# QUANTITATIVE MODEL VALIDATION OF MANIPULATIVE ROBOT SYSTEMS

BY

IMAN HERWIDIANA KARTOWISASTRO

A THESIS SUBMITTED FOR THE

DEGREE OF DOCTOR OF PHILOSOPHY

IN CONTROL ENGINEERING

CONTROL ENGINEERING CENTRE

DEPARTMENT OF ELECTRICAL, ELECTRONIC AND INFORMATION ENGINEERING

CITY UNIVERSITY, LONDON

FEBRUARY, 1991

CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## ACKNOWLEDGEMENTS

Foremost, I would like to express my thanks to my parents Mr. and Mrs. Hermawan Kartowisastro, and my uncle and aunt Mr. and Mrs. Herudi Kartowisastro for their constant moral support. From a technical point of view, I would like to express my thanks to my research supervisor, Professor P.D. Roberts for his sincere advice, guidance and painstaking corrections through out this work. I am also indebted to Professor M.H. Butterfield from the United Kingdom Atomic Energy Authority for the discussions about his method.

I also like to thank Dr. J.E. Ellis and Dr. G. Pooler for their help in computer programming, Dr. J. Lin for the discussions regarding the subject matter and Mr. J. Osman for the discussions and comments about this thesis.

Miss Z. Amini, Mrs. J. Rivellini and other research colleagues and staff in the Control Engineering Centre and the Information Engineering Centre also deserve thanks for their helps and supports.

Finally, I would like to thank the Agency For the Assessment And Application of Technology (B.P.P. Teknologi) in Indonesia for supporting this research through its Overseas Fellowship Program (O.F.P.).

## DECLARATION

The author grants power of discretion to the University Librarian to allow this thesis to be copied in whole or part without further reference to the author. This permission covers for study purposes only.

## ABSTRACT

This thesis is concerned with applying the distortion quantitative validation technique to a robot manipulative system with revolute joints. Using the distortion technique to validate a model quantitatively, the model parameter uncertainties are taken into account in assessing the faithfulness of the model and this approach is relatively more objective than the commonly visual comparison method. The industrial robot is represented by the TQ MA2000 robot arm. Details of the mathematical derivation of the distortion technique are given which explains the required distortion of the constant parameters within the model and the assessment of model adequacy.

Due to the complexity of a robot model, only the first three degrees of freedom are considered where all links are assumed rigid. The modelling involves the Newton-Euler approach to obtain the dynamics model, and the Denavit-Hartenberg convention is used through out the work. The conventional feedback control system is used in developing the model. The system behaviour to parameter changes is investigated as some parameters are redundant. This work is important so that the most important parameters to be distorted can be selected and this leads to a new term called the fundamental parameters.

The transfer function approach has been chosen to validate an industrial robot quantitatively against the measured data due to its practicality. Initially, the assessment of the model fidelity criterion indicated that the model was not capable of explaining the transient record in term of the model parameter uncertainties. Further investigations led to significant improvements of the model and better understanding of the model properties. After several improvements in the model, the fidelity criterion obtained was almost satisfied. Although the fidelity criterion is slighltly less than unity, it has been shown that the distortion technique can be applied in a robot manipulative system.

Using the validated model, the importance of friction terms in the model was highlighted with the aid of the partition control technique. It was also shown that the conventional feedback control scheme was insufficient for a robot manipulative system due to high nonlinearity which was inherent in the robot manipulator.

# NOMENCLATURE

Bold letters represent vectors or matrices

## Abbreviations

D-H : Denavit- Hartenberg

N-E : Newton-Euler

## Lower case

$a_i$ : D-H parameter, the distance along the common normal between the joint i axis and the joint i+1 axis.

$\mathbf{a}$ : constant parameter vector.

$\hat{\mathbf{a}}$ : optimised constant parameter vector.

$\bar{\mathbf{a}}_i$ : linear acceleration of the centre of mass of link i with respect to the base coordinate system and expressed in the base coordinate system.

$c_{ij}$ : steady state gain of the associate system.

$c_f$ : filter gain.

$\bar{\mathbf{c}}_i$ : a position vector of the centre of mass of link i from the origin of the coordinate system i-1 and expressed in the base coordinate system.

${}^i\bar{\mathbf{c}}_i$ : a position vector of the centre of mass of link i from the origin of the coordinate system i-1 and expressed in the coordinate system i.

$d_i$ : D-H parameter, the distance between the intersections of the joint i axis with the common normals $a_{i-1}$ and $a_i$.

dx : small translation along the $x'$ axis.

dy : small translation along the $y'$ axis.

$dz$ : small translation along the $z'$ axis.

$\mathbf{dT}$ : differential change of a homogenous transformation matrix.

$e_{min_i}$ : optimised residual error of the i-th output.

$e_i'$ : position error of joint i.

$\mathbf{e}$ : residual error vector.

$\mathbf{e}_{min}$ : optimised residual error vector of the overall system.

$\mathbf{e}'$ : position error vector.

$\mathbf{f}_i$ : force exerted at joint i to suport link i and the distal links.

$\mathbf{f}$ : linear or nonlinear function vector.

$g_{ij}$ : transfer function of the i-th associate system output driven by the j-th parameter variation input.

$h_{i1}$, $h_{i2}$ and $h_{i3}$ : coefficients of the cubic trajectory.

$i$ : joint number.

$m_i$ : mass of link i.

$n$ : number of degrees of freedom.

$n_i$ : gear ratio of joint i.

$\mathbf{n}_i$ : torque exerted at joint i to suport link i and the distal links.

${}^i\mathbf{nn}_i$ : torque at joint i due to the motion of link i alone and expressed in the coordinate system i.

$\mathbf{p}_i$ : translation vector of the origin of the coordinate system i from the origin of the coordinate system i-1 and expressed in the base coordinate system.

$\bar{\mathbf{s}}_i$ : position vector of the centre of mass of link i from the origin of the coordinate system i and expressed in the base coordinate system.

$q_i$ : joint i variable.

$\dot{q}_i$ : first derivative of joint i variable.

$\ddot{q}_i$ : second derivative of joint i variable.

$t_o$ : initial time.

$t_f$ : final time.

$u_i$ : input vector of the model of joint i with respect to link side.

$u$ : input vector of the overall system.

$v$ : input voltage vector applied to the amplifier input.

$v_i$ : linear velocity of the origin of the coordinate system i with respect to the base coordinate system and expressed in the base coordinate system.

$\dot{v}_i$ : linear acceleration of the origin of the coordinate system i with respect to the base coordinate system and expressed in the base coordinate system.

$x_i'$, $y_i'$, $z_i'$ : i-th coordiante system.

$x_i$ : state vector of the model of joint i with respect to actuator side.

$x$ : state vector of the overall system.

$\hat{x}$ : optimised state vector of the overall system.

$\tilde{x}_i$ : state vector of the model of joint i with respect to link side.

$y$ : model output vector.

$z$ : measured output vector.

Upper case

$A_i$ : length of link i of the TQ MA2000 robot model.

$A'_i$ : linear power amplifier gain of joint i.

$A$ : system matrix of the overall actuating system.

$A_i$ : system matrix of the i-th actuator.

$\tilde{A}_i$ : system matrix of the closed loop control of joint i.

$\bar{A}^{i}_{i-1}$ : D-H homogenous transformation matrix that transforms the coordinate system i to the coordinate system i-1.

B : input matrix of the overall acatuating system.

$B_i$ : input vector of the i-th actuator.

$\tilde{B}_i$ : input matrix of the closed loop control of joint i.

Ci : cos ($\theta_1$)

C2 : cos ($\theta_2$)

C3 : cos ($\theta_3$)

C23 : cos ($\theta_2+\theta_3$)

C : load matrix of the overall actuating system.

$C_i$ : load vector of the i-th actuator.

$\tilde{C}_i$ : disturbance vector of the closed loop control of joint i.

$C'_i$ : Coulomb friction at joint i.

C' : Coulomb friction vector.

D : inertial acceleration related symmetric matrix.

$\hat{D}$ : estimated inertial acceleration related symmetric matrix.

$E_i$ : back EMF voltage of the i-th actuator.

F : complete friction vector.

F' : filter transfer function.

$F_i$ : net force acting at the centre of mass of link i and expressed in the base coordinate system.

G : gravity torque vector.

$\hat{G}$ : estimated gravity torque vector.

G' : transfer function matrix of the overall associate system.

H : coriolis/ centrifugal torque vector.

$\hat{H}$ : estimated coriolis/ centrifugal torque vector.

$I_i$ : inertia tensor matrix of link i about its centre of mass and

expressed in the base coordinate system.

$^{i}\bar{I}_{i}^{*}$ : inertia matrix of link i about joint i and expressed in the coordinate system i.

$\tilde{I}_i$ : armature current of the i-th actuator.

$I_n$ : identity matrix of dimension nxn.

$J_{mi}$ : rotor inertia of the i-th actuator.

$J$ : actuator inertia matrix.

$J_a$ : Jacobian matrix of the system with respect to the differential parameter change.

$J_x$ : Jacobian matrix of the system with respect to the differential state change.

$K$ : total kinetic energy of the system.

$K_{Ti}$ : torque constant of the i-th actuator.

$K_{bi}$ : back EMF constant of the i-th actuator.

$K_{Pi}$ : proportional gain of joint i.

$K_{Ii}$ : integral gain of joint i.

$K_{Di}$ : derivative gain of joint i.

$K_{\theta i}$ : optical encoder/ counter assembly gain of joint i.

$K$ : integrated amplifier and actuator parameter matrix.

$K_P$ : proportional gain matrix.

$K_I$ : integral gain matrix.

$K_D$ : derivative gain matrix.

$L$ : Lagrangian function.

$L_i$ : rotor inductance of the i-th actuator.

$M_{ii}$ : inertial related acceleration value of joint i where actuator and gear ratio are taken into account.

$\hat{M}_{ii}$ : estimated inertial related acceleration value of joint i where

actuator and gear ratio are taken into account.

$MM_3$ : total mass of link 3.

MLOAD : mass at the robot tip.

MSQE : mean squared error value.

$M^*$ : inertial related acceleration matrix where amplifier, actuator and gear ratio are taken into account.

$\hat{M}^*$ : estimated inertial related acceleration matrix where amplifier, actuator and gear ratio are taken into account.

N : gear ratio matrix.

$N_i$ : net torque acting at the centre of mass of link i and expressed in the base coordinate system.

P : disturbance torque vector with respect to link side.

P : total potential energy of the system.

$Q_i$ : partial derivative matrix with respect to joint i variable.

R : rotation matrix.

$R^i_{i-1}$ : rotation matrix that rotates the coordinate system i to the coordinate system i-1.

$R'_1$ : radius of link 1 of the TQ MA2000 robot model.

$R_i$ : terminal resistance of the i-th actuator.

$S_i$ : disturbance at joint i where actuator and gear ratio are taken into account.

$\hat{S}_i$ : estimated disturbance at joint i where actuator and gear ratio are taken into account.

S1 : $\sin(\theta_1)$

S2 : $\sin(\theta_2)$

S3 : $\sin(\theta_3)$

S23 : $\sin(\theta_2+\theta_3)$

$S_{3x}$ : centre of mass position of link 3 from the origin of the coordinate system 3.

$S^{*}$ : disturbance vector where amplifier, actuator and gear ratio are taken into account.

$\hat{S}^{*}$ : estimated disturbance vector where amplifier, actuator and gear ratio are taken into account.

T : homogenous transformation matrix.

T : observation time.

$U_i$ : input voltage of the i-th actuator.

V : proportionally back EMF matrix.

W : weighting matrix.

X : distorted state vector.

$X^{o}$ : matched distorted state vector.

Y : distorted output vector of the model.

$Y^{o}$ : matched distorted output vector of the model.

Z : transfer matrix relating the joint angular acceleration with respect to actuator side with the first derivative of the state vector.

Greek symbols

$\alpha_i$ : D-H parameter, the angle between the joint i axis and the joint i+1 axis.

$\alpha(t)$ : time dependent parameter variation vector for distorting the model.

$\alpha^{o}(t)$ : time dependent parameter variation vector to obtain a perfect model-plant match.

$\theta_i$ : D-H parameter, the angle between $a_{i-1}$ and $a_i$.

$\theta$ : joint variable vector with respect to link side.

$\theta_d$ : desired angular displacement vector with respect to link side.

$\phi$ : rotation about the x' axis.

$\beta$ : rotation about the y' axis.

$\gamma$ : rotation about the z' axis.

$\boldsymbol{\phi}$ : model state variable difference vector.

$\boldsymbol{\phi}^o$ : model state variable difference vector in the perfect model-plant match condition.

$\boldsymbol{\psi}$ : difference between undistorted and distorted model output vector.

$\boldsymbol{\psi}^o$ : difference between undistorted and distorted model output vector in the perfect model-plant match condition.

$\Delta$ : differential operator of a homogenous transformation matrix.

$\omega_f$ : filter cut-off frequency.

$\Phi$ : white noise intensity.

$\zeta_{ij}$ : damping ratio of the associate system.

$\lambda^2$ : fidelity criterion of the model.

$\delta_x$ : small rotation of about the x' axis.

$\delta_y$ : small rotation of about the y' axis.

$\delta_z$ : small rotation of about the z' axis.

$\tau_i$ : output torque of the i-th actuator.

$\tau_{l_i}$ : generalized torque to drive link i with respect to link side.

$\tau_{m_i}$ : generalized torque of joint i with respect to actuator side.

$\boldsymbol{\tau}_l$ : generalized torque vector with respect to link side.

$\tau'_j$ : expected standard deviation of the j-th parameter.

$\sigma_{\alpha_j}$ : standard deviation of the j-th parameter.

$\upsilon_{mi}$ : total viscous friction of joint i with respect to actuator side.

$\tilde{\upsilon}_{mi}$ : viscous damping constant of actuator i.

$\tilde{\upsilon}_{ji}$ : viscous damping constant of joint i.

$\tilde{\upsilon}_{effi}$ : effective total viscous damping constant of joint i with respect to

actuator side.

$\upsilon$ : viscous friction vector with respect to link side.

$\mu$ : coefficient of Coulomb friction.

$\tilde{\mu}$ : effective coefficient of Coulomb friction.

# CHAPTER 1

# INTRODUCTION

## 1.1 Simulation

Simulation is the use of one system to imitate the behaviour of another system. A system itself is a collection of interacting elements which act together to achieve a desired goal. Systems can be studied by building prototypes or by building mathematical models. The former is undesirable due to some reasons, eg. expensive and risky. The purpose of mathematical models in a simulation is to aid the analysis, design and prediction without actually building and operating the real plant which in turn reduces the risks and costs. It also allows a study of the system sensitivity to perturbations in its parameters. On the other hand, if a proper design of experiments is not carried out, exercising a computer model can be very expensive in terms of computing time and man labour.

Simulation has been formally defined as "the process of designing a computerized model of a system (or process) and conducting experiments with this model for the purpose either of understanding the behaviour of the system or of evaluating various strategies for the operation of the system" [Shannon, 1975]. Nowadays, almost all simulations are performed on digital computers rather than on analogue computers.

Once a model has been obtained, it is important to check the validity of the model. Validation is a process of confirming that the model is adequate for representing the real plant and is capable of imitating its behaviour reasonably within its intended purposes. Hence, the purpose of validation is to scrutinise the model with

respect to its inadequacies. Without validation, a model is of very little use.

## 1.2 Industrial Robot

Due to an increasing demand both in quantity and quality of products, the trend in industry goes towards computer based automation such as the use of robot manipulators. An industrial robot has a high flexibility in performing some applications. Because of this reason, robot manipulators are more preferable than special purpose machines which can only perform predetermined functions with limited reprogrammable capabilities.

**Definition of an industrial robot [Scott and Husband, 1985] :**

**1. British Robot Association**

An industrial robot is a reprogrammable device designed to both manipulate and transport parts, tools or specialized manufacturing implements through variable programmed motion for the performance of specific manufacturing tasks.

**2. Robot Institute of America**

A robot is reprogrammable multifunctional manipulator designed to move material, parts, tools or specialized devices, through variable programmed motions for the performance of a variety of tasks.

In short, an industrial robot arm is a general purpose manipulator which has

several links connected commonly by revolute or prismatic joints or a combination of these two types of joints. It must possess intelligence with the aids of internal and external sensors and exhibit flexibility that can be achieved by reprogramming its movements through its controller.

The first mechanical manipulator was built at Argonne National Laboratory in 1947 [Lee, Gonzalez, Fu, 1986]. Nowadays, more advanced industrial robots are controlled by micro or mini computers with better internal and external sensors. With the advances of artificial intelligence, fast growing of microprocessor developments and advanced control system, industrial robots can perform from simple jobs such as pick and place operations to complicated tasks such as doing a surgery. Some companies in U.K. eg. Colne Robtics Co. Ltd., Powertran Cybernatics Ltd, UMI Group and Tequipment Ltd. offer robots for educational and research purposes.

**Advantages of using robots :**

- In industrialised countries where labour costs are high, production costs can be reduced by the use of robots.
- Robots work relatively faster than human beings, so productivities can be increased.
- Robots have a high degree of accuracy and repeatability which yield better quality products.
- Robots can replace human workers in hazardous environments.
- The use of robots can improve management control.

**Disadvantages of using robots :**

- In some countries the use of robots can be more expensive.
- Unemployment problems.
- The tasks that can be performed by robots are still relatively limited.

Current research interests are modelling, control, sensors, robot vision, robot languages, machine intelligence and system architecture.

### 1.3 General Description of Thesis

Numerous research in validating mathematical models quantitatively using the distortion technique were performed in the field of nuclear power plant [Butterfield, Sutton, 1979; Butterfield, 1981; Butterfield and Thomas, 1983; Butterfield and Thomas, 1986; Butterfield, 1989; Harrison and McCabe, 1989; Li, 1988, 1989] and in a linear system [Cameron, 1989]. Using this quantitative validation technique, the parameter uncertainties of the model are taken into account. To date however, no application of the distortion technique has been performed in the field of an industrial robot manipulator system. Only a few authors carried out works in robot model validation, eg. Gawthrop, Mirab and Li [1989]. They performed a robot model validation work where the frequency response of the model was compared to the frequency response of the real system. The parameter uncertainties of the model, however, were not taken into account in validating the model. Thus, it was based on visual comparison. This thesis presents a research work which was carried out in applying the distortion quantitative validation technique to a robot manipulative system with revolute joints. With the use of the distortion technique in validating a robot model quantitatively, where the model parameter uncertainties are taken into account, the result is relatively more objective than the visual comparison method used commonly by other authors.

The TQ MA2000 robot arm from Tequipment Ltd. was used in this work.

Chapter 2 gives the common procedures in a simulation study and in building mathematical models. The importance of parameter estimation and validation are also mentioned.

Chapter 3 is concerned with the mathematical model development of an industrial robot arm. It discusses the robot arm kinematics, the derivation of the dynamical model, the design of trajectory in which the arm follows to accomplish the tasks and the control system to govern the arm along a desired trajectory.

Chapter 4 discusses the mathematical development of the distortion technique to validate a model quantitatively. Mathematical analysis for both the time and frequency domain approaches are given in detail. An extension for a multiple measured variable case in the frequency domain approach is also given.

Chapter 5 describes the implementation of the distortion technique given in chapter 4 to validate a robot model quantitatively. Some modifications are needed in the computation of robot dynamics. Sensitivity analysis of the inertial parameters are discussed and a new term of the fundamental parameters is introduced.

A full description of the TQ MA2000 robot arm is given in chapter 6. Both hardware and software of this robot are explained in detail. The experiment carried out prior to exercising the distortion validation method is discussed and some experimental results are presented.

The application of the quantitative model validation technique described in chapter 4 to a robot manipulator system is given in chapter 7. Important model parameters are identified in a closed loop manner and the frequency domain method is then used to validate the TQ MA2000 robot arm model against measured data obtained from experiment. Validation results and discussions of the model are given.

Chapter 8 presents an application of a validated mathematical model in control system design. A partitioned control technique is used in studying the superiority/ inferiority of the models. The performance of a classical PID control system is also discussed.

Finally, chapter 9 gives the overall conclusions of this work and suggestions for future work.

The flowchart as shown in figure (1.1) gives a clear description of the organization of this thesis.

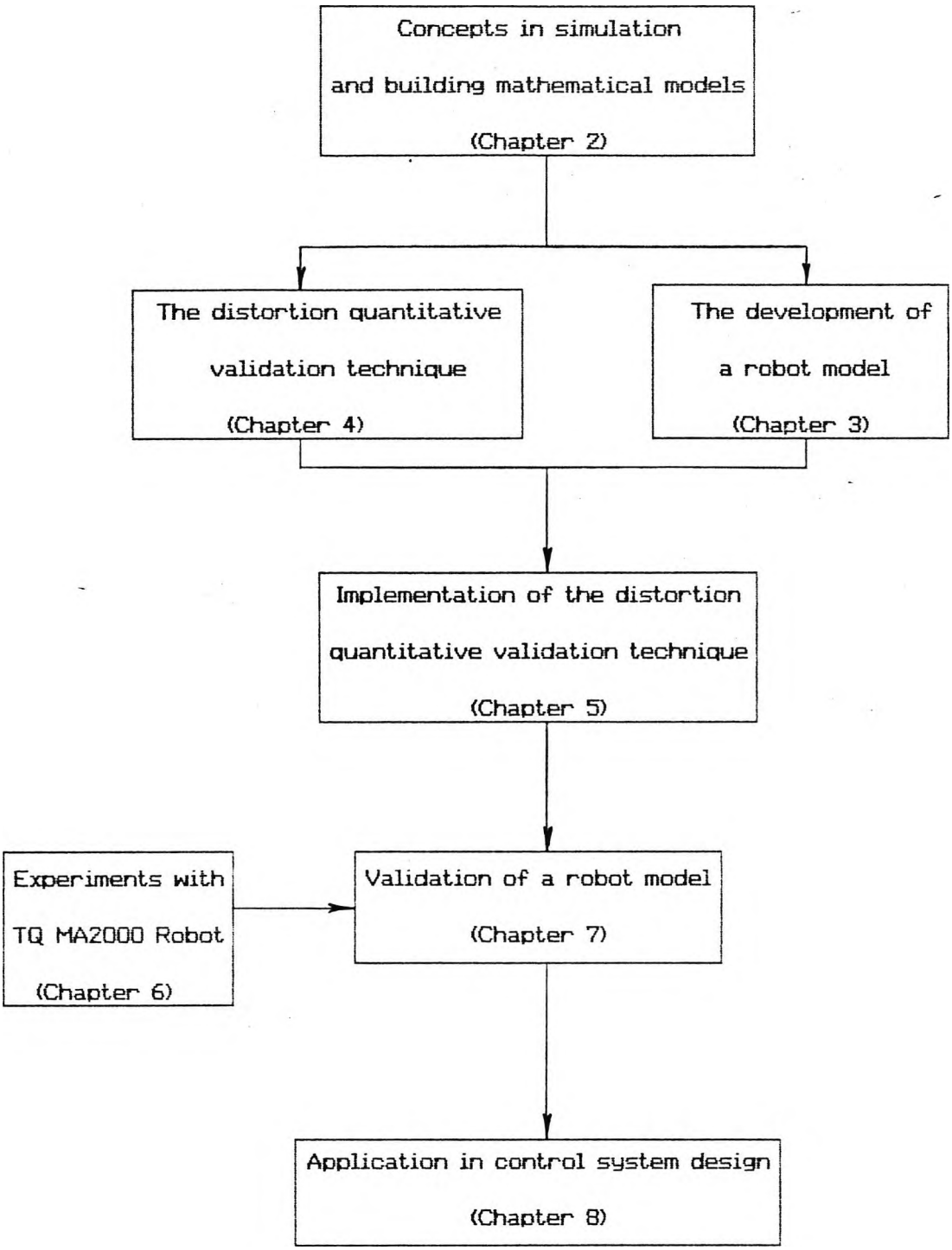



Figure 1.1

Flowchart of thesis

# CHAPTER 2

# MATHEMATICAL MODELLING

## 2.1 Introduction

Having introduced a general concept of simulation in chapter 1, a procedure to perform simulation tasks from developing a formulation of the mathematical model to using a validated model in some applications is given in this chapter. Important parts in a simulation - model building, parameter estimation and validation - are discussed in detail. Computer modelling and simulation has been introduced in university curricula since 1960s [Neelamkavil, 1987].

Mathematical modelling is an attempt to describe the behaviour of the real system in mathematical terms. A problem defined by a mathematical model may have a feasible solution, optimum solution, satisfactory solution or even no solution at all and the studies of computer modelling and simulation lean towards finding satisfactory solutions [Neelamkavil, 1987]. The main advantage of mathematical models is to provide a useful tool in learning the principles of operation, capabilities and limitations [Finkelstein and Watts, 1978]. For example, by having a good mathematical model, a company can work efficiently and has a capability of making products economically.

Since a mathematical model is an important part of a simulation work, the physical properties of the system which is to be modelled must be studied carefully. In order to have a useful mathematical model as well as confidence in using it, a mathematical model must be validated.

## 2.2 Principal Steps in a Simulation Study

1. Develop a formulation of the mathematical model of the system to be studied to the complexity it is intended for or to the limits of the computer capability in terms of the computing time and the software which are available. All simplifying assumptions used in developing the mathematical model are defined.

2. Create a computer program of the mathematical model for running on the digital computer system. This program in turn must be verified and if necessary some refinement or modification must be made.

3. Perform the parameter estimation of the model in such away so that the error between the plant recorded response and the model response is minimum.

4. Validate the model under a condition where it was intended to represent. Go back to step 1 if an improvement in the model is required.

Once the model has been validated, a simulation study such as safety analysis can be performed. Figure (2.1) shows the sequence in a simulation study.

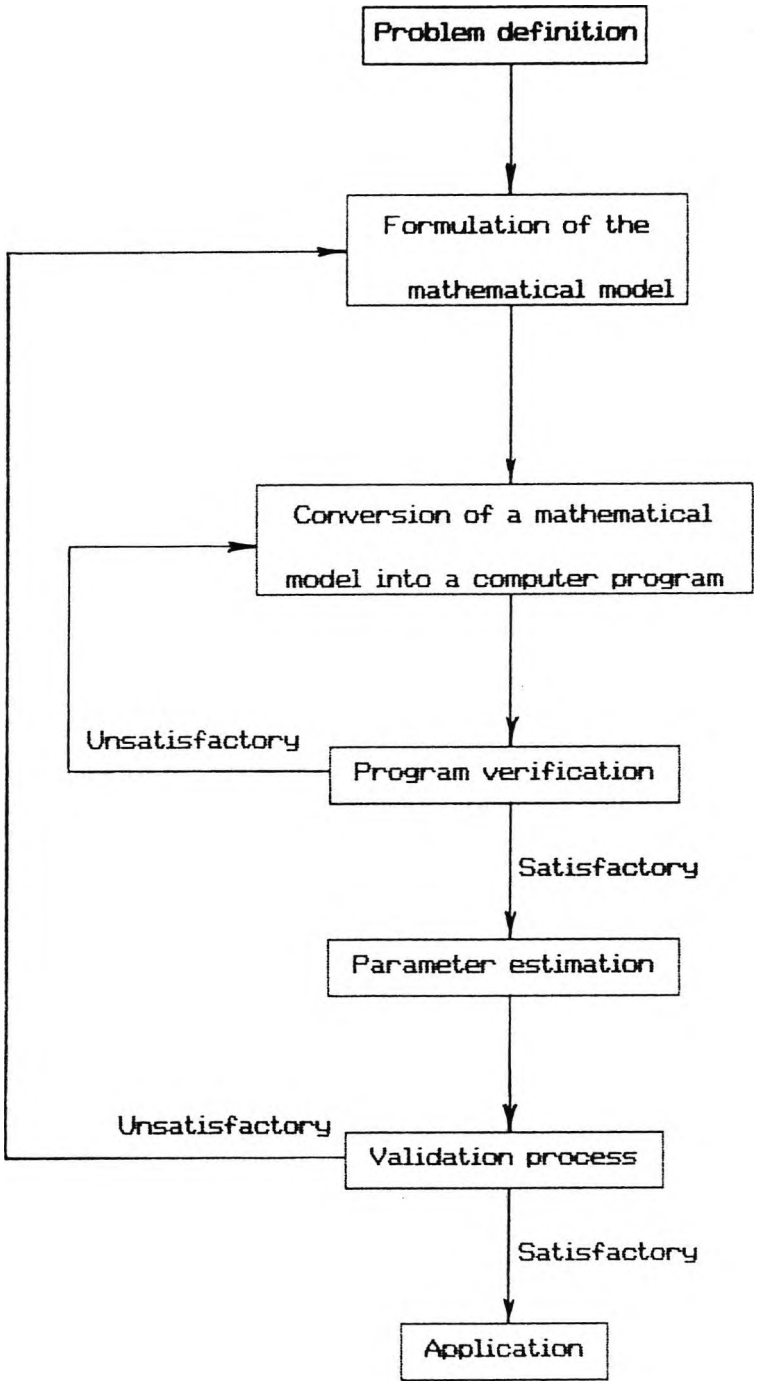



Figure 2.1

Procedures in a simulation study

### 2.2.1 Mathematical Model

A mathematical model is the heart of any simulation. By understanding thoroughly the physical properties of the system under investigation and putting physical principles as seemed relevant, a good mathematical model can be developed. Due to the complexity of the nature of the real system, some assumptions are usually made to simplify the model. Since this work involves the exercise of this model using a digital computer, the nature of the results depends on the simplifications which have been made.

Because of the nature of the problems, simulation studies are often carried out on dynamic systems where the time history of the system's behaviour is important. This in turn will involve the use of state equations in the model.

### 2.2.2 Computer Program

Mathematical models must be converted into computer code first before any simulation can be exercised. Simulations can be carried out using a special simulation language. The most important feature of this language is that its structures allow process modelling to be done directly, thus mathematical equation writing is greatly reduced. Another important feature is the run time command which allow the user to change parameter value(s) in the middle of a simulation run. There is also another type of simulation language which is based on the block diagram. These block structures provide the process modelling more direct to the user.

Ordinary languages such as FORTRAN can also be used in modelling since there is

a huge number of numerical libararies available. The biggest disadvantages of using this ordinary language are that it does not have the run time commannd facilities and it is difficult to debug the program. In both types of languages, the availability of ordinary differential equation algorithms plays an important role in a dynamic simulation.

### 2.2.3 Verification

Verification is concerned with the correctness, and consistency of the mathematical model implementation in the computer program, so that the mathematical model behaves the way an experimenter intends. This is because, although the programmer knows what the program is intended to do, the program may well do something else. Thus, ensuring the correctness of the program as a mathematical model to confirm that the model is a faithful representation of what was intended is important. However, it should be kept in mind that a verified simulation program does not guarantee the validity of the model.

### 2.2.4 Parameter Estimation

Optimisation has an important role in modelling, because the model which is developed has a set of independent parameters to determine so that it produces a good fit against the behaviour of the real system. This is because the mathematical model is only an approximation of the real system and a perfect measurement of the real system can never be achieved. The choice of technique to perform optimisation depends on the nature of the problem, eg. the availabilty of the derivatives of the objective function.

### 2.2.5 Validation

Once a model has been obtained, it is important to check the validity of the model. To validate means to prove that the model is an exact replica of the real system [Neelamkavil, 1987] and validation is a process of confirming that the model is adequate for representing the real plant and is capable of imitating its behaviour reasonably within its intended purposes. Hence, the purpose of validation is to scrutinise the model with respect to its inadequacies in comparison to the real system. Validation can be performed qualitatively such as by visual comparison to assess the goodness of fit between model and plant responses. Another method for validation is that the model validity is assessed by a quantitative means such as by quantifying the residual errors between model and plant. Without validation, confidence in performing a simulation work cannot be achieved and so a model is of very little use.

### 2.2.6 Application

After the validation stage has been passed, mathematical models are now ready for some applications. Typical applications include fault diagnostics, environment changes and control system design. Any operation which is dangerous to carry out in the real system can be conveniently performed using a mathematical model.

## 2.3 Model Building

Mathematical modelling is a technique to describe a system in the real world using mathematical terms. Application of mathematical models can be found in the fields of biological, physical and social sciences. It comprises parts taken from calculus,

linear algebra, statistics and various other fields within mathematics.

In a nonmathematical nature, a model is defined as an object or concept to represent something else which is scaled down from reality and converted to a form to make it comprehensible [Meyer, 1984]. For example a plane model prior to building a full scale one falls into this category and is called a physical model. A mathematical model itself is defined as a model which comprises of mathematical terms such as constants, variables, differential equations, etc [Meyer, 1984]. Building a mathematical model from scratch takes time and ingenuity. In most aspects, a mathematical model is superior to a physical model due to the advances in computer technology. From this point, the word model refers to a mathematical model.

The knowledge of a system under study falls into two categories. A priori knowledge is a case where the physical insight of the system is known theoretically such as its model structure. If the knowledge is obtained empirically from experimental data, such a knowledge is called a posteriori. A priori knowledge is usually restricted due to the uncertainty embedded in the system and its environment. This makes an estimation procedure necessary to perform. Both two types of knowledge are often used in combination to obtain good models. Figure 2.2 gives a more clear relation between these two types of knowledge [Eykhoff, 1974].

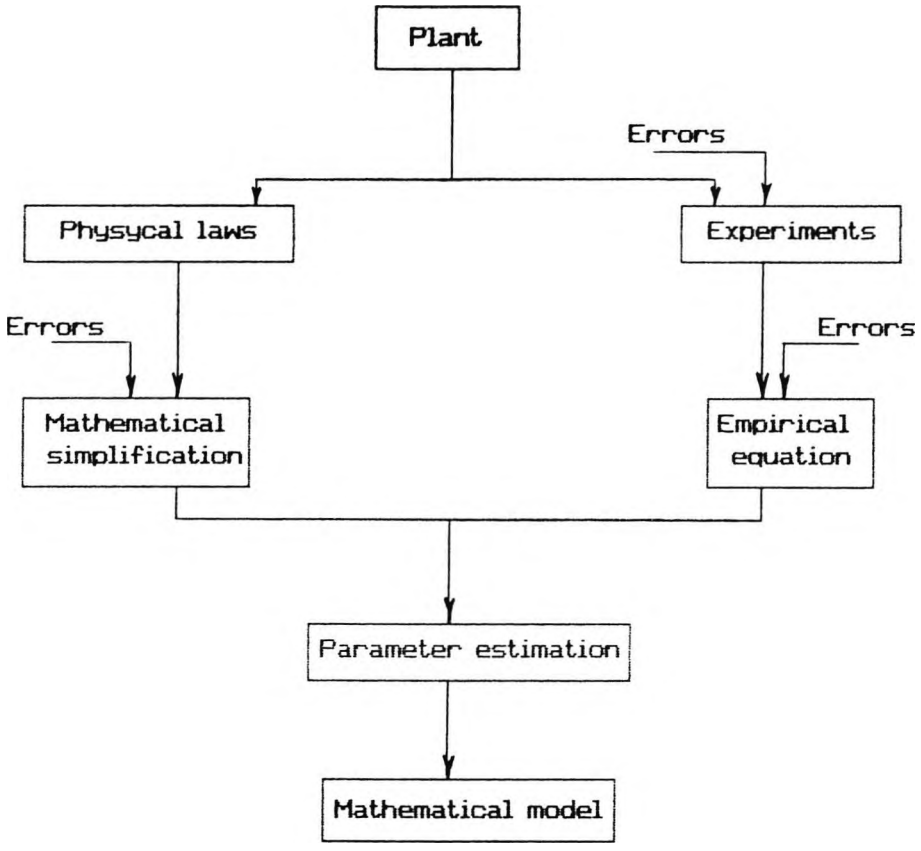



Figure 2.2

Process in modelling.

Roberts [1980] stated the difference between the real system and the model caused by modelling uncertainties as follows :

1. The model structure.

The mathematical model is only an approximation of the real system, so model simplification and lack of a priori knowledge of the real system cause uncertainties in the model.

2. The model parameters.

The parameter values, which are not selected in the parameter estimation

procedure and considered to have constant values, may not be correct and may change with different operating conditions.

3. The real system observations.

The measured output signals from the real system, as a response of the measured input signals, may be in error due to disturbance inputs, instrumentation and data processing error.

## 2.4 Parameter Estimation

It is rare to have a complete a priori knowledge of a model. Hence, a mathematical model is still susceptible to restricted a priori knowledge as well as simplification during the model building. For that reason, all unknown parameters associated with the model must be evaluated in order to complete the model. This procedure is regarded as parameter estimation. Parameter estimation is also defined as the determination of parameter values experimentally that govern the dynamic behaviour where it is assumed that the structure of the process model is known [Eykhoff, 1974]. The so called model reference technique is often used where the same input signal as applied to the plant is given to the model and the parameters of the model are then manipulated to give a minimum performance index.

The most commonly used parameter estimation method is known as the least square technique. This method is based on the principle that the sum of the square values of the differences between the plant response and model response that measures the degree of fit is at minimum. The performance index which is in a quadratic form can then be minimized. Some methods have been developed to include extensions from the original least square estimation technique eg. to take account

of uncertainties in the model structure and disturbances [Roberts, Leal and Georgantzis, 1979].

Another method which has been used in U.K.A.E.A., Winfrith, Dorset, England is based on the concept of the system frequency response [Butterfield and Thomas, 1983]. This method relies on the fact that the frequency response of an open loop system near its critical point is important in determining a closed loop performance. The error between an open loop frequency response of a plant and and open loop frequency response of a model is used as an objective function. This method has difficulties eg. open loop responses are not generally available and plant recorded measurements are more readily in the time domain rather than in the frequency domain.

For more complex models in which the analytical evaluation to find the model optimal parameters is limited, optimisation algorithms based on computer iterations are used. This method is more well known as a hill climbing method. Beginning with an intial guess of a parameter vector, this algorithm calculates the objective function. The algorithm then determines the next paramater vector to evaluate. This procedure is performed iteratively until the objective function converges to a minimum point. Unfortunately, nothing is known about finding a global mimimum or a local minimum and incorrect initial guess may also result a poor convergence. Two standard heuristics are usually used to get better results ie. by finding local minima beginning from widely varying starting values and choosing the best one; by perturbing a local minimum with a finite amplitude away from it and then check whetehер a better point is found. An optimisation procedure which uses a computer iteration algorithm by Nelder and Mead [Press, Flannery, Teukolsky and Vetterling, 1988] with upper and lower bounds is used throughout this work.

## 2.5 Model Validation

Every mathematical model are subjected to some errors due to limited a priori knowledge, measurement noise, quantization noise, simplification during the model building and uncertainties in the system. Prior to using a model in some applications, the model must pass a validation test. This in turn will give the user a full confidence in using the model.

In a general term, validation means conforming that a model is good enough for a specified purpose and capable of imitating the plant behaviour in a certain condition by comparing the model response against the plant response. Since the real system is never perfectly known, so an exact and perfect representation of the real system is never achieved. The result of this is that an absolute validation can only be approached, but never achieved [Neelamkavil, 1987].

A common validation method which is widely used is based upon the value of the minimum square error value between the plant response and the model response or using an eye judgment by overlaying both responses. This method gives less knowledge of the physical principles inside the system, and since there is no certain criterion how small the square error shoud be, this method is likely to give a subjective assessment. Because of this reason, not only optimum solutions, but also sensitivity studies on the model are important in validating the model. Another method which is relatively more objective is based upon model uncertainties [Butterfield and Thomas 1983]. This method also gives more knowledge of the physical principles and parameter sensitivities inside the system, and is applicable to highly nonlinear systems [Butterfield and Thomas 1983]. This

technique is called the Distortion Technique and will be used throughout this work.

## 2.6 Conclusions

In this chapter, the fundamental concepts of computer modelling and simulation as well as the philosophy of model building, parameter estimation and model validation have been presented. The procedure in performing a simulation task has been laid and this forms the foundation in doing the research work which will be presented through out this thesis.

# CHAPTER 3

# MATHEMATICAL MODEL OF A ROBOT MANIPULATOR

## 3.1 Introduction

Robotics is an interdisciplinary field where various disciplines in science and engineering give their own contributions. In general, the study of mechanics and control of robot manipulators is a collection of topics taken from mechanical engineering, electrical engineering, mathematics, computer science and control theory. Mechanical engineering provides tools to study the dynamics, electrical engineering gives contributions in designing sensors, computer controllers etc., mathematics contributes knowledge in spatial motions, computer science contributes programming and creating operating systems and control theory provides algorithms to perform desired motions.

Nowadays, the applications of industrial robots have increased enormously. Consequently, numerous research works are being performed in the robotics field. Alongside with the advances in computer technology, more advanced control algorithms have been proposed to perform more difficult manufacturing tasks in the industry. As a result, research work in the field of modelling robot manipulators become an important area. Many control algorithms depend on the accuracy of the mathematical model of a robot manipulator. Disturbances in the robot system can be greatly reduced if the dynamic behaviour of the robot manipulator is known accurately. Thus, the aim of mathematical modelling of industrial robots is to obtain information for control purposes.

A complete mathematical model of the system can be developed. Its application for

real time control purposes, however, in industry may be difficult since a high speed computer is needed. Due to economic reasons, only a relatively simple control algorithm is used in the industry. Many research works have been performed to overcome this problem ie. by developing efficient algorithms to obtain information about the dynamics of industrial robots [Huang, Lee, 1988; Li, 1989a; Li, 1989b; Vuskovic, Liang, Anantha, 1990].

In developing a mathematical model of robot systems prior to control realizations, there are subjects to deal with, namely kinematics, dynamics and trajectory descriptions. Kinematics is a subject which studies the analytical descriptions of the robot manipulator spatial motion as a function of time without regard to the torques/ forces that cause the motion. Dynamics deals with the mathematical equations which describes the dynamic behaviour of a robot manipulator. In order to perform some specified tasks, trajectories must be planned before hand. A trajectory is defined as a time history of position , velocity and acceleration for each degree of freedom where a degree of fredom is defined as an independent position variable [Craig, 1986]. Having obtained information in kinematics, dynamics and trajectory, control strategies can then be examined or developed.

### 3.2 Manipulator Description

An industrial robot may be considered as an open chain of bodies connected by joint as shown in figure (3.1). Each joint exhibits one degree of freedom. These bodies are commonly called links. An n degree of freedom industrial robot is composed of n joints, n links and a base. From now on, in the rest of this thesis, a robot refers to an industrial robot. All these links are numbered starting from the base which is usually called link 0 to link n which is the hand. Proximal links and

distal links refer to links close to the base and links distant from the base, respectively. At the last link, a tool is attached and it is usually called an end effector. Depending on the intended applications, an end effector may be a gripper, a welding torch or other devices.

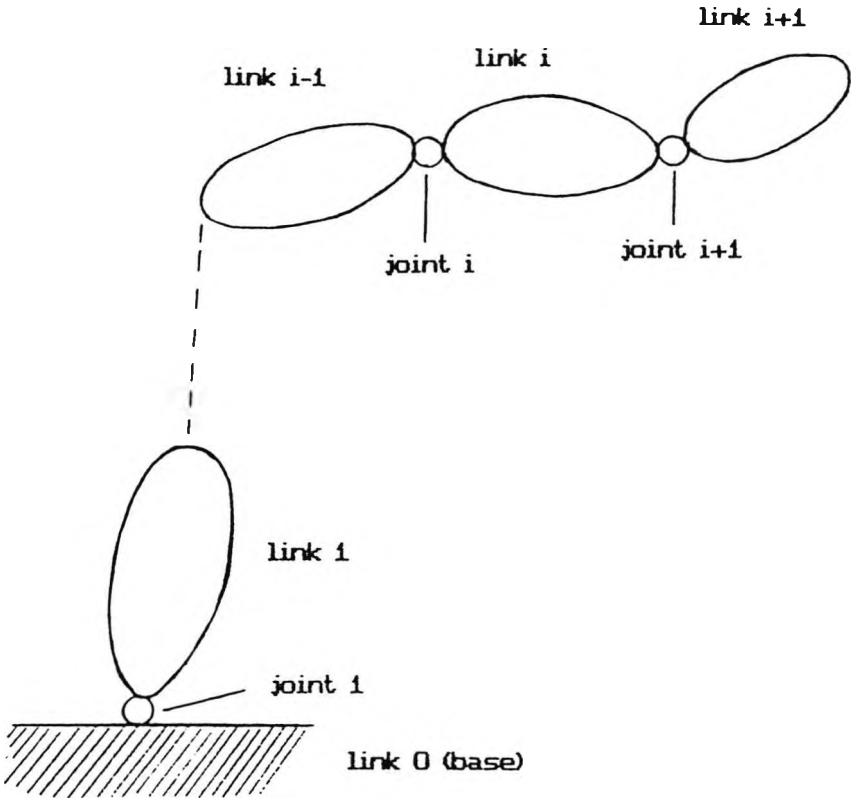



Figure 3.1

An industrial robot consists of a base, links and joints.

There are two types of joint which are commonly used in industrial robots, namely a revolute joint and a prismatic joint. In a rotary joint, link i rotates with respect to link i-1 while in a prismatic joint, link i slides with respect to link i-1, for i = 1 to n. These joints are equipped with position sensors which allow the relative position of the neighbouring links to be measured and sometimes are also equipped

with velocity sensors. More advanced robots may have torque/ force sensors in each joint. Each joint is driven by an actuator. This actuator may be electrically, hydraulically or pneumatically powered.

## 3.3 Denavit-Hartenberg (D-H) Representation

### 3.3.1 Denavit-Hartenberg Parameters

The relative motion of each joint results in the motion of a robot arm that gives the hand in the desired position and orientation. Denavit and Hartenberg [1955] proposed a spatial description between two neighbouring links as shown in figure (3.2) and the related parameters are called the Denavit-Hartenberg (D-H) parameters.

A joint axis i is established at the connection between link i-1 and link i. Thus, each link has two joints at both ends with the exception of link 0 and link n where they have only 1 joint. There are four parameters which are associated with each link in a robot manipulator system as shown in figure (3.2). In a revolute joint system, a prismatic joint system or a combination of both, there are two fixed parameters in each link, namely $a_i$ and $\alpha_i$. The parameter $a_i$ is the distance along the common normal between the joint i axis and the joint i+1 axis, and $\alpha_i$ is the angle between the joint i axis to the joint i+1 axis measured in a plane perpendicular to $a_i$. The other two parameters are $\theta_i$, which is the angle between $a_{i-1}$ and $a_i$ measured in a plane perpendicular to the joint i axis, and $d_i$ which is the distance between the intersections of the joint i axis with the common normals $a_{i-1}$ and $a_i$. For a revolute joint system, $\theta_i$ is the joint variable while $d_i$ remains constant and for a prismatic joint system, $d_i$ is the joint variable while $\theta_i$ remains

constant.

In many industrial robots, $a_i$ is simply the length of link i, $\alpha_i$ is the twist angle of link i and a typical value of $\alpha_i$ is either $0^o$ or $90^o$. In the rest of this thesis, the D-H parameters refer to $\alpha_i$, $a_i$, $\theta_i$ and $d_i$.

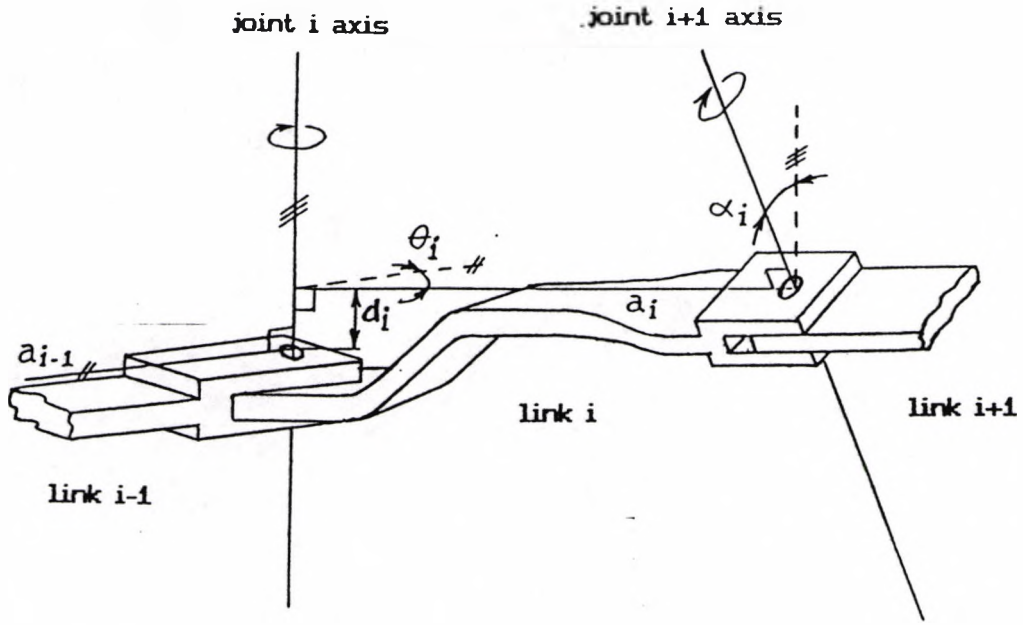



Figure 3.2

The D-H parameters in a robot manipulator.

### 3.3.2 Coordinate System Placement

To describe the spatial relationships of a robot arm motion which in turn gives the position and orientation of the hand, it is necessary to place a coordinate system (frame) to each link systematically and consistently. The Denavit -

Hartenberg concept gives a systematic orthonormal coordinate system assignment to an articulated chain of a robot arm. This orthonormal coordinate system is established at each link.

The coordinate system i is fixed at link i and located at joint i+1 as shown in figure (3.3). Since the coordinate system i and the coordinate system i-1 are fixed at link i and link i-1, respectively, when link i moves with respect to link i-1, the coordinate system i also moves with respect to the coordinate system i-1. The coordinate system 0 is attached to the base. This coordinate system is not moving and sometimes is called the base frame.

The assignment of every orthonormal coordinate system which is established for each link (except link n) follows the following rules [Denavit, Hartenberg, 1955] :

1. The $z_i$ axis lies along the axis of motion of the joint i+1.
2. The $x_i$ axis is perpendicular to a plane where the $z_{i-1}$ and the $z_i$ axes lie or along the common normal between the $z_{i-1}$ and the $z_i$ axes when they are parallel.
3. The $y_i$ axis is in a direction where it completes the right handed coodinate system.

The hand coordinate system is established by placing the $z_n$ axis parallel to the $z_{n-1}$ axis and the $x_n$ axis in normal to both the $z_{n-1}$ and $z_n$ axes. The $y_n$ is then established to complete the right handed coordinate system. In some cases, it is more convenient to have the origin of coordinate system 0 coincide with the origin of coordinate system 1.

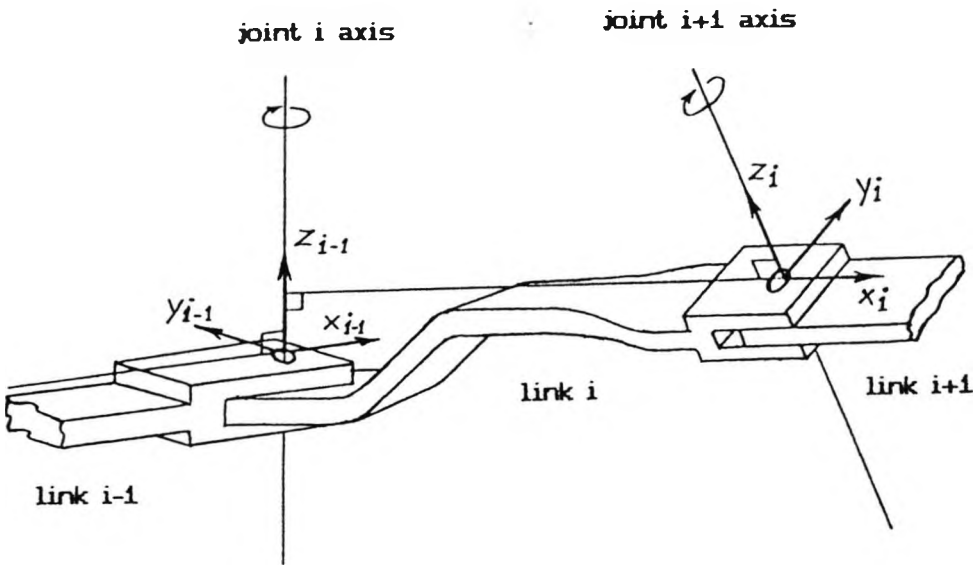



Figure 3.3

The Denavit-Hartenberg coordinate system.

### 3.4 Kinematics

In the study of robotics, the information about the location of an object is important. These objects can be the links of the robot, the end effector, the parts to be manipulated or other objects in the robot's environment (eg. some obstacles). This information is expressed in term of position and orientation.

When a coordinate system (x,y,z) is rotated an angle $\phi$ about its x axis, the rotation matrix $R_{x,\phi}$ which express the new coordinate system with respect to the old coordinate system is given by

$$R_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{3.1}$$

If a vector **b** is given as $(x_b,y_b,z_b)^T$ with respect to the coodinate system (**x**,**y**,**z**), then when this coordinate system is rotated an angle $\phi$ about the **x** axis, the new value of vector **b** with respect to the new coordinate system is obtained by performing the following operation,

$$\hat{\mathbf{b}} = R_{x,\phi}\mathbf{b} \tag{3.2}$$

where $\hat{\mathbf{b}}$ is the new value of vector **b** with respect to the new coordinate system.

The rotation matrices which express rotations of an angle $\beta$ about the **y** axis and of an angle $\gamma$ about the **z** axis are given by

$$R_{y,\beta} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{3.3}$$

$$R_{z,\gamma} = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

If more than one rotation is performed in a sequence, then the order of multiplication must be considered. To obtain a new value of vector **b** when the following sequence is performed; a rotation of an angle $\gamma$ about the **z** axis

followed by a rotation of an angle $\beta$ about the y axis and finally followed by a rotation of an angle $\phi$ about the x axis; the relationship is :

$$\hat{b} = R_{z,\gamma}\, R_{y,\beta}\, R_{x,\phi}\, b \qquad (3.5)$$

When a combination of rotation and translation is considered, a (4x4) homogenous transformation (rotation and translation) matrix T is introduced. This matrix consists of rotation, translation, scaling and perpective transformations. In the kinematics study, however, the scaling and perspective transformations are not used.

$$T = \begin{bmatrix} \text{Rotation matrix}_{3x3} & \text{Translation vector}_{3x1} \\ 0 & 1 \end{bmatrix} \qquad (3.6)$$

The translation vector gives the position of the new coordinate system with respect to the old coordinate system. In order to be able to perform the transformation appropriately, a '1' is added as the fourth element of a vector **b**, ie. $(x_b, y_b, z_b, 1)^T$. The procedure to perform this transformation is similar to the procedure of the rotational transformation. Thus, to express a coordinate system with respect to another coordinate system, it is necessary to know both the relative position and orientation between these two coordinate systems.

In a robot manipulator, a coordinate system i can be transformed into a coordinate system i-1. Based on the coordinate system assignment given in section 3.3.2, this task is carried out by performing a rotation, two translations and a final rotation as follows [Denavit, Hartenberg, 1955],

1. A rotation about the $z_{i-1}$ of $\theta_i$, so the $x_{i-1}$ axis is in parallel to the $x_i$ axis (pointing in the same direction).

2. A translatation of $d_i$ along the $z_{i-1}$ axis to locate the origin of the coordinate system i-1 at the point where the common normal between the $z_{i-1}$ and $z_i$ axes intersects the $z_{i-1}$ axis. The result is that the $x_{i-1}$ axis coincides with the $x_i$ axis.

3. A translation of $a_i$ along the $x_i$ axis, so both origins of the two coordinate systems coincide.

4. A rotation of $\alpha_i$ about the $x_i$ axis to bring the $z_{i-1}$ and $z_i$ axes into coincidence. The result is that the two coordinate systems have the same position and orientation.

The relationship of the D-H coordinate systems in the neighbouring links can be described using the 4x4 homogenous transformation matrix. By following the above operation to transform the coordinate system i to the coordinate system i-1, the D-H transformation matrix is obtained as follows [Denavit, Hartenberg, 1955]

$$\bar{A}^{1}_{i-1} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.7)$$

and the corresponding rotation matrix is given by

$$R^{i}_{i-1} = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i \\ \sin\theta_i & \cos\alpha_i\cos\theta_i & -\sin\alpha_i\cos\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i \end{bmatrix} \qquad (3.8)$$

For a multi link robot manipulator, the coordinate system i can be expressed with respect to the base coordinate system as

$$\bar{A}_0^{\,i} = \bar{A}_0^{\,1}\bar{A}_1^{\,2} \cdots\cdots \bar{A}_{i-2}^{\,i-1}\bar{A}_{i-1}^{\,i} \qquad (3.9)$$

Using this D-H homogenous transformation matrix, if the D-H parameters are known, one can obtain the position and orientation of the hand with respect to the base coordinate system.

### 3.5 Differential Solution

It is often necessary to compute the differential change of a 4x4 homogenous transformation matrix with respect to its variables. This knowledege is important in studying the dynamics of a robot manipulator where velocities of the joint variables are used extensively.

The differential transformation of a coordinate system T is defined as [Paul, 1981]

$$T + dT = T_{dx,dy,dz} \cdot T_{\delta_x,\delta_y,\delta_z} \cdot T \qquad (3.10)$$

where :

$dT$ = a differential change of a coordinate system T.

$T_{dx,dy,dz}$ = a differential translation along the x, y and z axes.

$T_{\delta_x,\delta_y,\delta_z}$ = a differential rotation about the x, y and z axes.

Defining,

$\delta_x$ = a small rotation of $\theta$ angle about the x axis.

$\delta_y$ = a small rotation of $\theta$ angle about the y axis.

$\delta_z$ = a small rotation of $\theta$ angle about the z axis.

and since for $\theta \approx 0$, $\sin\theta \approx \theta$ and $\cos\theta \approx 1$, from (3.1), (3.3), (3.4) and (3.6) the 4x4 homogenous differential rotation matrices (setting the translation vector to 0) are then expressed in the following form

$$T_{\delta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\delta_x & 0 \\ 0 & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.11)$$

$$T_{\delta_y} = \begin{bmatrix} 1 & 0 & \delta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\delta_y & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.12)$$

$$T_{\delta_z} = \begin{bmatrix} 1 & -\delta_z & 0 & 0 \\ \delta_z & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

If a differential rotation about the x, y and z axes is considered, then

$$T_{\delta_x,\delta_y,\delta_z} = T_{\delta_x} . T_{\delta_y} . T_{\delta_z} \tag{3.14}$$

Solving equation (3.14) yields (ignoring second and third order terms)

$$T_{\delta_x,\delta_y,\delta_z} = \begin{bmatrix} 1 & -\delta_z & \delta_y & 0 \\ \delta_z & 1 & -\delta_x & 0 \\ -\delta_y & \delta_x & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.15}$$

From equation (3.6) and setting the rotation matrix to a 3x3 identity matrix $I_3$, the 4x4 homogenous differential translation matrix is given by

$$T_{dx,dy,dz} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.16}$$

Substituting equations (3.15) and (3.16) to equation (3.10) gives the expression of dT

as

$$dT = \begin{bmatrix} 0 & -\delta_z & \delta_y & dx \\ \delta_z & 0 & -\delta_x & dy \\ -\delta_y & \delta_x & 0 & dz \\ 0 & 0 & 0 & 0 \end{bmatrix} .T \qquad (3.17)$$

or in the more compact form as

$$dT = \Delta \, . \, T \qquad (3.18)$$

All these differential relationships are expressed with respect to the base coordinate system.

In the case of the D-H transformation, the differential relationships with respect to the joint variables ($\theta_i$ or $d_i$) are obtained directly. From equation (3.7), the derivative of $\bar{A}^1_{i-1}$ with respect to the joint variable is given by

$$\frac{\partial}{\partial q_i}\bar{A}^1_{i-1} = \begin{bmatrix} -\sin\theta_i & -\cos\alpha_i\cos\theta_i & \sin\alpha_i\cos\theta_i & -a_i\sin\theta_i \\ \cos\theta_i & -\cos\alpha_i\sin\theta_i & \sin\alpha_i\sin\theta_i & a_i\cos\theta_i \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.19)$$

for a revolute joint.

$$\frac{\partial}{\partial q_i}\bar{A}^{i}_{i-1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.20)$$

for a prismatic joint.

Let, [Paul, 1972]

$$Q_i = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.21)$$

for a revolute joint, and

$$Q_i = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3.22)$$

for a prismatic joint.

The differential relationships now can be expressed as

$$\frac{\partial}{\partial q_i}\bar{A}^i_{i-1} = Q_i\ \bar{A}^i_{i-1} \tag{3.23}$$

For an n degree of freedom robot manipulator where i=1,....,n and $j \leq i$

$$\frac{\partial}{\partial q_j}\bar{A}^i_0 = \bar{A}^{j-1}_0\ \ Q_j\ \ \bar{A}^i_{j-1} \tag{3.24}$$

## 3.6 Dynamics

### 3.6.1 Overview of Robot Dynamics

Dynamics of a robot manipulator deals with mathematical equations of motion which gives the dynamic behaviour of the manipulator. A good dynamic model is important for controlling or simulating a motion of a robot arm accurately. The robot manipulator forward dynamics problem involves determining the motion of the manipulator resulting from a set of applied joint torques/ forces. The solution of this problem is a necessary requirement and, in fact, the heart of any dynamic simulation work. The inverse dynamics problem, as the name implies, uses knowledge of the joint position, velocities and accelerations of each link to calculate the required torques/ forces applied to each joint. Many advanced robot control schemes are based upon the accuracy of an inverse dynamics solution of the nonlinear robot dynamics.

The dynamic formulations of a robot arm can be obtained from basic physical laws such as the Lagrangian formulation and Newton law. The structure of these formulations, however, differ as they are required for various reasons and

intended purposes. Some are required to have fast computation time in performing real time control tasks, others are required to allow for control analysis. Since the complete dynamic model of a robot is complex, many researchers have endeavoured to develop efficient algorithms which can be implemented directly in the digital computer. Murray [1986] carried out a comparison among the then existing dynamics algorithms.

There are several techniques of modelling to derive the dynamic equations of a robot arm, such as :

- methods which are based on the Lagrangian equation [Paul, 1981; Li, 1989b].
- methods which are based on the Newton-Euler equations [Luh, Walker, Paul, 1980; Huang, Lee, 1988].

In general, two types of approaches are used in developing the dynamic equations, namely a closed form approach [Paul, 1981] and a recursive form approach [Luh, Walker, Paul, 1980; Hollerbach, 1980]. The former is preferable for state space control analysis purposes since this technique gives an explicit set of closed form differential equations. For real time control purposes, one would prefer the recursive form approach as this technique is very efficient in term of the computation time. The recursive nature, however, destroys the dynamic model structure if one wants to design the controller in the state space manner. From the above facts, there is some kind of trade off between the modelling and control aspects of a robot manipulator.

The dynamic description of a robot manipulator is a highly nonlinear multivariable system. The accuracy of modelling this complex system might be reduced by the structural deformation of the robot links since in reality each link to a certain

extent has a degree of flexibility [Huang, Lee, 1988]. In this research work, however, it is assumed that all links are rigid.

When the robot moves, kinetic energy which depends on the velocity of each joint exists in the system and, depending on the configuration, some joints suffer from gravity effects and thus potential energy also exists in the system. If the information about the total kinetic and potential energies in the system are evaluated, the Lagrangian function is then given by [Paul, 1981]

$$L = K - P \qquad (3.25)$$

where :

L = Lagrangian function.

K = Total kinetic energy of the system.

P = Total potential energy of the system.

Depending upon whether the joint is revolute or prismatic, the required generalized torque/ force at joint i in order to drive link i is obtained by forming the Lagrange-Euler equation which is given by

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = \tau_{1_i} \; ; \; i = 1,2, \ldots, n \qquad (3.26)$$

where :

$\tau_{1_i}$ = the generalized torque/force to drive link i

$q_i$ = the generalized joint i variable

$\dot{q}$ = the first derivative of generalized joint i variable

n = no. of degree of freedom

### 3.6.2 Newton-Euler (N-E) Approach

This approach is derived by analysing the net force and the net torque on an isolated link i, where the neighbouring links, ie. link i-1 and link i+1 are seen as exerting forces and torques to this link. Hence, in this apporoach, by having information on the angular and linear accelerations of link i, and the force and torque exerted by link i+1, one can obtain the generalized torque/ force applied to joint i in order to drive link i.

Consider a robot manipulator as shown in figure (3.4). When the robot links move, there is a propagation of velocities and accelerations from the base to the hand. Here, although the base does not move and, in fact because of this nature, the base actually propagates a linear acceleration to link 1 which is equal to the gravity acceleration. Thus, the angular/ linear velocity and the angular/ linear acceleration of link i with respect to the base coordinate system are cumulative. Since the coordinate system i is fixed to link i, this means that the coordinate system i is rotating and/ or translating with respect to the coordinate system i-1 and to the base coordinate system as well. As the robot manipulator which is used in this research work has revolute joints only, so the following derivation is intended for a revolute joint system only. From figure (3.4), some vectors are defined, namely

$\mathbf{p}_i$ = a translation vector of the origin of the coordinate system i from the origin of the coordinate system i-1 and expressed in the base coordinate system.

$\bar{\mathbf{s}}_i$ = a position vector of the centre of mass of link i from the origin of the

coordinate system i and expressed in the base coordinate system.

$\bar{c}_i$ = a position vector of the centre of mass of link i from the origin of the coordinate system i-1 and expressed in the base coordinate system.

