PARAMETER IDENTIFICATION
AND
MODEL BASED CONTROL OF DIRECT DRIVE ROBOTS

By

SOO HYUN KIM

THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

OF THE UNIVERSITY OF LONDON

AND THE DIPLOMA OF IMPERIAL COLLEGE

JULY 1991

COMPUTER AIDED ENGINEERING SECTION

DEPARTMENT OF MECHANICAL ENGINEERING

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXHIBITION ROAD

LONDON SW7 2BX

ENGLAND

ABSTRACT

The direct drive arm (IC DDR) sustaining the attribute of a parallel linkage mechanism is constructed at Imperial College. The positioning arm is mounted horizontally as SCARA type configuration. The structure is well suited for light assembly work. The motion control problem of the IC DDR becomes more crucial than in a conventional geared robot due to substantial increase in the load inertias reflected on the motors and the increased dynamic interactions between the arm links.

The research of this thesis covers four main themes: modelling of the IC DDR dynamics in discrete-time base, developing parameter identification algorithms, control methods by dynamic compensation, and real-time implementation.

The complete discrete-time dynamic model, which is inherently suitable for computer-controlled robot system, is developed. A feature of this discrete model is that the second derivatives(accelerations) are not included in this model structure. Rather than a conventional notion relating the torque/force to dynamic responses, a numerical input value, which is directly assigned to the amplifier device to activate the corresponding motor, and the joint angular position/velocity as output are related through the dynamic model structure by incorporating the motor dynamic model.

The parameters governing the dynamic behaviour of the model are crucial factors for the model accuracy, and especially important in its usage for control purpose. To achieve an accurate dynamic model, five identifier structures for parameter estimation are introduced. The estimation performances associated with a noise, limited dynamic response information(i.e., when only positional information are available), excitation type, input frequency, etc. are analyzed according to each identifier structure.

In the section on control, two typical motions and two smooth polynomial trajectories are designed, and their combinations are used to test the dynamic compensation control scheme based on the discrete-time dynamic model. The decoupling performance of discrete dynamic model is compared with those of conventional discrete-time dynamic model in inverse dynamic sense. To improve the tracking performance, off-line

feedforward and real-time full dynamic compensation control schemes are evaluated and compared with the independent joint control scheme. Further, stimulated by the outstanding characteristics of the model based dynamic compensation scheme, velocity dependent discrete dynamic model is reconstructed into the velocity independent discrete dynamic model, and its performance is also evaluated.

Finally, the structure of IC DDR controller is outlined. The parameter identification algorithm is experimentally applied on IC DDR to estimate the best candidate of actual parameters. The control scheme based on dynamic compensation is implemented at a sampling period 10 [msec], and their performances are tested through experiments.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Prof. C. B. Besant, for his encouragement, sponsoring me over the past three years. His guidance was crucial in organising this research and making it successful. I am particularly grateful to Dr. M. Ristic for giving advices at important stage. Also I wish to thank Dr.Husan Wang at R.D.Projects Ltd. for providing the stimulus for much of the work and valuable discussions.

I would like to thank GEC-MARCONI and Dr. Bardo for providing the funds to carry out the collaborative research program with the Korea Institute of Technology. I must also mention The British Council and the former representative in Korea, Mr. David Rogers, for arranging a fellowship during my research.

There are many colleagues in the Computer Aided Engineering Section who have provided a stimulating atmosphere. The colleagues I particularly wish to acknowledge are Mr. John Choi and Dr. Sen who helped me sort out problems in the writing of this thesis.

On a personal note, I would like to express my profound thanks to my wife, Sun for her endless love and support. And I can not finish without special thanks to my Mother and parents in-law, especially Mr. Chang Soo, Kim for his immense support throughout the research.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Robots for Automation and Assembly

The term 'automation' is used to describe the introduction of machines into factories or offices to perform tasks formerly done by people. The new technological view of automation at present is the emergence of flexibility as opposed to fixed or hard automation.

The flexibility of automation is the important criteria which distinguishes current technology from the past. As the level of sophistication increases, even the current fixed automation can perform very complicated tasks. However the machinery is dedicated to a fixed number of tasks and can only be changed by physical rearrangement. The complexity of the machinery means that it is expensive and therefore requires a large production size to justify its cost. If the production size is in a small batch size, the fixed automation can not be economic. The ideal flexible automation could be able to cope with a single product as well as products of many thousands.

Individual machine tool at present possesses flexibility due to computer numerical control and this is being extended by computer assisted manufacturing. The effort is being concentrated in incorporating the whole factory facility into an overall flexible manufacturing system. Elsewhere in a factory and in a design office, automation with a high degree of flexibility is already being used by many firms in the form of computer aided design. The linking of computer aided design (CAD) and computer assisted manufacturing (CAM) together into what is called, CAD/CAM is currently an area of much interest. The final goal is to link all the computer assisted technologies which include CAD, CAM and computer aided process planning together with office automation into a global computer controlled system for the factory called computer integrated manufacture (CIM).

Assembly means joining a number of separate parts together into a composite whole. Assembly tasks have been estimated to consume 53 % of time and 22 % of cost in the manufacture of a product[Owen 85]. In industry at present, these assembly operation are either performed manually or by fixed automation. Manual labour is used if the product size does not justify automated machinery on economic grounds or if the assembly task is too complex to automate. To introduce flexibility into automated assembly requires the use of robots. Only in the late 70's as robotics technology became more advanced, it was possible to consider building a flexible assembly system. Such systems are still in the research phase despite of the advertised claims of some manufacturers.

It is a small to medium batch size where the greatest advances can be made by introducing a sophisticated automation. In the future, flexibility of assembly for large production volume can be used to manufacture many variants on the same model. Given the amount of time and cost involved in assembly, it is clear that any advances in automating the process could bring substantial rewards if successful. It is for this reason that a great deal of current research is being conducted in robotics.

Before going onto descriptions of robots, a definition of robot would be helpful : *A robot is a reprogrammable device designed to both manipulate and transport parts, tools or specialized manufacturing implements through variable programmed motions for the performance of specific manufacturing tasks.* The programmability of a robot is essential for it to be flexible and hence fundamental to the concept of robots as a part of flexible automation. Robots in industry today can be reprogrammed by many programming methods such as lead through, coordinate entry programming etc. However these require comparatively long time and therefore the robots are rarely reprogrammed. Robots will not become flexible in practical sense until off-line programming, where the program can be generated without involving the robot, has become popular.

It is generally accepted that robots produce more consistent quality, more predictable output and are more reliable than humans. They also have the advantage over fixed automation as they are flexible and can be used on a variety of assembly operations. However fixed automation is quicker than robotic assembly. For a large production volume, robots no longer provide the best solution. The problem may be alleviated by increasing the speed of the robots and by making the reprogramming quicker. But it is

unlikely that robots will replace fixed automation where no flexibility is required. The other problem of assembly robots is their inability to deal with complex tasks. This could be solved beforehand by constraining the product design at initial manageable stage.

The cost analysis and justification of robotic assembly has not yet been fully developed and this is necessary before industry is willing to invest in such a system.

Consideration must now be given to the specifications of robots which make them suitable for assembly work. They are usually characterized by three factors, payload capacity, repeatability and accuracy. In addition, the speed of the robot is important when applied to assembly operations. The repeatability and accuracy of robots determine the actual position of the arm during an assembly sequence. Clearly the repeatability must be high so that the robot can return to the desired position within a small deviation. The absolute accuracy is not crucial if the robot is programmed on-line because any errors in position can be corrected for at the programming stage. If off-line programming is used, the absolute accuracy needs to be high

The task time depends not only on the maximum speed of the robot but also on its maximum acceleration. This can be deduced because even if the top speed is high, a low acceleration capability means that the robot can only reach its top speed after a long motion.

1.2 Direct Drive Robot

The first successful installation of an industrial robot was made by Unimation Inc. in 1961. The mechanical construction of the first Unimate industrial robot was hydraulically driven manipulator arm. It allowed the manipulator to perform versatile motions and to access a large work space compared to the space occupied by itself. However, the positioning accuracy at the endpoint of the manipulator was relatively low.

A robot arm is a positioning system, and this is particularly true in automatic assembly where the components must be correctly aligned before the assembly operation. Whilst

velocity control is necessary for some type of assembly such as glueing and welding, positioning is still critical . In any case, the velocity requirements are very low when compared to a general velocity servo system.

Typical industrial robots have a small percentage payload capacity (approximately 3 %) of its own weight[Lane 84], and their maximum acceleration is currently at most 0.1 to 0.5 g(gravity)[Asada 87]. Those are due to several factors. Perhaps the most important is the heavy weight restriction of the links for the purpose of maintaining rigidity.

One of the major drawbacks of today's robots is that they tend to be slow in general. It is true that the speed and the accuracy are not necessarily the primary goal in the design of robots. Other characteristics of robot such as flexibility, dexterity, intelligence, etc. are other important issues in the design of robots. But the important facts in connection with the usefulness of robot on production line are speed and accuracy. The speed could be severely limited by the weight of manipulator arm. As was mentioned above, many existing robot arms are made heavy for rigidity, which is necessary to secure an accurate placements and repeatability. Most of the existing manipulator can not perform task with a repeated position less than the order of 0.1mm. However many tasks, such as precision assembly of small electrical components, need repeated position tolerance on the order of 0.05mm. So this relatively poor endpoint accuracy of many current robots have restricted their applications to tasks that require small error tolerance. The automation of assembly tasks could be greatly enhanced if robots can operate at higher speeds with greater positional accuracy. This goals can not be achieved with the existing massive robot designs because of their comparatively slow motion due to its heavy weight.

For light-duty applications, electrically powered robots become the most prominent robot design. Virtually, all electrically driven arms include some form of mechanism to transmit the torque developed by the motors to the load. These transmission mechanisms are usually gear trains, ball screw, belts, chains and linkages. It is undesirable to locate the motors at the joint because their masses are added to the inertia of previous joint. If the joint is distant from the base, this effect can be much more increased. In addition, sometimes it is difficult to physically place a motor at the joint. In such case that the motor is not located at the joint, it is obvious that a mechanism must be used to transmit the torque to the corresponding joint. Similarly, some robot

configurations require prismatic motion at the joint. Hence a mechanism such as ball screw is needed to transform the revolute motion of the motor into the desired linear motion.

In spite of the needs above, the primary reason for using gears is to reduce the speed by an order of 1:100 because standard electric motors have been designed to produce their maximum power(but small torque) at several thousands rpm. As mentioned before, a robot is essentially a positioning system with speeds of less than half a revolution per second. So the requirement for low rotational speed and high torque has led to the use of a mechanical reducer between the motor and the load.

The mechanical reducer is required to provide a high reduction gear ratio while maintaining high precision. It is particularly important that the reducer should not introduce backlash and lost motion which are directly linked to the degradation of positioning accuracy. Even a small amount of backlash at proximal joint leads to a significantly large error at the distant arm tip. A robot arm with multiple degrees of freedom is more sensitive to reducer characteristics since several points of backlash may interact producing inaccuracy and possibly instability. But precise anti-backlash gears or other type of reducer with large reduction gear ratio inevitably introduce a considerably large friction. In the manipulators which have a significant gearing, the torque loss due to friction can be as high as 25% of the torque required to move the manipulator in typical situations[Craig 86]. A large friction leads to poor accuracy, repeatability and difficulties in control because friction is highly unpredictable and so difficult to be identified and also to be compensated for.

The alternative to all these problems is to design a manipulator of which the arm links are directly coupled to the motors so that the transmission mechanisms can be removed, namely direct drive robot. Recent developments in the use of rare-earth magnets in motors have improved the performance and have made this option possible. Even though it is anticipated that the control of a direct drive robot could be more complicated than that of a conventional geared robots, this is the problem of a computer software and control electronics. This contrasts with the mechanical problems of complicated precision gears and anti-backlash devices which are commonly used at present. Although friction has not been totally eliminated, it is greatly reduced and backlash is no longer a problem in the direct drive mechanism. An additional advantage of using direct drive technique is in the application of force

control. In many mechanical assembly operations, it is desirable to measure the force exerted at the arm tip which may be used for active compliance control.

Although the new rare-earth motors do produce considerably more torque at lower speeds than conventional motors, they have limits. So, sufficiently powerful motor must be specified at the design stage. An alternative for increasing the power of motor is to reduce the load on them. This leads into the consideration of attempting to minimize the inertia of the arm because there is no mechanism for reducing the load inertias reflected on the motors unlike geared arms where the inertias are reduced by the square of gear ratio. Composite material can be used as the new arm materials. As an instance, KIT(Korea Institute of Technology) designed the arm with composite material of high modulus graphite fiber. The arm weight can be reduced to more than 50 % whilst the static stiffness and first resonance frequency has been increased 50 % and 70 % respectively compared to the aluminum arm with same dimensions[KIT 90].

The first consideration in the design procedure is the actual configuration of the robot arm. The choice of configuration depends on the arm specification and the task it will be asked to perform. The analysis of automatic assembly operation shows that a single vertical movement is most common. This can be interpreted as the simple placement onto the desired location or insertion of a peg into a hole. The other required motion is only pick and place operations. Thus, the specification for an assembly robot can be stated that it must possess a general motion for positioning the component and a vertical motion for an actual assembly. The SCARA(Selected Compliance Articulated Robot Arm) configuration with its vertical motion at the tip of the arm and the horizontal pick-place motion of the arm seems suitable. The increasing interest in using robots in assembly has led to more SCARA type robot being designed, and this configuration is becoming standard for assembly robots.

The first work on direct drive arms was done at Carnegie-Mellon University[Asada 83]. The first of this prototype raised the major drawbacks in using direct drive technique to construct a practical arm. All the motors were placed at the joints in order to achieve direct drive mechanism. Hence the elbow motor became the load upon the preceding motor at the shoulder. This increasing inertia from tip to base may lead to excessive torque requirements upon the motor near the base. Therefore the major problem was that the motors were not properly specified for the desired loads and speed.

The problem of having motors located at the joint can be overcome by placing them at the stationary base. But this breaks the condition of direct drive as some mechanism should be used to transmit the torque from the base to the joint. Adept Technology uses steel bands in the design of the world's first commercial direct drive robot[Adept 85]. Strictly speaking, this is not a direct drive arm because a transmission mechanism is incorporated in the joint. The claim by Adept that the arm is direct drive is not dishonest because the design does not use any gears, harmonic drive or chains that are largely responsible for the problem of large friction, backlash and a degraded reliability. Although AdeptOne is not strictly a direct drive arm in the sense defined so far, the AdeptOne has been designed to eliminate the problems due to transmission mechanisms. Hence it is reasonable to relax the constraints for the definition of direct drive to include designs where simple transmission mechanism are used as long as the problems associated with conventional geared arm are eliminated or become insignificant.

Asada produced a design like the AdeptOne, in which the motors are located at the base. The design used a parallelogram construction of the arm links[Asada 84]. This idea of parallel drive mechanism is particularly suited to direct drive as it allows the motors to be installed at the base. But the two motors at the base in Asada's design are required to produce continuous torques due to gravity in order to keep the arm in a stationary position . It is known that high torque motor used in direct drive arm is not good at sustaining continuous torques for a long time because of heat dissipation problem.

The parallel drive configuration can be easily turned into a SCARA configuration by simply mounting it horizontally. In this configuration, the need to produce continuous torque against gravitational force can be eliminated and the torque developed by motor is used only to move the arms in the horizontal plane. As mentioned before, assembly task can be split into two distinct operations. By adopting the SCARA configuration, the design of an assembly robot can reflect these two operations such that horizontally mounted arms are used for a pick-place motion and a conventional geared mechanism can be utilized for the vertical motion at the tip of the arms.

The IC DDR(Imperial College Direct Drive Robot) has been designed to form a flexible assembly system for performing small batch size assembly task of printed circuit

boards. The robot has four degrees of freedom with maximum payload capacity 2 [Kg], target accuracy of less than 0.1mm and maximum acceleration capability more than 5 g(gravity). Such a high acceleration specification is assigned to achieve a minimum process cycle times. The mechanical construction makes use of graphite fiber epoxy composite material for achieving light weight and high stiffness of the positioning arm. The elbow joint is driven by the motor at the base through the unit ratio pulleys. This can not be called strictly direct drive. However, since there are no gearing mechanisms, the friction becomes insignificant and this could be reduced further by a good selection of bearing at the joints. Therefore the principal advantages of direct drive are maintained very well. The prototype IC DDR can be considered as falling into the wider definition of direct drive like AdeptOne. All of these design aims have been successfully implemented on IC DDR.

1.3 Research Objectives and Structure of The Thesis

Even though the direct drive arm has been shown to possess many advantages over conventional geared arm, the solution of one problem creates another problem. Unlike geared arms, the load inertias are not reduced but directly act on the actuating motor . In addition, the coupling torque and Coriolis and centrifugal torques become more significant due to high speed and acceleration compared to a conventional arm with a high gear ratio. Hence the robot becomes highly coupled nonlinear multivariable system as the movement of one arm produces disturbance torques at the other motors . These dynamic interactions between the arm links make the motion control problem difficult and is likely to result in large tracking errors.

The major concerns of this research are the development of a dynamic model suitable for computer controlled system, parameter identification problem and the motion control problem of the positioning arm. The wrist part has no problem because it has not serious dynamic interactions like the positioning arm part. Hence the wrist part of the robot is not covered in depth in this research.

A systematic procedure based on Lagrange-Euler method is applied for the derivation of IC DDR dynamic model in the first part of Chapter 2. The motor dynamics is integrated into the rigid body dynamic model of the arms, and the complete closed form

dynamic model is derived. Since the control action of a computer controlled system is performed in discrete time rather than in continuous time, a discrete time model is the natural way to express the dynamic behaviour of the system. A discrete-time dynamic model which is more accurate than the conventional discrete-time dynamic model and particularly suitable for control application is introduced at the last section of Chapter 2.

Chapter 3 deals with the parameter estimation problem which plays a decisive role in the dynamic behaviour of the developed model. Several identifiers are suggested for the parameter identification, and their performances are discussed under the various conditions.

Major considerations in control are given in Chapter 4. The accuracy and effectiveness of several dynamic models are compared in the sense of inverse dynamic calculation. The first control scheme is independent joint control which is popular in most of the industrial robots. Its performance and limitation on the highly coupled system are shown. The principles of dynamic compensation and control law partitioning for the highly coupled nonlinear system are discussed before the detailed performance evaluation of the dynamic compensation scheme. Adding a compensating control effort to counteract the dynamic coupling effects, merit of the proposed dynamic compensation scheme is investigated. In the last part of Chapter 4, the discrete-time dynamic model which does not require velocity or acceleration information is reconstructed and their performances in dynamic compensation scheme are also evaluated.

The final part of the work is devoted to the actual implementation of the proposed control scheme for IC DDR in Chapter 5. The experimental results on parameter estimation of IC DDR and real-time performances of the control scheme are presented. The conclusions and contributions of this research and areas for further study are addressed in Chapter 6.

2. DYNAMICS OF ROBOT SYSTEM

2.1 Introduction

The manipulator is an active mechanism which possesses separate drive for each degree of freedom. The manipulator arms are moved by controlling the actuators at each joint to get the desired motion. Because of its changing structure, the inertia seen by the actuator is changing throughout its movement. The motion of one joint will affect all the other joints through Coriolis and centrifugal forces, and the force due to gravity will also acts on a joint according to its position. Therefore, the dynamics of the robot motion is characterized by nonlinear, highly coupled, multivariable system.

Manipulator dynamics concerns the relationship between the motion of a mechanical kinematic chain of linkages and the forces applied by its actuators. The importance of manipulator dynamics is its use in simulation, analysis, and control. Simulation of dynamic response of a robot is a way of designing prototype manipulator and testing control strategies without the expense of working with actual manipulators. There are two applications related to the dynamics of a robot system. In the first application, if trajectory points are given, the joint torques to follow the specified trajectory can be calculated. This is useful for the application of controlling a robot system. The other one is to calculate how the mechanism will move under the application of given joint torques. This is useful for simulating the robot system behaviour. The formulation method which yields the solution to both application areas of dynamics is of interest. The number of computation required to form the model can be an important criterion from the point of view of on-line applicability.

The main approaches toward the dynamic equations of motion for robot system can be divided into two groups:

- Method based on Newton-Euler equation
- Method based on Lagrange-Euler equations

All other methods can be considered to be just variations of these twos. For instance, D'Alembert's principle of virtual work[Fu 87] is equivalent to the method of Lagrange equation because it is used to derive the Lagrange equation. The two approaches must,

of course, yield equivalent equations, but computational efficiency could differ. However, both methods can be used to derive equally efficient algorithm, provided that the proper choice of coordinate representation is adopted[Silver 82].

In the next section, the general procedure to form the dynamic model of a robot will be introduced based on Lagrange and Newton-Euler method.The design concept and mechanical structure of Imperial College Direct Drive Robot (IC DDR) will be described briefly in the first part of section 2.3, and the closed form of rigid body dynamic equation for the positioning arms is derived. In section 2.4, the model of actuating motor system is constructed, and its dynamic behaviour is discussed. the complete dynamic equations including motor dynamics are represented at the last part of this section. In the last section, the discretization method is discussed, and the discrete-time dynamic model which is much suitable for the computer controlled system will be derived using the first integral of motion[Landau 76].

2.2 Review of the Lagrange and Newton-Euler Method

The advantage of the Lagrange equations is that this method is straightforward and simple. The first step is to formulate the kinetic energy κ and the potential energy ρ of the manipulator, which can be found from summing the kinetic energy and potential energy of each link. In the Lagrange equations, the generalized coordinates which describe the position of objects in mechanical system must be chosen. It is convenient to choose joint variables q_i as the generalized coordinates. Then the Lagrange equation is

$$Q_i = \frac{d}{dt}\left(\frac{\partial \Lambda}{\partial \dot{q}_i}\right) - \frac{\partial \Lambda}{\partial q_i} \tag{2-1}$$

where Λ is the Lagrangian function $(\Lambda = \kappa - \rho)$ and Q_i are the generalized torques.

To describe the translational and rotational relationships between adjacent links, the Denavit-Hartenberg matrix representation[Denavit 55], which represents each link's coordinate system at the joint with respect to the previous link's coordinate system, is employed. The direct application of the Lagrangian dynamic formulation, together

with the Denavit-Hartenberg link coordinate representation, results in a convenient and compact description of the equations of motion.

Let ${}^{i}r_i$ be a point fixed and at rest in link i and expressed in coordinates with respect to the i-th link coordinate frame.

$$ {}^{i}r_i = (x_i, y_i, z_i, 1) \tag{2-2} $$

Let ${}^{0}r_i$ be the same point ${}^{i}r_i$ with respect to the base coordinate frame. The coordinate transformation matrix ${}^{i-1}A_i$ relates the spatial displacement of the i-th link coordinate frame to the (i-1)-th link coordinate frame, and ${}^{0}A_i$ relates the i-th coordinate frame to the base coordinate frame. Then ${}^{0}r_i$ is related to the point ${}^{i}r_i$ by

$$ {}^{0}r_i = {}^{0}A_i \; {}^{i}r_i \tag{2-3} $$

where,

$$ {}^{0}A_i = {}^{0}A_1 \, {}^{1}A_2 \ldots {}^{i-1}A_i \tag{2-4} $$

Assuming the link is rigid, the point ${}^{i}r_i$ fixed in link i will have zero velocity with respect to the i-th coordinate frame. The velocity of ${}^{i}r_i$ expressed in the base coordinate frame can be expressed as

$$ {}^{0}V_i \equiv V_i = \frac{d}{dt}\left({}^{0}r_i\right) = \left[\sum_{j=1}^{i} \frac{\partial {}^{0}A_i}{\partial q_j} \dot{q}_j\right] {}^{i}r_i \tag{2-5} $$

Let $d\kappa_i$ be the kinetic energy of a differential mass dm in link i, then

$$ d\kappa_i = \frac{1}{2} \mathrm{Tr}(V_i V_i^T)\, dm \tag{2-6} $$

In this formulation it is necessary to use the trace operator to form the tensor product $V_i V_i^T$ from which the inertia matrix J_i is found. Integrating over all differential masses, the total energy κ_i of the link is

$$ \kappa_i = \frac{1}{2} \mathrm{Tr}\left[\sum_{p=1}^{i} \sum_{r=1}^{i} U_{ip} \left(\int {}^{i}r_i \, {}^{i}r_i^T dm\right) U_{ir}^T \dot{q}_p \dot{q}_r\right] $$

$$= \frac{1}{2} \mathrm{Tr}\left[\sum_{p=1}^{i} \sum_{r=1}^{i} \mathrm{Tr}\left(U_{ip} J_i U_{ir}^T \right) \dot{q}_p \dot{q}_r \right] \tag{2-7}$$

where matrix U_{ij} is the rate of change of the point ${}^i r_i$ on link i relative to the base coordinate frame as q_j changes, and defined by

$$U_{ij} \equiv \frac{\partial {}^0A_i}{\partial q_j} \tag{2-8}$$

J_i is the inertia matrix which is expressed as $\int {}^i r_i \, {}^i r_i^T dm$ in link i coordinates. Each link's potential energy is given by,

$$\rho_i = -m_i \, g \, {}^0\bar{r}_r = -m_i \, g \, {}^0A_i \, {}^i\bar{r}_i \tag{2-9}$$

where ${}^i\bar{r}_i$ is a vector from the origin of link i to its center of gravity, $g=(0,0,g_c,0)$ is a gravity row vector. From Eqs.(2-7),(2-9) and applying (2-1) yields the necessary generalized torque Q_i for joint i.

$$Q_i = \sum_{j=i}^{n} \sum_{k=1}^{j} \mathrm{Tr}\left(U_{jk} J_j U_{ji}^T \right) \ddot{q}_k + \sum_{j=i}^{n} \sum_{k=1}^{j} \sum_{m=1}^{j} \mathrm{Tr}\left(U_{jkm} J_j U_{ji}^T \right) \dot{q}_k \dot{q}_m$$

$$- \sum_{j=i}^{n} m_j \, g \, U_{ji} \, {}^j\bar{r}_j \tag{2-10}$$

The complexity of this equation arises from dynamic interactions between links of the manipulator. It can be seen from the summations that the applied torque at a joint depends on the state of movement at all the other joints. There are three types of terms in this dynamic equation: inertial torques which are proportional to the joint accelerations $\ddot{q}_k$, velocity torques which are proportional to the product $\dot{q}_k \dot{q}_m$, and gravity torques. In addition, the velocity torques can be divided into centrifugal torques when k=m and into Coriolis torques when $k \neq m$. This dynamic equation is the closed form expression in the sense that the dependence of a joint torque on

movements at all joints is made explicit. In the closed form, most terms are reevaluated many times. Therefore the calculation of joint torque is too slow for real-time computation. Waters[Waters 79] noticed that the generalized torques could be expressed in the form of several backward recurrence relations (in the sense of the link numbering direction). With his formulation the number of additions and multiplications is reduced to the order of N^2. Hollerbach[Hollerbach 80] employed an additional forward recursion and increased the efficiency of the formulation done by Waters. This form of double recursion significantly reduces the amount of computation required to solve the Lagrangian model. The number of computation with his formulation could be decreased to the order of N.

While the Lagrangian dynamics were reworked with some effort into an efficient recursive form, this recursive nature falls out of application of the Newton-Euler equations. A Newton-Euler derivation begins with a free body analysis, in which each link is considered as a free body and obeying Newton's equation for linear movement,

$$F_i = m_i \bar{a}_i \tag{2-11}$$

and Euler's equation for angular acceleration

$$N_i = I_i \dot{\omega}_i + \omega_i \times I_i \omega_i \tag{2-12}$$

The most significant aspect of this formulation is that the computation time for the applied torques can be reduced significantly to allow real-time computation. This formulation results in a set of forward and backward recursive equations. But, instead of 4×4 transformation matrix in Lagrange formulation, this has messy vector cross-product terms.

Hooker[Hooker 65] and several people studied the motion of multi-bodies connected by means of joints based on Newton-Euler formulation. But the basic deficiency of their method lies in the fact that it does not permit the formulation of recursive kinematic and dynamic equations. Recently a number of papers have appeared on an efficient recursive Newton-Euler formulation of manipulator dynamics[Luh 80 / Orin 79 / Armstrong 79].

In order to compute inertial forces acting on the link it is necessary to compute the linear velocity and acceleration of the center of mass of link i respectively,

$$\bar{V}_i = \omega_i \times \bar{S}_i + V_i \tag{2-13}$$

$$\bar{a}_i = \dot{\omega}_i \times \bar{S}_i + \omega_i \times (\omega_i \times \bar{S}_i) + V_i \tag{2-14}$$

where $\bar{S}_i$ = position of center of mass of link i from the origin of the coordinate system (x_i, y_i, z_i)

$\bar{V}_i$ = linear velocity of the center of mass link i

$\bar{a}_i$ = linear acceleration of the center of mass of link i

and (x_i, y_i, z_i) is the moving-rotating coordinate system. Then from Fig. 2.1, the total external force F_i and moment N_i are those acting on link i by neighbouring links i-1 and i+1. That is,

$$F_i = f_i - f_{i+1} \tag{2-15}$$

$$N_i = n_i - n_{i+1} + (P_{i-1} - \bar{r}_i) \times f_i - P_i^* \times f_{i+1} \tag{2-16}$$

Using $\bar{r}_i - P_{i-1} = P_i^* + \bar{S}_i$, the above equation can be recast into recursive forms

$$f_i = F_i + f_{i+1} = m_i \bar{a}_i + f_{i+1} \tag{2-17}$$

$$n_i = n_{i+1} + P_i^* \times f_{i+1} + (p_i^* + \bar{S}_i) \times F_i + N_i \tag{2-18}$$

From the above recursive equations, the input torque at joint i can be computed as the sum of the projection n_i onto the Z_{i-1} axis. However, if joint i is translational, the input force at that joint is the sum of the projection of f_i onto the Z_{i-1} axis. Hence, the input torque or force for joint i is

$$\tau_i = \begin{cases} n_i^T z_{i-1} & \text{, joint i rotational} \\ f_i^T z_{i-1} & \text{, joint i translational} \end{cases} \tag{2-19}$$

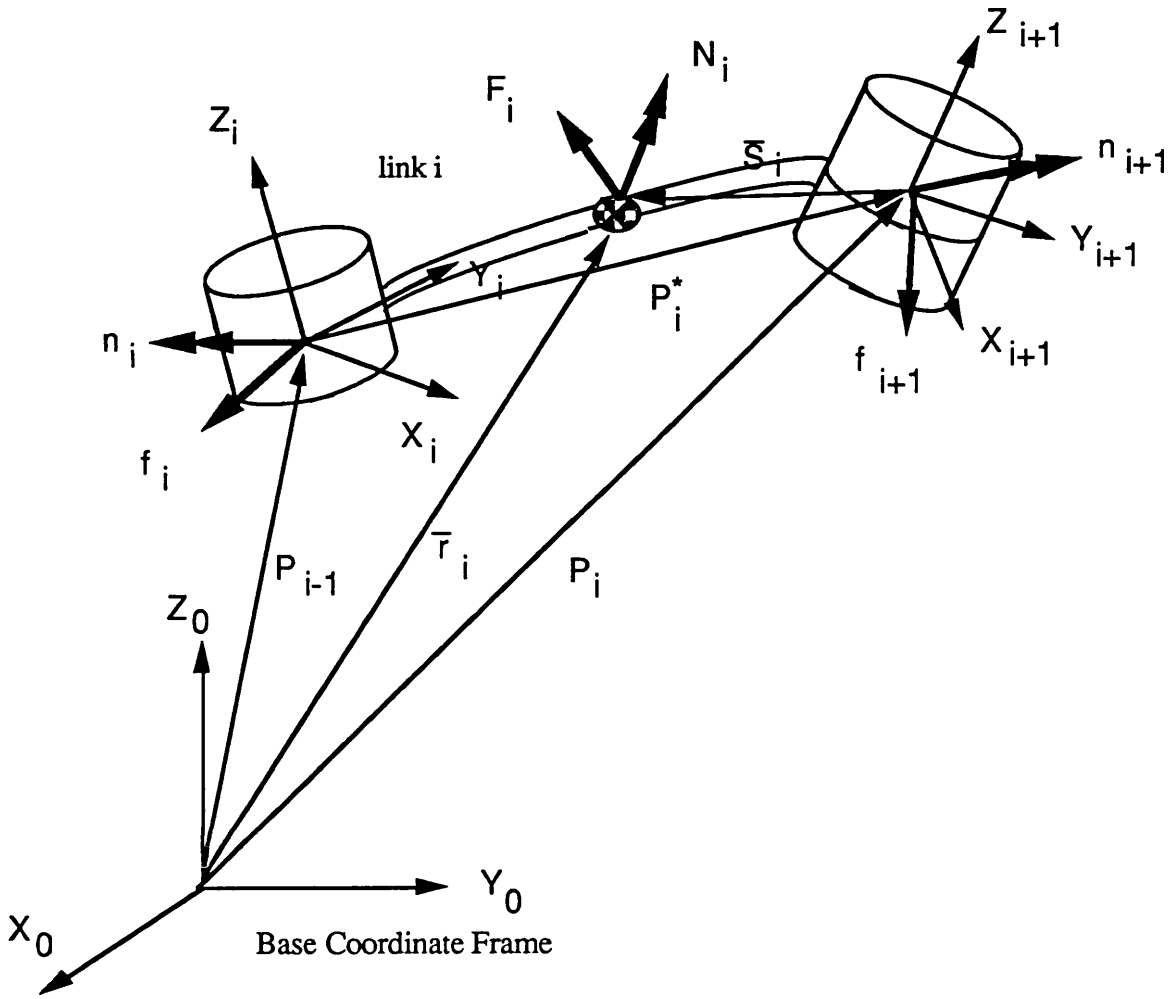

Figure 2.1 Forces and Moments on Link i

The complete procedure for computing joint torques or forces from the recursive equations is composed of two parts. Firstly, link velocities and accelerations are iteratively figured out from link 1 to link n and the Newton-Euler equations are applied to each link. Secondly, joint torques and forces are computed recursively from link n back to link 1.

2.3 Dynamic Model of the IC DDR Positioning Arms

2.3.1 Horizontally Mounted Arm

Fig.2.2 shows typical construction of a 2 DOF(degree of freedom) serial drive arm mechanism, in which the shoulder link is driven by a motor fixed on the base and the elbow link is driven by motor attached at the end of the shoulder link.

The arm mechanism in which each motor is located at the joint between adjacent links is referred to as a serial drive mechanism[Asada 84]. In this serial configuration, the weight of a motor itself becomes a load for the next motor down the serial link. Therefore the drive torque required for each motor increases from the distal joint to the base joint. Moreover, the reaction torque of the elbow motor acts on the shoulder motor. i.e., when the elbow motor accelerates in the clockwise direction, a counter-clockwise torque acts on the shoulder motor, and vice versa. Thus two motors have significant interactions.

Fig.2.3 shows an alternative arm drive mechanism using belt drive transmission for the second joint. Two motors located at the fixed base column drive the two links and produce a two-dimensional motion.

In the latter drive mechanism, the weight of one motor is not a load on the other because the motors are fixed on the stationary base. In contrast to the serial drive mechanism, the weight and reaction torque of one motor do not affect the other motor directly in this configuration. This is referred to as a parallel drive mechanism. The advantage of the parallel drive mechanism over the serial one are

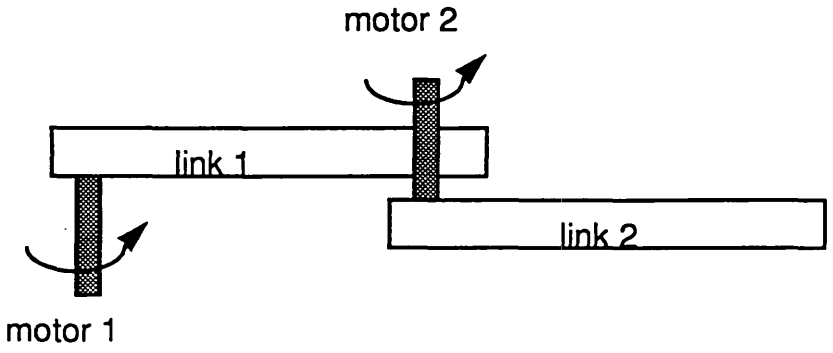

Figure 2.2(a) 2 DOF Serial Drive Arm

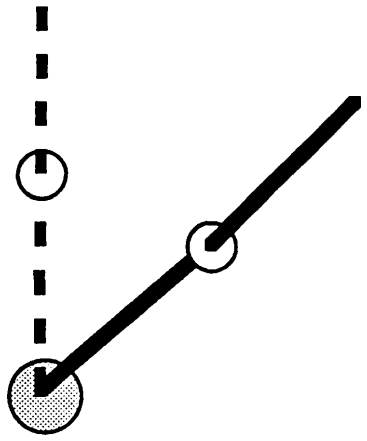

Figure 2.2(b) Arm Configuration When Joint 1 rotates

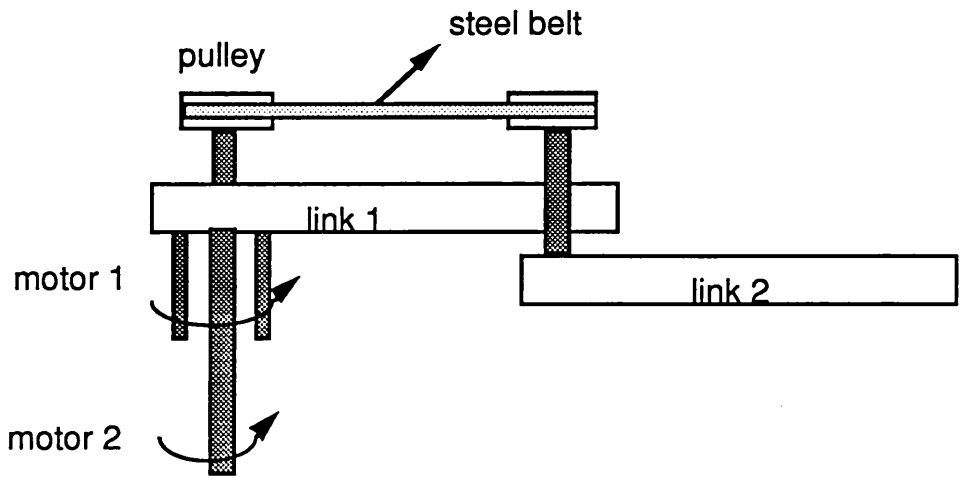

Figure 2.3(a) 2 DOF Horizontally Mounted Parallel Drive Arm
with Unity Pulley Mechanism

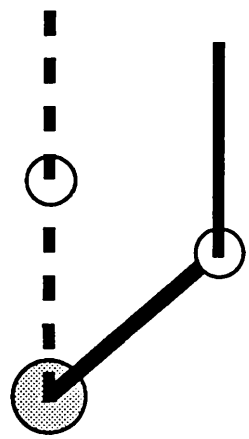

Figure 2.3(b) Arm Configuration When Joint 1 Rotates

- less weight of arm structure
- lower interaction and nonlinearity than the serial drive mechanism
- invariant diagonal elements of inertia matrix
- no nonlinear velocity torque which results from Coriolis effect

The dynamic equations and their properties will be dealt with in detail in the section 2.3.3.

2.3.2 Outline of the Mechanical Design

Fig.2.4 shows the sectional view of the direct drive robot developed at Imperial College, which is aimed at fast and precise assembly of electronic / mechanical components. The design emphasis has been put on high speed performance with accurate positioning and excellent repeatability. Generally, three attributes, i.e., accuracy, repeatability and speed of operation characterize the performance of a robot system. The desirable objective of a robot system can be achieved by maximizing all the three attributes simultaneously, but there exist conflicts of interest. For example, better accuracy may be obtained by reducing the speed of operation. Hence, the three attributes have to be compromised according to the actual requirements of work on objects.

The overall system of IC DDR consists of two subsystems. One is the horizontally articulated arm for positioning of the end-effector wrist. The second subsystem is the wrist mechanism(end mechanism) for handling an end-effector / gripper. The first two DOF constitute the positioning arm and the last two DOF constitute the wrist mechanism. The two subsystems differ in their task assignments and dynamic behaviours. The arm part is responsible for positioning the wrist part at a specified point in the working area. In contrast to the arm, the wrist part is responsible for fine motion in the workplace, which is relatively slower than the motion of arm part where rapid execution and precise position are required. Each submechanism is driven by the brushless high performance torque motors.

The arm links are manufactured from carbon fibre composite material which is ideally suitable for stiff and lightweight arms. In comparison with aluminum arm, the static and dynamic properties of the composite material arm is much improved: the total arm weight is reduced by 1.5 Kg, while the stiffness and natural frequency are improved by 15 % respectively. Table 2.1 shows the vibrational characteristics and static deflection for the positioning arm structure of IC DDR. The 1st natural frequency is due to the steel belt transmission mechanism and 2nd natural frequency is corresponding to the resonance frequency of the 1st mode bending vibration of arm structure.

TABLE 2.1 Stiffness and Natural Frequencies of the Positioning Arm Structure [KIT 89]

	Alumimum Arm	Composite Arm
1st Natural Frequency[Hz]	18	17
2nd Natural Frequency[Hz]	551	624
†Static Deflection [μm]	140	80

†measured at the tip of elbow arm under 1Kg vertical load

Since the driving motors for positioning mechanism are placed in the stationary column, the extremely lightweight arm structure has been realized. The first link(shoulder) has a rotational motion relative to the base column and is directly coupled to the motor rotator. The second link(elbow) is driven by the steel band pulley. The steel bands are clamped tightly on pulleys at their both ends. The shoulder pulley is divided into upper and lower part for adjusting the tension of the belt. The eccentric cam inside the shoulder pulley is designed to give the opposite directional rotation of the divided upper and lower pulley. An appropriate initial tension can be easily achieved by rotating the eccentric cam on shoulder pulley side. The common disadvantages such as backlash, lost motion, and unpredictable large friction in gear coupling do not appear in this clamped pulley transmission. Hence, the important features of direct drive are maintained.

For wrist mechanism, a precision ball screw is used to give a vertical motion and a torque resistant shaft is connected concentrically to the ball screw to produce a rotational motion.

When the ball screw nut is rotated, the shaft moves up and down, and if ball bushing nut of the torque resistant shaft is rotated, the shaft rotates. When both nuts are rotated in opposite directions, fast vertical movement can be achieved. To get a precise positional information, a resolver is installed concentrically along its rotor shaft of each driving motor. The analog signals from resolver are processed and calibrated by resolver to digital converter(RDC). Its maximum resultant resolution can be obtained up to 16 bits digital data.

Estimating the masses from engineering drawing data shows that the mass of the wrist mechanism is small in comparison with the mass of positioning arms. The operation of wrist mechanism over some object is mainly carried out when the positioning arm is at rest or nearly stationary. Since the inertia of the end-effector/gripper is much less than that of the positioning mechanism, the dynamic interaction between the two subsystem can be viewed as negligibly weak. Thus, it is reasonable to consider dynamic behaviour for the two subsystems separately rather than through the complicated whole dynamic equations. By this physical insight, a sophisticated procedure in the derivation of dynamic equations can be avoided, and at the same time the simpler form of dynamic equations make it easier to construct a proper controller for each subsystem. In this approach, the whole wrist part including an end-effector/gripper can be viewed as a lumped mass which is attached to the end of the last arm of positioning mechanism, and this lumped mass is included in the computation of inertia of the last link of the positioning system.

The dynamic equation of the wrist part mechanism are represented as linear differential equations[Kim 90]. Therefore, its control algorithm could be set up through the simple PID(proportional-integral-derivative) method or the well-established linear system theory.

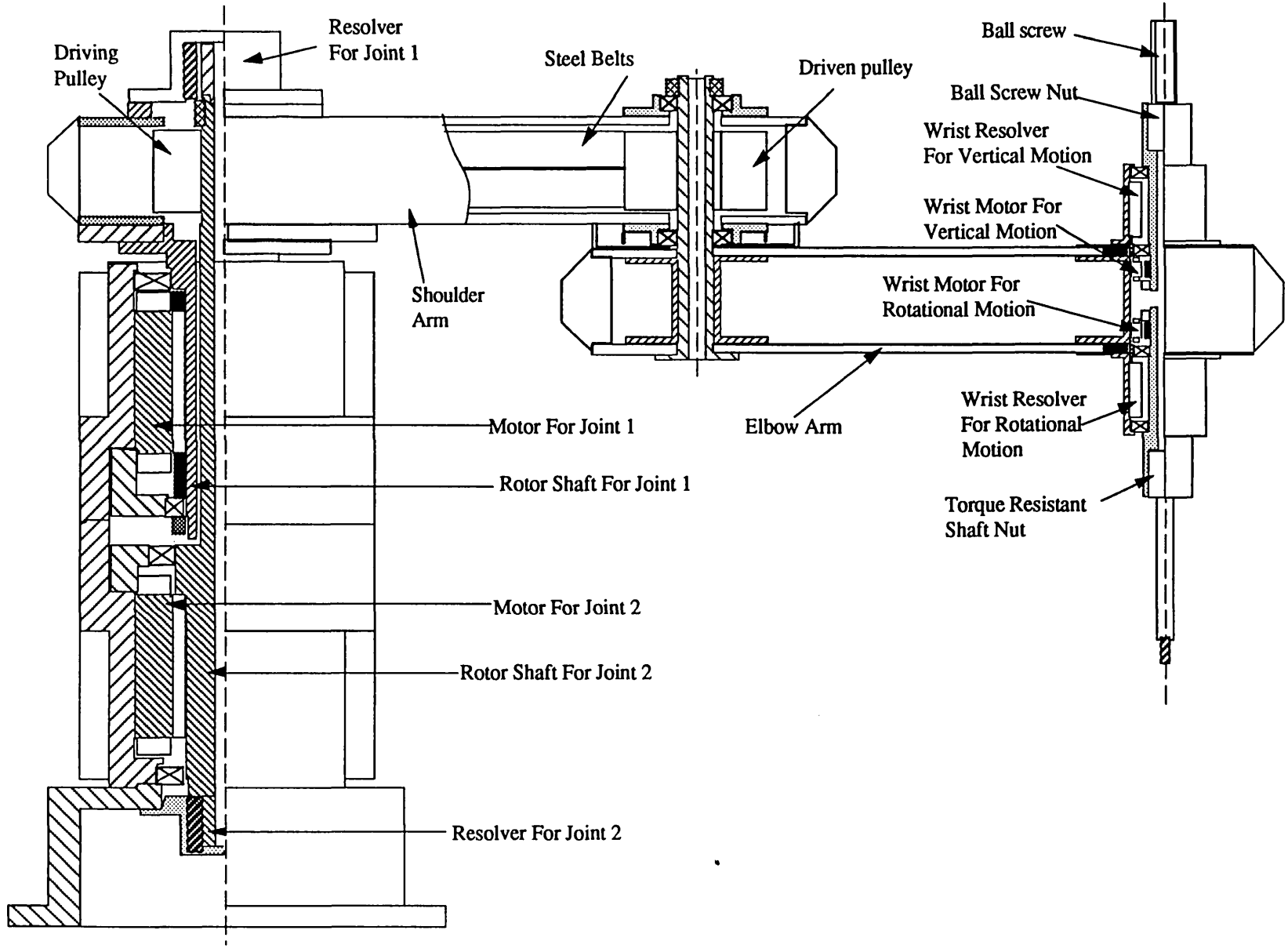

Figure 2.4 The Sectional View of IC DDR

2.3.3 Dynamic Equations of the Positioning Arm

In general, the dynamic equations of multi-link manipulator are highly nonlinear and consist of inertial loading, coupling reaction torques / forces between joints, and gravity loading effect. Especially the velocity generated reaction torques/forces depend both on the instantaneous velocities and the configuration of the links.

The recurrence representation of dynamic equations is quite efficient as a general means of computing the dynamics of any manipulator, but explicit representation of each term of dynamic equations is hidden in this procedure. An explicit closed form of dynamic equations can give a better insight for the investigation of the effects of various reaction and coupled terms. The main drawback of formulating the closed form of dynamic equations is simply that it requires a fair amount of desk calculations.

As described at section 2.3.2, the Imperial College Direct Drive Robot has four degrees of freedom excluding an end-effector / gripper. The first two joints are responsible for positioning of the wrist part in the horizontal plane, while the last two joints determine the direction and vertical position of an end-effector/gripper. The up-down motion of the wrist part has no dynamic effects on the positioning mechanism and the maximum torque delivered to the last two joints at the wrist mechanism is 0.41 [Nm] (0.4% of the peak torque of the first two joints). Therefore the dynamic coupling between two submechanisms (positioning arm and wrist part) can be considered negligibly small. For this reason, the wrist part is treated as an object attached at the end of the last link of the positioning mechanism.

The closed form dynamic model of IC DDR is derived based on Lagrangian formulation method[Lee 83] because of its simple and systematic procedure. The kinematic relationships between adjacent links are given by Denavit-Hartenberg matrix representation for each link. Fig.2.5 shows the schematic diagram with the physical dimensions and the established link coordinate system. The base coordinate is defined as the 0-th coordinate frame (x_0 y_0 z_0) which is the inertial coordinate frame of the robot. The four geometric parameters associated with each link are depicted in Table 2.2 based on the selected link coordinate frames.

TABLE 2.2 Link Coordinate Parameter

joint i	α_i	a_i	d_i	β_i
1	0	L	0	q_1
2	0	L	d_2	q_2

a_i : The orthogonal distance from Z_{i-1} to Z_i along the X_i axis (the distance of the link)

α_i : The angle from Z_{i-1} to Z_i about X_i axis (the twist of the link)

d_i : The distance from the origin of (i-1)th coordinate to the intersection of Z_{i-1} axis with the X_i axis along the Z_{i-1} axis (the distance between links)

β_i : The joint angle from X_{i-1} axis to the X_i axis about the Z_{i-1} axis (the angle between the links)

A transformation matrix ${}^{i-1}A_i$ which relates the i-th coordinate frame to the (i-1)th coordinate frame has the form as followings

$$
{}^{i-1}A_i = \begin{bmatrix} \cos\beta_i & -\cos\alpha_i \sin\beta_i & \sin\alpha_i \sin\beta_i & a_i \cos\beta_i \\ \sin\beta_i & \cos\alpha_i \cos\beta_i & -\sin\alpha_i \cos\beta_i & a_i \sin\beta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-20}
$$

Then the absolute velocity of the point r_i with respect to the base coordinate frame can be written as Eq.(2-5).

