ERRATA

p 12, l 19: : $\pm$ .025 → $\pm$ 0.25

p 14. awx 1.2.1, l 11: accompanied → accomplished

p 51, l 9: transverse → transpose

p 56, equ (3.17): X, Y, Z, → x, y, z

p 77, equ (4.10): $p_z = S_{234}l_4 + S_{23}l_3 + S_2l_2 + t[S_{234}S_5]$

p 86, equ (4.32): $\theta_5 = \text{atan } 2[S\alpha - \phi C\sigma_1 C\alpha - \phi]$

p 86, equ (4.33): $\theta_6 = \text{Atan } 2[S_5 S\sigma_1 C\sigma] + \gamma$

p 95, sec 4.4.1, l 18: The value of θ_1 → The values of θ_1 and θ_5

p 100, l 1: (4.??) → (4.95)

p 100, l 2: given → is given

p 100, equ (4.100): $\Delta\theta_i^2$ → $\delta\theta_i^2$

p 124, l 23: the one → one

p 141, l 15: x → $\dot{x}$

p 141, l 18: x_{n+1} → $\dot{x}_{n+1}$

p 144, l 11, l 14: $\underline{\omega}$ max → $\dot{\underline{\omega}}$ max

p 144, equ (5.60): $\underline{x}$ → $\dot{\underline{x}}$

p 156, equ (6.1): $f_{d,i} = m_i\ddot{r}_i$

p 156, equ (6.2): $N_{d,i} = I_i\ddot{\theta}_i + \dot{\theta}_i x\ I_i\dot{\theta}_i$

p 156, l 5: r_i → $\ddot{r}_i$

p 156, l 6: θ_i → $\dot{\theta}_i$

p 156, l 7: θ_i → $\ddot{\theta}_i$

p 157, equ (6.5): $F_i = f_{d,i} + F_{i,i+1} - mg$

p 160, l 5: assambly → assembler

p 178, l 5: 6800 → 68000

p 183, l 24: MC6000 → MC68000

p 183, l 24: MC6020 → MC68881

p 183, l 23: delete: or NS16032

p 184, l 5: [Darzel 19??] → [Dalzelle 1981]

COMPLEX MOTIONS WITH AN ANTHROPOMORPHIC ROBOT

by

PHILIP LLOYD NICCOLLS

A thesis submitted for the degree of

DOCTOR OF PHILOSOPHY

of the University of London

and also for the

DIPLOMA OF IMPERIAL COLLEGE

August 1984

Department of Mechanical Engineering

Imperial College of Science and Technology

London SW7 2BX

To My Parents

ABSTRACT

The work presented in this thesis can be divided into three areas: design of a robot; structuring of the robot control problem and the design of fast control algorithms; and the implementation of these algorithms using a distributed intelligence multi-processor system.

The section dealing with robot design, considers the major criteria for choice, the standard options available, and presents a robot designed to be simple and robust.

To enable the implementation of a distributed intelligence controller, the author has structured the robot control problem into semi-autonomous tasks. These include spatial vector profiling, interpolation, frame and co-ordinate transformation, and joint servo control. These areas have been analysed in detail and the author has devised new algorithms to markedly reduce the computational task of robot control.

The algorithms for robot control have been implemented using a distributed microcomputer system. The controller employs a mixture of relatively cheap 32/16 bit and 8 bit microprocessors. This control architecture provides a powerful and flexible robot controller, enabling a vast range of complex motions to be generated and executed by the robot.

ACKNOWLEDGEMENTS

First many thanks to the Science and Engineering Research Council and Baker Perkins Ltd. for their sponsorship of this work by means of a CASE award. Much help has also been given by RD Projects Ltd. and, Hazmac Handling Ltd., specifically with the mechanical design.

People within the Applied Mechanics and Nuclear Power Sections of Imperial College, and employees of RD Projects Ltd. have given invaluable help with this project. Whilst it is perhaps unfair to name individuals, Dr. John Mason, Dr. Ali Pak, and Dr. Robin Curtis have been particularly influential.

Crucial support has come from my supervisor Dr. Colin Besant, whose help and guidance has been instrumental in this work. I am indebted to his help.

Finally, and by no means least, thanks are also due to Kiki who has had the most difficult job of typing this thesis, a most thankless task.

LIST OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

1.1 THE INDUSTRIAL ROBOT

In the early 1960's George Devol and Unimation Incorporated introduced the first industrial robot. The basic idea was to build a machine that was flexible enough to do a variety of jobs automatically: a device that could be easily taught or programmed, so that if the part or process changed, the robot could adapt to its new job without expensive retooling. This contrasts markedly with the traditional concept of "hard" automation, whereby plant and equipment is dedicated to one specific task. It was the combination of a computer and a flexible manipulator that has helped open the door to new methods of manufacturing.

Dr. James S. Albus in a recent book, [Albus 1979], on the effect of computers and robots, wrote, "The human race is now poised on the brink of a new industrial revolution which will at least equal, if not far exceed, the first Industrial Revolution in its impact on mankind. The first Industrial Revolution was based on the substitution of mechanical energy for muscle power. The next industrial revolution will be based on the substitution of electronic computers for the human brain in the control of machines and industrial processes".

1.1.1 Basic Robot Elements

The following are the three basic components of an industrial robot, [Saveriano 1980]:

(i) Controller: The robot controller functions as the co-ordinating system of the robot. It can be any programmable device from a rotary drum switch to a full computer. In sophisticated industrial robots, the control computer is capable of a level of "artificial" intelligence and not only runs the robot through its programmed moves, but also integrates it with ancillary machinery, equipment and devices. The controller can also monitor processes and can make decisions based on system demand while at the same time reporting to a supervisory control.

(ii) Manipulator: The manipulator consists of the base and arm of the robot, including the power supply, usually hydraulic, electric or pneumatic. The manipulator is the component that provides movement in any number of degrees of freedom. The manipulator's movement can be described in relation to its co-ordinate system, which may be cylindrical, spherical, anthropomorphic, etc. Depending on the controller, movement can be servo or non-servo controlled and can be a point to point motion or a motion along a specified continuous path.

(iii) Tooling: The hand or gripper, sometimes call the "end effector" can be a mechanical, vacuum, or magnetic device for part handling. It can incorporate levels of compliancy to accommodate any slight misalignment. This can be in the form of passive compliance, whereby any correction is provided locally, or active compliance where sensors provide additional positional information for the robot controller.

1.1.2 Types of Industrial Robot

As varied as the definition of "robot", there are many ways to classify different types of robots. Under the broad classification of

industrial robots, categories often reflect the kind of work the robot is assigned to do, such as spray painting, welding, assembly, material handling etc. Three categories of robot can be defined:

(i) Simple Robots: Simple robots are also called "pick and place" devices and "limited sequence" manipulators. Simple robots are perhaps the most underrated and underutilized robots. These low cost, easy to maintain, fast and accurate devices can dramatically increase productivity in medium and long-run production industries.

Normally, these devices are restricted to three or four non-servo degrees of freedom. Mechanical stops are used on each axis to set the amount of travel; this is usually only two positions, i.e. up/down, right/left, in/out. Because they are very limited in the number of moves available to the manipulator, simple robots are very depedent on support equipment such as bowl feeders and part presenters. A general rule of thumb in robotics is that the higher the intelligence of the controller, and the greater the programmability and number of moves of the manipulator and tooling, the less dependent the robot will be on support equipment.

Simple robots are usually air-operated, repeatable to ±.025mm or better, and can operate as fast as a cycle per second. These robots cost anywhere from £1500 to £7000.

(ii) Medium Technology Robots: Medium technology robots have a greater memory capacity and are easier to teach than simple robots. Such robots have four to six degrees of freedom and are servo-controlled in most of their axes of movement.

Medium technology robots are usually used for single machine load/unload type jobs and are not capable of continuous path operations required for welding and spray painting applications. There are many jobs in manufacturing today that could be automated by using medium technology robots. Such units can cost from £7000 to £15000 and generally have a repeatability of ±1mm.

(iii) Sophisticated Industrial Robots: Sophisticated industrial robots are at the leading edge of manufacturing technology. These robots possess highly flexible and programmable manipulators and utilize controllers that exemplify the higher levels of artificial intelligence used in industrial automation. Such controllers can be interfaced with sophisticated sensory and inspection devices and also enable the robot to be taught even the most complex of jobs with relative ease. The sophisticated industrial robot has the capability of being integrated into a myriad of computer-controlled work cells and manufacturing systems.

Sophisticated industrial robots have a large on-board memory, capable of multiple programmes and the ability to change programs automatically, depending upon the requirements of the work cell or system in which they are working. These machines are easily programmed by pendant, terminal keyboard, or off-line programming, or any combination of the three. Limited voice control is becoming available. High level robot programming languages and software are being used. Sophisticated robots' controllers are usually micro or mini-computers; program storage can be on any number of available media. The manipulators have five or more degrees of freedom and are fully

programmable in all axes: these manipulators can operate either point to point or continuous path. Small sophisticated robot arms such as the Unimation PUMA have repeatability of ±.05mm carrying loads of a few pounds, larger sophisticated robots, such as Cincinnatic Milacron's T3 are repeatable to ±.05mm even when carrying heavier payloads over greater distances. Sophisticated robots cost in the range of £20000 to £70000 depending on configuration.

1.2 THE ROBOT CONTROL SYSTEM

1.2.1 The Development of Robot Controllers

The major difference between "hard" automation and a robotic system, is the fact that the latter is re-programmable. Hard automation, when installed, continuously carries out essentially the same procedure. It has limited ability to react to the environment and requires major conversion to accommodate a different product. The versatility of a robot is obtained by a multi axis mechanical configuration and the robot controller. The task sequencing of the earlier "first generation" robots was achieved by the use of hardware logic. The control of these robots, although satisfactory, was rather primitive and restricted. The advent of the microprocessor was important in that the logic tasking could be accompanied in software rather than hardware. The earlier microprocessors (four and eight bit units) had limited processing power, and as such realised only marginal increases in the overall control of industrial robots.

Industrial robots controlled either by hardware logic, or early microprocessors, assume importance in that they were the first generation of non-dedicated automation. They were however simple point to point motion devices, and programming them for anything other than simple pick-and-place tasks proved tediously time consuming. The amount of on-line programming, and perhaps more important the lack of facilities to include environmental sensing, restricted their usage for more sophisticated industrial tasks. Knowledge of the local environment, through the use of transducers, is one of the prerequisites for increased robot intelligence.

The features required of a modern robot control system are illustrated in Figure 1.1. The top level interfaces with both command devices and environmental sensing units. It is then necessary to interpolate with respect to time, thus defining the robot trajectory; to transform the trajectory into the base frame co-ordinate system or work space of the robot; and to convert from the trajectory in work space to the joint demands of the robot. It is then necessary to servo control each joint of the robot.

The development of LSI circuits provided the opportunity to enhance the robot control system. The lower end of the minicomputer market, (which was later to become the high end of the microcomputer market), encompassed processing units at a relatively low cost yet sufficiently powerful to significantly improve the control of the robot and incorporate some of the features described above. The major advantage obtained by the use of these processors was continuous path motion, previously difficult due to the numerical capabilities required to

accomplish this form of motion. The use of higher speed processors also enabled off-line programming and instigated high level robot programming languages.

The low cost of microprocessors in general and, the recent development of the more powerful 16/32 bit processors, have enabled considerably more sophisticated systems to be developed for robot control.

1.2.2 A Distributed Intelligence Control System

In the following section the term microprocessor is taken to mean a single processor chip, and microcomputer refers to the combination of processor chip and peripheral chips (memory, I/O, etc). A multiprocessor system is one which combines a number of processing chips, and a multi-microcomputer system is a subset of multiprocessor systems which combines semi-autonomous microcomputer units each with their own peripheral chips.

There are four principle levels at which improvements in performance of a computer system are possible, [Enslow 1977]:

(i) Devices and circuits; the basic hardware speed.

(ii) Function implementation; the algorithms implemented in the functional units ie. in the central processor, memory, and I/O.

(iii) System architecture; the topology for the interconnection of the functional units.

(iv) System software; the scope, speed and efficiency of the operating system and other supporting software.

Of the four levels mentioned, this section discusses improvements at the level of systems organisation, dealing specifically with a particular class of systems; multiprocessors.

The extremely low cost of microprocessors has led to the analysis and development of systems which previously would have involved only a single processor. These units incorporate more than one processor to accomplish the system task. Two overlapping approaches define multiple processor systems; multiprocessor and distributed intelligence.

(i) Multiprocessor systems improve cost performance by the use of interconnecting processors in a tightly coupled manner, such that processors share resources and function under a single operating system

(ii) Distributed intelligence systems use a number of processors (possibly in a multiprocessor configuration) each handling a number of different, but usually related, tasks.

Of the two options, multiprocessor systems often involve complex operating systems with high software and hardware overheads. This, coupled with the limitations for hardware modularity and expandability, suggests that the distributed intelligence concept would prove more suitable for the robot control problem.

The Distributed Intelligence Microcomputer Systems (DIMS) concept divides the control problem into small definable subsystems, each controlled by an individual processor. DIMS differs from multiprocessing in that multiprocessing uses many processors to handle one task, while distributed intelligence uses one processor per task of the system [Anderson 1975]. The DIMS concept distributes the intelligence throughout the system by means of the microcomputers, each acting semi-autonomously and communicating with other elements in the system. Each microcomputer has a dedicated task, idealy each task being isolated from that of any other microcomputer in the DIMS, ensuring hardware and software isolation.

There are a number of advantages which may be realised by the use of a DIMS approach, [Infotech 1977], compared to a single central processing unit robot controller:

(i) By the use of a number of relatively cheap microprocessors, it is possible to gain a large measure of computing power from an assembly of low cost, mass produced components.

(ii) Isolated tasks, each controlled by a single microcomputer, are a key design feature of the DIMS concept. When this is achieved the system reliability is very high. Another result of this autonomy is a relatively simple communications system, handling only limited data flow and synchronisation. Since each control element is independent, the system can be designed to ensure that the failure of a microcomputer is unlikely to corrupt the whole control system.

(iii) The DIMS approach allows expandability and adaptability, [Maekawa 1980], in that processing power may be easily added in increments. This is achieved in two ways. First; the upgrading of software (easily accommodated as the control functions are independent). Second; hardware expandability, such as the addition of another degree of freedom to the control system, or the upgrading of a particular card with one having greater processing power.

(iv) Reduced system cost and reduced complexity are achieved when standardisation, and interchangeability of microcomputers are possible. This feature must apply to both hardware and software to be truly effective.

(v) The problems of service and maintenance of the robot controller are simplified when using the DIMS approach. The controller can perform self-checking operations, thus allowing rapid evaluation of problems and providing guidance to service personnel in diagnosing and correcting problems. Because boards and software are of the same design debugging is possible by board swapping. Also a complete spare can be stored on site and installed quickly, thus reducing machine downtime at relatively little extra cost to the robot user.

The DIMS concept is considered in more detail in Chapter 7.

1.3 THE MECHANICAL ARM

1.3.1 Degrees of Freedom

Most industrial robots today incorporate five, six or seven degrees of freedom (dof). An indication of the added versatility each degree provides can be given, building from a single joint. A single dof can be thought of as a ram, which in a spherical polar coordinate system is the r-axis. Rotation in a vertical plane, and rotation in a horizontal plane add the ψ and θ axes. It may be envisaged that in three dimensional, (3-D), space three degrees of freedom are sufficient, and this is true merely to locate a point in space. However the orientation of a 3 dof robot is fixed. In 3-D space there are in fact, three position variables and three orientation variables. Each space variable, whether position or orientation requires at least one dof on the robot. For a number of applications a robot with five dof will suffice. This can be achieved by careful positioning of the robot and ancilliary equipment, (eg: machine tool loading), or where the sixth dof is redundant, (eg: welding). A six dof robot can achieve near full 3-D space manipulation, the restrictions being physical restraints.

A robot with greater than six dof exhibit kinematic redundancy and require additional criteria for axis movement. Often the added degrees of freedom are obtained by making a robot mobile, usually a linear motion in one or two directions.

1.3.2 Geometric Configuration

Two main linkage mechanisms appear in the designs of the robot (excluding wrist): linear and rotary. Various combinations of these result in four standard configurations: cartesian, cylindrical polar,

spherical polar, and anthropomorphic, (see Figure 1.2). These are termed the primary axes, and can be thought of as the "arm" of the robot. The robot wrist or secondary axis configuration, is usually a combination of rotary motions. The major variations reflect the number of axes of rotation that are coincidental, and the sequence and relative direction of each joint axis rotation.

1.3.3 Robot Drives

Three general methods of robot drive are currently employed in robotics: electric, hydraulic, and pneumatic. The benefits and limitations of each reflect the facility of power transmission to the prime mover, the power to weight ratio, and the controllability of method chosen. Electric motion require small leads to motor but have a relatively low power to weight ratio. Hydraulics on the other hand have a high power to weight ratio but suffer from bulky fluid transmission pipes, and costly, bulky power packs. Pneumatics are intermediate in the power/weight factor, and are a clean form of power transmission. However position control is much more difficult, especially static positioning.

Relating the robot drive to typical applications, pneumatics are used more for pick and place robots where mechanical stops can determine the static position. The choice between electric or hydraulic power depends on the weights which must be held and moved by the robot. Lighter weights, as for example in light assembly, are more suited to electrics. Hydraulics on the other hand are usually necessary when the robot must lift loads in excess of 100kg. There are however exceptions to this generalisation. For example the IBM 7565 is a low payload hydraulic robot.

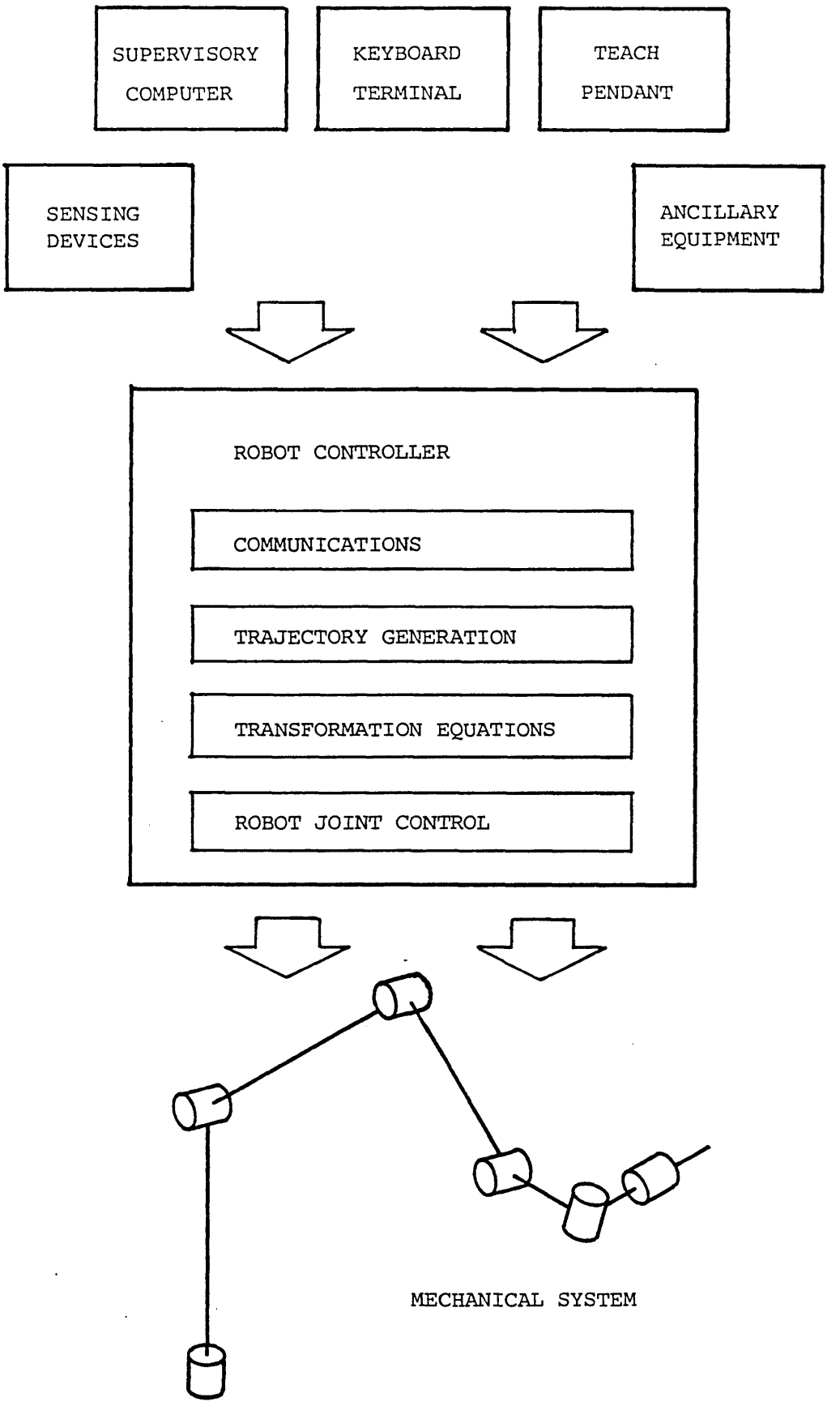
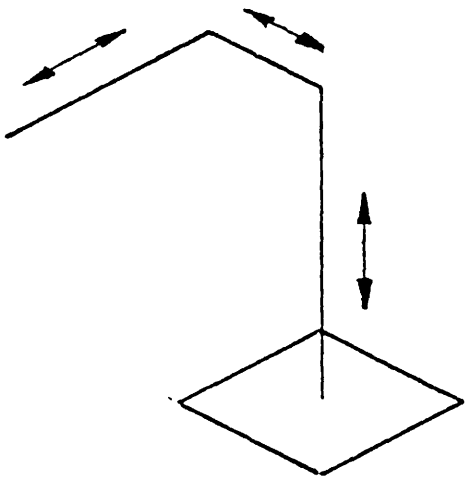
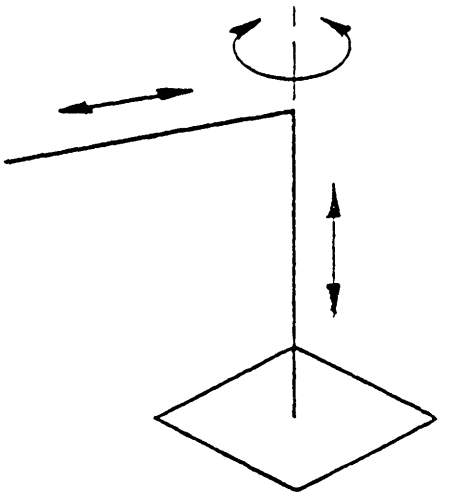


Figure 1.1 Features Required of a robot controller



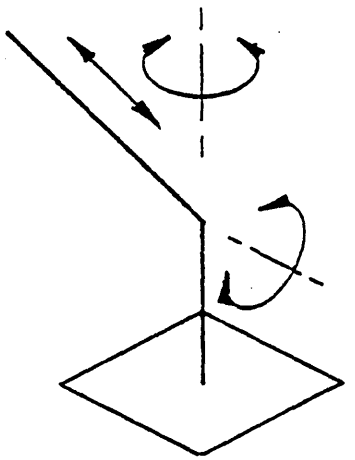
(i) CARTESIAN



(ii) CYLINDRICAL POLAR

(iii) SPHERICAL POLAR

(iv) ANTHROPOMORPHIC



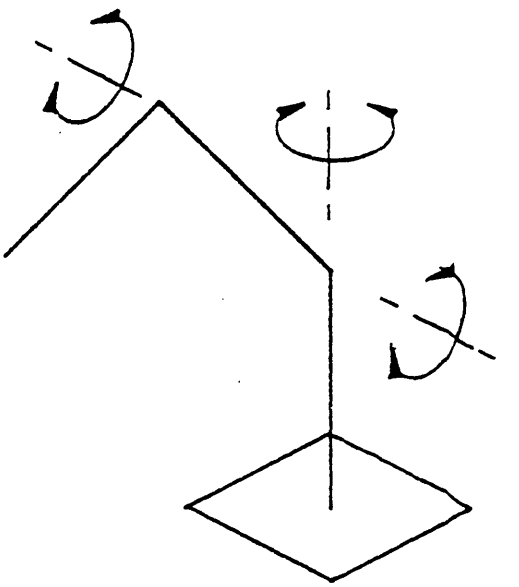


Figure 1.2 Standard Robot Primary Axes Configurations

2 THE DESIGNING OF A ROBOT MECHANICAL ARM

2.1 DESIGN STRATEGIES

The early general role of robot arms was reflected by their name of Universal Manipulators. In recent years however, robots, although still fairly flexible in their capabilities, are now being designed for one or two primary jobs. Machine loading, and materials handling robots are becoming more compact with fewer degrees of freedom, matched to specific machine tools, [CME 1980]. Process robots, (those that accomplish spraying, welding, shot peening, etc.), are now being tailored to be more easily taught, more durable in operation, and better at holding tolerances. Assembly dedicated robots, particularly those aimed at small part insertion, are light, fast, accurate robots, with an increasing level of intelligence using vision, tactile, and force sensing to accommodate changes in their environment.

Considering the mechanical aspects of the robot system three groups can be identified:

(i) Mechanical Arm: including the primary and secondary axes.

(ii) The gripper or end effector.

(iii) Ancillary equipment: including part presenters, conveyors, etc.

This section deals with the first of these items: the mechanical arm. In considering the arm, the design can be divided into two seperate aspects:

(i) The mechanical system.

(ii) Articulation requirements.

The first; the mechanical system, refers to the nature of the physical components that constitute the arm, and can be summarised as follows:

(i) Material characteristics of the robot "limbs" and joints.

(ii) Prime movers and drive systems.

(iii) Power transmission.

(iv) Feedback transducers.

The articulation of the robot represents such features as:

(i) Configuration of the primary axes.

(ii) Configuration of the secondary axes.

(iii) Additional degrees of freedom to increase the articulation of the robot.

The articulation of the robot has two underlying design criteria:

(i) Optimisation of the mechanical system.

(ii) Manoeuvrability of the robot in terms of manipulating the end effector within certain constraints, both in terms of functioning in confined volumes and avoiding articulation limitations such as singularities. (Dealt with in more detail in Section 4.3).

Considering both the design of the mechanical system and articulation requirements, a number of techniques and criteria have been proposed. These can be divided into the two groups as follows:

(i) Mechanical System. Techniques for designing the mechanical structure are predominantly involved with the dynamic response of the system. Some expand the kinematic specifications to develop dynamic variables to evaluate a design. Examples include: dimensionless constants, [Demaurex and Gerelle 1979], the mapping of inertia ellipsoids, (a geometric interpretation of the inertia tensors), [Asada 1982], and speed criteria and energy consumption, relating to the drives of the robot, [Vukobratovic et al 1978], [Vukobratovic et al 1980].

(ii) Robot Articulation. Research into robot articulation can be divided into two sections; structural synthesis and maneouvrability. Structural synthesis involves the mapping and structure of the workspace of the robot, [Lin 1982], and the analysis of special configurations. These techniques deal primarily with uni-joint configurations where a drive moves the robot without the aid of mechanisms. A more detailed analysis of kinematic chains and mechanisms in general, [Singimoto 1979], allows for more complex interaction of the robot joints and links.

The area of maneouvrability includes items such as the analysis of specific configurations of robots to indicate true maneouvrability of the end effector of the robot, [Kinoshita 1981]. Even though a robot

may have six dof, full three dimensional mobility is not necessarily achievable. Physical limitations and singularity impose restrictions on the mobility of the robot. The concept of redundancy, (whereby there are more degrees of freedom than boundary conditions), is also a feature of maneouverability. Research into the control of redundant manipulators [Brooks 1982], [Aspragathos 1983], enables redundant manipulators to be considered for tasks which would be impossible for six dof manipulators.

However the "art" of designing a robot arm is still an iterative technique embodying the classical principles of machinery design. It is also true to say that robot arm design is still, to some extent, an empirical science.

2.2 DESIGN SPECIFICATION AND CRITERIA FOR CHOICE

2.2.1 Design Specification

A design specification is drawn up with reference to certain design parameters which influence the role suitability, and performance of the robot. The most important parameters are as follows, [Warnecke and Schraft 1979]:

(i) Load lifting capacity.

(ii) Working envelope.

(iii) Positioning accuracy.

(iv) Positioning repeatability.

(v) Velocities and accelerations.

(vi) Static and dynamic stiffness.

(vii) Ease of maintenance.

(viii) Operational life span.

(ix) Safety aspects.

(x) Resistance to the work environment.

(xi) Initial and running costs.

These are inherent aspects of the mechanical system. Additional parameters relating to the articulation requirements are summarised as follows:

(i) Number and configuration of primary axes.

(ii) Number and configuration of secondary axes.

Careful consideration of the proposed job functions required of the robot enables the design specification to be drawn up based on the parameters summarised above. The importance of, and degree to which specific design criteria are expanded, is dependent upon the task required of the robot.

2.2.2 Criteria Effecting the Design of the Robot Limbs and Joints

Considering the robot limbs, two design aspects are relevant, material and spatial. Material requirements include:

(i) High strength to weight ratios.

(ii) High stiffness to weight ratios.

(iii) Good fatigue life.

(iv) High inherent damping.

(v) Resistance to the industrial environment and high structural resonant frequency.

(vi) Cost of material.

(vii) Cost, and ease, of manufacturing the robot components.

The choices available for the material and methods of manufacture are well defined. Experience obtained in the machine tool and areospace industries is particularly relevant. Table 2.1 compares the characteristics of three suitable materials: steel, aluminium alloy, and composite material. Research to evaluate the dynamic aspects of the limbs is well suited to the technique of finite element analysis, [Sung 1982], whilst less nummerically intensive techniques are sufficient for static stress and strain analysis, [Belolikov 1981].

Spatial design aspects relate to the actual shapes and dimensions of the limbs. Factors which may be applicable include:

(i) Housing of prime movers, transmission and sensory equipment.

(ii) Accessibility for maintenance.

(iii) Suitable cross sections to improve strength and stiffness.

(iv) Small cross section to allow maneouvrability in confined spaces.

Refering to the actual joints of the robot, the two choices of joint for a single degree of freedom are prismatic and rotary. (A robot joint can however include multiple degrees of freedom in a single unit. In these cases two or more input motions to the unit interact to give combined output motions). The major parameters relevant to joint designs include:

(i) Stiffness.

(ii) Weight, particularly if the prime mover is on the robot joint.

(iii) Bearing support.

(iv) Incorporation of power transmission or prime mover units.

(v) Cost.

Rotary joints prove advantageous for most of the above criteria, especially stiffness, bearing support, and incorporating power transmission. The advantages of prismatic joints relate to the accessibility of the robot to confined spaces, and a reduction in the computation required to achieve the inverse kinematic transformations and follow defined trajectories.

The stiffness of the joint is perhaps the most important single item, especially for high speed motion and positioning. Some of the limitations of the joint characterisitcs can be accommodated within the axis servo control unit. Alternatively mechanical methods of increasing the damping can be incorporated into the system. These include pneumatic dampers, to help compensate for the gravitational effects, and elastic supports at the joint, [Kamiya et al 1980]. The latter is particularly useful for the reduction of residual vibration at the joint.

2.2.3 Criteria Affecting the Selection of Prime Movers and Drive Systems

Although criteria for the choice of prime movers and drive systems are presented independently, a comparitive analysis is based on a matched prime mover/drive system combination. This provides a more relevant basis for the selection of the robot drive system.

The parameters most important when considering the choice of stationary positioning drives, (drives not carried on the moving limbs of the robot arm), are as follows, [Drexel et al 1980]:

(i) Nominal power and torque.

(ii) Starting and braking torque.

(iii) Rotational inertia (mechanical time constant).

(iv) Controllability/linearity.

(v) Positioning accuracy.

(vi) Noise and vibration.

(vii) Sensitivity to industrial environment.

(viii) Production, energy, and maintenance costs.

(ix) Behaviour in shutdown, power failure and operating safety.

(x) Cost of drive.

Additional parameters are important if the drives are actually on the moving limbs of the robot:

(i) Basic weight.

(ii) Power/weight ratio.

The main choices available to the robot design engineer are as follows:

(i)	Hydraulic:	cylinders limited rotation vane drives motors
(ii)	Electric:	DC torque motor (rare earth) DC servo motor AC servo motor (brushless, permanent magnet synchronous) stepper motor
(iii)	Pneumatic:	cylinder motor

Considering the system drive parameters, a number of aspects overlap the criteria relevant to the prime movers. These reflect some of the most important items when the prime mover/drive system combination is evaluated as a single unit. The parameters relevant to the choice of drive systems are as follows:

(i) Bandwidth
(ii) Hysteresis
} Frequency response.

(iii) Stability to environment.

(iv) Output power (continuous and peak).

(v) Efficiency.

(vi) System protection and diagnostics.

(vii) Modularity and expandability.

The major options available for drive systems are as follows:

(i)	Hydraulic:	low performance servovalve high performance servovalve
(ii)	Electric:	line frequency switched thyristor controller (SRC) pulse width modulated controller (PWM) linear amplifiers
(iii)	Pneumatic	valves valve/mechanical combination

A comparitive evaluation of a number of drive/prime mover combinations is presented in Table 2.2.

2.2.4 Power Transmission and Velocity Reduction

A number of robot designs contain the prime movers at the robot base. These designs require the transmission of motion, through the robot limbs to the joints. Often the transmission includes some velocity reduction. Although the transmission and velocity reduction are considered independently, a robot design may incorporate more than one option to achieve the desired response. The major requirements of the transmission/velocity reduction units are as follows:

(i) Low backlash.

(ii) High reduction (typically 1:50 - 1:500).

(iii) High efficiency.

(iv) Low added mechanical inertia.

(v) Comparatively high torque transmission (dependent on velocity reduction in system).

(vi) Low vibration and noise.

(vii) Low cost.

The major options available to the design engineer are listed below:

(i) Ball screws.

(ii) Spur gears.

(iii) Toothed rubber belts, (often with steel wire support).

(iv) Chain drives.

(v) Harmonic gears.

(vi) Mechanical linkages.

A comparison of these transmission options is given in Table 2.3.

2.2.5 Feedback Transducers

The area of feedback transducers is an important feature of robotics. This section deals primarily with positional feedback, although velocity and even acceleration feedback can be incorporated into the closed loop control system. There is also the field of enviromental sensing, an increasingly important area of robotics. These include; vision, tactile sensing and force sensing. Returning to positional feedback, the important criteria for choice are as follows:

(i) Minimal backlash between drive and transducer.

(ii) High resolution (if transducer incorporated at joint, resolution could be > 20 bit).

(iii) High bandwidth (primarily if transducer is on primary drive).

(iv) Mechanical/electrical reliability.

(v) Sensitivity to the environment.

(vi) Additional electronic requirements.

(vii) Cost.

The main options for positional measurement are as follows:

(i) Incremental optical encoders.

(ii) Absolute optical encoders.

(iii) Resolvers.

(iv) Inductosyns.

(v) Potentiometers.

A comparitive analysis of the above options is presented in Table 2.4.

2.3 DESIGN OF THE IMPERIAL COLLEGE ROBOT

The procedure adopted for designing the Imperial College robot followed a "top down" approach as follows:

(i) General specifications.

(ii) Mechanical system characteristics.

(iii) Mechanical design.

(iv) Motor and drive choice.

(v) Transmission medium.

(vi) Feedback transducers.

The two general specifications for the robot were:

(i) The robot was to be used for machine tool loading and unloading, and similar handling roles. There were however to be facilities for upgrading, to accommodate tasks such as welding.

(ii) The mechanical design was to be simple and robust. Any inherent performance limitations to be accommodated within the robot control system.

From these general specifications the following mechanical characteristics were determined:

(i) Lifting capacity of 25Kg including gripper.

(ii) A five dof anthropomorphic configuration, (see Figure 2.1).

(iii)

Base motion	$\pm 160^{o}$, speed 60^{o}/sec
Shoulder motion	$+240^{o}$, -60, speed 60^{o}/sec
Elbow motion	$\pm 120^{o}$, speed 60^{o}/sec
Wrist pitch	$\pm 120^{o}$, speed 120^{o}/sec
Wrist roll	$\pm 250^{o}$, speed 120^{o}/sec

(iv) Arm to reach two metres and to include the facility to "flip over" to reach objects behind robot.

(v) Positional accuracy and repeatability of ± 0.5mm.

The design procedure was based on standard analysis of stress and strain for the mechanical system, and torque requirements for robot joints. Full details of the design procedure are not presented. However the general description of the robot designed and constructed is included.

Based on the above specifications a prototype robot was designed and constructed to incorporate the following:

The mechanical construction used a combination of aluminium alloy and steel, (see Figure 2.2). The faster moving top limbs were fabricated from aluminium alloy. This was primarily to reduce the inertial loads near the end effector. The base, whose inertial effect is considerably less than the upper limbs, was fabricated using steel. This produced a stronger, more robust, base suited to the environment of the factory floor.

All transmission and gear reduction was achieved by the use of chains, (see Figure 2.3). A high load factor was used to minimise the effect of wear, and tensioning of the chains can be accomplished in situ.

The prime movers used for the Imperial College robot were d.c. servo motors. These are of compact "flat style" construction allowing units to be assembled incorporating the tachogenerator, brake and encoder. Fail safe brakes are necessary to hold the loads during any emergency. The tachogenerator provides velocity feedback to the servo amplifiers. The optical encoders fitted provide 512 square wave pulses per revolution, together with a once per revolution zero marker pulse. The encoders were fitted to the motors themselves, rather than the actual joints, to increase the measurable accuracy. (Position measurement at the joints can require up to 20 bit resolution with considerable increase in cost). For the prototype robot all motors were of the same rating.

The amplifiers used were transitor switched, pulse width modulated, servo amplifiers. These were matched to the motors used. The power element of the amplifier is a chopper consisting of four transistor switches. Adjustment is provided on each amplifier to modify the pre-amp or rate loop again. the tachogenerator scale factor and zero offset. For the d.c. servomotors used it was necessary to place an inductance in series with the motor armature. This ensures a low form factor to maximise efficiency and minimise motor heating.

Preliminary investigations, using the prototype robot, suggest that a better response of the mechanical system would be achieved by the use of harmonic drive gearboxes and toothed rubber belts. A modification of the motor ratings would also be necessary.

The robot, designed by the author, and manufactured with the help of Hazmac Handling Ltd., is shown in Figures 2.4 and 2.5.

2.4 PHYSICAL MODULARITY

Robots benefit from a flexibility and adaptability which sets them aside from other automated machines. However any over flexibility is a source of higher capital and operational costs and may lower reliability. In fact many robot designs are still not optimised for one specific task, but tend to be a result of a comprimise of many, often conflicting, requirements.

Two shortcomings of present day industrial robots have been identified as follows, [Surnin 1878]:

(i) A high degree of redundancy in robot capabilities compared to the requirements of many industrial applications.

(ii) A lack of kinematic versatility

The second point refers to the fact that most universal robots have a rigid kinematic structure which does not always best satisfy a specific application.

Statistical surveys, [Drimer 1980], have analysed information relating the kinematic requirements of a robot to a number of industrial applications, (including; machine tool loading, machines for working plastics, foundries, fabrication shops). The results of the research indicate that for at least 80% of robot applications, the requirement is for robots of four dof or less. Since most industrial robots exhibit five, or six dof, a large number of basic industrial tasks constitute an under-utilisation of robot capabilities.

An elegant solution to the problem of suitably matching the mechanics of a robot to a specific industrial task, is the concept of physical modularity. The task of fully investigating the design and manufacture is beyond the scope of this thesis. However a discussion of modular robots is presented below. This is incorporated due to the belief that industry will see an increasing role for modular robots, and that this concept is well reflected in the robot controller presented in this thesis.

An industrial robot of modular design is one in which the number of degrees of motion, (or freedom), hence the kinematic structure, are chosen to suit a specific application. It is constructed by combining appropriate design modules. These modules are functionally and structurally independent units. They can be employed individually, and in various combinations with other modules.

By employing a "plug together" approach, a modular robot can be easily adapted to suit a subsequent task requirement, if the original requirement becomes obsolete. This is achieved by reconfiguring or replacing the design modules of the robot.

Each module represents an autonomous mechanical unit. This simplifies the assembly of the final configuration, and enables easy adjustment during the operation of the robot. In a similar manner to the control modularity discussed in Chapter 7, the concept of physical modularity makes it possible to improve the reliability of individual modules, hence the robots configured from them.

There are, however, disadvantages associated with robots of modular construction. The increased number of interfacing surfaces, at the links of the kinematic chain inherent with the concept of modularity, will cause a reduction in stiffness. To correct for this will result in an increase in mass and dimensions of the design. In addition, the assembly of modules creates dificulties in developing standard interfacing surfaces, and the means to centre and fasten various module combinations. Power and communication connections must allow any combination of modules. This results in a significant redundancy of connections for each module. Finally, even though individual modules may be more reliable, the large number of "quick break" connections in links between modules, has a detrimental effect on design reliability.

It is the desire to eliminate the shortcomings, whilst maintaining the advantages of modular construction that reflects the work carried out in this field of mechanical design, [Velikovich 1978].

The simplest form of physical modularity, is the development of modules with a single degree of freedom. This is an approach adopted by the recently introduce "Robo-Arm" manufactured by Robotic Systems Ltd. Each module of the Robo-Arm features a printed motor, harmonic gearbox,

and resolver. The modules can be combined to give a robot with up to five dof.

The approach of single dof modules does however exhibit disadvantages. Heavy and bulky electric motors and gear units are mounted near the end effector, decreasing the payload and accessability of the robot. One solution, adopted by designers, is to construct the primary axis configuration from modules, and incorporate a wrist unit that is a complete independent module, [Surnin 1978]. The essence of modularity is still retained, in that the degrees of freedom of the wrist are variable. In addition the number of permutations of robot configuration is still high.

There are a number of advantages that can be realised by the use of a robot with modular design. Modularity allows the mechanical articulation to be configured, and reconfigured, to best suit the task required. This means that those variables which constitute a source of high capital costs (motors, drives, gearboxes etc) need satisfy only the articulation requirements of the specified task, reducing unecessary dof. There is also a potential for reliability which will decrease running costs of the system. These advantages, if realised, when matched to a control system with similar philosophy, could contribute to a complete robot system of cheaper capital and running costs.