$\hat{\omega}_i$ = an angular velocity of joint i with respect to the coordinate system i-1 and expressed in the base coordinate system.

$\dot{\hat{\omega}}_i$ = an angular acceleration of joint i with respect to the coordinate system i-1 and expressed in the base coordinate system.

$\omega_i$ = an angular velocity of joint i with respect to the base coordinate system and expressed in the base coordinate system.

$\dot{\omega}_i$ = an angular acceleration of joint i with respect to the base coordinate system and expressed in the base coordinate system.

$v_i$ = a linear velocity of the origin of the coordinate system i with respect to the base coordinate system and expressed in the base coordinate system.

$\dot{v}_i$ = a linear acceleration of the origin of the coordinate system i with respect to the base coordinate system and expressed in the base coordinate system.

$\bar{a}_i$ = a linear acceleration of the centre of mass of link i with respect to the base coordinate system and expressed in the base coordinate system.

The kinematic information is then given by

$$\bar{c}_i = p_i + \bar{s}_i \tag{3.27}$$

$$\hat{\omega}_i = z_{i-1}\,\dot{q}_i \tag{3.28}$$

$$\dot{\hat{\omega}}_i = z_{i-1}\,\ddot{q}_i \tag{3.29}$$

$$\omega_i = \omega_{i-1} + z_{i-1}\,\dot{q}_i \tag{3.30}$$

$$\dot{\omega}_i = \dot{\omega}_{i-1} + \omega_{i-1} \times \hat{\omega}_i + z_{i-1}\,\ddot{q}_i \tag{3.31}$$

$$v_i = v_{i-1} + \omega_i \times p_i \tag{3.32}$$

$$\dot{v}_i = \dot{v}_{i-1} + \dot{\omega}_i \times p_i + \omega_i \times (\omega_i \times p_i) \tag{3.33}$$

$$\bar{a}_i = \dot{v}_i + \dot{\omega}_i \times \bar{s}_i + \omega_i \times (\omega_i \times \bar{s}_i) \tag{3.34}$$

As joint i rotates about the $z_{i-1}$ axis, so $\hat{\omega}_i$, $\dot{\hat{\omega}}_i$ and the $z_{i-1}$ axis have the same direction as given by equation (3.28) and (3.29).

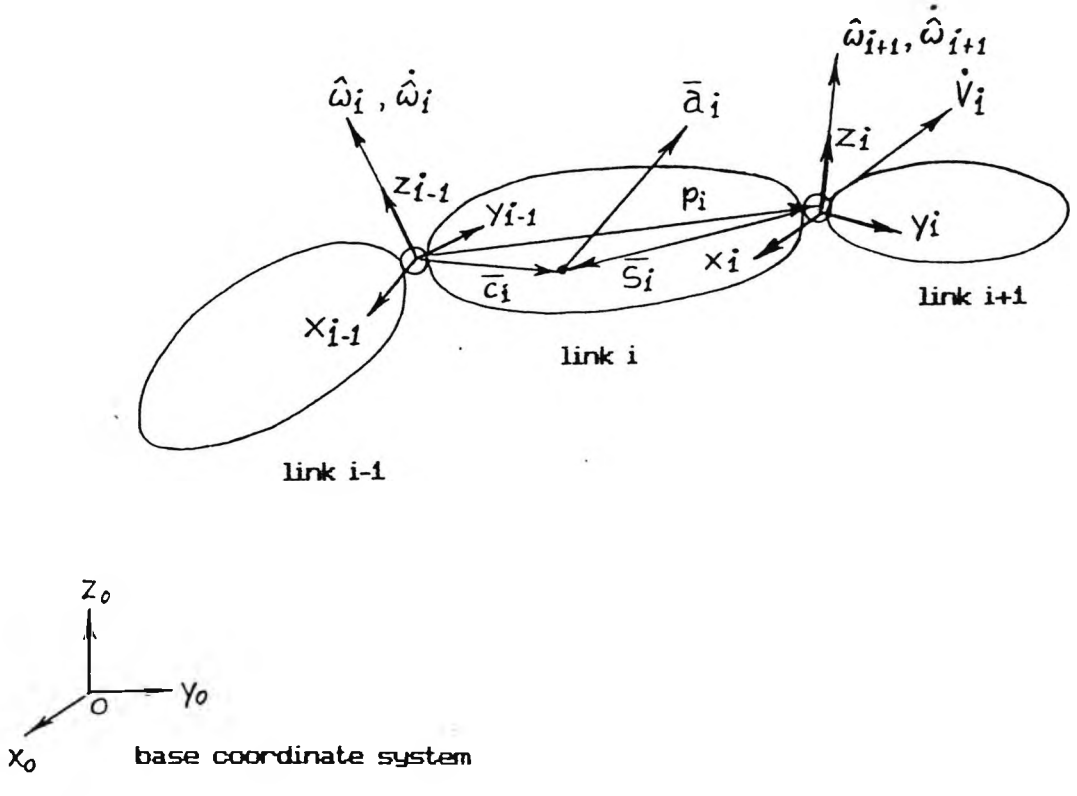



Figure 3.4

Vector assignments in the Newton-Euler approach.

Since the velocity and acceleration (both angular and linear) of link i are cumulative from the base, so the above equations are evaluated recursively from i=1 to i=n. Once the kinematic relationships are known, the forces and torques in the system can then be evaluated as shown in figure (3.5). As the name implies, the

Newton-Euler approach uses the Newton equation (for translation)

$$F_i = m_i \bar{a}_i \tag{3.35}$$

where :

$m_i$ = mass of link i.

$F_i$ = the net force acting at the centre of mass of link i and expressed in the base coordinate system.

the Euler equation (for rotation)

$$N_i = I_i\dot{\omega}_i + \omega_i \times (I_i\omega_i) \tag{3.36}$$

where :

$I_i$ = inertia tensor matrix of link i about its centre of mass and expressed in the base coordinate system. Detail of the inertia tensor matrix $I_i$ is given in appendix A.

$N_i$ = net torque acting on the centre of mass of link i and expressed in the base coordinate system.

and the d'Alembert principle which applies the static equilibrium conditions to problems in dynamics. Using the d'Alembert principle, the algebraic sum of external forces/ torques exerted on link i and the forces/ torques which resist the motion is zero.

From figure (3.5), applying the d'Alembert principle gives

$$f_i = f_{i+1} + F_i \tag{3.37}$$

$$n_i = n_{i+1} + N_i + c_i \times F_i + p_i \times f_{i+1} \tag{3.38}$$

where :

$f_i$ = force exerted at joint i to support link i and the distal links.

$n_i$ = torque exerted at joint i to support link i and the distal links

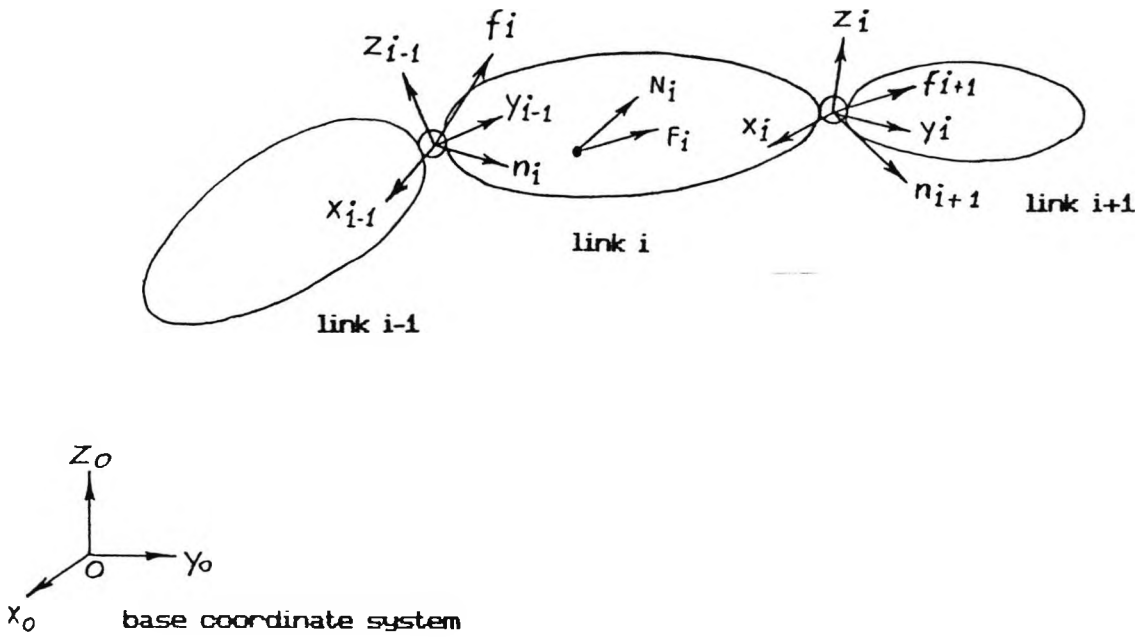



Figure 3.5

Forces and torques in a robot mechanical linkage

From the establishment of the D-H coordinate system, link i rotates about the $z_{i-1}$ axis. Thus, to drive link i, the applied input torque at joint i is the projection of $n_i$ onto the $z_{i-1}$ axis. Hence, the necessary torque given by the actuator to joint i is given by

$$\tau_{l_i} = n_i^T z_{i-1} \qquad (3.39)$$

where :

$\tau_{l_i}$ = applied input torque to drive link i given by the actuator to joint i with respect to the link shaft.

As link i must support the link above it, so equation (3.35) to (3.39) are performed recursively from i=n to i=1. It should be noted that as the linear acceleration of the base is assigned to the gravity acceleration, the value of $F_i$ given by equation (3.35) includes the gravity effect.

When the Newton-Euler algorithm is completely carried out, it yields a complete dynamic equation of motion and can be expressed in the following matrix form

$$\tau_l = D(\theta)\,\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) \qquad (3.40)$$

where :

$\tau_l$ = an nx1 generalized torque vector applied at joint i where i=1 to n.

$D(\theta)$ = an nxn inertial acceleration related symmetric matrix.

$H(\theta,\dot{\theta})$ = an nx1 coriolis/ centrifugal torque vector.

$G(\theta)$ = an nx1 gravity torque vector.

$\theta$ = an nx1 joint variable position vector with respect to the link side.

$\dot{\theta}$ = an nx1 joint variable velocity vector with respect to the link side.

$\ddot{\theta}$ = an nx1 joint variable acceleration vector with respect to the link side.

Equation (3.40) shows that the dynamic equation of a robot manipulator is composed of a set of highly nonlinear coupled second order differential equations where the D matrix is always non singular [Fu, Gonzalez and Lee, 1987].

To improve the efficiency in computation, the inertia tensor matrix $I_i$ as well as $p_i$, $\bar{s}_i$ and $\bar{c}_i$ are expressed in the coordinate system i [Luh, Walker, Paul, 1980].

## 3.7 Trajectory

### 3.7.1 Trajectory in General

In many applications, it is necessary to have a preplanned path (trajectory), so that the robot can move efficiently with or without any obstacle in the work space. A trajectory is defined as a time history of position, velocity and acceleration of each degree of freedom [Craig, 1986]. When there is no trajectory constraint, the problem is simply a positional control. Although for the purpose of designing or examining a control system one usually, for convenience, uses only trajectories which do not have constraints such as a step input, a ramp step input or a sinusoidal input; however it is appropriate to mention trajectories which are used in typical applications of the industrial robot, eg. in a pick and place task.

In designing a robot system, not only does the system control the motion of the robot, but it also needs to be capable of planning the motion trajectory from the initial joint configuration to the final joint configuration. Once the kinematic information, such as the initial and final positions and orientation of the hand in term of the joint coordinates, to manipulate an object has been given, the system

then generates the motion trajectory in order to accomplish the given task. The initial and final conditions are commonly defined as

$$\theta_i(t_o) = \theta_{io} \tag{3.41}$$

$$\theta_i(t_f) = \theta_{if} \tag{3.42}$$

$$\dot{\theta}_i(t_o) = \dot{\theta}_{io} \tag{3.43}$$

$$\dot{\theta}_i(t_f) = \dot{\theta}_{if} \tag{3.44}$$

where :

$i$ = joint number

$t_o$ = initial time

$t_f$ = final time

and normally, $\dot{\theta}_i(t_o)$ and $\dot{\theta}(t_f)$ are set to zero.

Logically, a smooth continuous trajectory is preferable than a jerky discontinuous one. This requires that the first and the second derivatives of the trajectory must exist to guarantee the continuity of the trajectory. The simplest approach to obtain this is by implementing a cubic trajectory as given in the following form

$$\theta_i(t) = \theta_{io} + h_{i1}t + h_{i2}t^2 + h_{i3}t^3 \tag{3.45}$$

Figure (3.6) shows this trajectory.

To avoid any collision with the environment's obstacles if any, a more detailed trajectory might be necessary than just the initial and final points only. Thus, some times the trajectory needs one or more via points so that a trajectory is

divided into some segments. For a trajectory with two segments as shown in figure (3.7), the initial and final conditions of the both segments are given as

$$\theta_{i1}(t_{o1}) = \theta_{io1} \tag{3.46}$$

$$\theta_{i2}(t_{o2}) = \theta_{i1}(t_{f1}) \tag{3.47}$$

$$\dot{\theta}_{i1}(t_{o1}) = \dot{\theta}_{io1} \tag{3.48}$$

$$\dot{\theta}_{i2}(t_{o2}) = \dot{\theta}_{i1}(t_{f1}) \tag{3.49}$$

$$\theta_{i2}(t_{f2}) = \theta_{i2f2} \tag{3.50}$$

$$\dot{\theta}_{i2}(t_{f2}) = \dot{\theta}_{i2f2} \tag{3.51}$$

and the trajectory segments are in the following form

$$\theta_{i1}(t) = \theta_{i1o} + h_{i11}t + h_{i12}t^2 + h_{i13}t^3 \tag{3.52}$$

$$\theta_{i2}(t) = \theta_{i2o} + h_{i21}t + h_{i22}t^2 + h_{i23}t^3 \tag{3.53}$$

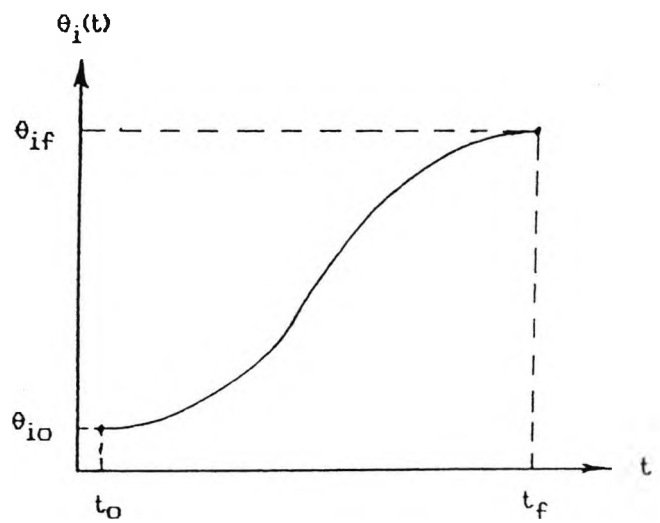



Figure 3.6

A cubic trajectory

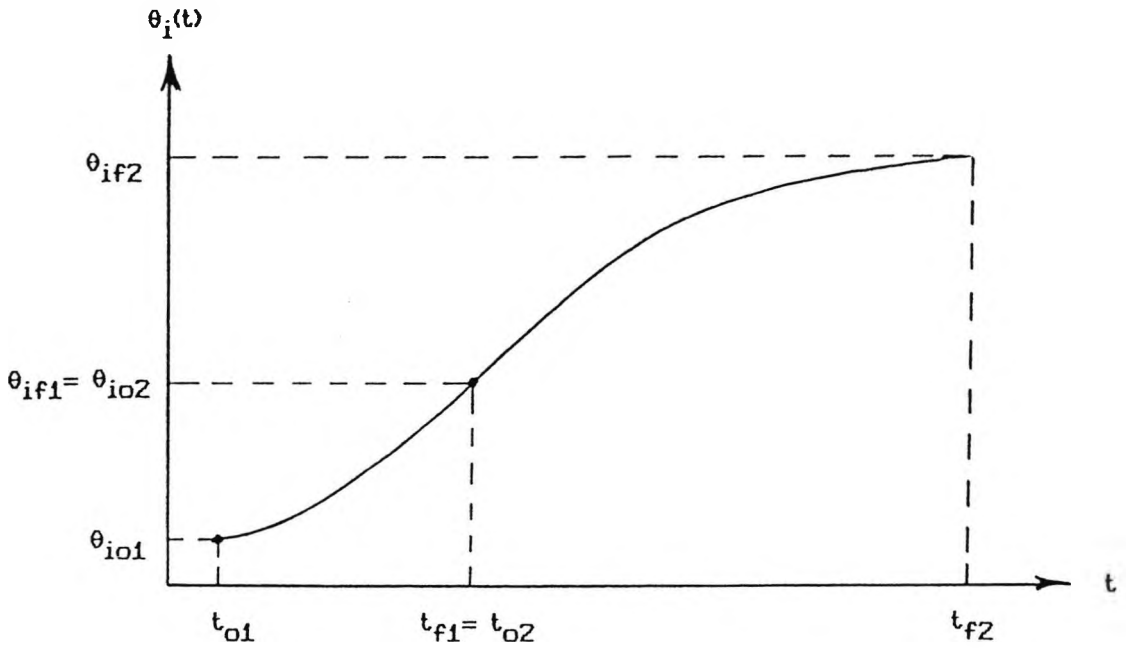



Figure 3.7

A cubic trajectory with two segments

### 3.7.2 Trajectory in a Pick And Place Task

Typical applications of an industrial robot are some sort of pick and place like tasks, eg. picking and placing objects to a conveyor belt and assembling printed circuit boards. Consider a robot which performs a picking motion. After the robot grips an object, the joints are commanded to move to their corresponding final points. If there is not any constraint imposed on the picking movement, the robot may drag the object instead of pick it up or even the robot tip may collide with the surface which supports the object. Since the object needs to be picked gently in order not to damage it as well as the robot itself, another consideration in planning a trajectory must be taken into account.

Paul [1972] suggested an additional constraint which must be considered. Instead of bringing the object directly to the final point, it is necessary to lift the arm up first in order to clear the supporting surface. Thus, as the hand starts moving, the motion must be in the normal direction and pointing away from the surface. This via point which is in the normal direction to the surface is called a lift off point. After passing this lift off point, the robot can bring the object to the desired location.

A similar constraint as in the picking movement is necessary when the robot wants to place the object in the desired location. A collision with the surface may happen if the arm directly places the object. Instead of placing it directly, the hand must pass another via point which is in the normal direction from the surface and this via point is called a set down point. Having passed this set down point, the hand then approaches the surface in the direction of the normal.

Hence two points alone, ie. the initial and the final points, are not sufficient in order to avoid collision with the supporting surface and two other via points, ie. the lift off and the set down points are necessary to be considered in generating a trajectory. Paul [1972] recommended that both the lift off and the set down points must be at least 25% of the length of the last link. Figure 3.8 shows this trajectory.

From figure (3.8), the trajectroy which has the initial, the lift off, the set down and the final points is divided into three, ie. a picking motion (from the initial point to the lift off point), a travelling motion (from the lift off point to the set down point) and a placing motion (from the setdown point to the final point). One approach to carry out this kind of trajectory is by assigning a fourth order

polynomial to both the picking and the placing motion as given by (with an appropriate subscript for the picking and the placing motion)

$$\theta_i(t) = \theta_{io} + h_{i1}t + h_{i2}t^2 + h_{i3}t^3 + h_{i4}t^4 \qquad (3.54)$$

and a third order polynomial to the travelling motion. Normally, the velocities and accelerations at the initial and final points are zero. To avoid any obstacle, the travelling motion can be further divided into some smaller cubic polynomial segments as detailed in section 3.7.1.

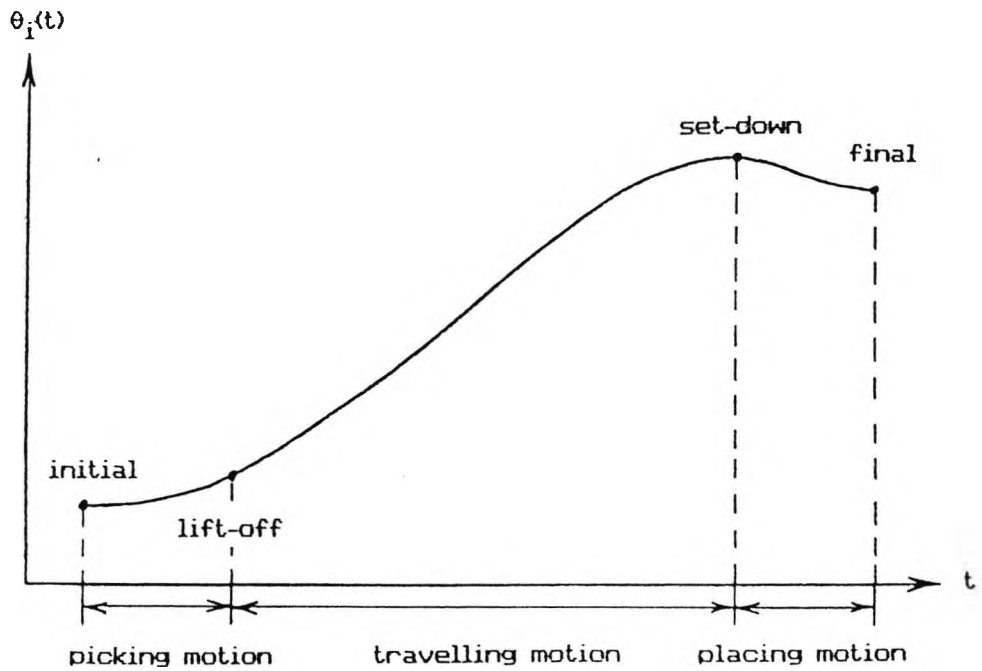



Figure 3.8

A trajectory in the pick and place task

## 3.8 Control System

### 3.8.1 General

In general, robot control is an application of microprocessor equipped controllers to carry out mechanical manipulation tasks in a desired manner. The main objective of the controller is to provide the necessary torques to all joints in such away so that the robot arm can follow the desired trajectory with a considerably small error. Since the controller is programmable, theoretically any control algorithm can be implemented. Hence, in more general, the aim of a robot control is to obtain a good performance of a robot manipulator with specified characteristics. A typical closed loop robot control system is shown below in figure (3.9).

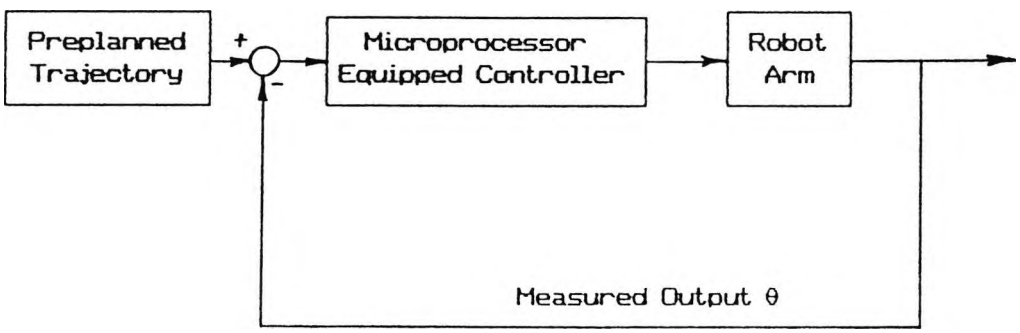



Figure 3.9

A closed loop robot control system

### 3.8.2 Complete Dynamic Robot Model

Before going into the control strategy, it is necessary to develop a robot model with an integrated actuator. The dynamics of the actuators have to be included in the model since they are part of the whole robot system. Here, an electrically powered actuator is considered, ie. a DC motor with permanent magnet. This motor is armature controlled and operated in its linear range.

The torque given to each joint is a function of the current and the torque constant. For joint i,

$$\tau_i = K_{Ti}\,\tilde{I}_i \tag{3.55}$$

where :

$\tau_i$ = output torque of the i-th actuator [Nm].

$K_{Ti}$= torque constant of the i-th actuator [Nm/Amp].

$\tilde{I}_i$ = armature current of the i-th actuator [Amp].

As the speed increases, the back EMF voltage, which is induced in the armature, increases proportionally and this tends to reduce the current.

$$U_i - E_i = \tilde{I}_i\,R_i + L_i\,\dot{\tilde{I}}_i \tag{3.56}$$

$$E_i = K_{bi}\,\dot{\theta}_{mi} \tag{3.57}$$

where :

$U_i$ = input voltage of the i-th actuator [Volt].

$E_i$ = back EMF voltage of the i-th actuator [Volt].

$K_{bi}$= back EMF constant of the i-th actuator [Volts/rad].

$R_i$ = terminal resistance of the i-th actuator [Ohm].

$L_i$ = rotor inductance of the i-th actuator [mH].

$\dot{\theta}_{mi}$ = angular velocity of the joint i with respect to the actuator side [Radian/s].

Since the rotor has an inertia value, so

$$\tau_i = J_{mi}\,\ddot{\theta}_{mi} + \tau_{mi} \tag{3.58}$$

where :

$J_{mi}$ = i-th actuator rotor inertia [$kgm^2$]

$\ddot{\theta}_{mi}$ = i-th joint angular acceleration with respect to the actuator side [$Radian/s^2$]

$\tau_{mi}$ = generalized torque of joint i with respect to the actuator side [Nm]

From equation (3.55), (3.56), (3.57) and (3.58)

$$J_{mi}\,\ddot{\theta}_{mi} = K_{Ti}\,\tilde{I}_i - \tau_{mi} \tag{3.59}$$

$$L_i\,\dot{\tilde{I}}_i = U_i - K_{bi}\,\dot{\theta}_{mi} - \tilde{I}_i\,R_i \tag{3.60}$$

Let the state vector be $\mathbf{x_i} = (\theta_{mi}, \dot{\theta}_{mi}, \tilde{I}_i)^T$, then the third order mathematical model of the actuator can be expressed in the state equation as

$$\begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \\ \dot{x}_{3i} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{K_{Ti}}{J_{mi}} \\ 0 & -\frac{K_{bi}}{L_i} & -\frac{R_i}{L_i} \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \\ x_{3i} \end{bmatrix} +$$

$$\begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_i} \end{bmatrix} U_i + \begin{bmatrix} 0 \\ -\frac{1}{J_{mi}} \\ 0 \end{bmatrix} \tau_{mi} \qquad (3.61)$$

As the electrical time constant is much smaller than the mechanical time constant, the inductance can be neglected and hence, equation (3.56) reduces to

$$U_i - E_i = \tilde{I}_i R_i \qquad (3.62)$$

Let the new state vector be $\mathbf{x_i} = (\theta_{mi}, \dot{\theta}_{mi})^T$, the second order mathematical model of the actuator is then given in the following form

$$\begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{K_{Ti}K_{bi}}{R_i J_{mi}} \end{bmatrix} \begin{bmatrix} x_{1i} \\ x_{2i} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{K_{Ti}}{R_i J_{mi}} \end{bmatrix} U_i + \begin{bmatrix} 0 \\ -\frac{1}{J_{mi}} \end{bmatrix} \tau_{mi} \qquad (3.63)$$

Or, in a more condensed form for the i-th actuator model

$$\dot{\mathbf{x}}_i = \mathbf{A}_i \mathbf{x}_i + \mathbf{B}_i U_i + \mathbf{C}_i \tau_{mi} \qquad (3.64)$$

and for an n degree of freedom robot manipulator

$$\dot{x} = A\,x + B\,U + C\,\tau_m \qquad (3.65)$$

where :

$x$ = a 2n x 1 state vector

$U$ = a n x 1 input voltage vector

$\tau_m$ = a n x 1 generalized torque vector with respect to the actuator side

$A = \mathrm{diag}(A_1, A_2, \ldots, A_n)$

$B = \mathrm{diag}(B_1, B_2, \ldots, B_n)$

$C = \mathrm{diag}(C_1, C_2, \ldots, C_n)$

The block diagram of a complete robot system with a second order actuator is shown in figure (3.10).

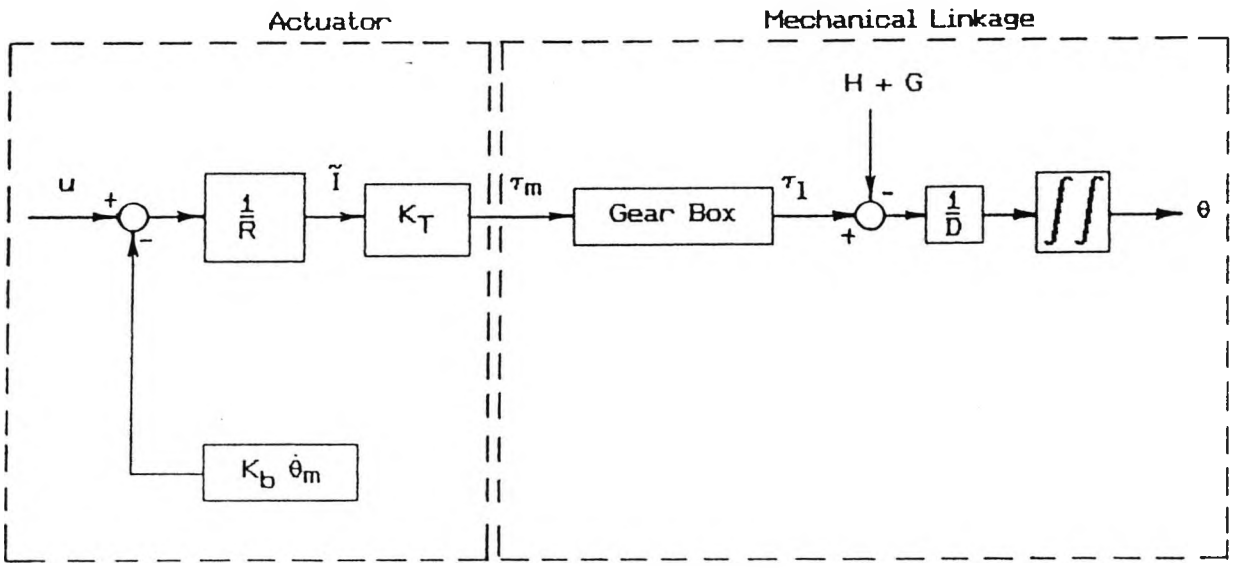



Figure 3.10

A complete robot model with an integrated actuator.

The generalized torque produced by the actuator has a relationship with the input generalized torque to the manipulator as

$$\tau_{mi} = r_i \, \tau_{li} \tag{3.66}$$

and the joint variable relationship is given as

$$\theta_i = r_i \, \theta_{mi} \tag{3.67}$$

where :

$r_i$ = gear ratio of joint i ($r_i \leq 1$)

Equation (3.40) can be rewritten in the following form (dropping $\theta$, $\dot{\theta}$ for brevity)

$$\tau_l = D \, \ddot{\theta} + P \tag{3.68}$$

where:

$$P = H + G \tag{3.69}$$

Let the gear ratio of a robot manipulator with n degree of freedom be

$$N = \begin{bmatrix} r_1 & & & \\ & r_2 & \bigcirc & \\ \bigcirc & & & \\ & & & r_n \end{bmatrix} \tag{3.70}$$

where :

$\mathbf{N}$ = an nxn gear ratio diagonal matrix.

From equations (3.66), (3.67), (3.68) and (3.70) the generalized torque vector with respect to the actuator shaft can now be expressed in the following form

$$\tau_m = \mathbf{N}\,\mathbf{D}\,\mathbf{N}\,\ddot{\theta}_m + \mathbf{N}\,\mathbf{P} \tag{3.71}$$

Let the transformation matrix between $\ddot{\theta}_m$ and $\ddot{x}$ be Z such that

$$\ddot{\theta}_m = Z\,\ddot{x} \tag{3.72}$$

where Z is an nx2n matrix and given as

$$Z = \begin{bmatrix} 0 & 1 & 0 & 0 & & 0 \\ 0 & 0 & 0 & 1 & & 0 \\ & & & & & \\ 0 & & & & 0 & 1 \end{bmatrix} \tag{3.73}$$

Substituting equation (3.72) into (3.71) yields

$$\tau_m = \mathbf{N}\,\mathbf{D}\,\mathbf{N}\,Z\,\ddot{x} + \mathbf{N}\,\mathbf{P} \tag{3.74}$$

Substituting (3.65) into (3.74) and manipulating gives

$$\tau_m = \left(I_n - \mathbf{N}\,\mathbf{D}\,\mathbf{N}\,Z\,C\right)^{-1}\left(\mathbf{N}\,\mathbf{D}\,\mathbf{N}\,Z\,(Ax + BU) + \mathbf{N}\,\mathbf{P}\right) \tag{3.75}$$

where :

$I_n$ = an nxn identity matrix

Hence, equation (3.65) now becomes

$$\dot{x} = \left(I_n + C\,(I_n - N\,D\,N\,Z\,C)^{-1}\,N\,D\,N\,Z\right) A\,x + \left(I_n + C(I_n - N\,D\,N\,Z\right) B\,U + C(I_n - NDNZC)^{-1}\,N\,P \qquad (3.76)$$

### 3.8.3 Control Strategies

The purpose of robot manipulator control is to keep the motion of the robot in a correct trajectory with a specified characteristic response. The problem becomes complicated since the nonlinear and coupling inertial torques, gravity torques and other disturbances are inherent in the system.

If it is assumed that the coupling inertial torques are small enough to be neglected, the control design becomes less complicated. Figure (3.11) shows a closed loop robot control system for joint i. The preplanned trajectory gives a set point to this joint as a function of time. If necessary, the desired velocity and acceleration are obtained by evaluating the first and second derivatives of the preplanned trajectory, respectively.

The position error is obtained by subtracting the observed position from the desired position. This negative position feedback reduces the position error. This position error is then converted to voltage by the means of a potentiometer or optical encoder/ counter assembly with a gain of $K_{\theta i}$ [Luh, 1983]. The velocity feedback is obtained by taking the first derivative of the position error. The

negative velocity feedback provides damping into the system.

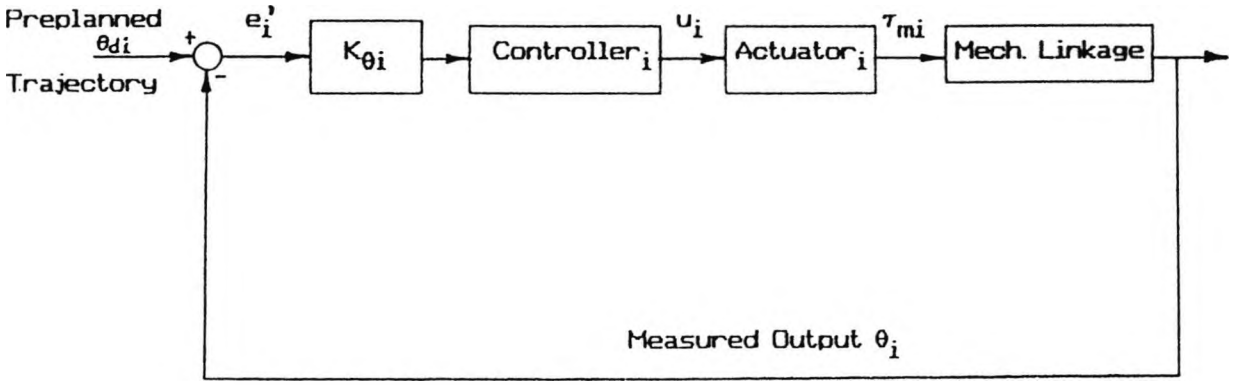



Figure 3.11

A closed loop robot control system of joint i

Combining equations (3.55), (3.57) and (3.62) gives

$$\tau_i = \frac{U_i - K_{bi}\dot{\theta}_{mi}}{R_i} K_{Ti} \qquad (3.77)$$

Substituting equation (3.58) into (3.77) and rearranging yields

$$U_i = \frac{R_i}{K_{Ti}} J_{mi} \ddot{\theta}_{mi} + \frac{R_i}{K_{Ti}} \tau_{mi} + K_{bi}\dot{\theta}_{mi} \qquad (3.78)$$

As in reality, only the measured position with respect to the link side can be obtained, equation (3.78) then becomes

$$U_i = \frac{R_i}{K_{Ti}} \frac{J_{mi}}{n_i} \ddot{\theta}_i + \frac{R_i}{K_{Ti}} \tau_{mi} + \frac{K_{bi}}{n_i} \dot{\theta}_i \qquad (3.79)$$

The generalized torque of joint i with respect to the actuator side can be expressed as

$$\tau_{mi} = n_i D_{ii} \ddot{\theta}_i + n_i P_i \qquad (3.80)$$

Substituting equation (3.80) into (3.79) gives

$$U_i = \frac{R_i}{K_{Ti}} \left( \frac{J_{mi}}{n_i} + n_i D_{ii} \right) \ddot{\theta}_i + \frac{K_{bi}}{n_i} \dot{\theta}_i + \frac{R_i}{K_{Ti}} n_i P_i \qquad (3.81)$$

If all terms other than the inertial torque are treated as disturbances, then equation (3.81) becomes

$$U_i = M_{ii} \ddot{\theta}_i + S_i \qquad (3.82)$$

where :

$$M_{ii} = \frac{R_i}{K_{Ti}} \left( n_i D_{ii} + \frac{J_{mi}}{n_i} \right) \qquad (3.83)$$

$$S_i = \frac{K_{bi}}{n_i} \dot{\theta}_i + \frac{R_i}{K_{Ti}} n_i P_i \qquad (3.84)$$

Figure (3.12) shows the revised block diagram of a closed loop system.

Consider a proportional + derivative controller which is employed in joint i as shown in figure (3.13). Since the coriolis and centrifugal terms are negligible in a low speed motion, so the principal disturbances are due to the gravity effect and back EMF. Working through the block diagram given in figure (3.13) and defining the state vector with respect to the link side, $\tilde{x}_i = (\theta_i, \dot{\theta}_i)^T$, this system can be expressed in the following form

$$\begin{bmatrix} \dot{\tilde{x}}_{1i} \\ \dot{\tilde{x}}_{2i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{x}_{1i} \\ \tilde{x}_{2i} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M_{ii}} \end{bmatrix} (U_i - S_i) \tag{3.85}$$

From figure (3.13),

$$e_i' = \theta_{di} - \theta_i \tag{3.86}$$

$$U_i = K_{\theta i} \left( K_{Pi}\, e_i' + K_{Di}\, \dot{e}_i' \right) \tag{3.87}$$

Combining equation (3.85), (3.86) and (3.87) yields

$$\begin{bmatrix} \dot{\tilde{x}}_{1i} \\ \dot{\tilde{x}}_{2i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{K_{Pi}K_{\theta i}}{M_{ii}} & -\frac{K_{Di}K_{\theta i}}{M_{ii}} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1i} \\ \tilde{x}_{2i} \end{bmatrix} +$$

$$\begin{bmatrix} 0 & 1 \\ \frac{K_{Pi}K_{\theta i}}{M_{ii}} & \frac{K_{Di}K_{\theta i}}{M_{ii}} \end{bmatrix} \begin{bmatrix} \theta_{di} \\ \dot{\theta}_{di} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{M_{ii}} \end{bmatrix} S_i \tag{3.88}$$

Let the input vector be $u_i = (\theta_{di},\ \dot{\theta}_{di})^T$, equation (3.88) can then be expressed in a more compact form as

$$\dot{\tilde{x}}_i = \tilde{A}_i\, \tilde{x}_i + \tilde{B}_i\, u_i + \tilde{C}_i\, S_i \tag{3.89}$$

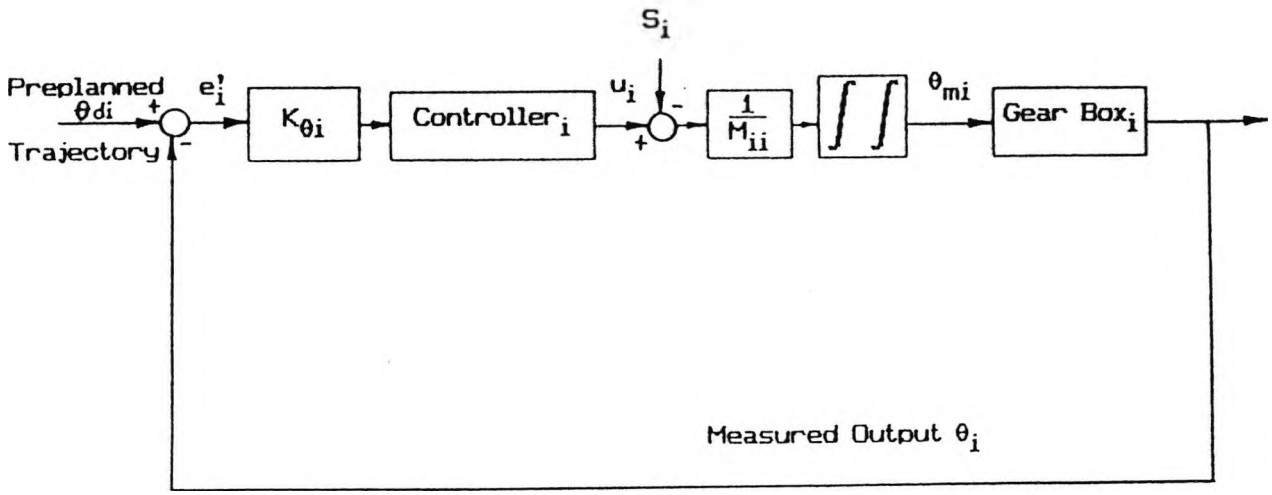



Figure 3.12

The revised block diagram of a closed loop system in robot control

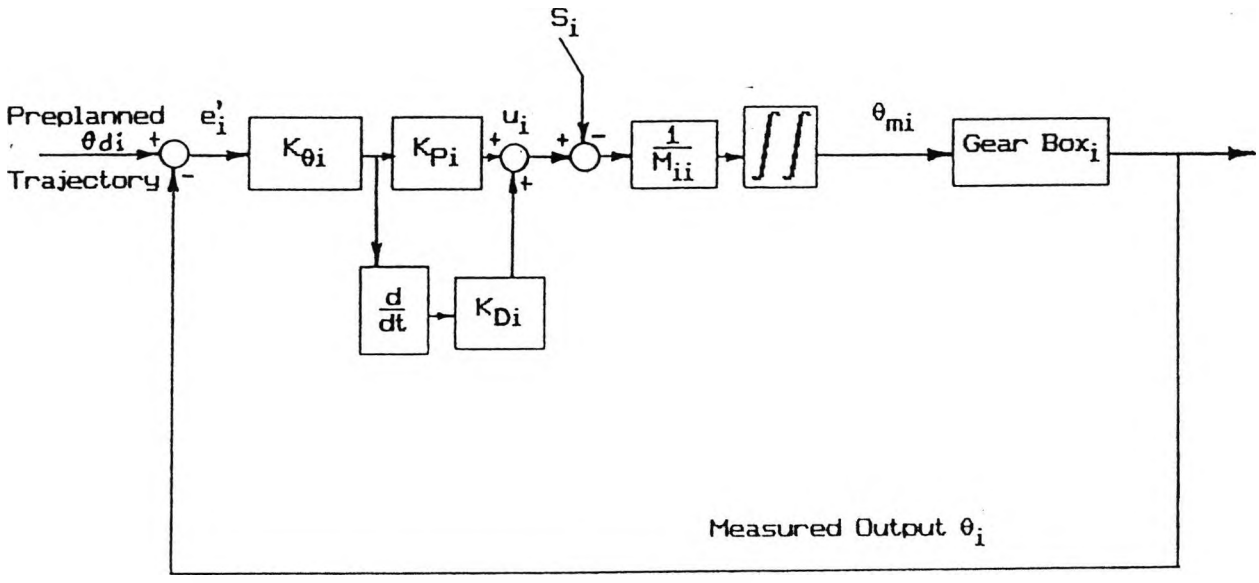



Figure 3.13

A PD control system

From here, it can be seen that the characteristic of the system which is given by the eigenvalues of matrix $\tilde{A}_i$ depends on the effective inertia $M_{ii}$. Since the effective inertia, which is a function of $\theta$, varies as the posture of the robot arm changes, it is not possible to maintain the system characteristic independent of the robot arm configuration.

The value of this effective inertia can be evaluated using the dynamics mathematical model which was developed in the preceeding section and is denoted as $\hat{M}_{ii}$. The computed model effective inertia value is then added to the system as a gain. Figure (3.14) shows this modification. With $\hat{M}_{ii}$ added as a gain in the system, equation (3.82) then becomes

$$\hat{M}_{ii} U_i = M_{ii} \ddot{\theta}_i + S_i \qquad (3.90)$$

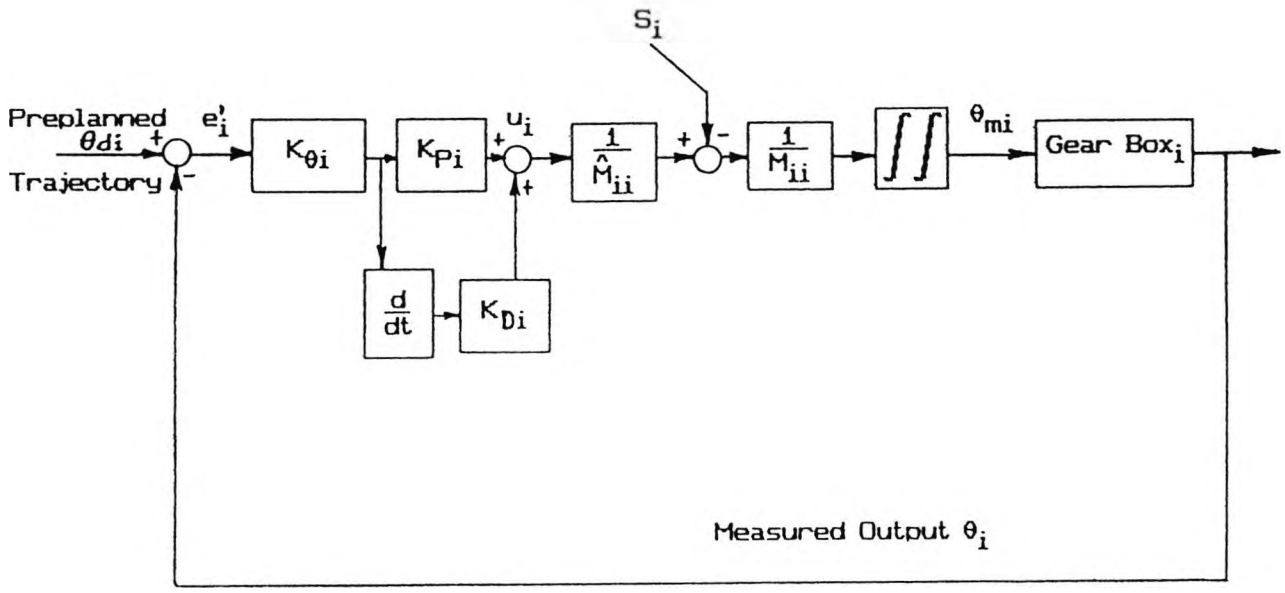



Figure 3.14

A linearized closed loop control system

If it is assumed that there is no modelling error, ie. $\hat{M}_{ii} = M_{ii}$, equation (3.88) becomes

$$\begin{bmatrix} \dot{\tilde{x}}_{1i} \\ \dot{\tilde{x}}_{2i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_{Pi}K_{\theta i} & -K_{Di}K_{\theta i} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1i} \\ \tilde{x}_{2i} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ K_{Pi}K_{\theta i} & K_{Di}K_{\theta i} \end{bmatrix} \begin{bmatrix} \theta_{di} \\ \dot{\theta}_{di} \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{1}{M_{ii}} \end{bmatrix} S_i \qquad (3.91)$$

Equation (3.91) shows that the characteristic of the system no longer depends on the effective inertia. To further improve the performance of the system, the disturbance $S_i$ should be eliminated from the system. With the aid of the dynamics mathematical model, the value of this disturbance model can be obtained and is denoted as $\hat{S}_i$. Again, with the assumption that the model is perfect, ie. $\hat{S}_i = S_i$, injecting this signal into the system will eliminate the disturbance $S_i$ from the system. Figure (3.15) shows this modification.

Having added this compensation signal into the system, equation (3.91) becomes

$$\begin{bmatrix} \dot{\tilde{x}}_{1i} \\ \dot{\tilde{x}}_{2i} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -K_{Pi}K_{\theta i} & -K_{Di}K_{\theta i} \end{bmatrix} \begin{bmatrix} \tilde{x}_{1i} \\ \tilde{x}_{2i} \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ K_{Pi}K_{\theta i} & K_{Di}K_{\theta i} \end{bmatrix} \begin{bmatrix} \theta_{di} \\ \dot{\theta}_{di} \end{bmatrix} \qquad (3.92)$$

and a critically damped response is obtained when

$$K_{Di} = 2\left(\frac{K_{Pi}}{K_{\theta i}}\right)^{\frac{1}{2}} \qquad (3.93)$$

To suppress the error quickly, $K_{Pi}$ must be set to a large value. In practice, however, the choice of $K_{Di}$ value is limited due to the presence of noise in differentiating the error.

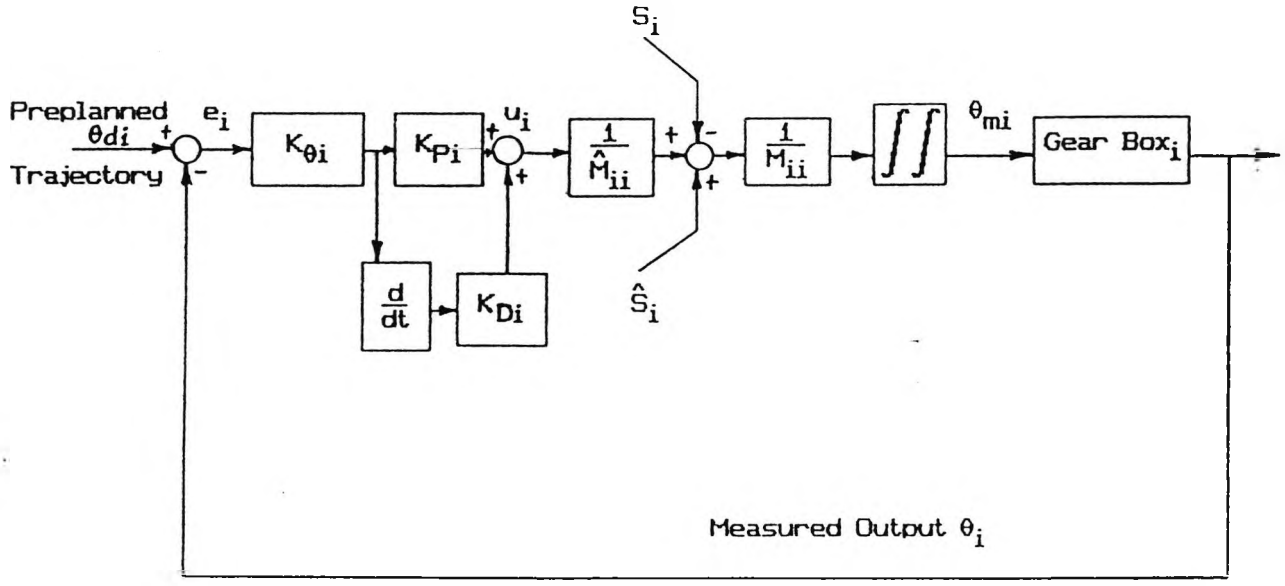



Figure 3.15

A linearized closed loop control system
with disturbance compensation signal

This control strategy is the basic principle of the computed torque technique [Markiewicz, 1973]. Some control algorithms which are based on this principle are the alpha computed torque technique [Benson, 1987] and the control partitioning

technique [Craig, 1986]. Other algorithms such as an adaptive control method can also be based on this technique but with an addition of an adaptive element [Craig, Hsu and Sastry, 1986]. The main drawback of the computed torque technique is its heavy dependence on the model reference. Many present industrial robots, however, still use a simple PID control law in a straight forward manner.

## 3.9 Conclusions

In this chapter, the foundation of all aspects in developing a mathematical model of an industrial robot which has a revolute joint system has been presented. A thorough understanding of the Denavit-Hartenberg concept which systematically gives a description of an industrial robot is necessary. This is obtained by understanding the D-H parameters and the coordinate system placement.

Based on the D-H concept, kinematics of an industrial robot can be developed. This involves the use of the D-H transformation matrix which describes the position and orientation of the robot hand. The differential change of the robot arm configuration with respect to a change of the joint variables has been presented.

Having obtained the kinematic information, the dynamic equations which give the dynamic behaviour of the robot can be developed. The accuracy and computational efficiency of a robot dynamic model plays an important role in a simulation. In this chapter, the robot arm dynamics has been developed with some assumptions, ie. all links are rigid and all motion are frictionless. The N-E approach has been chosen by the author in developing the robot arm dynamics since it is efficient in computation and relatively easy to analyse all forces and torques which exist in the system.

For the sake of a general understanding of a robot manipulator system, trajectory planning has been discussed briefly in term of the joint coordinates. In order to give a smooth continuous trajectory, a polynomial with the lowest order of a third order polynomial is used. In the case of pick and place like tasks, the trajectory is divided into three, the picking motion, the travelling motion and the placing motion. For this kind of task, a 4-3-4 trajectory has been presented.

As actuators are part of the system, it is necessary to develop a complete integrated mathematical model prior to designing a control system. The basic idea of the so called computed torque technique has been given to overcome nonlinearity problems in the system.

# CHAPTER 4

# THE DISTORTION TECHNIQUE

## 4.1 Introduction

The confidence in using a model for any purpose relies on the model validation technique which is used during the model development. Clearly, it is important to be able to justify the validity of the model for its intended purposes. Most validation techniques rely on a visual comparison while it is ambiguous how close a model response can be said to be close to a recorded response. The model validity is often assessed by a quantitative means such as the minimum squared error, minimum peak differences etc. These approaches also still exhibit an ambiguity since there is no certain rule which indicates how small a true minimum value should be. Also, by deriving the best fit parameters only, the model tends to be assumed as a black box model and not based on physical principles. Hence, the judgement of a model validity is frequently subjective.

In validating a model, the model parameter uncertainties should be taken into account. Any model can be made to match the measured transient response by distorting its parameters. If the distortion required to match the transient is acceptable in terms of known approximations in the model, then the model is considered capable of explaining the measured transient ie. the model is valid in terms of the model uncertainties. This is the idea which stands behind the distortion technique for validating a model quantitatively [Butterfield, Thomas, 1983], and also this technique is applicable to non-linear systems. This chapter describes the distortion technique which gives a relatively objective assessment in validating a model quantitatively.

The distortion quantitative validation technique was first developed at the United Kingdom Atomic Energy Authority, Winfrith, England [Butterfield, Thomas, 1983] and was applied to nuclear power plant models. Some institutions, eg. City University, London, do some research work for other applications.

## 4.2 Quantitative Model Validation Based on The Distortion Technique [Butterfield and Thomas, 1983]

This method is based on the preposition that a mathematical model can be made to match the plant transient response by varying the model parameters as functions of time. So that, the less the distortion, the better the model.

There are two approaches of the distortion technique :

1. Direct solution in the time domain.
2. Approximate solution using the transfer function concept.

A mathematical model of a plant can be represented as,

$$\dot{x} = f( x, u, a, t ) \qquad (4.1)$$

where : $x$ = a state vector in n dimension.

$u$ = an input vector in m dimension.

$a$ = a constant parameter vector in p dimension.

$f$ = a linear or nonlinear vector function.

and for a plant model with k output variables,

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \qquad (4.2)$$

where : **C** = a kxn coefficient matrix.

**D** = a kxm coefficient matrix.

**Y** = a kx1 output vector.

The residual error vector between the plant and the model is given by,

$$e(t) = z(t) - y(a,t) \qquad (4.3)$$

where : **z** = a vector of the measured output variables in k dimension.

To quantify the match between a model and a plant, a performance index, which is a function of the parameter vector ' **a** ', is defined by,

$$I = \frac{1}{T} \int_0^T \mathbf{e}(\mathbf{a}, t)^T \mathbf{W}\, \mathbf{e}(\mathbf{a}, t) \qquad (4.4)$$

where : **W** = a weighting matrix.

T = an observation time.

T = transpose operator.

For convenience, a performance index should be easily computed.

After minimising the performance index, given by equation (4.4), a fitted model is defined by,

$$\hat{\mathbf{x}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}, \hat{\mathbf{a}}, t) \qquad (4.5)$$

$$\hat{y} = C\hat{x} + Du \tag{4.6}$$

where ' ^ ' denotes the optimal condition.

So that, equation (4.3) can be written as,

$$e_{min}(t) = \hat{z}(t) - \hat{y}(\hat{a},t) \tag{4.7}$$

where $e_{min}$ is the residual error function and its norm has been minimised, at the optimal condition.

If the model parameters are varied as functions of time, the new model equations are given by,

$$\dot{X} = f(X, u, \hat{a} + \alpha(t), t) \tag{4.8}$$

$$Y = CX + Du \tag{4.9}$$

where : X = a distorted state vector.

$\alpha(t)$ = a time dependent parameter variation vector for distorting the model.

Y = a distorted output vector.

In most cases, it will be possible to drive the system with these parameter variations, so that the measured states of the plant are matched by the distorted state vector of the model [Butterfield and Thomas, 1983].

For a perfect model-plant match, equations (4.8) and (4.9) can be written as,

$$\dot{X}^{o} = f( X^{o}, u, \hat{a} + \alpha^{o}(t), t ) \qquad (4.10)$$

$$Y^{o} = CX^{o} + Du \qquad (4.11)$$

where : $X^{o}$ = a matched distorted state vector.

$\alpha^{o}(t)$ = a time dependent model parameter variation vector to obtain a perfect model-plant match.

$Y^{o}$ = a matched distorted output vector of the model.

and $'^{o}$ denotes a perfect match.

So that,

$$Y^{o}(t) = z(t) \qquad (4.12)$$

which implies a perfect model-plant matching. Hence, in the model distortion technique, a quantitative model validation is performed by judging the time dependent vector $\alpha^{o}(t)$.

### 4.3 Time Domain Approach

This method involves a direct computation in obtaining a time dependent model parameter variation vector, $\alpha^{o}(t)$, required to eliminate the residual error between the model and the plant. For a reasonable model, which implies that the parameter variations $\alpha^{o}$ should be small, equation (4.8) can be linearised using a Taylor

series theorem as,

$$\dot{X}^{\circ} = f( X^{\circ}, u, \hat{a}, t ) + J_a \alpha^{\circ} \tag{4.13}$$

$$Y^{\circ} = CX^{\circ} + Du \tag{4.14}$$

where $J_a$ is a Jacobian matrix in nxp dimension ie.

$$J_a = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \cdots & \frac{\partial f_1}{\partial a_p} \\ \frac{\partial f_2}{\partial a_1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial a_1} & \cdots & \cdots & \frac{\partial f_n}{\partial a_p} \end{bmatrix} \tag{4.15}$$

evaluated at $a = \hat{a}$

Differentiating equation (4.14) gives,

$$\dot{Y}^{\circ} = C\dot{X}^{\circ} + D\dot{u} \tag{4.16}$$

Substituting equation (4.13) into equation (4.16),

$$\dot{Y}^{\circ} = Cf( X^{\circ}, u, \hat{a}, t ) + CJ_a \alpha^{\circ} + D\dot{u} \tag{4.17}$$

and rearranging,

$$CJ_{\hat{a}}\alpha^{\circ} = \dot{Y}^{\circ} - ( Cf( X^{\circ}, u, \hat{a}, t ) + D\dot{u} ) \qquad (4.18)$$

Differentiating equation (4.12) and substituting into equation (4.18) yields,

$$CJ_{\hat{a}}\ \alpha^{\circ} = \dot{z} - ( Cf( X^{\circ}, u, \hat{a}, t) + D\dot{u} ) \qquad (4.19)$$

Other than $\alpha^{\circ}$, all terms in equation (4.19) are known.

Equation (4.19) can be written in a simplified form, ie.

$$D\alpha^{\circ} = \dot{z} - b \qquad (4.20)$$

where,

$$D = CJ_{\hat{a}} \ ; \ \text{a kxp matrix} \qquad (4.21)$$

$$b = Cf( X^{\circ}, u, \hat{a}, t ) + D\dot{u} \ ; \ \text{a kx1 vector} \qquad (4.22)$$

Provided k = p, equation (4.20), which is a set of linear equations, can be solved to obtain the values of $\alpha^{\circ}$ at each time interval. In the case where k < p, an optimisation procedure should be used, since the solution is not unique.

### 4.3.1 Criterion For Model Fidelity

By solving equation (4.20) during an observation time, T, the distribution of each time dependent model parameter variation to obtain a perfect match $\alpha_i^o(t)$, i=1 to p can be found and a typical graphical representation of $\alpha_i^o$ is shown in figure (4.1).

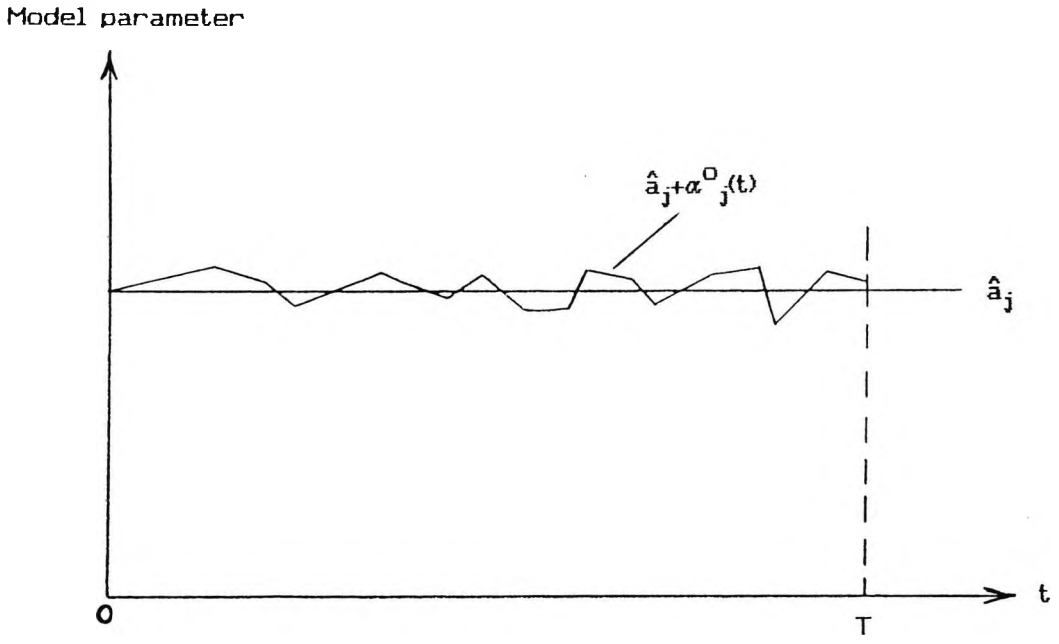



Figure 4.1

Distribution of time dependent model parameter variation

The mean squared value or variance of each parameter variation $\sigma^2_{\alpha^o_j}$ about its optimal value can be evaluated during an observation time, ie.

$$\sigma^2_{\alpha^o_j} = \frac{1}{T}\int_0^T (\alpha^o_j)^2\, dt \quad ; \quad \text{for } j = 1 \text{ to } p \tag{4.23}$$

The variances of all p model parameters given by equation (4.23) are then required to eliminate the error between the model response and the plant response. The basic concept in judging whether or not the model is capable of explaining the recorded transient in term of the parameter uncertainties is by observing the

variance of each model parameter. In order for the model to be considered valid, the variance of each model parameter, $\sigma^2_{\alpha^o_j}$ , should be less than or equal to its expected variance, $\tau'^2_j$ , ie.

$$\sigma^2_{\alpha^o_j} \leq \tau'^2_j \quad ; \quad \text{for } j = 1 \text{ to } p \tag{4.24}$$

Hence, provided the condition stated in equation (4.24) is satisfied, the model is considered capable of explaining the recorded transient in term of the parameter uncertainties.

## 4.4 Transfer Function Approach

For a reasonable model, where $\mathbf{X}$ does not differ greatly from $\mathbf{x}$ and $\boldsymbol{\alpha}$ is small, equation (4.8) can be approximated using a Taylor series theorem.

$$\dot{\mathbf{X}} = \mathbf{f}( \hat{\mathbf{x}}, \mathbf{u}, \hat{\mathbf{a}}, t ) + \mathbf{J}_x( \mathbf{X} - \hat{\mathbf{x}} ) + \mathbf{J}_a \boldsymbol{\alpha} \tag{4.25}$$

with the assumption, that, the vector function $\mathbf{f}$ is continuous in the vicinity of $\hat{\mathbf{x}}$ and $\hat{\mathbf{a}}$. $\mathbf{J}_x$ and $\mathbf{J}_a$ are the Jacobian matrices, evaluated at $\hat{\mathbf{x}}$ and $\hat{\mathbf{a}}$, respectively ie.

$$\mathbf{J}_x = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{4.26}$$

$$J_a = \begin{bmatrix} \frac{\partial f_1}{\partial a_1} & \frac{\partial f_1}{\partial a_2} & \cdots & \frac{\partial f_1}{\partial a_p} \\ \frac{\partial f_2}{\partial a_1} & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial a_1} & \cdots & \cdots & \frac{\partial f_n}{\partial a_p} \end{bmatrix} \qquad (4.27)$$

The state difference vector between the undistorted model and the distorted model is defined by,

$$\boldsymbol{\phi} = \mathbf{X} - \mathbf{x} \qquad (4.28)$$

and hence,

$$\dot{\boldsymbol{\phi}} = \dot{\mathbf{X}} - \dot{\mathbf{x}} \qquad (4.29)$$

When a system is said to be controllable at time $t_o$, it is possible by means of an unconstrained input to transfer the system from initial state $x(t_o)$ to any other state in a finite time interval [Ogata, 1981]. Provided that the system is controllable, a perfect model-plant match can be achieved by introducing the $\boldsymbol{\alpha}$ vector variations in the absence of any limit on the amplitude or frequency of variation of $\boldsymbol{\alpha}$.