The partial derivative of ${}^0A_i$ with respect to joint variable (q_1 or q_2) can be calculated easily and defined by

$$
\frac{\partial {}^0A_i}{\partial \beta_j} \equiv U_{ij} \tag{2-21}
$$

Another derivative which appears in Eq.(2-10) is defined as

$$
\frac{\partial U_{ij}}{\partial \beta_k} \equiv U_{ijk} \tag{2-22}
$$

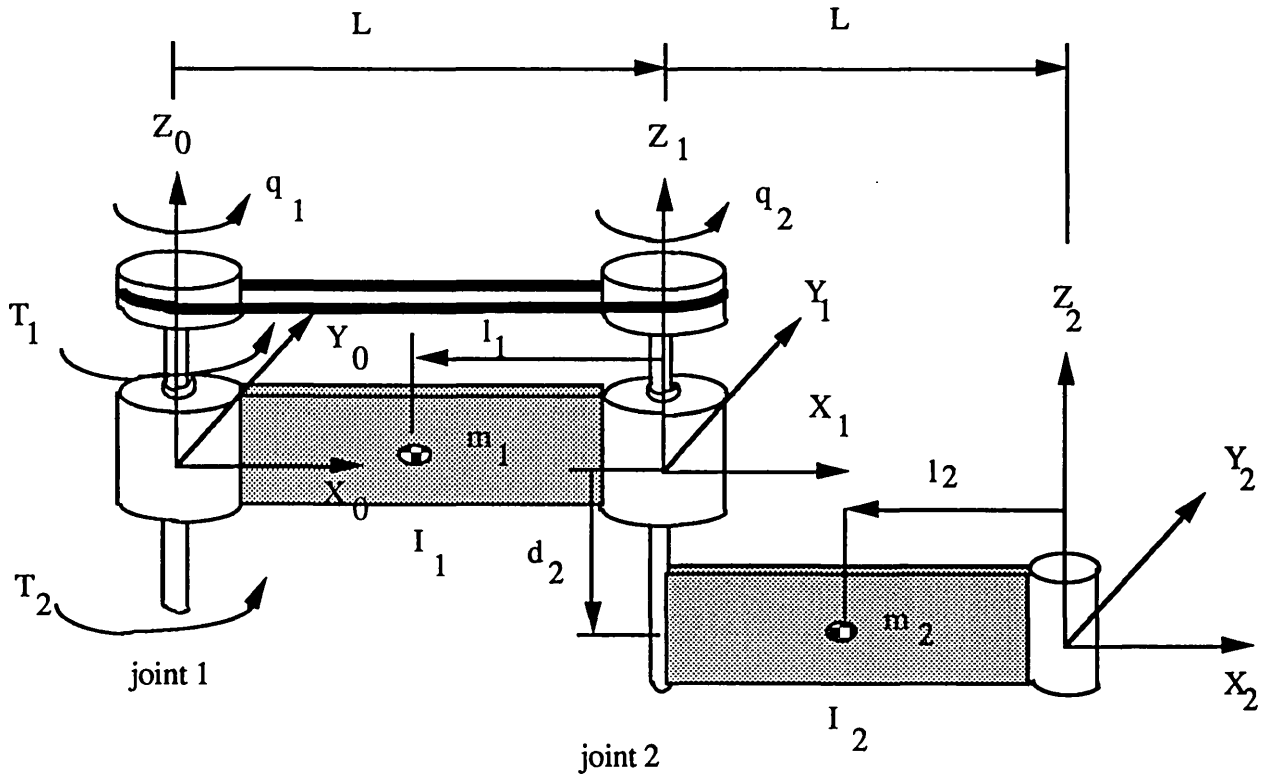

Figure 2.5 The Schematic Drawing of IC DDR and Its Coordinates Frame Assignments

By using this notation and neglecting the terms related to gravity because of the confined horizontal motion of the first two links, the required generalized torque Q_i for each joint i can be written in the simpler form

$$Q_i = \sum_{k=1}^{2} D_{ik}\dot{q}_k + \sum_{k=1}^{2}\sum_{k=1}^{2} h_{ikm}\dot{q}_k\dot{q}_m \tag{2-23}$$

where

$$D_{ik} = \sum_{j=\max(i,k)}^{2} Tr(U_{jk}J_jU_{ji}^T) \tag{2-24}$$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^{2} Tr(U_{jkm}J_jU_{ji}^T) \tag{2-25}$$

The coordinate transformation matrices ${}^{i-1}A_i$ (i=1,2) are obtained according to Eq.(2-20)

$${}^0A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & LC_1 \\ S_1 & C_1 & 0 & LS_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-26}$$

$${}^1A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & LC_2 \\ S_2 & C_2 & 0 & LS_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-27}$$

$${}^0A_2 = {}^0A_1\,{}^1A_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & L(C_{12}+C_1) \\ S_{12} & C_{12} & 0 & L(S_{12}+S_1) \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2-28}$$

From Eq.(2-21),

$$U_{11} = \frac{\partial}{\partial} \frac{{}^0A_1}{q_1} = \begin{bmatrix} -S_1 & -C_1 & 0 & -LS_1 \\ C_1 & -S_1 & 0 & LC_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2-29}$$

$$U_{21} = \frac{\partial}{\partial} \frac{{}^0A_2}{q_1} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -L(S_{12}+S_1) \\ C_{12} & -S_{12} & 0 & L(C_{12}+C_1) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2-30}$$

$$U_{22} = \frac{\partial}{\partial} \frac{{}^0A_2}{q_2} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -LS_{12} \\ C_{12} & -S_{12} & 0 & LC_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2-31}$$

where $C_i = \text{COS } q_i$, $S_i = \text{SIN } q_i$, $C_{ij} = \text{COS } (q_i+q_j)$, $S_{ij} = \text{SIN } (q_i+q_j)$
J_i in Eq.(2-24,25) is called the pseudo-inertia matrix on link i and expressed as

$$J_i = \int {}^i r_i \, {}^i r_i^T dm \tag{2-32}$$

Assuming that the link masses are evenly distributed on each X_i axis with some line mass density, then all the products of inertia can be zeroed out and pseudo-inertia matrices for the first two links can be represented as

$$J_1 = \begin{bmatrix} I_1 & 0 & 0 & -l_1 m_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -l_1 m_1 & 0 & 0 & m_1 \end{bmatrix} \tag{2-33}$$

$$J_2 = \begin{bmatrix} I_2 & 0 & 0 & -l_2 m_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -l_2 m_2 & 0 & 0 & m_2 \end{bmatrix} \tag{2-34}$$

Then using Eq.(2-26), the D_{ij} terms related to acceleration can be computed with these pseudo-inertia matrices.

$$D_{11} = Tr\,(U_{11}\,J_1\,U^T{}_{11}) + Tr(U_{21}\,J_2\,U^T{}_{21})$$
$$= I_1 - 2m_1\,l_1\,L + m_1\,L^2 + I_2 - 2m_2\,l_2\,L + 2m_2\,L^2 + 2\,(m_2\,L^2 - m_2\,l_2\,L)\,C_2 \qquad (2\text{-}35)$$

$$D_{12} = D_{21} = Tr(U_{22}\,J_2\,U^T{}_{21})$$
$$= I_2 - 2m_2\,l_2\,L + m_2\,L^2 + (m_2\,L^2 - m_2\,l_2\,L) \qquad (2\text{-}36)$$

$$D_{22} = Tr(U_{22}\,J_2\,U^T{}_{22})$$
$$= I_2 - 2m_2\,l_2\,L + m_2\,L^2 \qquad (2\text{-}37)$$

From Eq.(2-23), the Coriolis and centrifugal terms, h_1 and h_2 are

$$h_1 = h_{111}\dot{q}_1^2 + h_{112}\dot{q}_1\dot{q}_2 + h_{121}\dot{q}_1\dot{q}_2 + h_{122}\dot{q}_2^2 \qquad (2\text{-}38)$$
$$h_2 = h_{211}\dot{q}_1^2 + h_{212}\dot{q}_1\dot{q}_2 + h_{221}\dot{q}_1\dot{q}_2 + h_{222}\dot{q}_2^2 \qquad (2\text{-}39)$$

where, h_{ikm} can be obtained from Eq.(2-22) and (2-25)

$$h_{111} = Tr\,(\,U_{111}\,J_1\,U^T{}_{11}\,) + Tr\,(U_{211}\,J_2\,U^T{}_{21}) = 0 \qquad (2\text{-}40)$$
$$h_{112} = Tr\,(\,U_{212}\,J_2\,U^T{}_{21}\,) = (\,-m_2\,L^2 + m_2\,l_2\,L\,)\,S_2 \qquad (2\text{-}41)$$
$$h_{121} = Tr\,(\,U_{221}\,J_2\,U^T{}_{21}\,) = (\,-m_2\,L^2 + m_2\,l_2\,L\,)\,S_2 \qquad (2\text{-}42)$$
$$h_{122} = Tr\,(\,U_{222}\,J_2\,U^T{}_{21}\,) = (\,-m_2\,L^2 + m_2\,l_2\,L\,)\,S_2 \qquad (2\text{-}43)$$
$$h_{211} = Tr\,(\,U_{211}\,J_2\,U^T{}_{22}\,) = (\,m_2\,L^2 - m_2\,l_2\,L\,)\,S_2 \qquad (2\text{-}44)$$
$$h_{212} = Tr\,(\,U_{212}\,J_2\,U^T{}_{22}\,) = 0 \qquad (2\text{-}45)$$
$$h_{221} = Tr\,(\,U_{221}\,J_2\,U^T{}_{22}\,) = 0 \qquad (2\text{-}46)$$
$$h_{222} = Tr\,(\,U_{222}\,J_2\,U^T{}_{22}\,) = 0 \qquad (2\text{-}47)$$

Finally, the equations of motion in matrix form can be represented as:

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} = \begin{pmatrix} P_1 + P_3 + 2P_2C_2 & P_3 + P_2C_2 \\ P_3 + P_2C_2 & P_3 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix}$$

$$+\begin{pmatrix} -2P_2S_2\dot{q}_1\dot{q}_2 - P_2S_2\dot{q}_2^2 \\ P_2S_2\dot{q}_1^2 \end{pmatrix} \tag{2-48}$$

where, $P_1 = I_1 - 2m_1 l_1 L + m_1 L^2 + m_2 L^2$

$P_2 = m_2 L^2 - m_2 l_2 L$

$P_3 = I_2 - 2m_2 l_2 L + m_2 L^2$

and Q_1, Q_2 are the generalized torques. If motor is located at each joint, in other word, the arm structure is a serial drive mechanism, Q_1 and Q_2 in Eq.(2-48) correspond to the torques developed by each motor at joint. In these circumstances, the diagonal elements of the inertia tensor depend on the arm configuration. Moreover, the nonlinear velocity torque which results from Coriolis effect have an influence on the dynamic behaviour of the shoulder arm .

But the drive motors of IC DDR are mounted on the fixed base, the generalized torques, Q_1 and Q_2 can not be substituted for the actual torques directly as in the serial link case. In the parallel link drive mechanism such as IC DDR, the relationship between the generalized torques and actual torques developed by each motor at fixed base can be found through a virtual work. When there are actual torques acting upon each joint, the virtual work, δW by these applied torques can be written as

$$\delta W = \sum_i Q_i \, \delta q_i = \sum_i T_i \, \delta A_i \tag{2-49}$$

Here, δA_i is the angular displacements which is a function of the generalized coordinates q_1, q_2 and T_i is the actual torque developed by motor. The angular displacements A_i are the joint angles measured to the base inertial coordinate frame. i.e.,

$$A_1 = q_1 \tag{2-50a}$$

$$A_2 = q_1 + q_2 \tag{2-50b}$$

So, the virtual work has the form

$$\delta W = T_1 \, \delta A_1 + T_2 \, \delta A_2 = (T_1 + T_2) \, \delta q_1 + T_2 \, \delta q_2 \tag{2-51}$$

This can be regarded as the product of the generalized torques Qi acting over the generalized displacement δq_i so that the generalized torques have the relationship

$$Q_1 = T_1 + T_2 \tag{2-52a}$$
$$Q_2 = T_2 \tag{2-52b}$$

Eq. (2-48) is derived by considering only rigid body dynamics. The important source of forces that are not included in rigid body dynamics is friction. In some manipulators which have a high gear reduction with low backlash, over 25% of the total motor power is dissipated at the gearing[Craig 86]. The large friction could induce poor control performance and become a serious problem for high precision applications. In the IC DDR, the first two joints are driven with no gear reduction, hence friction is significantly reduced. It may only exist at the bearing elements supporting the joint drive shaft.

Generally, friction is highly unpredictable quantity that is difficult to estimate. However, it is important to represent these friction forces through a reasonable model. A simple one is the viscous friction model in which the resisting torque due to friction is proportional to the angular velocity of joint motion. The dynamic model including this type of friction torques can be written in terms of the actuating torques T_1, T_2 and actuator angles A_1, A_2.

$$\begin{pmatrix} T_1 - P_4 \dot{A}_1 \\ T_2 - P_5 \dot{A}_2 \end{pmatrix} = \begin{pmatrix} P_1 & P_2 C_{2\bar{1}} \\ P_2 C_{2\bar{1}} & P_3 \end{pmatrix} \begin{pmatrix} \ddot{A}_1 \\ \ddot{A}_2 \end{pmatrix} + \begin{pmatrix} -P_2 S_{2\bar{1}} \dot{A}_2^2 \\ P_2 S_{2\bar{1}} \dot{A}_1^2 \end{pmatrix} \tag{2-53}$$

Here,

$C_{2\bar{1}} = \mathrm{COS}[A_2(t) - A_1(t)]$, $S_{2\bar{1}} = \mathrm{SIN}[A_2(t) - A_1(t)]$

P_4, P_5 are the coefficients of viscous friction at each kinematic joint respectively.

The dynamic equation of the parallel mechanism Eq.(2-53) has differences as compared with that of the serial link mechanism Eq.(2-48). Namely, the diagonal elements of inertia matrix are invariant and the off-diagonal elements, which are the interactive inertia torque, is less than those of the serial link mechanism. In addition, the nonlinear velocity torque which comes from Coriolis effect disappears in Eq.(2-53).

The numerical values of the parameters can be roughly estimated from design data and manufacturer's specifications. But the coefficients of viscous friction are poorly known priori and may be estimated through experimental measurement.

2.4 Complete Dynamic Model of the Positioning System

2.4.1 Mathematical Model of Motor and Amplifier System

The performance of a direct drive arm is highly dependent on the performance of the motors. All the joints are powered by brushless DC motors which utilize Samarium Cobalt rare-earth magnet material. The rotor, which contains permanent magnets, is directly coupled to the joint axis. Since motor winding coils are part of the stationary outer stator, the motor has a better heat dissipation than a conventional DC torque motor. Moreover, brushless motor does not cause an electrical spark which may create a dangerous situation in explosive environments. The characteristics of the motors used for IC DDR are listed in TABLE 2.3

The amplifier provides the correct amount of current and voltage needed to drive motor. An amplifier in which the voltages vary continuously is called a linear amplifier. Their drawback is the operational inefficiency caused by the dissipation of large amount of power in the output stage transistors. Especially, the operations at low speed and high torque requirement invoke the high power dissipation in the linear amplifier. This power dissipation problem can be solved by using a switching amplifier which controls the applied motor voltage by varying the duty cycle ratio(mark space ratio) of the fixed supplying voltage. The output stage transistor of this type amplifier are turned on and off like a switch. When the transistor is turned on, the voltage across it is negligible; when the transistor is turned off, the voltage is large, but the current is nil. In either case, the power dissipated in the output transistor is very small.

PWM(Pulse Width Modulated) amplifiers with switching rate of 24 [KHz] are used to drive motors of IC DDR. The applied mean voltages to motors are adjusted according to a mark space ratio. The maximum swing voltage is 100 [Volts] to the high power

motors of positioning arm and 24 [Volts] for the low power motors of wrist mechanism.

TABLE 2.3 Brushless DC Motor Specification Used for IC DDR

Feature[unit]: symbol	Positioning Arm	Wrist Part
Model	INLAND/ RBE 03006-A50	INLAND/ RBE 01200
Stator Diameter [m]	1.27×10^{-1}	4.9×10^{-2}
No. of Poles	12	8
Viscous Damping [Nm/rad/sec]	2.62×10^{-3}	1.05×10^{-5}
Terminal Resistance [Ω]: R_i	2.7	3.1
Terminal Inductance [H]: D_i	1.4×10^{-2}	1.3×10^{-3}
Rotor Inertia [Kg m2]	1.53×10^{-3}	5.15×10^{-6}
Motor Constant [Nm/$\sqrt{W}$]	1.53	3.0×10^{-2}
Unhoused Weight [Kg]	5.9	8.8×10^{-2}
Peak Current [Amp]	37	7.7
Peak Torque [Nm]	93	0.41
Torque sensitivity [Nm/Amp]:K_{ti}	2.51	5.3×10^{-2}
Back EMF [Volt/rad/sec]:K_{bi}	2.513	5.29×10^{-2}

If a proper commutation method is implemented, the electrical equation for the brushless DC (BDC) motor is the same as that of a regular brushtype DC motor. Since the output from the PWM amplifier is a square wave train with varying mark space ratio, it is a nonlinear device. Even though the instantaneous input-output relationship of PWM amplifier is nonlinear, the applied mean voltage across the motor can be considered to be linear to the input of amplifier. Thus, the PWM amplifier can be seen as a linear element in the motor drive system.

Fig.2.6 shows the block diagram of the DC motor and its amplifier. The electrical relationship between applied mean voltage V_m and the resulted current i in the stator is given by the well-known equation.

$$V_m(t) = R\ i(t) + D\frac{d\,i(t)}{dt} + B(t) \tag{2-54}$$

where B(t) is the back EMF(Electromotive Force Voltage) due to the rotation of the permanent rotor, and is given by

$$B(t) = K_b\ \dot{A}(t) \tag{2-55}$$

here, K_b is called the back EMF constant and $\dot{A}(t)$ is the angular velocity of the rotor. A turning torque T_e on the shaft is given by the current i,

$$T_t = K_t\ i(t) \tag{2-56}$$

then, the dynamic equation of rotor shaft is represented as

$$J_r\ \ddot{A} = T_t - T \tag{2-57}$$

T is a torque from an external load. It is assumed that the shape of a generated torque T_t is kept flat without ripple during commutation, so the variations in K_b or K_t as a function of rotor position are neglected. Eqs. from (2-54) to (2-57) constitute a basic set of equations that model the driving system of the DC motor. As mentioned before, the relationship between input value U for PWM amplifier and the corresponding output value V_m is modelled as a linear element having an amplifier constant K_a.

Taking the Laplace transform to the basic set of equations and rearranging leads to the transfer function relating the input to the amplifier U(s) and angular velocity sA(s) as below:

$$\frac{s\,A(s)}{U(s)} = \frac{\dfrac{K_t K_a}{D J_r}}{s^2 + \dfrac{R}{D}s + \dfrac{K_t K_b}{D J_r}} \tag{2-58}$$

The block diagram representation of Eq.(2-58) is shown at Fig.2.7. Eq.(2-58) can be rewritten in terms of electrical time constant (τ_e) and mechanical time constant (τ_m) which are defined as followings.

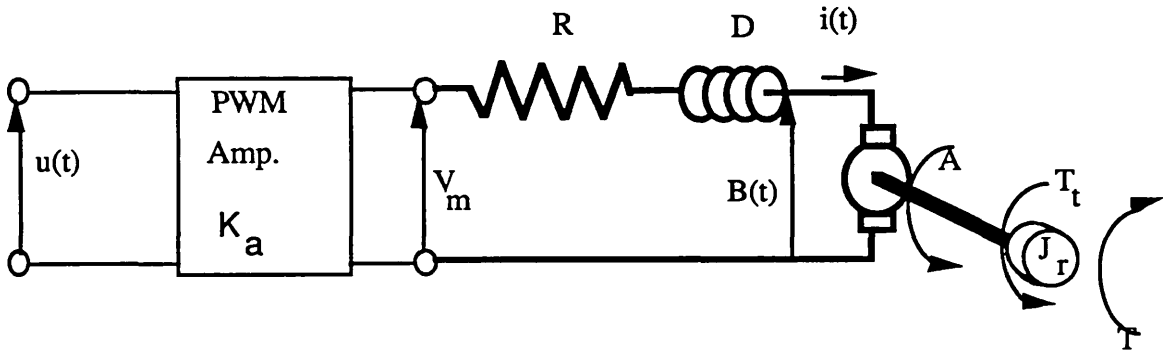

Figure 2.6 Simplified Block Diagram of DC motor and Amplifier

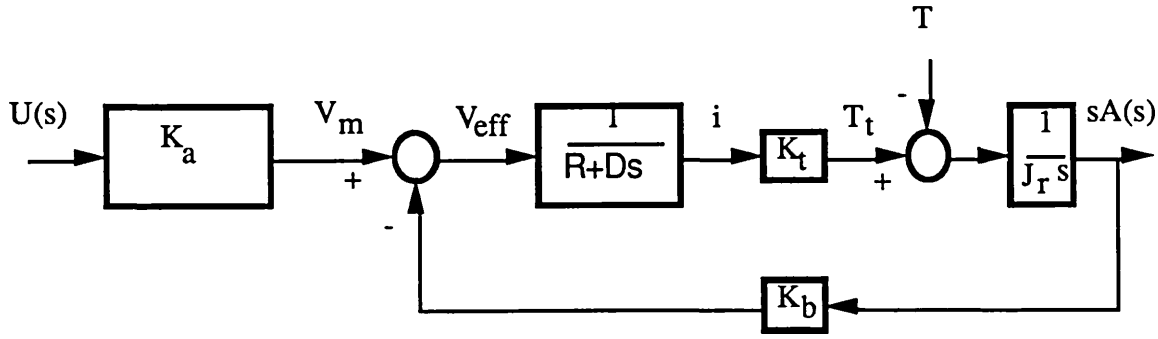

Figure 2.7 Block Diagram of Simple Joint Motor System(Open Loop)

$$\tau_e \equiv \frac{D}{R} \tag{2-59a}$$

$$\tau_m \equiv \frac{J_r R}{K_t K_b} \tag{2-59b}$$

The electrical time constant represents the time for motor current to rise up to 63% of its final steady state value, while the mechanical time constant means the time required for the rotor to reach 63% of its final speed after the application of a constant DC voltage. Then, using the time constants,

$$\frac{s\,A(s)}{U(s)} = \frac{\left(\frac{K_e K_a}{D J_r}\right)}{s^2 + \left(\frac{1}{\tau_e}\right)s + \left(\frac{1}{\tau_e \tau_m}\right)} \tag{2-60}$$

Before incorporating motor dynamics into the complete dynamics of robot system, it is helpful to investigate the dynamic properties of motor-amplifier system. The simple examination of Eq.(2-60) gives a reasonable ground on the selection of the differential order of the motor dynamic equation. Beside the inertia of the shaft-rotor assembly, the effective inertia felt by rotor has to be considered. The effective inertia of each joint is computed from the calculated parameters in TABLE 3.2, and listed in the second column of TABLE 2.4. Using the listed values in TABLE 2.3 and the effective inertias, the time constants can be calculated.

Clearly, the transfer function Eq.(2-60) has two real poles because the electrical time constants are fairly small in comparison with the mechanical time constants. Its two poles are represented as followings,

$$s_{1,2} = \frac{1}{2\tau_e}\left(-1 \pm \sqrt{1 - 4\,\tau_e/\tau_m}\,\right)$$

$$\cong -\frac{1}{\tau_m}\,,\ -\frac{1}{\tau_e} + \frac{1}{\tau_m} \tag{2-61}$$

From Eq.(2-61), it can be recognized that the dominant pole,which is closer to the origin, is determined by the mechanical time constant, and the 2nd pole, which is far from the origin, is mainly influenced by the electrical time constant. The dominant pole

is called the slow pole because the dynamic response governed by this pole decays slower than that by 2nd pole(fast pole). The resultant pole locations are given in TABLE 2.4 .

TABLE 2.4 Time Constants and Pole Locations

Motor(Joint)	Effective Inertia [Kgm^2]: †J_{ei}	Mech. Time Const. [sec]: τ_{mi}	Elect. Time Const. [sec]: τ_{ei}	Slow Pole Location	Fast Pole Location
Shoulder Arm	0.645	0.276	0.005185	-3.7	-189.2
Elbow Arm	0.300	0.128	0.005185	-8.2	-184.7

† $J_{e1} = P_1 + J_{r1}$, $J_{e2} = P_3 + J_{r2}$

As can be seen in this table , the fast poles are located far from the slow poles. This fact means that the dynamic response by the fast pole can be neglected without the loss of generality because of its fast decaying property. In other words, the mathematical model of motor system can be described good enough without the fast pole. Since the fast pole is created by the existence of the motor inductance, the inductance in Eq.(2-54) can be neglected on the basis of the above discussion. Then, the simplified equation of motor dynamics is represented as

$$T_i(t) = \eta_i U_i(t) - J_{ri} \ddot{A}_i - b_i \dot{A}_i \qquad (2\text{-}62)$$

Here, $\eta_i = (K_{ti} K_{ai})/R_i$, $b_i = (K_{ti} K_{bi})/R_i$

2.4.2 The Complete Closed Form of Dynamic Equation

In order to form the complete dynamic model of robot system, the dynamic models of the mechanical part and motor system have to be united. This can be constructed using the kinematic connection between joint motion and actuator motion. In general case, the relationship between joint variables and actuator angles may appear as a linear function. But each kinematic joint of the IC DDR is actuated directly by the corresponding motor except the linear up-down motion of the wrist mechanism, where

two motors work together to cause a linear joint motion. All the position sensors are installed concentrically at the rotational shafts of motor. Both motors used for the positioning arm have the same electrical characteristics, and the identical driving amplifiers are used. Hence, the parameters related to the electrical characteristics of motor are redefined,

$$\eta_1 = \eta_2 \equiv C_h$$
$$b_1 = b_2 \equiv b_h \tag{2-63}$$

and for the clarity of equations, the parameters appearing in Eq.(2-53) and (2-62) are reset as followings:

$$\begin{aligned}
(p_1 + J_{r1}) / C_h &\equiv \Theta_1 \\
p_2 / C_h &\equiv \Theta_2 \\
(P_3 + J_{r2}) / C_h &\equiv \Theta_3 \\
(P_4 + b_h) / C_h &\equiv \Theta_4 \\
(P_5 + b_h) / C_h &\equiv \Theta_5
\end{aligned} \tag{2–64}$$

Then, the complete dynamic equations for positioning arm including motor dynamics can be obtained,

$$U_1 = \Theta_1 \ddot{A}_1 + \Theta_2 C_{2\bar{1}} \ddot{A}_2 - \Theta_2 S_{2\bar{1}} \dot{A}_2^2 + \Theta_4 \dot{A}_1 \tag{2-65a}$$

$$U_2 = \Theta_2 C_{2\bar{1}} \ddot{A}_1 + \Theta_3 \ddot{A}_2 + \Theta_2 S_{2\bar{1}} \dot{A}_1^2 + \Theta_5 \dot{A}_2 \tag{2-65b}$$

where, $C_{2\bar{1}} = \mathrm{COS}\,[A_2(t) - A_1(t)]$
$S_{2\bar{1}} = \mathrm{SIN}\,[A_2(t) - A_1(t)]$

2.5 Discrete-time Dynamic Model

Digital computers are being used increasingly as tools for analysis and design of control system. In many areas, digital computers are outperforming their analog counterparts and are cheaper as well. Microcomputers of these days have computing

power greater than large main frames of the late seventies, and the ratio of price to performance is expected to drop substantially further. Based on this revolutionary development of modern computer technology, the approach to analysis,design and implementation of control system is changing drastically, and more advanced regulator can be introduced even for basic applications. The main features of computer controlled system can be illustrated as

- No need for a change in hardware wiring
- Easy realization of a complicated control algorithm

The computer controlled system contains both continuous time signals and sampled discrete time signals. Discrete-time system of which the behaviour is described at sampling time instants deals with sequences of numbers. So a natural way to represent such a system is to use difference equation.

In next section, the formulation procedure for discrete-time dynamic model of robot system will be introduced by using some fundamental physical properties such as the generalized momentum equation/energy equation. The discrete-time dynamic model derived through such a procedure guarantees the conservation of a certain invariant properties over each sampling interval, thus satisfies the fundamental principles of classical mechanics.

2.5.1 Discretization Method

The key motive for a dynamic model in discrete-time form is to get a suitable description of dynamic system under computer controlled environments for the forward or inverse dynamic applications.

Conventional approach to discrete-time dynamic model is to derive a difference equation whose solution is an approximation to that of differential equation through a numerical integration method. The topic of numerical integration is not simple, but the most elementary techniques which are based on the selection of the incremental area term of integration, are the forward rectangular rule(also known as forward Euler's rule), the backward rectangular rule, and the trapezoid rule(often Tustin's bilinear

rule)[Franklin 80]. The explicit forward and backward rectangular rules are straightforward to apply. The implicit trapezoid rule is more accurate than the rectangular rules because of taking the approximated integration area to be the average area of the two previous rectangular rules. TABLE 2.5 summarizes the substitution rules of those approximations.

TABLE 2.5 Single Step Numerical Integration Formula

Method	Approximation
Forward Rectangular Rule	$x(k+1) - x(k) = \Delta t\ \dot{x}(k)$
Backward Rectangular Rule	$x(k+1) - x(k) = \Delta t\ \dot{x}(k+1)$
Trapezoid Rule	$x(k+1) - x(k) = \Delta t\ [\dot{x}(k+1) + \dot{x}(k)]/2$

The dynamic equations of typical manipulators are characterized by the coupled, nonlinear second order differential equations as Eq.(2-53). Moreover, the inertia matrix (the coefficient matrix of second derivative vector of joint coordinate) shows that the system is a variable inertia system, i.e., the inertias are changed according to the arm configurations. Hence, the equations of motion are not directly integrable unlike an invariant, decoupled linear equations.

For the concerns of the differential order of equations, it is desirable to formulate the equations such that the second derivative terms with respect to time should not be included in the equations because measuring the acceleration is not usual on typical robot systems. When a robot system has only position sensors, the acceleration information has to be obtained by double-differentiation on the original position data. In this case, the processed data for acceleration could be deteriorated due to a noise in the original data even though a prefilter is applied before differentiation.

One way for the first order equations is to use Hamilton's equation instead of Lagrange's equation[Meirovitch 70]. But, in Hamilton's equation, the auxiliary variables appear in the form of momentum. Therefore, it is not appropriate for the applications because those auxiliary variables are not directly measurable quantities.

A complete solution of n-degree simultaneous differential equations requires 2n constants of integration. These constants may consist either of the initial values of the n coordinates q_i and n velocities $\dot{q}_i$ or of the values of n coordinates at two different time instants. Generally, complete solutions of equations of motion may not be possible, particularly it is impossible for the most of nonlinear differential equations. However, sometimes, the equations of motion exhibits certain peculiarities which furnish informations about its behaviour without actually obtaining the complete solutions of equations. This is the so-called first integrals of motion[Meirovitch 70]. These integrals contain the derivatives of the coordinates of one order lower than the order of the original differential equations. This feature is the foundation upon which the discrete-time dynamic model is developed.

Greenspan redefined the concept of work to fit discrete mechanics rather than forcing the discrete model to obey the principle of conservation of energy. His discrete mechanics is based on two basic assumptions, one is the constant inertia system, and the other is the constant acceleration in each sampling time interval[Greenspan 73]. Tourassis discretized the dynamic equations of the assumed cylindrical type robot model and generalized the replacement rules for discrete-time equations[Tourassis 85].

2.5.2 Discrete-time Dynamic Model of IC DDR

For the derivation of the discrete-time dynamic model of IC DDR, consider some properties of the Lagrangian. There exists a system for which a certain coordinate is absent from the Lagrangian Λ although its time derivative does appear in Λ. Then the corresponding Lagrange's equation is reduced to the form,

$$\frac{d}{dt}\left(\frac{\partial \Lambda}{\partial \dot{q}_i}\right) = Q_i \tag{2-66}$$

here, the quantity $\partial\Lambda/\partial\dot{q}_i$ can be considered as generalized momentum and the coordinate q_i is called ignorable or cyclic variable[Landau 76]. Eq.(2-66) constitutes the first integral of motion and corresponds to the law of the conservation of momentum when the generalized forces Q_i are not acting on the system. Next, turn to the case in which the Lagrangian does not explicitly depend on time, i.e.,

$$\frac{\partial \Lambda}{\partial t} = 0 \tag{2-67}$$

Then, recalling Lagrange's equation Eq.(2-1), the total time derivative of Λ can be written as

$$\frac{\partial \Lambda}{\partial t} = \sum_i \frac{\partial \Lambda}{\partial q_i} \dot{q}_i + \sum_i \frac{\partial \Lambda}{\partial \dot{q}_i} \ddot{q}_i \tag{2-68}$$

Replacing $\partial\Lambda/\partial q_i$, in accordance with Lagrange's equation by $(d/dt)\partial\Lambda/\partial\dot{q}_i - Q_i$ and rearranging gives,

$$\frac{d}{dt}\left(\sum_i \frac{\partial \Lambda}{\partial \dot{q}_i} \dot{q}_i - \Lambda \right) = \sum_i Q_i \dot{q}_i \tag{2-69}$$

The quantity $\{\sum(\partial\Lambda/\partial\dot{q}_i)\,\dot{q}_i - \Lambda\}$ remains constant during the motion when there are not external forces, and is called the energy of the system. This constitutes another integral of motion. A friction model is not included in Eq.(2-66) and Eq.(2-69). If the dissipative friction forces(proportional to velocity) are assumed, the equations can be modified as

$$\frac{d}{dt}\left(\frac{\partial \Lambda}{\partial \dot{q}_i} \right) = Q_i + Q_{di} \tag{2-70a}$$

$$\frac{d}{dt}\left(\sum_i \frac{\partial \Lambda}{\partial \dot{q}_i} \dot{q}_i - \Lambda \right) = \sum_i (Q_i + Q_{di})\dot{q}_i \tag{2-70b}$$

Here, Q_{di} are the generalized dissipative forces due to friction and can be expressed in terms of the generalized coordinates q_1, q_2.

$$Q_{d1} = -[p_4 \dot{q}_1 + p_5(\dot{q}_1 + \dot{q}_2)] \tag{2-71a}$$

$$Q_{d2} = -p_5(\dot{q}_1 + \dot{q}_2) \tag{2-71b}$$

On the other hand, the Lagrangian Λ of the positioning arm of IC DDR is expressed as

$$\Lambda = \frac{1}{2}(p_1 + p_3 + 2p_2 \cos q_2)\dot{q}_1^2 + \frac{1}{2}p_3\dot{q}_2^2 + (p_3 + p_2 \cos q_2)\dot{q}_1\dot{q}_2 \tag{2-72}$$

Since the coordinate q_1 does not appear in Lagrangian, q_1 is cyclic and the generalized momentum is obtained according to Eq.(2-66),

$$\frac{\partial \Lambda}{\partial \dot{q}_1} = (p_1 + p_3 + 2p_2 \cos q_2)\,\dot{q}_1 + (p_3 + p_2 \cos q_2)\,\dot{q}_2 \tag{2-73}$$

then, the direct integrable form of equation is obtained

$$\frac{d}{dt}\left[(p_1 + p_3 + 2p_2 \cos q_2)\,\dot{q}_1 + (p_3 + p_2 \cos q_2)\,\dot{q}_2\right] = Q_1 + Q_{d1} \tag{2-74}$$

Another integrable form of equation is obtained from Eq.(2-70b). In the positioning arm of IC DDR, there is no contribution of potential energy of positioning arm to Lagrangian, so the kinetic energy of the positioning arm is corresponding to the Lagrangian. The another form of equation is,

$$\frac{d}{dt}\left[\frac{1}{2}(p_1 + p_3 + 2p_2 \cos q_2)\,\dot{q}_1^2 + \frac{1}{2}p_3\dot{q}_2^2 + (p_3 + p_2 \cos q_2)\,\dot{q}_1\dot{q}_2\right]$$

$$= \dot{q}_1(Q_1 + Q_{d1}) + \dot{q}_2(Q_2 + Q_{d2}) \tag{2-75}$$

Using Eq.(2-71) and incorporating the motor dynamics Eq.(2-62), Eq.(2-74) and Eq.(2-75) can be recast in terms of actuator angles A_i.

$$\frac{d}{dt}\left[(p_1 + p_2 C_{2\bar{1}})\dot{A}_1 + (p_3 + p_2 C_{2\bar{1}})\dot{A}_2\right] = \eta_1 U_1 + \eta_2 U_2$$
$$- \frac{d}{dt}\left[J_{r1}\dot{A}_1 + J_{r2}\dot{A}_2 + (p_4 + b_1)A_1 + (p_5 + b_2)A_2\right] \tag{2-76a}$$

$$\frac{d}{dt}\left[\frac{1}{2}p_1\dot{A}_1^2 + \frac{1}{2}p_3\dot{A}_2^2 + p_2 C_{2\bar{1}}\dot{A}_1\dot{A}_2\right] = \frac{d}{dt}(\eta_1 U_1 A_1 + \eta_2 U_2 A_2)$$
$$- \frac{d}{dt}\left(\frac{1}{2}J_{r1}\dot{A}_1^2 + \frac{1}{2}J_{r2}\dot{A}_2^2\right) - (p_4 + b_1)\dot{A}_1^2 - (p_5 + b_2)\dot{A}_2^2 \tag{2-76b}$$

The direct integrability of Eq.(2-76) comes from the fact that the inputs U_i are kept constant over a sampling interval. But the right hand side of Eq(2-76b) can not be directly integrated because the involvement of friction model produces the squared terms of velocity. Therefore, it is inevitable to approximate these terms by some

numerical integration scheme. Though a multi-step numerical integration formula leads to higher degree of accuracy than a single step one, those numerical integration formulas produce difference equations with multi-step apart.

Since the joint friction of direct drive arm is much less than that of conventional gearing joints, the energy dissipation terms in Eq.(2-76b) does not make a great sense even though an accurate numerical scheme is adopted to approximate the integration of the indirectly integrable terms. A moderate and reasonable way to solve these indirectly integrable terms is to assume that the average velocity $\{[\dot{A}_i(k+1) + \dot{A}_i(k)]/2\}$ is constant during each sampling interval.

Then, using the parameters defined in Eq.(2-64), the integration of Eq.(2-76) over one sampling time interval leads to the difference equations as follows:

$$M(k+1) - M(k) = \Delta t\ [U_1(k) + U_2(k)] \tag{2-77a}$$

$$\begin{aligned} E(k+1) - E(k) = {} & U_1(k)[A_1(k+1) - A_1(k)] \\ & + U_2(k)[A_2(k+1) - A_2(k)] - E_d(k+1) \end{aligned} \tag{2-77b}$$

where,

$$\begin{aligned} M(k) = {} & [\Theta_1 + \Theta_2 C_{2\bar{1}}(k)]\,\dot{A}_1(k) + [\Theta_3 + \Theta_2 C_{2\bar{1}}(k)]\,\dot{A}_2(k) \\ & + \Theta_4 A_1(k) + \Theta_5 A_2(k) \end{aligned} \tag{2-77c}$$

$$E(k) = \frac{1}{2}\Theta_1 \dot{A}_1^2(k) + \frac{1}{2}\Theta_3 \dot{A}_2^2(k) + \Theta_2 C_{2\bar{1}}(k)\,\dot{A}_1(k)\,\dot{A}_2(k) \tag{2-77d}$$

$$\begin{aligned} & E_d(k+1) \\ & = \Delta t \left\{ \Theta_4 \left[\frac{\dot{A}_1(k+1) + \dot{A}_1(k)}{2} \right]^2 + \Theta_5 \left[\frac{\dot{A}_2(k+1) + \dot{A}_2(k)}{2} \right]^2 \right\} \end{aligned} \tag{2-77e}$$

One remarkable feature of the Eq.(2-77) is that the highest derivatives are the first order derivatives of the coordinates. The above equations can be arranged into an explicit form suitable for inverse dynamic applications. But there are many candidates in constituting the explicit form of the final discrete-time dynamic model.

A symbolic representation for the explicit form of discrete-time dynamic equation could be expressed as

$$\mathbf{U}(k) = \mathbf{H}(k+1)\dot{\mathbf{A}}(k+1) - \mathbf{H}(k)\dot{\mathbf{A}}(k) + \mathbf{C}[\mathbf{A}(k), \mathbf{A}(k+1), \dot{\mathbf{A}}(k), \dot{\mathbf{A}}(k+1)] + \mathbf{D}[\dot{\mathbf{A}}(k), \dot{\mathbf{A}}(k+1)] \tag{2-78}$$

here,

$$\mathbf{U}(k) = \begin{pmatrix} U_1(k) \\ U_2(k) \end{pmatrix} \quad \text{and} \quad \mathbf{A}(k) = \begin{pmatrix} A_1(k) \\ A_2(k) \end{pmatrix}$$

The first two terms in Eq.(2-78) are the change of total generalized momentum over one sampling period. The third term represents the centrifugal and Coriolis effects acting on the joint. The last is the dissipation term acting on the joint due to friction. If Eq(2-77) is solved for U_1, U_2 as algebraic equations, the inertial coefficient matrix **H** becomes dependent on **A** as well as $\dot{\mathbf{A}}$. It is not desirable that the inertial coefficient matrix **H** is dependent on velocities. As inspired by the inertial coefficient matrix in continuous-time equation Eq.(2-65), it can be reasonably inferred that the discrete inertial coefficient matrix should be dependent on the angular position vector **A** not on the angular velocity vector $\dot{\mathbf{A}}$, i.e., $\mathbf{H}(k+1)=\mathbf{H}[\mathbf{A}(k+1)]$. However, this form can not be achieved without some treatment on the squared terms of velocities in Eq.(2-77d). Since the energy difference equation Eq.(2-77b) produces the difference of the squared velocities at adjacent sampling instants, i.e., $[\dot{A}_i^2(k+1) - \dot{A}_i^2(k)]$, an introduction of the trapezoid rule in TABLE 2.5 could replace this term into $2[\dot{A}_i(k+1) - \dot{A}_i(k)][A_i(k+1) - A_i(k)]/\Delta t$. Then, after rearranging Eq.(2-77), the final form of the discrete-time dynamic model is obtained as follows.

$$\begin{pmatrix} U_1(k) \\ U_2(k) \end{pmatrix} = \frac{1}{\Delta t}\begin{bmatrix} \Theta_1 & \Theta_2 C_{2\bar{1}}(k+1) \\ \Theta_2 C_{2\bar{1}}(k+1) & \Theta_3 \end{bmatrix} \begin{pmatrix} \dot{A}_1(k+1) \\ \dot{A}_2(k+1) \end{pmatrix}$$

$$- \frac{1}{\Delta t}\begin{bmatrix} \Theta_1 & \Theta_2 C_{2\bar{1}}(k) \\ \Theta_2 C_{2\bar{1}}(k) & \Theta_3 \end{bmatrix} \begin{pmatrix} \dot{A}_1(k) \\ \dot{A}_2(k) \end{pmatrix}$$

$$+ \frac{1}{\{[A_1(k+1) - A_1(k)] - [A_2(k+1) - A_2(k)]\}} \times$$

$$\begin{bmatrix} -\frac{1}{2}\Theta_2\{\dot{A}_1(k+1)\dot{A}_2(k) + \dot{A}_1(k)\dot{A}_2(k+1)\}\{C_{2\bar{1}}(k+1) - C_{2\bar{1}}(k)\} \\ +\frac{1}{2}\Theta_2\{\dot{A}_1(k+1)\dot{A}_2(k) + \dot{A}_1(k)\dot{A}_2(k+1)\}\{C_{2\bar{1}}(k+1) - C_{2\bar{1}}(k)\} \end{bmatrix}$$

$$+\begin{bmatrix} \Theta_4\left\{\dfrac{\dot{A}_1(k+1) + \dot{A}_1(k)}{2}\right\} \\ \Theta_5\left\{\dfrac{\dot{A}_2(k+1) + \dot{A}_2(k)}{2}\right\} \end{bmatrix} \tag{2-79}$$

The resulting discrete-time dynamic equation becomes the first order difference equation in which second derivatives of coordinates are not included.

The most noteworthy property in Eq.(2-79) is that this discrete-time dynamic model takes account of the change of the inertia matrix over a sampling interval. This compact form of discrete-time dynamic model is quite suitable for trajectory planning and control.

3. PARAMETER IDENTIFICATION

3.1 Introduction

The elements of system identification are selection of model, experimental measurements and parameter estimation. The mathematical description of a system can be developed along two routes. One relies on the well established physical laws, and the other is based on assumed model and experiment. In Chapter 2, the dynamic models are derived in the forms of continuous-time/discrete-time dynamic equations based on the rigid body mechanics.

After a model structure is fixed, the adjustable parameters of the model determine the model accuracy. The experimental measurements of input-output data are used to estimate the parameters. At this stage, some prior knowledge about parameters can be combined. An input for exciting the system dynamics should be selected so that the input-output data from experimental measurements become informative. For a robot system, the input is the excitation signal to the actuator, and the position/velocity information of each joint is obtained as output data.

The model based control strategies such as the computed torque technique[Khosla 86a /An 87], decoupling method using nonlinear state feedback[Fournier 84], and resolved acceleration control[Wampler 88] incorporate a dynamic model in their control law to produce a desired motion by the computation of actuating torques/forces. The quality of control and the resulted accuracy of path following are highly dependent on the parameter values under these control schemes.

The mass, center of mass, inertia, kinematic specifications between joints, and the specifications of actuators constitute these parameters. In most cases, the basic physical quantities are combined to construct the lumped parameters of dynamic model. Determining those basic physical quantities from measurements is not easy. For the forward / inverse dynamic applications, identifying all the basic physical parameters(quantities) is not necessarily required. What matters is that rather than the

basic physical parameters, the lumped parameters (combined parameters of the basic physical quantities) can be estimated well enough to reproduce the input or output by using the suggested model with its identified parameters. If a robot is working under highly unknown and varying payload circumstances, there is a need to identify the mass and the inertial property of the payload on real-time to achieve precise trajectory following .

Previous work in the parameter estimation was mainly done on the identifying the mass of the payload and estimating the basic physical quantities such as inertias of link, mass, etc. Paul described two techniques of determining the mass of a load using the relationship between the steady state torque error and the steady state servo error[Paul 82]. In his first technique, an estimated mass was obtained from the information of joint torque/force, and the second technique determines the mass of an unknown load through the observation of torque/force at the wrist. But the centre of mass and the moment of inertia were not identified. Coiffet utilized the special test torques, and moved only one axis at a time to estimate the moment of inertia of a payload. He extended Paul's technique to estimate the centre of mass of a payload when the robot is at rest[Coiffet 83]. Mukerjee allowed general motion during load identification [Mukerjee 85]. His method requires full torque/force sensing which seems impractical, moreover, was not verified by implementation. Atkeson[Atkeson 85], An[An 86] developed an approach similar to Mukerjee's work using wrist torque/force sensor to estimate the inertial characteristics of a payload. Their approach was extended to identify the inertial parameters of all the links of a robot. Particularly, Atkeson tested his algorithm by an actual implementation on a PUMA 600 robot equipped with a torque/force sensor. Khosla modified the Newton-Euler formulation so that it becomes linear in parameters and developed both on-line and off-line parameter estimation procedures from the measurements of its actuating input torque and joint outputs[Khosla 85]. Craig presented an adaptive scheme[Craig 88]. His adaptive law was derived on the basis of Liapunov stability criteria. That adaptive law updated parameter estimates using the function of filtered servo error signal. Under this scheme, the servo error may go to zero, but the parameters are not guaranteed to go to their true values.

3.2 Parameter Calculation From Design Data

The aim of this section is to give reasonable values of parameters from engineering drawing data and design specifications. They may not reflect accurate values of parameters due to the approximation error of simplified model shape. However, these can serve as a good point of comparison for the estimated parameters by an identification experiment. Particularly, they can provide the dynamic model with appropriate values of parameters for simulation experiments which are important in testing the suggested estimation algorithm and control strategy. Thus, the calculation of parameters by no means carries little weight before determining these parameters from actual experimental measurements.

The moving parts of IC DDR are grouped into three sub assemblies according to their mechanical structures. The structures of subassemblies and their simplified constituent elements are described in [Kim 90]. 26 components are taken into account to build the whole of the moving parts. Each subassembly is modelled as a composite of hollow and solid cylinders, and rectangular parallelepipeds to approximate the constituent original shapes of its components. The resultant masses and inertias are also presented in [Kim 90]. All the parameters which are needed to describe the dynamic model are shown in Table 3.1.

TABLE 3.1 Calculated Parameters From Design Data

Description	Symbol	Unit	Value
Inertia of drive shaft assembly (shoulder arm)	J_{r1}	Kgm^2	1.37×10^{-2}
Inertia of drive shaft assembly (elbow arm)	J_{r2}	Kgm^2	6.21×10^{-3}
Mass (shoulder arm)	m_1	Kg	2.54
Mass (elbow arm)	m_2	Kg	5.13
Inertia of shoulder arm (to coordinate frame 1)	I_1	Kgm^2	0.131
Inertia of elbow arm (to coordinate frame 2)	I_2	Kgm^2	0.199
Center of Mass (shoulder arm)	l_1	m	0.175
Center of Mass (elbow arm)	l_2	m	0.131
Distance between joints	L	m	0.32

3.3 Parameter Identifier Structures

The target of parameter identification in this section is to estimate the lumped parameters defined in Eq.(2-64) rather than the elementary physical quantities.

The parameter identifier structures are obtained from the dynamic model of robot system. The previous work on the parameter identification of robot system requires the measurements of torque/force directly or to calculate them indirectly[Atkeson 85, An 85, Khosla 85]. Since the mathematical model of the motor system is incorporated in the dynamic equations, a numerical value assigned to the control device, the switching amplifier of the motor, is the input signal .

For the purpose of parameter identification, five identifier structures are suggested in this section. The first one comes from the continuous-time dynamic equations. The second structure is obtained from the continuous-time dynamic model but substitutes the higher order derivatives through numerical approximation rule. Other two come from the generalized energy difference and momentum difference equations(Eq.2-77a,b). Finally, the velocity dependent discrete-time dynamic model(Eq.2-79) is recast into the parameter identifier structure.

<u>Identifier Structure Using Continuous-time Dynamic Model (SC) :</u>

The continuous-time differential equations, Eq.(2-65) can be used directly to estimate the unknown parameters. Since the information about the dynamic responses are obtained only at sampling instants, the estimation model is expressed linear in parameters using the sampled values of input/output.

$$U_1(k) = \Theta_1 \ddot{A}_1(k) + \Theta_2[C_{2\bar{1}}(k)\ddot{A}_2(k) - S_{2\bar{1}}(k)\dot{A}_2^2(k)] + \Theta_4 \dot{A}_1(k) \qquad (3\text{-}1a)$$

$$U_2(k) = \Theta_2[C_{2\bar{1}}(k)\ddot{A}_1(k) + S_{2\bar{1}}(k)\dot{A}_1^2(k)] + \Theta_3 \ddot{A}_2(k) + \Theta_5 \dot{A}_2(k) \qquad (3\text{-}1b)$$

When using Eq.(3-1), it is immaterial whether the time dependent coefficients of Θ_i are linear or nonlinear functions of the joint responses. What matters here is that those coefficients of Θ_i are the known quantities at every sampling instant time, $k\Delta t$. Since

Eq.(3-1) have the property of being linear in the unknown parameters Θ_i, the well-known least square method can be applied for estimation.

Generally, most of robot systems are not equipped with acceleration sensors. Thus, to obtain the acceleration data, the joint velocities have to be differentiated when tachometers are installed, or the joint angles must be double-differentiated by a proper differentiating filter. If a noise is involved in its original data, differentiated signal can be distorted, particularly, it may be worse when double-differentiated even though prefiltering is applied to eliminate noise. Therefore, the requirement for acceleration information could become a disadvantage in using Eq.(3-1).

Identifier Structure Using Forward Euler Rule To Acceleration (**GD**) :

The identifier structure where the acceleration is not required seems to have a great merit compared to the previous identifier structure. The next identifier structure is inspired by Greenspan's discrete mechanics[Greenspan 73]. One realization of his discrete mechanics is to adopt the forward Euler rule for the velocity as in TABLE 2.4. But, some numerical approximation error is inevitably involved in this derivation. Applying the forward Euler rule to the joint angular velocity gives the replacement formula for the joint angular acceleration at each sampling instant as following,

$$\ddot{A}_i(k) = \frac{\dot{A}_i(k+1) - \dot{A}_i(k)}{\Delta t} \tag{3-2}$$

Substituting Eq.(3-2) into Eq.(2-65) gives

$$\begin{aligned} U_1(k) = {} & \Theta_1[\dot{A}_1(k+1) - \dot{A}_1(k)]/\Delta t + \Theta_2 C_{2\bar{1}}(k)[\dot{A}_2(k+1) - \dot{A}_2(k)]/\Delta t \\ & - \Theta_2 S_{2\bar{1}}(k)\dot{A}_2^2(k) + \Theta_4 \dot{A}_1(k) \end{aligned} \tag{3-3a}$$

$$\begin{aligned} U_2(k) = {} & \Theta_2 C_{2\bar{1}}(k)[\dot{A}_1(k+1) - \dot{A}_1(k)]/\Delta t + \Theta_3[\dot{A}_2(k+1) - \dot{A}_2(k)]/\Delta t \\ & + \Theta_2 S_{2\bar{1}}(k)\dot{A}_1^2(k) + \Theta_5 \dot{A}_2(k) \end{aligned} \tag{3-3b}$$

Eq.(3-3) with the trapezoid rule for the angular displacements forms other discrete-time dynamic model. Assuming that the coordinate-dependent inertias and the acceleration remain constant during each sampling period, the discrete-time Eq.(3-3) can be a

reasonable dynamic model. But another type of discrete-time dynamic model through the first integral of motion which is introduced in section 2.5.2 does not have such an assumption in the derivation. Therefore, the coordinate dependent inertia terms which are varied during the sampling period are positively accounted in the latter derivation procedure. After rearranging Eq.(3-3), another identifier structure which is free from acceleration data is obtained.

$$U_1(k) = \Theta_1[\dot{A}_1(k+1) - \dot{A}_1(k)]/\Delta t + \Theta_2\{C_{2\bar{1}}(k)[\dot{A}_2(k+1) - \dot{A}_2(k)]/\Delta t - S_{2\bar{1}}(k)\dot{A}_2^2(k)\} + \Theta_4\dot{A}_1(k) \tag{3-4a}$$

$$U_2(k) = \Theta_2\{C_{2\bar{1}}(k)[\dot{A}_1(k+1) - \dot{A}_1(k)]/\Delta t + S_{2\bar{1}}(k)\dot{A}_1^2(k)\} + \Theta_3[\dot{A}_2(k+1) - \dot{A}_2(k)]/\Delta t + \Theta_5\dot{A}_2(k) \tag{3-4b}$$

As was mentioned, the form free from acceleration looks a good feature, but a disadvantage is an error caused by numerical approximation for acceleration term that could degrade the accuracy of parameter estimation.

Identifier Structures From Momentum and Energy Difference Equation (**MD, ED**) :

Other two identifier structures for parameter estimation are the generalized momentum and energy difference equations, which are developed by direct integration of the first integral of motion. These two difference equations can be rearranged into suitable forms for parameter identification as follows:

Momentum Difference Model (**MD**) :

$$\begin{aligned}\Delta t\,[U_1(k) + U_2(k)] = {} & \Theta_1[\dot{A}_1(k+1) - \dot{A}_1(k)] + \\ & \Theta_2\{C_{2\bar{1}}(k+1)[\dot{A}_1(k+1) + \dot{A}_2(k+1)] - \\ & \quad C_{2\bar{1}}(k)[\dot{A}_1(k) + \dot{A}_2(k)]\} + \\ & \Theta_3[\dot{A}_2(k+1) - \dot{A}_2(k)] + \Theta_4[A_1(k+1) - A_1(k)] + \\ & \Theta_5[A_2(k+1) - A_2(k)]\end{aligned} \tag{3-5}$$

Energy Difference Model (**ED**) :

$$
\begin{aligned}
U_1(k)[A_1(k+1)-A_1(k)]+U_2(k)[A_2(k+1)-A_2(k)]= \\
&\Theta_1[\dot{A}_1^2(k+1)-\dot{A}_1^2(k)]/2+ \\
&\Theta_2[C_{2\bar{1}}(k+1)\dot{A}_1(k+1)\dot{A}_2(k+1)-C_{2\bar{1}}(k)\dot{A}_1(k)\dot{A}_2(k)]+ \\
&\Theta_3[\dot{A}_2^2(k+1)-\dot{A}_2^2(k)]/2+\Theta_4[\dot{A}_1^2(k+1)+\dot{A}_1^2(k)]\Delta t/2+ \\
&\Theta_5[\dot{A}_2^2(k+1)+\dot{A}_2^2(k)]\Delta t/2
\end{aligned}
\tag{3-6}
$$

Interestingly enough, the above identifier structures comprise all the parameters in their single equation respectively and furthermore accelerations are not needed. Any numerical approximation is not involved in the derivation of the generalized momentum difference equation. On the other hand, the generalized energy difference equation is derived by applying numerical integration to the dissipation term of friction. If the contributions of these dissipation terms are not great(this is true for the IC DDR), the error due to numerical approximation might not have a great influence on the parameter estimation performances. This effect will be investigated in the following simulation experiments.

<u>Identifier Structure From Velocity Dependent Discrete Dynamic Model (**VD**) :</u>

The last candidate is the velocity dependent discrete-time model (Eq.2-79). This candidate is not expected to show the best performances of parameter estimation as several stages of numerical approximation are applied to reach its final form. However, it is interesting and important to know whether their constituent parameters sustain their original meaning as defined in Eq.(2-64) in spite of several numerical approximations. Eq.(2-79) can be easily recast into the linear form in parameters. For brevity, their detailed equations will be omitted here because they are described fully in section 2.5.2. The identifier structures exploited for parameter estimation are listed in TABLE 3.2.

TABLE 3.2 Parameter Identifier Structure

Symbol	Eq.	Type	States Required for Estimation
SC	3-1	Sampled continuous-time Eq.	Acceleration, Velocity, Position
GD	3-3	continuous-time Eq. approximated by Forward Euler Rule	Velocity, Position
MD	3-5	Momentum Difference Eq.	Velocity, Position
ED	3-6	Energy Difference Eq.	Velocity, Position
VD	2-79	Velocity Dependent discrete Model	Velocity, Position

3.4 Estimation Method

The key issue in the previous section is how to choose the functions of nonlinear regression variables. The resulting identifier structures for parameter estimation can be regarded as a finite dimensional parametrization with nonlinear regression variables which can be computed from the observed data, and have the property of being linear in the unknown parameters.

Such a structure is called a linear regression and can be represented in a simple form of equation:

$$\hat{y}(k) = \Phi_1(k)\Theta_1 + \Phi_2(k)\Theta_2 + \quad \dots \quad + \Phi_n(k)\Theta_n = \Phi^T(k)\Theta \qquad (3\text{-}7)$$

Here, the sampling index k represents time. $\Phi(k)$ is a known regression vector and Θ is unknown parameter vector. This form is of importance since powerful and simple estimation methods can be applied for the determination of unknown parameters.

The first formulation and solution of an identification problem were given by Gauss in his famous determination of the orbits of planets. Gauss formulated the identification problem as an optimization problem and introduced the principle of least square, a method based on the minimization of the sum of the square of the error. This criterion has two advantages. First, large errors are heavily penalized: an error twice as large is

four times as bad. This usually accords with common sense, but there are exceptions. For instance, if a few observations are very poor or totally spurious misreadings, the best thing may be to ignore them altogether. The other advantage is mathematical tractability. The formula giving the least square estimates is obtained by simple matrix algebra, and the estimates are computed as the solution of a set of linear equation.