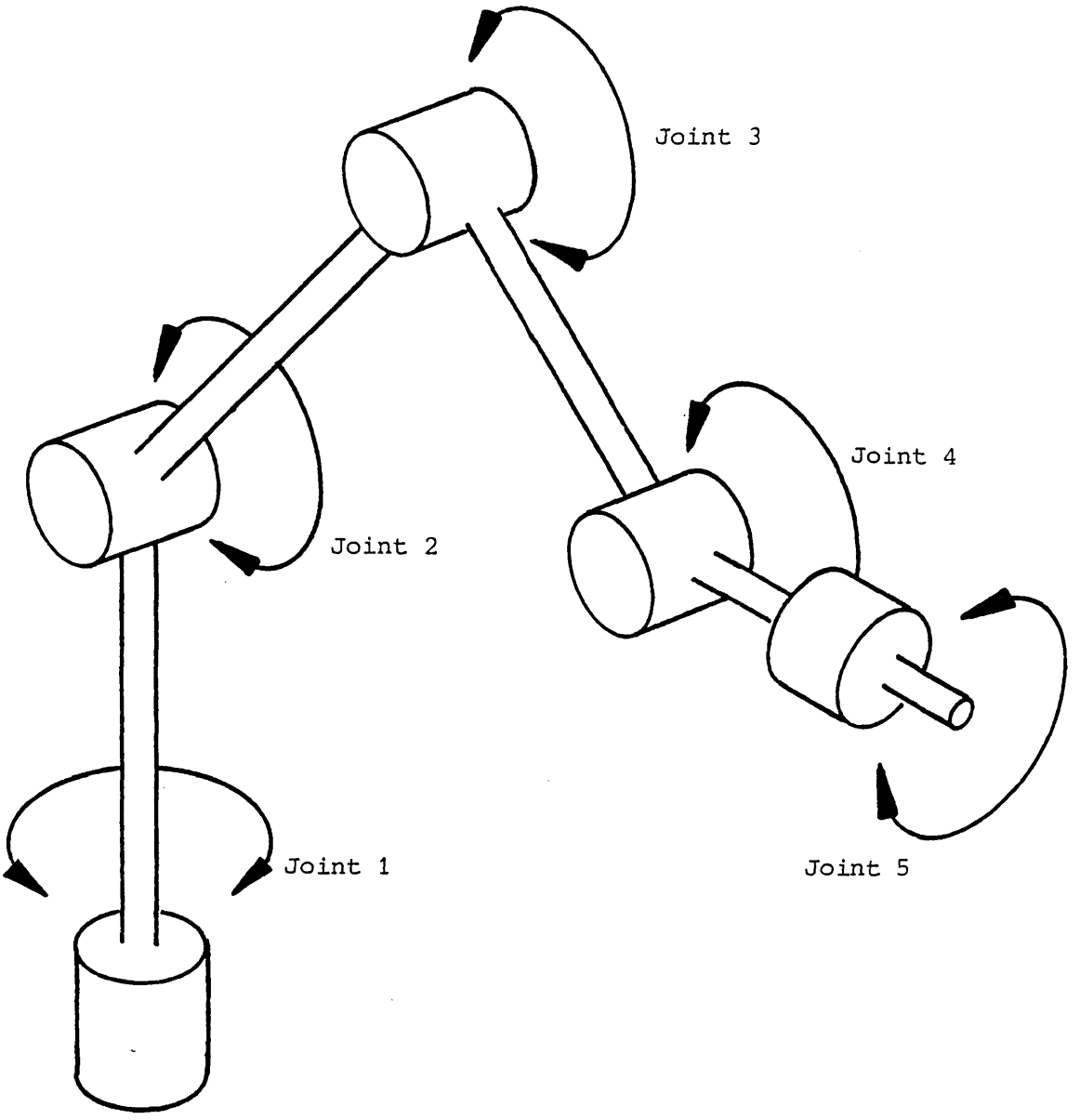


Figure 2.1 Five Degree of Freedom Anthropomorphic Configuration

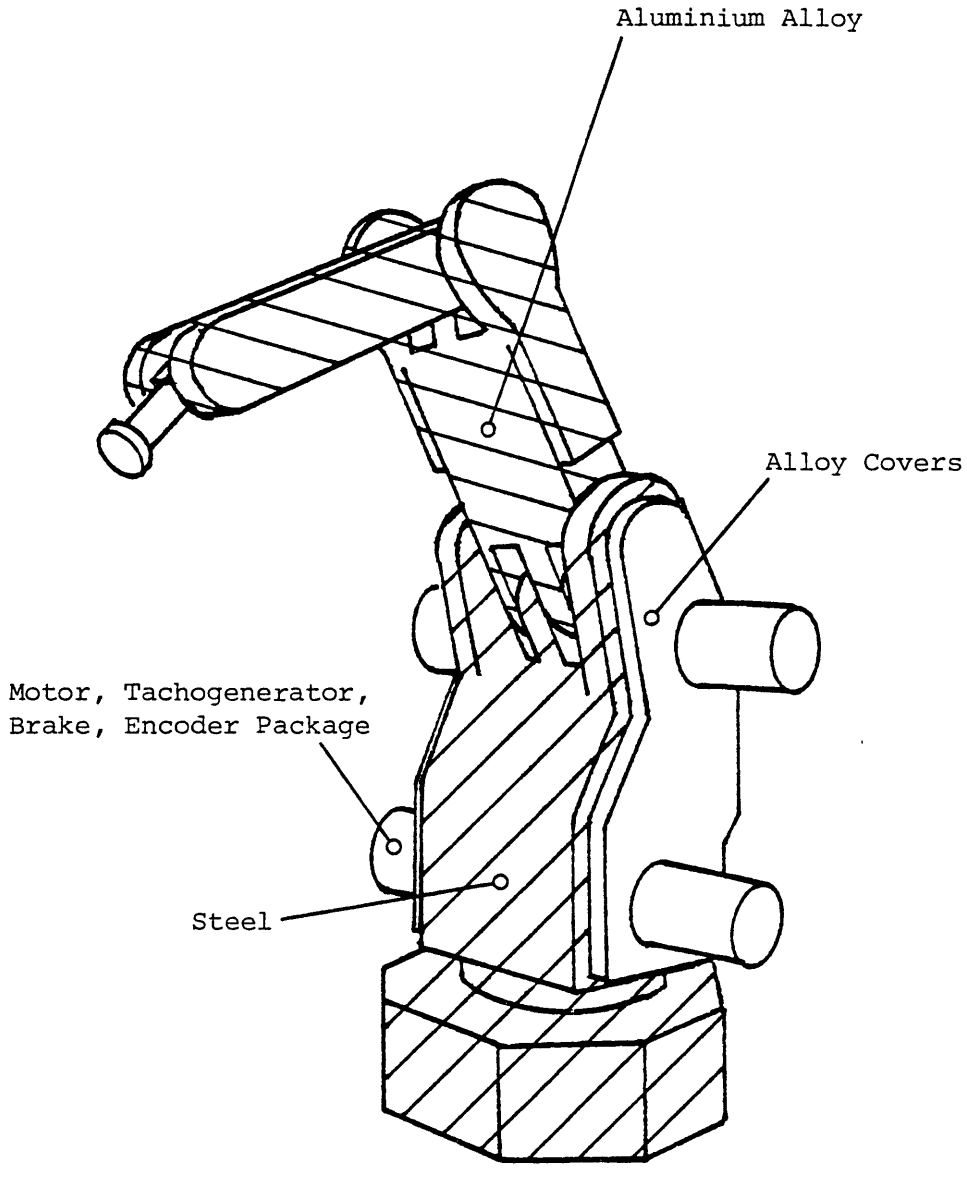


Figure 2.2 Mechanical Construction of the Imperial College Robot

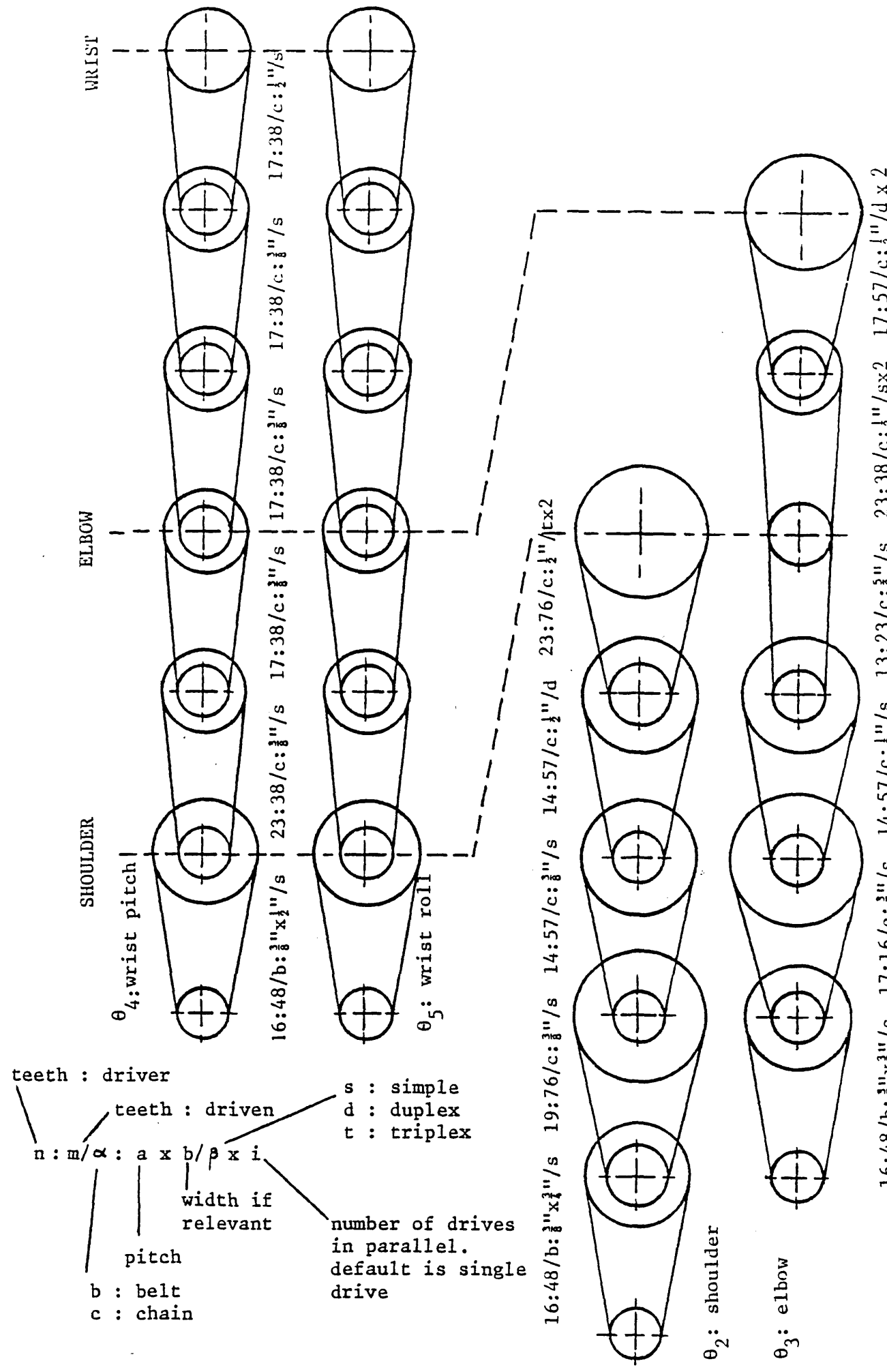


Figure 2.3 Chain Drive Schematic for the Imperial College Robot

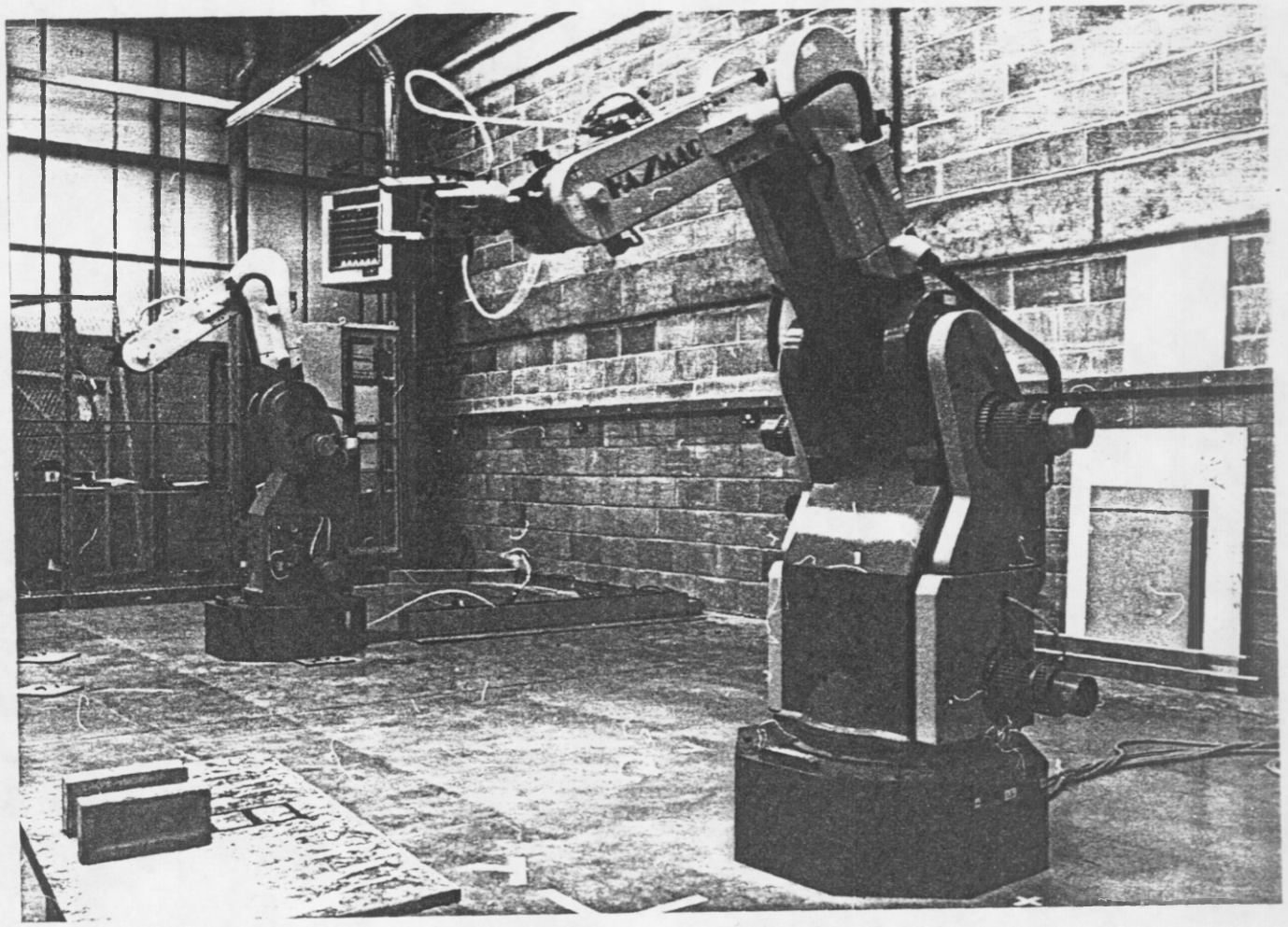


Figure 2.4 Imperial College Robot

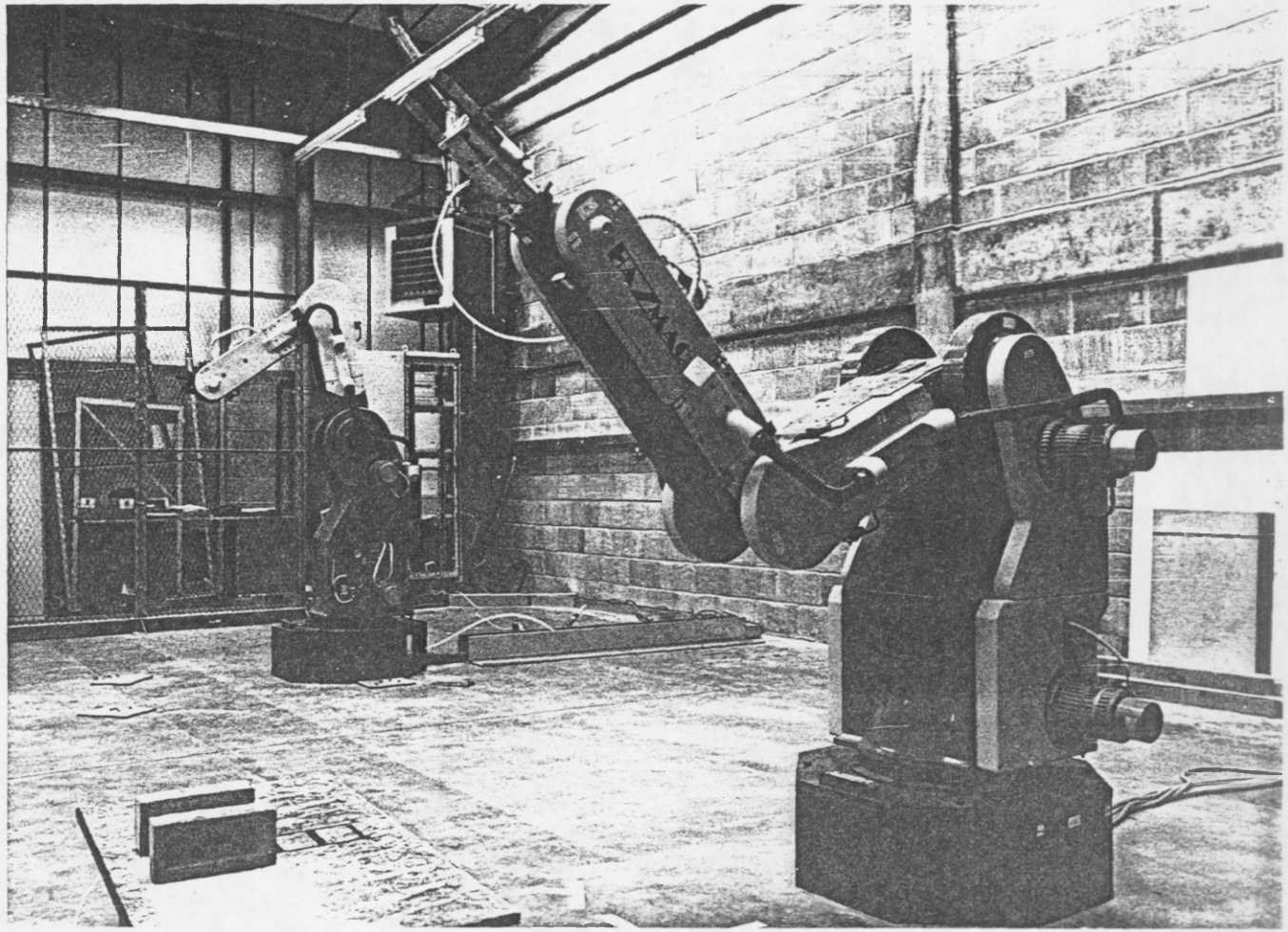


Figure 2.5 Imperial College Robot

	Strength to Weight	Stiffness to Weight	Inherent Damping	Resistance to Environment	Cost of Material	Cost & Ease of Maintenance
Steel	Low	Low/Medium	Medium	Medium	Low	Low
Aluminium Alloy	Medium	Medium	Medium/ High	Low/Medium	Low	Low/Medium
Composite Material	High	High	Medium/ High	High	High	High

Table 2.1 Comparison of Material Characteristics

	Power & Torque	Power to Weight	Mech. Inertia	Positioning Accuracy	Noise & Vibration	Sensitivity to Environment	Frequency Response	Efficiency	Protection & Diagnostics	Cost
Hydraulic Vane Actuator/High Quality Servovalve	High	High	Medium/ High	Medium/ High	Medium	Low	High	High	Medium	Medium/ High
DC Servo Motor/ PWM Amplifier	Medium	Medium/ High	Low	High	Low	Medium/ High	High	High	High	Medium
AC Permanent Magnet/PWM Synchro Amplifier	Medium	Medium	Low	High	Low	Low/ Medium	Medium	High	High	Medium
DC Torque Motor/ PWM Amplifier	Medium/ High	Medium/ High	Low	High	Low	Medium	High	High	High	High
Stepper Motor/ Translator Drive	Low	Low/ Medium	Low/ Medium	Medium/ High	Medium/ High	Medium/ High	Medium	Medium/ Low	High/ Medium	Medium
Pneumatic Cylinder/ Mechanical Damping	Low/ Medium	Low/ Medium	Low	Low	Medium/ High	Medium	Low	Low	Low	Low

Table 2.2 Comparison of Prime Mover/Drive Combinations

	Backlash	Speed Reduction	Efficiency	Inertia	Torque Transmission	Vibration	Size	Cost
Ball Screw	Low	Medium/ High	High	Medium	High	Low	Low	Medium
Spur Gears	Medium	Low	Medium/ High	High	High	Medium	Medium/ High	Low
Toothed Rubber Belts	Low	Low	Medium	Medium	Low	Low	Low/ Medium	Low
Chain Drives	Medium	Low	Medium/ High	Medium	High	Medium	Medium	Low
Harmonic Gears	Low	High	High	Low	High	Low	Low	High
Mechanical Linkages	Low	Low	Medium	Medium	High	Low	Medium/ High	Low

Table 2.3 Comparison of Transmission Options

	Resolution	Bandwidth	Mech/Elect Reliability	Sensitivity to Environment	Additional Electronic Requirements	Cost
Incremental Optical Encoder	High	High	Medium	Medium	Low	Medium
Absolute Optical Encoder	Medium	High	Medium	Medium	Low/ Medium	Medium/ High
Resolver	Medium/ High	Medium/ High	High	Low/ Medium	Medium/ High	Medium
Inductosyn	Medium	Medium/ High	High	Low	Medium	Low/ Medium
Potentiometer	Low	Low/ Medium	Medium	Low/ Medium	Low	Low

Table 2.4 Comparison of Positional Feedback Transducers

3 FRAME TRANSFORMATION

The two fundamental space systems relevant to a robotic arm are the base co-ordinate system, (normally Cartesian), and the robot joint system. This chapter is concerned with the first of these systems. The definition of the end effector in base Caresian co-ordinates, is given by the vector ${}^{b}\underline{X}$. The vector ${}^{b}\underline{X}$ comprises a position vector ${}^{b}\underline{x}$, and an orientation vector ${}^{b}\underline{\omega}$:

$$ {}^{b}\underline{X} = {}^{b}[\underline{x}, \underline{\omega}]^{T} \tag{3.1} $$

The superscript b indicates the reference co-ordinate frame, and T indicates the transverse. For full three dimensional space definition, the vectors ${}^{b}\underline{x}$ and ${}^{b}\underline{\omega}$, are of the form:

$$ {}^{b}\underline{x} = {}^{b}[x, y, z]^{T} \tag{3.2} $$

$$ {}^{b}\underline{\omega} = {}^{b}[\alpha, \beta, \gamma]^{T} \tag{3.3} $$

where x, y, z are the co-ordinates of the end point, and α, β, γ are orientation parameters.

The situation often arises when the position, or orientation vectors, (or both), are defined with respect to some other co-ordinate frame. The term "frame transformation" is given to the conversion of position or orientation, referenced to some secondary frame, to the equivalent parameters referenced to the robot base frame.

3.1 TRANSLATION AND ROTATION TRANSFORMATIONS

3.1.1 The Homogenous Transformation H

The two transformations of space relevant to robotics are linear translation, and rotation. Both of these transformations may be combined in a single 4x4 matrix; H, known as a homogeneous transformation matrix, [Roberts 1965]. Given a vector $\underline{u}$, its transformation to $\underline{v}$ is represented by:

$$\underline{v} = H\,\underline{u} \tag{3.4}$$

The matrix H comprises two distinct parts: a 3x3 rotation matrix R, and a 3x1 linear translation vector $\underline{L}$, where:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.5}$$

$$\underline{L} = [L_x, L_y, L_z]^T \tag{3.6}$$

The matrix H is then of the form:

$$H = \left[\begin{array}{ccc:c} & R & & L \\ \hdashline 0 & 0 & 0 & 1 \end{array}\right] \tag{3.7}$$

If the rotation matrix R is unitary, signifying no rotation, then multiplication by the matrix H is equivalent to vector addition.

3.1.2 The Rotation Matrix R

The frame rotation matrix R represents a rotation θ about a vector $\underline{k}$ and is signified by $R(\underline{k},\theta)$. The general form of $R(\underline{k},\theta)$ is as follows, [Hamilton 1969]:

$$\begin{bmatrix} k_x k_x \text{vers}\theta+\cos\theta & k_y k_x \text{vers}\theta-k_z\sin\theta & k_z k_x \text{vers}\theta+k_y\sin\theta \\ k_x k_y \text{vers}\theta+k_z\sin\theta & k_y k_y \text{vers}\theta+\cos\theta & k_z k_y \text{vers}\theta-k_x\sin\theta \\ k_x k_z \text{vers}\theta-k_y\sin\theta & k_y k_z \text{vers}\theta+k_x\sin\theta & k_z k_z \text{vers}\theta+\cos\theta \end{bmatrix} \qquad (3.8)$$

where:

$$\text{vers}\theta = (1-\cos\theta) \qquad (3.9)$$

$$k = k_x i + k_y j + k_z k \qquad (3.10)$$

It is, however, difficult to determine the vector $\underline{k}$ and rotation θ, for all but the elementary cases. This problem can be overcome by considering the equivalent of $R(\underline{k},\theta)$, easily obtained from a combination of rotations about the frame ordinates X, Y and Z. In addition, a rotation about a frame ordinate is defined by a much simpler form of the general rotation, Equation (3.8):

$$R(X,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \qquad (3.11a)$$

$$R(Y,\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \qquad (3.11b)$$

$$R(Z,\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (3.11c)$$

The equivalent rotation is found by multiplying the individual ordinate rotations.

3.2 POSITION AND ORIENTATION VECTORS

3.2.1 Vector Definition

To specify the positional state of a solid body in three dimensional space, it is necessary to specify six variables; three for position and three for orientation. The most familiar means of positional definition used in industry, is with reference to the cartesian co-ordinate system. This is therefore the co-ordinate frame used in most robot systems. The conventions adopted in this thesis has the origin at the robot base and ordinates as illustrated in Figure 3.1. Thus a point is defined by the vector ${}^{b}\underline{x}$, where:

$${}^{b}\underline{x} = {}^{b}[x, y, z]^{T} \qquad (3.12)$$

If, however some other co-ordinate system is required, then linear relationships exist which transform between co-ordinate systems. In the case of cylindrical polar co-ordinates:

$${}^{b}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {}^{b}\begin{bmatrix} r_c \cos\theta_c \\ r_c \sin\theta_c \\ z_c \end{bmatrix} \qquad (3.13)$$

where r_c, θ_c, and z_c refer to the co-ordinates of a cylindrical polar system. In the case of spherical polar co-ordinates:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}^b = \begin{bmatrix} r_s \cos\psi_s \cos\theta_s \\ r_s \cos\psi_s \sin\theta_s \\ r_s \sin\psi_s \end{bmatrix}^b \tag{3.14}$$

where r_s, ψ_s, and θ_s refer to co-ordinates of a spherical polar system.

Considering rotational definition, there are a number of methods available for defining the orientation of an object. The two most commonly used are pitch, yaw and roll (PYR), and Euler Angles (EUL). Each defines a series of rotations of a secondary co-ordinate frame, originally co-dimensioned with the base frame, about the base frame. Each individual rotation represents a rotation about an ordinate as described by Equations (3.11).

Referring to Figure 3.2, the orientation accomplished by the use of PYR is obtained by the sequence of rotations:

$$R(PYR) = R(Z,\alpha)\ R(Y,\beta)\ R(X,\gamma) \tag{3.15}$$

The orientation accomplished by means of Euler angles is defined by the series of rotations:

$$R(EUL) = R(Z,\phi)\ R(Y,\theta)\ R(Z,\psi) \tag{3.16}$$

The order in which the rotations are made, relate to the co-ordinate system at the time of rotation. If the sequence is taken from right to

left, then the rotations are taken with respect to the base frame ordinates. If the sequence is from left to right, then rotations are made with respect to the local frame ordinates. Considering R(PYR), the final orientation can be obtained by two series of rotations. R(PYR) can be defined by rotating α about Z, β about Y', and γ about X", (where the superscripts indicate the number of frame rotations that have occurred previously). Alternatively R(PYR) can be defined by rotating γ about X, β about Y, and α about Z. All the rotations in this case are about the base frame ordinates.

The method of specifying orientation chosen for work relating to this project is R(PYR). The major reason for this choice is the close relationship between R(PYR) and the conventions relevant to engineering drawings. This not only helps in the intuitive specification of orientation, but more important offers a comparable method of specification when the robot is programmed off-line and is integrated with an engineering design database. The PYR convention also aids the analysis of the inverse kinematics as described in Chapter 4. The rotation matrix R, specifying orientation is obtained by multiplying the ordinate rotation matrices, as specified in Equations (3.11). The sequence of multiplication is determined by Equation (3.15).

Thus a homogeneous matrix B, which describes the position and orientation with reference to the base axis system can be defined as:

$$B = {}^{b}\begin{bmatrix} C_\alpha C_\beta & C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma & C_\alpha S_\beta C_\gamma + S_\alpha S_\gamma & X \\ S_\alpha C_\beta & S_\alpha S_\beta S_\gamma + C_\alpha C_\gamma & S_\alpha S_\beta C_\gamma - C_\alpha S_\gamma & Y \\ -S_\beta & C_\beta S_\gamma & C_\beta C_\gamma & Z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.17)$$

where C_θ is $\cos(\theta)$ and S_θ is $\sin(\theta)$; ${}^bx$, ${}^by$, and ${}^bz$ refer to the co-ordinates of the end effector defined with respect to the base system.

3.2.2 Frame Transformation of Position Vectors

The two sets of parameters required to specify the state of a body in three dimensional space; position and orientation can be treated as two totally independent parameter groups. That is, the position may be defined with respect to a different reference frame than the orientation. A position vector referenced to a frame f; ${}^f\underline{x}$, is related to the equivalent base vector ${}^b\underline{x}$, by the relationship:

$$ {}^b\underline{x}_{,t} = {}^bH_{f,t} \; {}^f\underline{x}_{,t} \tag{3.18} $$

where

$$ {}^bH_{f,t} = \left[\begin{array}{c|c} {}^bR_{f,t} & {}^b\underline{O}_{f,t} \\ \hline 0 \;\; 0 \;\; 0 & 1 \end{array}\right] \tag{3.19} $$

The matrix ${}^bR_{f,t}$ is a 3x3 frame rotation matrix, and ${}^b\underline{O}_f$ is a 3x1 vector indicating the local frame origin. The subscripts; t reflect the fact that the values may be time dependent.

The frame rotation matrix ${}^bR_f$ will usually hold a constant value for a series of trajectories. If varied, ${}^bR_f$ will usually take a value that has been evaluated off-line. (Thus removing the need for on-line computation).

The frame origin, ${}^{b}\underline{O}_{f}$ will also hold a constant value for many applications required of the robot and, as for ${}^{b}R_{f}$, may be evaluated off-line. One example where the value of ${}^{b}\underline{O}_{f}$ is required to vary continuously, is when the robot is tracking a moving object. Typical examples of tracking arise when the robot integrates with a parts handling conveyor, or when interacting with items on an assembly line. These examples normally require a linear variation of ${}^{b}\underline{O}_{f}$. An example where ${}^{b}\underline{O}_{f}$ will vary circularly is when the robot interfaces with a carousselle or some similar circular conveyor system.

The computational task of the robot controller can be simplified by careful planning of the robot position and physical environment. When integrating a robot and continuously moving conveyor, if the conveyor motion corresponds to the Y axis of the robot, then the frame origin is defined by:

$$ {}^{b}\underline{O}_{ft} = {}^{b}[O_x, O_{y,t}, O_z]^{T}_{f} \qquad (3.20) $$

Thus ${}^{b}(O_{y,t})_{f}$ is the only time dependent variable that must be determined.

If the robot works with a surface that is not parallel to the XY base plane then careful positioning of the surface simplifies the form of ${}^{b}R_{f}$. By positioning the surface such that its orientation is specified by a single rotation about a base ordinate, then the form of ${}^{b}R_{f}$ has only four non-zero elements instead of the usual nine.

The two default values of ${}^{b}\underline{O}_{f}$ and ${}^{b}R_{f}$ occur when the secondary frames are either pure rotation or translation. If the base and frame origins are co-incident then ${}^{b}\underline{O}_{f}$ is equal to zero. When any secondary frame is a pure translation, R_F is then the unit matrix. Thus if the working frame of the robot is the base frame:

$$ {}^{b}Hf = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad (3.21) $$

3.2.3 Frame Transformation of Orientation Vectors

To evaluate the joint values of a robot, the orientation parameters of an object must be eventually defined with respect to the base frame, $({}^{b}\alpha,\ {}^{b}\beta,\ {}^{b}\gamma)$. Matrix methods for evaluating these base parameters involve the element comparison of the orientation definition matrix (Equation (3.13)), and the end effector definition matrix, defined with respect to the base frame, ${}^{b}T$, where:

$$ {}^{b}T = {}^{b}H_{f}{}^{f}T \qquad (3.22) $$

(The method of evaluating T is discussed in Section 4.3).

The elemental comparison of B and ${}^{b}T$ yield the following equations, [Paul 1981]:

$$ {}^{b}\alpha = \text{atan2}\ (n_y,\ n_x) \qquad (3.23) $$

$$ {}^{b}\beta = \text{atan2}\ (-n_z, Cb_{\alpha}n_x + Sb_{\alpha}n_y) \qquad (3.24) $$

$$b_\gamma = \text{atan2}\,(Sb_\alpha a_x - Cb_\alpha a_y, -Sb_\alpha o_x + Cb_\alpha o_y) \tag{3.25}$$

where, atan2 (a,b) is the four quadrant form of $\tan^{-1}$ (a/b), and:

$$n_x = r_{11}Cf_\alpha Cf_\beta + r_{12}Sf_\alpha Cf_\beta - r_{13}Sf_\beta \tag{3.26}$$

$$n_y = r_{21}Cf_\alpha Cf_\beta + r_{22}Sf_\alpha Cf_\beta - r_{23}Sf_\beta \tag{3.27}$$

$$n_z = r_{31}Cf_\alpha Cf_\beta + r_{32}Sf_\alpha Cf_\beta - r_{33}Sf_\beta \tag{3.28}$$

$$a_x = r_{11}(Cf_\alpha Sf_\beta Cf_\gamma - Sf_\alpha Sf_\gamma) + r_{12}(Sf_\alpha Sf_\beta Cf_\gamma - Cf_\alpha Sf_\gamma) + r_{13}Cf_\beta Cf_\gamma \tag{3.29}$$

$$a_y = r_{21}(Cf_\alpha Sf_\beta Cf_\gamma - Sf_\alpha Sf_\gamma) + r_{22}(Sf_\alpha Sf_\beta Cf_\gamma - Cf_\alpha Sf_\gamma) + r_{23}Cf_\beta Cf_\gamma \tag{3.30}$$

$$o_x = r_{11}(Cf_\alpha Sf_\beta Sf_\gamma - Sf_\alpha Cf_\gamma) + r_{12}(Sf_\alpha Sf_\beta Sf_\gamma + Cf_\alpha Sf_\gamma) + r_{13}Cf_\beta Sf_\gamma \tag{3.31}$$

$$o_y = r_{21}(Cf_\alpha Sf_\beta Sf_\gamma - Sf_\alpha Cf_\gamma) + r_{22}(Sf_\alpha Sf_\beta Sf_\gamma + Cf_\alpha Sf_\gamma) + r_{23}Cf_\beta Sf_\gamma \tag{3.32}$$

As before, C and S represent the cosine and sine of the angle subscripts.

A simpler form of the expressions for b_α, b_β and b_γ has been evaluated by Uchiyama, [Uchiyama 1979]. Again the technique uses element by element comparison to yield the following:

$$b_\alpha = \text{atan2}\,(n_y, n_x) \tag{3.33}$$

$$b_\beta = \text{atan2}\,(-n_z, (n_x^2 + n_y^2)^{1/2}) \tag{3.34}$$

$$^{b}\gamma = \text{atan2}\,(o_z, a_z) \tag{3.35}$$

where n_x, n_y, n_z are defined as in Equations (3.26), (3.27), and (3.28), and:

$$o_z = r_{31}(Cf_\alpha Sf_\beta Sf_\gamma - Sf_\alpha Cf_\gamma) + r_{32}(Sf_\alpha Sf_\beta Sf_\gamma + Cf_\alpha Sf_\gamma) + r_{33} Cf_\beta Sf_\gamma \tag{3.36}$$

$$a_z = r_{31}(Cf_\alpha Sf_\beta Cf_\gamma - Sf_\alpha Sf_\gamma) + r_{32}(Sf_\alpha Sf_\beta Cf_\gamma - Cf_\alpha Sf_\gamma) + r_{33} Cf_\beta Cf_\gamma \tag{3.37}$$

The method of orientation transformation devised by the author, involves the use of an imaginary extension of the tool. This extension comprises of three links set at right angles and configured according to the secondary frame parameters. By using a geometric technique, the simplest form of the equations can be determined in addition to the results presented above. The mathematical derivation is presented in Appendix B, the results of which are as follows:

$$^{b}\alpha = \text{atan2}\,(\delta^{b}(p1_y), \delta^{b}(p1_x)) \tag{3.38}$$

$$^{b}\beta = \text{atan2}\,(\delta^{b}(p1_z), L) \tag{3.39}$$

where L can be defined by any of the four equalities:

$$\text{(i)} \qquad L = (\delta^{b}(p1_x)^2 + \delta^{b}(p1_y)^2)^{1/2} \tag{3.40}$$

$$\text{(ii)} \qquad L = \delta^{b}(p1_x) Cb_\alpha + \delta^{b}(p1_y) Sb_\alpha \tag{3.41}$$

$$\text{(iii)} \qquad L = \delta^{b}(p1_x) / Cb_\alpha \tag{3.42}$$

(iv) $L = \delta^b(p1_y)/Sb_\alpha$ (3.43)

(Each of the four equalities varies in terms of mathematical operations and equality deterioration. The importance of these aspects are discussed fully in Chapter 7). Continuing:

$$ {}^b\gamma = \text{atan2}\,(\delta^b(p3_z),\delta^b(p2_z)) \tag{3.44} $$

The values of the vector components required to obtain ${}^b\alpha$, ${}^b\beta$, ${}^b\gamma$ are defined as follows:

$$ \delta^b(p1_x) = r_{11}Cf_\beta Cf_\alpha + r_{12}Cf_\beta Sf_\alpha - r_{13}Sf_\beta \tag{3.45} $$

$$ \delta^b(p1_y) = r_{21}Cf_\beta Cf_\alpha + r_{22}Cf_\beta Sf_\alpha - r_{23}Sf_\beta \tag{3.46} $$

$$ \delta^b(p1_z) = r_{31}Cf_\beta Cf_\alpha + r_{32}Cf_\beta Sf_\alpha - r_{33}Sf_\beta \tag{3.47} $$

$$ \delta^b(p2_z) = r_{31}(Cf_\gamma Sf_\beta Cf_\alpha + Sf_\gamma Sf_\alpha) + r_{32}(Cf_\gamma Sf_\beta Sf_\alpha - Sf_\gamma Cf_\alpha) + r_{33}Cf_\gamma Cf_\beta \tag{3.48} $$

$$ \delta^b(p3_z) = r_{31}(Sf_\gamma Sf_\beta Cf_\alpha - Cf_\gamma Sf_\alpha) + r_{32}(Sf_\gamma Sf_\beta Sf_\alpha + Cf_\gamma Cf_\alpha) + r_{33}\,Sf_\gamma Cf_\beta \tag{3.49} $$

3.3 METHODS OF DERIVING THE FRAME TRANSFORMATION MATRIX

As discussed earlier, the nine variables in the frame rotation matrix; ${}^bR_f$, will, for most applications remain constant for a large

number of tasks. That is they will not vary with respect to time, as in the case of the interpolation or orientation vectors. The derivation of ${}^{b}R_{f}$ requires the values of the matrix components where:

$$ {}^{b}R_{f} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = {}^{b}[\underline{r}_1, \underline{r}_2, \underline{r}_3]_f \qquad (3.50) $$

The four common methods for obtaining the frame rotation matrix and local frame origin are as follows:

(i) Alignment frame.

(ii) Tool frame.

(iii) Three point frame.

(iv) Off-line user defined frame.

Each of these is discussed in turn below.

3.3.1 Alignment Frame

This method of determining ${}^{b}R_{f}$ and ${}^{b}\underline{O}_{f}$ is used commerically by Cincinatti Milacron Inc., [Tarvin 1980]. The transformation parameters are obtained by the use of an alignment frame as illustrated in Figure 3.3. The corner of the frame, (denoted by PT:0) is placed at the required zero-reference point of the work environment, with the three legs of the frame placed along the X, Y, and Z axes of the required co-ordinate system. Special gauge marks are scribed on the frame at precise distances along the X, Y, and Z legs, the lengths being l_x, and l_y, and l_z respectively.

A pointer attached to the robot and effector is then directed to the four reference points on the frame, and the absolute co-ordinates recorded. The x, y, and z values of the markings; 0, X, Y, and Z are then used to determine ${}^{b}\underline{O}_f$ and ${}^{b}R_f$, as follows:

$$ {}^{b}\underline{O}_f = [O_x, O_y, O_z]^T \tag{3.51} $$

$$ {}^{b}\underline{r}_{1f} = [(X_x-O_x)/l_x, (Y_x-O_x)/l_y, (Z_x-O_x)l_z]^T \tag{3.52} $$

$$ {}^{b}\underline{r}_{2f} = [(X_y-O_y)/l_x, (Y_y-O_y)/l_y, (Z_y-O_y)l_z]^T \tag{3.53} $$

$$ {}^{b}\underline{r}_{3f} = [(X_z-O_z)/l_x, (Y_z-O_z)/l_y, (Z_z-O_z)l_z]^T \tag{3.54} $$

3.3.2 Tool Defined Frame

The values of parameters of ${}^{b}\underline{O}_f$ and ${}^{b}R_f$, for a tool defined frame, can be obtained directly from the robot input variables ${}^{b}x$, ${}^{b}y$, ${}^{b}z$, ${}^{b}\alpha$, ${}^{b}\beta$, ${}^{b}\gamma$:

$$ {}^{b}\underline{O}_f = [x, y, z]^T \tag{3.55} $$

$$ {}^{b}R_f = [R(Z, \alpha)\ R(Y, \beta)\ R(X, \gamma)]^{-1} \tag{3.56} $$

Because the matrix ${}^{b}R_f$ represents an orthogonal transformation, the inverse is equivalent to the transpose. Thus the required frame rotation matrix ${}^{b}R_f$ is given by:

$$ {}^{b}R_f = \begin{bmatrix} C_\alpha C_\beta & S_\alpha C_\beta & -S_\alpha \\ C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma & S_\alpha S_\beta S_\gamma + C_\gamma C_\alpha & C_\beta S_\gamma \\ C_\alpha S_\beta C_\gamma + S_\alpha S_\gamma & S_\alpha S_\beta C_\gamma - C_\alpha S_\gamma & C_\beta C_\gamma \end{bmatrix} \tag{3.57} $$

3.3.3 Three Point Frame Specification

This method of evaluating the local frame parameters, is similar in some respects to that of the alignment frame, in that a pointer attached to the robot end and effector is taken to points within the work environment. However no alignment frame is required, and only three points are necessary. These are at the local frame origin, a point along the required X axis, and a point along the required Y axis; denoted by O, X, and Y respectively. Three angles; α', β', and γ' are then obtained as follows:-

$$\alpha' = \text{atan2}\,(X_y - O_y,\ X_x - O_x) \tag{3.58}$$

$$\beta' = \text{atan2}\,(X_z - O_z,\ ([X_x - O_x]^2 + [X_y - O_y]^2)^{1/2}) \tag{3.59}$$

$$\gamma' = \text{atan2}\,(Y_z - O_z,\ ([Y_x - O_x]^2 + [Y_y - O_y]^2)^{1/2}) \tag{3.60}$$

The local frame origin ${}^{b}\underline{O}_f$ is defined by:

$${}^{b}\underline{O}_f = [O_x,\ O_y,\ O_z]^T \tag{3.61}$$

The three angles α', β' and γ' are equivalent to the tool orientation angles ${}^{b}\alpha$, ${}^{b}\beta$, ${}^{b}\gamma$. However ${}^{b}\alpha$, ${}^{b}\beta$, ${}^{b}\gamma$ represent the actual orientation parameters of the end effector, whereas α', β', and γ' represent the inferred orientation angles of the end effector if it were holding a reference frame. Thus the frame rotation matrix ${}^{b}R_f$ is of identical form as Equation (3.57) but with α', β' and γ' substituted for ${}^{b}\alpha$, ${}^{b}\beta$, and ${}^{b}\gamma$ respectively.

3.3.4 Off-Line User Defined Frame

There may be circumstances when a robot programmer will wish to define a frame. This may occur at the local level, i.e. when moving the robot by pendant, or at a higher level when interfacing with a high level language or a computer design/manufacture database. In these cases the programmer can utilise any of the methods discussed above except that the parameters for ${}^{b}H_{f}$ are input directly as numerical values instead of physically moving the robot to a specified position or configuration.

Frame rotation matrices are also required when using certain interpolated pathes (e.g. circles, ellipses). Their use in this context is discussed in Section 5.2.2.

3.4 CONCATENATION OF FRAME TRANSFORMATIONS

The mathematical form of the homogeneous frame transformation matrix; H, allows concatenation of spatially sequential frames to give a single transformation matrix. The general relationship between a position vector in frame j; ${}^{j}\underline{x}$, and a position vector in frame j+1; ${}^{j+1}\underline{x}$, is given by:

$$ {}^{j}\underline{x} = {}^{j}H_{j+1}\,{}^{j+1}\underline{x} \qquad (3.62) $$

where ${}^{j}H_{j+1}$ is the transformation matrix from frame j+1 to frame j.

Thus the relationship between a position vector in the base frame, (frame 0), and frame n is given by the equation:

$$ {}^{b}\underline{x} = {}^{b}H_1 \; {}^{1}H_2 \; {}^{2}H_3 \ldots {}^{n-1}H_n \; {}^{n}\underline{x} \tag{3.63} $$

An example that requires frame concatenation, occurs when two robots are working in a common frame, designated c. If the robots are required to work in some other frame; d, (a subset of frame c) then the base frame position vectors for each robot are given by the relationships:

For robot 1:
$$ {}^{b}\underline{x}_1 = {}^{b}H_{1c} \; {}^{c}H_d \tag{3.64} $$

For robot 2:
$$ {}^{b}\underline{x}_2 = {}^{b}H_{2c} \; {}^{c}H_d \tag{3.65} $$

where the numeric subscripts refer to the specific robot, ${}^{b}H_{1c}$ and ${}^{b}H_{2c}$ refer to the individual frame transformations from frame c to the base frame for robots 1 and 2 respectively. ${}^{c}H_d$ is the transformation from frame d to frame c.

Another example occurs when a dynamic frame is used. If two static frames are seperated by a time dependent or dyanamic frame, then the equivalent frame transformation matrix ${}^{b}H_{3,t}$ is given by:

$$ {}^{b}H_{3,t} = {}^{b}H_1 \; {}^{1}H_{2,t} \; {}^{2}H_3 \tag{3.66} $$

where the subscript shows the time dependency of both the dynamic and equivalent frames.

In a similar manner to the evaluation of the base position vectors, the base orientation vectors are obtained from the equivalent frame transformation matrix. In the case of base orientation vectors, the constituent orientation angles are obtained from the rotational sub-matrix; R, of the total transformation H. Again the position and orientation need not be specified with respect to the same frames.

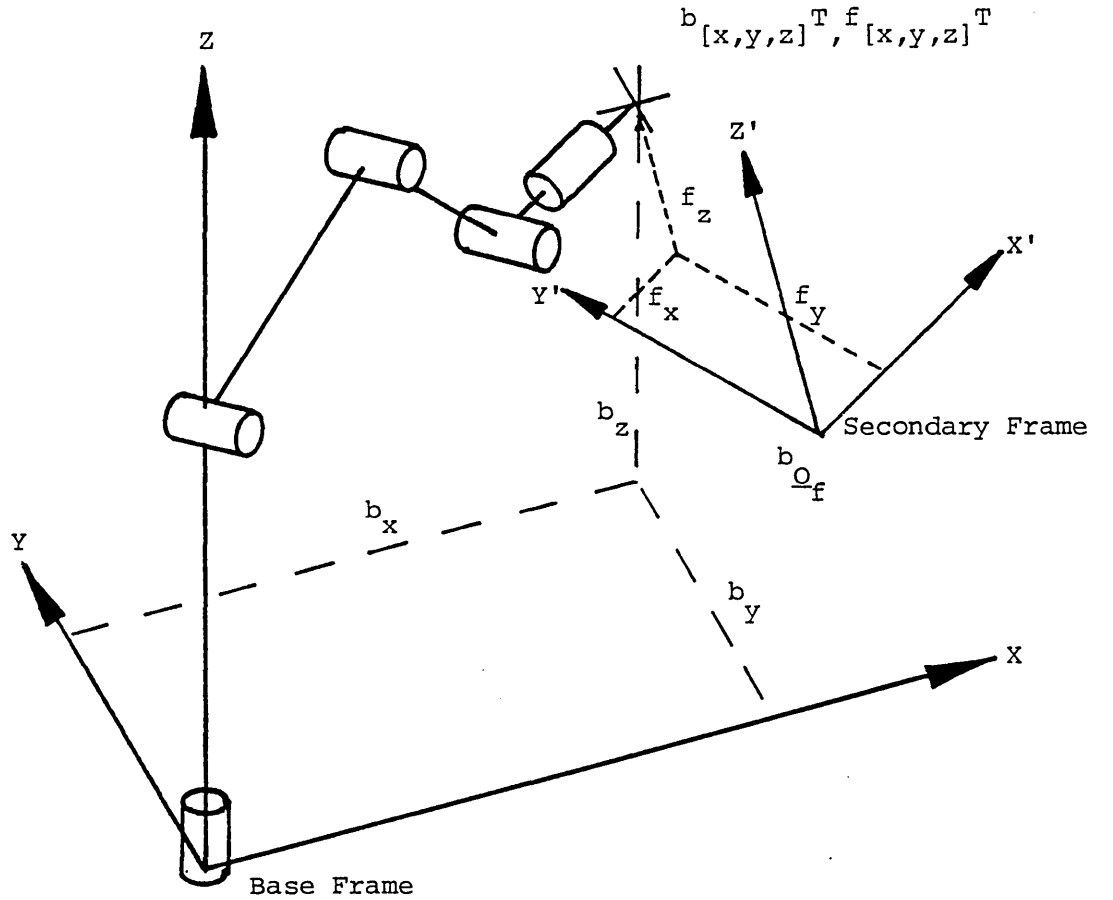


Figure 3.1 Conventions for Robot Co-ordinate Frames

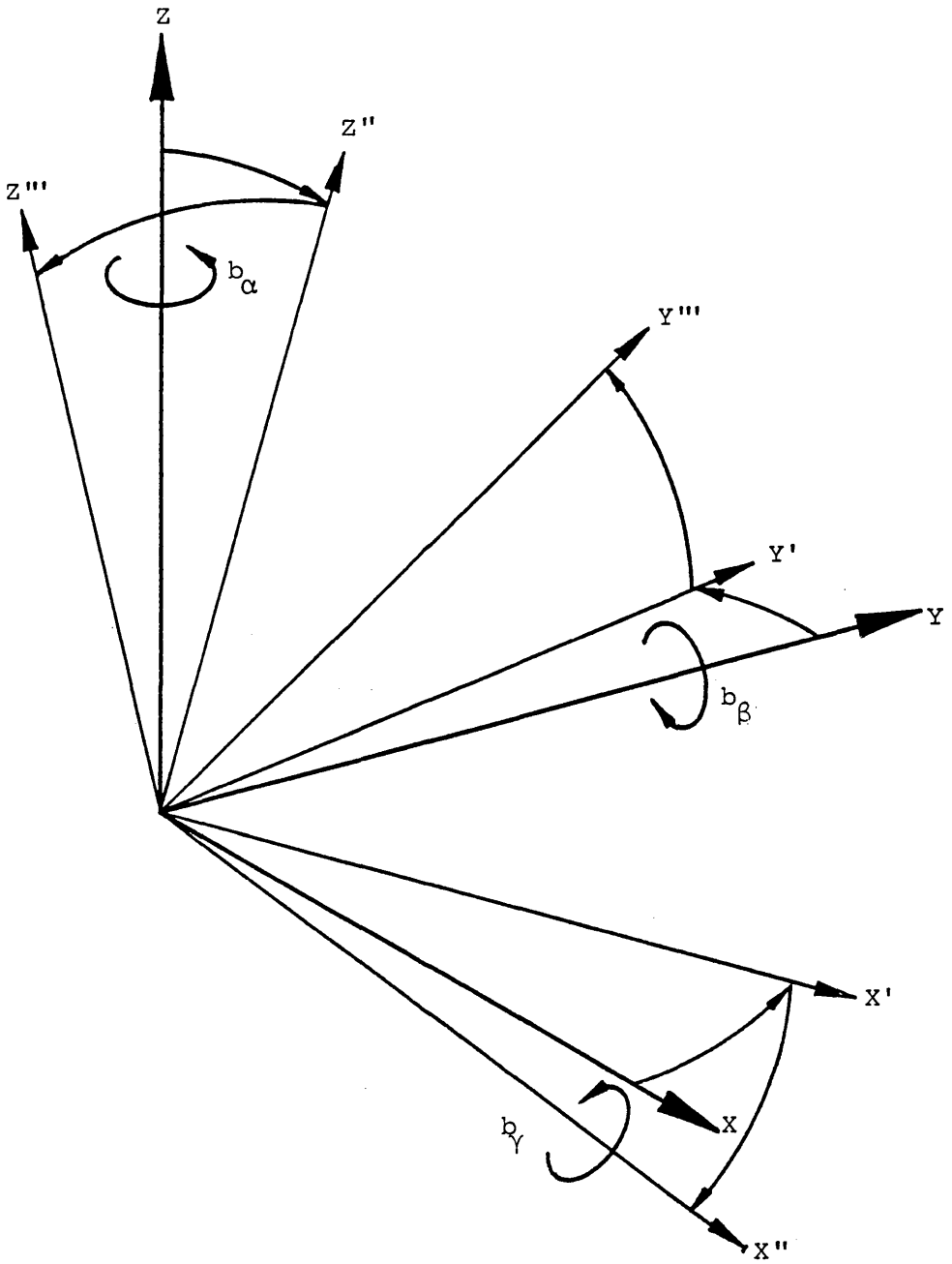


Figure 3.2 Convention for Pitch Yaw and Roll Orientation

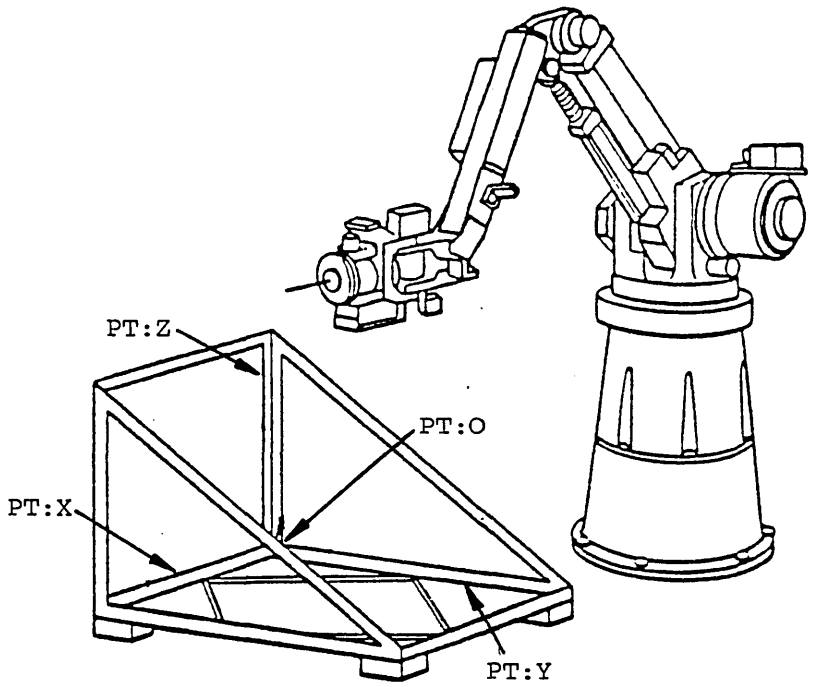


Figure 3.3 Secondary Frame Definition Using an Alignment Frame

4 CO-ORDINATE TRANSFORMATION

4.1 CONTINUOUS PATH MOTION

The first industrial robots used a point to point (PTP) mode of operation. Using this mode, the robot is taught points in the base co-ordinate space by means of a teach pendant. Each robot joint parameter (joint angle or linear position) is then recorded at the specified position. In achieving a sequence of taught points, the robot controller merely interpolates for, and servoes, each joint to provide synchronised motion. The subsequent path is co-ordinated in the sense that the robot passes through the sequence of taught points. There are however a number of disadvantages associated with this form of robot control:

(i) The path between taught points is non-definable and consists of a combination of joint motions.