Hence, for a perfect model-plant match,

$$\dot{X}^{\circ} = f(\hat{x}, u, \hat{a}, t) + J_x(X^{\circ} - \hat{x}) + J_a \alpha^{\circ} \tag{4.30}$$

$$Y^{\circ} = CX^{\circ} + Du \tag{4.31}$$

$$\phi^{\circ} = X^{\circ} - \hat{x} \tag{4.32}$$

$$\dot{\phi}^{\circ} = \dot{X}^{\circ} - \dot{\hat{x}} \tag{4.33}$$

Substituting equations (4.32) and (4.33) into equation (4.30) yields,

$$\dot{\hat{x}} + \dot{\phi}^{\circ} = f(\hat{x}, u, \hat{a}, t) + J_x \phi^{\circ} + J_a \alpha^{\circ} \tag{4.34}$$

Since, from equation (4.5), $\dot{\hat{x}} = f(\hat{x}, u, \hat{a}, t)$ then,

$$\dot{\phi}^{\circ} = J_x \phi^{\circ} + J_a \alpha^{\circ} \tag{4.35}$$

$$\psi^{\circ} = C\phi^{\circ} \tag{4.36}$$

The state equation, given by (4.35), is called the associate system where the time dependent model parameter variation vector $\alpha^{\circ}(t)$ and $\psi^{\circ}(t)$ are the input and output, respectively.

Based on equations (4.7) and (4.12),

$$\psi^{\circ}(t) = e_{min}(t) \tag{4.37}$$

so that,

$$\psi_i(t) = e_{min_i}(t) \tag{4.38}$$

The associate system, given by equations (4.35) and (4.36) has the transfer function matrix,

$$G(s) = C\,(sI_n - J_x)^{-1}\,J_a \tag{4.39}$$

where $I_n$ is an nxn identity matrix.

Hence,

$$\boldsymbol{\psi}(s) = G(s)\,\boldsymbol{\alpha}(s) \tag{4.40}$$

where G(s) is the transfer function matrix of the associate system in kxp dimension.

In a general form, equation (4.40) can then be written as,

$$\begin{bmatrix} \psi_1(s) \\ \psi_2(s) \\ \cdots \\ \psi_k(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) & \cdots & g_{1p}(s) \\ g_{21}(s) & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ g_{k1}(s) & \cdots & \cdots & g_{kp}(s) \end{bmatrix} \begin{bmatrix} \alpha_1(s) \\ \alpha_2(s) \\ \cdots \\ \alpha_p(s) \end{bmatrix} \tag{4.41}$$

### 4.4.1 Single Measured Variable Case

For a case in which there is one overridingly important variable which is available from a recorded measurement, equation (4.41) is reduced to

$$\begin{bmatrix} \psi_i(s) \end{bmatrix} = \begin{bmatrix} g_{i1}(s) & g_{i2}(s) & \cdots\cdots & g_{ip}(s) \end{bmatrix} \begin{bmatrix} \alpha_1(s) \\ \alpha_2(s) \\ \cdots\cdots \\ \alpha_p(s) \end{bmatrix} \qquad (4.42)$$

where $\psi_i$ is the output of the associate system of interest.

For a well developed model, the dynamic behaviour of the output of the associate system, $\psi_i$ , which is driven by each model parameter acting alone, can be assumed as a second order system. Hence,

$$g_{ij}(s) = \frac{\psi_i(s)}{\alpha_j(s)} = \frac{c_{ij}}{\frac{s^2}{\omega_{ij}^2} + 2\frac{\zeta_{ij}}{\omega_{ij}} s + 1} \qquad (4.43)$$

The input of the associate system, $\alpha_j$, is assumed to be represented by filtered white noise and the filter is given by

$$F(s) = \frac{c_f}{1 + \frac{s}{\omega_f}} \qquad (4.44)$$

where : $\omega_f$ = cut off frequency

$c_f$ = filter gain

By this condition, the parameter varies randomly about the optimal constant value $\hat{a}$ and its variation has a limited bandwidth. The filtered white noise itself with an intensity $\Phi$, then acts as a signal $\alpha_j$ of variance $\sigma^2\alpha_j$ . This signal in turn drives the associate system to produce the signal $\psi_i$ , which has a variance $\sigma^2\psi_i$ . Figures (4.2) and (4.3) show the associate system.

From figure (4.3), the variance of the input associate system, $\alpha_j$, can be evaluated in term of $\Phi$ as

$$\sigma^2\alpha_j = \frac{\Phi}{2\pi j} \int_{-j\infty}^{+j\infty} \left| \frac{c_f}{1 + \frac{s}{\omega_f}} \right|^2 ds \qquad (4.45)$$

Integrating equation (4.45) yields [Douce, 1963],

$$\sigma^2\alpha_j = \frac{c_f^2\, \omega_f}{2}\, \Phi \qquad (4.46)$$

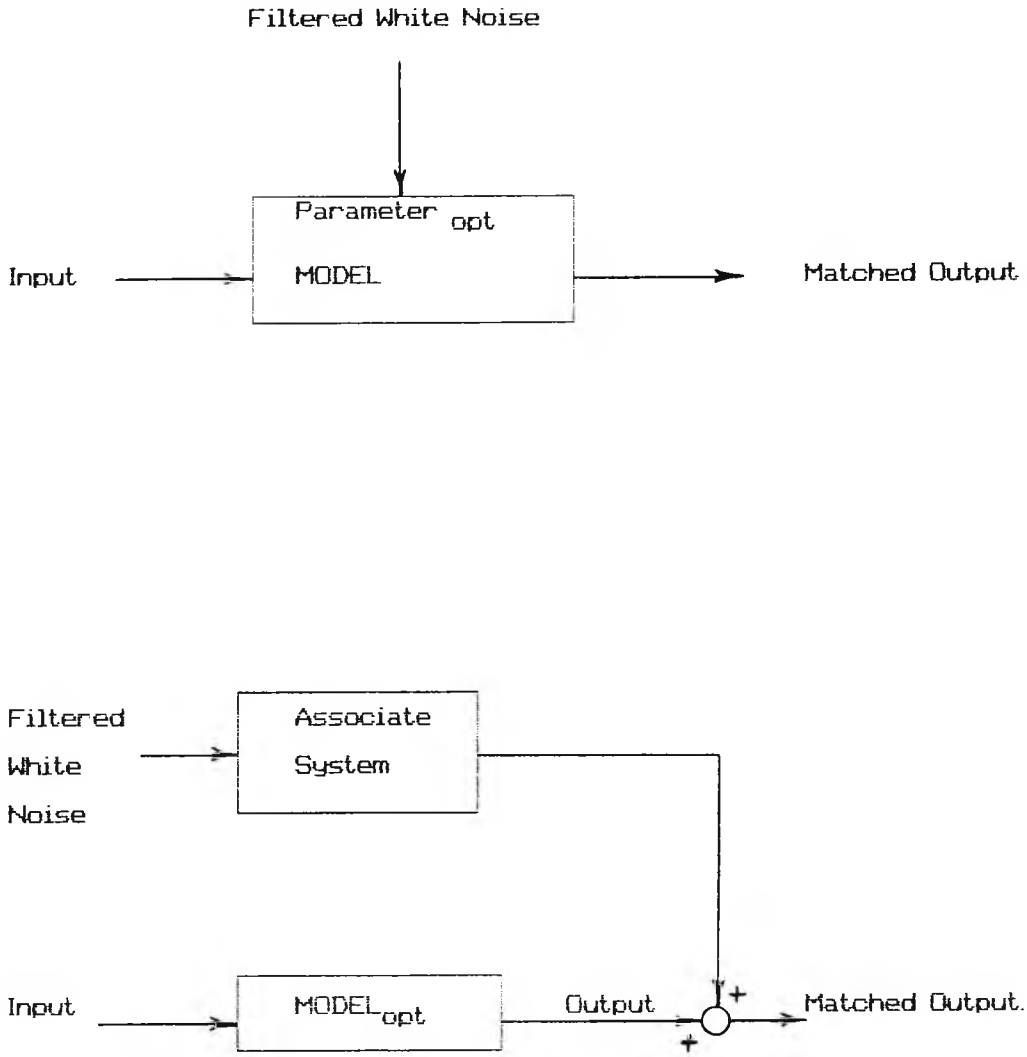



Figure 4.2

The associate system

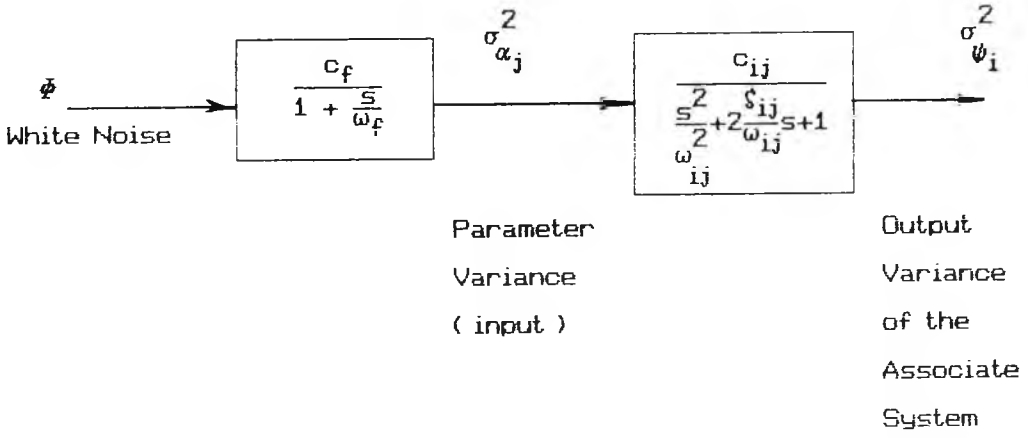



Figure 4.3

The associate system driven by the white noise

Using the same principle, the output of the associate system $\psi_i$ can be related to $\Phi$ as

$$\sigma^2_{\psi_i} = \frac{\Phi}{2\pi j} \int_{-j\infty}^{+j\infty} \left| \frac{c_f}{1 + \frac{s}{\omega_f}} \quad \frac{c_{ij}}{\frac{s^2}{\omega_{ij}^2} + 2\frac{\zeta_{ij}}{\omega_{ij}} s + 1} \right|^2 ds$$

$$= \frac{\Phi}{2\pi j} \int_{-j\infty}^{+j\infty} \left| \frac{c_f \, c_{ij}}{\frac{s^3}{\omega_f \omega_{ij}^2} + \frac{2\zeta\omega_{ij} + \omega_f}{\omega_f \omega_{ij}^2} + \frac{2\zeta\omega_f\omega_{ij} + \omega_{ij}^2}{\omega_f \omega_{ij}^2} + 1} \right|^2 ds \quad (4.47)$$

Integrating equation (4.47) yields [Butterfield and Thomas, 1983]

$$\sigma^2{}_{\psi_i} = \frac{c_f{}^2\, c_{ij}{}^2 \left(2\varsigma\omega_{ij} + \omega_f\right)}{4\varsigma\left(2\varsigma + \frac{\omega_{ij}}{\omega_f} + \frac{\omega_f}{\omega_{ij}}\right)} \quad \Phi \tag{4.48}$$

The relationship between $\sigma^2{}_{\alpha_j}$ and $\sigma^2{}_{\psi_i}$ is then obtained by dividing equation (4.46) by equation (4.48) which is given as

$$\frac{\sigma^2{}_{\alpha_j}}{\sigma^2{}_{\psi_i}} = \frac{2\varsigma_{ij}\left(2\varsigma_{ij} + \frac{\omega_{ij}}{\omega_f} + \frac{\omega_f}{\omega_{ij}}\right)}{c^2{}_{ij}\,(2\varsigma_{ij} + 1)} \tag{4.49}$$

With the assumption that the frequency contents of the input signal, $\alpha_j$, and the output signal, $\psi_i$, are roughly the same such that $\omega_f \simeq \omega_{ij}$, the relationship between the parameter variance, $\sigma^2{}_{\alpha_j}$, and the output variance, $\sigma^2{}_{\psi_i}$, can be written as

$$\frac{\sigma^2{}_{\alpha_j}}{\sigma^2{}_{\psi_i}} = \frac{4\varsigma_{ij}\,(\varsigma_{ij} + 1)}{c^2{}_{ij}\,(2\varsigma_{ij} + 1)} \tag{4.50}$$

where $\varsigma_{ij}$ depends on the shape of the minimum residual error and for most systems, $\varsigma_{ij}$ can be approximated to a value of 0.5 [Butterfield, 1989].

For a single i-th measured variable case, equation (4.7) can be rewritten as,

$$e_{min_i}(t) = z_i(t) - \hat{y}_i(\hat{\mathbf{a}},t) \tag{4.51}$$

The steady state gain of the associate system, $c_{ij}$ , can be evaluated by considering the residual error as given by equation (4.3). Equation (4.3) can be rewritten as ,

$$e_i(a,t) = z_i(t) - y_i( \hat{a} + \alpha_{js} , t ) \qquad (4.52)$$

where $\alpha_{js}$ is a constant value of parameter variation about $\hat{a}_j$.

From equation (4.43), the steady state condition of the associate system is obtained by,

$$\psi_{is} = c_{ij}\alpha_{js} \qquad (4.53)$$

where the subscript 's' indicates the steady state condition.

This yields, from equations (4.51) and (4.52), an error function

$$e_i(a,t) = z_i(t) - \hat{y}_i(\hat{a},t) + c_{ij}\alpha_{js} \qquad (4.54)$$

so that,

$$e_i(a,t) = e_{min_i}(t) + c_{ij}\alpha_{js} \qquad (4.55)$$

The mean squared error is given by,

$$\overline{e^2}_i = \frac{1}{T} \int_0^T ( e_{min_i} + c_{ij}\alpha_{js} )^2 \, dt \qquad (4.56)$$

$$= \frac{1}{T} \int_0^T e^2_{min_i} \, dt + \frac{1}{T} \int_0^T 2 \, e_{min_i} \, c_{ij} \, \alpha_{js} \, dt + \frac{1}{T} \int_0^T c_{ij}^2 \, \alpha_{js}^2 \, dt$$

Since for a linear system, $\frac{1}{T} \int_0^T e_{min_i} \, dt = 0$ , [Butterfield, Thomas, 1983], then

$$\overline{e^2}_i = \overline{e^2}_{min_i} + c_{ij}^2 \, \alpha^2_{js} \tag{4.57}$$

Hence,

$$c_{ij}^2 = \frac{\overline{e^2}_i - \overline{e^2}_{min_i}}{\alpha^2_{js}} \tag{4.58}$$

where the value of $\overline{e^2}_{min_i}$ was already obtained during the minimisation process. A typical mean squared error curve is shown in figure (4.4).

If, from the graph, $\alpha_{js} = \alpha_{jd}$ is selected where $\overline{e^2}_i$ is double $\overline{e^2}_{min_i}$ , equation (4.58) then becomes,

$$c^2_{ij} = \frac{\overline{e^2}_{min_i}}{\alpha^2_{jd}} \tag{4.59}$$

Since a mean squared error, usually, is non-symetrical, equation (4.59) can be written as,

$$c^2_{ij} = \frac{\overline{e^2}_{min_i}}{\left( \frac{\alpha_{jd-} + \alpha_{jd+}}{2} \right)^2} \tag{4.60}$$

where $\alpha_{jd-}$ and $\alpha_{jd+}$ are the descending and the ascending parameter variations to double the minimum mean squared error, respectively.

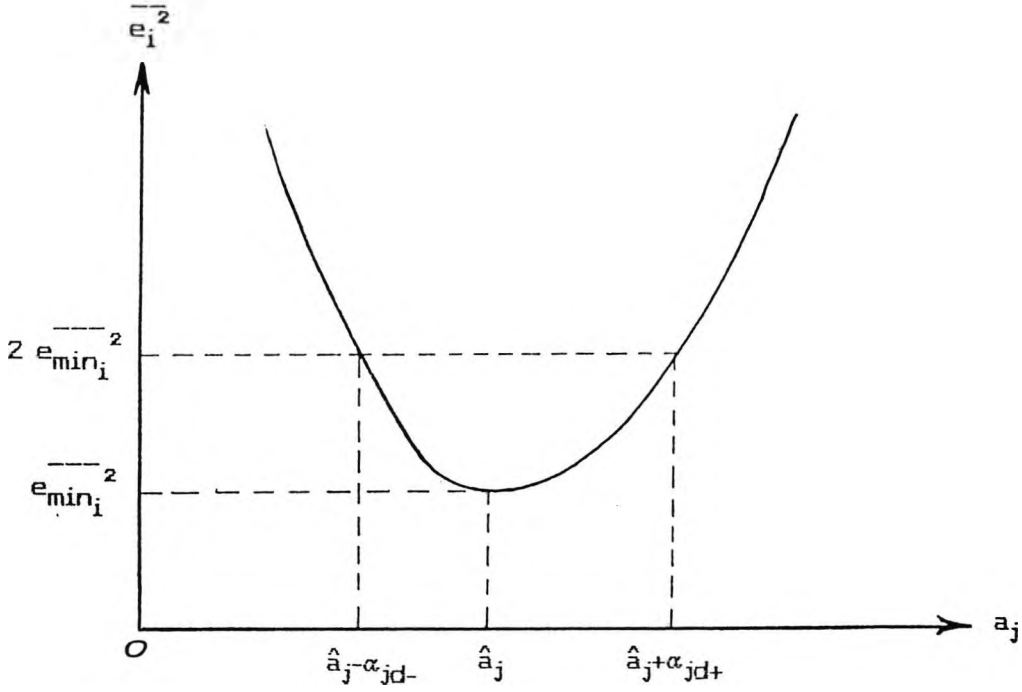



Figure 4.4

A typical mean squared error curve

Combining equations (4.50) and (4.60) yields,

$$\frac{\sigma^2_{\alpha_j}}{\sigma^2_{\psi_i}} = \frac{4\zeta_{ij}\,(\,\zeta_{ij}+1\,)}{(\,2\zeta_{ij}+1\,)}\;\frac{\left(\dfrac{\alpha_{jd-}+\alpha_{jd+}}{2}\right)^2}{\overline{e^2}_{min_i}} \qquad (4.61)$$

As in the perfect plant-model match, the output of the matched associate system is

equal to the minimum residual error and, in addition, the variance of the matched associate system is identical with the mean squared value of the output of the matched associate system, so that

$$\sigma^2_{\psi^o_i} = \overline{e^2}_{min_i} \qquad (4.62)$$

The input parameter variance is then given by,

$$\sigma^2_{\alpha^o_j} = \frac{4\zeta_{ij}(\zeta_{ij}+1)}{(2\zeta_{ij}+1)} \left(\frac{\alpha_{jd-}+\alpha_{jd+}}{2}\right)^2 \qquad (4.63)$$

Hence, $\sigma^2_{\alpha^o_j}$ is the required j-th parameter variance to eliminate the plant-model mismatch.

### 4.4.2 Multiple Measured Variable Case

It is not always possible to find a system which has one overridingly important measured variable. In this case, provided the plant has been carefully modelled, the distortion technique based on the transfer function approach can be applied simultaneously to all k measured variables. A combination of the mean squared error values, ie. $\overline{e^2}_i$ for i = 1 to k, is used as a performance index to fit the model.

For the sake of clarity, a multiple measured variable case is represented in a two measured variable case, but this approach extends to any finite number of measured variables. In this case, all associate systems, given by equation (4.43), are considered to have the same values of $\zeta$ and $\omega$, ie. $\zeta_{ij} = \zeta_j$ and $\omega_{ij} = \omega_j$ ; for i = 1 to 2.

When the model is in the optimal condition, then the residual errors

$$e_{min_1}(t) = z_1(t) - \hat{y}_1(\hat{a},t) \tag{4.64}$$

$$e_{min_2}(t) = z_2(t) - \hat{y}_2(\hat{a},t) \tag{4.65}$$

where $z_1$ and $z_2$ are the plant responses, and in the perfect plant-model match the associate system outputs

$$\psi^o_1(t) = e_{min_1}(t) \tag{4.66}$$

$$\psi^o_2(t) = e_{min_2}(t) \tag{4.67}$$

Hence,

$$\sigma^2_{\psi^o_1} = \overline{e^2}_{min_1} \tag{4.68}$$

$$\sigma^2_{\psi^o_2} = \overline{e^2}_{min_2} \tag{4.69}$$

$$\overline{e^2}_{min} = \overline{e^2}_{min_1} + \overline{e^2}_{min_2} \tag{4.70}$$

where $\overline{e^2}_{min}$ is the minimum total mean squared value.

If a constant value of the j-th parameter variation, $\alpha_{js}$, is applied about $\hat{a}_j$ to both associate systems, after the settling period

$$\psi_{1s} = c_{1j}\,\alpha_{js} \tag{4.71}$$

$$\psi_{2s} = c_{2j}\,\alpha_{js} \tag{4.72}$$

where, as before, the subscript ' s ' denotes the steady state condition.

Since,

$$e_1(a,t) = z_1(t) - y_1(\,\hat{\mathbf{a}} + \alpha_{js}\,,\,t\,) \tag{4.73}$$

$$e_2(a,t) = z_2(t) - y_2(\,\hat{\mathbf{a}} + \alpha_{js}\,,\,t\,) \tag{4.74}$$

where $\alpha_{js}$ is a constant value of parameter variation about $\hat{a}_j$ , then

$$e_1(a,t) = z_1(t) - y_1(\,\hat{\mathbf{a}}\,,\,t\,) + c_{1j}\,\alpha_{js} \tag{4.75}$$

$$e_2(a,t) = z_2(t) - y_2(\,\hat{\mathbf{a}}\,,\,t\,) + c_{2j}\,\alpha_{js} \tag{4.76}$$

Hence, for a single j-th parameter acting alone, equation (4.56) can be written as,

$$\overline{e^2} = \sum_{i=1}^{2} e^2{}_i = \frac{1}{T}\int_0^T \sum_{i=1}^{2} (\,e_{min_i} + c_{ij}\,\alpha_{js}\,)^2\,dt \tag{4.77}$$

$$= \frac{1}{T}\int_0^T \Big( (\,e_{min_1} + c_{1j}\,\alpha_{js}\,)^2 + (\,e_{min_2} + c_{2j}\,\alpha_{js}\,)^2 \Big)\,dt$$

$$= \frac{1}{T}\int_0^T e^2{}_{min_1}\,dt \;+\; \frac{1}{T}\int_0^T e^2{}_{min_2}\,dt$$

$$+ \; 2\, c_{1j}\, \alpha_{js}\, \frac{1}{T} \int_0^T e_{min_1}\, dt \; + \; 2\, c_{2j}\, \alpha_{js}\, \frac{1}{T} \int_0^T e_{min_2}\, dt$$

$$+ \; c^2_{1j}\, \alpha^2_{js}\, \frac{1}{T} \int_0^T dt \; + \; c^2_{2j}\, \alpha^2_{js}\, \frac{1}{T} \int_0^T dt$$

$$\overline{\overline{e^2}} = \sum_{i=1}^{2} \left( \frac{1}{T} \int_0^T e^2_{min_i}\, dt \; + \; 2\, c_{ij}\, \alpha_j\, \frac{1}{T} \int_0^T \hat{e}_i\, dt \; + \; c_{ij}{}^2\, \alpha^2_{js}\, \frac{1}{T} \int_0^T dt \right) \quad (4.78)$$

With the assumption, that, for a linear system, the second term of equation (4.78) at the optimal condition is negligible, equation (4.78) then becomes

$$\overline{\overline{e^2}} = \overline{\overline{e^2_{min_1}}} + \overline{\overline{e^2_{min_2}}} + c^2_{1j}\, \alpha^2_{js} + c^2_{2j}\, \alpha^2_{js} \quad (4.79)$$

Substituting equation (4.70) into equation (4.79) gives,

$$\overline{\overline{e^2}} = \overline{\overline{e^2_{min}}} + c^2_{1j}\, \alpha^2_{js} + c^2_{2j}\, \alpha^2_{js} \quad (4.80)$$

where,

$$c^2_{1j}\, \alpha^2_{js} + c^2_{2j}\, \alpha^2_{js} = c^2_j\, \alpha^2_{js} \quad (4.81)$$

as shown in figure (4.5). So that, equation (4.80) becomes,

$$\overline{\overline{e^2}} = \overline{\overline{e^2_{min}}} + c^2_j\, \alpha^2_{js} \quad (4.82)$$

Selecting $\alpha_{js} = \alpha_{jd}$, from figure (4.5), where $\overline{\overline{e^2}}$ is double $\overline{\overline{e^2}}_{min}$, yields

$$\overline{\overline{e^2}}_{min} = c^2_j \alpha^2_{jd} \tag{4.83}$$

The relationship between the variance of the j-th parameter and the variance of the outputs of the associate systems, based on equation (4.50), are

$$\sigma^2 \psi_1 = \frac{c^2_{1j} ( 2\zeta_j + 1 )}{4\zeta_j ( \zeta_j + 1 )} \sigma^2 \alpha_j \tag{4.84}$$

$$\sigma^2 \psi_2 = \frac{c^2_{2j} ( 2\zeta_j + 1 )}{4\zeta_j ( \zeta_j + 1 )} \sigma^2 \alpha_j \tag{4.85}$$

When a perfect-model plant matched is achieved, then from equations (4.68), (4.69) and (4.70)

$$\overline{\overline{e^2}}_{min} = \frac{( c^2_{1j} + c^2_{2j} ) ( 2\zeta_j + 1 )}{4\zeta_j ( \zeta_j + 1 )} \sigma^2 \alpha^\circ_j \tag{4.86}$$

Since $c^2_j = c^2_{1j} + c^2_{2j}$, ie. equation (4.81), equation (4.86) then becomes,

$$\overline{\overline{e^2}}_{min} = \frac{c^2_j ( 2\zeta_j + 1 )}{4\zeta_j ( \zeta_j + 1 )} \sigma^2 \alpha^\circ_j \tag{4.87}$$

Substituting equation (4.83) into equation (4.87) and rearranging yields,

$$\sigma^2 \alpha^\circ_j = \frac{4\zeta_j ( \zeta_j + 1 )}{( 2\zeta_j + 1 )} \alpha^2_{jd} \tag{4.88}$$

where $\alpha_{jd}$ is the average value of the descending and the ascending parameter

variations due to the non-symetrical curve of $\overline{e^2}$.

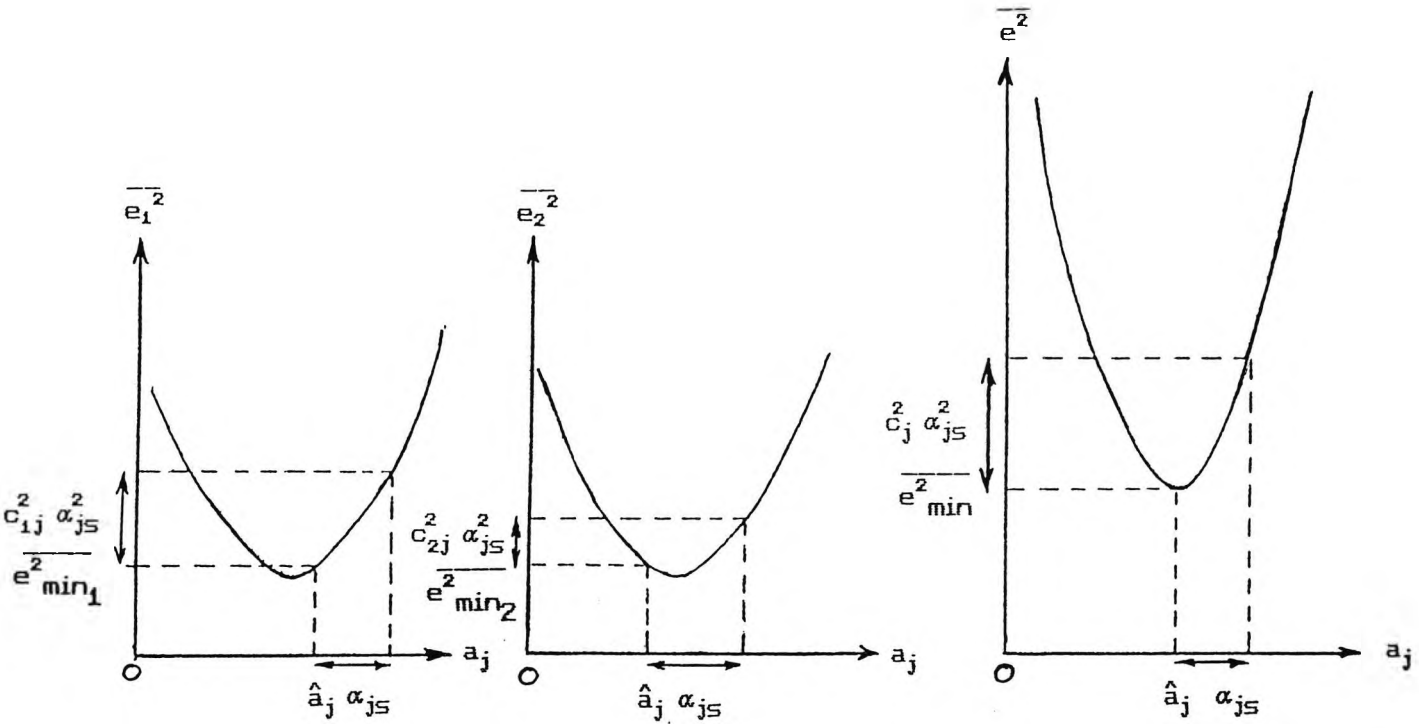



Figure 4.5

A mean squared error curve for a system with 2 measured variables

### 4.4.3 Criterion For Model Fidelity

As in the time domain approach, the model is considered capable of explaining the recorded transient if

$$\sigma^2_{\alpha^o_j} \leq \tau'^2_j \quad ; \text{for } j = 1 \text{ to } p \tag{4.89}$$

where $\tau'^2_j$ is the expected variance of the j-th parameter.

In the case where p parameters act simulataneously, only a fraction, $d_j$ , of the j-th parameter variance, $\sigma^2_{\alpha^o_j}$ is needed to eliminate the mismatch, ie.

$$d_j \sigma^2_{\alpha^o_j} \leq \tau'^2_j \quad ; \quad \text{for } j = 1 \text{ to } p \tag{4.90}$$

where,

$$\sum_{j=1}^{p} d_j = 1 \tag{4.91}$$

Defining,

$$\lambda^2_j = \frac{\tau'^2_j}{\sigma^2_{\alpha^o_j}} \tag{4.92}$$

and substituting equations (4.91) and (4.92) into equation (4.90) yields,

$$\lambda^2 = \sum_{j=1}^{p} \lambda^2_j \geq 1 \tag{4.93}$$

Equation (4.93) is then regarded as the fidelity criterion for the case where p parameters act simultaneously.

## 4.5 Advantages And Disadvantages of the Two Approaches

### 4.5.1 Time Domain Approach

Advantages :

1. In theory, this method is relatively more accurate.

Disadvantages :

1. The Jacobian matrix, $J_a$, must be evaluated at every time interval and it is not always easy in many cases to obtain this matrix.

2. An optimisation procedure might be needed at every time interval, if the number of parameters is more than the number of the first order differential equations, which have the information about the parameters.

3. Information is lost between each sampling time, hence an interpolation procedure should be performed.

4. Although in theory it is relatively more accurate, in practice differentiations are susceptible to noise.

5. It is difficult to implement.

### 4.5.2 Transfer Function Approach

Advantages :

1. It is more practical than the time domain approach, hence it is easy to implement.

Disadvantages :

1. It depends on the approximation of the second order associate systems and the white noise to represent the variance of each parameter.

## 4.6 Conclusions

The distortion technique quantitative validation technique has been outlined in this chapter. Both the time domain approach and the transfer function approach have been outlined. In both approaches, the basis of the technique is in the calculation of the amount of the distorted model parameters which are required to

eliminate the error between the model response and the plant response. The mathematical derivations of this technique have been described in detail. Advantages and disadvantages of the two approaches have been given.

In order to be applicable to a robot system which is multivariable and non-linear, the transfer function approach was extended to cope with the multiple measured variables case. An application of the transfer function domain approach in a robot system can be found in chapter 7. The transfer function domain approach is selected due to its ease of use and practicality in implementing the technique.

# CHAPTER 5

# IMPLEMENTATION OF THE DISTORTION TECHNIQUE

## 5.1 Introduction

Commonly, an industrial robot manipulator has six degrees of freedom where the first three degrees of freedom provide the position of the hand and the last three degrees of freedom give an orientation of the hand in performing a task. In this research work, however, due to the complexity of a six degree of freedom robot manipulator, only the first three degrees of freedom are considered. As the first three degrees of freedom give a major part of the operation, the dynamic behaviour and control system analysis can be sufficiently performed on them without losing the general understanding.

In implementing the distortion quantitative validation technique, one needs to know the importance of the parameters of the system. Thus, the physical behaviour of the system has to be studied before hand with respect to parameter changes. This is because some parameters are redundant in the sense that if some parameters take different values, the dynamic behaviour of the robot does not change. Knowledge about redundant parameters is important for parameter estimation and for obtaining an efficient inverse dynamics algorithm.

Neuman and Khosla [1985a, 1985b] stated some properties involving parameter identification in robot dynamics. Some work were carried out in identifying the non-redundant robot parameters [Mayeda, Yoshida and Osuka, 1988; Mayeda, Yoshida and Ohashi, 1989; Gautier and Khalil, 1989; Sheu and Walker, 1989]. This chapter presents the analysis of the robot parameters using the Newton-Euler approach.

This analysis then gives a basis in selecting important parameters in order to implement the distortion quantitative validation technique to a robot system.

Prior to performing a distortion quantitative validation technique which involves many repetitive tasks in exercising the model, it is appropriate to mention that the procedure in developing a simulation program is also important. Thus, the development of the simulation program is discussed in this chapter as well.

## 5.2 System Behaviour to Parameter Changes

The very first step in implementing the distortion quantitative validation technique is to have a good understanding of the system itself in how the system reacts if some parameters are distorted. For a robot with three degrees of freedom, the inverse dynamic equation which was given in equation (3.40) can be rewritten in the following form (eliminating $(\theta)$ for brevity)

$$\begin{bmatrix} \tau_{11} \\ \tau_{12} \\ \tau_{13} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{23} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} +$$

$$\begin{bmatrix} H_{111} & H_{112} & H_{113} & H_{122} & H_{123} & H_{133} \\ H_{211} & H_{212} & H_{213} & H_{222} & H_{223} & H_{233} \\ H_{311} & H_{312} & H_{313} & H_{322} & H_{323} & H_{333} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1\dot{\theta}_1 \\ \dot{\theta}_1\dot{\theta}_2 \\ \dot{\theta}_1\dot{\theta}_3 \\ \dot{\theta}_2\dot{\theta}_2 \\ \dot{\theta}_2\dot{\theta}_3 \\ \dot{\theta}_3\dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} G_1 \\ G_2 \\ G_3 \end{bmatrix} \qquad (5.1)$$

The first, second and third terms are inertial, coriolis/ centrifugal and gravity torques, respectively. All the above coefficients which are functions of $\theta$ contain a combination of inertial parameters, where the inertial parameters for each link are mass, centre of mass and inertia tensor matrix elements [Sheu, Walker, 1989].

Figure (5.1) shows a three degree of freedom manipulator with revolute joints which is under consideration. In chapter 3, it was discussed that all forces and torques propagate from the distal link backward to the proximal link. Thus, the study of parameter sensitivity starts form the distal link which is link 3.

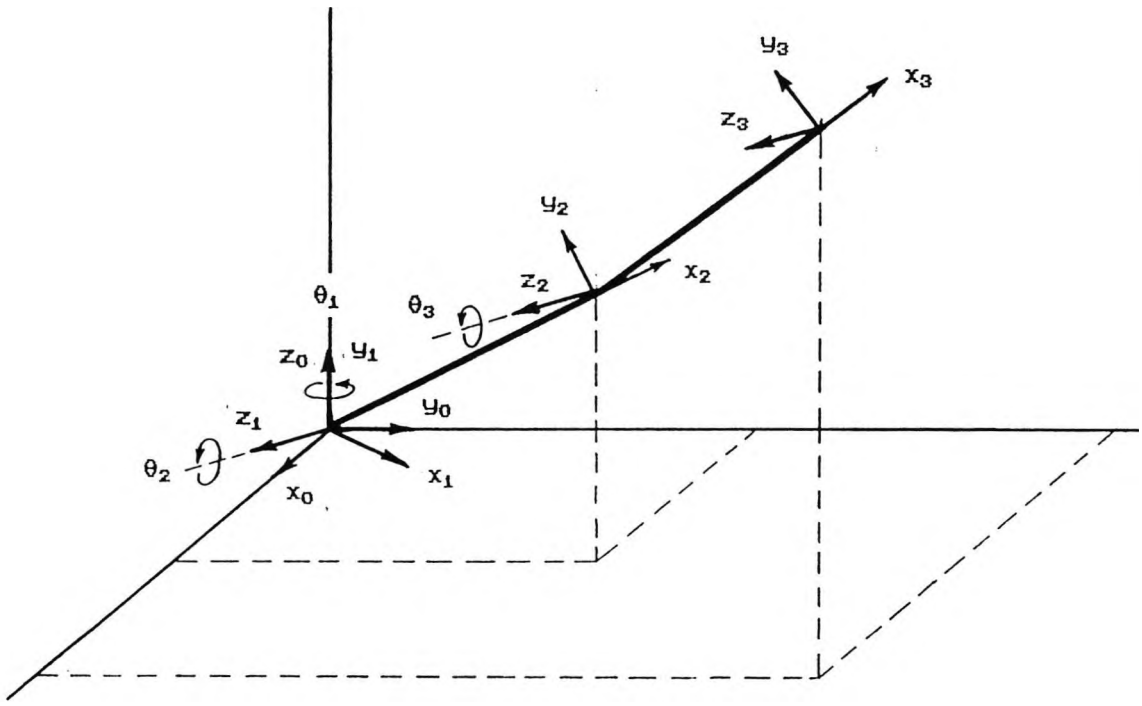



Figure 5.1

A three degree of freedom robot arm

As the forces and torques propagate from the distal link to the proximal link, not only do the joint variables $\theta$'s need to be considered, but also the twist angle $\alpha$'s of each link is equally important to be considered. From the Denavit-Hartenberg concept which was given in chapter 3, the twist angle value $\alpha_1$ between the $z_0$ and the $z_1$ axes is 90 degrees and the twist angle value $\alpha_2$ between the $z_1$ and the $z_2$ axes is 0 degree while the value of the twist angle $\alpha_3$ between the $z_2$ and the $z_3$ axes is not important in analysing the forces and torques propagation.

Let each link be isolated with its neighbouring links. With this condition, only the net forces/ torques and the forces/ torques at its own joint caused by the motion of its own link are considered. To make the problem easier in analysing the importance of each inertial parameter, the moment of inertia about the centre of mass is changed to about its own joint. By having the inertia tensor matrix evaluated about its own joint, the quadratic term of the centre of mass is eliminated [Neuman, Khosla, 1985a, 1985b].

Applying Newton and Euler equations to each link, the relationship of the torques/ forces and the inertial parameters on each link, based on An, Atkenson and Hollerbach [1985], with a modification can be expressed as

$$\begin{bmatrix} {}^{i}F_i \\ {}^{i}n_i - {}^{i}n_{i+1} - {}^{i}p_i \times {}^{i}f_{i+1} \end{bmatrix} = \begin{bmatrix} {}^{i}\dot{v}_{i-1} & \left({}^{i}\dot{\omega}_i \times\right) + \left({}^{i}\omega_i \times\right)\left({}^{i}\omega_i \times\right) & 0 \\ 0 & \left({}^{i}\dot{v}_{i-1} \times\right) & \left({}^{i}\dot{\omega}_i\right) + \left({}^{i}\omega_i \times\right)\left({}^{i}\omega_i\right) \end{bmatrix}$$

$$\begin{bmatrix} m_i \\ m_i\,{}^{i}c_{ix} \\ m_i\,{}^{i}c_{iy} \\ m_i\,{}^{i}c_{iz} \\ {}^{i}I^{*}_{ixx} \\ {}^{i}I^{*}_{ixy} \\ {}^{i}I^{*}_{ixz} \\ {}^{i}I^{*}_{iyy} \\ {}^{i}I^{*}_{iyz} \\ {}^{i}I^{*}_{izz} \end{bmatrix} \qquad (5.2)$$

where: ${}^{i}F_i$ = force at the centre of mass of link i expressed in the coordinate system i.

${}^{i}f_{i+1}$ = force at joint i+1 expressed in the coordinate system i.

${}^{i}n_i$ = torque at joint i expressed in the coordinate system i.

${}^{i}n_{i+1}$ = torque at joint i+1 expressed in the coordinate system i.

${}^{i}I^{*}_{i}$ = inertia matrix of link i about joint i and expressed in the

coordinate system i.

$m_i$ = mass of link i.

${}^{i}p_i$ = position of the origin of the coordinate system i from the origin of the coordinate system i-1 and expressed in the coordinate system i.

${}^{i}\bar{c}_i$ = position of the centre of mass of link i from the origin of the coordinate system i-1 and is expressed in the coordinate system i.

${}^{i}\omega_i$ = angular velocity of link i and expressed in the coordinate system i.

${}^{i}\dot{\omega}_i$ = angular acceleration of link i and expressed in the coordinate system i.

${}^{i}\dot{v}_{i-1}$ = linear acceleration of of the origin of the coordinate system i-1 and expressed in the coordinate system i.

Although the centre of mass ${}^{i}\bar{c}_{ix}$, ${}^{i}\bar{c}_{iy}$ and ${}^{i}\bar{c}_{iz}$ are not determined directly, their value can be factorized since $m_i$ is determined on its own [An, Atkenson and Hollerbach, 1985].

It is important to understand that since each link rotates about its own joint, only the torque component which is in the joint axis direction is necessary to be considered, ie. the generalized torque. If equation (5.2) is derived, it unleashes the significance of each inertial parameter with its relation to all kinematic variables as the following

$$
\begin{bmatrix} {}^{i}F_{i_x} \\ {}^{i}F_{i_y} \\ {}^{i}F_{i_z} \\ {}^{i}nn_{i_x} \\ {}^{i}nn_{i_y} \\ {}^{i}nn_{i_z} \end{bmatrix} = \begin{bmatrix} {}^{i}\dot{V}_{i-1_x} & -\omega_z^2 - \omega_y^2 & -\dot{\omega}_z + \omega_y\omega_x & \dot{\omega}_y + \omega_z\omega_x & 0 & 0 & 0 & 0 & 0 & 0 \\ {}^{i}\dot{V}_{i-1_y} & \dot{\omega}_z + \omega_x\omega_y & -\omega_z^2 - \omega_x^2 & -\dot{\omega}_x + \omega_z\omega_y & 0 & 0 & 0 & 0 & 0 & 0 \\ {}^{i}\dot{V}_{i-1_z} & -\dot{\omega}_y + \omega_x\omega_z & \dot{\omega}_x + \omega_y\omega_z & -\omega_y^2 - \omega_x^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & {}^{i}\dot{V}_{i-1_z} & -{}^{i}\dot{V}_{i-1_y} & \dot{\omega}_x & \dot{\omega}_y - \omega_z\omega_x & \dot{\omega}_z + \omega_y\omega_x & -\omega_z\omega_y & -\omega_z^2 + \omega_y^2 & \omega_y\omega_z \\ 0 & -{}^{i}\dot{V}_{i-1_z} & 0 & {}^{i}\dot{V}_{i-1_x} & \omega_z\omega_x & \dot{\omega}_x + \omega_z\omega_y & \omega_z^2 - \omega_x^2 & \dot{\omega}_y & \dot{\omega}_z - \omega_x\omega_y & -\omega_x\omega_z \\ 0 & {}^{i}\dot{V}_{i-1_y} & -{}^{i}\dot{V}_{i-1_x} & 0 & -\omega_y\omega_x & -\omega_y^2 + \omega_x^2 & \dot{\omega}_x - \omega_y\omega_z & \omega_x\omega_y & \dot{\omega}_y + \omega_x\omega_z & \dot{\omega}_z \end{bmatrix} \begin{bmatrix} m_i \\ m_i\,{}^{i}c_{x_i} \\ m_i\,{}^{i}c_{i_y} \\ m_i\,{}^{i}c_{i_z} \\ {}^{i}\overset{*}{I}_{i_{xx}} \\ {}^{i}\overset{*}{I}_{i_{xy}} \\ {}^{i}\overset{*}{I}_{i_{xz}} \\ {}^{i}\overset{*}{I}_{i_{yy}} \\ {}^{i}\overset{*}{I}_{i_{yz}} \\ {}^{i}\overset{*}{I}_{i_{zz}} \end{bmatrix} \tag{5.3}
$$

In equation (5.3), all other kinematic variables are subscripted and superscripted with i, and ${}^{i}nn_{i}$ is the torque at joint i due to motion of link i alone expressed in the coordinate system i. The components of $\overset{*}{I}$ are computed about joint i and expressed with respect to the coordinate system i. By observing equation (5.3) in conjunction with the N-E algorithm given in appendix B, how the system reacts if

parameters are distorted can then be analysed.

If only 1 joint is allowed to move at a time and the other joints are locked, in the case when link 2 or link 3 moves alone, only the angular kinematic variables in the directions of $z_2$ or $z_3$ axes exist in the system. When only joint 1 is allowed to move, only the angular kinematic variables in the direction of the $y_1$ exist in the system. Equations (5.4) to (5.9) express this behaviour, ie. when only 1 joint moves at a time.

$$ {}^{1}\omega_1 = \begin{bmatrix} 0 \\ \dot{\theta}_1 \\ 0 \end{bmatrix} \tag{5.4} $$

$$ {}^{2}\omega_2 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} \tag{5.5} $$

$$ {}^{3}\omega_3 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix} \tag{5.6} $$

$$ {}^{1}\dot{\omega}_1 = \begin{bmatrix} 0 \\ \ddot{\theta}_1 \\ 0 \end{bmatrix} \tag{5.7} $$

$$ {}^{2}\dot{\omega}_2 = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_2 \end{bmatrix} \tag{5.8} $$

$$ {}^{3}\dot{\omega}_{3} = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}_{3} \end{bmatrix} \qquad (5.9) $$

Consequently, among all elements of the inertia tensor matrix, only the ${}^{2}I^{*}_{zz2}$ and ${}^{3}I^{*}_{zz3}$ moments of inertia affect the generalized torque of joint 2 and joint 3, respectively and only the ${}^{1}I^{*}_{yy1}$ moment of inertia affects the generalized torque of joint 1. Equation (5.3) shows that the positions of the centre of mass of link 2 and link 3 in the z axis direction does not affect at all the generalized torque of joint 2 and joint 3, respectively, for any kind of movement.

If now joint 2 and joint 3 are allowed to move simultaneously, but still keeping joint 1 stationary, the system now is similar to a two link planar manipulator, where the hand moves in a two dimensional $X_1$-$Y_1$ (vertical) space as shown in figure (5.2). This system has the same property as before in the sense that only the z components of angular velocities and angular accelerations with respect to their own coordinate systems exist in the system. As a result , only the ${}^{2}I^{*}_{2zz}$ and the ${}^{3}I^{*}_{3zz}$ components of the inertia tensor matrices ${}^{2}I^{*}_{2}$ and ${}^{3}I^{*}_{3}$, respectively, are significant and no matter how much the other components of the inertia tensor matrix are varied the system behaviour remains unchanged. On the other hand, the z components of the linear accelerations of each link are zero.

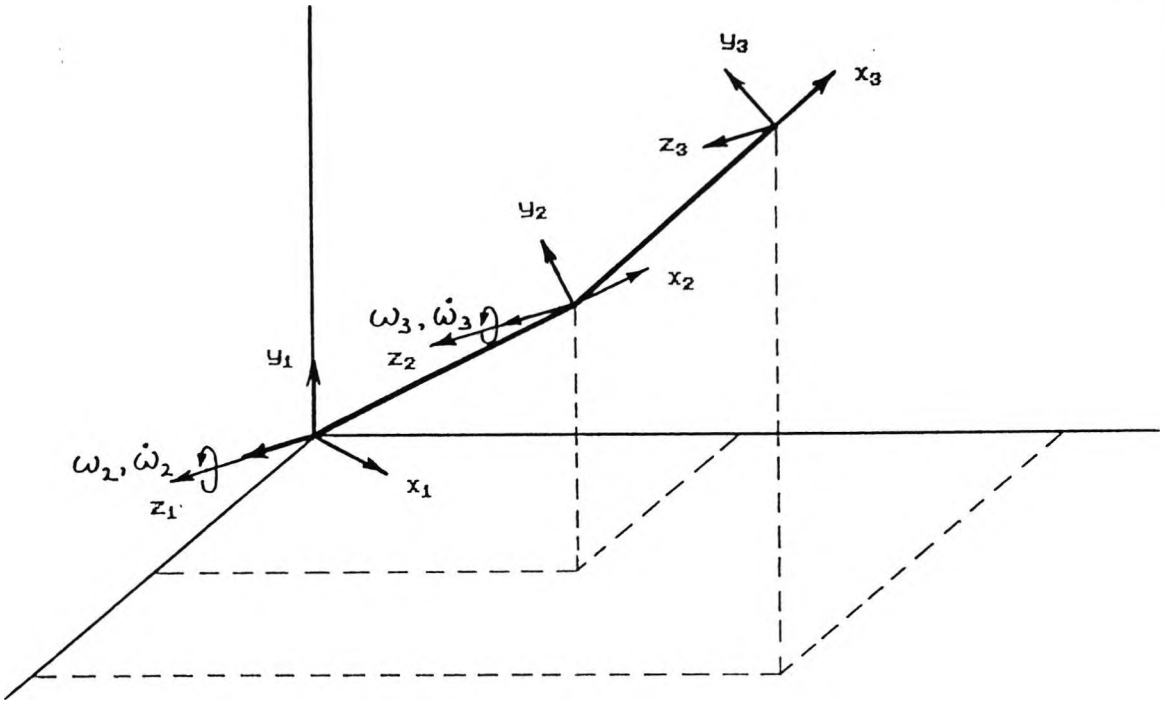



Figure 5.2

A three degree of freedom robot arm with joint 1 locked

When joint 1 is released and hence all links move simultaneously, the movement of joint 1 provides a third dimension movement to the robot. Now, the hand moves in a three dimensional $X_0$-$Y_0$-$Z_0$ space. As a result, both the angular velocities and the angular accelerations of link 2 and link 3 have the x, y and z components with respect to their own coordinate systems and for each joint the angular velocity and acceleration values are given as

$$^2\omega_2 = \begin{bmatrix} \dot{\theta}_1 S2 \\ \dot{\theta}_1 C2 \\ \dot{\theta}_2 \end{bmatrix} \tag{5.10}$$

$$^3\omega_3 = \begin{bmatrix} \dot{\theta}_1 S23 \\ \dot{\theta}_1 C23 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \tag{5.11}$$

$$ {}^{2}\omega_{2} = \begin{bmatrix} \dot{\theta}_1 S2 \\ \dot{\theta}_1 C2 \\ \dot{\theta}_2 \end{bmatrix} \qquad (5.12) $$

$$ {}^{3}\omega_{3} = \begin{bmatrix} \dot{\theta}_1 S23 \\ \dot{\theta}_1 C23 \\ \dot{\theta}_2 + \dot{\theta}_3 \end{bmatrix} \qquad (5.13) $$

where, $S2 = \sin(\theta_2)$, $C2 = \cos(\theta_2)$, $S23 = \sin(\theta_2+\theta_3)$ and $C23 = \cos(\theta_2+\theta_3)$.

This causes all components of the inertia tensor matrices $\overset{*}{I}$ of link 2 and link 3 to be significant to the system behaviour. As a three dimensional movement is provided by the movement of joint 1, the system then has a property that the importance of all inertia tensor matrix elements other than $\overset{*}{I}_{zz}$ for both link 2 and link 3 depend on the movement of joint 1. If joint 1 rotates with a considerably high speed, all elements of the inertia tensor matrices $\overset{*}{I}$ of link 2 and link 3 are significant to the system. On the other hand, if joint 1 rotates with a considerably low speed, the importance of all elements of the inertia matrix elements other than $\overset{*}{I}_{zz}$ for both link 2 and link 3 can be ignored.

The property cited above, however, does not apply to link 1. In the case of link 1, among all ten inertial parameters of link 1, only ${}^{1}\overset{*}{I}_{1yy}$ element of its inertia tensor matrix affects the behaviour of the system. The angular velocity and angular acceleration of link 1 are still represented by equations (5.4) and (5.7),

respectively. This is due to the restricted movement of link 1 since link 1 has only 1 degree of freedom about the $z_0$ or $y_1$ axis (the origin of the coordinate system 0 coincides with the origin of the coordinate system 1 and based on the D-H concept, the $z_0$ and $y_1$ axes have the same direction while the $z_1$ axis is in the direction of joint 2 axis). It should be noted due to the restricted motion, link 1 does not have both linear velocity and linear acceleration.

When the forces and torques from the distal links are considered, the torques/ forces propagation needs to be studied. As all forces and torques are treated as vectors, the orientations among all coordinate systems determine the propagation. Thus, apart from the joint variables $\theta$'s, the twist angles $\alpha$'s play an important role. The force exerted by each joint needs to be considered. This is because this force causes a torque which must be overcome by the proximal joints. Similarly, the x and y components of torques at joint 2 and joint 3 need to be considered as these torque components propagates and affects the generalized torque at joint 1. This propagation can be observed from the Newton-Euler algorithm which was presented in section 3.6.2.

Let link 3 move alone. As in this manner link 3 moves in a two dimensional (X1-Y1) space, thus the net force which exists in the system is in the direction of x and y only of the coordinate system 3. As a result, the centre of mass of link 3 in the direction of the z axis of the coordinate system 3 has no significance. On the other hand, the changes of the centre of mass of link 3 in the direction of the x and y axes of the coordinate system 3 affects the net force of link 3. This property still holds when joint 2 and joint 3 move simultaneously ie. the centre of mass of link 2 and link 3 in the direction of the $z_2$ axis and the $z_3$ axis, respectively, do not affect the net forces of the corresponding links and, hence,

they do not affect the generalized torque at joint 2 and joint 3 either. So, only the centre of mass of link 2 and link 3 in the direction of their corresponding **x** and **y** axes affect the system behaviour. In the case for link 1 when it moves alone, there is no translational motion due to its one degree of freedom movement limitation, hence no net force exists on link 1. If now all links move simultaneously, although the **z** components of the centre of mass vector of link 2 and link 3 do not affect the generalized torques at joint 2 and at joint 3, respectively, but they both affect the generalized torque at joint 1.

To summarize, the inertial parameters which affect the generalized torques of joint 1, joint 2 and joint 3 when all links move simultaneouly and link 1 moves with a considerably high speed are tabulated in table (5.1); and when all links move simulataneously and link 1 moves with a considerably low speed are tabulated in table (5.2).

| Generalized torque | Parameters |
|---|---|
| Joint 1 | All inertial parameters of link 3, link 2 and ${}^{1}I^{*}_{1yy}$ |
| Joint 2 | All inertial parameters of link 3 and link 2 except ${}^{2}\bar{c}_{2z}$ |
| Joint 3 | All inertial parameters of link 3 except ${}^{3}\bar{c}_{3z}$ |

Table 5.1

Inertial parameters which affect the generalized torques when link 1 moves with a considerably high speed

| Generalized torque | Parameters |
|---|---|
| Joint 1 | $m_3, {}^{3}\bar{c}_{3x}, {}^{3}\bar{c}_{3y}, {}^{3}I^{*}_{3zz}$, $m_2, {}^{2}\bar{c}_{2x}, {}^{2}\bar{c}_{2y}, {}^{2}I^{*}_{2zz}, {}^{1}I^{*}_{1yy}$ |
| Joint 2 | $m_3, {}^{3}\bar{c}_{3x}, {}^{3}\bar{c}_{3y}, {}^{3}I^{*}_{3zz}$, $m_2, {}^{2}\bar{c}_{2x}, {}^{2}\bar{c}_{2y}, {}^{2}I^{*}_{2zz}$ |
| Joint 3 | $m_3, {}^{3}\bar{c}_{3x}, {}^{3}\bar{c}_{3y}, {}^{3}I^{*}_{3zz}$ |

Table 5.2

Inertial parameters which affect the generalized torques

when link 1 moves with a considerably low speed

If the system now is simplified by considering link 2 and link 3 as a line with a uniform distribution of mass, the number of inertial parameters of link 2 and link 3 which affect the system behaviour are reduced to m, $c_x$, $I^{*}_{yy}$ and $I^{*}_{zz}$. Equation (5.3) then becomes

$$\begin{bmatrix} {}^{i}F_{ix} \\ {}^{i}F_{iy} \\ {}^{i}F_{iz} \\ {}^{i}nn_{ix} \\ {}^{i}nn_{iy} \\ {}^{i}nn_{iz} \end{bmatrix} = \begin{bmatrix} {}^{i}\dot{V}_{i-1x} & -\omega_z^2 - \omega_y^2 & 0 & 0 \\ {}^{i}\dot{V}_{i-1y} & \dot{\omega}_z + \omega_x\omega_y & 0 & 0 \\ {}^{i}\dot{V}_{i-1z} & -\dot{\omega}_y + \omega_x\omega_z & 0 & 0 \\ 0 & 0 & -\omega_z\omega_y & \omega_y\omega_z \\ 0 & -{}^{i}\dot{V}_{i-1z} & \dot{\omega}_y & -\omega_x\omega_z \\ 0 & {}^{i}\dot{V}_{i-1y} & \omega_x\omega_y & \dot{\omega}_z \end{bmatrix} \begin{bmatrix} \\ m_i \\ m_i\,{}^{i}cx_i \\ {}^{i}\overset{*}{I}_{iyy} \\ {}^{i}\overset{*}{I}_{izz} \\ \end{bmatrix} \quad (5.14)$$

In the case of link 1 only $\overset{*}{I}_{yy}$ affects the system behaviour and equation (5.3) becomes

$${}^{1}nn_{1y} = {}^{1}\dot{\omega}_{1y}\,{}^{1}\overset{*}{I}_{1yy} \quad (5.15)$$

Based on all the facts cited above, the important inertial parameters of each link which need to be considered in implementing the distortion quantitative validation technique are given in table (5.3).

By observing all inertial parameters which are given in table (5.3), all these parameters in fact are derived from the two most fundamental parameters for each link, ie. the mass of link and the length of link. Hence, a new term called the fundamental parameters is introduced to represent the mass and the length of each

link. These fundamental parameters will be used in implementing the distortion validation technique in chapter 7.

| Link no. | Parameter |
|---|---|
| 1 | $I^*_{1yy}$ |
| 2 | $m_2$ |
| | $^2\bar{c}_{2x}$ |
| | $I^*_{2yy}$ |
| | $I^*_{2zz}$ |
| 3 | $m_3$ |
| | $^3\bar{c}_{3x}$ |
| | $I^*_{3yy}$ |
| | $I^*_{3zz}$ |

Table 5.3

Effective inertial parameters of a simplified system

## 5.3 Simulation of the Robot Dynamics

In a dynamic simulation of a robot arm, the inverse dynamics equation needs to be converted into the forward dynamics equation, ie. given a set of generalized torques, the joint acceleration of each joint is to be computed which in turn is integrated twice to obtain the position. This torque is obtained from the controller output. A typical block diagram of a robot dynamics simulation is shown in figure (5.3).

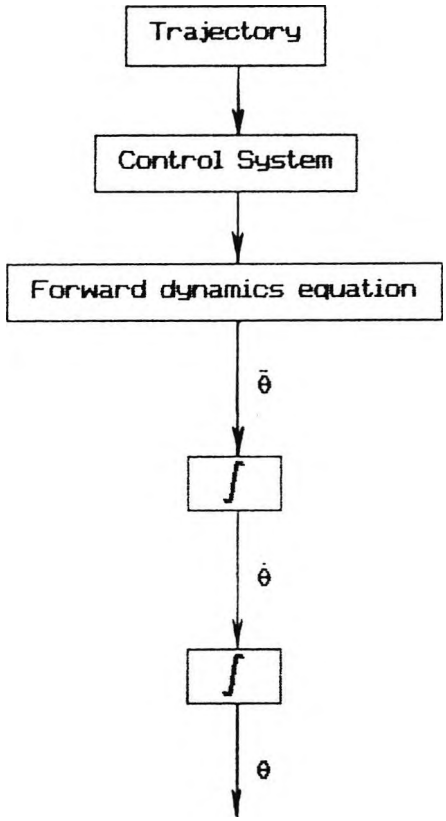



Figure 5.3

A block diagram of a robot computer simulation

In developing a simulation program of a robot manipulator dynamics system, it is preferable to have an inverse dynamics algorithm which has a flexible access for manipulating each parameter as well as computational efficiency. With all these considerations and reasons cited above, the most efficient general purpose algorithm [Hollerbach, 1980] ie. the Newton-Euler approach has been selected for deriving the manipulator mathematical model. An advantage of N-E method is that it is relatively easy to analyse the dynamic behaviour of robot manipulators. This is due to the familiarity of the Newton equation for translational motion and the

Euler equation for rotational motion.

To obtain the values of accelerations $\ddot{\theta}$'s, which in turn are integrated to obtain the velocity and position results, there are two methods to choose from. Either $\ddot{\theta}$'s are manually derived before hand or they are numerically calculated by a computer using an inverse dynamics algorithm. The first method, manually derivation by hand, has an advantage of having faster computation time. This is because only the final equation to produce $\ddot{\theta}$ is produced numerically by a computer. On the other hand, it has a big disadvantage because it lacks flexibility for manipulating the parameters. Since everything is done by hand, the resulting equations to be integrated are dependent on the assumed nature of the parameters.

For example, the centre of mass vector of a link which is assumed as a line with uniform distribution of mass is expressed as

$$ {}^{i}\bar{c}_i = \begin{bmatrix} A_i/2 \\ 0 \\ 0 \end{bmatrix} \qquad (5.16) $$

where $A_i$ is the length of link i.

If the offset distance, due to the structure of the robot arm, in joint 2 and joint 3 is not neglected, equation (5.16) then becomes

$$ {}^{i}\bar{c}_i = \begin{bmatrix} A_i/2 \\ 0 \\ d_i \end{bmatrix} \qquad (5.17) $$

where $d_i$ is the D-H offset distance parameter and not an inertial parameter.

As a result, in the final equations derived by hand, some terms are missing due to the assumption of zero offset distance elements in the centre of mass vector $\bar{c}$. Hence, in developing the model, if the offset elements are to be included, the equations should be rederived and this process is time consuming and inefficient.

Although the second method has a disadvantage of having longer computation time, it has an important feature, ie. flexibility in manipulating the parameters. Because the recursive set of equations use parameters in full without any neglected elements, there is no difference in the computational time with or without these elements. Another important advantage over the first method is that the risk of making errors is reduced particularly if a full complete model (including coriolis and centrifugal terms) is to be obtained. Because these terms are complicated, deriving them manually before hand is prone to error.

A subroutine in FORTRAN which performs the inverse dynamics problem in order to obtain the generalized torques of a robot with revolute joints, called NE1, has been created. Its inputs are the number of links, the Denavit-Hartenberg parameters and the inertial parameters, and its output is the generalized torque vector. It also has a flag which is used to determine whether the $Z_0$ axis is in a horizontal plane (perpendicular to the gravity acceleration vector) or in a vertical plane (parallel to the gravity acceleration vector), and a flag which is used to disable the gravity effect.

From equation (5.1), it is seen that each component of the generalized torque

vector can be computed separately. That is the gravity force vector is obtained by setting the $\ddot{\theta}$ and $\dot{\theta}$ to zero and the coriolis/ centrifugal force vector is obtained by setting the $\ddot{\theta}$ and gravity acceleration to zero. The inertial force vector is obtained by substracting the gravity and coriolis/ centrifugal force vectors from the known generalized torque vector. An algorithm by Walker and Orin [1982] is used to efficiently compute the D matrix.

The above method yields a convenient simulation program for implementing the distortion technique. An important aspect is that it is not necessary to rederive the dynamic equations when a parameter value is changed, eg. in the case of the centre of mass vector, and hence it has the flexibility for manipulating important model parameters. Figure (5.4) shows the flowchart of a forward dynamics simulation.