All the identifier structures derived for parameter estimation in section 3.3 have the form of linear regression in the unknown parameters Θ_i. Except for the first two models(**MD**, **ED**) in TABLE 3.2, **y**(k) in Eq.(3-7) becomes a vector of two simultaneous observations instead of a scalar. Thus two simultaneous linear equations have to be used for identification. Then vector Θ and matrix $\Phi^T(k)[2\times5]$ has to be formed as follows:

$$\hat{y}(k) = \Phi^T(k)\Theta + e(k) \tag{3-8}$$

where **e**(k) accounts for observation error (measurement noise) and modelling error, since even without observation error few models are perfect. The aim is to find the value $\hat{\Theta}$ of Θ which minimizes

$$J = \sum_{k=1}^{N} e^T(k)e(k) \tag{3-9}$$

To make the algebra tidy, collect all the sample vector **y**(1) to **y**(N) into a 2N-dimensional vector **Y**, all the matrices $\Phi^T(i)$ into $[2N\times5]$ matrix Ψ and all the $e^T(i)$ into 2N-dimensional vector **E**, giving

$$\mathbf{Y} = \Psi\Theta \quad + \mathbf{E} \tag{3-10}$$

and

$$J = \mathbf{E}^T\mathbf{E} = (\mathbf{Y}^T - \Theta^T\Psi^T)\,(\mathbf{Y} - \Psi\Theta) \tag{3-11}$$

Then the gradient of J with respect to Θ is

$$J_\theta = \left[\frac{\partial J}{\partial \theta_i}\right] = -2\Psi^T Y + 2\Psi^T\Psi\Theta \tag{3-12}$$

Therefore the Θ that makes the gradient of J zero is

$$\Psi^T\Psi\ \hat{\Theta} = \Psi^T Y \tag{3-13}$$

These equation is called the normal equation. If $\Psi^T\Psi$ is nonsingular, the explicit solution is written

$$\hat{\Theta} = \left(\Psi^T\Psi\right)^{-1}\Psi^T Y \tag{3-14}$$

The $\hat{\Theta}$ given by Eq.(3-14) is called the ordinary least square estimate of Θ. If none of the regressors(regression variables) are totally redundant by being a linear combination of the others at every sample data, the normal matrix $\Psi^T\Psi$ is positive definite, and the inverse of $\Psi^T\Psi$ exists. But it does not mean that the inverse is easy to compute accurately.

Even though the normal matrix is symmetric and positive definite, there are other efficient methods of solving sets of linear equations than general matrix inversion method. When the normal matrix is near singular, computing its inverse would be ill-conditioned. Ill-conditioning may prevent a satisfactory solution of the normal equations. This happens when at least one regressor is close to being linearly dependent on the other regressors. Ill-conditioning can cause Ψ to lose rank and consequently $\Psi^T\Psi$ to become singular. There are several ways to resolve this problem such as singular-value decomposition and ridge regression, etc[Norton 86].

The least square calculation for $\hat{\Theta}$ in Eq.(3-14) is referred to as a batch(off-line) calculation, since all the observations of y(k) and Φ(k), from which **Y** and Ψ are composed, are processed simultaneously and a single estimation of the parameter vector is produced. By contrast, recursive method processes one observation at a time and update the parameter estimates as more data become available. Recursive method allows monitoring of output or equation errors at each iteration. Hence, some isolated errors due to obscuring observations or a persistent drift in one or more of the parameters can be detected.

For the derivation of a recursive method, consider first the normal matrix,

$$\Psi^T(k+1)\Psi(k+1) = \Psi^T(k)\Psi(k) + \Phi(k+1)\Phi^T(k+1) \tag{3-15}$$

Define,

$$\mathbf{p}(k+1) = [\Phi^T(k+1)\Phi(k+1)]^{-1} \tag{3-16}$$

then from Eq.(3-15),

$$\mathbf{p}(k+1) = [\mathbf{p}^{-1}(k) + \Phi(k+1)\Phi^T(k+1)]^{-1} \tag{3-17}$$

By using matrix inversion lemma to Eq.(3-17), the next is obtained.

$$\mathbf{p}(k+1) = \mathbf{p}(k) - \mathbf{p}(k)\Phi(k+1) \left[\mathbf{I} + \Phi^T(k+1)\mathbf{p}(k)\Phi(k+1)\right]^{-1} \Phi^T(k+1)\mathbf{p}(k) \tag{3-18}$$

On the other hand, $\Psi^T(k+1)\mathbf{Y}(k+1)$ can be written as

$$\Psi^T(k+1)\mathbf{Y}(k+1) = \Psi^T(k)\mathbf{Y}(k) + \Phi^T(k+1)\mathbf{y}(k+1) \tag{3-19}$$

Now, substituting Eq.(3-18) and (3-19) into (3-14) forms the recursive equations as followings:

$$\hat{\Theta}(k+1) = \hat{\Theta}(k) + \mathbf{L}(k+1)[\mathbf{y}(k+1) - \Phi^T(k+1)\hat{\Theta}(k)] \tag{3-20a}$$

where, $$\mathbf{L}(k+1) = \mathbf{p}(k)\Phi(k+1)[\mathbf{I} + \Phi^T(k+1)\mathbf{p}(k)\Phi(k+1)]^{-1} \tag{3-20b}$$

$$\mathbf{p}(k+1) = [\mathbf{I} - \mathbf{L}(k+1)\Phi^T(k+1)]\mathbf{p}(k) \tag{3-20c}$$

Eq.(3-20) has a strong intuitive appeal. The new estimate $\hat{\Theta}(k+1)$ is obtained by adding a correction to the previous estimate $\hat{\Theta}(k)$. The correction is proportional to $[\mathbf{y}(k+1) - \Phi^T(k+1)\hat{\Theta}(k)]$, where the last term can be interpreted as the value of **y** at time k+1 predicted by the identifier structure. The correction term is thus proportional to the difference between the measured value of **y**(k+1) and the prediction

of $\mathbf{y}(k+1)$ based on the previous estimates of the parameters. The elements of the matrix $\mathbf{L}(k)$ are weighting factors that tell how the correction and the previous estimate should be combined.

Eq.(3-20) constitutes an algorithm for computing recursively. To use the recursive algorithm, initial values for $\mathbf{p}(k_0)$ and $\hat{\Theta}(k_0)$ are required. Since the matrix $\mathbf{p}(k)$ is defined as $[\Psi^T(k)\Psi(k)]^{-1}$, a possible solution to this could be to collect a sufficient number of initial data such that $\Psi^T(k_0)\Psi(k_0)$ has become invertible, and determine $\mathbf{p}(k_0)$, $\hat{\Theta}(k_0)$ using the batch identification algorithm(Eq.3-14). However, it is convenient to use the recursive equations from the start. An alternative way is to select $\mathbf{p}(0)=\mathbf{p}_0$(positive definite) and $\hat{\Theta}(0) = \Theta_0$ in Eq.(3-20). This gives [Ljung 87],

$$\mathbf{p}(k) = [\mathbf{p}_0^{-1} + \Psi^T(k)\,\Psi(k)]^{-1} \tag{3-21a}$$

$$\hat{\Theta}(k) = [\mathbf{p}_0^{-1} + \Psi^T(k)\,\Psi(k)]^{-1}\ [\mathbf{p}_0^{-1}\,\Theta_0 + \Psi^T(k)\,\mathbf{Y}(k)] \tag{3-21b}$$

Clearly, if $\mathbf{p}_0$ is selected to be large enough, the difference between Eq.(3-21b) and Eq.(3-14) is insignificant, and generally the larger $\mathbf{p}_0$ the smaller the influence of Θ_0 [Norton 86].

3.5 Simulation Experiments

In this section, the various problems related to the parameter estimation will be addressed through simulation experiments.

The continuous-time dynamic model (Eq. 2-65) is used to simulate the positioning mechanism of the IC DDR. For high accuracy requirement on the numerical integration of the differential equations, the variable-order, variable-step Adams routine[NAG 88] is used. This routine adjusts the step length automatically to meet prespecified accuracy tolerances without any serious accumulation of error over a long range of integration.

The important facts before going into details is the reasonable choice of the parameter values for simulation purpose. If the parameter values are determined inappropriately, unrealistic dynamic responses may be obtained from the numerical integration. For example, if the friction coefficients Θ_4 and Θ_5 are set to be excessive, then the numerical solution will contains rapidly decaying transient terms which is unrealistic especially for direct drive robots.

Hence the calculated parameters from the design data listed in TABLE 3.1 are very useful to figure out reasonable parameter values. The friction parameters p_4, p_5 are determined from the assumption that the dissipated energy due to friction model is 5 % of the total given energy over a simple accelerated motion[Kim 90]. The amplifier gain K_a is computed from the ratio of the assumed range of input values U_i to the maximum permissible design voltage(100 V) of the motor used. The parameter values Θ_i used throughout simulation experiments are listed in TABLE 3.3.

The sampling period for simulation is chosen to be 5 [msec], and the total simulation time is 10 seconds. During each sampling period, the inputs U_i are kept constant and the integration routine adjusts its internal step size until meeting the prescribed accuracy requirements.

TABLE 3.3 Parameter Values For Simulation Experiments

Symbol	Value
† K_a	3.26×10^{-3}
$C_h(=K_t K_a / R)$	3.03×10^{-3}
Θ_1	213
Θ_2	102
Θ_3	99
Θ_4	877
Θ_5	850

† assumed range of input values = $-30720 \le U_i \le 30720$

3.5.1 Effectiveness of Various Identifier Structures

The starting point of parameter identification is to investigate the appropriateness of the suggested identifier structures.

Simple square wave type of actuating input (1 Hz) is applied to each joint (excitation level 30 % for joint 1 and 15 % for joint 2). The dynamic responses are sampled at the sampling rate of 200 [Hz] during 5 [sec] movement under these excitation inputs.

The stepwise estimation performances of the suggested identifier structures using these dynamic responses are displayed from Fig.3.1a to Fig.3.1e. All the initial values of parameters are set to zero. When a new data is added, estimated values are updated through the recursive estimation algorithm. The graphs are plotted up to only 200 sampled points since all estimates are quite stable after this number of samples.

As can be seen clearly from Fig.3.1a to Fig.3.1e, the identifier structure **MD**, **SC** show their excellent ability to trace the perfect values of parameters. What is noteworthy here is that acceleration data is not needed for estimation in **MD** structure, while **SC** structure requires acceleration information for estimation. On the other hand, **ED** structure exhibits some error in estimation. This error is caused by the numerical approximation involved in the integration procedure on the friction terms of **ED** structure. But the errors in estimation are negligibly small as shown in TABLE 3.4(second row). The other identifier structures (**GD**, **VD**) shows the relatively larger estimation errors compared to the **ED** structure (TABLE 3.4). This can be easily understood because numerical approximations are more incorporated in the derivation of **GD** and **VD** structure than the **ED** model.

The resulting estimation values of parameters by the various identifier structures are summarized in TABLE 3.4. These estimates are obtained after 1000 samples.

TABLE 3.4 Estimates By the Various Identifier Structures

model	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	213.0	102.0	99.0	877.0	850.0
ED	213.11	101.96	98.87	876.89	849.90
SC	213.0	102.0	99.0	877.0	850.0
GD	203.57	90.83	93.29	930.96	884.03
VD	203.19	91.83	93.60	932.60	876.35

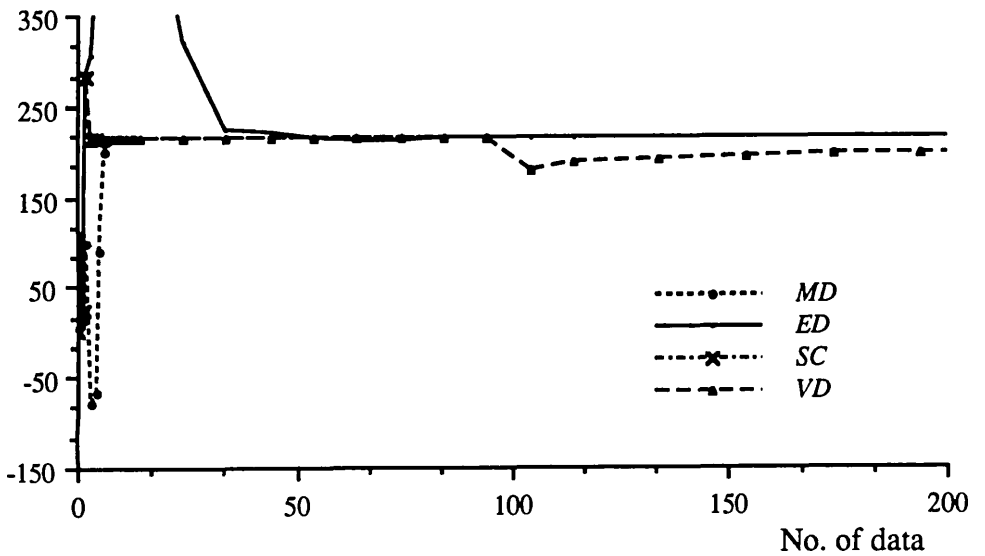

Figure 3.1a Stepwise Estimation of Parameter θ_1

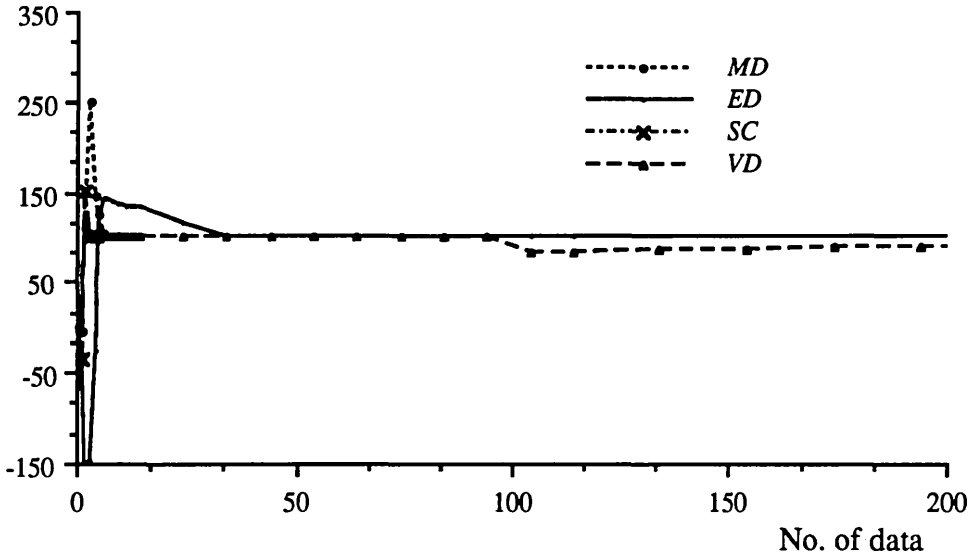

Figure 3.1b Stepwise Estimation of Parameter θ_2

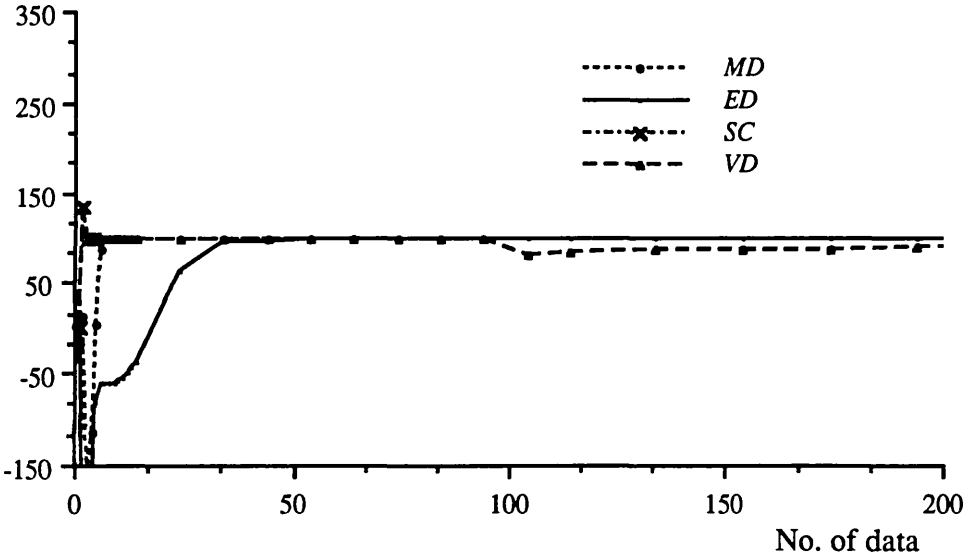

Figure 3.1c Stepwise Estimation of Parameter θ_3

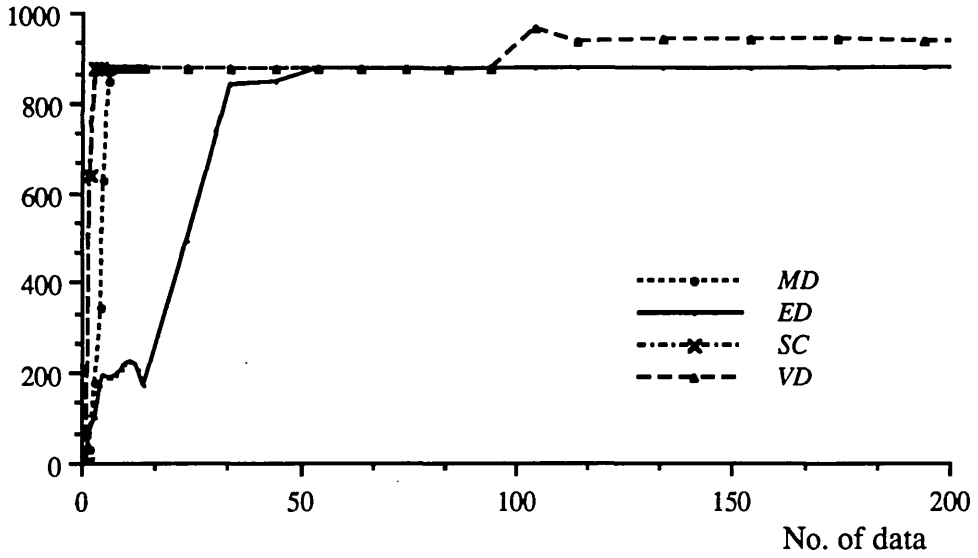

Figure 3.1d Stepwise Estimation of Parameter θ_4

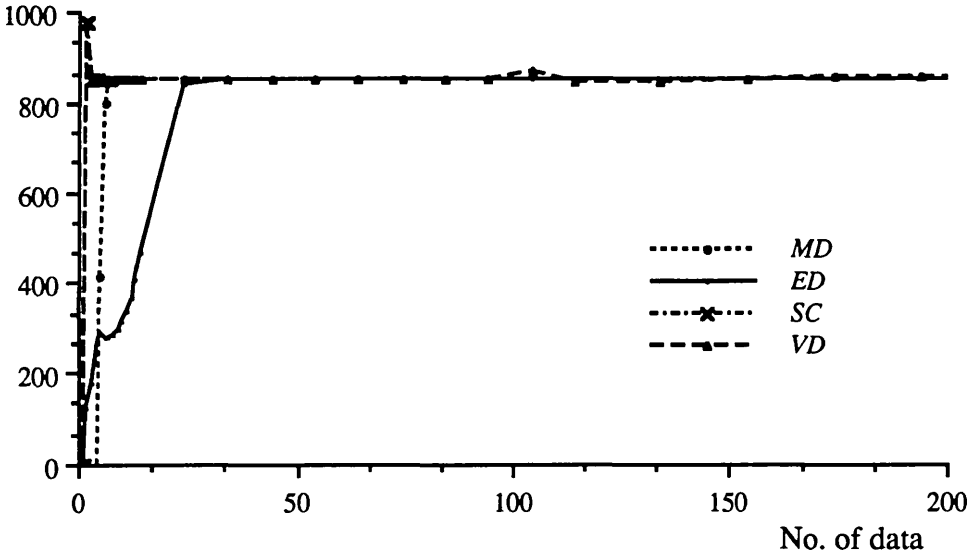

Figure 3.1e Stepwise Estimation of Parameter θ_5

3.5.2 Effect of Excitation Input

The test in this section is designed to examine the suitability of excitation input for parameter estimation purpose. As can be seen in the previous section, any specific planed movement is not required to get informative data for parameter identification . But a 'poor' excitation may fail to develop the dynamic responses which are informative for the identifier structures. The informative data sets for identification are closely related to the 'generally enough input' conditions. Clearly, an inappropriate excitation input may lead to a poorly developed dynamic response which is not informative enough for parameter identification.

Four types of excitation input are tested in this section and their shapes are presented in Fig. 3.2. Utilizing the different type of excitation input, the effect on the estimation performance of each identifier structure is investigated. The estimated parameter values of each identifier structure are presented from TABLE 3.5 to TABLE 3.9 according to the excitation input types after 1000 data points have been collected. Interestingly, the identifier model **MD** and **SC** give perfect estimation regardless of input excitation types (TABLE 3.5, TABLE 3.7). But for the identifier structure **ED**, the estimated parameter values (column 3 in TABLE 3.6)by the excitation type III (pulse train) shows slightly bigger error than those by the other types of excitation inputs. Particularly, the estimation error by the excitation type III is much more increased in the identifier structure **GD** and **VD** (Column 3 in TABLE 3.8 and 3.9). This poor estimation performance by excitation input type III is due to a weakly developed dynamic response which is not informative enough for identifier structure **ED**, **GD** and **VD**. But the excitation type I, II and IV can be generally accepted as proper excitation inputs. The resulting estimation values given in TABLE 3.5 ~ 3.9 are based on the ideal dynamic response data on which a realistic considerations such as measurement noise is not reflected. The influence of noise on parameter estimation will be discussed in the next section.

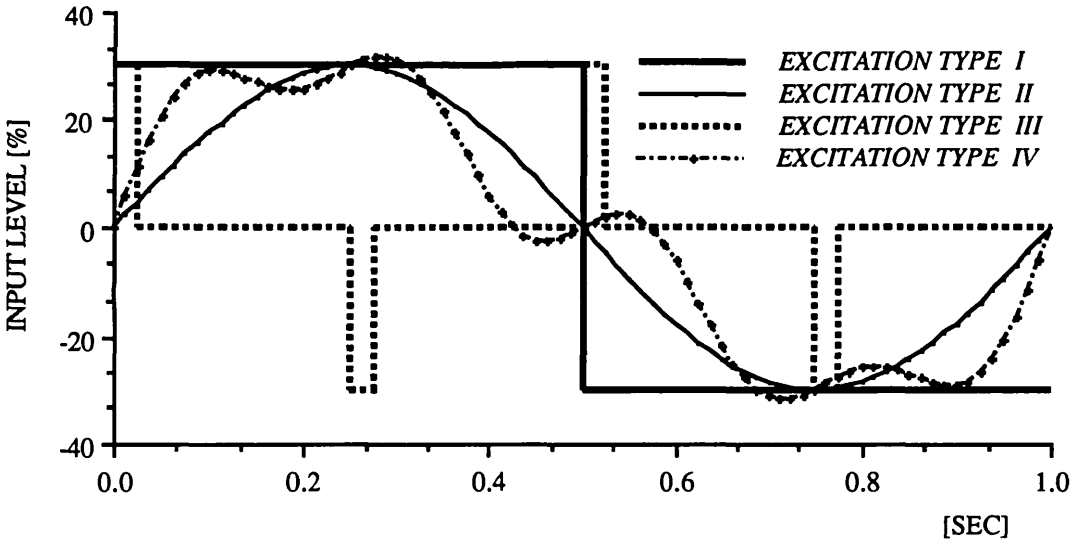

Figure 3.2 Types of Excitation Input

TABLE 3.5 Estimation By Identifier **MD**

Parameters	Excitation Type I	Excitation Type II	Excitation Type III	Excitation Type IV
$\hat{\Theta}_1$	213.0	213.0	213.0	213.0
$\hat{\Theta}_2$	102.0	102.0	102.0	102.0
$\hat{\Theta}_3$	99.0	99.0	99.0	99.0
$\hat{\Theta}_4$	877.0	877.0	877.0	877.0
$\hat{\Theta}_5$	850.0	850.0	850.0	850.0

TABLE 3.6 Estimation By Identifier **ED**

Parameters	Excitation Type I	Excitation Type II	Excitation Type III	Excitation Type IV
$\hat{\Theta}_1$	213.11	213.03	216.68	213.03
$\hat{\Theta}_2$	101.96	101.99	101.69	101.99
$\hat{\Theta}_3$	98.87	99.04	96.64	99.04
$\hat{\Theta}_4$	876.89	876.93	857.03	876.91
$\hat{\Theta}_5$	849.90	850.14	849.75	850.18

TABLE 3.7 Estimation By Identifier **SC**

Parameters	Excitation Type I	Excitation Type II	Excitation Type III	Excitation Type IV
$\hat{\Theta}_1$	213.0	213.0	213.0	213.0
$\hat{\Theta}_2$	102.0	102.0	102.0	102.0
$\hat{\Theta}_3$	99.0	99.0	99.0	99.0
$\hat{\Theta}_4$	877.0	877.0	877.0	877.0
$\hat{\Theta}_5$	850.0	850.0	850.0	850.0

TABLE 3.8 Estimation By Identifier **GD**

Parameters	Excitation Type I	Excitation Type II	Excitation Type III	Excitation Type IV
$\hat{\Theta}_1$	203.57	210.48	180.90	209.70
$\hat{\Theta}_2$	90.83	101.24	93.31	101.63
$\hat{\Theta}_3$	93.29	97.95	86.11	98.63
$\hat{\Theta}_4$	930.96	918.79	1387.8	922.47
$\hat{\Theta}_5$	884.03	892.37	1012.8	893.90

TABLE 3.9 Estimation By Identifier **VD**

Parameters	Excitation Type I	Excitation Type II	Excitation Type III	Excitation Type IV
$\hat{\Theta}_1$	203.19	208.30	182.61	207.47
$\hat{\Theta}_2$	91.83	101.04	97.32	101.66
$\hat{\Theta}_3$	93.60	96.52	86.36	97.22
$\hat{\Theta}_4$	932.60	919.67	1372.3	923.48
$\hat{\Theta}_5$	876.35	884.69	1022.9	886.03

3.5.3 Noise Influence

The simulation test in the previous sections are run on the assumption that all the information needed for the identification is available and perfectly free from noise. To reflect the realistic situation in estimation procedure, a substantial amount of noise should be added to the dynamic response in the form of random perturbations. To determine the reasonable perturbation ranges of noise, all the sensory devices for measuring position, velocity and acceleration are assumed to have 16-bit resolution and to be unreliable for the 4 least significant bits. The assumed sensible ranges of sensors and their perturbation bounds are given in TABLE 3.10.

TABLE 3.10 Perturbation Bound of Measurement Noise

Response	Sensible Range	Perturbation Bound
Position: [Deg]	±360	±0.088
Velocity: [Deg/Sec]	±570	±0.140
Acceleration:[Deg/Sec^2]	±9740	±2.378

Surprisingly, identifier model **MD** is highly sensitive to a slight noise in data and results in poor estimation when compared with the estimates by identifier **ED** and **SC** (TABLE 3.11). Particularly, when the dynamics of the system is excited by excitation type III and noise is added to the dynamic responses, it is clearly observed that the **MD** model leads to meaningless estimates (TABLE 3.12).

Since the positioning arm of IC DDR is mounted horizontally, the system energy is consists only of the kinetic energy. Therefore high energy level implicitly means a well developed velocity profiles of the system. Particularly, the evolution of total energy profile by excitation type I (Fig. 3.5) may offer comparatively better combination of high angular acceleration and well developed velocity than the other excitation inputs. In Fig.3.5, the total energy represents the accumulated sum of the incremental work done to the system, and the incremental work done to the arm over each sampling period is defined as belows.

TABLE 3.11 Estimates with Measurement Noise

(Excitation type I, level 30 %)

model	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	211.08	100.38	97.56	859.07	819.16
ED	213.09	102.01	98.93	876.83	849.51
SC	213.0	102.0	98.99	877.0	849.99
GD	203.57	90.83	93.30	930.97	884.03
VD	203.18	91.83	93.61	932.61	876.35

TABLE 3.12 Estimates with Measurement Noise

(Excitation type III, level 30 %)

model	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	-82.42	171.56	-133.46	294.62	-188.86
ED	212.42	100.13	96.54	839.10	858.87
SC	213.03	102.02	99.01	876.73	850.14
GD	177.96	91.01	84.54	1396.6	1007.5
VD	179.34	94.78	84.64	1382.0	1017.1

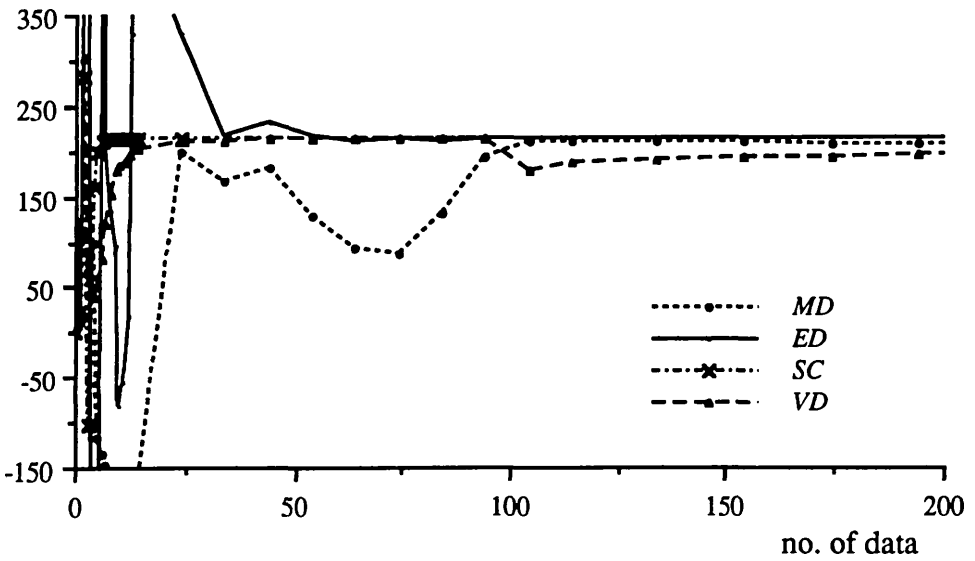

Figure 3.3a Stepwise Estimation of Parameter θ_1 With Measurement Noise (Excitation type I)

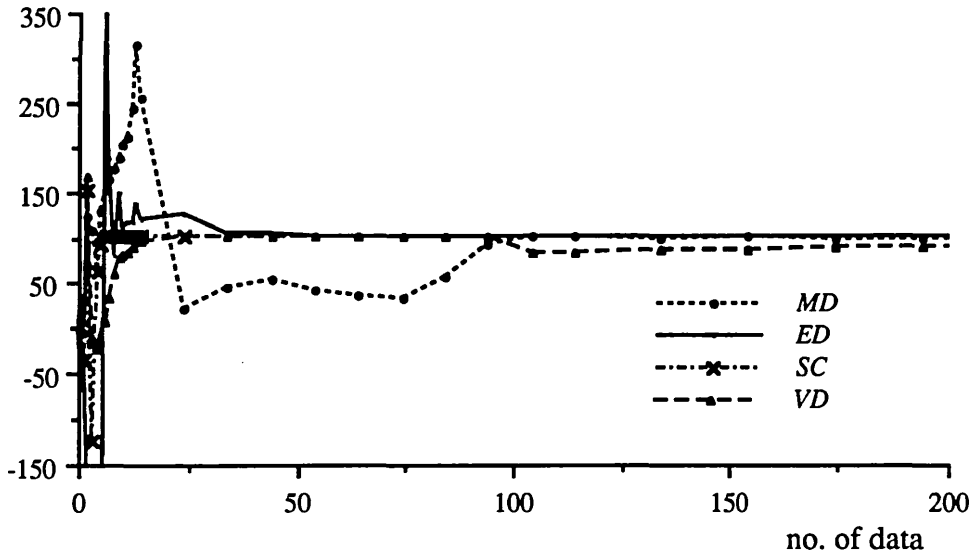

Figure 3.3b Stepwise Estimation of Parameter θ_2 With Measurement Noise (Excitation type I)

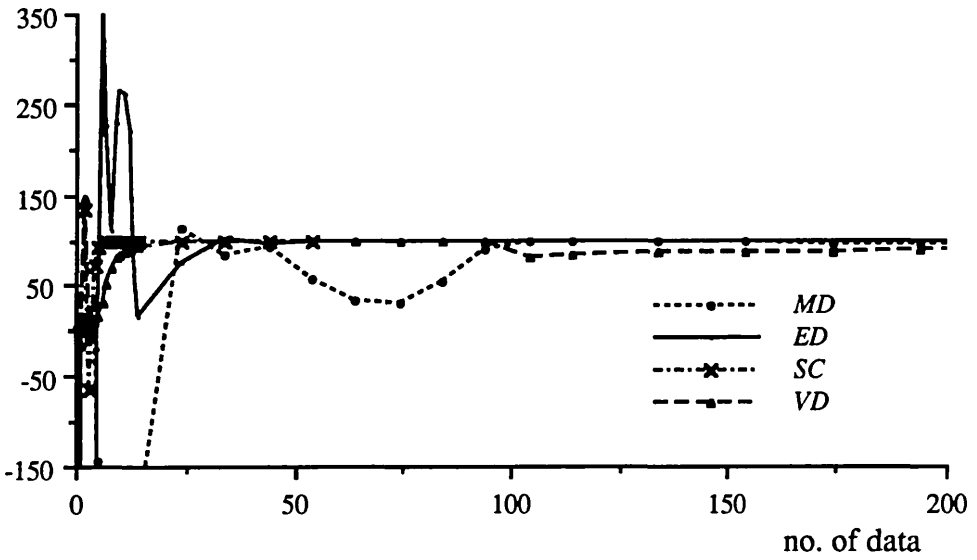

Figure 3.3c Stepwise Estimation of Parameter θ_3 With Measurement Noise (Excitation type I)

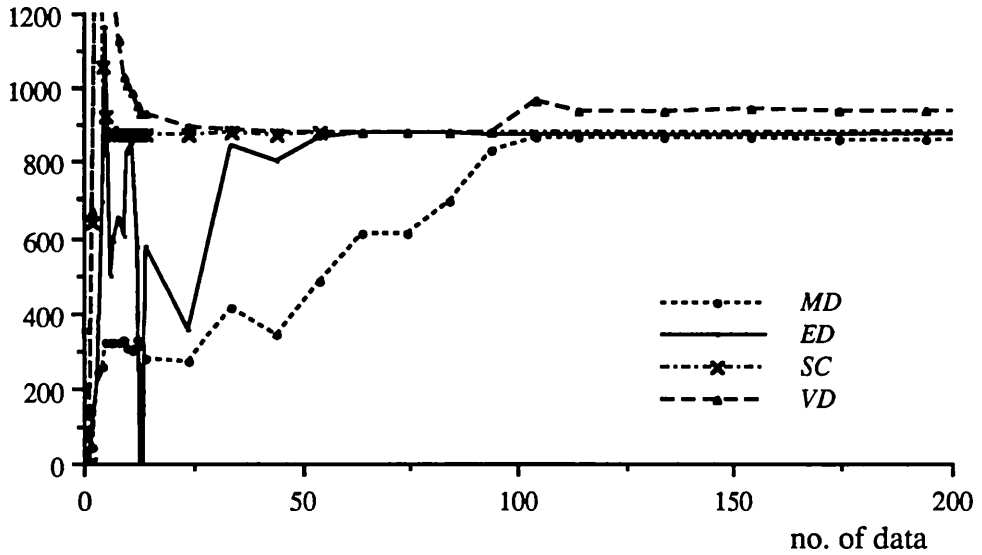

Figure 3.3d Stepwise Estimation of Parameter θ_4 With Measurement Noise (Excitation type I)

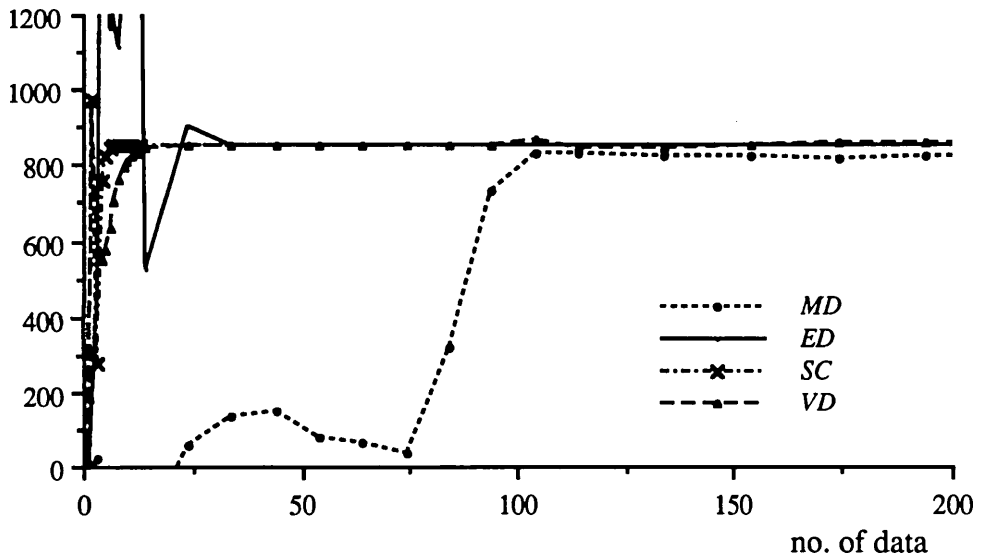

Figure 3.3e Stepwise Estimation of Parameter θ_5 With Measurement Noise (Excitation type I)

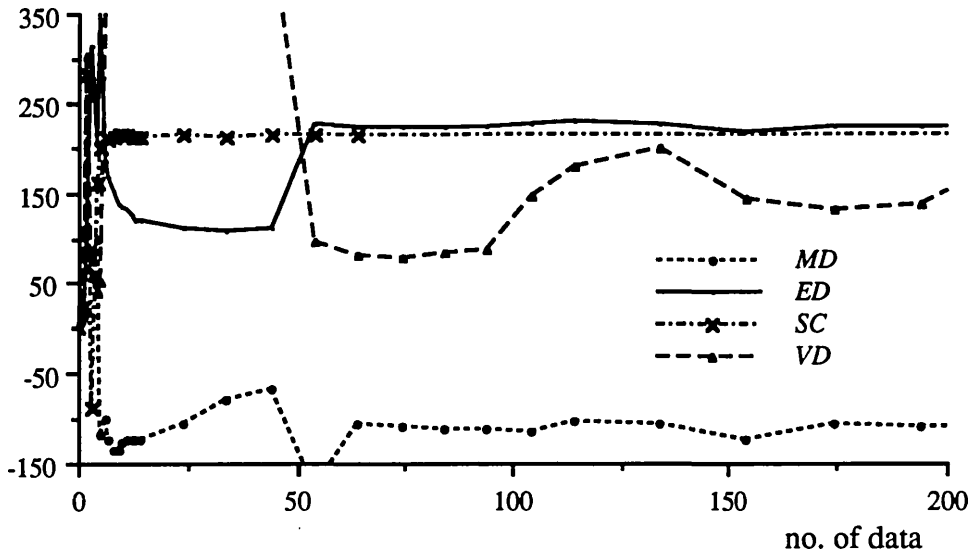

Figure 3.4a Stepwise Estimation of Parameter θ_1 With Measurement Noise (Excitation type III)

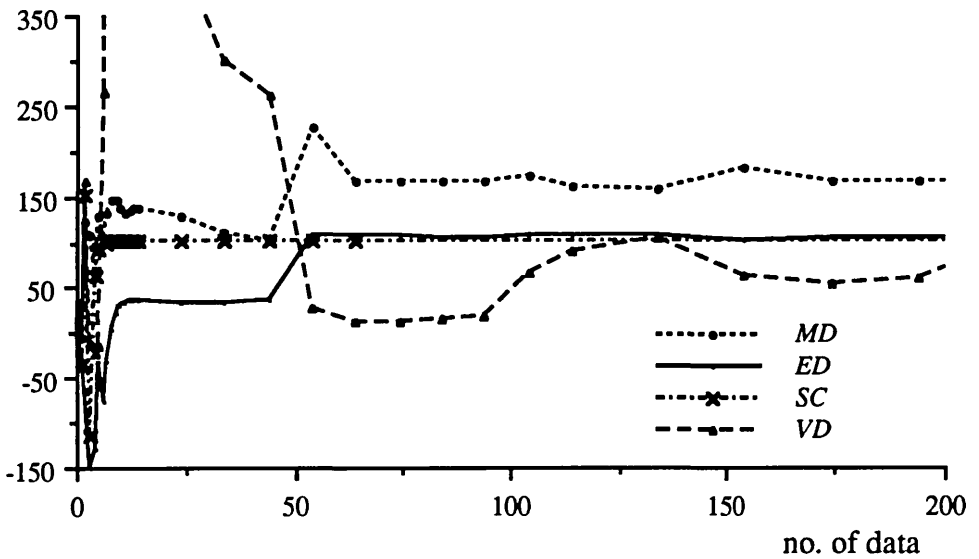

Figure 3.4b Stepwise Estimation of Parameter θ_2 With Measurement Noise (Excitation type III)

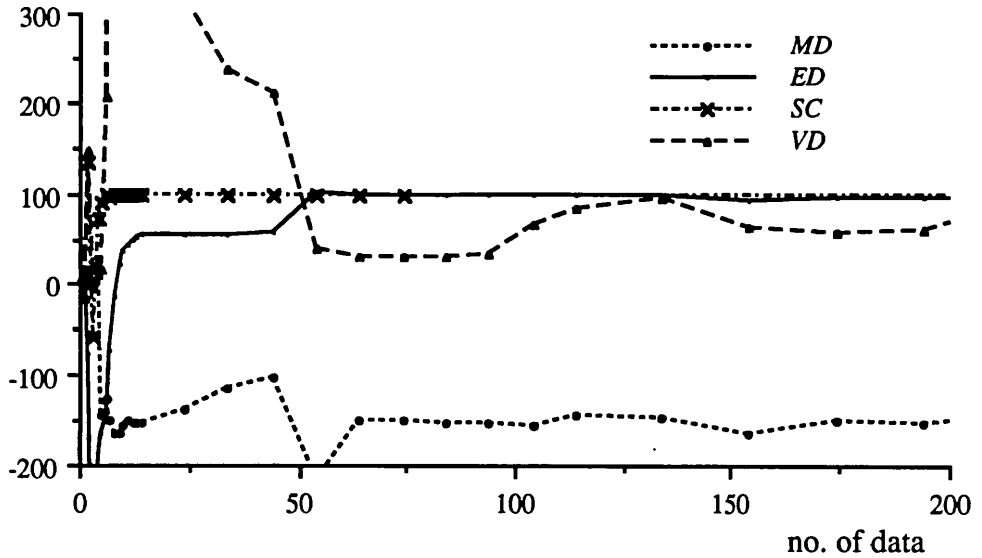

Figure 3.4c Stepwise Estimation of Parameter θ_3 With Measurement Noise (Excitation type III)

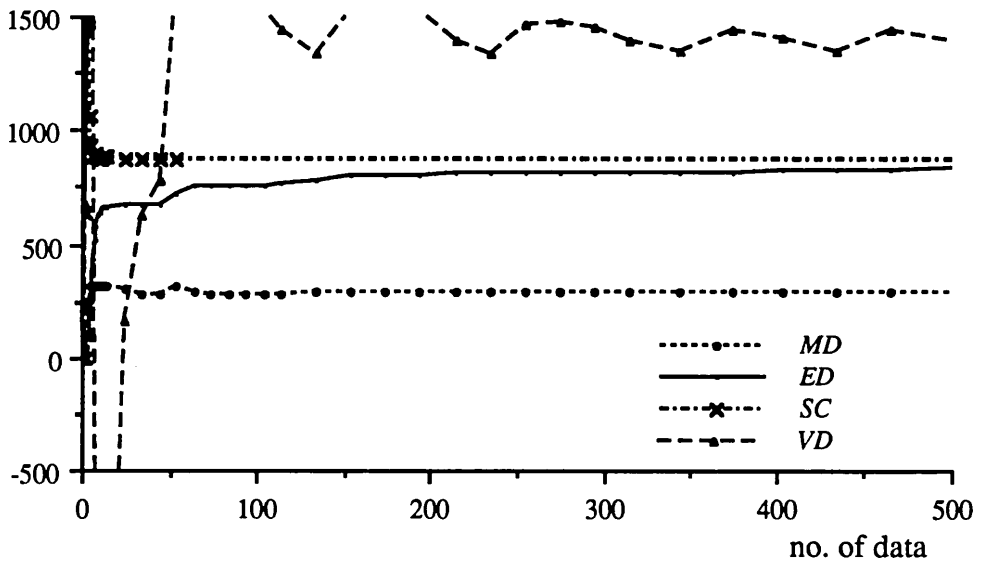

Figure 3.4d Stepwise Estimation of Parameter θ_4 With Measurement Noise (Excitation type III)

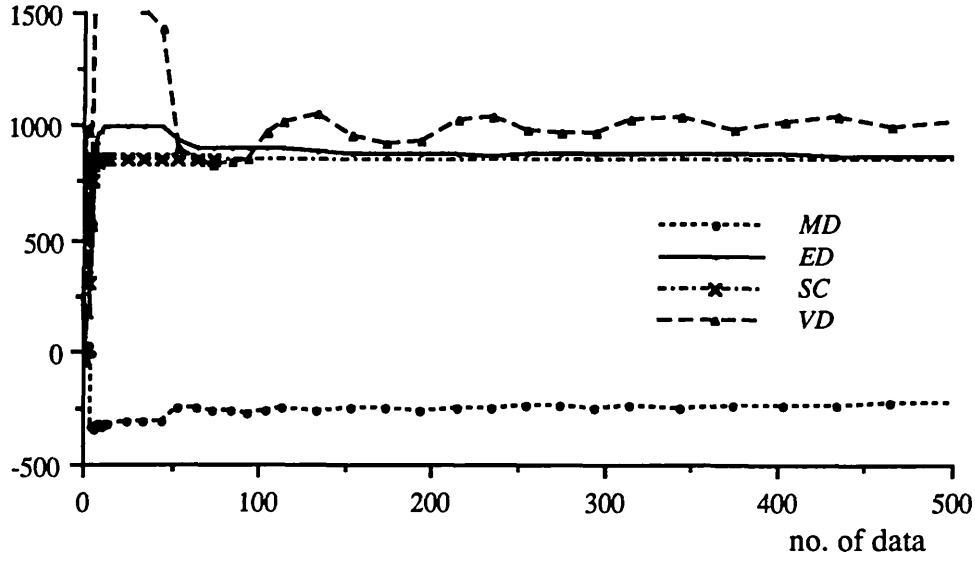

Figure 3.4e Stepwise Estimation of Parameter θ_5 With Measurement Noise (Excitation type III)

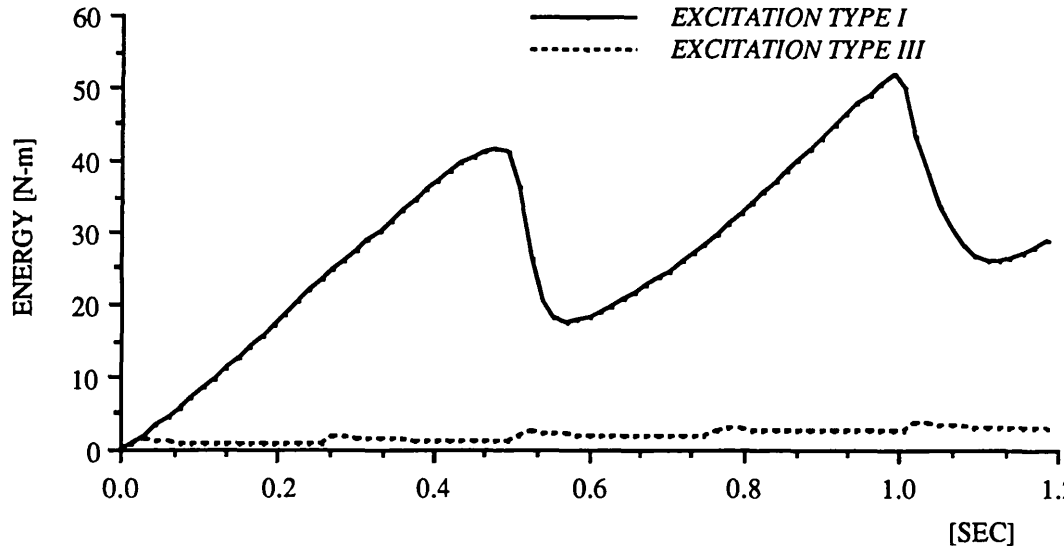

Figure 3.5 Total Energy Profile

$$\Delta w(k) = \sum_i T_i(k)[A_i(k+1) - A_i(k)] \tag{3-22}$$

Hence the excitation input which can produce the energy profile with high slope and large gap between top and bottom might result in more informative data for identification . From this argument, it can be said that the excitation type I can result in better estimation than the excitation type III (Fig. 3.6).

Without noise, identifier **MD** exhibits excellent estimation performance regardless of excitation type (refer to TABLE 3.5). But the results shown in TABLE 3.11 and 12(first row) reveal that if the excitation input is weak, then serious estimation errors are caused by a slight noise presence in the measurements. On the other hand, the identifier structure **ED** and **SC** sustain their good estimation performances though the estimation errors are slightly increased than in the noise-free case (refer to TABLE 3.6 and 3.7). The influence of noise on estimation error could be alleviated by applying a proper filter before calculating the estimates. The filtering problem will be discussed in the next section. The stepwise estimation profiles for parameters are displayed in Fig. 3.4 and 3.5. These graphs compare the influence of noise on the estimation performances of the suggested identifier structures under the different excitation inputs.

3.5.4 Influence of Initial Arm Posture

The test in this section is to examine whether any influence on parameter estimation can be made by a different choice of initial arm configuration. Three initial arm configurations which are tested in the simulation are listed in TABLE 3.13. The resulting estimates with the different initial postures are presented from TABLE 3.14 to TABLE 3.16 according to the identifier structures. As can be observed from these tables, the estimated parameter values are not affected by the changes in an initial arm posture. In other words, changing an initial arm configuration does not have an influence on the quality of informative data for identification. Fig. 3.7 illustrates the very similar energy profiles for the different initial arm postures. This graph can be again served as a backing material that the almost same quality of information is produced regardless of initial arm postures.

TABLE 3.13 Initial Arm Configuration

Configuration	Initial Angular Position of the Arm[Deg]	
	Joint 1	Joint 2
A	0	45
B	0	90
C	0	135

TABLE 3.14 Parameter Estimation For Different Initial Arm Postures(MD)

Parameter	Configuration		
	A	B	C
$\hat{\Theta}_1$	209.59	209.27	211.22
$\hat{\Theta}_2$	98.92	98.57	100.50
$\hat{\Theta}_3$	95.91	95.55	97.79
$\hat{\Theta}_4$	853.83	853.64	860.95
$\hat{\Theta}_5$	807.12	805.89	822.51

TABLE 3.15 Parameter Estimation For Different Initial Arm Postures(ED)

Parameter	Configuration		
	A	B	C
$\hat{\Theta}_1$	213.04	213.12	213.17
$\hat{\Theta}_2$	102.04	101.99	101.95
$\hat{\Theta}_3$	98.97	98.91	98.90
$\hat{\Theta}_4$	876.78	876.73	876.55
$\hat{\Theta}_5$	849.80	849.94	850.12

TABLE 3.16 Parameter Estimation For Different Initial Arm Postures(SC)

Parameter	Configuration		
	A	B	C
$\hat{\Theta}_1$	213.0	213.0	213.0
$\hat{\Theta}_2$	102.0	102.0	102.0
$\hat{\Theta}_3$	98.99	98.99	99.0
$\hat{\Theta}_4$	877.0	877.0	876.99
$\hat{\Theta}_5$	849.99	849.99	850.0

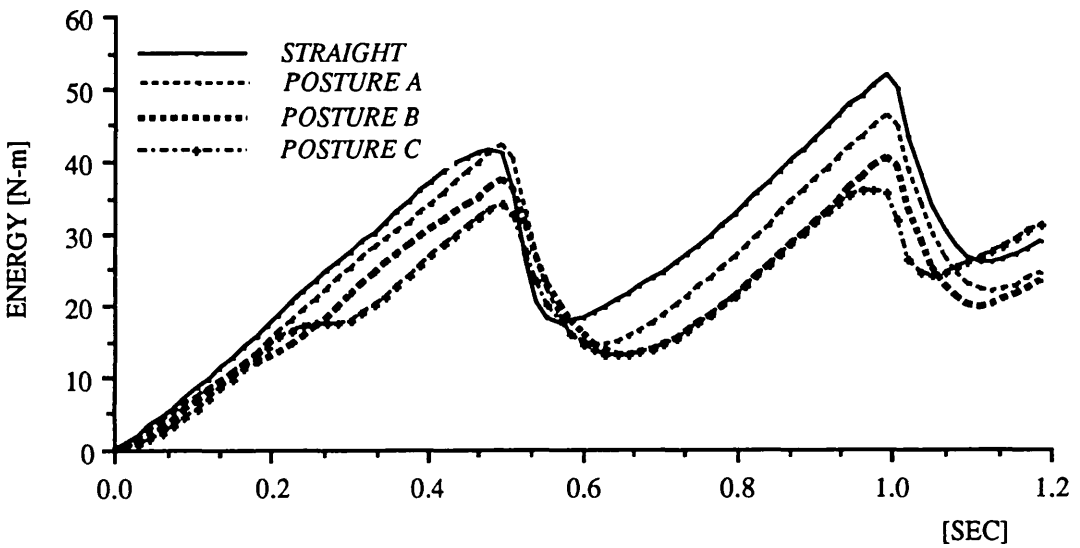

Figure 3.6 Total Energy Profile For Different Initial Arm Postures
(Excitation Type I)

3.5.5 Estimation Through Filtering Process

The parameter identification procedure requires states such as position, velocity, acceleration and force/torque according to its identifier structure. The quantitative measurement of these states is done using installed transducers.

In practical situations, most of industrial robots are equipped with only position sensing devices such as encoders or resolvers since they are primarily used as positioning devices. For instances, PUMA robot has encoders at each joint to measure joint angles, but no tachometers. In the CMU DD Arm, high resolution encoders are directly mounted on the joints[Rangan 82]. AdeptOne direct drive robot measures the joint position through the anti-backlash gear coupling using low resolution shaft encoders.

The IC DDR has been designed to have 16-bit resolution resolver at each joint without involving any transmission reduction to the motor shaft. However, the identifier structures introduced in this chapter need velocity or acceleration information in addition to the positional information. Thus the measured positional data should be processed to obtain the higher order derivatives. In the following section, a differentiating filter is designed to obtain the derivatives from positional data, and the estimation performance of each identifier structure using filtered data will be discussed.

3.5.5.1 Differentiating Filter

In principle, two types of digital filter can be defined according to their linear formulae, nonrecursive and recursive filters. In the recursive filter, the poles of transfer function can be placed anywhere inside the unit disc in z-plane. The consequence of this freedom is that selectivity can be easily be achieved with low order transfer functions. On the other hand, in nonrecursive filter with the pole fixed at the origin, selectivity can be achieved only by using a relatively higher order for transfer function. For the same filter specification, the required order in nonrecursive design can be 5 to 10 times higher than in a recursive design[Antoniou 79].

