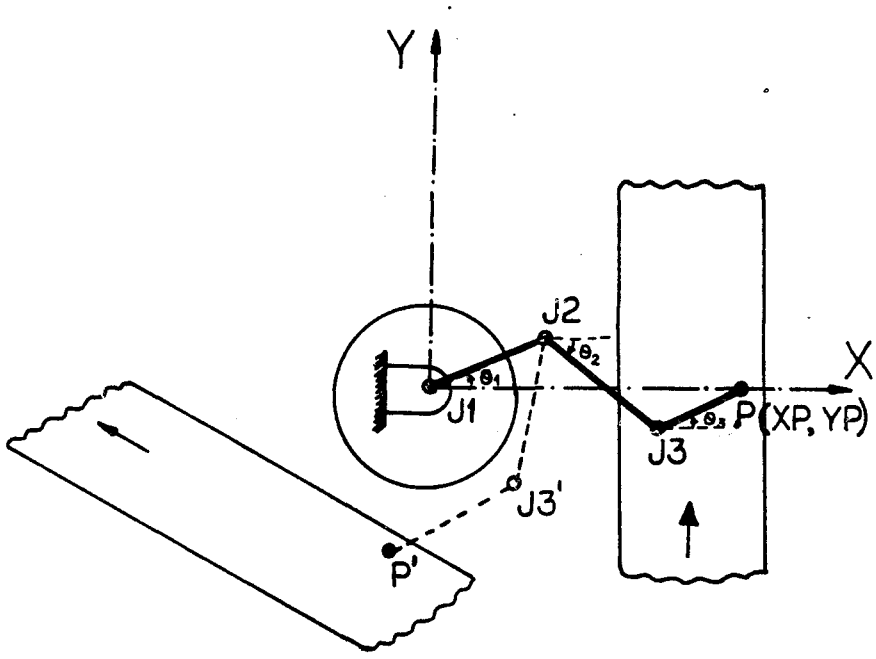


Figure 22 : 'Elliptical Arc' thru Path Generation



XP = DESIRED 'X' POSITION
YP = DESIRED 'Y' POSITION
SP = DESIRED SPEED

Method-1:

$$\begin{cases} XP = L_1 COS(\theta_1) + L_2 COS(\theta_2) + L_3 COS(\theta_3) \\ YP = L_1 SIN(\theta_1) + L_2 SIN(\theta_2) + L_3 SIN(\theta_3) \\ \theta_1 = \text{CONSTANT} \end{cases}$$

Method-2:

$$\begin{cases} XP = f(\theta_1, \theta_2, \theta_3) \\ YP = f(\theta_1, \theta_2, \theta_3) \\ SP = f(t, \theta_1, \theta_2, \theta_3) \end{cases}$$

Figure 23 : The concept of 'pick & place' and solution methodology

TABLE 6

Results for 'Pick & Place'

(Newton-Ralphson Iterations to achieve 'Desired' Position)

DESIRED POSITION : (XP, YP) = (2.5, 0.0)

INITIAL GUESS FOR ANGLES : θ_2 = -10.00, θ_3 = +60.00

ITER	X(I)'S = ANGLES	F(I)'S = ERRORS
0	-.1000E+02 , .6000E+02	.6448E+00 , .3493E+00
1	-.3865E+02 , .1123E+03	-.2152E+00 , -.2106E+00
2	-.2860E+02 , .1058E+03	-.1522E-01 , .7849E-02
3	-.2931E+02 , .1038E+03	.2368E-04 , -.3655E-03
4	-.2929E+02 , .1038E+03	-.6634E-07 , -.9243E-08

FINAL ANGLES : θ_2 = -29.29 , θ_3 = 103.8

CPU TIME FOR EXECUTION = .046 SEC.

TABLE 7

Use of 'pick & place' concept for Horizontal Straight-Line Path Generation (θ_1 = CONSTANT)

Constant Angle θ_1 = 0

INITIAL GUESS FOR ANGLES : θ_2 =10.00 & θ_3 = -60.00

DESIRED POSITION	FINAL ANGLES
(XP, YP)	θ_2 & θ_3
2.5 , 0.0	-29.29 & 103.8
2.6 , 0.0	-30.16 & 94.28
2.7 , 0.0	-30.16 & 85.72
2.8 , 0.0	-29.49 & 77.74
2.9 , 0.0	-28.26 & 70.05
3.0 , 0.0	-26.53 & 62.44
3.1 , 0.0	-24.27 & 54.66
3.2 , 0.0	-21.41 & 46.43
3.3 , 0.0	-17.75 & 37.25
3.4 , 0.0	-12.73 & 25.93
3.5 , 0.0	-.20 & .40

CHAPTER 9

CLOSURE

The introduction of ROBOPT through this research is the start to a new generation of robots. Recapitulating the research approach, a computer controlled high speed planar manipulator is developed. The computer aided design technique simplifies the optimal synthesis to a large extent. Torque model synthesis is carried out using a nonlinear programming package. These precalculated torque histories are supplied back in order to regenerate the desired path. Thus it can operate in real time applications. Also, the same robot may be used to carry out the pick & place applications. The robot operation could be highly simplified by dedicating a computer or a microprocessor that will control the robot by means of an 'easy to use' macro command language.

