Variable Structure Control of Robot Manipulators (The Example of the SPRINTA)

A Thesis submitted for the degree of Doctor of Philosophy

Pierre Nigrowsky

Department of Electronic and Computer Engineering, Brunel University

December, 1999

For the attention of candidates who have completed Part A

i) Attention is drawn to the fact that the copyright of a thesis rests with its author.

ii) A copy of a candidate's thesis is supplied to the Library on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the prior written consent of the author or the University, as appropriate.

Requests for such permission should be addressed in the first instance to the Head of Library Services.

Abstract

The subject of this thesis is the design and practical application of a model-based controller with variable structure control (VSC).

Robot manipulators are highly non-linear systems, however they form a specific class in the non-linear group. Exact mathematical descriptions of the robot dynamics can be achieved and further, robot manipulators have specific useful properties that can be used for the design of advanced controllers. The inclusion of the inverse dynamic description of the robot manipulator as a feedforward term of the controller (model-based controller) is used to transform two non-linear systems i.e. the controller and the robot, into one linear system. The limitation of this technique arises from the accuracy of the inverse dynamic model. The linearisation only takes place if the model is known exactly. To deal with the uncertainties that arise in the model, a control methodology based on variable structure control is proposed.

The design of the controller is based on a Lyapunov approach and engineering considerations of the robot. A candidate Lyapunov function of a pseudo-energy form is selected to start the controller design. The general form of the controller is selected to satisfy the negative definiteness of the Lyapunov function. The initial uncertainties between the actual robot dynamics and the model used in the controller are dealt with using a classical VSC regulator. The deficiencies of this approach are evident however because of the chattering phenomenum. The model uncertainties are examined from an engineering point of view and adjustable bounds are then devised for the VSC regulator, and simulations confirm a reduction in the chattering.

Implementation on the SPRINTA robot reveals further limitations in the proposed methodology and the bound adjustment is enhanced to take into account the position of the robot and the tracking errors. Two controllers based on the same principle are then obtained and their performances are compared to a PID controller, for three types of trajectory. Tests reveal the superiority of the devised control methodology over the classic PID controller. The devised controller demonstrates that the inclusion of the robot dynamics and properties in the controller design with adequate engineering considerations lead to improved robot responses.

Memorandum

This thesis is based on work carried out by the author in the Department of Electronic and Computer Engineering at Brunel University between October 1996 and September 1999.

All work and ideas in this thesis are original unless otherwise acknoledged in the text and by references. The work has not been submitted for another Degree in this University, nor in any other University.

The main contributions of this thesis include :

1. The proposal of a design methodology for controller for robot manipulators, based on the system properties and on engineering considerations.

2. The proposal and assessment of bound variations for variable structure controllers as to reduce the chattering phenomenum and the stress put on the system.

3. The application and assessment of a control methodology based on the two above principles to the SPRINTA robot.

Toute vérité jouit d'un moment de gloire entre deux infinis, lorqu'elle est complètement fausse et lorqu'elle est considérée comme évidente.

Every truth enjoys one moment of triumph between one infinity, when it is deemed false, and another infinity, when it is considered trivial.

Henry Poincaré

Acknowledgements

It is my pleasure to thank the many people who enabled me to perform the work presented in this thesis.

First, I want to thank my parents and family for their encouragement and unconditional support during many years.

I owe many thanks to Dr. Peter Turner, my supervisor for his constructive suggestion and guidance. I would like to thank Dr. John Marsh for the fruitful discussions on control and real-life problems.

Special thanks are in order for John Phillips, Clive Barret, Philip Cooke and John Newton for helping me in various aspects of my work.

My colleagues at the Control Systems Research Group, and in the Department have created a very pleasant atmosphere to work.

I greatly appreciate the financial support by the EPSRC and the Department of Electronic and Computer Engineering of Brunel University.

Finally, I would like to thank T. for having been so patient during all these years.

Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Background and Motivation

In the past twenty years, the field of robotics has stimulated an increasing interest from industrial and academic researchers. The general idea of the early 1980's was that robotics was no longer science fiction and very soon robots "would do wonderful things" for industry, the home, space exploration, and other areas. Unfortunately, robotics has not met these early expectations [27]. One explanation could be the significant difference between what is readily available in the laboratory and what is commercially available. Another explanation could be the general misconception about robotics and robot capabilities from films like *Star Wars*, which did very little to establish a realistic picture of robotics.

Although robotics can be found in different environments for various applications (medicine, space, underwater, etc. ...), they are most commonly found in an industrial environment, where their role is to perform a task in a repetitive manner such as welding, painting or assembling parts. Those robots which are far from human like, are referred to as industrial robots. They are made of four major components : the manipulator, the power conversion unit, the sensory devices and the computer control system.

Previously robot systems were designed in an *ad-hoc* fashion, the various components were developed separately without much interaction. The continuous need for better performance (speed and accuracy) of robot systems could no longer be fulfilled with such an approach. Robotics as a specific field was consequently established and a new way of thinking was required. Its interdisciplinary subjects: cybernetics, mechanics, bioengineering, electronics, information science, and automatic control science to mention the most important ones, demand robot engineers have a cross disciplinary skill to make the best of each discipline and foresee the possible benefit for robotics of the development in a particular field.

This thesis intends to address the performance improvement that can be obtained by robot manipulators from the control point of view. It emphasises the concept of a control methodology that considers the mechatronic nature of the system. This

provides some new insight for the control of robot manipulators and further supports an integrated approach for the design of robot systems.

1.2 Problem Formulation

This thesis concentrates on the control of a specific class of robot manipulators, that is the class of rigid robot systems with transmission directness. The rigid class refers to robots for which the links do not have flexibility, or for which the flexibility is negligible. The transmission directness concept means that the transmission between the motor and the joint of the robot does not involve a gearbox element, this idea is further defined in the next chapter. During the last 15 years, the control of such systems has been an extensive subject of interest. The requirement for greater accuracy and faster performance have lead robot engineers to look for new design approaches for mechatronic systems that overcome the limitation of the traditional approaches. The full potential of such new systems could however only be achieved by reconsidering the control methodology. The non-linear nature of robotic systems and the varying requirement imposed by the tasks they are required to do, is another reason for reconsidering the traditional control method. The development of control theory as well as a clearer understanding of the mechanical properties of robot manipulators have brought new perspectives for tackling the control problem.

From a general point of view, robot controllers may be classified into two categories, non-model based and model based. The first approach refers to the type of control technique which does not use any knowledge of the system in the formulation of the control algorithm. The second approach, on the other hand uses some of the mathematical descriptions and properties of the system for the generation of its control laws. The non-model based approach has been used in the early days of robotics because of the ease of implementation and relatively good performance. Nevertheless demand for better performance has pushed engineers and scientists to develop new control algorithms. Robot manipulators are highly non-linear systems and the particular inherent properties of these types of system mean that they form a special class within the non-linear family. As suggested by Slotine [82], a close analysis of these properties leads to more efficient and reliable controllers. Most

of the recent controllers systematically use the system properties either during the design phase or for their stability analysis. These type of controllers form the model based approach.

The controller design approach suggested in this thesis intends to reveal how the robot performance can be enhanced by considering the physical nature of the robot in the design phase.

1.3 Organisation of the Thesis

In Chapter 2 the dynamic equations describing a general robot system are introduced, and some fundamental properties of these dynamics are explained. Also alternative methods to the general geared transmission techniques are presented. The concept of the transmission of the SPRINTA is explained with its general geometry. Chapter 3 provides an overview of the various control methodologies for robot manipulators, and highlights of the specificities of each approach are presented. The control for the SPRINTA is addressed and a general direction for the design of its controller is given. The concept of Variable Structure Control with sliding mode is explained in Chapter 4. The application of the technique to robot manipulators is exposed, in line with its development.

A controller design methodology for robot manipulator based on VSC and sliding mode is presented in Chapter 5. The designed is established in a step by step way and is supported by simulations. The application to the SPRINTA of the controller with the necessary modifications are discussed in Chapter 6. The results obtained with the designed controllers for three different trajectories are shown and are compared with the results obtained with a classical PID. Finally, in Chapter 7 conclusions and suggestions for future research are given.

2 ROBOT MANIPULATORS

2.1 Robot Dynamics and Properties

The model based approach to controller design systematically uses the equations describing the system dynamics. The most commonly used description is the inverse dynamic equations. These equations give the torque required to achieve a given trajectory of the robot manipulator. The two most commonly used techniques to derive the inverse dynamics are termed Newton-Euler and Euler-Lagrange. Both methods lead to the same result, but they tackle the dynamic formulation problem from a different point of view.

The Newton-Euler formulation is based on two set of equations. The first set, referred to as outward recursion, propagates the kinematic information from the base of the manipulator to its end point. The second set of equations, inward recursion, determines the forces and moments acting on each link. They are computed from the end point down to the initial link using the kinematic information obtained from the outward iteration. This method is computationally efficient, and very suitable for simulation [101], [22]. A good understanding of the forces between the links including the direction in which they act is required however to apply this technique.

The Euler-Lagrange method is more general than the Newton-Euler technique and it is based on the energy in the system. The approach leads directly to a compact formulation of the inverse dynamics. In addition, very little knowledge about the system is required. A shortcoming of this approach is that a large number of algebraic operations is required, which makes it computationally inefficient. The introduction of friction acting between the links is also not so easy as in the Newton-Euler method.

The torque for each link T_i, is derived using equation (1).

$$T_i = \frac{d}{dt}\left\{\frac{\partial L}{\partial \dot{\theta}_i}\right\} - \frac{\partial L}{\partial \theta_i},\ i = 1, ..., n \tag{1}$$

where L is the Lagrangian function defined as the difference between the to-

tal Kinetic ($Ktotal$) and Potential energy ($Ptotal$), θ_i and $\dot{\theta}_i$ are respectively the position and velocity of the joint of the ith link.

$$L = Ktotal - Ptotal \tag{2}$$

$$Ktotal = \sum_{i=1}^{n} \frac{1}{2} M_i \dot{\theta}_i^T Jl^{i,com^T} Jl^{i,com} \dot{\theta}_i + \frac{1}{2} \dot{\theta}_i^T Ja^{i,com^T} R_i I_i R_i^T Ja^{i,com} \dot{\theta}_i \tag{3}$$

$$Ptotal = \sum_{i=1}^{n} M_i Gr^T P_i^{com} \tag{4}$$

where :

$Gr = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$, g is the gravitational acceleration ($9.81 m.s^{-2}$),

M_i is the mass of link i,

I_i is the inertia matrix of link i at the Center Of Mass (C.O.M.),

R_i is the rotation matrix of link i with reference to the base coordinate frame,

$Jl^{i,com}$ and $Ja^{i,com}$ are the Jacobian matrices for the linear and angular velocity of the C.O.M., and

P_i^{com} is the position vector of the C.O.M. of the link i with reference to the base coordinate frame.

The derivation of the Lagrange formulation leads to a compact form of the inverse dynamics, equation (5). Symbolic calculation of the Newton-Euler set of equations followed by some algebraic manipulations would give the same compact formulation of the inverse dynamics [7].

$$T = M(\theta)\ \theta + C(\theta, \theta)\ \theta + G(\theta) \tag{5}$$

where :

$M(\theta)$ is the manipulator inertia matrix,

$C(\theta, \dot{\theta})$ represents the Coriolis and centripetal effects,

$G(\theta)$ is the gravitational term,

$\ddot{\theta}$, $\dot{\theta}$ and θ are respectively column vectors of the angular acceleration, velocity and position of the joints.

Other approaches have been used to derive the inverse dynamics of the system, such as the D'Alembert [53], Hamiltonian [62], or Kane [45] methods. Although these methods start from different perspectives, their respective algebraic inverse dynamic representation can always be reformulated into the general form, equation (5) . This general representation of the inverse dynamics is very useful as it is usually the starting point for the design of modern controller algorithms.

To derive the equation of the inverse dynamics via the Newton-Euler, Euler-Lagrange, or any other method, the system has to be entirely defined in terms of positions and orientations. For this, a widely used convention is the Denavit-Hartenberg [39] representation (see APPENDIX A). The position and orientation of each link is defined by a frame attached to it. The overall location of the manipulator is defined by successively mapping the orientation and location from the end point frame down to the reference frame.

As mentioned earlier, the equations describing the robot manipulators, are highly non linear. Articulated mechanics have some common inherent properties however and a careful analysis, followed by a judicious use of these properties, leads to very effective controllers, as suggested by Slotine [82]. The five most commonly used properties for manipulators are : the symmetric and positive definiteness of the inertia matrix, the boundedness of the dynamical terms, the skew-symmetry, the parametrization and the passivity of the system.

2.1.1 Symmetric Positive Definite Inertia Matrix

The manipulator inertia matrix $M(\theta)$ is always symmetric and positive definite, and is also the case for the inertia tensor of each link [7]. The positive definiteness implies that the quadratic form of the manipulator inertia is always a positive scalar. Physically, it means that the kinetic energy is always positive for non-zero velocity.

$$K = 0.5 \cdot (\dot{\theta}^T M(\theta)\, \dot{\theta}) > 0 \text{ ; for } \dot{\theta} \neq 0 \qquad (6)$$

2.1.2 Boundedness Property

The dynamic terms $M(\theta)$, $C(\theta,\dot{\theta})$ and $G(\theta)$ in the inverse dynamic formulation vary with respect to the joint angles and their first derivatives. Since the range of variation for the joint angles is physically limited, the inertia, Coriolis/centripetal and gravitational matrices are bounded with respect to θ. The same argument holds for prismatic joints. In the case of the SPRINTA robot, which uses a special type of transmission element, such property is also verified.

$$\begin{aligned} \beta_1 \cdot I_n &\leq M(\theta) \leq \beta_2 \cdot I_n && (7) \\ 0 &\leq \left\| C(\theta,\dot{\theta})\,\dot{\theta} \right\|_2 \leq F(\theta,\dot{\theta}) && (8) \\ 0 &\leq G(\theta) \leq G_1 && (9) \end{aligned}$$

where :

β_1 and β_2 are positive constants and I_n is an $(n \times n)$ identity matrix,

$F(\theta,\dot{\theta})$ is a diagonal matrix of known scalar functions,

$\|...\|_2$ denotes the euclidean norm, and

G_1 is a diagonal matrix of positive scalars.

The definition $\beta_1 \cdot I_n \leq M(\theta)$ means that $(M(\theta) - \beta_1 \cdot I_n)$ is positive semidefinite, and similarly $M(\theta) \leq \beta_2 \cdot I_n$ means that $(M(\theta) - \beta_2 \cdot I_n)$ is negative semidefinite. The boundedness property of the inertia matrix can be also expressed as

$$\beta_1 \cdot I_n \leq \|M(\theta)\|_2 \leq \beta_2 \cdot I_n \qquad (10)$$

2.1.3 Skew-symmetry

The vector $C(\theta,\theta)\ \theta$ representing the centripetal and Coriolis effects has a unique definition. The definition of the matrix $C(\theta,\dot{\theta})$ is however not unique, using the Christoffel coefficients [55], the ith element for the centripetal and Coriolis vector may be written as :

$$\sum_{j=1}^{n} C_{ij}\,\dot{\theta}_j = \sum_{j=1}^{n}\sum_{k=1}^{n} h_{ijk}\,\dot{\theta}_j\dot{\theta}_k \tag{11}$$

with $h_{ijk} = \dfrac{\partial M_{ij}}{\partial \theta_k} - \dfrac{1}{2}\dfrac{\partial M_k}{\partial \theta_i}$

The ith term for the Coriolis and centripetal effect may be rewritten as [55]:

$$\sum_{j=1}^{n} C_{ij}\,\dot{\theta}_j = \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial M_{ij}}{\partial \theta_k}\,\dot{\theta}_j\dot{\theta}_k + \frac{1}{2}\sum_{k=1}^{n}\sum_{j=1}^{n}\left(\frac{\partial M_{ik}}{\partial \theta_j} - \frac{1}{2}\frac{\partial M_{jk}}{\partial \theta_i}\right)\dot{\theta}_k\dot{\theta}_j \tag{12}$$

Then :

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{n} \frac{\partial M_{ij}}{\partial \theta_k}\,\dot{\theta}_k + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{ik}}{\partial \theta_j} - \frac{1}{2}\frac{\partial M_{jk}}{\partial \theta_i}\right)\dot{\theta}_k \tag{13}$$

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{n} \dot{M}_{ij} + \frac{1}{2}\sum_{k=1}^{n}\left(\frac{\partial M_{ik}}{\partial \theta_j} - \frac{1}{2}\frac{\partial M_{jk}}{\partial \theta_i}\right)\dot{\theta}_k \tag{14}$$

Let :

$$W_{ij} = \sum_{k=1}^{n}\left(\frac{\partial M_{ik}}{\partial \theta_j} - \frac{1}{2}\frac{\partial M_{jk}}{\partial \theta_i}\right)\dot{\theta}_k \tag{15}$$

Then for all i, j and $W_{ij} = -W_{ji}$, the matrix $W = \dot{M} - 2C$ is skew-symmetric. The skew-symmetry gives the following property :

$$x^T\left[\dot{M}(\theta) - 2C(\theta,\dot{\theta})\right]x = 0, \text{ for all } x \in R^n. \tag{16}$$

The property in equation (16) is simply a statement that the forces, defined by $C(\theta,\dot{\theta})\,\dot{\theta}$, do no work on the system [69]. An important consequence of this is the following property.

2.1.4 Passive Nature of Manipulator

The passivity concept originated from network theory; a system is said to be passive if it does not create energy. A system is passive if for equal number of inputs $U(t)$

and outputs $Y(t)$, it satisfies the following inequality :

$$\int_0^t Y(\tau)^T U(\tau)\, d\tau \geq \alpha; \text{ for all } t > 0 \text{ and } \alpha > -\infty. \tag{17}$$

For a robot manipulator, considering the input as the torque vector and the output as the vector of joint velocities, the system defines a passive mapping from the input to the output. Using the Hamiltonian term H defined as the sum of the kinetic energy and the potential energy of the manipulator [69], the following relation holds

$$\frac{dH}{dt} = \dot{\theta} \,.T, \tag{18}$$

where :

$\dot{\theta}$ is the vector of joint velocities,

T is the input torque vector, and

H is the Hamiltonian sum of the kinetic and potential energy.

Then

$$\int_0^t \dot{\theta}(\tau)^T T(\tau)\, d\tau = H(t) - H(0) \geq -H(0) \tag{19}$$

This proves the passivity property of the manipulator. This last property has been unconsciously used for some time in control engineering. By closing the loop between the joint velocity and the torque, with a passive system, the whole system remains passive. Since the new system is dissipative, it is stable. Recently the passivity property has been used considerably as a systematic design philosophy for robot controllers [68].

2.1.5 Linearity in Parameters

By a suitable reformulation of the dynamic equations, the constant parameters defining the dynamics of the system, i.e. link masses, moments of inertia, etc... may appear as coefficients of a general co-ordinate function. By defining each coefficient as a separate parameter, a linear relationship results, so that the dynamic equations can be written as equation (20). Since all these constant parameters are usually

subject to inaccuracies, this linear formulation of the dynamic equation is very useful for adaptive control. This reformulation that separates the unknown or partially unknown parameters and the known time functions, is used for the formulation of the adaptive update rules [55].

$$T = M(\theta)\,\ddot{\theta} + C(\theta,\dot{\theta})\,\dot{\theta} + G(\theta) = Y(\theta,\dot{\theta},\ddot{\theta})\delta \tag{20}$$

where :

$Y(\theta,\dot{\theta},\ddot{\theta})$ is an $(n \times n)$ matrix of known functions, called regressor functions, and

δ is an $(n \times 1)$ vector containing the constant parameters.

2.2 Direct Drive and Associated Techniques

Traditional robotic arms use gearboxes as transmission elements between the actuators and the joints, which means that the load seen by the motor is reduced and the motor can be relocated from the arm. The drawback of this method is the presence of friction and either backlash or compliance in the gearbox. As robotic applications in the manufacturing industries demand ever increasing accuracy and speed of manipulations, the limitations imposed by the use of gearboxes become unacceptable. The development of low speed and high torque motors [102], [70], means that there is no longer a need for a gearbox that magnifies the torque and reduces the speed, between the motor and the joint. A design approach, called Direct Drive (DD), which is based on the use of these special motors has been proposed some 15 years ago [9] and [25]. In this approach, the shafts of the motors are directly coupled to the joints of the manipulators. Friction, backlash and compliance associated with the gearboxes are then eliminated. The backlash and friction problems associated with traditional gearboxes, have also been addressed by new type of gearboxes such as harmonic drives and cyclo drives. The CMU DD Arm model I [5] and the MIT I serial DDarm [3] are robots designed on the direct drive concept. In its strict form, direct drive presents serious limitations due to the motor shaft being directly connected to the joint so that it must be mounted on the link thereby reducing the

payload. To avoid this extra loading on the link, a transmission element is introduced that keeps as much directness as possible while allowing motor relocation. Robotic arms using this technique diverge from pure direct drive as a transmission element is allowed to be introduced between the shaft of the motor and the joint to actuate. This transmission element provides the maximum relocation freedom whilst introducing the minimum of detrimental effects associated with conventional transmissions. No distinction is however made between this technique and direct drive and it is commonly referred to as direct drive. Vines [100] proposed to distinguish the two and refers to those that use some form of transmission as Direct Transmission (DT).

Robot manipulators using direct transmissions still need torque motors, as the directness of the transmission is kept. A number of robot manipulators using this technique have been reported, the MIT DD model II, III and IV [9], [8], the Minnesota DDarm [47], the Adept-One [2] and the DD SCARA [54]. The three MIT models as well as the Minnesota DDarm use linkage mechanisms as transmission elements, to relocate the motors from the links and to keep as much directness as possible. The Adept-One and the DD SCARA involve a system of pulleys and steel bands as direct transmission elements.

The Direct Transmission philosophy has been proposed to achieve a greater level of actuator relocation freedom than is possible under the direct drive concept, but without introducing the majority of the negative aspects of the conventional transmissions. The enhanced relocation allows reduced actuator mobility, suggesting higher performance than for direct drive.

2.3 Non-Linear Direct Transmission

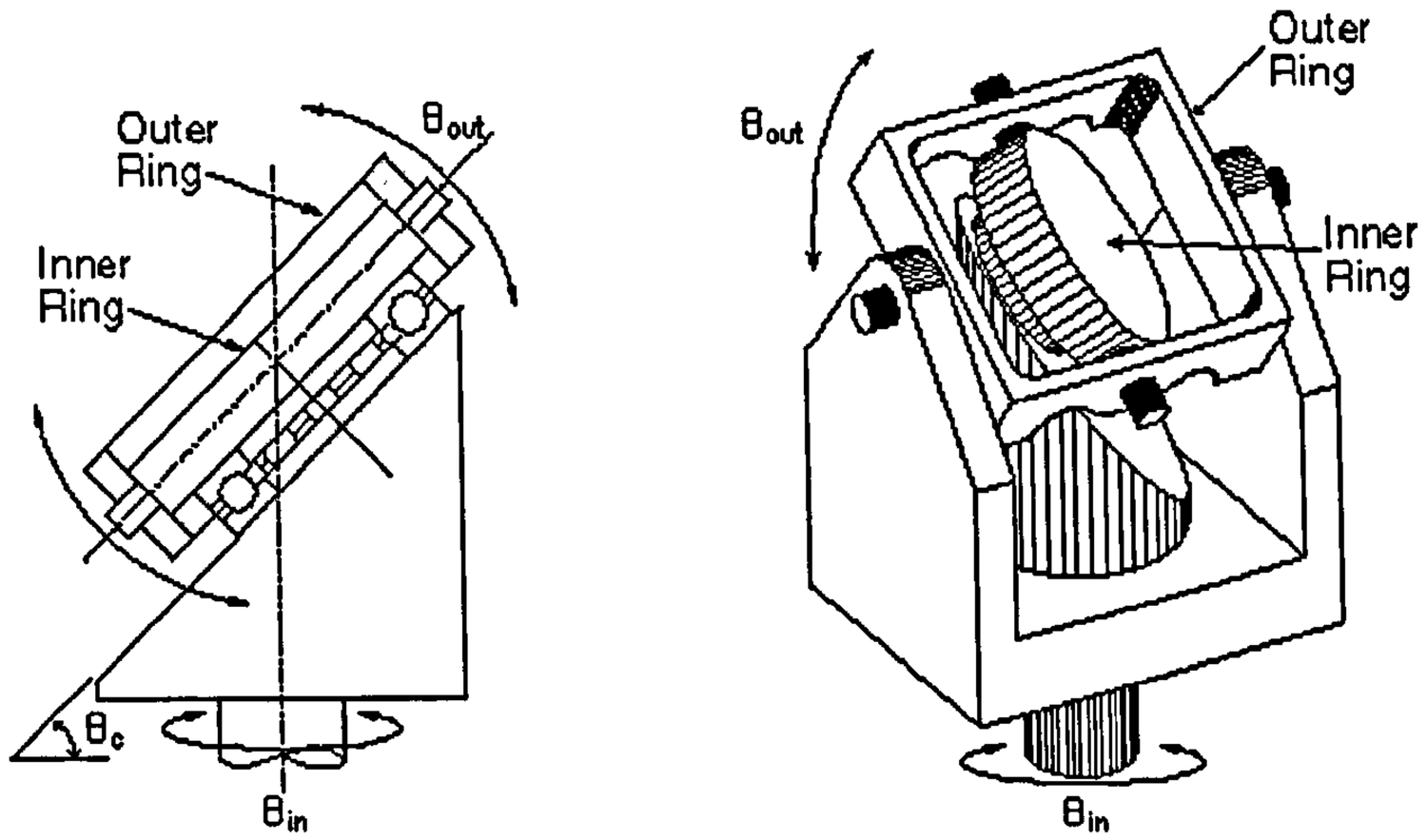
The direct drive and direct transmission techniques have some serious consequences on the control method. Since the reduction effect of the gearbox no longer exists, dynamic complexity, such as coupling and non-linearities are directly reflected at the motor shafts. Further, higher resolution feedback sensors are needed because of the absence of a reduction ratio. The SISO technique or independent joint control which is commonly used for traditional manipulators, is not appropriate anymore

to deal with the high level of dynamic complexity reflected at the motor shafts [8]. More advanced methods such as model based [3], [15] or computed torque control [4] are needed. Another method to deal with dynamic complexity, is to re-design the manipulator in such a way that the dynamic complexity is kept to a minimum. This is done by redistributing the masses and reformulating the structure of the manipulator. Takase et al [90] reduced the gravitational torque by designing an arm with a counterweight but this increases the inertia of the arm, and thus reduces the acceleration potential. The Minnesota direct drive robot [47] uses a four-bar linkage with a specific design structure so that the functional moving parts are balanced. This design arrangement eliminates gravitation torque thus requiring smaller actuators and amplifiers and a simpler controller. This balancing is only valid however for robots having the same specific geometrical structure.

Until now, the complexity introduced by direct drive and direct transmission has been tackled by using more complex controller algorithms or re-designing the mechanics of the manipulator. Vines [100], [93] proposed a new design, that combined the advantages of the direct transmission philosophy with the reduction ratio property of gearboxes. This design is based on the use of a specific type of transmission, referred to as non-linear direct transmission. The non-linear direct transmission keeps the directness of direct drive whilst providing a varying reduction ratio. Such transmissions provide all the advantages of the direct transmission, i.e. no friction, backlash and compliance whilst providing motor relocation. The varying reduction ratio can also allow some compensation for the variation in joint requirements. Depending on the task, the torque and the speed, as well as the accuracy requirements varies within the workspace. The dynamic complexity which dominates in direct drive and direct transmission, can also be attenuated by providing a reduction ratio. For example when the manipulator is in a position where the gravitational torque is significant, the non-linear transmission can be designed such that the load on the motor is attenuated.

A number of mechanical devices satisfy the criteria of a non-linear transmission topology, for example a non-parallel linkage [39], the Peaucellier linkages [39, p181], bands with elliptical pulleys [72]. Vines [100],[93] proposes the Gimbal Drive shown

in Figure 1 , which is an extension of the sinusoidal reciprocator [20, p83], as a non-linear transmission element suitable for robot manipulators.



a: 2D schematic b: Simple 3D model

Figure 1: The Gimbal Drive Transmission

The gimbal drive consists of a truncated cylinder, a rocking shaft (outer ring) and an inner ring. The input of the system is given by rotating the truncated cylinder and the output is obtained by the swing of the outer ring. With the inner ring restrained to rotate, the rotation of the truncated cylinder causes the inner ring to follow the oblique surface of the truncated cylinder. The restraining points of the inner cylinder are forced to have antagonistic movements, one is moving up while the other moves down. Their displacements result in the swing of the outer ring. The gimbal drive transmission has the following input-output function (equ. (21) and fig. 2).

$$\theta_{out} = \theta_{of\_out} + \tan^{-1}\left(\tan\theta_c \cdot \cos\theta_{in}\right) \tag{21}$$

where θ_{out} is the output angle, θ_{of_out} is the offset angle at the output, θ_c is the gradient of the truncated cylinder, and θ_{in} is the input angle.

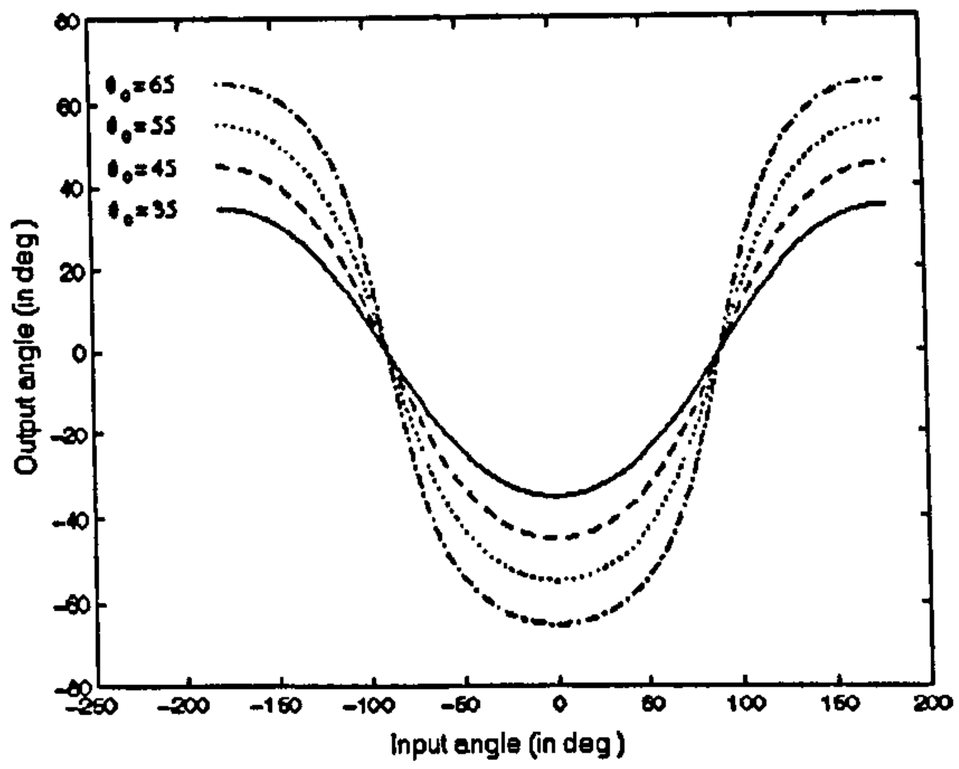


Figure 2: Transfer characteristic of the gimbal drive

2.4 The SPRINTA Concept

The non-linear direct transmission philosophy has been evaluated on the SPRINTA (Serial Parallel Robot Incorporating Non-linear Transmissioned Actuation) arm, Figure 3, which has been designed and built in the Department of Electronic and Computer Engineering at Brunel University [100]. The gimbal drives provide motor relocation through a link twist of 90°. This allows the relocation of the motors for the shoulder and elbow joint to the base of the arm. The three motors are placed at the base of the arm, they are coaxial and lie on the axis of the first joint. The waist joint is directly connected to the motor shaft, forming a direct drive system. The shoulder joint is driven directly via a gimbal drive element, while a parallel linkage is used between the elbow joint and its gimbal drive, as shown in figure 4. With coaxial motor shafts, the gimbal drive for the elbow has been accommodated inside the gimbal drive of the shoulder.

The workspace of the SPRINTA prototype is shown in figure 5, whilst robot dimensions and workspace characteristic are summarised in table 1.

The joints are driven by three NSK Megatorque® motors, two $RS0608FN003$ and one $RS0408FN003$ [67]. Each motor has its own driver which has been configured so that it responds to the command data with a torque proportional to the

Joint	Range†	Travel	θ_c	Link	Length
Waist	Unlimited	Unlimited	-	Base	565 mm
Shoulder	0° to 90°	90°	45°	Humerus	400 mm
Elbow	−120° to − 10°	110°	55°	Forearm	300 mm
†Angle above horizontal, for the elbow and shoulder joints.					

Table 1: SPRINTA parameters

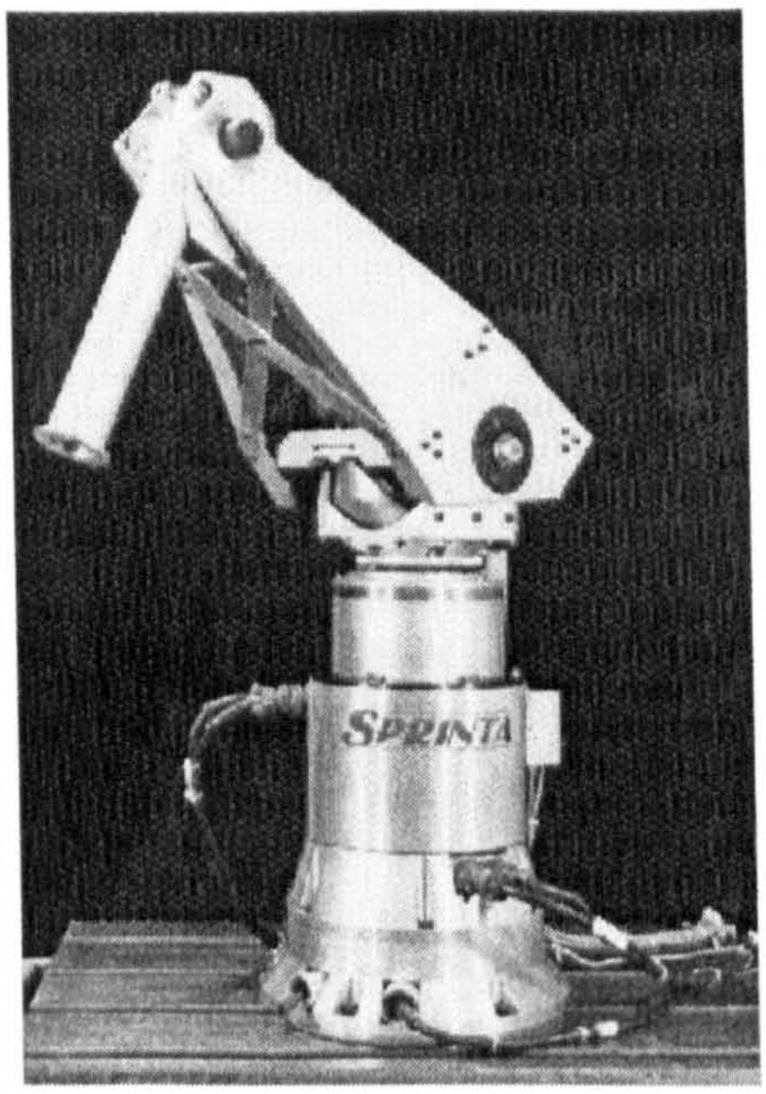


Figure 3: The SPRINTA

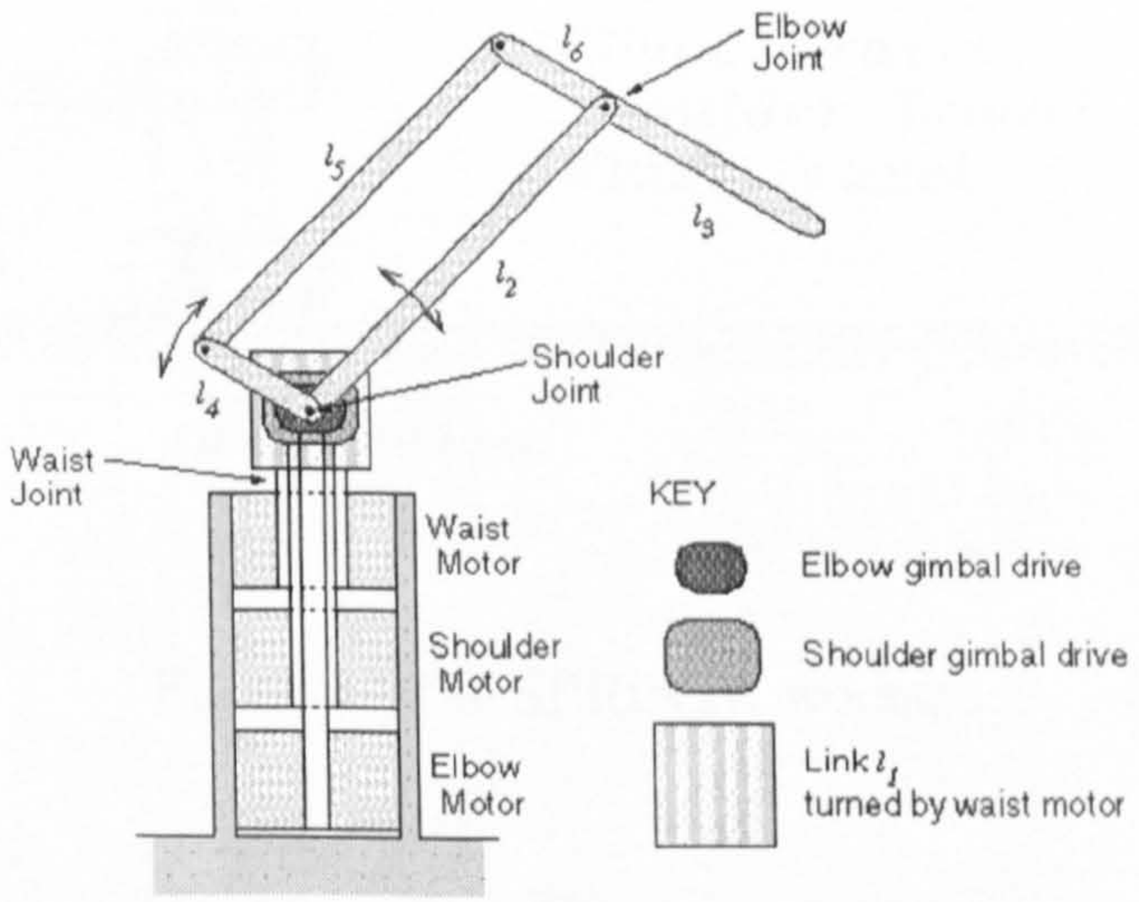


Figure 4: The SPRINTA Topology

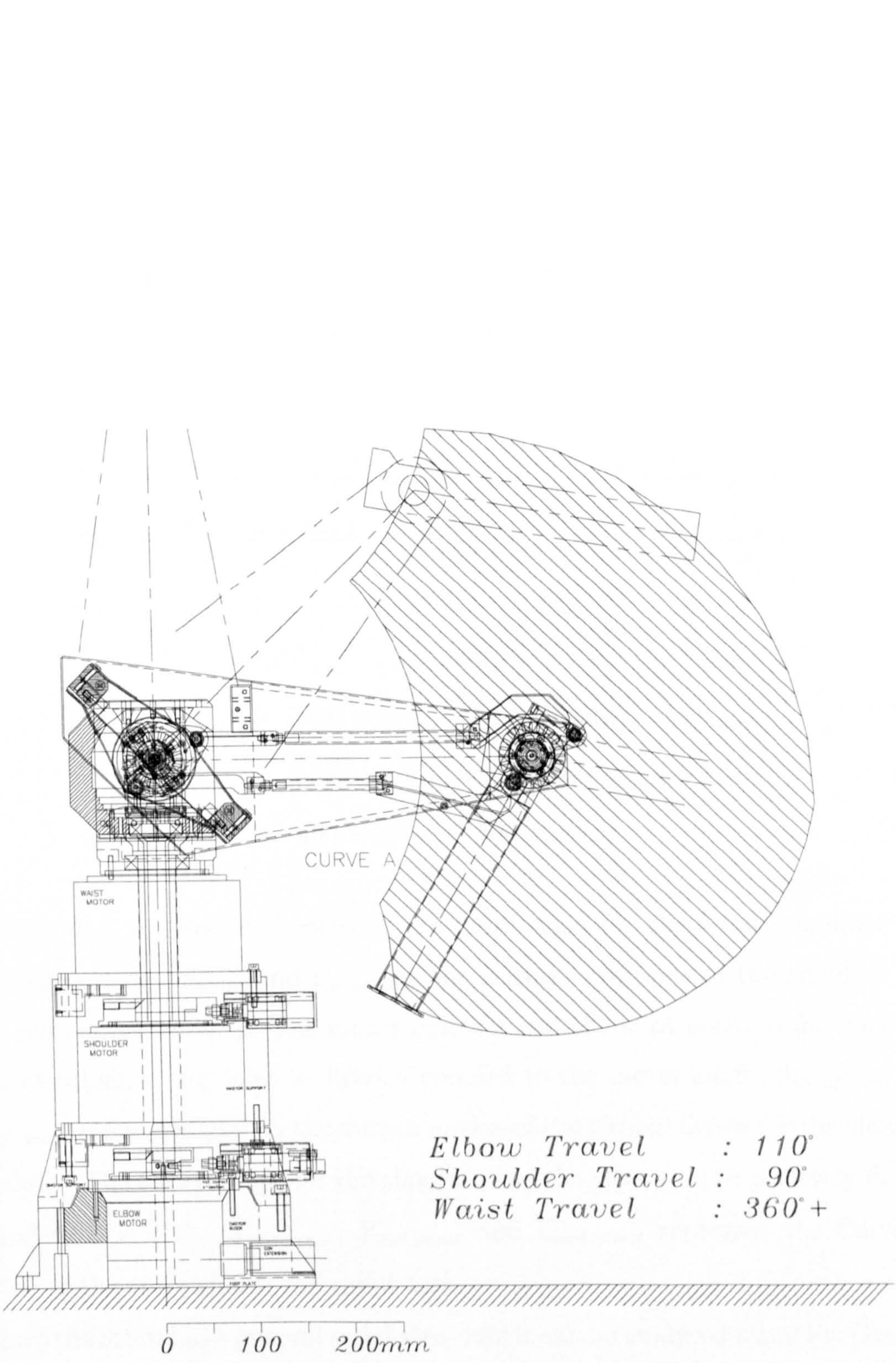


Figure 5: The SPRINTA workspace

maximum torque available. The three motors drivers have their own interface cards which are connected to a $QT960$ controller board [41]. The motor positions are sensed by incremental encoders, producing 614400 pulses per revolutions.