A very important advantage of the nonrecursive filter is that it can be implemented by using the Fourier series expansion. i.e., the differentiating filter can be designed by choosing a finite set of coefficients c_k as follows.

$$y_n = \sum_{k=-N}^{k=N} c_k x_{n-k} \tag{3-23}$$

Since any function can be written as the sum of even and odd function, the following identity can be given.

$$c_k = \frac{c_k + c_{-k}}{2} + \frac{c_k - c_{-k}}{2} \tag{3-24}$$

This expression shows that any nonrecursive digital filter can be written as the sum of a smoothing(even) filter portion and a differentiating(odd) filter portion[Hamming 89]. A smoothing filter portion can be viewed as a linear combination of the sum of symmetrically placed data, while a differentiating filter portion uses its differences. Substituting $x_n = e^{i\omega n} = z^n$ into the equation represents that they are simply the terms of the general Fourier series expansion.

The expression for derivative can be easily obtained from

$$\frac{d}{dt}\left[e^{j\omega t}\right] = j\omega e^{j\omega t} \tag{3-25}$$

Hence, designing a filter to estimate the derivative of some data is to approximate the transfer function of differentiator(here $j\omega$). As can be seen in Eq.(3-25), the process of differentiation amplifies high frequencies much more than low frequencies. That is why it is numerically difficult to differentiate data. Since high frequencies are often the noise, this means that the filter should properly cut off the frequencies after some value ω_c . This implies that a filter is asked to both differentiate and reject a significant amount of noise. Thus, the response of the differentiating filter can be defined as

$$H(e^{j\omega T}) \cong \begin{cases} j\omega & \text{for } |\omega| < \omega_c \\ 0 & \omega_c < |\omega| < \omega_s/2 \end{cases} \tag{3-26}$$

where ω_c is cut-off angular frequency and ω_s is sampling angular frequency. For a finite -order transfer function, $H(e^{i\omega T})$ can be written in terms of Fourier series expansion as was given in Eq.(3-23).

$$H(e^{j\omega T}) = \sum_{k=-N}^{k=N} h(kT)e^{-j\omega kT} \tag{3-27}$$

here,

$$h(kT) = \frac{1}{\omega_s}\int_{-\omega_s/2}^{\omega_s/2} H(e^{j\omega T})e^{j\omega kT}d\omega \tag{3-28}$$

substituting $e^{i\omega T} = z$ into Eq.(3-27) gives

$$H(z) = h(0) + \sum_{k=1}^{(N-1)/2} [h(-kT)z^k + h(kT)z^{-k}] \tag{3-29}$$

Eq.(3-29) is the form of a noncausal filter. Causality can be brought about by multiplying H(z) by $z^{-(N-1)/2}$. This modification is permissible since the amplitude response will remain unchanged. However the amplitude response of the filter shows the pass-band and stop band oscillation, which is known as Gibbs' oscillation, due to slow convergence of Fourier series[Antoniou 79]. This oscillation problem can be reduced by preconditioning h(kT) using a proper window function. Using Eq.(3-26) and (3-28), the coefficients of differentiating filter are expressed in terms of rotational frequencies.

$$h(0) = 0$$

$$h(kT) = \frac{1}{\omega_s}\int_{-\omega_c}^{\omega_c} j\omega e^{j\omega kT}d\omega = -\frac{2}{\omega_s}\int_0^{\omega_c} \omega \sin\omega kT\, d\omega$$

$$= \frac{2f_c}{kf_s^2}\cos(2\pi k\frac{f_c}{f_k}) - \frac{f_s}{k^2\pi}\sin(2\pi k\frac{f_c}{f_s}), \quad k \neq 0 \tag{3-30}$$

Where f_c, f_s are the cut-off frequency and the sampling frequency respectively.

Fig.3.7 shows the frequency responses of the several digital differentiating filters. The recursive filter is designed through PC-MATLAB[MATLAB 87] by using the Yule-Walker method[Friedlander 84] which performs a least square fit of the specified frequency response. The designed nonrecursive filter has the length 61 and Kaiser window function with factor 4.0 is applied to reduce the band oscillation problem. In an on-line application, the choice of filter depends on the hardware used for the implementation of the filter. The recursive filter, when implemented in fixed-point arithmetic, may have instabilities and could have large quantization noise, depending on the number of bits allocated to the coefficients[Parks 87]. On the other hand, the nonrecursive filter is naturally implemented in nonrecursive way and can guarantee a stable calculation. TMS320 family of digital signal processors have special instructions to facilitate the implementation of a nonrecursive filter. A length-N nonrecursive filter can be computed in about the same time as an recursive filter of order N/5 for the TMS320 family signal processor.

3.5.5.2 Estimation Using Filtered Dynamic Responses

The simulation tests in this section assume that only positional responses are available and contaminated by noise. In reality, this situation is frequently encountered when only positional sensing devices are available. Joint angles are differentiated and double- differentiated respectively using the digital differentiating filter designed in the previous section(nonrecursive, noncausal filter of length 61).

As shown in TABLE 3.17, identifier structures, **ED** and **SC** allow the good estimates of parameters. The estimated parameter values are fairly close to the true values though they contain some errors. Even if the system is weakly excited(using excitation type III), **ED** structure still gives a good estimation but the errors become bigger(TABLE 3.18). On the other hand, **SC** structure shows relatively bad estimation. In this weakly excited case, it is clearly observed that the identifier structure **ED** outperforms any other identifier structures. The relatively bad performance of **SC** structure is mainly ascribed to poor acceleration data because the acceleration information is seriously deteriorated over the double-differentiating procedure.

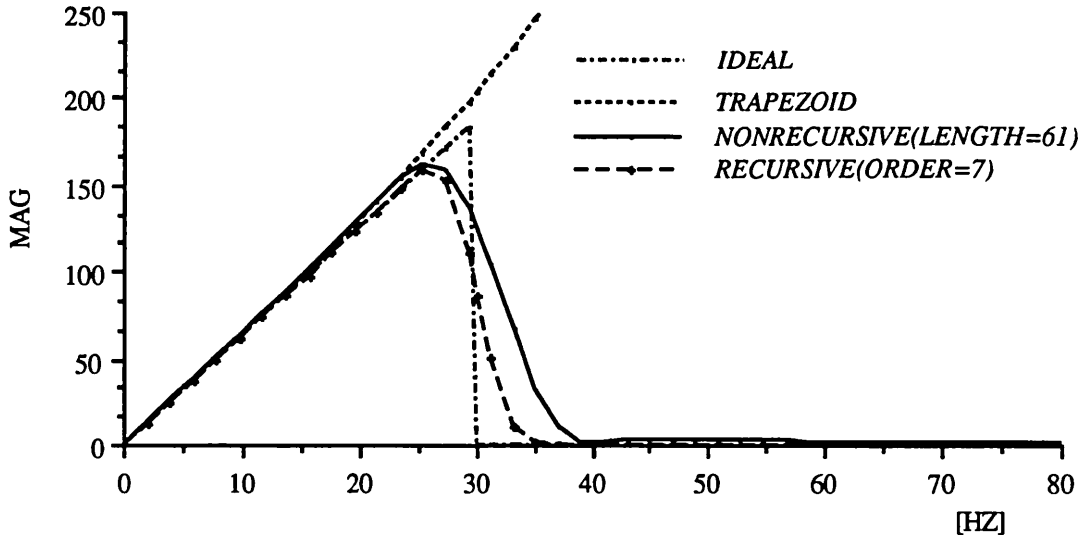

Figure 3.7 Frequency Response of Differentiating Filter
(Sampling Frequency:200 Hz,Cutoff Frequency :30 Hz)

TABLE 3.17 Estimated Parameter Values Using Filtered Data
(Excitation Type I, Level 30 %)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	180.12	78.74	72.67	800.62	679.44
ED	204.82	102.61	108.14	855.23	861.63
SC	206.80	95.63	96.48	899.07	858.41
GD	205.20	91.44	94.58	923.45	881.25
VD	205.04	92.70	94.88	924.40	872.71

TABLE 3.18 Estimated Parameter Values Using Filtered Data
(Excitation Type III, Level 30 %)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	-114.06	144.21	-150.37	325.89	-308.86
ED	185.11	90.72	102.78	1487.1	508.75
SC	61.73	-13.64	17.85	1573.5	662.86
GD	59.98	-11.98	17.72	1779.0	749.02
VD	54.91	-12.48	15.51	1780.9	747.76

TABLE 3.19 Estimated Parameter Values Using Filtered Data

(Excitation Type I, Level 50 %)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	202.03	95.29	93.01	846.00	789.20
ED	209.61	100.85	99.91	862.51	850.42
SC	205.74	97.89	95.83	897.38	851.98
GD	203.13	96.13	94.31	899.73	866.25
VD	203.60	97.44	94.30	919.27	859.76

TABLE 3.20 Estimated Parameter Values Using Filtered Data

(Excitation Type I, Level 5 %)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	22.66	-2.45	-20.46	83.71	-551.84
ED	162.91	81.22	133.44	911.59	806.96
SC	121.14	-5.76	32.30	1019.8	703.09
GD	118.29	-8.15	30.89	1043.2	707.68
VD	115.77	-7.94	29.19	1042.9	707.97

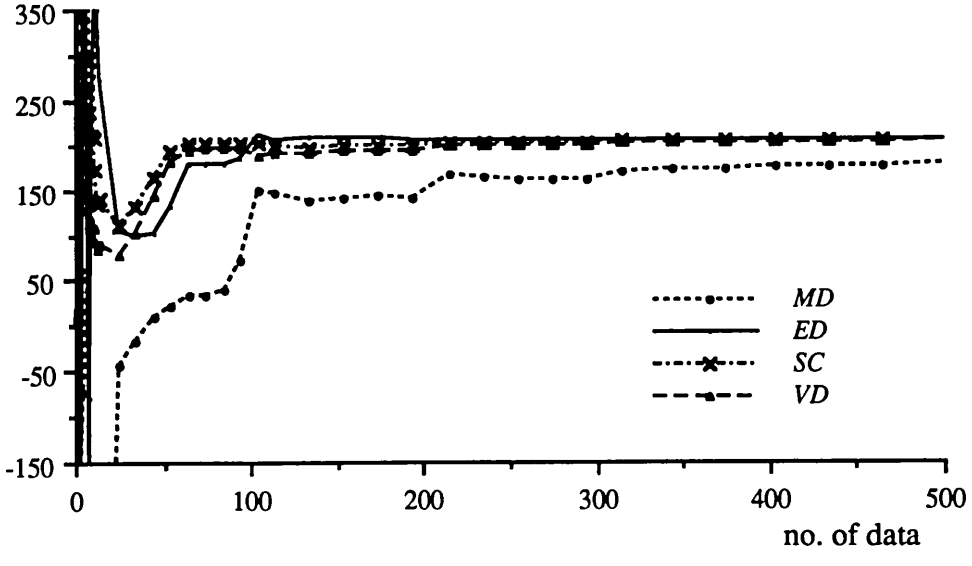

Figure 3.8a Stepwise Estimation of Parameter θ_1 Using Filtered Dynamic Responses (Excitation Type I)

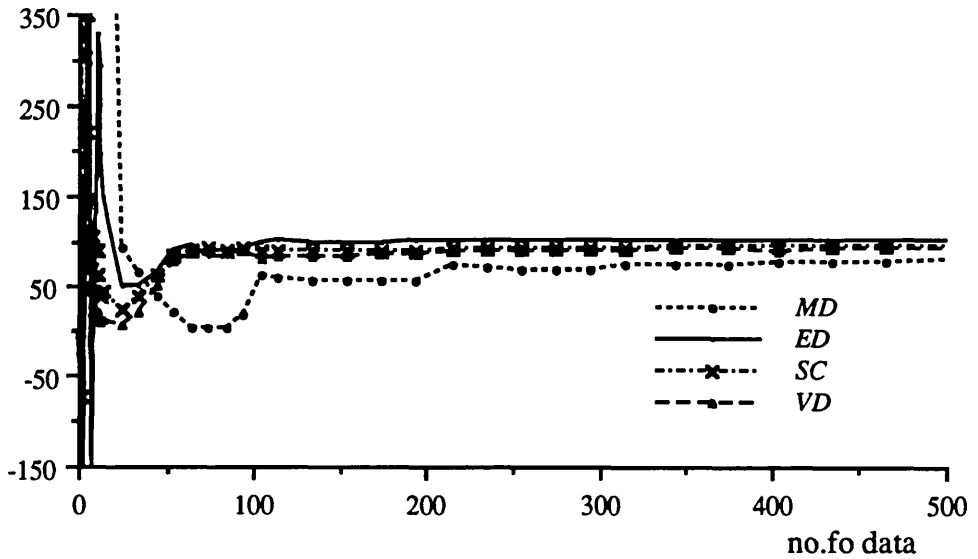

Figure 3.8b Stepwise Estimation of Parameter θ_2 Using Filtered Dynamic Responses (Excitation Type I)

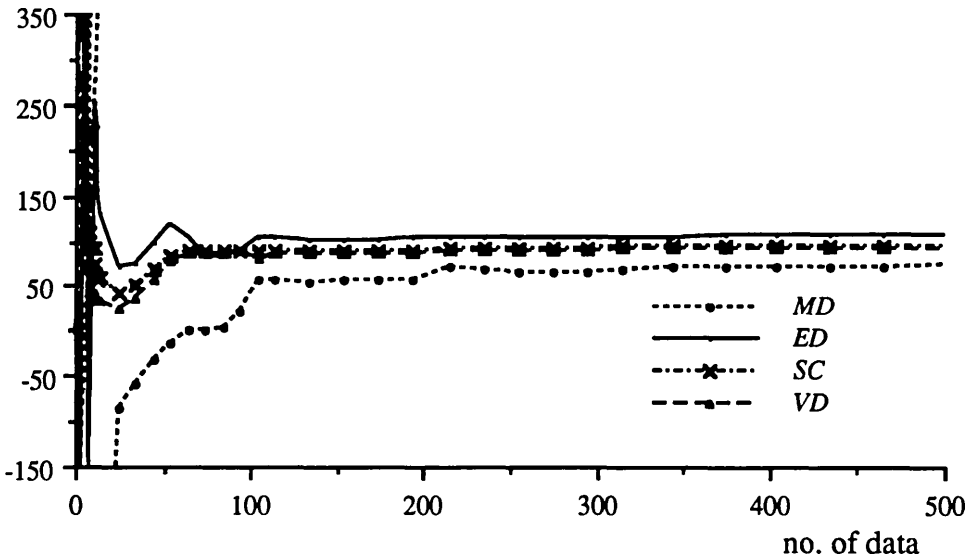

Figure 3.8c Stepwise Estimation of Parameter θ_3 Using Filtered Dynamic Responses (Excitation Type I)

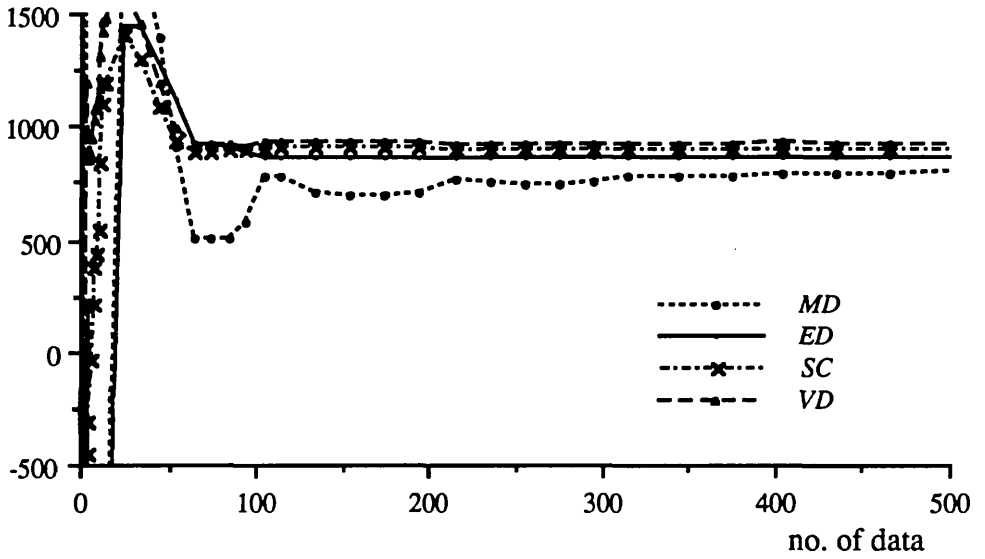

Figure 3.8d Stepwise Estimation of Parameter θ_4 Using Filtered Dynamic Responses (Excitation Type I)

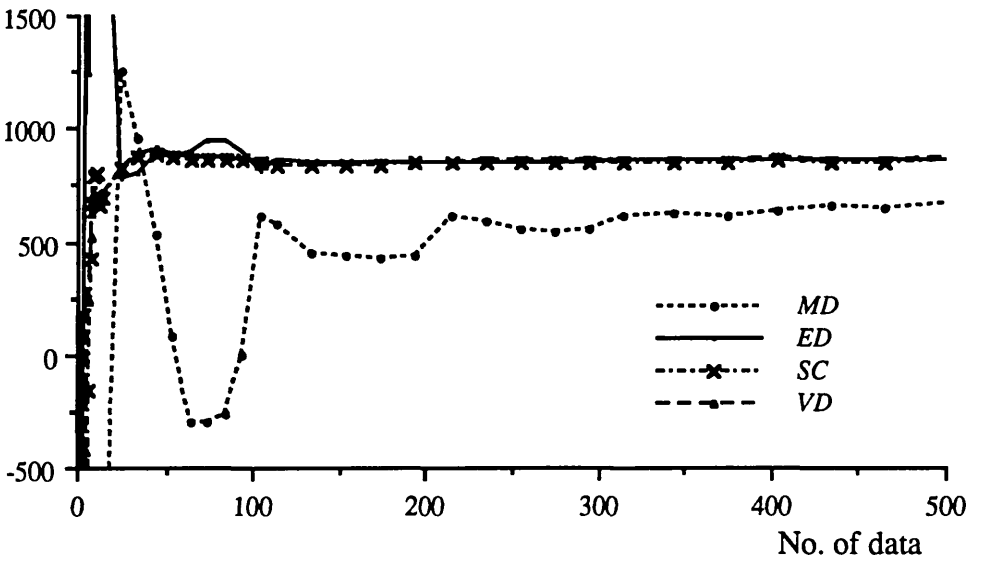

Figure 3.8e Stepwise Estimation of Parameter θ_5 Using Filtered Dynamic Responses (Excitation Type I)

The similar results are observed in TABLE 3.20 when the excitation type I (square wave) is applied instead of excitation type III (pulse train) but with a decreased excitation level (5 % of maximum permissible input limit). Surprisingly, even such a poor excitation condition, **ED** structure still represents its outstanding estimation performances compared to other identifier structures(second row in TABLE 3.20). In contrast, TABLE 3.19 shows the estimation results when the excitation input level is brought up to 50 % of maximum input using excitation type I. All the estimated parameter values are much improved. This clearly suggests that the increased excitation level results in more informative dynamic responses for parameter estimation purpose. Among those improved estimates, **ED** structure achieves the closest estimation to the true values. From this outstanding performance in **ED** structure, it could be inferred that the approximation error, which is involved in the derivation procedure of identifier structure **ED**, is negligible, and probably the deteriorated acceleration data due to the double-differentiation of positional data contaminated by noise has worse influence than the small approximation error of identifier **ED**.

Graphs from Fig.3.8a to Fig.3.8e indicates the stepwise estimation performances of identifier structures under the excitation input type I (level 30 %). These figures show the well stabilized estimated values after the sufficient data (approximately 400 data) are acquired.

3.5.6 Effect of Excitation Frequency

The final simulation experiment is concerned about the influence of excitation frequency on estimation. TABLE 3.21 shows the resulting estimates under the excitation type I (level 30 %) with frequency of 5 Hz. The estimated values are worse than the 1 Hz case with the same excitation input type (refer to TABLE 3.17). The differences in total energy profile of the system during the excitation are illustrated in Fig.3.9. Referring to the discussions on the meaning of energy level given in section 3.5, this can be interpreted such that 5 Hz excitation input is delivering the less informative data for estimation. Further, in practical situations, excitation input of high frequency is more likely to develop an unmodelled structural dynamics (vibrations). This unmodelled dynamic responses may cause an undesirable effect on estimation such as a biased offset of estimated values. Hence, the low frequency excitation could offer more desirable information for estimation.

TABLE 3.21 Estimated Parameter Values With 5 Hz Input Frequency

(Excitation Type I, level 30 %)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	-39.80	150.67	-105.92	853.26	182.77
ED	214.50	106.70	111.60	1203.1	669.62
SC	186.67	70.69	82.73	1329.5	936.34

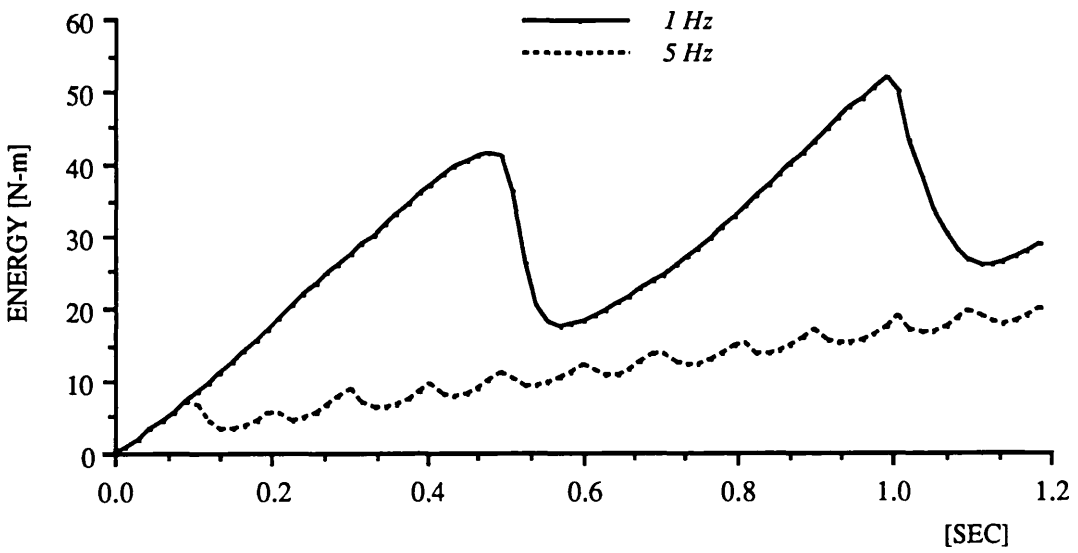

Figure 3.9 Total Energy Profile With Different Input Frequencies

(Excitation Type I)

4. CONTROL STRATEGY

4.1 Introduction

The complexity of robot dynamics casts challenging problems for controller design and creates the need for the development of sophisticated control algorithms to achieve high speed requirement and positional accuracy at the same time.

The basic principle of control system design for a complex mechanical system is to construct a hierarchical structure. Hierarchical organization of control system is vertical so that higher control level deals with wider aspects of overall system behaviour than a lower level. A higher control level communicates with a lower level giving it instructions and receiving relevant information required for decision making. After obtaining information from a lower level, higher level makes decisions by taking account of general decision made by upper higher level, and forward necessary information to a lower level for execution.

The number of levels in a hierarchical control system depends on the complexity of the tasks for which the robot is intended. Three control levels are most often encountered: the top level, the middle level, and the low level. For instance, the top level can recognize the obstacles in the operating space and the conditions under which a task is being performed, and makes decision on how the task is to be accomplished. The middle level divides the imposed operation into elementary movements and perform the distribution of the elementary movements to each joint motion. The lowest level executes the imposed motion by means of appropriate actuators. The low control levels can be realized in various modes, and their capability for realizing the final accurate motion will determine the organization and complexity of the higher control levels.

The control system mentioned above represents the complete control system in a broad sense. The research in this chapter is concerned with the problem of synthesizing the two lower control levels of a robot system. In other words, the problem for accurate realization of the functional movement prescribed by the top level will be discussed.

The concept of a controller design can be broadly classified into two categories: non-adaptive control scheme and adaptive control scheme. In either case, the dynamic model of object system can be used in the controller design. If the dynamic model of a robot system is positively reflected in the design of the controller, a simplified control law can be realized because the nonlinearities and dynamic interactions can be accounted by the dynamic model. On the other hand, if a simplified dynamic model is involved in the control scheme, the burden of compensating for the nonlinearities and interactions is shifted on the feedback control law. Thus the control law often becomes more complex form than in the model based case.

The quality of control performance based on the dynamic model depends on the extent to which the real robot dynamics are taken into account. Generally, the top level does not consider the actual system dynamics but only prescribes actions to be implemented by the lower level. But the lower level takes account of the dynamic behaviour of the robot to achieve the high speed motion requirement and the accurate tracking performance.

The theoretical basis for control scheme that utilizes nonlinear state feedback was developed by Freund. In his method, decoupling and linearization of the entire system was achieved by the cancellation of the unwanted system dynamics[Freund 82]. Fournier investigated the implementational difficulties associated with applying a nonlinear state feedback control law[Fournier 84]. The computed torque technique is a similar scheme to decouple the dynamic behaviour of a robot system[An 87, Khosla 86b]. This scheme is extended to operate in the cartesian space and is called the resolved acceleration controller[Luh 80b, Wampler 88].

One important barrier in the implementation of the control scheme based on the dynamic model was the intensive real-time computational requirement of the inverse dynamics. This led researchers to develop different forms of computer implementations of dynamic equations in order to achieve real-time computation[Luh 80a, Horak 84].

Vukobratovic linearized dynamic model around a specific trajectory to circumvent the complexity of robot dynamic model[Vukobratovic 83]. This approach steps from the global nonlinear control problem to a local linear time-varying control problem.

However the application of linearization method is limited by the fact that it is based upon an approximated model.

To overcome the shortcoming of modelling errors, attempts to synthesize adaptive control algorithms have been made for robot systems. The basic idea in model-reference adaptive control is to design a control signal which will force the controlled system to respond in a desirable fashion as specified by the choice of a stable and linear time-invariant reference model. For example, a model-reference adaptive control scheme was proposed in [Dubowsky 79]. This scheme utilized a second-order linear system as the reference model for each joint. The gains in the position and velocity feedback loops are adjusted by a gradient algorithm[Astrom 89] to offset model following errors. But this approach is impractical because it assumed the existence of counterbalance to cancel all of the gravitational force/torque and it neglected all the inertial, centrifugal and Coriolis coupling forces/torque. An improved scheme was recently proposed in [Craig 88]. This uses adaptive algorithm to update the parameters of the closed form dynamic model of robot. In his experiments, a well preplanned trajectory was used for estimation of parameters inside the adaptive algorithm. However, in practical circumstances, it is difficult to state that the sufficient excitation condition for successful on-line estimation of parameter could always be met in the actual trajectory carried out by the robot. Adaptive algorithm of [Koivo 83] used a linear auto-regressive model for each joint and computes the parameters of the model by on-line recursive algorithms. This method is also limited to the case that robot has almost constant joint inertias. Other model-reference adaptive control scheme utilizing different design procedures have been proposed in [Lackey 86, Middleton 88, Goldenberg 89].

Self-tuning control algorithms and model-reference adaptive control scheme share similar structure. These two classes were originally derived from very different points of view, but have recently come to be recognized as closely related[Ljung 78]. Self-tuning control can be understood as, in a sense, the simplest possible adaptive control algorithm. The design of self-tuning controller for robot system starts from assuming that a linear model of the robot is known. The unknown parameters of the model are estimated on-line and substituted in the controller design to compute the required control signals[Farsi 86, Wahab 85, Walters 82, Tzafestas 86]. The disadvantage of this method is that the robot must be modelled as a linear time-varying process.

In spite of the growing list of adaptive control schemes for a robot system, most of these schemes have been based on the approximated linear model and the assumption of slow parameter variation. But these factors may lead the controlled system not to satisfy the convergence criteria in a global sense[Ortega 88]. Even those adaptive control schemes for robots perform fairly well in simulation stage, very little amount of experimental evaluation through actual implementation has been reported. Moreover, the question about the usefulness of adaptive control schemes remains very much alive[Astrom 80]. The principal reason is that all controlled processes for which adaptive control might be suitable are essentially nonlinear, stochastic and slowly varying parameter systems, and therefore difficult to control and to analyse by conventional control method[Jacobs 81]. Further, Vukobratovic states *"Few efforts have been made to analyse the necessity of adaptive control for robot system"* . *It seems that most of the parameter variations in practical situation could be compensated by sufficiently robust control"* [Vukobratovic 82].

4.2 Joint Motion Planning

The curve along which the robot moves from the initial location to the final location is called trajectory. A trajectory planning scheme generally interpolates or approximates a desired path by a class of polynomial functions and generates a sequence of time based points. Trajectory can be specified either in joint space coordinates or in cartesian space coordinates. Cartesian trajectory planning is more straightforward because it is easier to visualize the hand position in cartesian coordinates than in joint coordinates. Trajectory planning in cartesian space can be done in two steps: by selecting a set of interpolation points in cartesian coordinates along the user-specified cartesian path, and by specifying a class of functions to approximate these path segments. For trajectory planning in joint space, time history of all joint angles and their first two time derivatives are planned to describe the desired motion of robot. Planning in joint space has the advantage that the trajectory is planned directly in terms of the controlled variable during the motion.

Trajectory planning can be interpreted as a part of the control scheme in a broad sense. The trajectory planning in this section is to supply test motion profile for evaluating the performance of a control scheme. Since the IC DDR is a highly nonlinear and coupled system, it is not possible to characterize its behaviour from one particular test motion.

Two types of trajectory are designed based on smooth polynomial function. The basic difference between the two types is the acceleration profile. These two trajectories will be used to generate the intermediate points between the initial and final points in joint space at a chosen sampling rate.

There are many smooth functions which can be used to interpolate between the initial and goal joint coordinate values. For generating a smooth motion, at least four constraints on smooth function are evident. Two constraints on the function values come from the selection of initial and final values. Additional two constraints are that the function is continuous in velocity. These four constraints can be satisfied by a polynomial of third order. Then the cubic polynomial trajectory has the form as below in terms of normalized time variable τ

$$S(\tau)=J_s + v_s t_r + (3\delta - v_f t_r - 2v_s t_r)\tau^2 + (v_f t_r + v_s t_r - 2\delta)\tau^3 \qquad (4\text{-}1)$$

Where $t_r = t_f - t_s$: the real time required for the desired motion segment.
$\tau = (t - t_s) / (t_f - t_s)$: normalized time variable $\tau \in [0, 1]$
t : real time in second
t_s, t_f : initial and final time in seconds
J_s : initial value of joint angle
v_s, v_f : initial and final velocities of joint motion
δ : travelled angle during t_r

The position, velocity and acceleration profiles of cubic polynomial trajectory are depicted in Fig.4.1. Position and velocity change smoothly in this trajectory profile, but the discontinuity of acceleration occurs at the start and end of the motion. For practical reasons, it is desirable to keep the continuity of acceleration profile because sudden changes of acceleration could induce the undesirable structural vibrations.

The acceleration can not be specified independently at the both ends of the motion in the cubic polynomial trajectory. For the smooth evolution of acceleration profile near the beginning and the end of each trajectory segment, a quartic polynomial motion is splined together with the cubic polynomial motion. This splined cubic polynomial trajectory has the following three submotion segments: the first segment is fourth order polynomial trajectory satisfying the motion from initial position to the intermediate lift-

up position, the midmotion segment is a cubic polynomial achieving the motion from the lift-up position to the set-down position, and the last motion segment is also a fourth order polynomial from the set-down position to the final position. The total travelling time is equally divided into three time periods. This trajectory profile is shown in Fig.4.2. The polynomial equations in each submotion segment are expressed in normalized time as follows:

Lift-up motion segment (S_I):

$$S_I = \left\{(\delta_1 - \sigma) - \frac{1}{3}v_s t_r - \frac{1}{18}a_s t_r^2\right\}\tau^4 + \sigma\tau^3 + (\frac{1}{18}a_s t_r^2)\tau^2 + (\frac{1}{3}v_s t_r)\tau + J_s \tag{4-2a}$$

$$v_1 = \frac{3}{t_r}(4\delta_1 - \sigma) - 3v_s - \frac{1}{3}a_s t_r \tag{4-2b}$$

$$a_1 = \frac{54}{t_r^2}(2\delta_1 - \sigma) - 36\frac{v_s}{t_r} - 5a_s \tag{4-2c}$$

$$\sigma = \frac{1}{4}(11\delta_1 - 2\delta_2 + \delta_3) - \frac{t_r}{24}(19v_s + v_f) - \frac{t_r^2}{216}(29a_s - 3a_f) \tag{4-2d}$$

Intermediate motion segment (S_{II}):

$$S_{II} = (\delta_2 - \frac{1}{3}v_1 t_r - \frac{1}{18}a_1 t_r^2)\tau^3 + (\frac{1}{18}a_1 t_r^2)\tau^2 + (\frac{1}{3}v_1 t_r)\tau + J_1 \tag{4-3a}$$

$$v_2 = 9\frac{\delta_2}{t_r} - 2v_1 - \frac{1}{6}a_1 t_r \tag{4-3b}$$

$$a_2 = \frac{54\delta_2}{t_r^2} - 18\frac{v_1}{t_r} - 2a_1 \tag{4-3c}$$

Set-down motion segment (S_{III}):

$$S_{III} = \left\{9\delta_3 - \frac{t_r}{3}(4v_2 + 5v_f) - \frac{t_r^2}{18}(a_2 - a_f)\right\}\tau^4 + \left\{-8\sigma_3 + \frac{t_r}{3}(3v_2 + 5v_f) - \frac{t_r^2}{18}a_f\right\}\tau^3 + (\frac{1}{18}a_2 t_r^2)\tau^2 + (\frac{1}{3}v_2 t_r)\tau + J_2 \tag{4-4}$$

Where δ_1, δ_2 and δ_3 are the differences of joint angles between successive submotion segments and defined as

$$\delta_1 = J_1 - J_s \tag{4-5a}$$
$$\delta_2 = J_2 - J_1 \tag{4-5b}$$
$$\delta_3 = J_f - J_2 \tag{4-5c}$$

The maximum acceleration and deceleration occur at the start and end of motion in the cubic polynomial trajectory, while those maximums in the splined cubic polynomial trajectory are obtained during the lift-up and set-down motion period. If the maximum attainable acceleration is set to be equal to that of the cubic polynomial motion, then the travelled angles of each submotion segment become

$$\delta_1 = \delta_3 = \frac{5\delta}{24} \tag{4-6a}$$
$$\delta_2 = \frac{7\delta}{12} \tag{4-6b}$$

Two illustrative test motion are chosen to provide insight into the effect of dynamic compensation control scheme. In the first test motion, joint 1 and 2 are commanded to move in reverse direction, and the travelling time for each joint motion is equal so that the two joints can reach the destination angles at the same time. The desired cartesian trajectory are shown in Fig.4.3. The second test motion moves the joint 1 and 2 in same direction. Its cartesian trajectory is also presented in Fig.4.4 and trajectory profiles in joint space are shown from Fig.4.5a to Fig.4.5c.

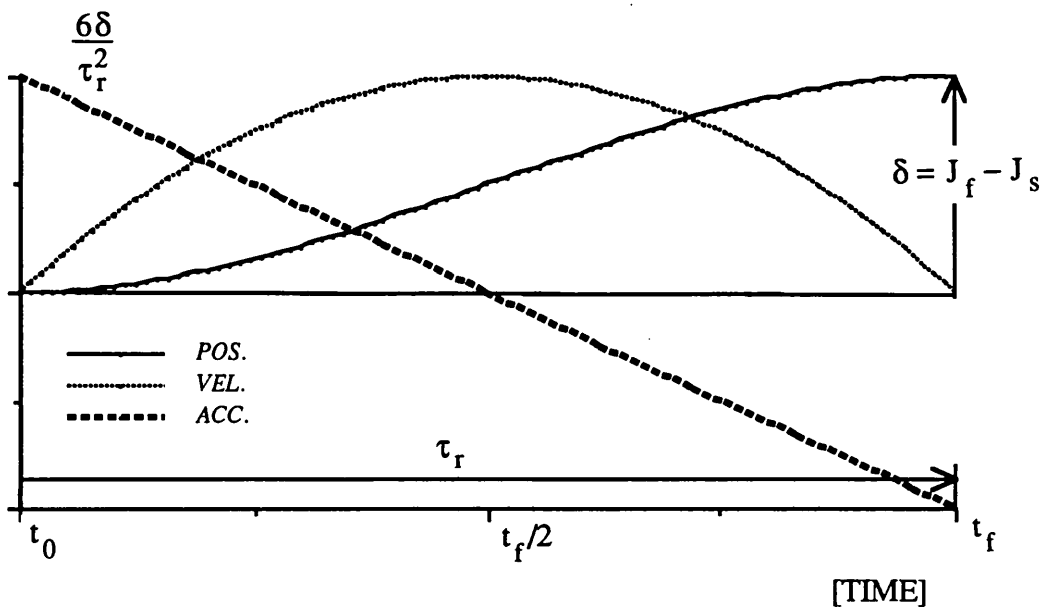

Figure 4.1 Cubic Trajectory($v_o = v_f = 0$)

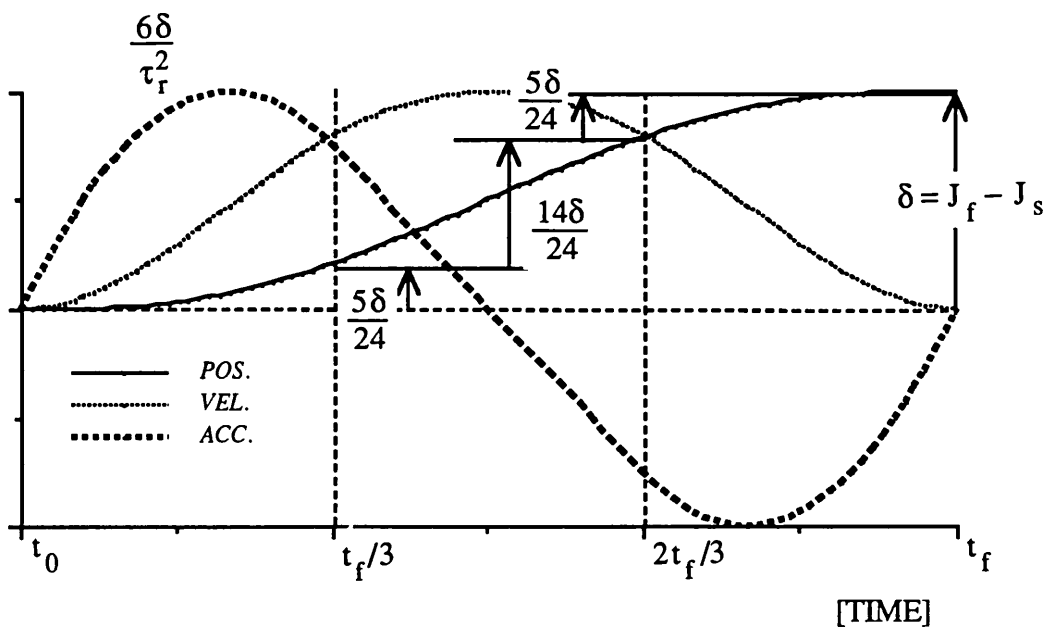

Figure 4.2 Splined Cubic Trajectory($v_o = v_f = 0$, $a_o = a_f = 0$)

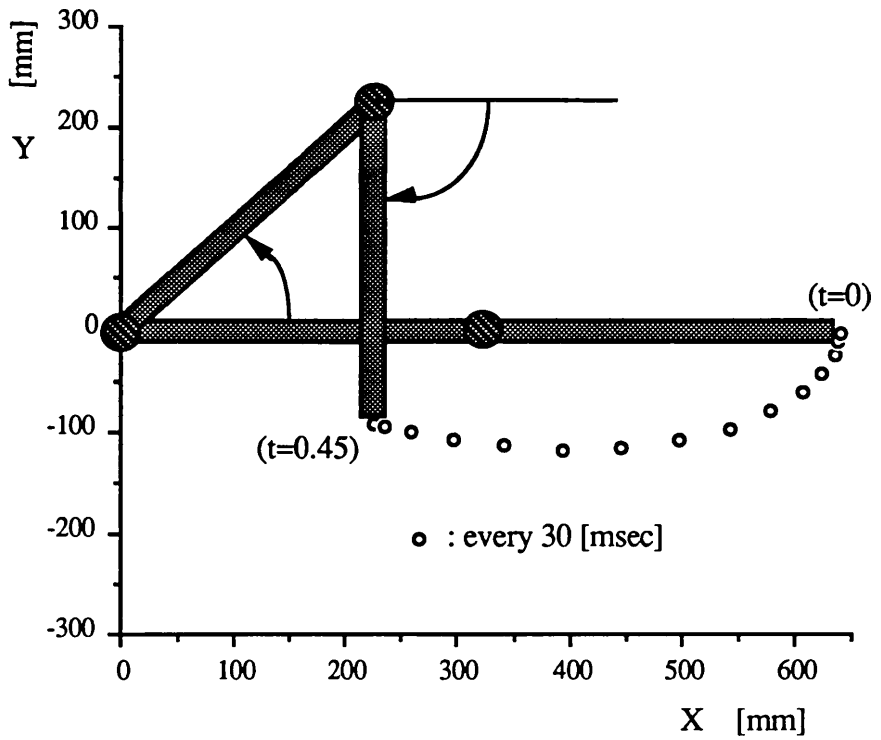

Figure 4.3 Desired Path of Motion I

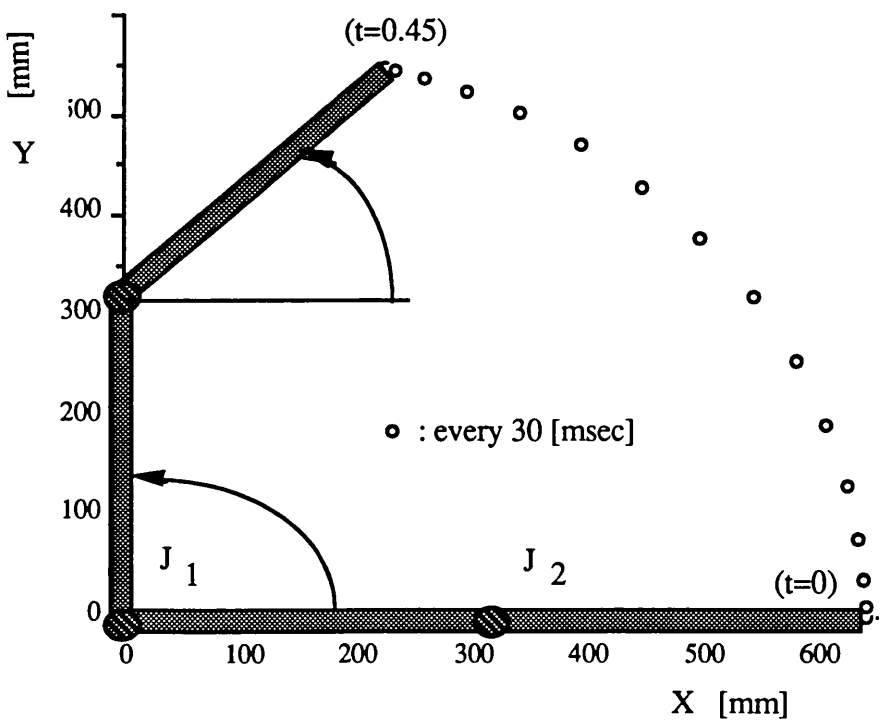

Figure 4.4 Desired Path of Motion II

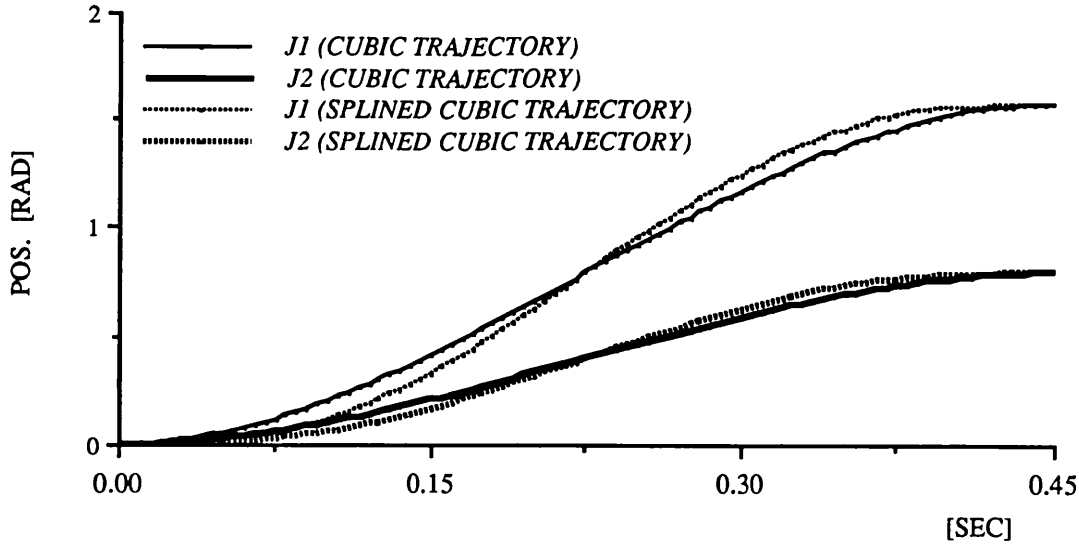

Figure 4.5a Desired Joint Position Profile of Motion II

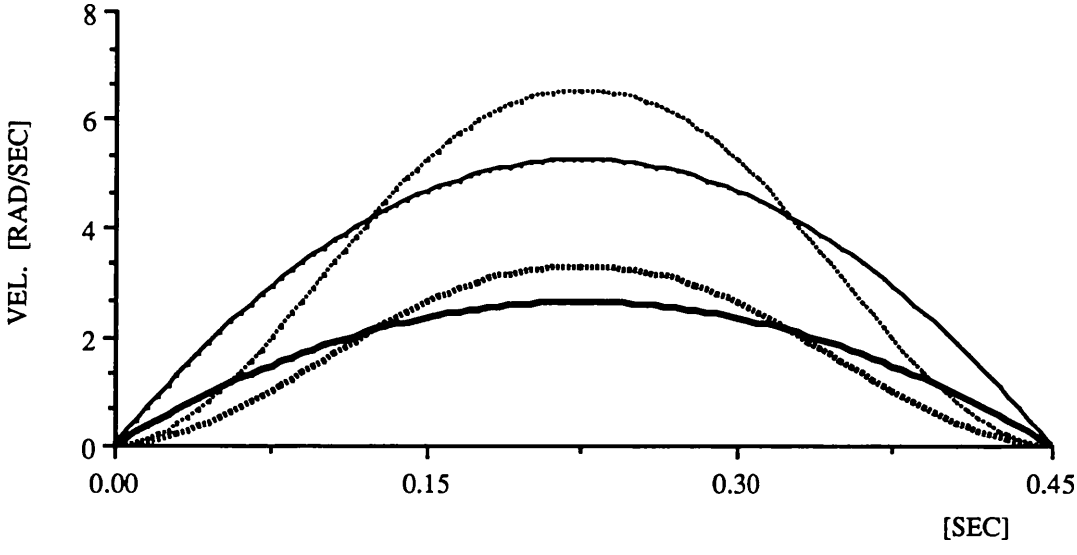

Figure 4.5b Desired Joint Velocity Profile of Motion II

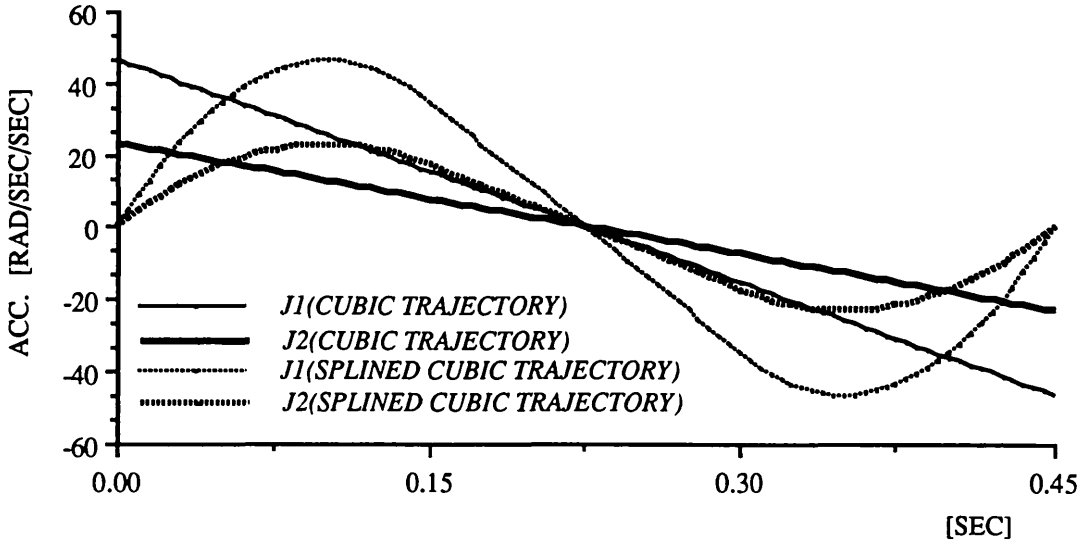

Figure 4.5c Desired Joint Acceleration Profile of Motion II

4.3 Independent Joint Control

Independent joint control is by far the most popular controller for robots. Independent joint control scheme is distinguished from a complicated control scheme based on a dynamic model. In the most of conventional commercial robots, independent joint control is suitable because of high gear ratios and relatively slow movements. A new generation of robots based on direct drive technology is capable of moving at speeds which are an order of magnitude higher than conventional geared robot. The independent control scheme for this kind of high speed robots could not perform well because the control actions are taken for each joint without considering the motions of other joints. However this type of control is still attractive because of its simplicity and can provide a good basis for performance comparison with other complicated control schemes.

In this section, the design procedure of independent joint controller will be presented, and its performance on the IC DDR will be investigated through simulation experiments. The first step in the design of independent joint controller is to obtain the linear decoupled dynamic model for each joint by simply assuming that each joint motion is linear and independent from other joints.

If neglecting the coupling and nonlinear terms from Eq.(2-79), the dynamic model for each joint becomes

$$U_1 = \frac{\theta_1}{\Delta t}\dot{A}_1(k+1) - \frac{\theta_1}{\Delta t}\dot{A}_1(k) + \theta_4\left[\frac{\dot{A}_1(k+1) + \dot{A}_1(k)}{2}\right] \tag{4-7a}$$

$$U_2 = \frac{\theta_3}{\Delta t}\dot{A}_2(k+1) - \frac{\theta_3}{\Delta t}\dot{A}_2(k) + \theta_5\left[\frac{\dot{A}_2(k+1) + \dot{A}_2(k)}{2}\right] \tag{4-7b}$$

Thus the whole system is regarded as two independent linear system. The z-transformation of Eq.(4-7) gives the transfer function as follows.

$$\dot{A}_i(z) = \frac{\alpha_{i2}}{z + \alpha_{i1}} U_i(z), \qquad i = 1, 2 \tag{4-8a}$$

where,

$$\alpha_{11} = \frac{-2\theta_1 + \Delta t\theta_4}{2\theta_1 + \Delta t\theta_4} \tag{4-8b}$$

$$\alpha_{12} = \frac{2\Delta t}{2\theta_1 + \Delta t\theta_4} \tag{4-8c}$$

$$\alpha_{21} = \frac{-2\theta_3 + \Delta t\theta_5}{2\theta_3 + \Delta t\theta_5} \tag{4-8d}$$

$$\alpha_{22} = \frac{2\Delta t}{2\theta_3 + \Delta t\theta_5} \tag{4-8e}$$

The basic structure of the controller for this system is depicted in Fig.4.6. This controller is duplicated for each joint, and control action at each joint is totally independent from other joints.