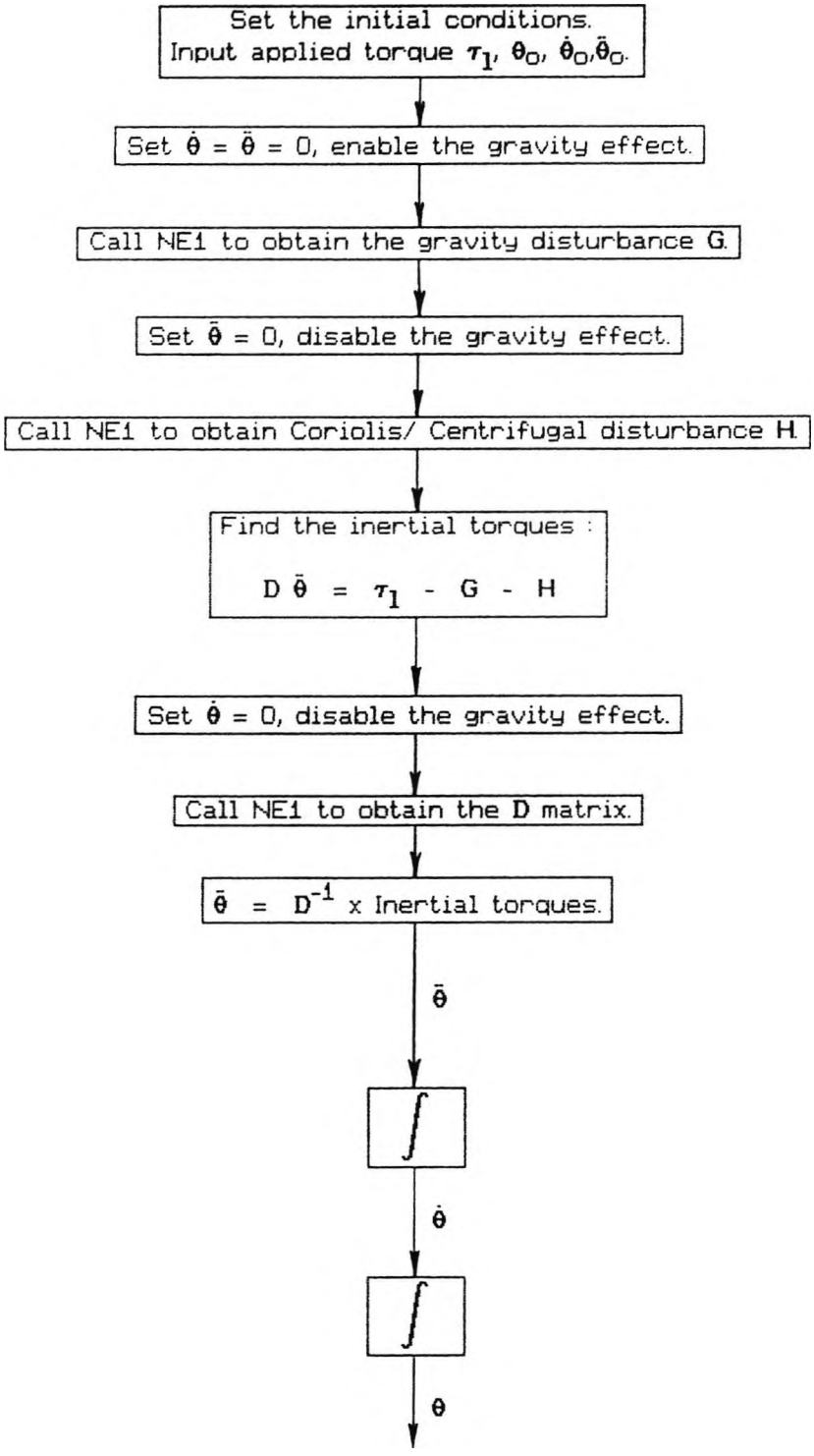



Figure 5.4

Flowchart of a forward dynamics simulation

## 5.4 Conclusions

In this chapter, foundations to implement the distortion validation technique on a robot system with revolute joints have been given. A knowledge of how the system reacts if some parameters are distorted is important. The sensitivity analysis of all inertial parameters have been carried out. Based on this knowledge, important parameters which are to be distorted can be selected.

The efficient inertial parameters for a simplified robot system are summarised in table 5.3. A new term has been introduced and is called the fundamental parameters term where the all inertial parameters depend on the value of these new parameters. For each link, the fundamental parameters are the mass and the length. These fundamental parameters will be used later in validating a model using the distortion quantitative validation technique.

To carry out a computer simulation work of a robot dynamics, the N-E algorithm approach has been chosen for both its computation efficiency and ease in analysing the robot dynamic behaviour. Using a flow chart given in figure (5.4) a forward dynamics problem can be carried out. The D matrix is then computed using an algorithm which was developed by Walker and Orin [1982] since it gives the most efficient general purpose algorithm for computing forward dynamics problems [Murray, 1986].

The information which has been given in this chapter will be used in developing and validating robot models in chapter 7.

# CHAPTER 6

# EXPERIMENTS WITH TQ MA2000 ROBOT

## 6.1 Introduction

In a robot system, it is necessary to have a computer system which is used in the servo system. By controlling the movement of each joint of the robot, a good flexibility of the robot arm can be achieved, so that, a multipurpose robot can be obtained.

The task of this computer system in controlling the movement of the links of a robot arm can be as simple as picking up an object from one place and moving it to another place to as complex as picking up an object with a certain orientation and moving it to another place by following a particular trajectory. Whatever the tasks are, a well designed controller as well as a good controlling servo algorithm are required.

Basically, there are two types of controller system ie. single processor system and multi processor system. In the first system, the coordination of all joint actions is controlled by a single processor based motor controller. The disadvantage of this system is that it is incapable of implementing a complex control algorithm due to its limited processing speed. This system is usually adopted for a pick and place task and the TQ MA2000 falls in this category. On the other hand, the second system is composed of a number of local processors, each of which controls a single joint only. This system can be used for assembly work eg. PUMA 600.

For both single processor and multi processor systems, they have two control levels. The low level control, ie. the motor controller, is for controlling the position/ orientation of the hand as well as the speed of the joint axes, where the measured informations are obtained from the internal sensors eg. potentiometer and shaft encoder. The high level control, which is done by the host computer, defines the trajectories for all joints. The information signals are obtained from both the internal sensors and the external sensors (process input terminals in TQ MA2000) and the resulting set points are then sent to the motor controller.

This chapter gives a description of the TQ MA2000 robot arm and an experiment which was carried out. This experiment was carried out prior to performimg the distortion quantitative validation method. The computer architecture is explained as well as its software system. For advanced experiments, this robot needs some modifications both in hardware and in software due to its limited capability.

Some experimental results are presented. The transient responses could not be obtained in a longer period due to the limited memory of the BBC computer. A ramp-step input signal was used since it was the simplest way and it did not require a lot of memory to create this kind of trajectory.

## 6.2 TQ MA2000 in General

The TQ MA2000 robot manufactured by Tequipment Ltd. is a low cost educational robot with unique design based on industrial robots. Mechanically, the robot consists of an arm with 6 joints as shown in figure (6.1). Joint 1, joint 2 and joint 3 provide the waist , shoulder and elbow motions, respectively. These first three joints, called the major axes [Tequipment, 1984], give the position of the hand and

perform the major operation in many applications. The remaining three joints, called the minor axes [Tequipment, 1984] complete the robot motion by giving the orientation of the hand. The motions which are given by joint 4, joint 5 and joint 6 are called the pitch, the yaw and the roll. The range of movement of the waist, shoulder and elbow are $270^{o}$ and the range of movement of the pitch, yaw and roll are $180^{o}$. This allows the robot to reach most points inside a hemisphere work space. A gripper which is pneumatically powered, is attached to the hand and this allows the robot to perform operations such as a pick and place task. Basically, the gripper works as an on/ off device and a variety of tools can be attached in order to carry out specific tasks.

This robot has 6 DC motors with permanent magnets to actuate its six joints where each joint is operating with a closed loop control. The TQ MQ2000 robot arm is not a direct drive robot arm. A torque is provided by the DC motor to the joint through a gear box and pulleys. Detail about the actuator and gear ratio technical information is given in appendix C.

The teach pendant provides a friendly facility for the user to communicate with the robot through the host computer. Information signals such as position of each joint, speed and mode of movement can be given to the robot arm. Some mode of movements are point to point operation with absolute position, point to point operation with relative position to the current position, point to point operation through the lead by nose method and continuous path operation. For other and more detail modes of movement, one can refer to the user's manual [Tequipment, 1984]. Programming the robot to perform a certain task can also be done through this teach pendant where some editing functions are provided.

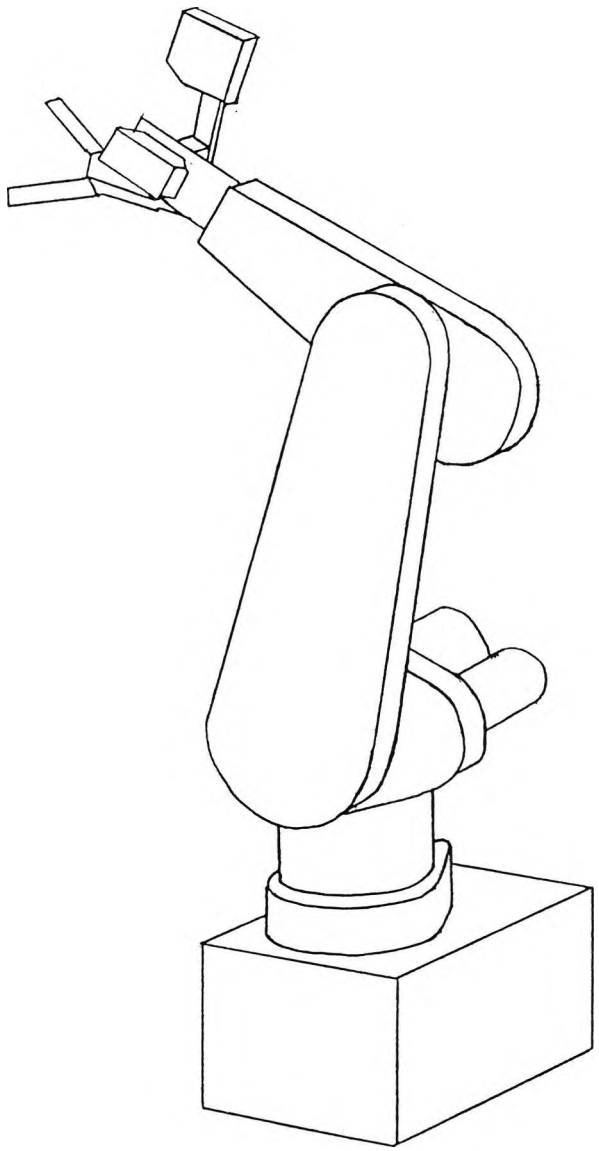


Figure 6.1

TQ MA2000 robot arm

The robot has also additional input and output ports for connecting with other devices. The input ports receive signals from the outside world which are to be tested and the output ports send signals to the outside world which are to be set. All these capabilities allow the robot to be used as a part of a flexible manufacturing system as well as a stand alone system.

The host computer can be one of the following : BBC model B micro computer, OU Hektor micro computer or IBM PC micro computer. The one which is used in the control engineering computer laboratory is the BBC model B micro computer where the host computer is further controlled by a PDP 11 computer. Figure (6.2) shows the computer network in the control engineering computer laboratory.

### 6.3 Hardware

The TQ MA2000 robot system in the computer control laboratory comprises of a robot arm, a BBC microcomputer with 32 KRAM as the host computer, a teach pendant, a 1 MHz 6502 based motor controller interface and an operating system software. Figure (6.3) shows the configuration of the system. The controller interface contains an ADC and DAC circuit, a servo system circuit and a power amplifier circuit. The control algorithm and the default values of the PID parameters are held in EPROM, and the work space and shared memory are held in RAM. The shared memory, which is accessible from both the host computer and the motor controller, contains set points, measured positions, positional errors, PID gain values and PID outputs.

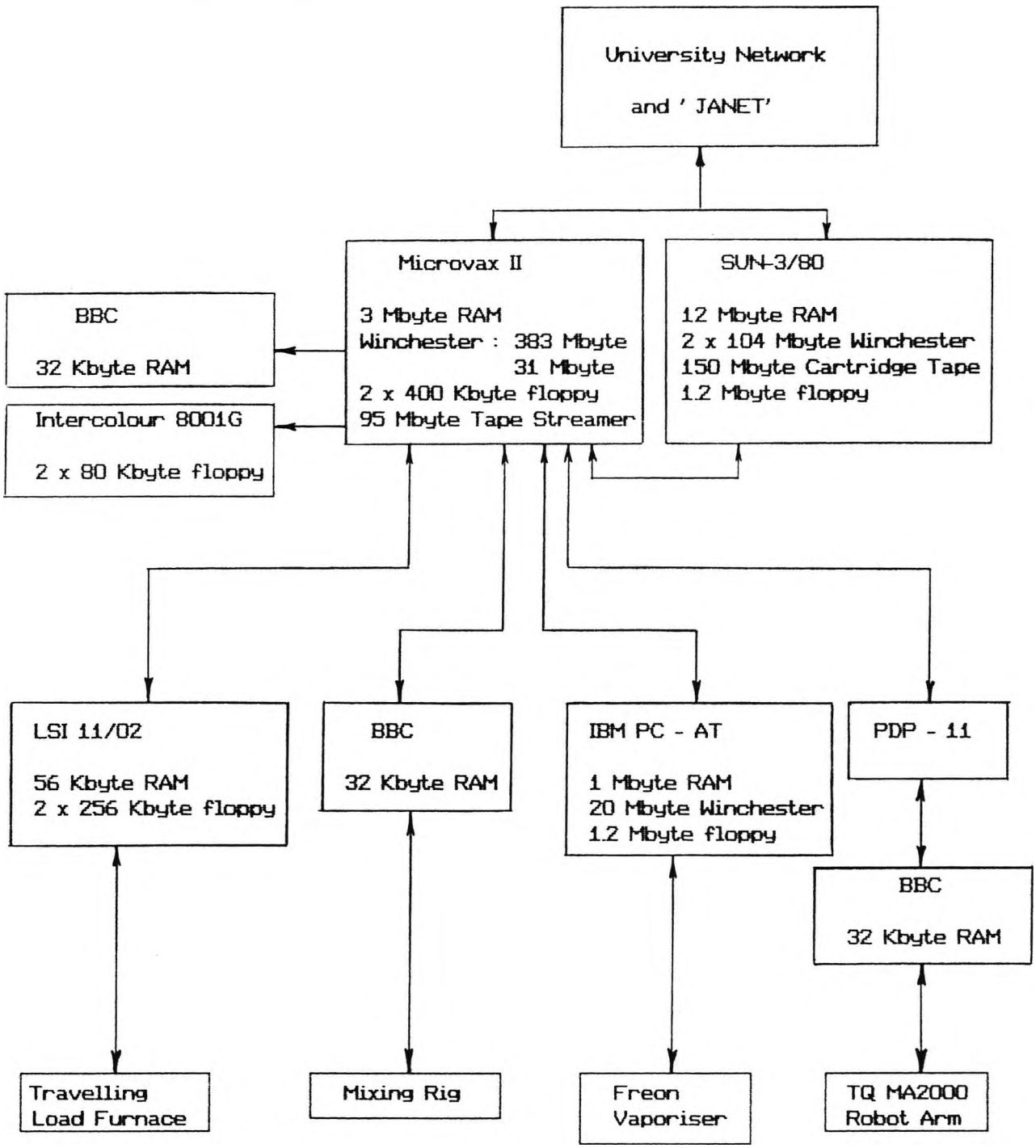



Figure 6.2

Computer network in the control engineering computer laboratory

The main task of the motor controller is to perform real time PID control of all the six joints. The set points, which are sent from the host computer, and the measured positions are computed using a conventional PID algorithm by the motor controller in order to give the required torques to the coresponding actuators. Output of this process is then passed to an 8 bit DAC converter. The resulting analogue signal is fed to a Pulse Width Modulation circuit prior to amplifying it. Output of this power amplifier with a frequency of 15.6 KHz then drives the corresponding DC motor. Using a potentiometer, the response of each joint is measured and sent back to a motor controller via a 12 bit ADC converter. Figure (6.4) shows the block diagram of the feedback control system.

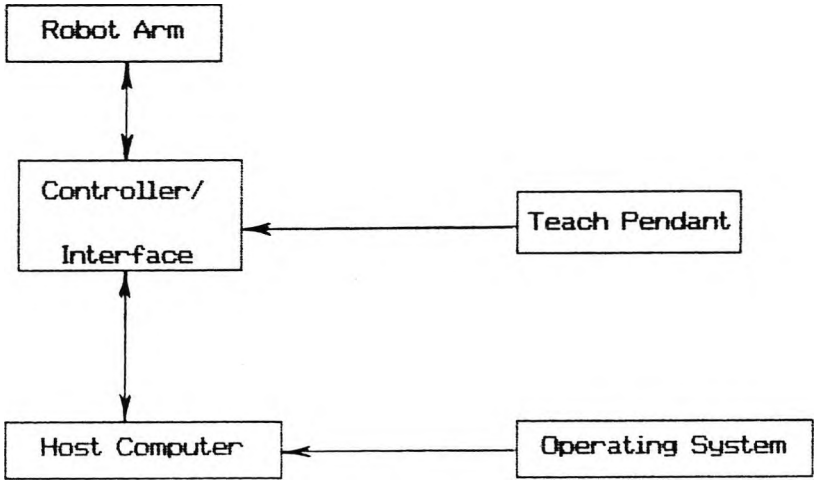



Figure 6.3

TQ MA2000 robot system

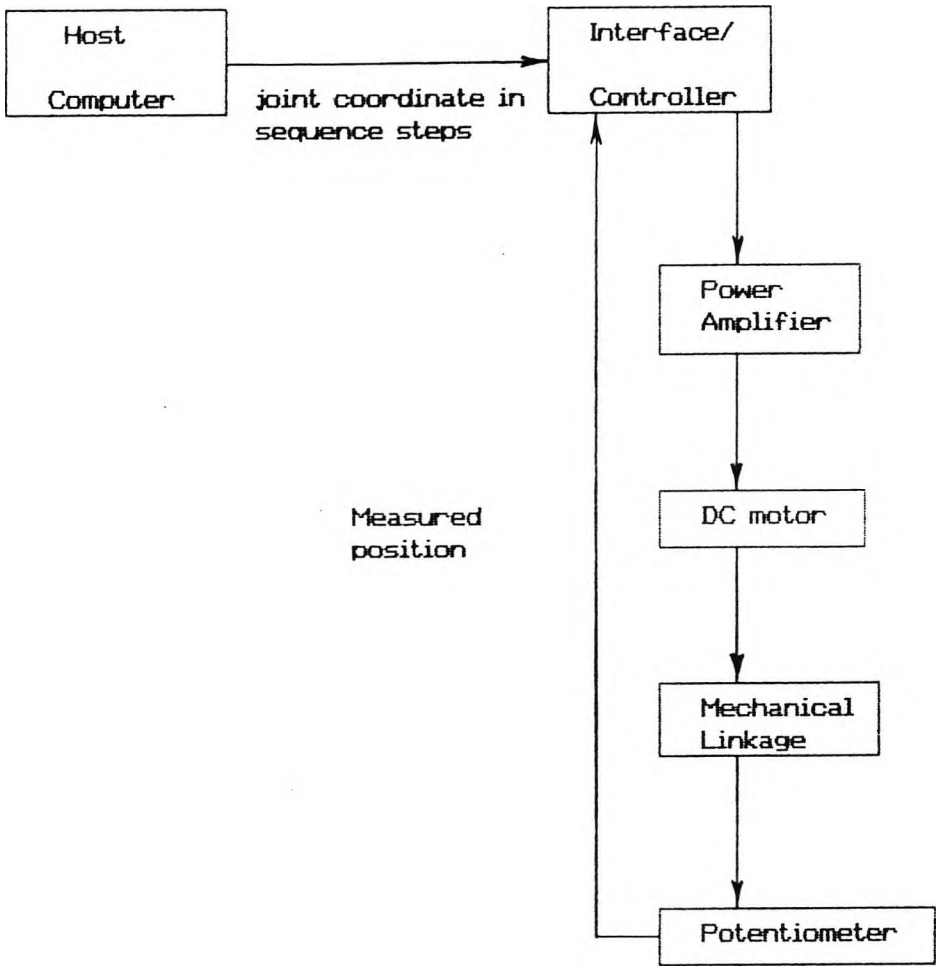



Figure 6.4

The feedback control system of TQ MA2000 robot arm

The host computer interprets the user's commands for each robot joint configuration and translate them into a series of set points which in turn are sent to the motor controller. It also performs the required calculation for the process inputs and process outputs.

Understanding the communication protocol between the host computer and the motor controller is important since the shared memory cannot be accessed at the same time by both the host computer and the motor controller. The motor controller has a higher priority to access the shared memory than the host computer. So, if the shared memory is being accessed by the host computer, the motor controller will not wait for an access and as a consequence a new set point will then be missed. To protect the robot from an uncertain condition, if the error is less or greater than -127 or +127 ADC units respectively, the host computer will temporarily stop sending the set points.

## 6.4 Software

The operating system of the TQ MA2000 is written in BASIC and in assembly. The BASIC routines interface the use of the teach pendant in order to give commands to the robot. The assembly routines interface the host computer to the shared memory contained in the motor controller. There are 4 teaching modes available for the user and these are under two types of operation.

1. Point to point operation.

The teaching modes which fall under this category are :

- Drive through using the teach pendant.
- Lead by nose
- Off line by the host computer.

2. Continuous path operation.

There is only one teaching mode for this kind of operation ie. lead by nose. When the arm is led by nose, all the position during that period are stored in memory.

Hence, the length of period depends on the memory available in the host computer. For the BBC computer, it is less than 60 seconds. When it is played back, all positions recorded during that period are sent to the motor controller as the set points. Using this technique, the robot can move along a specified path.

Assembly routines which interface the host computer to the shared memory are :

- DRIVESA :

Sends the set point for the current motion, being driven under direct control of the teach pendant.

- LEADSA :

Used when in Lead by Nose mode and reports back the current values for each motion.

- CONTPATH :

Executes a series of ministeps which constitute a continuous path step. This series of mini steps are stored in address TOP+&800 to &EEEE. Refer to BBC Manual [Coll, Allen, 1982] for the information of TOP.

- SAMOVE :

Moves the robot from the current step to the next step. This routine also creates a trajectory between each step in such away so that all joints arrive at their own destinations (the posture configuration) at the same time.

- REPORTG :

Reports the PID gains for each motion.

- SETG :

Sends new PID gains for each motion.

- REPORTEPP :

Reports the Errors, Positions and Powers for each motion.

Each command is stored in a step and each step has an information of the position of each link. This position is expressed in posture value. For the major and minor axes which can rotate up to $270^{o}$ and $180^{o}$ respectively, they have 1000 units of posture values. These posture values are scaled to ADC units prior to sending them to the shared memory.

More information about the BASIC and assembly routines can be found in TQ MA2000 Hardware and Software manual [Tequipment, 1986]. All the available routines provide a highly interactive interface between the human operator and the robot. Moreover, the facilities of process inputs and process outputs give the robot an intelligence to interact with outside world. But from the control engineering experimental point of view, this facility is not as important as in the industrial field and using the BASIC interpreter means slower speed and less available numerical libraries.

### 6.5 Experiment

In doing the experiments, none of the four available teaching method was used. The block diagram of the system while in the play back mode is given in figure (6.5). Given a set of posture values with range 0-999 for all six joints, a BASIC

procedure called PATH calculated the biggest increment among all six joints. The posture values of the current step and the next step were scaled to ADC units and an assembly routine called SAMOVE was then called. This SAMOVE routine generated a series of set points, ie. the trajectory, between the current step and the next step for all joints and this trajectory depended on the value of the selected parameter RATE which acted as a delay. The algorithm inside SAMOVE created a trajectory for each joint in such away so that the final set points of all joints occured at the same time. As a result, if the increment between the current step and the next step of each joint was different to each other, each link would move with different speed. Hence, the parameter RATE in the teach pendant defined the rate of movement of the whole system.

Using a point to point operation, the interval time between having finished the current step and starting executing the next step, if done by the existing original program, would take a considerable amount of time. This caused all joints to achieve their steady state conditions before executing the next step. Another difficulty arose since the actual set points generated by SAMOVE were not available for the user. So, the actual input signal of each joint, which was really important in modelling this robot, could not be obtained. The only available facility provided by the manufacturer is an analogue port of the robot output response.

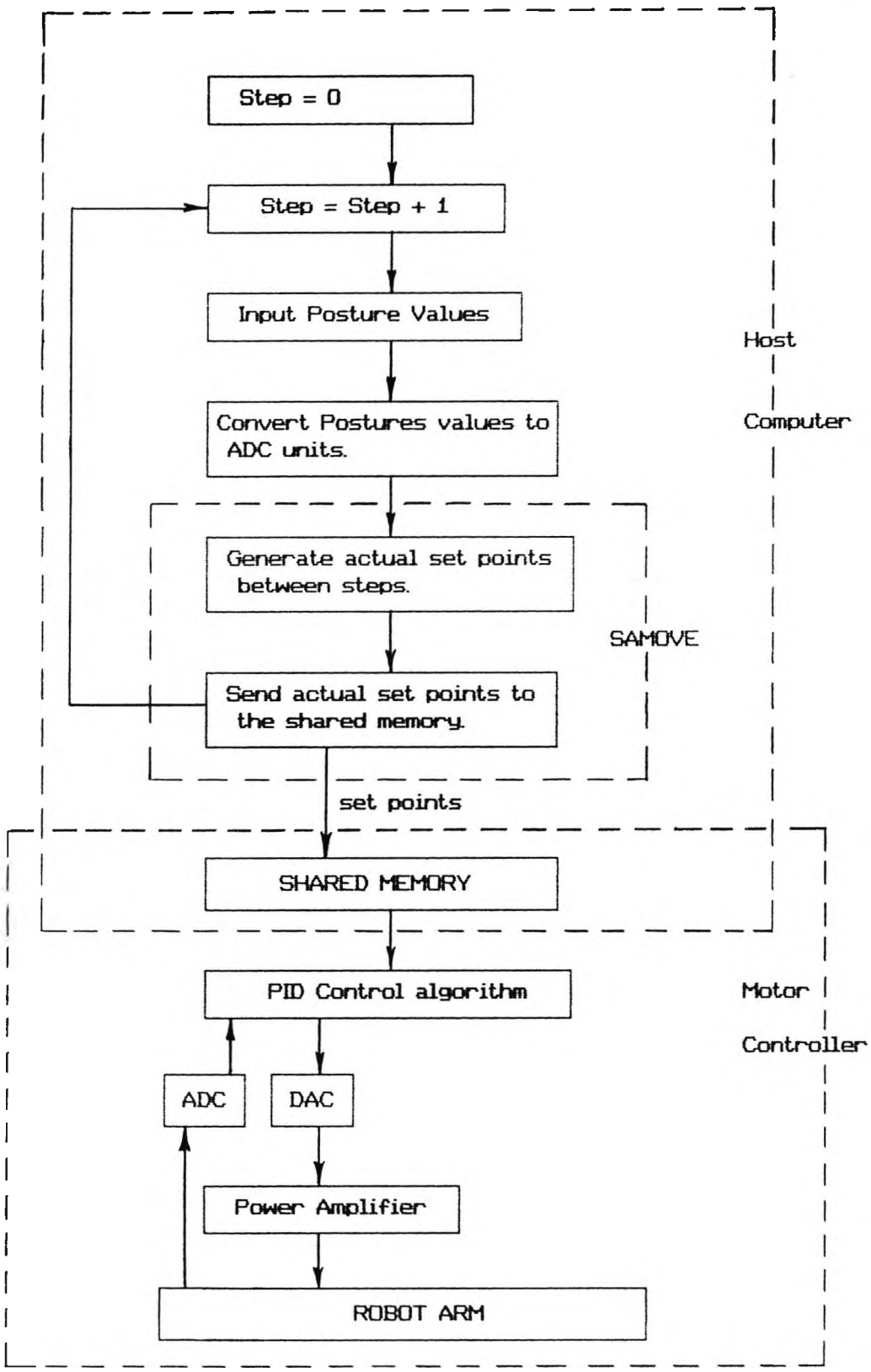



Figure 6.5

Block diagram of the system in play back mode

Using a continuous path operation, the steady state conditions between each step could be overcome since some procedures were by-passed in this operation. In this operation, all ministeps which constitute a continuous path step, were stored in memory. This operation, however, created another problem since the continuous path was generated by leading the robot arm by nose (ie. by hand). This method yielded an unreliable continuous path as it was difficult to create a good continuous path manually. The set points generated by SAMOVE, as in the point to point operation, could not be obtained either.

Considering all the difficulties cited above, none of the four available teaching modes could be used. Instead, a modification of a continuous path mode was created. Using the BBC computer with BASIC as a language to govern the robot, it was virtually impossible to create a sophisticated trajectory. The reason was the processing speed of BASIC was very slow and the robot might have settled down before the host computer finished computing the next point in the trajectory. Also, due to the limited memory of the BBC computer, the trajectory could not be stored in an array. Instead, every point had to be sent to the motor controller immediately by calling SAMOVE (SAMOVE still generated the actual set points between the trajectory points generated in BASIC and these actual set points were the set points which were sent to the motor controller ie. to the shared memory).

Since the *FX command in BBC BASIC to send an output only to a printer (not the VDU and the printer to increase the speed) did not work and the printer did not have a sufficient buffer, the resulting output was stored in an array. Hence, the observation time for the output response was limited to the available memory in the host computer.

Using the similar principle to the continuous path mode, but instead of leading the robot by nose to create the trajectory, a ramp step trajectory applied to the first three joints was created using BASIC with the assumption that the interval time between each point in the trajectory was much smaller than the transient time. This trajectory has been chosen to be simple but capable of providing the dynamic behaviour of the robot. The trajectory algorithm inside SAMOVE routine was made less significant by giving the same small increments to the joints. Joint 4, 5 and 6 were kept constant. In this manner, the wrist and the gripper were assumed as a load. Using this technique, the SAMOVE routine was used only to send the ramp step trajectory points to the shared memory in the motor controller since the same small increment of the ramp step trajectory points applied to joints 1, 2 and 3 fooled the trajectory algorithm inside this routine (SAMOVE routine could not be modified since it was not accessible).

In order to be able to carry out this technique, some BASIC procedures in the original program, which were not needed in this experiment, were deleted to overcome the out of memory space problem. It was found that using the REPORTEPP routine to obtain the output response of the robot was not fast enough ie. the robot reached its steady state condition between every pair of trajectory points. An alternative way to obtain the output response faster was by accessing the location of the shared memory directly.

Each measurement of the output response of the robot was stored in the shared memory in a location of two bytes since a 12 bit ADC converter was used. These locations are :

- joint 1 : Low byte FC20

High byte FC28

-joint 2 : Low byte FC21

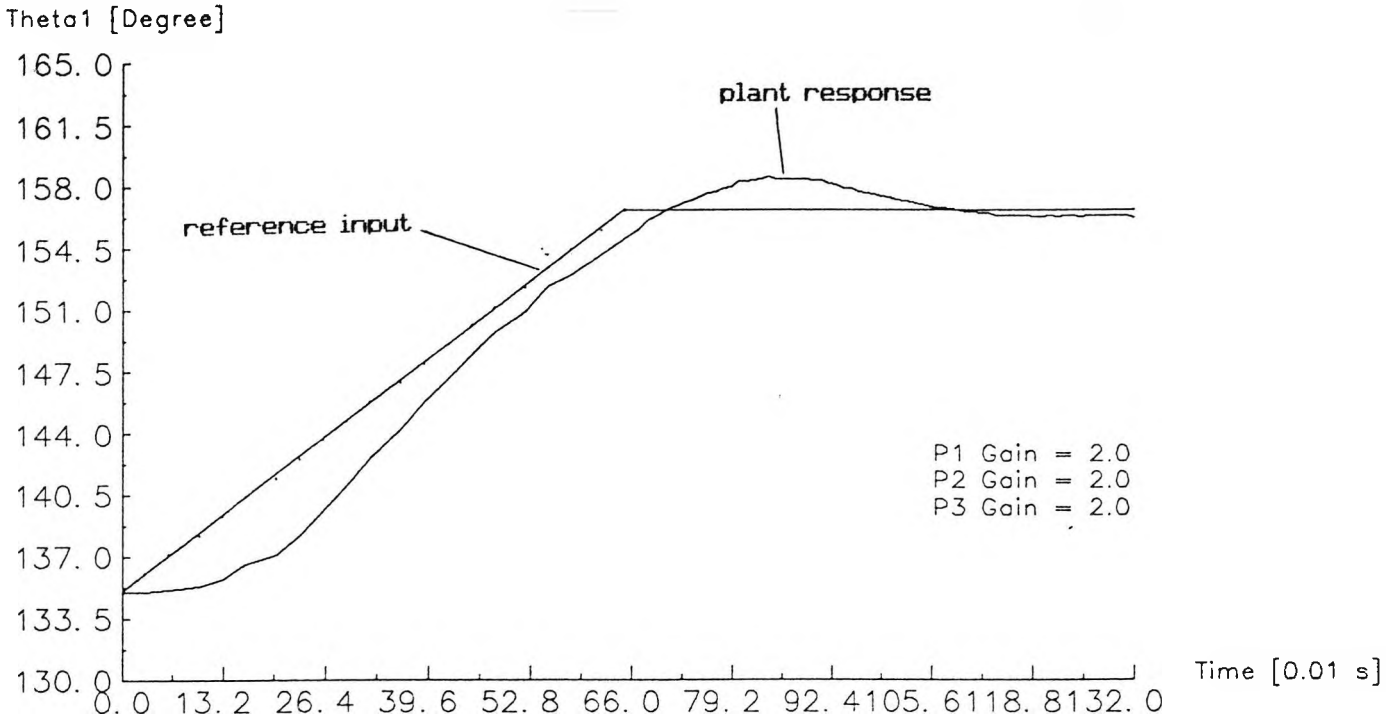
High byte FC29

-joint 3 : Low byte FC22

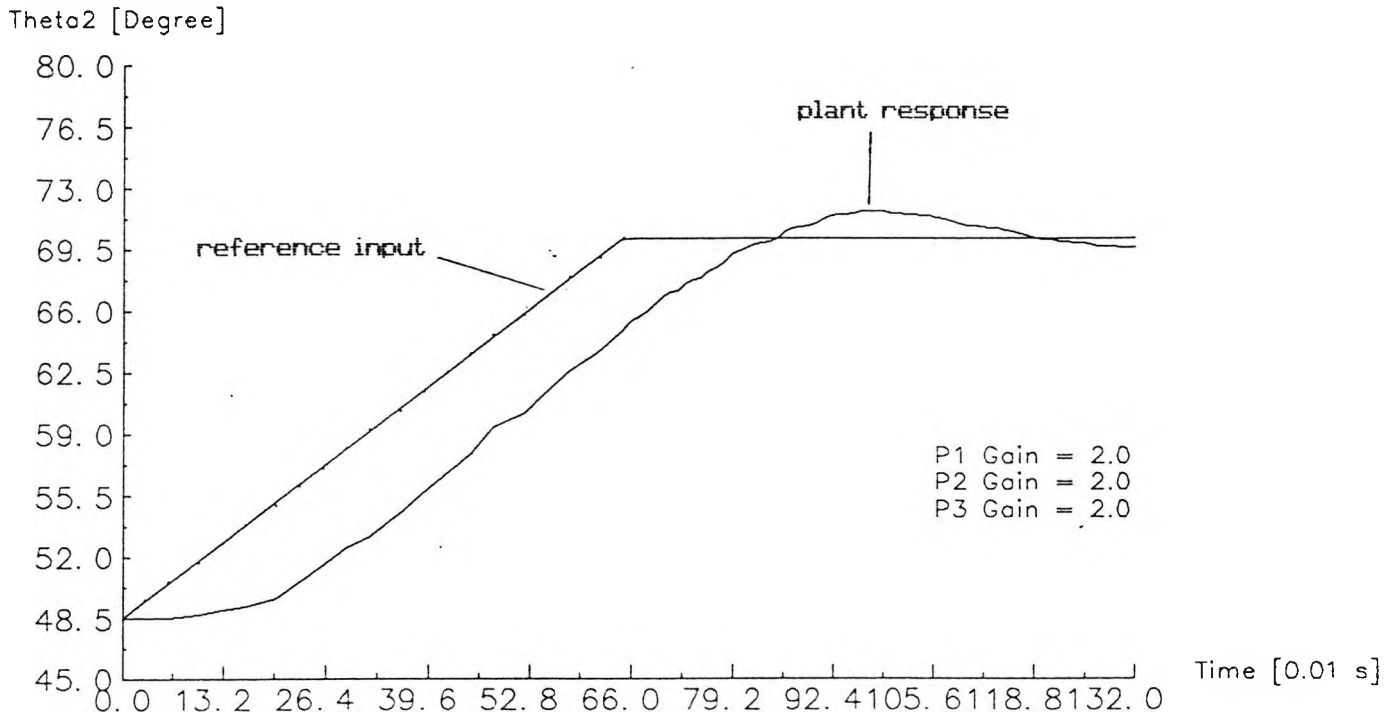
High byte FC2A

To obtain a good dynamics result, the robot was made under damped by setting the Integral and Derivative gains to zero. Since the adjustable PID gains with range from 0 to 20 are not the actual PID gains, this way also simplified the procedure to obtain the actual gain (in this case only the proportional gain). From the information given by the manufacturer, the power value which was obtainable using REPORTEPP routine was not the actual power which drove the motor. Instead, it was the output of the PID controller before amplified by the power amplifier. So, by knowing the values of error and power given by REPORTEPP, the actual proportional gain of each joint was obtained experimentally at some different postures.

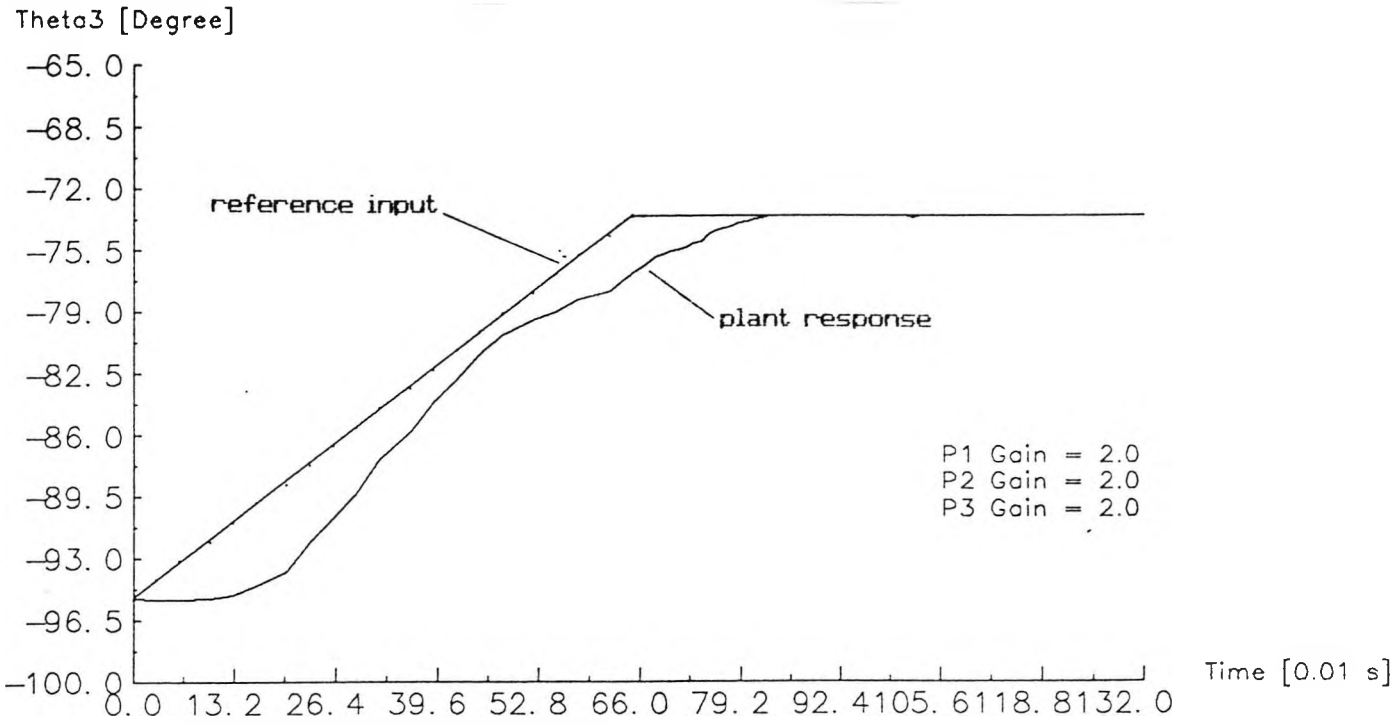
In the experiment, joint 1, joint 2 and joint 3 were commanded to move $21.6^{o}$ in 0.65 seconds. The joints then were commanded to remain constant after this. The adjustable proportional gains of all joints were set to 2. Figure (6.6) shows the responses of joint 1, joint 2 and joint 3. Due to the nature of communication protocol between the host computer and the motor controller (see section 6.3), not every trajectory points could be sent to the motor controller by SAMOVE at the same time interval. So, linear least square fitting was applied to the input signal in producing the plots. This linearly fitted input signal will be used in the modelling.




(a) Response of joint 1




(b) Response of joint 2




(c) Response of joint 3

Figure 6.6

Experimental results of TQ MA2000 robot

## 6.6 Conclusions

An experiment to obtain the dynamic behaviour of a TQ MA2000 robot arm has been carried out. This robot has a lot of capability to carry out works in the industrial field. The teach pendant gives an interactive interface between a man and the robot. The process inputs and process outputs facility allows the robot to communicate with the outside world. In short, this robot reflects industrial practice with a slower processing speed.

From the control engineering experimental point of view, this robot needs some modifications both in the computer hardware and in software. For example, in doing a control experiment, it is preferable to have the operating system done in

FORTRAN or C rather than in BASIC since speed is more important than the ease of use (the teach pendant is not really necessary for this kind of experiment). Also, to have more memory capacity to create bigger programs and to record all responses of the experiment, and to gain more speed, an IBM AT compatible host computer is preferable. It was felt that the job which was done using the BBC computer with BASIC language as a host computer was extremely limited. The SAMOVE routine should be modified in order to have a greater flexibility in generating a trajectory. Finally, the control algorithm inside the EPROM in the motor controller should also be modified to include some additional summing points to allow more advanced technique eg. feedforward control. But, since the current control rate is 20 ms, which is the lower limit for an industrial robot [Khosla and Neuman, 1985], this speed limit should be considered in modifying the control algorithm inside the EPROM.

The same input signals applied to joint 1, jont 2 and joint 3 and the response of each joint of the robot arm which was obtained from the experiment as well as the corresponding controller gain will be used in developing and vaidating the robot models in the next chapter.

# CHAPTER 7

# VALIDATION OF TQ MA2000 ROBOT ARM MODEL

## 7.1 Introduction

In developing a control system of a robot manipulator, knowledge of inertial parameters of the manipulator ie. mass, centre of mass and moment of inertia as well as parameters of actuators is important. Most manipulators are usually designed based on their kinematic information and not on their dynamic information. This is because it is difficult to obtain knowledge of dynamic information in which the manipulators have irregular shapes and nonuniform mass distribution, even by dismantling and weighing each component, and not even the manufactures know them.

In this chapter, another approach to obtain dynamics information of a manipulator is carried out, ie. by identifying the parameters of a mathematical model which was derived in chapter 3 through some simplifications in a closed loop manner. There are many papers which discuss parameter estimation of a robot manipulator with some experiments eg. [Olsen and Bekey, 1985; An, Atkenson and Hollerbach, 1985; Atkenson, An and Hollerbach, 1986; Moon, Chung, Cho and Gweon, 1986; West, Papadopoulos, Dubowsky and Cheah, 1989]. Most of them, however, use the inverse dynamics method in estimating the parameter values ie. the errors of generalized torques are used to identify the parameters. This chapter deals with estimating parameters of the first three links in a forward dynamics manner in which the errors of joint response are used to identify the parameters and sensitivity of each parameter are investigated. The mathematical model will then be validated quantitatively using the distortion technique which is based on the trasfer

function technique since it is more practical over the time domain approach. In the previous work, the distortion quantitative validation technique was applied to a two link planar robot arm [Kartowisastro, 1989].

## 7.2 Basic Model

As derived in chapter 3, the dynamic behaviour of a robot arm with n degrees of freedom is given by

$$\tau_1 = D(\theta)\,\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) \qquad (7.1)$$

where : $\tau_1$ = an nx1 generalized torque vector

D = an nxn symmetric inertial matrix

H = an nx1 coriolis/centrifugal vector

G = an nx1 gravity vector

$\theta$ = an nx1 joint variable vector

The above equation of motion is the basic equation where energy dissipation is not considered. This is the model which is commonly used. By applying the distortion technique method to validate the model quantitatively, it is now possible to tell whether the model is capable of explaining the transient record or not. Each link of the robot manipulator has ten inertial parameters, ie. the mass, the three elements of the centre of mass vector and the six elements of the inertia tensor matrix [Sheu and Walker, 1989]. All these inertial parameters, however, depend on the two most fundamental parameters, namely the length of link and the mass of link as discusssed in chapter 5. For each link, two fundamental parameters have been selected for implementing the distortion technique, with the exception of link

3 where another parameter is taken into account and also considered as a fundamental parameter ie. mass of load.

In this mathematical model, each link is assumed rigid and considered as a line with uniform distribution of mass. Link 1 is assumed to have a cylindrical shape with a radius of $R'_1$, and the end effector at the end of link 3 is treated as a load. The joint and motor of each link is assumed massless. These assumptions yield a simple mechanical model where the centre of mass of each link lies in the centre of the corresponding link with the exception of link 3 where the existence of load shifts the centre of mass along its x axis.

From figure 7.1, the Denavit-Hartenberg parameters are

$$\alpha_1 = \pi/2 \tag{7.2}$$

$$\alpha_2 = 0 \tag{7.3}$$

$$\alpha_3 = 0 \tag{7.4}$$

$$d_1 = 0 \tag{7.5}$$

$$d_2 = B_2 \tag{7.6}$$

$$d_3 = B_3 \tag{7.7}$$

$$a_1 = 0 \tag{7.8}$$

$$a_2 = A_2 \tag{7.9}$$

$$a_3 = A_3 \tag{7.10}$$

and $\theta_i$'s are the joint variables for i=1,2,3.

The vectors used in Newton-Euler approach, as shown in figure 7.2 are given by

$$ {}^{1}p_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.11} $$

$$ {}^{2}p_2 = \begin{bmatrix} A_2 \\ 0 \\ B_2 \end{bmatrix} \tag{7.12} $$

$$ {}^{3}p_3 = \begin{bmatrix} A_3 \\ 0 \\ -B_3 \end{bmatrix} \tag{7.13} $$

$$ {}^{1}\bar{s}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.14} $$

$$ {}^{2}\bar{s}_2 = \begin{bmatrix} -A_2/2 \\ 0 \\ 0 \end{bmatrix} \tag{7.15} $$

$$ {}^{3}\bar{s}_3 = \begin{bmatrix} -S_{3x} \\ 0 \\ 0 \end{bmatrix} \tag{7.16} $$

$$ {}^{1}\bar{c}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{7.17} $$

$$ {}^{2}\bar{c}_2 = \begin{bmatrix} A_2/2 \\ 0 \\ B_2 \end{bmatrix} \qquad (7.18) $$

$$ {}^{3}\bar{c}_3 = \begin{bmatrix} A_3 - S_{3x} \\ 0 \\ -B_3 \end{bmatrix} \qquad (7.19) $$

where :

${}^{i}p_i$ = a translation vector of the origin of the coordinate system i from the origin of the coordinate system i-1 and expressed in the coordinate system i.

${}^{i}\bar{s}_i$ = the centre of mass of link i from the origin of the coordinate system i and expressed in the coordinate system i.

${}^{i}\bar{c}_i$ = the centre of mass of link i from the origin of the coordinate system i-1 and expressed in the coordinate system i.

$S_{3x}$ = the centre of mass of link 3 in the **x** direction from the origin of the coordinate system 3.

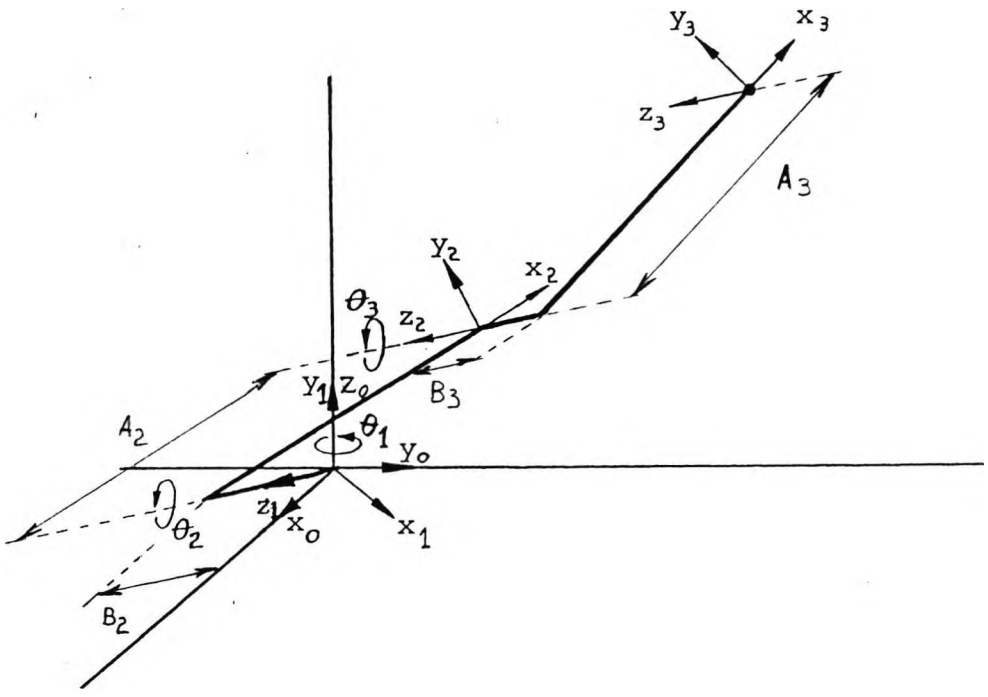



Figure 7.1

The D-H representation of TQ MA2000 robot arm.

Since the mass of the end effector is assumed to be concentrated at the distal end of link 3, the addition of this mass to link 3 will alter the centre of mass of the integrated link 3 to a new position. Hence, the inertia matrix of this link will also be affected. Consequently, this in turn will change the dynamic behaviour of the robot arm.

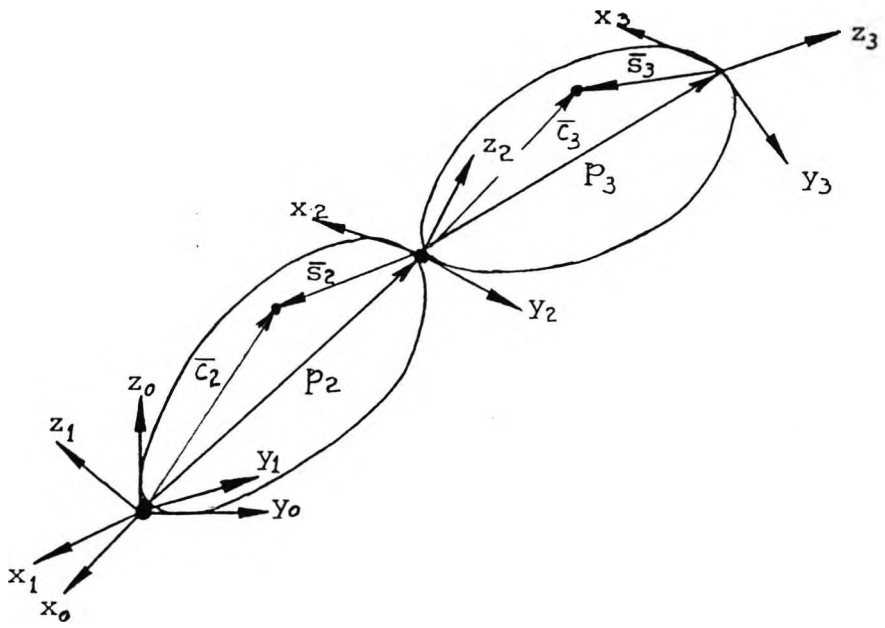



Figure 7.2

Vectors used in the Newton-Euler equations.

Assuming that link 3 and the mass of end effector, MLOAD, may be isolated as shown in figure 7.3, then the new position of the centre of mass of link 3 may be determined as :

$$MM_3 = m_3 + MLOAD \tag{7.20}$$

From basic mechanics,

$$m_3 \times A_3/2 + MLOAD \times A_3 = MM_3 \times (A_3 - S_{3x}) \tag{7.21}$$

Thus,

$$S_{3x} = A_3 - \frac{m_3 \times A_3/2 + MLOAD \times A_3}{MM_3} \qquad (7.22)$$

where $MM_3$ is the mass of link 3 after the mass of end effector is taken into account.

The inertia tensor matrix of each link about its joint, which is expressed in its own coordinate frame, is given by

$$ {}^{1*}I_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & {}^{1*}I_{1yy} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (7.23)$$

$$ {}^{2*}I_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & {}^{2*}I_{2yy} & 0 \\ 0 & 0 & {}^{2*}I_{2zz} \end{bmatrix} \qquad (7.24)$$

$$ {}^{3*}I_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & {}^{3*}I_{3yy} & 0 \\ 0 & 0 & {}^{3*}I_{3zz} \end{bmatrix} \qquad (7.25)$$

where :

$$ {}^{1}I^{*}_{1yy} = \frac{1}{2} m_1 R'^{2}_{1} \tag{7.26} $$

$$ {}^{2}I^{*}_{2yy} = {}^{2}I^{*}_{2zz} = \frac{1}{3} m_2 A_2^{2} \tag{7.27} $$

$$ {}^{3}I^{*}_{3yy} = {}^{3}I^{*}_{3zz} = \frac{1}{3} MM_3 A_3^{2} \left( \left( \frac{S_{3x}}{A_3} \right)^{3} + \left( \frac{A_3 - S_{3x}}{A_3} \right)^{3} \right) + $$

$$ MM_3 (A_3 - S_{3x})^{2} \tag{7.28} $$

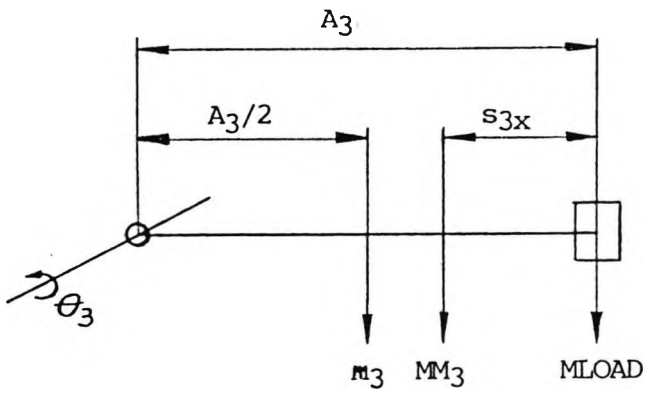



Figure 7.3

Link 3 with load.

Figure 7.4 shows the block diagram of the mathematical model. The power amplifier is assumed linear and there is no back lash in the gear box.

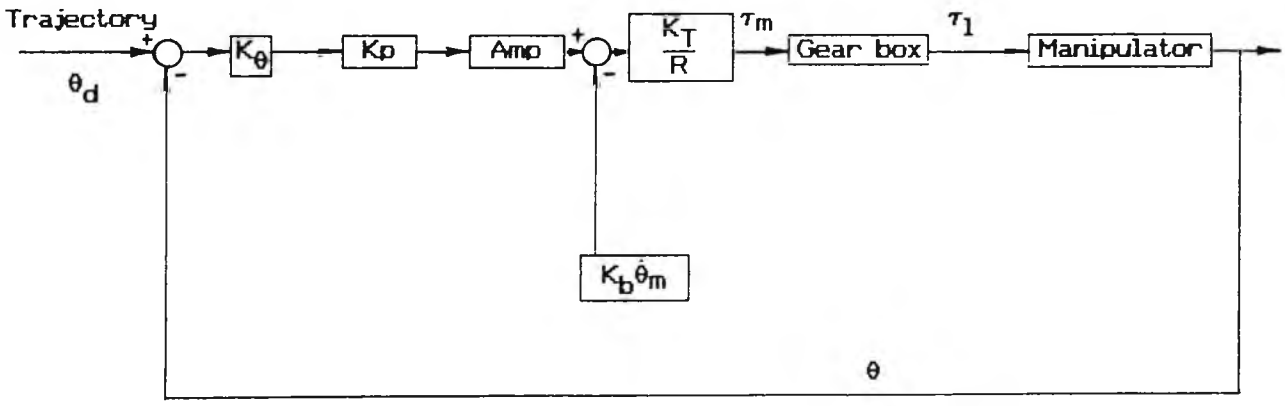



Figure 7.4

Closed loop system block diagram.

Simplification is further made by neglecting the inductance of the actuator since the electrical time constant is much smaller than the mechanical time constant. For the sake of convenience, the information about actuators used by TQ MA2000 [TecQuiptment, 1986; Escap, 1986/87] given in the appendix C is written in S.I. units as shown in table 7.1 where both link 1 and link 3 use the same actuators ie. MAXON and link 2 uses ESCAP.

The dimensions from the technical drawing which is obtained from the manufacturer have been used to obtain the unoptimised parameters. The weight of each link was obtained by evaluating its volume and then multiplying by its density. The density of aluminium which is used by the robot is found to be 2698 $kg/m^3$ [Anderson and Haupin, 1978]. These unoptimised parameters were used as starting points in excercising the hill climbing process. From all the information obtained, the unoptimised parameter values are tabulated in table 7.2 where MSQE stands for

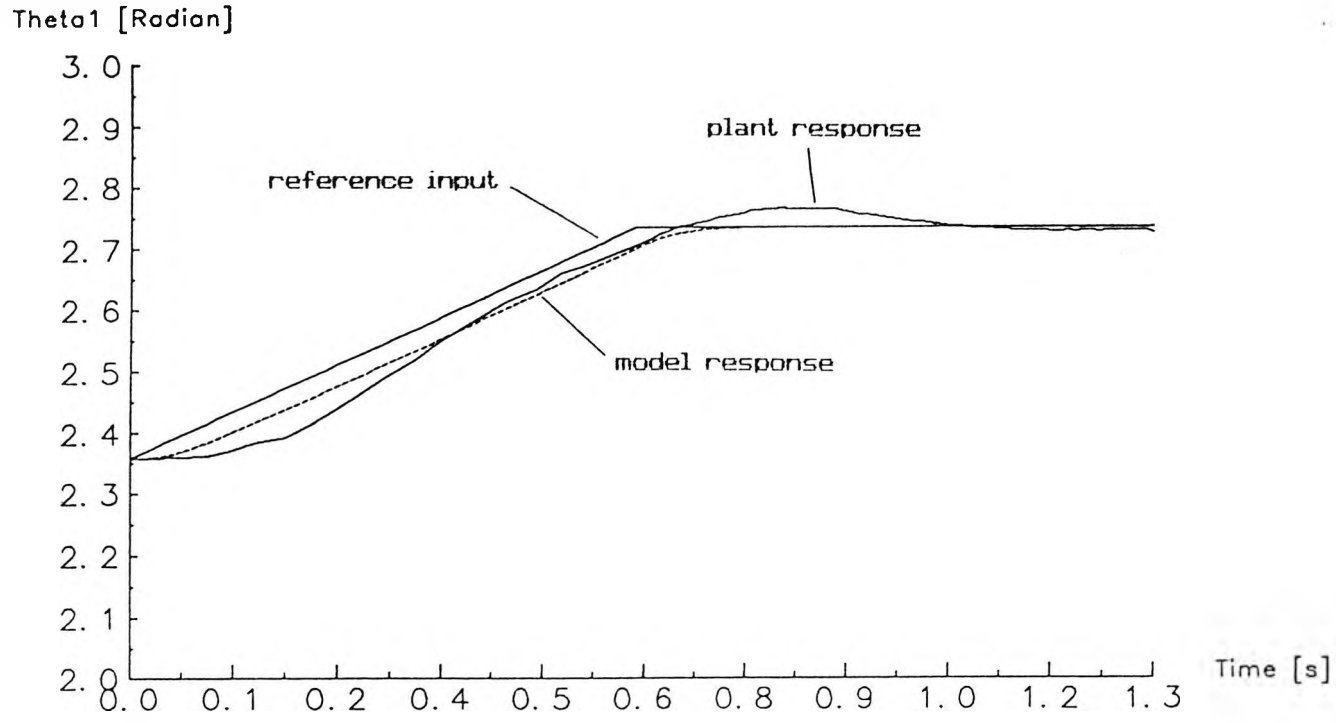
mean squared error. Figures (7.5), (7.6) and (7.7) show the responses of the unoptimised model and their corresponding errors when a ramp step input is given to each joint.

| | | MAXON | ESCAP |
|---|---|---|---|
| Power output | Watt | 11 | 12 |
| Nominal voltage | Volt | 18 | 15 |
| No load current | mA | 43.3 | 20 |
| Terminal resistance | Ohm | 5.49 | 4.5 |
| Rotor inductance | mH | 0.82 | 0.6 |
| Torque constant | mNm/A | 20.8 | 33 |
| Rotor inertia | $kgm^2.10^{-7}$ | 18.3 | 32 |
| Back EMF constant | V/rad/s | Not available | 0.0334 |
| Mass | kg | 0.174 | 0.230 |
| Viscous damping constant | $Nms/rad.10^{-6}$ | Not available | 1 |

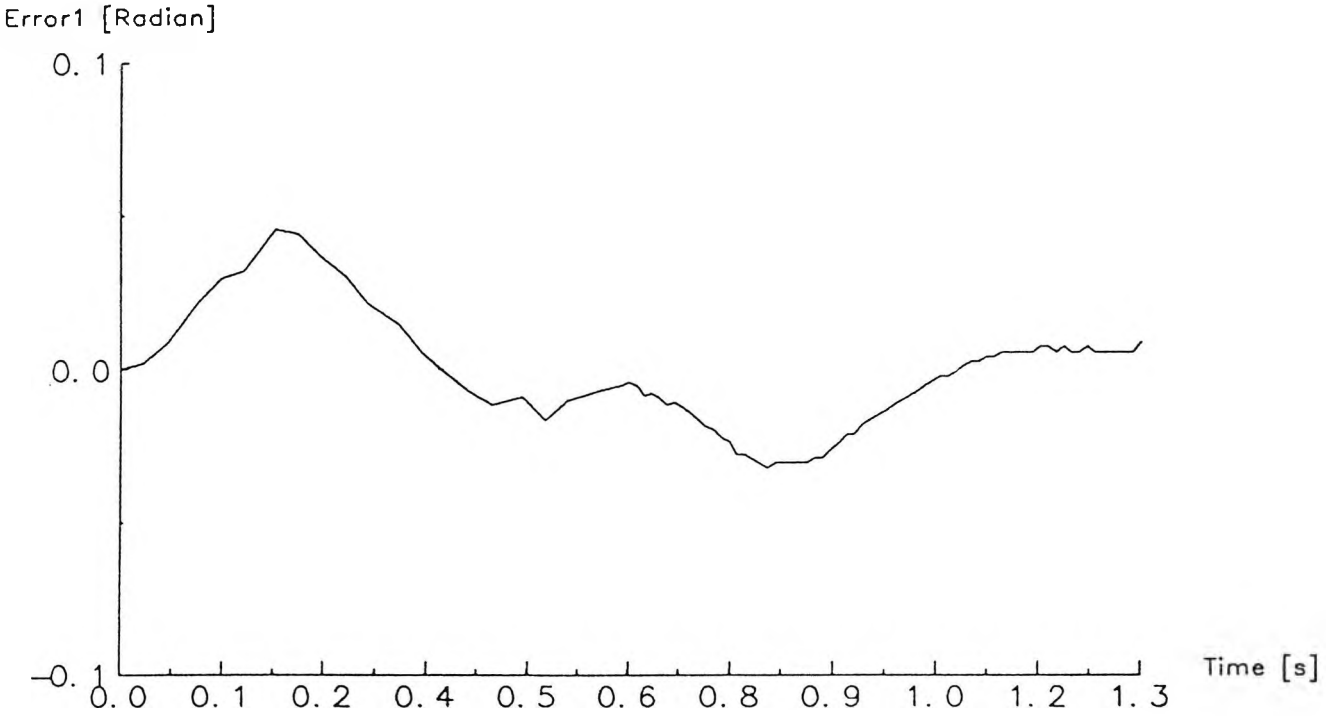
Table 7.1

Technical data of actuators

By assuming initially that the value of back EMF constant of actuator 2 is correct, there are 8 parameters to identify, namely mass of link 1, $m_1$; mass of link 2, $m_2$; mass of link 3, $m_3$; Mass of load, MLOAD; radius of link 1, $R'_1$; length of link 2, $A_2$; length of link 3, $A_3$; and back EMF constant of actuator 1 and actuator 3, $K_{b1-3}$



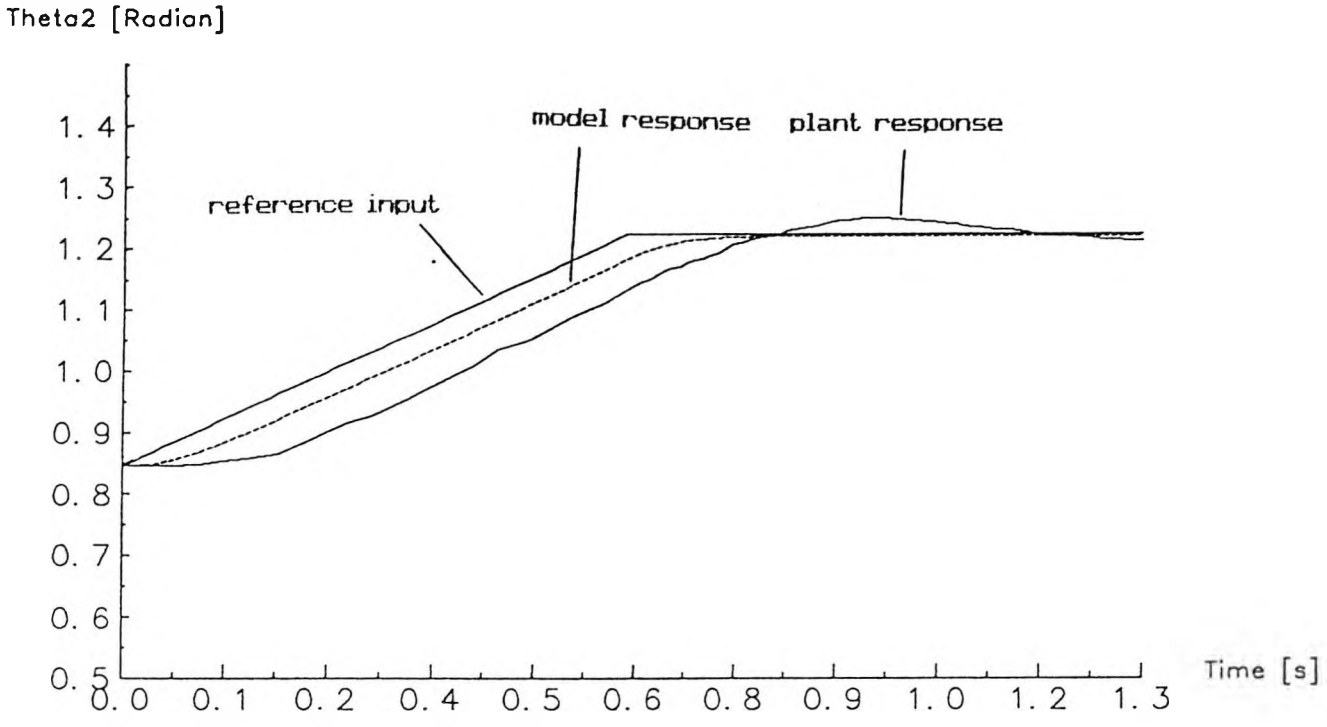

(a) Joint 1 response.