With the base frame located at the centre of the waist joint, the kinematic model for the SPRINTA is given by equations (22-26).

$$\theta_{sh} = \arctan\left[-\tan(\theta_{c\_sh}).\cos(\theta_{sh\_m} - \theta_{wa})\right] + \theta_{sh\_off\_set} \quad (22)$$

$$\theta_{el} = \left[\arctan -\tan(\theta_{c\_el}).\cos(\theta_{el\_m} - \theta_{wa})\right] + \theta_{el\_off\_set} \quad (23)$$

$$X_{end\_point} = \left[400.\cos(\theta_{sh}) + 300.\cos(\theta_{el})\right].\cos(\theta_{wa}) \quad (24)$$

$$Y_{end\_point} = \left[400.\cos(\theta_{sh}) + 300.\cos(\theta_{el})\right].\sin(\theta_{wa}) \quad (25)$$

$$Z_{end\_point} = 400.\sin(\theta_{sh}) + 300.\sin(\theta_{el}) \quad (26)$$

Where θ_{wa}, θ_{sh} and θ_{el} represent the joint angle for the waist, shoulder and elbow respectively. θ_{sh_m} and θ_{el_m} are the angular positions of the shoulder and elbow motors respectively. The motor position and the joint position for the waist are identical since this joint is directly coupled to the motor shaft. $\theta_{sh\_off\_set}$ and $\theta_{el\_off\_set}$ are offset angles on the output angles of the gimbal drives for the shoulder and elbow. The cut-off angle for the shoulder and the elbow are respectively $\theta_{c\_sh} = 45°$ and $\theta_{c\_el} = 55°$. X_{end_point}, Y_{end_point} and Z_{end_point} represent the Cartesian position of the end point of the robot arm.

Since the robot has a general serial structure it can be analysed using the Denavit-Hartenberg convention [39] (see APPENDIX A), and the dynamics can be derived using the Euler-Lagrange method (see APPENDIX B).

As it can be seen from APPENDIX B, the exact modelling of the SPRINTA is a lengthy and difficult exercise. The derivation of the inverse dynamics, could not be performed by hand and the scientific software MATHEMATICA® has to be used to complete the task. The outcome results in a set of highly non-linear and coupled

inverse dynamic equations for the SPRINTA. These equations could not be used for practical implementation on a controller, and so they must be simplified.

2.5 Concluding Remarks

Robot manipulators are non-linear systems, they can however be described mathematically (dynamic equations). These mathematical descriptions of the manipulator have specific properties, as mentioned earlier that are very useful for the design of controllers. The uses of these properties is even more crucial to achieve high performances (speed and accuracy) when direct drive or associated technique are employed, as non-linearities and joint coupling are directly reflected on the shaft of the motor. In the case of the SPRINTA, some of the non-linearities can be reduced, but the directness of the approach means that the physics of the system has to be exploited to achieve good performances.

3 CONTROL OF MANIPULATORS

In general, the problem of controlling a manipulator is to determine the time history of the generalised forces (forces or torques) to be developed by the joint actuators so as to guarantee execution of the commanded task while satisfying given transient and steady-state requirements.

Demand for high performance robots has led to the development of various advanced control techniques. Two general controller approaches can be distinguished for robot manipulators, the model based approach and the non-model based approach. The model based controllers consider some of the structures of the system in their design procedures. In contrast non-model based controllers do not take account of the system properties. The first group can be divided into two classes: controllers based on feedback linearisation or computed torque techniques, and passivity-based controllers. The second group which is merely an independent joint control approach, re-grouped the traditional class of Proportional+Derivative (P.D.), Proportional+Integral+Derivative (P.I.D.), phase lead and phase lag controllers. In independent joint control, an n-joint robot manipulator is considered to be made of n single input single output (SISO) systems, and this control approach has been used successfully for many years. Various adaptive and robust controllers have been developed for both model based and non-model based controllers.

Another control method which has not been mentioned earlier is that of Variable Structure Control (V.S.C.). As originally developed, this is not model-based, but recent developments have incorporated V.S.C. into model-based methods [105]. Adaptive versions have also been formulated [107] [11]. Due to their special properties, variable structure controllers hold a specific position on the set of control methodology and will be treated as a group on its own.

The choice of which control technique to use is very much dependent on the task to be performed. For instance, the need for trajectory tracking control may lead to controller implementations which differ from those allowing point-to-point control. The driving system of the joints has also an effect on the type of control strategy used. If a manipulator is actuated by electric motors with reduction gears of a

high ratio, the presence of these gears tends to linearize the dynamics of the system reflected on the shafts of the motors. With a high reduction ratio, the joints can be considered to be decoupled from each other. The disadvantage of the reduction of the effect of the non-linearities is the occurrence of joint friction, elasticity and backlash that may significantly limit system performance. Alternatively, a robot actuated with direct drives eliminates the drawbacks due to friction, elasticity and backlash but the effect of non-linearities and coupling between the joints becomes highly significant. As a consequence, different control strategies have to be selected to obtain high performance.

3.1 Model Based Controllers

This group of controllers can be divided into the feedback linearisation and the passivity-based classes. The feedback linearisation or computed torque method relies on cancellation of the non-linearities in the robot dynamics by inverting its dynamic equations. The passivity-based approach uses the passive nature of the robot manipulator in the design procedure (see section 2.1.4). It attempts to reshape the system's natural energy in such a way that tracking control is achieved. The passivity based approach for robot control arose from the theory used in electrical circuit theory analysis.

3.1.1 Feedback Linearisation

The idea behind this methodology is to transform a non-linear system into a new linear system incorporating the robot and its controller. The methodology is different from the traditional linearisation (Jacobian linearisation) used in modelling of dynamical systems [61]. The task is to design a controller which will cancel the non-linearities in the system for the entire operating workspace. The idea of feedback linearisation can only be applied to a special class of non-linear systems, those which can be described in companion form or controllable canonical form [85]; i.e. $x(n) = f(x) + b(x)u$. Computed torque is a special case of feedback linearisation for a class of non-linear systems. It is sometimes referred to as inverse dynamics since the basic idea is to compute the inverse of the system dynamics as the input torque.

A large number of studies and experiments have been performed [59], [71], [3], [4], [51]; and an extensive list of variants of the computed torque method can be found in the literature. It has proved to be an efficient method when the model is known accurately and has been employed successfully in industry.

The controller uses the general inverse dynamic equation of the robot manipulator, equation (5), as the basic design starting point. Usually the controller is composed of two parts, the feedforward and the feedback elements, as shown in figure 6. The feedforward part attempts to cancel the non-linearities in the system, while the feedback is used to stabilise the system and drive the error to zero. This is done by augmenting the acceleration demand by the signal U, which is a function of the tracking error. In practice, the system parameters can not be known exactly, so the computed torque controller uses an estimated version of the system dynamics.

$$T = \widehat{M}(\theta)\,(\ddot{\theta}_d - U) + \widehat{C}(\theta,\dot{\theta})\,\dot{\theta} + \widehat{G}(\theta) \tag{27}$$

where :

$\widehat{M}(\theta)$, $\widehat{C}(\theta,\dot{\theta})$ and $\widehat{G}(\theta)$ are estimates of the system parameters,

$\ddot{\theta}_d$ is the acceleration demand, and

U is the feedback error signal.

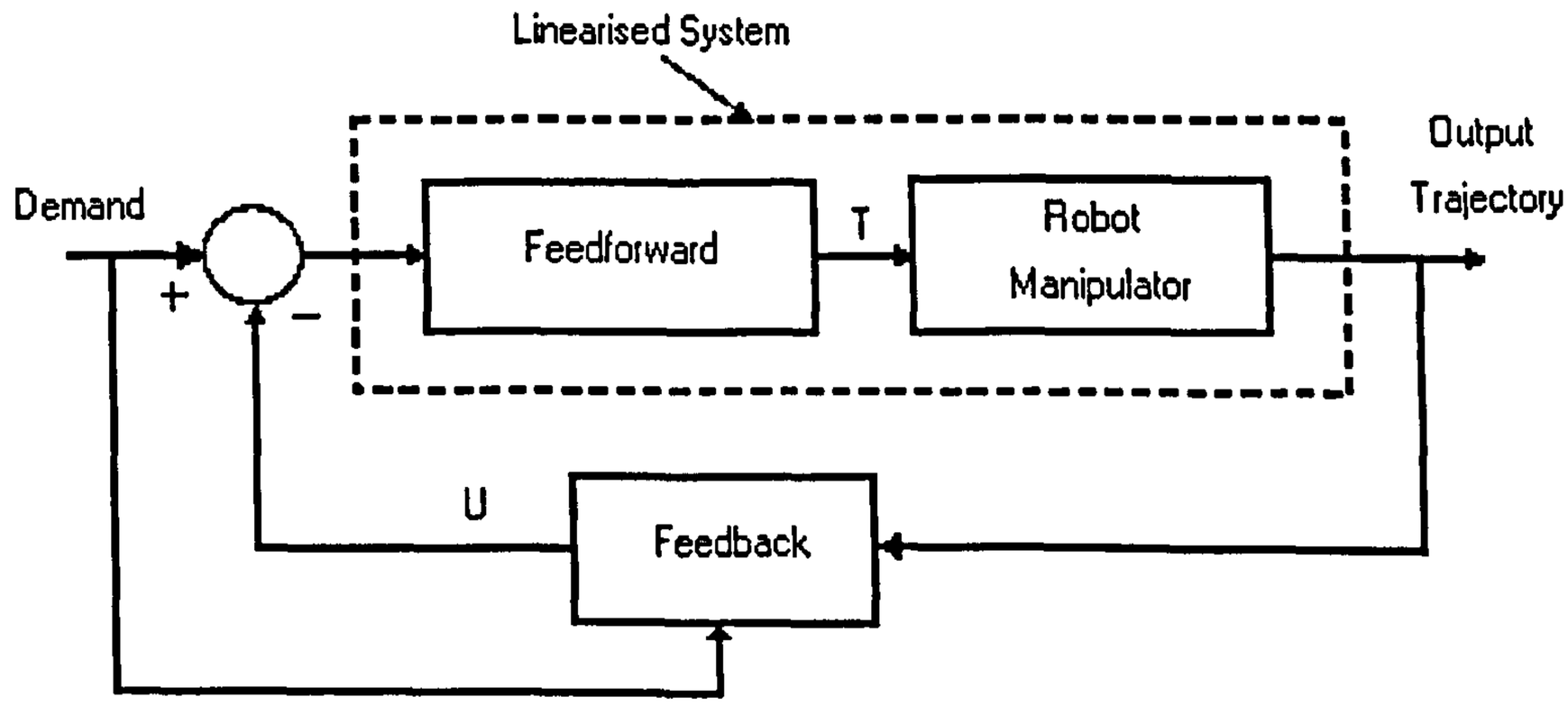


Figure 6: General Computed Torque Controller

The system uncertainties do not cause problems in terms of control as long as the estimated model is sufficiently close to the real one. The feedback loop is used

to correct for the system uncertainties and to guarantee acceptable performance. As the uncertainties increase however, the system performance decays rapidly such that the system may become unstable [34], [84]. When the uncertainties are within an acceptable range, the outer, or feedback loop is designed in such a way that the error converges to zero. To achieve this, a number of control laws may be employed. The most common and simplest to implement are the P, P.D. and P.I.D. type outer loop control [4]. P. type outer loop control is the most simple, where the feedback error signal $u(t)$, is directly proportional to the position error, equation (28). The P.D. controller incorporates a derivative term in its control law to improve the transient response of the system, i.e. the error signal is augmented by a velocity error, equation (29). The P.I.D. controller is an extension of the P.D. type with an integral term, equation (30). The integral element is used to remove the steady state error which may be present on the system because of Coulomb friction or other disturbances.

$$u(t) = K_p E \tag{28}$$

$$u(t) = K_p E + K_d \dot{E} \tag{29}$$

$$u(t) = K_p E + K_d \dot{E} + K_i \int E.dt \tag{30}$$

where :

K_p K_d and K_i are positive diagonal matrices,

$E = \theta_d - \theta$, is the column vector of the position errors,

$\dot{E} = \frac{dE}{dt} = \dot{\theta}_d - \dot{\theta}$, is the column vector of the velocity errors, and

$\int E.dt$ is the column vector of the integral of the position error with respect to time.

Although theoretically based techniques exist to tune the elements Kp, Kd and Ki, an empirical approach is usually employed in practice. The effectiveness of the entire algorithm equation (27), is very much dependent on the accuracy of the model used as the feedforward term. This dependance has motivated engineers and scientists to search for a way to render the system insensitive to model uncertainty. Two alternative philosophies have been proposed to achieve this: robust and adaptive

control (discussed in section 3.3).

In summary, the computed torque approach is based on the feedback linearisation concept. Such controllers are designed in two steps. Initially, the structure of the feedforward term that provides linearisation is established, with an estimate of the system dynamics. The second step consists of defining $u(t)$ such that the control objective is achieved.

3.1.2 Use of Passivity in Controller Design

In recent years, the passivity-based approach to robot control has gained much attention [82], [69], [15], [68]. As opposed to computed torque control, this method tackles the robot control problem by exploiting the physical structure of the manipulator, and specifically its property of passivity, equation (19). A passivity-based controller relies on the passive structure of the manipulator, which is expressed by the skew-symmetry property, equation (16). This property means that there is a passive mapping from the input torque to the joint velocity, i.e. the arm cannot create energy, it is a dissipative system. From a control point of view, the passivity property states that the system is stable. The idea of the passivity approach controller is to close the loop from the velocity to the input torque with a passive element [85], [83], [55], as shown in figure 7. Since the path from torque to velocity is passive, as is the path from velocity to torque, the overall system composed of both systems is also passive and hence stable.

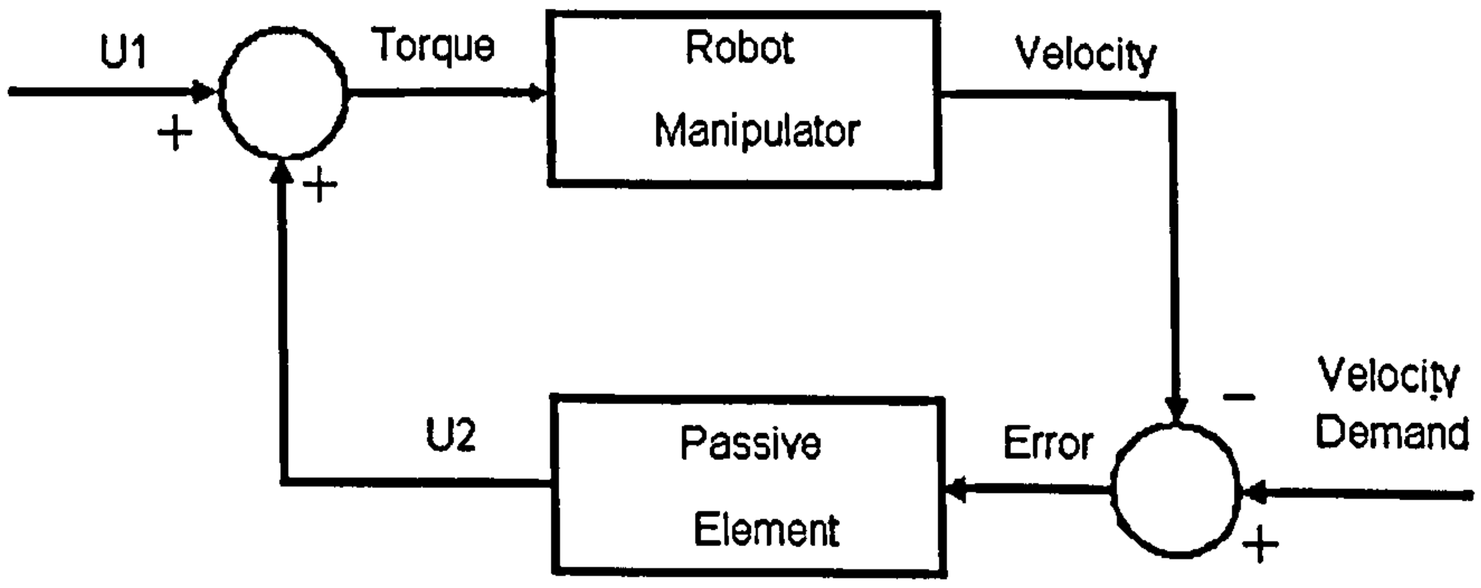


Figure 7: Passive Loop

The structure shown in figure 8, preserves the passivity property of the system

and guarantees asymptotic stability of the complete system. The input signal U_1 is defined to be bounded, it is used to provide the regulation point. The passive loop from velocity to torque is obtained using the velocity error (note also that the velocity demand $\dot{\theta}_d$ is defined to be bounded). The velocity error is then used as a filtered version of the actual velocity to close the loop. The stability is maintained as long as the loop element is passive or described by a strictly proper, stable, rational function. To satisfy the condition of passivity, the feedback element may be a P., P.D. or P.I.D. type function, equations (28) to (30). The feedforward part that generates the signal U_1 is usually chosen to be a model-based form, i.e. a computed torque based structure. As shown by Berghuis [15], any passivity-based controller can be fitted into the general structure of figure 8, where the feedback element is a passive function of the manipulator error and the feedforward term is designed according to the robot inverse dynamics. The controller can be expressed by the general form, equation (31).

$$T = \widehat{M}(\theta_f)\,\ddot{\theta}_f + \widehat{C}(\theta_f, \dot{\theta}_f) + \widehat{G}(\theta_f) + \Phi \tag{31}$$

Where:

Φ represents the closed loop passive element, which can be P., P.D. or P.I.D. type, and

θ_f, $\dot{\theta}_f$ and $\ddot{\theta}_f$ are respectively functions of the actual position, velocity and acceleration of the robot.

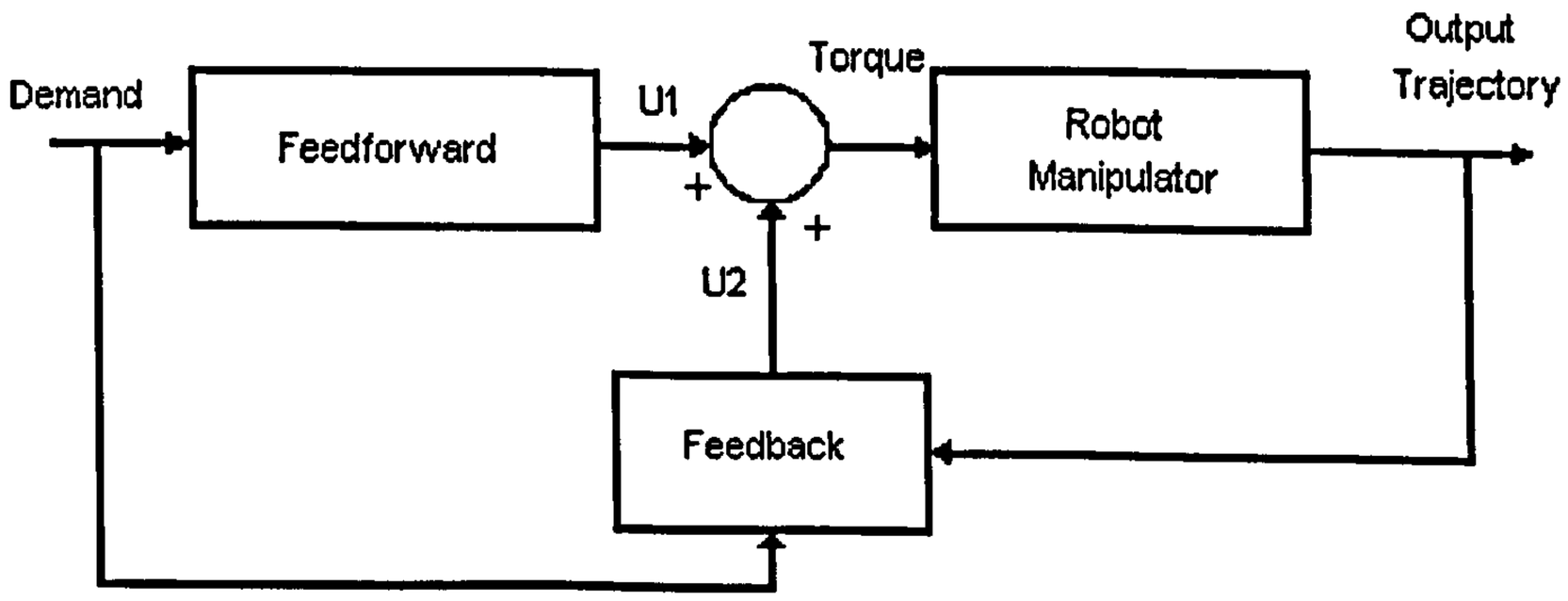


Figure 8: General Passivity-Based Controller

A number of controllers with a form similar to equation (31) have been proposed [49], [46], although they are not systematically referred to as passivity-based. Nevertheless, their structures fit the general passivity form and the stability of such controllers is guaranteed by the passive nature of the algorithm. Just as for the computed torque method, passivity-based controllers exist in adaptive and robust versions. Since an estimated model of the robot is used for the feedforward term, the performance of the system depends on the accuracy of this model. An adaptation mechanism or a robust controller may be necessary to correct for the uncertainties and unmodelled dynamics of the system or to compensate for parameter changes, e.g. load variations.

In summary, the use of passivity for robot control consists of constructing a controller such that the closed-loop system matches a desired energy function that resembles the natural energy contents of the open-loop system, by using a computed torque like structure for the feedforward terms. In this way, passivity of the robot can be preserved in the closed-loop. Moreover, the inclusion of damping via a velocity feedback loop guarantees the asymptotic stability of the overall closed loop system.

3.2 Non-model Based Controllers

The simplest control strategy that can be thought of is the one that decomposes an n-joint manipulator into n-independent systems allowing independent joint control. This control approach has been used since the early age of robotics, and is referred to as the single input single output (S.I.S.O.) technique or independent / classical joint control. Classical joint control has been applied to robotic manipulators because of its ease of implementation and reliability. The effectiveness of S.I.S.O. control is confirmed for manipulators with relatively low joint interaction reflected at the actuators. This situation is only valid for slowly moving links and highly geared transmissions. The reduction ratios in the transmissions reduce the link interactions and the variation in inertia as seen by the actuators. For directly driven links and manipulators where the inertia matrix varies for the desired trajectory within the range of 1:4, the interaction between links may be significant [10]. In this case independent joint control may not provide acceptable results and the interaction

between the links has to be accommodated in the controller design via a model-based like approach.

Without loss of generality, the actuator is taken as a rotary electric DC motor, figure 9. The position, velocity and acceleration are respectively represented by θ, $\dot{\theta}$ and $\ddot{\theta}$, I is the average inertia experienced by the motor shaft, Ra is the armature resistance, and Kv and Kt are respectively the torque and back emf constants. Gv denotes the voltage gain of the power amplifier, Va and Vc are respectively the armature voltage and the input voltage. The joint interaction is represented by D, [21].

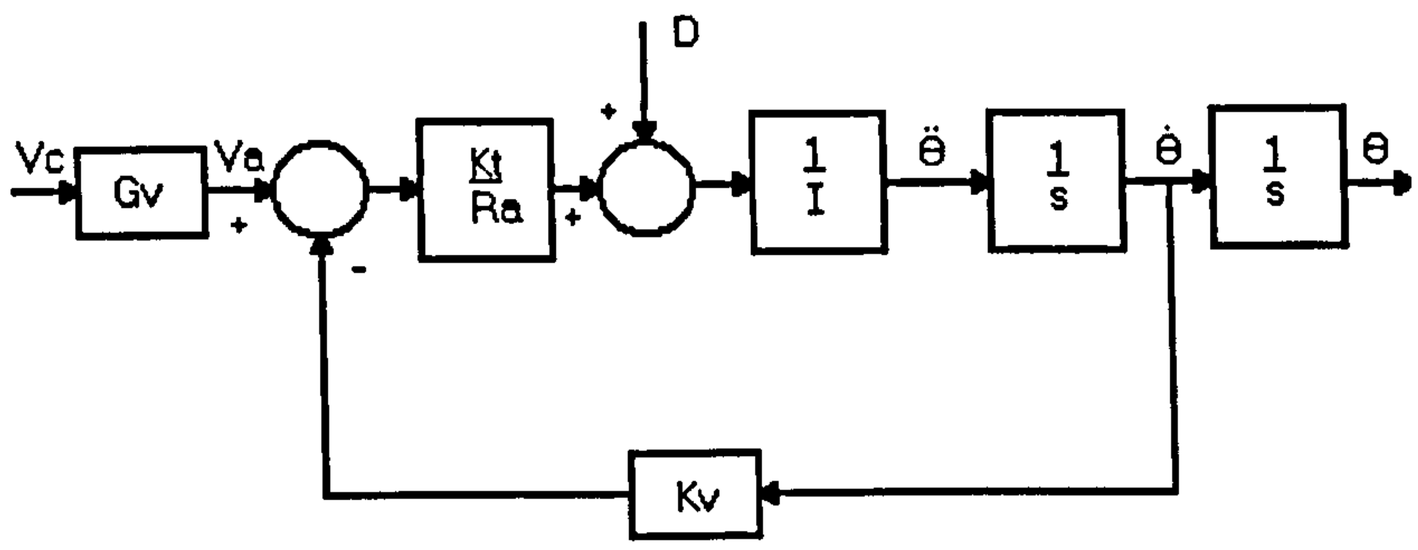


Figure 9: Model assumption of a joint drive system for SISO control

If the model in figure 9 is assumed to be accurate, then classical linear control techniques can be applied, figure 10. An error signal is generated from the difference between the demand signal and the system response, and fed through a controller of the P.I. or P.I.D. type, equations (32) and (33) respectively. Coupling effects between joints due to varying configurations during motion are treated as disturbance inputs. Effective rejection of these disturbances is usually achieved by a large gain and integral action [58].

P.I. type :

$$Vc = K_p E + K_i \int E \tag{32}$$

P.I.D. type :

$$Vc = K_p E + K_d \dot{E} + K_i \int E \tag{33}$$

The terms have the same definition as for equations (28) to (30).

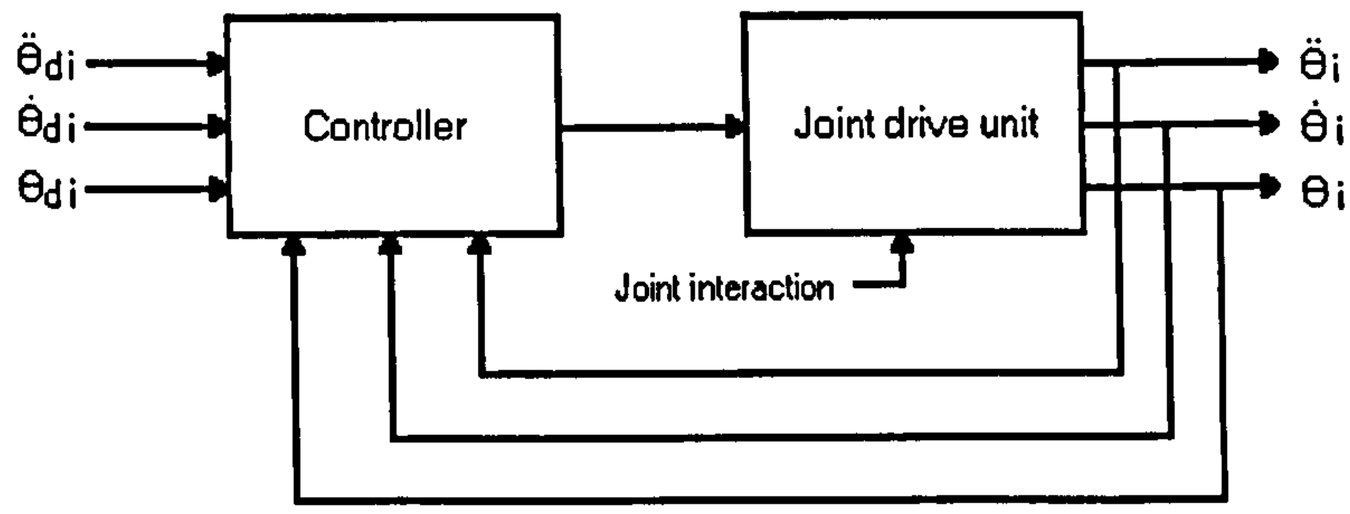


Figure 10: General independent joint control

Usually for industrial robots, independent joint control gives reasonably good results for set point regulation as well as for trajectory following. Although, the validity of the scheme can be appreciated from an engineering point of view, and the large number of applications confirm its effectiveness [24], [3], it is only very recently that theoretical proofs have justified the application and the stability of such schemes in robotic applications [75], [87].

The main advantage of this type of control technique is that very little knowledge of the manipulator dynamics is needed for implementation. In addition the computation requirements for these type of controllers are extremely low and the underlying principles are well established.

A number of variants of these schemes have been proposed to enhance the manipulator responses [6], [71], [55]. The P.D. plus gravity compensation technique, equation (34), is one of them [50] [89]. Although, this type of controller uses some physical knowledge of the mechanical system, it has been classified in the non-model-based controller class, as the major part of the controller does not use knowledge of the system dynamics. The gravity compensation term replaces the integral action of PID, as it mainly affects the steady state performance of the manipulator. This type of system may be viewed as a hybrid type controller. It should be noted that the control signal generated by the controller represents the torque T.

$$T = K_p E + K_d \dot{E} + \widehat{G}(\theta) \tag{34}$$

where :

K_p, E, K_d and $\dot{E}$ are defined as in equation (29) and $\widehat{G}(\theta)$ is the estimate of the gravitational term.

An extensive amount of work has been done on independent joint control and a large family of controllers and hybrid controllers for S.I.S.O. control have been proposed. Luh [58] gives a survey on the subject. As for the previous class, adaptive and robust schemes have been proposed to alleviate the problems of joint interaction, unmodelled dynamics and parameter variations [50], [91].

3.3 Dealing with Uncertainties and Parameter Variations

Robot manipulators are non-linear systems, and are subject to variations while operating (torque variation within the workspace or variation of the load carried). To deal with these situations the controller must have some built-in properties to accommodate these variations. Further, in the case of model-based control, the model used very often presents some uncertainties, either in its structure (approximation of the model) or in its parameters.

Approximations on the plant to be controlled are introduced to simplify the model, and the system parameters are only known to within a certain accuracy. In either case, the task is to design a controller which will give satisfactory response in the presence of these uncertainties (uncertainties which may be in some cases very important).

Two major philosophies have been developed to deal with uncertainty. Robust control, in which the structure is fixed but a satisfactory response is acheived for a given class of plant, and adaptive control, which attempts to determine the uncertainties and hence improve the performance. Adaptive control of robot manipulators is usually applied to the model-based family. As it implies that the structure of the model is very close to the real system.

3.3.1 Robust Controllers

Many of the previously discussed classes of controllers exist in robust versions or are intrinsically robust, e.g. P.I.D., V.S.C. with sliding mode. The robustness properties may however be valid to within a certain range. The aim of the robustness property for the controller is to render the system insensitive to parameter mismatches, i.e. uncertainty in the dynamic terms of the model-based controller; and to disturbances, i.e. joint interactions, measurement noises and noises affecting the system itself. A robust system gives a satisfactory response in any situation for which it has been designed. If no disturbances and mismatch are present, the system does not achieve optimum response but instead gives a response within a predefined acceptable range. Robust controllers are usually simpler to implement than their adaptive counterparts and no time is required for tuning the controller (adaptation time) to the particular plant.

For robot manipulators, disturbances affecting the system can be bounded by some known functions. The uncertainties or unmodelled dynamics of the system can also be bounded by some functions. These functions may be used to define the rejection range of the controller (its robustness). It is important that the disturbances are not over estimated too much as this leads to excessive control effort. For a manipulator, an adequate use of the system dynamics properties leads to a less conservative controller and lower control effort.

Various methodologies have been suggested to design the robust control scheme [1], [55]. The Lyapunov approach is one in which a candidate Lyapunov function is selected to define the robustness of the controller. The most commonly selected Lyapunov function is the pseudo-kinetic energy function [35]. It is formed using a function of the error and the inertia matrix of the manipulator. The stability and robustness of the scheme are guaranteed by establishing a strictly negative bound on the derivative of the Lyapunov function. This must be valid for any point in the operating range of the manipulator. This approach is very conservative (garantee the existence of the Lyapunov function is a sufficient and not a necessary condition for stability), but it may nevertheless be a starting point for further refinements. Passivity-based controllers are also part of the robust control family. As discussed

previously, the stability of the system can be guaranteed by closing the loop from velocity to torque using a passive element. Another approach for the design of a robust controller for robot manipulators is the variable structure system in conjunction with sliding mode [104]. One of the main features of this approach is that the error only needs to be driven to a switching surface, after which the system is in sliding mode and will not be affected by modelling uncertainties and/or disturbances within a defined range. This specific type of control methodology will be discussed in more detail in Chapter 4.

3.3.2 Adaptive Controllers

Adaptive control is the other main approach for dealing with uncertainties. Research in adaptive control started in the early 1950' s in response to the development of high performance aircraft, but research did not progress very much due to the lack of a global theory. It was abandoned until the late 1960's, when interest in adaptive controllers started again.

In many robot control problems where some form of model based control is used, some of the parameters of the plant to control are partially or totally unknown (the inertia, mass, length, etc...). It may also happen that the process, during its normal operation, is subject to changes in parameters. The robot may be required to manipulate loads which may vary in size, weight and mass distribution. In order to give consistent responses in terms of speed and accuracy in any of these situations, the controller has to adapt itself to cope with these varying or uncertain parameters. The basic idea in adaptive control is to estimate on-line the unknown parameters, or alternatively the parameters of the controller to give satisfactory response. Adaptive controllers can be divided into two categories : the Model-Reference Adaptive Controller, usually referred to as M.R.A.C., and the Self-Tuning Controller (S.T.C.).

In the M.R.A.C., a model of the plant is used to generate the ideal output that the plant response must follow. The controller parameters are updated in a way that the plant achieves perfect tracking of the reference model response. The M.R.A.C. can be schematically represented by four elements, as shown in figure 11. The reference model, which is an approximation of the real system, has to

satisfy two conditions. It specifies the desired dynamic behaviour of the system (rise time, settling time and/or overshoot) but must also provide responses which can be achieved by the system. The parameters of the controller are adjusted such that the plant produces a response identical to the reference model. The stability and convergence to zero of the signal errors are guaranteed by an adequate choice of adaptive laws for the parameters. The adaptation law should also guarantee the convergence of the controller parameters. The design procedure for such a controller can be considered in four steps : first select a reference model which is relatively close to the real plant, but simple in structure, and then choose a control law containing variable parameters. The third step is to design adaptation laws for the parameters, and the final stage in the design is to check the convergence properties of the system [52].

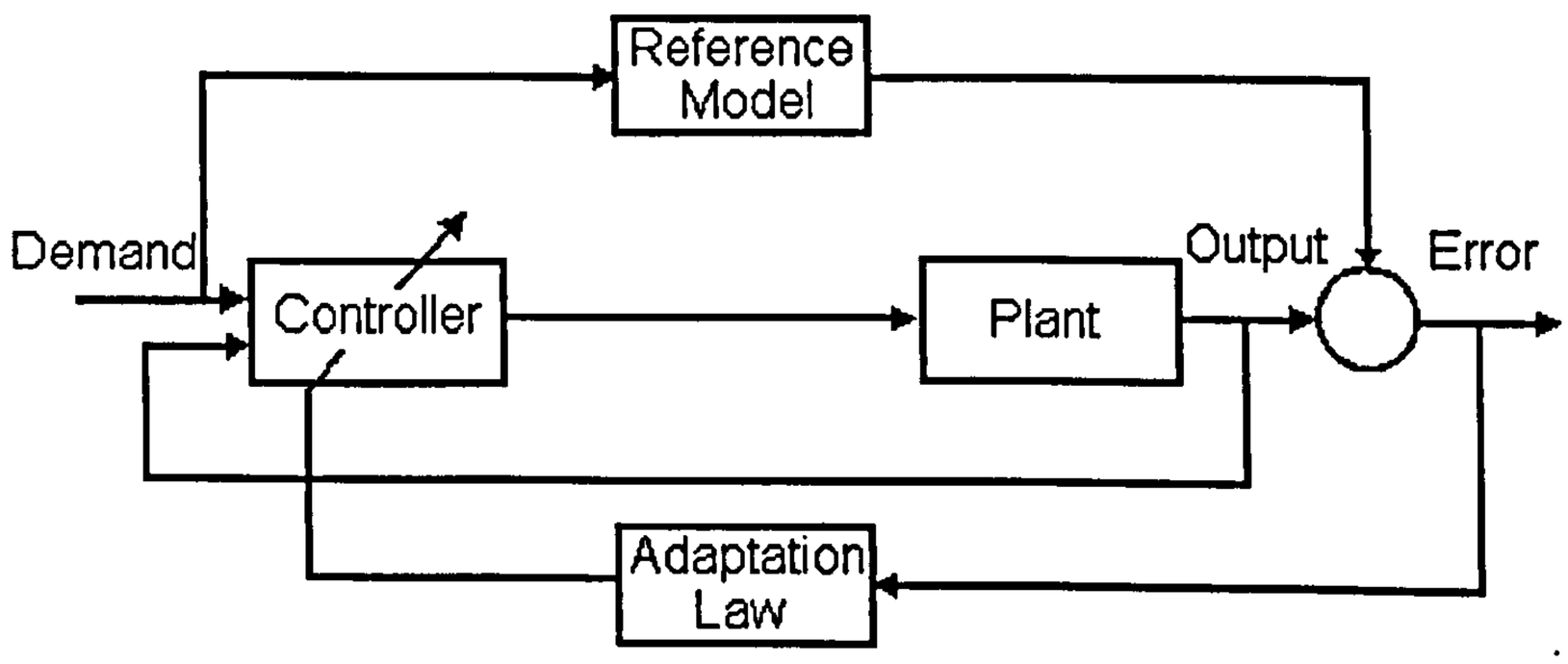


Figure 11: Model Refence Adaptive Control Scheme

In some applications, the reference model is not explicit. It is incorporated into the controller and parameters are directly adapted so that the reference model matches the plant.

For the self-tuning controller, no pre-estimation of the plant is required as the parameters of the controller are adapted using on-line estimation. One simple approach is to estimate the unknown parameters of the plant and then compute the controller parameters from these results. This method is referred to as indirect adaptive control. A more straightforward technique called direct adaptive control, reformulates the estimated parameters in terms of controller parameters.

From a general point of view, the main difference between M.R.A.C. and S.T.C. is that in the former, the parameters of the plant are tuned such that the error between the reference and the actual plant vanishes. In the latter, the controller parameters are estimated such that the input-output error is minimized. Some similarities exist however between the two schemes [85]. In robotics, both schemes have been employed, for example Seraji [77] proposed a self-tuning P.I.D. type controller. It seems that M.R.A.C. however has been the most active research field. The reason may be that using the linearity in parameters property, equation (20), the uncertainty on the inverse dynamic equation can be expressed as simple coefficients. Formulating the reference model in terms of the controller and putting the uncertainties as coefficients, estimation mechanisms can be used to evaluate the uncertainties, and update the reference model so that it matches the plant.