(ii) The generation of closely defined paths requires the teaching of numerous points along the required path. This is cumbersome when teaching and requires the recording and storage of an excessively large number of points (typically 15000 for a five minute program).

(iii) Adaptive control is virtually impossible, in that if a non pre-programmed deviation to the path is required, it cannot be easily accommodated into the program.

One commonly adopted solution to these problems involves the dynamic generation of joint parameters from specified positions and orientations. These being defined with respect to the robot base co-ordinate system. The robot controller generates a series of interpolation points, in base space, between two specified points. These points are transformed into joint space to provide the sequence of joint parameters necessary to accomplish the required path. This task of parameter mapping, or co-ordinate transformation, involves two distinct operations.

(i) Evaluation of the necessary physical robot joint parameters to achieve the position and orientation required.

(ii) The transformation from the required physical joint parameters to the actual robot drive parameters.

The mapping of base co-ordinate space parameters to the robot joint space parameters, is one of the major tasks required of the robot controller and is defined as:

$$\underline{X}(t) \rightarrow \underline{\theta}(t) \tag{4.1}$$

where:

$$\underline{X} = [x, y, z, \alpha, \beta, \gamma]^T \tag{4.2}$$

$$\underline{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T \tag{4.3}$$

θ_i is the ith joint value of the robot.

The above relationship applies to a robot exhibiting six degrees of freedom. The relationship between $\underline{\theta}$ and $\underline{X}$ comprises a set of highly non-linear coupled equations. Methods for the calculation of $\underline{\theta}$ given $\underline{X}$ are discussed in this chapter.

4.1.1 Joint Frame Assignation

The method commonly adopted for assigning co-ordinate frames to each link of a manipulator, is based on the Denavit Hartenberg convention, [Denavit 1955]. This convention is embodied in the concept of A matrices. The A matrix relating to the co-ordinate frame of link n to link n-1 of a manipulator is defined as follows:

$$ {}^{n-1}A_n = \begin{bmatrix} \cos\theta & -\sin\theta\cos\alpha & \sin\theta\sin\alpha & a\cos\theta \\ \sin\theta & \cos\theta\cos\alpha & -\cos\theta\sin\alpha & a\sin\theta \\ 0 & \sin\alpha & \cos\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.4) $$

where, referring to Figure 4.1:

a : the distance between the z_{n-1} axis and the z_n axis;

α : the angle between the z_{n-1} axis and the z_n axis about the x_n axis;

d : the z_{n-1} distance between the x_n axis and the x_{n-1} axis;

θ : the angle between the x_n axis and the x_{n-1} axis about the z_{n-1} axis.

With most commercial manipulators the A matrices for the links simplify to the form:

$$ {}^{n-1}A_n = \left[\begin{array}{c|c} {}^{n-1}R_n & {}^{n-1}\underline{x}_n \\ \hline 0 & 1 \end{array}\right] \qquad (4.5) $$

where ${}^{n-1}R_n$ refers to a rotation matrix of the form of Equations (3.11) and describes a rotation about one of the co-ordinate frame axes. ${}^{n-1}\underline{x}_n$ is the vector describing the origin of frame n with respect to the origin of frame n-1.

4.2 EVALUATION OF THE DIRECT KINEMATIC EQUATIONS

Two sets of kinematic equations are relevant to robotics:

(i) Direct kinematics: which maps the robot joint space to the co-ordinate base space; $\underline{\theta}(t) \rightarrow \underline{X}(t)$.

(ii) Inverse kinematics: which maps the co-ordinate base space to the robot joint space; $\underline{X}(t) \rightarrow \underline{\theta}(t)$.

This section deals with the first of the two equation sets, involving the evaluation of the position and orientation vectors from the robot joint parameters. Of the two, the direct kinematics is far easier mathematically, yielding a unique solution describing the robot end effector.

A straightforward method of obtaining the direct kinematic equation involves the use of the A matrix for each link of the robot. The base description matrix ${}^{b}T$ is obtained as follows:

$$ {}^{b}T = {}^{0}A_1 \, {}^{1}A_2 \, {}^{2}A_3 \, {}^{3}A_4 \, {}^{4}A_5 \, {}^{5}A_6 \, E \qquad (4.6) $$

The matrix ${}^{b}T$ is equivalent to that of Equation (3.22) specifying the homogeneous form of the base space parameters: x, y, z, α, β, γ, and the matrix E describes the end effector geometry. The parameters of the six degree of freedom robot, illustrated in Figure 4.2, are shown in Table 4.1. ${}^{b}T$ can be written in the form, [Paul 1981]:

$$ {}^{b}T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4.7) $$

The components of the base definition vector $\underline{X}$ are defined as follows:

$$ \left.\begin{aligned} {}^{b}x &= p_x \\ {}^{b}y &= p_y \\ {}^{b}z &= p_z \\ {}^{b}\alpha &= \text{atan 2}\,(n_y, n_x) \\ {}^{b}\beta &= \text{atan 2}\,(-n_z, \cos{}^{b}\alpha n_x + \sin{}^{b}\alpha n_y) \\ {}^{b}\gamma &= \text{atan 2}\,(\sin{}^{b}\alpha a_x - \cos{}^{b}\alpha a_y, \cos{}^{b}\alpha o_x - \sin{}^{b}\alpha o_y) \end{aligned}\right\} \qquad (4.8) $$

A simpler form of ${}^{b}\gamma$ can be obtained by referring to Equation (3.35) whereby:

$$ {}^{b}\gamma = \text{atan 2}\,(o_z, a_z) \qquad (4.9) $$

The components of ${}^{b}T$ for the six degree of freedom anthropomorphic arm are as follows:

$$
\begin{aligned}
p_x &= C_1[C_{234}\, l_4 + C_{23} l_3 + C_2 l_2] + t\,[C_1 C_{234} S_5 + S_1 C_5] \\
p_y &= S_1[C_{234}\, l_4 + C_{23} l_3 + C_2 l_2] + t\,[S_1 C_{234} S_{54} - C_1 C_5] \\
p_z &= S_{234} l_4 + 523 l_3 + 52 l_2 + t\,[S_{234} S_5] \\
n_x &= C_1 C_{234} S_5 + S_1 C_5 \\
n_y &= S_1 C_{234} S_5 - C_1 C_5 \\
n_z &= S_{234} S_5 \\
a_x &= C_1[C_{234} C_5 C_6 - S_{234} S_6] - S_1 S_5 C_6 \\
a_y &= S_1[C_{234} C_5 C_6 - S_{234} S_6] + C_1 S_5 C_6 \\
a_z &= S_{234} C_5 C_6 + C_{234} S_6 \\
o_z &= S_{23} + C_5 C_6 + C_{234} S_6
\end{aligned}
\qquad (4.10)
$$

Where, for simplification, S_i, C_i represent $\sin(\theta_i)$, $\cos(\theta_i)$ and $S_{j\ldots k}$, $C_{j\ldots k}$ represent $\sin(\theta_j+\ldots+\theta_k)$ and $\cos(\theta_j+\ldots+\theta_k)$ respectively.

For comparitive purposes, the number of arithmetic operations required to define the direct kinematics are presented in Table 4.2.

4.3 EVALUATION OF THE INVERSE KINEMATIC EQUATIONS

The methods for evaluating the inverse kinematic equation set can be divided into three main techniques: recursive, iterative, and direct.

An example of a recursive technique is that developed at the University of Genova, [Gaglio 1981]. This method is easily applied to

the simpler geometric configurations of six dof manipulators, but has specific relevance to robots with greater than six dof. The technique relies on pre-defined knowledge of both the required hand position and orientation, and arm configuration. Given each new configuration the recursive algorithm achieves the desired geometry by a series of rotations of each degree of freedom. If the manipulator is redundant, in that there are more than six degrees of freedom, the technique can incorporate additional criteria (eg to constrain the robot arm within some "safe" working volume, or to optimally adjust the arm configuration while executing a trajectory, [Benati 1980]). The method of expressing how an orientation of a body is modified by an assigned rotation about an arbitrary axis, is achieved by the use of Rodrigues vectors and the associated vector algebra.

Methods of solving the inverse kinematic problem using iterative techniques, [Derby 1982], are predominantly based on the Unified Theory of Mechanisms proposed by Duffy, [Duffy 1980], [Lin 1982]. The theory uses the geometric laws of spherical triangles as expressions for direction cosines, relating to each link of a manipulator. The laws pertaining to spherical polygons are extended to dual angles, which allows the description of the corresponding spatial polygons to be incorporated into a unified mathematical form. The application of this method of kinematic analysis yields a single equation, describing the spatial configuration of a manipulator. The equation is in the form of a polynominal specific to that geometric configuration. The major disadvantage of this technique is that the solution of the roots of a polynomial over degree four, requires an iterative approach. The degree of polynomials describing the majority of industrial robots are either

eight or sixteen, and as such would require an iterative on-line algorithm for real time control.

There are three major techniques used for direct evaluation of the inverse kinematics: dual number quarternions, [Yang 1964], matrices, [Paul 1981] and direct geometric analysis. The major difference between the use of quarternions or matrices, involves the mathematics used to describe the spatial relationship of the links of a manipulator. The disadvantage of matrices is that they are moderately expensive to store, unlike quarternions, and that computations on them require more operations than quaternions if describing rotational operators. However matrix representations are easier to understand, since link co-ordinate frames can be composed using the ordinary rules of matrix multiplication. When applied to the analysis of inverse kinematics, a marginal reduction in processing requirements is obtained by the use of matrix methods, [Taylor 1979]. The analytical derivation of the inverse kinematics using a matrix technique is illustrated in Section (4.3.2).

The use of direct geometric analysis provides the most compact form of the inverse kinematic equations. These require fewer arithmetic functions when used for real time control of a manipulator. The equation sets derived by the author for the majority of five and six dof manipulators are presented in Section (4.3.3).

There are also techniques for manipulator control that do not require the computation of inverse kinematics. These are however more computationally intensive than control methods involving inverse kinematics. An example of this form of control is the Pseudo Resolved Motion Rate Technique.

4.3.1 Pseudo Resolved Motion Rate Technique

This method of mapping co-ordinate space to joint space is based on manipulator control concept proposed by Whitney, [Whitney 1972]: Resolve Motion Rate Control. The advantage of this technique is the fact that the inverse kinematics need not be solved directly.

The Pseudo Resolved Motion Rate (PRMR) technique involves the Jacobian matrix of the kinematic system; $\underline{J}(\theta)$. Expressing the mapping relationship between $\underline{X}$ and $\underline{\theta}$ as:

$$\underline{X} = \underline{X}(\theta) \tag{4.11}$$

the Jacobian is obtained by the differentiation of this relationship, yielding:

$$\dot{\underline{X}} = J(\theta)\ \dot{\underline{\theta}} \tag{4.12}$$

where the elements of $J(\theta)$ relate the change in a world co-ordinate parameter for a given change in a joint parameter. A typical element of the Jacobian is of the form:

$$J_{ij} = \frac{\partial X_i}{\partial \theta_j} \tag{4.13}$$

Such that:

$$\delta X_i \approx \frac{\partial X_i}{\partial \theta_j}\ \delta\theta_j \tag{4.14}$$

For a non-redundant manipulator, the Jacobian is non-singular and may be inverted to give:

$$\dot{\underline{\theta}} = J^{-1}(\theta)\ \dot{\underline{X}} \tag{4.15}$$

$$\delta\underline{\theta} \approx J^{-1}(\sigma)\ \delta\underline{X} \tag{4.16}$$

where: $$\delta\underline{\theta} = [\delta\theta_1, \delta\theta_2, \delta\theta_3, \delta\theta_4, \delta\theta_5, \delta\theta_6]^T \tag{4.17}$$

$$\delta\underline{X} = [\delta x, \delta y, \delta z, \delta\alpha, \delta\beta, \delta\gamma]^T \tag{4.18}$$

The form of Equation (4.16) can be used to directly control the output variables; $\delta\underline{\theta}$, given $\delta\underline{X}$. However Equation (4.16) is an approximation and not an equality, therefore second order errors occur. These are cummulative, but can be accommodated if the robot accuracy requirements are not too stringent. A control structure can be designed which takes account of the second order errors, [Lien 1980]. This structure compares the actual and required base space robot position by means of the direct kinematics, expressed in Equation (4.11). The actual base space position of the robot at time t, $\underline{X}_{a,t}$ is given by:

$$\underline{X}_{a,t} = \underline{X}(\theta),\underline{\theta}_t \tag{4.19}$$

Specifying the required position of the robot as $\underline{X}_{r,t}$, the second order position error $\delta^2\underline{X}$ can be computed from:

$$\delta^2\underline{X}_t = \underline{X}_{r,t} - \underline{X}_{a,t}$$

$$= \left[\int_{t=o}^{t=t} \delta\underline{X}_t\right] - \underline{X}(\theta) \tag{4.20}$$

This term is then incorporated into the controller as a feedback term, (see Figure 4.3).

The advantage of this technique is in the fact that the inverse kinematics need not be solved analytically. This is especially useful if the kinematic structure of the robot is one which does not lend itself to analytic analysis, (as for example when the yaw motion is before the pitch). The disadvantages included the fact that the technique is numerically intensive, (see Table 4.2). Also since the Jacobian is a function of the joint parameters, the inverse must be computed at regular intervals to minimise the correction factor $\delta^2\underline{X}$, which can be seen as a measure of the trajectory error. In some instances the inverse must be computed for each set of increments, [Lien 1980]. Direct calculation of the Jacobian, and its inverse is time consuming, therefore techniques are incorporated which lessen the amount of processing required. These are discussed further in Section 4.4.

It is possible to combine the original concept of Resolved Motion Rate Control (RMR) and PRMR techniques to give an hybrid controller, [Uchiyama 1982]. The controller uses RMR to provide trajectory control, and switches at the end of a demand vector to PRMR for final positioning control.

4.3.2 Analytical Solution of Inverse Kinematics Using 'A' Matrices

The direct kinematic equation relating the base matrix ${}^{b}T$ to the A matrices for each robot link, has been introduced, (Section (4.2)):

$$ {}^{b}T = A_1 A_2 A_3 A_4 A_5 A_6 E \tag{4.21} $$

This equation can also be written in the form

$$ {}^{b}T = U_1 E \tag{4.22} $$

where, in general, the matrix U_n describes the position and orientation of link six with respect to link co-ordinate frame n-1:

$$ U_n = A_n A_{n+1} \cdots A_6 \tag{4.23} $$

The five matrix equations used to define the joint parameters are obtained as follows, [Paul 1981]:

$$ \left.\begin{aligned} A_1^{-1}\ {}^{b}T &= U_2 \\ A_2^{-1} A_1^{-1}\ {}^{b}T &= U_3 \\ A_3^{-1} A_2^{-1} A_1^{-1}\ {}^{b}T &= U_4 \\ A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1}\ {}^{b}T &= U_5 \\ A_5^{-1} A_4^{-1} A_3^{-1} A_2^{-1} A_1^{-1}\ {}^{b}T &= U_6 \end{aligned}\right\} \tag{4.24} $$

The matrix elements of the left hand sides of these equations are functions of the elements of ${}^{b}T$ and of the first n-1 joint variables. The matrix elements of the righthand sides are either zero, constants, or functions of the nth to 6th joint variables. As matrix equality implies element by element equality twelve equations are obtained from each matrix equation. Equating elements of these matrix equations frequently result in equations yielding joint variables explicitly.

Referring to the six degree of freedom anthropomorphic arm, illustrated in Figure 4.2, the application of this technique yields the following values for the joint parameters:

$$
\begin{aligned}
\theta_1 &= \text{atan2}\ (p_y,\ p_x) \\
\theta_{234} &= \text{atan2}\ (a_z,\ C_1 a_x + S_1 a_y) \\
p'_x &= C_1 p_x + S_1 p_y - C_{234}\, l_4 \\
p'_y &= p_z - S_{234} l_4 \\
C_3 &= p'^2_x + p'^2_y - l_3^2 - l_2^2 / 2 l_2 l_3 \\
\theta_3 &= \text{atan2}\ ((1-C_3^2)^{1/2},\ C_3) \\
\theta_2 &= \text{atan2}\ ((C_3 l_3 + a_2)\, p'_y - S_3 l_3 p'_x,\ (C_3 l_3 + a_2)\, p'_x + S_3 l_3 p'_y) \\
\theta_4 &= \theta_{234} - \theta_3 - \theta_2 \\
\theta_5 &= \text{atan2}\ (C_{234}(C_1 a_x + S_1 a_y) + S_{234}\, a_z,\ S_1 a_x - C_1 a_y) \\
\theta_6 &= \text{atan2}\ (-C_5(C_{234}(C_1 o_x + S_1 o_y) + S_{234} o_z) + S_5(C_1 o_x - C_1 o_y), \\
&\quad - S_{234}\,(C_1 o_x + S_1 o_y) + C_{234} o_z)
\end{aligned}
\qquad (4.25)
$$

where:

$$
\begin{aligned}
p_x &= {}^b x - t C_\alpha C_\beta \\
p_y &= {}^b y - t S_\alpha C_\beta \\
p_z &= {}^b z - S_\beta \\
o_x &= C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma \\
o_y &= S_\alpha S_\beta S_\gamma + C_\alpha C_\gamma \\
o_z &= C_\beta S_\gamma \\
a_x &= C_\alpha C_\beta \\
a_y &= S_\alpha C_\beta \\
a_z &= -S_\beta
\end{aligned}
\qquad (4.26)
$$

The angles α, β, γ are specified with respect to the base frame, but the superscript has been ommitted for clarity.

The arithmetic requirements of this technique are given in Table 4.2.

4.3.3 Direct Evaluation of Inverse Kinematics

The method of direct kinematic analysis of the inverse kinematics yields explicit solutions of the robot joint parameters. Devised by the author, they involve the transmission of parameters through a mechanical node, termed the wrist point. The parameters transferred between the secondary and primary axes are the position, lateral orientation, and variable and fixed vertical orientation ($\underline{W}$, ϕ, σ and ω respectively). The direction of transfer is dependent upon the configuration of secondary axes.

The method of direct kinematic analysis reduces substantially the mathematical task of evaluating the inverse kinematics. This is especially important when the inverse kinematics are evaluated in real time for continuous path motion.

The solutions to the inverse kinematics, are given for two forms of three dof wrist (RPR), (PYR), and for a two dof wrist (RP) with a non linear end effector (such as a welding rod), and a simple gripper. The solutions for primary axes are given for anthropomorphic, spherical polar, cylindrical polar, and cartesian configurations. A full discussion of the method of analysis is given in Appendix B.

In the following solutions the abreviations S, C, and T refer to the sine, cosine and tangent of the subscript angles.

(i) PYR three dof wrist

The configuration and relevant parameters of the PYR wrist are illustrated in Figure 4.4a. The following parameters are defined:

$$x' = {}^{b}x - l_t C_\alpha C_\beta \tag{4.27}$$

$$y' = {}^{b}y - l_t S_\alpha C_\beta \tag{4.28}$$

$$\phi = \text{atan2}\,[y', x'] \tag{4.29}$$

Note however that in this, and all subsequent wrist equation sets, the value of ϕ may be redefined by the base axes. The value of σ is given by:

$$\sigma = \text{atan2}\,[T_\beta, C_{\alpha-\phi}] \tag{4.30}$$

The joint parameters are then defined as follows:

$$\theta_4 = \pi - (\omega + \sigma) \qquad (\omega \text{ obtained from primary axes}) \tag{4.31}$$

$$\theta_5 = \text{atan2}\,[T_{\alpha-\phi} C_\sigma, 1] \tag{4.32}$$

$$\theta_6 = \text{atan2}\,[S_5 T_\sigma, 1] + \gamma \tag{4.33}$$

The wrist point is then defined by:

$$Wx = x'-l_4C_\sigma C_\phi \tag{4.34}$$

$$Wy = y'-l_4C_\sigma S_\phi \tag{4.35}$$

$$Wz = {}^bz+l_tS_\beta+l_4S_\sigma \tag{4.36}$$

(ii) RPR three dof wrist

The RPR wrist configuration is illustrated in Figure 4.4b. For this wrist configuration the end of the primary axes is taken to included joint 4, as this joint is always co-incident with link 3 of the primary axes. The wrist point is defined as:

$$Wx = {}^bx-l_tC_\alpha C_\beta \tag{4.37}$$

$$Wy = {}^by-l_tS_\alpha C_\beta \tag{4.38}$$

$$Wz = {}^bz+l_tS_\beta \tag{4.39}$$

For the RPR wrist configuration $\sigma=\omega$ and is obtained from the primary axes. The joint values for the wrist are then obtained as follows:

$$\phi = \text{atan2 [Wy, Wx]} \tag{4.40}$$

$$K = S_\beta C_\sigma-C_\beta C_\alpha-\phi S_\sigma \tag{4.41}$$

$$\theta_4 = \text{atan2 [K,}C_\beta S_\alpha-\phi] \tag{4.42}$$

$$\theta_5 = \text{atan2}\,[KS_4+C_\beta S_{\alpha-\phi}C_4,\ C_\beta C_{\alpha-\phi}C_\sigma+S_\beta S_\sigma] \tag{4.43}$$

$$\theta_6 = \text{atan2}\,[C_5C_\sigma S_4-S_\sigma S_5,\ C_\sigma C_4] + \gamma \tag{4.44}$$

(iii) PR two dof wrist

The PR configuration is illustrated in Figure 4.5a. As with RPR, the wrist point excludes joint 4, as this is included in the primary axes. The wrist parameters are obtained as follows:

$$Wx = {}^bx-l_tC_\alpha C_\beta \tag{4.45}$$

$$Wy = {}^by-l_tS_\alpha C_\beta \tag{4.46}$$

$$Wz = {}^bz+l_tS_\beta \tag{4.47}$$

$$\phi = \text{atan2}\,[Wy,\ Wx] \tag{4.48}$$

$$\sigma = 2\text{atan2}\,[S_\beta \pm (S_\beta^2+C_\beta^2C_{\alpha-\phi}^2-C_\delta^2)^{\frac{1}{2}},\ C_\delta+C_\beta C_{\alpha-\phi}] \tag{4.49}$$

The dual solution arises from the two physical possibilities that fulfil the required parameters. In most cases the choice of sign can be taken as the sign of β. The joint values are defined as:

$$\theta_4 = \pi-(\sigma+\omega) \tag{4.50}$$

$$\theta_5 = \text{atan2}\,[S_{\alpha-\phi},\ T_\beta C_\sigma-S_\sigma C_{\alpha-\phi}] \tag{4.51}$$

The physical restrictions imposed by using a simple gripper as the tool, (see Figure 4.5b), produce the trivial solutions for this configuration as:

$$\phi = \text{atan2}\,[{}^{b}y, {}^{b}x] \tag{4.52}$$

The value of ϕ, as before, may be re-defined by the base axes.

$$Wx = {}^{b}x - l_t C_\phi C_\beta \tag{4.53}$$

$$Wy = {}^{b}y - l_t S_\phi C_\beta \tag{4.54}$$

$$Wz = {}^{b}z + l_t S_\beta \tag{4.55}$$

$$\theta_4 = \pi - (\omega + \beta) \tag{4.56}$$

$$\theta_5 = \gamma \tag{4.57}$$

It is possible to combine any of wrist or secondary axes configuration with any of the primary axes configurations shown in Figure 4.6. Considering the four main types of primary axes configurations, the solutions, are as follows:

(i) Anthropomorphic primary configuration

Parameters relevant to the anthropomorphic robot are illustrated in Fiugre 4.6d. The following are defined:

$$R_1{}^2 = Wx^2 + Wy^2 \tag{4.58}$$

$$R_2^2 = R_1^2+(W_z-l_1)^2 \tag{4.59}$$

$$\psi = \text{atan2}\,[W_z-l_1, R_1] \tag{4.60}$$

The value of the primary joints are as follows:

$$\theta_2 = \frac{\pi}{2} - \arccos\left[\frac{l_2^2+R_2^2-l_3^2}{2l_2R_2}\right] - \psi \tag{4.61}$$

$$\theta_3 = \frac{\pi}{2} - \arccos\left[\frac{l_2^2+l_3^2-R_2^2}{2l_2l_3}\right] \tag{4.62}$$

However if, as in many cases, l_2 and l_3 are equal, signifying equal limb length, the solution simplifies to:

$$\theta_3 = \pi - \arcsin\left[\frac{R_2}{2l}\right] \tag{4.63}$$

$$\theta_2 = \frac{\theta_3}{2} + \psi \tag{4.64}$$

The value of parameters passed through the wrist point are:

$$\phi = \phi \tag{4.65}$$

$$\omega = \theta_2+\theta_3 \tag{4.66}$$

(ii) Spherical polar primary configuration

The spherical polar configuration is illustrated in Figure 4.6c.

Given:

$$R_1^2 = Wx^2+Wy^2 \tag{4.67}$$

The robot joint parameters are as follows:

$$R_S = [R_1^2+(Wz-l_1)^2]^{\frac{1}{2}} \tag{4.68}$$

$$\theta_S = \text{atan2}\,[Wy,Wx] \tag{4.69}$$

$$\psi_S = -\text{atan2}\,[Wz-l_1,R_1] \tag{4.70}$$

The parameters passed through the wrist point are as follows:

$$\phi = \phi \tag{4.71}$$

$$\omega = \pi/2 + \psi_S \tag{4.72}$$

(iii) Cylindrical polar primary configuration

The cylindrical polar configuration is illustrated in Figure 4.6b. The joint parameters are defined as follows:

$$R_C = (Wx^2 + Wy^2)^{\frac{1}{2}} \tag{4.73}$$

$$\theta_C = \text{atan2}\,[Wy,Wx] \tag{4.74}$$

$$Z_C = Wz \tag{4.75}$$

The parameters passed through the wrist point are as follows:

$$\phi = \phi \tag{4.76}$$

$$\omega = \pi/2 \tag{4.77}$$

(iv) Cartesian primary configuration

The cartesian configuration is illustrated in Figure 4.6a. The primary joint parameters are defined as follows:

$$X_L = Wx \tag{4.78}$$

$$Y_L = Wy \tag{4.79}$$

$$Z_L = Wz \tag{4.80}$$

The wrist parameters are set as follows:

$$\phi = \phi \tag{4.81}$$

$$\omega = \pi/2 \tag{4.82}$$

The above equation sets have been verified by comparing the results of the inverse kinematic analysis with established techniques. For comparitive purposes the mathematical requirements of a six dof mechanical arm are given in Table 4.2. The comparison takes on more significance if the relative costs, in terms of processing time, of the

various functions are taken into account. This factor is discussed in more detail in Chapter 7.

4.3.4 Non-aligned Tool Frame

In the case of six (or greater) dof arms, a tool that does not reflect the co-ordinate frame of the end effector can be easily accommodated. A tool whose frame co-ordinate matrix is given by T can be incorporated within the frame transformation section of the robot controller. The tool configuration is included as a static frame of the form T^{-1}. Where for a homogeneous matrix given:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.83}$$

then:

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -p.n \\ o_x & o_y & o_z & -p.o \\ a_x & a_y & a_z & -p.a \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.84}$$

where "." represents the vector dot product.

4.4 DISCUSSION OF THE INVERSE KINEMATICS

There are a number of problems associated with the application of

the inverse kinematic equations. These problems are of two types:

(i) Mathematical multiplicity of solutions.

(ii) Equation degeneracy.

Mathematical multiplicity refers to the fact that there is usually more than one configuration of the robot which will achieve a desired position and orientation of the end effector. These include such items as an "over" or "under" configuration of limbs two and three, and if the "shoulder" can reach back on itself, creating two base positions 180^{o} apart. These ambiguities are either solved by restraining the inverse kinematic equations, or specifying configuration information when programming the robot. Another problem associated with mathematical evaluation, is the definition of rotary joints within the range $\pm 180^{o}$. Unexpected motions of the robot may occur if the motion requires a change from $+180^{o}$ to +181. Instead the robot may rotate 359^{o}, to -179^{o}. This is a more difficult situation to anticipate and requires additional information relating to the continuity required of a motion.

The problems of equation degeneracy can best be described with reference to the Jacobian matrix; J. Solution degeneracy arises when the matrix J is non-invertable. Two cases arise:

(i) The manipulator is in a singular configuration. In this case the determinant of J is zero and J cannot be inverted. This situation is dependent only on the configuration of the robot.

(ii) The manipulator is redundant. In this case J is not square and again cannot be inverted. This situation arises when a manipulator has greater than six dof, or when manipulators are co-operating in a common space.

4.4.1 Manipulator Singularity

There are certain configurations of a manipulator that are termed singular. Mathematically, singular configurations represent a discontinuity in the relationship between joint and base space. Physically, singular configurations represent positions of the manipulator where two or more axes can achieve the same motion of the end effector. This can be interpreted as a form of redundancy, whereby there are more degrees of freedom than boundary conditions. Relating singularity to the Jacobian, singular configurations are determined when the determinant of the Jacobian is equal to zero. Thus the Jacobian cannot be inverted by classical means. Defining the robot configuration as C, and the singular point, C_s, then:

$$C_s = C \mid_{Det(J)=0} \tag{4.85}$$

Singular positions can also be related to specific joints when the value of joint i; θ_i, is given by:

$$\theta_i = atan2(0,0) \tag{4.86}$$

An example of a singular configuration, arises with the simple gripper 5 dof anthropomorphic robot, when the gripper rotate axis is colinear with the base z axis. The value of θ_1 cannot then be defined explicitly.

Another example is the RPR 3 dof wrist when the pitch joint, θ_5 is equal to zero. In this case parameters are physically constrained such that:

$$\beta = \omega = \varepsilon \tag{4.87}$$

$$\alpha = \theta_1 = \phi \tag{4.88}$$

where β and α are orientation parameters, θ_1 is the base rotation, and ε is the constant defined.

The value of θ_4 is defined by:

$$\theta_4 = \text{atan2}(C_\sigma S_\beta - C_\beta C_{\alpha-\phi} S_\sigma, C_\beta S_{\alpha-\phi}) \tag{4.89}$$

Substituting the relevant parameters, for this arm geometry given, yields:

$$\theta_4 = \text{atan2}(C_\varepsilon S_\varepsilon - C_\varepsilon C_0 S_\varepsilon, C_\varepsilon S_0) \tag{4.90}$$

$$\theta_4 = \text{atan2}(0,0) \tag{4.91}$$

In this case a rotation of θ_4 achieves the same effect as rotating θ_6.

The concept of singularity can be expanded to give a measure of the "quality" of control of a robot. Although absolute redundancy occurs at singular points, there is a reduction in controllability near, or approaching, singular configurations. Figure 4.7 shows the base joint demand/time relationship when a trajectory passes near a singular point.

As the trajectory nears a singular point, the velocity demands increase drastically. Since a dof is in effect lost, the maneouverability of the robot is also reduced. These two factors combined constitute a reduction in the quality of control.

It has been suggested that a measure of the quality to be expected from co-ordinate transformation, is the condition number of the Jacobian, [Daniel 1983]. The condition number of the Jacobian C[J] is defined as:

$$C[J] = \frac{\sigma_{max}[J]}{\sigma_{min}[J]} \qquad (4.92)$$

where $\sigma[J]$ is a singular value of J, defined as the positive square roots of the eigenvalues of JJ^T.

The value of the condition number gives information on the joint torques required of a given task, and the smoothness of control. Daniel has produced isometric maps of the condition number inverse, against the working volume of the robot. Areas of low or zero value indicate the worst conditions, values of high or one, the best conditions. It is suggested that reference to these maps would provide valuable information concerning the robot path and arm configurations.

The programmer should attempt to avoid singular positions when evaluating a robot trajectory. If a trajectory nears a singular configuration the excessive joint demands can be accomplished within the control structure as discussed in Chapter 7. However there exist smooth

trajectories in base space near to singular positions where C[J] becomes so large that jittery motion will take place no matter how slowly the arm is programmed to move.

One solution to the problem of singularity is to incorporate redundancy into the manipulator. That is to use manipulators with greater than six dof where the additional dof is not a simple additional motion of the base. This is the technique that the human arm uses to avoid singular positions.

4.4.2 Manipulator Redundancy

The method commonly adopted to solve the inverse kinematic equations for a redundant manipulator, is the method of Generalised Inverses. Considering the Jacobian and Equation (4.12):

$$\dot{\underline{X}} = J(\theta)\dot{\underline{\theta}} \tag{4.93}$$

The generalised inverse of J is every matrix G which satisfies the equation, [Coiffet 1981]:

$$JGJ = J \tag{4.94}$$

Equation (4.93) can then be written as:

$$\dot{\underline{\theta}} = G\dot{\underline{X}} \tag{4.95}$$

One form of the generalised inverse is that of Principle Variables. The attraction of this solution is its simplicity. It involves holding some

variable or variables fixed. The generalised inverse is obtained by taking a submatrix of J; J_S, where J_S is square. This involves extracting vectors from J to leave J_S. The vectors extracted refer to the principle variables to be held constant. The generalised inverse is then obtained by inverting J_S and substituting for the variables held constant.

An example of the use of principle variables is for a redundant manipulator required to work in confined spaces. Aspragathos, [Aspragathos 1983], illustrates this technique for a five dof primary axis arm which is two fold redundant. The arm is required to work within a toroidal workstation, access to which is via a small port. The task of approaching, and entering the workstation is divided into various phases dependent on current position. For each phase, specific principle variables are held constant allowing solution of the inverse kinematics. Thus the manipulator is co-ordinated to allow complete passage into the workstation.

Since an infinite number of generalised inverses exist, it becomes possible to choose a solution which minimises some specific criterion. A quadratic technique often employed is that to minimise some criterion C, where:

$$C = \frac{1}{2} \dot{\underline{\theta}}^T W \dot{\underline{\theta}} \qquad (4.96)$$

where W is a square, positive weighting function.

It can be shown that the solution to Equation (4.??),(found through the use of Lagrange multipliers, [Fournier 1980]), given by:

$$\underline{\dot{\theta}} = W^{-1}J^T(JW^{-1}J^T)^{-1}\underline{\dot{X}} \qquad (4.97)$$

In this case the generalised inverse G is defined as:

$$G = W^{-1}J^T(JW^{-1}J^T)^{-1} \qquad (4.98)$$

A specific use of the weighting funciton is the unit matrix, I. Using I as the weighting function, minimising C represents the minimisation of the Euclidian norm of the joint articulation:

$$C = \frac{1}{2}\ \underline{\dot{\theta}}^2 \qquad (4.99)$$

Or, considering the incremental motion of each joint, $\delta\theta_i$ for an n link manipulator:

$$C = \frac{1}{2} \sum_{i=1}^{n} \dot{\Delta\theta}_i^2 \qquad (4.100)$$

This minimises the displacement of each joint.

This technique has been employed for the control of a dual manipulator system performing co-operative tasks, [Brooks 1982]. Brooks considers a combined generalised inverse, and kinematic analysis. This involves using the generalised inverse to specify certain joints, and using inverse kinematics to solve the remaining joints. This has the

advantage of eliminating the positional errors associated with the use of the Jacobian alone.

4.4.3 Matrix Inversion

The inversion of matrices using classical mathematics is problematic for two reasons. First; the process of inverting a matrix is mathematically, and computationally intensive. (Since the Jacobian is a funciton of arm configuration it must be re-evaluated at frequent intervals). Secondly; the Jacobian often has a number of elements which are zero or approaching zero. These can cause ill-conditioning of the determinant, and inversion degeneracy. Therefore a number of methods have been proposed to facilitate the inversion of the Jacobian matrix:

(i) The storage of several pre-computed Jacobian inverses [Whitney 1972], [Horn 1977]. This method consists of generating a table of Jacobian inverses as a function of joint parameters. A specific inverse J^{-1}, is evaluated by interpolating between values obtained from the table. Traditionally, the major disadvantage of this method is the number of values that must be stored. However this is becoming less of a problem as low cost, rapid memory storage becomes commercially available.

(ii) The Jacobian/Jacobian inverse relationship, [Renaud 1979]. This method consists of inverting the matrix once, at time zero. Subsequent inverses are then calculated by means of the relationship:

$$J^{-1}{}_{t+\delta t} = J^{-1}{}_{t}\,(2I - J_{t+\delta t} J^{-1}{}_{t}) \qquad (4.101)$$

where the subscripts refer to time, and I is the unit matrix.

(iii) Recursive technique to solve specifically for the pseudo-inverse; J^*. The pseudo-inverse provides a solution which minimises the Euclidian norm of movements, $\Sigma\delta\theta_i{}^2$ mentioned previously. Defining a sub-matrix J_k of J, where the subscript refers to the first k columns of the Jacobian, a recursive relationship exists relating $J_k{}^*$, and $J_{k-1}{}^*$ [Greville 1960]:

$$J_k = \begin{bmatrix} J_{k-1}{}^* - D_k B_k^T \\ B_k^T \end{bmatrix} \qquad (4.102)$$

and $D_k = J_{k-1}{}^* \underline{j}_k$

$$C_k = \underline{j}_k - J_{k-1} D_k$$

$$\text{if } C_k \neq 0 \;;\; B_k^T = C_k^T / C_k^T C_k \qquad (4.103)$$

$$\text{if } C_k = 0 \;;\; B_k^T = D_k^T J_{k-1}{}^* / (1 + D_k^T D_k)$$

where $\underline{j}_k$ is the k th column of the Jacobian J.

The initialisation of the algorithm is based on the first column of J. The value of $J_1{}^*$ is determined as follows:

$$\text{if } \underline{j}_1 = 0 \qquad J_1{}^* = 0^T$$

$$\text{if } \underline{j}_1 \neq 0 \qquad J_1{}^* = \underline{j}_1^T / (j_1^T j_1) \qquad (4.104)$$

Thus, having determined J_1^* the subsequent pseudo-inverses can be obtained by the recursive relationship of Equation (4.102). The form of the pseudo-inverse once evaluated can be applied continuously in the control of the manipulator, updating only the elements which are a function of configuration.

The robot controller, developed in this thesis, does not at present control redundant manipulators. However the philosophy employed is designed to allow inclusion at a later stage, if required.

4.5 DERIVATIVE MOTION

4.5.1 Introduction

Considering the general family of "manipulators", the open loop chain can be considered as a set of kinematic links, each with an associated drive, (motor). Furthermore it is possible to distinguish three basic groups of manipulators by considering a classifying equation relating the kinematic links and motors of the manipulator, [Kalabin 1978]. The classifying factor X is identified as follows:

$$X = \omega_i - \omega_{mi} \qquad i = 1, 2 \ldots n \qquad (4.105)$$

where ω_i is the number of degrees of freedom of the $_i$-th element of the open loop chain (relative to the base element); ω_{mi} is the number of degrees of freedom of the element containing the motor which drives element i; n is the number of degrees of freedom of the manipulator.

The three main classifying groups are as follows:

(i) X<1: This infers that the motor is on the element being driven (X=0), or even that the motor is attached to an element having greater mobility than the element driven by the motor. (As for example in some types of excavator, where the motor powering the caterpillar drive is contained in a tower which can rotate relative to the vertical axis). For manipulators, this type of kinematic chain is not used in practice.

(ii) X=1: This infers that the motor element is connected directly with the element being driven (either directly or via transmissions). This often means that the actuating element is driven by a motor (and reduction unit) incorporated directly in the moving joint of the manipulator (cf Cincinnati Milacron T3).

(iii) X>1: This infers that the motor is attached to an element having less degrees of freedom than the element of the kinematic pair being driven by it. Transmission of the motion is provided by various types of mechanisms, via the joints of open loop chain. (In the particular case where X=i then the motor is housed in the base element). It is this class of manipulator which experiences derivative motion.

Derivative motion is the term given to the motion of a specific axis which occurs as a result of the motion of some other axis, (or axes). The phenomenon is linearly independent. For example if the motion of

axis four is derivatively linked to axes one, two and three, then it is possible to look at the derivative motion due to each axis seperately. Each derivative motion can be added to yield a total derivative motion. The phenomenon of derivative motion constitutes a set of liner equations, easily shown in matrix form.

It is useful here to discriminate between two major forms of derivative motion; planar transmission, and differential gear.

Planar transmission is the term given to the transfer of motion through an arm via each joint. The transmission mechanism can be gear (Cincinnati Milacron; T3 746), disc and rod (ASEA; IRb 6 robot), or chain/belt (IC, Hitachi PW10II robot). The use of gear, chain, or belt allows a reduction between the joints, the disc and rod necessitates a 1:1 ratio between the joints.

Considering the motion of gears as illustrated in Figure 4.8a, the derivative motion of gear θ_0 is defined as the change in angular orientation of gear 2 relative to the connecting link 1:

$$\theta_o = z\Delta\theta_i = z(\theta_i' - \theta_i) \qquad (4.106)$$

Where θ_i is the absolute angular orientation of the connecting link between gears 1 and 2. $\Delta\theta_i$ is the change in angular orientation of the link; z is the ratio of gear teeth:

$$z = \frac{z_1}{z_2} \qquad (4.107)$$

Gear 1 is assumed held fixed in space.

Considering the motion of axes linked by chain, as illustrated in Figure 4.8b. The equation of the derived motion is as follows:

$$\theta_D = -z\Delta\theta_i = -z(\theta_i' - \theta_i) \qquad (4.108)$$

where as before θ_i is the absolute value of the angular orientation of the connecting link; $\Delta\theta_i$ is the change in orientation of the connecting link; z is the ratio of sprocket teeth:

$$z = \frac{z_1}{z_2} \qquad (4.109)$$

The negative sign reflects the fact that the rotation of sprocket 2 is in the opposite direction from the motion of the connecting link.

Equation (4.108) is also that which applies to the disc and rod arrangement. This can be considered as the specific case of the chain transmission with the gear ratio between each joint equal to unity. This has the interesting effect of retaining a fixed absolute orientation in space for an element of a robot. Thus a kinematic configuration which employs this form of transmission arrangement could be used to reduce the number of driven degrees of freedom required of a manipulator. By using only three degrees of freedom and the transmission mechanism discussed above, a large working envelope in 3-D space can be achieved with a fixed orientation of the end effector. This would have numerous applications in, for example, pick and place, or simple drilling applications.

Differential gear derivative motion is that form of motion most often experienced in wrist assemblies. Consider Figure 4.9a; the assembly comprises a common link; 1, gear 1 and gear 2. The input parameters are θ_{i1} and θ_{i2}. The physical output parameters are θ_{o1} and θ_{o2}. The relevant kinematic relationships are as follows, [Mizutani 1981]:

$$\theta_{o1} = \theta_{i1} \tag{4.100}$$

$$\theta_{o2} = a\theta_{o1} - b\theta_{o2} \tag{4.111}$$

where:

$$a = r_{i1}/r_{o2} \tag{4.112}$$
$$b = r_{i2}/r_{o2}$$

Thus the output θ_{o1} is dependent only on θ_{i1} whereas θ_{o2} depends on both θ_{i1} and θ_{i2}. This can be expressed more succinctly in matrix form. (Which constitues a sub matrix of the overall derivative motion matrix discussed later).

$$\begin{bmatrix} \theta_{o1} \\ \theta_{o2} \end{bmatrix} = \begin{bmatrix} 1 & o \\ a & -b \end{bmatrix} \begin{bmatrix} \theta_{i1} \\ \theta_{i2} \end{bmatrix} \tag{4.113}$$

In fact in equation (4.113) θ_{o1} and θ_{i1} are identical, however θ_{i1} is the input from the drive and θ_{o1} is the output to the load.

Another wrist assembly commonly met, is illustrated in Figure 4.9b. In this case input is via two bevel gears; θ_{i1} and θ_{i2}; the output as before is θ_{o1} and θ_{o2}. In this case the derivative motion matrix is :

$$\begin{bmatrix} \theta_{o1} \\ \theta_{o2} \end{bmatrix} = \begin{bmatrix} a/2 & a/2 \\ -b/2 & b/2 \end{bmatrix} \begin{bmatrix} \theta_{i1} \\ \theta_{i2} \end{bmatrix} \tag{4.114}$$

Again: $a = r_{i1}/r_{o1}$, $b = r_{i2}/r_{o2}$

However for co-ordinate transformation the inverse of the above equations are more important, in that for specific ouput angles it is required to evaluate the motions of the input angles.

Thus inverting the derivative motion matrix for single level arrangement yields:

$$\begin{bmatrix} \theta_{i1} \\ \theta_{i2} \end{bmatrix} = \frac{1}{b} \begin{bmatrix} b & o \\ a & -1 \end{bmatrix} \begin{bmatrix} \theta_{o1} \\ \theta_{o2} \end{bmatrix} \tag{4.115}$$

Similarly for the double level arrangement:

$$\begin{bmatrix} \theta_{i1} \\ \theta_{i2} \end{bmatrix} = \frac{2}{a.b} \begin{bmatrix} b/2 & -a/2 \\ b/2 & a/2 \end{bmatrix} \begin{bmatrix} \theta_{o1} \\ \theta_{o2} \end{bmatrix} \tag{4.116}$$

4.5.2 Analysis Sequence for the Total Derivative Motion Matrix

Using the IC robot as an example the total derivative matrix will be determined. The basic relationship for the planar transmission derivative motion is as follows:

$$\theta_{D,ij} = K_{ij}\Delta\theta_j \tag{4.117}$$

where: $\theta_{D,ij}$ is the derivative motion of joint i due to the motion of joint j; K_{ij} is the inverse of the gear ratio of the transmission medium (gears or chain) between joint i and j; $\Delta\theta_j$ is the change in angle of joint j.