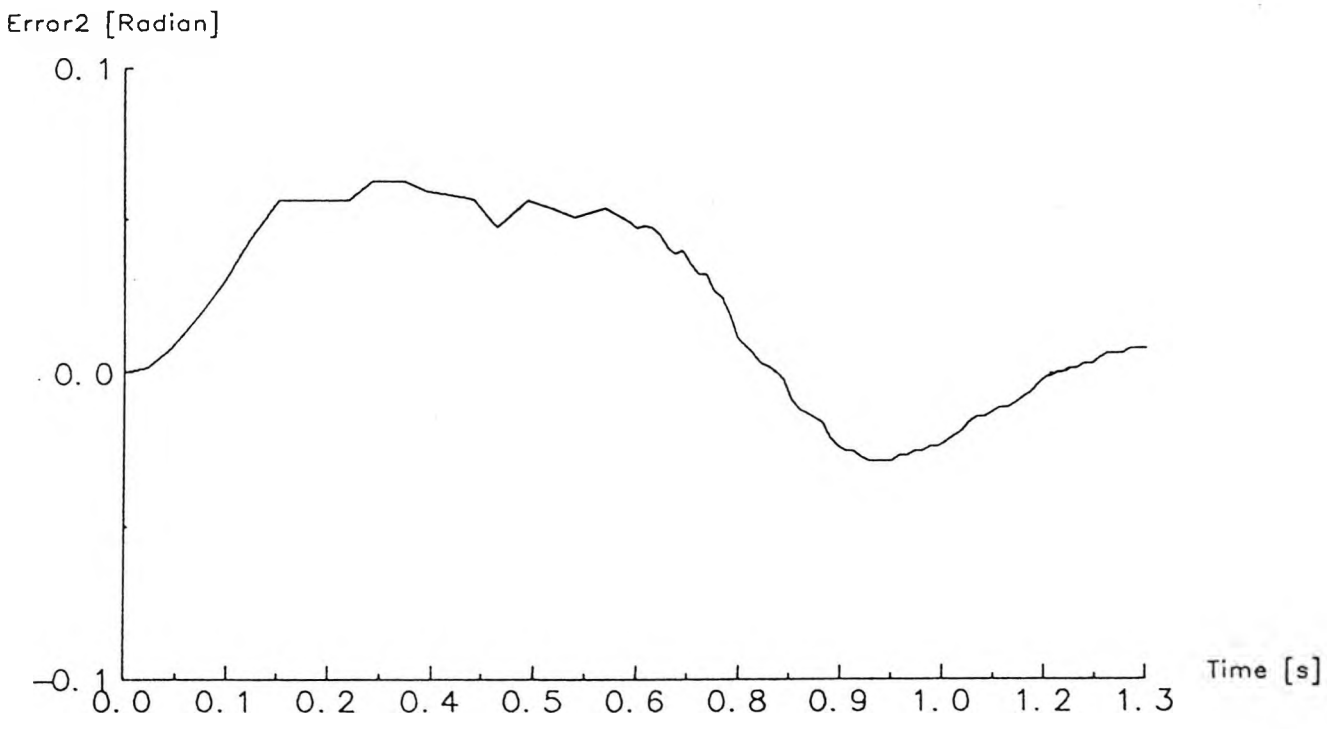

(b) Joint 1 error.

Figure 7.5

Unoptimised model.



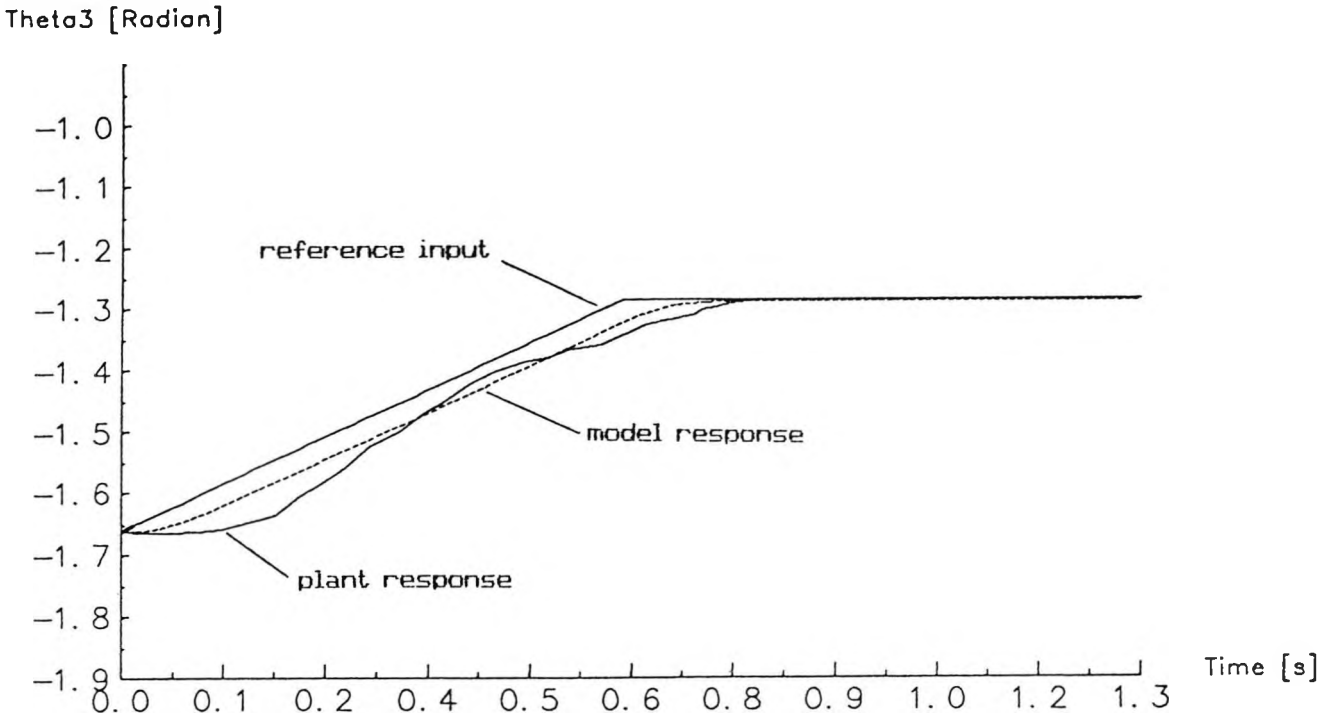

(a) Joint 2 response.



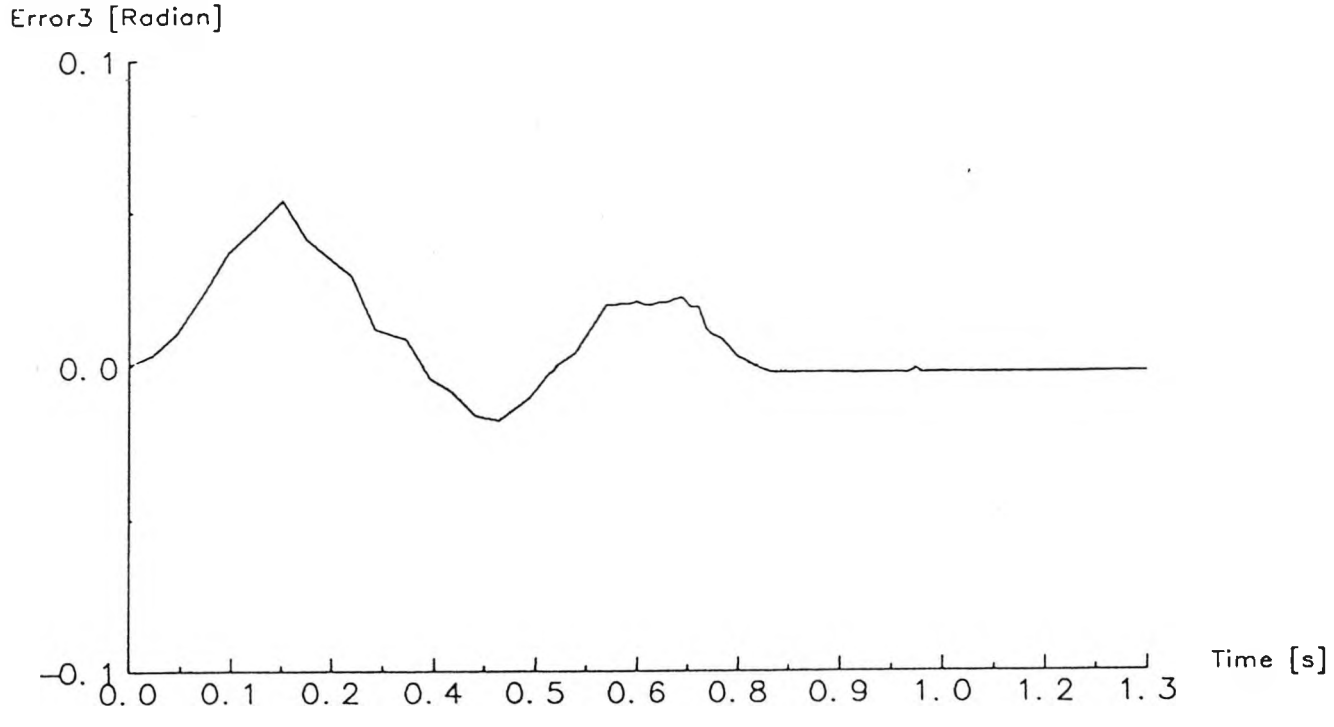

(b) Joint 2 error.

Figure 7.6

Unoptimised model.




(a) Joint 3 response.




(b) Joint 3 error.

Figure 7.7

Unoptimised model.

Since all these parameter values are always positive, the lower bounds of all parameters are assigned to zero. Assigning upper bounds of parameters are difficult without any understanding of physical knowledge. Upper bounds are then determined by evaluating the maximum gravity force caused by the upper bound of each link. This gravity force in turn must be less than the maximum torque that the corresponding actuator can give. For link 1 where gravity force does not exist, the upper bound is simply 5 times its unoptimised value (there appear to be no formal technique for choosing this value of 5 times, so the number had to be chosen so that the result could be obtained). The upper bound of back EMF constant is found by trial and error from simulation exercises.

| | | |
|---|---|---|
| $m_1$ | kg | 2.0468 |
| $m_2$ | kg | 0.4040 |
| $m_3$ | kg | 0.2321 |
| MLOAD | kg | 1.0 |
| $R'_1$ | m | 0.0493 |
| $A_2$ | m | 0.23 |
| $A_3$ | m | 0.137 |
| $K_{b1-3}$ | V/rad/s | 0.03 |
| MSQE | $Radian^2$ | $2.1428.10^{-3}$ |

Table 7.2

Unoptimised parameters of the basic model.

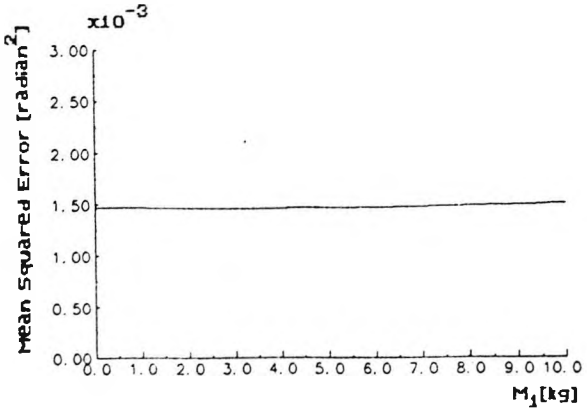
The same ramp step input signal as applied to the real system during an experiment is given to the model under study. The difference between the output of

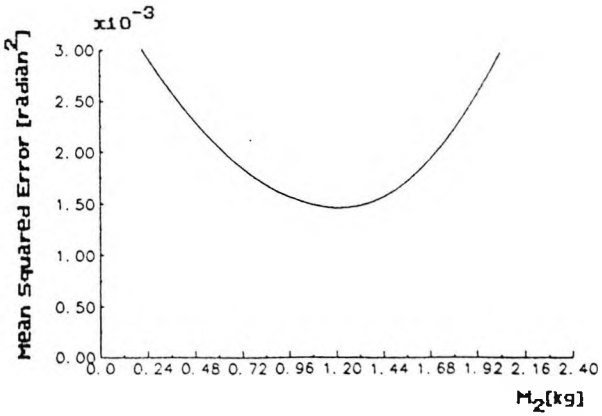
the real system and the output of the model in terms of the combined mean squared error of joint 1, joint 2 and joint 3 is then minimised to obtain an optimum model. The parameter optimisation curves are obtained and shown in figure (7.8) and the result of optimised parameters are tabulated in table 7.3. Figures (7.9), (7.10) and (7.11) show the plant and fitted model responses and their corresponding errors.

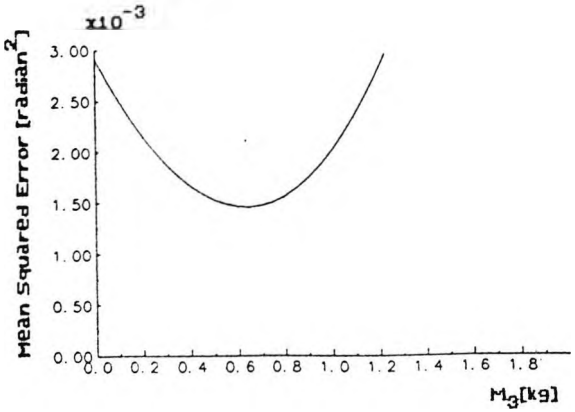
| | | |
|---|---|---|
| $m_1$ | kg | 2.93606 |
| $m_2$ | kg | 1.22802 |
| $m_3$ | kg | 0.64424 |
| MLOAD | kg | 1.11327 |
| $R'_1$ | m | 0.05376 |
| $A_2$ | m | 0.89820 |
| $A_3$ | m | 0.32245 |
| $K_{b1-3}$ | V/rad/s | 0.03776 |
| MSQE | $Radian^2$ | $1.4651.10^{-3}$ |

Table 7.3

Optimised parameters of the basic model.



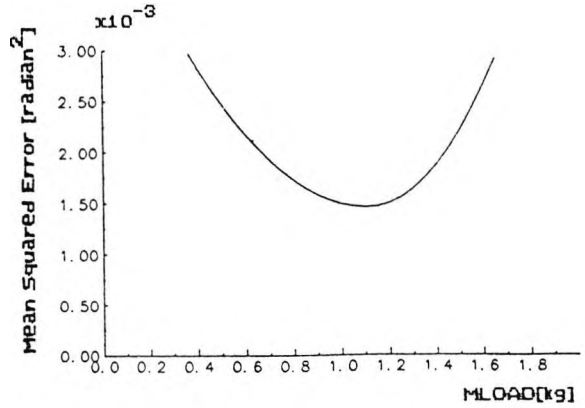
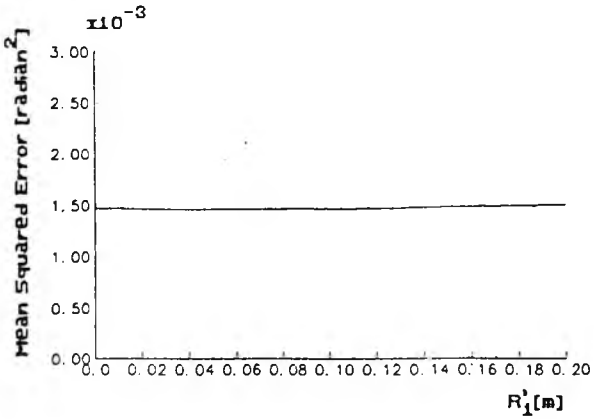
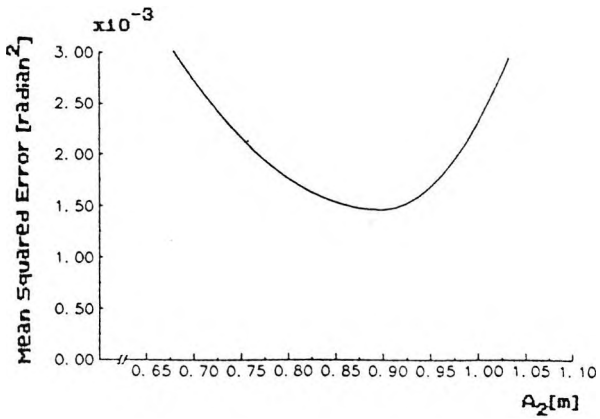



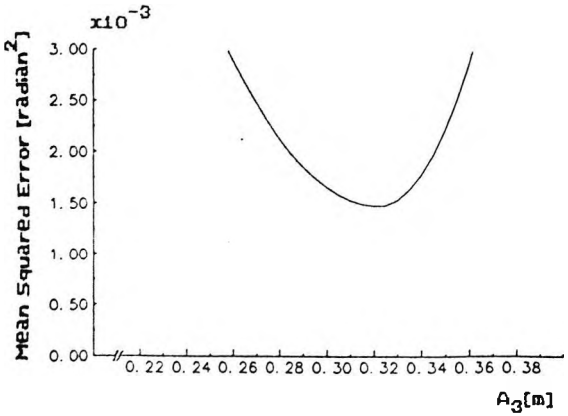

Figure 7.8

Parameter optimisation curves

of a basic model



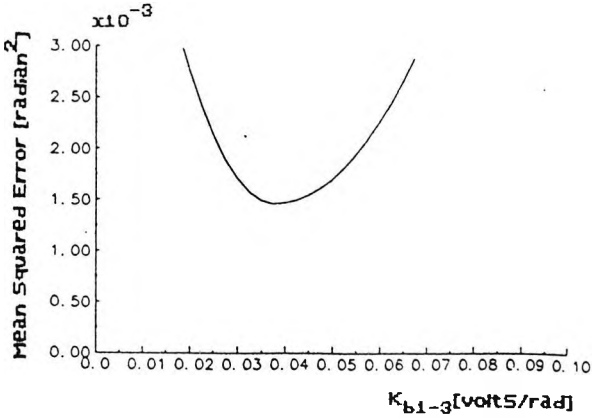
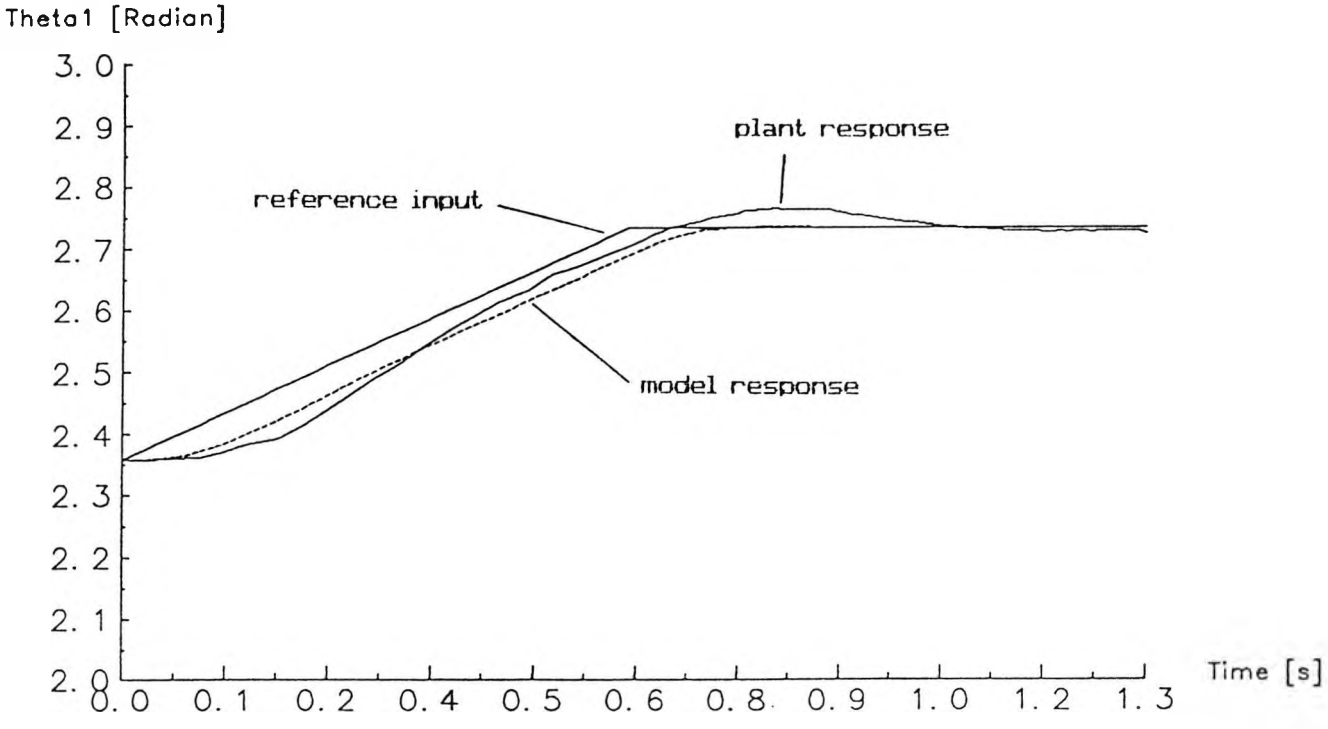
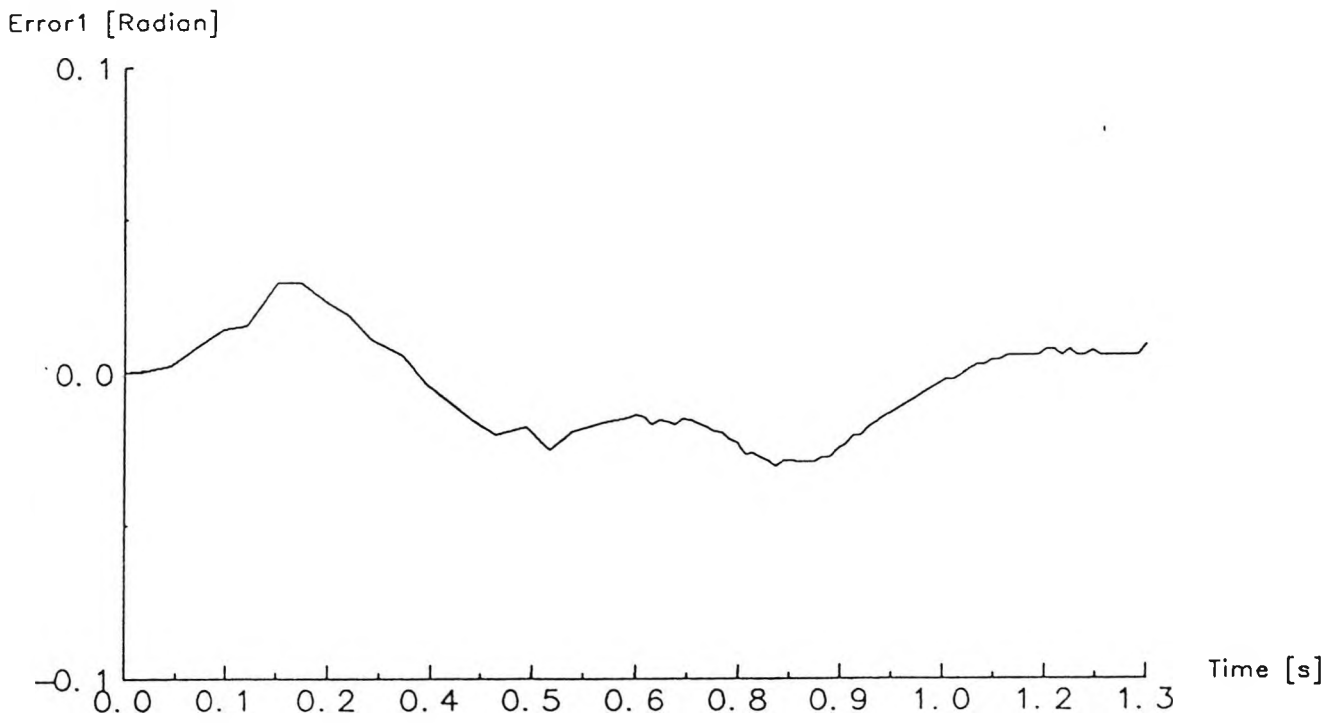




Figure 7.8

(Continued)



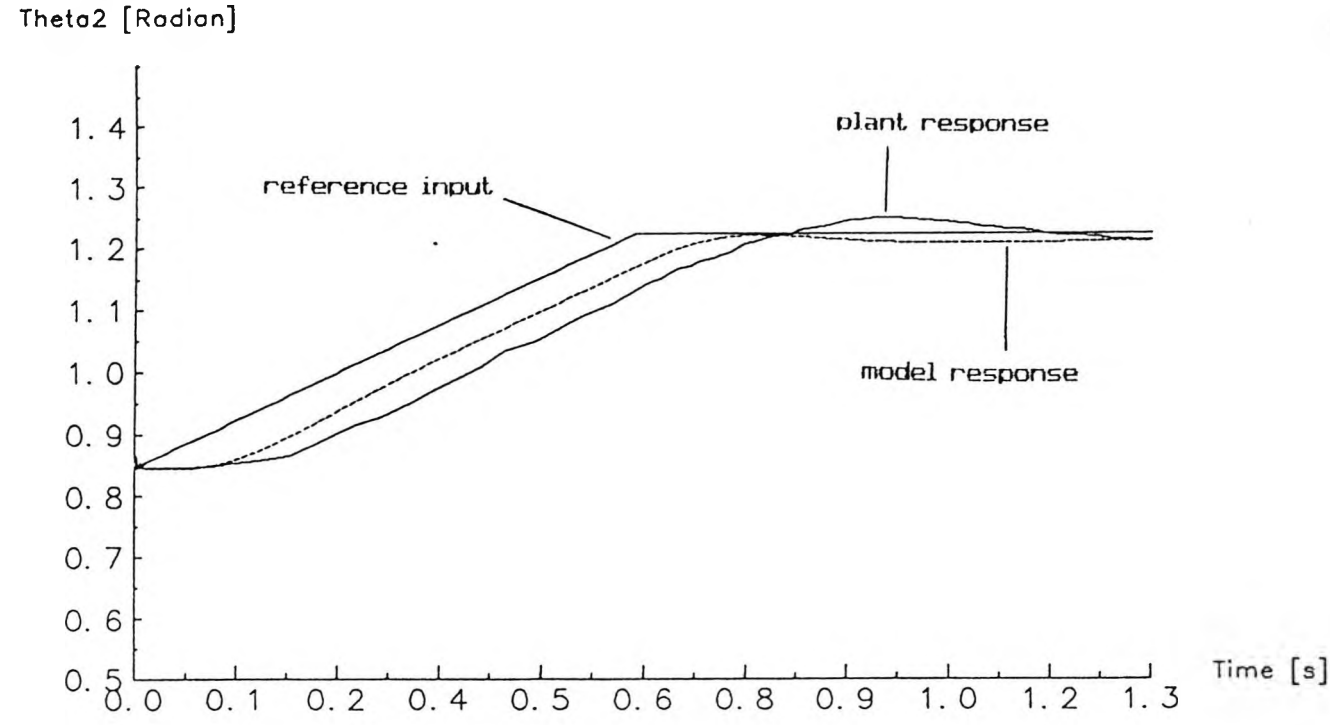

(a) Joint 1 response.



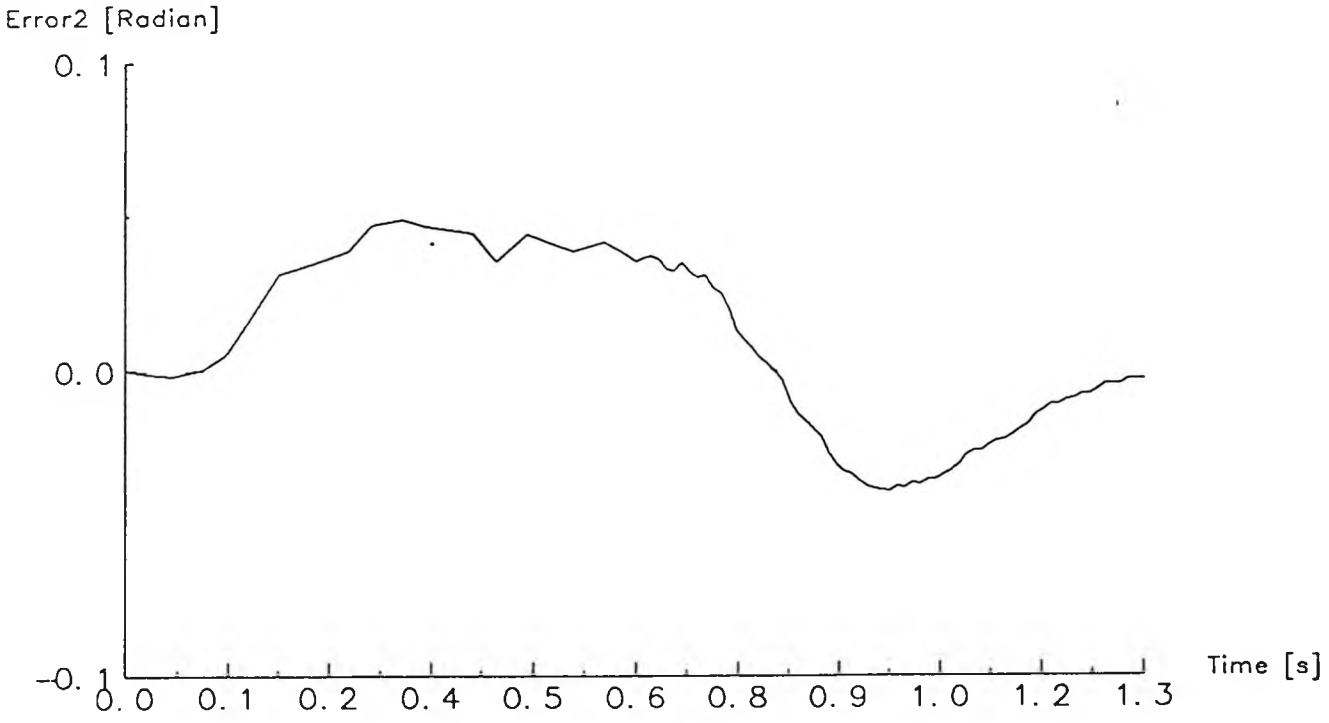

(b) Joint 1 error.

Figure 7.9

Basic model.



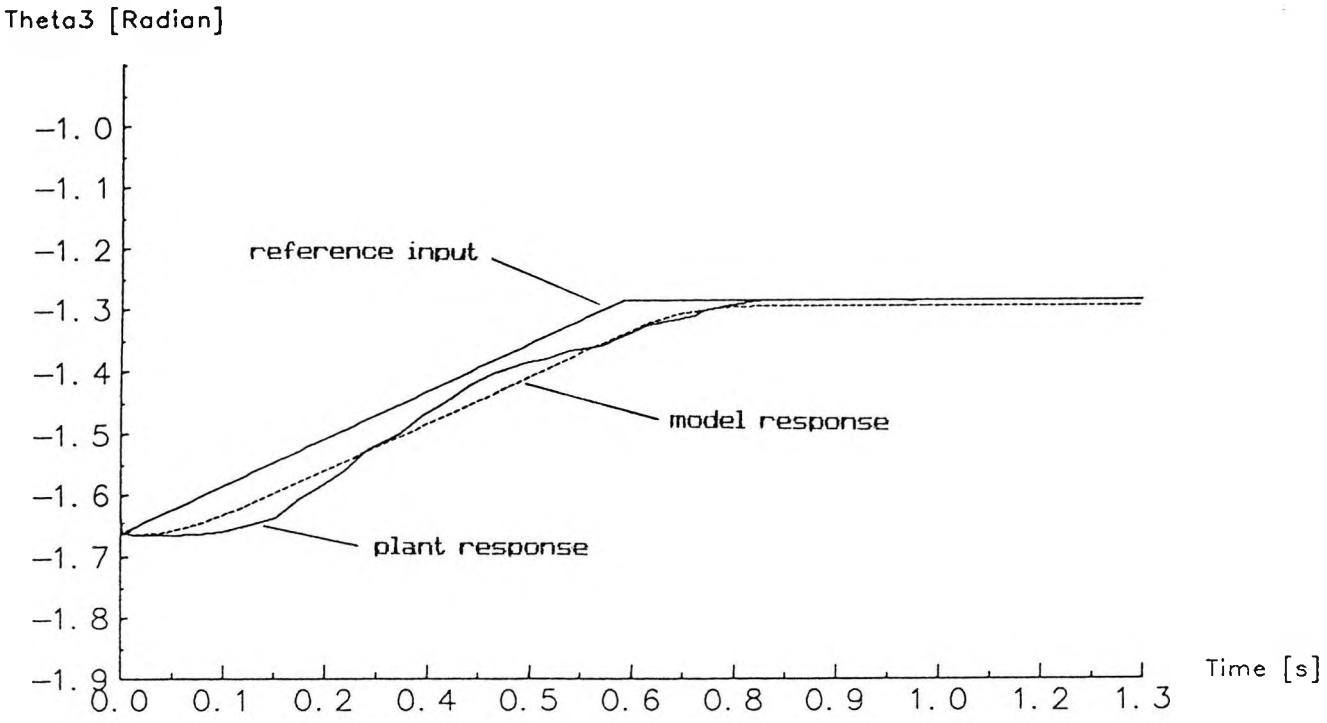

(a) Joint 2 response.



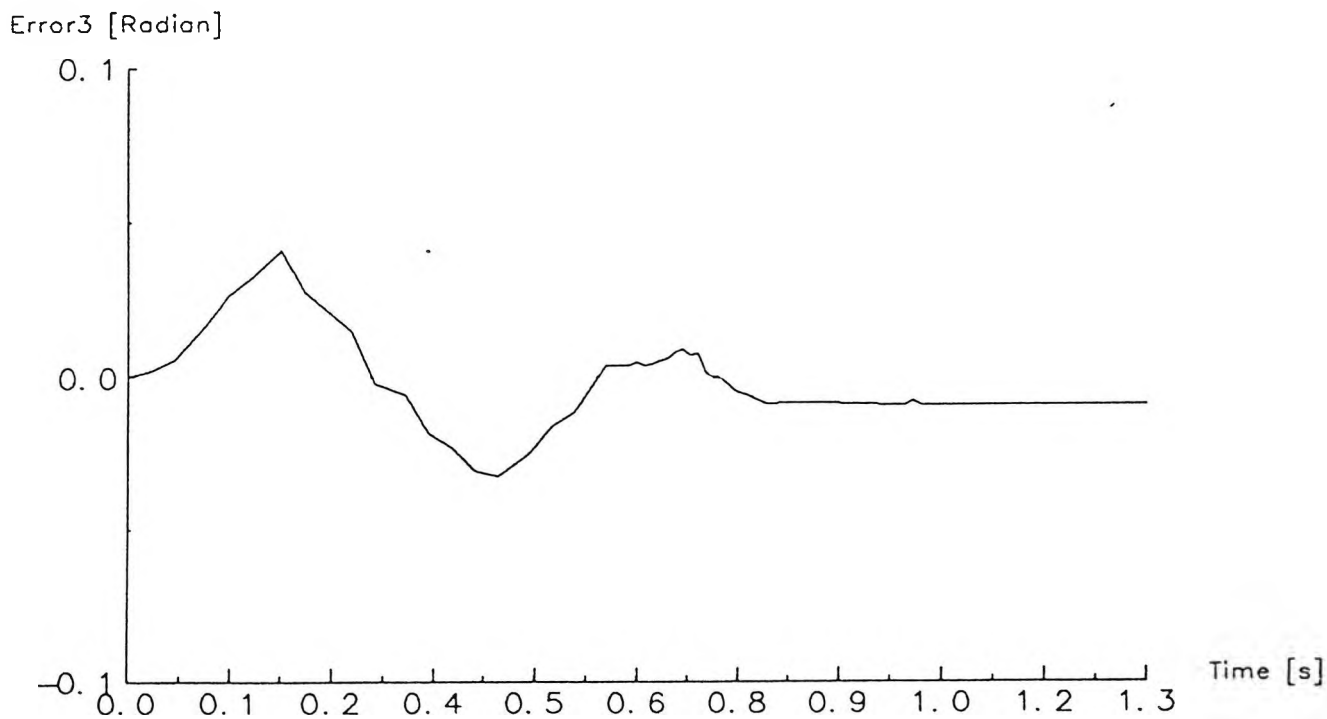

(b) Joint 2 error.

Figure 7.10

Basic model.




(a) Joint 3 response.




(b) Joint 3 error.

Figure 7.11

Basic model.

### 7.2.1 Sensitivity Analysis

From the parameter optimisation curves in figure (7.8), it can be seen that the fundamental parameters of the distal link are more sensitive than the fundamental parameters of the proximal link. The modified Newton-Euler (N-E) algorithm, given in appendix B, can explain this sensitivity behaviour. The centre of mass of each link is a function of its length, so that the length of each link determines the value of its centre of mass. From the backward iteration of the N-E algorithm, which gives dynamic terms of the system, it is seen that the changes of the centre of mass and mass of the distal link affect the generalized torques of this link as well as the proximal links while the changes of the centre of mass and mass of the proximal link only affect its own link and do not affect the distal links. In another words, the parameters of the distal link are more sensitive than the parameters of the proximal link. In addition, the centre of mass value of link 2 is only affected by its length and not by its mass. As a consequence, its linear acceleration is also affected. Hence, for link 2, its length is more sensitive than its mass.

In the case of link 1, its motion is restricted where it has only one degree of freedom about its $Y_1$ axis. Neither $m_1$ nor $R'_1$ affect the movement of the manipulator directly. They only affect the movement of the robot through its moment of inertia about the $Y_1$ axis. In another words, the fundamental parameters of link 1 affect only the rotational component and not the translational component of the robot movement. Because of this and since link 1 is the proximal link, the mass and the radius of link 1 are insensitive in practice.

### 7.2.2 Fidelity of The Model

From a set of parameter optimisation curves, which are shown in figure (7.8), the $\alpha_{jd-}$ and $\alpha_{jd+}$ values to double the minimum mean squared error value can be determined. A damping ratio of 0.5 is assigned since it provides a reasonable rule covering a wide range of conditions [Butterfield, 1989]. The variance of each parameter to eliminate on its own the differences between the measured response of the real system and the model output can then be determined and their standard deviations are tabulated in table 7.4 where ' ››› ' denotes a very large value.

| | | |
|---|---|---|
| $m_1$ | kg | ››› |
| $m_2$ | kg | 0.9342 |
| $m_3$ | kg | 0.4001 |
| MLOAD | kg | 0.6560 |
| $R'_1$ | m | ››› |
| $A_2$ | m | 0.1674 |
| $A_3$ | m | 0.0557 |
| $K_{b1-3}$ | V/rad/s | 0.0263 |

Table 7.4

Standard deviation of parameters of the basic model.

The parameters involved are expected to have standard deviations, from their corresponding optimal values, of 3% for the radius or length of each link and 10% for the mass of each link and load. The assignment of an expected standard deviation of the back EMF constant is performed by assuming that it has a

Gaussian distribution and its mean value is the optimal value of $K_{b1-3}$. From its physical properties, the back EMF constant value always has a positive value. Choosing an area under the Gaussian tail at $K_{b1-3} = 0$ which lies three standard deviations from the mean value is considered to be reasonable and gives a very small error function value.

The fidelity criterion as given in chapter 4 is then evaluated to assess the validity of the model quantitatively. For this basic model, applying the criterion gives

$$\sum\lambda^2 = \left(\frac{\tau' m_1}{\sigma m_1}\right)^2 + \left(\frac{\tau' m_2}{\sigma m_2}\right)^2 + \left(\frac{\tau' m_3}{\sigma m_3}\right)^2 + \left(\frac{\tau' MLOAD}{\sigma MLOAD}\right)^2 + \left(\frac{\tau' R_1'}{\sigma R_1}\right)^2 +$$

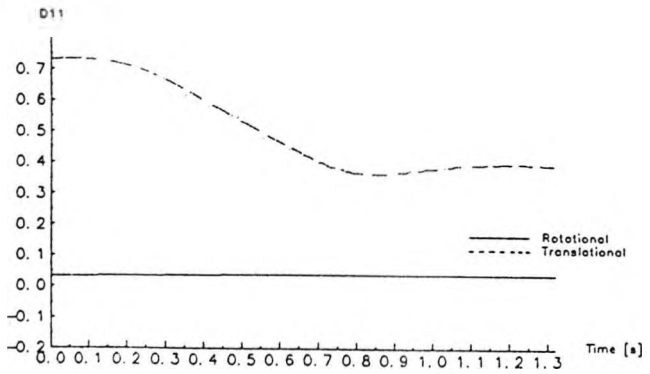
$$\left(\frac{\tau' A_2}{\sigma A_2}\right)^2 + \left(\frac{\tau' A_3}{\sigma A_3}\right)^2 + \left(\frac{\tau' K_{b1-3}}{\sigma K_{b1-3}}\right)^2$$

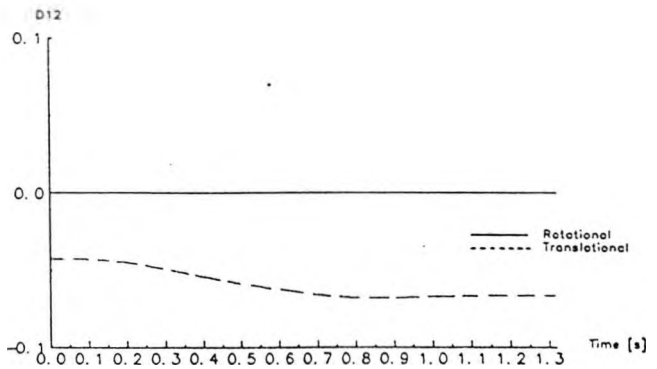
$$= 0.2449$$

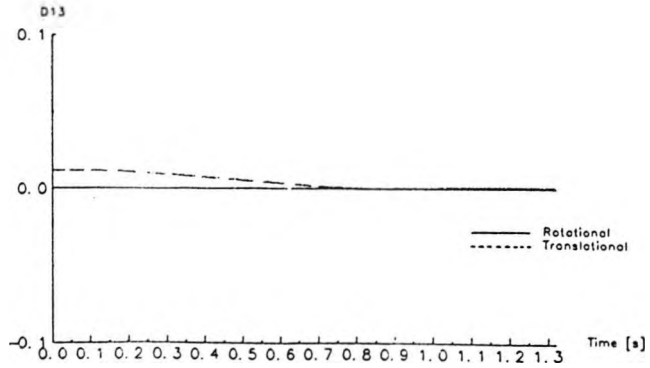
The value of $\sum\lambda^2$ , because it is considerably less than inity, does not satisfy the fidelity criterion of the distortion technique and hence the model is not capable of explaining the recorded response of the real system. Clearly, the model needs improvement.

A further investigation is performed. From figures (7.9) to (7.11), the model response of joint 2 has the largest discrepancy where the real system response is more sluggish. A scrutiny of the system behaviour is performed by observing each component of the D matrix. The rotational and translational components of $D_{ij}$ elements along the trajectory are computed and plotted as shown in figure (7.12). Since the D matrix is a function of $\theta$, the value of each element depends on the

nature of the trajectory. This set of plots shows that the parameters of link 1 are not sensitive since the values of $D_{12}$ and $D_{13}$ elements are very small. This is due to the fact that $\alpha_1 = \pi/2$ and $\alpha_2 = 0$. It can be seen that $D_{22}$ is much larger than $D_{23}$. This implies that link 2 suffers not too much interaction caused by link 3 in comparison with its own inertial value. The behaviour of the system is more clearly understood by studying the components of each torque which exists significantly in the system ie. inertial torques and gravity torques. For the same trajectory, these components are then computed and plotted. Figures (7.13) and (7.14) show the inertial torques and gravity torques, respectively. The amount of interactions among the links are easily noticeable. It is clearly shown that interactions are mainly caused by the gravity torques while interactions due to the induction of inertial torques are relatively small. The sawtooth like shapes of the inertial torques are due to tracking a ramp input signal for a period of 0.65 secs while, after the input signal becomes constant, the inertial torques shapes are smooth and eventually die out in the steady state condition. The interaction inertial torques induced by accelerations of other joints are relatively small in comparison with their own inertial torques. The maximum inertial torques on joint 2 and joint 3 which are caused by their own acceleration are only slightly larger than their corresponding gravity torques. While the inertial torques have large values only at the time when there is a sudden change in the trajectory and zero when the speed is constant, the gravity torques have relatively more constant values. From the plots, it is noticed that joint 2 suffers much larger interaction caused by gravity torques than interaction caused by inertial torques. Since the same trajectories are applied to all joints, there is no significant difference between the interaction inertial torque on joint 2 induced by the acceleration of joint 3 and the interaction inertial torque on joint 3 induced by the acceleration of joint 2.



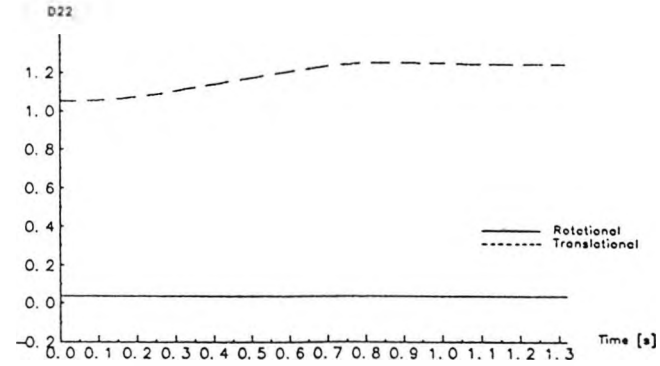
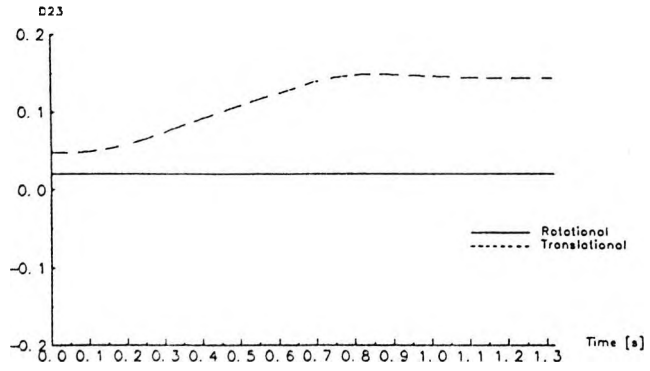




Figure 7.12

Elements of D matrix



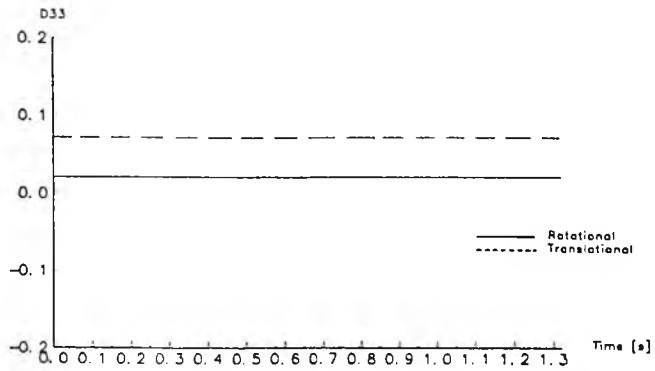
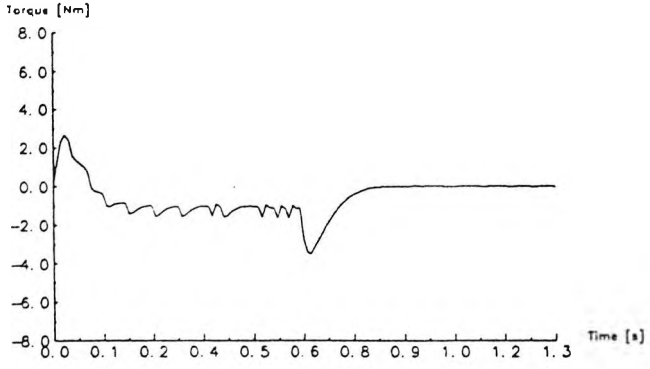




Figure 7.12

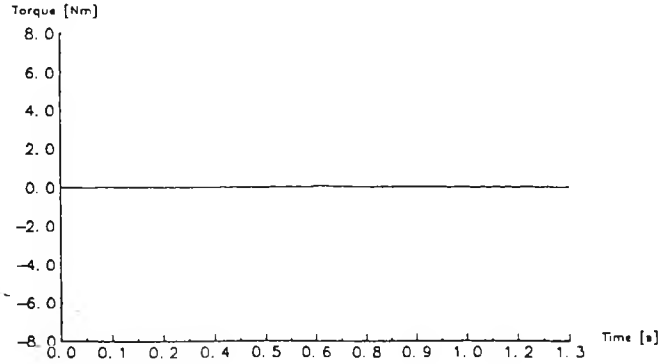
(Continued)






(a)

Inertial torque at joint 1.



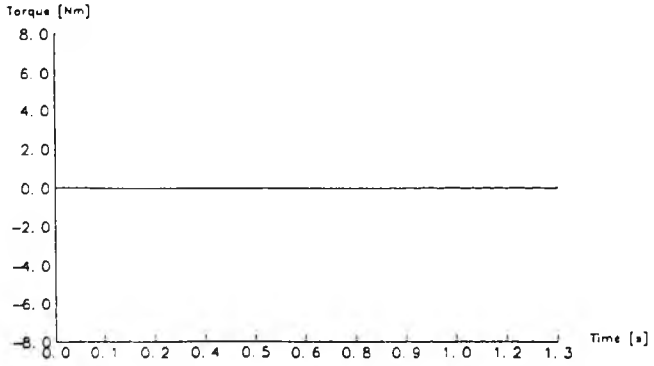

(b)

Interaction inertial torque at joint 1,

induced by joint 2.

Figure 7.13

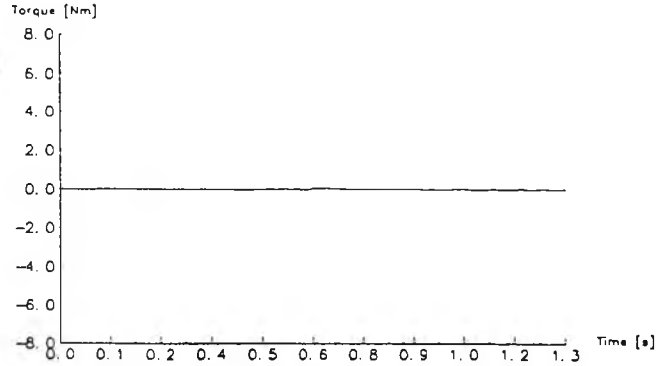
Inertial torques.




(c)

Interaction inertial torque at joint 1,

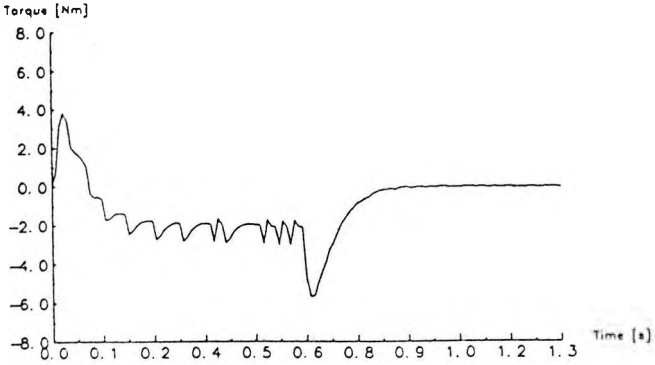
induced by joint 3.




(d)

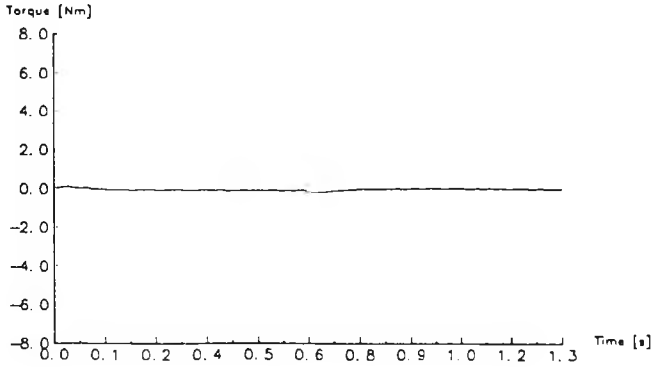
Interaction inertial torque at joint 2,

induced by joint 1.




(e)

Inertial torque at joint 2.



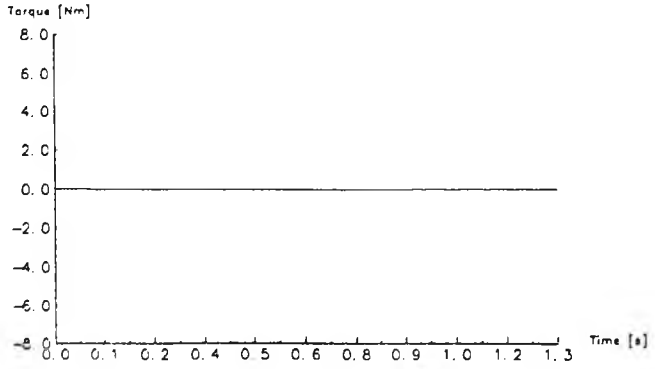

(f)

Interaction inertial torque at joint 2,

induced by joint 3.

Figure 7.13

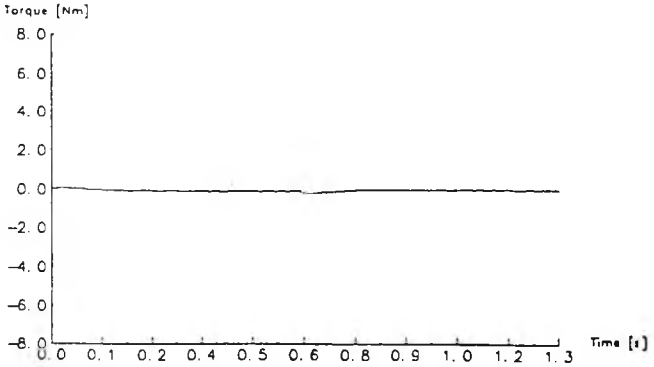
(Continued)




(g)

Interaction inertial torque at joint 3,

induced by joint 1.




(h)

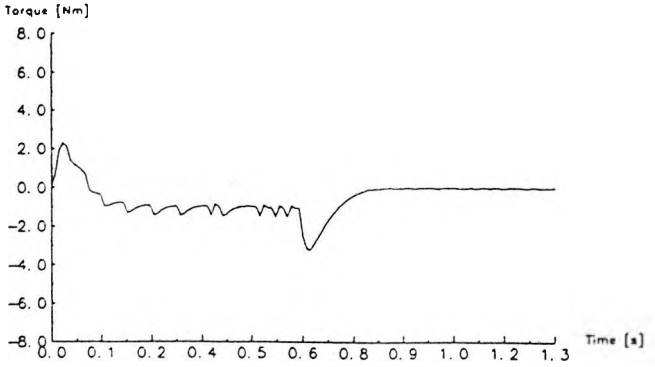
Interaction inertial torque at joint 3,

induced by joint 2.

Figure 7.13

(Continued)



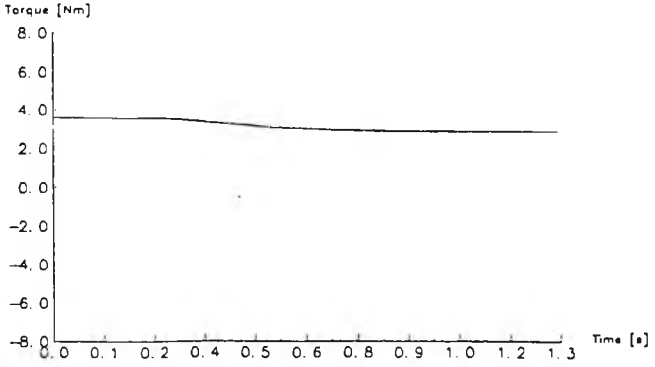



(i)

Inertial torque at joint 3.

Figure 7.13

(Continued)



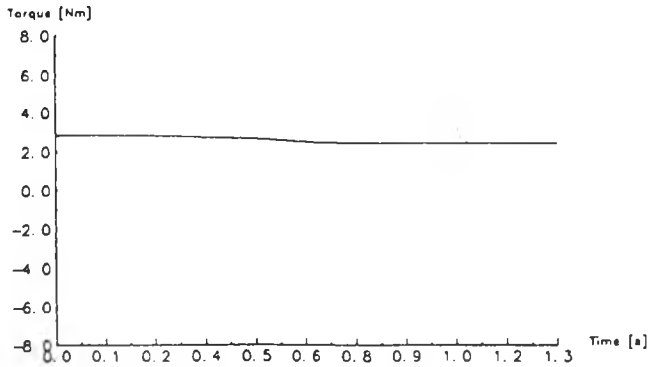

(a)

Gravity torque at joint 2,

caused by link 2 and link 3.

Figure 7.14

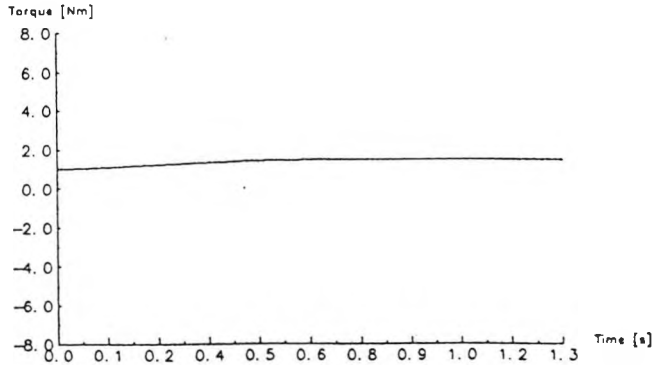
Gravity torques.




(b)

Gravity torque at joint 2,

caused by link 3 only.




(c)

Gravity torque at joint 3.

Figure 7.14

(Continued)

This knowledge implies that the fundamental parameters have an important role in the interactions of joints while other parameters such as back EMF constant give insignificant interactions. In other words, parameters other than fundamental parameters affect only their corresponding joints. Hence, it leads to an idea that parameters of link 2 other than mass and length can be manipulated to obtain a better fit of joint 2 response. This parameter is the coefficient of back EMF. It must be understood that the **D** matrix for a particular trajectory is a function of mass and length of link only and not a function of back EMF coefficient. Although the value of back EMF coefficient of motor 2 is available from the data sheet, its value still needs to be optimised due to a change of magnetic field. Hence, to improve the model, a new parameter, back EMF coefficient $K_{b2}$, should be optimised.

With a new parameter $K_{b2}$ to be optimised, there are 9 parameters to identify. The upper and lower bounds are assigned in the same manner as $K_{b1-3}$ and the prescribed value from the data sheet is used as a starting value. Figures (7.15), (7.16) and (7.17) show the model responses and its errors. It can be seen that joint 2 has a better fit than it had in the previous model. The parameter optimisation curves are then obtained and shown in figure (7.18) and the result of optimised parameters are tabulated in table 7.5. The standard deviations of each corresponding parameters in order to eliminate the errors between the recorded response of the real system and the model output are tabulated in table 7.6.

Using the approach previously used, the expected standard deviation of $K_{b2}$ is determined in the same manner as $K_{b1-3}$ by assuming that it has a Gaussian distribution. The fidelity criterion is then evaluated to assess the validity of the model quantitatively. Applying the criterion gives

$$\sum\lambda^2 = \left(\frac{\tau' m_1}{\sigma m_1}\right)^2 + \left(\frac{\tau' m_2}{\sigma m_2}\right)^2 + \left(\frac{\tau' m_3}{\sigma m_3}\right)^2 + \left(\frac{\tau' MLOAD}{\sigma MLOAD}\right)^2 + \left(\frac{\tau' R'_1}{\sigma R_1}\right)^2 +$$

$$\left(\frac{\tau' A_2}{\sigma A_2}\right)^2 + \left(\frac{\tau' A_3}{\sigma A_3}\right)^2 + \left(\frac{\tau' K_{b1-3}}{\sigma K_{b1-3}}\right)^2 + \left(\frac{\tau' K_{b2}}{\sigma K_{b2}}\right)^2$$

$$= 0.4431$$

A definite improvement in $\sum\lambda^2$ is clearly seen but the model still fails to satisfy the fidelity criterion.

| | | |
|---|---|---|
| $m_1$ | kg | 2.050781 |
| $m_2$ | kg | 0.401821 |
| $m_3$ | kg | 0.229105 |
| MLOAD | kg | 1.017110 |
| $R'_1$ | m | 0.049466 |
| $A_2$ | m | 0.235155 |
| $A_3$ | m | 0.129856 |
| $K_{b1-3}$ | V/rad/s | 0.037760 |
| $K_{b2}$ | V/rad/s | 0.074697 |
| MSQE | $Radian^2$ | $9.9884.10^{-4}$ |

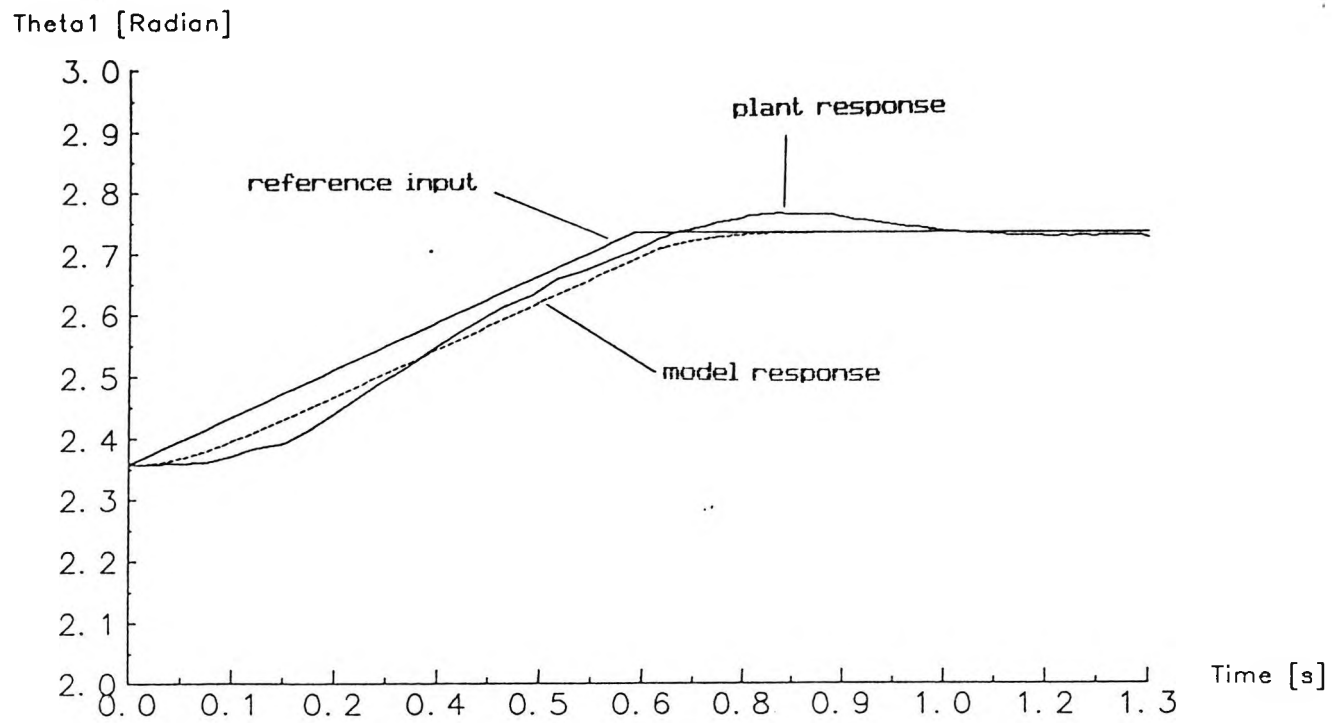
Table 7.5

Optimised parameters of the basic model with $K_{b2}$ optimised.