Using the linearity in parameters property, a number of adaptive controllers for robotics have been proposed most of them belonging to the MRAC scheme [23], [48], [26], [60], [76] . Although the adaptation mechanisms are different for all these controllers they all rely on the linearity in parameters property, equation (20) and below.

$$M(\theta)\,\ddot{\theta} + C(\theta,\dot{\theta})\,\dot{\theta} + G(\theta) = Y(\theta,\dot{\theta},\ddot{\theta})\delta$$

Selecting an initial control law of the form :

$$T = -\hat{M}(\theta).\dot{e} - \hat{C}(\theta,\dot{\theta})e + \hat{G}(\theta) - K.r \tag{35}$$

where e, $\dot{e}$ and r may be defined as :

$$e = \dot{\theta}_d - \Lambda\,\theta = \dot{\theta}_d - \Lambda(\theta - \theta_d)$$

$$\dot{e} = \ddot{\theta}_d - \Lambda\,\dot{\theta}$$

$$r = \dot{\theta} - e = \dot{\theta} + \Lambda\,\theta$$

With K, Λ diagonal matrices of positive gains, $\hat{}$ denotes the estimates. In terms of the linear parameterization of the robot dynamics, the control law can be expressed as :

$$T = Y(\theta, \dot{\theta}, \ddot{\theta}, e, \dot{e})\,\hat{\delta} - K \cdot r \tag{36}$$

Combining equation (36) with equation (5) yields

$$M(\theta)\,\dot{r} + C(\theta, \dot{\theta})r + K \cdot r = Y(\theta, \dot{\theta}, \ddot{\theta}, e, \dot{e})\,\tilde{\delta} \tag{37}$$

The parameter estimate $\hat{\delta}$ may be computed using standard methods, such as gradient or least squares [44]. For example, using the gradient update law

$$\dot{\hat{\delta}} = -\Gamma^{-1} Y^T(\theta, \dot{\theta}, \ddot{\theta}, e, \dot{e})r \tag{38}$$

where Γ is diagonal positive definite matrix.

A number of important refinements to this initial result are possible [84], [83], [46], [69].

Another type of adaptive controller for robot manipulators has recently appeared in the literature with the development of Neural Networks [56], [57]. Very often with this type of controller, some form of model-based element is used and the uncertainties are compensated by the neural network part. The neural network is either used to estimate the parameter uncertainties and then adjust the model-based element, or to simply compensate mismatch between the model used and the real system. In the latter, no parameter identification is performed, so it is usually simpler and requires less computational load. The drawback of these new type of adaptive controllers is the amount of computer power required for implementation and the time taken to tune the neurons before acceptable performances are obtained.

3.4 The Case of the SPRINTA

The SPRINTA uses some form of transmission directness, in the sense that the motors are not directly coupled to the joint but directness is still kept [100], i.e. no gearing element. In this situation the link interactions reflected at the motor shaft may not be negligible. The significance of joint interaction at the motor shaft would lead to the choice of a model-based controller in order to maximise

the manipulator performance. In the derivation of the inverse dynamics it has been found however that the exact model of the robot could not be used for the controller design (APPENDIX B). This is due to the complexity of the equations, which make their on-line computation very slow. The model has to be greatly simplified for implementation purpose, and as a result significant uncertainties are introduced in the system structure as well as in the parameters themselves. Adaptive controllers can deal with parameter uncertainties, however they are not good at dealing with unmodelled dynamics, as is the case of the SPRINTA model. A neural network approach can not be used because of the limited computer power available on the existing system.

The other technique to deal with uncertainty is robust control. Such a controller may not provide optimum responses but can deal with parametric uncertainties as well as unmodelled dynamics. One type of robust controller, which is inherently robust and has been applied to various type of plant (linear as well as non-linear), is the variable structure controller in conjunction with sliding mode. This type of controller which has not been presented previously is described in the next chapter. This type of controller requires very little computer power, is very efficient to cope with the uncertainty, and it can be designed in conjunction with a model-based structure controller.

4 VARIABLE STRUCTURE SYSTEM & SLIDING MODE

4.1 Background

Variable Structure Control (VSC) in conjunction with sliding mode constitutes a special class of control techniques that renders the system robust to parameter variations as well as to disturbances. This technique provides a systematic approach for the control of linear as well as non-linear plant.

The theory of VSC was pioneered in the Soviet Union during the early 1950's and 1960's by Professors S. V. Emel'yanov and E. Barbashin and their co-workers [28], [16], [13], [14]. The study and development of this control technique is closely related to the concept of state space methods initiated by A. A. Andronov [17], [92]. The idea behind VSC originated with the use of relay control [63], where the control element is deliberately allowed to change the sign of its control effort in accordance with some functions of the error signal and its derivatives. A fundamental property that can arise with variable structure systems is the so called sliding mode. A sliding mode occurs when the system behaviour is restricted to a specific region of the system dynamics referred as switching line or switching curve. This line is defined by the switching conditions of the variable structure controller. When the system state reaches the switching line, it is bound to remain in it. The dynamic of the system on the switching line is referred to as sliding mode. While in this mode, the system dynamics are entirely defined by the equation of the switching condition, making the system insensitive to disturbances and parameter variations.

The technique received very little attention in the West, due to the shortage of literature dedicated to it in English and the lack of applications. It is only in the late 1960's and early 1970's that VSC with sliding mode started to penetrate the Western world, as scientific documents became available in English [97], [95], [96], [43], and the development of fast switching electronic devices made VSC with sliding mode potentially applicable to real systems [29], [31], [104], [105].

In its most simple form, the output of the VSC controller varies between two

extreme values according to the sign of the error function, S.

$$U = sgn(S) \tag{39}$$

with :

$$sgn(S) = \begin{cases} +1 \text{ if } S > 0 \\ -1 \text{ if } S < 0 \end{cases} \tag{40}$$

where :

$S(x)$ is a function of the state errors, satisfying $S(0) = 0$.

$S(x) = 0$, is commonly referred to as the discontinuity surface, sliding manifold/surface or switching curve/line/surface. Using a state space representation of the state errors, S represents a curve separating the plane in two and passing through the origin. If the sign of the control law U, can be arbitrarily specified in such a way that the system behaviour always converges toward the discontinuous surface S, it can be guaranteed that when this surface is reached the system behaviour will be restricted to this surface and will slide along it to the origin of the plane. This phenomenon, called sliding mode, is the result of the high frequency switching of the control law. When the state motion reaches the sliding surface, it overshoots by an infinitesimally small value creating the sign change of the control law; the motion is redirected toward the surface, and overshoot on the other side occurs. The control law is once again reversed, and since this phenomenon takes place at infinitely high frequencies, the motion is restricted to the sliding surface. A Russian mathematician A. F. Filippov [30] proposed a rigorous definition of the behaviour in sliding mode in the early 1960's, figure 12. Sliding mode corresponds to a limiting behaviour because control switching occurs infinitely fast, so the system behaviour in sliding mode f^o, at a specific point is defined as the 'average' solution between the two system behaviours f^+ and f^- ,on both sides of the discontinuous surface.

$$f^o = (1 - \alpha) \cdot f^+ + \alpha \cdot f^- \tag{41}$$

where α is a scalar such that $0 < \alpha < 1$, and f^o is tangential to the surface S at

the point of interest.

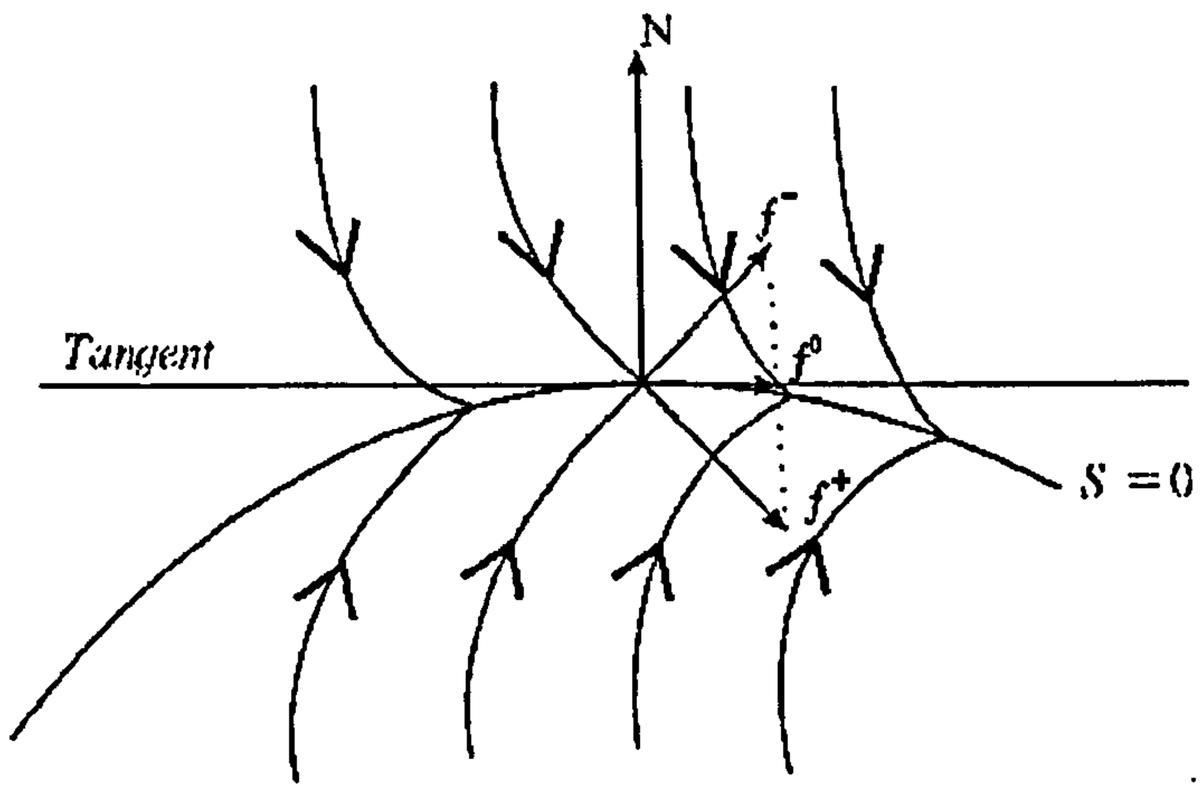


Figure 12: Filippov's interpretation of sliding mode

In practice, the switching occurs at a finite frequency so pure sliding mode does not take place. Instead, the representative point (system behaviour) is restricted to a neighbourhood of the sliding surface, which is directly dictated by the switching frequency. In this condition, the control signal oscillates at the switching frequency, and this phenomenon is referred to as chattering [98]. Chattering is undesired as it leads to high frequency oscillations of the plant which cause severe vibrations that damage the plant.

4.2 General Procedure

The design of a variable structure controller with sliding mode may consist of three phases. The first one is to define a switching surface so that the plant restricted to the surface has desired dynamics, such as stability to the origin, tracking, regulation, etc. The second phase is to derive the control law with its associated switching logic such that the state is driven toward the sliding surface for any given initial condition and maintained on it after interception. The last phase consists of selecting a mechanism/law to eliminate or reduce the chattering.

4.2.1 Sliding Surface

The design of the switching surface is dictated by the system requirements, which can be transcribed in terms of the behaviour of the state error. One remarkable property on sliding mode is that when the system reaches the discontinuous surface it remains on it and slides along it to the origin of the state plane. If the origin is defined as the required system behaviour to attain, one has to select the discontinuous surface S as a function of the system state errors, so that the system after reaching the switching surface will converge to the origin defined as zero state errors. In the case of a tracking control problem of a SISO mechanical system, the sliding surface can be defined as

$$S = \dot{\tilde{x}} + \lambda \tilde{x} \tag{42}$$

where $\dot{\tilde{x}}$ is the velocity error and $\tilde{x}$ is the position error of the system and λ is a positive scalar. The function S can be interpreted as the weighted sum of the position error and the velocity error.

The equation (42), represents a straight line of slope λ in the phase plane representation of the state errors, figure 13. When the system is in sliding mode, $S = 0$, and the dynamic of the state error is defined by a first order differential equation (43), for which the solution is simply an exponential type function of time constant λ, equation (44).

$$\dot{\tilde{x}} + \lambda \tilde{x} = 0 \tag{43}$$

$$\tilde{x} = -\lambda \cdot e^{-\lambda . t} \tag{44}$$

For an nth order system with m inputs, m sliding surfaces are generally selected and defined similarly to the SISO case.

Remark 1 The maximum number of possible sliding surfaces for an nth order system with m inputs, is $2^m - 1$. The first m sliding surfaces are given by each scalar switching function S_i for each input. By considering the intersection of two surfaces S_i and S_j as another sliding surface, the total number of such intersections equals the number of combinations of m taken two at a time, i.e. $m(m-1)/2$. Further

intersections involving multiple surfaces S_i constitute another sliding surface, and the last sliding surface is given as the intersection of all surfaces. On each of the switching surfaces, there may be a sliding mode that is described by a differential equation of the same dimension as the switching surface. Therefore, it is possible to have a total of $2^m - 1$ different sliding modes. This way of defining sliding modes was implied in the hierarchical control scheme of Utkin [98].

4.2.2 Control Law

The selection of the control law is the second major step in the design of a VSC with sliding mode. Its role is to guarantee that the states converge toward the sliding surface and are maintained on it thereafter. One method to derive the control law is based on the reaching condition, which is a sufficient condition for the system to reach the discontinuous surface and for the sliding mode to exist. The condition may be expressed as :

$$\begin{aligned} \dot{S} &> 0, \text{ when } S < 0 \\ \dot{S} &< 0, \text{ when } S > 0 \end{aligned} \tag{45}$$

which can be formalised as :

$$\dot{S}\,.S < 0 \tag{46}$$

The reaching condition, equation (45) has a physical interpretation in state space, figure 13. When the system is not in the sliding surface $S \neq 0$, its value expresses how far the system is from it, and whether it is above or below it. $\dot{S}$ represents the gradient of the system behaviour in the state space. If for all the operating points of the system, the gradients are always directed toward the sliding surface, then the system will converge toward the discontinuous manifold and reach it, and the sliding mode would then take place. The above conditions are formalised by the reaching condition, equation (45). The control law has to be selected to guarantee the sliding condition.

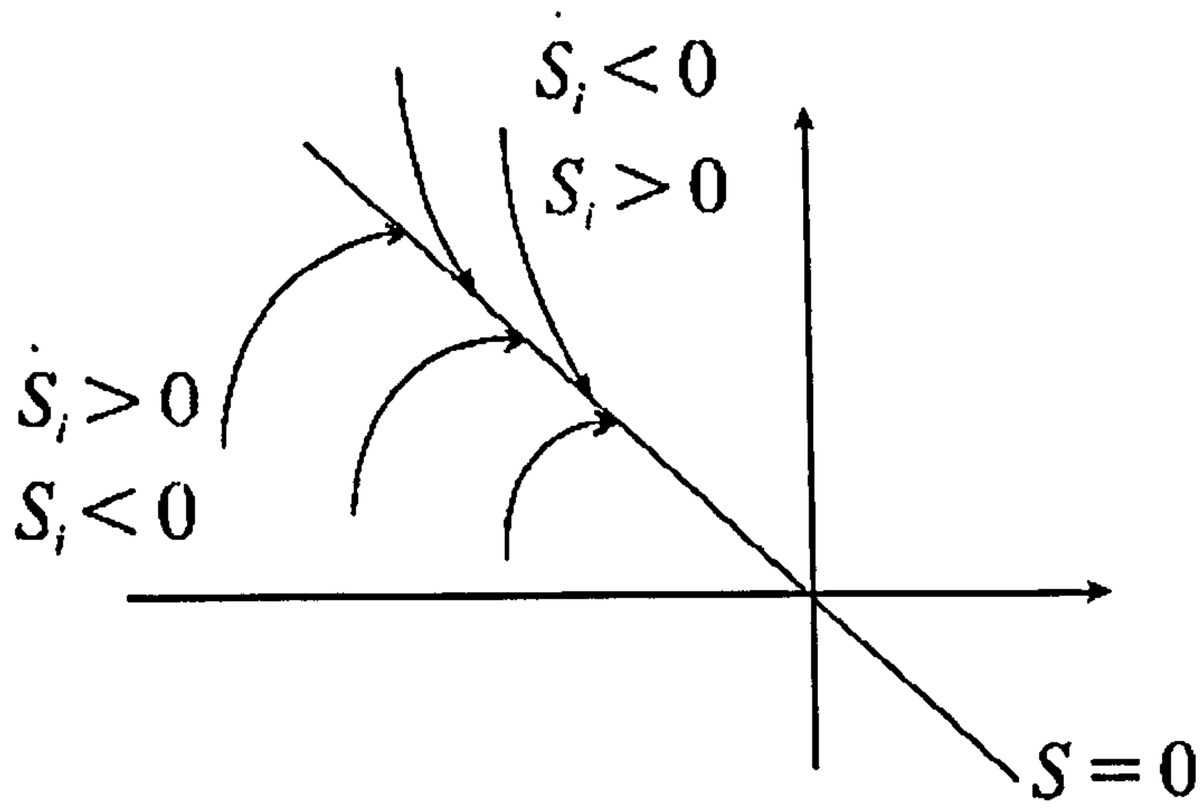


Figure 13: Sliding Condition

4.2.3 Chattering Reduction

Chattering is the result of a finite switching frequency of the discontinuous element while in sliding mode, and the control effort oscillates at this frequency. Such a phenomenon is undesirable and potentially dangerous for the plant as it may excite some resonant frequencies in the plant under control [73], [74], [38].

The problem of chattering has been addressed by a number of researchers [86], [80], [103], [106], [73]. Most of the techniques used to reduce chattering are based on a modification of the $sgn\,(S)$ function. It usually involves a continuous approximation of the discontinuous function in the neighbourhood of the switching surface, when $S \rightarrow 0$.

Slotine suggested replacing the sgn function by the saturation function, equation (47). The new function defines a boundary layer around the sliding surface.

$$U\,(S) = Sat\,(S) \tag{47}$$

where :

$$Sat\,(S) = \begin{cases} +1 & \text{for } S > \Phi \\ S/\Phi & \text{for } |S| < \Phi \\ -1 & \text{for } S < -\Phi \end{cases} \tag{48}$$

with $\Phi > 0$.

Φ is the threshold value for entering the boundary layer. Outside the boundary layer, the control is identical to the ideal relay characteristic. Within the boundary layer however, the control is a high-gain, linear control. The use of such a function alleviates the chattering phenomenon, but it is obtained at the price of a reduction in robustness. The system states do not converge anymore to the origin of the state space but to its neighbouring area, and the size of this attraction surface is defined by Φ. K. S. Yeung *et al.* [103], proposed reducing the width of the boundary layer as it approaches the origin of the phase plane by making Φ proportional to the error, equation (49).

$$\Phi = a + \frac{|E|}{b} \tag{49}$$

A different approach to the chattering problem was suggested by Zinober et al. [107], which puts a low pass filter at the output of the controller to reduce or eliminate the chattering. Again chattering reduction is obtained at the expense of the robustness.

In all the mentioned techniques, chattering is tackled after it appears on the system, instead of tackling it at the source. In the following chapter, a technique to reduce the chattering will be proposed and applied on a robot controller.

4.2.4 Existing Applications of VSC

Since the acceptance of the theory of VSC and sliding mode theory in the Western world, the number of reported applications has kept growing in various areas (mobile robot guidance [37], servomechanisms [18] [32], furnace temperature control [88], teleoperated systems [94], helicopter flight regulation [79], chemical process regulation [78], control of electrically driven vehicle [99] and others). Although variable structure control has not been systematically used as a universal controller, over the years, it has been successfully implemented on various types of processes, both linear and non-linear. Whatever the application, the controller is based on invariant sliding manifolds for the states. When the system states reach the manifolds, the

system behaviour becomes insensitive to disturbances and is entirely dictated by the manifolds.

4.3 Variable Structure Controllers for Robot Manipulators

The earliest reported application of VSC with sliding mode to robot manipulator control is due to Young [104]. The robot manipulator was modelled using an analogue computer. The control law used was switched between two values, one positive and the other negative, equation (50). The sliding surface was designed to give point to point regulation, equation (51). The control law was designed using the hierarchical switching order [98], in which the sliding mode occurs earlier on switching planes of higher orders. The manifolds are nested and sliding mode on each manifold appears in a predefined order, the occurrence of sliding mode is guaranteed for every sliding surface as well as the order of appearance on them. The speed of response is slow however since sliding mode must occur in a specific order. Further, the coupling between manifolds implied by this "sliding hierarchy", makes the control law design complex. Another limitation of the proposed scheme is that very little knowledge of the system is used to established the control law, which inevitably leads to high gain control law.

The control law used by Young is :

$$T_i = \begin{cases} T_i^+ \text{ if } S_i(\tilde{\theta}_i, \dot{\theta}_i) > 0 \\ T_i^- \text{ if } S_i(\tilde{\theta}_i, \dot{\theta}_i) < 0 \end{cases} \tag{50}$$

and the sliding surface used for set point regulation :

$$S_i(\tilde{\theta}_i, \dot{\theta}_i) = \lambda_i \tilde{\theta}_i + \dot{\theta}_i \tag{51}$$

The control law used by Young, equation (50), has been shown to be effective for set point regulation of robot manipulators. The control effort is however unnecessarily large which generates a high level of chattering. Real application of a such scheme is thus not possible as the chattering may excite unmodelled high-frequency dynamics of the robot and damage it. In addition, this controller does not exploit

the physics of the robot and is therefore less effective than controllers that do use such information.

Intuitively if one can use some knowledge of the system in the controller to improve the control effort and use a VSC algorithm to compensate for the unknown part, the switching law can be reduced which would at the same time reduce the chattering amplitude. This approach was used by Slotine *et al.* [86], [81] for the tracking control of a robot manipulator. In the design of the control law, Slotine considers the sliding manifolds to be independent, hence sliding mode can occur on the manifold in a free order as opposed to the hierarchical approach. The complex and nested control laws imposed by the hierarchical control to achieve sliding mode in a specific order is then avoided. The speed of response is increased as sliding modes may occur in parallel on the different switching surfaces.

The controller uses a general model based structure, with switching terms to guarantee the sliding mode and to account for the uncertainties of the model used, equation (53). The sliding surface is selected to achieve tracking error, equation (52).

$$S = (\dot{\theta} - \dot{\theta}_d) + \lambda(\theta - \theta_d) = \dot{\tilde{\theta}} + \lambda \tilde{\theta} \tag{52}$$

$$Ta = -\hat{M} \cdot \ddot{\theta}_r - \hat{C}(\theta, \dot{\theta}) \dot{\theta}_r + \hat{G}(\theta) + Ts \tag{53}$$

$\dot{\theta}_r$ represents a pseudo velocity reference, formed from the desired velocity and the position error, equation (54).

$$\dot{\theta}_r = \dot{\theta}_d - \lambda \tilde{\theta} \tag{54}$$

$$S = \dot{\theta} - \dot{\theta}_r \tag{55}$$

The term Ts is a column vector which is selected so that the sliding condition for each discontinuous surface is guaranteed, equation (56).

$$\dot{S}_i \, . S_i < 0 \tag{56}$$

Using the computed torque equation (53) and the sliding condition equation (56), Slotine established that control can be achieved using :

$$Ts_i = K_i . sgn(S_i) \tag{57}$$

where K_i is a constant term which bounds the system mismatches and uncertainties of the ith control torque term, and guarantees sliding mode.

The scheme described by Slotine, presents a number of improvements and advantages over the controller due to Young. Firstly, Slotine control algorithm is not based on hierarchical control, which means that the sliding manifolds are independent, and so the control law is simpler. The second improvement is the use of an estimate of the system inverse dynamics as feedforward terms for the controller (model-based structure). This leads to improved performance over the scheme due to Young. Further, the exploitation of the Lagrange-Euler equations provide a reduction in chattering as the system uncertainties are reduced and some of the joint coupling is accommodated in the model-based feedforward term. In its initial form however, the Slotine control algorithm equation (53) can not be practically implemented on a robot manipulator because of the strict discontinuous term and the relatively large magnitudes of K_i, equation (57). Even though, the model based feedforward term has allowed a reduction in the values of K_i, its magnitude to guarantee that the system reaches the sliding surface and remains on it, is still large, which results in unacceptable chattering. Slotine overcomes the problem by using a saturation like function, equation (48), instead of the *sgn* one. This defines a boundary layer around the sliding surface, so exact sliding mode does not occur anymore, but instead the system remains within the vicinity of the sliding surface and converges towards the origin of the plane.

In the scheme proposed by Slotine, the controller has to be derived by considering the sliding condition, equation (56) for each sliding surface. The analysis of sliding conditions for each sliding surface is an involved process as it results in coupled sets

of equations, and significant assumptions have to be made to establish the control law. Further, an estimate of the inverse inertia matrix is required. Bailey *et al.* [12] propose to use a Lyapunov function to carry out the controller design. The candidate Lyapunov function is selected so that it characterises the system tracking performance. The control laws are then selected so that the negative definiteness of the derivative of the Lyapunov function is guaranteed. Bailey *et al.* define a sliding variable for each joint of the robot manipulator, in a similar way as Slotine, equation (52). The candidate Lyapunov function forms a quadratic equation using this sliding variable S and the inertia matrix M of the robot manipulator, equation (58).

$$V = \frac{1}{2} S^T . M . S \tag{58}$$

Since S may be viewed as the weighted sum of the position and velocity errors, and the inertia matrix M is positive definite, the function satisfies the requirement of a Lyapunov function and at the same time characterises the tracking error for the system. The proposed control law is of the form of a computed like torque, similar to Slotine, equation (53). The uncertainties are bounded by considering their sum to be bounded. This means that the sliding mode on each individual sliding surface is not guaranteed but only at the intersection of all sliding surfaces. By over estimating these bounds however, sliding mode on individual discontinuous surfaces will eventually occur in a free order manner.

Such an approach has the advantage over that proposed by Slotine by being more general, and of having the design conducted at the same time as the stability of the system being established. Also no inverse inertia matrix estimation is needed. The chattering problem is not tackled in the scheme proposed by Bailey *et al* as only simulation work is presented. In a real application, the finite switching problem would need to be addressed.

K. S. Yeung *et al.* [103] re-used the design approach proposed by Bailey *et al.*, a candidate Lyapunov function of the form of a pseudo-energy function is selected to design the control law as well as establishing the system stability, equation (58). The controller proposed by K. S. Yeung *et al.* is only concerned with point to point

regulation of the robot manipulator. In the proposed scheme, n sliding surfaces are selected for the n joints of the manipulator, where each sliding surface is defined as the weighted sum of the joint velocity and the joint position error, equation (59). In essence, the proposed scheme is close to the one proposed earlier by Bailey *et al.*, but Yeung *et al.* address the problem of chattering in the design of the control laws.

$$S_i = \dot{\theta}_i + \lambda(\theta_i - \theta_{di}) \tag{59}$$

The chattering problem during transient responses is tackled using a sliding sector. The system state is forced towards the sliding sector, and after entering it the system behaviour is maintained in it and converges to the origin of the phase plane. Chattering at the regulation point (origin of the phase plane) is dealt with by a saturation function. Yi-Feng Chen *et al.* [19] extend the scheme of K. S. Yeung *et al.* [103] to trajectory control of a robotic arm. The problem with the scheme due to Yeung *et al.* and its extended version by Chen *et al.* is that both need an estimate of the derivative of each term of the inertia matrix to establish the control law. Such requirements impose tedious calculations and result in a complex control law. Even though the scheme is an improvement over the algorithm due to Bailey *et al.*, it is not paragon.

Gorez *et al.* [36] proposes a controller that uses sliding mode and variable structure control as an initial design start. The design relies on a Lyapunov function, that represents a pseudo-kinetic energy function similar to the one used by Bailey *et al.* equation (58). A sliding manifold for each joint is defined as a nonlinear function of the position error, equation (60). The sliding surface is only concerned with achieving point-to-point regulation. Such definition smooths the sliding manifold as the position approaches to the desired position (i.e. : $\tilde{\theta}_i = 0$), which results in slow dynamics far from the desired position and faster dynamics when approaching this position. The sliding manifold defines a first order system with a varying time constant [36]. With such a definition of the sliding manifold the steady state error at the regulation point can be kept very low.

$$S_i = \dot{\theta}_i + \dot{\theta}_{i,m} .sat(\frac{\tilde{\theta}_i}{\varepsilon_i}) \quad (60)$$

$$S = \dot{\theta} + v(\tilde{\theta}) = \dot{\theta} - \dot{\theta}_r \quad (61)$$

where $\dot{\theta}_i$ is the velocity of the ith joint, $\dot{\theta}_{i,m}$ denotes the upper bound on the magnitude of the velocity of the ith joint, which is a (positive) design parameter. The term ε_i is a positive constant design parameter and $\tilde{\theta}_i$ represents the position error of the ith joint. $v(\tilde{\theta})$ and $-\dot{\theta}_r$ represent a fictitious velocity reference that is a function of the position error as well as the maximum velocity magnitude.

The bounds on the system uncertainties are established by considering each term from a practical point of view. Such an approach diverges greatly from previous approaches as the finite velocity, maximum reaching position, and maximum motor torque constraints for the robot are considered to establish the bounds for the control law. The idea is further enhanced by the same author [35], where a complete step-by-step approach is proposed. The controller designed initially starts from the sliding manifold definition, equation (60) and a pseudo-kinetic energy function as Lyapunov function as in Bailey *et al.* [12]. New terms are progressively introduced in the Lyapunov function and in the control law. Such an approach underlines the physical meaning of the system bounds and makes the selection of the control actions easier. The resulting controller may be viewed as a model based controller with PID-like auxiliary terms, equation (62).

$$Ta = -\hat{M} .v'(\tilde{\theta}).\dot{\theta} + \hat{G}(\theta) - K_s(S) - K_r(r) \quad (62)$$

where K_s denotes a diagonal matrix with positive entries and which provides a control action proportional to S such that $K_s(S) = K_s.S$. This is similar to a PD-like action with a bound on the P term. The term $K_r(r)$ provides an integral like term, as it is defined as a nondecreasing positive diagonal function, and $\dot{r} = S$.

The first main idea of the scheme proposed by Gorez *et al.* is the use of nonlinear sliding manifolds to vary the system dynamics as the position approaches the desired

position, so that the steady state error at the regulation point is low. The second main point is the design methodology, where the physical nature of the terms to be bound are considered to establish the control law. The overall design results in a computed torque controller with a PID like regulator term. The scheme is only concerned with point to point regulation, and further the sliding variables are used to start the design but the resulting control laws do not guarantee that all sliding modes occur.

4.4 Summary

The theory of Variable Structure Control with sliding mode has been initially developed in the former Soviet Union some forty years ago. The idea is derived from discontinuous control action, such as relay control. The research on the topic was mainly theoretical and research reports were mainly in Russian which may explain the delay in spreading to the Western World. Nowadays, VSC and sliding mode is well established and its applications are found in many different fields (from mobile robot guidance to chemical process regulation and helicopter flight regulation).

Young was the first to use the variable structure control and sliding mode theory to control a robot manipulator. His scheme was crude and the physics of the system were ignored in the control law. Further, the sliding surfaces were nested, which imposed a slow response of the system. Slotine *et al.* proposed having independent sliding surfaces and using the system dynamics in the control law. However, the design of the control law relies on the sliding condition, which requires the analysis of each joint equation and an estimation of the terms of the inverse inertia matrix. Such analysis becomes difficult because of the nested nature of the equations for each joint and the inverse inertia matrix estimation. Bailey *et al.*, propose using a Lyapunov function in the form of a quadratic function, that characterises the system tracking error as a whole. The resulting control laws are similar to those obtained by Slotine, but the approach is more systematic and there is no need to evaluate the inverse inertia matrix. The chattering problem however is not addressed which in a practical application would be a serious issue. Yeung *et al.* and Chen *et al.* extended the scheme of Bailey to respectively point to point regulation and trajectory tracking.

Both address the chattering problem in a similar way, by generating sliding sectors. Their schemes suffer from the need to have estimates of each entry of the inverse of the inertia matrix. Their schemes benefit from the systematic approach of Bailey *et al.* and tackle the chattering phenomenon, but the inverse inertia matrix requirement make the derivation of the control law tedious. Gorez *et al.* propose a control methodology that departs from the earlier Bailey *et al.* proposal. Non linear sliding surfaces are used to vary the system dynamics as it approaches the regulation point. Further, the design is conducted in a step-by-step fashion where the physical nature of the system is considered to establish the bounds and to select the control action. The resulting controller is a PID-like controller with computed torque terms but where sliding mode is no longer guaranteed. The sliding variables defined at the start only represent virtual sliding surfaces. The proposed scheme of Gorez *et al.* is only concerned with point to point regulation, which in many industrial applications is not sufficient.

There have been a large number of other schemes proposed in the literature which have not been mentioned. Only those which present a significant point in the development of the variable structure control with sliding mode applied to robot manipulators has been reported. Further, adaptive versions of controller that used VSC and sliding mode are omitted, since they represent an intermediate class, i.e. robust-adaptive controllers.

5 PROPOSED CONTROLLER FOR ROBOT MANIPULATORS

5.1 Design Procedure

The proposed controller is aimed at achieving trajectory tracking, such that position as well as velocity demand must be tracked by the manipulator as close as possible. The sliding surface is then selected so that it represents the weighted sum of the position error and of the velocity error, equation (63). This definition of the sliding surface is identical to the one used by Slotine *et al.* [86], Bailey *et al.* [12] and Yi-Feng Chen *et al.* [19]. The control laws force the system to reach the sliding surface S, and when it is reached the system dynamics are restrained to this surface and converge to the state where $(\tilde{\theta}, \dot{\tilde{\theta}}) = (0, 0)$. This state is the ideal case, where the position as well as the velocity error are null, and tracking performances are optimum.

$$S = (\dot{\theta} - \dot{\theta}_d) + \lambda(\theta - \theta_d) = \dot{\tilde{\theta}} + \lambda . \tilde{\theta} \tag{63}$$

The time derivative of S is given as :

$$\dot{S} = \ddot{\tilde{\theta}} + \lambda . \dot{\tilde{\theta}} \tag{64}$$

The next step in the design is to select the control laws so that the system behaviour is forced toward the sliding surface and remains on it thereafter. The reaching condition, equation (45), can be used for this effect [86]. The problem with such an approach is that it requires the estimation of the inverse inertia matrix and tedious analysis of the resulting inequalities. The Lyapunov function V of the form of a pseudo-energy function, equation (58), is selected to conduct the derivation of the control laws.

$$V = \frac{1}{2} S^T . M(\theta) . S \tag{65}$$

where S^T is the transpose of the sliding vector S and $M(\theta)$ is the inertia matrix

of the robot manipulator. The time derivative of V is given by :

$$\dot{V} = S^T.M(\theta).\dot{S} + \frac{1}{2}S^T.\dot{M}(\theta).S \tag{66}$$

$$\dot{V} = S^T.M(\theta).(\ddot{\tilde{\theta}} + \lambda.\dot{\tilde{\theta}}) + \frac{1}{2}S^T.\dot{M}(\theta).S \tag{67}$$

$$\dot{V} = S^T.M(\theta).\ddot{\theta} + S^T.M(\theta).(-\ddot{\theta}_d + \lambda.\dot{\tilde{\theta}}) + \frac{1}{2}S^T.\dot{M}(\theta).S \tag{68}$$

Now define :

$$e = (-\dot{\theta}_d + \lambda.\tilde{\theta}) \tag{69}$$

$$\dot{e} = (-\ddot{\theta}_d + \lambda.\dot{\tilde{\theta}}) \tag{70}$$

The terms e and $\dot{e}$ are direct functions of the velocity demand augmented by the weighted position error, and of the acceleration demand augmented by the velocity error respectively.

The inverse dynamics of the robot can be represented as equation (71). The terms $M(\theta)$, $C(\theta,\theta)$, $G(\theta)$, and $F(\theta,\theta)$ represent the inertia matrix, Coriolis/centripetal effect, gravitational torque and friction respectively. The Coriolis/centripetal term is defined so that it verifies the skew symmetric property, equation (16). The friction term is a discontinuous function of the velocity and position, the friction term of each joint is decoupled from each other and is dissipative, so that $\dot{\theta}_i.F_i(\theta,\dot{\theta}) > 0$, for all $\dot{\theta}_i \neq 0$. For robots that use the concept of transmission directness, as in the case of the SPRINTA, it can be assumed that friction is non existent/negligible [100].

$$Ta = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta}).\dot{\theta} + G(\theta) + F(\theta,\dot{\theta}) \tag{71}$$

Then

$$M(\theta)\ddot{\theta} = Ta - C(\theta,\dot{\theta}).\dot{\theta} - G(\theta) - F(\theta,\dot{\theta}) \tag{72}$$

Substituting equation (72) and equation (69) and its derivative into equation

(68), gives:

$$\dot{V}= S^T.\left[Ta - C(\theta,\dot{\theta}).\,\dot{\theta} - G(\theta) - F(\theta,\dot{\theta}) - C(\theta,\dot{\theta}).e + C(\theta,\dot{\theta}).e\right] + \tfrac{1}{2}S^T.\,\dot{M}(\theta).S \tag{73}$$

$$\dot{V}= S^T.\left[Ta - G(\theta) - F(\theta,\dot{\theta}) - C(\theta,\dot{\theta}).S + C(\theta,\dot{\theta}).e + M(\theta).\,\dot{e}\right] + \tfrac{1}{2}S^T.\,\dot{M}(\theta).S \tag{74}$$

$$\dot{V}= S^T.\left[Ta - G(\theta) - F(\theta,\dot{\theta}) + C(\theta,\dot{\theta}).e + M(\theta).\,\dot{e}\right] + S^T.\left[\tfrac{1}{2}.\,\dot{M}(\theta) - C(\theta,\dot{\theta})\right].S \tag{75}$$

Using the skew-symmetry property, equation (16), the expression is reduced to:

$$\dot{V}= S^T.\left[Ta - G(\theta) - F(\theta,\dot{\theta}) + C(\theta,\dot{\theta}).e + M(\theta).\,\dot{e}\right] \tag{76}$$

Letting $Ta = \hat{G}(\theta) + \hat{F}(\theta,\dot{\theta}) - \hat{C}(\theta,\dot{\theta}).e - \hat{M}(\theta).\,\dot{e} + Ts$, and substituting it into equation (76) gives :

$$\dot{V}= S^T.\left[\tilde{G}(\theta) + \tilde{F}(\theta,\dot{\theta}) - \tilde{C}(\theta,\dot{\theta}).e - \tilde{M}(\theta).\,\dot{e} + Ts\right] \tag{77}$$

where :

$\hat{(\,.\,)}$ is the parameter estimate and

$\tilde{(\,.\,)}=\hat{(\,.\,)} - (\,.\,)$, is the parameter error.