Denoting $\Delta\theta_{rj}$ as the required change in angular position of joint j in space and $\Delta\theta_{aj}$ as the actual change of angular position as seen by the drive motor, then a series of equations can be obtained. The table is simplified if there is no relationship between axes. In this case K is equal to zero. This usually occurs when j>i except for axes 4, and 5 where the effect of differential gear derivative motion occurs. Thus, for the Imperial College robot:

$$\begin{aligned}
\Delta\theta_{a2} &= K_{22}\Delta\theta_{r2} \\
\Delta\theta_{a3} &= K_{32}\Delta\theta_{a2} + K_{33}\Delta\theta_{r3} \\
\Delta\theta_{a4} &= K_{42}\Delta\theta_{a2} + K_{43}\Delta\theta_{a3} + K_{44}\Delta\theta_{r4} + K_{54}\Delta\theta_{r5} \\
\Delta\theta_{a5} &= K_{52}\Delta\theta_{a2} + K_{53}\Delta\theta_{a3} + K_{54}\Delta\theta_{r5} + K_{55}\Delta\theta_{r5}
\end{aligned} \tag{4.118}$$

which on substituting for the θ_a's on the right handside, and configured in matrix form, yields:

$$\Delta\begin{bmatrix}\theta_{a2}\\ \theta_{a3}\\ \theta_{a4}\\ \theta_{a5}\end{bmatrix} = \begin{bmatrix} K_{22} & 0 & 0 & 0 \\ K_{32}K_{22} & K_{33} & 0 & 0 \\ K_{42}K_{22}+K_{43}K_{32}K_{22} & K_{43}K_{33} & K_{44} & K_{45} \\ K_{52}K_{22}+K_{53}K_{32}K_{22} & K_{53}K_{33} & K_{54} & K_{55}\end{bmatrix}\Delta\begin{bmatrix}\theta_{r2}\\ \theta_{r3}\\ \theta_{r5}\\ \theta_{r5}\end{bmatrix} \tag{4.119}$$

As stated above the value of K_{ij} is the inverse of the gear ratio between i and j except for the inter-dependence of axes four and five. The constants for joints four and five can be determined from equation (4.113), substituting the relevant parameters:

$$\Delta \begin{bmatrix} \theta_{a4} \\ \theta_{a5} \end{bmatrix} = \frac{1}{b} \begin{bmatrix} b & o \\ a & -1 \end{bmatrix} \begin{bmatrix} \theta_{r4} \\ \theta_{r5} \end{bmatrix} \qquad (4.120)$$

where θ_{aj} is the actual rotation of the drives at the wrist, these being orientated in the same plane as joints two, and three.

Thus in Equation set (4.119); $K_{44} = 1$

$$K_{45} = 0 \qquad (4.121)$$

$$K_{54} = a/b$$

$$K_{55} = -1/b$$

Note that in all other cases $K_{ii}=1$

The other matrix elements can be obtained from the chain and gear ratios, and have the following values:

$$K_{32} = -0.180517$$

$$K_{42} = -0.024244$$

$$K_{52} = -0.024244$$

$$K_{43} = -0.089536 \qquad (4.122)$$

$$K_{53} = -0.089536$$

$$K_{54} = -1$$

$$K_{55} = -1$$

The signs reflect the relative motions of the joints. However to determine the motion of each joint required to compensate for the derivative motion, the negative value of all non-diagonal elements must be taken. Substituting the above constants into equation (4.119) and negating all non-diagonal elements, yields the following derivative compensation matrix for the Imperial College robot:

$$\Delta \begin{bmatrix} \theta_{a1} \\ \theta_{a2} \\ \theta_{a3} \\ \theta_{a4} \\ \theta_{a5} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0.180517 & 1 & 0 & 0 \\ 0 & -0.008082 & 0.089536 & 1 & 0 \\ 0 & -0.008082 & 0.089536 & 1 & -1 \end{bmatrix} \Delta \begin{bmatrix} \theta_{r1} \\ \theta_{r2} \\ \theta_{r3} \\ \theta_{r5} \\ \theta_{r5} \end{bmatrix} \qquad (4.123)$$

The joint parameter θ_1 has been added for completeness, but does not contribute to any derivative motion. The above matrix must be evaluated, after co-ordinate transformation, to determine the joint values for servo control.

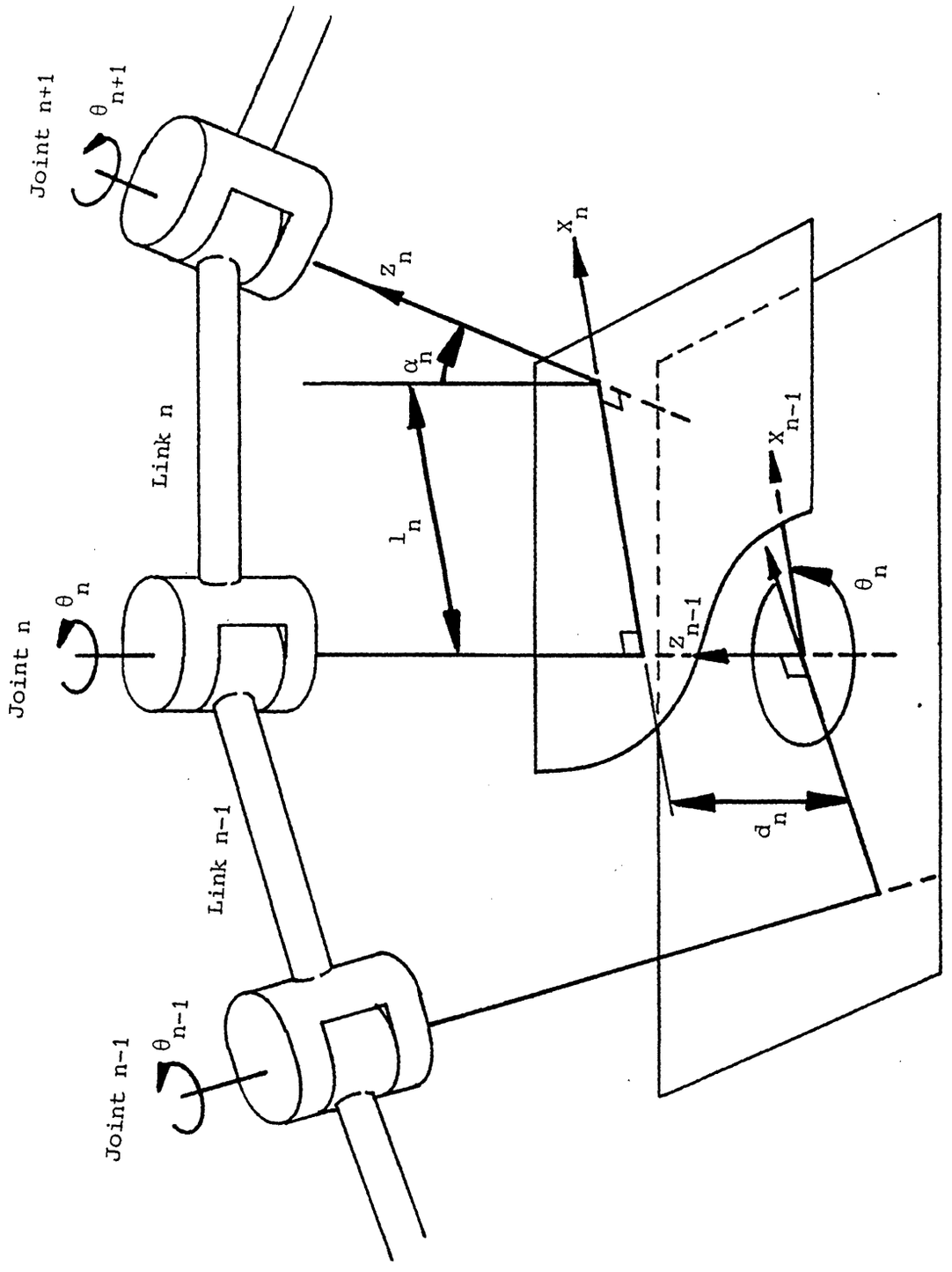


Figure 4.1 Denavit Hartenberg Convention for Link Parameters

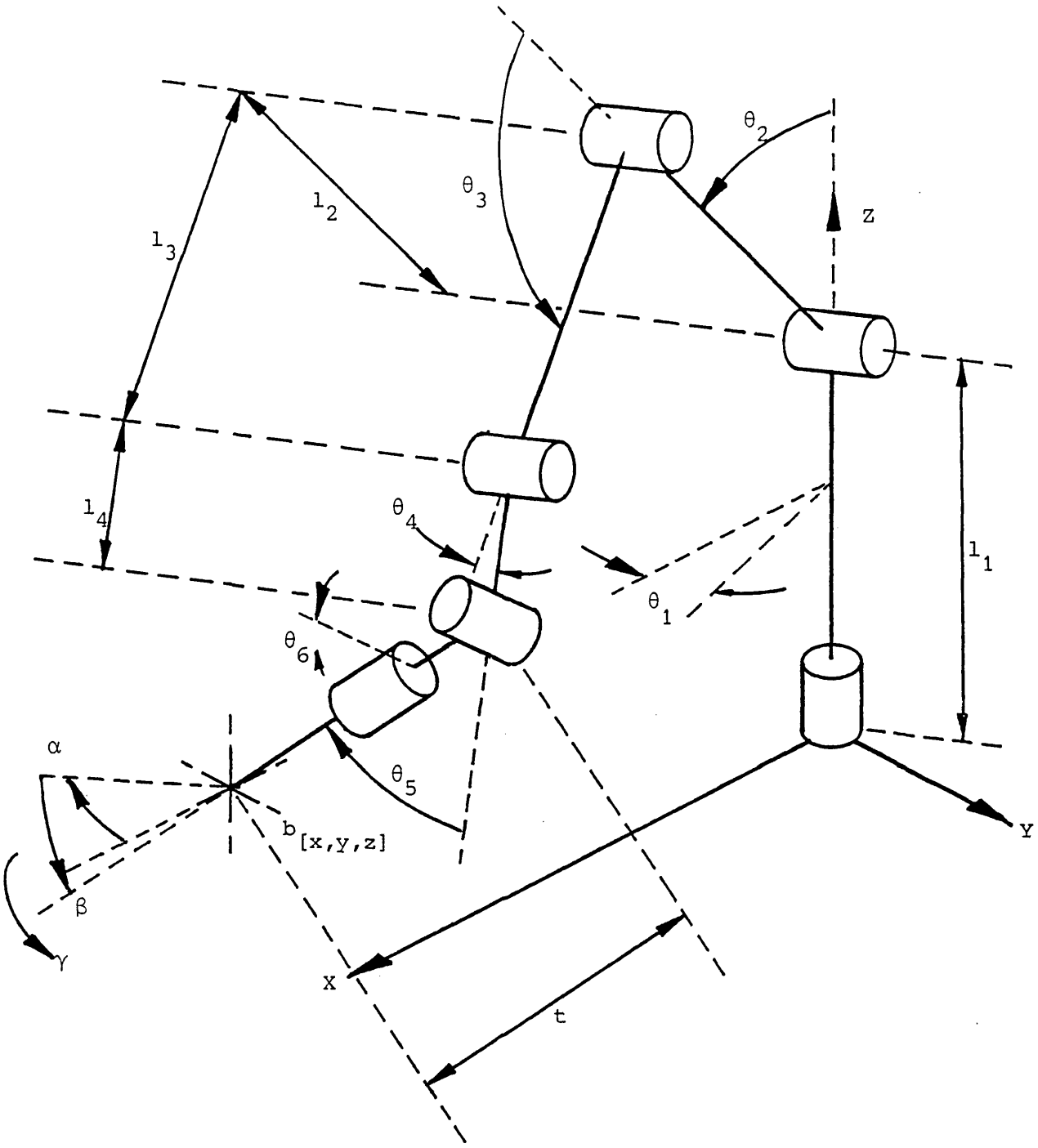


Figure 4.2 Six Degree of Freedom Anthropomorphic Robot Parameters

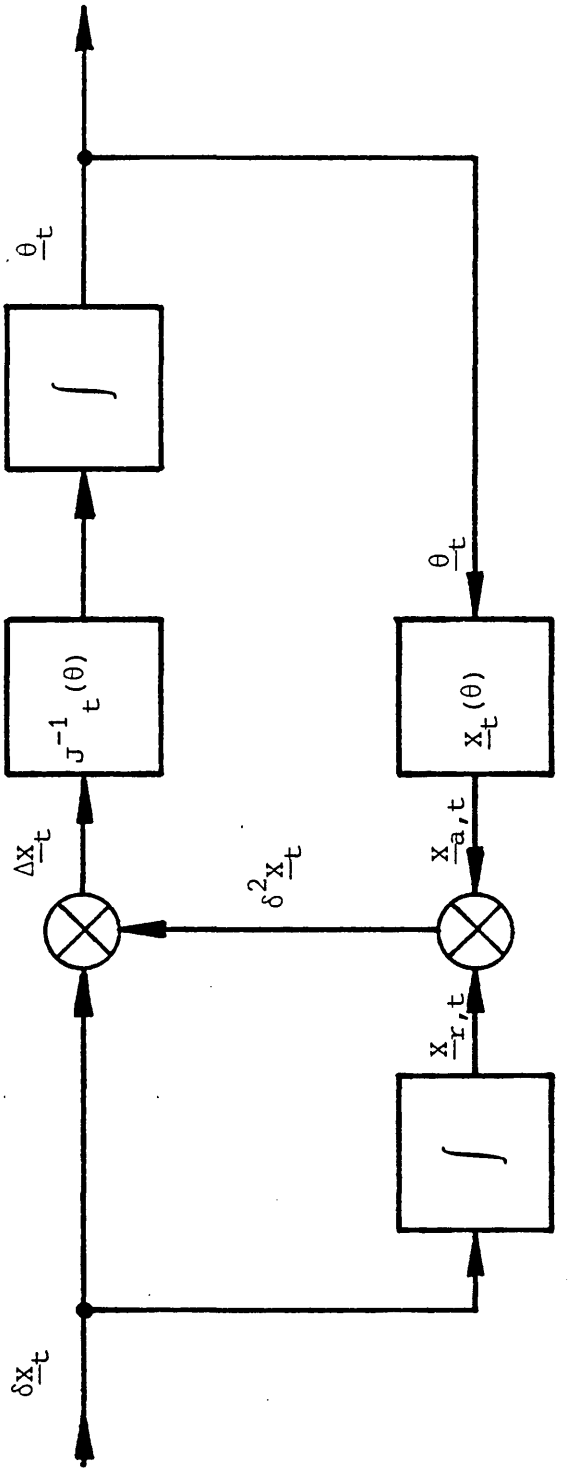
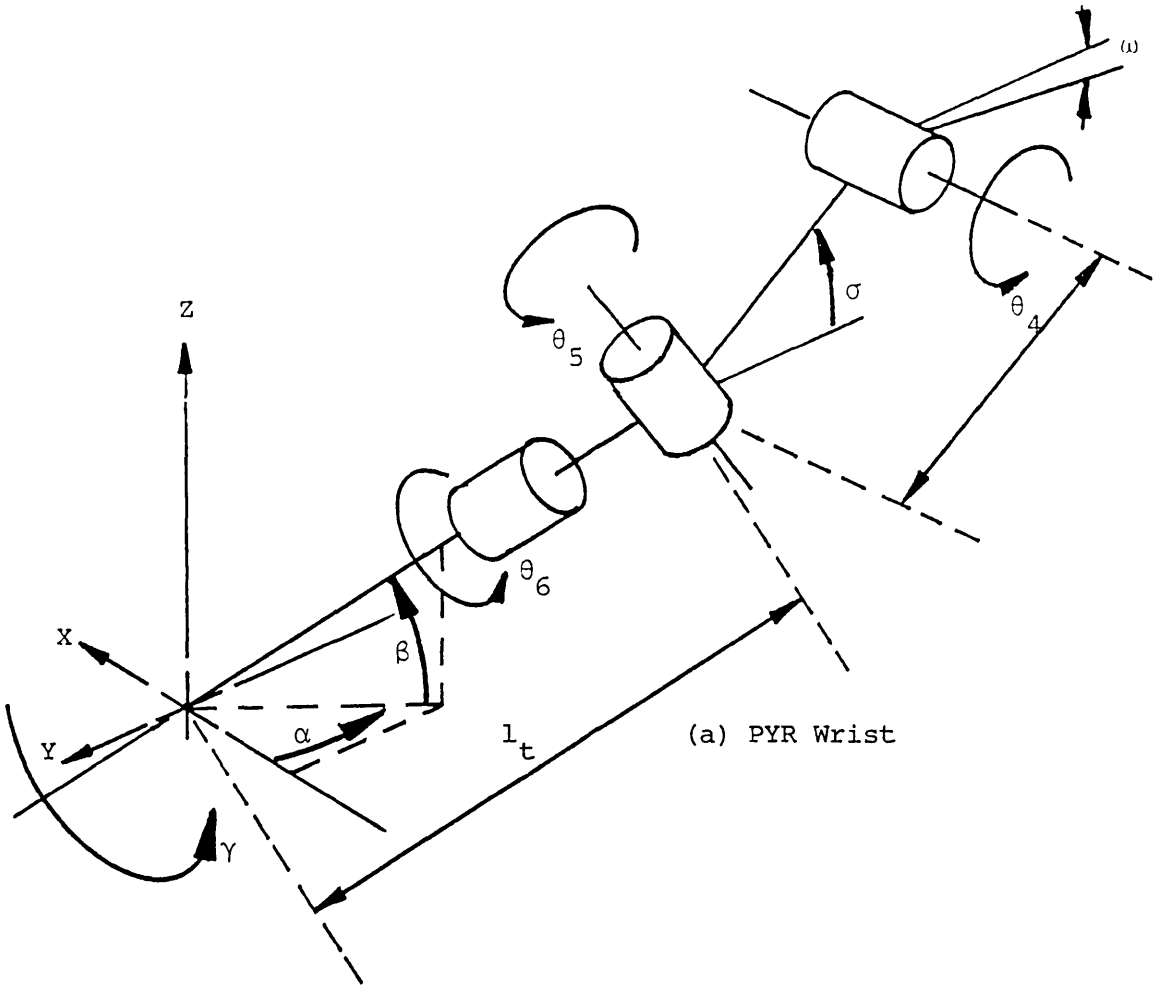


Figure 4.3 Structure of Pseudo Resolved Motion Rate Robot Controller



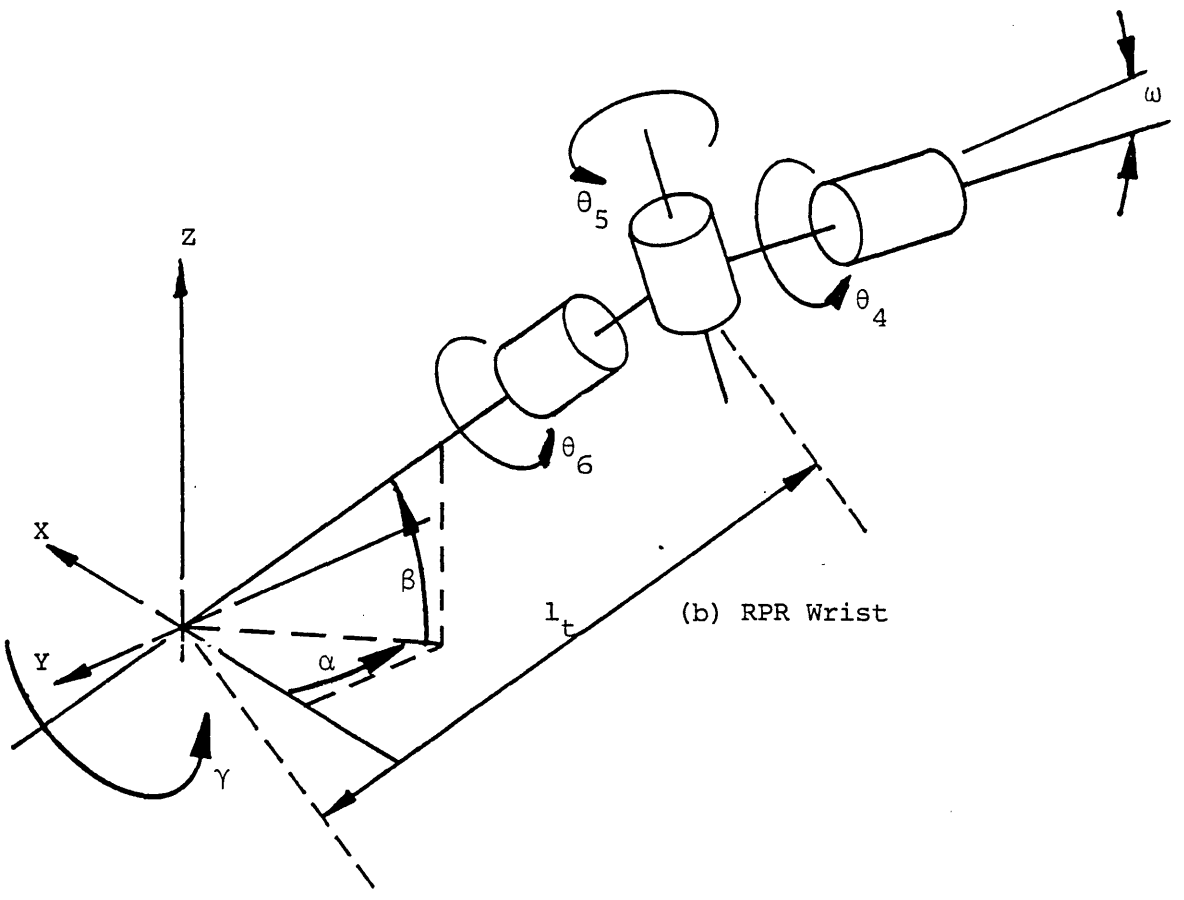
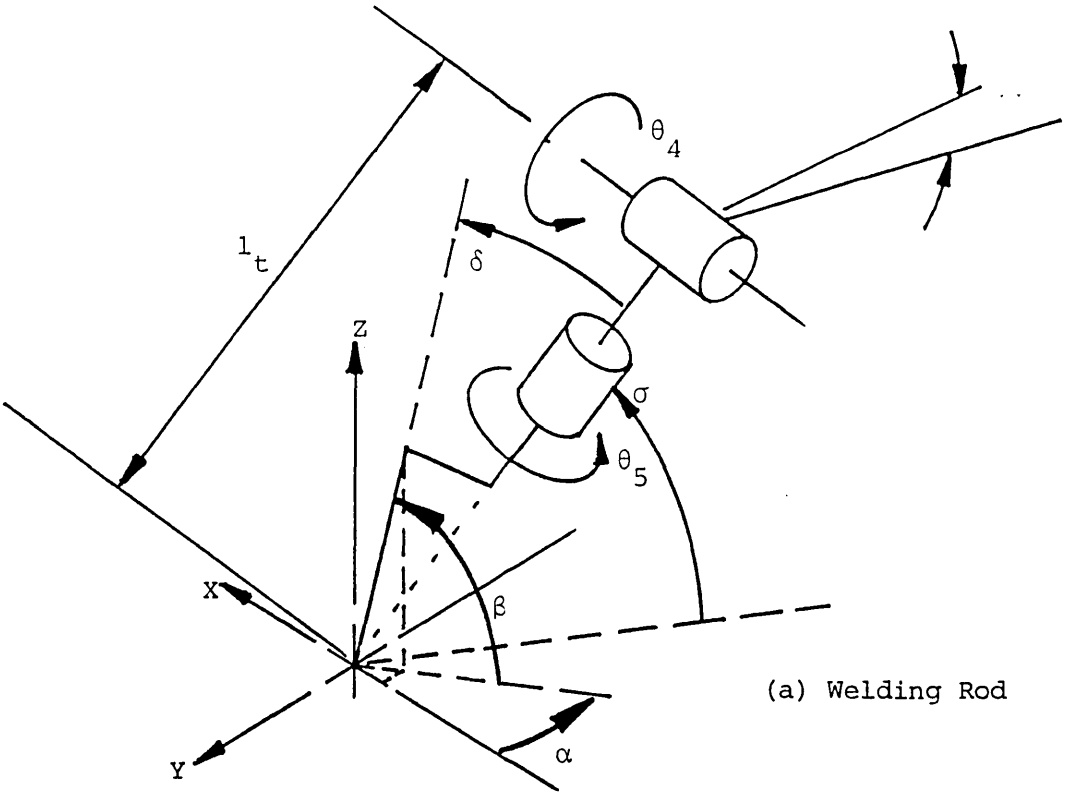
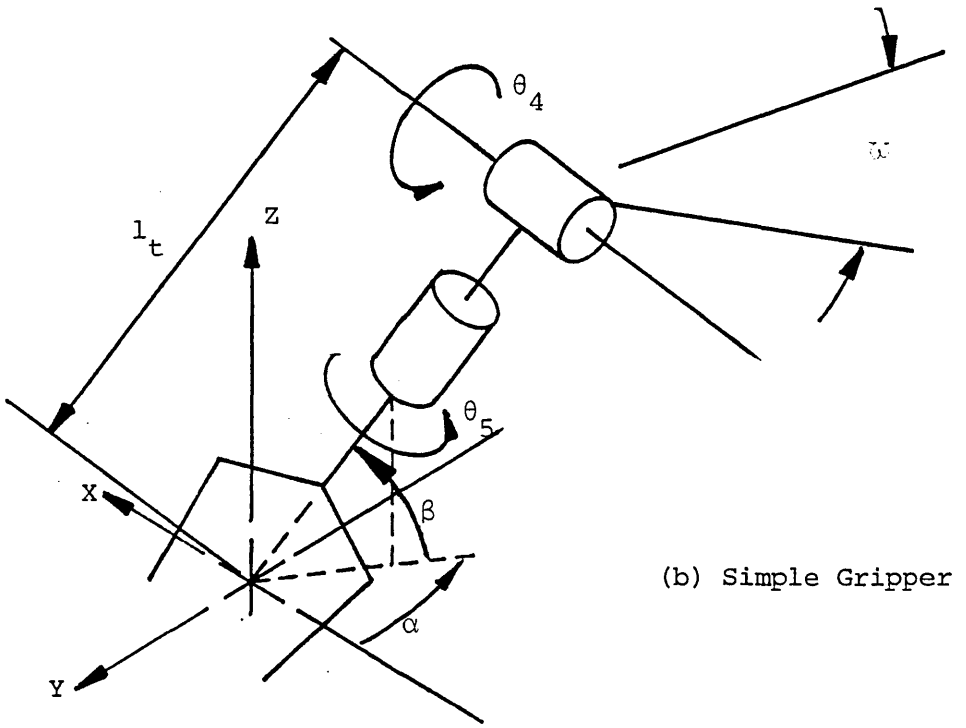


Figure 4.4 Three dof Wrist Parameters

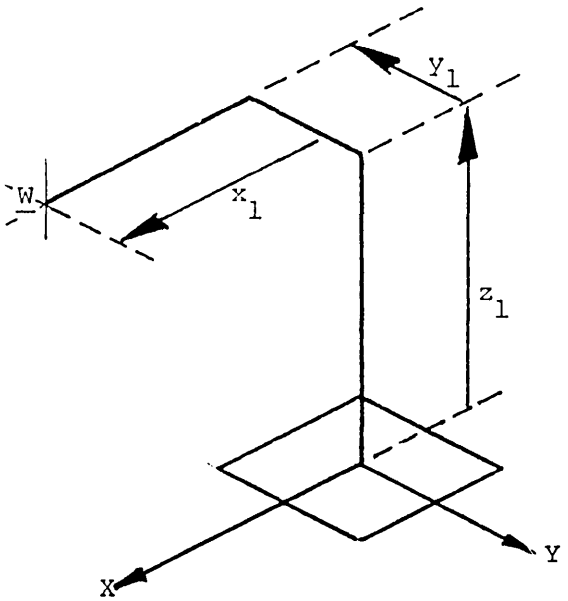


(a) Welding Rod

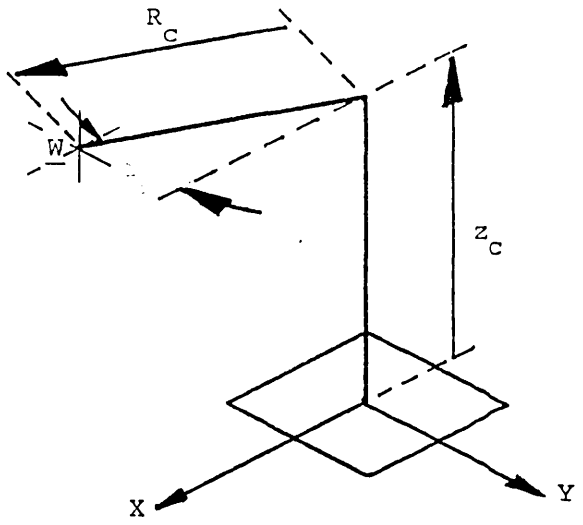


(b) Simple Gripper

Figure 4.5 Two dof Wrist Parameters



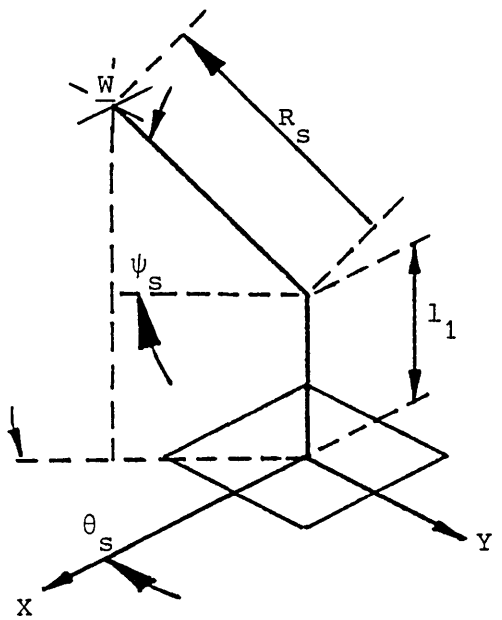
(a) Cartesian



(b) Cylindrical Polar

(c) Spherical Polar

(d) Anthropomorphic



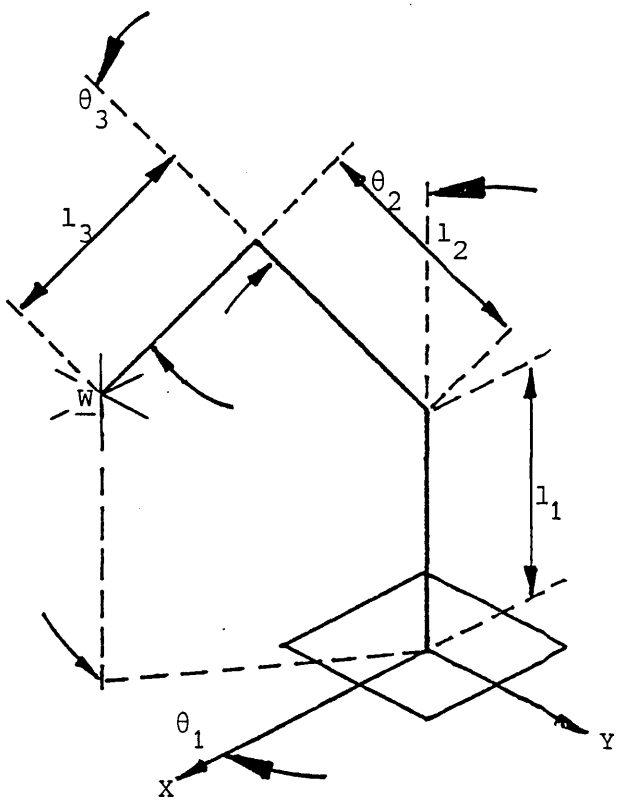


Figure 4.6 Primary Axes Parameters

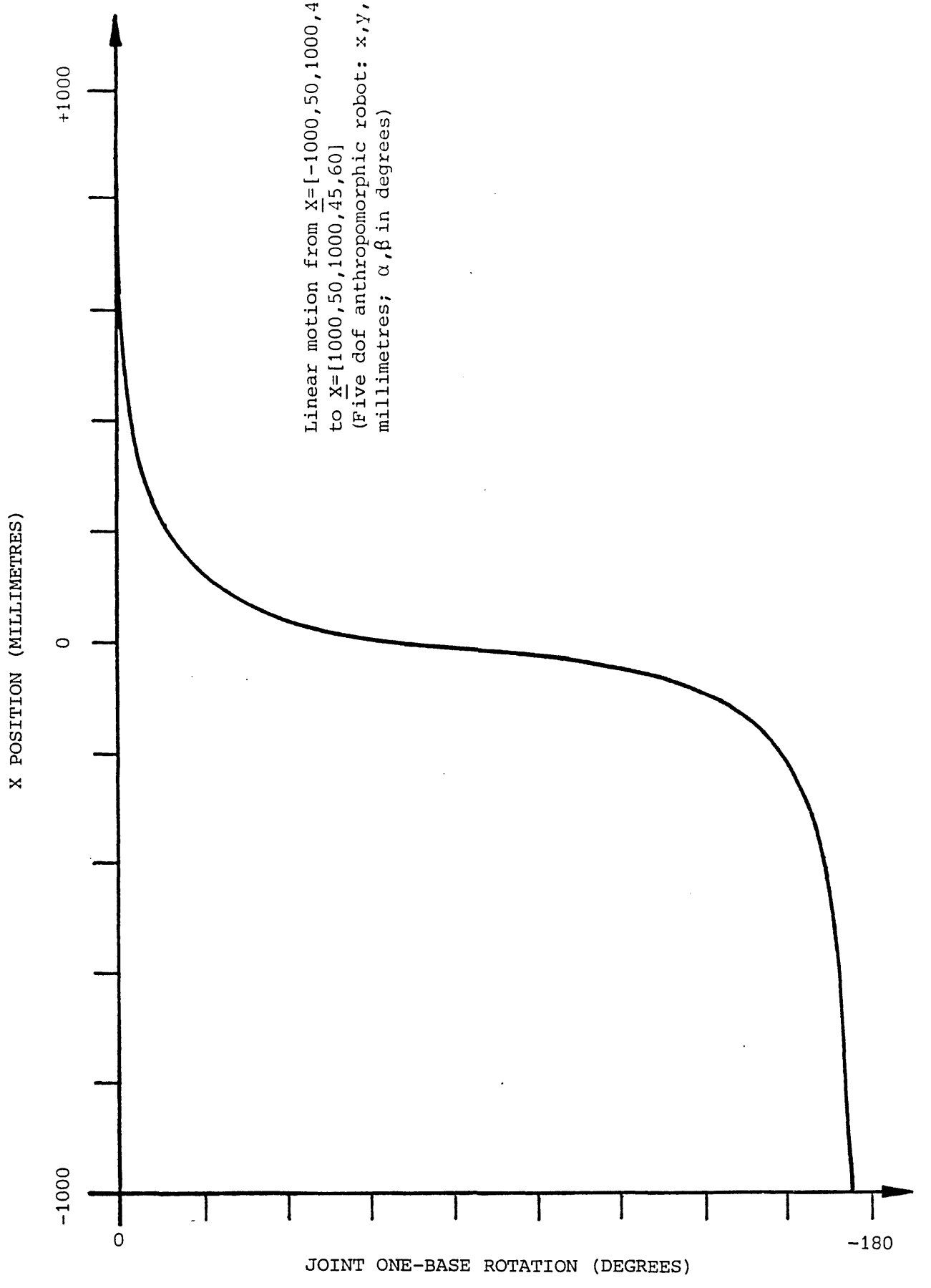
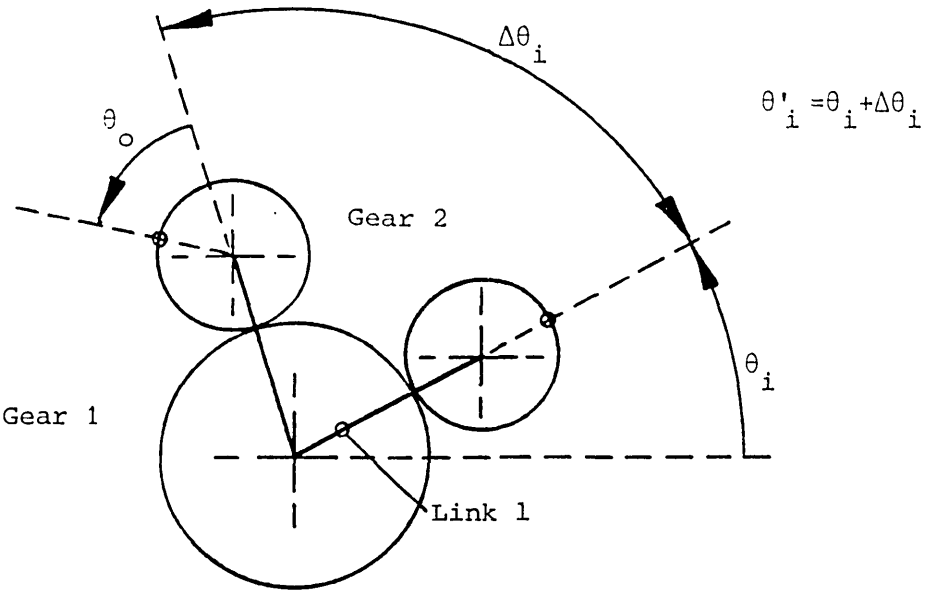


Figure 4.7 Joint Demand Requirements for a Motion Near a Singular Point



(a) Gear Transmission

(b) Chain Transmission

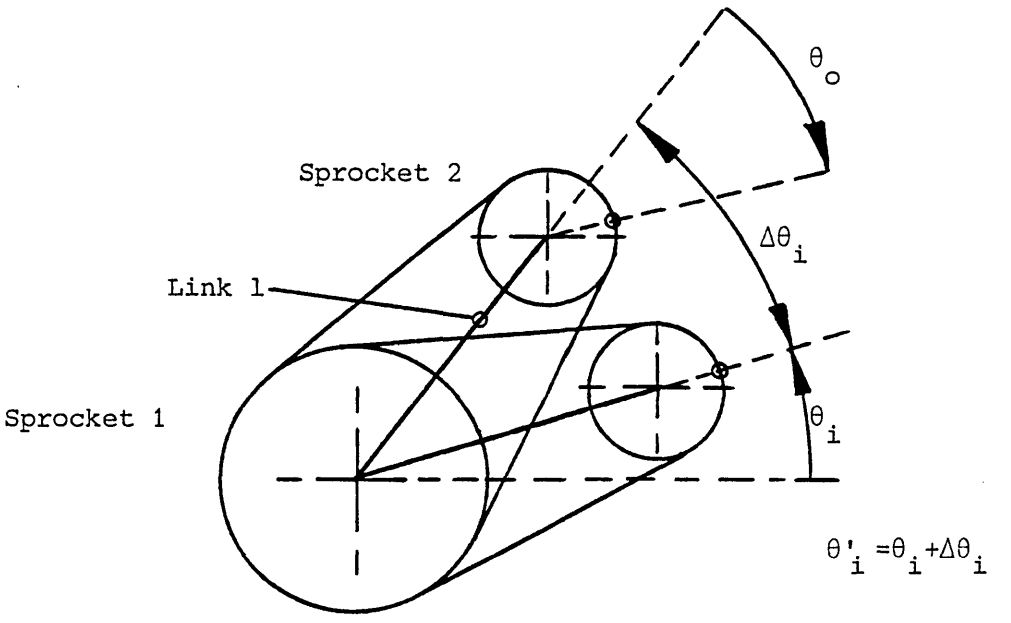
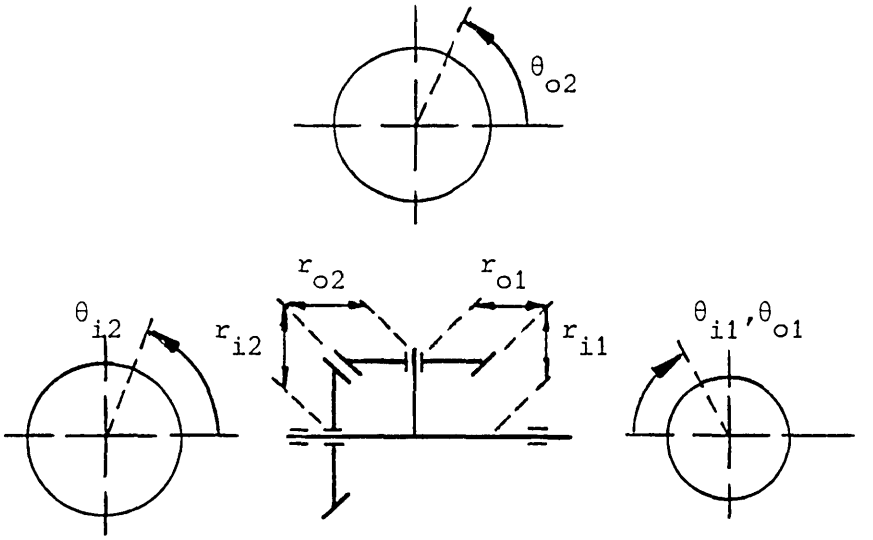


Figure 4.8 Planar Derivative Motion



(a) Two Bevel Gear Arrangement

(b) Three Bevel Gear Arrangement

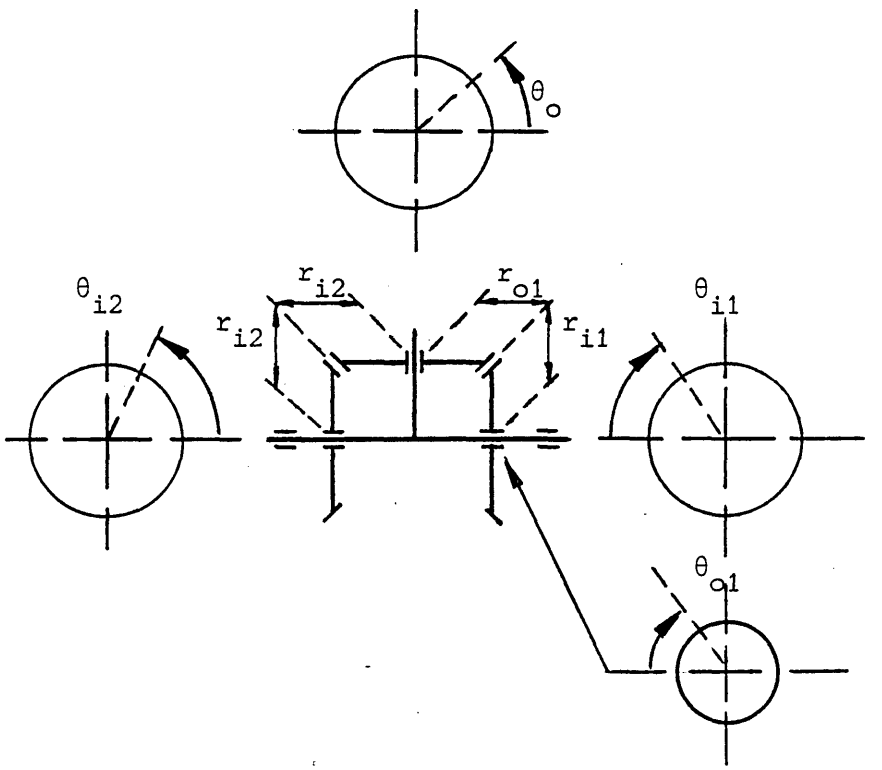


Figure 4.9 Differential Gear Mechanisms

Link	Variable	α	l	d
1	θ_1	90^o	0	0
2	θ_2	0^o	l_2	0
3	θ_3	0^o	l_3	0
4	θ_4	-90^o	l_4	0
5	θ_5	90^o	0	0
6	θ_6	0^o	0	0

Table 4.1 Link Parameters for the Six dof Robot

ARITHMETIC FUNCTION	INVERSE KINEMATIC TECHNIQUE		
	PRMR	MATRIX	GEOMETRIC
Arctangent	9	6	6
Sine/Cosine	14	27 (14)*	19 (8)*
Multiply/Divide	126	38	27
Add/Subtract	78	21	18
Square Root	-	1	1

*Refers to the minimum number of trigonometric functions that must be evaluated

Table 4.2 Arithmetic Requirements of Inverse Kinematics

5 TRAJECTORY GENERATION

5.1 TRAJECTORY REQUIREMENTS

The motion of a robot is determined by a series of paths in space, whether in workspace or joint space. Mathematically a path is a function of a single normalised parameter; s, the amount of path traversed. Considering workspace there are two associated paths, one pertaining to position, the other to orientation. In joint space there is only one path relating to the joints of the robot. A robot trajectory also includes velocity and acceleration information relating to the path.

The task of trajectory generation, irrespective of the trajectory co-ordinate space, can be subdivided into two distinct modules:

(i) Path or Vector Profiling. This module of trajectory specification produces reference values of the path parameter; s. The reference values are determined with respect to velocity and acceleration constraints.

(ii) The transformation of the reference values; s, into specific interpolation points along the path. The interpolation points can be in any of the robot space systems introduced in Chapter 3, (frame, base, joint). If a joint space system is used, the interpolation points can be given as demand input directly to the servo control algorithms. If any other space system is used, then the interpolation points are input parameters for subsequent transformation algorithms.

Expanding point (ii) above, if one considers work space trajectory generation, then within the control hierarchy of the robot controller the tasks of trajectory generation and joint servo control are seperated by frame transformation, co-ordinate transformation, and compensation for derivative motion. The two tasks are however closely coupled in two ways:

(i) If the robot is physically orthogonal, (as for a simple cartesian robot), the task of trajectory generation, and interpolation, reflects the actual positional requirements of the joints themselves. This is also true if a non-orthogonal robot is programmed in a point-to-point mode of path generation;

(ii) For the majority of tasks, (excepting trajectories through singular points), the time dependent shapes of trajectories in work space see similar path shapes at the joint trajectory level, and hence servo control level. Figure 5.1 illustrates this point by showing the continuous nature of the robot joint demands for a straight line, continuous motion, in base space. Therefore, the control of accelerations and velocities of trajectories in work space will also control the accelerations and velocities in joint space.

In the machine tool industry, the vector profiling, and interpolation techniques currently employed are usually the one of the following:

(i) Reference-Pulse interpolation, [Koren 1981],

(ii) Reference-Word interpolation (Alternatively known as Sampled-data interpolation), [Masory 1982].

The basic difference between these two methods of interpolation reflects the way in which the reference values for each interpolation axis are generated. In Reference-Pulse interpolation, a sequence of reference pulses is generated for each axis of motion, each pulse representing a motion of one basic length-unit, (BLU), of co-ordinate travel. With the Sampled-Data technique the reference value is generated as a binary word.

All Reference-Pulse interpolations are based on an iterative technique controlled by an interrupt clock. At each interrupt, a single iteration of the interpolation routine is executed, which in turn can provide an output pulse that increases the co-ordinate demand reference by one BLU. Therefore the maximum attainable velocity or co-ordinate demand, is inversely proportional to the execution time of a single iteration, coupled with the additional control functions carried out by the processor.

With Reference-Word interpolators, there is no restriction on the maximum velocity demand of an axis, but the interpolator algorithm is more complex than that required of a Reference-Pulse interpolator.

The actual interpolation algorithms are based on a two dimensional grid of BLU mesh. Separate algorithms for linear, [Bresenham 1965], and conic sections, [Pitteway 1967], evaluate the combination of X and Y BLU

movements to best achieve the desired spatial path. One difficulty with this form of interpolation technique is the correction for velocity differences when both an X and Y motion takes place simultaneously [Toko 1979].

5.2 VECTOR PROFILING

The term vector profiling, as used in this thesis, refers to the relationship between the amount of path traversal and time. The path traversal specifies a distance along a path, irrespective of the actual spatial form of the path. As stated previously, the vector profile can be defined with respect to any of the space systems relevant to robotics. The definition of vector profiling, which bests allows comparison, is the vector velocity/time relationship.

To evaluate the physical response of the robot for various velocity profiles, it is best to compare the profiles generated in joint space. Mujtaba, [Mujtaba 1977], has compared the response of various velocity profiles in joint space, including linear, cosine, polynomial, and critically damped profiles, (see Figure 5.2). The research showed that the linear velocity profile executes a given vector at least ten percent faster than the more complex profiles. (A critically damped profile is three times slower).

If the robot is to execute a continuous path motion, vector profiling is best carried out before frame or co-ordinate transformation. This ensures matching of servo commands in joint space.

To execute vector profiling for a continuous path in joint space, requires some degree of pre-processing of the work space path, to determine the associated joint space paths of the constituent axes. These paths would then require synchronisation to ensure the correct work space trajectory. It is simpler to profile the vector in work space prior to co-ordinate transformation. This is the technique employed in this thesis, in conjunction with linear profiling of the velocity demand.

The major disadvantage of profiling in work space, is the discontinuous nature of the co-ordinate transformation near singular points. This disadvantage can however, be overcome by monitoring the joint servo commands. This is discussed in more detail in Chapter 7.