A reference position A_{id} is compared to an actual position A_i, and the difference is multiplied by a position gain K_{pi} to produce a control signal for the actuator. To

provide stability, a damping term is added to the control signal based on the actual velocity $\dot{A}_i$ multiplied by a velocity gain K_{vi}

$$U_i = K_{pi}(A_{id} - A_i) - K_{vi}\dot{A}_i \tag{4-9}$$

One feature of this feedback control is that there is heavy damping action during the fast motion segment. To remedy this situation and eliminate the following error due to a constant velocity demand, proportional-derivative feedback with velocity reference is proposed in Fig.4.7. This requires the desired velocities to be specified by a trajectory planner. Now the control signal to the actuator is modified as follows.

$$U_i = K_{pi}(A_{id} - A_i) - K_{vi}(\dot{A}_{id} - \dot{A}_i) \tag{4-10}$$

These two types of independent joint control do not include the dynamic model of the robot and are purely driven by the error signal. From the block diagram Fig.4.6, the transfer function relating output variable $A_i(z)$ to the reference input $A_{id}(z)$ is given as follows.

$$A_i(z) = \frac{\frac{2}{\Delta t}K_{pi}H_i(z)}{1 + \frac{2}{\Delta t}K_{pi}H_i(z)} A_{id}(z) \tag{4-11}$$

When a velocity reference is added, the input-ouput relationship is modified as,

$$A_i(z) = \frac{\frac{2}{\Delta t}K_{pi}H_i(z)}{1 + \frac{2}{\Delta t}K_{pi}H_i(z)} A_{id}(z) + \frac{K_{pi}H_i(z)}{1 + \frac{2}{\Delta t}K_{pi}H_i(z)} \dot{A}_{id}(z) \tag{4-12}$$

where $H_i(z)$ is the forward-path transfer function from $E_{pi}(z)$ to $A_i(z)$, and represented as

$$H_i(z) = \frac{\alpha_{i2}\frac{\Delta t}{2}(z+1)}{[z + (\alpha_{i1} + \alpha_{i2}K_{vi})](z-1)} \tag{4-13}$$

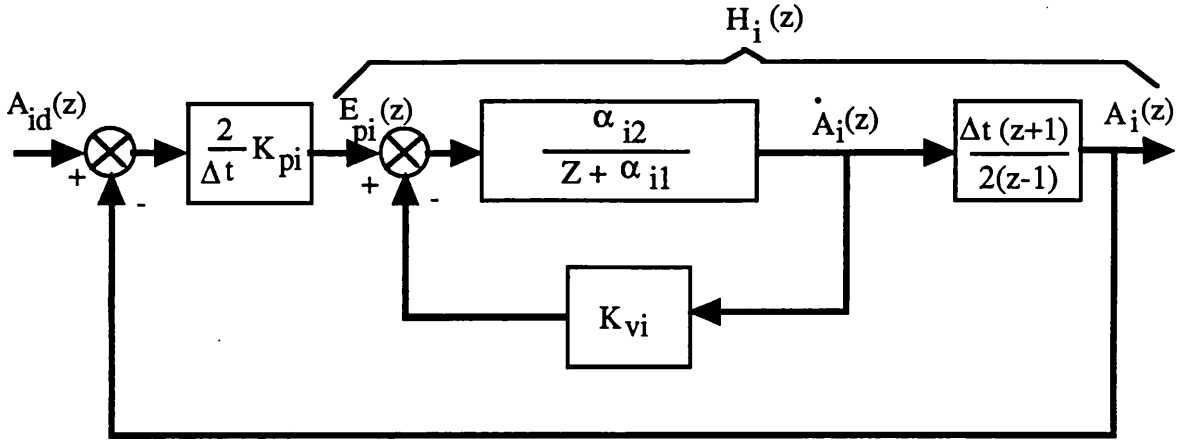

Figure 4.6 Independent Joint Control Without Velocity Reference

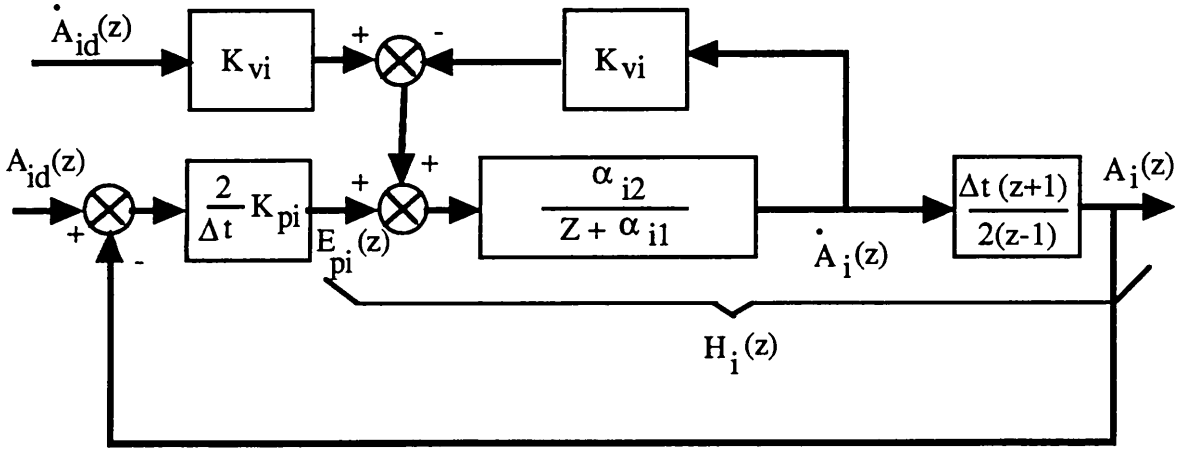

Figure 4.7 Independent Joint Control With Velocity Reference

The characteristic equation contains the information necessary to determine basic characteristics of the system response, and in this case the characteristic equation of both controllers are identical regardless of the inclusion of additional feedforward of the velocity reference inputs. The pole locations are determined by the characteristic equation as below.

$$1+\frac{2}{\Delta t}K_{pi}H_i(z)=0=z^2+d_1z+d_2 \tag{4-14a}$$

$$d_1=\alpha_{i1}+\alpha_{i2}K_{vi}+\alpha_{i2}K_{pi}-1 \tag{4-14b}$$

$$d_2=-(\alpha_{i1}+\alpha_{i2}K_{vi})+\alpha_{i2}K_{pi} \tag{4-14c}$$

There are two gains K_{pi} and K_{vi} which can be adjusted to select the pole location for desirable system responses. When the poles of the transfer function are placed at z_1 and z_2, the position and velocity gains K_{pi} and K_{vi} for each joint are computed according to the following equations.

$$K_{pi}=\frac{1}{2\alpha_{i2}}[z_1z_2-(z_1+z_2)+1] \tag{4-15a}$$

$$K_{vi}=\frac{1}{2\alpha_{i2}}[1-(z_1+z_2)-z_1z_2-2\alpha_{i1}] \tag{4-15b}$$

On the other hand, using the relationship $e^{sT}=z$, the coefficients of characteristic equation can be expressed in terms of damping factor ζ and undamped natural frequency ω_n of the second order system,

$$d_1=-2e^{-\zeta\omega_nT}\mathrm{COS}(\omega_n\sqrt{1-\zeta^2}\ T) \tag{4-16a}$$

$$d_2=\left(\frac{\alpha}{2}\right)[\mathrm{TAN}^2(\omega_n\sqrt{1-\zeta^2}\ T)+1] \tag{4-16b}$$

Since excessive overshoot is usually undesirable in the control of robots, the gains have to be adjusted such that the system is critically damped or slightly overdamped. To

prevent exciting structural vibrations and system stability, the undamped angular natural frequency must be limited to at least the half of the structural resonance frequency of the arm[Paul 82]. IC DDR has its first resonance frequency at near 16 [Hz][KIT 89]. If the system is designed to be critically damped with the undamped natural frequency of 8 [Hz], the desired pole location can be obtained from Eq.(4-16). The gain values based on this pole location are listed at TABLE 4.1.

TABLE 4.1 Feedback Gains (pole location=0.778)

	K_{pi} ($\times 10^3$)	K_{vi} ($\times 10^4$)
Joint 1	1.061	1.717
Joint 2	0.498	0.763

Fig. 4.8a shows how closely the responses follow the reference input (ramp) for joint 1, and Fig. 4.8b represents the position tracking errors. The response profile of independent joint control without the velocity reference lags behind the reference position input. The inclusion of the velocity reference significantly reduces such tracking errors as can be seen in Fig.4.8b.

Fig.4.9 and 4.10 show the position tracking error curves when this independent joint controller(with velocity reference) is applied to the IC DDR at sampling rate of 200 [Hz]. In these graphs, the tracking error is defined to be the cartesian distance between the desired and actual end position of the arm. Test motion I is slightly slower than test motion II, but still capable of providing insight into the influence of dynamic interaction during the motion in the independent joint control scheme. In both test motions, large tracking errors occur. These results illustrate that in the presence of substantial interaction between the joints, independent joint control scheme is not adequate for following the prescribed trajectory satisfactorily.

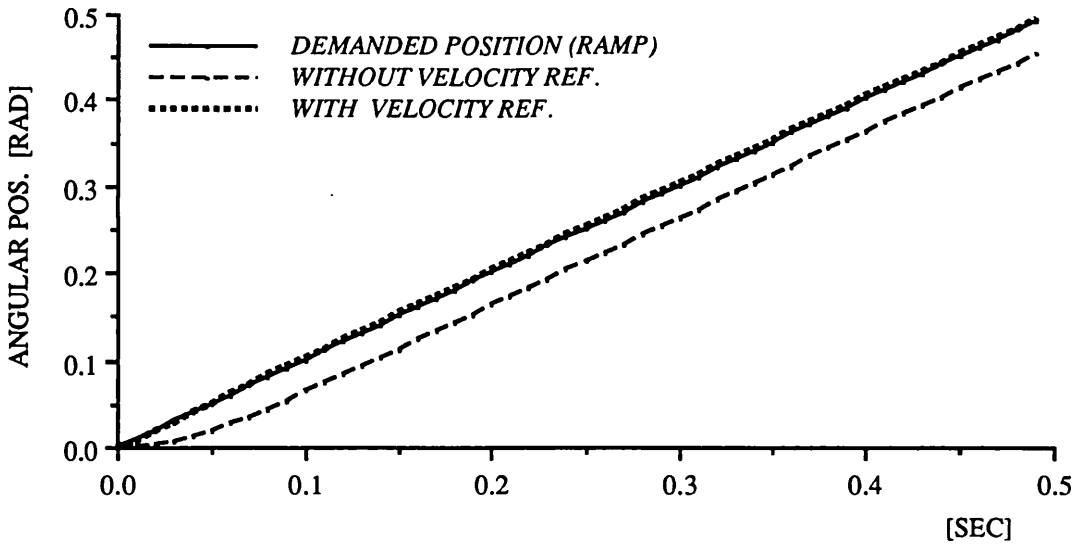

Figure 4.8a Ramp Responses (Joint 1)

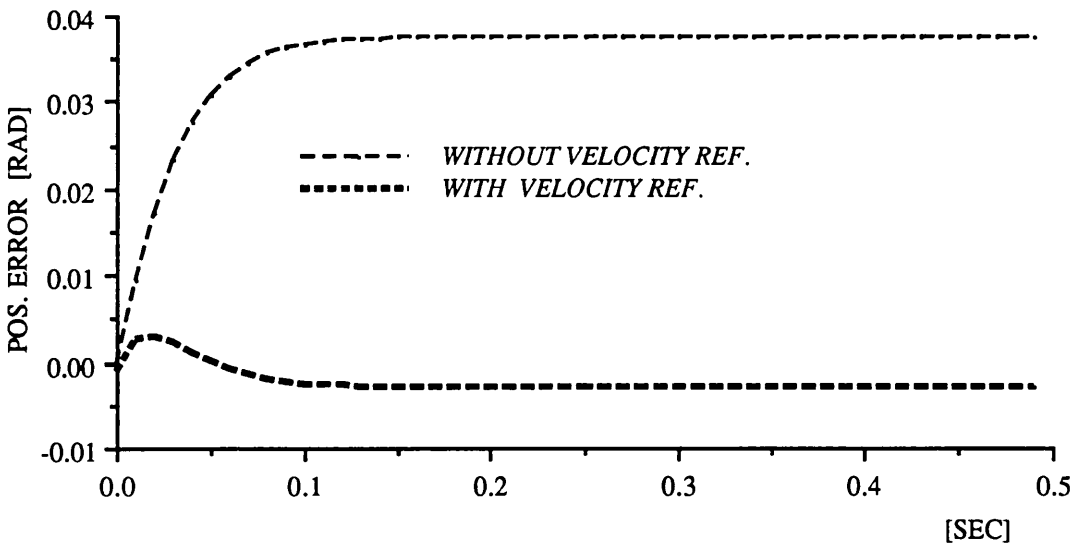

Figure 4.8b Response Error (Joint 1)

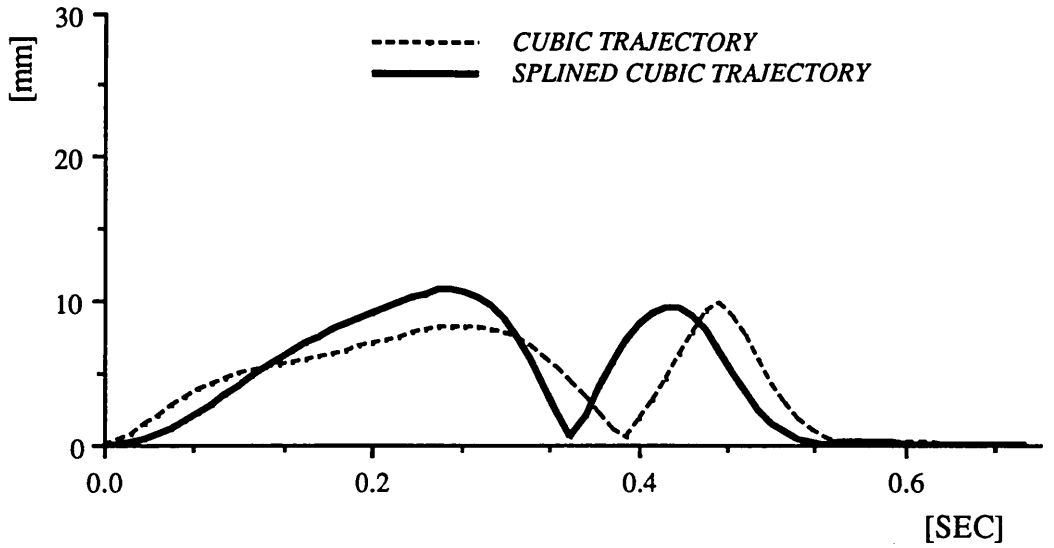

Figure 4.9 Tip Position Tracking Error of Independent Joint Control (Motion I)

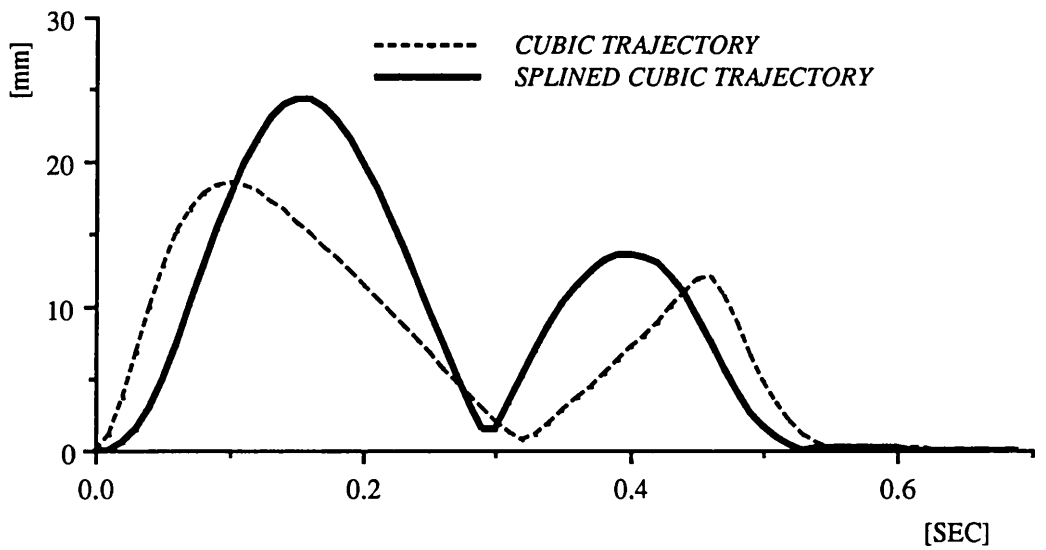

Figure 4.10 Tip Position Tracking Error of Independent Joint Control (Motion II)

4.4 Open Loop Performance of Inverse Dynamic Models

An important role for a robot dynamic model is in the generation of the joint actuating inputs required to follow the prespecified trajectory profile. Through the inverse dynamic computation, a prediction can be made about the required inputs for each joint to follow the prespecified trajectory. If the dynamic model is exact and external disturbances are absent, the application of actuating inputs through the inverse dynamic computation can accomplish an accurate trajectory tracking. In this section, the inverse dynamic performance of the newly developed discrete dynamic model (**VD** : Eq.2-79) will be compared with the continuous-time dynamic model (**SC** : Eq.2-65) and the conventional discrete dynamic model(**GD** : Eq.3-3). If the resulting trajectory caused by the actuating inputs of inverse dynamic computation does not satisfy the prescribed trajectory, these tracking errors will become the burden to be corrected by an additional feedback controller for driving the robot back to the desired trajectory. However, for some robots which move slowly and have weak dynamic interactions between joints, the inverse dynamic computation will become a lesser factor.

In an ideal world where the robot model and needed parameters are known exactly, the open loop control through the inverse dynamic computation would suffice for control purposes. But in a practical situation, the dynamic model can only be an approximation of the actual dynamics, and particularly the parameters of the dynamic model are difficult to identify precisely. Therefore the actuating inputs provided by the inverse dynamic computation are only assumed to reduce the strong interactions between joints as much as possible and to make the system appear to be almost uncoupled.

In the dynamic compensation control scheme, the decoupling performance by the inverse dynamic computation would have a decisive influence on the overall control quality. The open loop tracking performances of the three dynamic models (**SC**, **GD**, **VD**) are evaluated with reference to the desired trajectory. The desired trajectory supplies the demanded states of each joint to the inverse dynamic model at a sampling period of 5 msec, and the generated actuating inputs for joints are kept piecewise constant over the sampling intervals.

Fig.4.11 shows the open loop tracking performance of the arm by the inverse dynamic computation when test motion I with cubic interpolation trajectory is demanded. The inverse dynamic model **GD** and **VD** do not require the acceleration information of the desired trajectory, while the continuous-time dynamic model needs the acceleration to generate the actuating signals for joints. In principle, the inverse dynamic calculation of the continuous-time dynamic model relies upon the continuous-time trajectory data. But in the computer-controlled system, the actuating input signals are computed and applied at equally spaced sampling instants. Therefore the changes of the desired states over the sampling period can not be reflected on the computation of actuating inputs in the continuous-time dynamic model. Thus the inverse dynamic calculation by continuous-time dynamic model can not provide the accurate actuating signals until the sampling rate is sufficiently increased such that the changes of the state over a sampling period are negligible. The dynamic model **GD**, though the acceleration information is not needed in the inverse dynamic computation, is also not expected to generate the accurate actuating inputs to follow the demanded trajectory because the simple numerical approximation rule is used to replace the acceleration terms.

In contrast to the two inverse dynamic models (**SC**, **GD**) mentioned above, the inverse dynamic calculation based on the model **VD** results in small tracking errors for following the demanded trajectory. Fig. 4.12 represents the actuating input profile generated by the inverse dynamic model **VD**. The actuating input profiles by the model **SC** and **GD** also show the same pattern except the small magnitude differences. Thus those graphs are not included for clarity. The sudden changes of actuating input profiles in the vicinity of the final time of the demanded trajectory (t=0.45 sec) occur due to the discontinuity of the demanded acceleration.

Fig.4.13 shows the open loop position error under the same test motion but the splined cubic trajectory is applied instead of the cubic trajectory. In this case, the actuating inputs are smoothly changed (Fig.4.14). Inverse dynamic model **GD** and **VD** exhibit similar performance regardless of the interpolation types of the trajectory. Particularly it can be observed that the performance of the inverse dynamic model **SC** is very sensitive to the selection of interpolation type in motion generation. This simulation results reveal that the smooth acceleration profile is important to the performance of the inverse dynamic model **SC**.

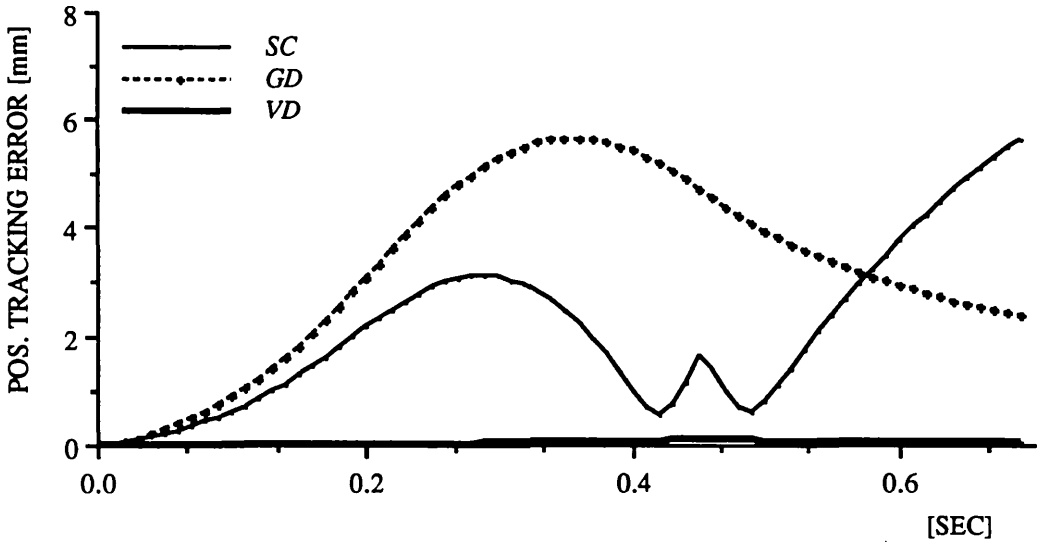

Figure 4.11 Tip Position Tracking Error(Motion I, Cubic Trajectory)

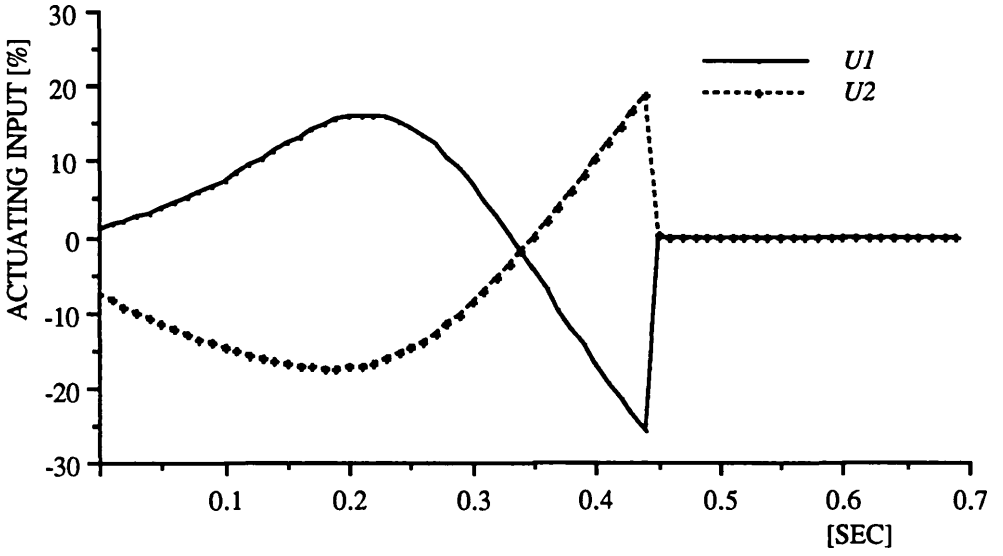

Figure 4.12 Actuating Inputs By Inverse Dynamic Model **VD**
(Motion I, Cubic Trajectory)

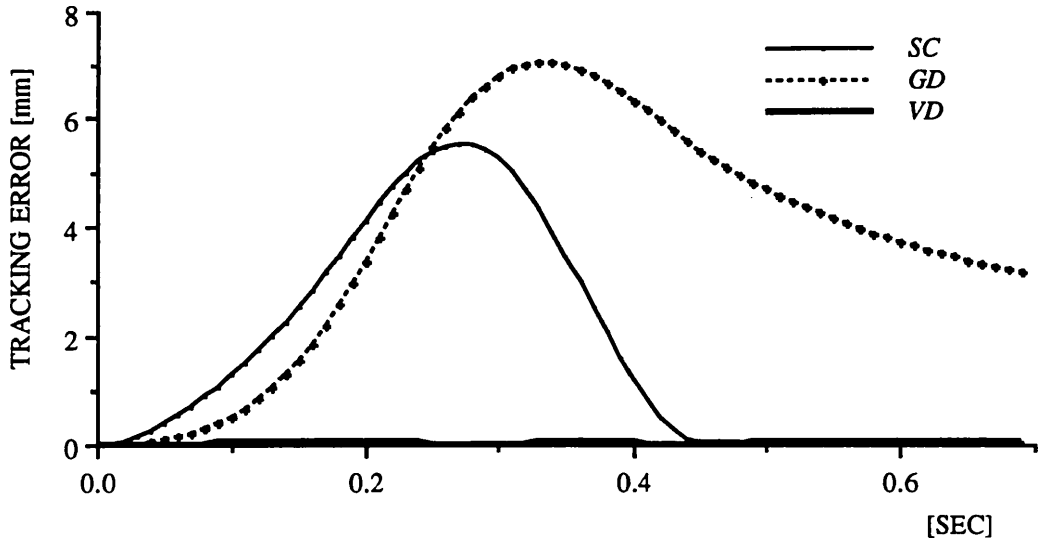

Figure 4.13 Tip Position Tracking Error(Motion I, Splined Cubic Trajectory)

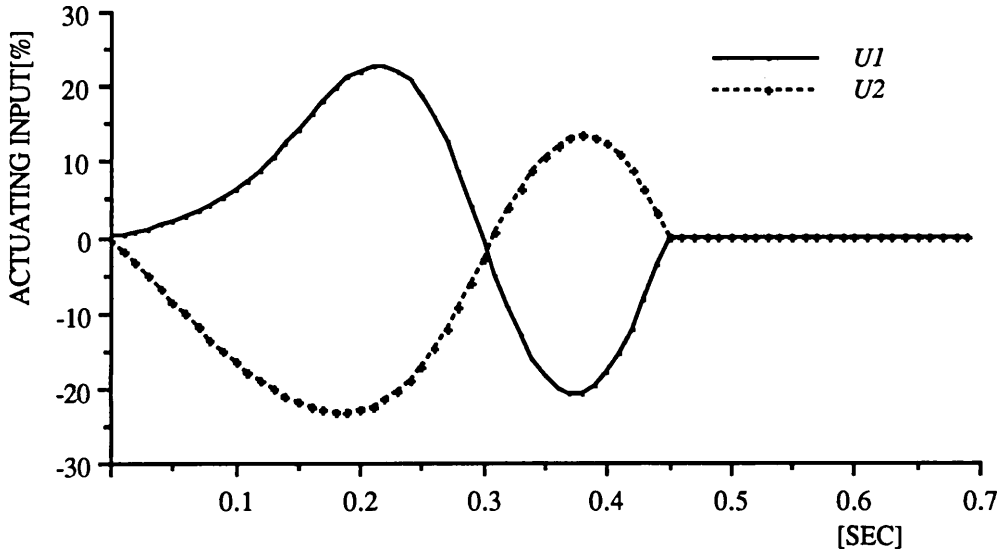

Figure 4.14 Actuating Inputs By Inverse Dynamic Model **VD**
(Motion I, Splined Cubic Trajectory)

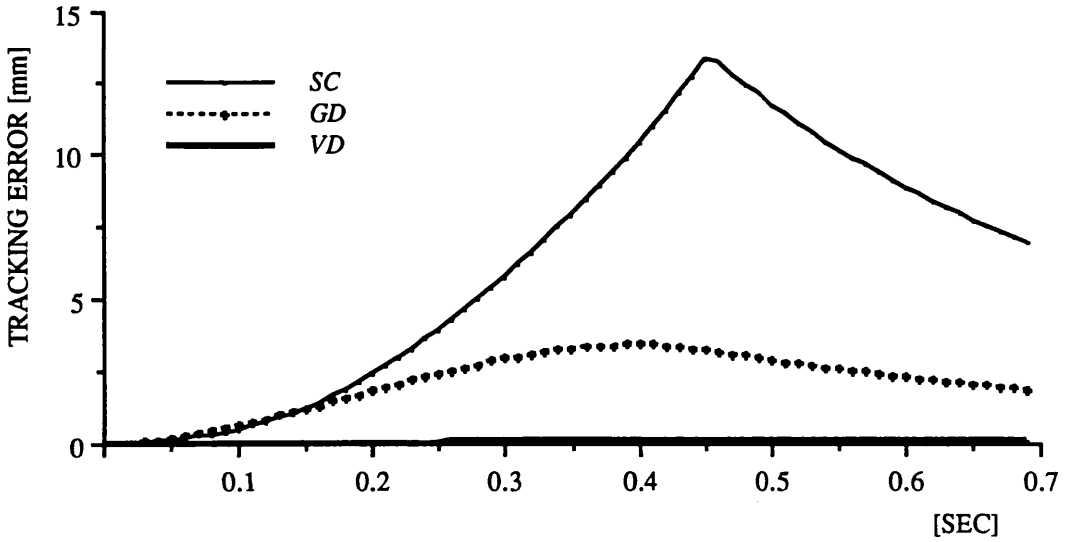

Figure 4.15 Tip Position Tracking Error(Motion II, Cubic Trajectory)

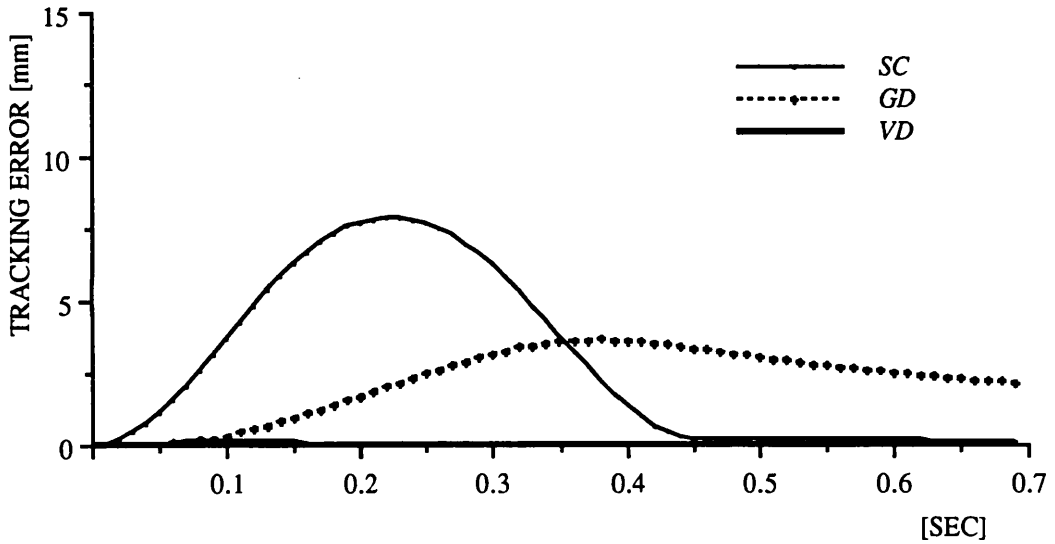

Figure 4.16 Tip Position Tracking Error(Motion II, Splined Cubic Trajectory)

TABLE 4.2 Maximum Errors

Motion	Trajectory Type	Model	Angular Position [DEG]		Angular Velocity [DEG]		Tip Position Error [mm]
			Joint 1	Joint 2	Joint 1	Joint 2	
Motion I	Cubic Trajectory	SC	0.714	-0.471	-3.45	6.92	5.65
		GD	0.432	-0.772	2.50	-4.36	5.63
		VD	0.004	-0.011	0.044	-0.076	0.081
	Splined Cubic Trajectory	SC	0.468	-0.941	3.41	-6.81	5.54
		GD	0.543	-0.955	3.59	-6.10	7.07
		VD	-0.006	0.011	0.072	-0.115	0.054
Motion II	Cubic Trajectory	SC	-1.29	-1.29	-7.74	-4.64	13.3
		GD	0.416	0.342	1.93	1.54	3.38
		VD	0.014	0.005	0.017	-0.019	0.095
	Splined Cubic Trajectory	SC	0.953	0.482	6.77	3.38	7.87
		GD	0.460	0.436	2.76	2.75	3.66
		VD	0.008	-0.005	0.035	-0.049	0.055

Fig.4.15 and 4.16 show the open loop position error under the test motion II with the different interpolation types. The inverse dynamic model **VD** is still showing its outstanding performance when compared with the two other inverse dynamic models. The velocity error profiles are not presented here for brevity, but the maximum errors in joint angle, joint velocity and tip position of the arm are summarized in TABLE 4.2. From this table, it can be seen that the inverse dynamic model **VD** results in almost 100 times smaller errors than the other inverse dynamic models. Conclusively, the simulation experiments in this section clearly illustrate the efficacy of the dynamic model **VD** in the inverse dynamic application for its potential usage in control.

4.5 Feedforward Dynamic Compensation

The first step towards the improvement of control performance is to add the feedforward term in the basic independent joint controller. Since the poor tracking performance of independent joint controller is mainly due to not taking account of the nonlinear and coupling effects, incorporating feedforward terms for cancelling dynamic interactions into the independent joint controller could improve the control performance. The feedforward computation predicts the actuating inputs to compensate the nonlinear and coupling terms between the joints. Hence the actuating signals for the joints are augmented as the sum of feedforward part and the feedback part of independent joint control.

$$\mathbf{U} = \mathbf{U}_f + \mathbf{K}_v\dot{\mathbf{E}} + \mathbf{K}_p\mathbf{E} \tag{4-17a}$$

Here $\mathbf{U}_f$ is a feedforward actuating input which may allow the system dynamics to be decoupled , and is calculated as below:

$$\mathbf{U}_f = \frac{\theta_2}{\Delta t}\begin{pmatrix} C_{2\bar{1}}(k+1)\dot{A}_{2d}(k+1) - C_{2\bar{1}}(k)\dot{A}_{2d}(k) \\ C_{2\bar{1}}(k+1)\dot{A}_{1d}(k+1) - C_{2\bar{1}}(k)\dot{A}_{1d}(k) \end{pmatrix} + \tag{4-17b}$$

$$\frac{\theta_2[\dot{A}_{1d}(k+1)\dot{A}_{2d}(k) + \dot{A}_{1d}(k)\dot{A}_{2d}(k+1)][C_{2\bar{1}}(k+1) - C_{2\bar{1}}(k)]}{2\{[A_{1d}(k+1) - A_{1d}(k)] - [A_{2d}(k+1) - A_{2d}(k)]\}}\begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

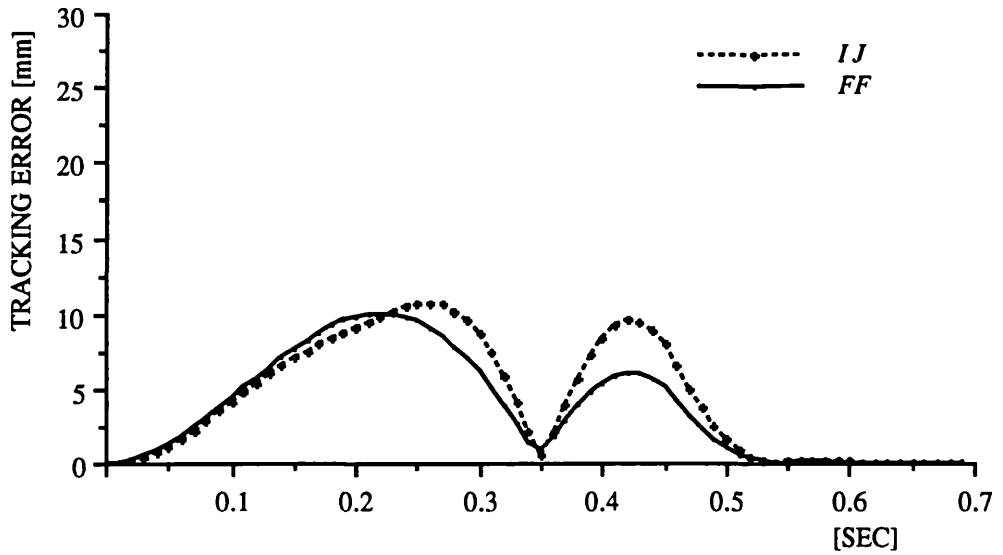

Figure 4.17 Tip Position Tracking Error of **FF** Scheme
(Motion I, Splined Cubic Trajectory)

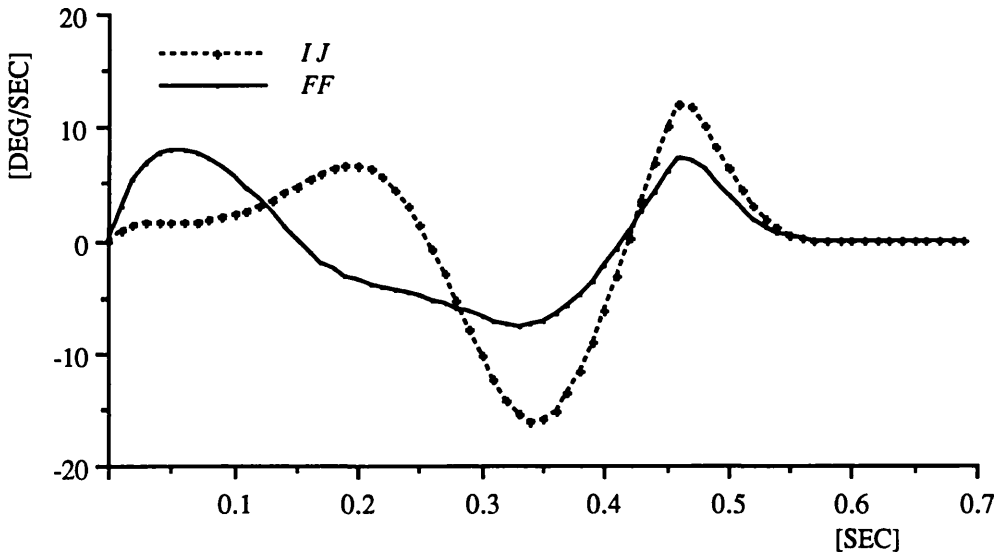

Figure 4.18a Velocity Error of **FF** Scheme For Joint 1
(Motion I, Splined Cubic Trajectory)

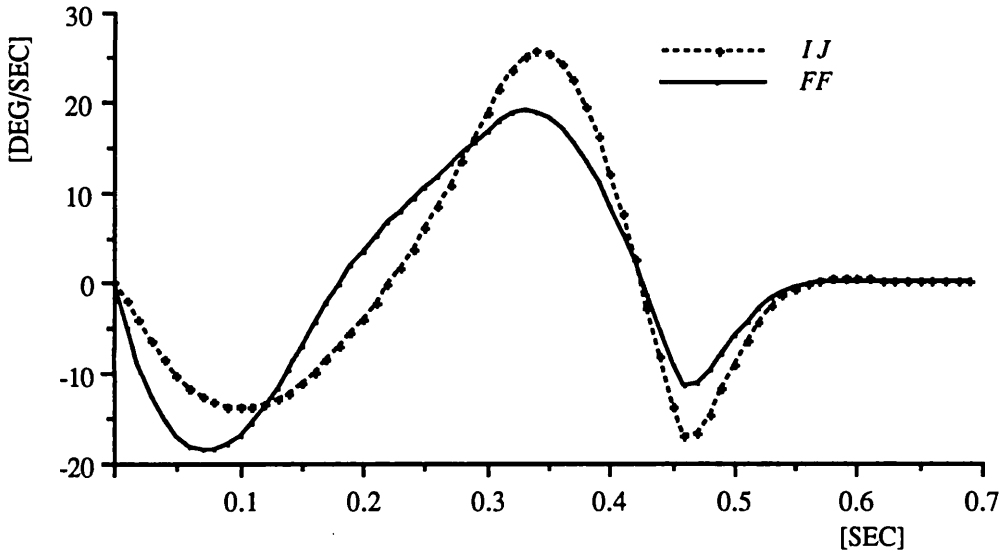

Figure 4.18b Velocity Error of **FF** Scheme For Joint 2
(Motion I, Splined Cubic Trajectory)

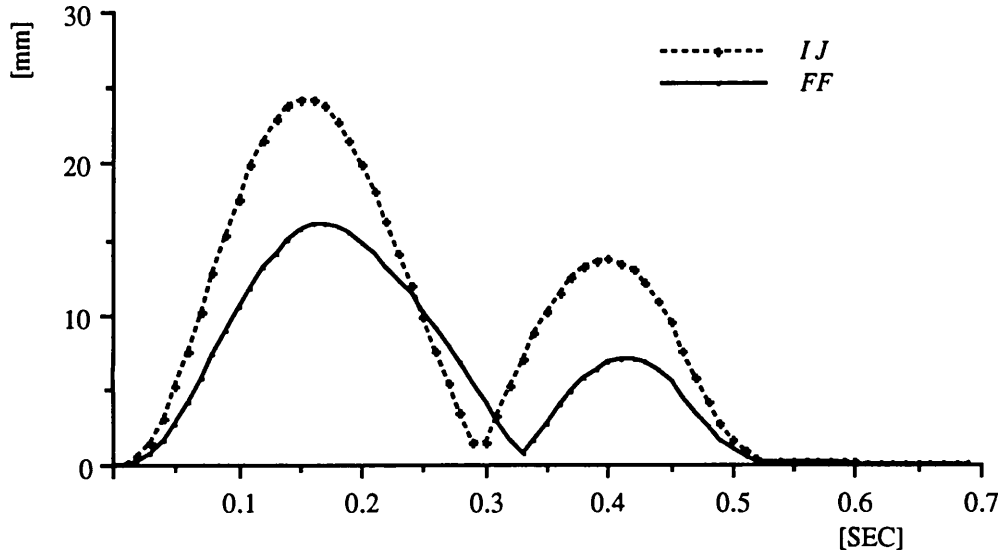

Figure 4.19 Tip Position Tracking Error of **FF** Scheme
(Motion II, Splined Cubic Trajectory)

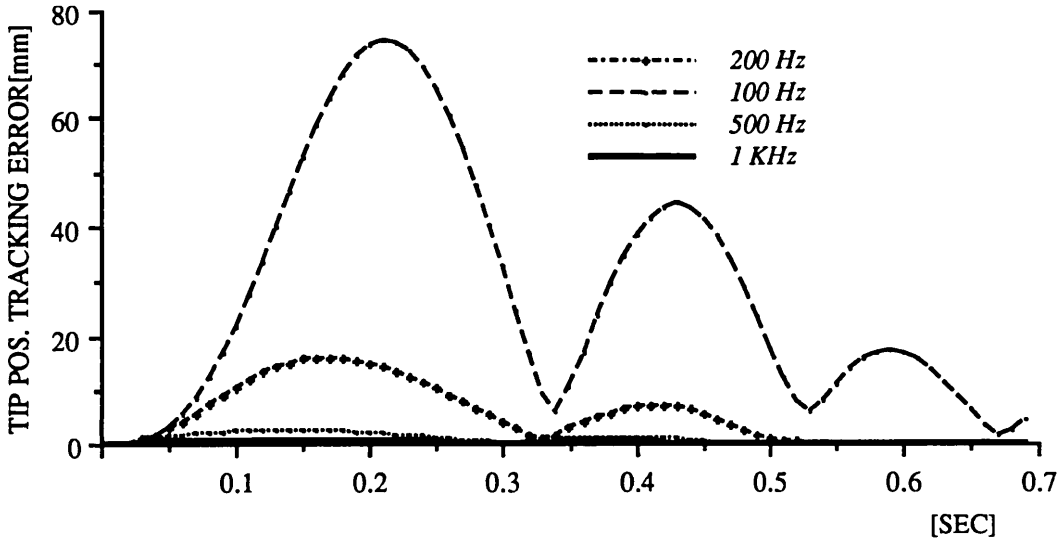

Figure 4.20 Influence of Sampling Frequency On **FF** Scheme
(Exact Parameters, Motion II, Splined Cubic Trajectory)

TABLE 4.3 Maximum Tip Position Error

Motion	Trajectory	Max. Tip Position Error [mm]	
		I J	**F F**
I	Cubic	9.81	7.46
	Splined Cubic	10.7	10.0
II	Cubic	18.5	11.7
	Splined Cubic	24.2	16.0

TABLE 4.4 Tip Position Errors With Different Sampling Frequencies
(Motion II, Splined Cubic Trajectory)

Frequency[Hz]	Max. Error [mm]
100	74.4
200	16.0
500	2.63
1000	0.654

Since these feedforward terms are functions of the desired states only, they can be computed off-line. In the past, this was an advantage over the real-time dynamic compensation scheme, but nowadays, it is not necessarily an important issue because of the substantial increases in the computational power of the control system.

This feedforward controller (**FF**) reduces the peak trajectory errors as summarized in TABLE 4.3. Fig.4.17 and 4.19 compare the tracking performance of feedforward control scheme with the independent joint control scheme under the different test motions. As can be observed in Fig.4.18a and 4.18b, the velocity tracking errors for joint 1 and 2 are also decreased by incorporating the feedforward terms. Adding the feedforward term has surely contributed to the reduction of the trajectory errors but does not help as much as was initially hoped.

Fig. 4.20 shows the effect of sampling rate on the tip position tracking error of the arm when test motion II (with splined cubic trajectory) is demanded. Increasing the control sampling period from 5 [msec] to 10 [msec] results in a big degradation of the tracking accuracy, while an increase in the sampling rate improves the tracking performance significantly. The maximum tracking errors at the tip position of the arm are listed in TABLE 4.4.

The point of weakness on the feedforward control scheme is that the feedback portion of the controller acts independently of the dynamics and probably produces perturbations at neighbouring joints. In other words, a corrective input at one joint perturbs the other joints, whereas ideally speaking, the corrective inputs were intended to cancel the nonlinear and coupling interaction between the joints. This problem can be treated better in the real-time full dynamic compensation scheme presented in the following section.

4.6 Control Partitioning and Decoupling by Inverse Dynamic Model

In a practical situation, even supposing that the inverse dynamic model is exact, it is extremely difficult to identify the true parameters of the model. In addition, there always exists some unpredictable external disturbances which can not be perfectly cancelled. For these practical reasons, robots can not be controlled solely in the open

loop fashion. As mentioned earlier, the control problem for the robot is inherently nonlinear and multivariable problem. This means that much of linear control theory can not be directly applicable.

In this section, the controller design will be restricted to the method using the inverse dynamic model. The complicated nonlinear system will be reduced to the simple decoupled unit mass system through the technique called control law partitioning[Craig 86]. The controller for this system is decomposed into two parts. One part of the control law is the model based part in that it makes use of the dynamic model. This part of the control law sets up the system so that it appears to be a unit mass system. The second part of control law is the error driven part in that it forms error signals by differencing desired and actual states and multiplying these errors by appropriate gains. Thus, the error driven part of the control law is called the servo part. Since the complicated nonlinear system is reduced to a simply decoupled unit mass system, the design of the servo part becomes very simple.

Fig. 4.21 shows the structure of control system through the partitioned control law. The model based part of the control system appears in the form,

$$\mathbf{U}(k) = \mathbf{R}\,\dot{\mathbf{A}}^{*}(k) + \mathbf{S} \tag{4-18}$$

For a system of n degrees of freedom, **U**, $\dot{\mathbf{A}}^{*}$, **S** are [n×1] vectors, and **R** is an [n×n] matrix. The matrix **R** is not necessarily diagonal, but should be chosen to decouple the system dynamic equations. If **R** and **S** are properly chosen, then the system appears to consist of a n independent unit mass system. With this structure of the control law, the result of combining the discrete dynamic model (**VD**) and Eq.(4-18) becomes

$$\mathbf{H}[\mathbf{A}(k+1)]\,\dot{\mathbf{A}}(k+1) - \mathbf{H}[\mathbf{A}(k)]\dot{\mathbf{A}}(k) + \mathbf{C}[\mathbf{A}(k),\mathbf{A}(k+1),\dot{\mathbf{A}}(k),\dot{\mathbf{A}}(k+1)]$$
$$+ \mathbf{D}[\dot{\mathbf{A}}(k),\dot{\mathbf{A}}(k+1)] = \mathbf{R}\,\dot{\mathbf{A}}^{*}(k) + \mathbf{S} \tag{4-19}$$

Clearly, in order to make the system appear as the independent unit mass system from input vector $\dot{\mathbf{A}}^{*}(k)$, **R** and **S** must be chosen as

$$\mathbf{R} = \mathbf{H}[\mathbf{A}(k+1)] \tag{4-20a}$$

$$\mathbf{S} = -\mathbf{H}[\mathbf{A}(k)]\dot{\mathbf{A}}(k) + \mathbf{C}[\mathbf{A}(k), \mathbf{A}(k+1), \dot{\mathbf{A}}(k), \dot{\mathbf{A}}(k+1)] + \mathbf{D}[\dot{\mathbf{A}}(k), \dot{\mathbf{A}}(k+1)] \tag{4-20b}$$

Therefore, the matrix **R** represents inertia matrix of the arm. One important property of the inertia matrix is that all dependence on **A** comes in the form of the trigonometric functions of cosine and sine. Since cosine and sine are bounded for any value of their arguments and they appear only in the numerators of the elements of **H**, where **H** is bounded for all **A.** The several properties of **H** can be stated as [Craig 88].

- symmetric
- positive definite and bounded above and below
- inverse exists and is positive definite and bounded

Putting Eq.(4-20a, b) into Eq.(4-19) and using these properties, the next equation is derived.

$$\dot{\mathbf{A}}(k+1) = \dot{\mathbf{A}}^{*}(k) \tag{4-21}$$

This is the equation of the independent unit mass system. Now it is easy to design the servo part of the control law. Select $\dot{\mathbf{A}}^{*}$ by incorporating the linear feedback ,then

$$\dot{\mathbf{A}}^{*}(k) = \dot{\mathbf{A}}_{d}(k+1) + \mathbf{K}_{v}\dot{\mathbf{E}}(k) + \frac{2}{\Delta t}\mathbf{K}_{p}\mathbf{E}(k) \tag{4-22}$$

here $\dot{\mathbf{A}}_{d}$ is commanded input (one step ahead desired velocity), $\mathbf{K}_v$ and $\mathbf{K}_p$ are diagonal gain matrices and **E** (k) is the error vector, i.e.,

$$\mathbf{E}(k) = \mathbf{A}_{d}(k) - \mathbf{A}(k) \tag{4-23}$$

The factor ($2/\Delta t$) in Eq.(4-22) is introduced so that both $\mathbf{K}_v$ and $\mathbf{K}_p$ are dimensionless. Then the selected feedback law of servo part leads to the closed loop error equation,

$$\dot{\mathbf{E}}(k+1) + \mathbf{K}_{v}\dot{\mathbf{E}}(k) + \frac{2}{\Delta t}\mathbf{K}_{p}\mathbf{E}(k) = \mathbf{O} \tag{4-24}$$

This equation describes the error behaviour of the whole controlled system. Upon applying Trapezoid rule in TABLE 2.5, Eq.(4-24) can be shown to have the characteristic equation as belows:

$$z^2 + (K_{pi} + K_{vi} - 1)z + (K_{pi} - K_{vi}) = 0 \tag{4-25}$$

If the feedback gains K_{pi} and K_{vi} are selected so that the roots (poles) of Eq.(4-25) are located inside the unit disc in z-plane, the stable error behaviour can be achieved. To avoid oscillation and overshoot in the closed-loop system, the roots(poles) should be placed on the real axis between z=0 and z=1. Particularly, in the critical damping case, errors are suppressed in the fastest way which does not cause overshoot. In this situation, the feedback gains have the relationship.

$$(K_{pi} + K_{vi} - 1)^2 = 4\,(K_{pi} - K_{vi}) \tag{4-26}$$

More generally, if the roots of the characteristic equation are placed at z_1 and z_2, the gains K_{pi} and K_{vi} are computed according to the following equations

$$K_{pi} = \frac{1}{2}(1 - z_1)\,(1 - z_2) \tag{4-27a}$$

$$K_{vi} = K_{pi} - z_1 z_2 \tag{4-27b}$$

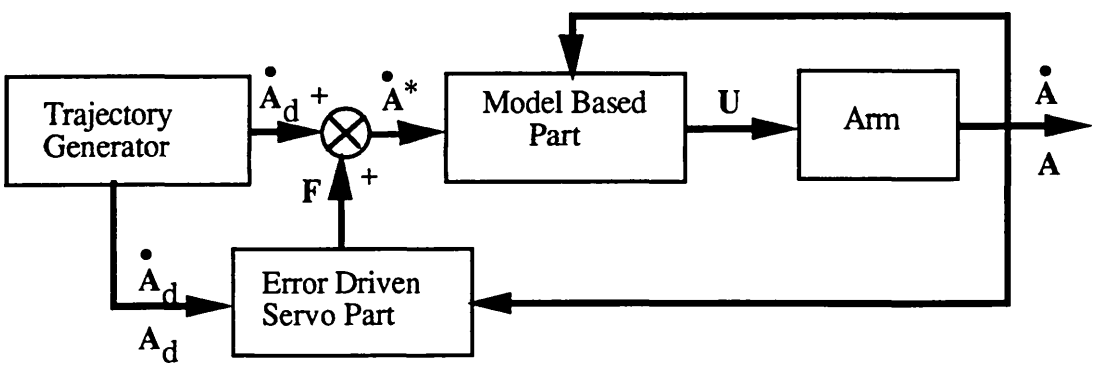

Figure 4.21 Conceptual Block Diagram of Control Scheme Using partitioned Control·Law

4.7 Real-time Dynamic Compensation

The derivation of control law shown in the last section is performed on the basis of two implicit assumptions:

- perfect knowledge of system parameters
- all state information is available

In the actual implementation stage, these assumptions could not be realized. The perfect values of parameters in the dynamic model can not be known. Especially, the implementation of control law requires the unavailable state $[\mathbf{A}(k+1), \dot{\mathbf{A}}(k+1)]$ at the k-th sampling instant for the on-line computation. Hence practical aspects have to be considered for realizable implementation.

Instead of the unavailable data $[\mathbf{A}(k+1), \dot{\mathbf{A}}(k+1)]$ and the perfect values of parameters, the unavailable states $[\mathbf{A}(k+1), \dot{\mathbf{A}}(k+1)]$ are replaced by the desired states $[\mathbf{A}_d(k+1), \dot{\mathbf{A}}_d(k+1)]$, and the estimated values of parameter are used in the computation of model based part. With these reasonable replacements, the ideal form of Eq.(4-20) is turned into the realizable forms as below:

$$\hat{\mathbf{R}} = \hat{\mathbf{H}}[\mathbf{A}_d(k+1)] \tag{4-28a}$$

$$\begin{aligned}\hat{\mathbf{S}} = &-\hat{\mathbf{H}}[\mathbf{A}(k)]\dot{\mathbf{A}}(k) + \hat{\mathbf{C}}[\mathbf{A}(k), \mathbf{A}_d(k+1), \dot{\mathbf{A}}(k), \dot{\mathbf{A}}_d(k+1)] \\ &+ \hat{\mathbf{D}}[\dot{\mathbf{A}}(k), \dot{\mathbf{A}}_d(k+1)]\end{aligned} \tag{4-28b}$$

The caret '^' symbolizes the estimated parameter values in contrast to the perfect values. By a similar procedure to the previous section, Eq.(4-28) leads to the closed-loop system equation:

$$\dot{\mathbf{A}}(k+1) = \dot{\mathbf{A}}^*(k) - \Gamma(k+1) \tag{4-29}$$

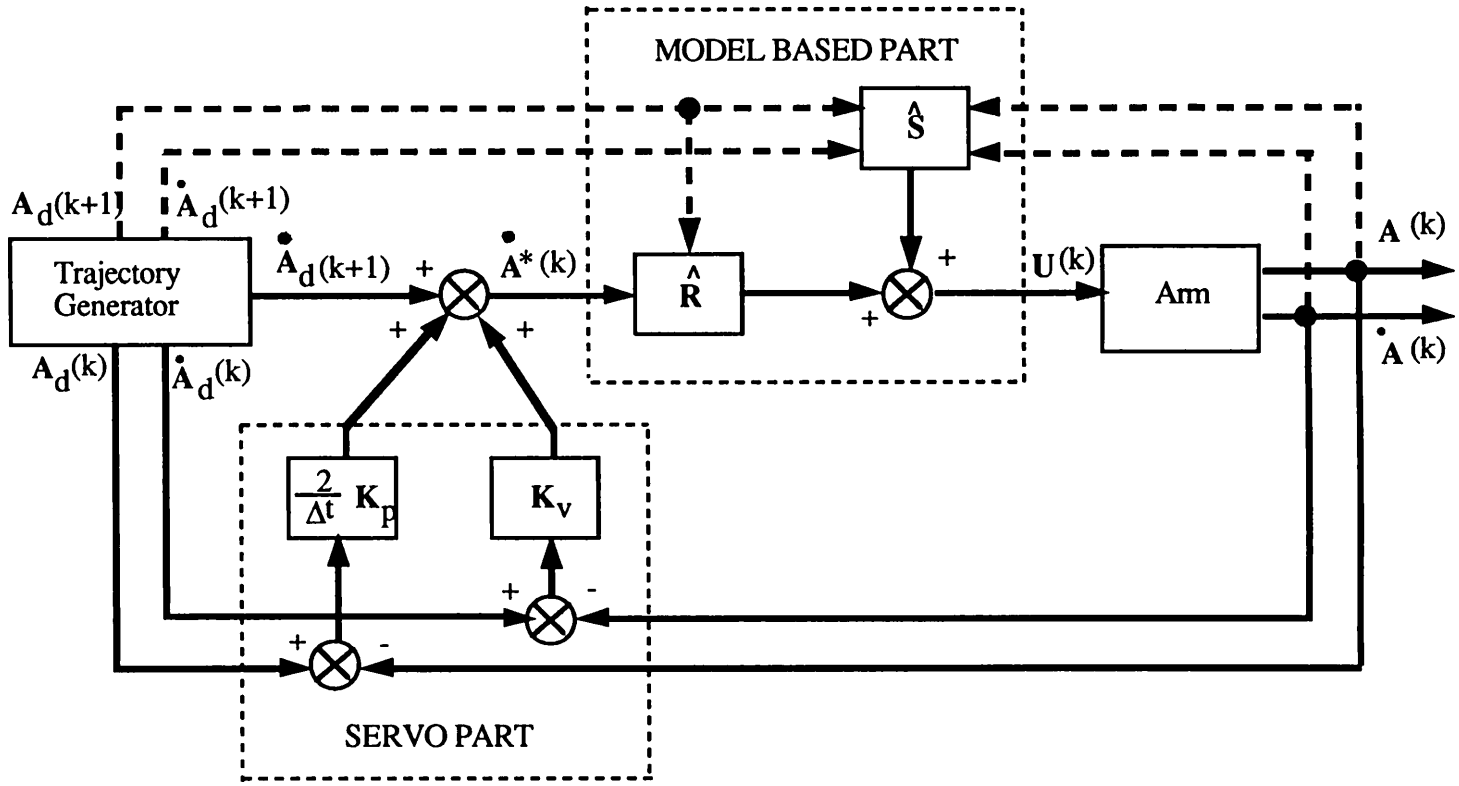

Figure 4.22 Realizable Implementation of The Model Based Control Scheme

Where $\Gamma(k+1)$ is the nonlinear error driving vector due to the errors in parameters and errors arising from the replacements $[\mathbf{A}(k+1),\ \dot{\mathbf{A}}(k+1)] \rightarrow [\mathbf{A}_d(k+1),\ \dot{\mathbf{A}}_d(k+1)]$. Its detailed expression is given in the next equation.

$$\begin{aligned}\Gamma(k+1) = \hat{\mathbf{H}}[\mathbf{A}_d(k+1)]^{-1}\Big\langle \Big\{\mathbf{H}[\mathbf{A}(k+1)] - \hat{\mathbf{H}}[\mathbf{A}_d(k+1)]\Big\}\dot{\mathbf{A}}(k+1) - \\ \Big\{\mathbf{H}[\mathbf{A}(k)] - \hat{\mathbf{H}}[\mathbf{A}_d(k)]\Big\}\dot{\mathbf{A}}(k) + \Big\{\mathbf{C}[\mathbf{A}(k), \mathbf{A}(k+1), \dot{\mathbf{A}}(k), \dot{\mathbf{A}}(k+1)] \\ - \hat{\mathbf{C}}[\mathbf{A}(k), \mathbf{A}_d(k+1), \dot{\mathbf{A}}(k), \dot{\mathbf{A}}_d(k+1)]\Big\} + \Big\{\mathbf{D}[\dot{\mathbf{A}}(k), \dot{\mathbf{A}}(k+1)] \\ - \hat{\mathbf{D}}[\dot{\mathbf{A}}(k), \dot{\mathbf{A}}_d(k+1)]\Big\}\Big\rangle \qquad (4\text{-}30)\end{aligned}$$

Even if the model and its constituent parameters are exact, the error driving vector $\Gamma(k+1)$ remains non-zero because of the aforementioned replacements for the states. Another property of this error driving vector is its inaccessibility at k-th sampling instant because it depends upon the actual states at (k+1)-th sampling instant.