The next step in the design procedure is to define Ts so that the negative semi-definiteness of $\dot{V}$ is guaranteed, equation (78). To derive Ts, the physics of the robot manipulator is considered from a practical point of view. Such an approach has been proposed and used by Gorez *et al.* [36], [35]. The idea was initially suggested by Slotine at a more theoretical level [82]. By considering the system from a practical standpoint, some engineering assumptions can be made so that the condition for the Lyapunov function to exist can be established, equation (78), and at the same time the control law can be derived.

$$\dot{V} = S^T . \left[\tilde{G}(\theta) + \tilde{F}(\theta, \dot{\theta}) - \tilde{C}(\theta, \dot{\theta}).e - \tilde{M}(\theta) . \dot{e} + Ts \right] \leqslant 0 \qquad (78)$$

Assumption 1 In the case of rotational joints, the gravitational torque is represented as a trigonometric function (sin or cos) of the joint displacement. Since trigonometric functions are cyclic, they are bounded. For translational joints, the translation is limited by physical constraints, minimum and maximum possible displacement of the joint. In both case, the gravitational torque can be upper bounded, and so can its estimate and the error resulting from the difference between the gravitational torque and the estimate. The gravitational torque error $\tilde{G}_i(\theta)$ for each link can be bounded as in equation (79).

$$\tilde{G}_i(\theta) < K_{g_i} \qquad (79)$$

Assumption 2 The inertia matrix is a function of the position of the joints, with rotational joint each entry of the inertia matrix is a trigonometric function (sin or cos) of the joint position. This implies the boundedness of each of the inertia terms. In the case of translational joints the boundedness on the inertia term is implied by the finite maximum displacement of the joint. Since all the terms of the inertia matrix are bounded, so are the estimates and the error on them.

$$\tilde{M}_{ij}(\theta) < K_{m_{ij}} \qquad (80)$$

Assumption 3 The $C(\theta, \dot{\theta})$ matrix relates the velocity/position to torque. This matrix depends on the positions and velocities of the joints. The part which is a function of the joint position is bounded, the justification is identical to the one used in assumptions 1 & 2. For a robot manipulator, the velocity of the joint is bounded by the maximum velocity achievable by the motor or by the safety limits that are imposed. It can be concluded that the matrix $C(\theta, \dot{\theta})$ is bounded as is the estimate. It can then be said that the error $\tilde{C}(\theta, \dot{\theta})$ is also bounded.

$$\tilde{C}_{ij}\,(\theta,\dot{\theta}) < K_{c_{ij}} \tag{81}$$

Assumption 4 The $F(\theta,\dot{\theta})$ term is a non linear function of the velocity and position. Past the stiction point or Coulomb friction level however, this term becomes a direct function of the velocity (viscous friction). Since the velocity of the joint can be assumed to be bounded, it can be assumed that each friction term is bounded and so are its estimate and the error on the estimate.

$$\tilde{F}_i\,(\theta,\dot{\theta}) < K_{f_i} \tag{82}$$

Assumption 5 The function $e = (-\dot{\theta}_d + \lambda.\tilde{\theta})$, represents a velocity demand function, which is augmented by the weighted position error. The velocity demand is generated by software according to some predefined profiles. An upper limit of the velocity demand is then established. The position error is assumed to remain within a bounded set during the operation of the manipulator (such assumption is very weak, but simulation reveals it is plausible). The function e can be considered to be upper bounded.

$$e_i < K_{e_i} \tag{83}$$

Assumption 6 For $\dot{e} = (-\ddot{\theta}_d + \lambda.\dot{\tilde{\theta}})$, a similar argument as for e may be used. $\dot{e}$ represents an acceleration function augmented by the weighted velocity error. The demand acceleration is bounded by predefined acceleration profiles used to generate the demand trajectory. The velocity error is considered to remain within a bounded set during operation (such assumption is very weak, but simulation however reveals it is plausible). It can be assumed that the function $\dot{e}$ is bounded by a constant.

$$\dot{e}_i < K_{\dot{e}_i} \tag{84}$$

Using all the previous assumptions, each row of the inverse dynamics error ($\tilde{G}(\theta) - \tilde{C}(\theta,\dot{\theta})e - \tilde{M}(\theta).\dot{e}$) can be bounded by a constant coefficient K_i, equation (85).

$$\left|\tilde{G}(\theta) - \tilde{C}(\theta,\dot{\theta})e - \tilde{M}(\theta).\dot{e}\right|_i < K_i \tag{85}$$

With a bound on the system error, the negative definiteness of $\dot{V}$ can be guaranteed by taking $Ts_i = -K_i.sign(S_i)$. The initial controller can be expressed as in equation (86), its structure is identical to the one used by Slotine *et al.* and Bailey *et al.*, figure 14.

$$Ta = -\hat{M}(\theta).\dot{e} - \hat{C}(\theta,\dot{\theta}).e + \hat{G}(\theta) - K.sgn(S) \tag{86}$$

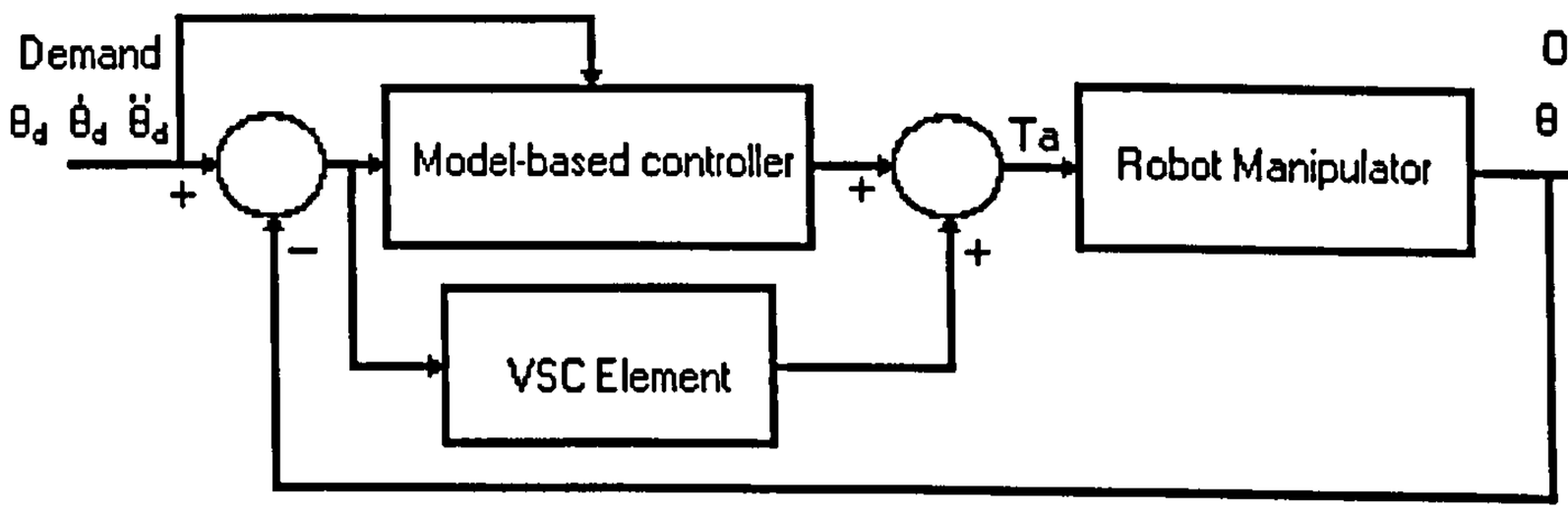


Figure 14: Structure of the Variable Structure Controller

5.1.1 Simulations

The initial controller algorithm is tested by simulation on a planar 2 degrees of freedom robot manipulator. The robot kinetic terms are identical to the ones used by Asada *et al.* [7, p. 229] and Gao *et al.* [33], and its dynamic equations are given in Appendix C. The robot dynamics as well as the controller has been coded in C. A fourth order Runge-Kutta integration algorithm has been used. The integration is performed with an incrementation time of 0.00001 sec., while the torque is updated

at a slower rate of 0.0002 sec . to have a realistic effect of the chattering phenomenon. The trajectories used for the simulation are piecewise differentiable and shown in figures 15-17.

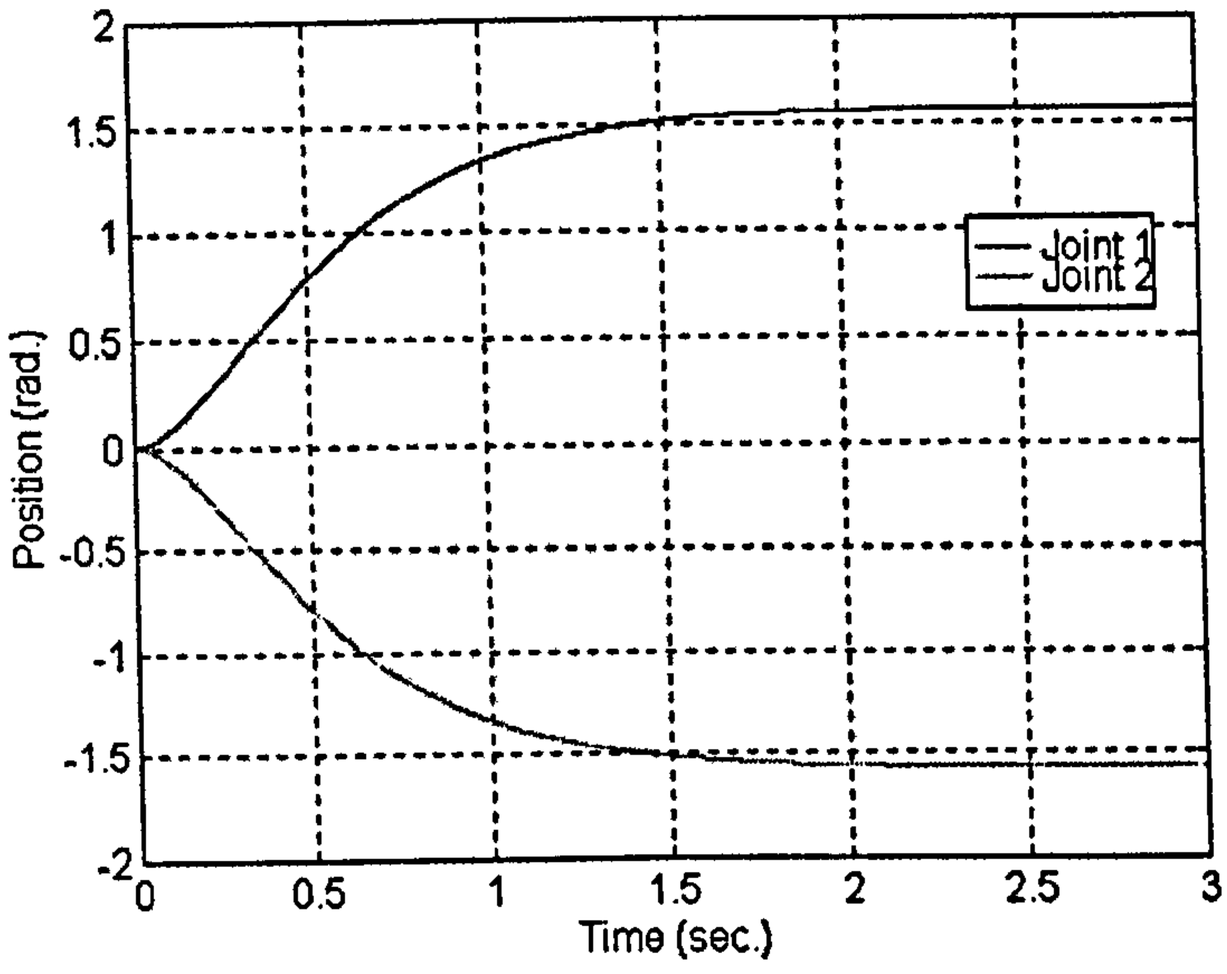


Figure 15: Position Demand Trajectories

The controller is of the form of equation (86), and an error of 20% has been introduced on the kinetic parameters used to establish the model-based controller part, and to demonstrate the effectiveness of the VSC scheme. The simulation results are presented in figures 18-22. The convergence to zero of the tracking error is observed from figure 20 and figure 21. When the sliding manifold is reached, the system behaviour is entirely dictated by it and slides along it until the origin $(0, 0)$ is reached. Such a result is achieved however at the cost of a substantial amount of chattering on the applied torque, figure 22. For practical implementation, this amount of chattering is unacceptable as it would inevitably damage the robot manipulator. The technique traditionally used to reduce the chattering is to replace the discontinuous *sgn* function by a continuous one with saturation like modes, as mentioned previously. Such an approach is very general and does not consider the detail of the problem. By reconsidering the nature of the uncertainties on the model based controller, refinement and technically sound solutions can be obtained to greatly reduce the chattering.

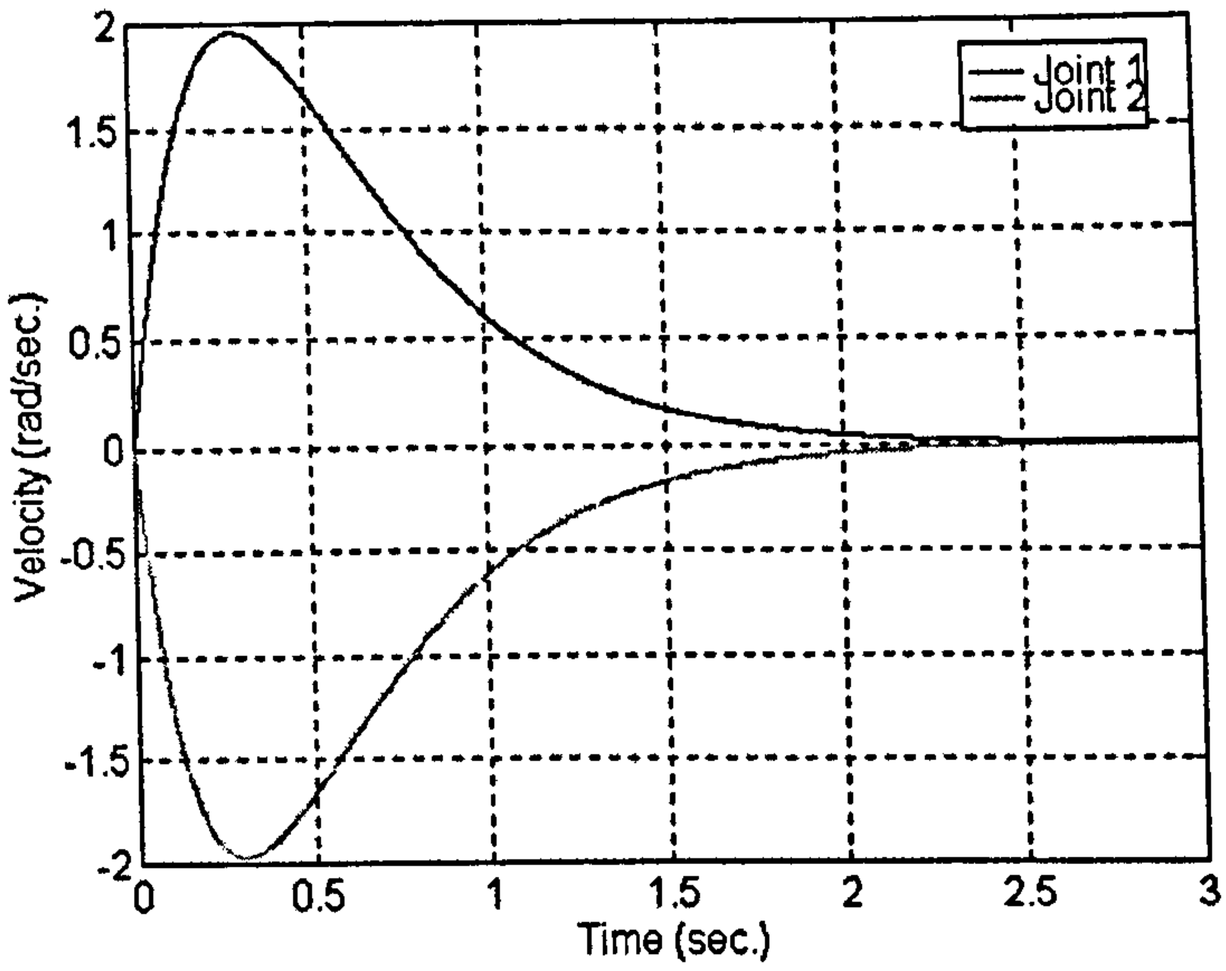


Figure 16: Velocity Demand Trajectories

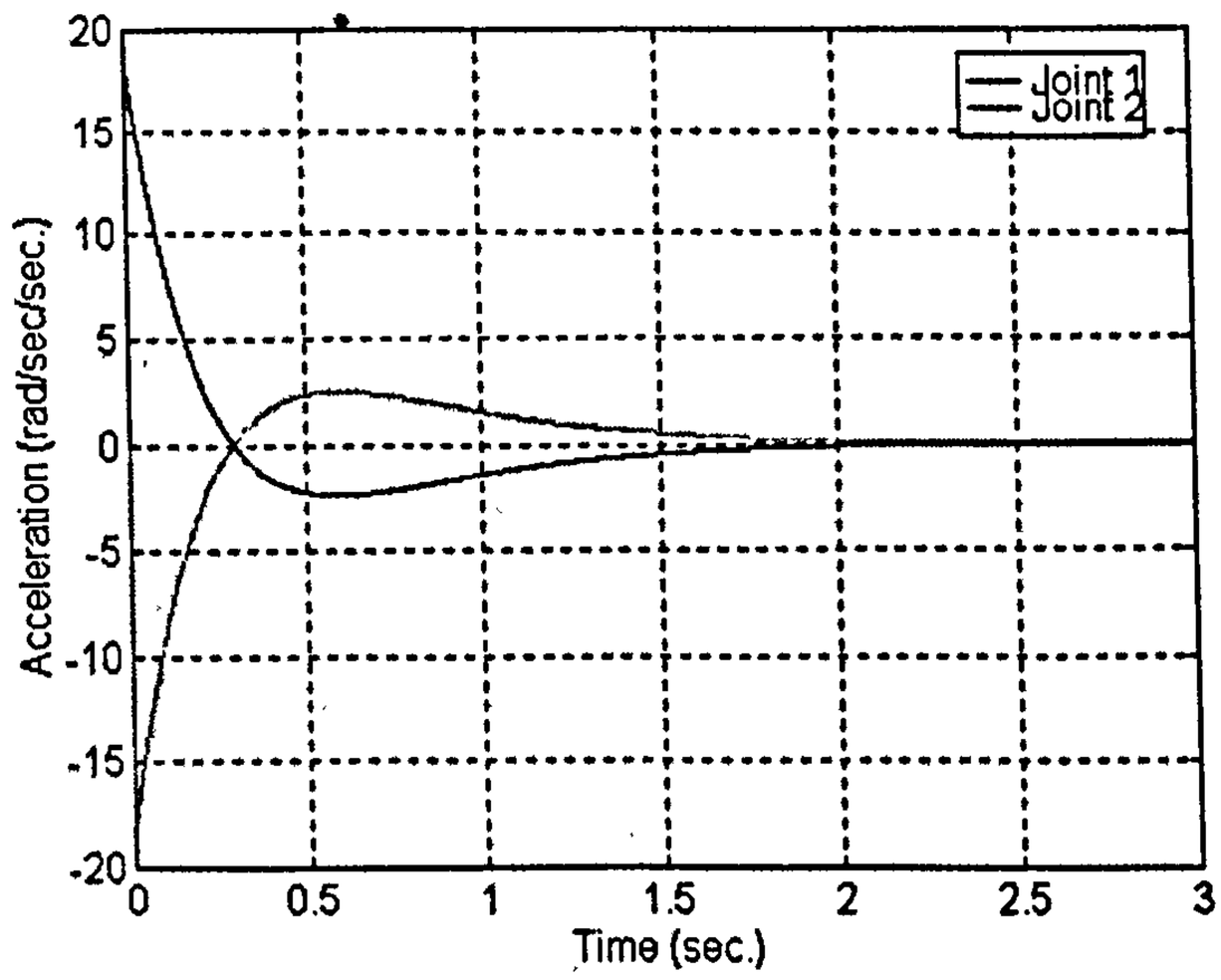


Figure 17: Acceleration Demand Trajectories

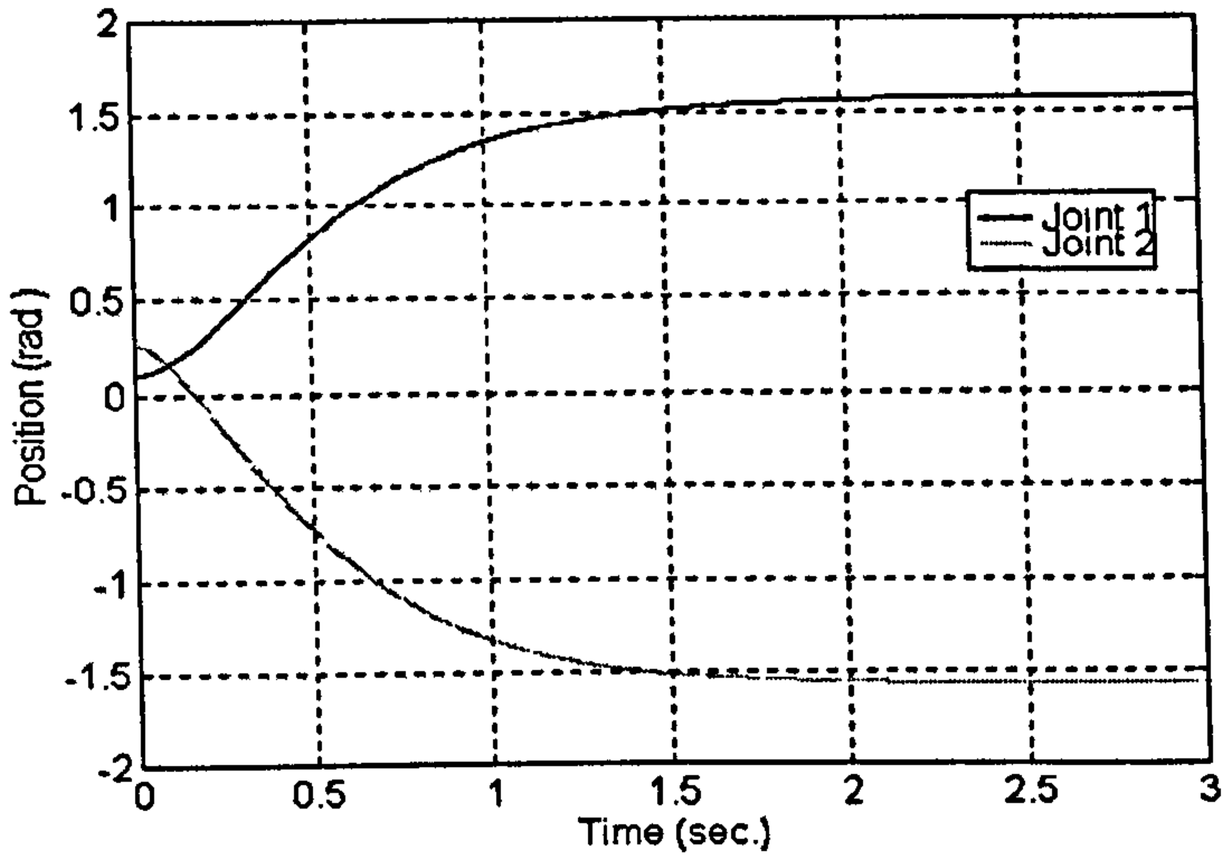


Figure 18: Joint Positions

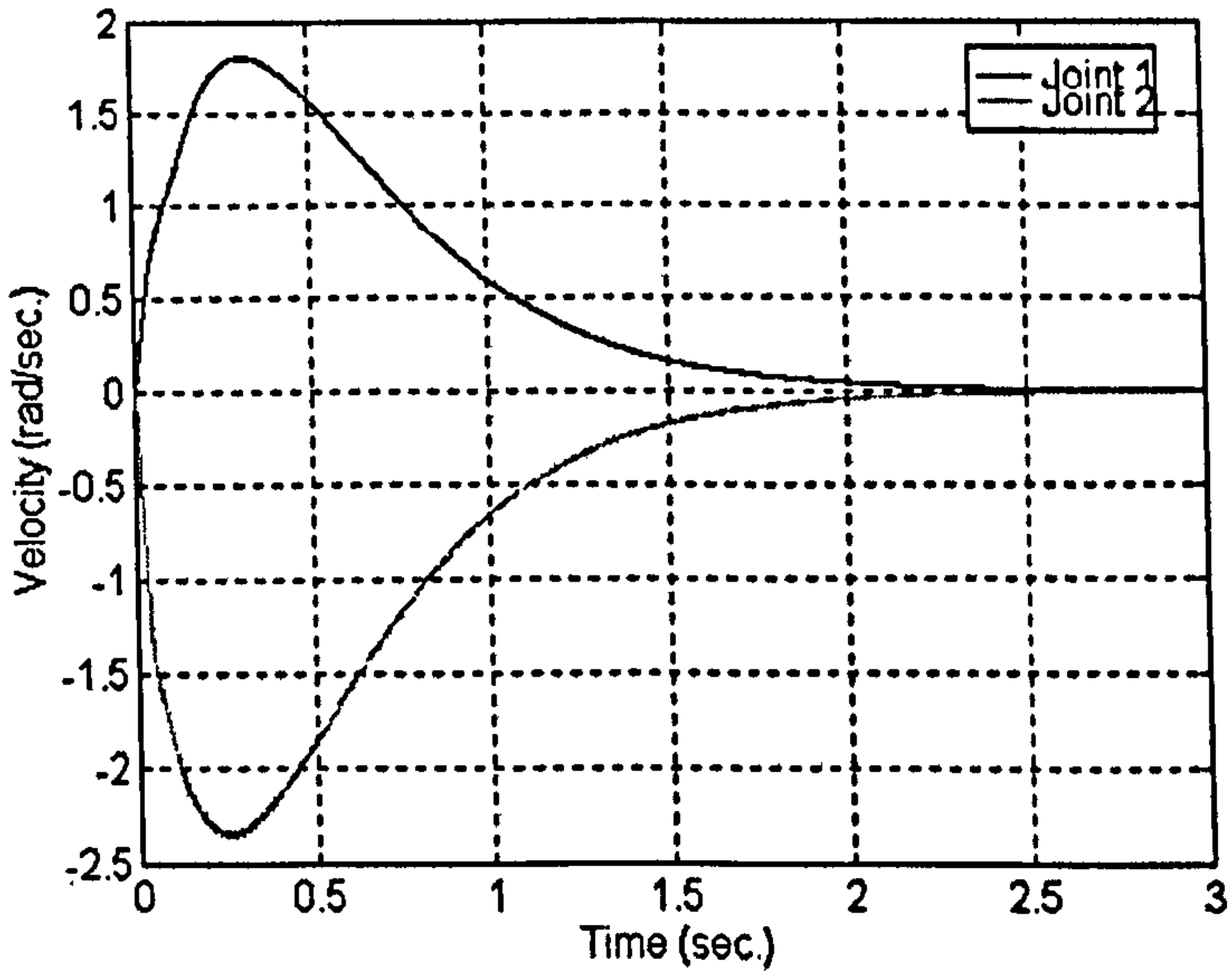


Figure 19: Joint Velocities

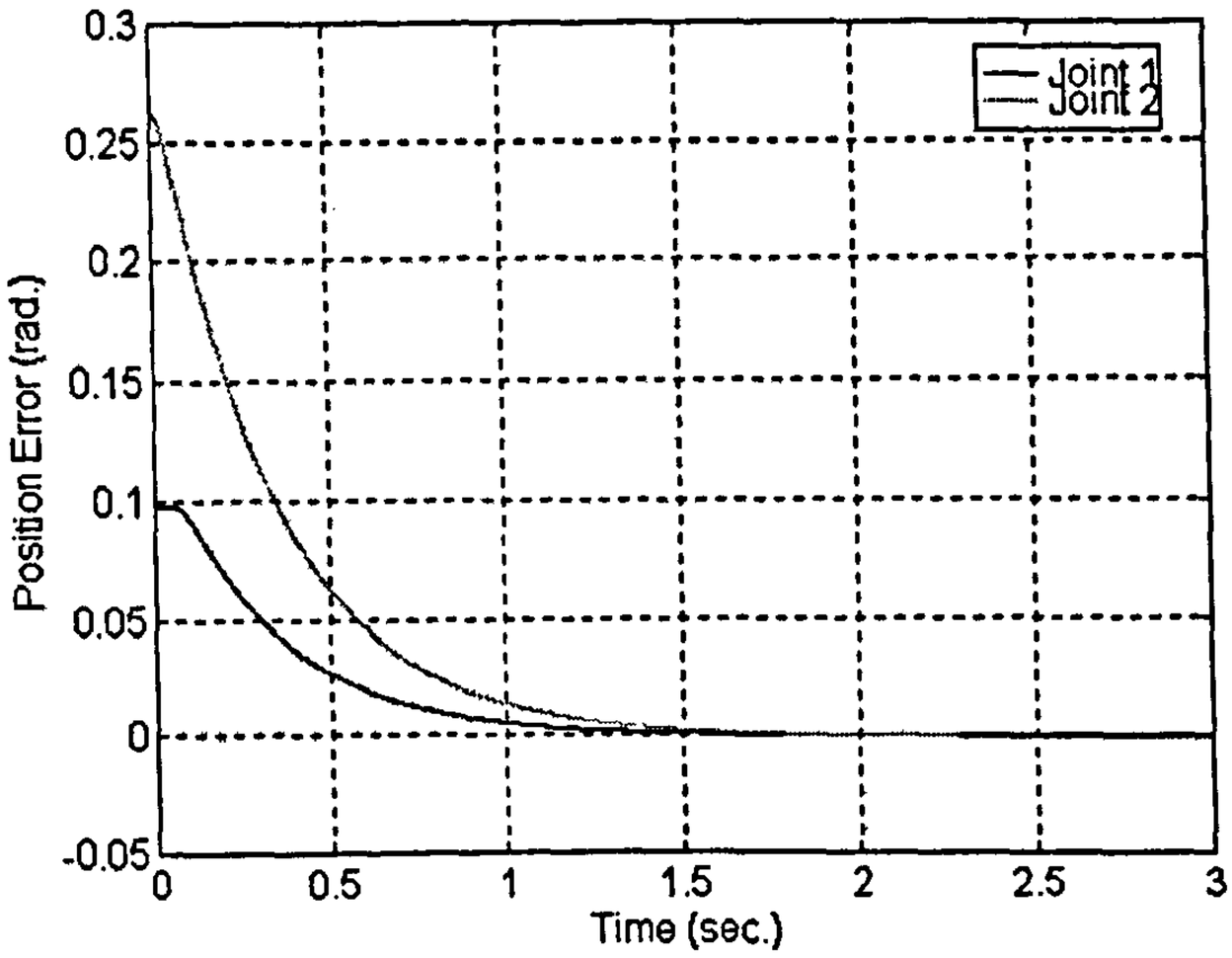


Figure 20: Position Errors at the Joints

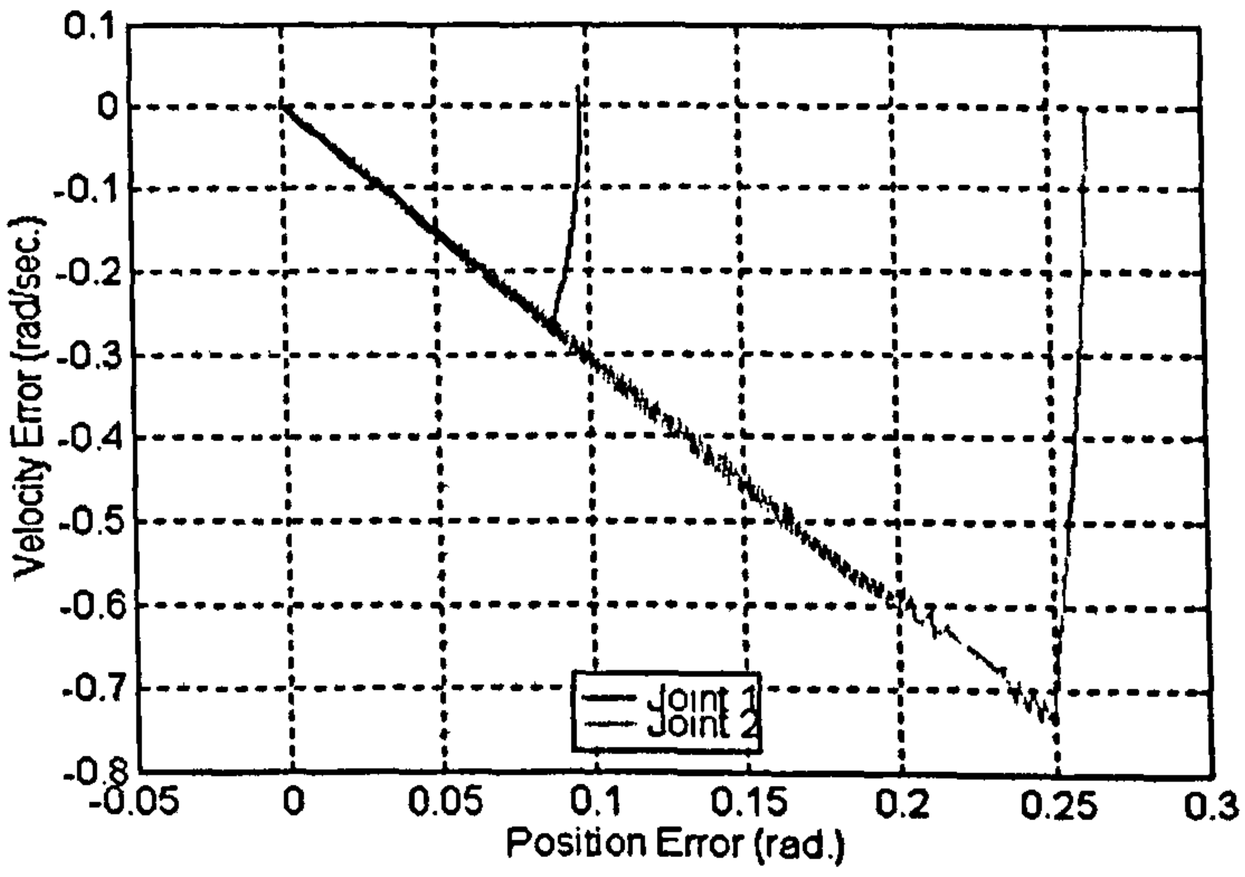


Figure 21: State Space Representation of the Trajectory Errors

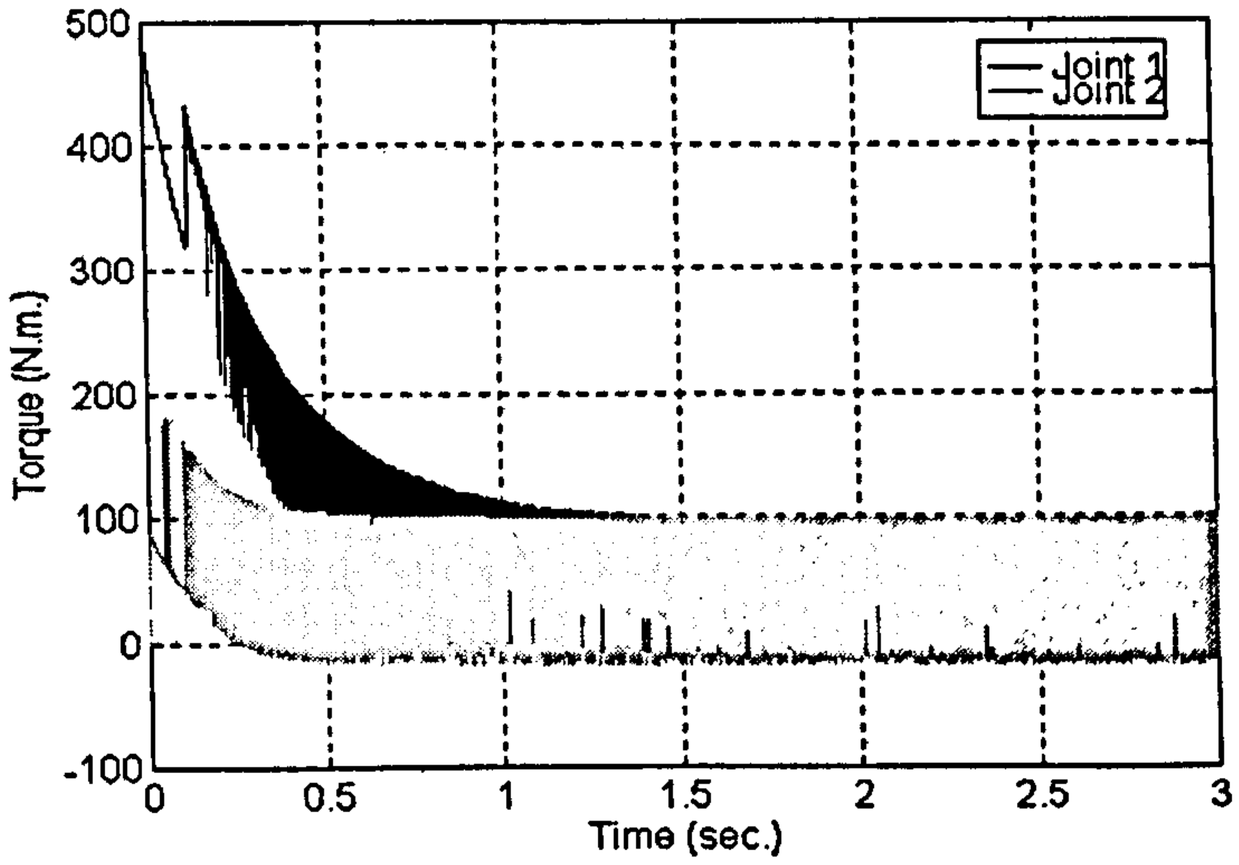


Figure 22: Applied Torque for the Joint Trajectory Demands

5.2 Design Re-Evaluation

The control laws have been determined by guaranteeing the negativeness of the derivative of the Lyapunov function, equation (78). The conditions for the Lyapunov derivative have been established by bounding the uncertainties with a fixed term K, equation (85). Even though the assumptions made to obtain this result are technically sound, the bounds of the unknown parameters are selected conservatively and their values are taken as being fixed. By re-examining assumption 5 and assumption 6, major improvement can be made on the establishment of the bounds for the terms e and $\dot{e}$.

Assumption 5' The function $e = (-\dot{\theta}_d + \lambda.\tilde{\theta})$, represents a velocity demand function, which is augmented by the weighted position error. As the typical velocity profile with upper and lower bounds are used to generate the velocity demand, the latter can be upper bounded by a term of fixed magnitude. The next term to be bounded in this function is the position error. Previously it is assumed that the position error remains within a certain bounded set. Such an assumption is very weak even though simulations as well as experiments has shown its validity. Further, to bound the position error the term used must have a large magnitude, which leads to excessive chattering. A natural choice

to bound the position error is to make the bound a function of the position error. In the simplest case, the bound is a linear function of the position error. The function e can be bounded as follow :

$$e_i < K_{e_i} + K_{1i}.\left|\tilde{\theta}_i\right| \tag{87}$$

where K_{e_i} and K_{1i} are constant positive terms.

Assumption 6' The assumption 6 for $\dot{e} = (-\ddot{\theta}_d + \lambda.\dot{\tilde{\theta}})$, is similarly revised. $\dot{e}$ represents an acceleration function augmented by the weighted velocity error. The demand acceleration is bounded by the predefined acceleration profile used to generate the demand trajectory. The velocity error can be bounded by a function of itself, instead of a constant term. The bound for the function $\dot{e}$ can then be taken as :

$$\dot{e}_i < K_{\dot{e}_i} + K_{2i}.\left|\dot{\tilde{\theta}}_i\right| \tag{88}$$

where $K_{\dot{e}_i}$ and K_{2i} are constant positive terms.

The inverse dynamic error $(\tilde{G}(\theta) - \tilde{C}(\theta,\dot{\theta})e - \tilde{M}(\theta).\dot{e})$ can then be bounded as

$$\left|\tilde{G}(\theta) - \tilde{C}(\theta,\dot{\theta})e - \tilde{M}(\theta).\dot{e}\right|_i < K_i + K_{1i}.\left|\tilde{\theta}_i\right| + K_{2i}.\left|\dot{\tilde{\theta}}_i\right| . \tag{89}$$

where K_i is the positive sum of all the constant bounds, and K_{1i} and K_{2i} are fixed positive terms. Define :

$$K_i(\tilde{\theta}_i,\dot{\tilde{\theta}}_i) = K_i + K_{1i}.\left|\tilde{\theta}_i\right| + K_{2i}.\left|\dot{\tilde{\theta}}_i\right| \tag{90}$$

The negativeness of $\dot{V}$ can be guaranteed by taking

$$Ts_i = -K_i(\tilde{\theta}_i,\dot{\tilde{\theta}}_i).sgn(S_i) \tag{91}$$

With such bounds on the system uncertainties, the chattering is greatly reduced. Chattering is however not completely overcome because of the term K_i, and the magnitude of the chattering is directly proportional to the magnitude of K_i. To overcome this problem, K_i is removed and K_{1i} and K_{2i} are slightly overestimated to compensate for the missing term. Hence the control law is :

$$Ta = -\widehat{M}(\theta)\dot{e} - \widehat{C}(\theta,\dot{\theta})e + \widehat{G}(\theta) - K(\tilde{\theta},\dot{\tilde{\theta}}).sgn(S) \tag{92}$$

where

$$K(\tilde{\theta},\dot{\tilde{\theta}}) = \left[K_1(\tilde{\theta}_1,\dot{\tilde{\theta}}_1), K_2(\tilde{\theta}_2,\dot{\tilde{\theta}}_2), K(\tilde{\theta}_3,\dot{\tilde{\theta}}_3), \ldots\right]^T$$

and

$$K_i(\tilde{\theta}_i,\dot{\tilde{\theta}}_i) = K_{1i}.\left|\tilde{\theta}_i\right| + K_{2i}.\left|\dot{\tilde{\theta}}_i\right| \tag{93}$$

The improved scheme is evaluated on the planar robot manipulator used previously.

5.2.1 Simulations

The 2 degrees of freedom planar manipulator is used to test the modified controller scheme. The same demand trajectories are used with identical integration and torque update intervals. The results are shown in figures 23-27. It appears that the chattering problem is solved using this type of control strategy. The system is still forced toward the sliding surface, and when reached sliding within its vicinity to the origin of the state occurs. The position errors however do not converge to zero exactly but to its neighbourhood, so an offset on the position then occurs. This offset is due to the fact that the boundedness of the uncertainty is no longer guaranteed when $\tilde{\theta}_i$ and $\dot{\tilde{\theta}}_i$ are close to zero since $K_i(\tilde{\theta}_i,\dot{\tilde{\theta}}_i)$ tends zero. To overcome this problem, the control action has to be extended to overcome the offset whilst not introducing chattering.