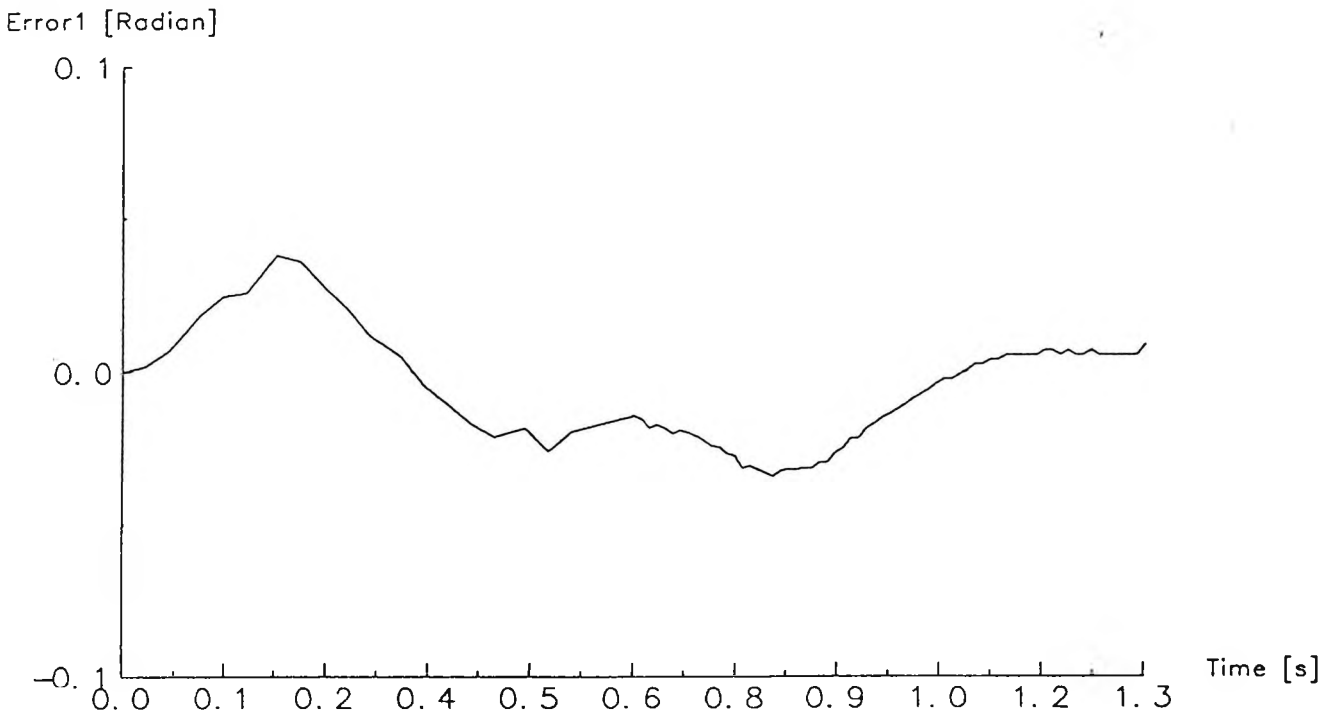
| | | |
|---|---|---|
| $m_1$ | kg | ››› |
| $m_2$ | kg | 0.2734 |
| $m_3$ | kg | 0.1207 |
| MLOAD | kg | 0.4238 |
| $R'_1$ | m | ››› |
| $A_2$ | m | 0.0294 |
| $A_3$ | m | 0.0133 |
| $K_{b1-3}$ | V/rad/s | 0.0255 |
| $K_{b2}$ | V/rad/s | 0.0692 |

Table 7.6

Standard deviation of parameters of the basic model with $K_{b2}$ optimised.



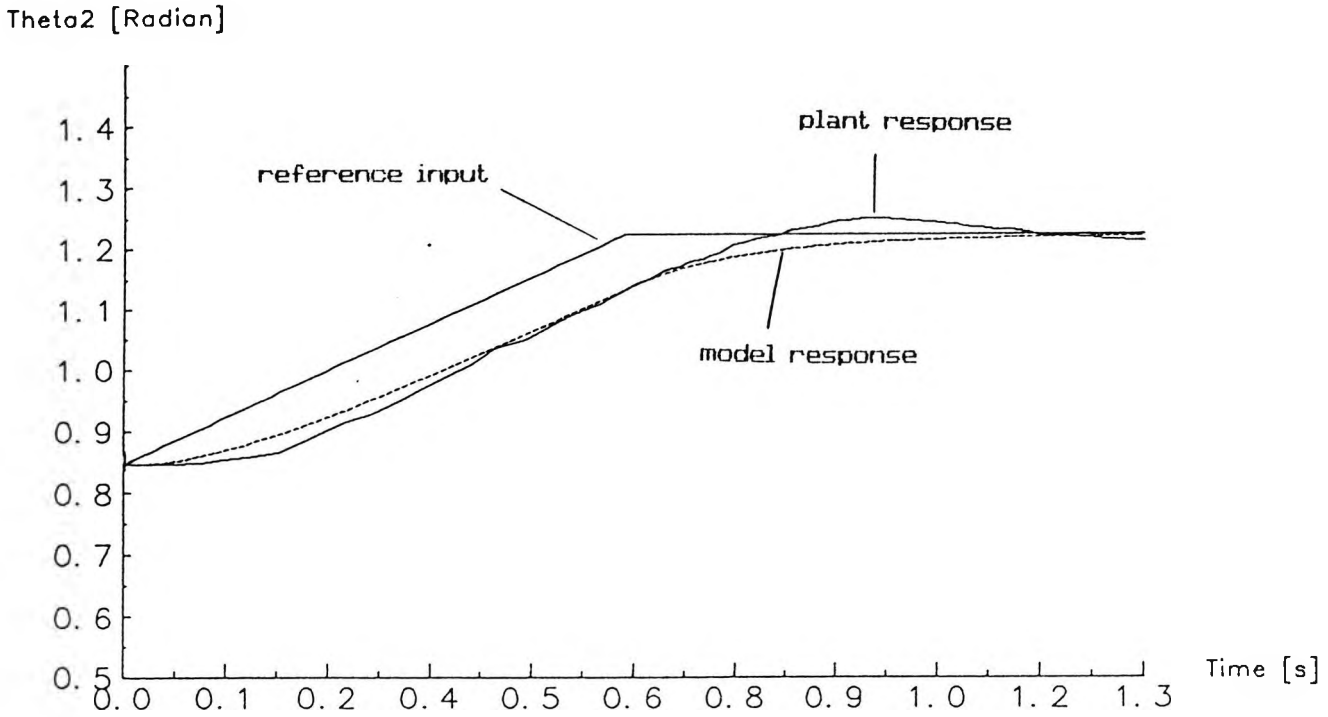

(a) Joint 1 response.



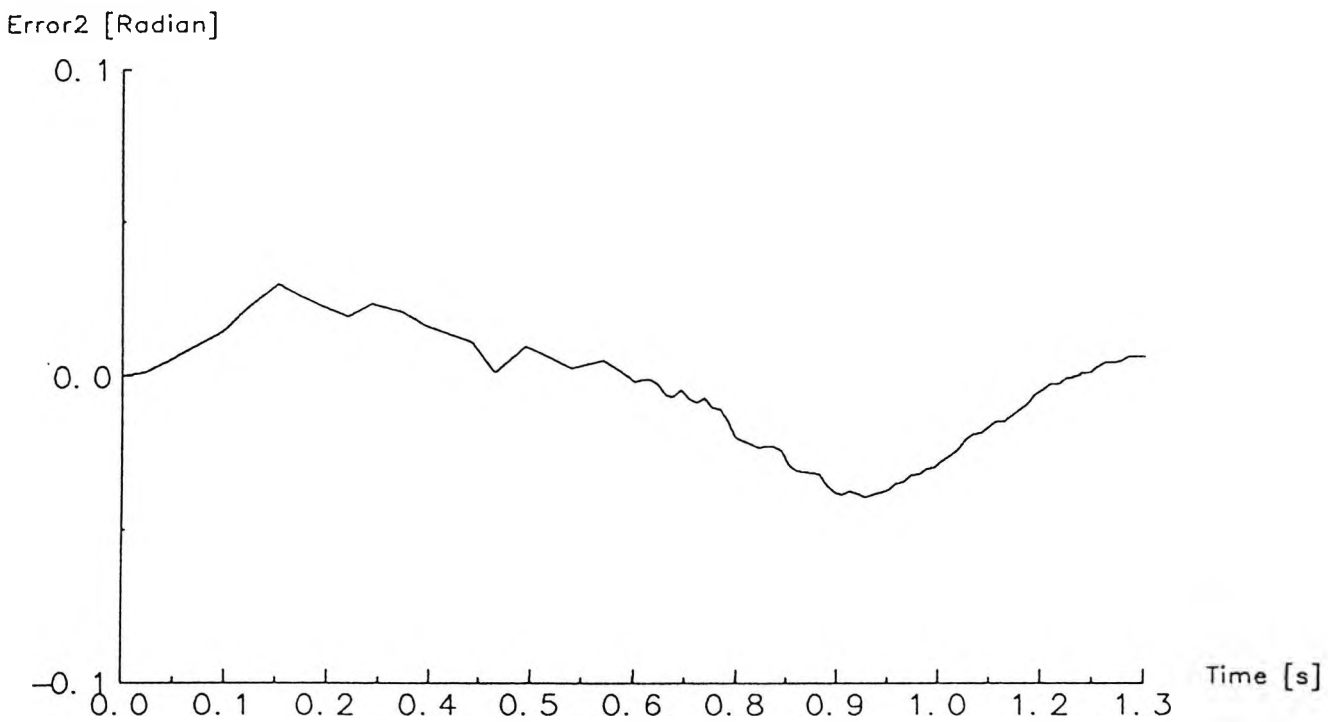

(b) Joint 1 error.

Figure 7.15

Basic model with $K_{b2}$ optimised.



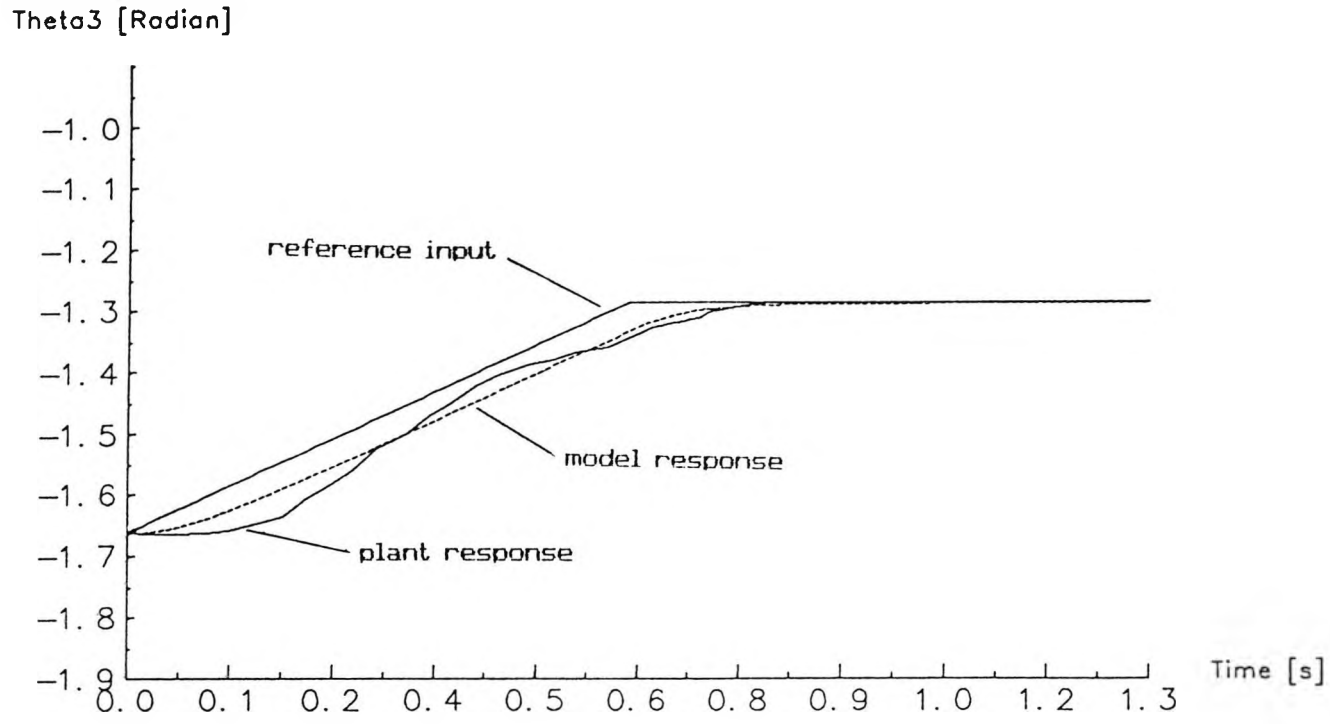

(a) Joint 2 response.



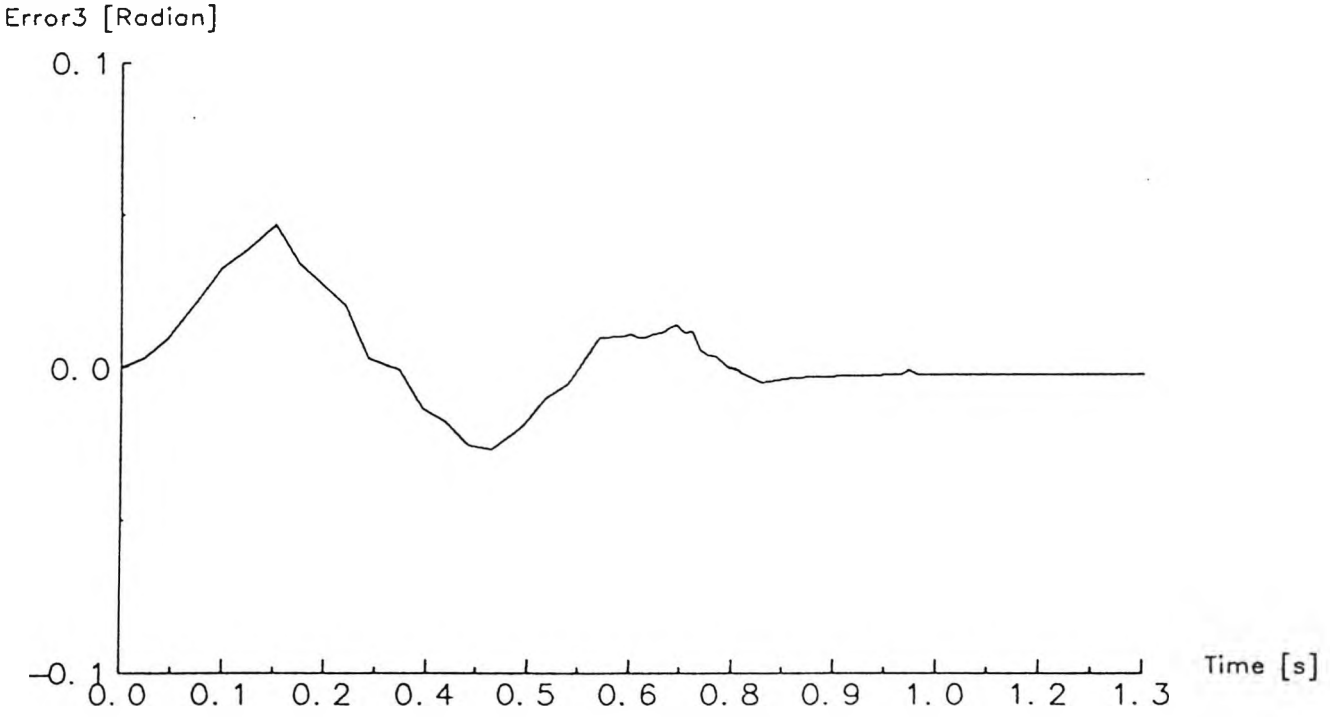

(b) Joint 2 error.

Figure 7.16

Basic model with $K_{b2}$ optimised




(a) Joint 3 response.




(b) Joint 3 error.

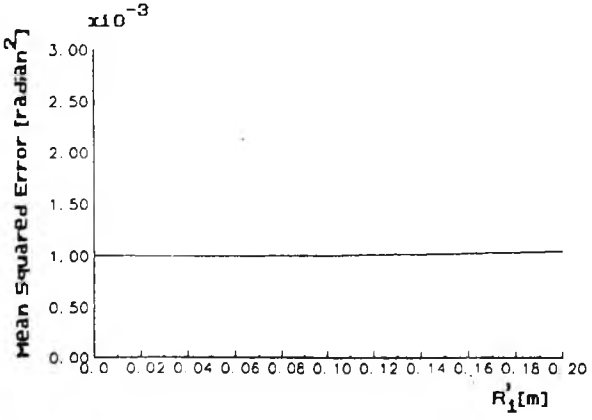
Figure 7.17

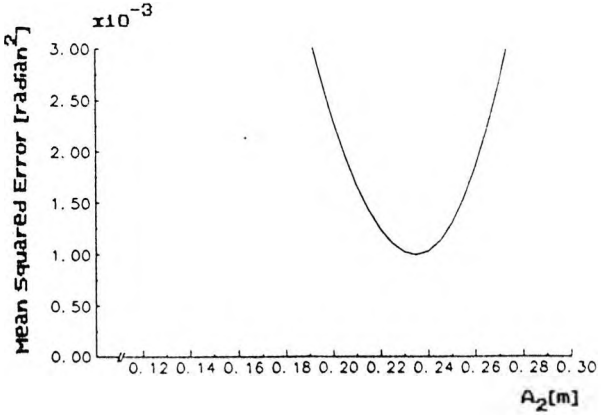
Basic model with $K_{b2}$ optimised.

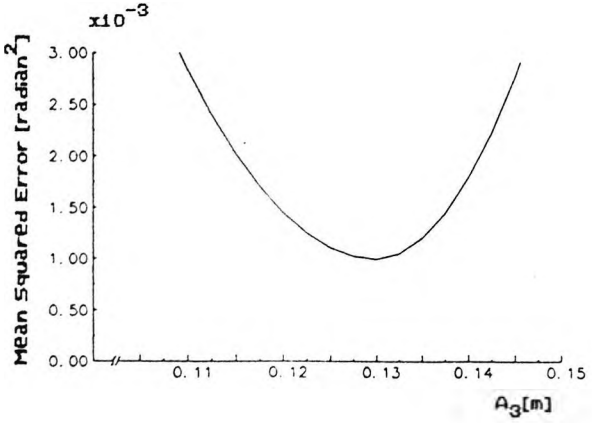
Figure 718

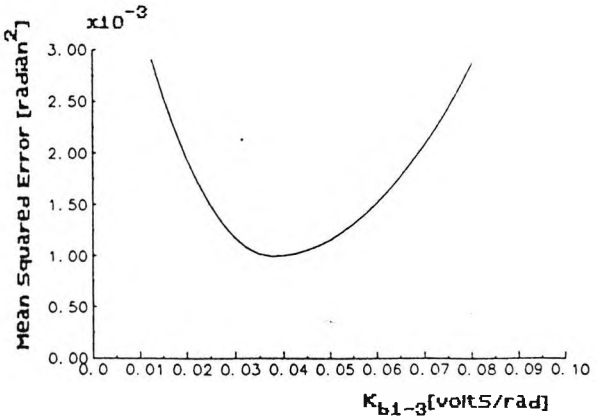
Parameter optimisation curves

of a basic model with $K_{b2}$ optimised.



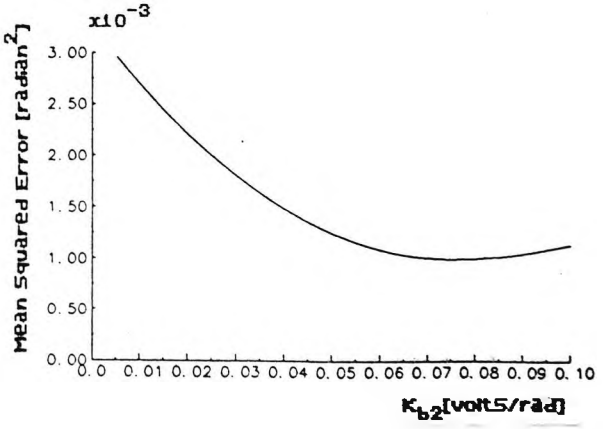




Figure 7.18

(Continued)

## 7.3 Model With Viscous Friction

In order to be more realistic, the model must include some sort of energy dissipation. Viscous friction is not dependent upon position, instead it is dependent upon velocity. Moon, Chung and Cho [1986] and Armstrong [1988] performed experiments to estimate the value of viscous friction using inverse dynamics identification but its sensitivity was not investigated. To include viscous friction on each joint, equation (7.1) is then modified as

$$\tau_1 = D(\theta)\,\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) + \upsilon(\dot{\theta}) \qquad (7.29)$$

where : $\upsilon(\dot{\theta})$ = an n x 1 viscous friction vector

Figure (7.19) is a closed loop block diagram of a system where viscous friction is included.

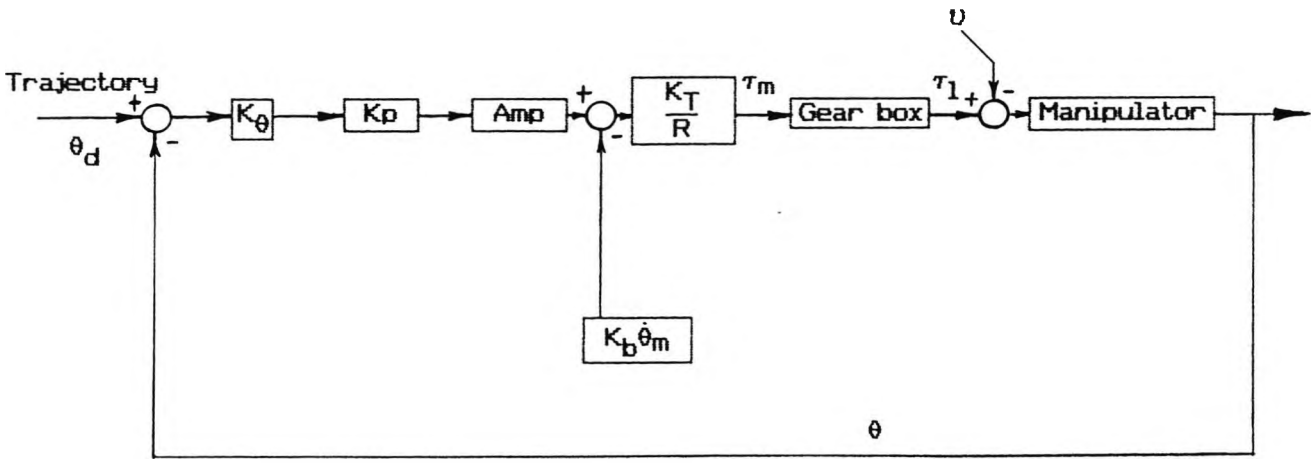



Figure 7.19

Closed loop system block diagram with viscous damping.

For joint i, expressing the viscous friction with respect to the actuator side gives

$$\upsilon_{mi} = \tilde{\upsilon}_{mi}\,\dot{\theta}_{mi} + n_i\,\tilde{\upsilon}_{ji}\,\dot{\theta}_i$$

$$= \left( \tilde{\upsilon}_{mi} + n^2\,\tilde{\upsilon}_{ji} \right) \dot{\theta}_{mi}$$

$$= \tilde{\upsilon}_{effi}\,\dot{\theta}_{mi} \qquad (7.30)$$

where : $\tilde{\upsilon}$ = viscous damping constant [Nms/rad]

$n$ = gear ratio ($\leq 1$)

and subscripts : m = actuator

j = joint

i = no. of joint

eff = effective value

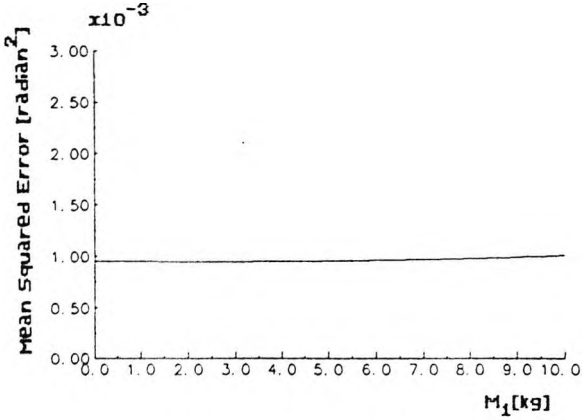
Since the viscous damping constant is not a fundamental parameter, its existence will only affect its own joint response. With the viscous damping constant introduced as a new parameter, there are 12 parameters to be identified ie. with an addition of 3 viscous damping constant effective values. Since the value of effective viscous damping constant is always positive, its lower bound is assigned to zero and its upper bound is found by trial and error from simulation exercises. For all joints, zero values are used as starting points in optimising the parameters. The results obtained by optimising this model are given in table 7.7. This optimisation yields an optimised back EMF coefficient value of motor 2 closed to the prescribed value in the data sheet due to the introduction of viscous

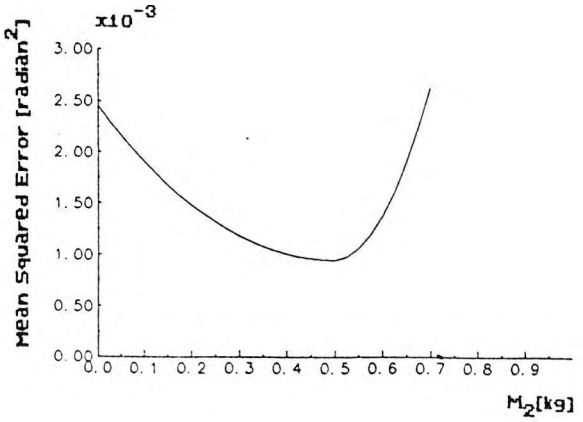
friction. Figure (7.20) shows the corresponding parameter optimisation curves. Although from table 7.7 the mean squared error value does not change much, figure (7.20) shows that the parameters are more sensitive. Using this set of optimised parameters, the responses and errors of each joint are shown in figures (7.21), (7.22) and (7.23). From figure (7.20), the standard deviations of each corresponding parameter in order to eliminate the errors between the recorded response of the real system and the model output can be evaluated and are tabulated in table 7.8.

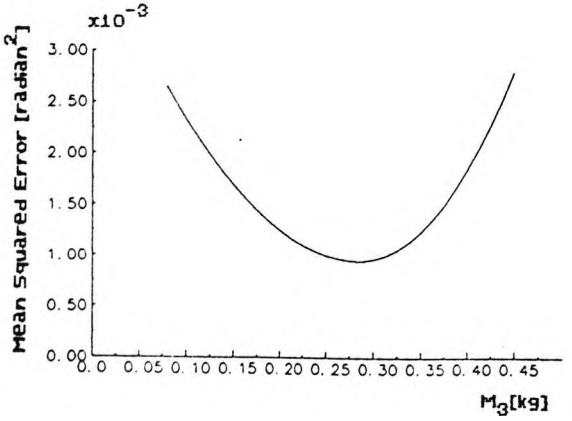
| | | |
|---|---|---|
| $m_1$ | kg | 2.475884 |
| $m_2$ | kg | 0.497314 |
| $m_3$ | kg | 0.286792 |
| MLOAD | kg | 1.219998 |
| $R'_1$ | m | 0.060944 |
| $A_2$ | m | 0.280288 |
| $A_3$ | m | 0.167525 |
| $K_{b1-3}$ | V/rad/s | 0.034953 |
| $K_{b2}$ | V/rad/s | 0.037130 |
| $\tilde{\upsilon}_{eff1}$ | Nms/rad | $3.0.10^{-6}$ |
| $\tilde{\upsilon}_{eff2}$ | Nms/rad | $2.46.10^{-4}$ |
| $\tilde{\upsilon}_{eff3}$ | Nms/rad | $1.7.10^{-5}$ |
| MSQE | $Radian^2$ | $9.4940.10^{-4}$ |

Table 7.7

Optimised parameters of the model with viscous friction.



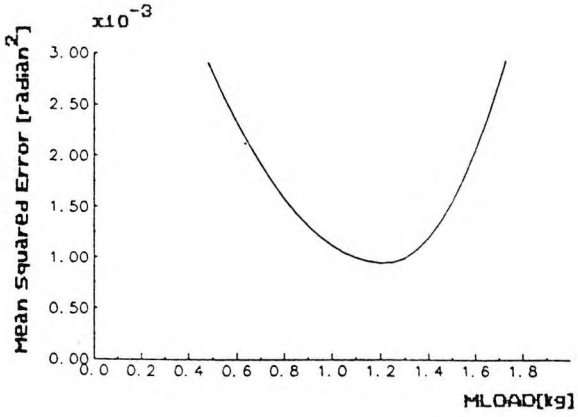
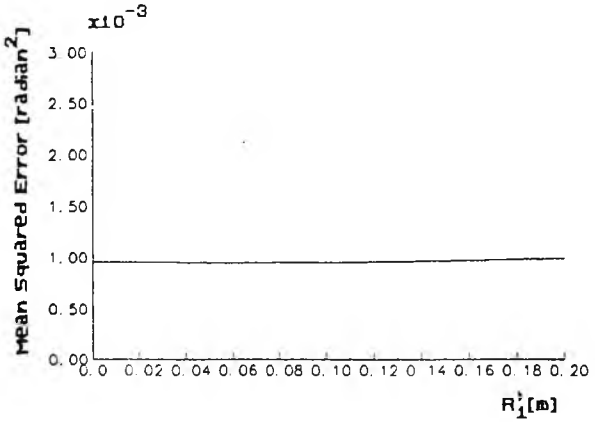
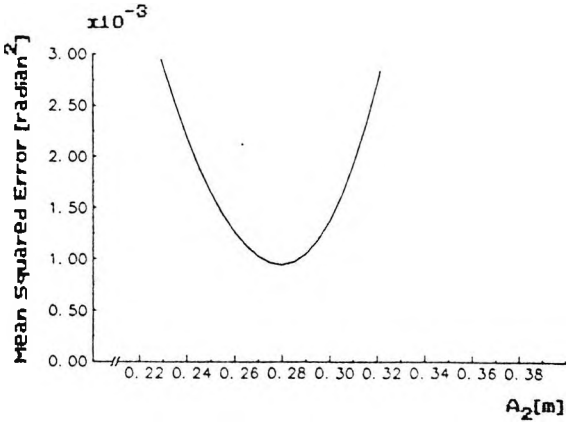



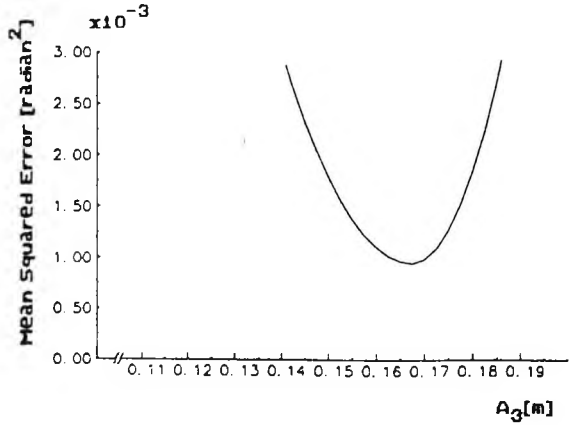

Figure 7.20

Parameter optimisation curves

of a model with viscous friction.



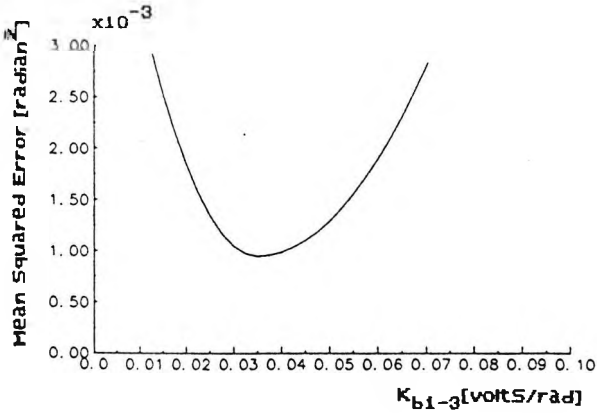
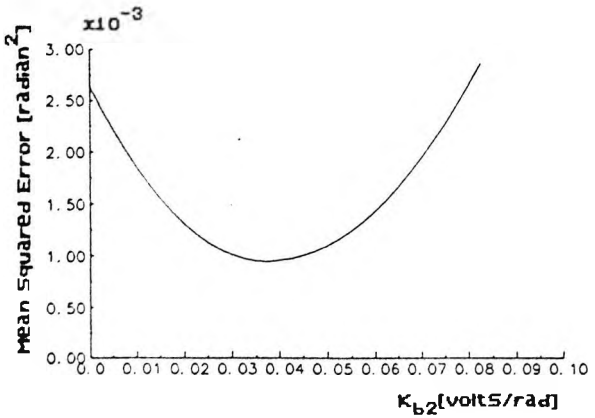
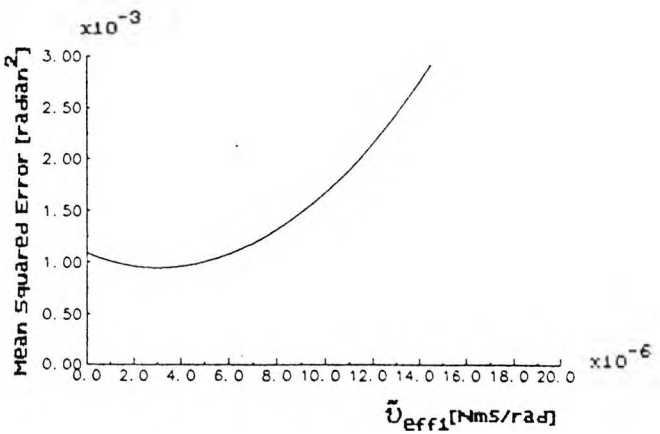
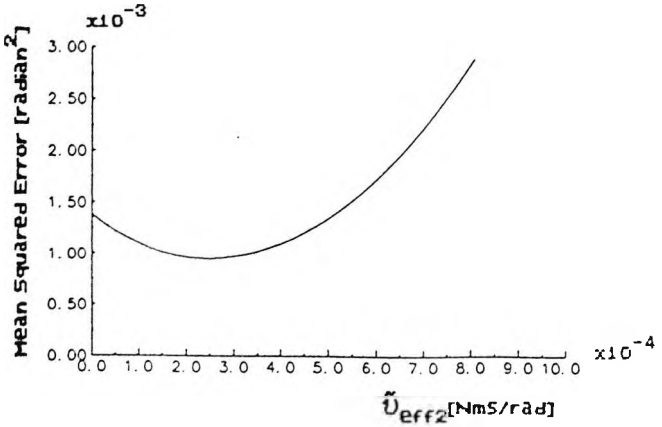




Figure 7.20

(Continued)



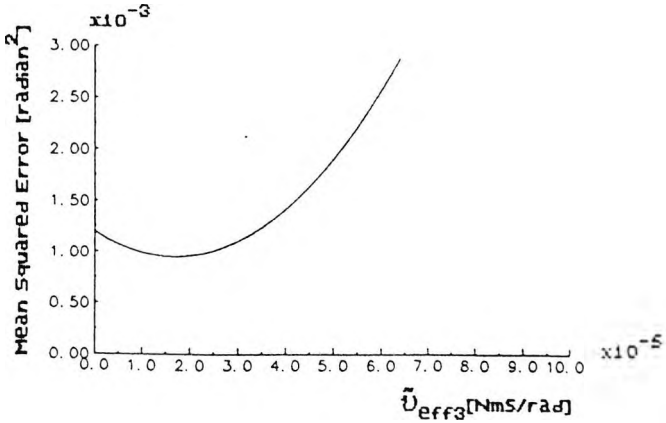




Figure 7.20

(Continued)

Since all surfaces in ambient atmosphere are contaminated in a number of forms [Sarkar, 1980], the roughness between two surfaces may change. For example, metals may oxidize and this will result an obstruction in the movement. Hence, one expects an uncertainty in the frictional resistance. An expected standard deviation of viscous damping constant is obtained by assuming its distribution has a Gaussian distribution and its optimal value is the Gaussian mean value. As in the back EMF constant, an area under the Gaussian tail at $\tilde{\upsilon}_{eff} = 0$ is chosen to be three standard deviations from its mean value.

| | | |
|---|---|---|
| $m_1$ | kg | ››› |
| $m_2$ | kg | 0.2930 |
| $m_3$ | kg | 0.1380 |
| MLOAD | kg | 0.4547 |
| $R'_1$ | m | ››› |
| $A_2$ | m | 0.0313 |
| $A_3$ | m | 0.0158 |
| $K_{b1-3}$ | V/rad/s | 0.0218 |
| $K_{b2}$ | V/rad/s | 0.0311 |
| $\tilde{\upsilon}_{eff1}$ | Nms/rad | $8.0.10^{-6}$ |
| $\tilde{\upsilon}_{eff2}$ | Nms/rad | $3.95.10^{-4}$ |
| $\tilde{\upsilon}_{eff3}$ | Nms/rad | $3.3.10^{-5}$ |

Table 7.8

Standard deviation of parameters of the model with viscous friction.

This model is then assessed in the same way as before in order to check its

validity quantitatively. Applying the criterion gives

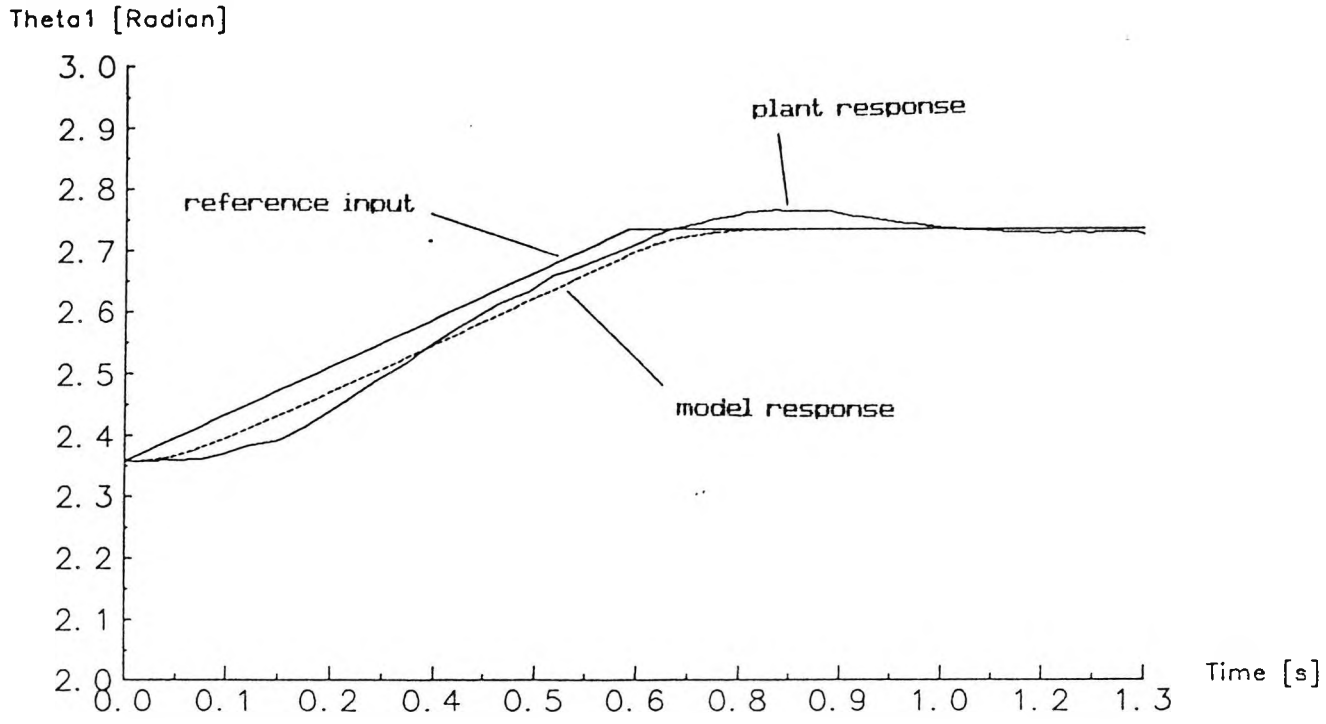
$$\sum\lambda^2 = \left(\frac{\tau' m_1}{\sigma m_1}\right)^2 + \left(\frac{\tau' m_2}{\sigma m_2}\right)^2 + \left(\frac{\tau' m_3}{\sigma m_3}\right)^2 + \left(\frac{\tau' MLOAD}{\sigma MLOAD}\right)^2 + \left(\frac{\tau' R'_1}{\sigma R_1}\right)^2 +$$

$$\left(\frac{\tau' A_2}{\sigma A_2}\right)^2 + \left(\frac{\tau' A_3}{\sigma A_3}\right)^2 + \left(\frac{\tau' K_{b1\text{-}3}}{\sigma K_{b1\text{-}3}}\right)^2 + \left(\frac{\tau' K_{b2}}{\sigma K_{b2}}\right)^2 + \left(\frac{\tau' \tilde{\upsilon}_{eff1}}{\sigma \tilde{\upsilon}_{eff1}}\right)^2 +$$

$$\left(\frac{\tau' \tilde{\upsilon}_{eff2}}{\sigma \tilde{\upsilon}_{eff2}}\right)^2 + \left(\frac{\tau' \tilde{\upsilon}_{eff3}}{\sigma \tilde{\upsilon}_{eff3}}\right)^2$$

$$= 0.5973$$

Although the fidelity criterion $\sum\lambda^2$ is greater than the previous model, it is still not fully capable of explaining the transient record.

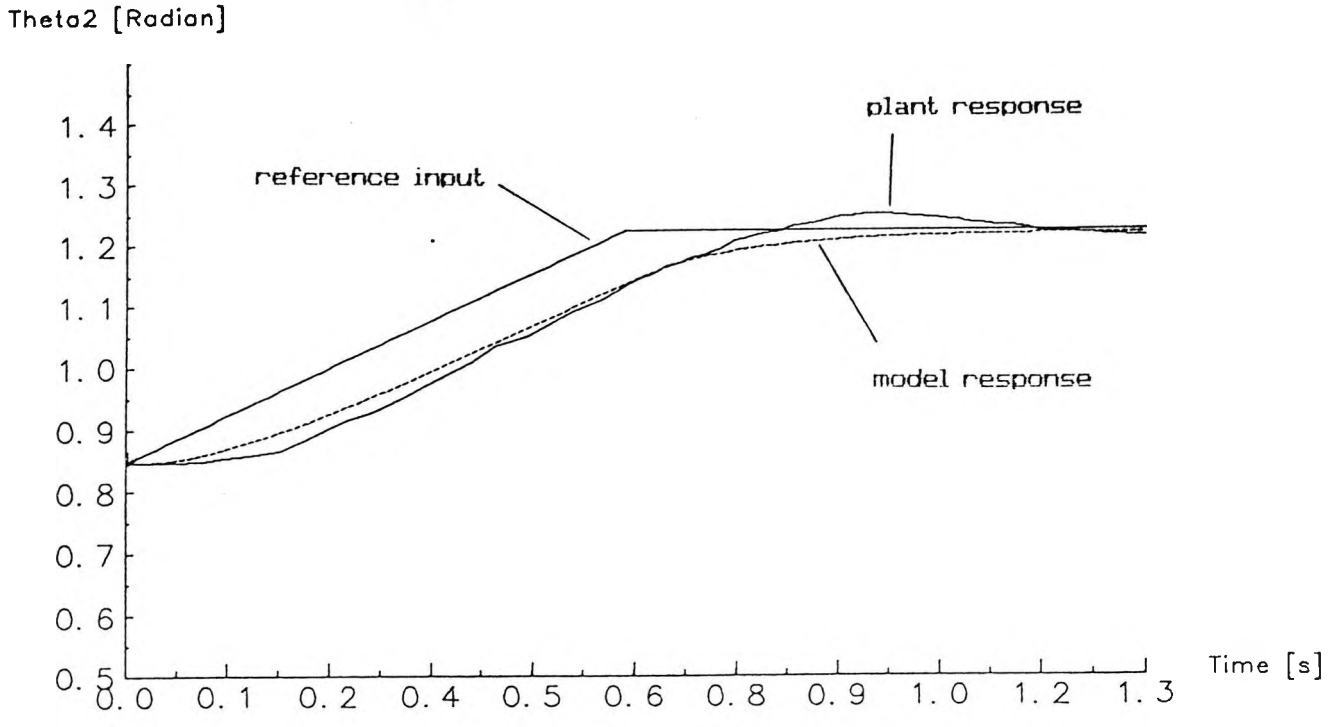



(a) Joint 1 response.

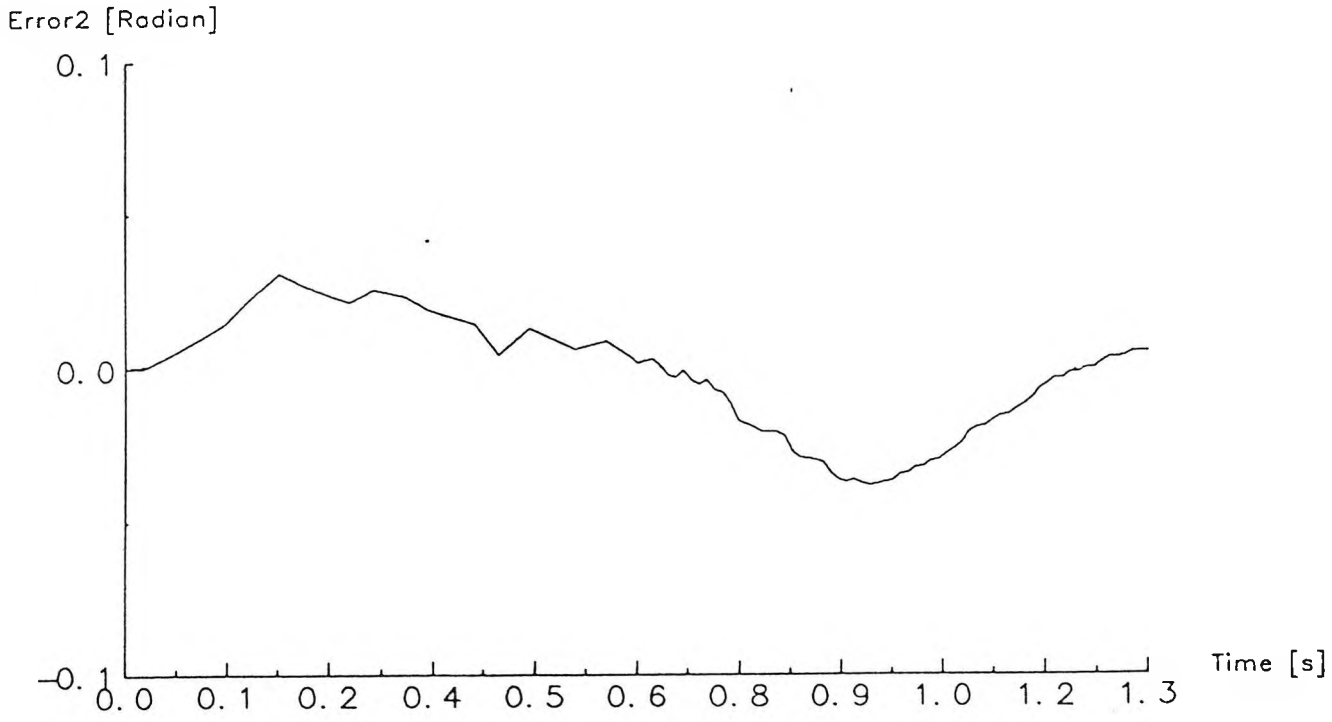

(b) Joint 1 error.

Figure 7.21

Model with viscous friction.



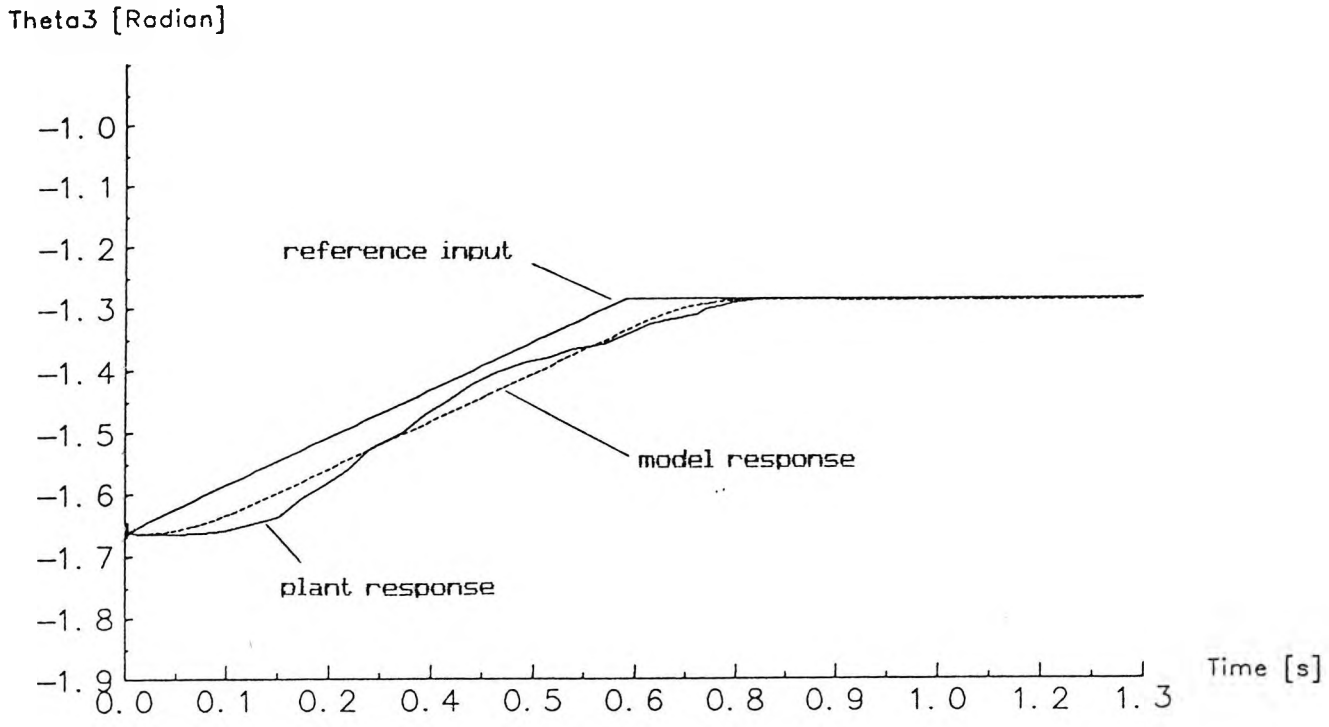

(a) Joint 2 response.



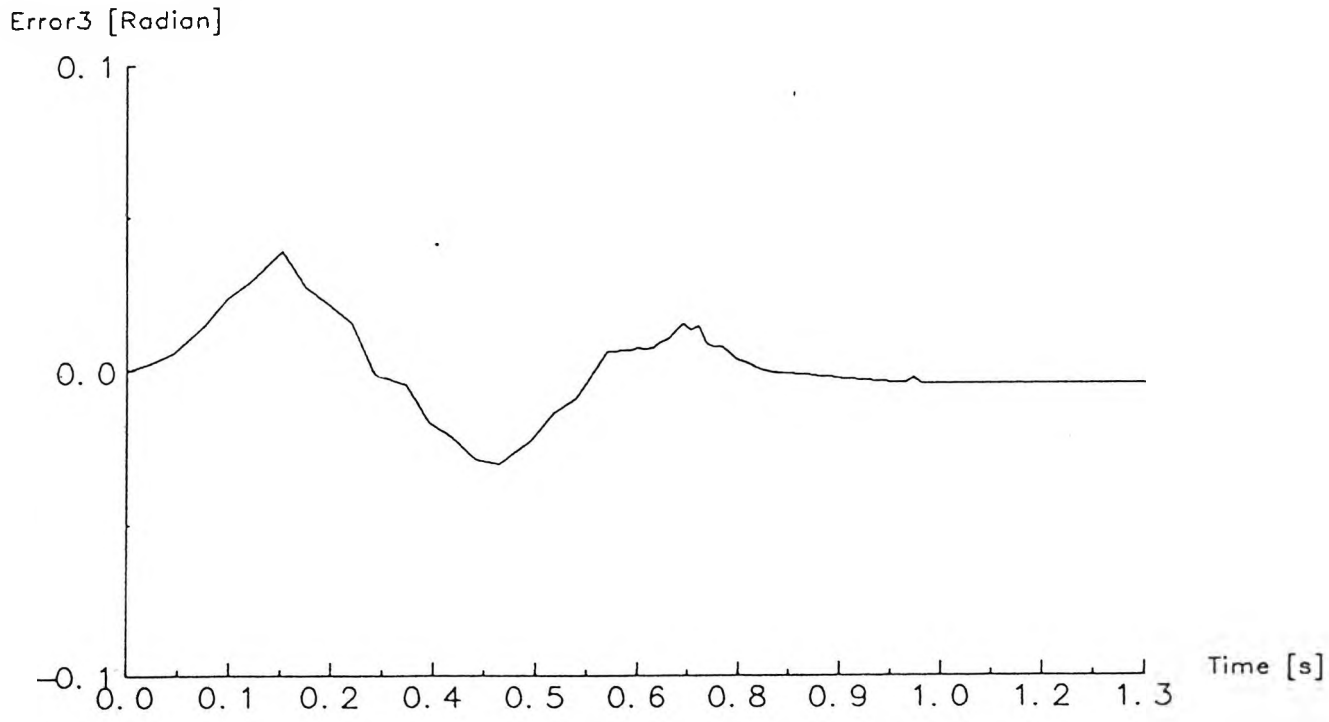

(b) Joint 2 error.

Figure 7.22

Model with viscous friction




(a) Joint 3 response.




(b) Joint 3 error.

Figure 7.23

Model with viscous friction.

### 7.4 Model With Viscous And Coulomb Friction

Another term which is also to be considered is Coulomb friction since it is a redundant energy. There are not many papers which discuss this topic and some of them do not provide simulation results [Kubo, Anwar, Tomizuka, 1986; Gogoussis, Donath, 1987,1988; Armstrong, 1988]. Prior to modelling the Coulomb friction, there are some laws of friction which must be understood.

**Amontons' laws of friction [Sarkar, 1980] :**

- The friction force is proportional to the normal force of the moving object.
- The friction force does not depend on the apparent contact area.

The proportionality between the normal force and the friction force determines the characteristic of the static friction coefficient $\mu_s$ and the kinetic friction coefficient $\mu_k$ where $\mu_k < \mu_s$. These two coefficients are not the intrinsic property of a material, instead they depend on the roughness of the contact area surfaces.

**Coulomb's law of friction [Sarkar, 1980] :**

- The friction force between two surfaces does not depend on the relative velocity between the two surfaces.

This law is valid only over a limited range of speed [Sarkar, 1980]. If the speed increases up to a certain limit, the friction coefficent will fall considerably.

The three laws stated above are used in studying the behaviour of Coulomb friction in the rest of this section. When the two contact surfaces are about to move relatively to each other, the friction rises to a certain value. This friction is regarded as static friction ie. when the two surfaces still stick together. When it achieves a maximum value, the static friction then disappears since the two surfaces are now moving relatively to each other. In this moment, the interface slips so that the friction falls to a lower value ie. the kinetic friction. The nature of the motion is in such a manner that it sticks, slips and is repeated for the whole process. Hence, it does not produce a constant smooth friction, instead it fluctuates about a certain mean value, and for contact surfaces which have similar materials, the friction fluctuates in a random manner [Sarkar, 1980]. Figure (7.24) shows the variation of friction coefficient with time.

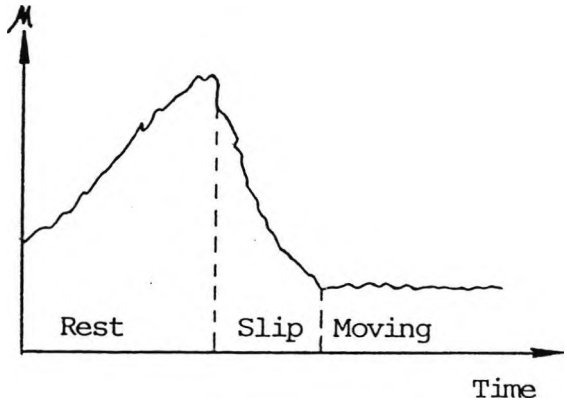



Figure 7.24

Variation of friction coefficient $\mu$.

This behaviour was modelled by Tustin [1947] and assumed as having an exponential decay from its static value to its kinetic value. The equation is given by

$$F(v) = F_0 - F_C\left(1 - e^{(-v/V_C)}\right) \qquad (7.31)$$

where: $F(v)$ = the friction as a function of velocity.

$F_0$ = the maximum static friction.

$F_C$ = the difference between the static and kinetic friction.

$v$ = velocity of motion.

$V_C$ = a constant which gives the characteristic velocity at the system transition to kinetic friction.

Due to the presence of joint compliances, Coulomb friction can generate limit cycles at low speed [Gogoussis and Donath, 1987, 1988]. In modelling the Coulomb friction, some simplifications are made. Static and kinetic friction coefficients are assumed equal as shown in figure (7.25), joint compliances are assumed negligible, the speed is high enough to avoid oscillation (jerkiness), and the dead band in the system due to the stick slip property is assumed negligible since the trajectory does not have zero velocity except in the initial and final points. All joints have equal coefficients of friction since they are made of the same material and underwent the same manufacturing processes.

Although it is true that Coulomb friction depends on the total forces at a joint under consideration, but since the total of the inertial forces exerted by the other links is zero when the corresponding joint accelerations are zero and since the coriolis/ centrifugal forces are insignificant in low speed, only gravity forces are important, which affect the magnitude of Coulomb friction. This is particularly true for a trajectory which is used in this research work as can be seen from figures (7.13) and (7.14). Moreover, not many applications use trajectories which have many acceleration points along them.

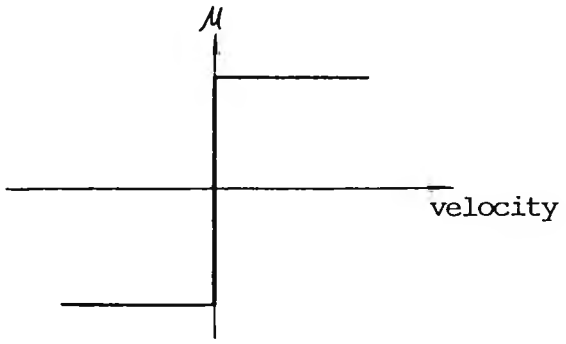



Figure 7.25

Simplified Coulomb friction coefficient.

In evaluating Coulomb friction, the weight of each link at its centre of mass can be projected to obtain a component which acts as a normal force to a joint. So, each link suffers a normal force caused by the upper links as shown in figure (7.26). As the manipulator moves along a trajectory, the normal forces of all joints change accordingly. Hence, Coulomb friction causes highly nonlinear coupling in the system.

Based on figure 7.26 as shown below, Coulomb friction in a manipulator system can be approximated as

$$C'_1 = \tilde{\mu}\left(m_2 g\sin(\theta_2)\cos(\theta_2) + MM_3 g\sin(\theta_2+\theta_3)\cos(\theta_2+\theta_3)\right) \tag{7.32}$$

$$C'_2 = \tilde{\mu}\left(m_2 g\sin(\theta_2) + MM_3 g\sin(\theta_2+\theta_3)\cos(\theta_3)\right) \tag{7.33}$$

$$C'_3 = \tilde{\mu} MM_3 g\sin(\theta_2+\theta_3) \tag{7.34}$$

where : $C'_1$ = Coulomb friction torque at joint 1.

$C'_2$ = Coulomb friction torque at joint 2.

$C'_3$ = Coulomb friction torque at joint 3.

$\tilde{\mu}$ = effective coefficient of friction which gives the proportionality of Coulomb friction torque on the normal force at each joint.

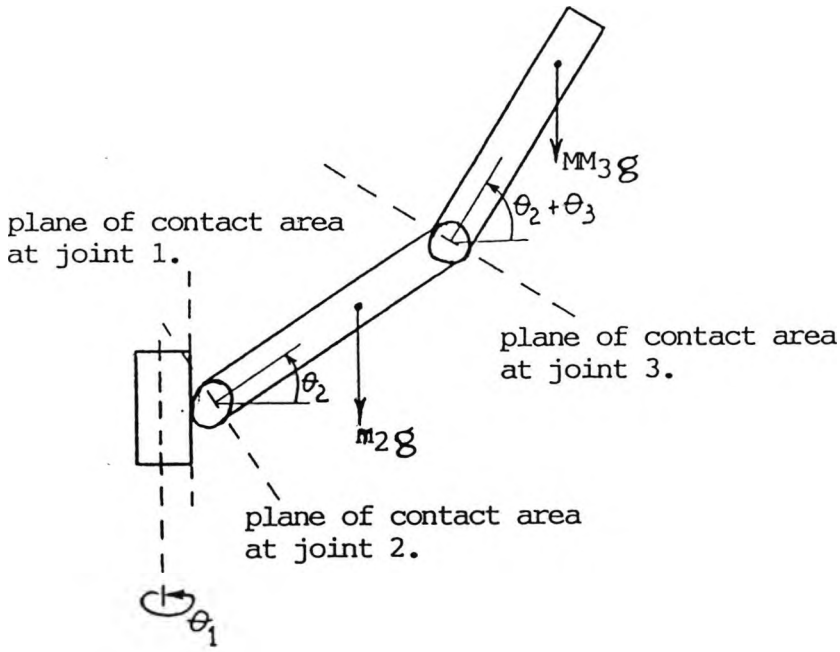



Figure 7.26

Contact areas in a manipulator system.

Taking Coulomb friction into account, the dynamic equation of the manipulator now becomes

$$\tau_1 = D(\theta)\,\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) + U(\dot{\theta}) + C'(\theta) \qquad (7.35)$$

where : $C'(\theta)$ = an n x 1 approximate Coulomb friction vector

Combining viscous and Coulomb frictions yields

$$\tau_1 = D(\theta)\,\ddot{\theta} + H(\theta,\dot{\theta}) + G(\theta) + F(\theta,\dot{\theta}) \qquad (7.36)$$

where : $F(\dot{\theta}) = \upsilon(\dot{\theta}) + C'(\theta)$

$= \text{an } n \times 1 \text{ friction vector}$

Figure (7.27) shows a closed loop block diagram of a system where complete friction is included.

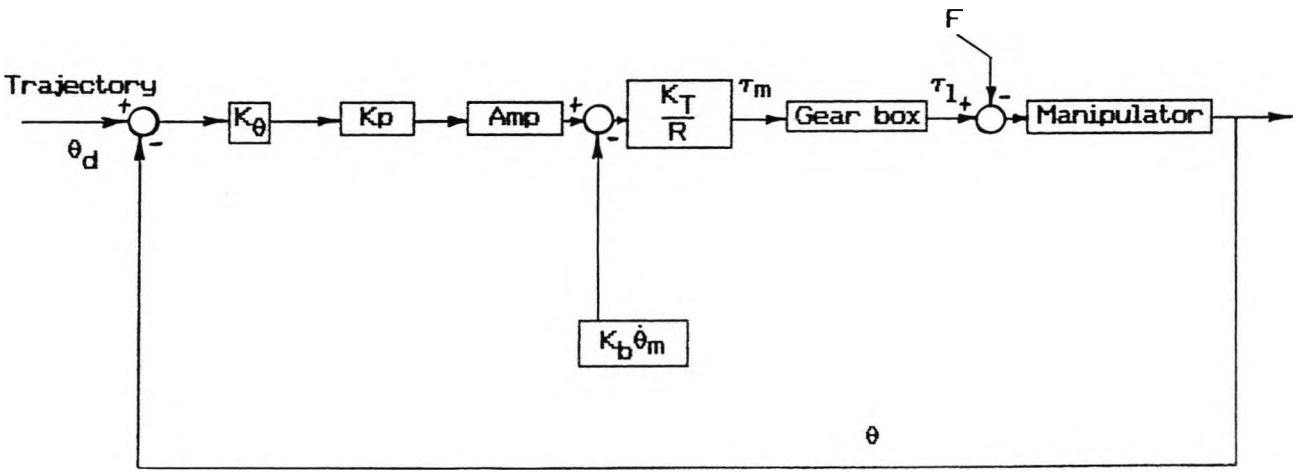



Figure 7.27

Closed loop system block diagram with complete friction.