Combining $\dot{\mathbf{A}}^*$ in Eq.(4-22) and the Trapezoid rule to Eq.(4-29) results in the following difference equation for the error vector $\mathbf{E}(k)$.

$$\begin{aligned}&\mathbf{E}(k) + [\mathbf{K}_p + \mathbf{K}_v - \mathbf{I}]\mathbf{E}(k-1) + [\mathbf{K}_p - \mathbf{K}_v]\mathbf{E}(k-2) \\ &= \frac{\Delta t}{2}[\Gamma(k) + \Gamma(k+1)] \qquad (4\text{-}31)\end{aligned}$$

The feedback gain matrices $\mathbf{K}_v$ and $\mathbf{K}_p$ can be selected in the same way as discussed in the previous section under the assumption that the error driving vector is zero. Fig.4-22 shows the possible implementation of this model based control scheme. Thin dashed lines indicates that $\hat{\mathbf{R}}$ and $\hat{\mathbf{S}}$ are functions of the current states and the desired states(one-step ahead).

4.7.1 Full Dynamic Compensation

In the full dynamic compensation control scheme, the feedback control part sends corrective signals through the inverse dynamic model as can be seen in Eq.(4-18), where $\dot{\mathbf{A}}^*$ can be thought as a nominal velocity rather than the actual velocity. This feedback control action is distinguished from the feedforward dynamic compensation

scheme in which the feedback controller acts independently of the dynamic compensation. In contrast to the feedforward compensation scheme in which the dynamic compensation relies upon only the desired state information, the computation of full dynamic compensation scheme is done on the basis of the actual state information.

In principle, the full dynamic compensation scheme(**FDC**) should be more accurate than the feedforward compensation scheme(**FF**) because the action of feedback control is decoupled through the inverse dynamics and the actual dynamic states are reflected in the compensation.

The performances of the control scheme presented in this chapter can be compared to each other only if the same criteria are used for the design of controller gain matrices. Therefore the controller gains are selected by placing the closed loop poles for each joint at the same location(z=0.778) as in the case of independent joint control scheme.

The tip position error profiles for **FDC** and **FF** schemes are shown in Fig.4.23. It can be seen that the **FDC** scheme significantly reduces the tracking error compared to the **FF** scheme. **FDC** scheme exhibits the maximum tip position error of 0.022 [mm] for test motion I with splined cubic trajectory and 0.025 [mm] for test motion II. The maximum tip position error for the different test motions are summarized in TABLE 4.5.

The good estimation of the constituent parameters of the dynamic model is undoubtedly an important factor in the dynamic compensation control scheme. Determining these parameters from measurements or computer models is generally difficult and involves some degree of uncertainty in the estimated parameter values. This is the common objection to the dynamic compensation control scheme. But there has not been any concrete rational basis for the source of errors or the bounds on errors in estimation of parameters. Though the work by [An 86] suggested that the link mass can be accurately identified to within only a few percent of error, this results could be a guideline for the uncertainty in the estimation of the inertial parameters.

If some models for actuator dynamics, its electronic hardware system and friction of the mechanical system,etc. are incorporated into the dynamic model as a whole, there might exist much bigger errors in the estimation of parameter. Spong et al. made an

assumption of ± 50 % error in parameter estimation for verifying a robust control formulation[Spong 84]. Therefore, ± 50 % error bound of parameter estimation is assumed as the worst parameter estimation error for reasonable argument in the next simulation experiments.

The performances of **FDC** and **FF** scheme are compared using two different incorrect parameter sets, i.e., 50 % overestimated values and 50 % underestimated values. The maximum tip position error occurs in the vicinity of t=0.15[sec] over the 0.45 [sec] movement. The average tracking errors of **FDC** scheme for both incorrect parameter sets are less than one-half of those of **FF** scheme. The tracking performance at the tip position with incorrect parameter values are displayed in Fig.4.24 and 4.26. The error profiles for joint angular velocities are included in Fig.4.25 for the case of 50% underestimated parameter.

For the **FDC** scheme with incorrect parameter values, the simulation experiments show that significant tracking errors can arise (the maximum tip position error of 13.4 [mm] for 50% overestimated parameter values and 4.13 [mm] for 50% underestimated parameter values: motion II with splined cubic trajectory). Interestingly, the tracking error for the underestimated parameter case is nearly three times larger than the overestimated parameter case.

This tracking error comes from both the realizable replacement and the incorrectness of parameter estimation. If the parameters are exact, the nonlinear error driving disturbance Γ of Eq.(4-30) could be negligible, but the error in parameter estimation will increase the error driving disturbance as shown in Fig.4.29. This graph shows that the error driving disturbances are smaller when the parameter is overestimated than when the parameter is underestimated.

The tip position error profiles of **FDC** scheme with incorrect parameter values are presented in Fig.4.27 and 4.28. The maximum tip position errors by incorrect parameter values are listed in TABLE 4.6. The case of the perfect parameter values can serve the achievable lower bound of the tracking error of **FDC** scheme as shown in TABLE 4.5.

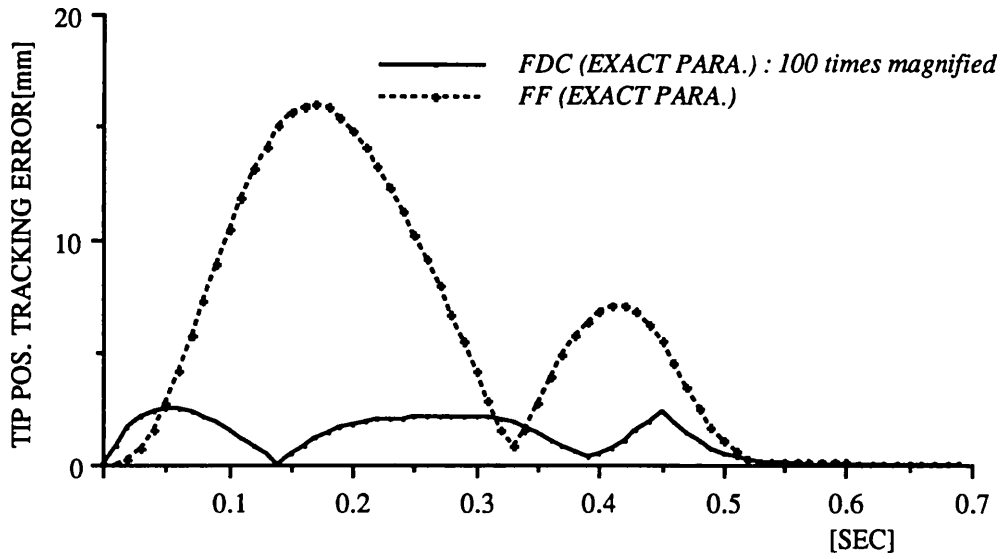

Figure 4.23 Comparison of Performance Between **FF** and **FDC** Scheme (Motion II, Splined Cubic Trajectory)

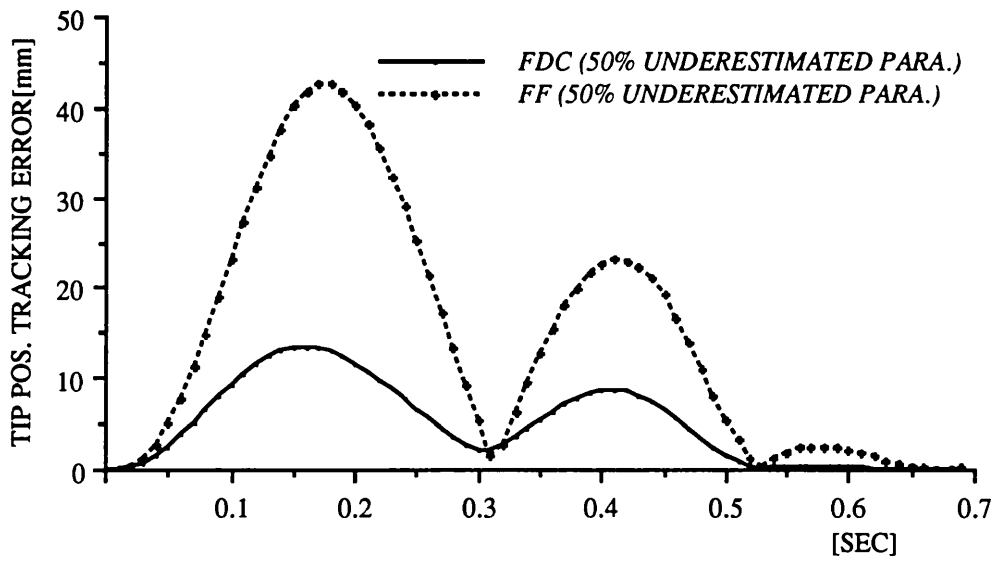

Figure 4.24 Comparison of Performance Between **FF** and **FDC** Scheme (Motion II, Splined Cubic Trajectory, 50% Underestimated Parameter Values)

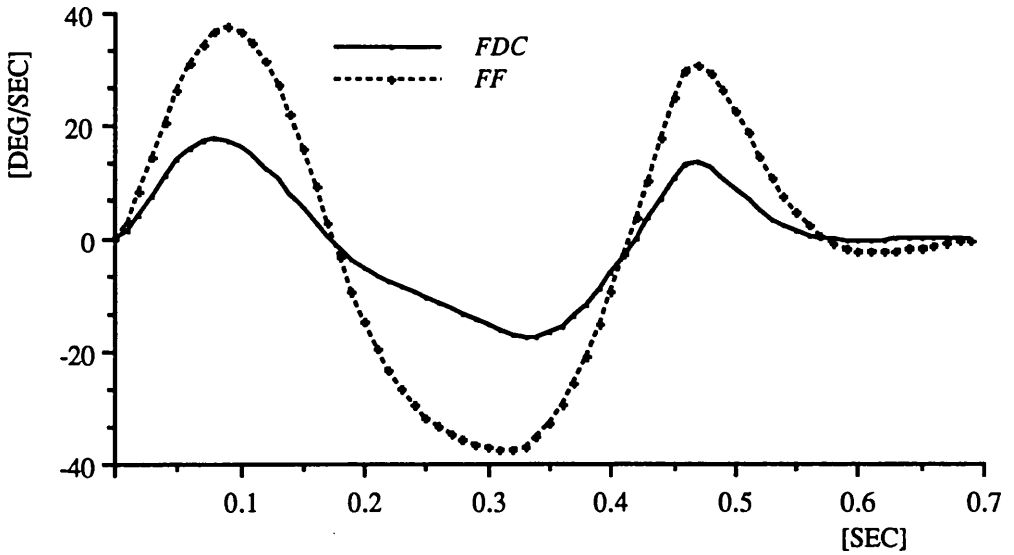

Figure 4.25a Velocity Error Profile For Joint 1
(Motion II, Splined Cubic Trajectory, 50% Underestimated Parameter Values)

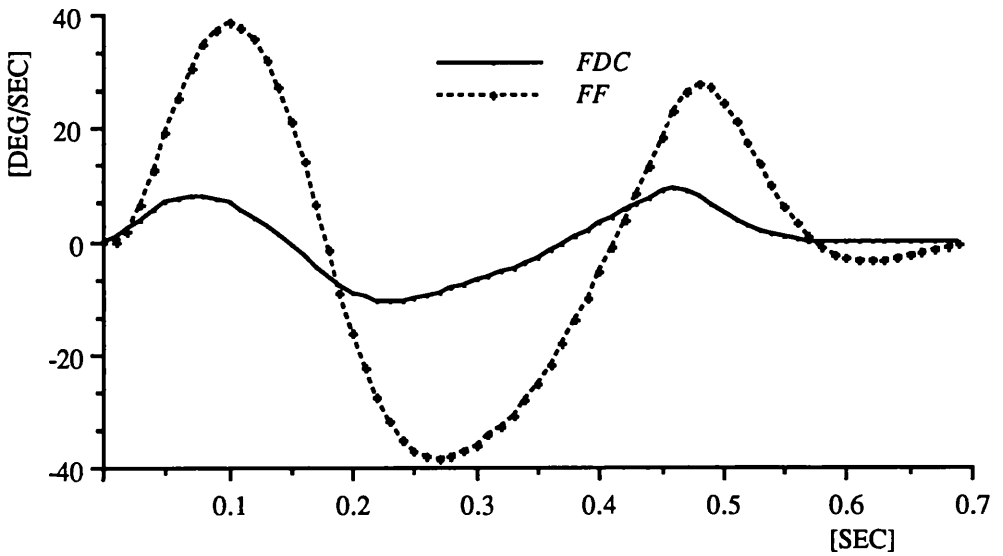

Figure 4.25b Velocity Error Profile For Joint 2
(Motion II, Splined Cubic Trajectory, 50% Underestimated Parameter Values)

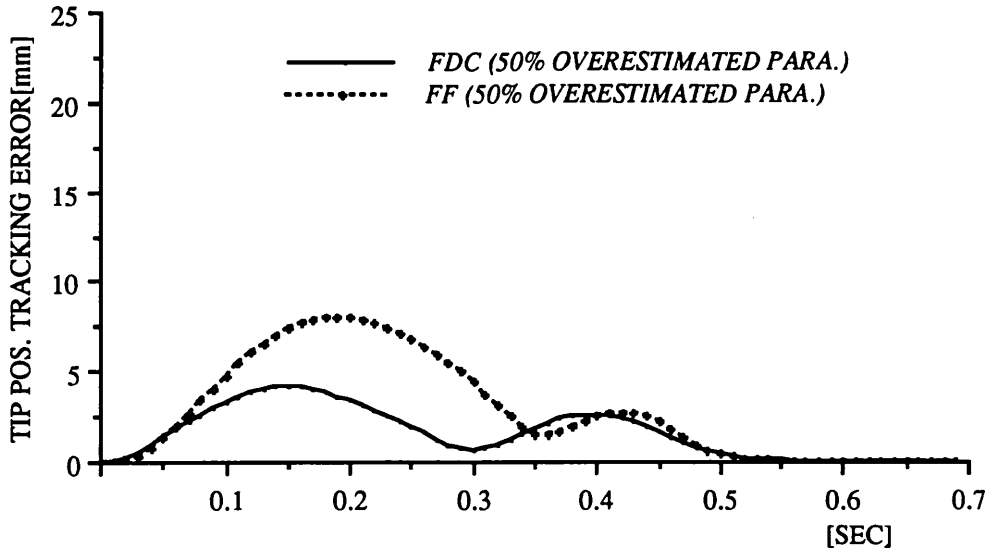

Figure 4.26 Comparison of Performance Between **FF** and **FDC** Scheme
(Motion II, Splined Cubic Trajectory, 50% Overestimated Parameter Values)

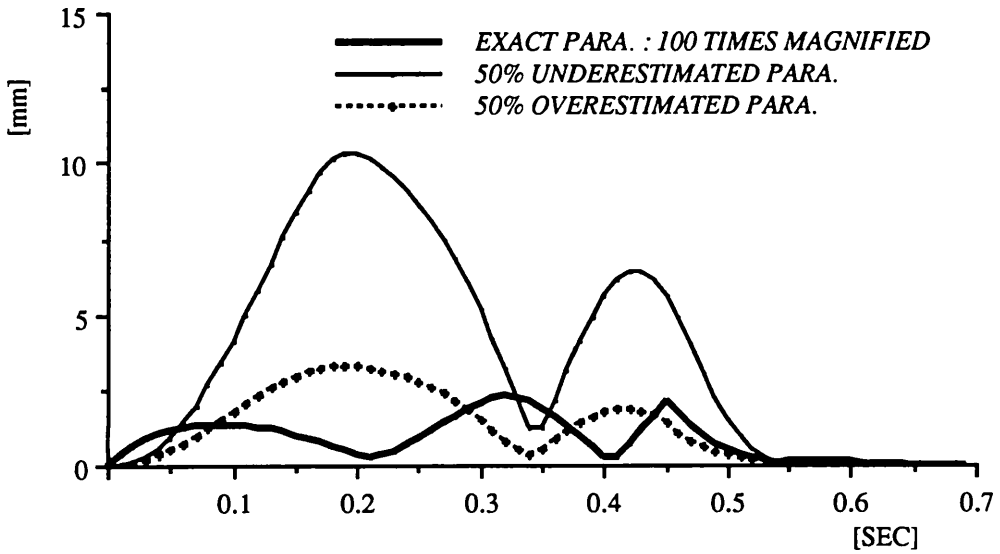

Figure 4.27 Tip Position Tracking Error of **FDC** Scheme
(Motion I, Splined Cubic Trajectory)

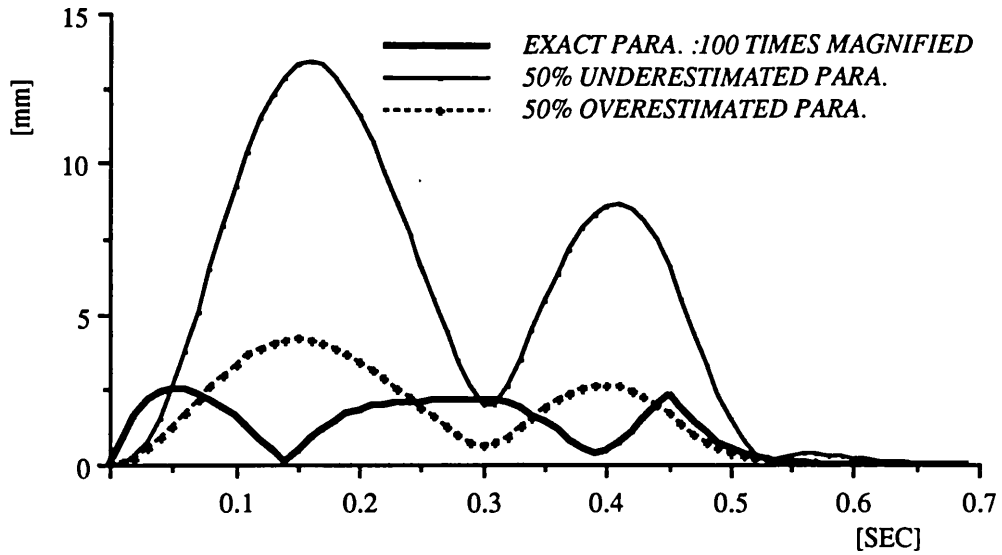

Figure 4.28 Tip Position Tracking Error of **FDC** Scheme
(Motion II, Splined Cubic Trajectory)

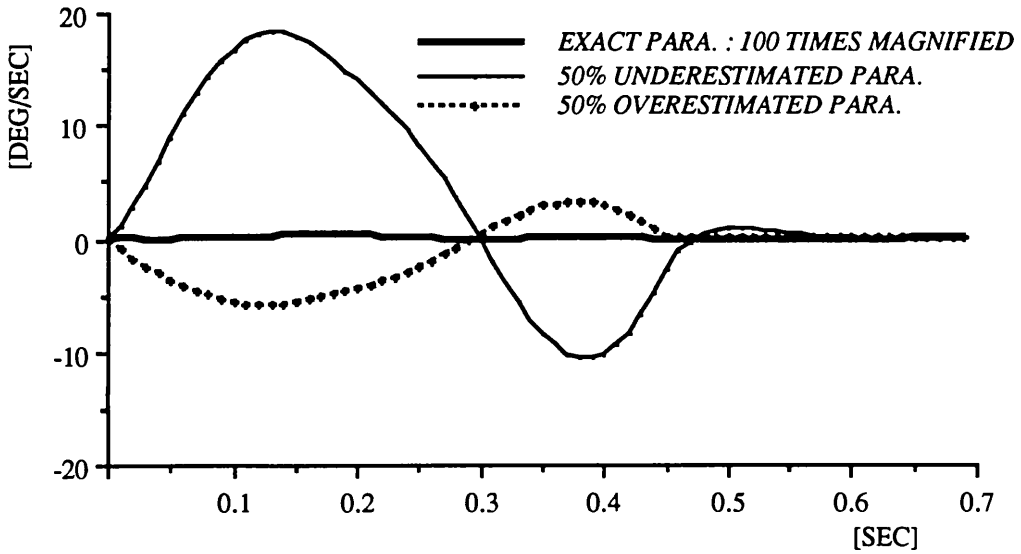

Figure 4.29a Error Driving Disturbances For Joint 1
(Motion II, Splined Cubic Trajectory)

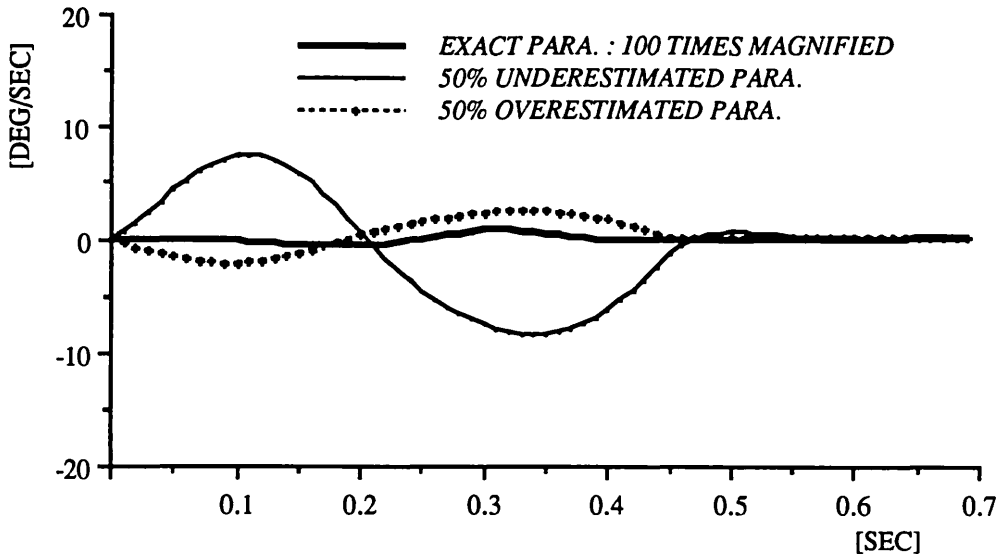

Figure 4.29b Error Driving Disturbances For Joint 2
(Motion II, Splined Cubic Trajectory)

TABLE 4.5 Maximum Tip Position Error

Motion	Trajectory	Max. Tip Position Error [mm]	
		F F	**F D C**
I	Cubic	7.46	0.017
	Splined Cubic	10.0	0.022
II	Cubic	11.7	0.013
	Splined Cubic	16.0	0.025

TABLE 4.6 Maximum Tip Position Error of **FDC** scheme with Incorrect Parameters

Motion	Trajectory	Max. Tip Position Error [mm]	
		50% Underestimated	50% Overestimated
I	Cubic	8.02	2.58
	Splined Cubic	10.3	3.28
II	Cubic	10.1	3.04
	Splined Cubic	13.4	4.13

4.7.2 Influence of Sampling Frequency

In the previous section, the performances of control scheme are evaluated at the selected sampling rate 200 Hz. The effect of changing the control sampling period will be investigated in this section.

The actual position gain matrix ($\frac{2}{\Delta t}\mathbf{K}_p$) is the function of the sampling period of the control system. The higher the sampling rate the larger the value of the actual position gain will be achieved. Since the servo stiffness of controlled system[Paul 81] is governed by the proportional gain matrix, a higher sampling rate implies higher stiffness also. The elements of gain matrices are selected so that the pole location remains fixed regardless of the change of sampling rate and satisfy the critical damping condition. This criteria for selecting the gains based on identical pole location will make the comparison meaningful.

In Fig.4.30, the tracking performances of **FDC** scheme are depicted according to the change of sampling rate. Fig.4.31a,b represent the error driving disturbance profile for each joint. Decreasing the sampling period reduces the influence of realizable replacements and parameter error, and thus improves tracking accuracy. From the above observations, it can be deduced that increasing the sampling rate results in a noteworthy improvement of tracking performance. Fig.4.32 shows the effect of sampling rate on tracking performance when the 50% overestimated parameter values are used for **FDC** scheme. Even under the circumstance of a parameter error, decreasing the sampling period significantly improves the tracking performance. The maximum tracking error at the tip position are given in TABLE 4.7. Particularly, in the case of underestimated parameter values, increasing the control sampling period from 5 [msec] to 10 [msec] results in a big degradation of the tracking performance. From this simulation experiment, it can be concluded that higher sampling rates are important for better tracking performance of control scheme because higher sampling rate results in stiffer system and they are effectively capable of reducing the error driving disturbances.

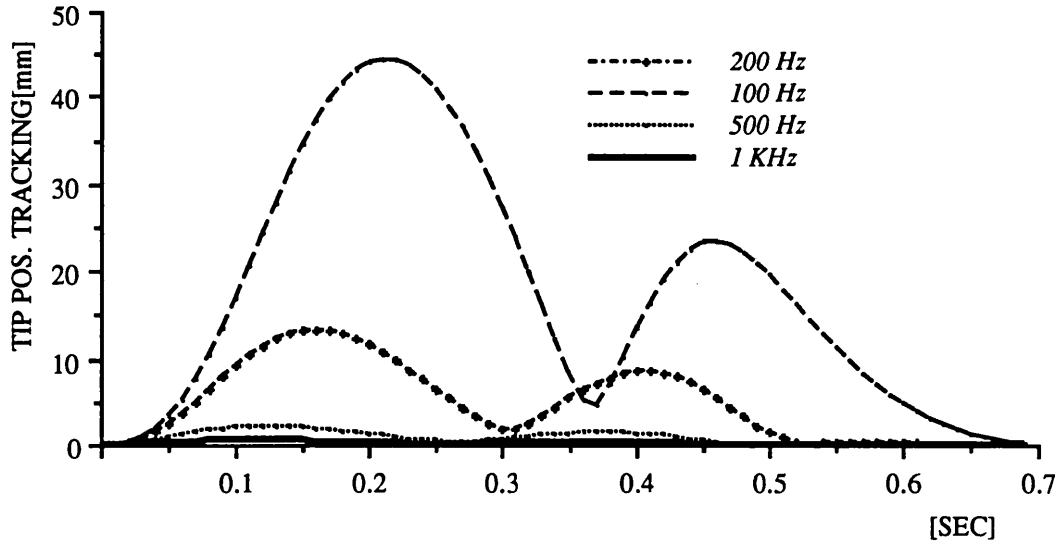

Figure 4.30 Influence of Sampling Frequency On **FDC** Scheme
(Test Motion II, Splined Cubic Trajectory, 50% Underestimated Parameter)

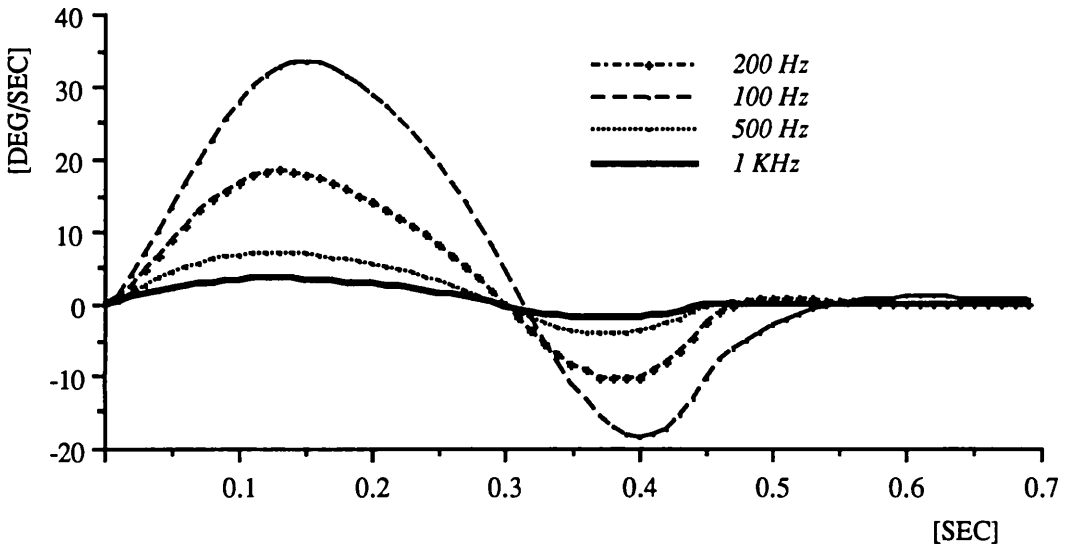

Figure 4.31a Error Driving Disturbances For Joint 1
(Test Motion II, Splined Cubic Trajectory, 50% Underestimated Parameter)

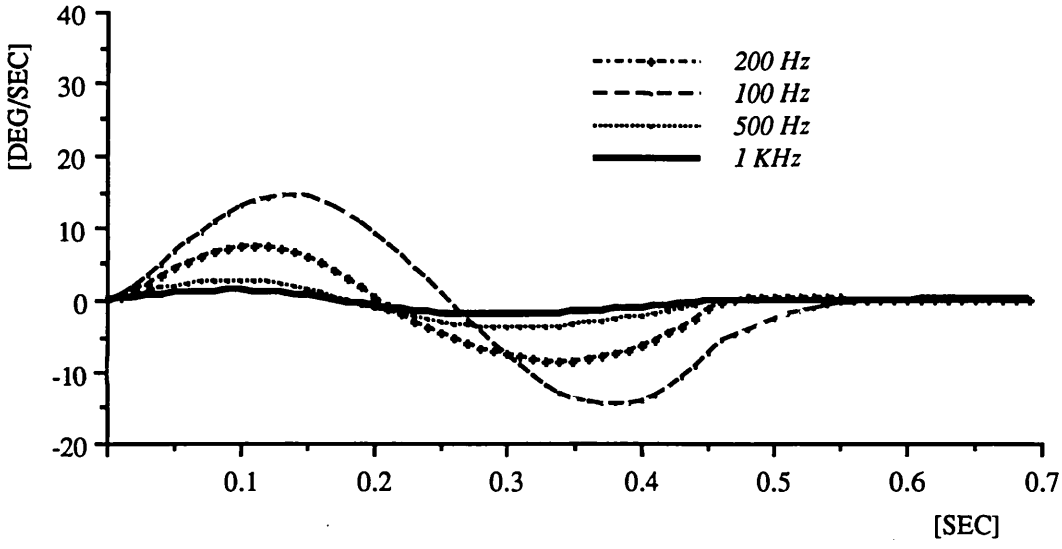

Figure 4.31b Error Driving Disturbances For Joint 2
(Test Motion II, Splined Cubic Trajectory, 50% Underestimated Parameter)

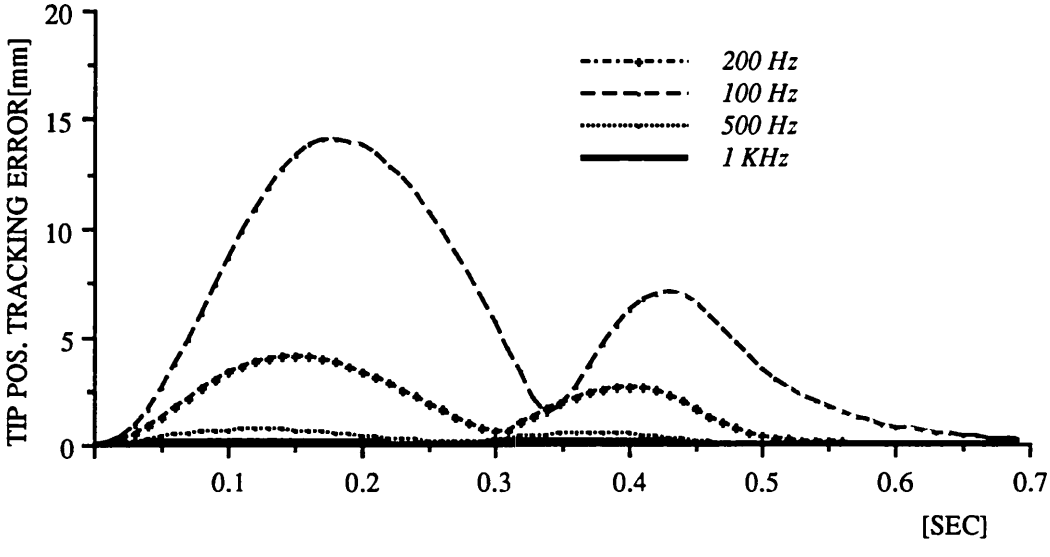

Figure 4.32 Influence of Sampling Frequency On **FDC** Scheme
(Test Motion II, Splined Cubic Trajectory, 50% Overestimated Parameter)

TABLE 4.7 Tip Position Error For Different Sampling Frequency
(Motion II, Splined Cubic Trajectory)

Frequency [Hz]	Max. Tip Position Error [mm]		
	Perfect Parameter	50% Underestimated	50% Overestimated
100	0.154	44.5	14.1
200	0.025	13.4	4.13
500	0.002	2.17	0.708
1000	0.0003	0.54	0.179

4.8 Reconstruction of Velocity Independent Discrete Dynamic Model

The real-time dynamic compensation scheme based on the dynamic model **VD** shows its distinctive performance. However this control scheme requires the real-time information of velocity in generating the control signals. Although arrangements for sensors are sometimes used, vast majority of the robots have only a positional sensor at each joint. In this circumstance, direct access to velocity information is not available. Hence, if the dynamic model could be expressed to be dependent only on positional information, it will be very attractive in the implementational point of view because additional sensory installation or any indirect estimating procedure for the higher order derivatives can be eliminated.

The determination of derivatives of a function f(x) in terms of discrete values of f(x) can only be made with a certain degree of approximation error. Therefore, to eliminate the dependence on velocity information from the dynamic model, a numerical approximation for velocity is inevitable. Applying this numerical approximation could result in undesirable performance of the inverse dynamic model . But the almost perfect performance of the inverse dynamic model **VD** allows room for further approximation which might not cause too much degradation of model accuracy.

Many numerical formulas for the first derivatives are available[Young 72]. Once a numerical formula is selected, the accuracy attainable for the first derivative f'(x) is limited by the accuracy of the value of f(x) rather than a smaller step size. In this section, four point numerical formula is chosen for the first derivative as below.

$$\dot{f}(k) = \frac{1}{6\Delta t}[11f(k) - 18f(k-1) + 9f(k-2) - 2f(k-3)] \tag{4-32}$$

By the help of the above approximation formula for the first derivative, the discrete dynamic model which simply relies upon positional information can be reconstructed as follows:

$$\begin{aligned}\mathbf{U}(k) = {} & \mathbf{G}_1[\mathbf{A}(k+1)]\mathbf{A}(k+1) + \mathbf{G}_2[\mathbf{A}(k+1),\mathbf{A}(k)]\mathbf{A}(k) \\ & + \mathbf{G}_3[\mathbf{A}(k+1),\mathbf{A}(k)]\mathbf{A}(k-1) + \mathbf{G}_4[\mathbf{A}(k+1),\mathbf{A}(k)]\mathbf{A}(k-2) \\ & + \mathbf{G}_5[\mathbf{A}(k)]\mathbf{A}(k-3) + \mathbf{C}_p[\mathbf{A}(k+1),\mathbf{A}(k),\mathbf{A}(k-1),\mathbf{A}(k-2),\mathbf{A}(k-3)] \\ & + \mathbf{D}_p[\mathbf{A}(k+1),\mathbf{A}(k),\mathbf{A}(k-1),\mathbf{A}(k-2),\mathbf{A}(k-3)]\end{aligned} \tag{4-33}$$

The detailed expression of each term is given in APPENDIX A.

4.8.1 Choice of Parameter Set and Inverse Dynamics

The reconstructed discrete dynamic model (**PD** : Eq.4-33) has a big advantage over the previous dynamic models because the need for explicit velocity information is completely eliminated. The structure of Eq.(4-33) has a form similar to a nonlinear ARX(Auto-Regressive Exogeneous) model[Ljung 87]. In the general case of nonlinear ARX model, the number of unknown parameters corresponds to the number of independent nonlinear terms introduced into a model structure. But unknown parameters can be regrouped and reparametrized through some prior knowledge for algebraic or physical relationship between the parameters. Hence, in the case of Eq.(4-33), it could be interpreted that the minimum parameter set (five parameters) is achieved since the model structure of Eq.(4-33) originates from the well-established rigid body mechanics. However it is essential to have some insight into the selection of parameter values. Since several stages of numerical approximation are involved to reach the final form of Eq.(4-33), there might exist some possibility that the parameter values which are directly estimated by Eq.(4-33) could result in better performance than the true parameter values defined in Eq.(2-62). The parameter values estimated directly from the model **PD** are listed in TABLE 4.8.

TABLE 4.8 Parameter Estimation Using dynamic model **PD**

(Excitation type I, Input level 30%)

	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
True Value	213	102	99	877	850
Estimate By **PD**	203.24	91.70	93.70	933.10	876.71

Two different sets of values in the above table are used in the computation of inverse dynamic model **PD**. Fig.4.33 gives clear answer about the choice of parameter values. This graph compares the tip position errors according to the different set of parameter values. Tracking errors by the true parameter values(solid line, magnified 100 times for clarity) are far less than tracking errors by the parameter values directly estimated from the dynamic model **PD**. From this result, it can be concluded that the parameter set in the dynamic model **PD** is still sustaining the identical significance as the parameter set in the original continuous-time dynamic model(Eq. 2-79) and the velocity independent dynamic model(**PD**) itself is not adequate for the parameter identification purpose.

For the dynamic model **VD**, the actuating inputs are computed by substituting the desired joint positions and velocities into the dynamic model, but the dynamic model **PD** utilizes only position information to generate the actuating signals. The joint error profiles by the two different inverse dynamic models(**VD**, **PD**) are depicted in Fig.4.34a (when motion II with cubic trajectory is demanded). These graphs show that the joint errors by the inverse dynamic model **PD** are comparable to those by the **VD** model. In Fig.4.35(a, b), the evolution of the actuating input for joints are compared. The actuating input profiles by the inverse dynamic model **PD** are almost identical to those by the **VD** model except the small ripples in the vicinity of switching points(start and end) of the demanded trajectory. These ripples are caused by the approximation error of the finite formula for the first derivatives. The sudden changes in the joint error profiles in Fig.4.34(a,b) are induced by the ripples of actuating inputs. If the cubic trajectory is replaced by the splined cubic trajectory with the same motion type, the smooth evolution of tip position error profile is observed (Fig. 4.36). In addition, Fig.4.36 shows the quite similar performances of the two dynamic model (**VD**, **PD**). In conclusion, the simulation experiments in this section reveal that the accuracy of the dynamic model **PD** is almost equivalent to that of dynamic model **VD**.

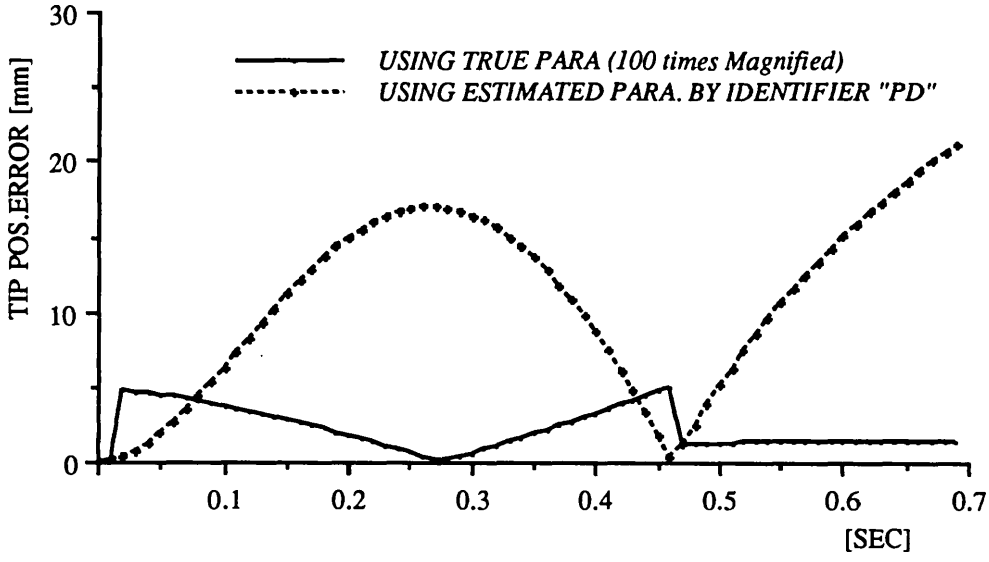

Figure 4.33 comparison of Inverse Dynamic Performance
With Two Different Sets of Parameter Value(Motion I, Cubic Trajectory)

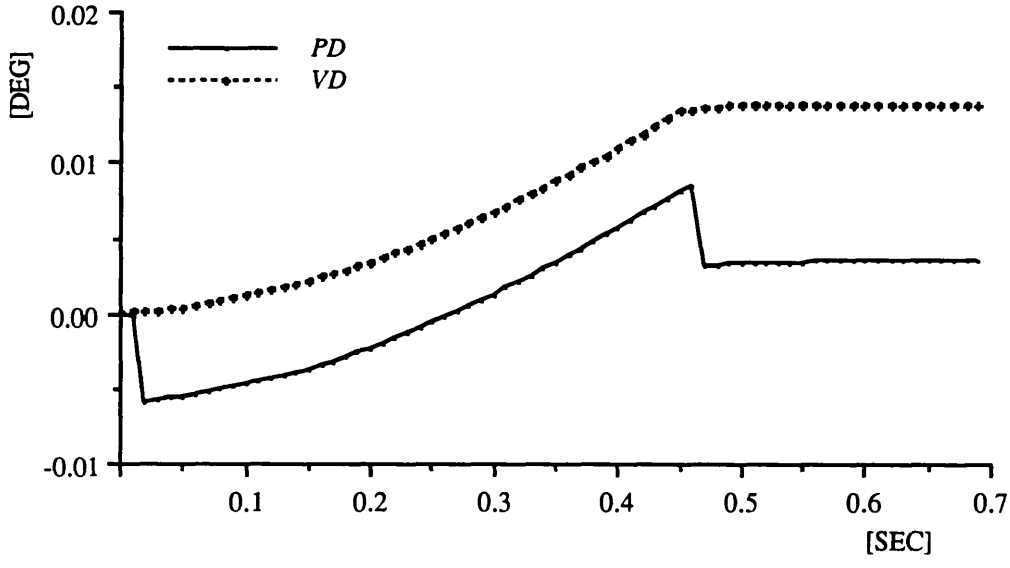

Figure 4.34a Position Error For Joint 1 (Motion II, Cubic Trajectory)

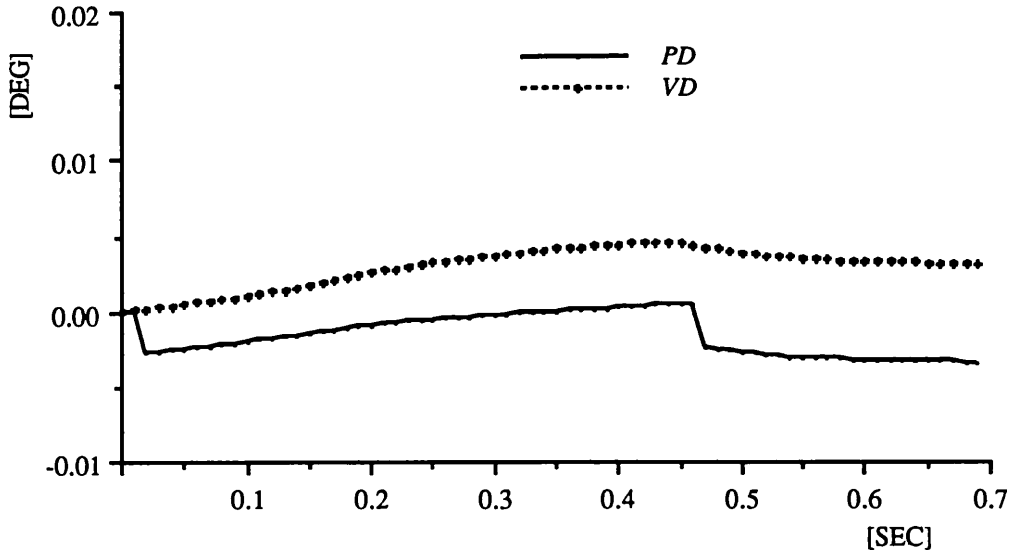

Figure 4.34b Position Error For Joint 2 (Motion II, Cubic Trajectory)

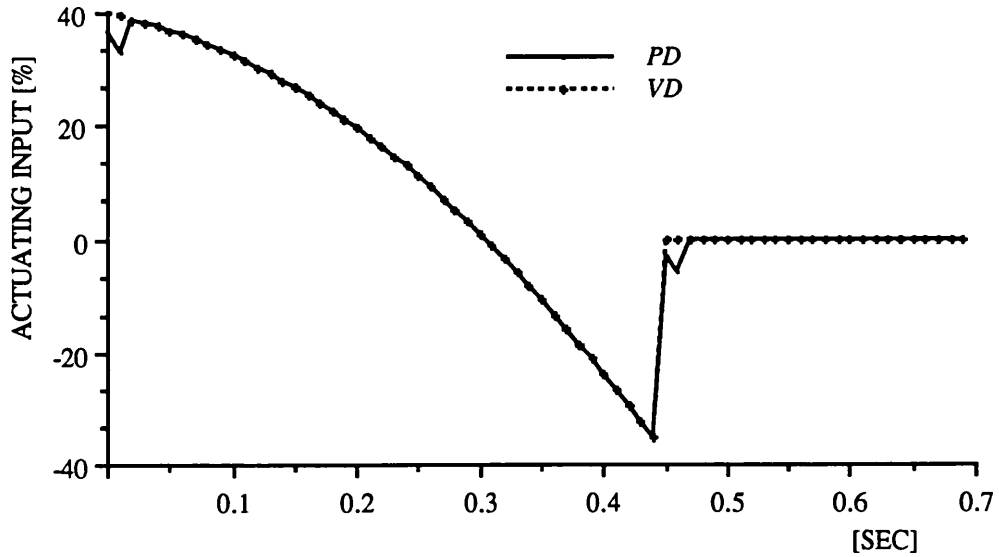

Figure 4.35a Input Profile For Joint 1 (Motion II, Cubic Trajectory)

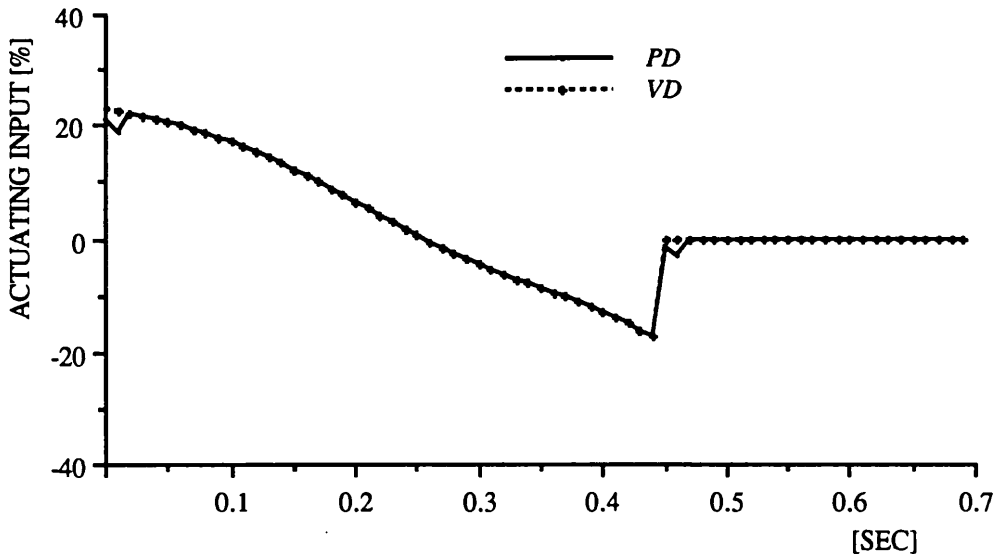

Figure 4.35b Input Profile For Joint 2 (Motion II, Cubic Trajectory)

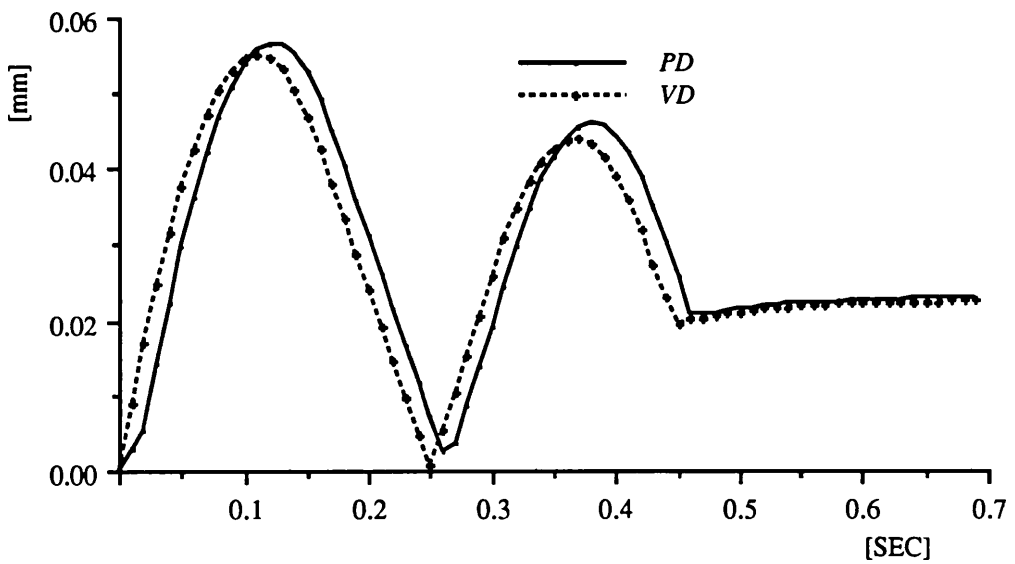

Figure 4.36 Tip Position Tracking Error (Motion I, Splined Cubic Trajectory)

4.8.2 Dynamic Compensation Based On Velocity Independent Discrete Model

The reconstructed dynamic model **PD** eliminates the need for tachometers and accelerometers in its implementation stage. The method of computing the actuating inputs for joints is a little bit different from the case of the dynamic model (**VD**). i.e., the commanded velocity input vector $\dot{\mathbf{A}}_d(k+1)$ in Eq.(4-22) is replaced by a commanded position input vector $\mathbf{A}_d(k+1)$, and the servo part of control law is only dependent on the past positional errors. Thus the nominal position vector $\mathbf{A}^*(k)$ is changed as

$$\mathbf{A}^*(k) = \mathbf{A}_d(k+1) + \mathbf{K}_1\mathbf{E}(k) + \mathbf{K}_2\mathbf{E}(k-1) + \mathbf{K}_3\mathbf{E}(k-2) \tag{4-34}$$

here $\mathbf{K}_i$ is the feedback gain matrices. Using Eq.(4-33), the actuating inputs for joints can be computed

$$\begin{aligned}\mathbf{U}(k) = {} & \hat{\mathbf{G}}_1\mathbf{A}^*(k) + \hat{\mathbf{G}}_2\mathbf{A}(k) + \hat{\mathbf{G}}_3\mathbf{A}(k-1) + \hat{\mathbf{G}}_4\mathbf{A}(k-2) \\ & + \hat{\mathbf{G}}_5\mathbf{A}(k-3) + \hat{\mathbf{C}}_p + \hat{\mathbf{D}}_p\end{aligned} \tag{4-35}$$

For realizable computation, the inaccessible (k+1)-th joint states at k-th sampling instant are replaced by the desired states. By the similar procedure given in section 4.7, the closed-loop system equation can be written as below:

$$\mathbf{A}(k+1) = \mathbf{A}^*(k) - \Gamma_p(k+1) \tag{4-36a}$$

where $\Gamma_p(k+1)$ is the error driving disturbance vector and is expressed as belows,

$$\begin{aligned}\Gamma_p = \hat{\mathbf{G}}_1^{-1}\Big\langle & \left\{\mathbf{G}_1 - \hat{\mathbf{G}}_1\right\}\mathbf{A}(k+1) + \left\{\mathbf{G}_2 - \hat{\mathbf{G}}_2\right\}\mathbf{A}(k) + \left\{\mathbf{G}_3 - \hat{\mathbf{G}}_3\right\}\mathbf{A}(k-1) \\ & + \left\{\mathbf{G}_4 - \hat{\mathbf{G}}_4\right\}\mathbf{A}(k-2) + \left\{\mathbf{G}_5 - \hat{\mathbf{G}}_5\right\}\mathbf{A}(k-3) \\ & + \left\{\mathbf{D}_p - \hat{\mathbf{D}}_p\right\} + \left\{\mathbf{C}_p - \hat{\mathbf{C}}_p\right\}\Big\rangle\end{aligned} \tag{4-36b}$$

Substituting Eq.(4-34) into (4-36a), the difference equation for tracking error vector **E**(k) can be obtained.

$$\mathbf{E}(k+1) + \mathbf{K}_1\mathbf{E}(k) + \mathbf{K}_2\mathbf{E}(k-1) + \mathbf{K}_3\mathbf{E}(k-2) = \Gamma_p(k+1) \qquad (4\text{-}37)$$

The feedback gain matrices $\mathbf{K}_i$ can be selected in the same way discussed in section 4.6. $\mathbf{K}_i$ is selected such that the pole of the error equation (Eq. 4-37) is placed at the same location(z=0.778) for the purpose of comparison with the previous results.