9.1 Robotic Development Philosophy

Eventhough the industrial robots today prove their compatability and productivity, a lot more work needs to be done before thay will be accepted universally both by human philosophy and industrial psychology.

Rather than spending an immense amount of energy, time and money after developing sophisticated control and sensing capabilities, there is a need to exploit the present state-of-art of technology for robotic development today. Nowadays, the more popular forms of robots are the 'humanoids'. Let us remember that the robots are for productivity, the robots are for safety, the robots are for 'quality of life'. All I need to say is, let us not be carried away from our fundamental goal of robotic development, the goal to spare the humans from repetitive, laborious tasks so that the human minds can be utilized for more creative thoughts.

9.2 Recommendations For Future Enhancements

Robotics is the area of promising future. At present, 5 % of the assembly lines of the US industry incorporates robots. It is forecasted that by the turn of the century, more than half of the assembly lines will be employing robots. The progress in other industrial operations is also encouraging.

There is an immediate need to produce high speed manipulators to work in real-time environments. In this thesis research, optimal synthesis and dynamic simulation techniques are applied to design such a robot. There are more needs to develop the concept of KINETO-ELASTODYNAMICS, advanced control theories, efficient actuators and special purpose hands for high speed robots. Each high speed robot should go through a rigorous testing of vibrations, inertial loading and elastic deformations at rated high speeds.

During the formulation of ROBOPT theory and development of software, the objective function in the optimal synthesis technique was considered to be the summation of both positional and velocity errors. For constant, high speed path generation requirement, "trade-off" was observed between positional accuracy and constant velocity maintainance. Also, in the presence of tight constraints and variable bounds, the approach becomes too much iterative and sometimes fails to yield the required accuracy. A more 'positive' approach, particularly for constant velocity requirement, is recommended. That is to say, rearrangement of the differential equations of motion to incorporate the constant speed requirement shall make optimization efficient. The optimal synthesis concept could be improvised to produce an EXPERT SYSTEM for the robots of

the future. This will eliminate the dependency of a problem on the initial design variables guess, eliminating the 'insight' factor from a design cycle.

An improved CAD approach is recommended in the form of the block diagram of figure 21. Also, three dimensional wire-frame model for ROBOPT can be easily generated to make dynamic simulation more powerful to visualize the robot motions.

9.3 Epitome

In today's competitive industry, a challenge for productivity and efficiency must be met with by intelligent application of computer aided design to robotics. A sincere effort is made towards this goal through the introduction of ROBOPT.

LIST OF REFERENCES

1. Ardayfio, D.D. and Pottinger, H.J., "Computer Control of Robotic Manipulators", Mechanical Engineering, August 1982.

2. Atrobolevsky, I.I., "General Problems in Theory of Machines and Mechanisms", Mechanisms and Machine Theory, Vol. 10, pp. 125, 1972.

3. Barrett, P.E., "High Speed, High Precision Robotics", Project Report, Purdue University, 1981.

4. Bhagat, B.M. and Willmert, K.D., "Finite Element Vibrational Analysis of Planar Mechanisms", Mechanisms and Machine Theory, Vol. 11, pp. 47, 1976.

5. Bhatt, V.D., "User's Manual PS300", Design Optimization Laboratory, University of Arizona, 1983.

6. Boykin, W.H. and Diaz, R.G., "Robotic Actuators: A Technology Assessment", Proceedings of International Computer Technology Conference, Vol. 1, 1981.

7. Burckhardt, C.W. and Helms, D., "Some General Rules for Building Robots", Third Conference on Industrial Robot Technology, 1976.