The lower bound of Coulomb friction coefficient is assigned to zero while its upper bound is found by trial and error from simulation exercises and zero value is used as its starting point in the optimisation process. The results of optimising

the model with complete friction are presented in table 7.9.

| | | |
|---|---|---|
| $m_1$ | kg | 2.517390 |
| $m_2$ | kg | 0.506972 |
| $m_3$ | kg | 0.290461 |
| MLOAD | kg | 1.250307 |
| $R'_1$ | m | 0.062899 |
| $A_2$ | m | 0.285755 |
| $A_3$ | m | 0.168410 |
| $K_{b1-3}$ | V/rad/s | 0.033444 |
| $K_{b2}$ | V/rad/s | 0.036948 |
| $\tilde{\upsilon}_{eff1}$ | Nms/rad | $2.0.10^{-6}$ |
| $\tilde{\upsilon}_{eff2}$ | Nms/rad | $2.53.10^{-4}$ |
| $\tilde{\upsilon}_{eff3}$ | Nms/rad | $2.4.10^{-5}$ |
| $\tilde{\mu}$ | | 0.128486 |
| MSQE | $Radian^2$ | $7.4877.10^{-4}$ |

Table 7.9

Optimised parameters of the model with complete friction.

From table 7.9, it can be seen that introducing Coulomb friction can reduce the minimum mean squared error over 20%. Figure (7.27) shows the parameter optimisation curves and the output responses with their errors of this model are shown in figures (7.28), (7.29) and (7.30). The fundamental parameter curves show that the parameter sensitivities rise considerably. This is because normal forces depend on these parameters. The standard deviations of each corresponding

parameter are then evaluated from figure (7.27) and given in table 7.10 below. The expected standard deviation of Coulomb friction coefficient is determined in the same manner as in the viscous damping constant ie. with the assumption it has a Gaussian distribution truncated three standard deviations from its optimal value.

Applying the fidelity criterion gives

$$\sum\lambda^2 = \left(\frac{\tau' m_1}{\sigma m_1}\right)^2 + \left(\frac{\tau' m_2}{\sigma m_2}\right)^2 + \left(\frac{\tau' m_3}{\sigma m_3}\right)^2 + \left(\frac{\tau' MLOAD}{\sigma MLOAD}\right)^2 + \left(\frac{\tau' R'_1}{\sigma R_1}\right)^2 +$$

$$\left(\frac{\tau' A_2}{\sigma A_2}\right)^2 + \left(\frac{\tau' A_3}{\sigma A_3}\right)^2 + \left(\frac{\tau' K_{b1\text{-}3}}{\sigma K_{b1\text{-}3}}\right)^2 + \left(\frac{\tau' K_{b2}}{\sigma K_{b2}}\right)^2 + \left(\frac{\tau' \tilde{\upsilon}_{eff1}}{\sigma \tilde{\upsilon}_{eff1}}\right)^2 +$$

$$\left(\frac{\tau' \tilde{\upsilon}_{eff2}}{\sigma \tilde{\upsilon}_{eff2}}\right)^2 + \left(\frac{\tau' \tilde{\upsilon}_{eff3}}{\sigma \tilde{\upsilon}_{eff3}}\right)^2 + \left(\frac{\tau' \tilde{\mu}}{\sigma \tilde{\mu}}\right)^2$$

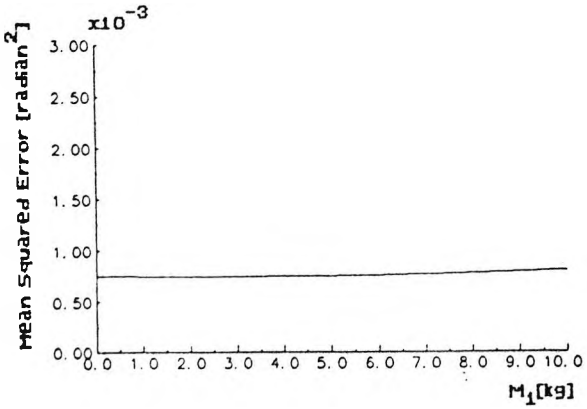
$$= 0.9313$$

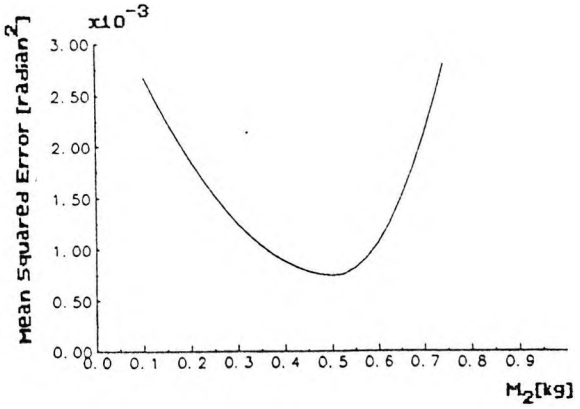
The above result for the fidelity criterion of this model indicates that the model is almost considered capable of explaining the plant transient record. Joint 3 has the largest discrepancy in comparison with the other two links. This is because link 3, with its wrist and gripper, has the most irregular shape and it is difficult to evaluate or even to approximate its moment of inertia. Furthermore, the equation to obtain the moment of inertia of each link is based on the assumption that each link has a regular shape as stated in section 7.2.

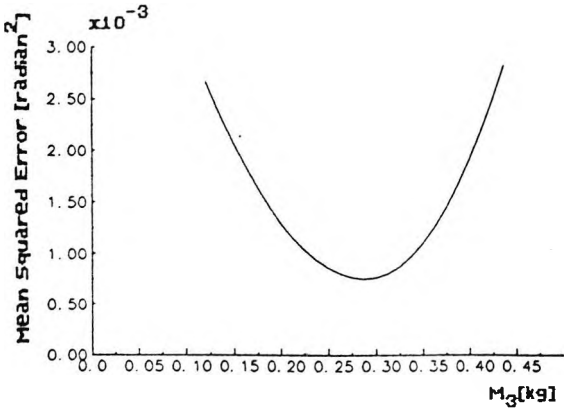
| | | |
|---|---|---|
| $m_1$ | kg | ››› |
| $m_2$ | kg | 0.2113 |
| $m_3$ | kg | 0.0989 |
| MLOAD | kg | 0.3684 |
| $R'_1$ | m | ››› |
| $A_2$ | m | 0.0253 |
| $A_3$ | m | 0.0127 |
| $K_{b1-3}$ | V/rad/s | 0.0193 |
| $K_{b2}$ | V/rad/s | 0.0296 |
| $\tilde{\upsilon}_{eff1}$ | Nms/rad | $4.0.10^{-6}$ |
| $\tilde{\upsilon}_{eff2}$ | Nms/rad | $3.14.10^{-4}$ |
| $\tilde{\upsilon}_{eff3}$ | Nms/rad | $3.8.10^{-5}$ |
| $\tilde{\mu}$ | | 0.1129 |

Table 7.10

Standard deviation of parameters of the model with complete friction.



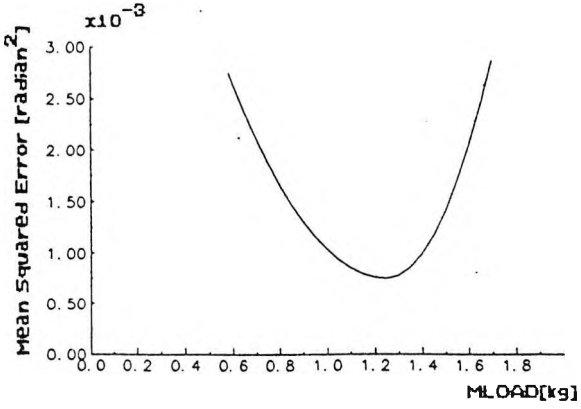
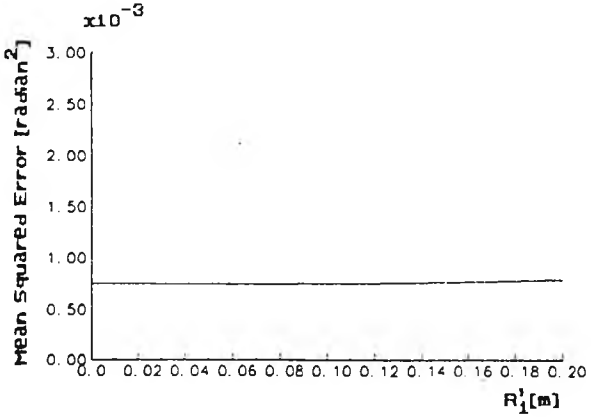
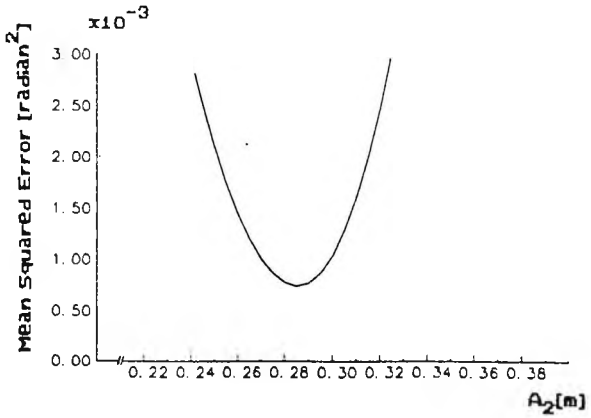



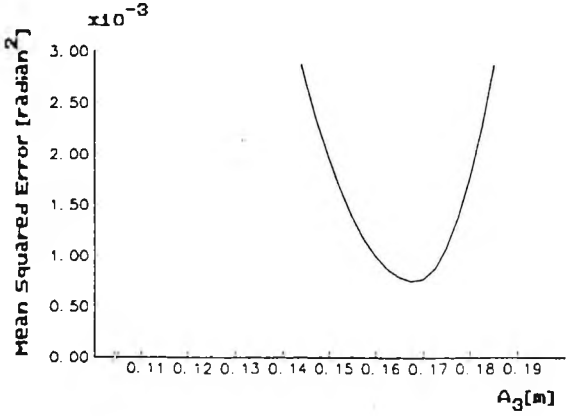

Figure 7.28

Parameter optimisation curves

of a model with viscous and Coulomb friction



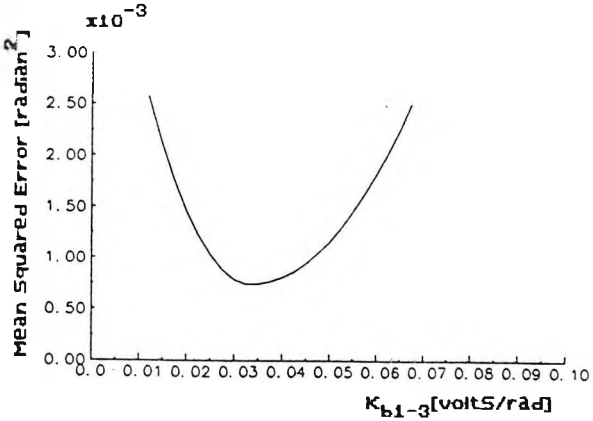
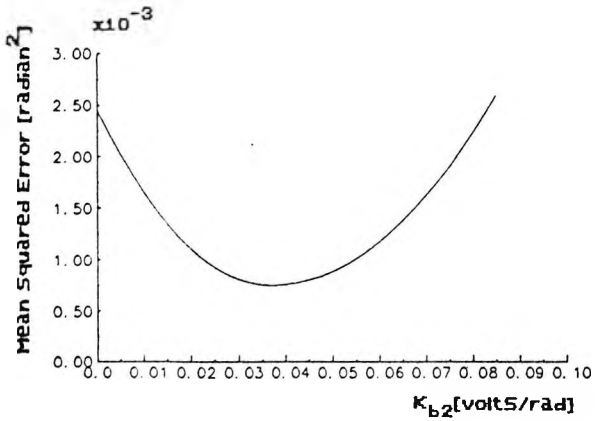
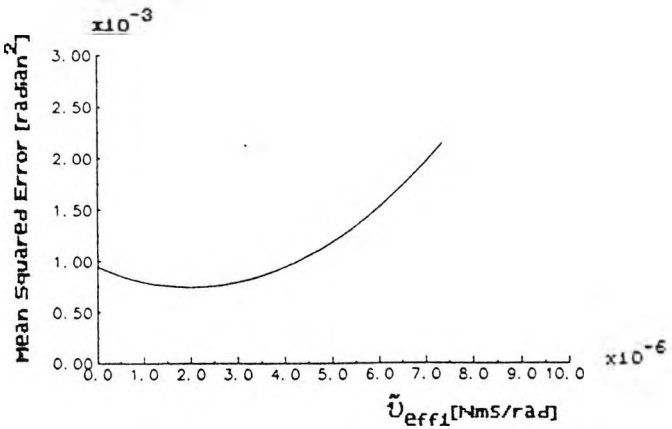
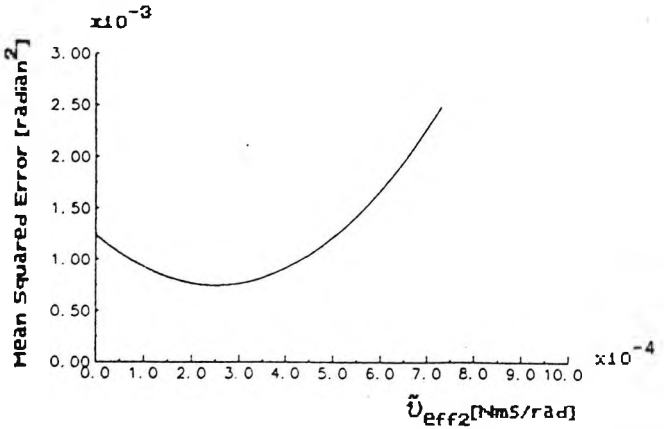



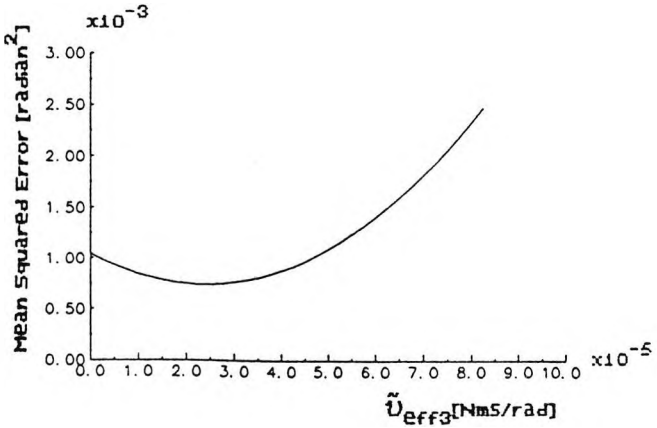

Figure 7.28

(Continued)



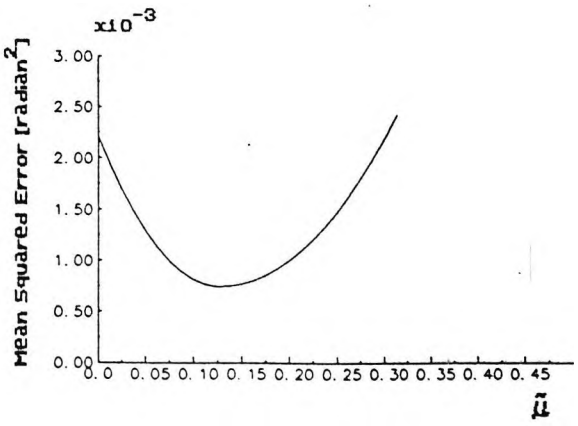
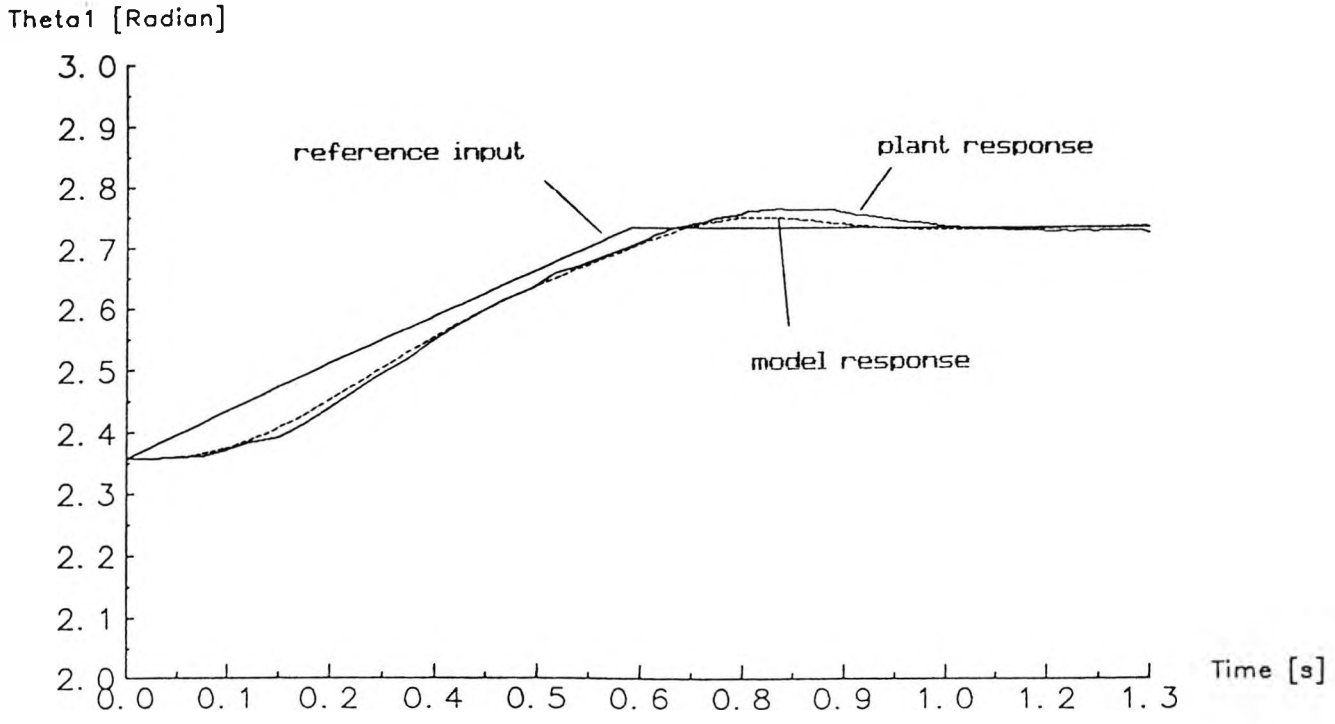
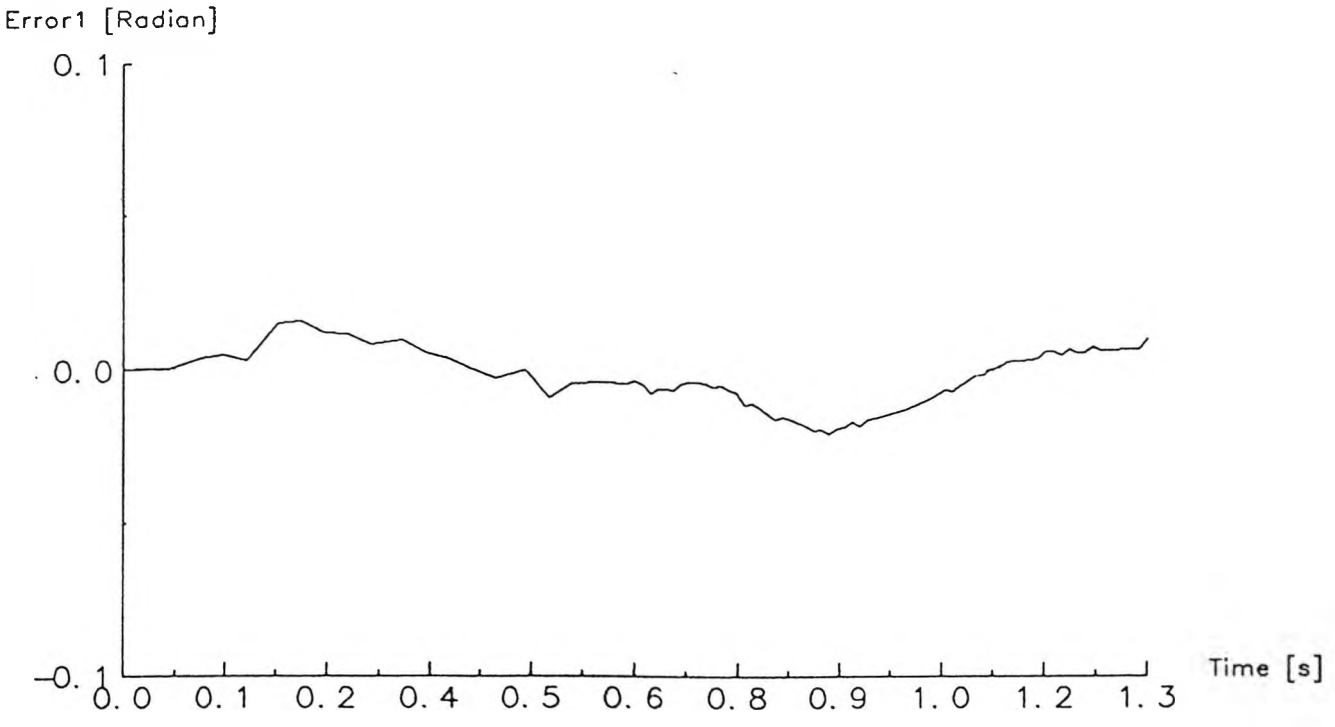




Figure 7.28

(Continued)



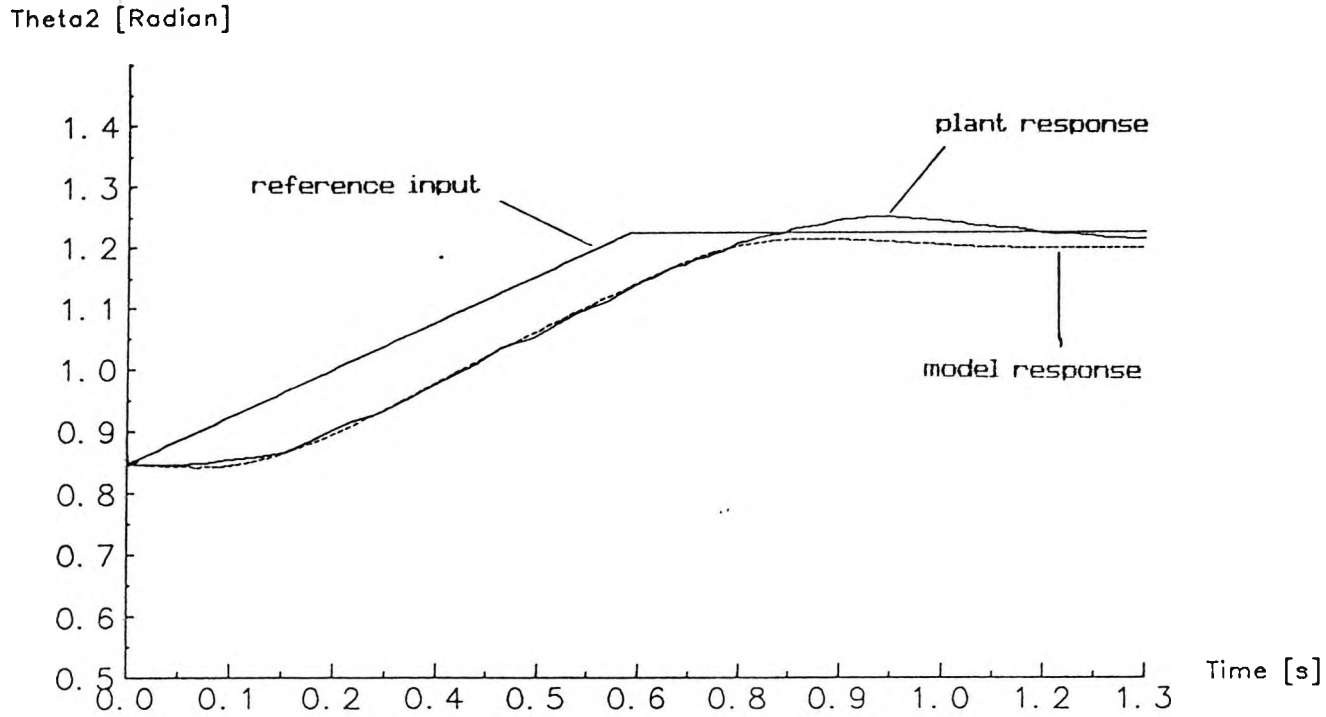

(a) Joint 1 response.



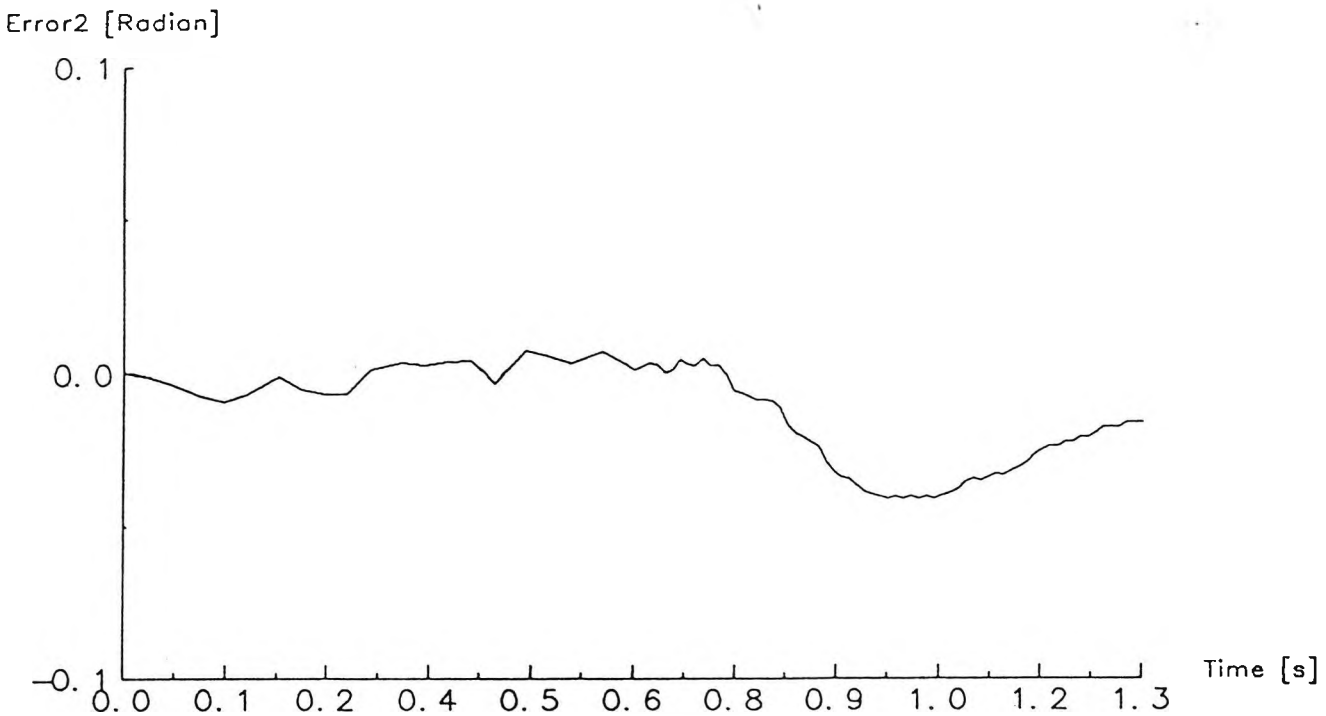

(b) Joint 1 error.

**Figure 7.29**

Model with viscous and Coulomb friction.



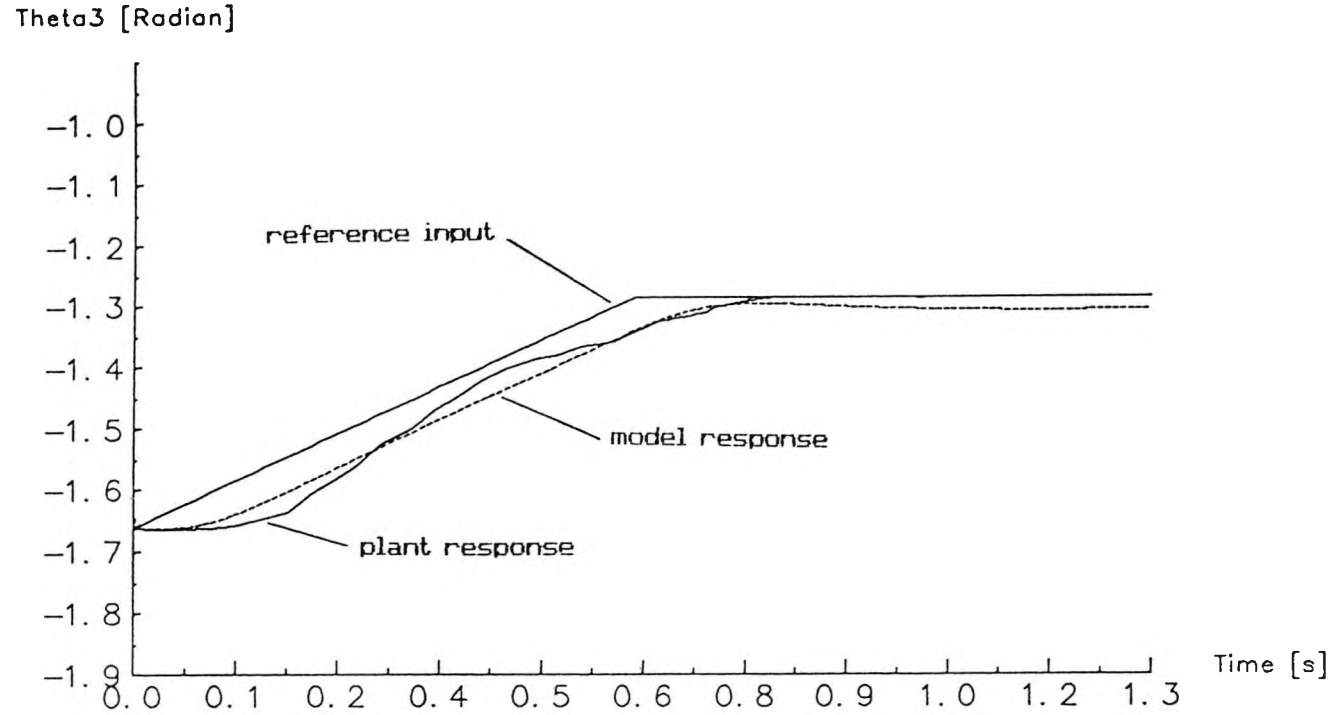

(a) Joint 2 response.



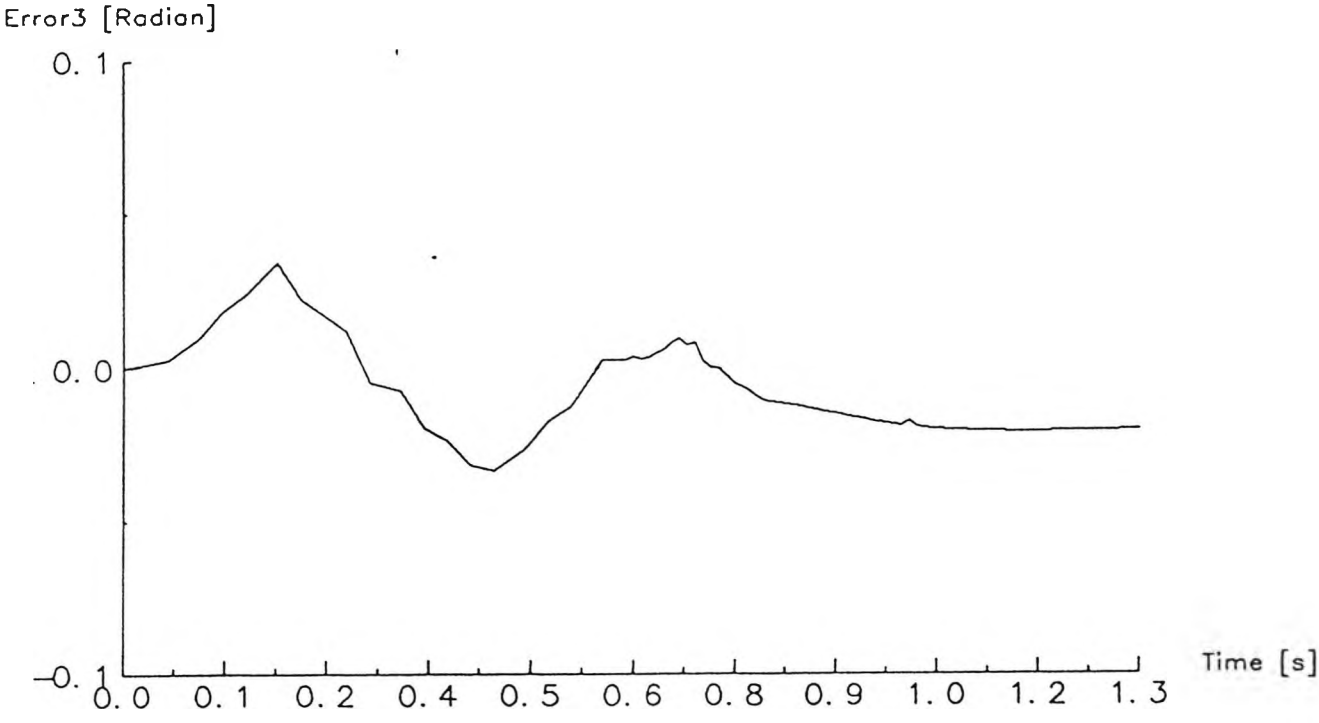

(b) Joint 2 error.

**Figure 7.30**

Model with viscous and Coulomb friction.




(a) Joint 3 response.




(b) Joint 3 error.

**Figure 7.31**

Model with viscous and Coulomb friction.

## 7.5 Conclusions

The main objective of the distortion technique for validating a model quantitatively is the importance to know sensitivities of the model parameters. If a model has sensitive parameters, this means that the model equations that govern the dynamic system behaviour really represent the real system. On the other hand, if a model has a small mean squared error value, the model equations do not necessarily represent the real system since a small mean squared error may be achieved by merely manipulating the parameter values. The fidelity criterion, $\sum\lambda^2$, indicates how sensitive the model is. The weakness of this technique is in assigning the expected standard deviation of each parameter. It is difficult to assign this value and unfortunately there is no certain rule to do this task. So, assigning these values is based on knowledge of the physical properties of the system.

In applying this technique, the progression of $\sum\lambda^2$ from a simple model to the most complex one is the most important thing to be noted. This will indicate whether the model is improving or not. In the simplest model where the back EMF constant of motor 2 was not identified, although the mean squared error value looks small, the fidelity criteria is very poor. This condition certainly indicates that further investigation of the system is necessary. It is then found that the back EMF constant of motor 2 needs to be identified since this parameter is susceptible to a change due to wear. Having identified this parameter, significant improvements both in the fidelity criterion and in the mean squared error value are obtained. In order to be more realistic, some sort of energy dissipation approximation is introduced into the model ie. the existence of friction since the friction effect may be quite large and is about 25% of the torques required to actuate the

manipulator in typical situations [Craig, 1986]. Introducing viscous damping does not greatly reduce the mean squared error value, but the parameter sensitivities which are the main point of the distortion technique increase. By introducing an approximate Coulomb friction into the model, the parameter sensitivities improve significantly, especially the fundamental parameters, and this leads to a $\sum\lambda^2$ of 0.9313.

From the basic model to the last model which has Coulomb friction approximation, the fundamental parameters of link 1 are not sensitive. This is because link 1 is the proximal link which means any changes in its parameters do not affect the upper links as can be understood from the Newton-Euler algorithm. Although the fidelity criterion is still slightly less than unity, it has been shown that the distortion technique can be applied to validate a robot arm quantitatively.

In a real time control situation in which computing time is very crucial, based on the results obtained, the interactions due to the inertial torques can be omitted as long as the trajectory does not involve much acceleration. Only interactions due to the gravity effects are taken into consideration. Hence, by omitting the inertial torque interactions, the computing time can be greatly reduced.

# CHAPTER 8

# APPLICATION IN CONTROL SYSTEM DESIGN

## 8.1 Introduction

One application of a simulation is to study a control system with the aid of a mathematical model. This work is very important since all control systems which use model based techniques rely on the mathematical model being used. New control systems can be developed and existing control systems can be examined through a simulation study using a validated mathematical model to represent a real system. There are many robot control techniques around belonging to the class of nonlinear multivariable systems which can be studied through a simulation. Many control algorithms are proposed based on simulation studies rather than on experimental studies. This is due to the fact that in simulation studies, computation time is not as important as in the case of experimental studies.

This chapter describes a simulation study using a robot mathematical model with both viscous and Coulomb friction, which was developed and validated in chapter 7, to represent the real system. Two control schemes are examined, one of them does not ignore the nonlinearities of the robot while the other is a very simple control law commonly used in industry which does not take the nonlinearities into account. The former, a partitioned servo control scheme [Craig, 1986] is compared with a built in ordinary PID control scheme which is used by the TQ MA2000 robot arm. The aim of this chapter is to do a comparison among the three models which were developed in the preceeding chapter and to show the insufficiency of the classical PID control scheme. Using a partitioned servo control law, the importance of modelling a friction term in a robot manipulator is highlighted. Confidence in

modelling friction effects is important although the trend of robot design is to reduce the friction of each joint [Fernandez, Bae and Everett, 1990].

### 8.2 Partitioned Servo Control Scheme [Craig, 1986]

This algorithm falls in the class of a computed torque technique and is based on the preposition that a set of highly nonlinear and coupled second order differential equations which governs the dynamic behaviour of a robot manipulator can be modified in such away that it appears as a new system which has a set of linear uncoupled second order differential equations. In addition, this new system has a unit inertia which further simplifies the problem. The salient advantage of a system which is linear and decoupled is that the configuration of all links does not affect the characteristics of the control system employed in the system. The evaluation of a control system is sufficiently performed in a single configuration. This in turn will yield a good control system in all areas inside the workspace. Details on the control law partitioning technique can be found in [Craig, 1986].

Disturbances, such as gravity effects and friction, in the system can be eliminated by injecting a compensation signal, and inertia coupling can be decoupled by manipulating the corresponding controlled signal. As in the other model based control techniques, this technique relies heavily on the accuracy of the mathematical model which is used to linearize and decouple the system. Lack of knowledge of parameters is the major problem in this technique and among them is friction effects. The contributions of friction effects which were modelled in chapter 7 are examined.

Since the result of the control partitioning technique depends on the accuracy of

the model being used to compute the required torques, three different cases are examined in which three different models with their corresponding optimised parameter values which were developed in chapter 7 are used to linearize and decouple the system. These three different models are the basic model, the model with viscous friction and the model with complete friction with $\sum\lambda^2$ of 0.4431, 0.5973 and 0.9313, respectively. With each model in each case is used in evaluating the inverse dynamics to obtain the required torques, the inferiority/superiority of each model can then be examined and compared with other models.

Figure (8.1) and figure (8.2) show the block diagram of the system which utilizes a control law partitioning technique. Unless stated, all variables are related to the variables in chapter 7 and '^' denotes an optimised parameter value of the model.

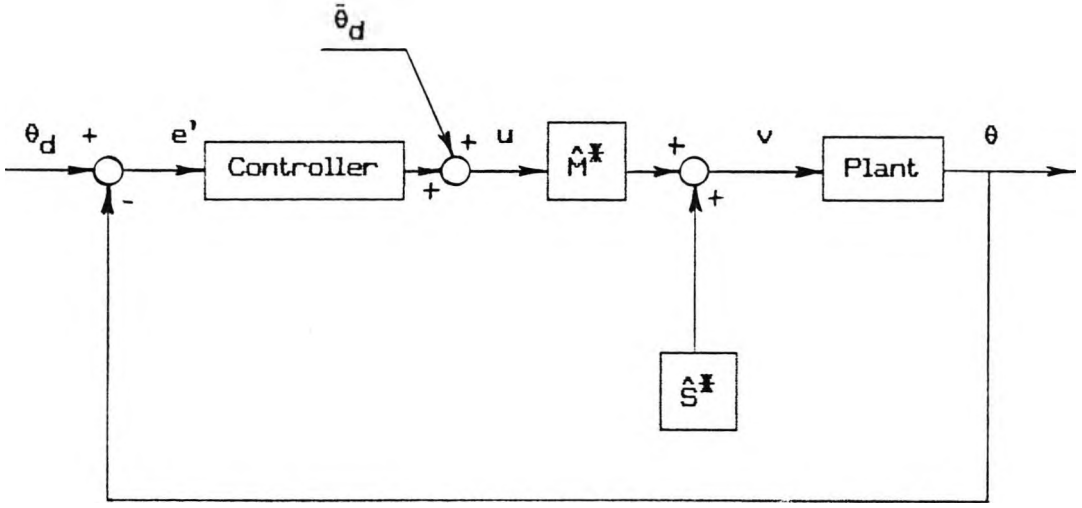



Figure 8.1

Block diagram of a closed loop system employing a control law partitioning technique.

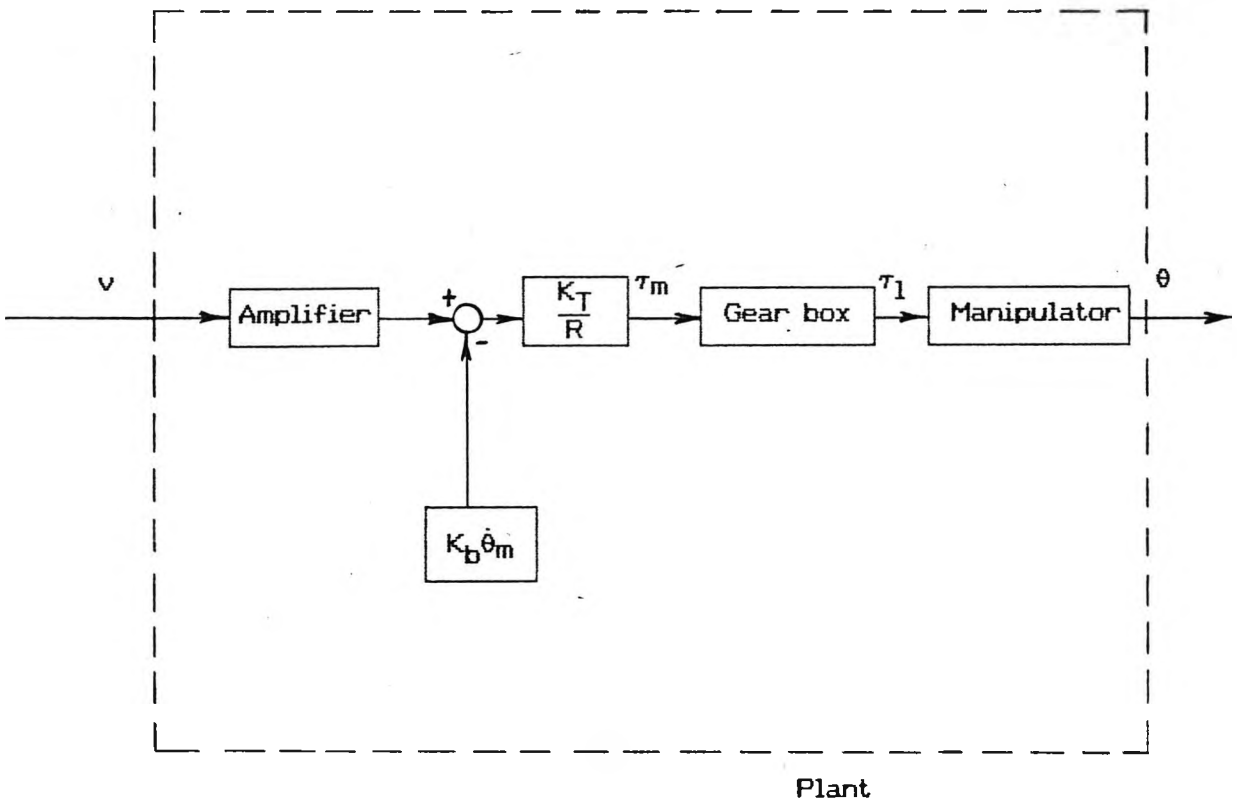



Figure 8.2

Block diagram inside a plant.

The generalized torque vector of a three degree of freedom manipulator, with respect to the link side, can be expressed in the following form

$$\tau_l = D(\theta)\,\ddot{\theta} + P(\theta,\dot{\theta}) \qquad (8.1)$$

where: $\tau_l$ = a 3x1 generalized torque vector with respect to the link side.

$D$ = a 3x3 symmetric inertial acceleration related matrix.

$P$ = a 3x1 disturbance vector with respect to the link side.

$\theta$ = a 3x1 joint variable vector with respect to the link side.

$\dot{\theta}$ = a 3x1 joint velocity variable vector with respect to the link side.

$\ddot{\theta}$ = a 3x1 joint acceleration variable vector with respect to the link side.

Expressing the generalized torque vector with respect to the actuator side gives the following (dropping $\theta$ and $\dot{\theta}$ for brevity)

$$\tau_m = N\,D\,\ddot{\theta} + N\,P \tag{8.2}$$

where: $\tau_m$ = a 3x1 generalized torque vector with respect to the actuator side.

$N$ = a diag($n_1$, $n_2$, $n_3$) gear ratio matrix.

Taking the actuator inertia into account, then the required output torque of each actuator, $\tau$, in order to drive each joint including its own inertia is given by

$$\tau = (N\,D + J)\,\ddot{\theta} + N\,P \tag{8.3}$$

where: $J$ = a diag($J_1$, $J_2$, $J_3$) actuator inertia matrix and $J_i = \frac{J_{mi}}{n_i}$.

From figure (8.2), a parameter matrix is introduced

$$K = \text{diag}(K_1, K_2, K_3) \tag{8.4}$$

$$K_i = \frac{R_i}{A_i K_{Ti}} \tag{8.5}$$

where: $K$ = an integrated amplifier and actuator parameter matrix.

$R_i$ = terminal resistance [Ohm].

$A_i$ = gain of the linear power amplifier.

$K_{Ti}$ = torque constant of the actuator [Newton.m/Amp].

and let the proportionally back EMF matrix be

$$V = \text{diag}(V_1, V_2, V_3) \tag{8.6}$$

$$V_i = \frac{K_{bi}}{A_i} \tag{8.7}$$

where: $K_{bi}$ = back EMF constant of actuator [Volt.s/radian].

Based on figure (8.2), where $\dot{\theta}_i = n\dot{\theta}_{mi}$, the relationship between the voltage which is applied to the input of the amplifier and the response of the robot arm is given by

$$v = K (N D + J) \ddot{\theta} + K N P + N^{-1} V \dot{\theta} \tag{8.8}$$

If the second and the third terms are treated as disturbances, equation (8.8) then becomes

$$v = M^{*} \ddot{\theta} + S^{*} \tag{8.9}$$

$$M^{*} = K (N D + J) \tag{8.10}$$

$$S^{*} = K N P + N^{-1} V \dot{\theta} \tag{8.11}$$

Equation (8.9) is the complete inverse dynamics equation of a three degree of freedom manipulator where both the actuator and the linear amplifier are taken into account. When a PID controller is employed at each joint then, from figure (8.1), the applied input voltage to the amplifier, $v$, is computed by the inverse dynamics equation, which is given by equation (8.9), as

$$v = \hat{M}^{*} \left( K_P (\theta_d - \theta) + K_I \int (\theta_d - \theta)\, dt + K_V (\dot{\theta}_d - \dot{\theta}) + \ddot{\theta}_d \right) + \hat{S}^{*} \qquad (8.12)$$

where: $K_P$ = a diag($K_{P1}$, $K_{P2}$, $K_{P3}$) proportional gain matrix.

$K_I$ = a diag($K_{I1}$, $K_{I2}$, $K_{I3}$) integral gain matrix.

$K_D$ = a diag($K_{D1}$, $K_{D2}$, $K_{D3}$) derivative gain matrix.

A subscript 'd' denotes the desired value (preplanned trajectory).

Let the error vector be

$$e' = \theta_d - \theta \qquad (8.13)$$

and equating equation (8.9) and (8.12) yields

$$K_P e' + K_I \int e'\, dt + K_D \dot{e}' + \ddot{e}' = \hat{M}^{*-1} \left( (M^{*} - \hat{M}^{*}) \ddot{\theta} + (S^{*} - \hat{S}^{*}) \right) \qquad (8.14)$$

From equation (8.14), the objective of the partitioned control technique is accomplished, ie. the system becomes linear and decoupled, and appears as a unit inertia system, only if the dynamic mathematical model is accurate which implies that $\hat{M}^{*} = M^{*}$ and $\hat{S}^{*} = S^{*}$. Thus, since it is assumed that there is no modelling error, from equation (8.14) the characteristic equation for joint i is then given by

$$K_{Pi}\, e'_i + K_{Ii} \int e'_i\, dt + K_{Di}\, \dot{e}'_i + \ddot{e}'_i = 0 \qquad (8.15)$$

Or, in Laplace form,

$$s^3 + K_{Di}\, s^2 + K_{Pi}\, s + K_{Ii} = 0 \qquad (8.16)$$

As the new system is linear, choosing the right values of controller gains for a specific characteristic will be valid for all joint configurations. By locating the roots of the characteristic equation in the right places, a critical damped response can be obtained. In a third order system, a critical damped resopnse can be approximated by having equal values for the first two roots and placing the third real root far enough to the left of the first two roots. Since the transient time of the third root is much faster, its transient response is then negligible.

Assigning,

$$K_P = 11.0 \qquad (8.17)$$

$$K_I = 5.0 \qquad (8.18)$$

$$K_D = 7.0 \qquad (8.19)$$

for all joints will put the first two roots at -1 and the third root at -5. Since the transient time of the third root is five times quicker than the transient times of the first two roots, the transient time of the third root is then fast enough to be neglected [DiStefano III, Stubberrud and Williams, 1976]. By having all roots in this configuration, the system can be approximated to have a critical damped response.

Since it is assumed that there is no modelling error, consequently the characteristic of the system which is given by equation (8.15) is assumed to apply to all three cases in designing a control system. This results that the PID controller gains are equal for all cases. However, although in all cases the PID controller gains are equal, each model gives its own values of $\hat{M}^{*}$ and $\hat{S}^{*}$ in order to make the system linear and decoupled with unit inertia. Equation (8.14) shows that the performance of the system depends on the accuracy of the model being used and this is to be examined in the following cases. The basic model and the model with viscous friction, which were developed in chapter 7, will be used to compute the values of $\hat{M}^{*}$ and $\hat{S}^{*}$ in case 1 and case 2, respectively, where the plant is represented by a model with complete friction. So, how true the assumption that there is no modelling error in designing the control system using the control partitioning technique, can be observed from the resulting responses. The importance of the inclusion of the Coulomb friction in the model is examined by implementing the model with complete friction to compute the values of $\hat{M}^{*}$ and $\hat{S}^{*}$ and is given in case 3.

### 8.2.1 Case 1 : Basic Model

The dynamic behaviour of the robot manipulator is approached by the basic model as

$$\hat{\tau}_1 = \hat{D}(\theta)\,\ddot{\theta} + \hat{H}(\theta,\dot{\theta}) + \hat{G}(\theta) \qquad (8.20)$$

and

$$\hat{P} = \hat{H}(\theta,\dot{\theta}) + \hat{G}(\theta) \qquad (8.21)$$

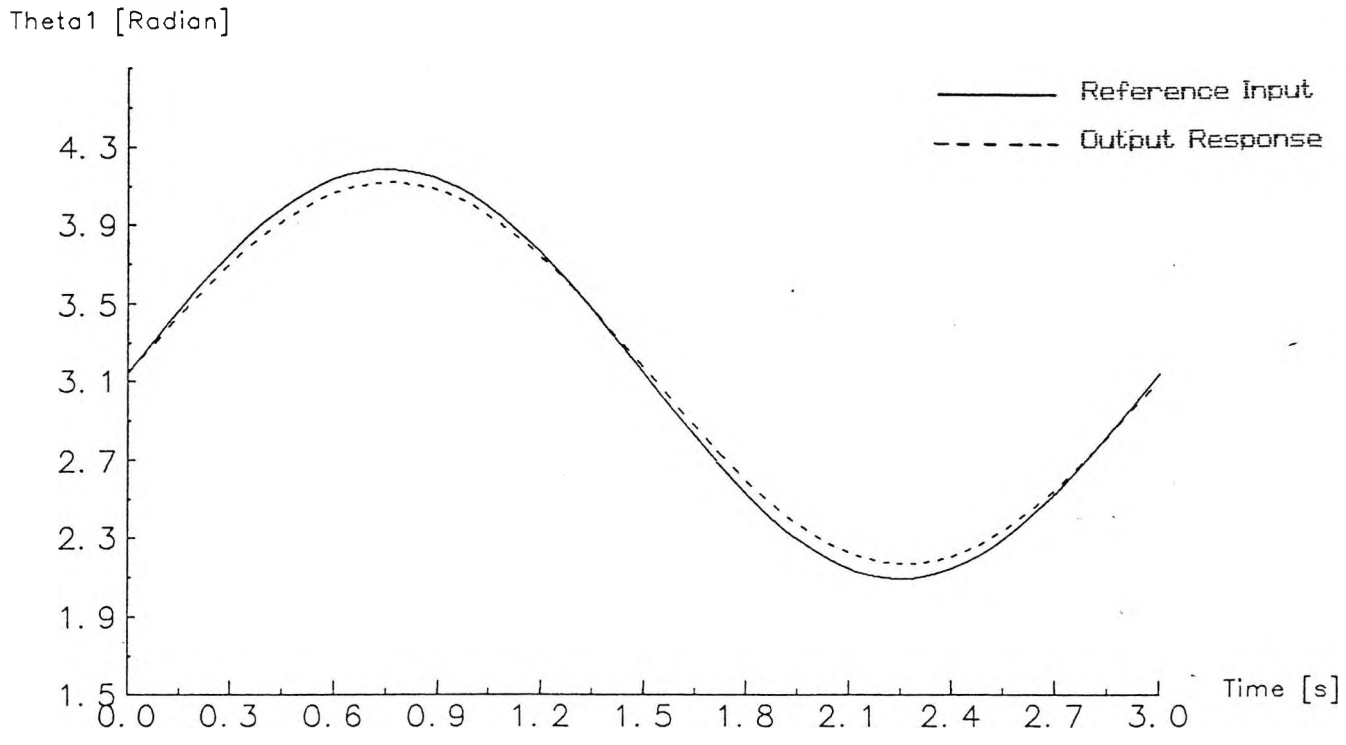
Equation (8.10) and (8.11) then are approached by (dropping $\theta,\dot{\theta}$)

$$\hat{M}^{*} = K\,(N\,\hat{D} + J) \tag{8.22}$$

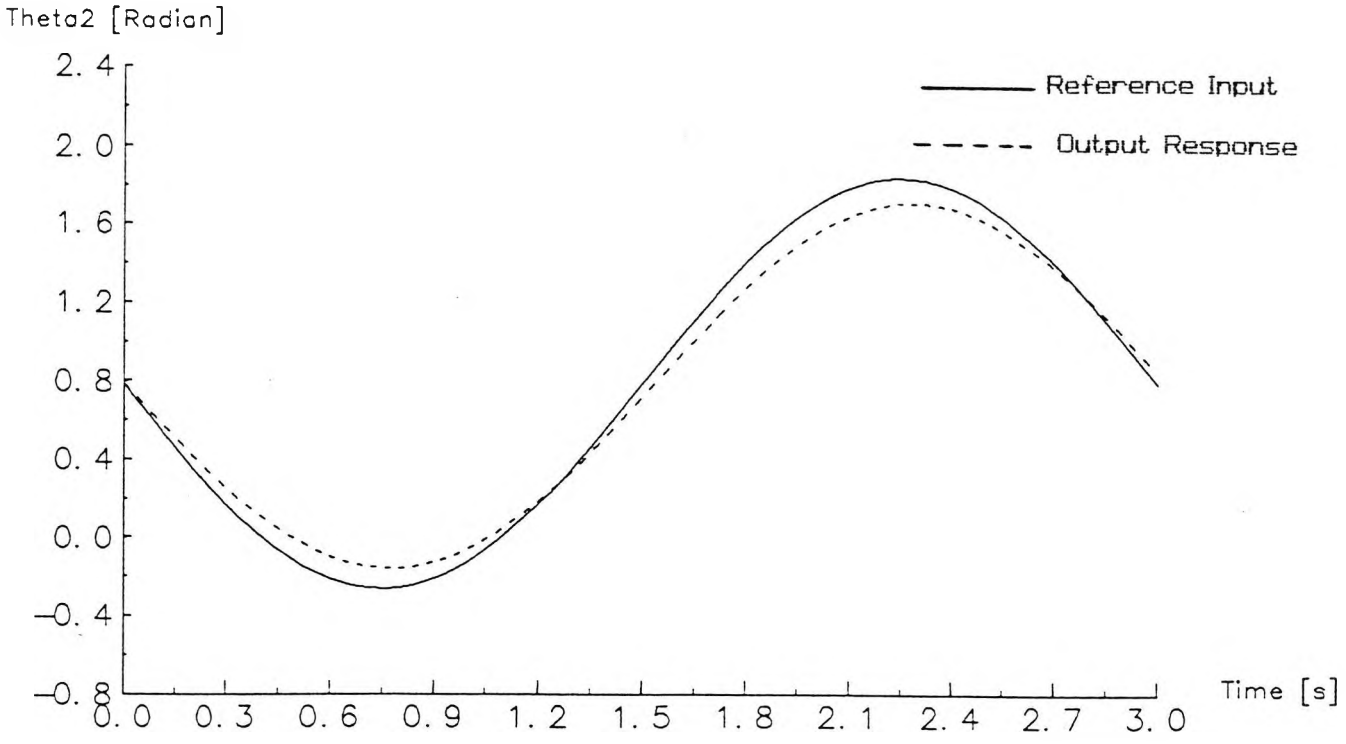
$$\hat{S}^{*} = K\,N\,(\hat{H} + \hat{G}) + N^{-1}\,\hat{V}\,\dot{\theta} \tag{8.23}$$

For convenience, a sinusoidal trajectory with a period of 3 seconds is used for all joints. The controller gains for all joints are given by equations (8.17), (8.18) and (8.19).

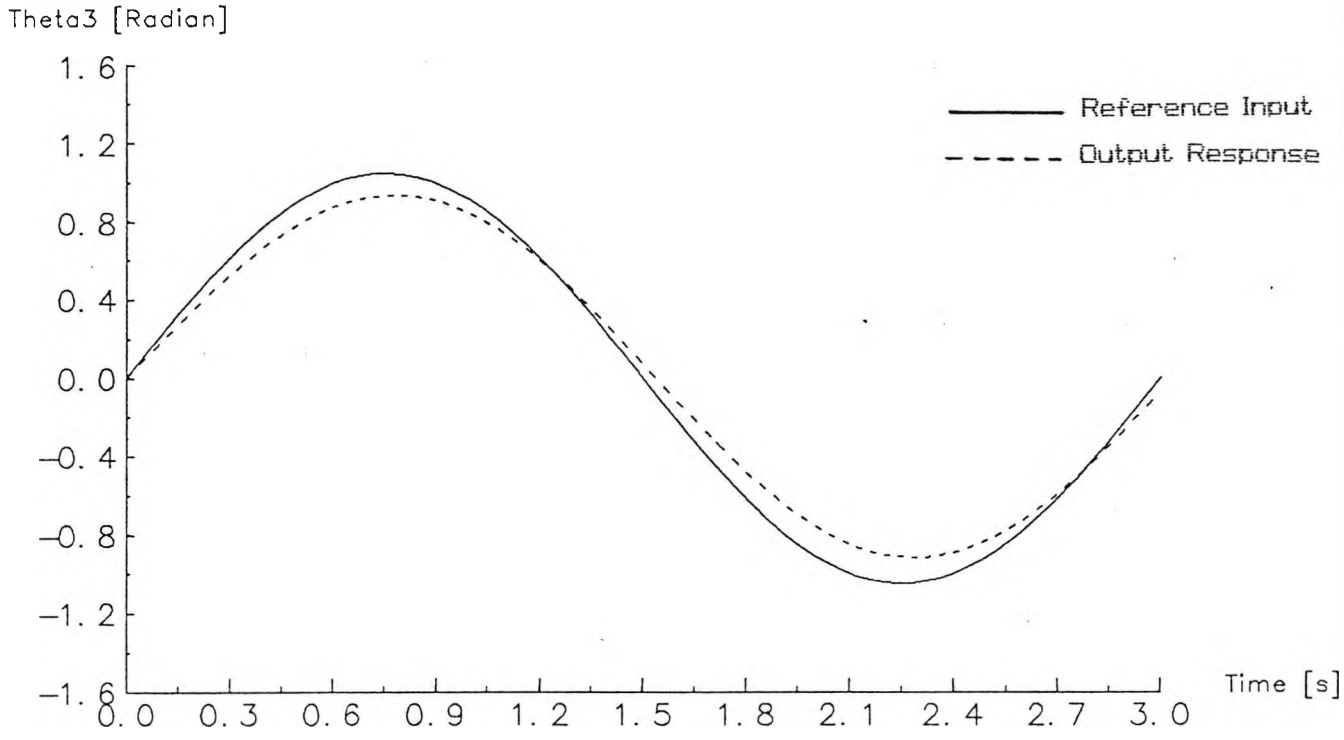
Initially, link 1 is at 180 degrees, link 2 is at 45 degrees (0.7854 radian) and link 3 is at 0 degrees. An amplitude of 60 degrees (1.0472 radian) is given to all joint trajectories where the trajectory of joint 2 has a different phase angle. Figure (8.3) shows the responses where the basic model with its optimised fundamental parameters (mass and length of each link) and back EMF coefficient values is used to compute the values of $\hat{M}^{*}$ and $\hat{S}^{*}$. It is seen that the output responses of all joints have significant reductions in the amplitude in comparison with their corresponding input trajectories.




(a) Joint 1 response




(b) Joint 2 response




(c) Joint 3 response

Figure 8.3

The system response with a basic model to compute $\hat{M}^*$ and $\hat{S}^*$

### 8.2.2 Case 2 : Model With Viscous Friction

With a model incorporating viscous friction, $v$, used to compute $\hat{M}^*$ and $\hat{S}^*$, the dynamic behaviour of the robot is approximated by

$$\hat{\tau}_1 = \hat{D}(\theta)\,\ddot{\theta} + \hat{H}(\theta,\dot{\theta}) + \hat{G}(\theta) + \hat{v}(\dot{\theta}) \qquad (8.24)$$

and

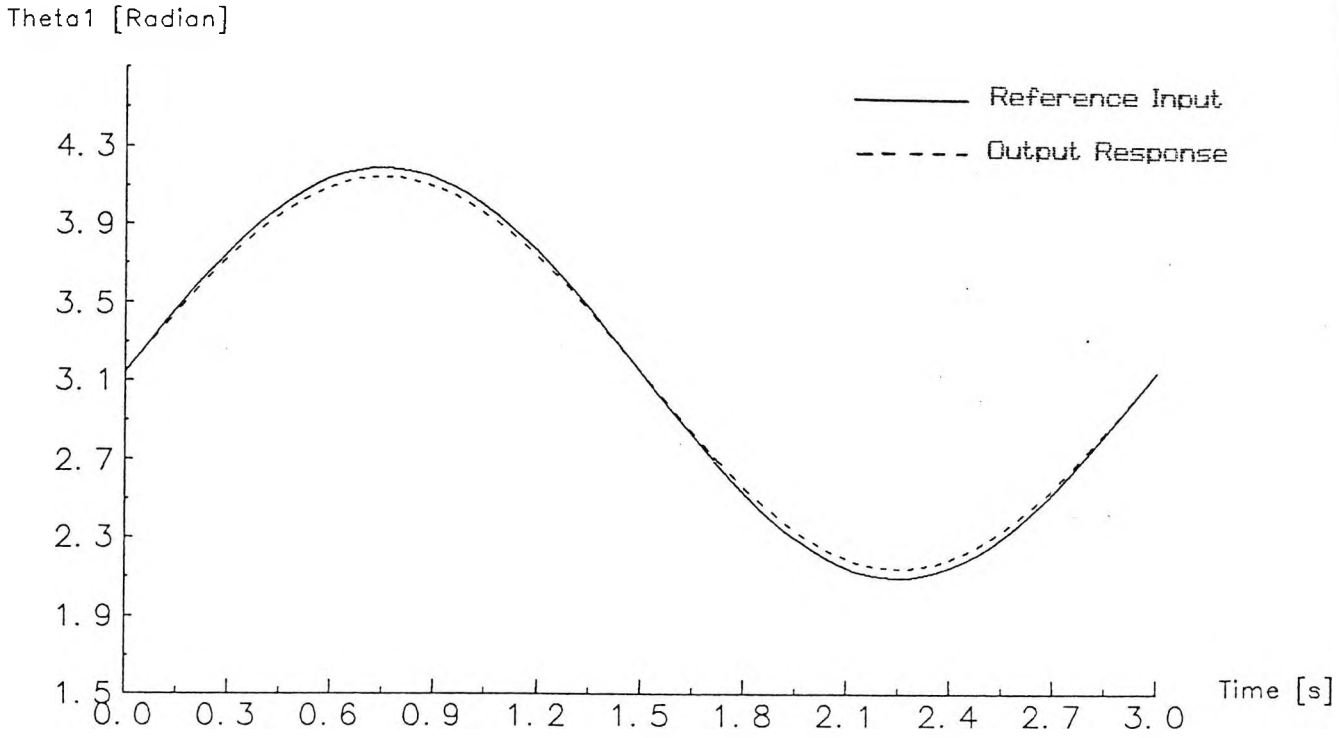
$$\hat{P} = \hat{H}(\theta,\dot{\theta}) + \hat{G}(\theta) + \hat{v}(\dot{\theta}) \qquad (8.25)$$

The expression to approximate $\hat{M}^*$ is similar to the one in the previous case, ie.