Another consideration in trajectory generation, is the number of interpolation points required along the path to achieve a desired accuracy of planned motion. This is reflected in the frequency of update of the path traversal. Taylor, [Taylor 1979], suggests the use of bounded deviation paths. This method generates a number of "knot" points, which relect the successive positions along a path where co-ordinate transformation is required. By executing joint interpolation between the specified knots, a path is evaluated within pre-defined limits of accuracy. However, as with vector profiling in joint space, an ammount of pre-processing is required. The bounded deviation path algorithm evaluates the difference between the desired path, and joint space interpolated position, at the midpoint between knots. If the difference exceeds the desired accuracy, an additional knot point is placed at the midpoint. The algorithm is then applied

recursively. Eventually the path is defined by a sufficient number of specified knots to execute joint interpolation between the knots, within the accuracy requirements.

As with joint space vector profiling, the computational overheads associated with the pre-processing requirements of the algorithm, makes its use as an on-line system exceedingly time consuming. Off-line processing could be used, however the required robot path must be known in advance. One of the aims of this project was to construct a trajectory generation system that is adaptive, in the sense that changes in the required path can be accommodated on-line. Therefore, the system devised incorporates trajectory generation which provides a reference word update at each transformation loop time. The reference-word; (1), specifies the distance to be traversed during the subsequent interpolation and transformation algorithm execution time; (t_T).

One of the major tasks of the vector profiling algorithm, is to determine the deceleration profile. This is to provide a decreasing series of velocity demands to enable a smooth approach to the destination position. This lessens the impulsive effect on the mechanical system, reducing oscillation about the final position.

Refering to the profile illustrated in Figure 5.3, a period of of acceleration; (t_{acc}), is followed by a period of contant velocity; (t_{cv}). This in turn is followed by a period of deceleration; (t_{dcc}).

The point at which to commence deceleration can be determined either with reference to time, or better with reference to distances. As

discrete functions, the following lengths are defined:

$$L_{acc} = \sum_{t=o}^{t=t_{acc}} l_t \tag{5.1}$$

$$L_{dcc} = \sum_{t=o}^{t=t_{dcc}} l_t \tag{5.2}$$

$$L_t = \sum_{t=o}^{t=t} l_t \tag{5.3}$$

$$L_T = \sum_{t=o}^{t=T} l_t \tag{5.4}$$

Where L_{acc}, L_{dcc} are the lengths required to accelerate and decelerate, L_t is the path traversal at time t; L_T is the total distance covered when executing a specific trajectory; and l_t is the reference-length at time t. The terms L_{acc} and L_{dcc} can be evaluated beforehand for a known velocity.

The point at which to commence deceleration is when:

$$L_{dcc} \geq L_T - L_t \tag{5.5}$$

For symmetrical velocity profiles, the term L_{dcc} can be replaced by L_{acc}. This can then be determined on line.

For any spatially symmetrical profile (PTP, linear, circular) the total vector length can easily be determined. Given a parametric form of a vector r, such that:

$$r = f(t^*) \tag{5.6}$$

where $f(t^*)$ is some function of the parametric variable t^*. Then the spatial length of the path (L_T) is given by, [Kreysig 1979]:

$$L_T = \int_{t^*=start}^{t^*=finish} (r.r)^{\frac{1}{2}}\, dt \tag{5.7}$$

where r.r is defined by:

$$r.r \equiv \left[\frac{\partial x}{\partial t^*}\right]^2 + \left[\frac{\partial y}{\partial t^*}\right]^2 + \left[\frac{\partial z}{\partial t^*}\right]^2$$

However, the form of the integral is too complex to evaluate L_T for ellipses and cubic splines in real time. There is also the fact that the actual, and executed reference-word lengths are not necessarily equal. Therefore the technique developed, by the author, for the current project, involves the evaluation of the linear distance, L, between the present and required finishing point:

$$\begin{aligned} L &= | \underline{x}_t - \underline{x}_T | \\ &= [(x_t - x_T)^2 + (y_t - y_T)^2 + (z_t - z_T)^2]^{\frac{1}{2}} \end{aligned} \tag{5.9}$$

where $\underline{x}_t$, $\underline{x}_T$ are the current and final positions. This definition of L applies providing the total path length allows the required orientation changes to be achieved. If they cannot be accomplished in the time defined by L_T then:

$$L = | \underline{\omega}_t - \underline{\omega}_T |_{max} \tag{5.10}$$

where the subscripts again refer to the current and final values of the maximum orientation parameter change. The compatability of $\underline{x}$ and $\underline{\omega}$ relies on the fact that they are defined in similarly scaled units.

Thus the point at which deceleration should commence is given when:

$$L_{dcc} \leq L \tag{5.11}$$

For a specific deceleration profile there is a fixed relationship between l and L. For linear deceleration

$$l = k L \tag{5.12}$$

where the constant k:

$$0 < k < 1 \tag{5.13}$$

Thus as the robot approaches the finishing position, the velocity will approach zero.

This technique will also correct for the fact that the time at which deceleration should commence will often occur between successive loop times. The actual deceleration profile using this technique is illustrated by the broken line in Figure 5.3.

5.3 INTERPOLATION

The robot path can be defined with respect to a number of levels of co-ordinate space. The level of the space gives an indication of the amount of transformation that is necessary in order to provide the implicit robot joint demands. The robot joint space, defined as level 0, requires no transformation. Level 1 is base space, requiring co-ordinate transformation. Level 2, frame space, requires additional frame transformation.

In levels one, and above, the method of defining the space parameters can also vary. Paul, [Paul 1979], suggests the direct interpolation of the homogeneous matrices describing the state of the end effector, such that:

$$[H]_t = f(1)\ \Delta[H] + [H]_o \qquad (5.14)$$

where $[H]_t$ is the configuration at time t; $[H]_o$ the configuration at t=o; $\Delta[H]$ is the total change in configuration along the trajectory; and f(1) is a function of the trajectory parameter 1.

Taylor, [Taylor 1979], suggests a similar mechanism based on quarternion representation of the end effector.

The technique devised by the author for the present project, involves the interpolation of the position and orientation vectors directly ($\underline{x}$ and $\underline{\omega}$ respectively), whereby:

$$\underline{x}_t = x(1)\ 1 + \underline{x}_{t-1} \tag{5.15}$$

$$\underline{\omega}_t = \omega(1)\ 1 + \underline{\omega}_{t-1} \tag{5.16}$$

where $\underline{x}_t$, $\underline{\omega}_t$ are the vectors at time t; $\underline{x}_{t-1}$ and $\underline{\omega}_{t-1}$ are the vectors at time $t=t-1$; $x(1)$ and $\omega(1)$ are functions of 1.

Thus the two components of the total end effector description vector, $\underline{X}$, are considered individually. This coupled with the PYR mode of orientation description enables the execution of a large range of complex paths.

The basic modes of interpolation that have been devised for this project are as follows:

(i) Linear;

(ii) Circular;

(iii) Elliptical;

(iv) Cubic Spline;

(v) Spatial Function.

It is the combination of these interpolation modes, discrete position and orientation vectors, and various levels of co-ordinate space that allow a versatility in trajectory generation. The interaction of the various modes will now be discussed further. Those interpolation modes relevant to position are discussed first. These are followed by those modes used for orientation interpolation.

5.3.1 Linear Positional Interpolation

The linear mode of interpolation, when used with respect to position, is used for both point-to-point (PTP) and continuous linear path motion. The interpolation function $x_L(1)$ is of the form:

$$x_L(1)_t = \sum_{t=o}^{t=t} l_t/L_T \qquad (5.17)$$

where $\sum_{t=o}^{t=t} l_t$ is the accumulated value of 1 from the vector start; (t=o), to the position at time t; and L_T is the total length of vector.

With PTP motion, the joint space vector at time t, $\underline{\theta}_t$ is defined by:

$$\underline{\theta}_t = x_L(1)_t \, \Delta\underline{\theta} + \underline{\theta}_o \qquad (5.18)$$

where $\Delta\underline{\theta}$ is the change in joint vector; and $\underline{\theta}_o$ is the joint vector at t=o. The value of L_T, used to define $x_L(1)_t$, is given by the largest of the components of $\Delta\underline{\theta}$.

For continuous path motion, the cartesian position vector at t=t, x_t is given by:

$$\underline{x}_t = x_L(1)_t \, \Delta\underline{x} + \underline{x}_o \tag{5.19}$$

where $\Delta\underline{x}$ is the change in position vector; with $\underline{x}=\underline{x}_o$ at t=o. The value of L_T in this case is given by:

$$L_T = (\Delta x^2+\Delta y^2+\Delta z^2)^{\frac{1}{2}} \tag{5.20}$$

5.3.2 Circular Positional Interpolation

The method of accomplishing conic sections in space is achieved by evaluating the plane of the conic section, deriving the required frame transformation, and generating interpolation vectors within the plane. If the robot is "taught by leading", the variables pertaining to the frame transformation matrix are obtained as follows:

Three points are taught in the robot, the start, finish and an intermediate point on the circle; designated $\underline{x}_s$, $\underline{x}_f$, and $\underline{x}_i$ respectively. The plane containing the conic section, P is defined as:

$$P = [a, b, c, d] \tag{5.21}$$

where, using the Hessian Normal form of plane definition, [Smith 1897]:

$$a = \begin{vmatrix} y_s & z_s & 1 \\ y_f & z_f & 1 \\ y_i & z_i & 1 \end{vmatrix} \qquad b = \begin{vmatrix} x_s & z_s & 1 \\ x_f & z_f & 1 \\ x_i & z_i & 1 \end{vmatrix}$$

$$c = \begin{vmatrix} x_s & y_s & 1 \\ x_f & y_f & 1 \\ x_i & y_i & 1 \end{vmatrix} \qquad d = \begin{vmatrix} x_s & y_s & z_s \\ x_f & y_f & z_f \\ x_i & y_i & z_i \end{vmatrix} \tag{5.22}$$

Defining m as $(a^2+b^2+c^2)^{\frac{1}{2}}$, a vector; n, is then obtained from the expression:

$$n = [a/m, b/m, c/m, 1]^T \tag{5.23}$$

which represents the outward pointing normal of plane P, at a distance of -d/m from the base frame origin. Two angles for frame rotation can thus be defined:

$$\beta' = \text{atan2}\,[c, (a^2+b^2)^{\frac{1}{2}}1] \tag{5.24}$$

$$\alpha' = \text{atan2}\,[b, a] \tag{5.25}$$

Defining the initial frame rotation matrix R_1 as:

$$R_1 = (Z, \alpha')\,(Y, \beta') \tag{5.26}$$

Then the transformation of plane P to the YZ plane is achieved by the inverse R_1^{-1}. By pre-multiplying $\underline{x}_s$, $\underline{x}_f$, and $\underline{x}_i$ by R_1^{-1} the y,z values of the vectors transformed to the YZ plane can be obtained, ($\underline{x}_s'$,$\underline{x}_f'$,$\underline{x}_i'$ respectively).

The centre of the circle, (y_c, z_c), can be obtained from the three transformed vectors by means of the intersection of chord normals.

The frame origin for the frame transformation ${}^b\underline{O}_f$ is the transformed centre of the circle in base frame co-ordinates. Thus:

$${}^b\underline{O}_f = [-d/m, y_c, z_c]^T \tag{5.27}$$

The parameters required for circular positional interpolation, in addition to the frame rotation matrix and frame origin, are the circle radius; r, and angle of rotation; δ where:

$$r = [(z_s-z_c)^2 + (y_s-y_c)^2]^{\frac{1}{2}} \tag{5.28}$$

$$\delta = \delta_1-\delta_2 \tag{5.29}$$

and: $$\delta_1 = \text{atan2}[z_f-z_c, y_f-y_c] \tag{5.30}$$

$$\delta_2 = \text{atan2}[z_s-z_c, y_s-y_c] \tag{5.31}$$

The relevant parameters are illustrated in Figure 5.4. These parameters if defined by "teach by leading" are pre-processed off-line. Alternatively the parameters can be determined mathematically from information contained within a CAD database. However defined, they can be modified on-line to achieve larger arcs of the defined circle, or circles of different radii, concentric with the defined circle.

Thus the parameters required for execution of the circle are ${}^{b}R_f$, ${}^{b}\underline{O}_f$, r, δ_2 and δ. The frame interpolation vector ${}^{f}\underline{x}$ is determined from planar circular interpolation, using the reference length l. The angle of arc at time t; θ_t, is determined from l by:

$$\theta_t = 2\arcsin [l/2r] + \theta_{t-1} \tag{5.32}$$

where θ at t=o is obtained from the starting point i.e. δ_2. ${}^{f}\underline{x}_t$ is thus given by:

$$ {}^{f}\underline{x}_t = [r\cos\theta_t,\ r\sin\theta_t,\ o]^T \qquad (5.33) $$

from which the base interpolation vector at t=t, ${}^{b}\underline{x}_t$ can be obtained thus:

$$ {}^{b}\underline{x}_t = {}^{b}R_f\ {}^{f}x_t + {}^{f}\underline{O}_b \qquad (5.34) $$

The base interpolation vector, (plus orientation vector), is then used to determine the required joint parameters at time t.

5.3.3 Elliptical Positional Interpolation

If the parameters required for the elliptical interpolation are not determined from stored data, then "teaching by leading" involves more parameters than required for circular interpolation. The equation for the ellipse could be obtained from teaching six points, rotating the plane, and solving simultaneously the general equation:

$$ a_e z^2\ b_e y^2 + 2c_e yz + 2d_e z - 2e_e y = f_e \qquad (5.35) $$

where a_e, b_e, c_e, d_e and f_e are constants.

It is however easier to solve using either a cubic spline for short elliptical segments, or restrict the operator to teaching specific points as follows:

either: The two extremities of the major axis, and one extremity of the minor axis;

or: the two extremities of the minor axis and one extremity of the major axis,

and: the start and finishing points of the elliptical path.

As with the circular interpolation, the parameter passed from the vector profiling unit is the reference length l. The plane of the ellipse is determined using the Hessian Normal form of plane definition. The third rotation brings the major axis parallel with the Y axis of the base frame. Thus the required ellipse is considered as shown in Figure 5.4, with the frame origin at the ellipse centre.

The method used to determine the required interpolation points, based on the reference word length is as follows:

Direct evaluation based on the distance between points on the ellipse requires the solution of a quartic equation. Time constraints make this technique unsuitable. However an approximation, devised by the author, based on the elliptical tangent, has been shown by computer modelling to give exact positions along the ellipse, with a velocity error of less than three percent. Given the parametric form of the ellipse

$$y = a\cos\theta_e \tag{5.36}$$

$$z = b\sin\theta_e \tag{5.37}$$

The angle of the tangent to the ellipse $\sigma_{e,t}$ at a point given by $\theta_{e,t}$ is given by:

$$\sigma_{e,t} = \text{atan2}\,[-b\cos\,\theta_{e,t},\ a\sin\,\theta_{e,t}] \tag{5.38}$$

The approximation for each subsequent point required on the ellipse is obtained as follows:

$$y_{t+1} = \pm l \cos \sigma_{e,t} \quad \text{if} \quad \sigma_{e,t} < 45^{o} \tag{5.39}$$

$$z_{t+1} = \pm l \sin \sigma_{e,t} \quad \text{if} \quad \sigma_{e,t} > 45^{o} \tag{5.40}$$

The sign is determined by considering whether the total change in position requires a positive or negative change in elliptical angle. Thus given either y or z, at time t+1, the corresponding value of z or y can be determined from the equation of the ellipse.

The change of term evaluation at 45^{o} means that the equations are never ill conditioned, and no ambiguity occurs at 90^{o} points.

5.3.4 Cubic Spline Positional Interpolation

The cubic spline interpolation algorithm devised by the author, generates a smooth continuous path through a sequence of given points. Although a plane can be generated using three points, discontinuities will occur if a subsequent point is not within the given plane. The technique devised uses the x y z position of points, with a cubic spline generated for each orthogonal plane. Given a frame origin at each start point of a spline generation, the required splines are given by:

$$x = a_x l_s^3 + b_x l_s^2 + c_x l_s \tag{5.41}$$

$$y = a_y l_s^3 + b_y l_s^2 + c_y l_s \tag{5.42}$$

$$z = a_z l_s^3 + b_z l_s^2 + c_z l_s \tag{5.43}$$

Where l_s is a component of the straight line distance between subsequent points and a, b, c are constants. Considering the x axis to illustrate the technique, the spline is an 'x' function of the straight line distance between subsequent points. Given that:

$$x = a_x l_s^3 + b_x l_s^2 + c_x l_s + d_x \tag{5.44}$$

The parameters a_x, b_x, c_x and d_x can be obtained from the following boundary conditions:

$$x = x_n \quad \text{at } l_s = o \tag{5.45}$$

(This term in fact sets d to zero as the origin is taken as the first point in the spline. This yields the three term cubic function as in Equations (5.41), (5.42), (5.43)).

$$x = x_{n+1} \text{ at } l_s = L_1 \tag{5.46}$$

$$\dot{x} = \dot{x}_n \quad \text{at } l_s = o \tag{5.47}$$

$$\dot{x} = \dot{x}_{n+1} \text{ at } l_s = L_1 \tag{5.48}$$

The subscripts refer to the points n, and n+1, and x refers to the differential of x with respect to l_s. x_n is either zero if starting, or passed from the previous spline. L_1 is the 'x' distance between points n and n+1. x_{n+1} is given by:

$$\dot{x}_{n+1} = \frac{x_{n+2} - x_n}{L_1 + L_2} \tag{5.49}$$

where L_2 is the 'x' distance between points n+1 and n+2.

This method of deriving the velocity ensures continuity in both velocity and acceleration when passing from one spline to another. The form of y and z can be obtained in a similar fashion.

The parameter l_s, the straight line distance is obtained from the reference-word length l, referring to the distance along the cubic spline, as follows:

The straight line distance between two consecutive points along a splined path, can be approximated by the tangent at a given point. The vector components of the tangent can be obtained from the constituent spline functions. The tangential vector; $\underline{n}$ is specified by:

$$\underline{n} = \left[\frac{\partial x}{\partial l_s}, \frac{\partial y}{\partial l_s}, \frac{\partial z}{\partial l_s}\right]^T \tag{5.50}$$

where the subscripts indicate a differential with respect to l_s. The vector components are defined as follows:

$$\frac{\partial x}{\partial l_s} = 3a_x l_s^2 + 2b_x l_s + c_x \tag{5.51}$$

$$\frac{\partial y}{\partial l_s} = 3a_y l_s^2 + 2b_y l_s + c_y \tag{5.52}$$

$$\frac{\partial z}{\partial l_s} = 3a_z l_s^2 + 2b_z l_s + c_z \tag{5.53}$$

The vector length, equivalent to the straight line distance, is given by:

$$|\underline{n}| = \left[\left(\frac{\partial x}{\partial l_s}\right)^2 + \left(\frac{\partial y}{\partial l_s}\right)^2 + \left(\frac{\partial z}{\partial l_s}\right)^2 \right]^{\frac{1}{2}} \qquad (5.54)$$

This scalar can then be used to approximate the next value of l_s; $l_{s,t+1}$ based on the reference-word length l_{t+1} as follows:

$$l_{s,t+1} = \frac{l_{t+1}}{|n|_t} \qquad (5.56)$$

where $|n|_t$ is based on $l_{s,t}$.

Computer modelling has shown that this technique ensures constant velocity along a spline within four percent. However a lower bound on $|n|$ must be given to prevent the collapse of Equation (5.56) at the beginning and end of a motion as $|n|$tends to zero.

To illustrate the technique Figure 5.5 shows the relationship between x, and y, holding z constant. The values of x and y have been obtained independently from the reference word 1. The lower bound on $|n|$ was taken to be 0.2.

5.3.5 Linear Orientation Interpolation

Linear orientation interpolation is used most frequently in conjunction with linear positional interpolation, and cubic spline interpolation. As for linear interpolation, the linear form of $\omega(l)$; $\omega_L(l)$ at time t is given by:

$$\omega_L(l)_t = \sum_{t=o}^{t=t} l_t/L \qquad (5.57)$$

and:

$$\underline{\omega}_t = \omega_L(1)\ \Delta\underline{\omega} + \underline{\omega}_o \tag{5.58}$$

where $\sum_{t=o}^{t=t} l_t$ is the summation of reference-word lengths l_t; $\underline{\omega}_t$ is the orientation vector at time t; $\Delta\underline{\omega}$ is the change in $\underline{\omega}$; and $\underline{\omega}_o$ is the orientation vector at the start of motion. L is for most cases defined as for linear interpolation, as:

$$L = (\Delta x^2 + \Delta y^2 + \Delta z^2)^{\frac{1}{2}} \tag{5.59}$$

However it is necessary to check that the required change in orientation can be accomplished in the time required for linear motion. (The extreme case is when $\underline{x} = o$). Given the maximum orientational velocity $\underline{\omega}_{max}$, then the positional form of L can be used if:

$$\frac{\Delta\underline{\omega}_{max}}{\Delta\underline{x}} < \frac{\dot{\underline{\omega}}_{max}}{\underline{x}} \tag{5.60}$$

Where $\underline{x}$ is the specified velocity for the path required; $\Delta\underline{\omega}_{max}$ is the largest component change of $\underline{\omega}$. Similarly $\underline{\omega}_{max}$ refers to the maximum possiblevelocity of a single component of $\underline{\omega}$. This assumes that the orientation motions are decoupled, which is a reasonable approximation for the instances that this will occur.

If condition 5.60 cannot be satisfied then the value of L is taken to be the largest component change of $\underline{\omega}$:

$$L = \Delta\underline{\omega}_{max} \tag{5.61}$$

This value of L must also be used in the positional interpolation algorithms that are executed in conjunction with the orientational interpolation.

5.3.6 Spatial Function Orientation Interpolation

The spatial function mode of interpolation is used to associate the orientation vector with the current positional vector. This mode of interpolation is especially important for such tasks as welding or adhesive application. The primary requirement is a plane of motion and a current vector defined within the plane.

Considering Figure 5.7, the orientation at time t; parameters $\alpha_{o,t}$, $\beta_{o,t}$ and $\gamma_{o,t}$, are specified with respect to the tangent to the current position. Thus the frame orientation parameters, specified with respect to the plane, are given by:

$$^{f}\alpha_t = \alpha_{o,t} + \sigma_t \tag{5.62}$$

$$^{f}\beta_t = \beta_{o,t} \tag{5.63}$$

$$^{f}\gamma_t = \gamma_{o,t} \tag{5.64}$$

Where σ_t is the angle of the tangent to the path at the current position. The orientation vector is then determined by use of the frame rotation matrix, as discussed in Section 3.4.

This mode of orientational interpolation can also be used with linear and cubic spline modes of positional interpolation, providing that the paths are specified within a single plane.

In a similar fashion to the spatial function interpolation, a combination of interpolation modes can be used to create such items as weaves along a path. This is particularly useful when welding. Given the plane of motion, and direction within the plane, then frame positional parameters; ${}^f x_t$ and ${}^f y_t$, can be defined by:

$$ {}^f x_t = x_{i,t}\, f_1(1)\cos\sigma_t \tag{5.65} $$

$$ {}^f y_t = y_{i,t} + f_1(1)\sin\sigma_t \tag{5.66} $$

where $f_1(1)$ can be a sinusoid, or waveform, to be superimposed on the primary motion; $x_{i,t}$, and $y_{i,t}$ are the positions determined from the primary motion at t; and σ_t is the tangential angle at time t.

A similar equation can be used to provide a motion perpendicular to the plane, such that

$$ {}^f z_t = f_2(1) \tag{5.67} $$

where $^{f}z_t$ is the z position with respect to the plane at time t, and $f_2(l)$ is a function such, but not necessarily the same, as $f_1(l)$.

It is felt that the combination of interpolation modes discussed above, relating independently to position and orientation, provide the means to accomplish a vast range of paths for a robot arm.

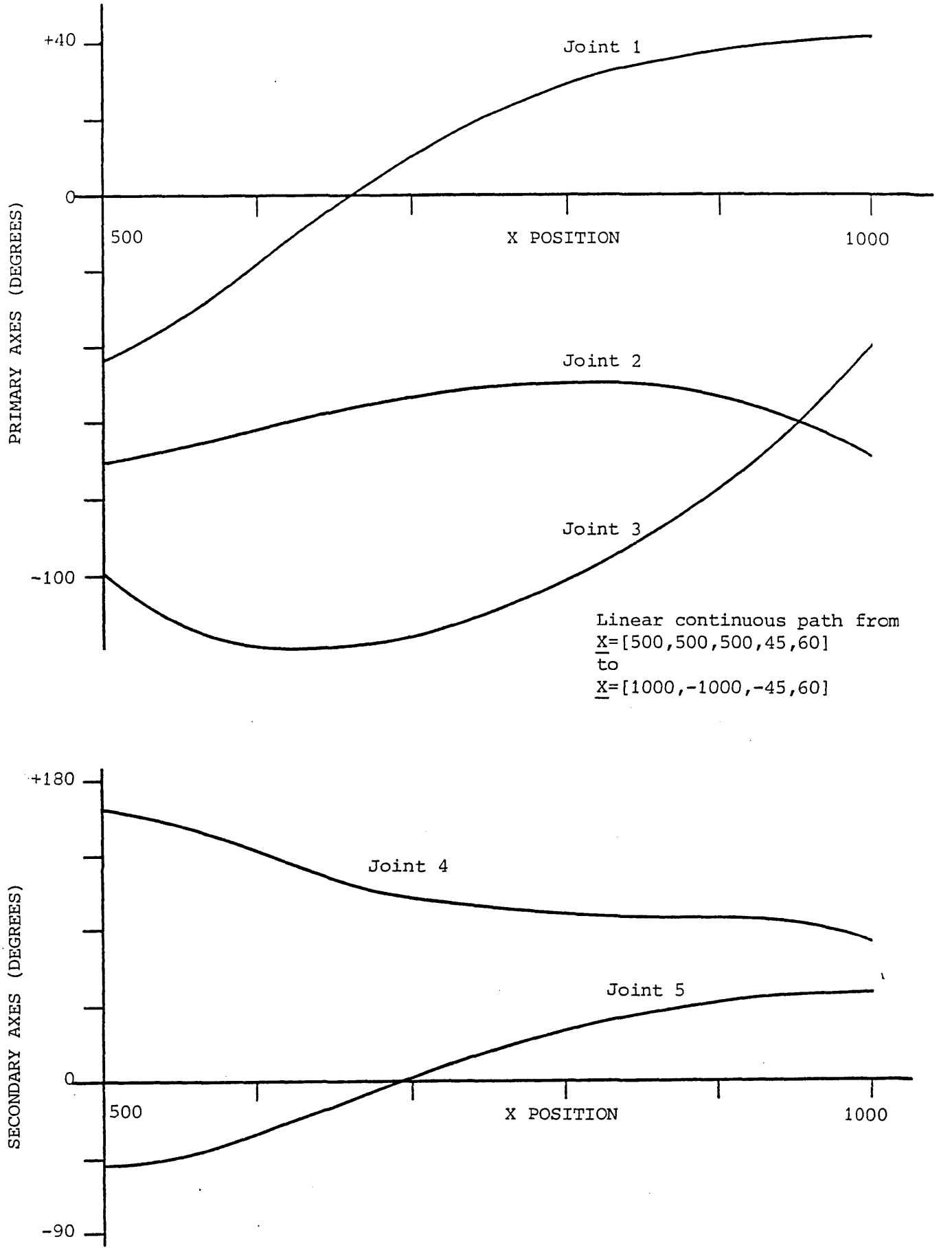


Figure 5.1 Variation of Robot Joint Parameters for Linear Continuous Path Motion

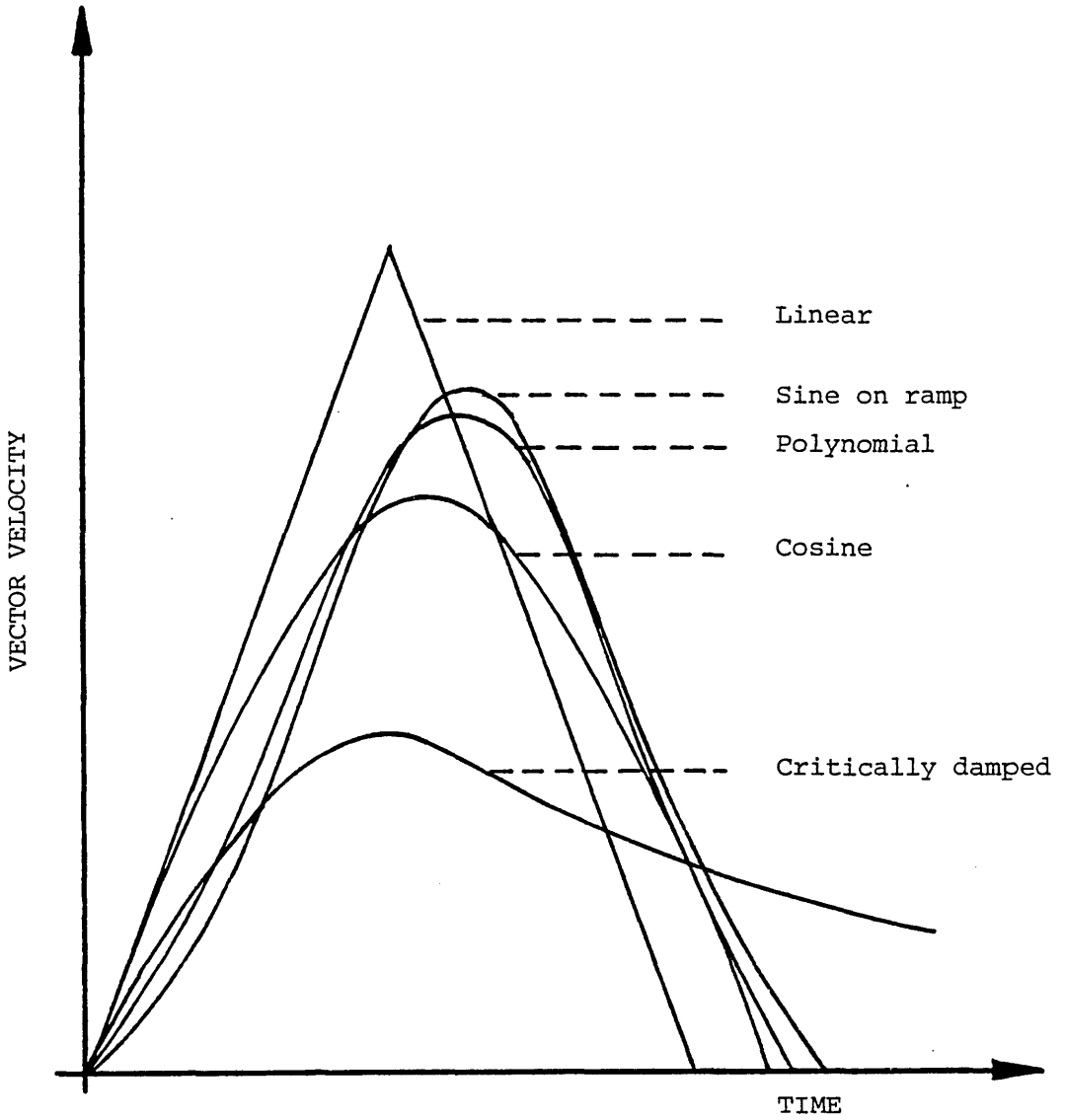


Figure 5.2 Vector Velocity Profiles

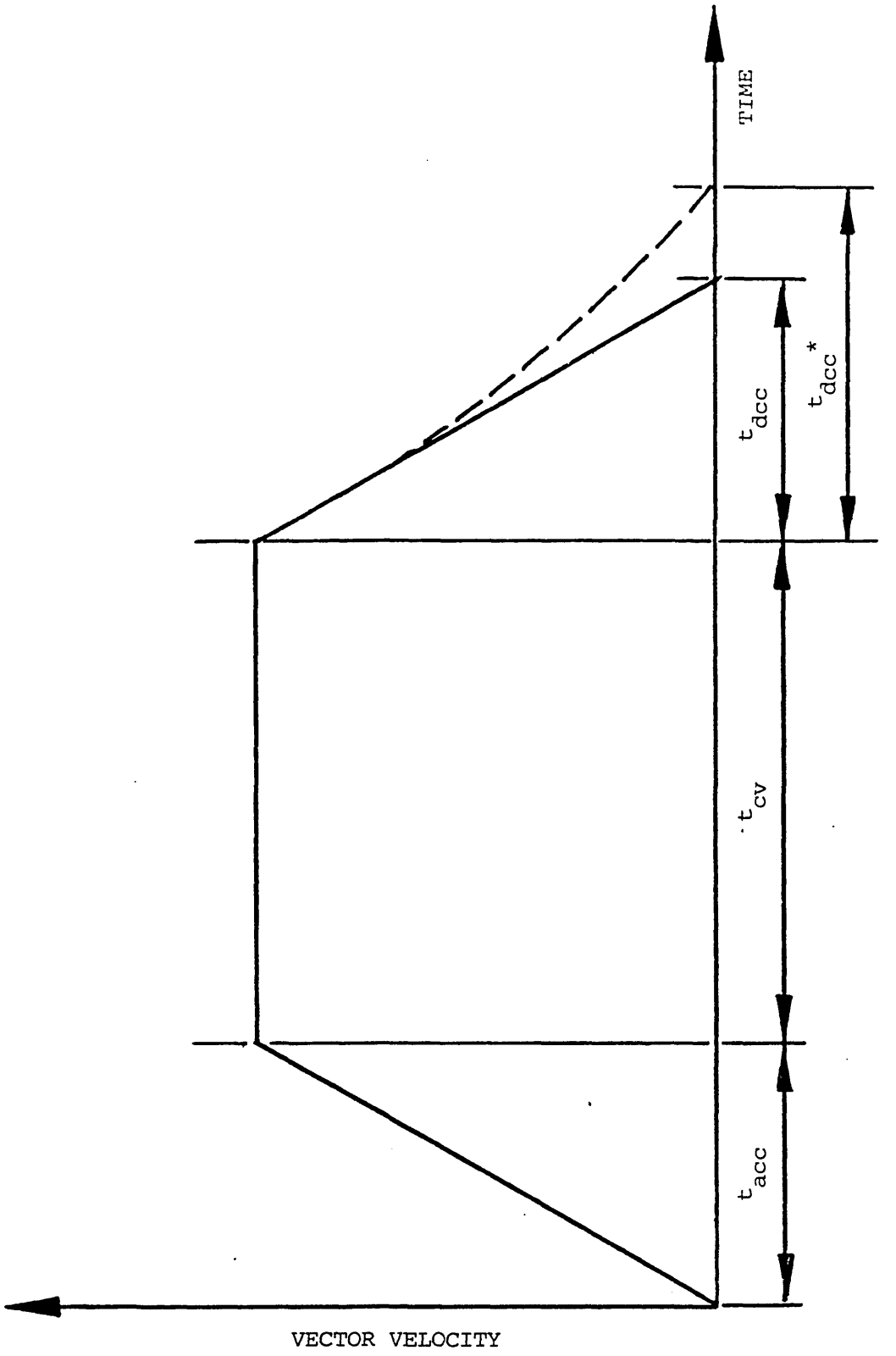


Figure 5.3 Vector Velocity/Time Characteristics

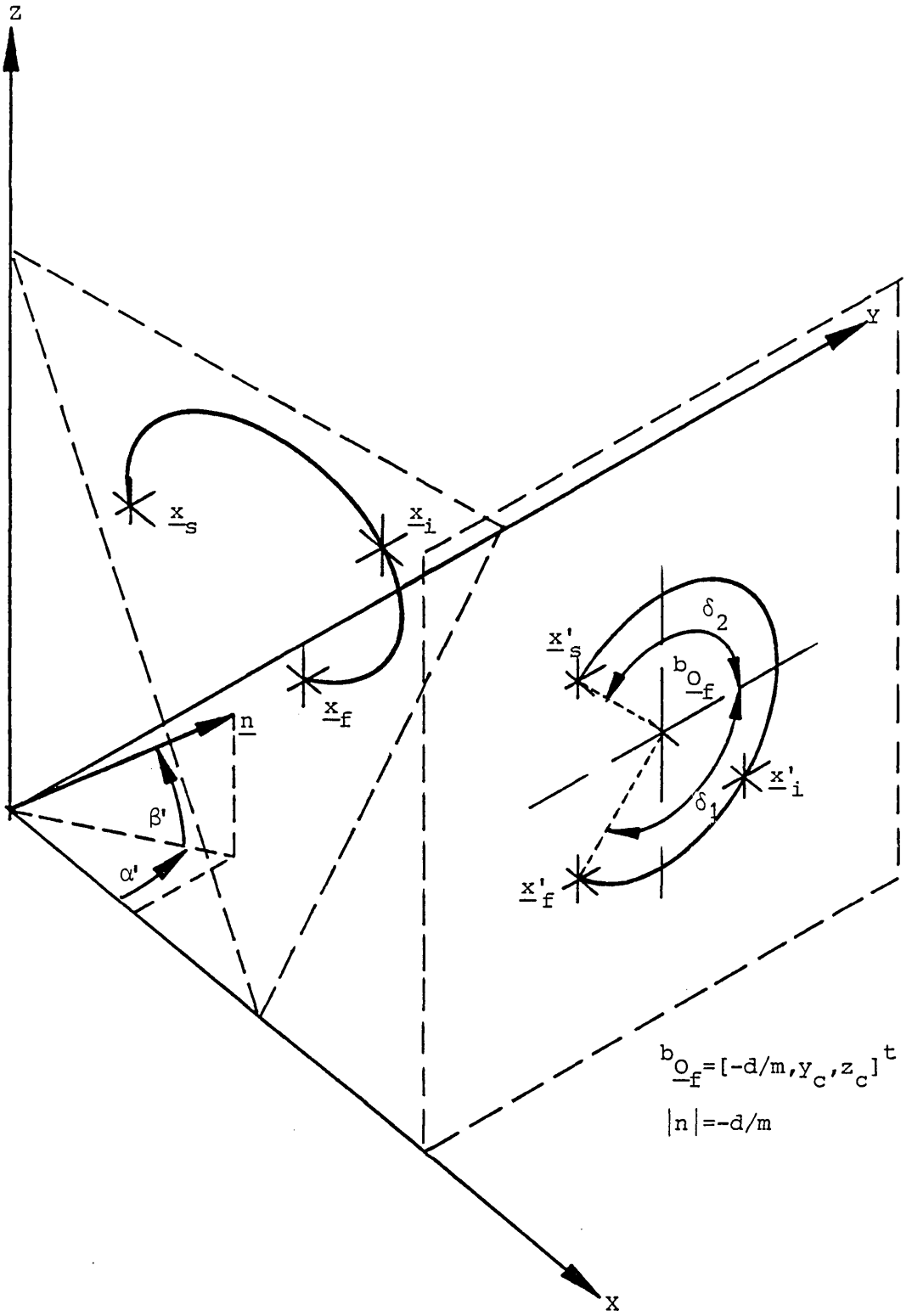


Figure 5.4 Circular Interpolation Parameters

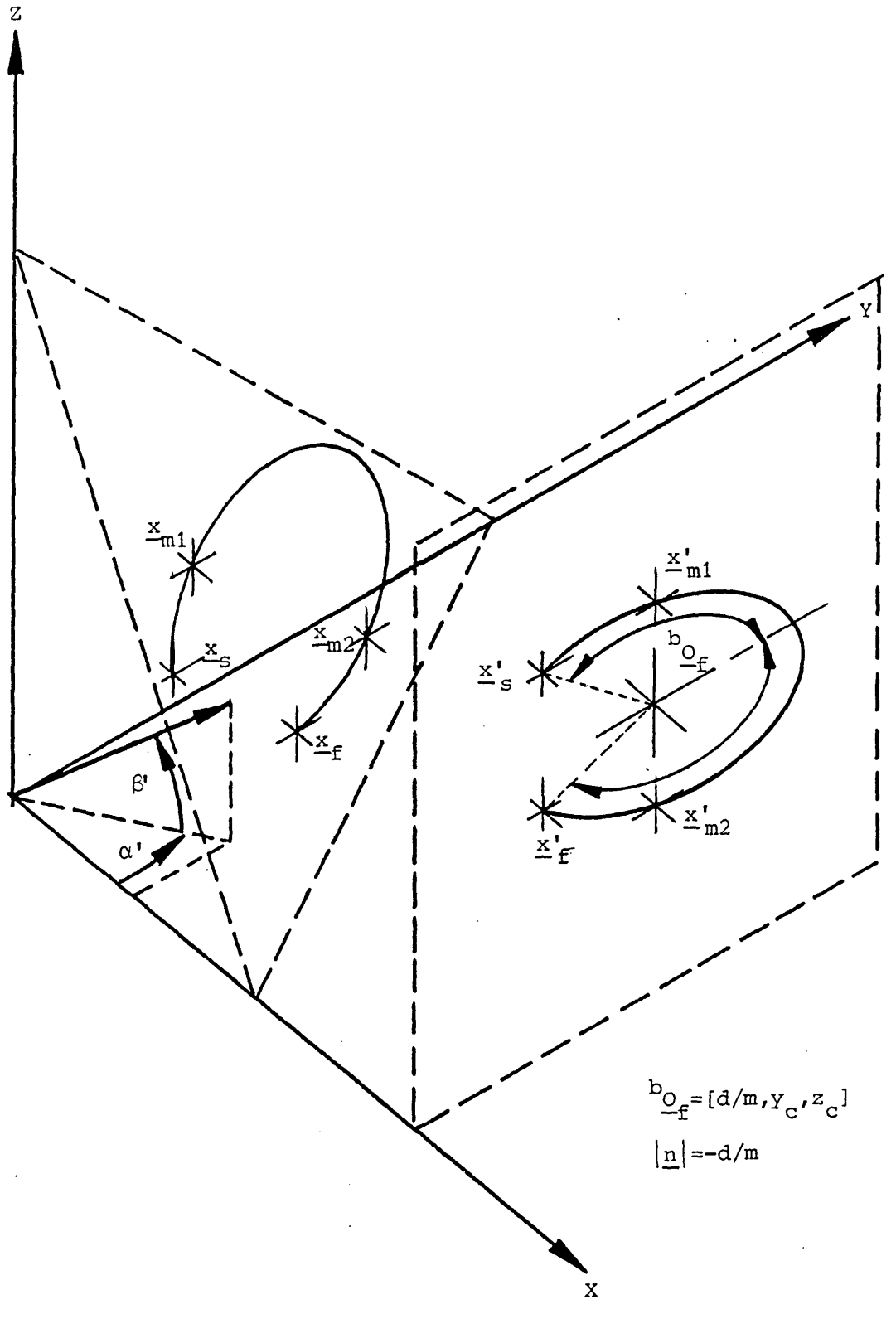


Figure 5.5 Elliptical Interpolation Parameters

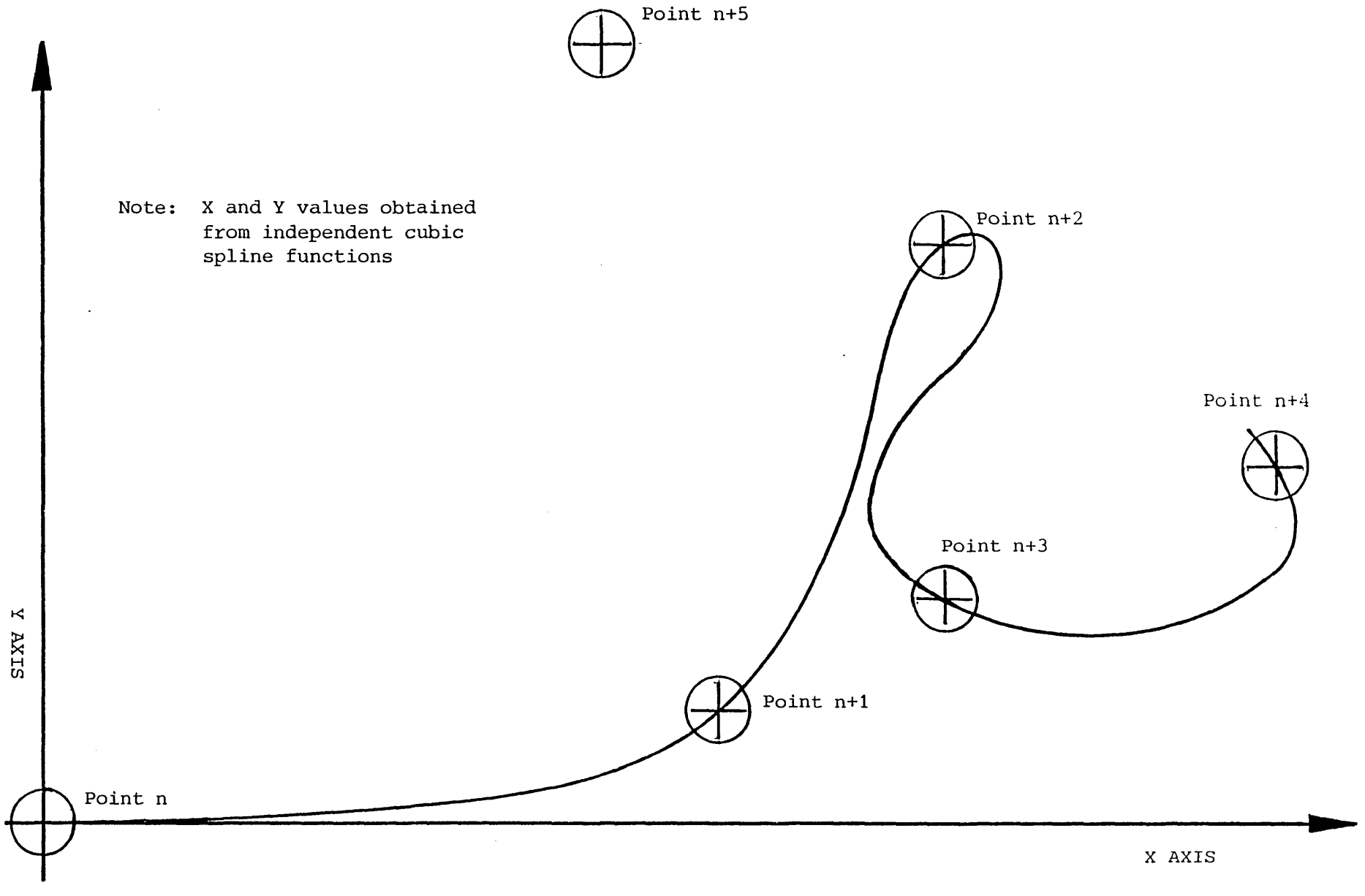


Figure 5.6 Combined Path Using Independent Cubic Splines

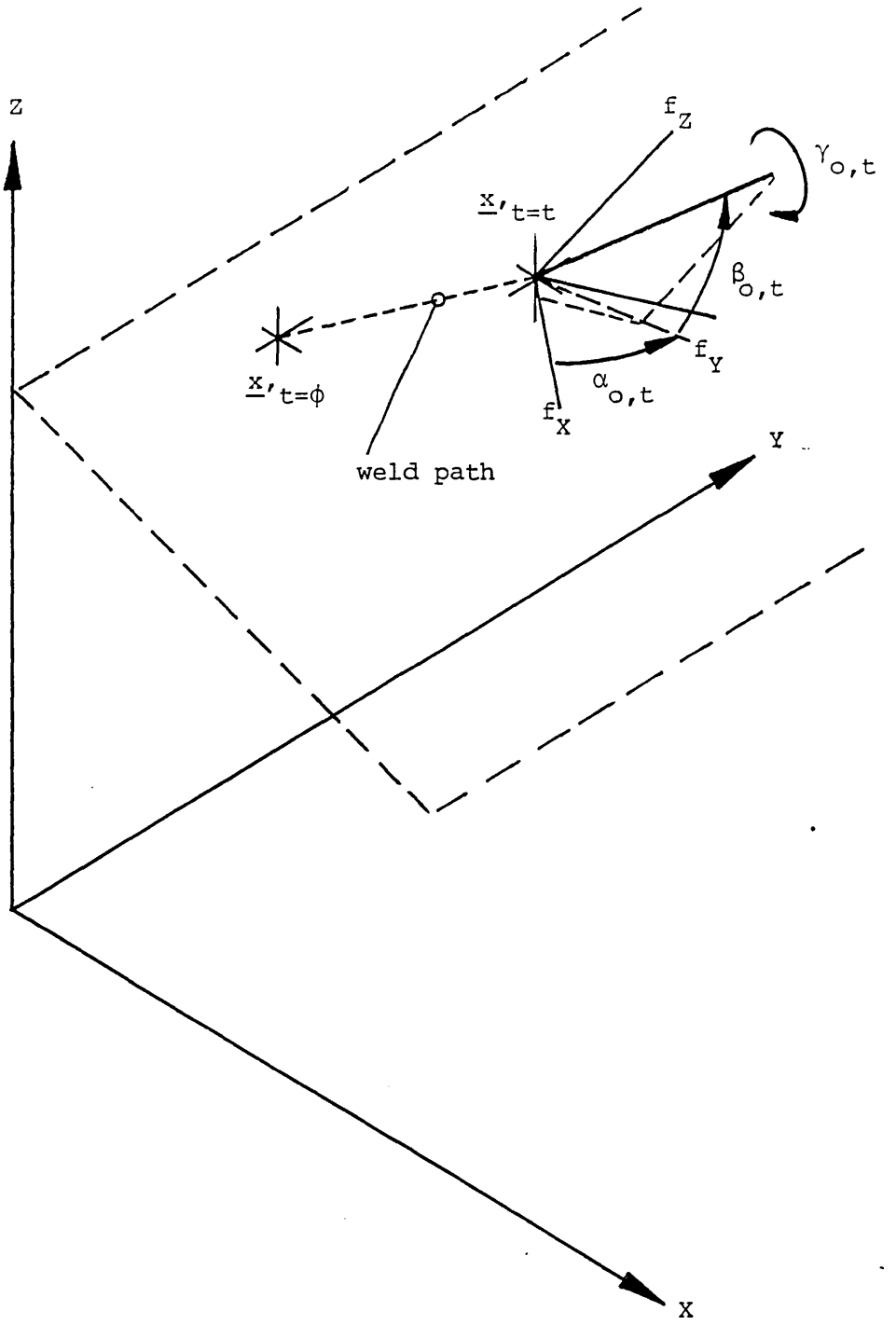


Figure 5.7 Spatial Function Orientation Parameters

6 ROBOT JOINT CONTROL

At low speeds it is possible to consider the robot arm as a mechanically uncoupled system. The control strategy employed if the system is assumed uncoupled, considers each joint seperately and is termed independent joint control. Thus the demands and responses of each joint are considered without regard to the effect due to the motion of any other joint. At higher speeds, however, this assumption of independence becomes less valid due to:

(i) Varying effective moments of inertia.