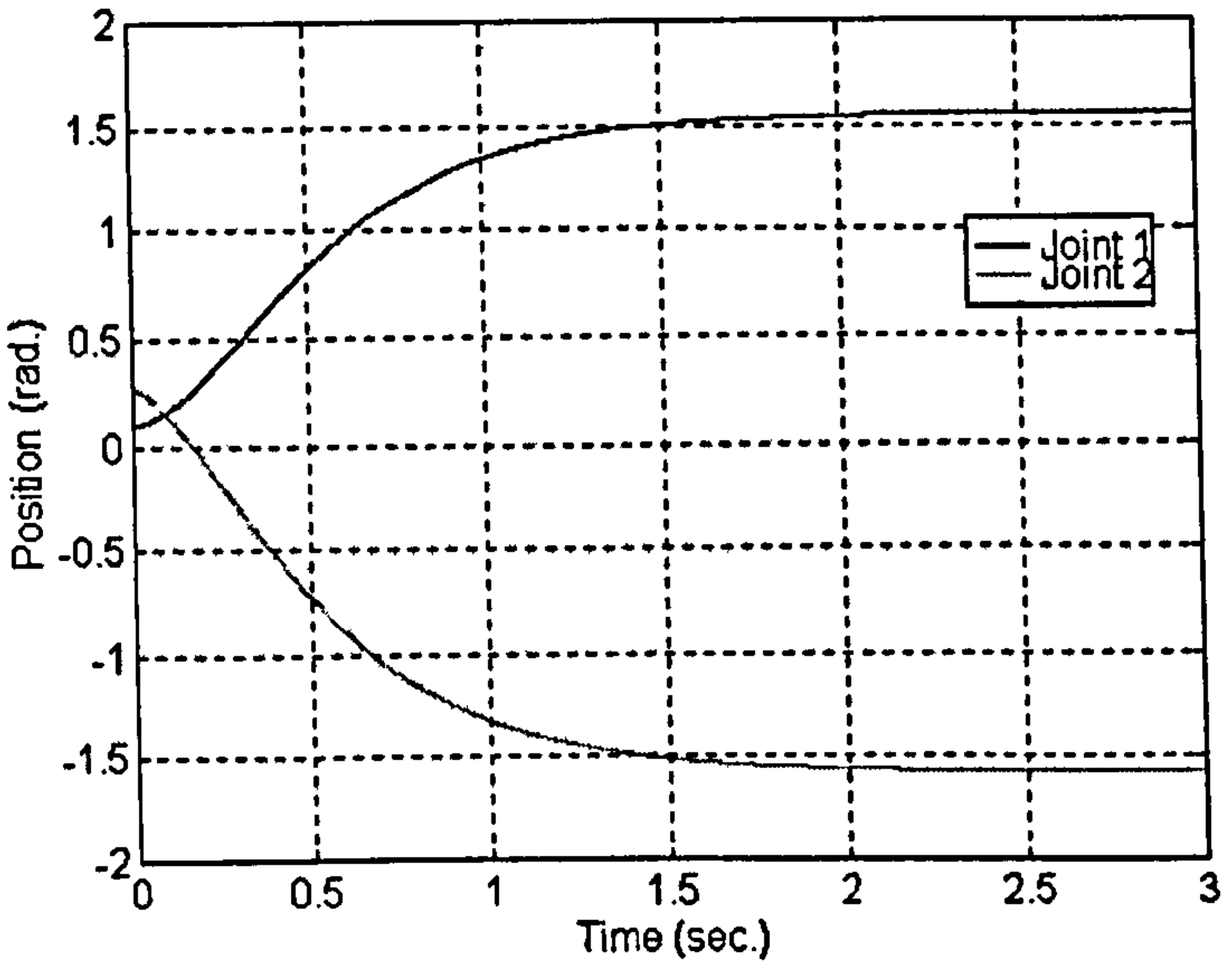


Figure 23: Joint Positions

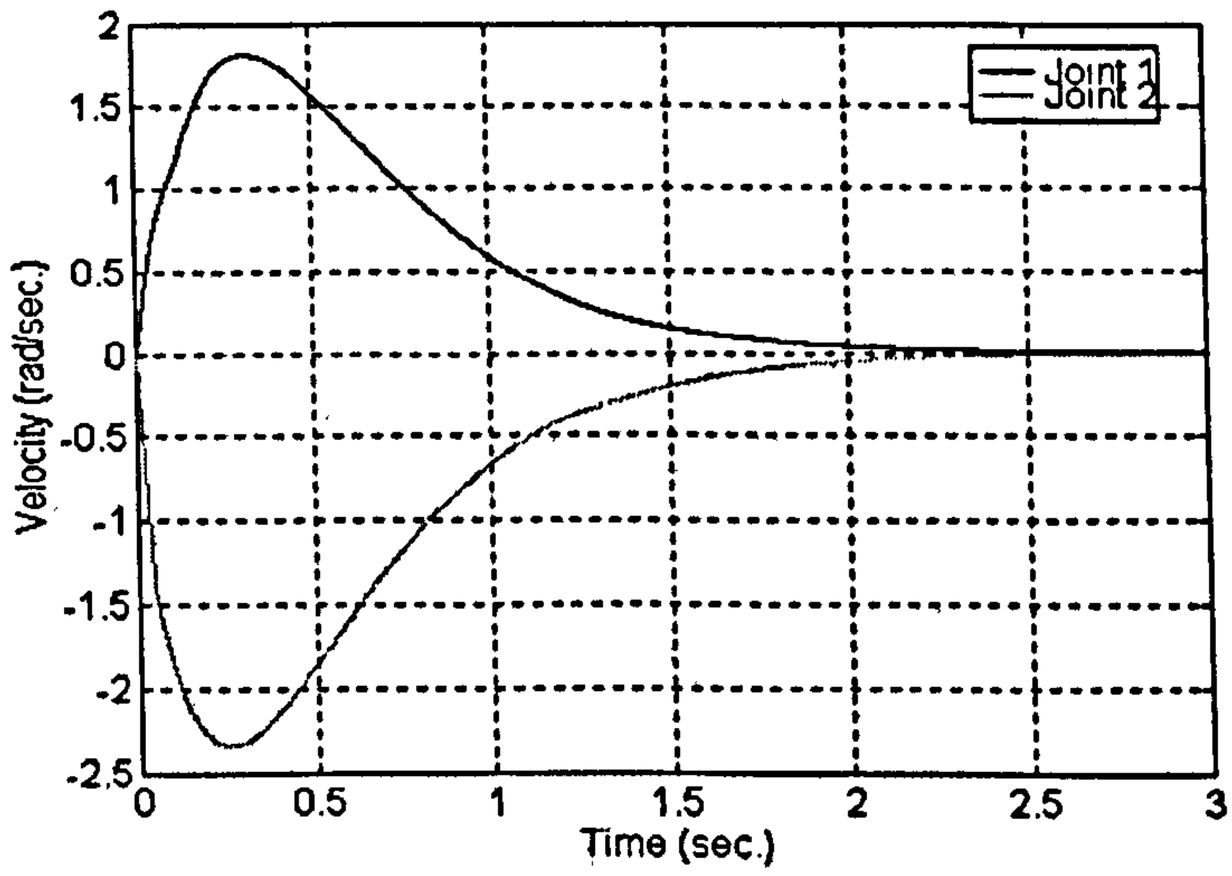


Figure 24: Joint Velocities

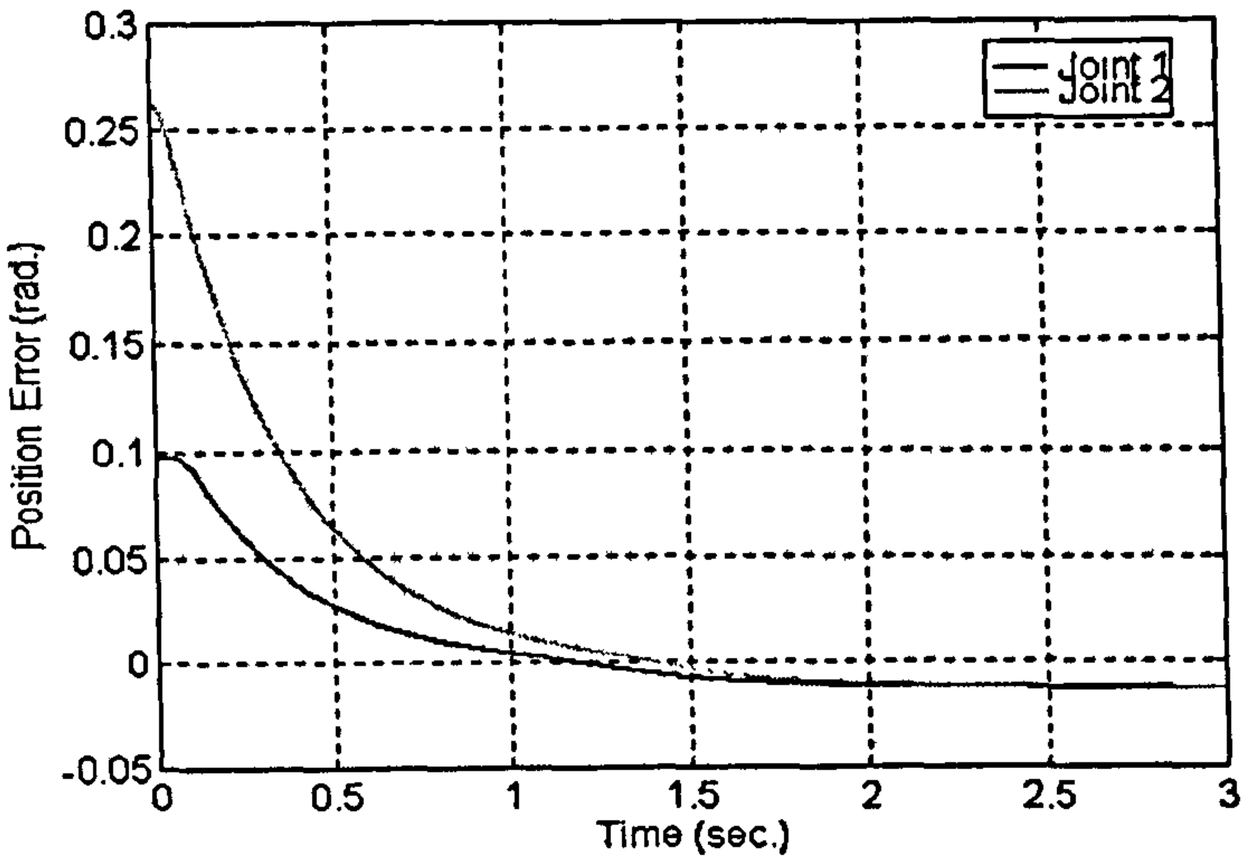


Figure 25: Position Errors at the Joints

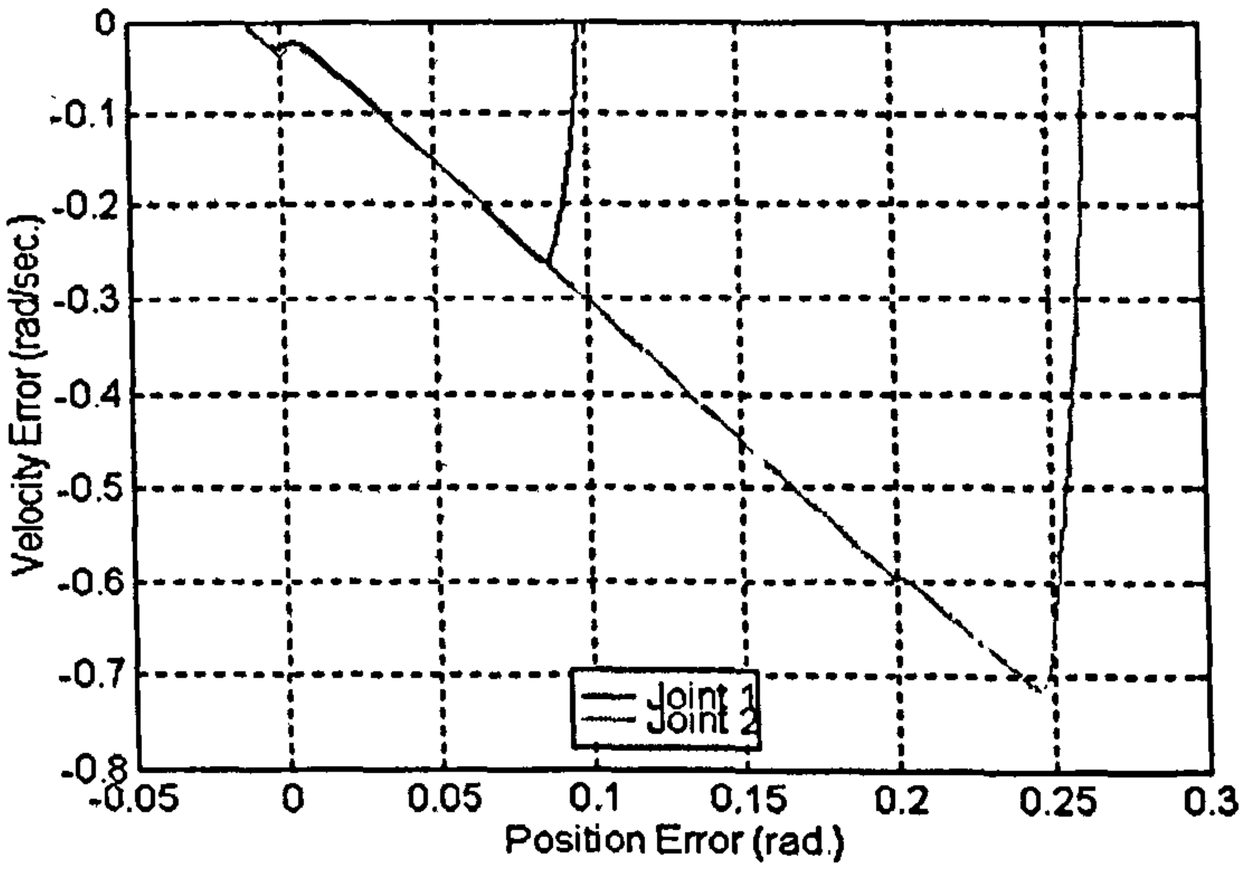


Figure 26: State Space Representation of the Trajectory Errors

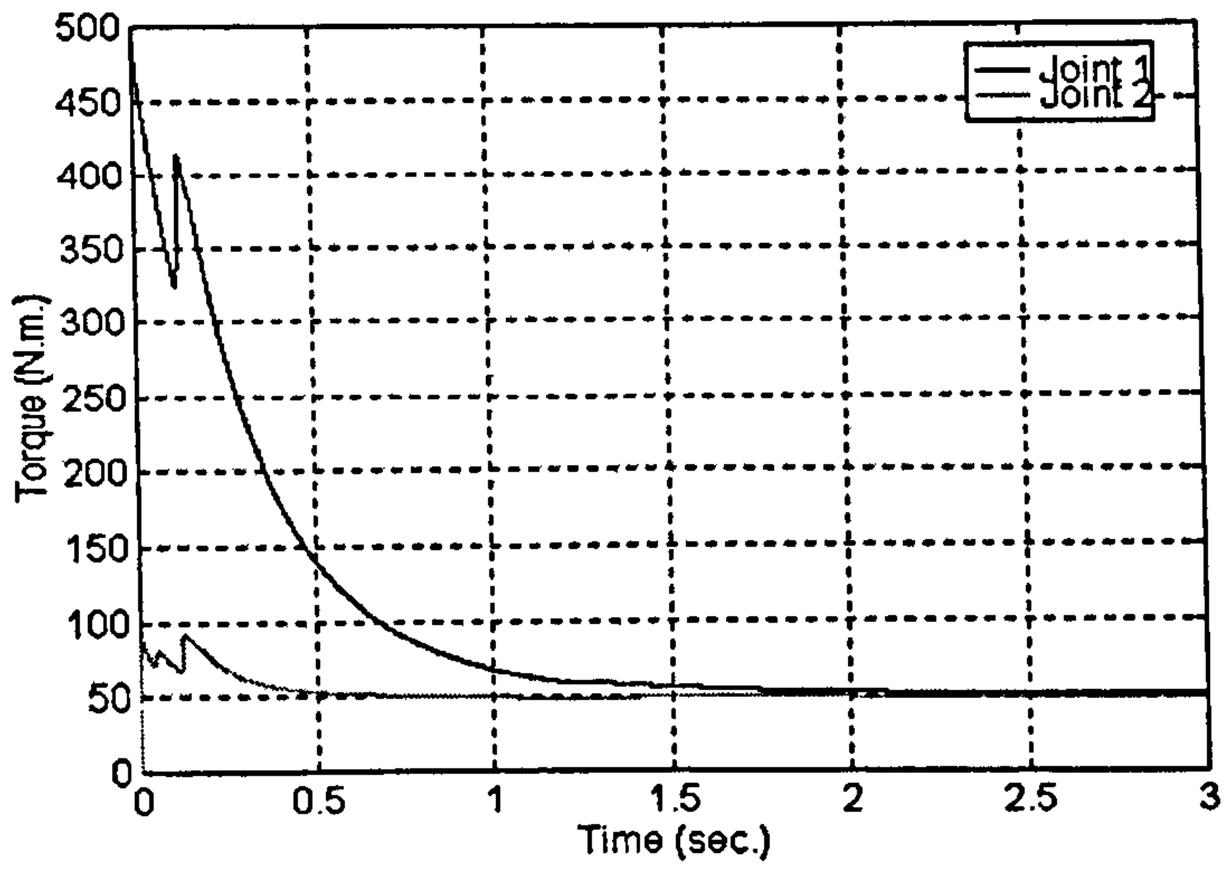


Figure 27: Applied Torque for the Joint Trajectory Demands

5.3 Offset Resetting

As demonstrated previously, indexing the bounds of the variable structure controller with the tracking error can drastically reduce the chattering. The exact convergence to zero of the position error however is no longer guaranteed and an offset occurs at the regulation point. To remove the offset, an integral action needs to be introduced in the control law. The controller is then of the form :

$$Ta = -\widehat{M}(\theta)\dot{e} - \widehat{C}(\theta,\dot{\theta})e + \widehat{G}(\theta) - K(\tilde{\theta},\dot{\tilde{\theta}}).sgn(S) + K_I.\int \tilde{\theta}\ .dt \qquad (94)$$

The term K_I is a column vector of constant terms for the integral action, all the other terms are defined as previously. The scheme is tested on the 2 degrees of freedom planar robot manipulator.

5.3.1 Simulations

The extended version of the controller with the reset action is tested on the robot manipulator used in the previous sections. Integration time is kept to 0.00001 sec and the torque update interval remains at 0.0002 sec, as in the previous simulations. Identical trajectory demands are used. The results of the simulation are shown in figures 28-32. There is no chattering on the control torque and the position errors

converge exactly to zero. The overall system behaviour is similar to a classic VSC scheme. The system is forced toward a sliding surface and when reached it slides along it up to the regulation point.

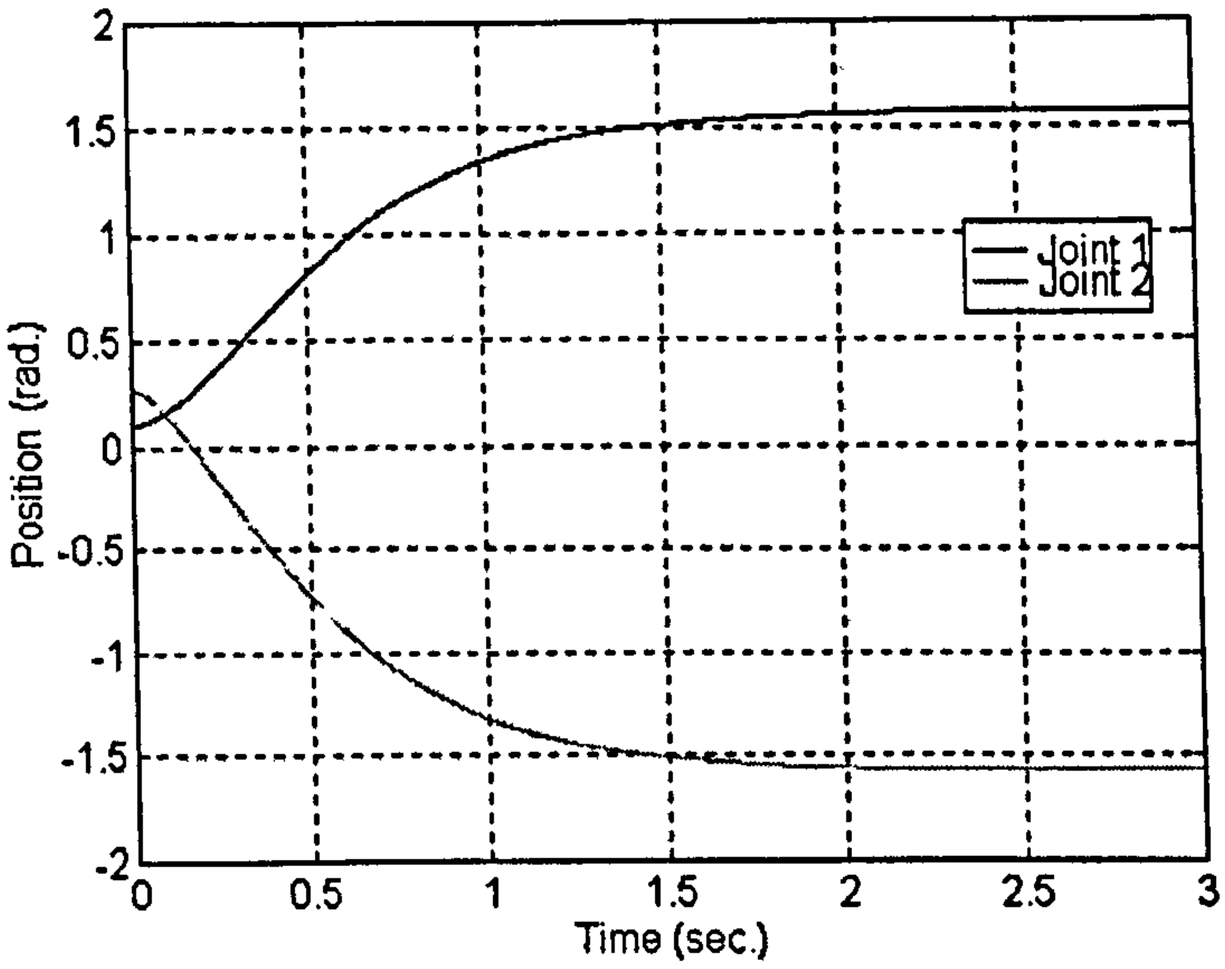


Figure 28: Joint Positions

5.4 Conclusion

In this chapter, a variable structure controller for manipulators is presented with its technically sound design approach. Simulation results of the proposed controller are given to illustrate its performance.

A Lyapunov function is used to conduct the controller design, a classic candidate Lyapunov function of the form of a pseudo-energy function is selected, $V = \frac{1}{2}S^T.M(\theta).S$. The controller parameters are derived so that the conditions for the Lyapunov function to exist are met. A computed-torque like feedforward term is primarily selected, and the next step of the design to bound the system uncertainties and mismatches is approached from an engineering perspective. The mismatch between the computed-torque like feedforward term and the robot system is decomposed into simple elements and analysed individually. Bounds for each term of the mismatch are derived from a practical and engineering point of view. The initial

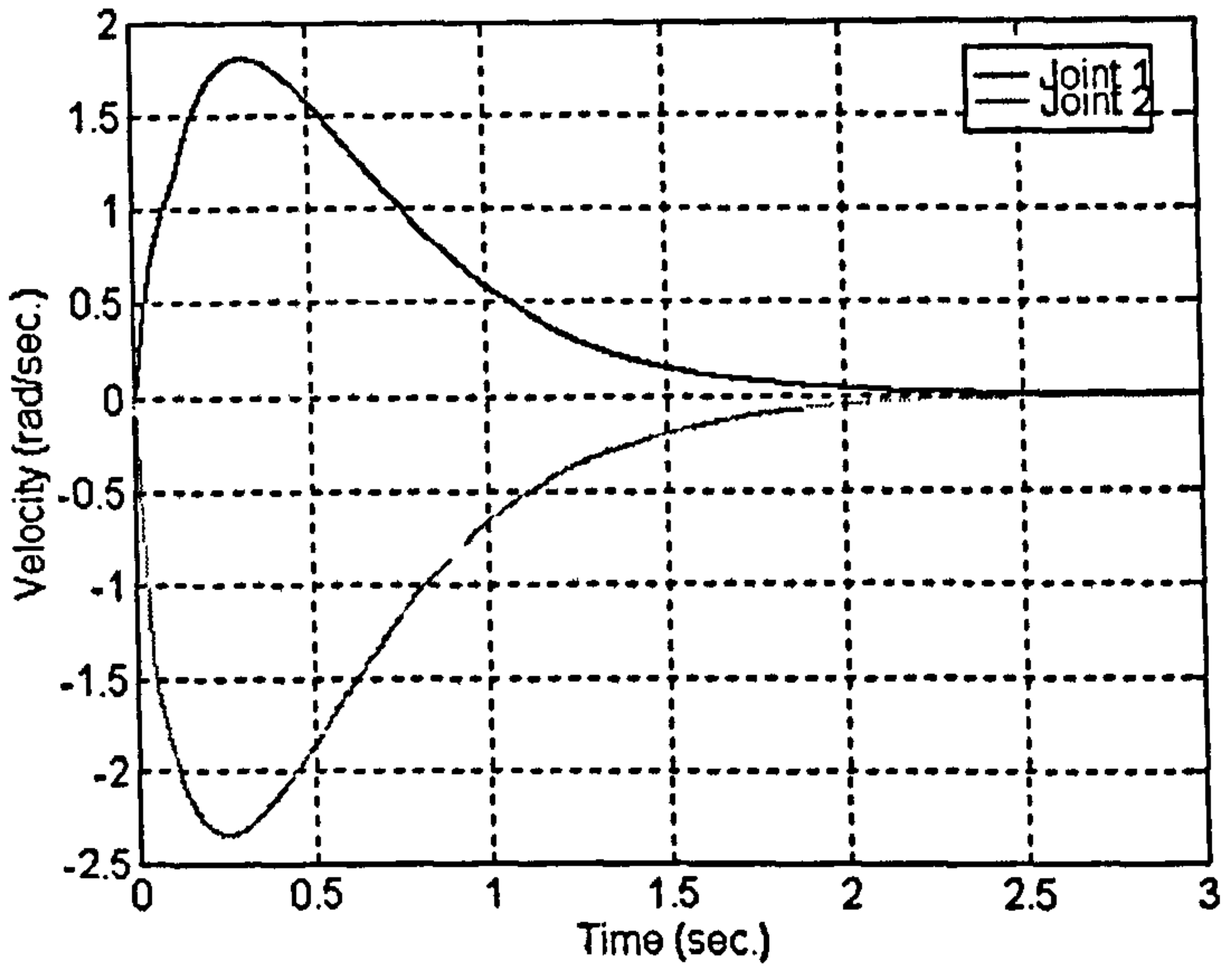


Figure 29: Joint Velocities

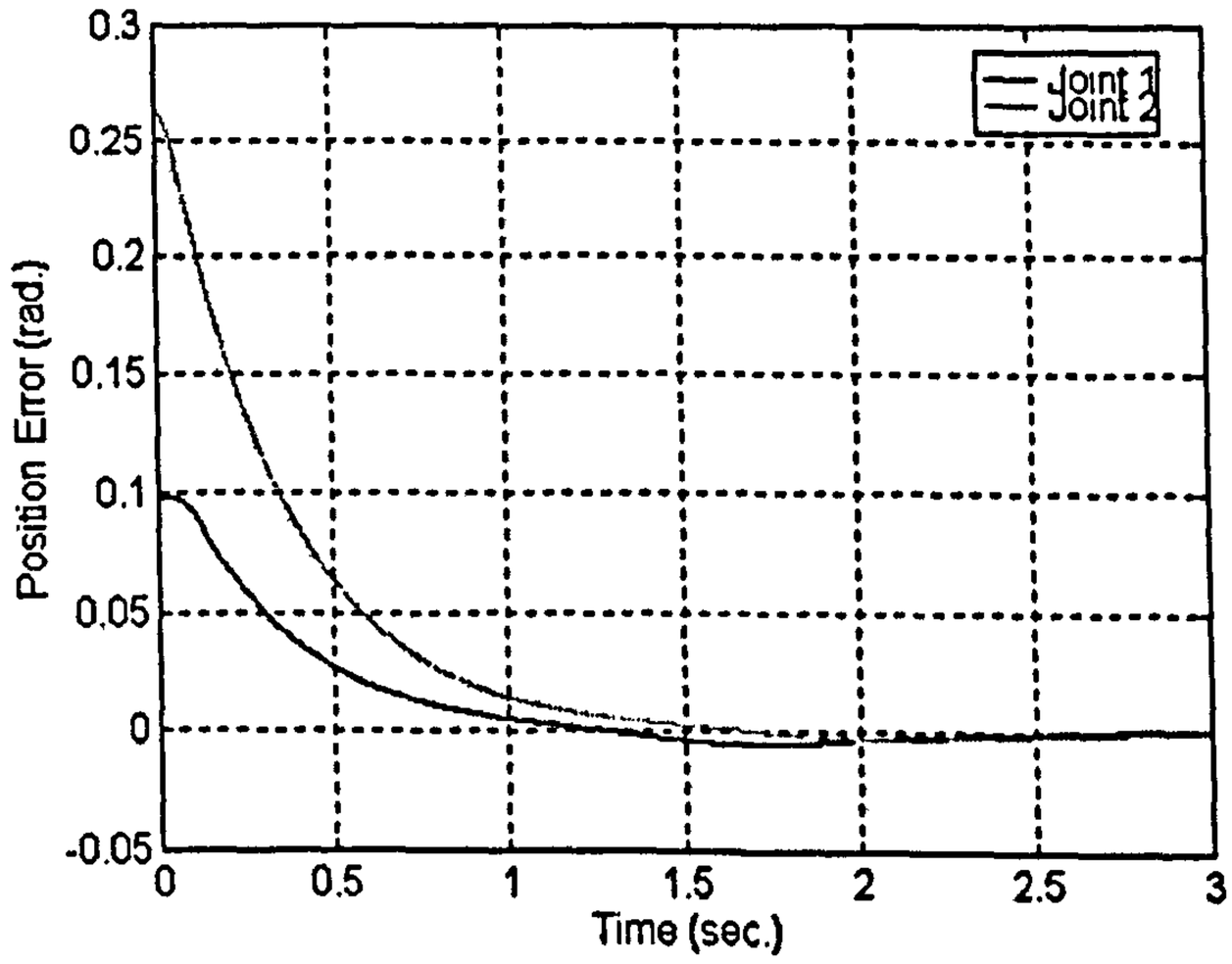


Figure 30: Position Errors at the Joints

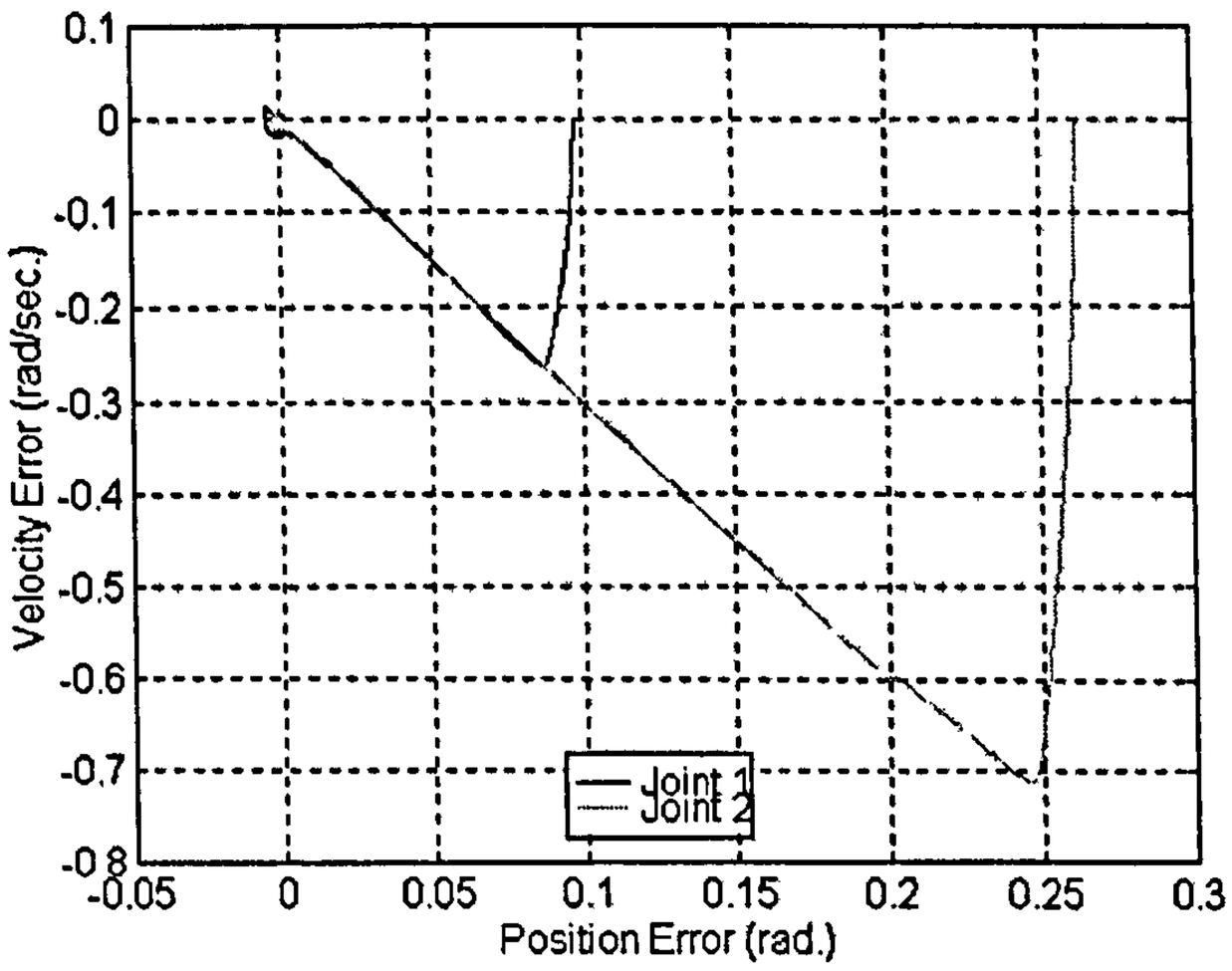


Figure 31: State Space Representation of the Trajectory Errors

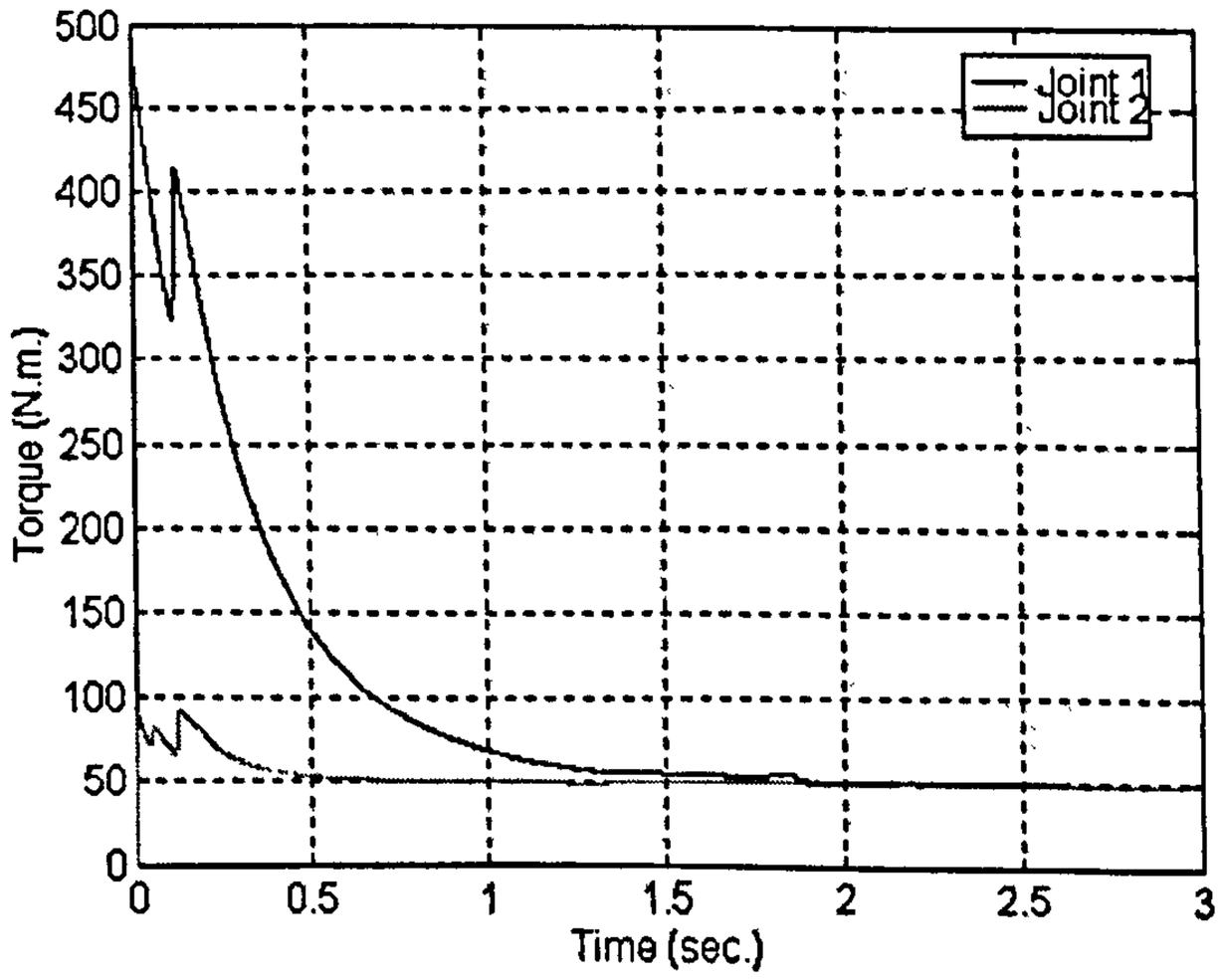


Figure 32: Applied Torque for the Joint Trajectory Demands

controller assumes fixed bounds for the VSC part, but simulations reveal large chattering on the torque as expected since no particular action has been taken against it. The simulations confirm the functionality of the controller, whilst underlining its impracticality because of the chattering.

Re-examining some of the initial assumptions suggests improvements to reduce the chattering magnitude can be obtained. Indexing the bound magnitudes with the position and velocity errors reduces the possible chattering at the regulation point and along the sliding surface. By making the bound a linear function of the position and velocity $K(\tilde{\theta}, \dot{\tilde{\theta}})$ with $K(0,0) = 0$, chattering is drastically reduced but an offset occurs at the regulation point, as shown by the simulations. The latter is tackled by introducing an integral action to the controller. The controller is then of the form, equation (95) :

$$Ta = -\widehat{M}(\theta)\dot{e} - \widehat{C}(\theta,\dot{\theta})e + \widehat{G}(\theta) - K(\tilde{\theta}, \dot{\tilde{\theta}}).sgn(S) + K_I.\int \tilde{\theta}\ .dt \qquad (95)$$

Simulations reveal the exact convergence to zero of the position error [65]. Tests on a real robot of the proposed controller has demonstrated its effectiveness. The performance obtained from the controller on a real system outperformed traditional control techniques such as PID control and computed-torque control [66].

6 IMPLEMENTATION ON THE SPRINTA

To implement the devised controller, a dynamic model of the SPRINTA is required. Initially, the kinetic parameters (inertia, mass, centre of mass) of the SPRINTA have to be derived,which is done with C.A.D. tools, whilst the calculation of the inverse dynamic equations is conducted using MATHEMATICA®. The Euler-Lagrange general method for the inverse dynamic calculation is employed, because of its systematic nature and the compactness of the results, see APPENDIX B.

The results reveal the highly non-linear nature of the SPRINTA and the dramatic effect of the non-linear transmission on the dynamic model of the manipulator (APPENDIX B). The non-linear transmission introduces complexity in the inverse dynamic equations, and generates cumbersome terms. The actual dynamic model for the robot manipulator is unmanageable and can not be used for on-line computation of the torque. This implies that simplification of the model is required. Inevitably such model simplification increases the mismatch between the actual robot dynamics and the model used to compute the torque and results in poorer performance. Extra emphasis has to be put on the robust part of the controller.

With such large uncertainties resulting from the use of a poor model, the bounds of the variable structure controller have to be large to guarantee the converge of the error for the entire workspace. The need for adjusting the bounds of the VSC element is even more crucial as their magnitude is large.

6.1 Initial Implementation Considerations

In order to be implementable, the dynamic model of the SPRINTA has to be greatly simplified. All the cumbersome terms are removed and only their averages are considered. The terms in the inertia matrix and in the Coriolis / Centripetal equations are taken as constant. Experimental tests reveal that the gravitational terms have very little influence on the control performance and since their computation is difficult and time consuming, they have been omitted in the estimated dynamic model of the SPRINTA. The form of the dynamic equations of the SPRINTA which is used for the feedforward element are as follows, equation (96).

$$\begin{bmatrix} M_{11} & M_{12} & 0 \\ M_{21} & M_{22} & 0 \\ 0 & 0 & M_{33} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} C_{11}.\dot{\theta}_1 & -C_{12}.\dot{\theta}_2 & 0 \\ -C_{21}.\dot{\theta}_1 & C_{22}.\dot{\theta}_2 & 0 \\ 0 & 0 & C_{33}.\dot{\theta}_3 \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (96)$$

where :

$M_{11}, M_{12}, M_{21}, M_{22}$ and M_{33} are constant inertia parameters, and $C_{11}, C_{12}, C_{21}, C_{22}$ and C_{33} are constant parameters for the Coriolis/centripetal effect.

The devised controller, equation (95), has been tested with the computed torque model of the SPRINTA given in equation (96). The large inaccuracy on the model-based feedforward term imposes a large magnitude for the bounds of the VS Controller. Despite the fact that these bounds are variable, high vibration problems of the manipulator are encountered at steady state. The vibrations are due to the large control effort derived from the bounds. Further it is observed that the vibration problem was worse for some locations of the endpoint of the manipulator. As the endpoint gets closer to the boundaries of the workspace, the magnitude of vibration increases dramatically, triggering the safety routine that turns off the entire system.