The position tracking performance of **FDC** scheme using the dynamic model **PD** is shown in Fig.4.37 and 4.38. If the cubic trajectory is demanded, the tip position errors by **PD** model exhibit sharp increases near the switching points(start and end points) of the desired trajectory(Fig.4.37). On the other hand, if the cubic trajectory is replaced by the splined cubic trajectory, these peaks are not observed and the tracking errors by the **PD** model keep the profiles lower than those by the **VD** model as shown in Fig.4.38. The maximum tracking errors at the tip position under the various test trajectories are listed in TABLE 4.9. Under the cubic trajectory, the maximum tracking errors by the **PD** model are approximately 2 times bigger than those by the **VD** model. But for the splined cubic trajectory, the maximum tracking errors by the **PD** model are decreased by half of those by the **VD** model. Hence, in utilizing **PD** model, the smooth evolution of the desired position profile is important to reduce the tracking error.

However, the results shown above are based on the perfect parameter estimates in the model. If incorrectness in parameter estimation is involved, the smoothness in the desired trajectory is not the key factor to determine the tracking error because the influence of parameter errors becomes more dominant. Fig.4.39 and 4.41 illustrate the tip position errors under the different motion types. These graphs show that the tracking errors by the **PD** model are bigger than those by **VD** model, and it can be seen that smooth trajectory is not helpful to reduce the tracking error as in Fig.4.38.

TABLE 4.10 shows the maximum tip position errors of FDC scheme when the parameters are incorrect. Comparing this table(TABLE 4.10) with the results from TABLE 4.6 (maximum tracking errors by the **VD** model), it can be seen that the maximum tracking errors by the **PD** model are larger (from 20 % upto 50 %) than the errors by **VD** model.

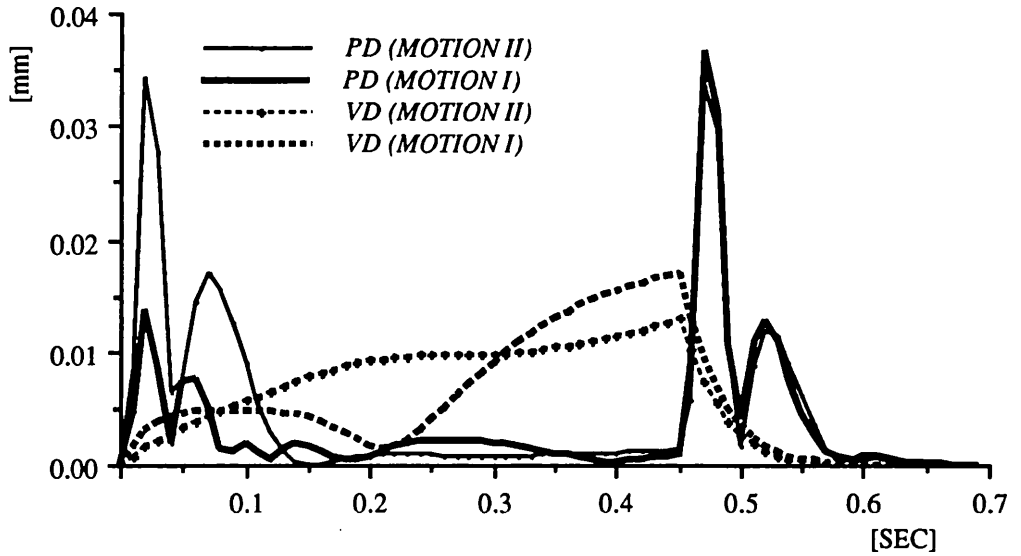

Figure 4.37 Performance of Two Different Discrete Dynamic Models on **FDC** Scheme (Cubic Trajectory)

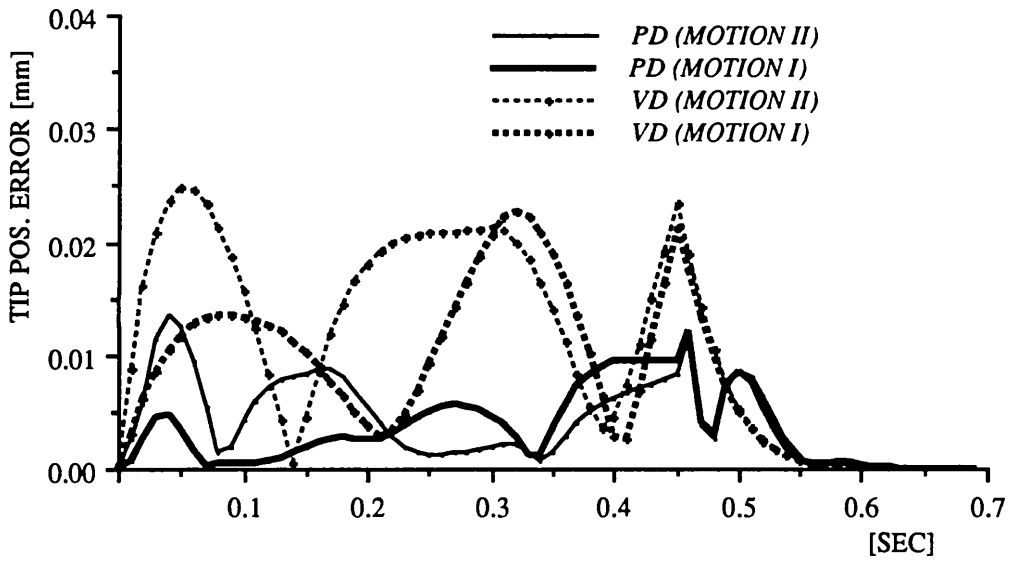

Figure 4.38 Performance of Two Different Discrete Dynamic Models on **FDC** Scheme (Splined Cubic Trajectory)

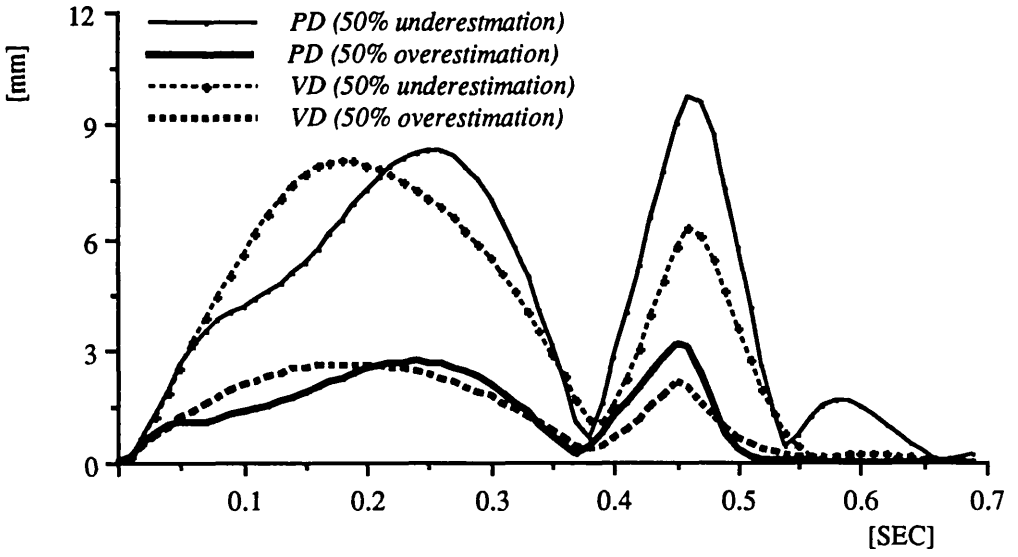

Figure 4.39 comparison of Tip Position Tracking Error
(Motion I, Cubic Trajectory)

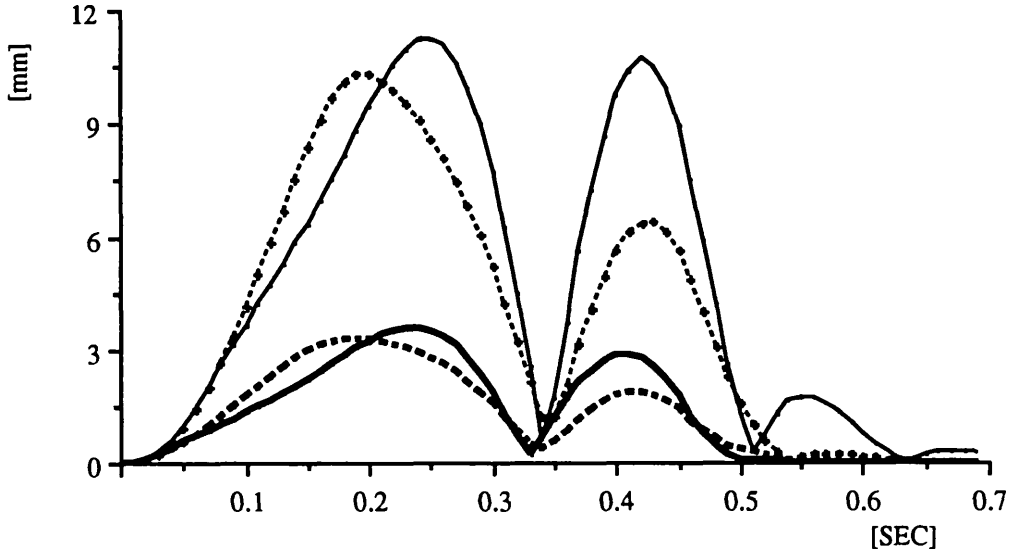

Figure 4.40 comparison of Tip Position Tracking Error
(Motion I, Splined Cubic Motion)

TABLE 4.9 Maximum Tip Position Errors of **FDC** scheme

Motion	Trajectory	Max. Tip Position Error [mm]	
		V D	**P D**
I	Cubic	0.017	0.037
	Splined Cubic	0.022	0.012
II	Cubic	0.013	0.034
	Splined Cubic	0.025	0.014

TABLE 4.10 Maximum Tip Position Error Using Dynamic model **PD**
With Incorrect Parameter

Motion	Trajectory	Max. Tip Position Error [mm]	
		50% Underestimated	50% Overestimated
I	Cubic	9.75	3.14
	Splined Cubic	11.2	3.58
II	Cubic	16.9	4.89
	Splined Cubic	22.1	6.50

5. CONTROLLER IMPLEMENTATION AND PERFORMANCE EVALUATION

Although many simulation results on the control of robot have been published, there have been few real-time implementation and performance evaluation of the control scheme based on dynamic model. The major reasons for this are the lack of a suitable robot system and the difficulty to estimate the constituent parameters for implementing a model based control scheme. The IC DDR project at Imperial College has overcome these difficulties and constructed the prototype direct drive robot including its customized controller system based on multi-microprocessor architecture (Motorola MC68020 and Texas Instruments TMS320C25 digital signal processors). A photograph of IC DDR is given in Fig.5.1a and its controller is shown in Fig.5.1b.

The performance of the model based control scheme depends greatly on the accuracy of the dynamic model. The model based control scheme utilizes the robot dynamic equations to calculate the actuating inputs necessary to drive the robot along the demanded trajectory. Generally, this model is highly nonlinear and is a function of the constituent parameters. The parameters of the dynamic model comprise the link inertial parameters, actuator characteristics and other relevant system parameters. In reference [An 86], the link inertial parameters were estimated. Their experiments needed a full force/torque sensing for estimation of the inertial parameters. But the actuator characteristic and mechanical friction, which also play an important role in the whole dynamic system, were not dealt with. The approach in [Khosla 87] was to estimate the parameters from the detailed drawings of the robot by approximating the mechanical structure as a combination of simplified geometric solid models. However this approach can not provide accurate parameters due to the approximation error in the simplified geometric solid model. Even when the inertial parameters are determined by a CAD database, there exists a sizable discrepancy between the experimental estimates and the CAD-modelled parameter values[An 86].

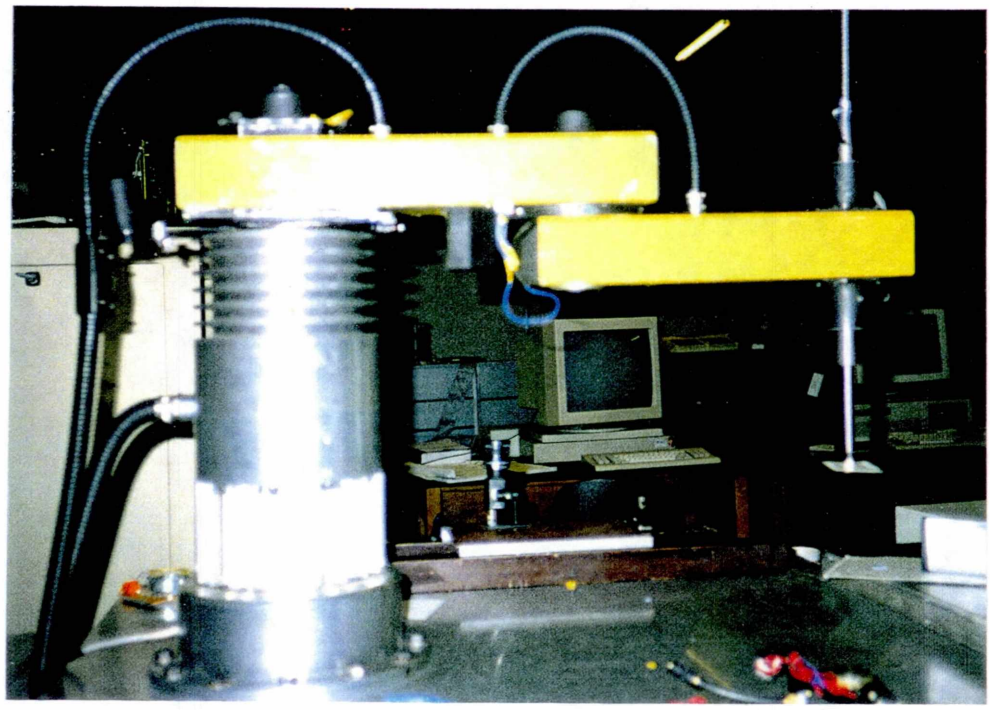

Figure 5.1a Photograph of Imperial College Direct Drive Robot

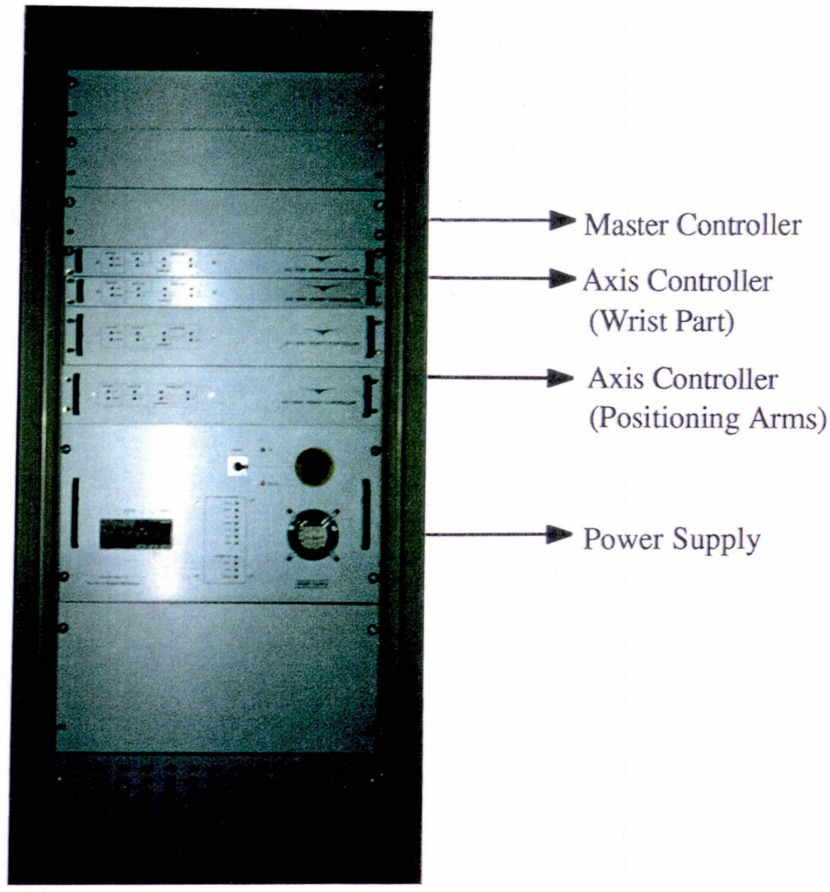

Figure 5.1b Photograph of the Controller System For IC DDR

Recently, a few papers in the area of trajectory tracking performances through the actual implementation were presented[Youcef-Toumi 87 / An 87 / Khosla 89]. In the approach adopted in reference [Youcef-Toumi 87], the arm dynamics were made decoupled and inertially invariant through appropriate mechanical design and mass redistribution technique. For these special arms, the controller design was made simple and controlled independently because the coupling torques caused by Coriolis and centrifugal forces were completely eliminated. Therefore the effect of interaction between the links on control performance did not appear. This arm structure requires a very careful mechanical design. The results of the experimental implementation were reported in [An 87 / Khosla 89]. In both papers, the continuous-time dynamic models were used rather than discrete-time dynamic model.

In this research, the experimental results for the parameter identification algorithms and real-time dynamic compensation scheme based on the velocity independent discrete dynamic model (**PD**) will be demonstrated on the IC DDR. The control performance of the real-time dynamic compensation scheme will be compared to the best tuned performance of the independent joint control scheme.

5.1 Structure of IC DDR Controller

The overall hardware structure of IC DDR controller consists of two hierarchical subsystems: the master controller and the axis controller (local controller). Incorporation of a higher level of coordination (by the master controller) reduces the complexity of the controller design. The coordination level (master controller) can support the application-oriented software, for example, kinematics software for contouring operations, software for automated assembly, obstacle avoidance, robot control languages, etc. Furthermore, the coordination level controller supervises the lower level controllers(axis controllers) and provides information for better control performance which can not be handled by the lower level controller. Each local controller(axis controller) is responsible for the elementary operations to realize the movement prescribed by the upper level controller, i.e., generating the appropriate commutation signals for the actuator, gathering the output states of joint, etc. The detailed tasks executed by the axis controller will be described in the next section.

5.1.1 Outline of Hardware Configuration

The hardware configuration of the master controller is shown in Fig.5.2. The master control board is divided into two sections : the communication and control computer section(**CCC**), and the axis control computer section(**ACC**). The **CCC** section consists of a Motorola MC68020 32-bit microprocessor running at 12.5 MHz and a MC68881 floating point co-processor. The static RAM sites take 32K chips giving 128K of RAM, and EPROM sites are giving 768K for system program storage. The **CCC** communicates with the **ACC** via 512K of dynamic RAM. There are also asynchronous communication channels using MC68681 DUARTs. One channel is for debugging and other channels are intended for the connection of peripheral devices. The **ACC** also consists of the second Motorola MC68020 microprocessor with provision for a floating point co-processor. It executes the control software, and there are digital interface on the memory expansion port which is used for connection between the master control board and the axis control board. The other I/O bus interface is a serial data bus running at 1 Mbits/sec for the additional future subsystems which can be linked to the robot system.

The axis controller board can be divided into three function blocks. Firstly the position measuring part of the motor shaft, and secondly the circuits for generating the mark space ratio for the output stages of the three phase motor windings, and the last block is the processor system.

The processor adopted is TMS320C25 digital signal processor which is operating at 32 MHz. It can access 8K words of program memory and 8K words of high speed 16-bit data RAM. The processor controls several peripherals necessary for the operation of motor. It reads the angular positions of the motor shaft via the on-board 16-bit resolver decoder and sends an 8-bit value to control the mark space ratios of three pulse width modulators. In this configuration, the output transistors (high power MOSFETs) connected to motor windings act merely as switches. By varying the mark space ratio at the high switching rate of 24 KHz, the required input voltages for the three phase windings of each motor can be properly adjusted.

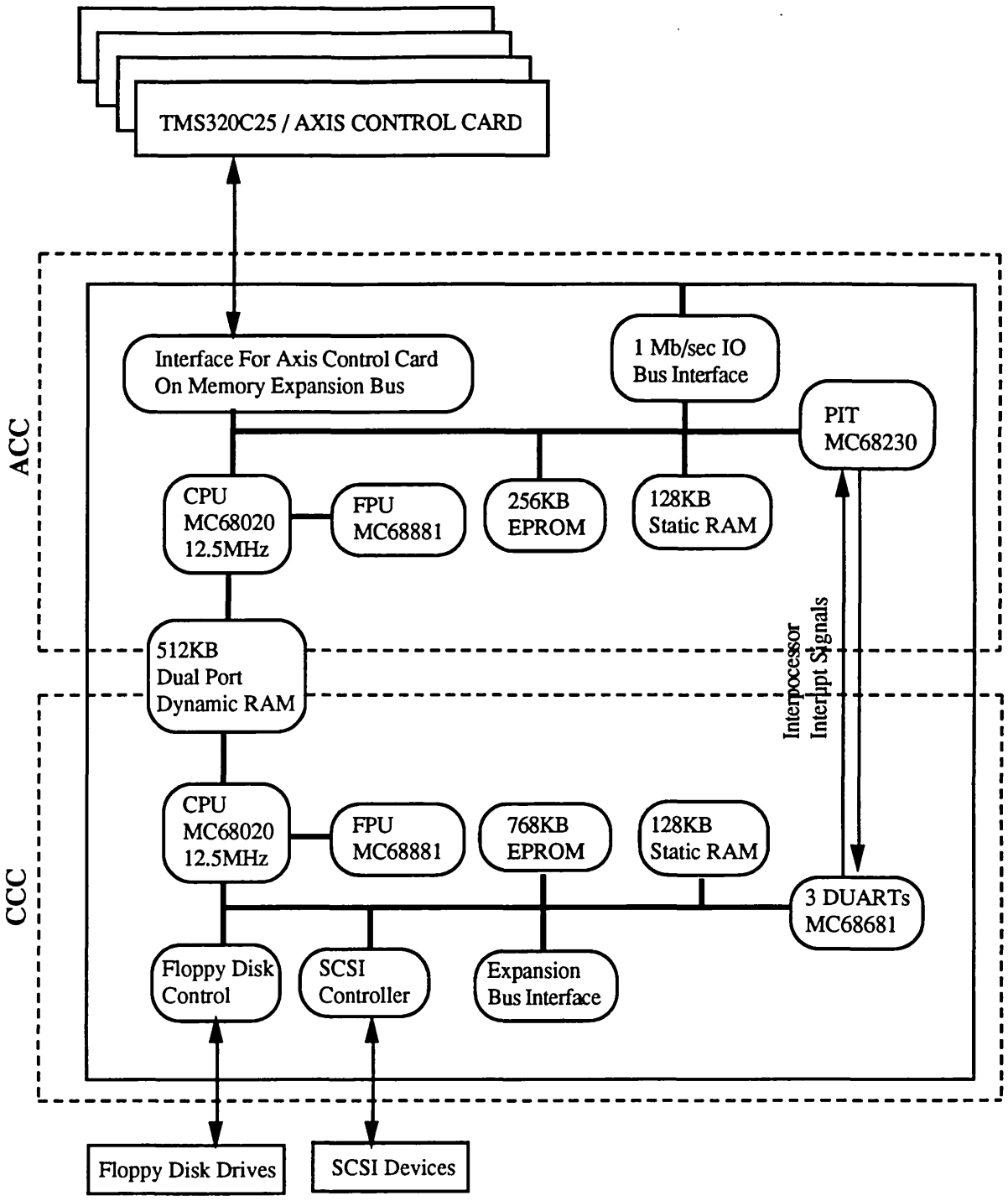

Figure 5.2 Block Diagram of Master Controller Board
(RDPT 1990)[RDPT 89]

Communication with the master controller is performed by a standard Motorola Expansion Bus. A 50-way ribbon cable is daisy chained together between the four axis control boards, coming from a Motorola Expansion Bus interface card on the master controller board. The address range that a particular axis control board occupies in the memory space of the master control board is selected via a 8-way dip switch so that it appears sequentially in the address space of the master controller.

The position measuring device used is a resolver which consists of active rotary coils energized from a sine wave oscillator and two static pickup coils 90° out of phase. The on-board resolver-to-digital converter(Analog Devices AD2580) converts the reference signal together with the two pickup coil signals into a 16-bit position value.

5.1.2 Software For Axis Controller

The IC DDR employs the high torque BDC(Brushless DC) motors. Unlike the conventional DC motor, the rotor of BDC motor consists of permanent magnets, while the stator consists of windings. Thus the rotor and the stator are interchanged. In the BDC motor, the mechanical commutation is replaced by electric switching circuits. Hence the key in using BDC motor and its amplifier system is commutation.

Commutation means knowing when and how much voltage to apply to which motor phase to provide rotation and torque in its desired direction. Brush type motor does this by the mechanical arrangement of brushes and commutation bars. There are two prevalent means of commutation in use today : 6-step commutation and sinusoidal commutation[Inland 87]. Sinusoidal commutation method develops smoother output torque than 6-step method, especially at low rotational speeds. But the sinusoidal commutation method tends to be more complicated than the 6-step commutation method.

The torque produced by a motor is directly related to the current applied to the windings. A common construction of windings is that of a three phase motor. To reduce torque ripple, the current flowing into each winding is varied continuously in accordance with rotor position. In order to modulate the motor current, a rotor position sensing device is needed to accurately generate sinusoids as a function of position.

The implementation block diagram of BDC motor drive system for IC DDR is depicted in Fig.5.3. In the model based control scheme, the master controller generates the elementary demand in the form of an internally expressed voltage signal which will be decoded and executed by the axis controller to realize the desired functional movement of the motor. The BDC motors used for the positioning arms of IC DDR are capable of handling a current of 40 [Amps] at the supplied voltage 100 [Volts]. Modern high power MOSFETs allow this supplied voltage to be switched very rapidly in the order of 100 [nsec], and hence the mean current into each winding can be controlled smoothly and accurately by means of pulse width modulation(PWM) technique where the output transistors act merely as on-off switches. The voltage applied to each winding of the motor is modulated to produce the sinusoidal wave shape as below.

$$V_a = V_d \, SIN(\theta_f) \tag{5-1a}$$
$$V_b = V_d \, SIN(\theta_f + \delta) \tag{5-1b}$$
$$V_c = V_d \, SIN(\theta_f + 2\delta) \tag{5-1c}$$

Where field angle θ_f and phase shift angle δ represent the electrical angles, and δ is selected such that the three sinusoidal phase voltages are electrically spaced 120° apart. The field angle is defined as the angle between the magnetic field created by the motor windings and the rotor. To obtain the field angle, a resolver and RDC(resolver-to-digital converter) are used to provide the rotor position. A side benefit of the resolver system is that the rotor position data directly corresponds to the angular position data for each joint of the positioning arm. But this physical position information of the rotor shaft can not be used directly in the modulation of phase voltages since the field angle is not equal to the mechanical angular displacement of the rotor shaft.

The field angle is measured from the nearest field angle boundary which is always 90° field angle behind the point of field balance(defined as the equilibrium shaft position for the particular three phase voltage combination according to Eq.(5-1). If the rotor has n magnetic poles, there exist n points of field balance. Therefore the field angle repeats n cycles for each complete rotation of the rotor shaft. The algorithm to get the field angle from positional data of rotor is outlined below:

- Subtract *RotOffset* from the angular position of rotor
- Repeat subtraction (360°/n) until negative

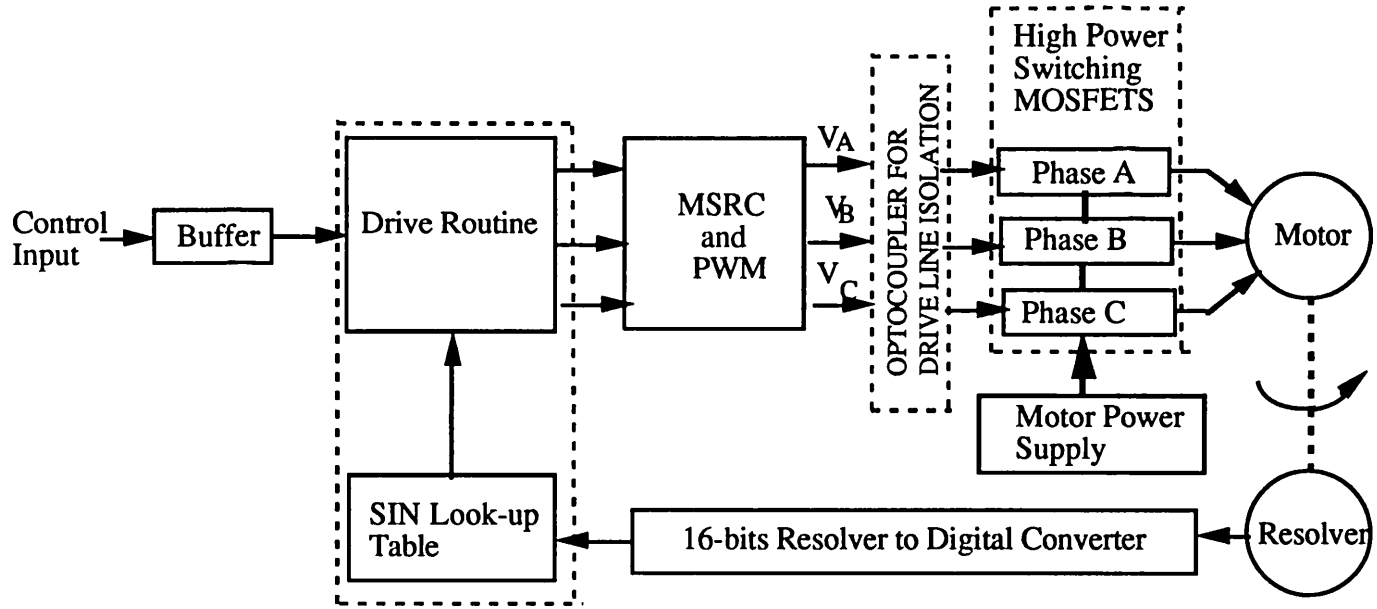

Figure 5.3 Brushless DC Motor Drive System

- Add position (360°/n)
- Multiply n

This results in the field angle. The *RotOffset* angle in the above algorithm is to make the alignment between the resolver reading angle and rotor shaft position. In other words, the field angle boundary is matched to the starting point of RDC reading. The detailed procedure to obtain *RotOffset* angle is included in APPENDIX B. Then this field angle is used in conjunction with a sine look-up table to provide the voltage modulation waveforms. The details of sine table and its usage are also given in APPENDIX B.

The software for driving the BDC motor is implemented in a synchronized way where the critical task is always active while other peripheral tasks are well attended. The reasons for this style instead of using multiple asynchronous process with various interrupt priorities can be summarised as below:

- Crucial tasks such as watch dog triggering are under tight supervision
- All tasks are performed with prescribed regularity
- Time dependent tasks(e.g., resolver reading) can be easily done
- Ease of coding and debugging

In the present implementation, there are only two interrupts to be handled. One is the system reset which happens on power-up or when the system is restarted. The other is the hardware interrupt and comes from the mark space ratio counter(MSRC) after the system has completed its start-up process. This interrupt happens at a regular rate of 24 [KHz], i.e., the rate of modulating the actuating voltage signals for the motor winding. The MSRC interrupt is chosen as the effective system clock because all other activities are synchronized with it.

The individual tasks are designed such that they rely only on a well defined set of global variables and no calls to other tasks occur. The individual tasks used in motor driving system are defined in TABLE 5.1. A group of selected tasks are allocated into the time slices over every sixteen MSRC interrupt period. The length of each time slice is 41.6 [μsec]. The repeated pattern of task sequence is presented in Fig.5.4. Particularly, the tasks *'Sequencer'* and *'Safety'* always precede the tasks of a specific group requested at a selected time slice.

TABLE 5.1 Definition of Tasks

Task Name	Purpose	Action
Sequencer	Update the Task Counter(TC) and Call tasks according to TC	- Update TC with mod. 16 - Call task for present time slice
Safety	Check Dangerflag, trigger watch dog	- If Dangerflag is raised or external request for shut down - go to shut down procedure
Actuation	Send content of submodulation buffer(SMB) to MSRCs	- Read one byte from SMB for each MSRC - Write the bytes to ports
ReadResolver	Read the current position from resolver chip	- Select port - Wait for bus to settle - Task reading - Subtract offset - Put result into resolver read buffer(RRB)
ProcessReadings	Filter out noise by low pass filter using the data of RRB to produce a 16-bit position data	- Filter RRB - Update position
GetMode	Read the operation mode	- Get mode from interface
GetData	Read demand data	- Get demand data from interface
SendData	Write required data	- Send required data to interface
PhaseDemands	Takes the demand data and generate eight submodulated MSRCs for each demand voltage	- Get current position - Compute three SINEs - Convert each phase demand voltage into eight SMPVs - Put the SMPVs into SMB
SetLimits	Limits are set to prevent run away motion of robot	- Check position against limits - Set demand limit
Application task	Reserved for further application procedure	- Run an application procedure

The control voltage demands for each motor are passed as unsigned 16-bit data. The task *'PhaseDemand'* truncates this voltage demand into 11-bit data and computes a set of three phase voltage demands according to Eq.(5-1). But the ultimate actuating of each motor is done via three MSRCs which have only 8-bit resolution running at the counting speed of 6.144 MHz. Therefore each phase voltage demand of 11-bit data is further transformed into the eight submodulated phase voltage (SMPV) demands (8-bit long)which can be directly loaded into the MSRCs. Three sets of eight SMPVs are stored in submodulation buffer(SMB) for later use by the task *'Actuation'*. The most significant byte of 11-bit phase voltage demand determines the basic value of the submodulated phase voltage demands and the last 3 bits modify the SMPVs according to the prescribed patterns as shown in TABLE 5.2. The principle of this pattern is to evenly distribute the SMPVs according to the residual(decimal value of the last 3 bits) of each phase voltage demand and to achieve the enhanced resolution for the real actuating phase voltages applied to the windings of the motor. Finally the resulting SMPVs are sequentially executed by the task 'Actuation' at every two interrupts as shown in Fig.5.4.

TABLE 5.2 Modification Pattern of Submodulated Phase Voltage Demand

Array of SMPV demands	Last three bits of Phase Voltage Demand							
	000	001	010	011	100	101	110	111
SMPV(1)	0	*	*	*	*	*	*	*
SMPV(2)	0	0	0	0	0	*	*	*
SMPV(3)	0	0	0	0	*	0	*	*
SMPV(4)	0	0	0	*	0	*	0	*
SMPV(5)	0	0	*	0	*	*	*	*
SMPV(6)	0	0	0	0	0	0	*	*
SMPV(7)	0	0	0	*	*	*	*	*
SMPV(8)	0	0	0	0	0	0	0	0

* add 1 to the basic MSRC value

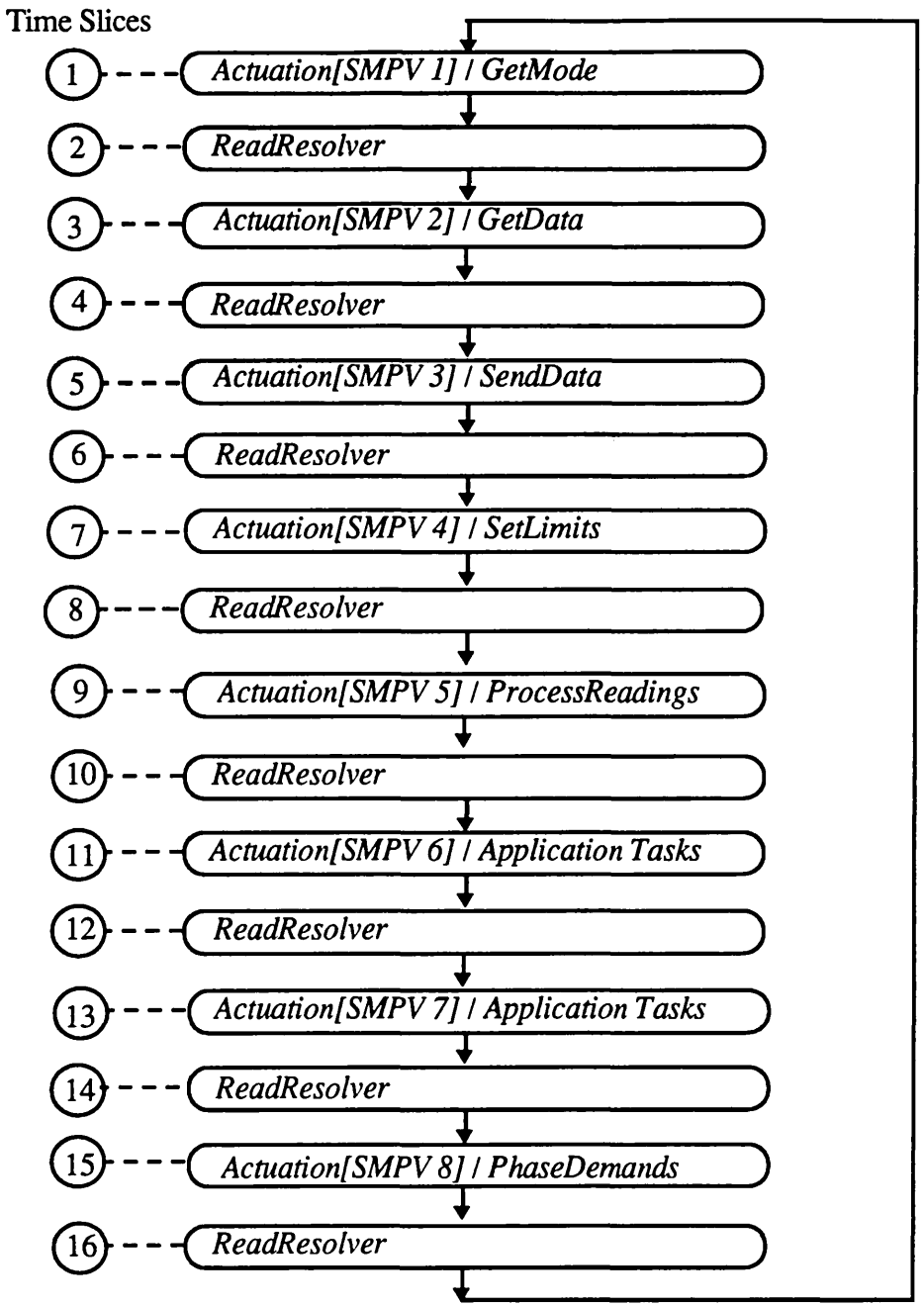

Figure 5.4 Sequence Diagram of Tasks

5.2 Experimental Parameter Estimation On IC DDR

In this section, the parameter identification algorithm is implemented on the IC DDR. The inertial parameters, kinematic data and the characteristics of the motor drive system are lumped into five numerical parameters as defined in Eq.(2-64). These lumped parameters constitute the complete set of parameters for the various identifier structures given in Chapter 3.

The key feature of this identification procedure is that it does not require any special test movement and arrangement for torque/force measuring devices. The IC DDR does not have any tachometers or accelerometers for measuring joint velocities or accelerations. Thus to obtain the joint velocities and accelerations, the measured joint angles are differentiated and differentiated again by the designed digital filter with cutoff frequency of 30 Hz.

The input for the object system(IC DDR) is the demanded voltage signal for actuating each motor. This input value is a number within the range of ± 26000. The input/output data is sampled at the rate of 100 Hz. The oscillating square wave input is selected to excite the motion because the simulation experiments in Chapter 3 reveal that excitation input of square wave type allows better estimation than the other types of excitation input. The input profiles for the joints are shown in Fig.5.5. The frequencies of excitation inputs are chosen 1Hz for joint 1 and 1.5 Hz for joint 2. In this case, the level of excitation input are set to 5000 for both joints(19.2 % of the maximum permissible input value), and the phase shift between the two excitation inputs is determined empirically so that the mechanical hard limits are not touched while the robot is in motion.

The estimation results according to each identifier structure are listed in TABLE 5.3. These estimates use 400 input/output data sampled during 4 second movement. For investigating the feasibility of the experimental estimates, the calculated parameter values from the design data (given in TABLE 3.3) can serve as a good point of comparison. A direct numerical comparison between the calculated values and experimental values has no meaning because all the calculated parameter values can be floated in proportion to the reciprocal of the input voltage gain(refer to C_h in TABLE 3.3). However the ratio between the parameters can be used as a good guideline for comparison.

In the IC DDR, the five parameters can be classified into two groups : the first is Θ_1, Θ_2 and Θ_3 which are related to the inertial properties of the arm, and the second is Θ_4 and Θ_5 which are related to friction. Interestingly, the experimental results given in TABLE 5.3 show a very similar trend when compared to the simulation results given in TABLE 3.17 which are obtained in the presence of measurement noise. In fact, there exists some noise in raw resolver readings. It is observed that the lowest 3 bits of the 16-bit data of resolver readings lose its significance due to noise.

In the light of the simulation results given in TABLE 3.17, it can be deduced that the experimental results in TABLE 5.3 are fairly good estimates of the actual parameter values. Particularly, the estimated values by identifier structure **ED** (in the second row of TABLE 5.3) show the ratio of 1 : 0.47 : 0.57 for the first group of parameters(Θ_1: Θ_2: Θ_3), and approximately 1:1 for the second group of parameters(Θ_4 : Θ_5). This results match fairly well with the ratios of the calculated parameter values from the design data(1 : 0.48 : 0.47 for the first group, and 1 : 1 for the second group). On the other hand, the values estimated by the identifier structure **MD**(first row in TABLE 5.3) are distinguished from the other estimated values. As predicted in simulation experiments, the identifier structure **MD** could result in poor estimates in the presence of a noise in the measured data. Hence it can be inferred that the values estimated by **MD** are not as accurate as the estimates by **ED**.

In the presence of noise, increasing the excitation level of the input improves the signal to noise ratio in the measured output data and can provide more informative data for better estimation. The simulation experiment has shown that an increased excitation level leads to the improvements in parameter estimation, especially the identifier structure **ED** allows better estimates than the other identifier structures (refer to TABLE 3.19).

In practical situations, the excitation levels of inputs can not be increased as desired because the mechanical hard limits are easily reached while the robot is in motion. Thus, the excitation level and phase shift between the two excitation inputs are experimentally adjusted not to touch the mechanical hard limit during the motion.

The new experimental excitation inputs (level is increased up to 23.1 % of maximum permissible input) for joints are presented in Fig.5.6a. For better estimation, 1000

input/output data were collected at the sampling rate of 100 [Hz] during 10 second motion under these new excitation inputs. The measured joint angles are displayed in Fig.5.6b, and the calculated angular velocities and accelerations by the differentiating filter are represented in Fig. 5.6c and 5.6d respectively.

The estimated parameter values under this new excitation inputs are listed in TABLE 5.4. Comparing these new estimation values with the simulation results given in TABLE 3.19, it can be easily found that the estimated parameter values in both tables show a close similarity though the magnitudes are different. In the simulation results given in TABLE 3.19, the identifier structure **ED** shows outstanding estimation performance, and identifies the parameters within 1.7 % error to the true parameter values. Hence this comparison suggests that the experimental parameter values by **ED** (second row in TABLE 5.4) are the best candidates for the actual parameter.

The stepwise estimation profile for each parameter is represented in Fig.5.7. As can be seen in this figure, the estimated values are stabilized after around 400 sampled data points.

The ratios for the first and second group of parameters are 1 : 0.49 : 0.56 (=Θ_1: Θ_2: Θ_3) and 1: 1.3(=Θ_4 : Θ_5) respectively. The ratio for the first group matches well with the ratio obtained from the design data(1 : 0.48 : 0.47), but the ratio for the second group(1: 1.3) is slightly different from the previous ratio(approximately 1 : 1). However the ratio for the second group does not carry the same importance as the first group because the parameter values of the second group (calculated parameter values in TABLE 3.3) are computed from the simple assumption for frictional dissipation energy based on the viscous friction model. Therefore it seems rather reasonable that the estimated value of Θ_5 is 30 % greater than Θ_4 because the joint 2 for elbow arm has longer drive train than joint 1 due to the pulley mechanism and could have bigger frictional dissipation. These estimated parameters(second row in TABLE 5.4) will be used in the performance evaluation of real-time dynamic compensation scheme, and their adequacy for control purposes will be justified through experiments.

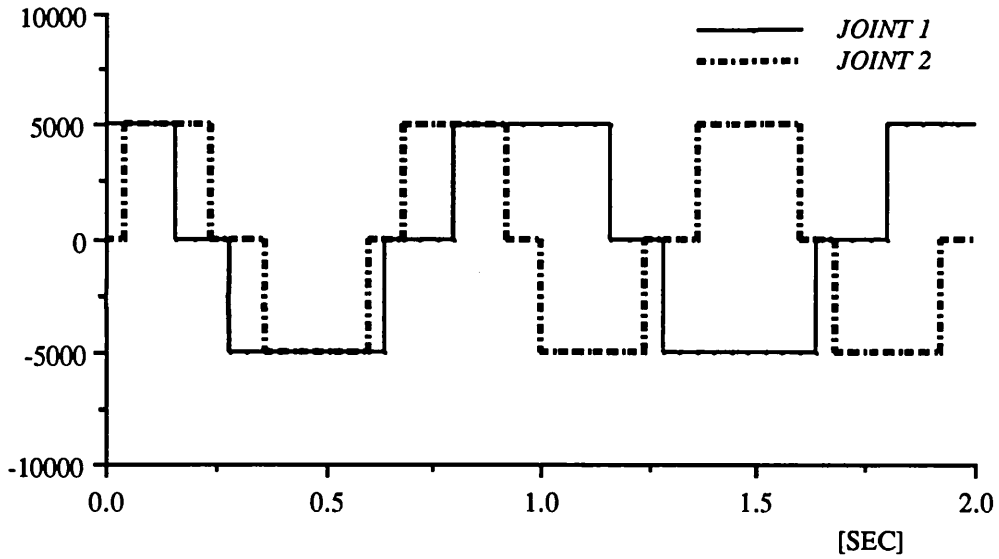

Figure 5.5 Excitation Input Profile(level.5000)

TABLE 5.3 Estimated Parameters of IC DDR

(Excitation Input level : 5000)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	399.61	230.50	191.66	426.13	588.77
ED	454.83	214.88	257.03	844.36	844.28
SC	421.56	190.90	223.26	886.71	876.29
GD	426.37	190.91	224.81	875.46	960.66
VD	422.98	191.48	220.35	880.45	959.56
PD	422.89	191.0	220.05	884.44	959.64

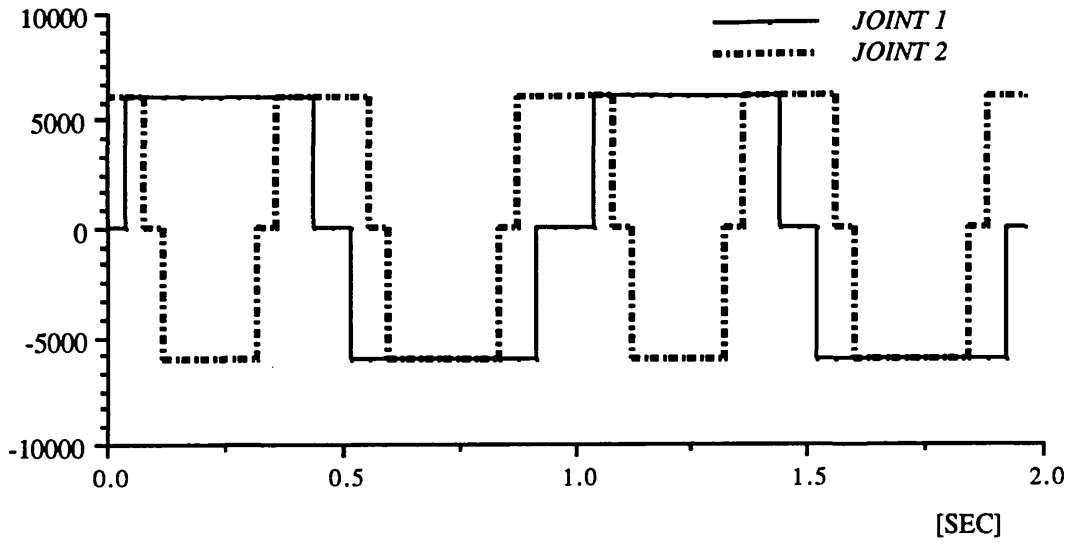

Figure 5.6a Excitation Input Profile(level.6000)

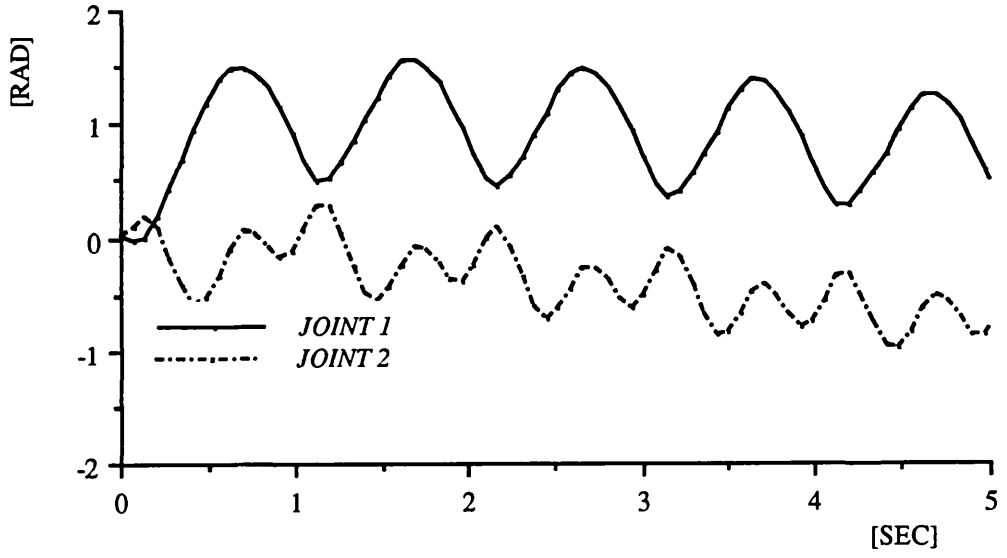

Figure 5.6b Measured Joint Angles

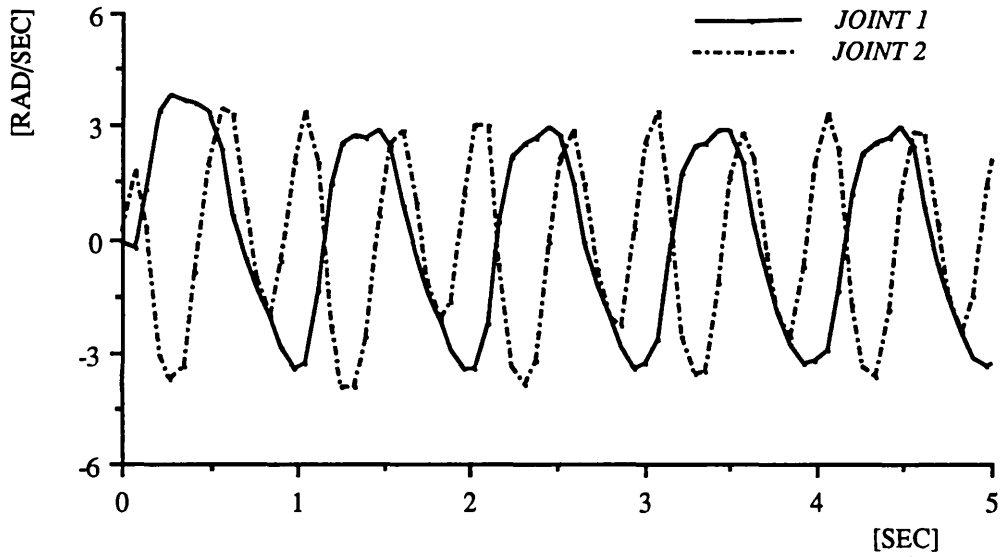

Figure 5.6c Calculated Angular Velocities

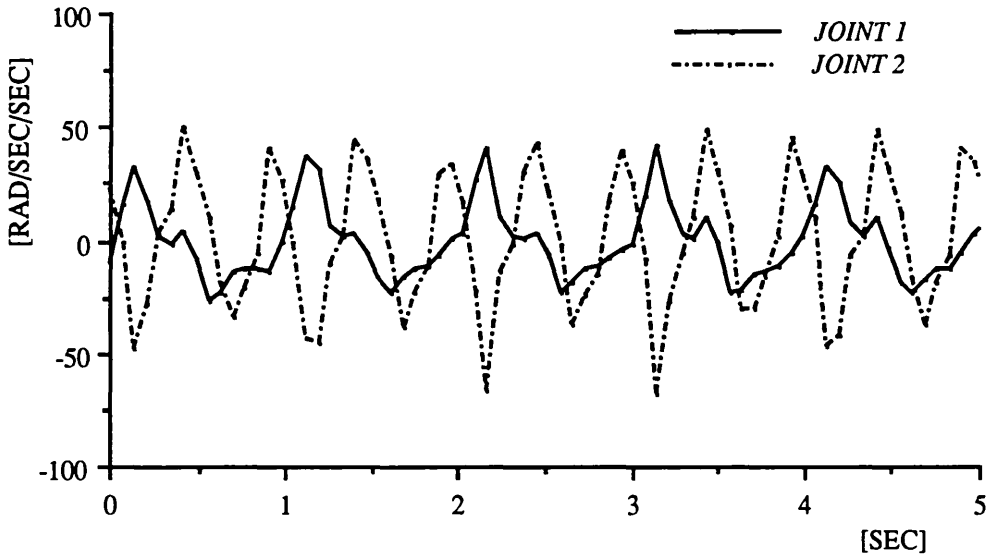

Figure 5.6d Calculated Angular Accelerations

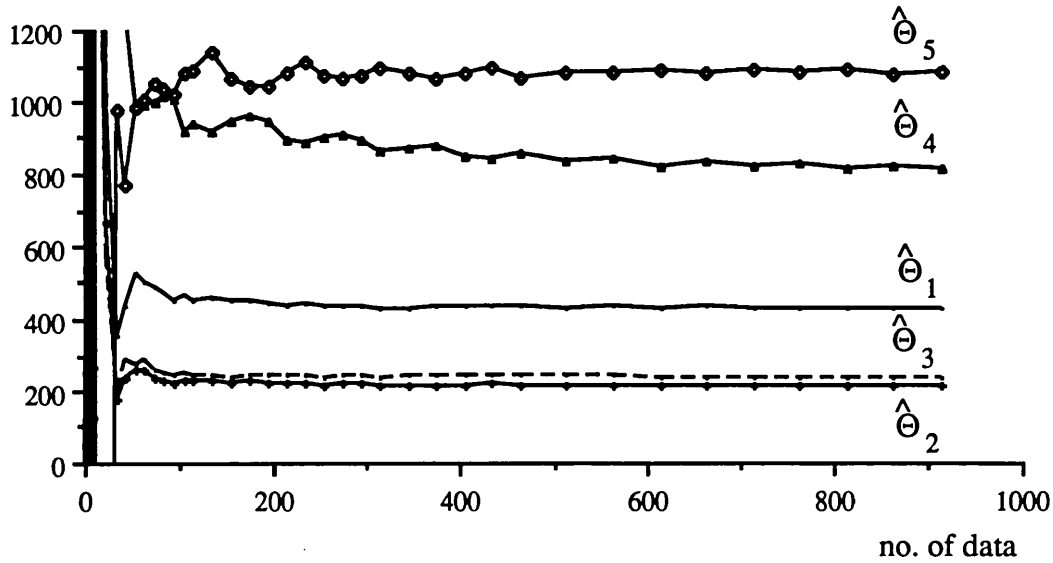

Figure 5.7 Stepwise Estimation By The Identifier Structure **ED**

TABLE 5.4 Estimated Parameters of IC DDR

(Excitation Input level : 6000)

Identifier	$\hat{\Theta}_1$	$\hat{\Theta}_2$	$\hat{\Theta}_3$	$\hat{\Theta}_4$	$\hat{\Theta}_5$
MD	424.94	207.87	215.41	834.82	860.46
ED	432.28	213.26	240.40	822.28	1073.4
SC	417.32	196.19	217.95	875.84	805.30
GD	422.44	195.77	217.76	924.52	942.82
VD	418.93	196.76	213.28	942.62	920.77
PD	419.60	196.41	212.21	946.88	920.48

5.3 Experimental Performance of Control Schemes

In this final section, the experimental performances of real-time dynamic compensation scheme(**FDC**) based on the dynamic model **PD** are compared with the independent joint control scheme(**IJ**). The distinguishing features of the complete dynamic model **PD** can be restated as

- Inherently discrete-time dynamic model
- Actuator dynamics are incorporated. Hence, a control input for each joint is expressed as a scaled numeric number, which will be directly used to activate the switching amplifier of the motor in the axis controller, rather than the physical quantities such as joint torque/force.
- Velocity and acceleration information is not required to be measured because the dynamic model is only dependent on position information

Nominal position input $A_i^*(k)$ is selected with four positional feedback gains as belows:

$$A_i^*(k) = A_{id}(k+1) + K_{i0}\, E_i(k) + K_{i1}\, E_i(k-1) + K_{i2}\, E_i(k-2) + K_{i3}\, E_i(k-3) \qquad (5\text{-}2)$$

Feedback gains K_{ij} (j=0,..,3) for each joint are determined by placing the poles of the closed-loop error equation (Eq.4-37). In this experiment, the feedback gains are selected by locating the pole at z=0.6 for each joint.