(ii) Torque coupling between the degrees of freedom.

(iii) Forces proportional to velocity product terms.

The way in which the mechanical system interacts can be evaluated by the analysis of the manipulator dynamics.

6.1 MANIPULATOR DYNAMICS

The two main approaches to derive the dynamic equations of motion are:

(i) Newton-Euler equations.

(ii) Lagrange equations.

The derivation of the dynamics using the Newton-Euler technique involves the free body analysis of each link. The dynamic forces and torques

acting on link i, can be determined by the use of Newton's Law; (i), and Eulers Law; (ii) as follows:

$$(i) \qquad f_{d,i} = m_i r_i \tag{6.1}$$

$$(ii) \qquad n_{d,i} = I_i \theta_i + \theta_i x I_i \theta_i \tag{6.2}$$

where f_i is the acting force on mass m_i with acceleration r_i; n_i is the acting torque on a link with inertia; I_i, angular velocity θ_i and angular acceleration θ_i. x is the vector cross product.

The linear acceleration of the centre of mass, and the angluar velocity and acceleration of each link, are obtained from the trajectory requirements and kinematics of the manipulator. Equations (6.1) and (6.2) can then be solved to give the dynamic forces and torques.

The interaction of links, i.e. the forces and torques acting on a link i from links i-1 and i+1, can be determined by static analysis using d'Alemberts theorems:

$$f_i = f_{i-1,i} - f_{i,i+1} + m_i g \tag{6.3}$$

$$n_i = n_{i-1,i} - n_{i,i+1} + \underline{d}_{i,i-1} x f_{i-1,i} - \underline{d}_{i,i+1} x f_{i,i+1} \tag{6.4}$$

where g is the gravity vector, and $f_{j,k}$ and $n_{j,k}$ are the effective forces and torques acting on link j from link k. $\underline{d}_{i,i-1}$ is the vector from the centre of gravity of link i to joint i-1. Similarly $d_{i,i+1}$ is the vector from the centre of gravity to the joint i.

The final step in solving for the inverse dynamics is to combine the Newton Euler equations with the statics. The combined form of the equations are as follows, [Luh et al 1980a]:

$$F_i = f_{d,i} + F_{i,i+1} \tag{6.5}$$

$$N_i = N_{i+1} + n_{d,i} + \underline{d}_{i,i-1} \times f_{d,i} + (\underline{d}_{i,i+1} + \underline{d}_{i,i-1}) \times F_{i,i+1} \tag{6.6}$$

where F_i and N_i are the net force and torque acting on link i.

The second method of evaluating manipulator dynamics is the Lagrange technique. The kinetic energy K and potential energy P of the manipulator as a whole is obtained from the summation of the kinetic and potential energy of each link (K_i, P_i respectively):

$$K = \sum_{i=1}^{n} K_i \tag{6.7}$$

$$P = \sum_{i=1}^{n} P_i \tag{6.8}$$

In the Lagrange equations it is necessary to define a generalised co-ordinate system to express the mechanics. It is useful to take the joint co-ordinate system as the generalised system. Defining the Lagrangian L by:

$$L = K + P \tag{6.9}$$

The generalised force f_i reflecting the torque required at a particular joint is obtained from the Lagrange equations:

$$f_i = \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{q}_i}\right] - \frac{\partial L}{\partial q_i} \qquad i=1,\ldots,n \tag{6.10}$$

where q_i is the joint i space variable.

Uicker, [Uicker 1965], applied the above equations to the problem of manipulator dynamics. The closed form of the system dynamics for an n linked manipulator is given by the Lagrangian equations:

$$f_i = \sum_{j=1}^{n}\left[\sum_{k=1}^{j}\left[tr\left[\frac{\partial W_j}{\partial q_i} J_j \frac{\partial W_j{}^T}{\partial q_k}\right]\right]\ddot{q}_k\right]$$

$$+ \sum_{k=1}^{j}\sum_{k=1}^{j}\left[tr\left[\frac{\partial W_j}{\partial q_i} J_j \frac{\partial^2 W_j{}^T}{\partial q_k \partial q_l}\right]\dot{q}_k\dot{q}_l\right] - m_j g^T \frac{\partial W_j}{\partial q_i} {}^j r_j \Bigg] \tag{6.11}$$

Where:

f_i is the generalised force term;

J_i is the inertia tensor expressed with respect to the current joint co-ordinate system j. It is obtained from the distribution of masses m, within the link j, at positions ${}^i p_i$ defined with respect to the local link origin:

$$J_j = \int ({}^i p_i \; {}^i p_i{}^T)\, dm; \tag{6.12}$$

m_j is the mass of link j;

g is the gravity vector;

${}^i r_i$ is the co-ordinate of the centre of mass of link j defined with respect to the local joint co-ordinate system;

t_r is the trace operator;

W_j is the combined homogeneous matrix definition of position:

$$W_j = A_0 \cdot A_1 \cdots A_j; \qquad (6.13)$$

q is the generalised joint variable with first and second time derivatives of q, $\dot{q}$ respectively.

The computation of the closed form of Lagrange equations has been undertaken, [Luh et al 1980a], and required 7.9 seconds when evaluated using Fortran on a PDP11/45 computer. Attempts to reduce the computational time required have been explored. Configuration Space Control, [Horn 1977], uses the technique of tabularisation of the dynamics. Position dependent terms are tabulated and the approximate dynamic response is determined by table look-up. There are however the problems of memory storage, configuration sensitivity, interpolation and table generation. The latter is especially important if a large range of loads is carried by the robot.

A method, which requires less computation than the closed form of the dynamic equations, is that of recursive analysis. This technique involves the expression of linear and angular velocities and accelerations starting from the base and working to the end effector. The second stage of the recursion is to determine the forces and torques on each joint. These are evaluated by working from the end effector back to the base. This formulation has been applied to both the Lagrange technique, [Hollerbach 1980], and Newton Euler equations, [Luh et al 1980a].

The combination of analytical techniques that are less mathematically intensive and modern powerful microprocessors, makes it possible to evaluate the manipulator dynamics on line (the Newton-Euler recursive technique can be evaluated in 4.5msec in floating point asambly on a PDP11/45, [Luh et al 1980a]). Given the interaction of the manipulator links, this can be incorporated within the robot controller as a torque predictor in the servo control of the joint drives.

If, however, the dynamic model is used for robot control there are a number of problems associated with the model.

(i) As velocities and accelerations increase, perturbation effects become more important. These can arise from such items as:
errors in length and mass;
elasticity of the links and transmission system;
backlash in transmission and friction.

(ii) The model must be controllable; however, in practice some variables cannot be controlled and others require modification of the robot design.

It is therefore important to incorporate other forms of control if dynamic modelling is used as a predictor for torque terms.

6.2 SERVO CONTROL OF THE ROBOT JOINTS

6.2.1 Control Strategies

The method by which specified input demands to the robot joints are converted to the output demands to the robot drives is termed the control strategy. The main types of control strategy relevant to robotics are as follows, [Brady et al 1982]:

(i) Open-loop control. This is the simplest form of control strategy. The joint input demands are converted directly to output drive demands with no feedback from the mechanical system. This form of control is used typically with stepper motor drive systems.

(ii) Linear control. Linear control incorporates a set, linear feedback relationship. A linear feedback law may have structure constraints in which only certain inputs and outputs are interconnected, as with the case when each joint is considered independently from any other joints.

(iii) Non-linear control. This class of control law includes: bang-bang control, global non-linear control, model reference control, and self-tuning or adaptive control.

Most of the present industrial robots incorporate linear control, usually independent joint control. The common control law applied is of proportional-integral-derivation (PID) type, [Lee 1981]. A further analysis of this control technique is given in Section 6.2.2

A number of pseudo-linear controls have been suggested, often of multi-variable form. Examples include Resolved Motion Rate Control (RMR), [Whitney 1969], and, more recently, an expansion of RMR; Resolved Acceleration Control, [Luh 1980b]. The latter deals directly with control of the robot end effector in cartesian base space. The control law is based on an internal dynamic model of the robot, and considers errors between the desired and actual position and oreintation of the end effector.

Referring to non-linear control, a number of strategies have been proposed. Freund, [Freund 1975], considered the general state space description of a non-linear system:

$$\underline{x}(t) = A(x,t) + B(x,t)\underline{u}(t) \tag{6.14}$$

$$\underline{y}(t) = C(x,t) + D(x,t)\underline{u}(t) \tag{6.15}$$

where $\underline{x}(t)$ is the state vector; u(t) and y(t) are the input and output vectors respectively; A, B, C, and D are matrices with compatable dimensions in which the elements are non-linear functions of the state of the system.

Freund showed that the general state space description could be adapted to a robot manipulator, and linearised to correspond to each drive of the robot. Zaballa, [Zaballa 1978], showed that this technique can be extended to any industrial manipulator.

Adaptive control, based on a model reference, has been investigated, [Savidis 1976], [Dubowsky 1981], [Le Borgue 1981]. The typical approach is to derive sub-optimal dynamic model of the robot. An adaptive technique is then employed to compensate for the difference between the actual robot and assumed model. The technique continuously updates parameters of the model, based on the response of the physical system.

The use of a non-linear control law has advantages in the response of the robot, but are complex to implement. There are also the real time requirements for robot control. Typically the loop closure frequency is in the order of 100Hz. The compromise of adaptive control, and a simplified model compensated by parameter tuning perhaps offers the optimal approach at the present time, [Vaha 1983].

6.2.2 Control Strategy Employed on the Imperial College Robot

As an initial stage, the servo control strategy employed in this project, is independent joint control. This is a linear decentralised control system, whereby the interaction of the robot joints is not considered as an input to the system. The interaction is monitored by considering the feedback from the mechanical system. The control law employed is that of proportional, integral, derivative (PID). Given an error in the required and actual position, of e, the mathematical form ofthe PID controller is as follows:

$$\theta d = K_D \frac{de}{dt} + K_p e + K_I \int e \, dt \qquad (6.16)$$

where θd is the motor drive demand; and K_D, K_p, K_I are the derivative, proportional and integral gain terms. The primary purpose of each gain term is as follows:

(i) K_p - proportional gain: to improve speed of response.

(ii) K_I - integral gain: assures steady state tracking of the input demand.

(iii) K_D - derivative gain: enhances stability and reduces the tendancy toward oscillation. (Often compensating for the de-stabilising effect of the integral gain).

The PID control law may be expressed in the following discrete time format:

$$\theta_d = K_D \frac{(e_t - 2e_{t-1} + e_{t-2})}{t_s} + K_p e_t + K_I \sum_{i=o}^{i=t} e_i t_s \qquad (6.17)$$

Where the subscripts refer to the sample instant in time; and t_s is the servoloop closure, or sampling time.

For this project the technique used to evaluate the gain constants was Root Locus Plots. This was accomplished by means of the Control Design Suite resident on the Imperial College mainframe computer. The initial stage was to determine the response characteristics of the major components of the control system. The two relevant modules are the microprocessor module, including PID, and the amplifier/motor/joint system module.

The constituent blocks of the microprocessor module are shown in Figure 6.1. The transfer function $G_\mu(s)$ can then be evaluated as:

$$G_\mu(s) = K_\mu \left[\frac{K_D S^2 + K_p S + K_I}{S} \right] \qquad (6.18)$$

where S is the laplace complex variable, and K_μ is the microprocessor overall gain, including the encoder and DAC gains.

Considering the robot drive/joint module, the general block diagram is shown in Figure 6.2. One simplification of this model is the exclusion of a velocity error integrator found with the GEC Gemdrive Axis Controller employed. The function of this integrator, is to ensure minimal velocity following error, specifically when varying loads are applied to the mechanical system. This velocity tracking can be treated as an extension of positional feedback, and ignored for the purpose of this analysis. This, and other simplifications introduced later, can be justified when considering the errors inherent in assuming independent, static inertial loads.

Taking the model of Figure 6.2, the effect of the back EMF and current feedback can be found by examining the drive/joint sub-module. Assuming L and f are zero the transfer function; $G_1(s)$, is given by:

$$G_1(S) = \frac{K_{a2}/K_b}{\left[\frac{R + K_i K_{a2}}{K_b K_t}\right] Js+1} \qquad (6.19)$$

The explanation of the various constants is given in Table 6.1. Given that $K_{a2} \gg R$, and $K_{a2} \gg K_t K_b$ (from manufacturers literature), the transfer function of $G_1(s)$ simplifies to:

$$G_1(s) = \frac{K_t}{K_i J S} \qquad (6.20)$$

Including the effect of velocity feedback gives the transfer function of drive/joint module, $G_j(s)$:

$$G_j(s) = \frac{1/K_{th}}{\frac{K_i J}{K_t K_{a2} K_{th}} S + 1} \tag{6.21}$$

If this is expressed in the form:

$$G_j(s) = \frac{K_{ma}}{K_j J s+1} \tag{6.22}$$

then the values of K_{ma} and K_j can be determined mathematically from manufacturers literature. However, a number of gains are tuned manually. Therefore an empirical derivation of K_{ma} and K_j was used. Monitoring the response of a drive/motor combination gave the value of the constants as:

$$K_{ma} = 14 \tag{6.23}$$

$$K_j = 3.7 \tag{6.24}$$

The Root Locus technique shows how the open loop response of a system affects the closed loop response. Since $K\mu$ includes the gain of the encoder, the position closed loop response can be obtained by considering the forward open loop characteristics of $G\mu(s)G_j(s)/s$ with unitary feedback.

The input to the Root Locus package is in the form of a polynomial such that:

$$G(s) = K \left[\frac{a_2 s^2 + a_1 s + a_0}{b_3 s^3 + b_2 s^2 + b_1 s + b_0} \right] \tag{6.25}$$

Where for the robot servo control:

$$\begin{aligned} K &= K_\mu K_{ma} \\ a_2 &= K_d \\ a_1 &= K_p \\ a_0 &= K_i \end{aligned} \tag{6.26}$$

$$\begin{aligned} b_3 &= K_j J \\ b_2 &= 1 \\ b_1 &= 0 \\ b_0 &= 0 \end{aligned} \tag{6.27}$$

The analysis using Root Locus is an iterative process whereby new values of K_d, K_p, K_i are evaluated based on the Root locus plot obtained from previous values. The way in which the PID gains are altered is very much an empirical process. The major criterion when selecting the gains was to give a response with a coefficient damping G>0.707. This gives a response to a step input with an overshot of less than 4%.

A typical root locus plot is illustrated in Figure 6.3. This was for the shoulder axis of rotation with a PID gain ratio of 1:13:0.015. Comparative analysis is easier when considering a proportional gain

of 1. This is compensated by changing the overall gain of the system; K_μ.

Note that the response improves with increased gain, and theoretically a response can be obtained with zero overshoot. However in reality the gain is limited by the power and response of the motors and the power of the drives.

Taking each joint of the robot, the PID constants were evaluated for each axis controller. The discrete time values of the constants, $K_{\mu D}$, $K_{\mu P}$, $K_{\mu I}$ were then obtained from:

$$K_{\mu D} = K_D / t_s \tag{6.28}$$

$$K_{\mu P} = K_p \tag{6.29}$$

$$K_{\mu I} = K_I \, t_s \tag{6.30}$$

where t_s is the servo loop closure time.

A list of the constant implemented within the axis controller cards is given in Table 6.2.

A full analysis of axis control algorithms has yet to be carried out. Further developments, including a simulation package of the dynamic response of the robot arm, will allow more sophisticated joint control algorithms to be designed and studied.

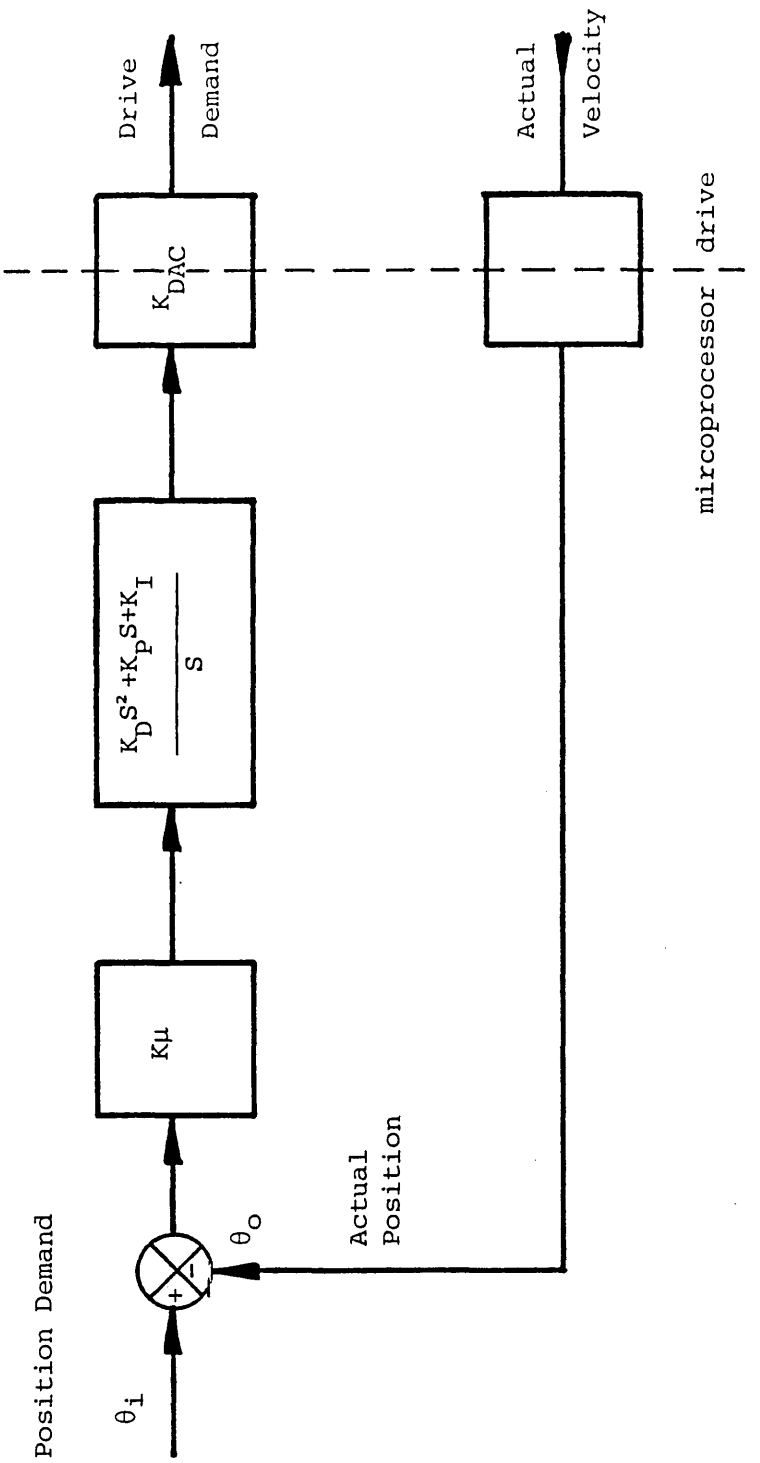


Figure 6.1 Microprocessor Control Transfer Functions

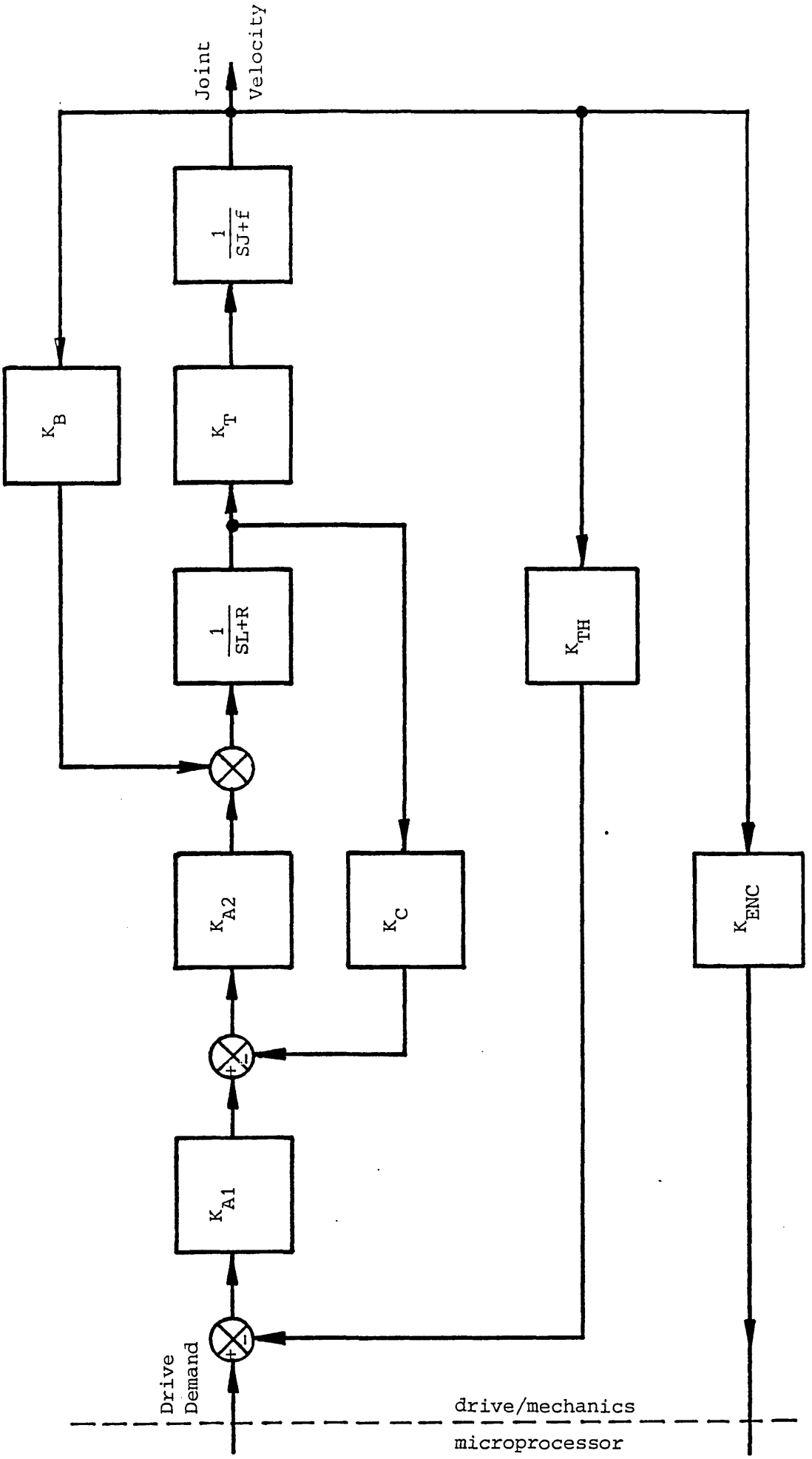


Figure 6.2 Robot Drive Transfer Functions

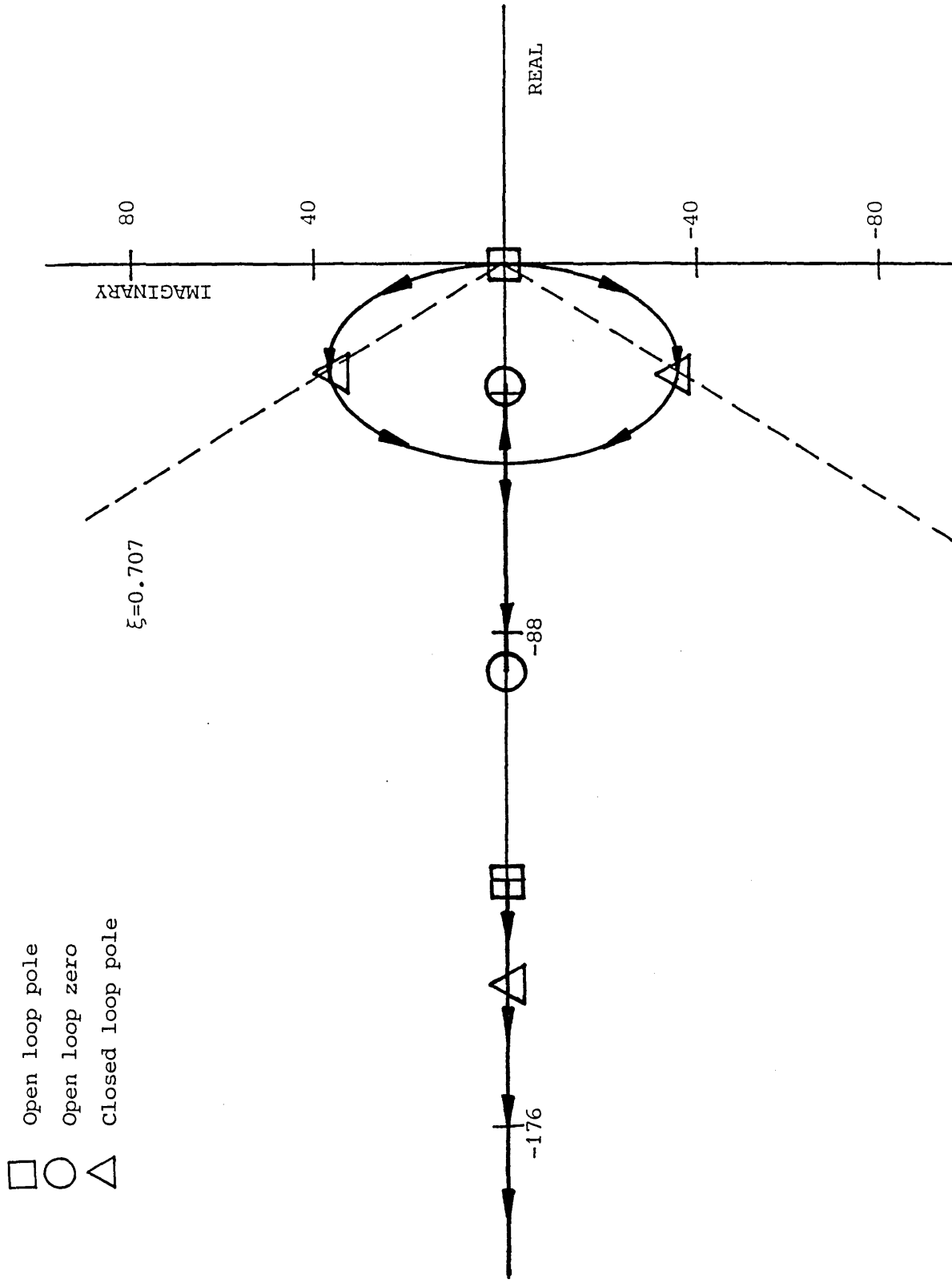


Figure 6.3 Root Locus Plot for Robot Axis Two

SYMBOL	DEFINITION
K_μ	Microprocessor Gain
K_D	Derivative Constant
K_p	Proportional Constant
K_I	Integral Constant
K_{DAC}	Gain of DAC
K_{ENC}	Encoder Gain
K_{A1}	Control Amplifier Gain
K_{A2}	Driver Inductance
L	Armature Inductance
R	Armature Resistance
K_T	Motor Constant
J	System Inertia
f	Friction
K_B	EMF Constant
K_c	Current Feedback Gain
K_{TH}	Tacogenerator Gain

Table 6.1 Robot Drive Transfer Function Constants

JOINT	CONSTANTS		
	PROPORTIONAL	INTEGRAL	DERIVATIVE
1 Base	1.0	0.069	3.6
2 Shoulder	1.0	0.13	1.5
3 Elbow	1.0	0.15	1.2
4 Wrist Pitch	1.0	0.16	1.1
5 Wrist Roll	1.0	0.16	1.1

Table 6.2 PID Constants for Each Robot Joint

7. DISTRIBUTED INTELLIGENCE ROBOT CONTROL

The task structure required of a continuous path robot controller is illustrated in Figure 7.1. All the components, apart from High Level Interface, have been discussed in previous chapters. High Level Interface is the major connecting phase between items external to the robot and the robot controller. It is closely related with the control structure and is therefore included in this Chapter. The requirements of this phase are discussed first, followed by a discussion of the overall control stategy employed in this project.

7.1 HIGH LEVEL INTERFACE

As robots become less insular, the number of external items, with which it must communicate, increases in number. Two groups of external devices can be determined. First, peripherals required of a typical stand alone computer system. Second, peripherals relevant to robot control. Examples of standard computer peripherals include:

(i) Visual Display Unit (VDU).

(ii) Graphics screen.

(iii) Disc storage.

(iv) Tape storage.

Those items relevant to a robot controller, when for example working within a Flexible Manufacturing System, (FMS), may include:

(i) Teach Pendant.

(ii) Vision, tactile and force sensors.

(iii) Task dependent I/O.

(iv) Hierarchial levels of plant communication e.g. cell controller, plant mainframe computer.

(v) High level robot languages.

(vi) Design and manufacture computer database communication.

The two groups are not always distinct. For example, modern teach pendants incorporating displays are similar in many respects to a VDU.

High level communication, environmental sensing and robot languages are closely related. Their importance to more advanced robotic systems, is reflected in the large amount of research carried out in these fields over recent years. In addition, a number of commercially available robot programming languages have appeared since 1979:

(i) AL: Produced at the Stanford Artificial Intelligence Laboratory and is based on concurrent Pascal. AL when pre-compiled to p-code can run on a PDP-11/45, running four robot arms simultaneously.

(ii) AML: Designed by IBM, this language is structured and interactive. The language runs on an IBM series/1 mini-computer.

(iii) HELP: Offered by General Electric Company, this language is interpretive, based on Pascal. The language runs on a DEC LSI-11/2.

(iv) JARS: Developed at the Jet propulsion Laboratory. Again the language is an extension to Pascal, incorporating subroutines relevant to robot control. It is run on a DEC PDP-11/34.

(v) MCL: Written by McDonnell Douglas Corporation, this language is an extension of APT. The language is orientated towards the programming of work cells. The McAuto proprietry version (MCL/11) uses a DEC PDP-11.

(vi) RAIL: An Automatix Inc. product, designed to control both robots and vision systems. The language is interpretive, running on a Motorola 68000 system.

(vii) VAL: Designed by Unimation Inc. for Unimation robots. A Basic like language run on a DEC LS1-11/23. An updated version VAL-II has recently become available.

The major reasons for the use of high level textual robot programming languages have been defined as follows, [Gruver et al 1983]:

(i) Teaching points by the use of teach pendant can be cumbersome for many operations, (e.g. palletizing).

(ii) As enviromental sensing becomes more widespread, an easy means of interfacing the robot and sensing devices becomes important.

(iii) A textual language permits robots to be programmed and simulated off-line, before implementation.

(iv) CAD/CAM can be integrated with robot programming.

There may be additional reasons, for example a controller may be required to operate in multi-task mode, both for a single robot and, if the system expands, to control a multiple robot cell, [Tyridal 1980]. A high level language that permits task orientated programming substantially eases the task of co-ordinating more than one manipulator. There are however a number of limitations associated with many of the current robot programming languages, [Soroka 1983]. Some of these limitations deal with the language, others with robot control. Items such as, who will program robots, program flexibility, and robot independence are relevant to commercial languages.

The use of high level languages also allow the inclusion of artificial intelligence for robot control. There is much research at the present time investigating the uses of artificial intelligence for decision making, and such items as obstacle avoidance, [Cameron 1982].

Although the scope of this project does not at present cover many of the aspects of high level languages or artificial intelligence, these items must be borne in mind when attempting to design a robot controller.

7.2 A DISTRIBUTED INTELLIGENCE ROBOT CONTROLLER

Recent years have seen a lot of interest in the use of distributed systems for robot control. A number of these incorporate minicomputers, such as the LSI-11/02, or PDP-11/34, [Gini et al 1980]. Others incorporate microprocessors entirely, using a mixture of 16 and 8 bit

processors in hiearchical structures, [Barbera et al 1979], [Albus et al 1981]. A number of distributed control systems have, in the last few years, been incorporated into industrial robot controllers. Siemens have developed a 16 bit system for point-to-point robots, [Becker 1980]. ASEA have incorporated two Motorola 6800 units for a continuous path robot, [Holmer 1982]. Cincinatti Milarcron have used the Intel 8086/87 and 8088 microprocessors for their new continuous path controller.

An introduction to distributed intelligence multi-microcomputer systems (DIMS) was presented in Section 1.2. In summary the four main advantages that may be realised by the implementation of a DIMS controller are as follows:

(i) Throughput: the ammount of processing required.

(ii) Flexibility: in terms of both hardware and software.

(iii) Reliability: of the hardware and data structures.

(iv) Cost effectiveness: the relative cost and merits of a DIMS approach compared to a single processor unit.

A number of aspects must be considered when investigating the use of a DIMS controller. These include:

(i) The determination of whether the use of a DIMS control can provide a substantially better controller system, which cannot be achieved by means of a single processor system. Also, if it can, is the improvement necessary, given the task of the robot.

(ii) The evaluation of whether a DIMS approach will provide an increase in cost effectiveness. This being achieved by either an equal capability for a lower cost than a single processor system, or an increased processing capability for equivalent cost.

(iii) It is important that the control problem is easily divided into discrete, semi-antonomous control functions. If no easily definable control boundaries exist, then the subsequent data transfer load between control modules could contribute to the total corruption of the system.

(iv) The use of a DIMS approach requires the development of a loosely coupled executive operating system; to enable each microcomputer to function within the system as a whole. The executive operating system will be responsible for items such as the administration of the parallel and hierarchical processing, and the intercommunication of the individual modules.

(v) Careful analysis must be carried out to ensure that system expandability, in terms of both hardware and software, can be accommodated. Degradation of either input/output control, or system communication must be avoided.

If these criteria can be sufficiently fulfilled, then additional benefits in cost can be realised, including:

(i) Exploiting the cost advantages of high volume production units, high levels of semi-conductor integration, and rapid design and implementation using the latest technology.

(ii) The cost of additional processors is low enough to allow the stocking of spares, reducing downtime costs.

(iii) Complicated software systems employed with mini-processors are eleminated.

It was felt that the opportunities and benefits offered by DIMS were sufficient to justify a distributed system for robot control. This was therefore the concept employed by the author for the development of a robot controller.

Three major features that must be considered for a DIMS controller are as follows:

(i) Software or task structure.
(ii) Hardware structure.
(iii) Inter-processor communications.

These items are closely related, and there are certainly areas of overlap. Thus although treated separately, each item was not considered in isolation.

Methodologies for the design of distributed control systems for machine tools and robots usually follow the "top down" approach,

[Duffie 1981]. Using this strategy the sequence of the design is of the form:

(1) Identify the requirements of the system from a global point of view. This includes simple models of the control process and indications of time constraints. this stage also includes consideration of high level communications, and system flexibility and reliability.

(2) To partition the process into semi-autonomous control functions. The sub-processes defined should require minimal inter-processes communication, and items such as real time response should be contained within a single process when possible.

(3) Choose the suitable processors to fulfil the task sub-process requirement. Considerations here include computation, interface, and response times. A large number of criteria affect processor choice, including:

(i) Programming flexibility, (instruction set).

(ii) Architecture, (speed, wordlength etc).

(iii) Software/firmware, (development, debugging).

(iv) Memory type and memory size.

Other criteria relevant to the complete system include availability, reliability, and technical support.

(4) Communication network design. This includes the network architecture and communications protocol.

(5) Allocation of sub-processes to processors, including software and hardware design. This stage will also include the development of an executive operating system.

The design strategy of the DIMS robot controller design for this project will now be discussed in more detail.

7.2.1 Software Modules of the DIMS System

The major software or process task requirements of a robot controller can be grouped as shown in Figure 7.1. These modules reflect the work defined within this thesis. there are also additional modules that may be incorporated at a later date, such as dynamic compensation, or adaptive control.

There is an inverse relationship between algorithm complexity and system bandwidth. To ensure adequate control, the robot joints must be servoed between 60Hz and 200Hz, [Paul 1980]. This factor influences the number of processors required to accomplish robot control at an adequate rate for smooth motion.

The number of software modules incorporated for a specific controller depends on the level of sophistication required, and physical specifications of the robot. This in turn depends on the anticipated task of the robot.

7.2.2 Hardware Structure

As with software modules, the hardware structure is dependent on the level of sophistication required of the controller.

The relationship between controller sophistication and the flexibility of a DIMS concept is discussed in more detail in Section 7.3.

The real time computation necessary of a continuous path robot controller requires greater processing power than is available with traditional 8 bit microprocessors. It was therefore necessary to evaluate the performance of the more recent 16 bit processing units. The major options of 16 bit processors were:

(i) 8086 - Intel.

(ii) MC68000 - Motorola.

(iii) TMS9900 - Texas Instruments.

(iv) Z8000 - Zilog.

The Intel 8086 is primarily an upgrade of the 8080 family which tends to limit programming options. However the 8087 arithmetic co-processor dramatically increases mathematical processing times. Motorola and Zilog have based their instruction set on an analysis of the most frequently used instructions. The Motorola 68000 is a pseudo 32 bit processor, having 32 bit registers and a powerful instruction set. The Zilog 8001 is again a powerful processor combining features of both mini and microcomputer systems. The TMS9900 is slower than the other processors, primilarly due to its earlier introduction.

The mathematical intensity of robot control suggested the Intel 8086/87 combination or NS16032. However anticipated arithmetric upgrading of the MC6000 (MC6020), and Z8001 (Z8004) reduced this

advantage. The relative merits of each offered no clear indication for choice.

The deciding factor was the compatability of the MC68000 with the 6800 family. Experience had already been obtained with MC6809 processors for machine tool control [Darzel 19??] and the use of the MC68000 allowed upward mobility with the same manufacturer. Although early work was carried out using modified 68000 processor boards, the two later systems that offered the best features were the Cifer 68000 Auxiliary Processor Board, and the Sage II. The hardware specification of each is given in Appendix A, however both support the IEEE 488 interface bus was chosen as the communication bus. In addition MC6809 boards, developed at Imperial College were used for the less mathematically intensive processes (again detailed in Appendix A).

The various hardware configurations possible when using a DIMS concept and modulator software are discussed in more detail in the following Section.

7.2.3 Network Architectures

Interprocessor communication takes place along a link between interfaces, the bandwidth of which must satisfy the data transmission rates of the system. In addition the network design must not inhibit the DIMS potential of flexibility, reliability, performance, and cost effectiveness. A number of network structures are available, [Duffie 1980]. Illustrated in Figure 7.2, these include:

(i) Loop Network: Each processor is connnected to two neighbours. A message is placed in the loop and passes around the loop. An

address linked to a message determines the destination processor. Flexibility, in terms of loop expansion is good, however a single link failure will effectively halt all transmissions.

(ii) Completely Interconnected Network: Each proecessor is connected to every other processor. This means direct processor communication. The logical complexity of the interface is low, but the flexibility is poor. The number of interconnections increases with the square of the number of processors. However reliability is high.

(iii) Star Network: A central processor acts as distributor for all messages. Flexibility may be poor, due to all processors being linked to the central processor. Also failure in the central node would halt all communications. Interface complexity is, however, low for all interfaces.

(iv) Global Bus Network: With this structure all processors communicate via a global bus. Messages can be sent directly between processors, but some bus management may be necessary if data transmissions are high. Flexibility is good, an additional processor is simply connected to the bus. Interfacing is easy due to the simple bus structure. Limitations include bus failure, halting all communications, and bus bandwidth. The latter can impose severe restrictions on data transmissions.

(v) Irregular Networks: Processors in an irregular network can be connected to any other processor. This allows a number of paths of communication between two processors. Flexibility and reliability is good, however interface complexity is high.

The relative merits and limitations of each must be considered for each type of application. For this project a global base network was chosen. The fact that the separate processor cards are not physically distributed enables this form of communication to be easily interfaced by means of a processor back plane. There are also standard interface protocols often handled by dedicated chips. The global bus standard adopted for this work was the IEEE-488 General Purpose Interface Bus (GPIB).

This is an asynchronous, parallel instrumentation bus, having eight parallel data lines, and eight parallel control/status lines. Communications are possible up to a data rate of one mega-bit per second.

There is also the advantage of custom chips (TMS9914A) which relieve the burden of handshaking and bus protocol.

The reasons for using the IEEE-488 are mainly historical. Previous work had produced processor routines for the chip interface. Also it was felt that the system could be devised and tested using this bus and if necessary modified at a later stage, when improved bus standards relating to the manufacturing industry become established.

The various hardware configurations possible when using a DIMS concept and modular software is discussed in more detail in the following section.

7.3 SYSTEM IMPLEMENTATION

7.3.1 Distributed Control for a Five dof Robot

The hardware and software modularity concepts employed in the current project allows considerable system flexibility and expandability. The implementation of DIMS concepts can be illustrated by considering the distributed control system developed for the Imperial College five dof robot.

The constituent microcomputers, used for the robot control are contained in the robot control cabinet (See Figure 7.3). The cabinet also houses the dc servo amplifiers, power supplies for the amplifiers and control system interlocks, relays, contactors and protection circuitry.

The control microcomputers reside in the controller rack, on double hieght Eurocards. The constituent cards talk via backplane communication using the IEEE-488 GPIB. The card/bus interface is affected by means of the TMS 9914A GPIB chip, and two interface buffer driver chips. The rack has its own internally mounted switch-mode power supply.

The processors employed, and the role of each within the overall control strategy is now discussed in more detail.

(1) 68000 Based Microcomputer No. 1: This processor acts as system supervisor and is responsible for the following tasks:

(i) Major operating system and peripheral interface. This includes items such as memory management and data handling, program editing and software development.

(ii) High level communications interface. This includes communications with sensing devices, and higher level controllers, as well as the teach pendant and VDU/keyboard.

(iii) Error monitoring and system diagnostics.

(iv) Execution of the robot operating system. This refers to control of the GPIB, acting as bus master, and the intercommunications of the microcomputer based components of the system.

(v) Robot command language interface. At the present time this role is limited to the execution of robot commands termed pseudo-level one commands. These commands are virtual commands that can be acted upon independently. At present they are interpreted directly. They are virtual in the sense that higher level languages will accomplish the same task, but may not use the commands directly. An example of the basic commands, and their implementation is presented in Table 7.2.

(2) 68000 Based Microcomputer No. 2: This processor acts on the numerically intensive tasks, and executes the algorithms required for:

(i) Interpolation.

(ii) Frame transformation.

(iii) Co-ordinate transformation including derivative motion.

This board also includes monitoring software to accomplish diagnostics. These diagnostic facilities not only include functional monitoring, but also positional singularities and articulation limitations.

The board can be software configurated to accomplish point-to-point and continuous path modes of motion. The co-ordinate transformation algoriths can be easily altered to reflect various robot configurations and degrees of freedom.

(3) Five 68B09 Axis Control Boards: There is a dedicated board for each axis of robot motion, allowing the system to be matched to the physical configuration of the robot. (This concept is well reflected in the modular design of robots as discussed in Section 2.4). The major functions of each axis card is as follows:

(i) Servo loop closure by means of proportional or PID algorithms. There are facilities to alter the servo constants.

(ii) Hardware and software limits. Software limits may be down loaded or defined at teaching. The joint limits cause the joint brake to be applied.

(iii) Monitoring the joint following error for both mechanical failure and trajectory accuracy.

(iv) Notification to the Vector Profiling Unit to enable base space velocities and accelerations to reflect the required path accuracy.

(v) Recording of absolute joint position. Necessary if the robot is used in an inspection capacity. The axis board checks the joint value when an inspection probe measures a component.

(vi) Error checking by comparison of software joint values with joint potentiometer measurement.

(4) A 68B09 Board for Velocity Profiling: Although a seperate board for velocity profiling was used initially, the close coupling of velocity profiling and interpolation will require the incorporation of these two software modules on the same processor board. The role of the velocity profiling module is:

(i) Reference word generation. This includes acceleration and deceleration profiles. The profiles can be default values or downloaded from the system supervisor.

(ii) Velocity demand correction. If the robot is required to move as fast as possible, or within trajectory limits, the unit can adapt the reference word demands. The demand values can be decreased by fixed proportions of the absolute value.

(iii) Diagnostics, both in terms of self checking, and velocity and acceleration demands.

(iv) Interaction with the supervisor and interpolation units to notify of possible vector termination. The unit can then respond, either to decelerate to a static position, or change the velocity demands for the next vector to be executed.

The overall system is initialised by driving the robot to a known position. The reference position may be determined by limit switches, potentiometers, or some other datumning device.

(5) Board for Handling Input/Output, (I/O): To allow the controller to interface at a low level with peripheral items an I/O card is included in the rack. The signals are buffered and opto-isolated. Several types of input and output signals must be considered.

At present a single Sage II is emulating the tasks of both 68000 processors. This allows software development and gradual upgrading of the system. The original intention was to include the Cifer board as the second 68000 processor. This reflects the fact that the Cifer is a cheaper single board computer. However the amount of data that was found necessary to pass between the system supervisor and transformation units conflict with the aims of a loosely coupled system. The use of a different, faster communications bus or direct memory acess by the two 68000 systems could relieve this problem. However, even if the communications problems are sorted out, there are also the difficulties involved with software development on the Cifer board.

The possibility of a single board Sage II, and a 10MHz version to be available shortly, makes the combatibility of two Sage units a feasible implementation.

7.3.2 System Versatility

The system versatility is obtained through hardware and software modularity. Apart from the correspondence between the number of axis cards and dof of the robot, software module can be incorporated into various cards.

For example, a simple version of the control structure would be for a pick-and-place four dof robot. In this case on-line transformation is not required, interpolation being in joint space. The system would then use only one 68000 microcomputer and four axis boards. This would still allow programming in base space, a useful facility, but when running would not need the added computational ability of a second 68000 board. However the system could be easily upgraded at a later stage to allow continuous path motion by the addition of the second board and minor software changes.

Similarly a sophisticated robot controller, incorporating all modes of vector profiling and interpolation, and including dynamic rotational frames may require three 68000 boards. Two boards would divide the tasks of vector profiling, interpolation, frame transformation, and co-ordinate transformation. One board would act as system supervisor.

Apart from the inclusion of software functions developed in this project the system can be expanded to perform even more sophisticated

roles. Examples include dynamic modelling and adaptive control. These may be accommodated by hardware additions or processor upgrading.

7.3.3 Algorithm Software

Due to the real time requirements of robot control, the efficiency of the internal algorithms is a crucial aspect of the control system. An analysis of the computational requirements and relative processing power of the constituent microprocessor showed the trajectory generation and transformation algorithms, executed on the 68000, to be the most critical section.

Considering the algorithm execution requirements, it was necessary to implement the algorithms in processor assembler code. A comparison of high level and assembler implementation of algorithms on the Sage II showed a decrease in the execution time by a factor of 12 when using assembler code.