To resolve the vibration problem, two measures are adopted. First of all the state space of each joint is divided into three zones, as shown in figure 33. The first zone, *zone*1, is defined by the two vertical lines distant from the Y-axis by the length Γ, and by the two lines parallel to the sliding surface $S = 0$ and distant from it by the length Δ. The second zone, *zone*2 is presented by the zone between the two vertical lines distant from the Y-axis by the length Γ, minus *zone*1. The last zone, *zone*3 is located outside the two vertical lines. Such partitioning of the phase plane of the tracking error, provides further refinement for the bound adjustment.

The second step which is taken to tackle the vibration problem for some locations of the manipulator endpoint within the workspace, is to identify these locations and to draw a contour, as shown in figure 34. The control law is further adjusted according to the endpoint location. The shaded area in figure 34 represents a vertical

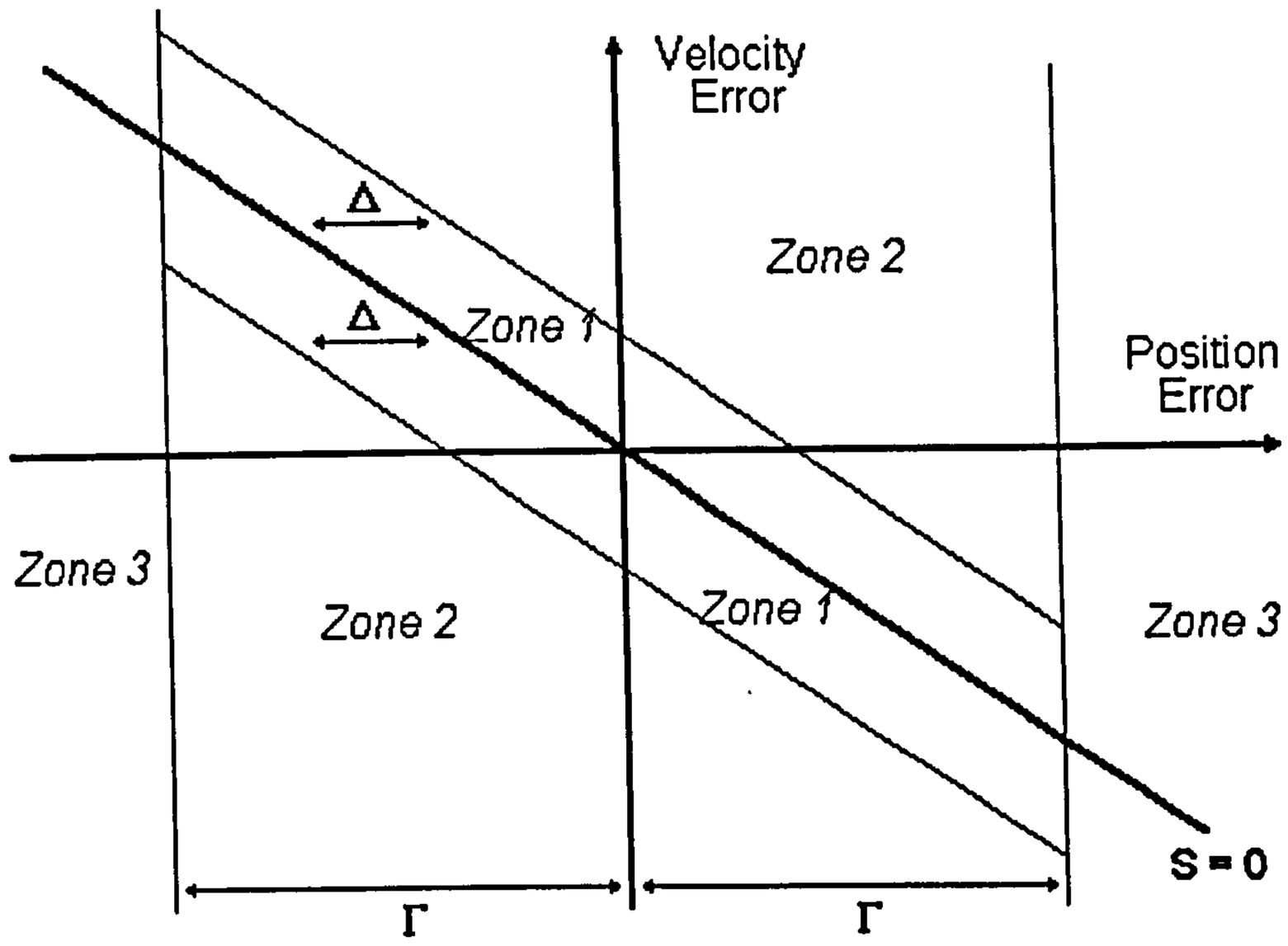


Figure 33: State Space Representation of the Tracking Error

section of the workspace of the SPRINTA. Locations for which the vibration problem is important are represented on the cross section of the workspace as the outside contour connecting the dots. When the endpoint is outside the contour defined by the dots, the robot is much more subject to vibration, which in some cases leads to the complete shut down of the system.

In order to take into account this sensitivity with respect to the endpoint location in the control law, the contour which has been identified, is mapped into motor coordinates, as shown in figure 35. The transformation from endpoint coordinates to motor coordinates is indispensable since the control law operates in motor coordinates. To be used effectively, the contour obtained in motor coordinates is approximated by the equation of a circle. This approximation of the vibration contour is preferred, as it provides an easy to use algebraic representation of the contour. The circle is centered on the point $(40,000; -100,000)$, where the units are in number of counts of the incremental resolver, and its radius is defined as Ψ. The vibration is only affected by the endpoint position in the vertical plane, i.e. the

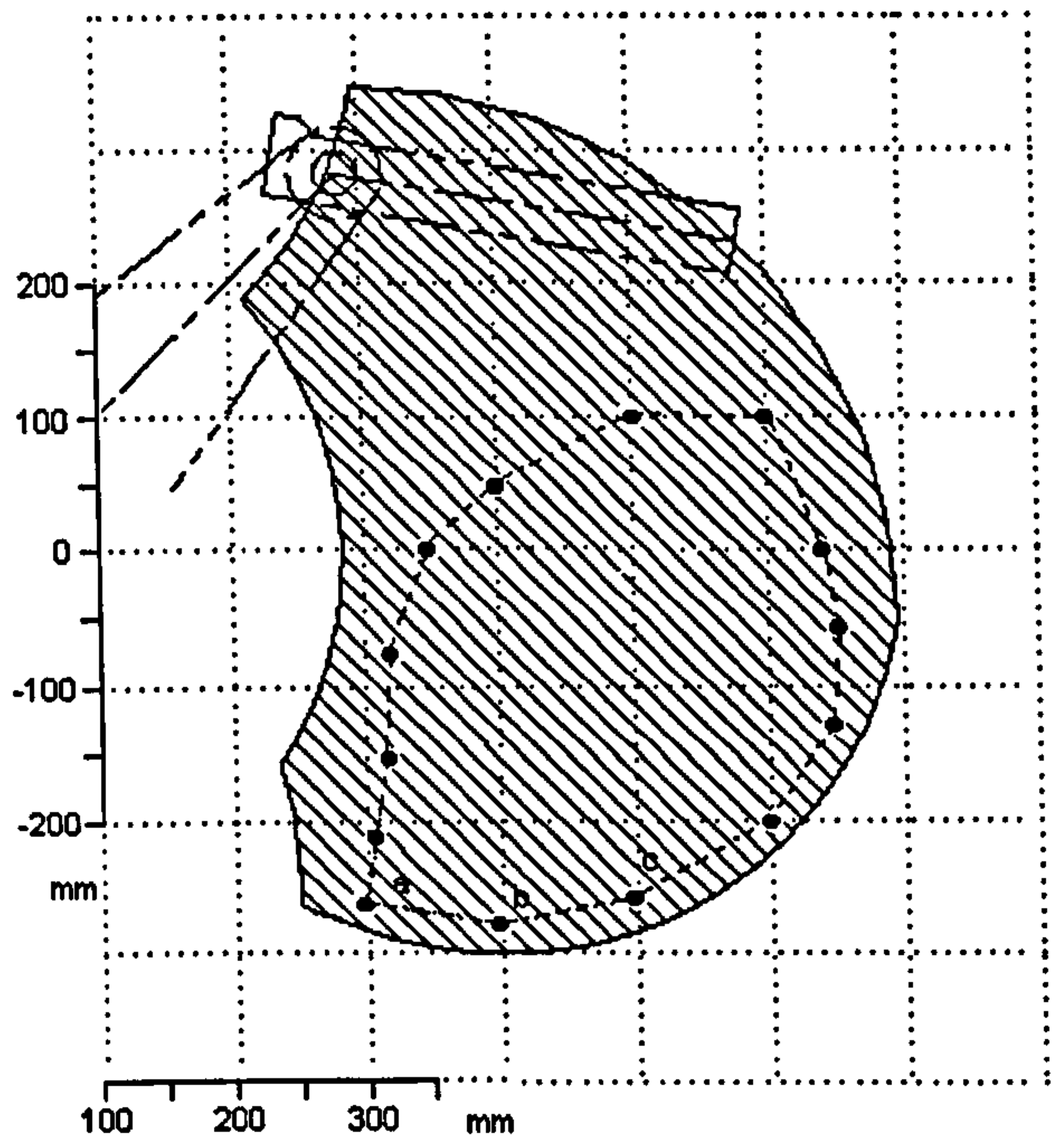


Figure 34: Contour of the Large Vibration Zone of the Endpoint

vibration problem is independent of the waist location. For this reason, the contour defined in motor coordinates is given in differential coordinates, i.e. (coordinates of the shoulder motor)-(coordinates of the waist motor) and (coordinates of the elbow motor)-(coordinates of the waist motor). The subtraction of the waist coordinates from the coordinates of shoulder and elbow motors is required to remove the mechanical differential coupling between the shaft of the waist motor and the two other shafts (shoulder and elbow). The obtained mapping in the motor coordinates is only concerned with the location of the endpoint in the vertical plane and not its orientation in the horizontal plane.

6.2 First Controller Implementation and Evaluations

The controller that has been successfully implemented and which provides acceptable response for the entire workspace of the robot manipulator, has the following structure :

$$Ta = -\widehat{M}(\theta)\dot{e} - \widehat{C}(\theta,\dot{\theta})e - K(\tilde{\theta},\dot{\tilde{\theta}},\Delta,\Gamma,\Psi).sgn(S) + K_I.\int \tilde{\theta}\ .dt \qquad (97)$$

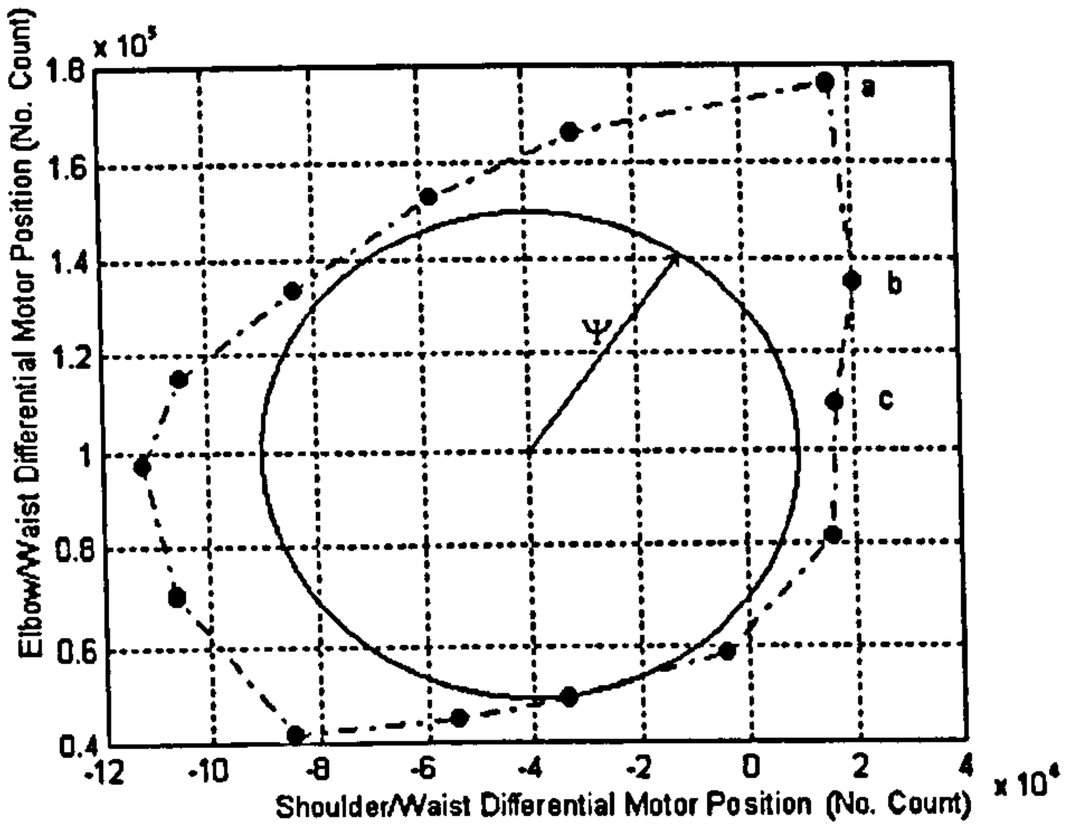


Figure 35: Contour of the Large Vibration Zone Mapped into Motor Co-ordinates

where :

$\widehat{M}(\theta)$ and $\widehat{C}(\theta, \dot{\theta})$ are defined in equation (96),

$K(\tilde{\theta}, \dot{\tilde{\theta}}, \Delta, \Gamma, \Psi)$ is the column vector of the varying bounds,

Δ defines the distance in the phase plane between the line $S = 0$ and the two lines parallel to the sliding surface. Each joint has its own phase plane and its own specific value for Δ_i.

Γ represents the distance in the phase plane between the vertical axis and the two lines parallel to it.

Ψ represents the radius of the circle in the differential motor coordinates and which is centered on $(40,000; -100,000)$.

The bounds variation with respect to the position and velocity errors, $\tilde{\theta}$ and $\dot{\tilde{\theta}}$, is proportional, as initially defined in equation (93). The terms Δ, Γ and Ψ which define zones in the phase plane and in the workspace affect only the rate of proportionally between the bounds and the position and velocity errors as expressed in equation (98).

$$K(\tilde{\theta}, \dot{\tilde{\theta}}, \Delta, \Gamma, \Psi) = K_1(\Delta, \Gamma, \Psi).\left|\tilde{\theta}_i\right| + K_2(\Delta, \Gamma, \Psi).\left|\dot{\tilde{\theta}}_i\right| \tag{98}$$

When the system state error of one of the joint is in *zone*3, i.e. $\left|\tilde{\theta}_i\right| > \Gamma_i$,

the position error is relatively large, the term $K_{1i}(\Delta, \Gamma, \Psi)$ is at its maximum value, while the value for $K_{2i}(\Delta, \Gamma, \Psi)$ is reduced by a fixed amount. In *zone*3, the position error is dominant in the calculation of the bounds. When the system is in *zone*2, i.e. $\left|\widetilde{\theta}_i\right| < \Gamma_i$ and $|S_i| > \Delta_i$, the term $K_{1i}(\Delta, \Gamma, \Psi)$ is slightly reduced, while $K_{2i}(\Delta, \Gamma, \Psi)$ is at its maximum. In *zone*1, i.e. $\left|\widetilde{\theta}_i\right| < \Gamma_i$ and $|S_i| < \Delta_i$, $K_{1i}(\Delta, \Gamma, \Psi)$ is further reduced and $K_{2i}(\Delta, \Gamma, \Psi)$ is reduced to zero. The adjustment of the terms with respect to the zones is represented in figures 36 and 37. The amount by which the terms $K_{1i}(\Delta, \Gamma, \Psi)$ and $K_{2i}(\Delta, \Gamma, \Psi)$ are varied with respect to the zones, is different for each joint, but the direction of variation is identical for all the joints.

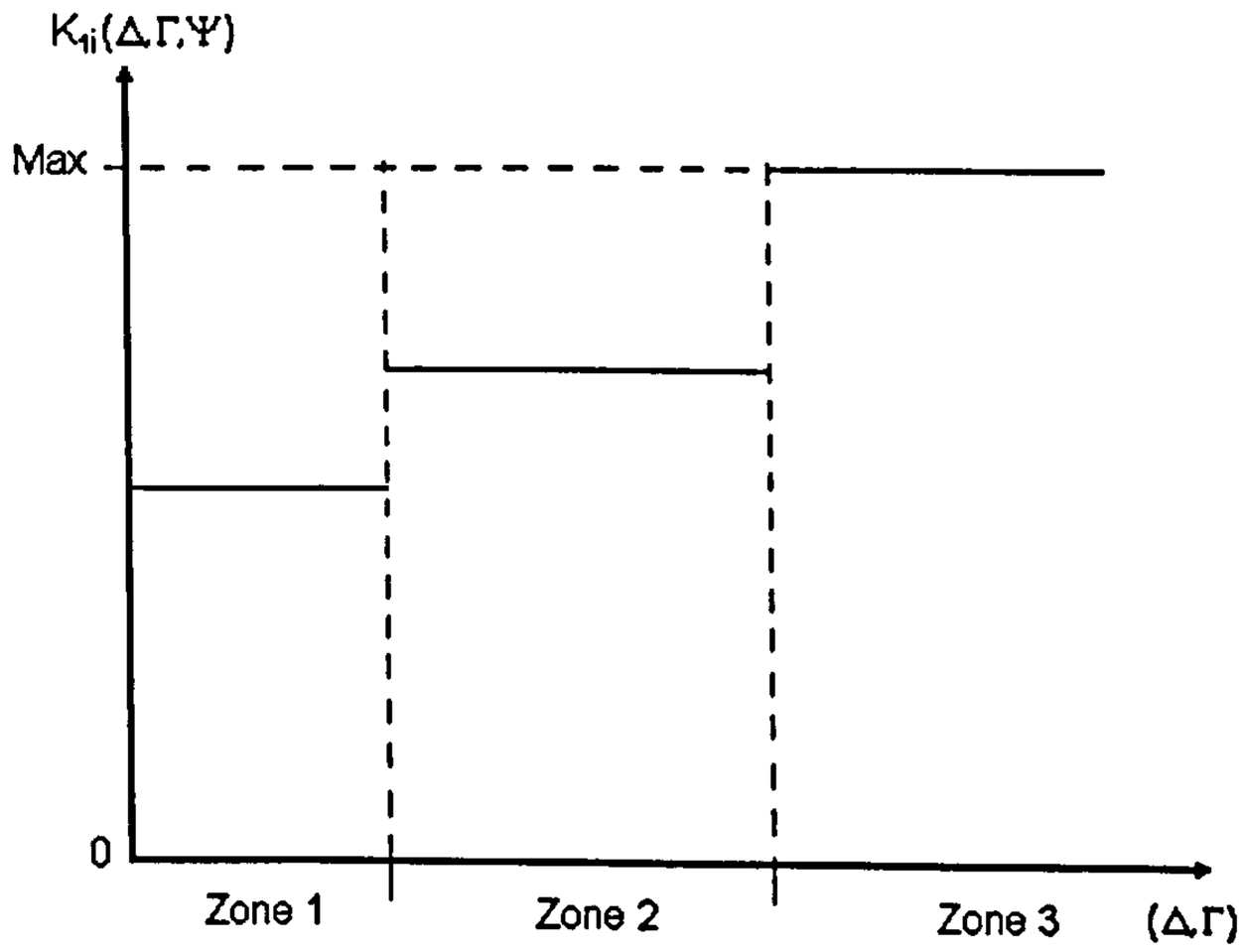


Figure 36: Variation of $K_{1i}(\Delta, \Gamma, \Psi)$ with Respect to the Zones

The position of the endpoint within its workspace, affects the terms $K_{1i}(\Delta, \Gamma, \Psi)$ and $K_{2i}(\Delta, \Gamma, \Psi)$ in a binary way, i.e. the endpoint is within the circle of radius Ψ or it is not. If the endpoint is within the circle the coefficient of the bounds ($K_{1i}(\Delta, \Gamma, \Psi)$ and $K_{2i}(\Delta, \Gamma, \Psi)$) are unaffected. On the other hand if the endpoint is mapped outside the circle, then the coefficients of the bounds are divided by a specific amount, this amount is different according to the joint and the type of coefficients ($K_{1i}(\Delta, \Gamma, \Psi)$ or $K_{2i}(\Delta, \Gamma, \Psi)$).

The amount by which the coefficients of the bounds are affected and their triggering values (Δ, Γ) are determined experimentally.

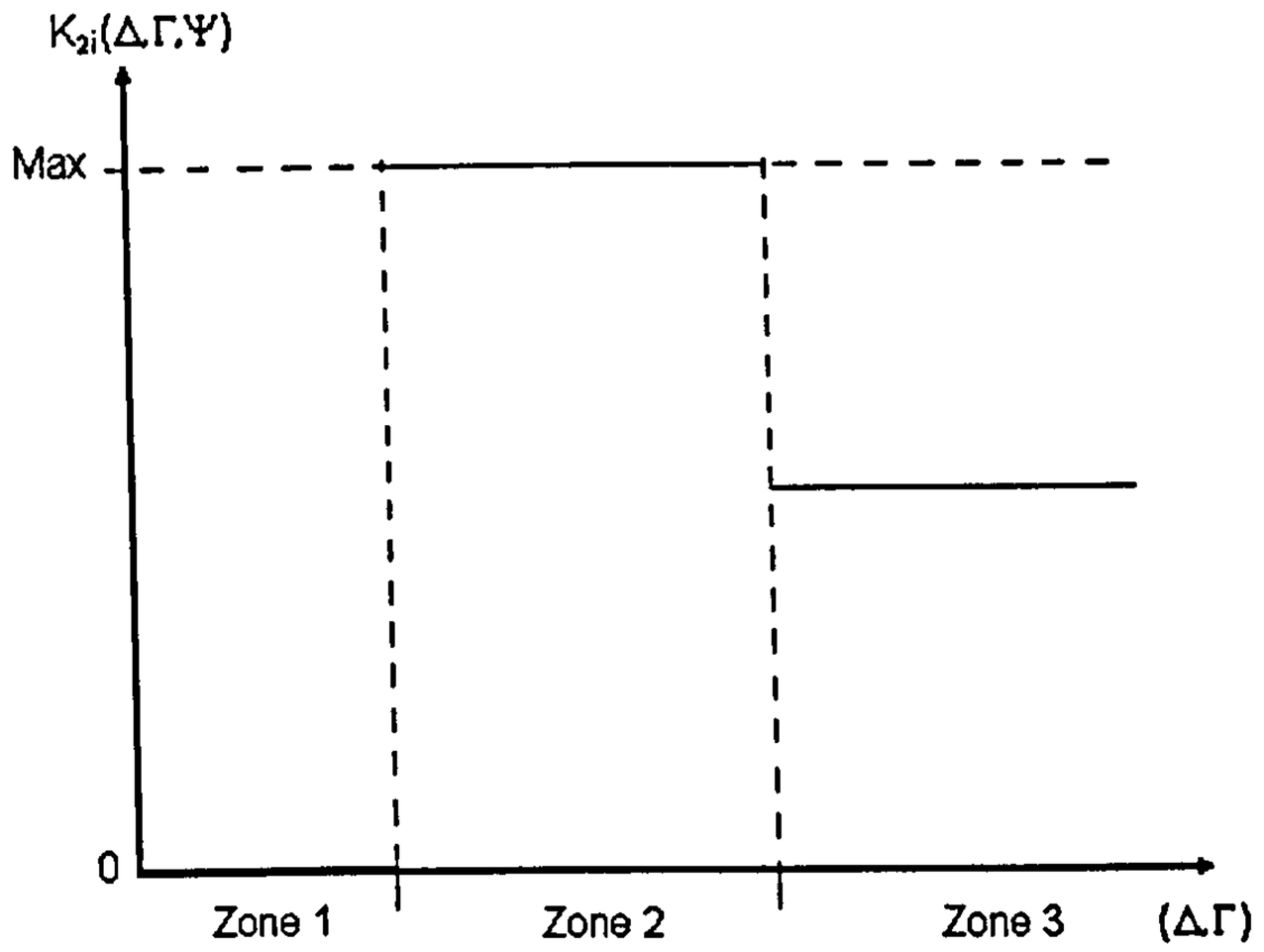


Figure 37: Variation of $K_{2i}(\Delta, \Gamma, \Psi)$ with Respect to the Zones

The devised controller equation (98), has been implemented on the SPRINTA, and tested for several types of trajectory; a straight line, a goalpost test and a circle in the vertical plane.

6.2.1 Straight Line Motion

The first type of trajectory used to evaluate the controllers is the straight line motion. The end point of the robot is required to move from the Cartesian location $[650\ 0\ -100]_{XYZ}$ to the location $[0\ 550\ -100]_{XYZ}$ in 1 sec, (all co-ordinates are in mm), with the origin of the co-ordinate frame located at the centre of the waist joint. This test is used to assess the tracking performance as well as the steady state performance of the controller. Rectilinear interpolation is used for this movement, the sampling time for the controller has been set to $1.5 milli$ sec. The trajectory demand in motor co-ordinates is given in figure 38, whilst the system response is shown in figures 39-41. The units used for the position at the motor is the number of pulses from the incremental encoder. The encoder produces 614400 pulses per revolution, which corresponds to a resolution of $0.59 milli$ deg or $0.01 millirad$. For all figures the torque demand value is given in the form of a digital data. The drivers have been configurated so that they respond to the command data with

torque proportional to the maximum torque available at the speed, i.e. the motor torque limit slightly decreases with speed, [67]. The maximum limits for the torque are given for -2047 and 2047, since the motors do not have the same rating those limits correspond to different values of torque in N.m . Figure 42 is a top view of the demand and actual end point trajectory of the SPRINTA robot, whilst figure 43 is a perspective in 3 dimensions of the point to point trajectory.

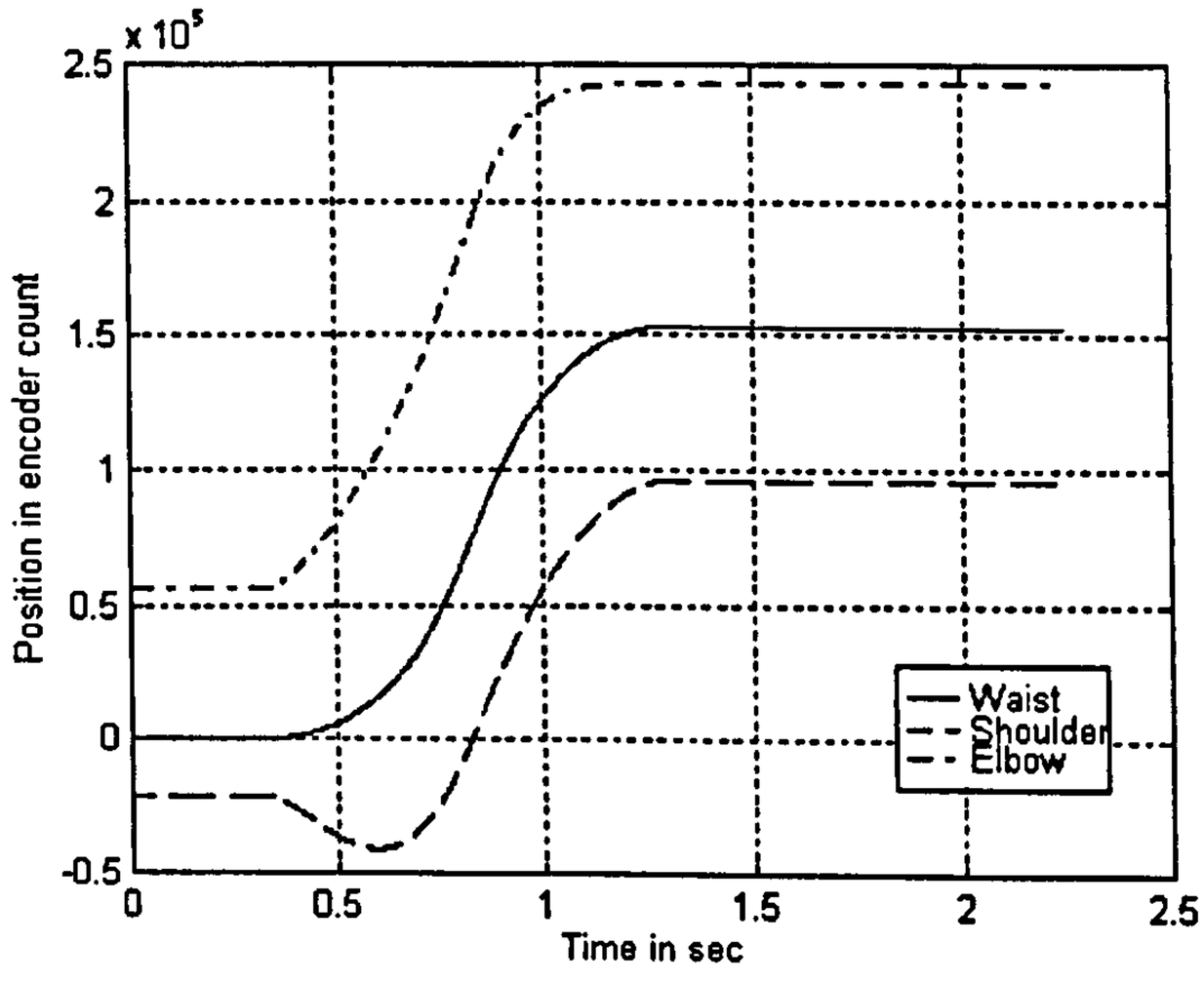


Figure 38: Straight Line Trajectory Demands in Motor Co-ordinates

The maximum position errors for the waist, shoulder and elbow joints are 629, 1936 and 5766 encoder pulses i.e. $0.0064rad$, $0.02rad$ and $0.059rad$ respectively. At the end of the movement, the largest position error of the three motor shafts is less than 1000 encoder pulses ($0.01rad$), and the position errors converge to a steady state error of less than 200 encoder pulses or less than $0.002rad$ within 1 sec. In Cartesian co-ordinates, the maximum deviation of the end point of the SPRINTA from the demand trajectory in less than $25mm$, as shown in figures 42 and 43.

6.2.2 GoalPost Test

The second type of test is the goalpost test, in which the path is similar to a pick and place movement. It is an industrial standard test that indicates the maximum

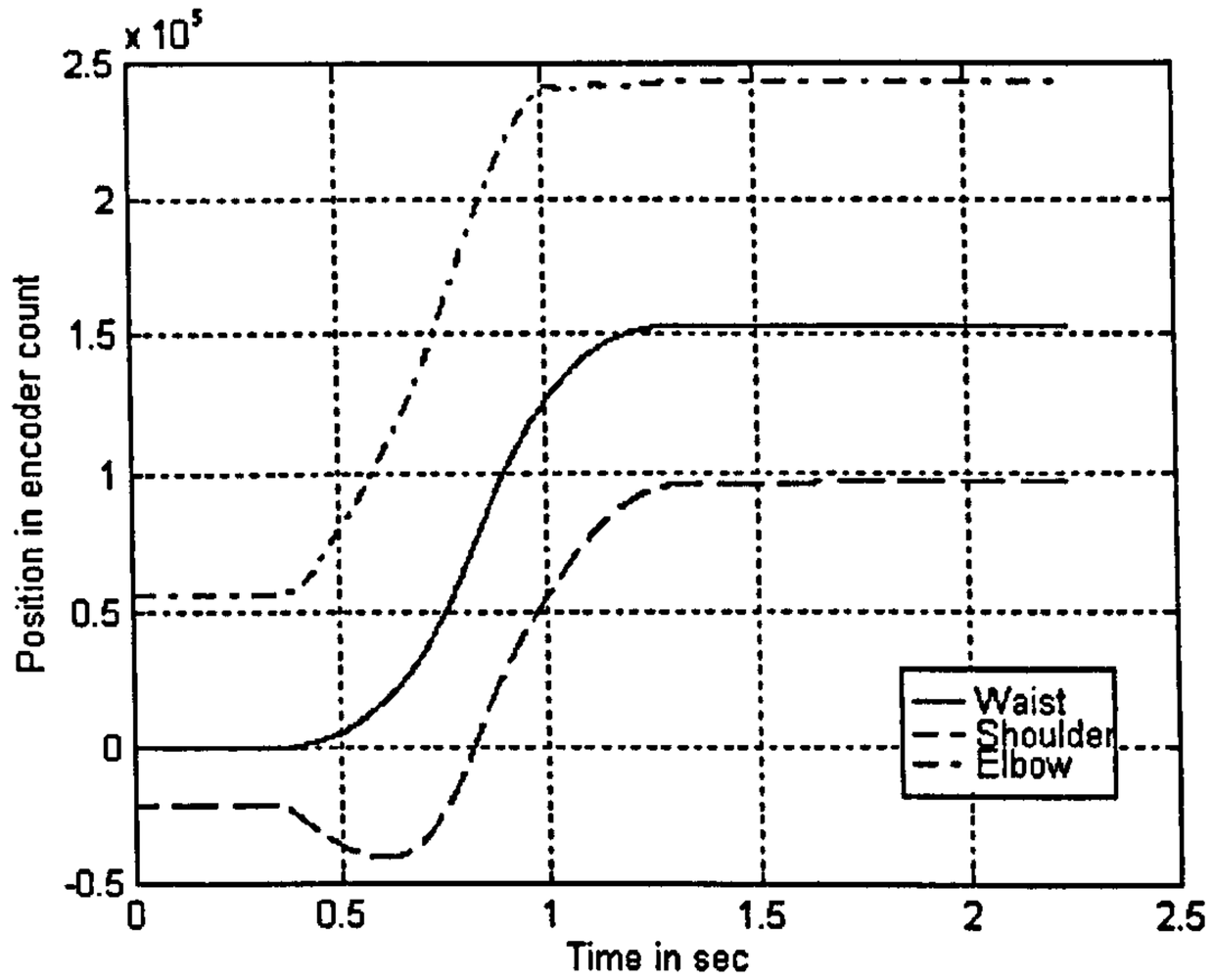


Figure 39: Response Trajectories in Motor Co-ordinates for Straight Line Motion

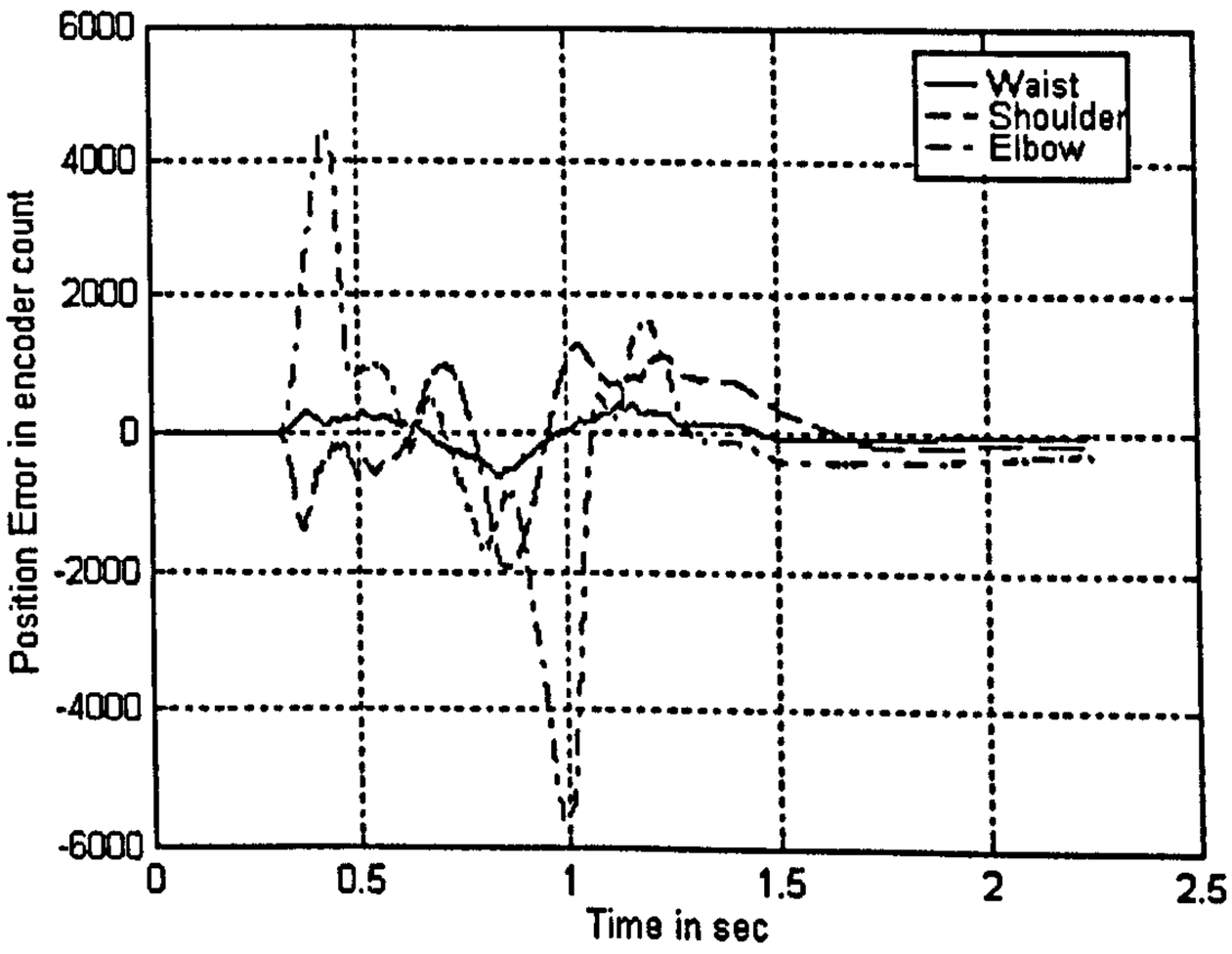


Figure 40: Position Errors in Motor Co-ordinates for the Straight Line Trajectory