8. D.G.," FORTRAN 77 Reference Manual", Data General Corporation, #093-000162-02, 1983.

9. Engelberger, J.F., "Robotics in Practice", 1983.

10. E & S, "PS300 User's Manual", Evans & Sutherland Computer Corporation, E & S #901172-007 P4, Version P4.V01, 1982.

11. E & S, "PS300 Programmer Training Class Notebook", # 901172-107 NC, 1982.

12. E & S, "PS300 Application Notes", # 901172-026, 1982.

13. E & S, "PS300 Setup and Operation Guide", # 901172-026, 1983.

14. E & S, "User's Manual for Host Resident PS300 I/O Subroutines", # 901172-064 P5, 1983.

15. Freund, E., "Direct Design Methods for the Control of Industrial Robots", Computers in Mechanical Engineering, April 1983.

16. Fulton, R.E., "Using CAD/CAM to improve Productivity: NASA IPAD Project Approach", Mechanical Engineering, Nov. 1981.

17. Gabriel, G.A. and Ragsdell, K.M., "OPT Users Manual", Version 2.0, DOL, University of Arizona, 1984.

18. Gerald, C.F., "Applied Numerical Analysis", Addison-Wesely Publishers, Second Edition, 1980.

19. Gevarter, W.B., "An overview of Artificial Intelligence and Robotics, Volume II: Robotics", National Technical Information Service, PB 82-204439, 1982.

20. Greenberg, M.D., "Foundations of Applied Mathematics", Prentice-Hall Inc., 1978.

21. Heer, Ewald, "Robots and Manipulators", Mechanical Engineering, Nov. 1981.

22. I.R., "Industrial Robots of the World", The Industrial Robot, March 1983 - Sept. 1983.

23. Kincaid, D.R., "RKAM User's Manual", University of Texas.

24. Kretch, S.J., "Robotic Animation", Mechanical Engineering, Aug. 1982.

25. Lowen, G.G. and Jandrasits, W.G., "Survey of Investigations into Dynamic Behavior of Mechanisms containing links with Distributed Mass and Elasticity", Mechanism and Machine Theory, Vol. 7, pp. 3, 1972.

26. Makino, H., "A Kinematic Classification of Robot Manipulators", Third Conference on Industrial Robot Technology & Sixth International Simposium on Industrial Robots, paper F2, 1976.

27. Newman, W.M. and Sproull, R.F., "Principles of Interactive Computer Graphics", McGraw-Hill Book Company, 1979.

28. Paul, Burton, "Kinematics and Dynamics of Planar Machinery", Prentice-Hall Inc., pp. 5, 1979.

29. Paul, R.P., "Mechanical Manipulators: Mathematics, Programming, and Control", MIT Press, 1981.

30. Paul, R.P., "Manipulator Cartesian Path Control", IEEE Transactions on Systems, Man, and Cybernetics, Nov. 1979.

31. Paul R.P., "Modelling, Trajectory Calculation and Servoing of a Computer Controlled Arm", Stanford Artificial Intelligence Laboratory, Stanford University, AIM 177, 1972.

32. Paul, R.P.C., Walker, M.W. and Luh, J.Y.S., "On-line Computational Scheme for Mechanical Manipulators" Journal of Dynamic systems, Measurement and Control, ASME Transactions, Vol. 102, June 1980.

33. Ragsdell, K.M., "MELIB Users Manual", Design Optimization Laboratory, University of Arizona, 1984.

34. Reklaitis, G.V., Ravindran, A. and Ragsdell, K.M., "Engineering Optimization Methods and Application", John-Wiley & Sons, 1983.

35. Root, R.R. and Ragsdell, K.M., "BIAS Users Manual", Version 2.0, Design Optimization Laboratory, University of Arizona, 1984.

36. Roth, Bernard, "A table of Contemporary Manipulator Devices", Robotics Age, May 1983 - Jan. 1984.

37. Roth, Bernard, "Robots State of the art in regard to Mechanisms Theory", Journal of Mechanisms, Transmissions and Automation in Design, ASME transactions, Vol. 105, March 1983.

38. Sandor, G.N. and Imdad Imam, "A general Method of KINETO-ELASTODYNAMIC Design of High Speed Mechanisms", Mechanism and Machine Theory, Vol. 8, 1973.

39. Seering, W.P., "Directions in Robot Design", Journal of Mechanisms, Transmissions, and Automation in Design, Transactions of the ASME, Vol. 105, March 1983.

40. Suh, C.H. and Radcliffe, C.W., "Kinematics & Mechanisms Design", John Wiley & Sons, 1978.

41. Vukobratovic, M., "Engineering Concepts of Dynamics and Control of Robots and Manipulators", Proceedings of International Computer Technology Conference, Vol. 1, pp. 212, 1981.

42. Whitney, D.E., "The Mathematics of coordinated Control of Prosthetic Arms and Manipulators", Journal of Dynamic Systems, Measurement, and Control, ASME Transactions, Dec. 1972.