An important additional factor in timing is the internal accuracy of elementary routines. A basic guide to the internal accuracy can be obtained by considering the resolution requirements of the physical system. The greatest demands on resolution occur at the base and shoulder joints of the robot. A resolution of ±0.5mm at a 2m reach requires a resolution of 1 : 12600; a data accuracy of 14 bits. The measurement of the system should be at least four times the required accuracy, giving a data accuracy of 16 bits. This was therefore reflected in the initial data structure employed in the system. A floating point format as shown in Figure 7.4(a) was used. The structure employed a 16 bit mantissa, and seven bit exponent, giving a range (in decimal) of:

$$nnnnn \times 10^{-37} < N < nnnnn \times 10^{37} \qquad (7.1)$$

where n represents a decimal significant figure, and N is the possible internal value that can be accommodated.

The use of a floating point structure rather than integer, reflects the data range required. This is incorporated primarily to stabilise the inverse kinematic evaluation near ill conditioned values.

As the arithmetic requirements increased, with the addition of more complex algorithms, it was found that the 16 bit internal data structure was unsatisfactory. The structure was therefore upgraded to a 24 bit mantissa, as illustrated in Figure 7.4(b). This structure still maintained the data range but increased the decimal equivalent of significant bits to seven. The increased data accuracy was however at a cost of processing time. A comparison of 16 and 24 bit mantissa basic functions is given in Table 7.1. The algorithms for evaluating the elementary functions (sine, cosine etc) reflected the internal architecture of the processor, [Cody 1980]. It can be seen from Table 7.1 that the time overheads for the trigonometric evaluations are relatively high. Therefore techniques to reduce the complexity of trigonometric functions were investigated.

Considering the 68B09 cards used for axis servo control, and velocity profiling, the arithmetical demands were less stringent. However execution times were more important, specifically for servo control. The internal architecture of the 68B09 processor makes the use of floating point algorithms more time intensive relative to the 68000

processor. Therefore an internal data structure of 24 bit integer was employed.

Processors exist which have fast processing times for arithmetic funcitons. These are however expensive processors manufactured for the military market. (An example of a military processor is the AN/AYK-14(V) which can evaluate a 32 floating point sine function in 24μs, [Cassola 1982]). Most consumer market processor chips have relatively slow arithmetic function processing times. One technique that can be employed to aid processing of trigonometric functions; incorporates look up tables.

Given a function; sin $(x+\delta x)$, this can be expanded by means of the Taylor series:

$$\sin(x+\delta x) = (x+\delta x) - \frac{(x+\delta x)^3}{3!} + \frac{(x+\delta x)^5}{5!} \qquad (7.2)$$

Expanding and collecting terms:

$$\sin(x+\delta x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} \ldots + \delta x\left(1 - \frac{x^2}{2!} + \frac{x^4}{4!} \ldots\right) - \frac{\delta x^2}{2}$$
$$\left(x - \frac{x^3}{3!} + \frac{x^5}{5!} \ldots\right) \qquad (7.3)$$
$$= \sin x + \delta x \cos x - \frac{\delta x^2}{2} \sin x$$

This expansion can be used with a look up table to decrease the computation time required to evaluate the sine and cosine functions.

A technique for the arctangent function provides a similar expansion for use with look up tables. For example the first two terms for the arctangent series are:

$$\tan(x+\delta x) = \tan x + \frac{\delta x}{1+x^2} \qquad (7.4)$$

Again computational savings can be obtained when using an expansion technique and look up tables for the arctangent function.

When the relative costs, in terms of computation of the arithmetic routines, are compared, a choice can be made between various equation options. For example, the four methods of defining the paramater l, given in equations (3.40) to (3.43), when compared in the light of computation requirements, suggest the use of the format:

$$l = \delta x \cos\alpha + \delta y \sin\alpha$$

The computational cost of arithmetic routines also illustrates the importance of the less numerically intensive algorithms, developed in this thesis.

The first DIMS system employed involved the use of one 68000 unit for interpolation, frame and co-ordinate transformation. This card was also used for limited high level interface. In addition five 68B09 cards were used for servo control, one for each robot joint. Another 68B09 card was employed for vector profiling.

A servo loop closure time of 8msec was achieved using the 68B09 cards. However the transformations algorithms required 27msec. Therefore an additional interpolation was carried out on the joint demand values to match the 68000 and 68B09 processors. This allowed a servo frequency of 100Hz with a marginal reduction in accuracy.

The next stage is to package the transformation algorithms onto an independent 68000 board. In addition the mathematical routines, specifically trigonometric algorithms, require a decrease in processing time. This can be achieved by using the trigonometric methods discussed previously.

An evaluation of the Control system is linked to the testing of the mechanical system. Conceptually the use of DIMS has been proven. The technical benefits of DIMS have been illustrated. A more detailed study is required to evaluate the financial implications of DIMS.

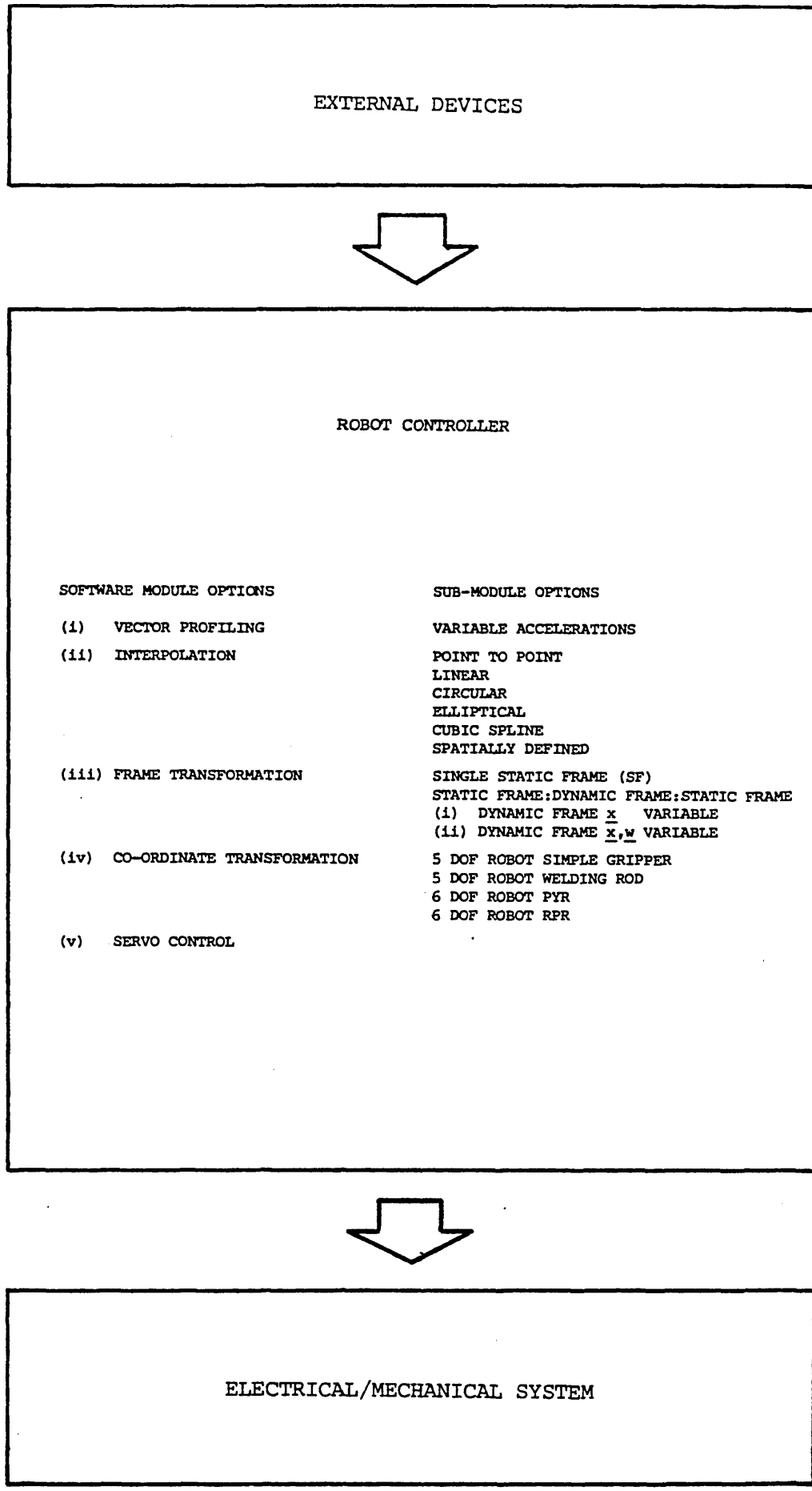
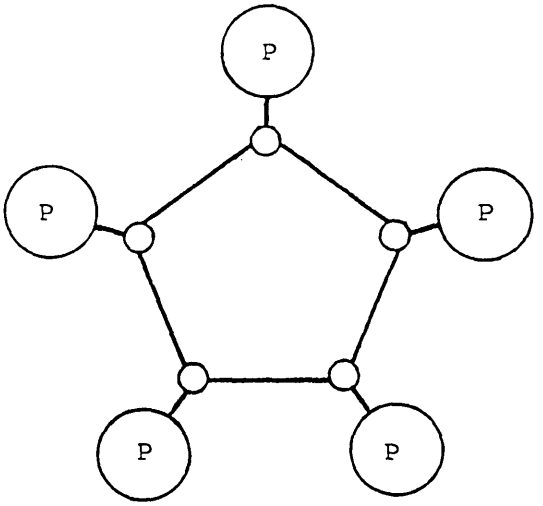
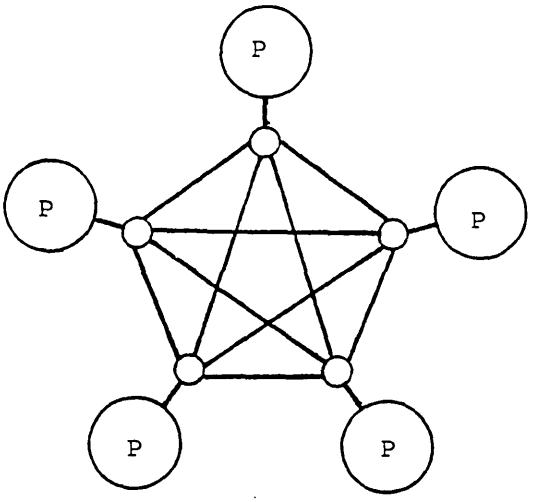


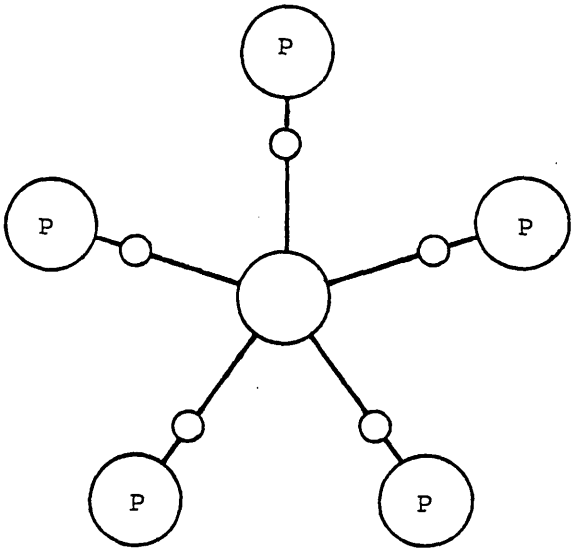
Figure 7.1 Task Structure for a Continuous Path Robot



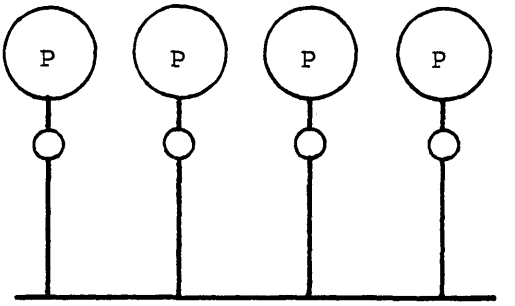
Loop network



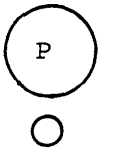
Comletely interconnected network



Star network

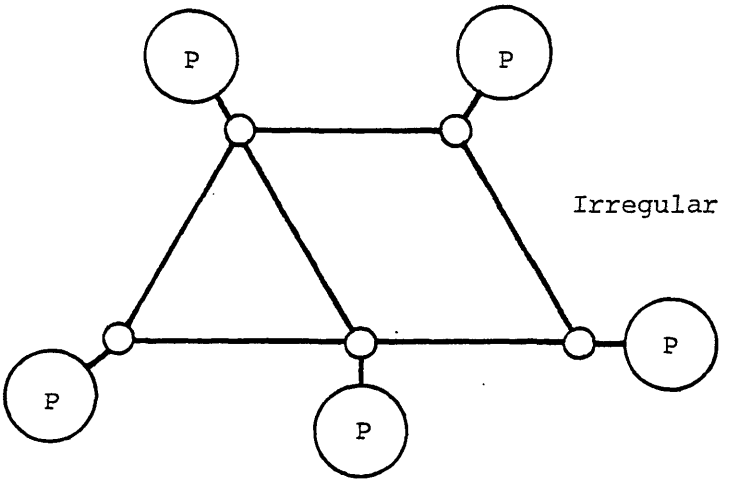


Global bus network



processor

interface



Irregular network

Figure 7.2 Network Architectures

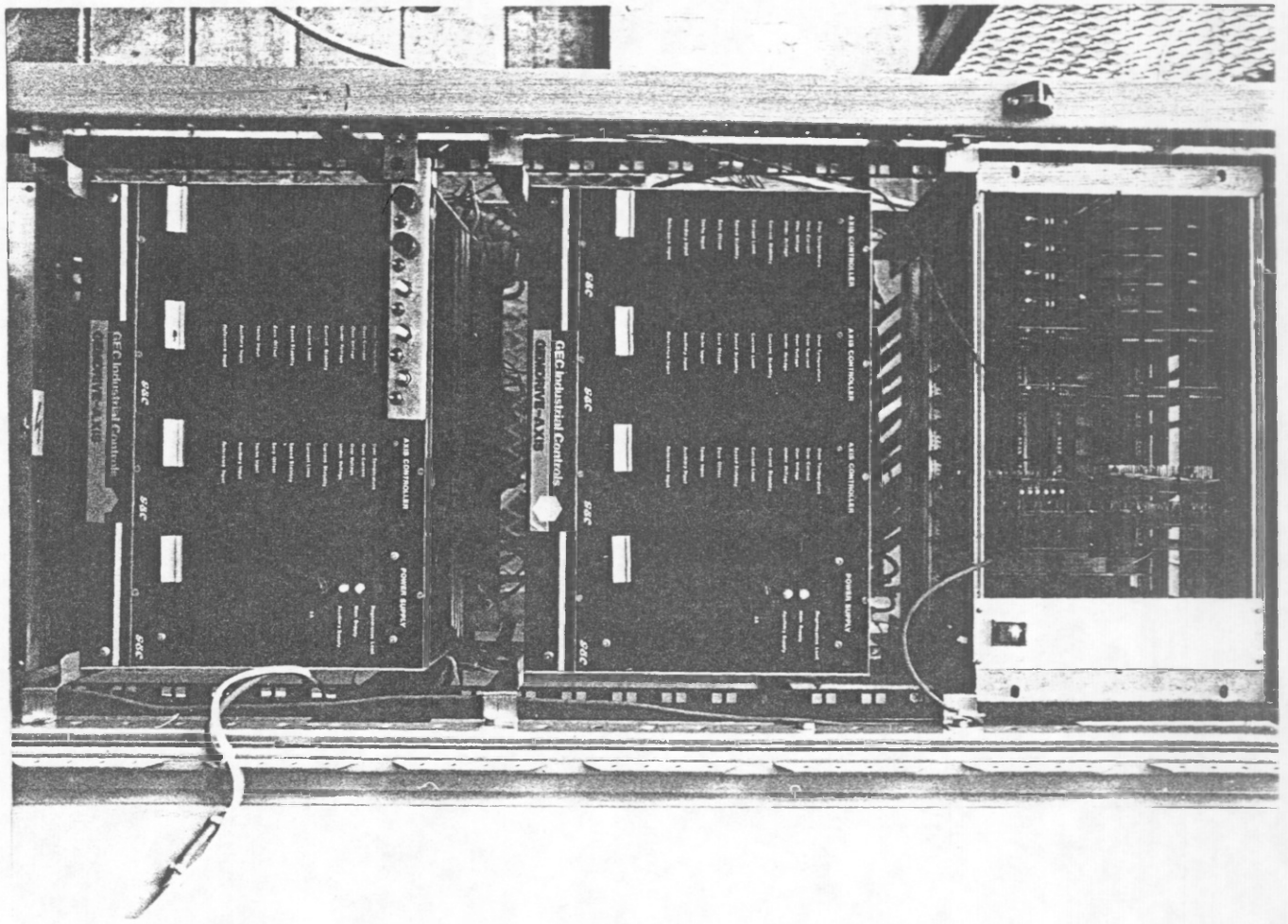
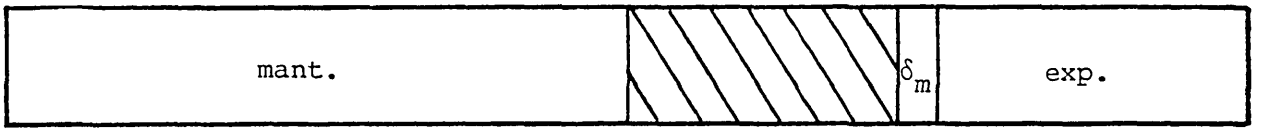
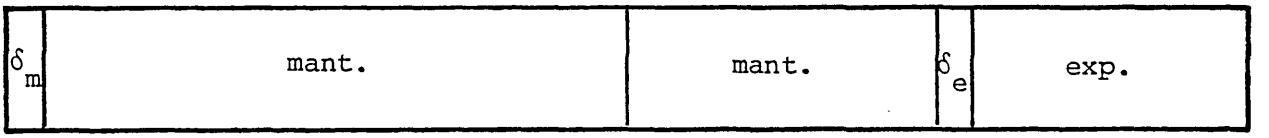


Figure 7.3 Imperial College Robot Control Cabinet



(a) 16 bit mantissa



(b) 24 bit mantissa

mant.	mantissa
exp.	exponent
δ_m	sign mantissa
δ_e	sign exponent

Figure 7.4 Floating Point Data Format

FUNCTION	TIME (μsec)	
	24 bit mantissa	16 bit mantissa
Add/Subtract	60	30
Multiply	70	27
Divide	220	35
Sine/Cosine	670	350
Arctangent	1570	600
Square Root	870	240

Table 7.1 Algorithm Execution Times

COMMAND	SOURCE	DESTINATION	DATA FIELDS
PTP Interpolation	Delegator	Traj Gen/Transf^n	-
Elliptical Interpolation	Delegator	Traj Gen	5
Robot Configuration	Delegator	Tranf^n	1
Move Relative to Current Frame	Delegator	Traj Gen	12
Set End Effector Parameters	Delegator	Transf^n	4
Excessive Following Error	Axis Cards	Traj Gen	-

Table 7.2 Examples of Pseudo Level One Commands

8 CONCLUSIONS

The work, presented in this thesis, can be divided into three areas. First; the mechanical and electrical design of a robot. Second; the structuring of the robot control problem into semi autonomous tasks, and devising algorithms for those tasks. Third; the implementation of the various algorithms, using a distributed processor architecture, to provide a flexible robot controller. These areas are treated in sequence, relative to their introduction in this thesis.

In Chapter 2, features important in the mechanical design of a robot arm have been introduced. Although various techniques for robot design have been suggested non of these successfully provide a complete methodology for mechanical design. One feature of the robot designed by the author was the use of chain. This was intended to provide a cheap robust method of transmission and gear reduction. Although the mechanical performance was satisfactory, when analysed in detail, only marginal cost savings could be realised. This was due to the machining and assembly cost of the various pinions and jockeys. In addition the take up of chain wear, looked likely to prove problematic. Self adjusting tensioners were difficult to employ without introducing areas of transmission vibration and non-linearities. The tensioners eventually employed requires human adjustment, and it was in this area that problems were envisaged. Scheduling maintenance and ensuring correct tensioning is difficult to achieve in the industrial environment. Therefore it was decided that the next mechanical prototype should employ harmonic gears and, if necessary, toothed rubber belts.

There have also been some problems with bearings, specifically in the shoulder joint. Although deep groove ball bearings were used, it is felt that a taper roller pair, as employed in the base, is necessary.

The overall performance of the mechanical and electrical design has, to the present time, been analysed in only a qualitative manner. A quantitative analysis requires both the physical robot and controller. A strategy for performance evaluation is an important item that is, at present, being implemented. Apart from the work carried out by Warneke and Schraft, [Warnecke and Schraft 1979], other strategies for testing robots have been suggested, [Anshin 1978].

The search for lighter, stiffer robots involves aspects of both material construction, and drive mechanisms. An area of interest, although expensive at the present time, involves the investigation of rare earth, direct torque motors, and composite limb structures. The latter, composite materials, lends itself to finite element analysis, both for design and performance monitoring.

Chapter 3 introduced the concept of co-ordinate frame transformation. It was felt that the separation of positional and orientation parameters allowed a more flexible frame transformation structure. These parameter groups, when treated separately, enable the frame transformation algorithm to function in a manner that allows more complex motions of the robot arm. An example is that of welding, where constant rod orientation, relative to the direction of motion, and detailed rod motions, can be easily superimposed on the basic welding direction of travel.

For both frame and co-ordinate transformations, an analytical technique has been developed by the author based on three dimensional geometric analysis. This compares with elemental matrix techniques previously employed. Both techniques are intuitive. The matrix technique requires inspection of the matrix elements before deciding mathematical treatment. The geometric technique requires inspection of the geometry before mathematical treatment. One difficulty with geometric analysis, is the care required with angle conventions. This aspect is perhaps less difficult with matrix techniques. However, providing care is taken to determine the conventions beforehand, it is felt that this will not cause too much inconvenience when applying geometric analysis.

Chapter 3 also defined the pitch, yaw and roll (PYR) convention employed for defining orientation of the robot end effector. It is felt that this convention, although not employed extensively in other research, provides a more complementary orientation definition for the engineering environment. The PYR orientation is intuitively managable by engineers, and can be easily obtained from engineering drawings. This latter point becomes significant as more robots become linked to the engineering databases used with Computer Aided Design and Manufacture equipment.

Chapter 4 involved co-ordinate transformation, the mapping of world to joint co-ordinates. To provide inverse kinematic equations, the technique of geometric analysis was again employed. The savings in arithmetic complexity, and therefore computational time, by using this technique is significant when compared to existing matrix techniques.

Algorithms have been designed, and proved, for all major robot primary and secondary configurations.

The technique of Pseudo Resolved Motion Rate (PRMR) control was also discussed. The way in which this technique, involving the Jacobian matrix, and the inverse kinematic equations can be combined to allow control of redundant manipulators was presented.

As the power of available microprocessors increases, the real time restraints of robot control become less critical. It is envisaged that in the future, the requirement for detailed analysis of the inverse kinematics will not be as important. When the performance of cheap, easily available, microprocessors reach a more advanced specification, it is felt that the PRMR technique will become used in more robot controllers. The simplicity of implementation and similarity with requirements of redundant manipulator control, makes PRMR a powerful technique in the area of co-ordinate transformation.

The phenomenon of derivative motion is an overhead in computation, that can be avoided. Its avoidance, however, requires the drive for a specific joint to be either at the joint or on the joint carrying limb. The need, to place the heavy, bulky drive assemblies toward the base of the robot means that compensation for derivative motion is necessary at present. Either improvements in the electrical and mechanical construction, or the use of more powerful microprocessors will eventually relieve this burden.

Chapter 5 discussed the area of trajectory genration. This area can be usefully divided into two areas. First; there is the task of vector generation with respect to time. Secondly; there is the task of spatial interpolation, the transformation of the vector time requirements into the actual spatial path. Algorithms for both these aspects have been devised by the author.

The framework for vector generation involve the velocity/time plane. A linear velocity profile was employed with self correction for spatial inaccuracies. Quantitatively, the use of velocity profiling provided a marked improvement in motion performance. However, a qualitative analysis is necessary to enable the merits of various velocity profiles to be compared. It is felt that for high performance manipulators, acceleration generation will also be required. Again this may be an area for future work.

Algorithms to enable the fast generation of a large range of interpolated paths, have been devised by the author. One spatial path important in welding is that of the ellipse. A fast algorithm has been devised which has only slight computational overheads compared to circular interpolation. A method for generation smooth paths passing through specified points, has been introduced based on cubic splines.

In accordance with the separation of positional and orientation parameters, the interpolation techniques apply to both seperately. As has been suggested, this allows more complex robot motions of the robot. The example of robot welding allows three interpolation algorithms to run seperately, and be superimposed to give a weave along the weld path,

with the welding rod to be held at a specified orientation to the motion.

Chapter 6 discussed the requirements of robot joint control. As an introduction to the difficulties involved, the dynamic interaction of the robot joints was illustrated mathematically. Time requirements allowed only the implementation of a three term controller, treating each joint independently. The area of joint control is however of utmost importance as the requirements for increased speed of robot motion grow. Thus the need for fast dynamic evaluation, and adaptive control becomes apparent. Much research is at present being conducted in this area, and this field is suggested for further study. Specifically the area of fast self tuning control algorithms offers much scope in robot control.

Again performance testing of the robot with various control algorithms is needed for further evaluation.

Chapter 7 draws the various techniques and algorithms presented, into a Distributed Intelligence Multi microcomputers System (DIMS). By dividing the robot control problem into semi-discrete functions it has been possible to implement a robot controller that employs DIMS. The use of a distributed processor system allows great scope in terms of expandability and flexibility. Various configurations of multi-processor architecture have been tried. The criterion for a semi-automonous, loosely coupled system suggests that a direct link, co-processor combination for frame and co-ordinate transformation is necessary. This may be achieved by a separate, high speed communications bus or Direct Memory Address (DMA).

The DIMS concept employed allows processor balancing between the number of processor cards, degree of control sophistication required, and physical configuration of the robot. Any change can be easily accommodated by insertion, or removal, of processor cards, and minor software changes.

Chapter 7 also introduced those items peripheral to direct robot control. Of specific importance is environmental sensing; vision, tactile and force. To enable a robot to respond and interact with the engineering environment, it is crucial that information is obtained using these peripheral devices. Much research is being undertaken at present in these fields. A DIMS controller will allow implementation of advances in these fields.

One other aspect introduced in Chapter 7 was that of high level languages. Many requirements of robots can be fulfilled at present, using low level teaching. However as robots are required to make more decisions concerning possible actions, the need for high level languages emerges. Requirements are diverse. Apart from interaction and control of environmental sensing, decisions are required relating to possible situations that may arise. Thus great interest surrounds artificial intelligence with reference to robot control which would allow more sophisticated roles to be undertaken.

It is felt that a DIMS philosophy allows advances in algorithmic techniques and items such as dynamic analysis to be easily incorporated at a later stage. By breaking up the control problem onto separate processors each can be treated individually and easily adopted into the

control system. The algorithms presented are relevant to the state of robot control at the present time. By using DIMS it is easier to update the features of robot control as improved techniques,and increased market requirements occur.

REFERENCES

Albus, J.S.

"Peoples Capitalism: The Economics of the Robot Revolution", College Park, 1979, p1.

Albus, J.S., Barbera, A.J., Fitzgerald, M.L.

"Hierarchical Control for Sensory Interactive Robots", Proc. 11th Int. Symp. on Ind. Robots, Tokyo, Japan, Oct. 1981, p497.

Anderson, L.H.

"Distributed Intelligence Microcomputer Systems (DIMS)", ISA-75 Industry Orientated Conf., ISA Reprint 75-504 Instrument Soc. of America, Oct. 1975.

Anshim, S.S.

"Acceptance Tests for Industrial Robots", Machines and Tooling, Vol. 49, No. 7, p11, 1978.

Asada, H.

"A Characteristics Analysis of Manipulator Dynamics Using Principle Transformations", Proc. Amer. Control Conf., Virginia, USA, June 1982, p1186.

Aspragathos, N., Hewit, J.R.

"The Use of Redundancy in a Manipulator", IEE Colloquium Control Division, Nov. 1983.

Barbera, A.J., Albus, J.S., Fitzgerald, M.L.
"Hierarchical Control of Robots Using Microcomputers", Proc. 9th Int. Symp. on Ind. Robots, Washington D.C., March 1979, p405.

Becker, H.
"Robot Control - A Microprocessor Control System for Industrial Robots", Siemens Power Engineering 11, No. 7 Systems and Equipment, 1980, p191.

Belshikov, S., Kerytko, O., Tress, K.
"Consideration of the Influence of Local Deformations of Industrial Robot Links Produced Upon Resilient Errors of Positioning", Proc. 11th Int. Symp. on Ind. Robots, Tokyo, Japan, Oct. 1981.

Benati, M., Morasso, P., Tagliasco, V.
"The Inverse Kinematic Problem for Anthropomorphic Manipulator Arms", Techn. Rep., E.E. Dept., Univ. Genoa.

Brooks, T.L.
"Optimal Path Generation for Cooperating or Redundant Manipulators", Proc. 2nd Int. Comp. Eng. Conf., Vol. 2, San Diego, California, Aug. 1982, p119.

Bollinger, J.G., State, G., Van Brussel, H., Dinsdale, J.
"Digital Controls and Feed Drives State-of-the-Art and New Developments", Annals CIRP, Vol. 9, No. 2, 1980.

CME

"Machine Tool Robots", Chartered Mechanical Engineer, Vol. 27, No. 4, April 1980.

Cameron, S.

"The Clash Detection Problem", DAI Working Paper No. 126, University of Edinburgh, Sept 1982.

Cassola, R.L.

"Floating Point Algorithm Design", Computer Design, System Design/Software, June 1982, p107.

Cody, W.J., Waite, W.

"Software Manual for the Elementary Functions", Prentice Hall, New Jersey, 1980.

Coiffet, P.

"Modeling and Control", Robot Technology, Vol. 1, Hermes Publishing, France, 1981.

Daniel, R.W., Cook, I.C.

"The Relationship Between Trajectory Sensitivity and Manoeuverability as a Design Aid for Revolute Fixed-Arm Industrial Robots", 1983.

Dalzelle, D.T.

"Intelligent Machine Tools and the Microprocessor", PhD Thesis, University of London, 1981.

Demaurex, M.O., Gerelle, E.G.R.
"Can I Build This Robot", Proc. 9th Symp. on Ind. Robots, Washington D.C., March 1979, p621.

Denavit, J., Hartenberg, R.S.
"A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", J. Appl. Mech., pp215-221, June 1955.

Derby, S.J.
"General Robot Arm Simulation Program (GRASP): Part 2, Methods of Joint Solutions and the Reachable Volume", Proc. Second Int. Comp. Conf., Vol. 2, San Diego California, August 1982.

Drexel, P., Schmalenback, E., Schweizer, M., Haas, R., Wanr, G.
"New Manipulatory Systems as Technical Aids in the Work Process", Part 4: Drives for Manipulating Systems Research Report HA 80-030, July 1980, pp1-216.

Drimer, D., Oprean, A., Petrescu, C., Petrescu, M.
"Research on the Construction of Industrial Robots", Proc. 10th Int. Symp. on Ind. Robots, Milan Italy, March 1980, p453.

Dubowsky, S., Des Forges, D.T.
"The Application of Model-Reference Adaptive Control to Robotic Manipulators", Jnl. Dyn. Syst. Meas. Control, Vol. 101, No. 3, 1979, p193.

Duffy, J.

"Analysis of Mechanisms and Robot Manipulators", Edward Arnold (Pub.) Ltd., 1980.

Duffie, N.A., Bollinger, J.G.

"Distributed Computing Systems for Multiple-Processor Industrial Control", Annals CIRP, Vol 29, Jan. 1980, p357.

Duffie, N.A.

"An Approach to the Design of Distributed Machinary Control Systems", IEEE-IAS, Philadelphia, Oct 1981, p1.

Enslow, P.H. Jnr.

"Multiprocessor Organisation - A Survey", Computing Surveys, Vol. 9, No. 1, March 1977.

Fourmer, A.

"Robotic Movement Generation. Application of Generalised Inverses and Pseudo-inverses", Thesis, Montpellier, 1980.

Freund, E.

"The Structure of Decoupled Non-linear Systems", Int. Jnl. of Control, Vol 21, No. 3, 1975.

Gaglio, S. et al.

"Computation of Inverse Kinematics and Inverse Dynamics in Manipulator Arm Control", Proc. 11th Int. Symp. on Ind. Robots, Tokyo Japan, Oct. 1981, p553.

Gini, G. et al.
"Distributed Robot Programming", Proc. 10th Int. Symp. on Ind. Robots, Milan Italy, March 1980, p61.

Greville, T.N.E.
"Some Applications of the Pseudo-inverse of a Matrix", SIAM Review II, 1960, p15.

Gruver, W.A., Soroka, B.I., Craig, J.J., Turner, T.L.
"Evolution of Commercially Available Robot Programming Languages", Proc. 13th Int. Symp. on Ind. Robots, Vol. 2, Chicago Illinois, April 1983.

Hamilton, W.R.
"Elements of Quarternions", Chelsea Publishing Co., New York, 1969.

Hollerbach, J.M.
"A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity", IEEE Trans. on Syst. Man. and Cybernetics, Vol. 10, No. 11, p730.

Holmer, A.
"New Generation of Robot Controllers", ASEA Journal, Vol. 55, No. 6, 1982, p139.

Horn, B.K.P., Raibert, M.H.
"Configuration Space Control", MIT, A.I. Memo, No. 458, Dec. 1977.

Infotech.
"Multiprocessor Systems", Infotech State of the Art Report, Infotech Intl. Ltd., Maidenhead, 1977.

Kalabin, I.V.,
"Differential Mechanisms for Manipulators", Machines and Tooling, Vol. 49, No. 7, 1978, p26.

Kamiya, Y., Yokoyama, Y., Takano, M.
"High Speed Accurate Positioning of Robot Arm - Reduction of its Residual Vibration", Bull. Japan Soc. of Prec. Engg., Vol. 14, No. 2, June 1980, p9.

Kinoshita, G.
"The Maneouvrability of a Manipulator with Multi-Joints", Proc. 11th Int. Symp. on Ind. Robots, Toyko Japan, Oct. 1981.

Koren, Y., Masory, O.
"Reference-Pulse Circular Interpolators for CNC Systems", Trans. ASME, J. of Eng. for Ind., Vol. 103, Feb. 1981, p131.

Kreysig, E.
"Advanced Engineering Mathematics", John Wiley and Sons, New York, USA, p376.

Le Borgne, M., Ibarra, J.M., Espiau, B.
"Adaptive Control of High Velocity Manipulators", Proc. 11th Int. Symp. on Ind. Robots, Tokyo Japan, Oct. 1981, p227.

Lee, B.J., Nho, T.S.
"The Positioning Control of Robot Using Microcomputer", Proc. 11th Symp. on Ind. Robots, Tokyo Japan, Oct. 1981, p237.

Lien, T.K.
"Co-ordinate Transformations in CNC Systems for Automatic Handling Machines", CIRP Journal: Manufacturing Systems, Vol. 9, No. 1, 1980, p49.

Lin, Y.C., Duffy, J.
"The Mapping and Structure of the Workspace of Robot Manipulators with Revolute and Prismatic Pairs", Proc. 2nd Int. Comp. Conf., Vol. 2, San Diego California, August 1982.

Luh, J.Y.S., Walker, M.W., Paul, R.P.C.
"On Line Computational Scheme for Mechanical Manipulators", Jnl. of Dyn. Syst. Meas. and Control, Vol. 102, No. 2, 1980a, p69.

Luh, J.Y.S., Walker, M.W., Paul, R.P.C.
"Resolved Acceleration Control of Mechanical Manipulators", IEEE Trans. Automatic Control, Vol. 25, No. 3, 1980b, p468.

Maekawa, M.
"Extensibility and Adaptability of Distributed Computing Systems", Proc. IEEE Comp. Soc. Fourth Int. Computer Software and Applications Conf., Oct. 1981.

Masory, O., Koren, Y.

"Reference-word Circular Interpolations for CNC Systems", Trans. ASME, J. of Eng. for Ind., Vol. 104, Nov. 1982, p400.

Mizutani, T., Hasegawa, K.

"On Applications of Differential Gear Mechanism to Manipulator", Proc. 11th Int. Symp. on Ind. Robots, Tokyo Japan, Oct. 1981, p611.

Paul, R.P.

"Kinematic Control Equations for Simple Manipulators", IEEE Trans. on Systems, Man., and Cybernetics, Vol. SMC-11, No. 6, June 1981.

Paul, R.P.

"Robot Manipulators Mathematics Programming and Control", MIT Press, 1982.

Renaud, M.

"Robot Manipulator Control", Proc. 9th Int. Symp. on Ind. Robots, Washington DC, March 1979, p463.

Roberts, L.G.

"Homogeneous Matrix Representation and Manipulation of N-Dimensinoal Constructs", MIT Lincoln Labs Document MS-1045, May 1965.

Saridis, G.N., Lee, G.S.G.

"Heuristic Control in Trainable Manipulators", Int. Automation and Control Conf., 1976, p712.

Saveriano, J.W.

"Industrial Robots Today and Tomorrow", Robotics Age, Summer 1980, Robotics Publishing Corporation.

Soroka, B.I.

"What Can't Robot Languages Do", Proc. 13th Int. Symp. on Ind. Robots, Vol. 2, Chicago Illinois, April 1983.

Sugimoto, K.

"An Approach to the Structural Synthesis of Robots", Proc. 9th Int. Symp. on Ind. Robots, Washington DC, March 1979.

Sung, C.K., Thompson, B.S.

"The Design of High Speed Machinary: Material Selection", Proc. 2nd Int. Comp. Eng. Conf., Vol. 2, San Diego California, August 1982.

Surnin, B.N.

"Design Features of Modular Type Robots", Machines and Tooling, Vol. 49, No. 7, 1978, p17.

Tarvin, R.L.

"Considerations for Off-line Programming a Heavy Duty Industrial Robot", 10th Int. Symp. on Ind. Robots, Milan Italy, March 1980, p109.

Taylor, R.H.

"Planning and Execution of Straight Line Manipulator Trajectories", IBM J. Research and Development, 1979, pp424-436.

TOKO

TOKO K3701 Function Generator Chip Users Manual, 1979.

Tyridal, P.

"New Ideas in Multi-task Real Time Control System for Industrial Robots", Proc. 10th Int. Symp. on Ind. Robots, Milan Italy, March 1980, p659.

Uchiyama, M.

"A Study of Computer Control of Motion of a Mechanical Arm", Bull. JSME, Vol. 22, No. 173, Nov. 1979.

Uchiyama, T., Tamamushi, K., Nishimoto, K., Akita, T., Araki, M.

"A Small Articulated Robot with High Positioning Resolution", Proc. 12th Int. Symp. on Ind. Robots, Paris, June 1982, p231.

Uicker, J.J.

"On the Dynamic Analysis of Spatial Linkages Using 4x4 Matrices", PhD Thesis, Dept. Mech. Eng. and Astronautical Sciences, Northwestern University, 1965.

Vaha, P., Halme, A.

"Adaptive Digital Control for a Heavy Manipulator", Internal Communication, University of Oulu, 1983.

Velikovich V.B., and Kritsku, D.R.

"Modular System of Design Calculations for Industrial Robots", Machines and Tooling, Vol. 49, No. 7, 1978, p21.

Vukobratovic, M., Potkonjak, V., Hristic, D.
"Dynamic Method for the Evaluation and Choice of Industrial Manipulators", Proc. 9th Int. Symp. on Ind. Robots, Washington DC, March 1979, p549.

Vukobratovic, M, Potkonjak, V., Hristic, D.
"Contribution to the Computer-Aided Design of Industrial Manipulators", 11th Int. Symp. on Ind. Robots, Tokyo Japan, August 1981, p545.

Warnecke, H.J., Schraft, R.D.
"Industrial Robots; Application Experience", German: Kranskopf-Verlag G.m.b.H., 1979, English: I.F.S. Publications Ltd., 1982.

Whitney, D.E.
"Resolved Motion Rate Control of Manipulators and Human Prostheses", IEEE Trans. Man-Machine Systems, 1969, p47.

Whitney, D.E.
"The Mathematics of Co-ordinate Control of Prosthetic Arms and Manipulators", Trans. ASME, J. Dyn. Syst. Meas. and Control, pp303-309, Dec. 1972.

Yang, A.T., Frendenstein, F.
"Application of Dual-Number Quarternion Algebra to the Analysis of Spatial Mechanisms", Trans. ASME, J. Appl. Mech., June 1964.

Zabala, J.
"Control of Robot Manipulators Based on Dynamic Modelling", Thesis, Toulouse, July 1978.

APPENDIX A HARDWARE SPECIFICATION OF UNITS USED IN THE ROBOT CONTROLLER

A.1 Sage II Specification

The general specification of the Sage II unit, used as the system supervisor, and initially for transformation equations, is as follows:

MC68000 16-bit processor 2 million instructions per second

Multi-colour status LED

Sage expansion bus: 16-bit data bus, 24-bid address bus

All input and output is interrupt driven, optionally polled

128K to 512K byte dynamic memory

Byte level parity checking

Real-time clock

Task scheduler

Two RS232-C serial ports

Parallel printer port

IEEE-488 GPIB port

Easy to interface BIOS

DEBUGGER for software development

Choice of 48TPI or 96TPI floppy disc drives

Low power requirements (70 watts)

Switching power supply

UCSD p-System Software with Pascal, FORTRAN and BASIC

A.2 Cifer 6800 Auxiliary Processor Board

The hardware specification for the Cifer 68000 board intended for the trajectory generation and transformation software modules is as follows:

MC68000 Processor running at 8MHa

256KB RAM with no wait states

Up to 32KByte EPROM

Memory mapping of RAM, providing 4 maps each of 64x4K Byte segments. Each segment may be designated as INACCESSABLE, READ ONLY, or READ-WRITE SUPERVISOR and USER modes

IEEE4888 interface using TMS9914A

RS232C port running at 9600 Baud with Handshakes and clock output

50 Hz timer interrupt

External ABORT button generating non-maskable interrupt

Hardware reset at power-up or by IEEE488 IFC line or by connection of an external button

Status by means of on board LED's

The processor directly supports 16MBytes of logical address space, and all memory and I/O devices are mapped into this space. The memory mapping is only applicable to the RAM with the EPROM and I/O devices appearing at fixed addresses.

A.3 Axis Controller Boards

The robot axis controller cards are single-board microcomputer units based on the MC68B09 microprocessor circuit. A large number of peripheral integrated circuits provide the board with a wide range of capabilities. If all the available facilities are not required, then the board may be operated in a partially-populated state. In its complete form, each axis controller card comprises:

MC68B09 microprocessor
6 Timer/Counters
2K bytes RAM
8K bytes ROM/EPROM
12-bit digital to analogue (D/A) converter
10.24V voltage reference
Analogue to digital (A/D) converter
CMOS LSI interpolation pulse generator
8 digital inputs
8 digital outputs
Voltage-controlled oscillator
TMS 9914 GPIB Controller and interface drivers

The above auxiliary circuits and additional on-board logic circuitry permit each axis controller card to be interfaced to the IEEE-488 instrumentation bus, an incremental optical encoder, an axis position potentiometer, forward and reverse limit switches, initialisation/ data-tunning switches, motor brake control circuit, a dc servo controller amplifier, and various input/output and diagnostic signals.

The Vector Profiling unit initially used for velocity profiling is a partially populated version of an axis controller card.

A.4 Input/Output Cards

A series of digital I/O lines under software control are fed to the external input/ouput cards within the robot controller rack. Digital outputs are first buffered and then fed through an opto-isolator to the output stage which may be either a Darlington transistor or Triac. Several outpu options are normally necessary, e.g. +24V dc at 1A drive current or 110V ac at 5A drive current capability. The status of any output line may be read when required. The inputs form external devices are conditioned and reduced to an acceptable level before being fed through an opto-isolator and into a digital buffer/line driver. Both inputs and outputs are available on 6U height double Eurocards in blocks of 16 circuits. Only one type of circuit is available on each card. For heavy current relays or contractors, which must be fitted to a DIN rail situated below the rack of servo controller amplifiers in the controller cabinet.

APPENDIX B DERIVATION OF TRANSFORMATION EQUATIONS

This Appendix contains the detailed calculations involved to determine the equations for the transformation of orientation from a secondary frame to base system and inverse kinematic analysis.

B.1 Orientation Transformation from a Frame to a Base System

The method devised for determining the transformation of orientation from a frame to base system uses an imaginary extension of the robot tool. The extension is of three links of unit length set a right angles to each other. This extension is set according to the orientation referenced with the secondary frame ${}^{f}\alpha$, ${}^{f}\beta$, ${}^{f}\gamma$. Four points are then specified corresponding to the common point and the ends of the extension links:

$$ {}^{f}\underline{x}_0 = [x_0, y_0, z_0]^T \qquad (B.1) $$

$$ {}^{f}\underline{x}_1 = [x_1, y_1, z_1]^T \qquad (B.2) $$

$$ {}^{f}\underline{x}_2 = [x_2, y_2, z_2]^T \qquad (B.3) $$

$$ {}^{f}\underline{x}_3 = [x_3, y_3, z_3]^T \qquad (B.4) $$

The relative positions of each vector can then be specified by means of ${}^{f}\alpha$, ${}^{f}\beta$, and ${}^{f}\gamma$ (see Figure B.1), as follows:

$$ {}^{f}\underline{x}_1 = {}^{f}\underline{x}_0 + \delta^{f}\underline{x}_1 \tag{B.5} $$

$$ {}^{f}\underline{x}_2 = {}^{f}\underline{x}_0 + \delta^{f}\underline{x}_2 \tag{B.6} $$

$$ {}^{f}\underline{x}_3 = {}^{f}\underline{x}_0 + \delta^{f}\underline{x}_3 \tag{B.7} $$

where from geometric inspection:

$$ \delta^{f}\underline{x}_1 = {}^{f}\begin{bmatrix} C_\beta\, C_\alpha \\ C_\beta\, S_\alpha \\ -S_\beta \end{bmatrix} \tag{B.8} $$

$$ \delta^{f}\underline{x}_2 = {}^{f}\begin{bmatrix} C_\gamma\, S_\beta\, C_\alpha + S_\gamma\, S_\alpha \\ C_\gamma\, S_\beta\, S_\alpha - S_\gamma\, C_\alpha \\ C_\gamma\, C_\beta \end{bmatrix} \tag{B.9} $$

$$ \delta^{f}\underline{x}_3 = {}^{f}\begin{bmatrix} S_\gamma\, S_\beta\, C_\alpha - C_\gamma\, S_\alpha \\ S_\gamma\, S_\beta\, S_\alpha + C_\gamma\, C_\alpha \\ S_\gamma\, C_\beta \end{bmatrix} \tag{B.10} $$

Where S and C refer to the sine and cosine of the subscripts respectively and the superscript f indicates that all orientation parameters are specified with respect to the secondary frame.

The four points specified, can be transformed to the base co-ordinate system by means of the frame rotation matrix ${}^{b}R_f$:

$$ {}^{b}\underline{x}_0 = {}^{b}R_f\ {}^{f}\underline{x}_0 \tag{B.11} $$

$$ {}^{b}\underline{x}_1 = {}^{b}R_f \, {}^{f}\underline{x}_1 = {}^{b}R_f \, {}^{f}\underline{x}_0 + {}^{b}R_f \, \delta^{f}\underline{x}_1 \qquad (B.12) $$

$$ {}^{b}\underline{x}_2 = {}^{b}R_f \, {}^{f}\underline{x}_2 = {}^{b}R_f \, {}^{f}\underline{x}_0 + {}^{b}R_f \, \delta^{f}\underline{x}_2 \qquad (B.13) $$

$$ {}^{b}\underline{x}_3 = {}^{b}R_f \, {}^{f}\underline{x}_3 = {}^{b}R_f \, {}^{f}\underline{x}_0 + {}^{b}R_f \, \delta^{f}\underline{x}_3 \qquad (B.14) $$

The relative positions of the vectors in base space is determined by executing in effect the reverse of the procedure outlined above. The relative positions of the vectors in base space is determined by:

$$ \delta^{b}\underline{x}_1 = {}^{b}\underline{x}_1 - {}^{b}\underline{x}_0 = {}^{b}R_f \delta^{f}\underline{x}_1 \qquad (B.15) $$

$$ \delta^{b}\underline{x}_2 = {}^{b}\underline{x}_2 - {}^{b}\underline{x}_0 = {}^{b}R_f \delta^{f}\underline{x}_2 \qquad (B.16) $$

$$ \delta^{b}\underline{x}_3 = {}^{b}\underline{x}_3 - {}^{b}\underline{x}_0 = {}^{b}R_f \delta^{f}\underline{x}_3 \qquad (B.17) $$

Since absolute lengths are preserved on rotational transformations, the values of ${}^{b}\alpha$, ${}^{b}\beta$, and ${}^{b}\gamma$ can be determined from ${}^{b}\delta\underline{x}_1$ ${}^{b}\delta\underline{x}_2$ ${}^{b}\delta\underline{x}_3$.