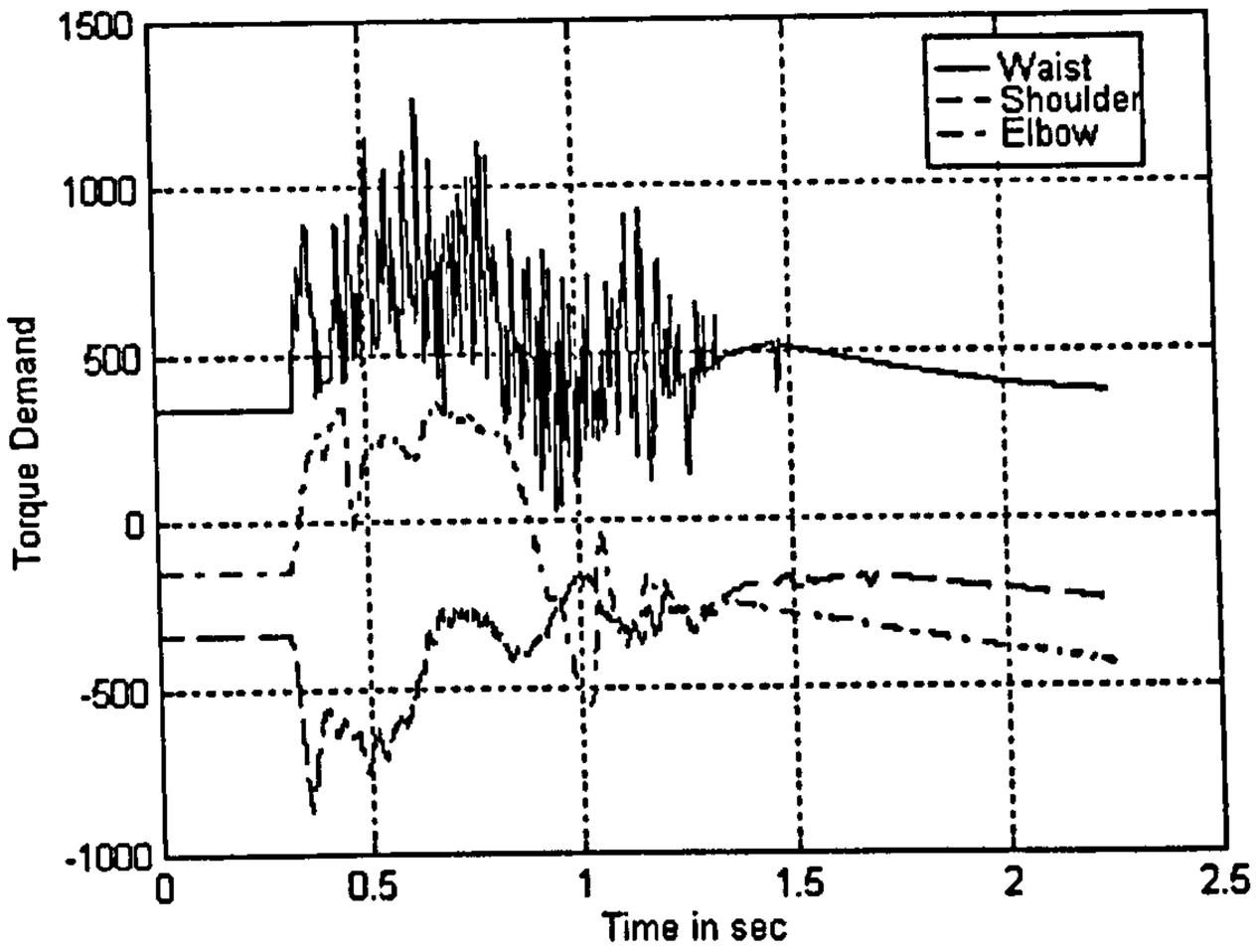


Figure 41: Torque Demands for Straight Line Motion

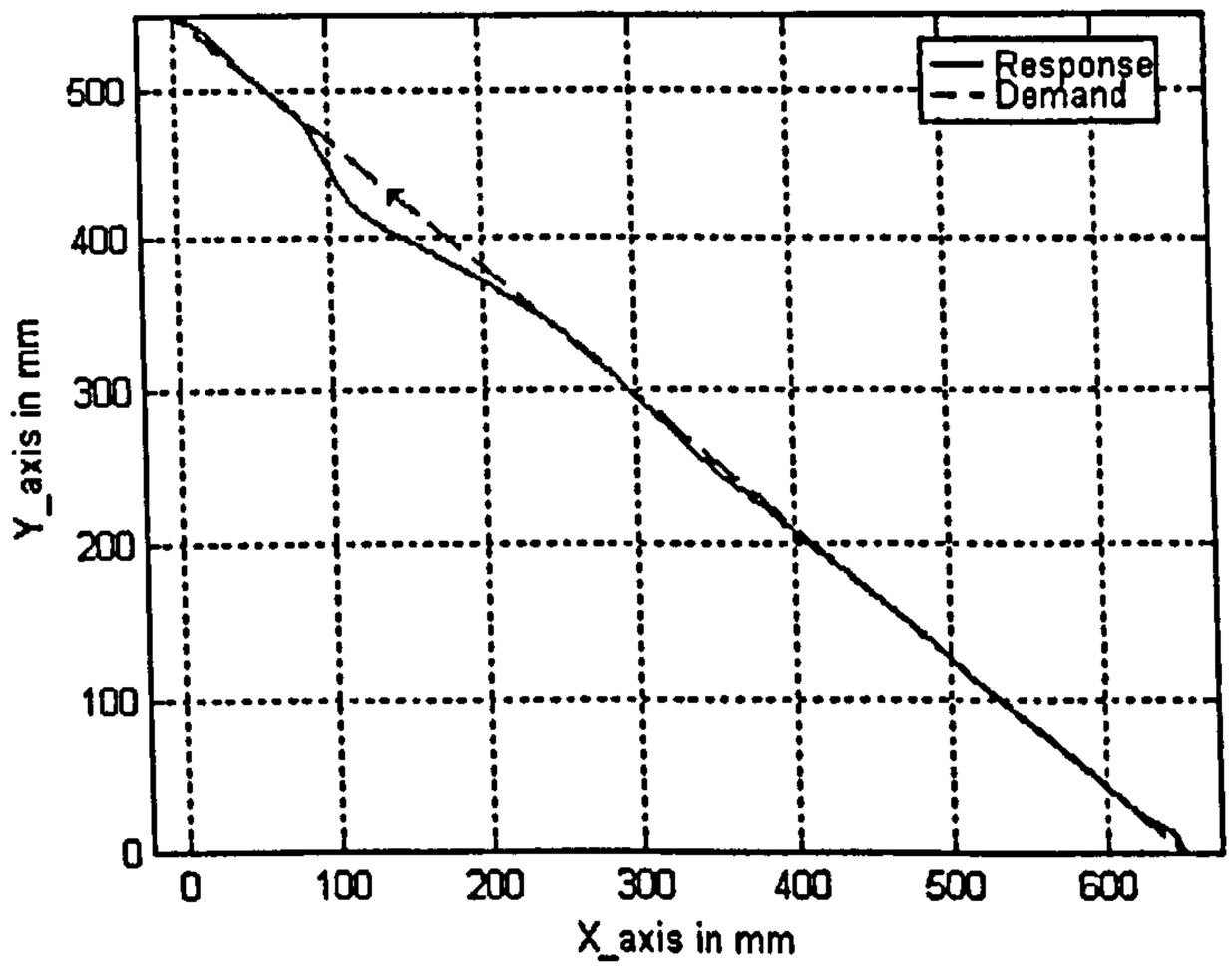


Figure 42: Top View of the End Point Displacement for the Straight Line Trajectory

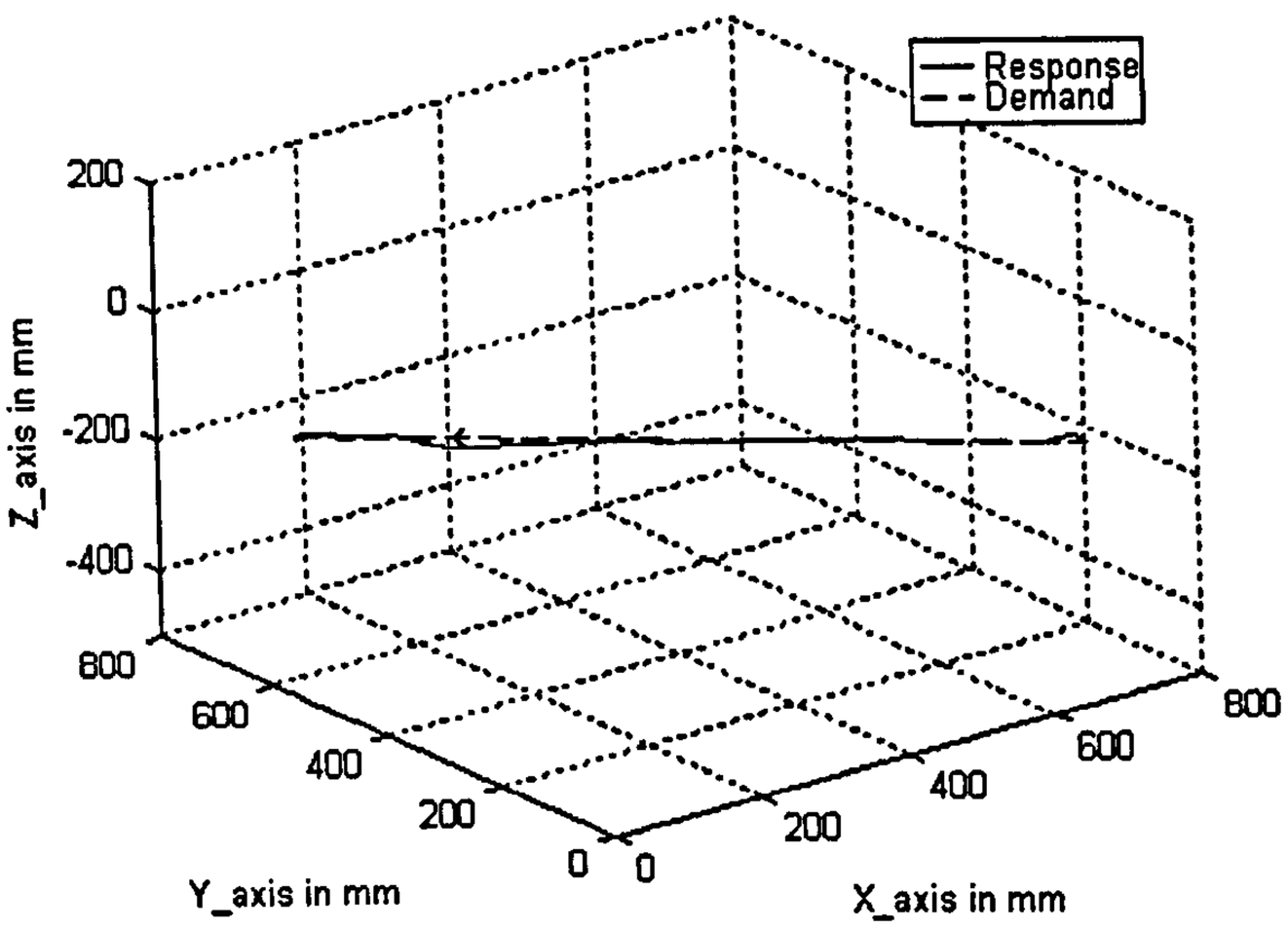


Figure 43: 3D View of the End Point Displacement for the Straight Line Trajectory

speed of motion a robot arm can consistently perform. The path is $300mm$ across and $25mm$ height as shown in figure 44.

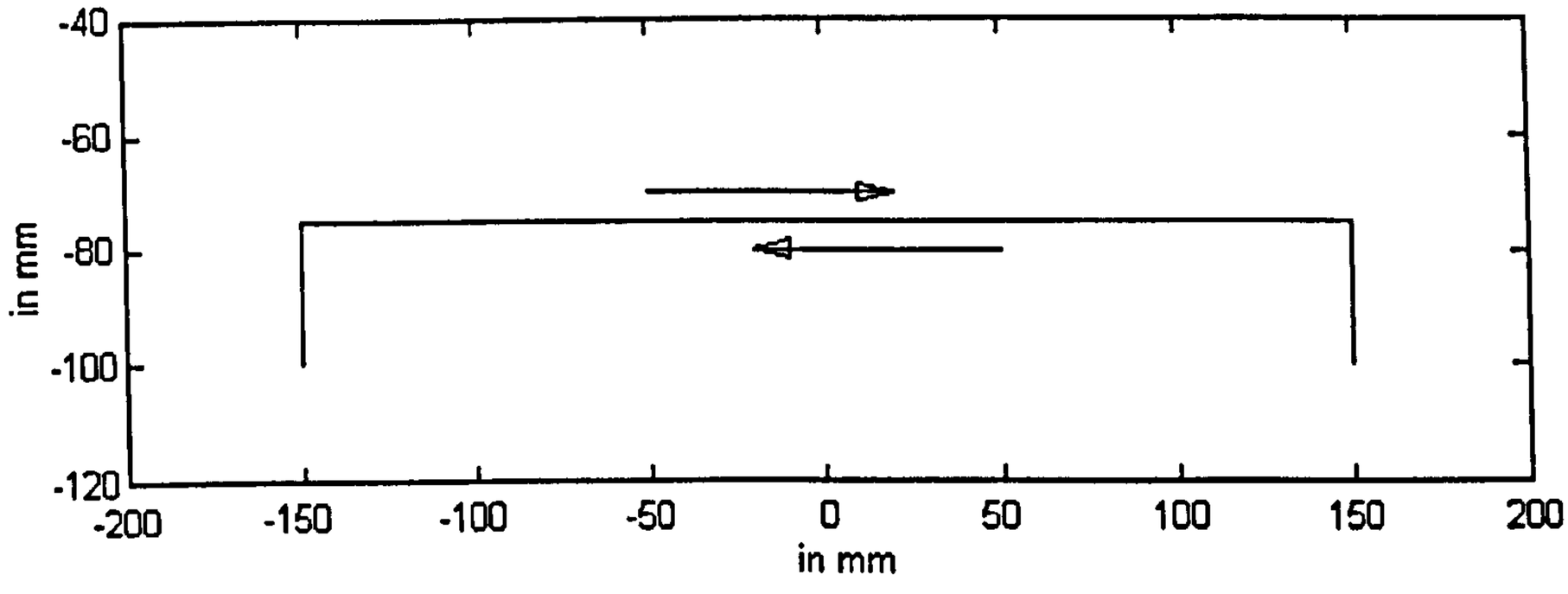


Figure 44: Goalpost Test Path

The SPRINTA is capable of executing the goalpost test (forward and return movement) in 0.7 sec in a permanent cyclic manner. The assessment of the controllers is performed at the maximum speed (0.7 sec) at which the goalpost test can be executed. The test has been performed between the point $[150\ 600\ -100]_{XYZ}$ and the point $[-150\ 600\ -100]_{XYZ}$ (all co-ordinates are in mm and the origin of

the co-ordinate frame is located at the centre of the waist joint). As for the straight line test, rectilinear interpolation is used for this movement with a sampling time of $1.5 milli$ sec for the controller. The trajectory demand in motor co-ordinates is given in figure 45, whilst the system response is shown in figures 46-48.

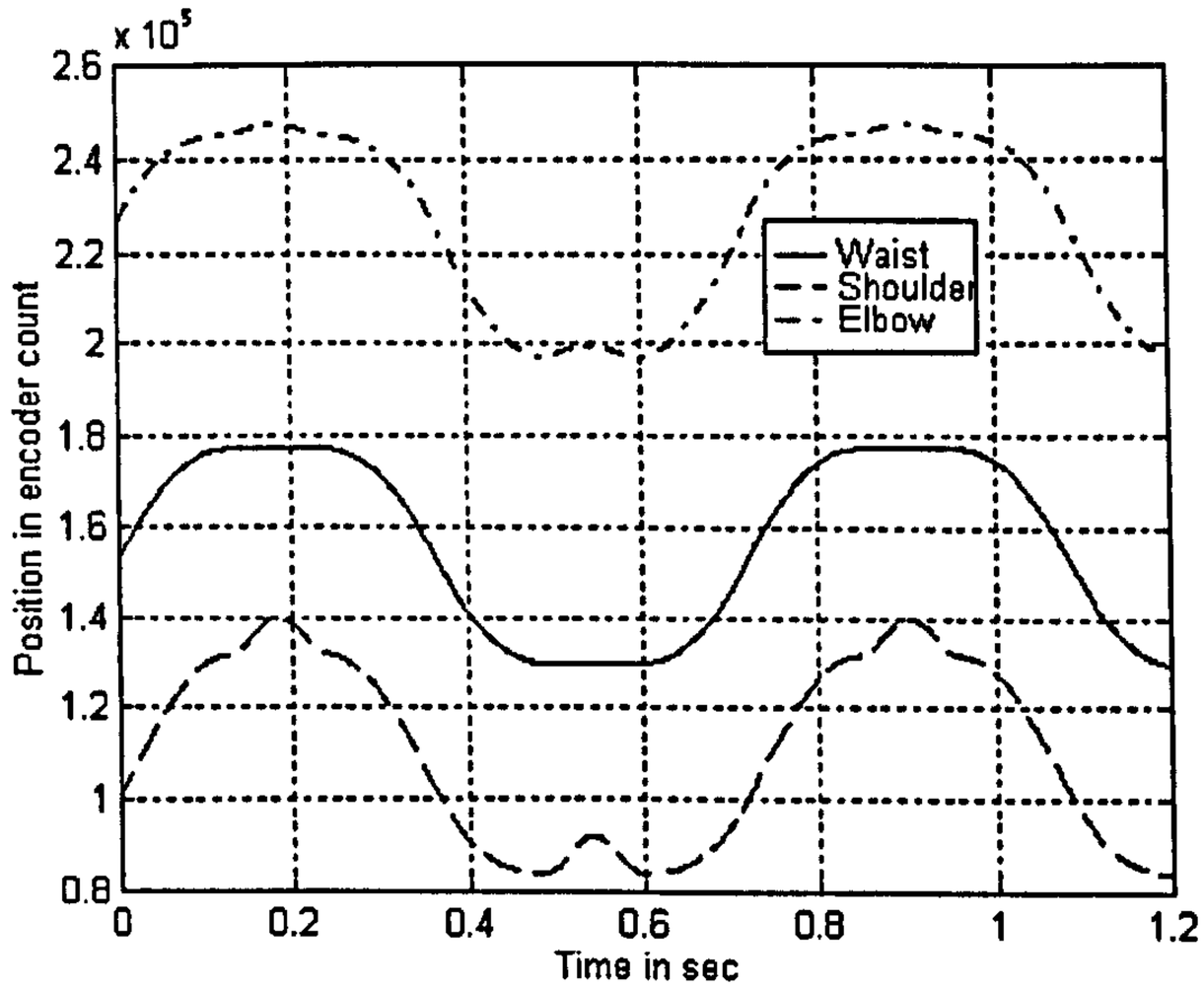


Figure 45: Goalpost Test Trajectory Demands in Motor Co-ordinates

The maximum position errors for the waist, shoulder and elbow joints are 1135, 9339 and 6020 encoder pulses or $0.012 rad$, $0.095 rad$ and $0.062 rad$ respectively. In Cartesian co-ordinates, the maximum error represents a deviation of the end point from the horizontal demand trajectory of less than $25 mm$, as can be seen from figures 49 and 50.

6.2.3 Circle Trajectory in the Vertical Plane

The third and last test is a circle in a vertical plane with its centre is located at the Cartesian position $[0\ 550\ -100]_{XYZ}$ (all co-ordinates are in mm) and a radius of $100 mm$. The circular trajectory is executed continuously with a period of 0.8 sec for each revolution. As for the two previous tests, rectilinear interpolation with a controller sampling time of $1.5 milli$ sec is used. The corresponding trajectory demand in motor co-ordinates is represented in figure 51 whilst the response of the system is shown in figures 52-54.