The desired trajectory for generating the demanded position A_{id} in Eq.(5-2) are chosen to be a cycloid trajectory in joint space.

$$A_{id}(t) = \delta_i\, [t / t_f - 1/2\pi\, \mathrm{SIN}(2\pi\, t / t_f)] \qquad (5\text{-}3)$$

Where δ_i and t_f represents the total travelling angular displacement and the ending time of the trajectory. This trajectory has the smooth evolution of velocity and acceleration profile though they are not explicitly used in the control scheme. But the smoothness of velocity/acceleration contributes to minimize an undesirable mechanical vibration at the beginning and the end of movement by avoiding the possible sudden big changes of actuating input for the motor.

In the independent joint control scheme, linear control law is designed for each joint based on the assumption that the joints are decoupled and linear. The control inputs applied to the joints at each sampling instant are computed as

$$U_i(k) = b_{i1}\, U_i(k-1) + a_{i1}\, E_i(k) + a_{i2}\, E_i(k-1) + a_{i3}\, E_i(k-2) \qquad (5\text{-}4)$$

Each joint is tested independently with various angle demands while the other joint is locked. In this way the coefficients of Eq.(5-4) are adjusted and tuned empirically to produce the best performance for each separate joint movement. The addition of $U_i(k-1)$ in Eq.(5-4) acts as preview term. From the control theory point of view, addition of $U_i(k-1)$ term creates a feedforward zero in the transfer function, thus increases the speed of response of the system. The independent control scheme runs at 1.5 [KHz] sampling rate, while the real-time dynamic compensation scheme is implemented at 100 [Hz]. For comparison purposes of the two control schemes, an identical sampling rate seems reasonable. But at low sampling frequencies, tracking error by the **IJ** scheme are very large and out of the range of reasonable comparison due to serious interactions between the joints under the high speed movement. For this reason, the best performances of **IJ** scheme have been compared with those of **FDC** scheme at the sampling rate of 15 times higher than the **FDC** scheme.

Unlike the linear systems, for which a specific demand(such as ramp) can be used to evaluate the control performance, a particular choice of demanded movement can not characterize the whole control performance of a highly nonlinear and coupled system such as IC DDR. Four test movements, which are illustrative enough to provide insight into the effect of dynamic compensation, are arranged in this experiments(TABLE 5.5).

In the first two test movements, the joints move in the opposite direction, and in the last two the joints are commanded to rotate in the same direction. Joint 1 is commanded to move two times faster than joint 2 in the test movement I(opposite direction) and III(same direction). For test movement II(opposite direction) and IV(same direction), joint 2 moves approximately two times faster than joint 1. All the joint movements start from its home position. Home position is defined as the fully stretched arm configuration.

TABLE 5.5 Test Movement Specification

Test Movement	Final Joint Angle [deg]		t_f [sec]
	Joint 1	Joint 2	
I	-90	40	0.60
II	40	-90	0.56
III	90	45	0.60
IV	90	150	0.72

The tracking error of each joint and the tip position error under the test movement I are shown in Fig.5.8a and 5.8b respectively. For the **FDC** scheme, the tracking errors for joint 1 range from -0.60 [deg] to 0.31 [deg] and joint 2 goes from -0.43 [deg] to 0.45 [deg], while **IJ** scheme shows approximately 12 times bigger tracking errors of joint angle than **FDC** scheme. The maximum tip position error is 46.70 [mm] for **IJ** scheme and 3.86 [mm] for **FDC** scheme.

When the test movement II is commanded, almost identical results of tracking error are observed, but the error patterns for joints are reversed (refer to second row in TABLE 5.6). Fig.5.10a and 5.10b represent the tracking errors for the joints and the tip position tracking error respectively when the arms move in the same direction but joint 1 is two times faster than joint 2(test movement III). Comparing these results with the case of test movement I, the tracking error for **IJ** scheme are much increased, especially the tracking errors for joint 1 range from -10.8 [deg] to 12.23 [deg] (tracking error for joint 1 vary from -8.67 [deg] to 1.54 [deg] for test movement I) and maximum tracking error of tip position is increased approximately two times than the case of test movement I. However, **FDC** scheme maintains the almost same magnitude of tracking error compared to the case of test movement I. **FDC** scheme reduces the tracking errors significantly for joint 1 and joint 2 with peak error of only 0.63 [deg] and -0.71 [deg] respectively. The maximum tip position error by **FDC** scheme is reduced 20 times less than that of **IJ** scheme.

In the following high speed experiment of test movement IV, the joints move from (0,0) [deg] to (90,150) [deg] with peak velocities of 125 [deg/sec] and 208 [deg/sec], and peak accelerations of 1091 [deg/sec^2] and 1818 [deg/sec^2] for joint 1 and joint 2 respectively.

To give a clear idea of the relative performances, the desired trajectories and joint position tracking profiles are displayed in Fig.5.11a and 5.11b. At the same time, tracking errors for each joint and tip position error profile are also given in Fig. 5.11c and 5.11d. These figures demonstrate again the superior tracking performances of **FDC** scheme to **IJ** scheme. In this high speed movement, the tracking errors of **FDC** scheme for joint 1 vary from -0.38 [deg] to 0.68 [deg] and from -0.92 [deg] to 0.35 [deg] for joint 2, while the tracking errors of **IJ** scheme ranges from -7.54 [deg] to 7.55 [deg] for joint 1 and from -4.0 [deg] to 18.24 [deg] for joint 2. The tip position tracking error of **IJ** scheme is very large and reaches the maximum value of 122.6 [mm]. However, as can be seen in Fig.5.11a and 5.11b, the **IJ** scheme still can be used as a practical alternative for point-to-point application of industrial robots with relaxed tracking accuracy requirement. For comparison purpose, the maximum tracking errors are summarised in TABLE 5.6. In the interpretation of experimental results, it should be noticed again that the results of **IJ** scheme are obtained at 15 times faster sampling rate than **FDC** scheme. As was expected in the simulation analysis of the previous chapter, increasing the sampling rate would upgrade the tracking performance of **FDC** scheme. But the present sampling rate for the **FDC** scheme is sufficient for demonstrating the efficacy of the real-time dynamic compensation over the fixed control scheme such as **IJ** scheme.

TABLE 5.6 Maximum Tracking Errors

Test Movement	Max. Joint Angle Error [deg]				Max. Tip Position Error [mm]	
	FDC		**IJ**			
	Joint 1	Joint 2	Joint 1	Joint 2	**FDC**	**IJ**
I	-0.60	0.45	-8.67	5.28	3.86	46.70
II	0.52	-0.57	4.98	-8.62	4.25	46.16
III	0.63	-0.71	12.23	-7.02	4.07	90.58
IV	0.68	-0.92	7.55	18.24	3.78	122.58

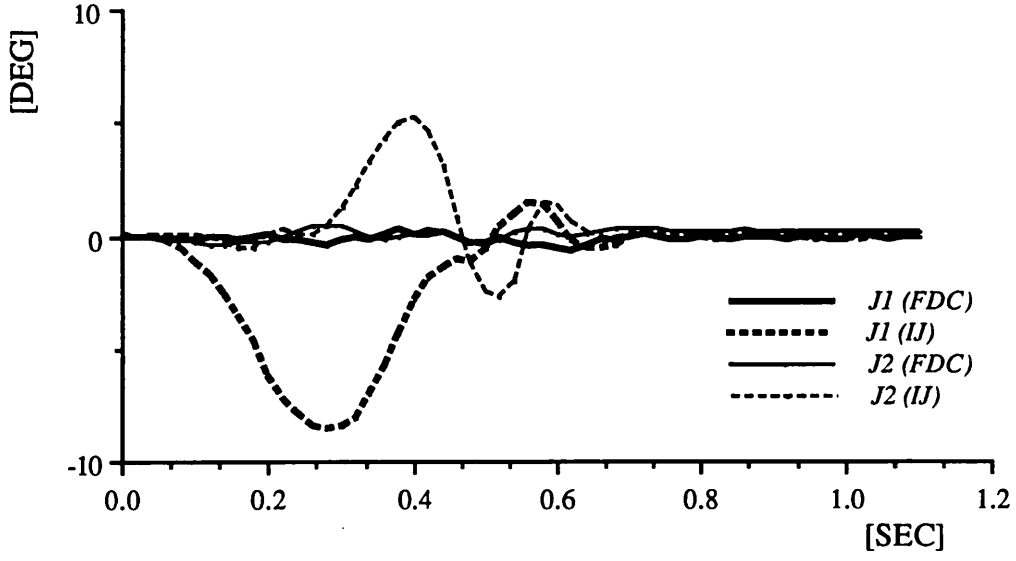

Figure 5.8a Joint Angle Tracking Error For Movement I

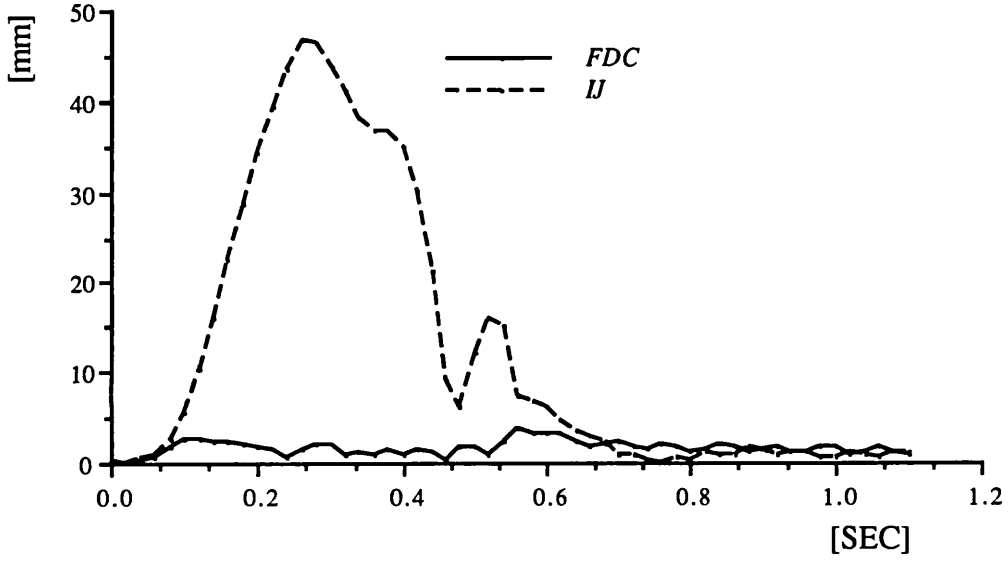

Figure 5.8b Tip Position Tracking Error For Movement I

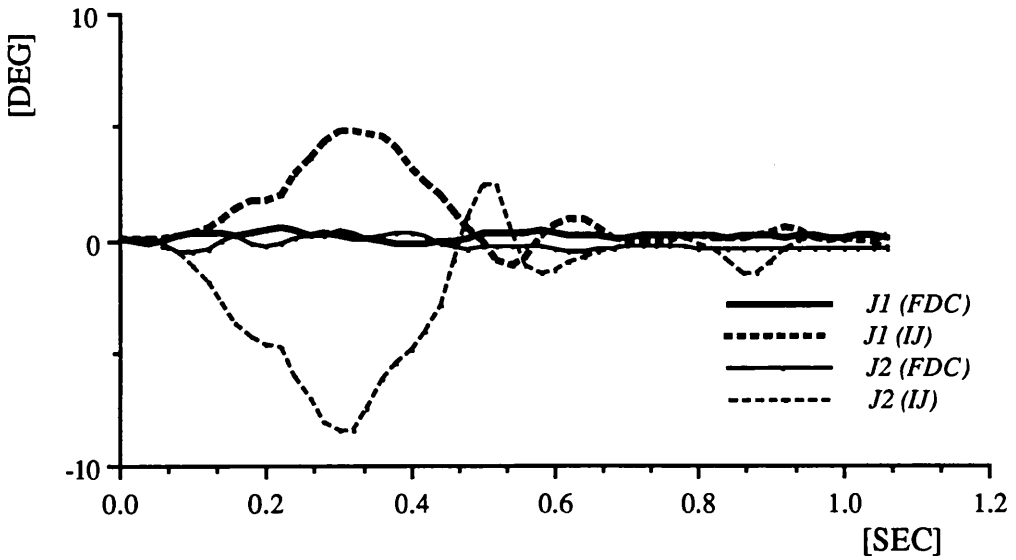

Figure 5.9a Joint Angle Tracking Error For Movement II

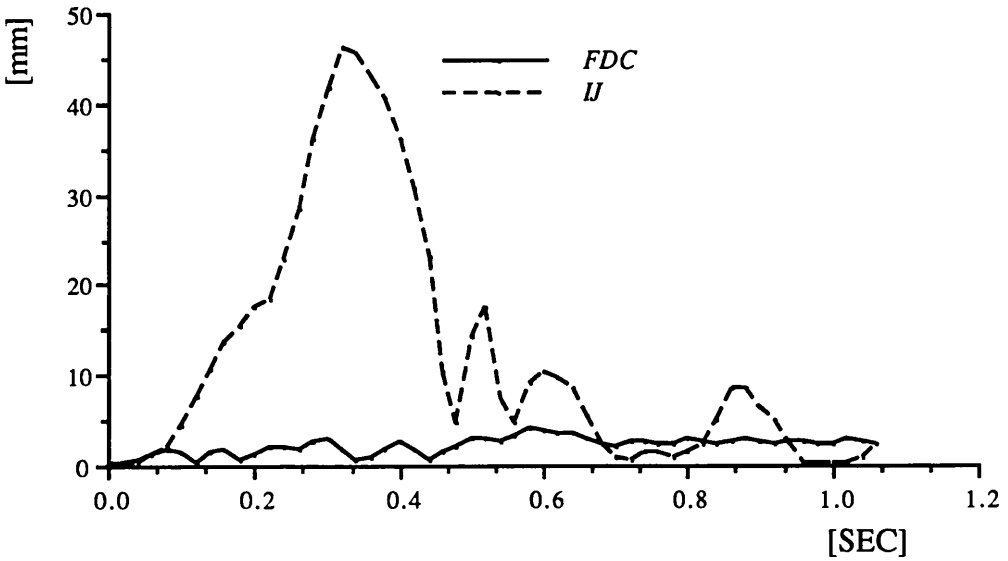

Figure 5.9b Tip Position Tracking Error For Movement II

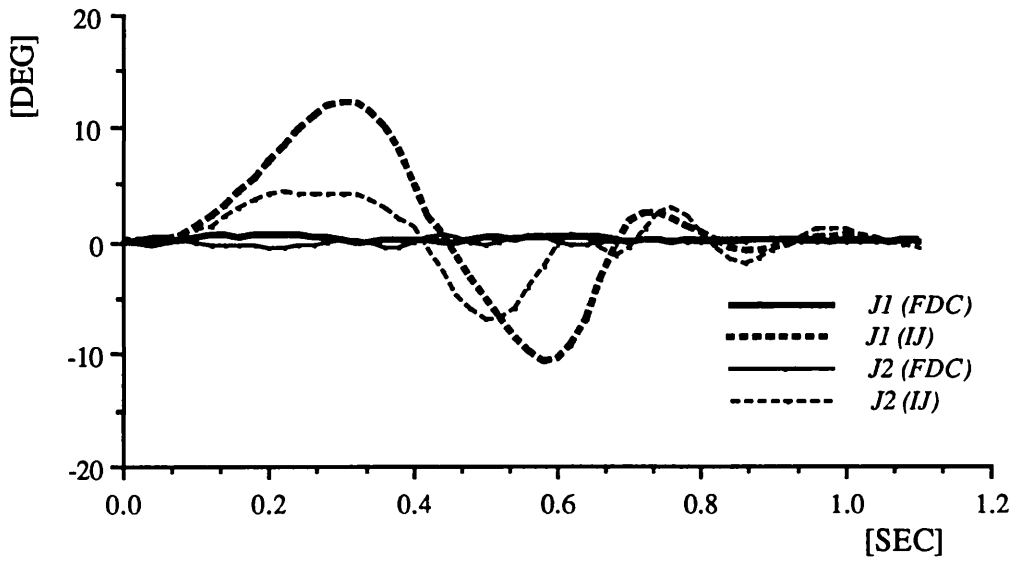

Figure 5.10a Joint Angle Tracking Error For Movement III

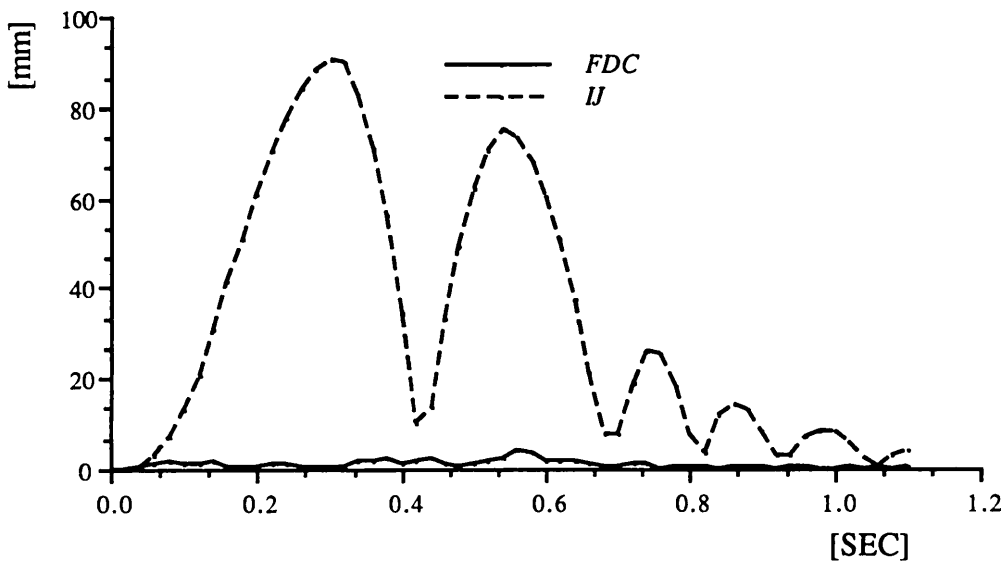

Figure 5.10b Tip Position Tracking Error For Movement III

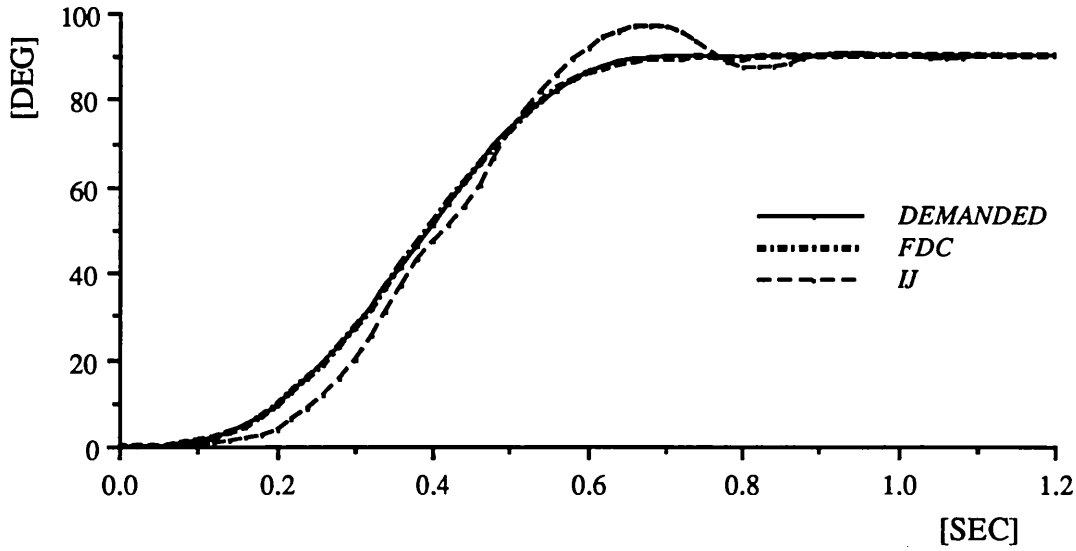

Figure 5.11a Angular Position Tracking of Joint 1 for Movement IV

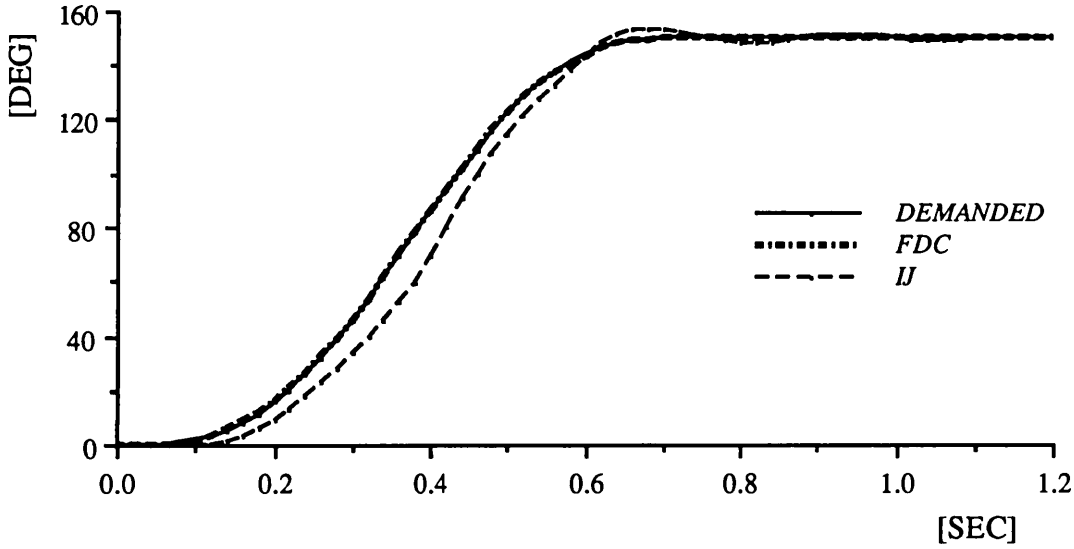

Figure 5.11b Angular Position Tracking of Joint 2 for Movement IV

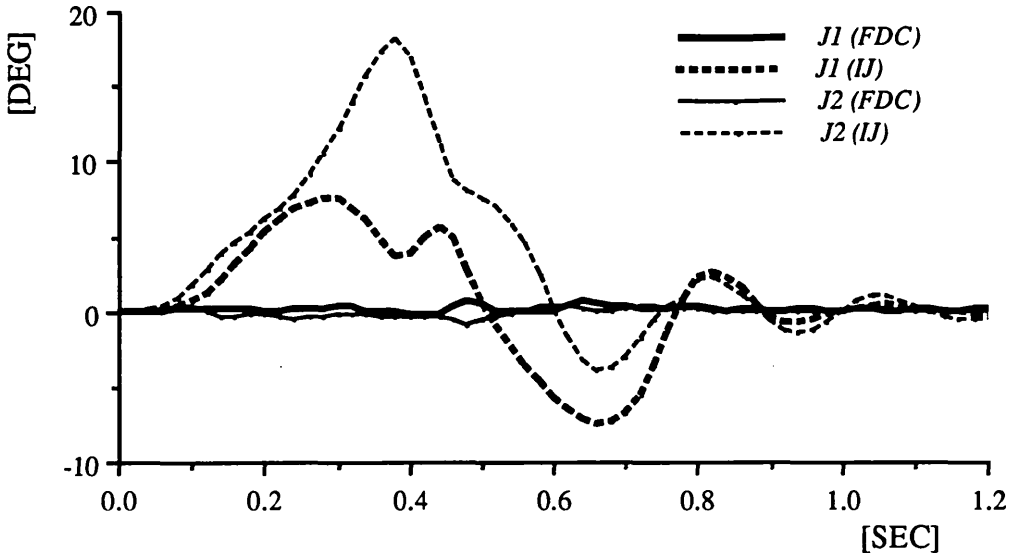

Figure 5.11c Joint Angle Tracking Error For Movement IV

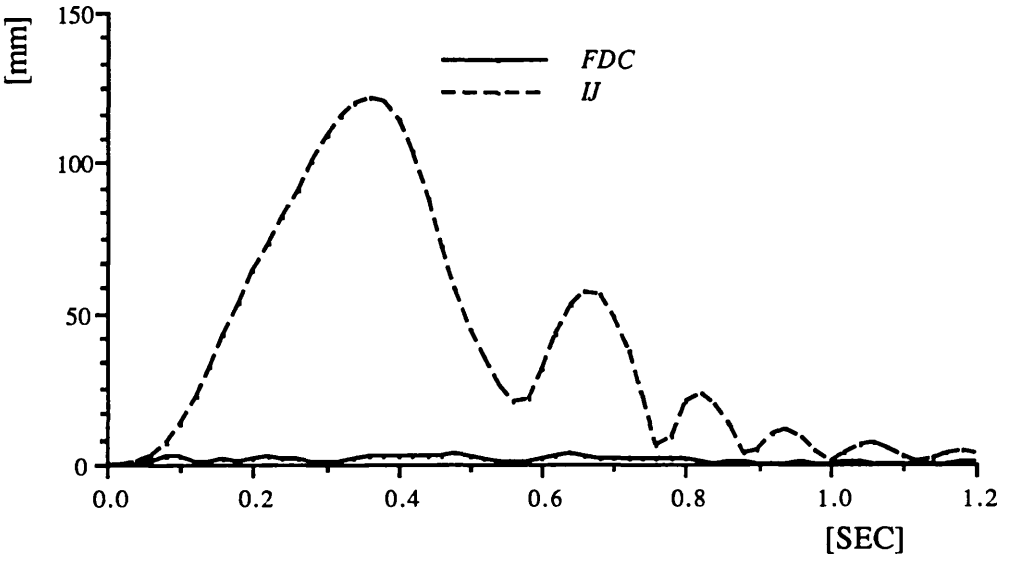

Figure 5.11d Tip Position Tracking Error For Movement IV

6. CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

This research has contributed primarily to the following four areas: formulation of the discrete dynamic model of robot system which is particularly suitable for a computer-controlled robot system, identification of parameters, analysis and evaluation of model based control scheme, implementation of parameter identification algorithm and performance evaluation of real-time dynamic compensation scheme on the direct drive robot. The conclusions and suggestions for further work will be focused on these main areas and presented in the next two sections.

6.1 Conclusions

The SCARA type direct drive robot named IC DDR has been designed. The positioning arm is mounted horizontally, thus avoids gravity forces acting upon the motors of the principal axes. Furthermore the motors of the positioning arm are mounted at the base to reduce the arm weight. This structure is well suited for high speed applications of assembly-type works. Since the motor rotors are directly coupled to the loads, the drive system has no backlash, low friction and high mechanical stiffness. However, with the aforementioned advantages over a conventional geared robot, the problem of arm dynamics is still alive, and the control problem becomes more crucial due to substantial increase of interactions between the arm links.

The systematic procedure for the formulation of robot dynamics is introduced and applied to derive the arm dynamic model of IC DDR. Actuator dynamics are treated as an integral part of the arm dynamics. Each joint actuator system consisting of a high torque BDC motor-resolver combination and a driving amplifier is modelled. Initially a third order model is derived but simplified into the second order model by neglecting a fast pole(located far from origin in z-plane) which is mainly influenced by the electrical

time constant. This dynamic model of the motor system is incorporated to construct the closed-form dynamics model of the whole system.

A natural way to describe a computer controlled system is to use a difference form of dynamic equations. Through investigating certain invariant properties of the Lagrange equation, integration factors which allow the nonlinear differential equations to be integrated over the sampling period are found. Using this integrable form, the complete discrete dynamic model of IC DDR is obtained. A key feature of this discrete dynamics model is that the second derivatives(accelerations) are eliminated through the integration procedure.

The constituent parameter values of the dynamic model have a decisive role in the performance of model based control scheme. There are two distinct uses for parameter identification. For control purpose, matching the input/output behaviour of the dynamic model is a great concern, while for recognition, what matters is to match the estimated values to real parameter values.

In the section of parameter identification, five identifier structures are introduced, and their estimation performances associated with a measurement noise, types of excitation input and excitation frequencies, etc. are analyzed through the simulation experiments. At the same time, the parameters are calculated from the engineering drawings and design specifications by modelling relevant mechanical parts as a combination of simple geometric shapes. These calculated parameter values are used throughout the simulation experiments and also used as a guideline for the experimental results of parameter estimation. The important feature of this identification procedure is that it does not require any special test movement and no sensory data from torque/force measuring devices. Particularly, when only positional data with a substantial measurement noise is obtained, the energy difference equation (identifier structure **ED**) shows the best estimates over the other identifier structures.

In the section on control, a detailed design procedure for the construction of a conventional position controller is presented. Trajectory following errors of this simple controller are analyzed, but for the highly coupled robot system such as IC DDR, decentralized scheme such as independent joint control on the axis level is not sufficient to deliver the full potential power due to strong dynamic interactions between joints.

The decoupling performance of the newly developed discrete dynamics model(**VD**) is compared with those of continuous dynamic model(**SC**) and conventional discrete dynamic model(**GD**) in the inverse dynamic sense. In these simulation experiments, the piecewise-constant actuating inputs for joints are computed using the dynamic model along with the reference trajectory, and these inputs are applied in open loop fashion to actuate the robot along the reference trajectory. The inverse dynamic performance of the newly developed discrete dynamic model(**VD**) is shown to be superior to that of the other dynamic models(**SC**, **GD**), thus dynamic model **VD** is adopted for use in the subsequent dynamic compensation control scheme.

For improvement in the tracking performance by dynamic compensation, firstly a feedforward compensation scheme is investigated as a method to cancel out interactions between the joints. The weak point in the feedforward scheme is that the linear feedback portion of control law acts independently and a corrective input at one joint could perturb the other joints, whereas ideally speaking, the corrective inputs should cancel out the nonlinear and coupling interactions between the joints. Thus, the simulation experiments have shown that feedforward scheme serves to improve the tracking performance, but not as much as expected at first.

On the other hand, in the real-time full dynamic compensation scheme, the linear feedback portion sends its output through the dynamic model. The real-time full dynamic compensation scheme(**FDC**) is more accurate than the feedforward control scheme because the action of feedback control is decoupled through the inverse dynamics and the actual dynamic states are reflected in the compensation. Therefore if the modelling and parameter values of the arm are exact, then the arm would follow the prespecified trajectory with little error.

It is observed that the **FDC** scheme can achieve a remarkable trajectory tracking performance by effectively cancelling a dynamic interaction. However, in reality, the compensation could not be expected to be perfect due to modelling and parameter errors. Especially, the **FDC** scheme based on an underestimated parameter values exhibits greater tracking errors than in the case of an overestimated parameter values. Although a rigourous mathematical verification of the relationship between a possible instability and the marginal extent of parameter errors is not given in this thesis, the simulation experiments demonstrate that **FDC** scheme is still robust under a sizable

amount of parameter errors and results only in the increased tracking errors of trajectory. In such circumstances, it is shown that high sampling rates are important because they result in a stiffer system that is capable of effectively rejecting disturbances.

The aforementioned control scheme and the dynamic model require real-time information of velocities in generating the input signals for joints. Since the vast majority of robots have only a positional sensor at each joint, eliminating the dependence of velocity information in both control law and dynamic model is quite attractive from the implementational point of view. Hence the velocity dependent discrete dynamic model(**VD**) has been used to construct a new form of dynamic model(**PD**) which is expressed in difference equations using only positional information. This velocity independent discrete dynamic model(**PD**) has a great implementational advantage because an additional sensory installation for direct measurement of velocities or an indirect estimating procedure for the higher derivatives can be excluded. The trajectory following performance of **FDC** scheme based on this new dynamic model is also evaluated, and found to be comparable to the results based on the velocity dependent discrete dynamics model(**VD**).

The hardware of the IC DDR controller has been designed to consist of two hierarchical levels: the master controller consisting of dual 32-bits MC68020 microprocessors and axis controllers based on TMS320C25 digital signal processor. The master controller coordinates the axis controllers, while axis controller for each joint is responsible for the elementary operations to realize the command prescribed by the master controller.

The results of experimental implementation of the parameter identification algorithm are presented, and the estimation performances of various identifier structures are evaluated. The real-time full dynamic compensation scheme has been implemented at the sampling period 10 [msec], and its trajectory tracking performances are compared to the best tuned independent joint control scheme which runs at 15 times faster sampling rate than the dynamic compensation scheme. The experimental results show that a dynamic compensation scheme can improve trajectory following accuracy significantly ,and verify that the estimated parameters and dynamic model of the arm are accurate for control purposes.

6.2 Suggestions For Further Work

The dynamic compensation scheme based on dynamic model is a good approach to control the robot when parameters are sufficiently known and computing power of control computer is sufficient. When the knowledge about parameters is poor, the dynamic compensation scheme may not decouple and linearize the system dynamics as intended, thus may cause the system to be unstable. Based on simulation results and experiments, the dynamic compensation scheme appears to be robust to the bounded parameter errors. Though many researchers conjecture the robustness of the model based dynamic compensation scheme, there has been no rigourous proof of the robustness problem. Therefore proving the robustness will be a good way to extend the application spectrum of the model based dynamic compensation scheme.

The primary source of trajectory tracking errors in the dynamic compensation scheme is twofold: lack of precise knowledge of true parameter values and the realizable replacements for an inaccessible state information. Even if parameters are accurately identified, the error driving disturbances can not be zeroed out due to these realizable replacements. Thus incorporating some adequate prediction model in adaptive or fixed form into the control law may reduce the error driving disturbances and could improve the tracking performance.

In the control scheme discussed so far, the desired trajectory is given in joint space. However, the robot end-effector often has to follow straight line or other path shape described in cartesian coordinates. In this case, the kinematics and other transformation loop should be incorporated into the control scheme. The design of cartesian based controller which can suppress the cartesian tracking errors uniformly over all possible configurations can be an another research direction.

Finally, the next research area of more sophisticated control scheme is a hybrid position/force control problem. In parts-mating assembly works, control of a contact forces is important. A hybrid position/force controller could enhance a positional accuracy one step higher by monitoring a contact force. Incorporating force control capability in control system could make important progress toward using robots for extremely precise assembly works.

REFERENCES

[Adept 85] Adept Technology Inc., "The AdeptOne Manipulator", 1985.

[An 85] Chae H. An, C. G. Atkeson, J. M. Hollerbach, "Estimation of Inertial Parameters of Rigid Body Links of Manipulators", Proceedings of 24th Conf. on Decision and Control, pp990-995, Dec. 1985.

[An 86] Chae H. An, C. G. Atkeson, J. M. Hollerbach, "Estimation of Inertial Parameters of Rigid Body Links of Manipulators", A. I. Memo 887, MIT Artificial Intelligence Lab., Feb., 1986.

[An 87] Chae H. An, C. G. Atkeson, "Experimental Evaluation of Feedforward and Computed Torque Control", Proceedings 1987 IEEE Int'l Conf. on Robotics and Automation, pp165-168.

[Antoniou 79] Andreas Antoniou, "Digital Filters: Analysis and Design", McGraw-Hill, 1979.

[Armstrong 79] W. W. Armstrong, "Recursive Solution to the Equations of Motion of an N-linked Manipulator", Proc. 5th World Congress on Theory of Machines and Mechanisms, Montreal, July, pp1343-1346, 1979.

[Asada 83] H. Asada, T. Kanade,"Design of Direct-Drive Mechanical Arms", ASME J. of Vibration, Acoustics, Stress, and Reliability in Design, Vol. 105, pp312-316, 1983.

[Asada 84] H. Asada, K. Yousef-Toumi, "Analysis and Design of A Direct Drive Arm with A Five-Bar-Link Parallel Drive Mechanism", Trans. of ASME, J. of Dynamic Systems, Measurement, and Control, Vol. 106, pp225-230, Sep. 1984.

[Asada 87] H.Asada, K. Youcef_Toumi, "Direct-Drive Robots: Theory and Practice", MIT Press, 1987.

[Astrom 80] K. J. Astrom, "Why use Adaptive Techniques For Steering Large Tankers", Int'l J. of Control, Vol. 32, No. 4, pp689-708, 1980.

[Astrom 89] K. J. Astrom, Bjorn Wittenmark, "Adaptive Control", Addison-Wesley, 1989.

[Atkeson 85] Christopher G. Atkeson, C. H. An, J. H. Hollerbach, "Rigid Body Load Identification for Manipulators", Proceedings of 24th Conf. on Decision and Control, pp996-1002, 1985.

[Coiffet 83] P. Coiffet, "Robot Technology. vol. 2: Interaction With the Environments", Kogan Page, London, 1983.

[Craig 86] John. J. Craig, "Introduction to Robotics", Addison-Wesley,1986.

[Craig 88] John. J. Craig, "Adaptive Control of Mechanical Manipulators", Addison-Wesley, 1988.

[Denavit 55] J. Denabit, R. S. Hartenberg, "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices", J. Applied Mechanics, pp215-221, June 1955.

[Dubowsky 79] S. Dubowsky, D. T. Desforges, "The Application of Model-Referenced Adaptive Control to Robotic Manipulators", Trans. ASME, J. Dynamic System, Measurement and Control, Vol. 101, pp193-200, 1979.

[Farsi 86] M. Farsi, J. W. Finch, K. Warwick, et al., "Simplified PID Self-Tuning Controller for Robotic Manipulators", Proceedings of 25th Conf. on Decision and Control, pp1886-1887, Athens, Greece, Dec. 1986.

[Fournier 84] S. J. Fournier, R. J. Schilling, "Decoupling of A Two-Axis Robotic Manipulator Using Nonlinear State Feedback: A Case Study", Int'l J. of Robotics Research, Vol.3, No. 3, pp76-86, 1984.

[Franklin 80] Gene F. Franklin, J. David Powell, "Digital Control of Dynamic Systems", Addison-Wesley, 1980.

[Freund 82] E. Freund, "Fast Nonlinear Control With Arbitrary Pole-Placement For Industrial Robots and Manipulators", Int'l J. Robotics Research, Vol. 1, No. 1, pp65-78, 1982.

[Friedlander 84] B. Friedlander, B. Porat, "The Modified Yule-Walker Method of ARMA Spectral Estimation", IEEE Trans. on Aerospace Electronics Systems, Vol. AES-20, No.2, pp158-173, Mar. 1984.

[Fu 87] K. S. Fu, R. C. Gonzalez, C. S. G. Lee, "Robotic-control, Sensing, Vision, and Intelligence", McGraw-Hill, 1987.

[Goldenberg 89] A. A. Goldenberg, J. A. Apkarian, H. W. Smith, "An Approach to Adaptive Control of Robot Manipulators Using the Computed Torque Technique", Trans. ASME, J. of Dynamic Systems, Measurement, and Control, Vol. 111, pp1-8, 1989.

Greenspan 73] Daniel Greenspan, "Discrete Models", Addison-Wesley, 1973.

[Hamming 89] R. W. Hamming, "Digital Filters", Prentice-Hall, 3rd ed., 1989.

[Hollerbach 80] John M. Hollerbach, "A Recursive Lagrangian Formulation of Manipulator Dynamics and A Comparative Study of Dynamics Formulation Complexity", IEEE Trans. on Sys., Man and Cybernetics SMC-10, pp730-736, Nov. 1980.

[Hooker 65] W. W. Hooker, G. Margulies, "The Dynamical Attitude Equations For A n-body Satellite", J. of Astronautical Science, XII(4), pp123-128, Winter 1965.

[Horak 84] D. T. Horak, "A Fast Computational Scheme for Dynamic Control of Manipulator", American Control Conference, pp625-630, June 1984.

[Inland 87] Inland Motor, "Brushless Motors and Drive Systems", Kollmorgen Corp., 1987.

[Jacobs 80] O. L. R. Jacobs, "When is Adaptive Control Useful ?", Proc. IMA Conf. on Control Theory at Sheffield, Academic Press, 1980.

[Khosla 85] P. K. Khosla, T. Kanade, "Parameter Identification of Robot Dynamics", Proceedings of 24th Conf. on Decision and Control, pp1754-1760, Dec. 1985.

[Khosla 86a] P. K. Khosla, T. Kanade, "Experimental Evaluation of the Feedforward Compensation and Computed-Torque Control Schemes" American Control Conference, pp790-798, June 1986, Seattle, WA.

[Khosla 86b] P. K. Khosla, "Real-Time Control and Identification of Direct-Drive Manipulators", Ph.D. Thesis, Carnegie-Mellon Univ. 1986.

[Khosla 87] P. K. Khosla, "Choosing Sampling Rates for Robot Control", Proc. 1987 IEEE Int'l Conf. on Robotics and Automation, pp169-174.

[Khosla 89] P. K. Khosla, T. Kanade, "Real-time Implementation and Evaluation of Computed Torque Scheme", IEEE Trans. on Robotics and Automation, Vol.5, No.2, pp245-253, 1989.

[Koivo 83] Antti J. Koivo, Ten-Huei Guo, "Adaptive Linear Controller For Robotic Manipulator", IEEE Trans. on Automatic Control, Vol. AC-28, No. 2, pp162-171, 1983.

[Kim 90] S. H. Kim, "Dynamic Equations of IC DDR and Computation of Constituent Parameters", 2nd Year Technical Report, Imperial College, 1990.

[KIT 89] Korea Institute of Technology, "A Study of The Development of Direct Drive Robots and Their Applications(II)", Report No. TR-01-89, Korea, 1989.

[KIT 90] Korea Institute of Technology, "A Study of the Development of Direct Drive Robots and Their Application", Report No. TR-01-90, Korea, 1990.

[Lackey 86] John D. Lackey, "Adaptive Position Control of Robot Manipulators", Conf. on Applied Motion Control, pp63-69, 1986.

[Landau 76] L. D. Landau, E. M. Lifshitz, "Mechanics", 3rd ed., Pergamon Press, 1976.

[Lane 84] J. S. Lane, S. L. Dickerson, "Contribution of Passive Damping to the Control of Flexible Manipulators", Proceedings of the 1984 Int'l Computers in Engineering Conf. and Exhibit, Vol. 1,pp175-180, Las vegas, Aug. 1984.

[Lee 83] C.S.G. Lee, R.C. Gonzalez, K.S. Fu, "Tutorial on Robotics", IEEE Computer Society, pp93-102,1983.

[Ljung 78] L. Ljung, Y. D. Landau, "Model Reference Adaptive Systems and Self-Tuning Regulators-Some Connections", 7th IFAC Congress, Helsinki, 1978.

[Ljung 87] L. Ljung, "System Identification : Theory For The User", Prentice-Hall, 1987.

[Luh 80a] J. Y. S. Luh, M. W. Walker, R. P. Paul, "On-Line Computational Scheme for Mechanical Manipulator", Trans. of ASME, J. of Dyn. Sys., Meas. and Control, Vol.102, pp69-76, June 1980.

[Luh 80b] J. Y. S. Luh, M. W. Walker, R. P. Paul. "Resolved-Acceleration Control of Mechanical Manipulators", IEEE Trans. Automatic Control, Vol. AC-25, No. 3, pp468-474, 1980.

[MATLAB 87] The Mathematics Inc., "PC-MATLAB V. 3.10", 1987.

[Meirovitch 70] Leonard Meirovitch, "Methods of Analytical Dynamics", McGraw-Hill, 1970.

[Middleton 88] R. H. Middleton, D. C. Goodwin, "Adaptive Computed Torque Control for Rigid Link Manipulations", System & Control Letters, Vol. 10, pp9-16, 1988.

[Mukerjee 85] A. Mukerjee, D. H. Ballard, "Self-Calibration in Robot Manipulators", Proceedings of IEEE Conf. on Robotics and Automation, pp1050-1057, St. Louis, 1985.

[NAG 88] The Numerical Algorithms Group Ltd, "The NAG Fortran Library Manual-MARK13", 1988.

[Norton 86] J. P. Norton, "An Introduction to Identification", Academic Press, 1986.

[Orin 79] D. E. Orin, R. B. McGhee, M. Vukobratovic, G. Hartoch, "Kinematic and Kinetic Analysis of Open-chain Linkages Utilizing Newton-Euler Methods", Mathematical Biosciences 43, pp107-130, Feb. 1979.

[Ortega 88] R. Ortega, M. W. Spong, "Adaptive Motion Control of Rigid Robots: A Tutorials", Proceedings of the 27th IEEE Conf. on Decision and Control, Austin,Texas,Dec. 1988.

[Owen 85] T. Owen, "Assembly With Robots", New Technology Modular Series, Kogan Page, London, 1985.

[Parks 87] T. W. Parks, C. S. Burrus, "Digital Filter Design", John Wiley & Sons Inc., 1987.

[Paul 82] R. Paul, "Robot Manipulators: Mathematics, Programming, and Control", MIT Press, 1982.

[Rangan 82] Kasturi V. Rangan, "Position and Velocity Measurement by Optical Shaft Encoders", Carnegie-Mellon University, CMU-R1-TR-82-8, 1982.

[RDPT 89] RDP Technology Ltd.,"RDP 1990 Series Hardware Reference Manual', May 1989, U.K.

[Silver 82] W. M. Silver, "On the Equivalence of Lagrangian and Newton-Euler Dynamics for Manipulators", Int'l J. of Robotics Research, Vol. 1, No.2, pp60-70,1982.

[Spong 84] M. W. Spong, J. S. Thorpe, J. M. Kleinmaks, "The Control of Robot Manipulator With Bounded Input. Part II: Robustness and Disturbance Rejection", Proc. 23rd IEEE Conf. on Decision and Control, pp1047-1052, Las Vegas, Dec. 1984.

[Tourassis 85] V. D. Tourassis, "Dynamic Modelling and Control of Robotic Manipulators", Ph.D. Thesis, Carnegie-Mellon University, 1985.

[Tzafestas 82] S. G. Tzafestas, G. I. Stassinopoulos, M. Farsi, J. W. Finch, K. Warwick, "Decentralized PID Self-Tuning Control of Industrial Robots", Proceedings of 25th IEEE Conf. on Decision and Control, pp1888-1891, Athens, Greece, Dec. 1986.

[Vukobratovic 82] M. Vukobratovic, D. Stokic, "Control of Manipulation Robots, Theory and Application", Springer-Verlag, 1982.

[Vukobratovic 83] M. Vukobratovic, N. Kircanski, "Decoupled Control of Robots Via Asymtotic Regulators", IEEE Trans. on Automatic Control, Vol. AC-28, No. 20, pp978-981, 1983.

[Wahab 85] W. Wahab, P. E. Wellstead, "Pole-Assignment Self-Tuning Control of A Robot Manipulator Arm", Presented at The IEEE Control System Special Interest Meeting at Hull Univ.(U. K.), 6th Nov. 1985.

[Walters 82] R. G. Walters and M.M. Bayoumi, "Application of A Self-tuning Pole-Placement Regulator to an Industrial Manipulator", IEEE Conf., pp323-329, 1982.

[Wampler 88] C. W. Wampler, II., L. J. Leifer, "Applications of Damped Least Square Methods to Resolved-Rate and Resolved-Acceleration Control of Manipulators", Trans. of ASME, J. of Dyn., Sys., Mea., and Control, Vol. 110, pp31-38, Mar. 1988.

[Waters 79] R. C. Waters, "Mechanical Arm Control", Artificial Intelligence Lab., MIT, AIM 549, Oct. 1979.

[Youcef-Toumi 87] K. Youcef-Toumi, A. T. Y. Kuo, "High Speed Trajectory Control of A Direct Drive Robot", Proc. of 26th Conf. on Decision and Control, pp2202-2209, LA. Dec. 1987.

[Young 72] David M. Young, R. T. Gregory, "A Survey of Numerical Mathematics", Vol.1, Addison-Wesley, 1972.

APPENDIX A

VELOCITY INDEPENDENT DISCRETE DYNAMIC MODEL

$$G_1 = \frac{11}{6(\Delta t)^2}\begin{bmatrix} p_1 & p_2 C_{2\bar{1}}(k+1) \\ p_2 C_{2\bar{1}}(k+1) & p_3 \end{bmatrix}$$

$$G_2 = \frac{1}{6(\Delta t)^2}\begin{bmatrix} -29p_1 & -p_2\{18C_{2\bar{1}}(k+1) + 11C_{2\bar{1}}(k)\} \\ -p_2\{18C_{2\bar{1}}(k+1) + 11C_{2\bar{1}}(k)\} & -29p_3 \end{bmatrix}$$

$$G_3 = \frac{3}{2(\Delta t)^2}\begin{bmatrix} 3p_1 & p_2\{C_{2\bar{1}}(k+1) + 2C_{2\bar{1}}(k)\} \\ p_2\{C_{2\bar{1}}(k+1) + 2C_{2\bar{1}}(k)\} & 3p_3 \end{bmatrix}$$

$$G_4 = \frac{1}{6(\Delta t)^2}\begin{bmatrix} -11p_1 & -p_2\{2C_{2\bar{1}}(k+1) + 9C_{2\bar{1}}(k)\} \\ -p_2\{2C_{2\bar{1}}(k+1) + 9C_{2\bar{1}}(k)\} & -11p_3 \end{bmatrix}$$

$$G_5 = -\frac{1}{3(\Delta t)^2}\begin{bmatrix} p_1 & p_2 C_{2\bar{1}}(k) \\ p_2 C_{2\bar{1}}(k) & p_3 \end{bmatrix}$$

$$C_p = -\frac{p_2\{C_{2\bar{1}}(k+1) - C_{2\bar{1}}(k)\}}{72(\Delta t)^2\{[A_1(k+1) - A_1(k)] - [A_2(k+1) - A_2(k)]\}}\begin{bmatrix} \lambda \\ -\lambda \end{bmatrix}$$

$$\begin{aligned}\lambda = {} & 121\,[A_1(k+1)A_2(k) + A_1(k)A_2(k+1)] \\ & - 198[A_1(k+1)A_2(k-1) + 2A_1(k)A_2(k) + A_1(k-1)A_2(k+1)] \\ & + 99[A_1(k+1)A_2(k-2) + A_1(k-1)A_2(k) + A_1(k)A_2(k-1)\end{aligned}$$

$$
\begin{aligned}
&\quad + A_1(k-2)A_2(k+1)] \\
&- 22[A_1(k+1)A_2(k-3) + A_1(k-2)A_2(k) + A_1(k)A_2(k-2) \\
&\quad + A_1(k-3)A_2(k+1)] \\
&+ 324[A_1(k)A_2(k-1) + A_1(k-1)A_2(k)] \\
&- 162[A_1(k)A_2(k-2) + A_1(k-1)A_2(k-1) + A_1(k-1)A_2(k-1) \\
&\quad + A_1(k-2)A_2(k)] \\
&+ 81[A_1(k-1)A_2(k-2) + A_1(k-2)A_2(k-1)] \\
&- 18[A_1(k-1)A_2(k-3) + 2A_1(k-2)A_2(k-2) + A_1(k-3)A_2(k-1)] \\
&+ 4[A_1(k-2)A_2(k-3) + A_1(k-3)A_2(k-2)]
\end{aligned}
$$

$$
\mathbf{D}_p = \frac{1}{12\Delta t}\begin{bmatrix} p_4\{11A_1(k+1) - 7A_1(k) - 9A_1(k-1) + 7A_1(k-2) - 2A_1(k-3) \\ \\ p_5\{11A_2(k+1) - 7A_2(k) - 9A_2(k-1) + 7A_2(k-2) - 2A_2(k-3) \end{bmatrix}
$$

APPENDIX B

OFFSET ANGLE(*RotOff*) AND SINE TABLE

Offset Angle

The average offset angle between the raw resolver reading(RRR) and field angle can be found as belows:

- Choose a small field angle A_f(<10d) and apply the phase voltage according to the Eq.(5-1).
- Subtract R[90° dfa] (resolver reading value which corresponds to 90 degrees field angle) from RRR (since the field equilibrium position is 90 degrees field angle ahead of A_f).
- Subtract R[360° dfa] (resolver reading values which corresponds to 360 degrees field angle) from RRR until the remainder is less than R[360° dfa].
- take average of the N field equilibrium position.

This result is the offset (*RotOff*).

Sine Table

Sine table is used for generating the demand phase voltage for each winding of motor. The input to sine table is the field angles. The table only needs to cover one quarter of the 360 degrees range due to the symmetric nature of sine function, a table of 800H entries(11-bit resolution, from 0 to 7FFH) is selected.

The algorithm to use the sine table is,

- Right shift the 16-bits input angle to get a 11-bit number.
- If the number is bigger than half(=3FFH=180 degrees) then
  number=number-half
  sign=-1
- if the number is bigger than quarter(1FFH=90 degrees) then
  number=quarter-number
- Get the number and times the sign.