Referring to Figure B.2:

$$ {}^{b}\alpha = \text{atan2}[\delta^{b}y_1, \delta^{b}x_1] \qquad (B.18) $$

$$ {}^{b}\beta = \text{atan2}[\delta^{b}z_1, l] \qquad (B.19) $$

where the value of l can be specified by means of any of the following:

(i) $$ l = ((\delta^{b}x_1)^2 + (\delta^{b}y_1)^2)^{\frac{1}{2}} \qquad (B.20) $$

(ii) $\quad 1 = {}^{\delta b}x_1 \, Cb_a + {}^{\delta b}y_1 \, Sb_\alpha$ (B.21)

(iii) $\quad 1 = {}^{\delta b}x_1 / Cb_\beta$ (B.22)

(iv) $\quad 1 = {}^{\delta b}y_1 / Sb_\alpha$ (B.23)

Referring again to Figure B.2 the value of ${}^{b}\gamma$ is obtained by considering the z components of ${}^{\delta b}\underline{x}_2$ and ${}^{\delta b}\underline{x}_3$:

$${}^{b}\gamma = \text{atan2}({}^{\delta b}z_3/Cb_\beta, {}^{\delta b}z_2/Cb_\beta) \quad \text{(B.24)}$$

$$\rightarrow \quad {}^{b}\gamma = \text{atan2}({}^{\delta b}z_3, {}^{\delta b}z_2) \quad \text{(B.25)}$$

In terms of the elements of ${}^{b}R_f$ and the orientation parameters specified with respect to the secondary frame, the relevant base parameters required to specify ${}^{b}\alpha$, ${}^{b}\beta$, ${}^{b}\gamma$ are as follows:

$${}^{\delta b}x_1 = {}^{f}[r_{11}C_\beta C_\alpha + r_{12}C_\beta S_\alpha - r_{13}S_\beta] \quad \text{(B.26)}$$

$${}^{\delta b}y_1 = {}^{f}[r_{21}C_\beta C_\alpha + r_{22}C_\beta S_\alpha - r_{23}S_\beta] \quad \text{(B.27)}$$

$${}^{\delta b}z_1 = {}^{f}[r_{31}C_\beta C_\alpha + r_{32}C_\beta S_\alpha - r_{33}S_\beta] \quad \text{(B.28)}$$

$${}^{\delta b}z_2 = {}^{f}[r_{31}(C_\gamma S_\beta C_\alpha + S_\gamma S_\alpha) + r_{32}(C_\gamma S_\beta S_\alpha + S_\gamma C_\alpha) + r_{33}C_\gamma C_\beta] \quad \text{(B.29)}$$

$${}^{\delta b}z_3 = {}^{f}[r_{31}(S_\gamma S_\beta C_\alpha + C_\gamma S_\alpha) + r_{32}(S_\gamma S_\beta S_\alpha + C_\gamma C_\alpha) + r_{33}S_\gamma C_\beta] \quad \text{(B.30)}$$

where r_{ij} is the ith row jth colum component of ${}^{b}R_f$.

B.2 Detailed Analysis of Inverse Kinematic Equations

The following sections deal with the geometric analysis of the manipulators to determine the direct inverse kinematic equations.

The first stage with five or six dof manipulators is to determine the co-axial position of the end effector $\underline{x}_c$, thence θ, the inferred base rotation. Referring to Figure B.3:

$$x_c = {}^b(x - rC_\beta C_\alpha) \tag{B.31}$$

$$y_c = {}^b(y - rC_\beta S_\alpha) \tag{B.32}$$

$$z_c = {}^b(z - rS_\beta) \tag{B.33}$$

where ${}^b\underline{x}$ refers to the base position of the end effector and the superscript reflects the fact that all parameters are defined with respect to base space. The value of θ is then determined as:

$$\theta = \text{atan2}(y_c, x_c) \tag{B.34}$$

This expression is valid for cylindrical polar, spherical polar, and anthropomorphic configurations. With cartesian robots θ is taken as zero. Given θ the wrist parameters for five and six dor manipulators can be determined.

B.2.1 Two dof Wrist Configuration

The specific cases of two dof wrist configurations can be determined by considering the general case of a pitch, roll wrist with a tool offset at some angle δ, (see Figure B.4). By using an imaginary right angled triange set from the central axis of the end effector, the general solution can be obtained. Refering to Figure B.4:

$$hS_\delta S_\phi = hS_\beta - hC_\delta S_\sigma \tag{B.35}$$

Squaring this expression:

$$h^2S^2_\delta S^2_\phi = h^2S^2_\beta + h^2C^2_\delta S^2_\sigma - 2h^2S_\beta C_\delta S_\sigma \tag{B.36}$$

Also:

$$h^2S^2_\delta S^2_\phi = h^2C^2_\beta + h^2C^2_\delta C^2_\sigma - 2h^2C_\delta C_\beta C_\sigma C_{(\alpha-\theta 1)} \tag{B.37}$$

Cancelling h^2 and adding gives:

$$\delta^2_\delta(S^2_\phi+C^2_\phi) = (S^2_\beta+C^2_\beta) + C^2_\delta(S^2_\sigma+C^2_\sigma) - 2C_\delta(S_\beta S_\sigma+C_\beta C_\sigma C_{(\alpha-\theta 1)})$$

$$\rightarrow \quad \delta^2_\delta = 1+C^2_\delta - 2C_\delta(S_\beta S_\sigma+C_\beta C_\sigma C_{(a-\theta 1)}) \tag{B.38}$$

$$\rightarrow \quad C_\beta C_\sigma C_{(\alpha-\theta 1)} + S_\beta S_\sigma - C_\delta = 0$$

Note that if $\delta=90^o$ then the equation solves for σ as:

$$\sigma = -\text{atan2}(C_{(\alpha-\theta 1)}, T_\beta) \tag{B.39}$$

To solve for equation (B.38), substitute half angle equivalents for S_σ and C_σ, where:

$$S_\sigma = \frac{2t}{1+t^2} \tag{B.40}$$

$$C_\sigma = \frac{1-t^2}{1+t^2} \tag{B.41}$$

Where:

$$t = \tan(\sigma/2) \tag{B.42}$$

The equation can then be expressed in the form

$$\sigma = 2\text{atan2}(S_\beta \pm (S^2{}_\beta + C^2{}_\beta C^2{}_{(\alpha-\theta 1)} - C^2{}_\delta)^{\frac{1}{2}}, C_\delta + C_\beta C_{(\alpha-\theta 1)})$$

$$\rightarrow \quad \sigma = 2\text{atan2}(S_\beta \pm (S^2{}_\delta - C^2{}_\beta S^2{}_{(\alpha-\theta 1)})^{\frac{1}{2}}, C_\delta + C_\beta C_{(\alpha-\theta 1)}) \tag{B.43}$$

Note that if $\delta=0$ then the equation gives the relationship:

$$\sigma = \beta \tag{B.44}$$

To determine θ_5 refer again to Figure B.4:

$$l_5 = hS_\beta \tag{B.45}$$

$$o_5 = hC_\beta S_{(\alpha-\theta 1)} \tag{B.46}$$

The value of h_5 is given by:

$$\begin{aligned} h_5 &= l_5 - hC_\beta C_{(\alpha-\theta 1)} T_\sigma \\ &= h(S_\beta - C_\beta C_{(\alpha-\theta 1)} T_\sigma) \end{aligned} \tag{B.47}$$

The value of a_5 is given by:

$$\begin{aligned} a_5 &= h_5 C_\sigma \\ &= hC_\sigma (S_\beta - C_\beta C_{(\alpha-\theta 1)} T_\sigma) \end{aligned} \tag{B.48}$$

θ_5 can then be determined :

$$\begin{aligned} \theta_5 &= \text{atan2}(o_5, a_5) \\ &= \text{atan2}(C_\beta S_{(\alpha-\theta 1)}, C_\sigma (S_\beta - C_\beta C_{(\alpha-\theta 1)} T_\sigma) \\ &= \text{atan2}(S_{(\alpha-\theta 1)}, T_\beta C_\sigma - C_{(\alpha-\theta 1)} S_\sigma \end{aligned} \tag{B.49}$$

Note that if $\delta=90^o$ and σ is re-specified then Equation (B.49) simplifies to:

$$\theta_5 = \text{atan2}(S_{(\alpha-\theta 1)} C_\sigma, C_{(\alpha-\theta 1)}) \tag{B.50}$$

The additional parameter required to solve for the primary axes configuration, is the wrist position $\underline{x}_\omega$, where:

$$x_\omega = x_c - l_\omega C_\sigma C_{(\alpha-\theta 1)} \tag{B.51}$$

$$y_\omega = y_c - l_\omega C_\sigma C_{(\alpha-\theta 1)} \tag{B.52}$$

$$z_\omega = z_c - l_\omega S_\sigma \tag{B.53}$$

Note that the above equations also hold true when specify the inverse kinematic equation set for a five dof arm exhibiting a pure pitch roll secondary axis configuration. This configuration is usually used when the end effector is a gripper.

B.2.2 Pitch Yaw Roll (PYR) Three dof Wrist

The solution of a PYR wrist is an extension of the solution for the two dof wrist, with $\delta=90^o$. Thus:

$$\sigma = -\text{atan2}(C_{(\alpha-\theta 1)}, T_\beta \tag{B.54}$$

$$\theta_5 = \text{atan}(S_{(\alpha-\theta 1)} C_\sigma, C_{(\alpha-\theta 1)}) \tag{B.55}$$

The value of θ_6 is found by determining the angle required to bring roll axis of the orientation back to the vertical. The actual angular offset is then added to this compensatory angle. Referring to Figure B.5, the compensatory angle θ_6' is determined as follows:

Considering the imaginary cylinder extending from the σ plane, then ν_6, the offset from the cylinder axis and the vertical, is given by:

$$\nu_6 = \nu S_\sigma \tag{B.56}$$

Thus:

$$o_6 = \nu_6 S_{\theta 5} = \nu S_\sigma S_{\theta 5} \tag{B.57}$$

The cylinder height, a_6:

$$a_6 = \nu C_\sigma \tag{B.58}$$

Thus the value of θ_6' is given by:

$$\theta_6' = \text{atan2}(S_\sigma S_{\theta 5}, C_\sigma) \tag{B.59}$$

Thus the total angle θ_6 is given by:

$$\theta_6 = \theta_6' + \gamma \tag{B.60}$$

The wrist point is defined as before, and the value of θ_4 is determined with respect to the primary axes configuration.

B.2.3 Roll Pitch Roll (RPR) Three dof Wrist Configuration

The RPR wrist configuration is illustrated in Figure B.6. With this configuration the value of ω is determined from the primary axes configuration. The value of θ_1 is determined from $\underline{x}_c$ as detailed previously. Refering to Figure B.6, and taking l=1 for clarity:

$$o_\beta = S_\beta \tag{B.61}$$

$$a_\beta = C_\beta \tag{B.62}$$

$$a_\omega = a_\beta C_{(a-\theta 1)} = C_\beta C_{(\alpha-\theta 1)} \tag{B.63}$$

$$o_\omega = a_\omega T_\omega = C_\beta C_{(\alpha-\theta 1)} T_\omega \tag{B.64}$$

Thus the value of ν is defined by:

$$\nu = o_\beta \text{-} o_\omega = S_\beta \text{-} C_\beta C_{(\alpha \text{-} \theta 1)} T_\omega \tag{B.65}$$

From ν, o_4 can be defined:

$$o_4 = \nu C_\omega = C_\omega S_\beta \text{-} C_\beta C_{(\alpha \text{-} \theta 1)} S_\omega \tag{B.66}$$

Also:

$$a_4 = a_\beta S_{(\alpha \text{-} \theta 1)} = C_\beta S_{(\alpha \text{-} \theta 1)} \tag{B.67}$$

Thus θ_4 can be defined by:

$$\theta_4 = \text{atan2}(o_4, a_4) \tag{B.68}$$

$$\theta_4 = \text{atan2}(C_\omega S_\beta \text{-} C_\beta C_{(\alpha \text{-} \theta 1)} S_\omega, C_\beta S_{(\alpha \text{-} \theta 1)}) \tag{B.69}$$

θ_5 is determined as follows:

$$e_4 = \nu S_\omega = S_\beta S_\omega \text{-} C_\beta C_{(\alpha \text{-} \theta 1)} T_\omega S_\omega \tag{B.70}$$

$$h_\omega = a_\omega / C_\omega = C_\beta C_{(\alpha \text{-} \theta 1)} / C_\omega \tag{B.71}$$

(The value of h_σ may appear ill-conditioned as $\omega \rightarrow 90^\circ$ however the physical limitations of ω, dependent on the primary axis, means that ω will usually be less than 80°).

Thus a_5 is determined by:

$$a_5 = e_4+h_\omega \tag{B.72}$$

$$a_5 = S_\beta S_\omega - C_\beta C(\alpha-\theta 1) T_\omega S_\omega + C_\beta C(\alpha-\theta 1)/C_\omega \tag{B.73}$$

Which on rearranging terms gives:

$$a_5 = S_\beta S_\omega + C_\beta C(\alpha-\theta 1) C_\omega \tag{B.74}$$

The value of θ_5 can be determined in four ways:

$$\text{(i)} \qquad o_5 = (a_4^2+o_4^2)^{\frac{1}{2}} \tag{B.75}$$

$$\text{(ii)} \qquad o_5 = a_4 C_{\theta 4} + o_4 S_{\theta 4} \tag{B.76}$$

$$\text{(iii)} \qquad o_5 = a_4/C_{\theta 4} \tag{B.77}$$

$$\text{(iv)} \qquad o_5 = o_4/S_{\theta 4} \tag{B.78}$$

Thus θ_5 is given by:

$$\theta_5 = \text{atan2}(o_5, a_5) \tag{B.79}$$

Using for example the definition of o_5, (ii):

$$\theta_5 = \text{atan}(C_\beta S_{(\alpha-\theta 1)}, C_{\theta 4}(S_\beta S_\omega + C_\beta C(\alpha-\theta 1) C_\omega) \tag{B.80}$$

The value of θ_6 is determined by using an imaginary cylinder rotated twice from the vertical, as illustrated in Figure B.7. Given the vertical distance between the cylinder planes, ν, the cylinder height h:

$$h = \nu C_\omega C_{\theta 4} \tag{B.81}$$

Also:

$$m_1 = \nu S_\omega \tag{B.82}$$

$$m_2 = \nu C_\omega S_{\theta 4} \tag{B.83}$$

The value of o_6 is given by:

$$o_6 = m_1 S_{\theta 5} - m_2 C_{\theta 5} \tag{B.84}$$

$$o_6 = \nu S_\omega S_{\theta 5} - \nu C_\omega S_{\theta 4} C_{\theta 5} \tag{B.85}$$

The value of θ_6' the angle required to bring the roll orientation back to the vertical given by:

$$\theta_6' = \text{atan2}(o_6, h) \tag{B.86}$$

$$\theta_6' = \text{atan}(S_\omega S_{\theta 5} - C_\omega S_{\theta 4} S_{\theta 5}, C_\omega C_{\theta 4}) \tag{B.87}$$

Thus the total angle θ_6 is given by:

$$\theta_6 = \theta_6' + \Upsilon \tag{B.88}$$

B.2.4 Primary Axes Configurations

The four configurations most commonly found for the primary axes of an individual robot are, (see Figure B.8):

(i) Cartesian

(ii) Cylindrical Polar

(iii) Spherical Polar

(iv) Anthropomorphic

Given the wrist point $\underline{x}_c$, the primary axes parameters can be determined as follows:

For a cartesian robot:

$$X_r = x_c \tag{B.89}$$

$$Y_r = y_c \tag{B.90}$$

$$Z_r = z_c \tag{B.91}$$

For a cylindrical polar configuration:

$$R_c = (x_c^2+y_c^2)^{\frac{1}{2}} \tag{B.92}$$

$$\theta_c = \text{atan2}(y_c,x_c) \tag{B.93}$$

$$Z_c = z_c \tag{B.94}$$

For the spherical polar configuration:

$$z'_c = z_c - l_s \tag{B.95}$$

$$R_s = (x_c^2 + y_c^2 + z'_c{}^2)^{\frac{1}{2}} \tag{B.96}$$

$$\theta_s = \text{atan2}(y_c, x_c) \tag{B.97}$$

$$\psi_s = \text{atan2}(z_c, (y_c^2 + x_c^2)^{\frac{1}{2}}) \tag{B.98}$$

For the anthropomorphic configuration, the analytical technique is as follows, referring to Figure B.10:

$$a_\psi = (x_c^2 + y_c^2)^{\frac{1}{2}} \tag{B.99}$$

(Alternatively the value of a_ψ can be specified in terms of x_c, y_c and θ_1 as per Equations (3.40), (3.41), (3.42) (3.43).

$$o_\psi = z_c - l_1 \tag{B.100}$$

Thus angle ψ can be determined by:

$$\psi = \text{atan2}(o_\psi, a_\psi) \tag{B.101}$$

The values of h_ψ can be determined from ψ, o_ψ and a_ψ again as per Equations (3.40) to (3.43). Taking definition (i):

$$h_\psi = (o_\psi^2 + a_\psi^2)^{\frac{1}{2}} \qquad \text{(B.102)}$$

The values of θ_2' and θ_3' can then be determined from h_ψ, l_2, and l_3. However as with the Imperial College robot, l_2 and l_3 are often equal in length, simplifying the equations:

$$\theta_3' = 2\arcsin(h_\psi/2l_2) \qquad \text{(B.103)}$$

$$\theta_2' = 90-\theta_3'/2 \qquad \text{(B.104)}$$

Since h_ψ is limited in range due to the physical limitations of the robot, the equations are well conditioned. However the singular point of $h_\psi=2l_2$, signifying the extreme reach of the robot should be avoided if possible.

Thus from θ_2' and θ_3' the values θ_2 and θ_3 can determined:

$$\theta_2 = 90-(\theta_2+\psi) \qquad \text{(B.105)}$$

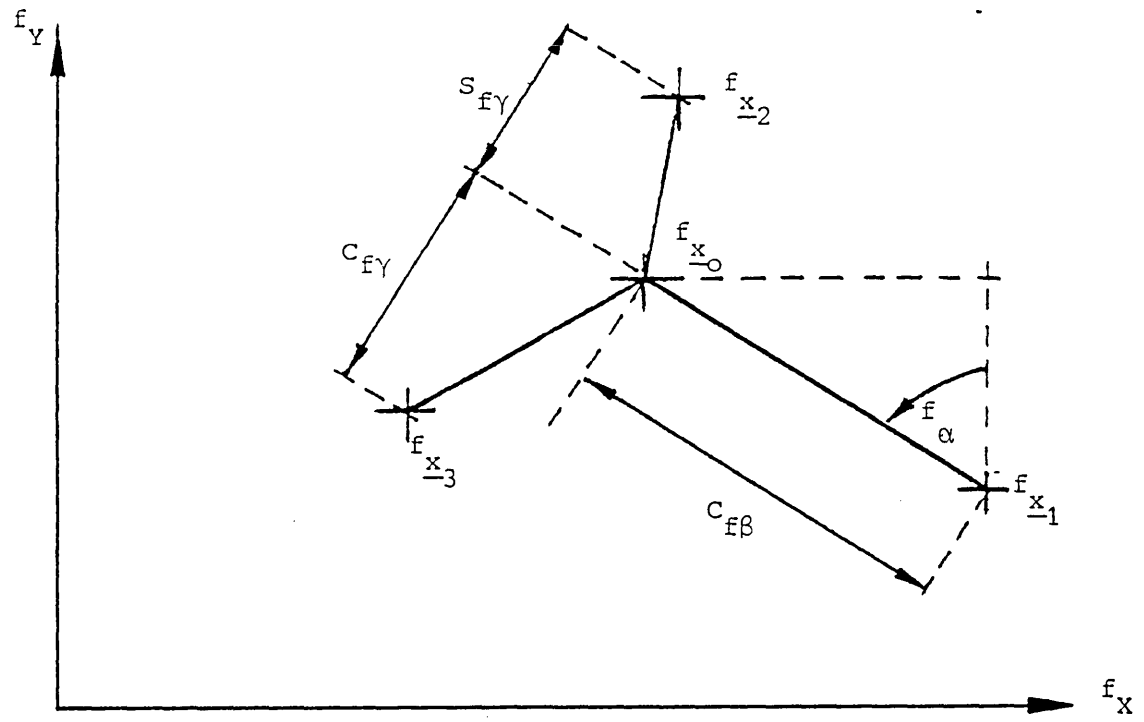
$$\theta_3 = 180-\theta_3' \qquad \text{(B.106)}$$

The value of ω passed to the RPR wrist configuration reflects the orientation of the end of the primary axes. Therefore; $\omega=o$ for the cartesian and cylindrical polar configurations, $\omega=\psi s$ for spherical polar, and $\omega=90-(\theta_2+\theta_3)$ for the anthropomorphic configurations. Given ω and σ from the primary and secondary axes respectively (when applicable), the value of θ_4 can be determined:

2 dof wrist : $\theta_4 = \sigma - \omega$ (B.107)

3 dof PYR wrist : $\theta_4 = \sigma - \omega$ (B.108)

3 dof RPR wrist : not applicable (B.109)



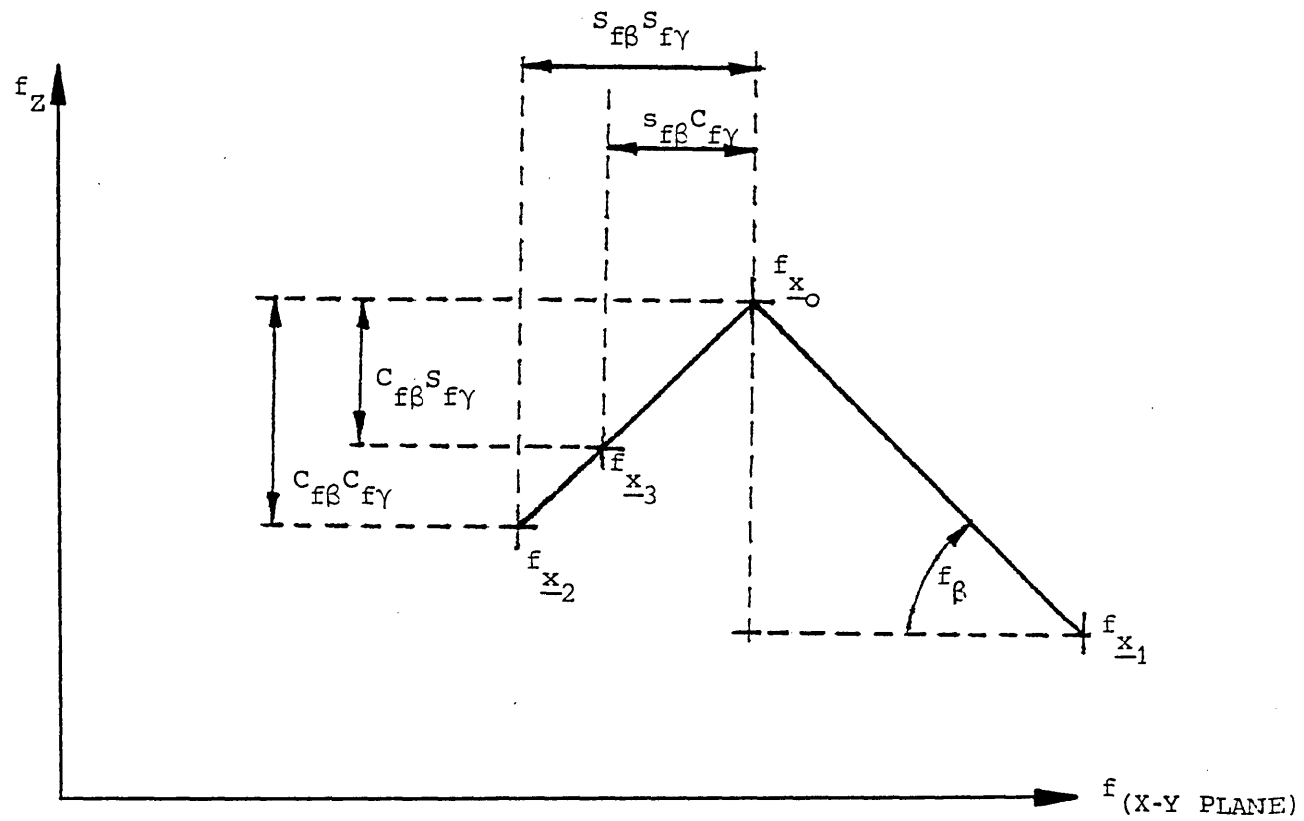


Figure B.1 Tool Extension Parameters in Secondary Frame

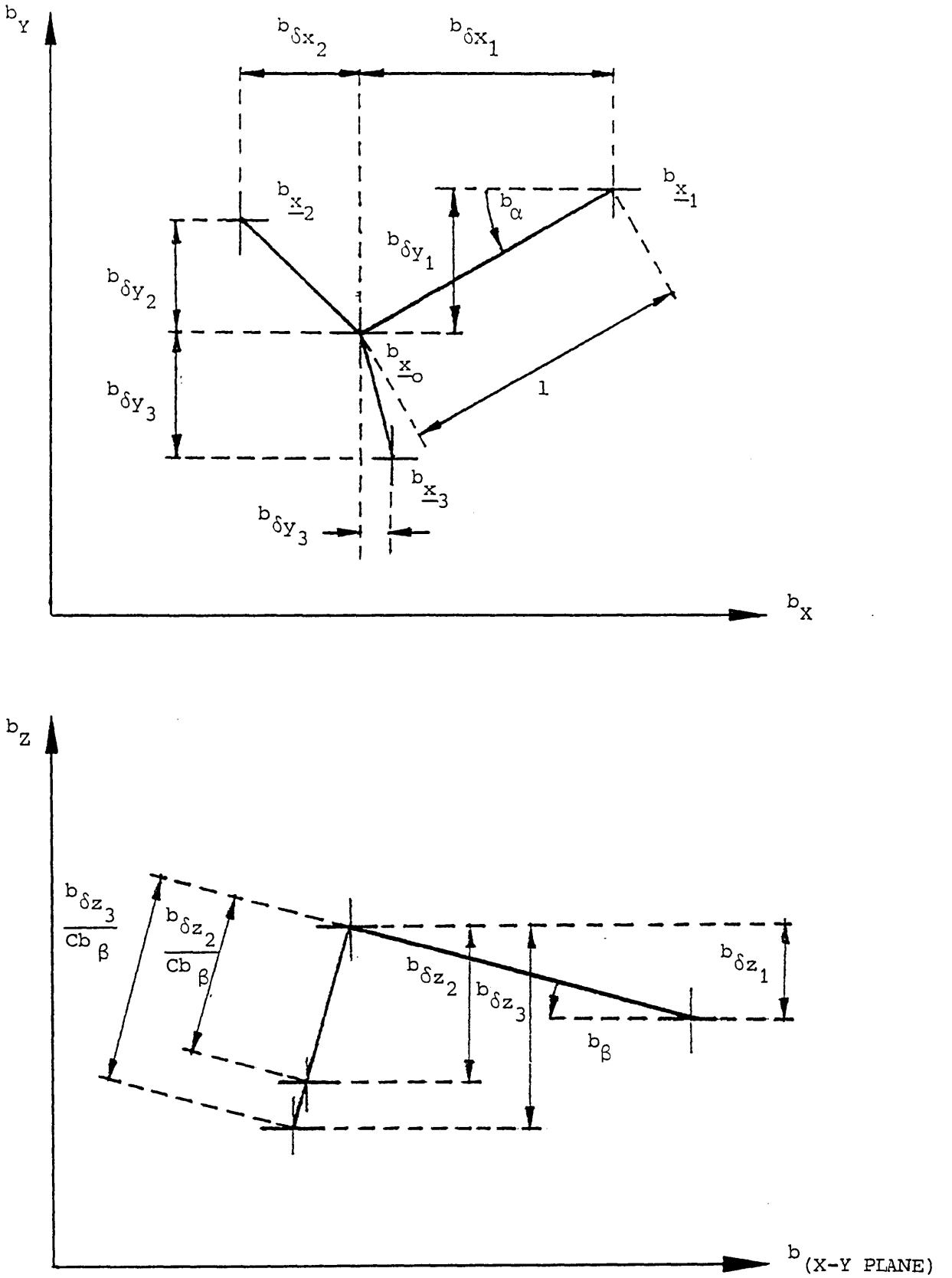


Figure B.2 Tool Extension Parameters in Base Frame

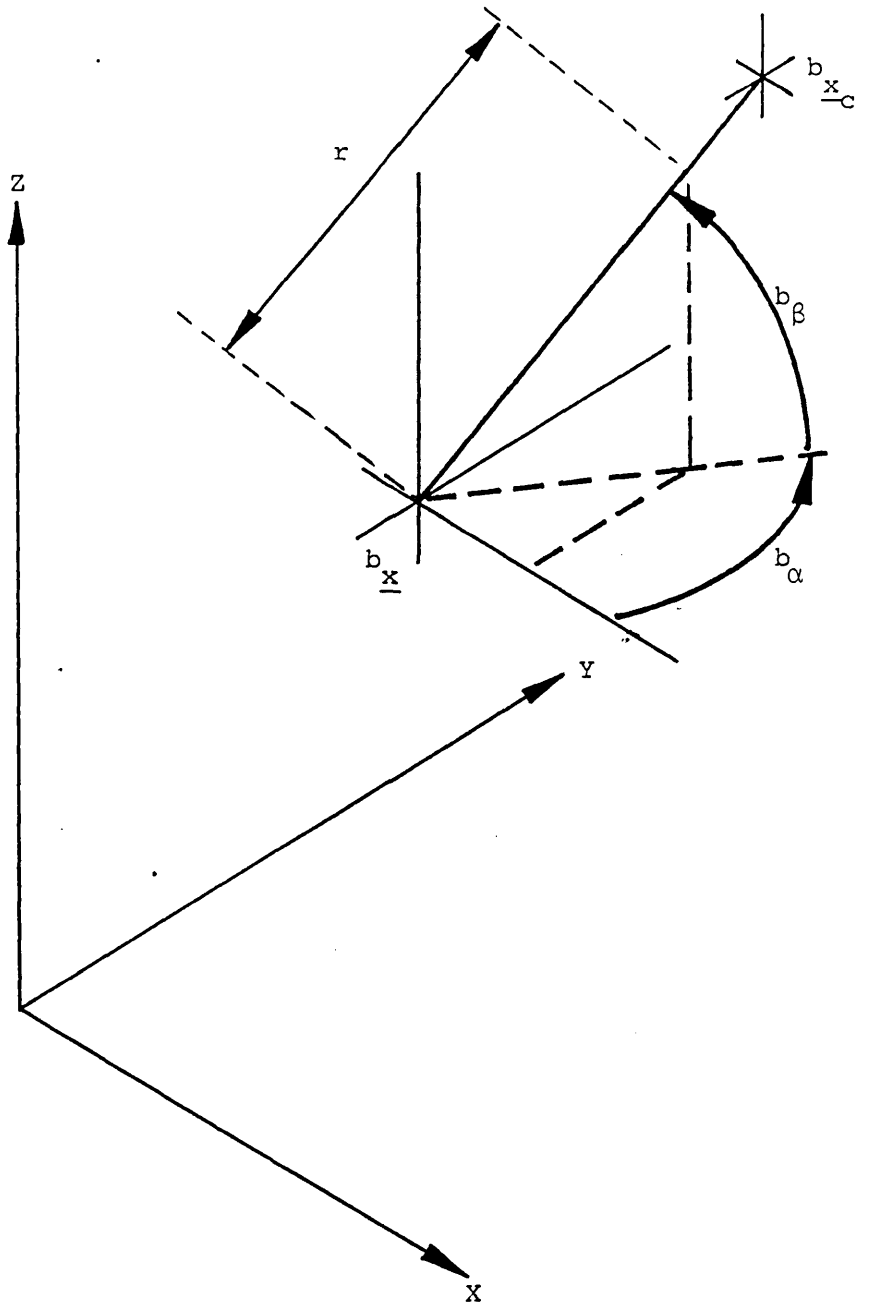
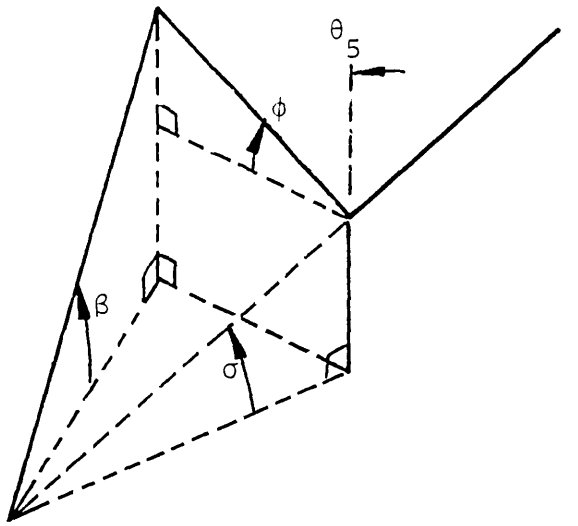
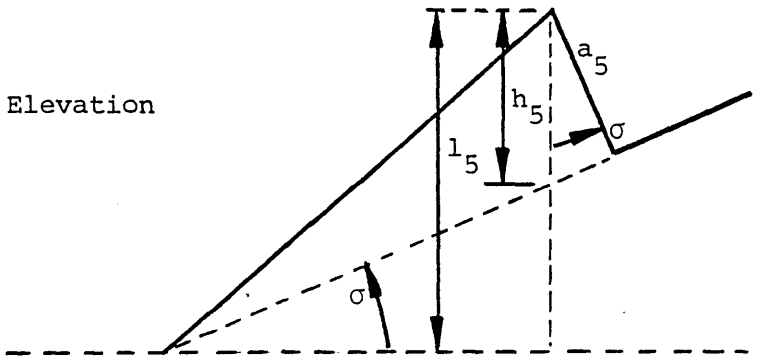


Figure B.3 Co-axial Position of End Effector





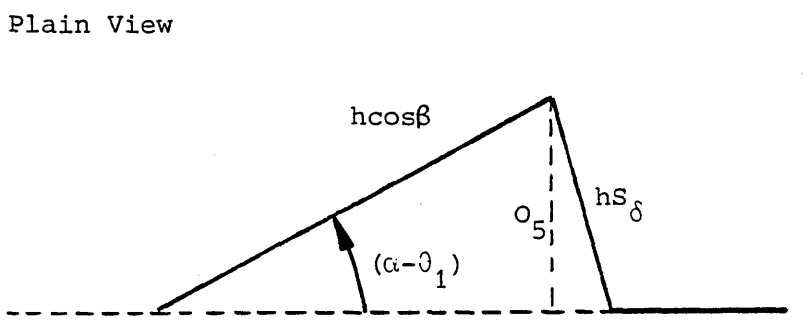
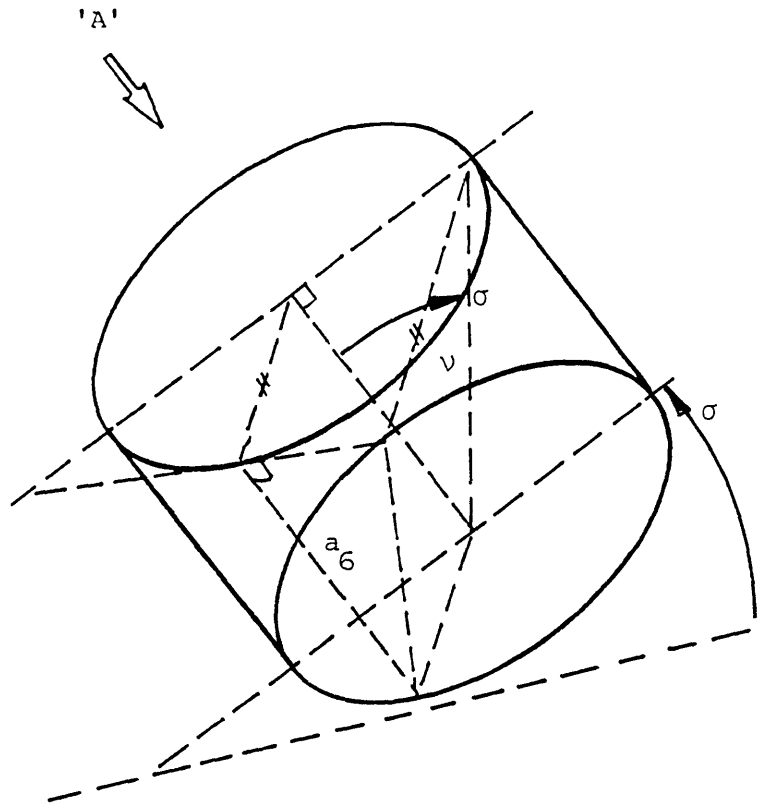


Figure B.4 Two dof Wrist Inverse Kinematic Evaluation



View on 'A'

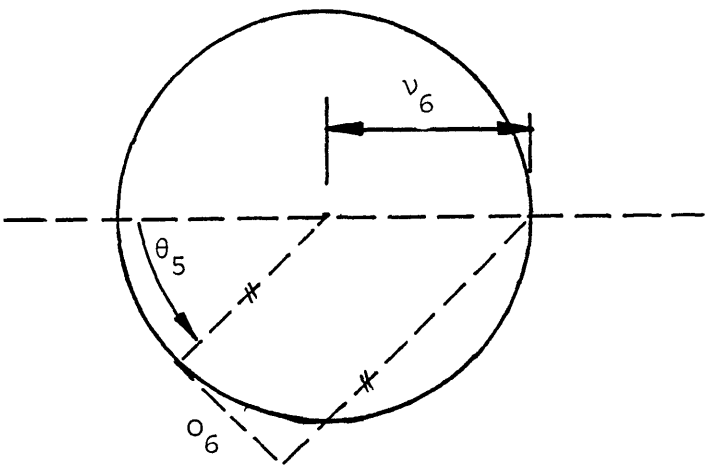


Figure B.5 Evaluation of θ_6 for PYR Wrist Configuration

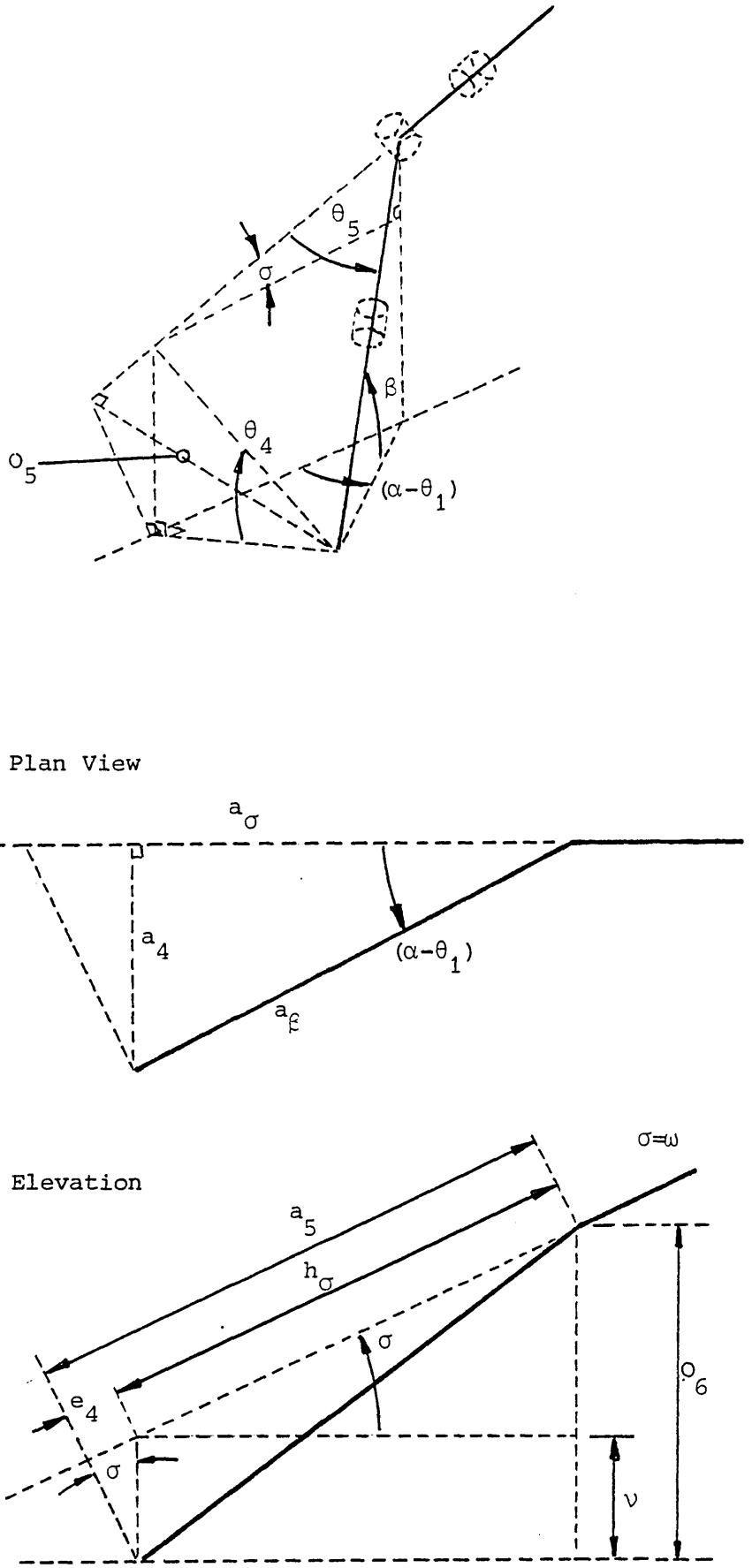


Figure B.6 Three dof RPR Wrist Kinematic Evaluation

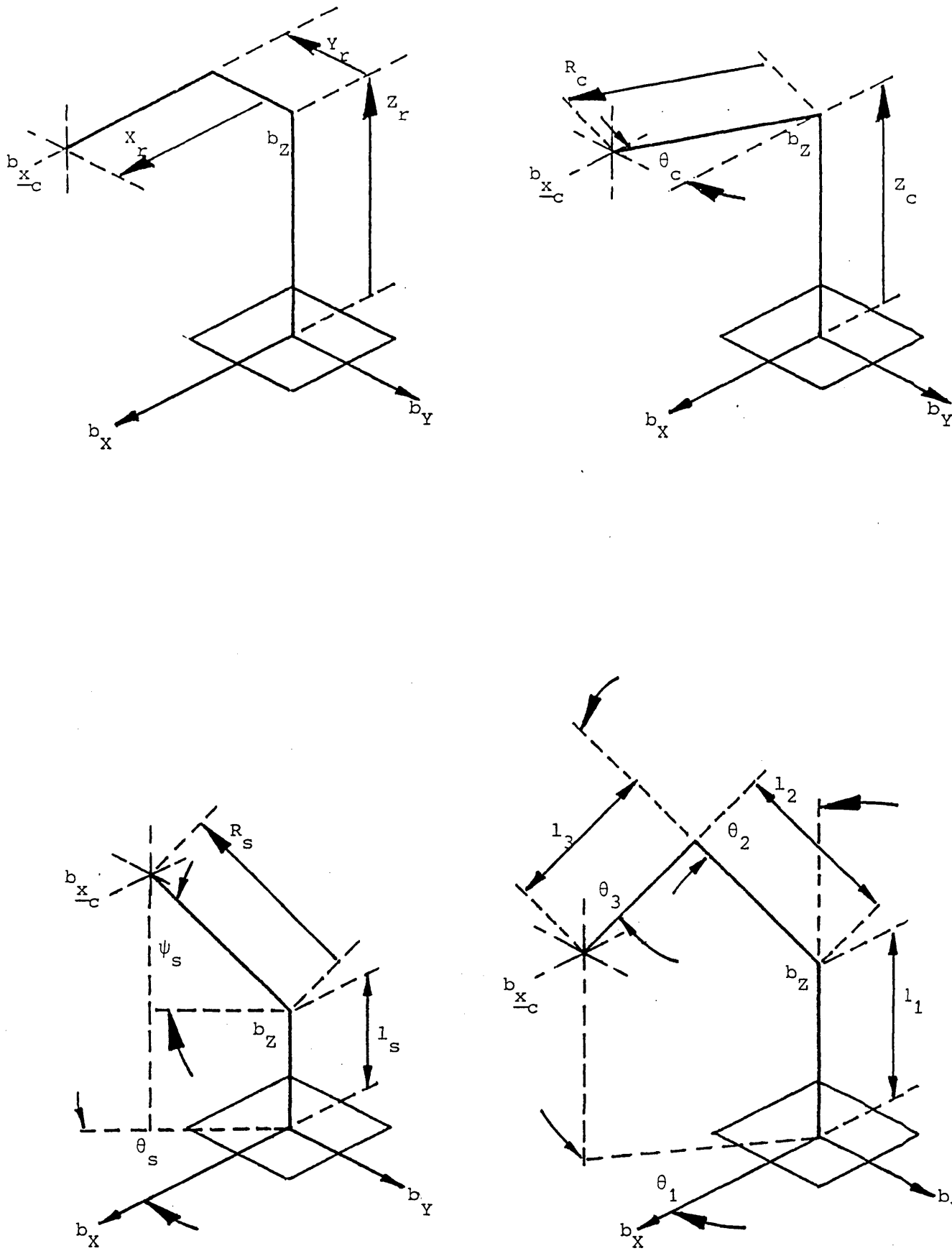
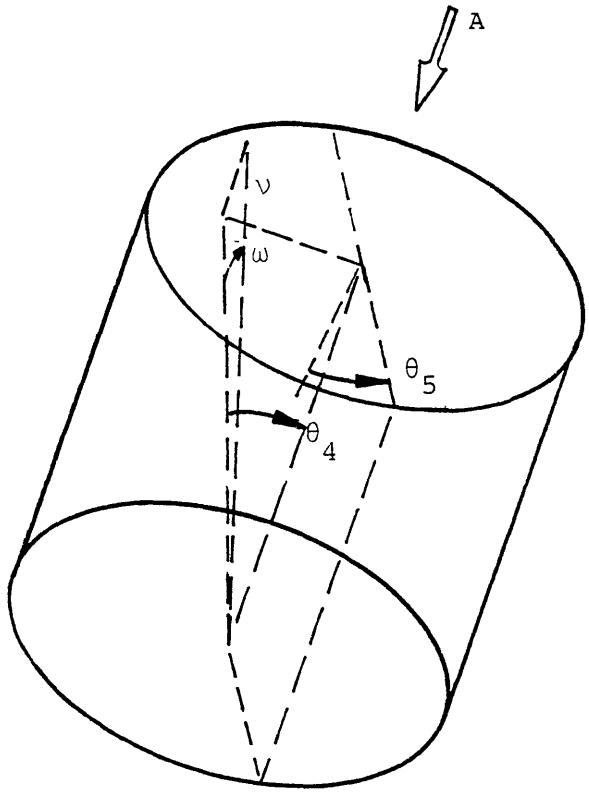


Figure B.8 Evaluation of Primary Axes Parameter



View on 'A'

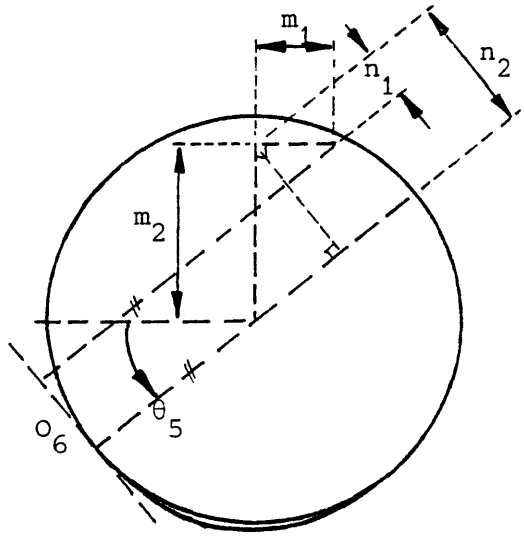


Figure B.7 Evaluation of θ_6 for RPR Wrist Configuration