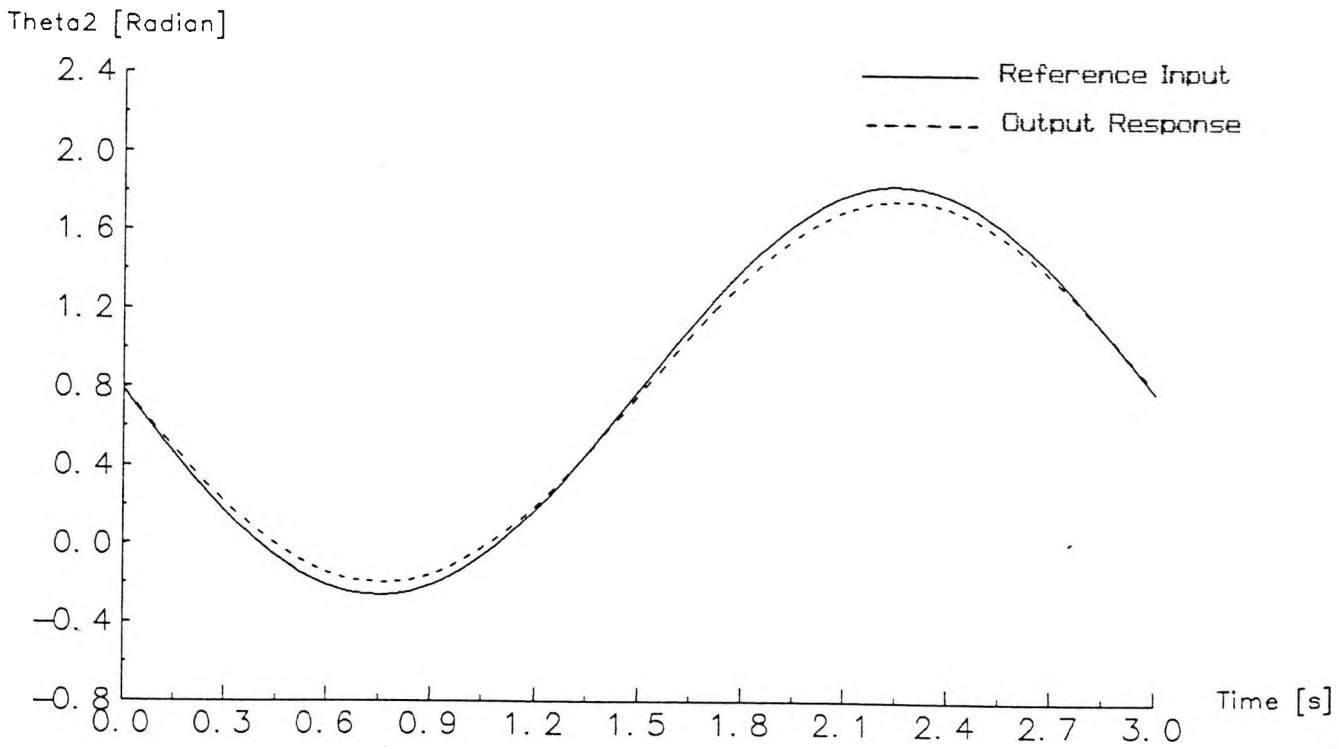
equation (8.22) but with different optimised fundamental parameter values while the disturbance is approximated as

$$\hat{S}^{*} = K N (\hat{H} + \hat{G} + \hat{v}) + N^{-1} \hat{V} \dot{\theta} \qquad (8.26)$$

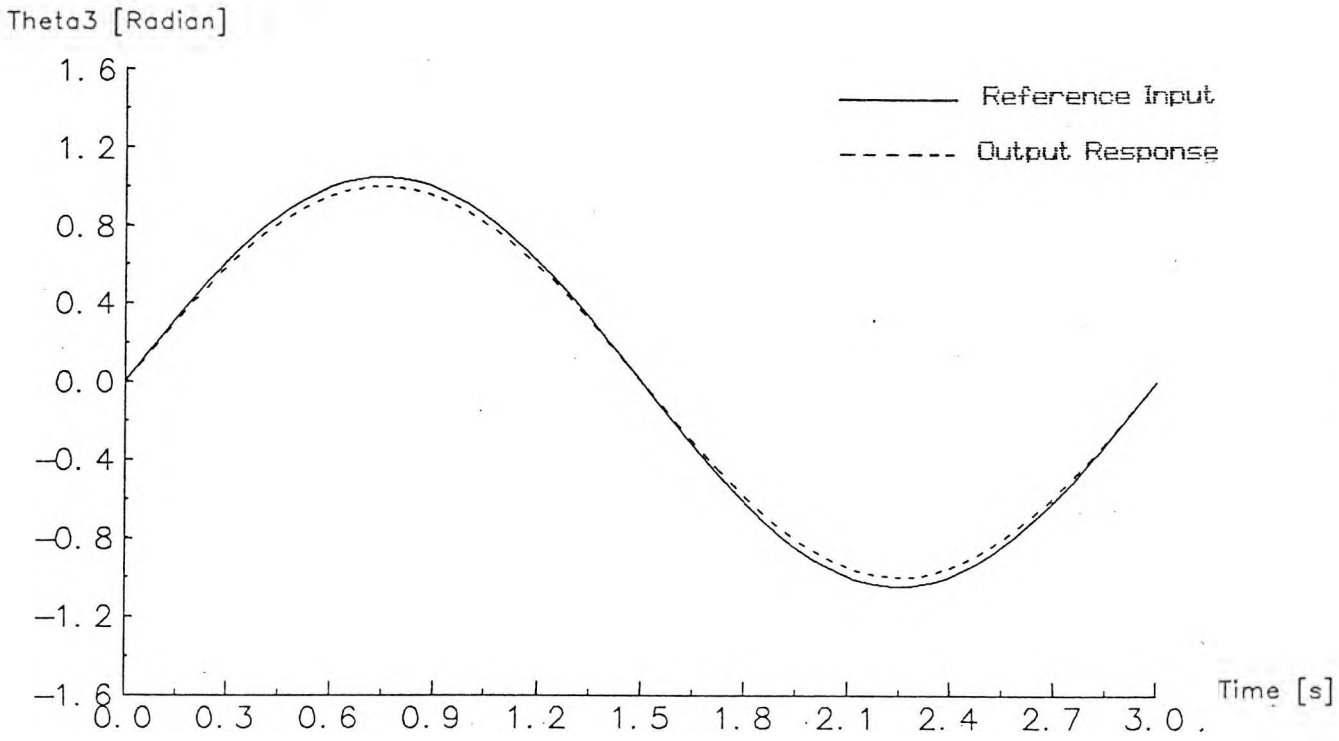
The initial conditions of link 1, link 2 and link 3 are the same as in case 1, ie. link 1 is at 180 degrees, link 2 is at 45 degrees (0.7854 radian) and link 3 is at 0 degrees. Sinusoidal trajectories with the amplitude of 60 degrees (1.0472 radian) and the period of 3 seconds as in case 1 are given to all joints. Using the same initial values, trajectories and controller gains as in case 1, the accuracy of this model in linearising and decoupling the system can then be observed. Figure (8.4) shows the responses where the model incorporating viscous friction with its corresponding optimised parameter values is used to compute the values of $\hat{M}^{*}$ and $\hat{S}^{*}$. The response of each joint is noticeably better, where the amplitude is enlarged, than the responses in the previous case.




(a) Joint 1 response




(b) Joint 2 response




(c) Joint 3 response

Figure 8.4

The system response with a model incorporating viscous friction to compute $\hat{M}^{*}$ and $\hat{S}^{*}$

### 8.2.3 Case 3 : Model With Viscous And Coulomb Friction

Coulomb friction, C', is added into the model in order to improve the system performance, and the required torque to drive each joint is approximated by the inverse dynamics equation as

$$\hat{\tau}_1 = \hat{D}(\theta)\,\ddot{\theta} + \hat{H}(\theta,\dot{\theta}) + \hat{G}(\theta) + \hat{v}(\dot{\theta}) + \hat{C}'(\theta) \qquad (8.27)$$

and

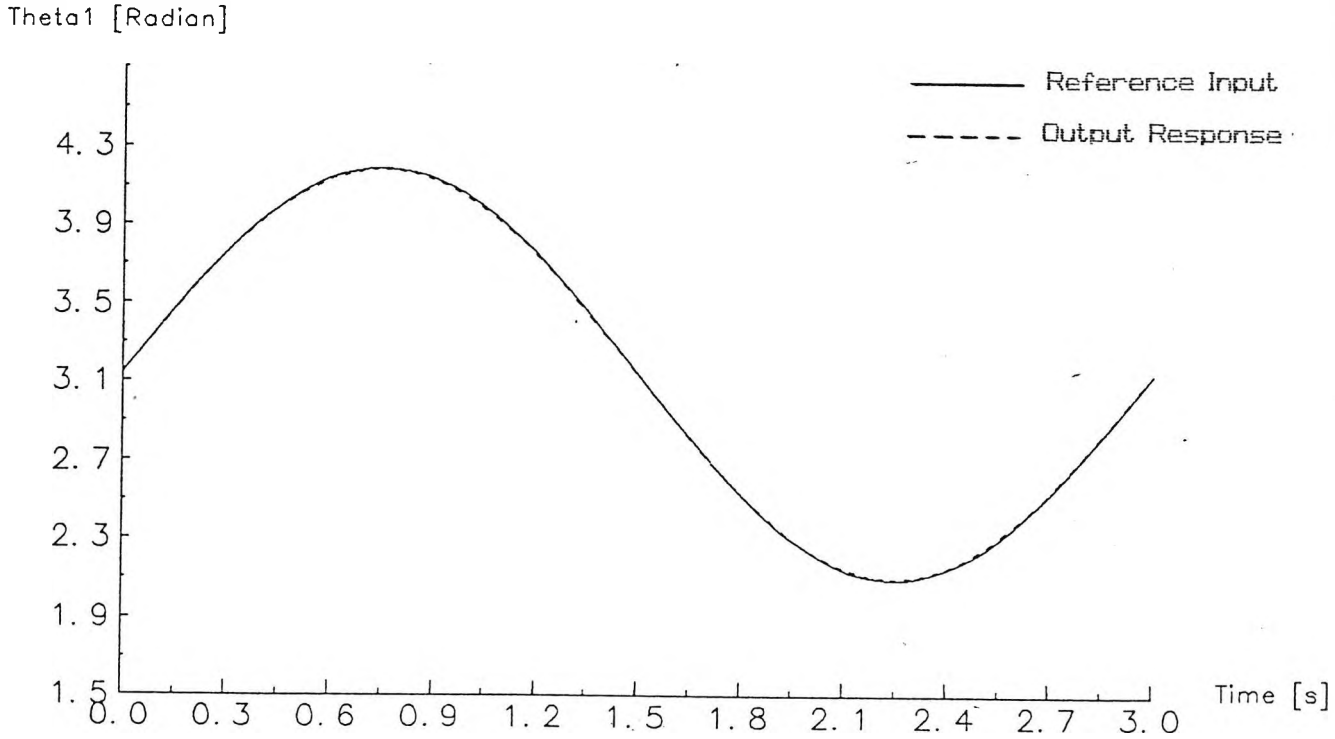
$$\hat{P} = \hat{H}(\theta,\dot{\theta}) + \hat{G}(\theta) + \hat{v}(\dot{\theta}) + \hat{C}'(\theta) \qquad (8.28)$$

and as in the previous cases, the expression to approximate $\hat{M}^{*}$ is given by equation (8.22) with different optimised fundamental parameter values. By giving the addition of a Coulomb friction term, the disturbance is then approximated as

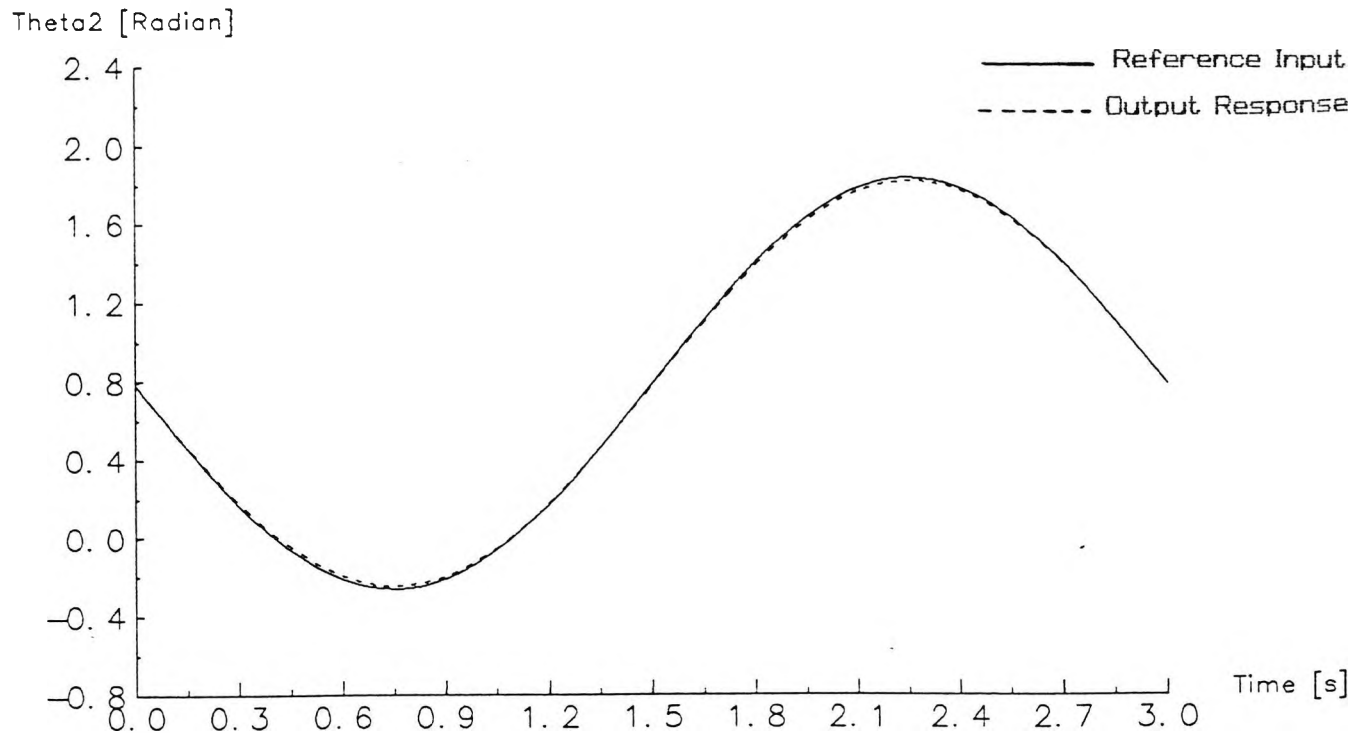
$$\hat{S}^{*} = K N (\hat{H} + \hat{G} + \hat{D} + \hat{C}) + N^{-1} \hat{V} \dot{\theta} \qquad (8.29)$$

The same initial conditions as in the previous cases are applied to the link 1, link 2 and link 3 where link 1 is at 180 degrees, link 2 is at 45 degrees (0.7854 radian) and link 3 is at 0 degrees. The input trajectories to each joint are not changed as in case 1 and case 2 with the same amplitude and period. The controller gains for all joints are kept constant as in the previous cases, so that the resulting performance of each joint response can be compared with the previous results. Figure (8.5) shows the responses where the model incorporating both viscous and Coulomb friction with its corresponding optimised parameter values is used to compute the values of $\hat{M}^{*}$ and $\hat{S}^{*}$. The result shows that the overall system performance is further improved in comparison with the performance in case 2 where all joint responses have good agreements with the sinusoidal inputs.

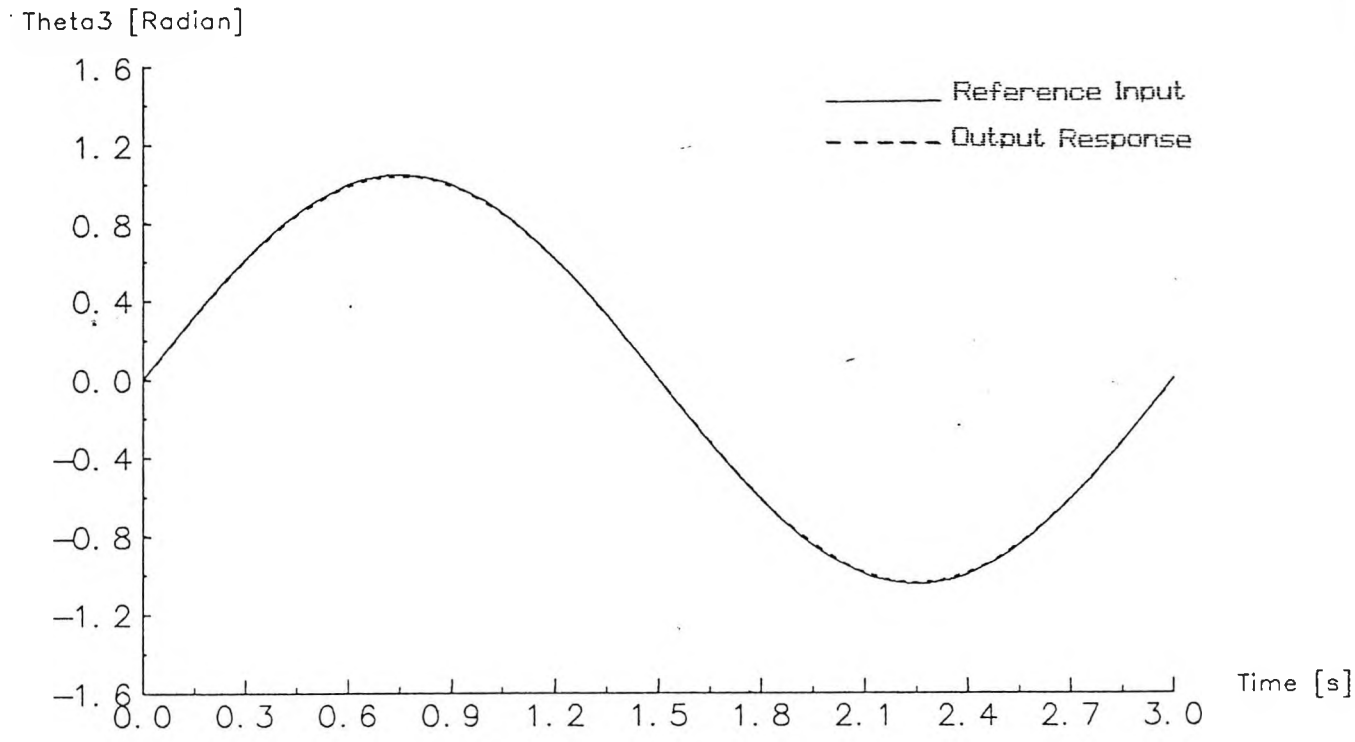
With a partitioned control technique employed in the system, the gain settings can be kept constant for all areas inside the work space without degrading the responses of all joints.




(a) Joint 1 response




(b) Joint 2 response




(c) Joint 3 response

Figure 8.5

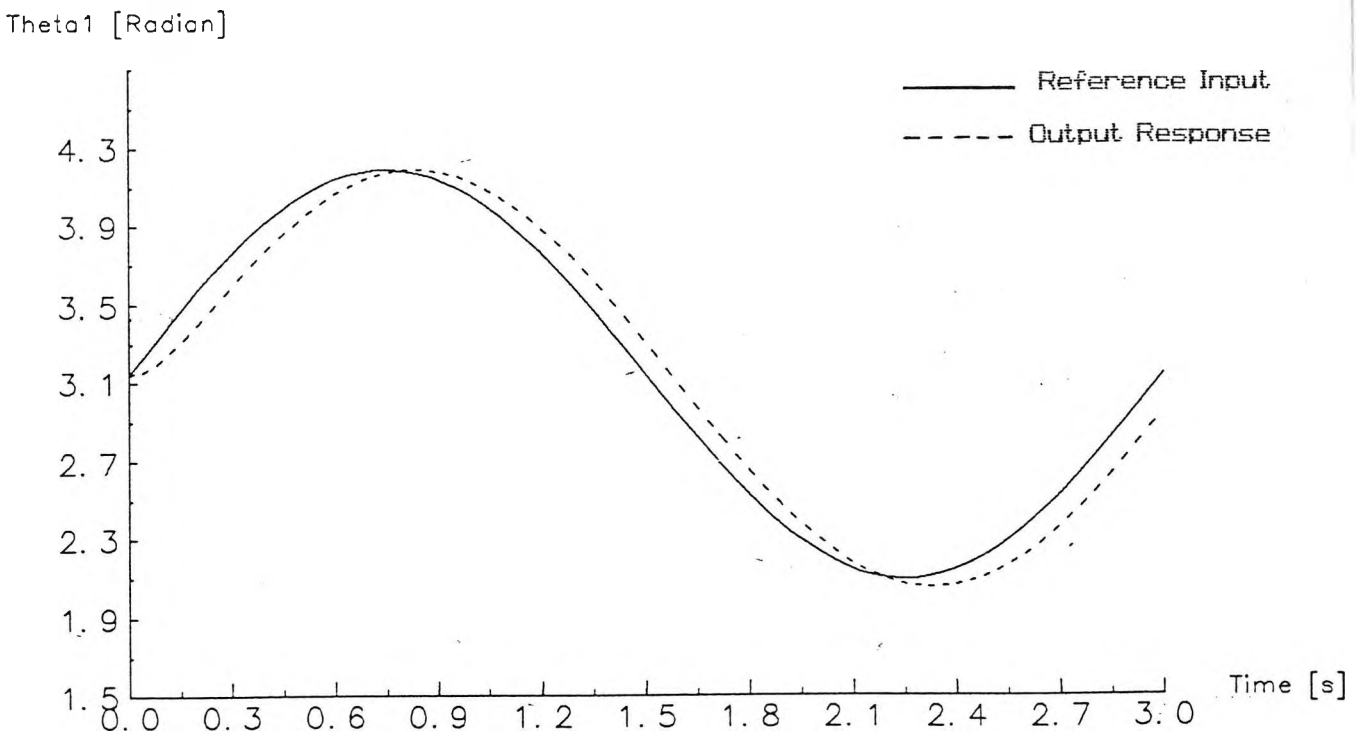
The system response with a model incorporating both viscous and Coulomb friction to compute $\hat{M}^{\star}$ and $\hat{S}^{\star}$

## 8.3 Classical PID Control Scheme

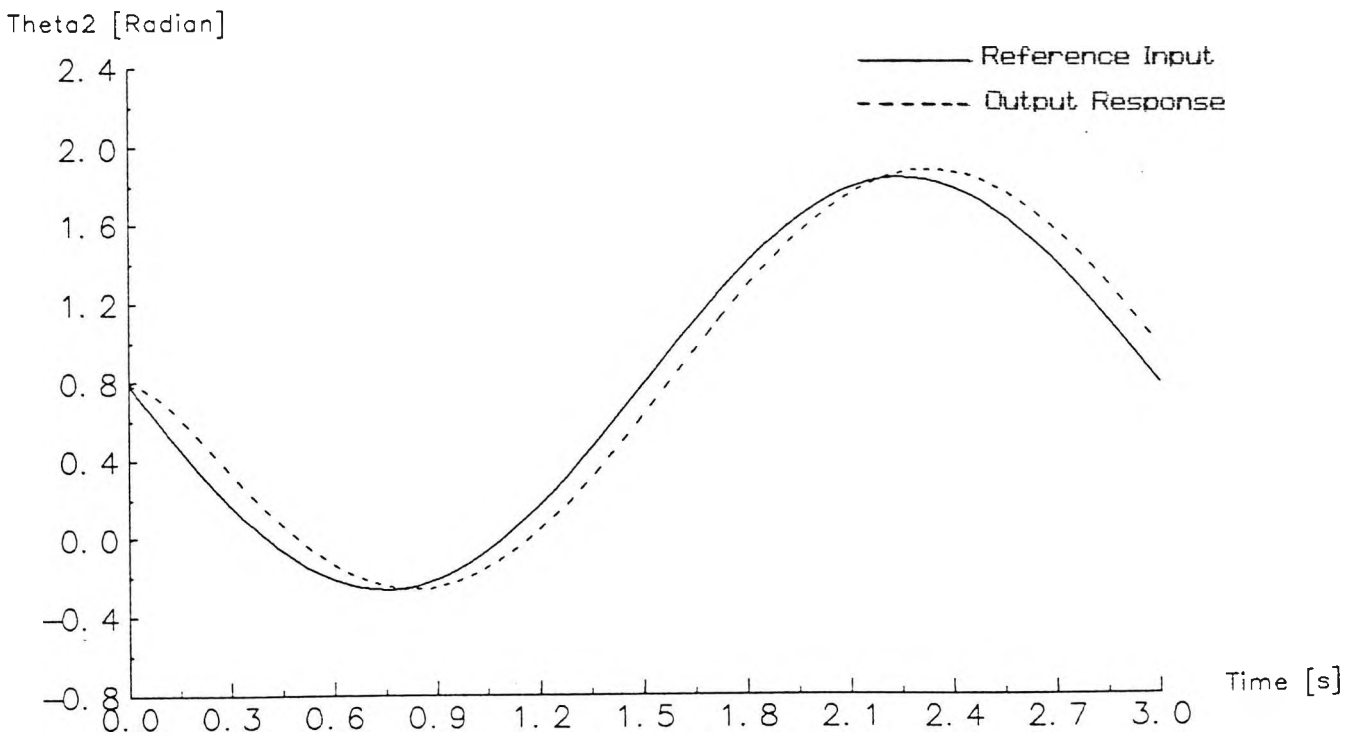
The classical PID control scheme is expected to have a poorer performance when applied to a robot manipulator in comparison to the PID control partition scheme. This classical technique is one of the earliest techniques applied to industrial robot manipulators. Nonlinearities in the robot mathematical model mean gravity and friction effects change as the robot moves from one configuration to another configuration. Things get even worse as the inertial acceleration related matrix D also changes. Hence, it is impossible to select fixed gains to a desired characteristic for all areas inside the work space

Initially, link 1 is at 180 degrees, link 2 is at 45 degrees (0.7854 radian) and link 3 is at 0 degrees. These initial conditions are the same as in the partitioning control technique case. In order to have an idea how the system behave under this classical PID control scheme in comparison with the performance of the partitioning control technique, the same sinusoidal trajectories as in the partitioning control technique case are employed to all joints. The controller gains of each joint, which are selected by trial and error so that the best responses of all joints can be obtained, are as follows $K_{P1}$ = 25.4, $K_{I1}$ = 5.2 and $K_{D1}$ = 21.9; $K_{P2}$ = 29.2, $K_{I2}$ = 3.8 and $K_{D2}$ = 23.6; $K_{P3}$ = 22.8, $K_{I3}$ = 4.8 and $K_{D3}$ = 21.7. Figure 8.6 shows the responses of each joint. In this gain setting, all joints follow their corresponding trajectory nicely, although each joint exhibits a phase lag.

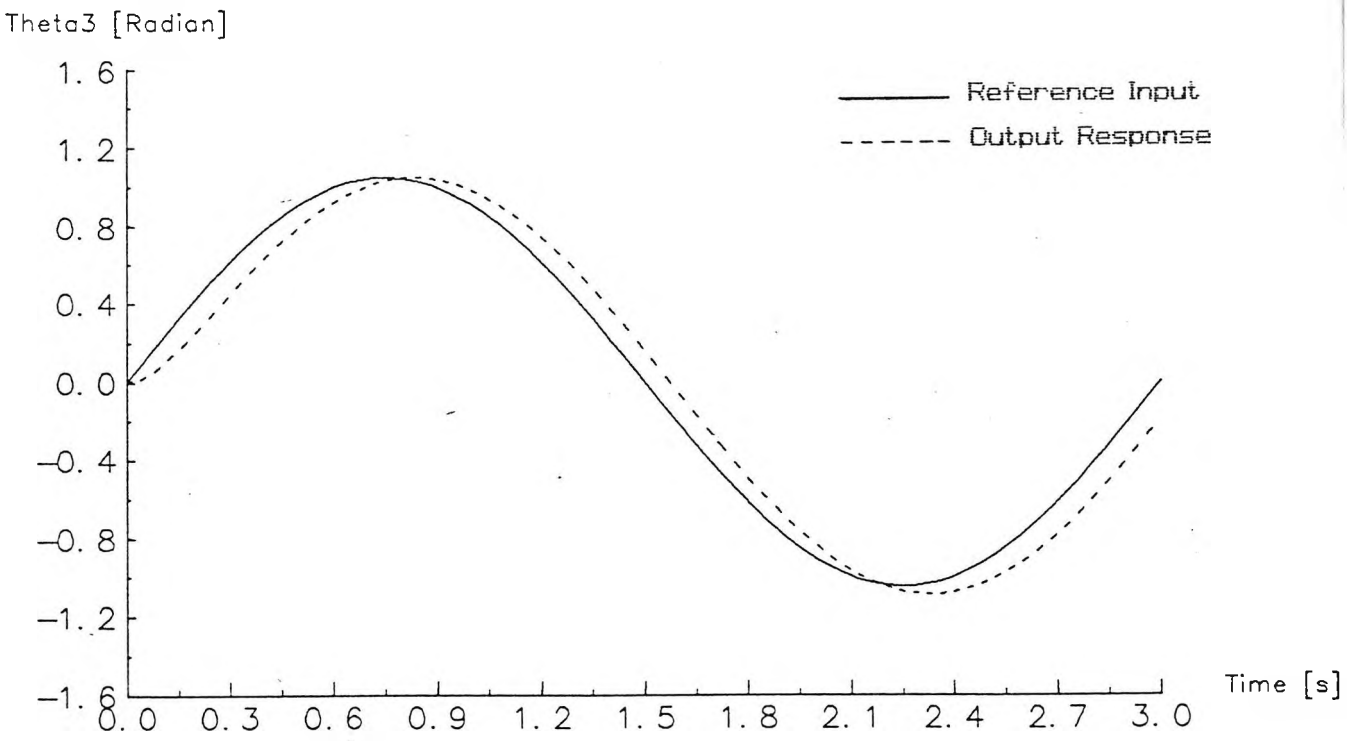
To show nonlinearities in a robot control system, a different initial value is given to joint 2 while initial values of other joints remain constant. No changes are made to all controller gains and trajectory of each joint. In this case, the initial value of joint 2 is at 90 degrees (1.5708 radian). With this configuration, as link 2 and link 3 are at right angles (vertical position), the inertia value suffered by joint 1 due to its own joint acceleration is at its minimum and gravity effects at both joint 2 and joint 3 are at their minimum too. All these changes give significant effect to the responses of all joints. Figure (8.7) shows that joint 2 and joint 3 suffer considerably overshoots.




(a) Joint 1 response



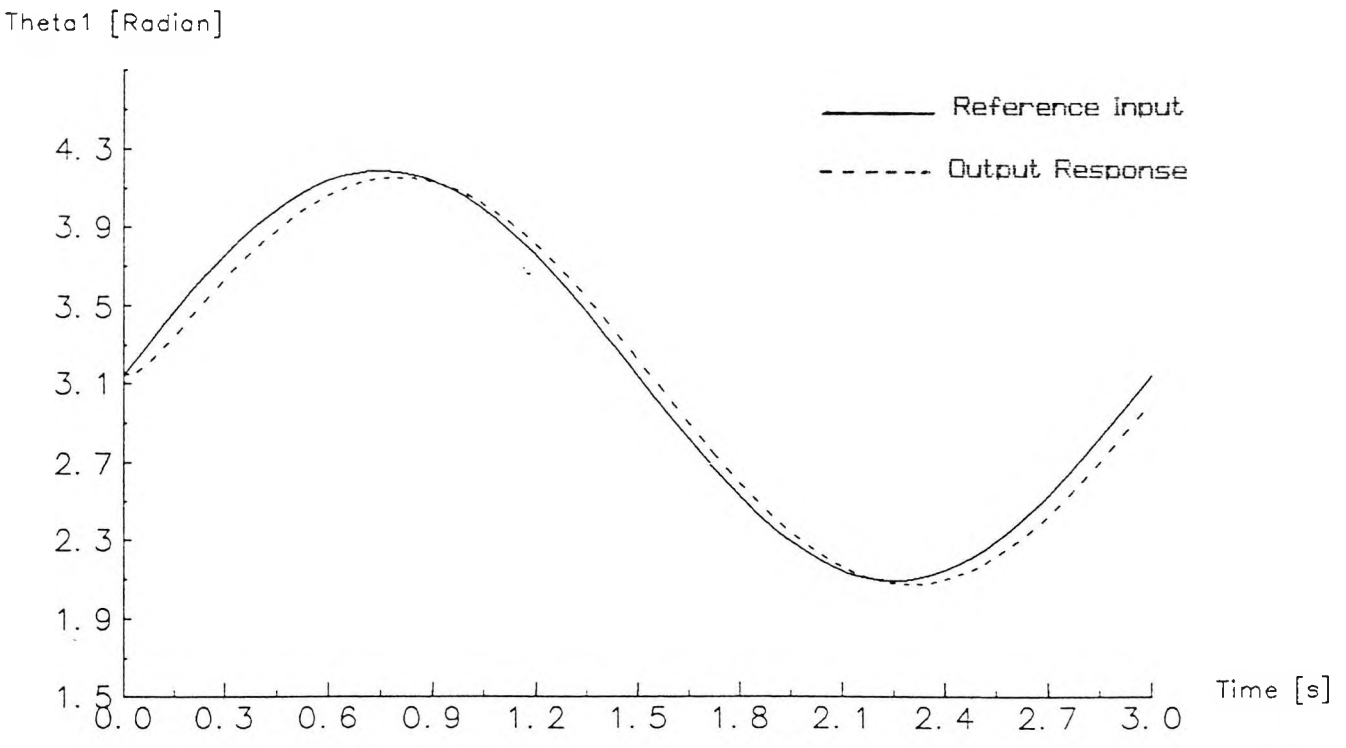

(b) Joint 2 response



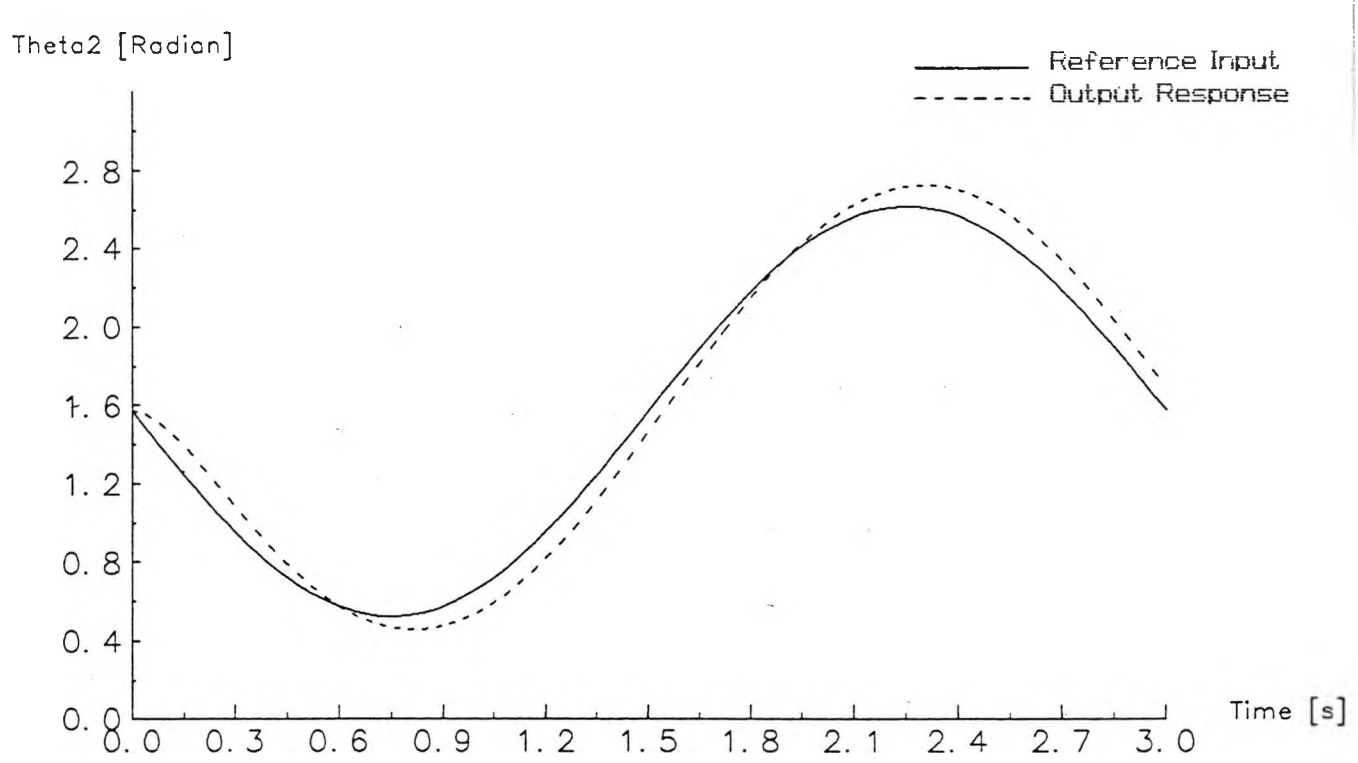

(c) Joint 3 response

Figure 8.6

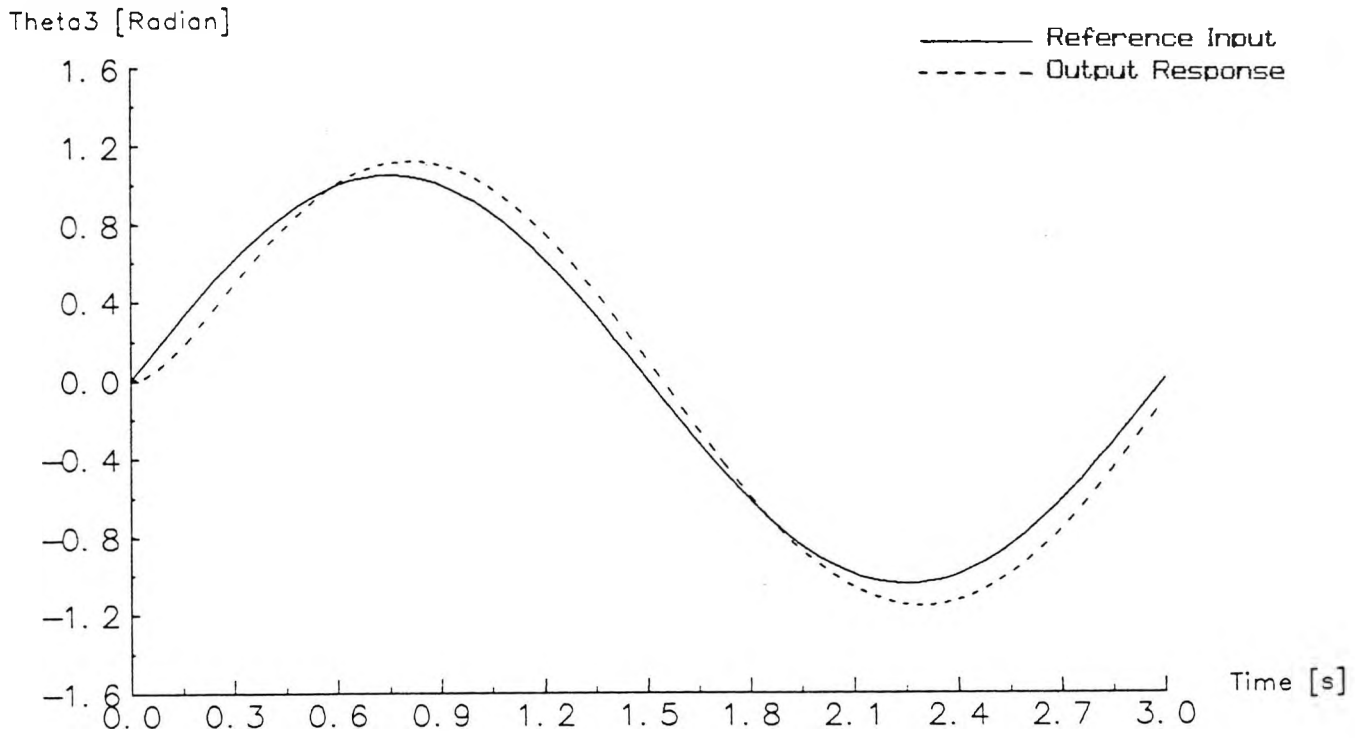
Joint responses using a classical PID control system




(a) Joint 1 response




(b) Joint 2 response




(c) Joint 3 response

Figure 8.7

Classical PID control system with overshoots.

In a practical situation, an average gain is selected to overcome the problem of nonlinearity. This average gain is usually selected with respect to the centre of the work space. For the sake of safety, an overdamped response is preferable to a critical damped or underdamped response.

## 8.4 Conclusions

Simulation studies show that nonlinearities inherent in the robot system cannot be neglected. There are many control techniques around proposed to overcome this problem [Li, 1989a; Fernandez, Bae and Everett, 1990], but the problem is back to the availabilities of expensive high speed computers to carry out the algorithms. Khosla and Kanade [1988] performed a real time evaluation of a model based control scheme. Karam and Warwick [1989] proposed a novel practical microprocessor based controller. To reduce the computational burden in model based control of manipulators, some customised computer architectures have been proposed such as to compute the inverse dynamics [Khosla and Ramos, 1988; Vuskovic, Liang and Anantha, 1990]. Through a simulation study, it is shown that the individual PID technique is not suitable, but for economic reasons, present day manipulators are still controlled with a simple PID control scheme and average gains are chosen in the centre of the work space.

A simulation study also shows that the larger the value of $\sum\lambda^2$ in the model the better the response and that omitting friction effects can cause considerable discrepancy between the plant and model responses. Although a friction effect is relatively less siginificant than the gravity effect, its existence cannot be neglected in modelling and at least an approximation of the friction term must be carried out. In spite of the fact that a partitioned control law is rarely used in

many practical situations due to the need of a second derivative of a trajectory [Craig, 1986] and a high speed computer, this control algorithm is very useful in studying the behaviour of a robot manipulator eg. in the existence of friction.

# CHAPTER 9

# CONCLUSIONS AND SUGGESTIONS FOR FUTURE WORK

## 9.1 Conclusions

The fundamental concepts of mathematical modelling and computer simulation as well as model building have been presented. Principal simulation steps have been included in this study and this lead to the realization of the importance of model validation. The distortion technique to validate a mathematical model quantitatively has been thoroughly reviewed. Both the time domain approach and the frequency domain approach have been outlined. In order to be applicable to a robot system, the frequency domain approach has been extended from a single measured variable case to a multiple measured variable case.

Various aspects concerning an industrial robot have been given. The derivation of a mathematical model of an industrial robot with revolute joints has been studied in detail. The Newton-Euler approach in deriving the robot dynamics equation has been chosen since it leads to a deep understanding of how a robot arm behaves. And also, from the distortion quantitative validation technique point of view, this method is the most suitable.

Implementing the distortion technique to a robot system has been discussed. This covers from selecting important parameters to creating a simulation program. In order to be able to select important parameters which are to be distorted, a study about the importance of inertial parameters has been carried out and this leads to introducing a new term called the fundamental parameters where all inertial

parameters depend on the value of these fundamental parameters. For each link, the fundamental parameters are the mass and the length.

An experiment to obtain recorded transient measurements has been carried out using a TQ MA2000 robot arm which is located in the computer control laboratory. Some modifications in the software were necessary to be done before measurements could be carried out. A ramp step input signal was applied to joint 1, joint 2 and joint 3 as this was the simplest trajectory but was still capable of giving dynamic properties of the robot. Based on this result, the robot mathematical model was validated.

The distortion technique based on the transfer function approach has been used to validate robot models quantitatively since it is more practical over the time domain approach. As the model parameter uncertainties are taken into account in validating the model, the analysis in evaluating the parameter sensitivities has given further insight into the dynamic behaviour of the robot. This has led to an idea that if the trajectory does not involve much acceleration, the interactions due to the inertial torques can be omitted. Although the distortion technique is relatively more objective in validating a model quantitatively, this technique has a weakness ie. in assigning the expected standard deviation of each parameter. With a simplified model where each link is assumed rigid and considered as a line with a uniform distribution of mass, the model with complete friction can produce a reasonably good fit with the exception of link 3 since it is difficult to find its dynamic information due to its highly irregular shape.

Finally, the application of a validated robot model to represent a real system for the control purposes has shown that the friction term is important and hence must

be included in the model. Examining the built in classical PID control scheme which is used by the TQ MA2000 robot arm has indicated that the classical PID technique is not suitable for controlling a robot arm. So, this is a challenge in the future to modify the control algorithm used by the TQ MA2000 robot arm.

## 9.2 Suggestions For Future Work

As mentioned before, although the frequency domain approach is more practical to implement over the time domain approach, this approach suffers from some assumptions, eg. the use of a second order transfer function for representing the discrepancy between model and plant responses. This approximation should be investigated further by applying this technique to many other systems. A more rigorous mathematical derivation would raise the credibility of this approach.

On the other hand, the time domain approach, which is relatively more accurate suffers from the burden in the mathematical difficulty. A simplification of this mathematical problem should be investigated, so this technique can be more easily implemented.

In the second area, the controller board of this robot which accomodates the control algorithm needs a major modification. It needs a more powerful and faster microprocessor as well as more RAM to store the necessary coefficient values. The architecture should allow more summing points, so more advanced control algorithms can be implemented. An analogue power amplifier is preferable rather than a pulse width modulation one since it allows many other aspects of parameter estimation to be done and this may lead to further control research work. All these modifications will lead to a more advanced controller with faster sampling

rate. A faster host computer is also preferable since this will allow more sophisticated trajectories to be used.

In order to do advanced works, the robot needs force/ torque sensors such as strain gauges and a more sophisticated end effector. But the modification in the mechanical part of the robot is less necessary than the modification in the controller.

# REFERENCES AND BIBLIOGRAPHY


**An, C.H., Atkenson, C.G. and Hollerbach, J.M.** (1985)

Estimation of Inertial Parameters of Rigid Body Links of Manipulators.

IEEE International Conference on Robotics and Automation, p. 990-995.

**Armstrong, B.** (1988)

Friction : Determination, Modelling and Compensation.

IEEE International Conference on Robotics and Automation, p. 1422-1427.

**Atkenson, C.G., An, C.H. and Hollerbach, J.M.** (Fall 1986)

Estimation of Inertial Parameters of Manipulator Loads and Links.

The International Journal of Robotics Research, Vol. 5, No. 3, p. 101-119.

**Benson, R.** (1987)

Control of Robot Manipulators.

Robots 11, International Symposium on Industrial Robots, p. 10.31-10.49.

**Butterfield, M.H. and Sutton, T.J.** (June 1979)

Experience in Model Validation on SGHWR During Normal Operation And Following A Trip.

IAEA/NPPCI Specialists Meeting on Control System Comissioning and Dynamic Model Validation, Harrogate, p. 313-327.

**Butterfield, M.H.** (1981)

A Method of Model Verification For Large Perturbation Transients With Examples From Nuclear Power.

IEE International Conference on Control and its Applications, Warwick, p. 196-201.

**Butterfield, MH. and Thomas, P.J.** (March 1983)

Methods of Quantitative Model Validation For Linear and Non-Linear Systems, With Applications to Nuclear Power Plant.

U.K.A.E.A.; AEEW - R 1634.

**Butterfield, MH. and Thomas, P.J.** (October 1986)

Methods of Quantitative Validation For Dynamic Simulation Models, Parts 1 and 2.

Transactions of the Institute of Measurements and Control, Vol. 8, No. 4, p. 182-219.

**Butterfield, MH.** (16 May, 1989)

A Method of Quantitative Validation based on Model Distortion.

One day Symposium on Model Validation,

Organised by The Institute of Measurement and Control.

**Cameron, R.G.** (1989)

Model Validation by the Distortion Method : Linear State Space Systems.

Research Report, R393.

Department of Control Engineering, University of Bradford, Bradford, England.

**Coll, J. and Allen, D.** (1982)

The BBC Microcomputer User Guide.

British Broadcasting Corporation.

**Craig, J.J.** (1986)

Introduction to Robotics - Mechanics and Control.

Addison-Wesley.

**Craig, J.J., Hsu, P. and Sastry, S.S.** (1986)

Adaptive Control of Mechanical Manipulators.

IEEE International Conference on Robotics and Automation, p. 190-195.

**Denavit, J. and Hartenberg, R.S.** (June 1955)

A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices.

Journal of Applied Mechanics, p. 215-221.

**Desloge, E.A.** (1982)

Classical Mechanics, Vol 1.

John Wiley and Sons.

**DiStefano III, J.J., Stubberud, A.R. and Williams, I.J.** (1976)

Feedback and Control System.

McGraw Hill.

**Douce, J.L.** (1963)

An Introduction to the Mathematics of Servomechanisms.

The English Universities Press.

**Escap** (1986/1987)

Product Catalogue, Portescap, Ltd.

**Eykhoff, P.** (1974)

System Identification : Parameter and State Estimation.

John Wiley and Sons.

**Fernandez R, B., Bae, G.W. and Everett, L.J.** (1990)

Control of Robot Manipulator Through Robust Sliding Linearization.

IEEE International Conference on Robotics and Automation, p. 124-129.

**Finkelstein, L. and Watts, R.D.** (1978)

Mathematical Models of Instruments - Fundamental Principles.

Journal of Physics E, Scientific Instruments, Vol. 11, p. 841-855.

**Fu, K.S., Gonzalez, R.C. and Lee, C.S.G.** (1987)

Robotics : Control, Sensing, Vision and Intelligence.

Mc Graw-Hill.

**Gautier, M. and Khalil, W.** (1989)

Identification of the Minimum Inertial Parameters of Robots.

IEEE International Conference on Robotics and Automation, p. 1529-1534

**Gawthrop, P.J., Mirab, L. and Li, X.** (16 May, 1989)

Robot model Validation

One Day Symposium on Model Validation,

Organised by The Institute of Measurement and Control.

**Gogoussis, A. and Donath, M.** (1987)

Coulomb Friction Joint And Drive Effects in Robot Mechanisms.

IEEE International Conference on Robotics and Automation, p. 828-836.

**Gogoussis, A. and Donath, M.** (1988)

Coulomb Friction Effects on the Dynamics of Bearings and Transmissions in Precision Robot Mechanisms.

IEEE International Conference on Robotics and Automation, p. 1440-1446.

**Harrison, T.A. and McCabe, E.M.** (16 May, 1989)

Application of the Distortion Method to Validation of the Code NUMEL for Boiler Dynamics and Control.

One day Symposium on Model Validation,

Organised by The Institute of Measurement and Control.

**Hollerbach, J.M.** (November 1980)

A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity.

IEEE Transactions On Systems, Man, and Cybernatics, Vol. SMC-10, No. 11, p. 730-736.

**Huang, Y. and Lee, C.S.G.** (September 1988)

Generalization of Newton-Euler Formulation of Dynamics Equations to Nonrigid Manipulators.

Journal of Dynamic Systems, Measurement, and Control, Vol. 110, p. 308-315.

**Karam, K.Z., Warwick, K.** (7 November 1989)

A microprocessor based adaptive controller for robotic manipulators.

IEE Colloquium on " Controllers for robotic applications - Concepts and implementation ", p. 4/1-4/4.

Kartowisastro, I.H. (October 1990)

Application of the Distortion Technique to A Two Link Planar Robot Arm.

Research Memorandum, CEC/IHK-98,

Control Engineering Centre, City University, London.

**Khosla, P.K. and Neman, C.P.** (1985)

Computational Requirements of Customized Newton-Euler Algorithms.

Journal of Robotic Systems, 2(3), page. 309-327.

**Khosla, P.K. and Kanade, T.** (February 1988)

Experimental Evaluation of Nonlinear Feedback and Feedforward Control Schemes for manipulators.

The International Journal of Robotics Research, Vol. 7, No. 1, p. 18-28.

**Khosla, K.P. and Ramos, S.** (1988)

A Comparative Analysis of the Hardware Requirements for the Lagrange-Euler and Newton-Euler Dynamics Formulations.

IEEE International Conference on Robotics and Automation, p. 291-296.

**Kubo, T., Anwar, G. and Tomizuka, M.** (1986)

Application of Nonlinear Friction Compensation to Robot Arm Control.

IEEE International Conference on Robotics and Automation, p. 722-727.

**Lee, C.S.G., Gonzalez, R.C. and Fu, K.S.** (1986)

Tutorial on Robotics. 2nd edition.

IEEE Computer Society Press.

**Li, C.J.** (August 1989a)

An Efficient Method for Linearization of Dynamic Models of Robot Manipulators.
IEEE Transaction of Robotics and Automation, Vol 5, No 4, p. 397-402.

**Li, C.J.** (December 1989b)
A New Lagrangian Formulation of Dynamics for Robot Manipulators.
Journal of Dynamic Systems, Measurement, and Control, Vol. 111, p. 559-567.

**Li, C.L.R.** (February 1988)
Methods of Quantitative Model Validation Based on Model Parameter Distortion With Applications to The Nuclear Power Industry.
A Ph.D. Thesis, The City University, London.

**Li, C.L.R.** (16 May, 1989)
Comparison of TIme Domain and Transfer Function Implementation of the Distortion Technique For Quantitative Model Validation.
One day Symposium on Model Validation,
Organised by The Institute of Measurement and Control.

**Luh, J.Y.S., Walker, M.W. and Paul, R.P.C.** (June 1980)
On-Line Computational Scheme for Mechanical Manipulators.
Transactions of ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 102, p. 69-76.

**Luh, J.Y.S.** (May/ June 1983)
Conventional Controller Design for Industrial Robots - A Tutorial.
IEEE Transactions on Systems, Man, And Cybernatics, Vol. SMC-13, No. 3, p. 298-316.

**Markiewicz, B.R.** (March, 1973)

Analysis of the Computed Torque Drive Method and Comparison with the Conventional Position Servo for a Computer Controlled Manipulator.

Technical Memorandum 33.601, Jet Propulsion Laboratory, Pasadena, CA.

**Mayeda, H., Yoshida, K. and Osuka, K.** (1988)

Base Parameters of Manipulator Dynamic Models.

IEEE International Conference on Robotics and Automation, p. 1367-1372.

**Mayeda, H., Yoshida, K. and Ohashi, K.** (1989)

Base Parameters of Dynamic Models For Manipulators With Rotational And Translational Joints.

IEEE International Conference on Robotics and Automation, p. 1523-1528.

**Meyer, W.J.** (1984)

Concepts of Mathematical Modelling.

McGraw-Hill.

**Moon, J.I., Chung, W.K., Cho, H.S. and Gweon, D.G.** (30 Sept 1986 - 2 Oct 1986)

A Dynamic Parameter Identification Method for the Puma-760 Robot.

Proceedings 16th International Symposium on Industrial Robots, 8th International Conference on Industrial Robot Technology, Brussels, Belgium, p. 55-65.

**Murray, J.J.** (10 Septenber, 1986)

Computational Robot Dynamics.

A Ph.D. Dissertation, Carnegie Mellon University.

**Neelamkavil, F.** (1987)

Computer Simulation And Modelling.

John Wiley and Sons.

**Neuman, C.P. and Khosla, P.K.** (1985)

Identification of Robot Dynamics : An Application of Recursive Estimation.

Advances in Adaptive Systems Theory, Narendra, K.S., ed.

Plenum Publishing Corporation, New York, p. 175-194.

**Neuman, C.P. and Khosla, P.K.** (1985)

Identification of Robot Dynamics : An Application of Recursive Estimation.

Proceedings 4th Yale Workshop on Applications of Adaptive Systems Theory, p. 42-49.

**Ogata, K.** (1981)

Modern Control Engineering.

Prentice Hall of India.

**Olsen, H.B. and Bekey, G.A.** (1985)

Identification of Parameters In models of Robots With Rotary Joints.

IEEE International Conference on Robotics and Automation, p. 1045-1049.

**Paul, R.P.C.** (August, 1972)

Modelling, Trajectory Calculation And Servoing of a Computer Controlled Arm.

A Ph.D. Dissertation, Stanford University.

Paul, R.P. (1981)

Robot Manipulators : Mathematics, Programming and Control.

The MIT Press.

**Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T.** (1988)

Numerical Recipes, The Art of Scientific Computing.

Cambridge University Press.

**Rasmussen, H.** (7 November 1989)

A single board computer for robot control.

IEE Colloquium on " Controllers for robotic applications - Concepts and implementation ", p. 5/1-5/4.

**Roberts, P.D., Leal, D.J. and Georgantzis, G.** (1979)

Parameter Estimation in Uncertain Mathematical Models of Industrial Plant.

Control System Commissioning and Dynamic Model Validation Specialists Meeting, Harrogate.

London : Central Electricity Generating Board, p. 255-267.

**Roberts, P.D.** (1980)

Verification of Mathematical Dynamic Models,

Final Report DSS/PDR/213,

Department of Systems Science,

City University (England).

**Sarkar, A.D.** (1980)

Friction and Wear.

Academic Press.

**Scott, P.B. and Husband, T.M.** (1985)

The Robotics Revolution : The complete Guide For Managers And Engineers.

Basil Blackwell.

**Shannon, R.E.** (1975)

Systems Simulation, The Art And Science.

Prentice-Hall, Inc.

**Sheu, S.Y. and Walker, M.W.** (1989)

Basis Sets for Manipulator Inertial Parameters.

IEEE International Conference on Robotics and Automation, p. 1517-1522.

**TecQuipment.** (1984)

TQ MA2000 User's Manual.

**TecQuipment.** (1986)

TQ MA2000 Hardware and Software Manual.

**Tustin, A.** (1947)

The Effects of Backlash and of Speed-Dependent Friction on the Stability of Closed-Cycle Control Systems.

Journal of the Institution of Electrical Engineers, p. 143-151.

**Vuskovic, M., Liang, T. and Anantha, K.** (1990)

Decoupled Parallel Recursive Newton-Euler Algorithm for Inverse Dynamics.

IEEE International Conference on Robotics and Automation, p. 832-838.

**Walker, M.W. and Orin, D.E.** (September 1982)

Efficient Dynamic Computer Simulation of Robotic Mechanisms.

Journal of Dynamic Systems, Measurement and Control, Vol. 104, p. 141-147.

**West, H., Papadopoulos, E., Dubowsky, S. and Cheah, H.** (1989)

A Method For Estimating the Mass Properties of A Manipulator by Measuring the Reaction Moments At Its Base.

IEEE International Conference on Robotics and Automation, p. 1510-1516.

# APPENDIX A

For a rigid body which is free to move in three dimensions, there are an infinite number of possible rotation axes. A complete way of characterising the mass distribution of a rigid body is the inertia tensor, which can be thought of as a generalisation of the scalar moment of inertia of an object.

The inertia tensor matrix relative to a coordinate system i may be expressed in the matrix form as the 3x3 symmetric matrix.

$$ {}^{i}\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} $$

where

$$ I_{xx} = \iiint_V ( y^2 + z^2 ) \rho \, dv $$

$$ I_{yy} = \iiint_V ( x^2 + z^2 ) \rho \, dv $$

$$ I_{zz} = \iiint_V ( x^2 + y^2 ) \rho \, dv $$

$$ I_{xy} = \iiint_V xy \, \rho \, dv $$

$$ I_{xz} = \iiint_V xz \, \rho \, dv $$

$$ I_{yz} = \iiint_V yz \, \rho \, dv $$

and the rigid body is composed of differential volume elements, dv, with material density, $\rho$, and located with a vector $( x', y', z' )^T$ with respect to a coordinate system i.

## APPENDIX B

The recursive Newton-Euler equations of motion in this appendix is a modification of the existing algorithm [Luh, Walker and Paul, 1980] where in the backward iteration, the inertia matrix of link i is evaluated about joint i and expressed in the coordinate system i.

Forward iteration :

$$ {}^{i}\omega_i = {}^{i}R_{i-1} \left( {}^{i-1}\omega_{i-1} + z_0 \dot{q}_i \right) $$

$$ {}^{i}\dot{\omega}_i = {}^{i}R_{i-1} \left( {}^{i-1}\dot{\omega}_{i-1} + z_0 \ddot{q}_i + {}^{i-1}\omega_{i-1} \times z_0 \dot{q}_i \right) $$

$$ {}^{i}\dot{V}_i = {}^{i}\dot{\omega}_i \times {}^{i}p_i + {}^{i}\omega_i \times \left( {}^{i}\omega_i \times {}^{i}p_i \right) + {}^{i}R_{i-1} \left( {}^{i-1}\dot{V}_{i-1} \right) $$

$$ {}^{i}\bar{a}_i = {}^{i}\dot{V}_{i-1} + {}^{i}\dot{\omega}_i \times {}^{i}c_i + {}^{i}\omega_i \times \left( {}^{i}\omega_i \times {}^{i}c_i \right) $$

Backward iteration :

$$ {}^{i}F_i = m_i \, {}^{i}\bar{a}_i $$

$$ {}^{i}nn_i = m_i \, {}^{i}c_i \times {}^{i}\dot{V}_{i-1} + {}^{i}I_i^{*} \, {}^{i}\dot{\omega}_i + {}^{i}\omega_i \times \left( {}^{i}I_i^{*} \, {}^{i}\omega_i \right) $$

$$ {}^{i}f_i = {}^{i}R_{i+1} \, {}^{i+1}f_{i+1} + {}^{i}F_i $$

$$ {}^{i}n_i = {}^{i}nn_i + {}^{i}R_{i+1} \, {}^{i+1}n_{i+1} + {}^{i}p_i \times \left( {}^{i}R_{i+1} \, {}^{i+1}f_{i+1} \right) $$

where :

${}^{i}R_{i+1}$ = a 3x3 rotation matrix which transforms any vector in the coordinate system i+1 to the coordinate system i.

$q_i$ = i-th joint variable.

${}^{i}p_i$ = position of the origin of coordinate system i from the origin of coordinate system i-1and expressed in the coordinate system i.

${}^{i}c_i$ = position of the centre of mass of link i from the origin of coordinate system i and expressed in the coordinate system i.

${}^{i}\omega_i$ = angular velocity of link i expressed in the coordinate system i.

${}^{i}\dot{\omega}_i$ = angular acceleration of link i expressed in the coordinate system i.

${}^{i}\dot{v}_{i-1}$ = linear acceleration of the origin of coordinate system i-1 and expressed in the coordinate system i.

${}^{i}I_i^*$ = inertia matrix of link i about joint i and expressed in the coordinate system i.

$m_i$ = mass of link i.

${}^{i}F_i$ = force at the centre of mass of link i and expressed in the coordinate system i.

${}^{i}f_i$ = force at joint i expressed in the coordinate system i.

${}^{i}nn_i$ = moment at joint i due to motion of link i alone expressed in the coordinate system i.

${}^{i}n_i$ = moment at joint i due to motions of link i and the distal links, and expressed in the coordinate system i.

## APPENDIX C

## TECHNICAL SPECIFICATION OF TQ MA2000 ROBOT ARM

**Repeatability** : ± 2 mm.

**Accuracy** : ± 3 mm.

**Power Transmission** :

**Waist** : Motor - Maxon 2332-908-11-151-000
Driving through 100:1 gearbox
and 60:10 pulleys.

**Shoulder** : Motor - Escap 34L11-224E-5
Driving through 123:1 gearbox
and 62:12 pulleys.

**Elbow** : Motor - Maxon 2332-908-11-151-000
Driving through 100:1 gearbox
and 40:12 pulleys.

**Gain of power amplifiers of joint 1, 2 and 3** : 28

**Joint position tranducer** : Plastic film potentiometer with linearity of ± 0.25 %
(main axes only).

| | | MAXON | ESCAP |
|---|---|---|---|
| Power output | Watt | 11 | 12 |
| Nominal voltage | Volt | 18 | 15 |
| No load current | mA | 43.3 | 20 |
| Terminal resistance | Ohm | 5.49 | 4.5 |
| Rotor inductance | mH | 0.82 | 0.6 |
| Torque constant | mNm/A | 20.8 | 33 |
| Rotor inertia | $kgm^2.10^{-7}$ | 18.3 | 32 |
| Back EMF constant | V/1000 rpm | Not available | 3.5 |
| Mass | g | 174 | 230 |
| Viscous damping constant | $Nms/rad.10^{-6}$ | Not available | 1 |