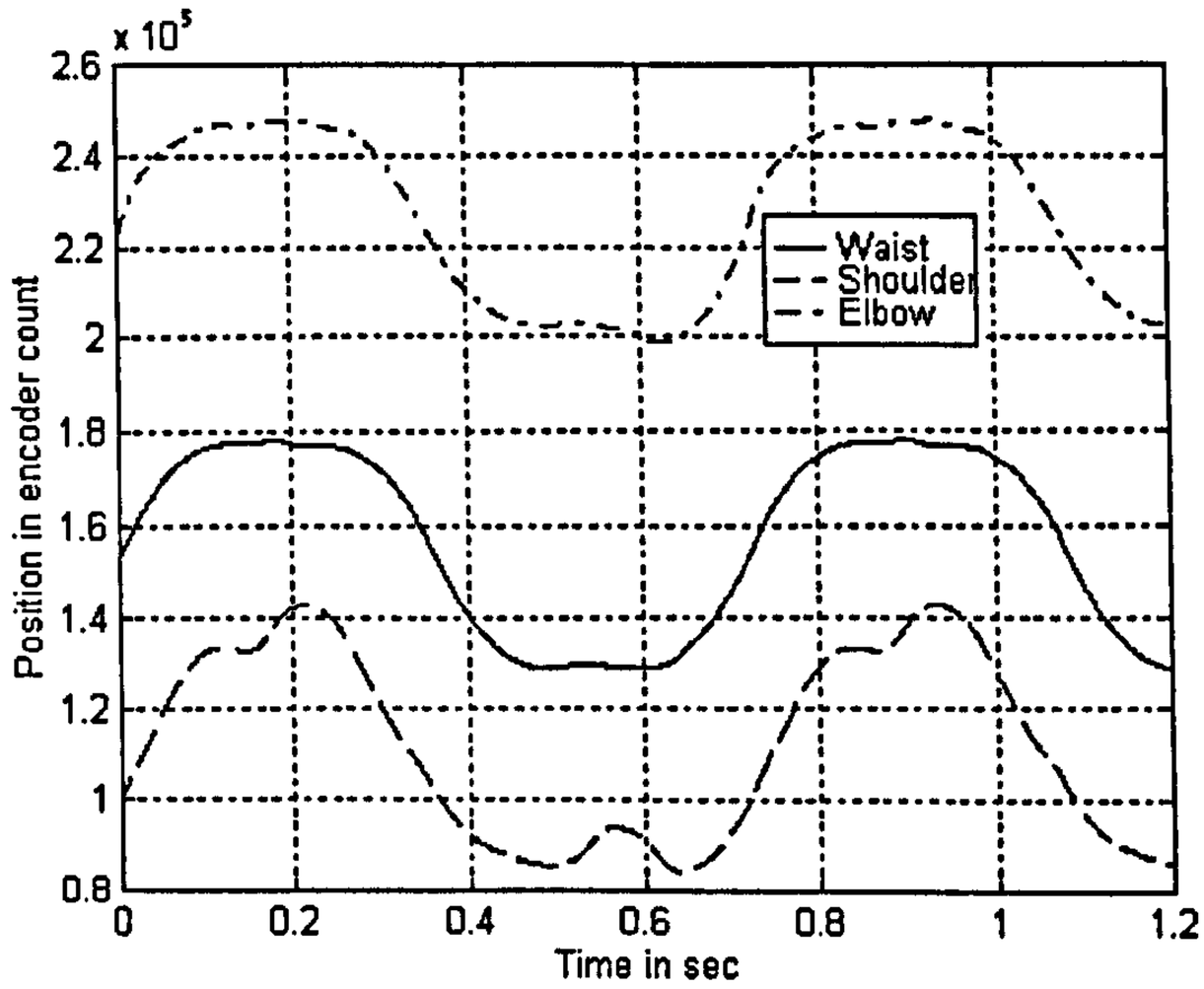


Figure 46: Response Trajectories in Motor Co-ordinates for the Goalpost Test

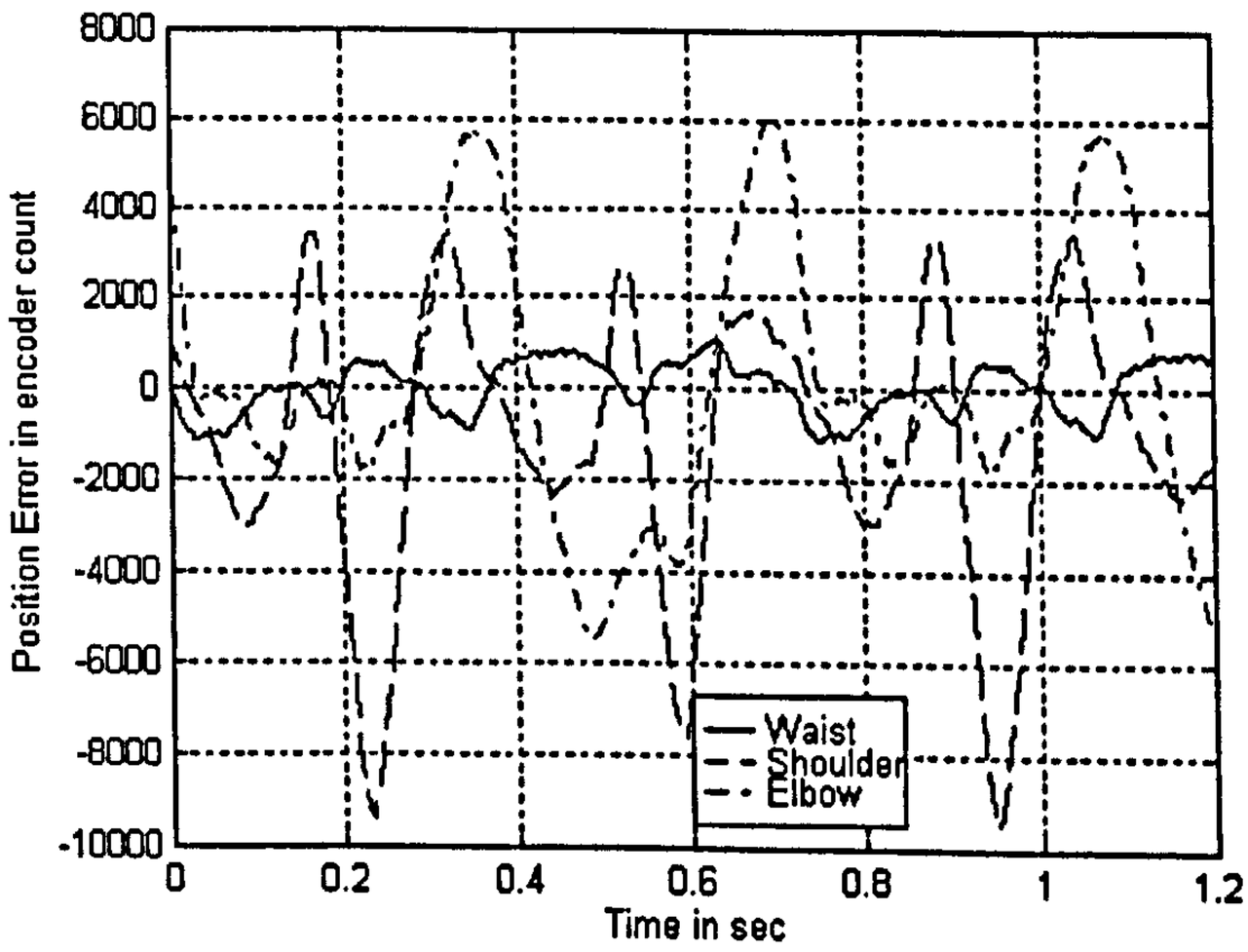


Figure 47: Position Errors in Motor Co-ordinates for the Goalpost Test

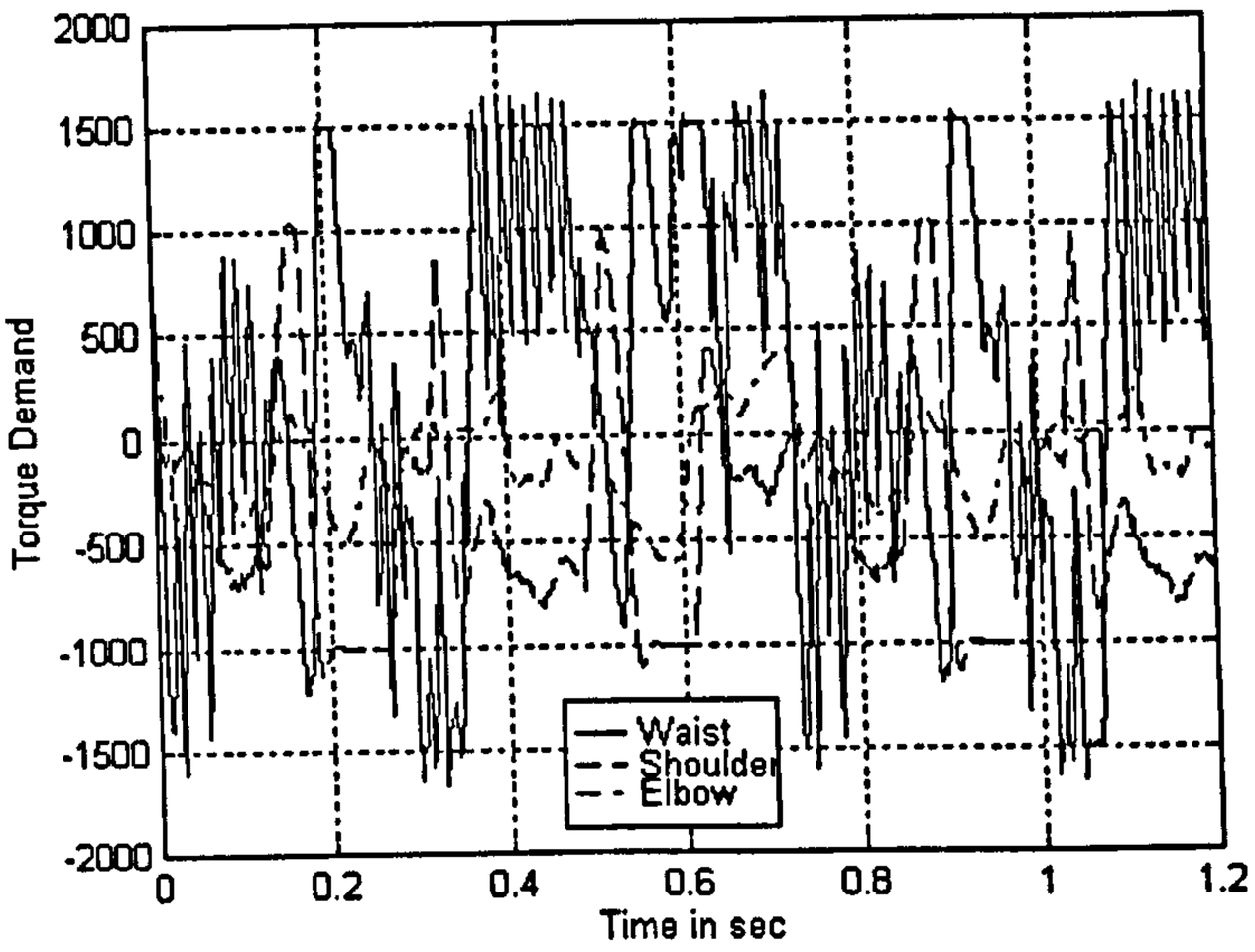


Figure 48: Torque Demands for the Goalpost Test

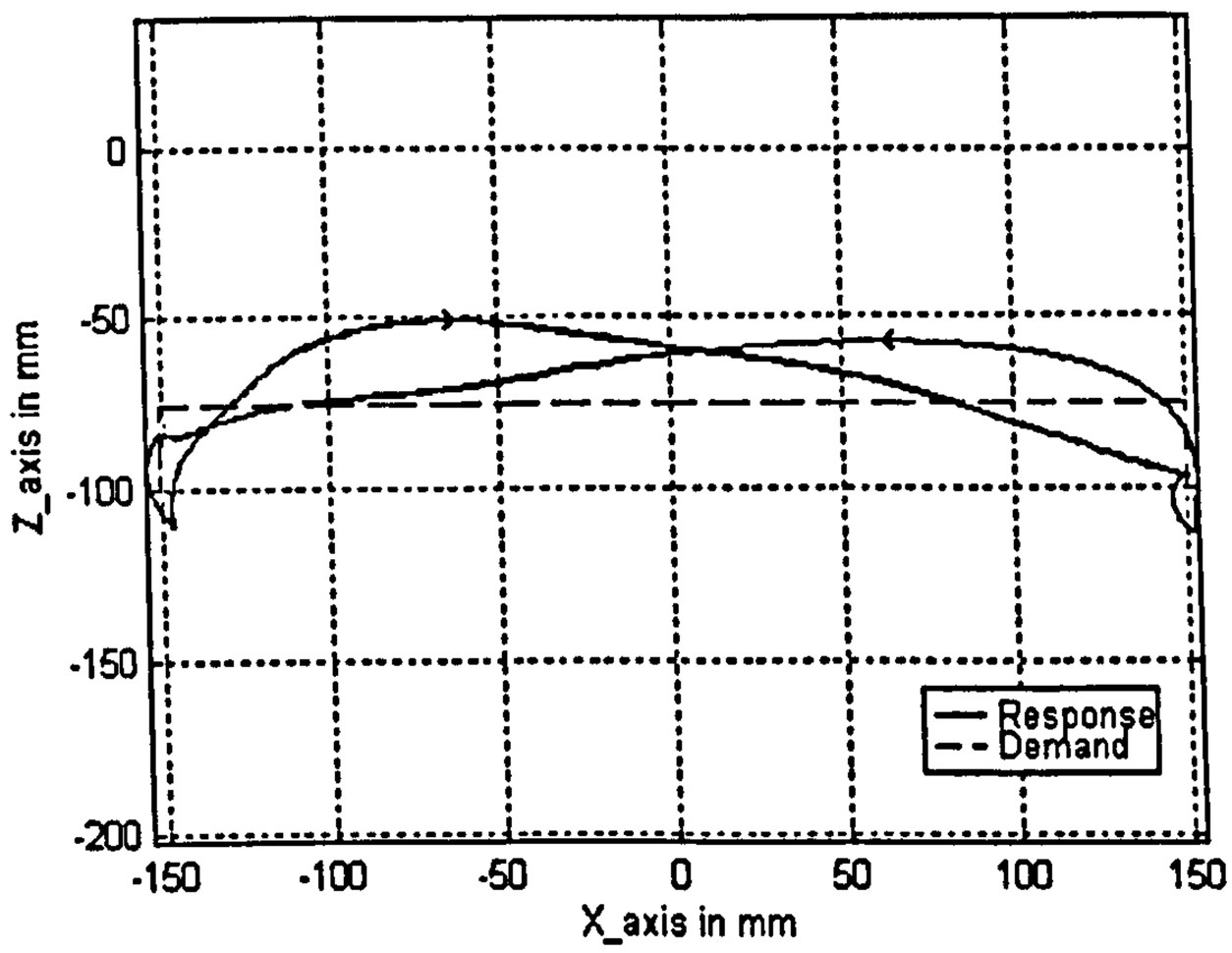


Figure 49: Front View of the End Point Displacement for the Goalpost Test

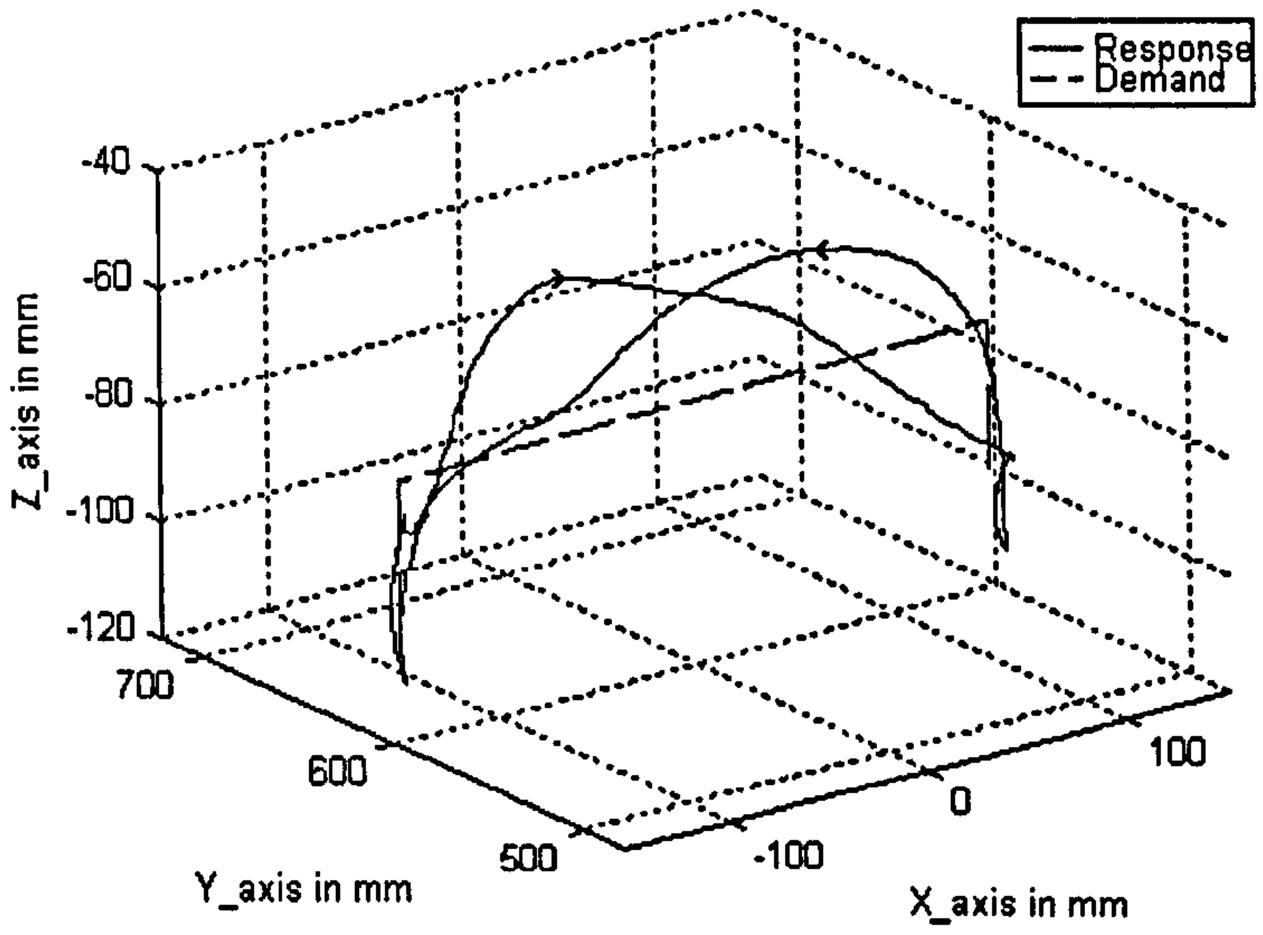


Figure 50: 3D View of the End Point Displacement for the Goalpost Test

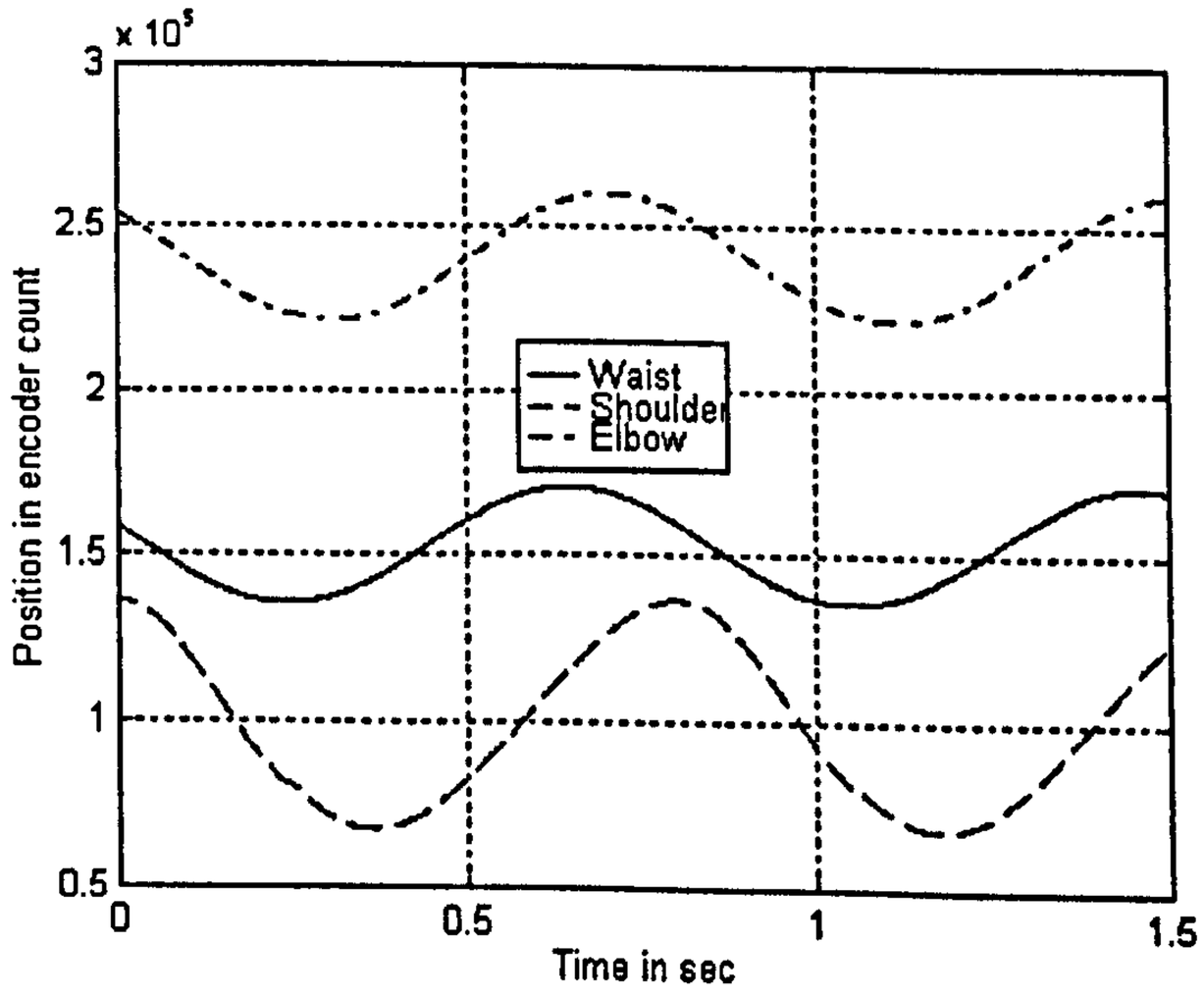


Figure 51: Circle Trajectory Demands in Motor Co-ordinates

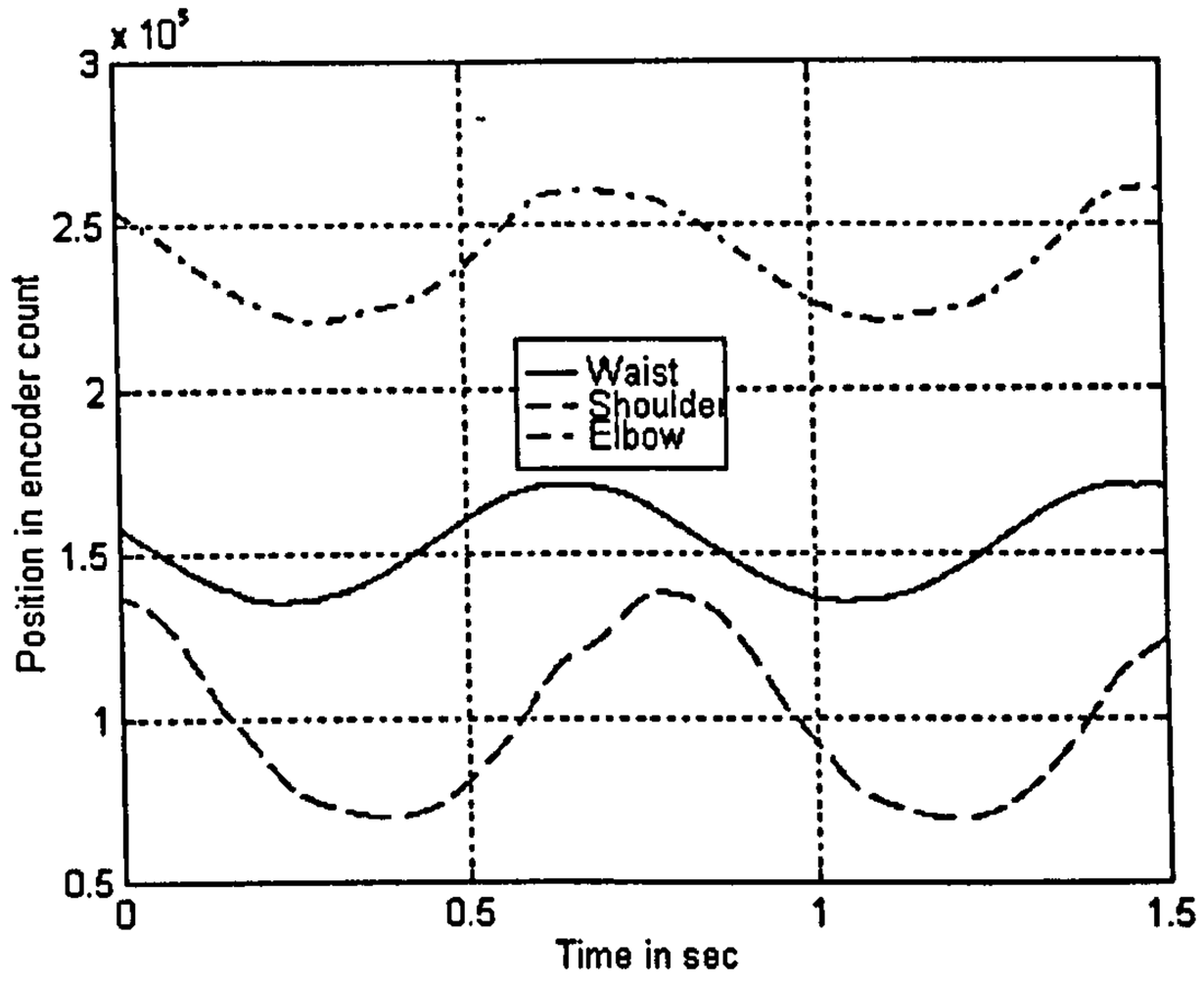


Figure 52: Response Trajectory in Motor Co-ordinates for the Circle Motion

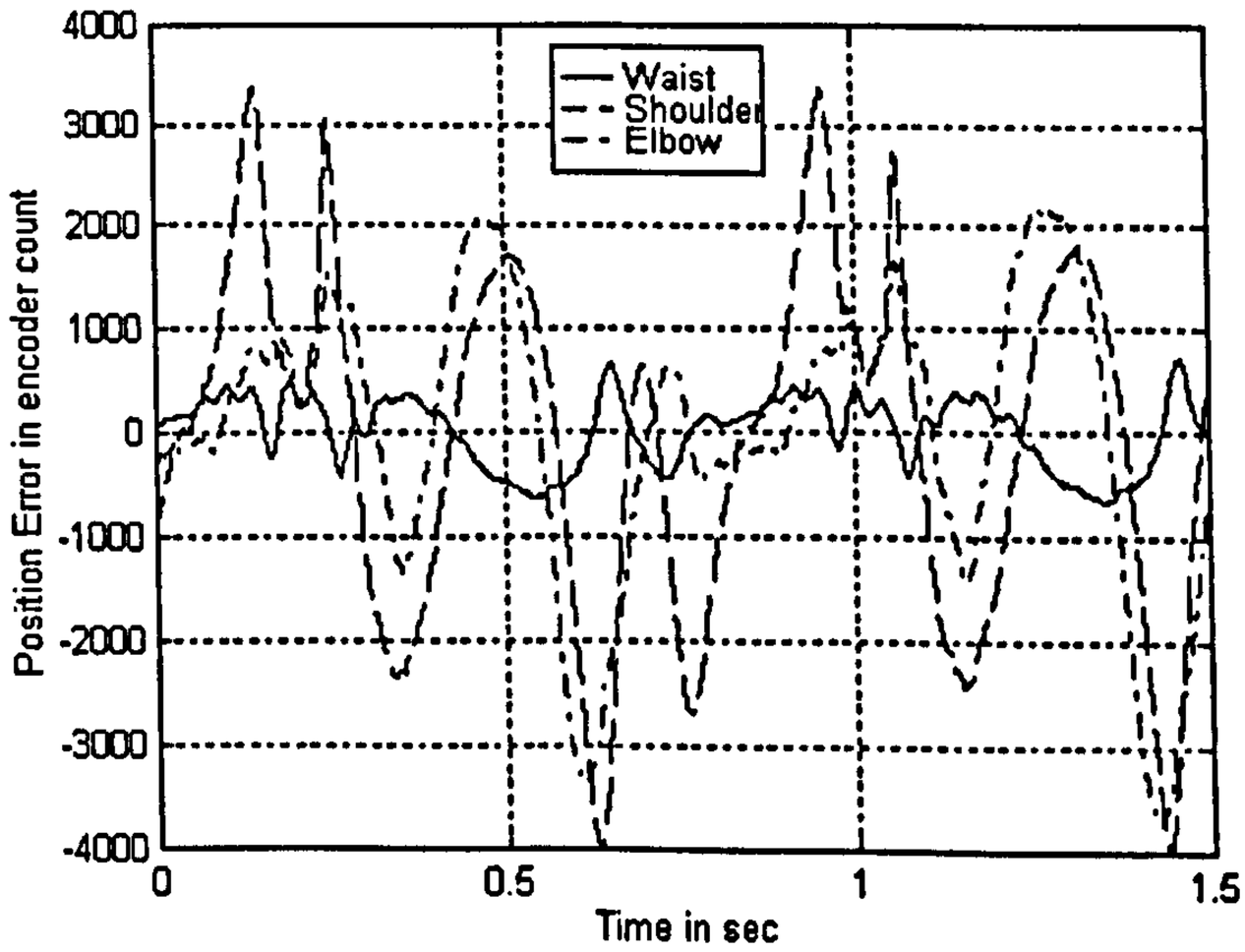


Figure 53: Position Errors in Motor Co-ordinates for the Circle Trajectory

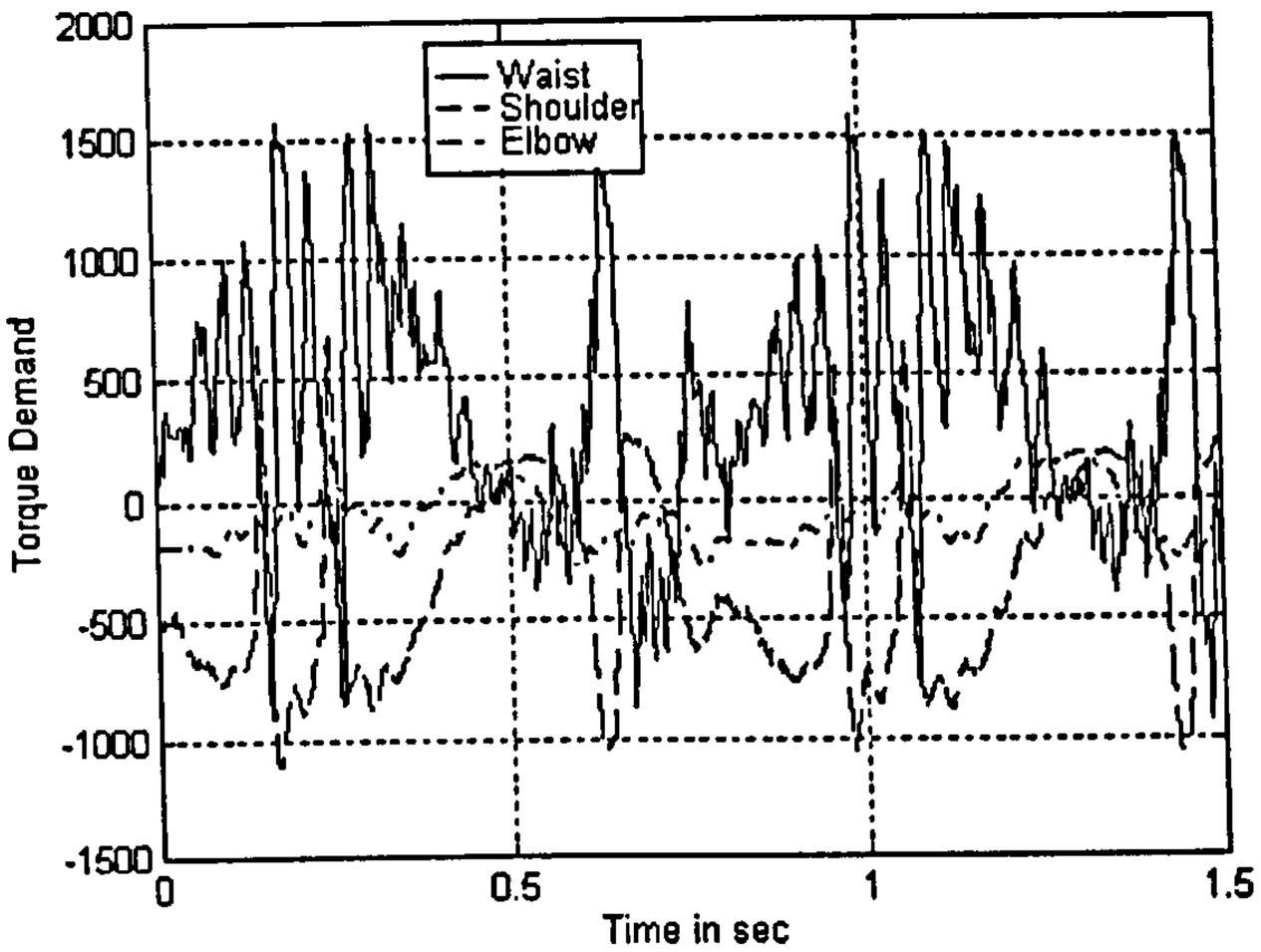


Figure 54: Torque Demands for the Circle Motion

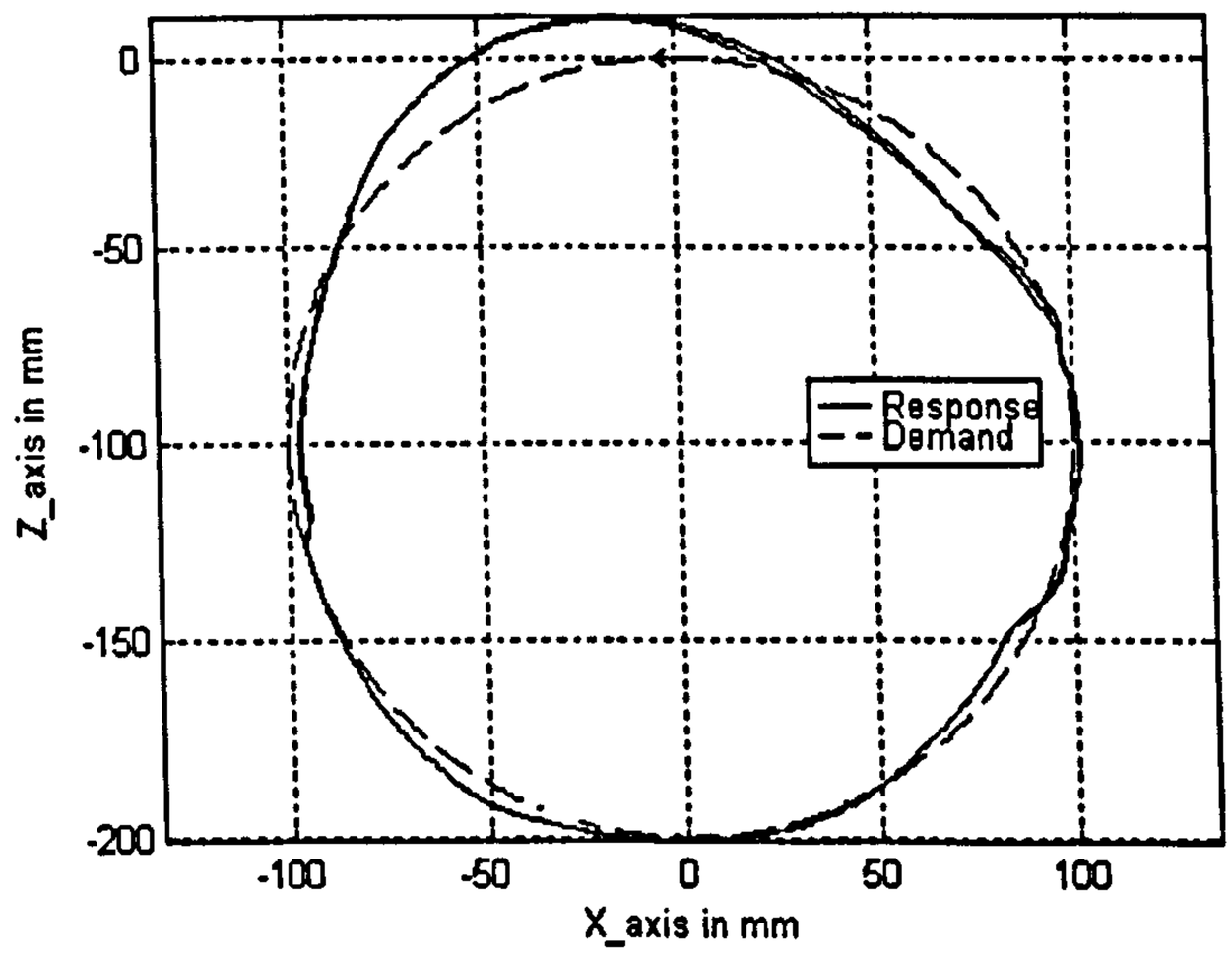


Figure 55: Front View of the End Point Displacement for the Circle Trajectory

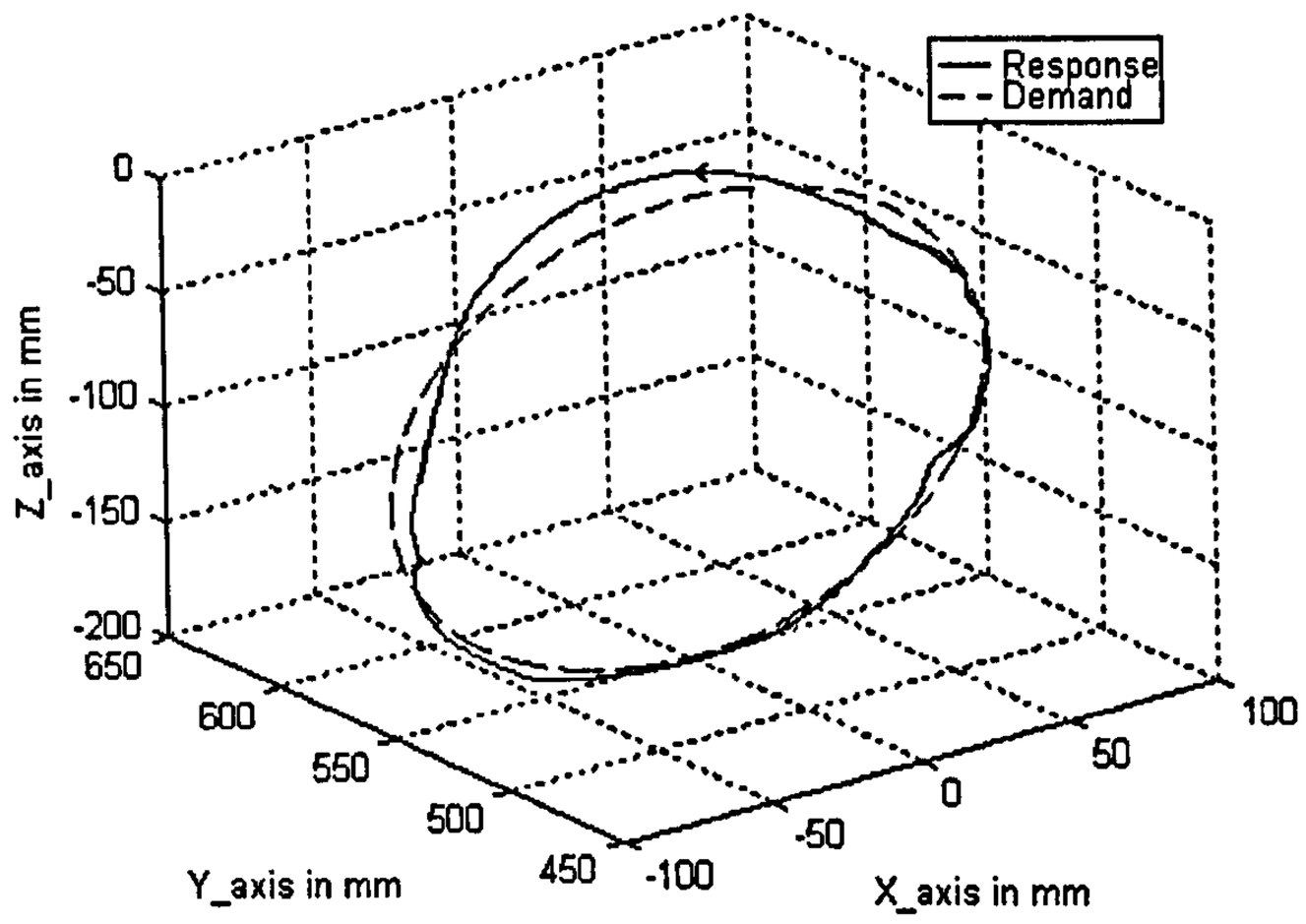


Figure 56: 3 D View of the End Point Displacement for the Circle Trajectory

The maximum position errors for the waist, shoulder and elbow joints are 668, 3998 and 3369 encoder pulses or $0.007rad$, $0.041rad$ and $0.035rad$ respectively. In Cartesian co-ordinates, the maximum error represents a deviation of the end point from the demand trajectory of the circle of less than $15mm$, as can be seen from figures 55 and 56.

6.3 Second Controller Implementation and Evaluation

The second controller that is implemented on the SPRINTA, is an improved version of the previous controller. It has the same control structure as the first one, equation (97). The difference is in the way the bounds $K_1(\Delta, \Gamma, \Psi)$ and $K_2(\Delta, \Gamma, \Psi)$ are adjusted with respect to the position in the phase plane and the equivalent end point position in motor co-ordinates. The transition from one zone to another is made gradual, during this transition the bounds are varying progressively. The three previously defined zones in the phase plane, figure (33) are kept and transition regions are defined around their limits. Similarly, the transition from the inside to the outside of the circle defined in motor co-ordinates, figure 35 is smoothed. Two circles (Circle 1 and Circle 2) concentric to the circle of figure 35 are defined, with

Circle 1 inside the circle of figure 35, and Circle 2 outside. Circle 1 and Circle 2 define the beginning and the end of the transition region for the smoothing of the bounds. Figures 57 and 58 show the adjustment of bounds $K_1(\Delta, \Gamma, \Psi)$ and $K_2(\Delta, \Gamma, \Psi)$ respectively, with respect to the zones and endpoint location. In figures 57 and 58, the magnitudes of the bound are given as percentages of the maximum value as the values are different for each joint.

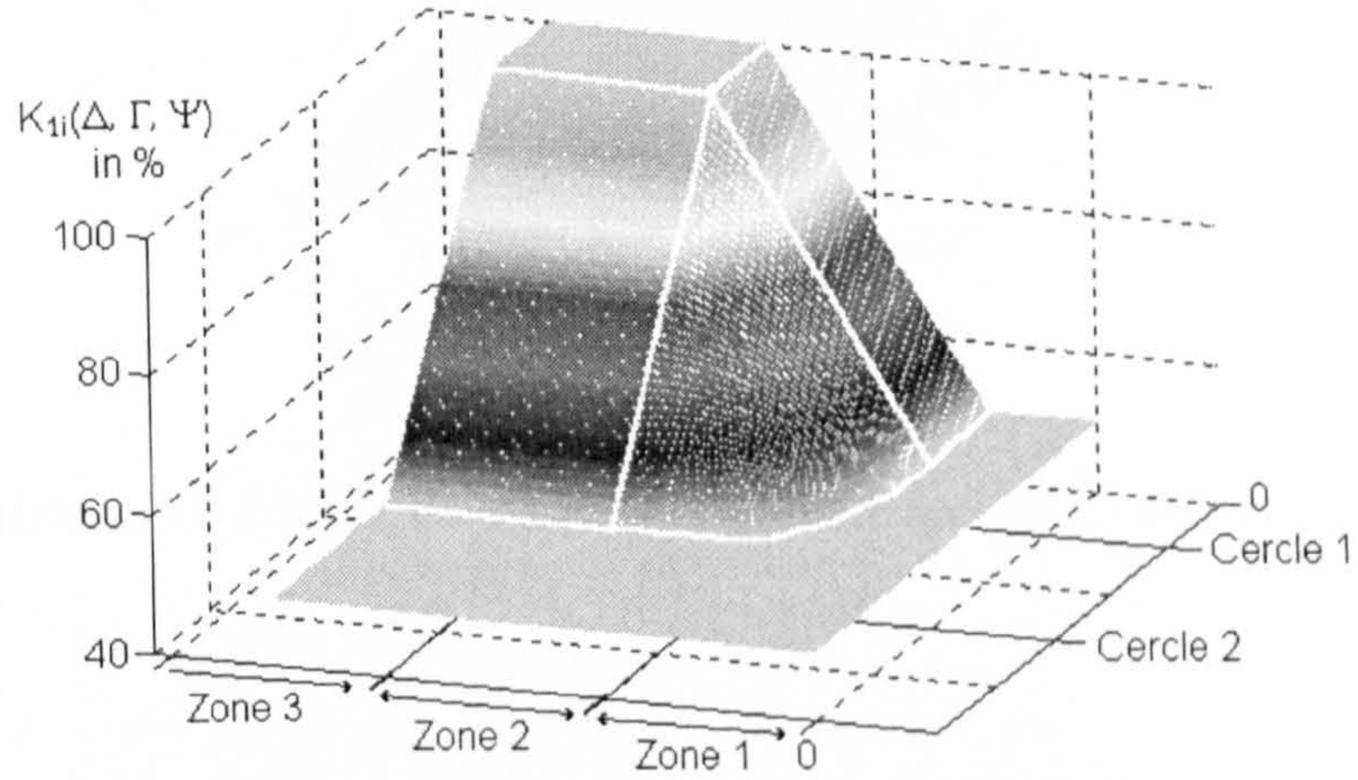


Figure 57: Variation of $K_{1i}(\Delta, \Gamma, \Psi)$ with Respect to the Zones and the End Point Location

As previously, the controller, equation (97) with the improved bound adjustment is implemented on the SPRINTA and evaluated on the straight line, goalpost test and circle trajectory.

6.3.1 Straight Line Motion

The straight line test is executed between the Cartesian location $[650\ 0\ -100]_{XYZ}$ to the location $[0\ 550\ -100]_{XYZ}$ in 1 sec, (all co-ordinates are in mm). Rectilinear interpolation is used for this movement, the sampling time for the controller is set to $1.5 milli$ sec. The position demands for the shafts of the three motors are given in figure 38. The system response is shown in figures 59-61. Figure 62 is a top view of the demand and actual end point trajectory of the SPRINTA robot, whilst figure 63 is a perspective in 3 dimensions of the point to point trajectory.

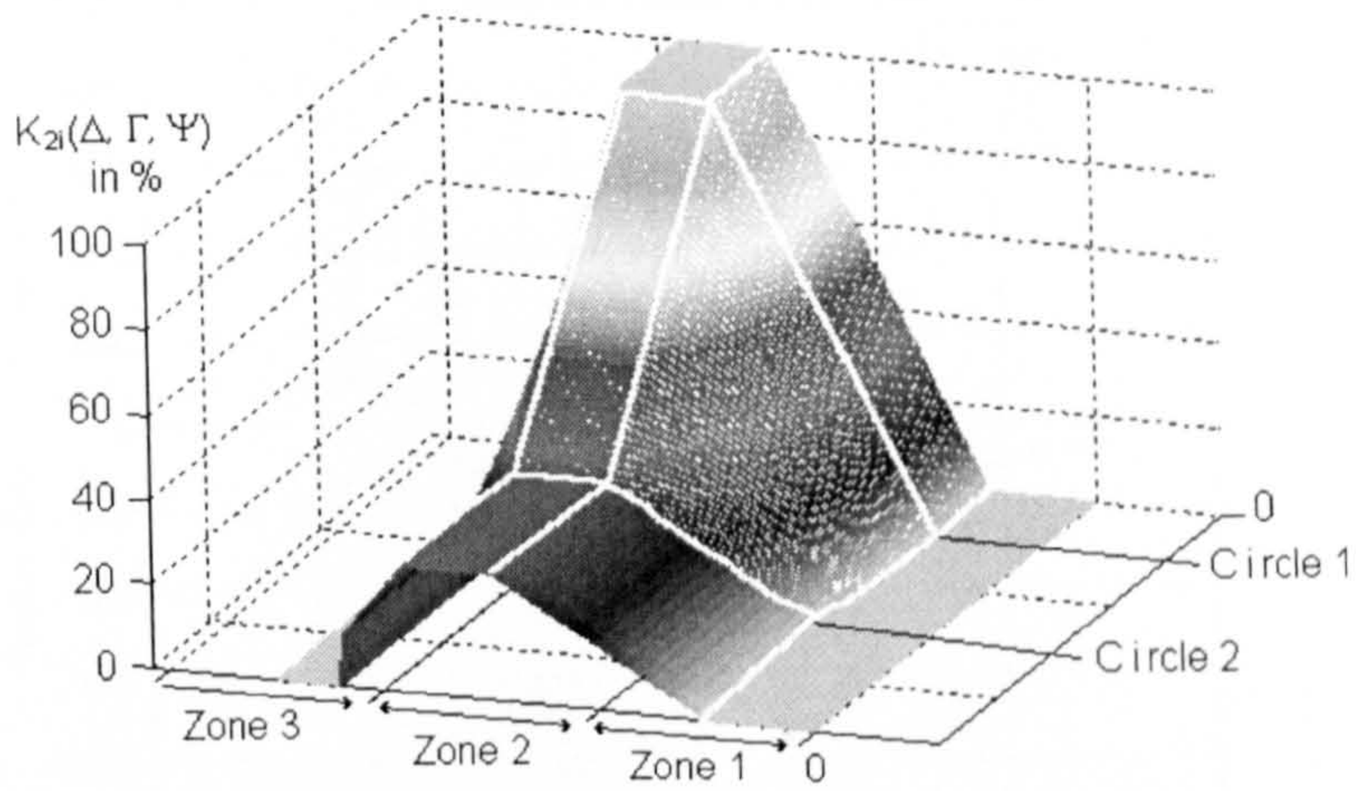


Figure 58: Variation of $K_{2i}(\Delta, \Gamma, \Psi)$ with Respect to the Zones and the End Point Location

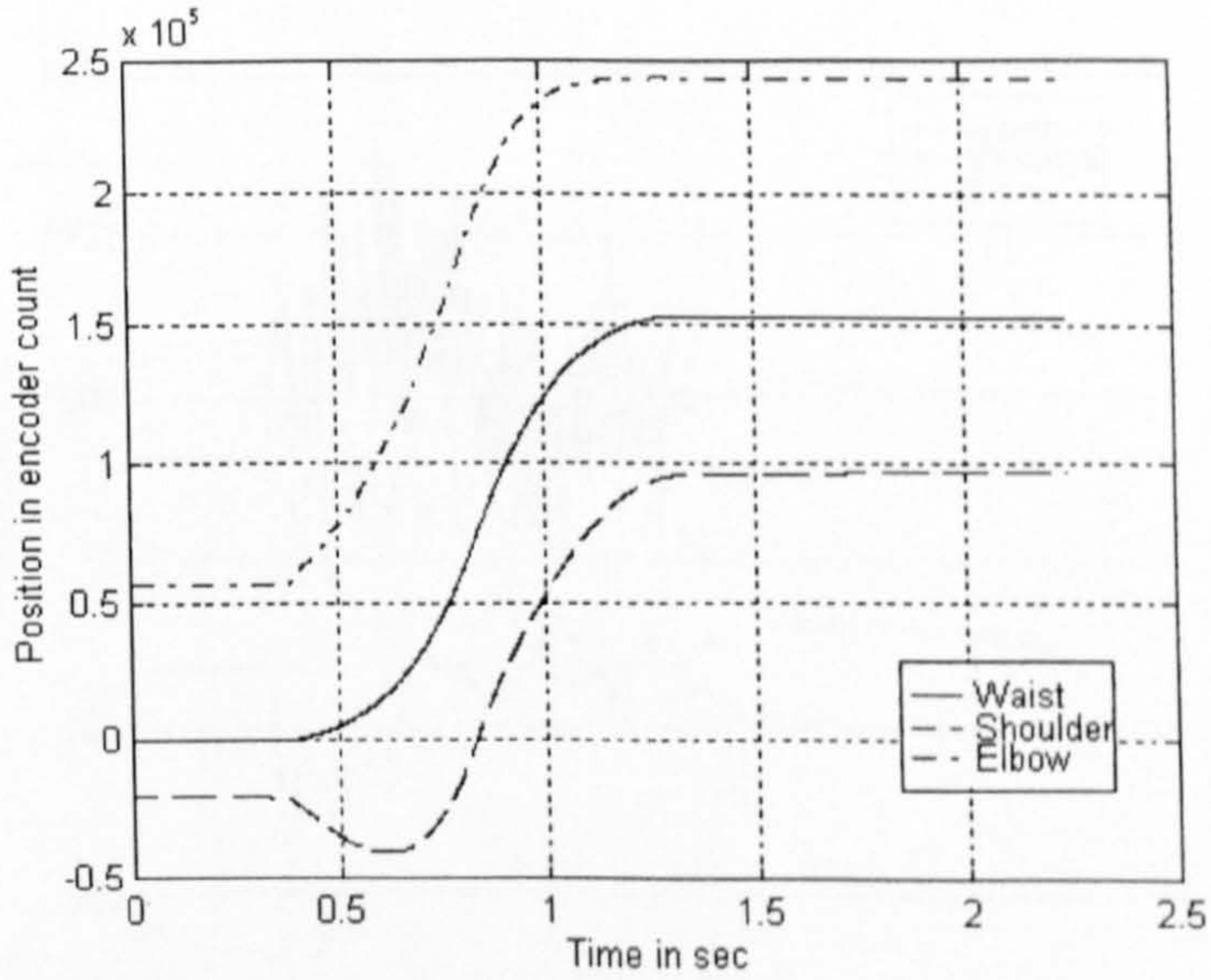


Figure 59: Response Trajectrories in Motor Co-ordinates for the Straight Motion

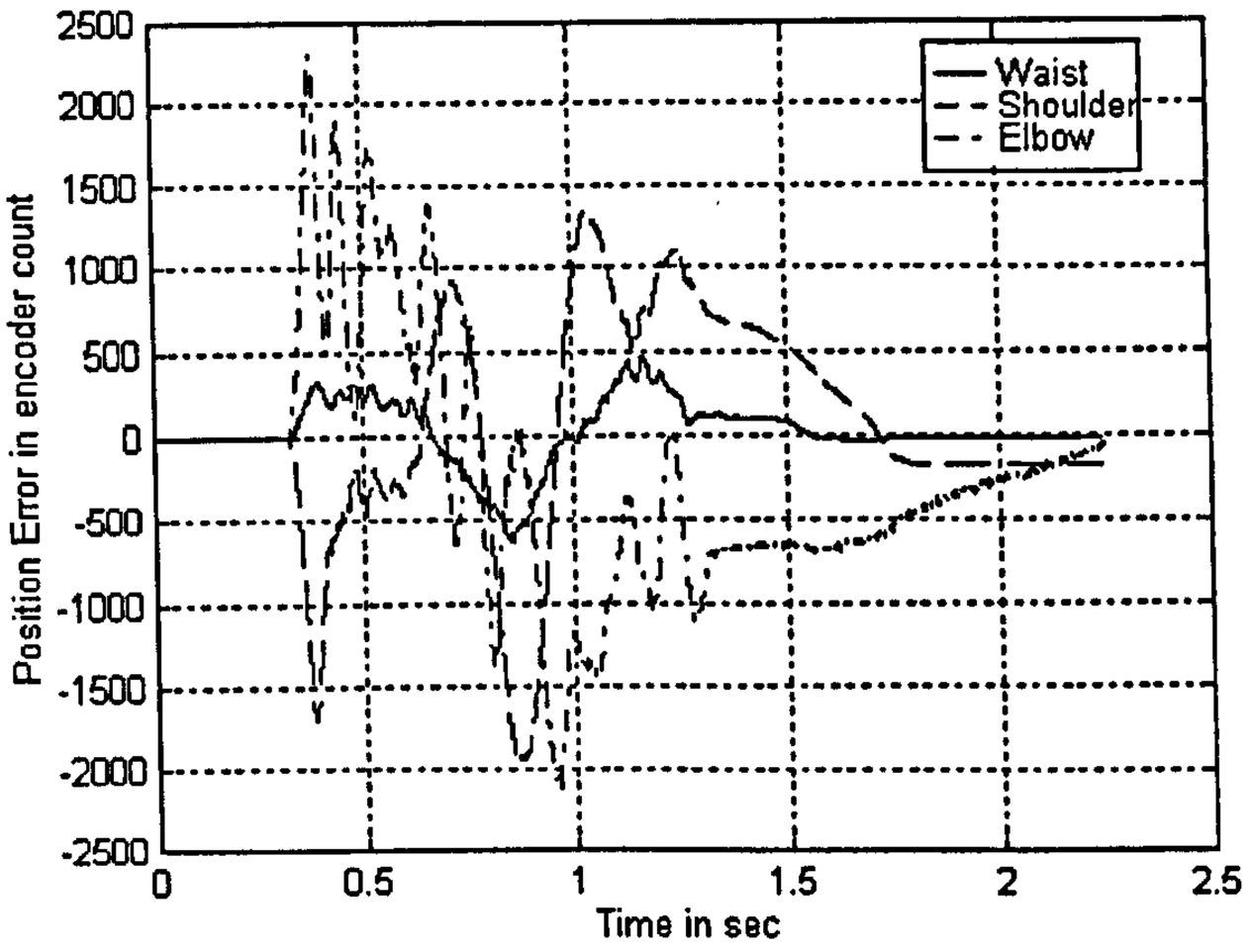


Figure 60: Position Errors in Motor Co-ordinates for Straight Line Trajectory

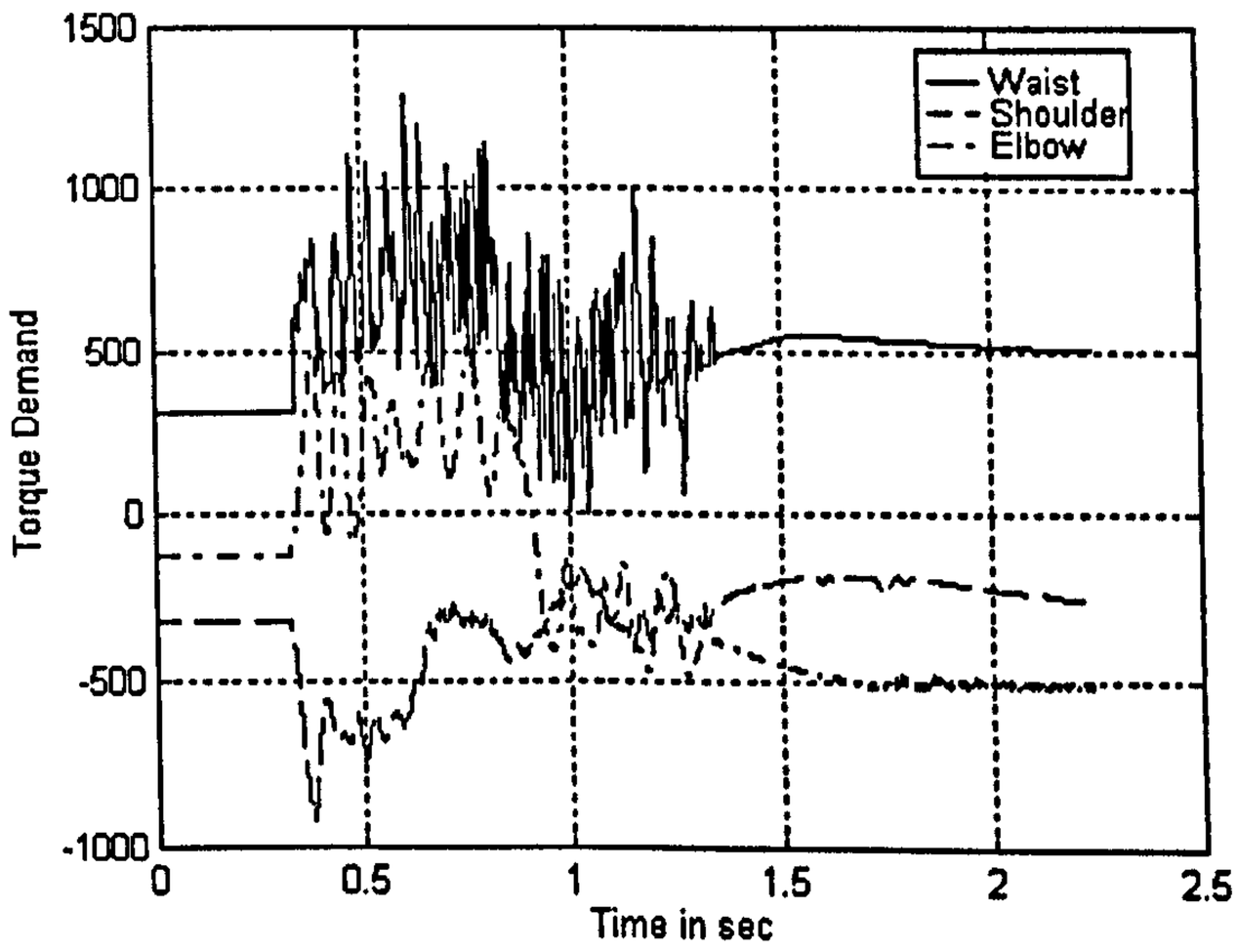


Figure 61: Torque Demands for Straight Line Motion

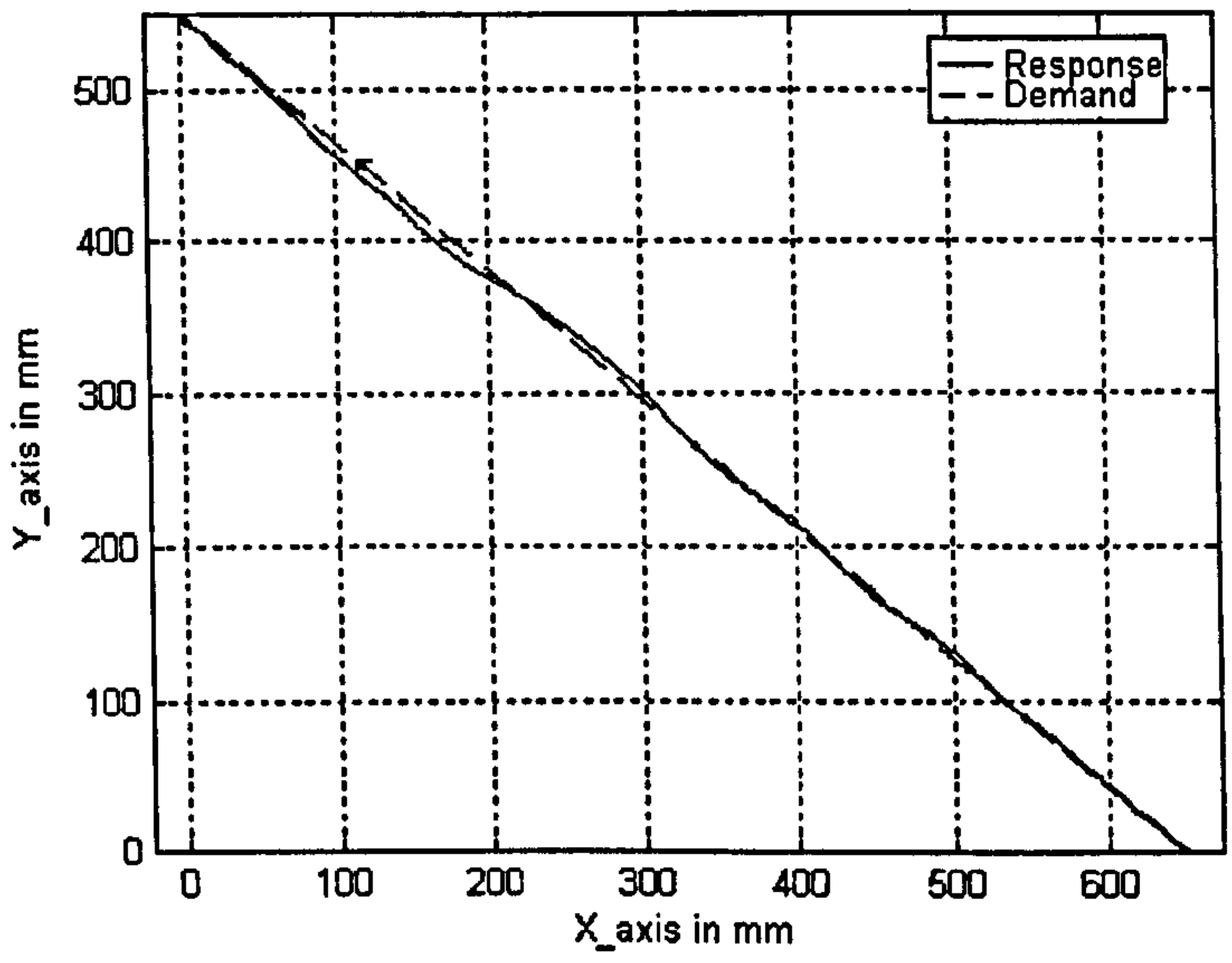


Figure 62: Top View of the End Point Displacement for the Straight Line Trajectory

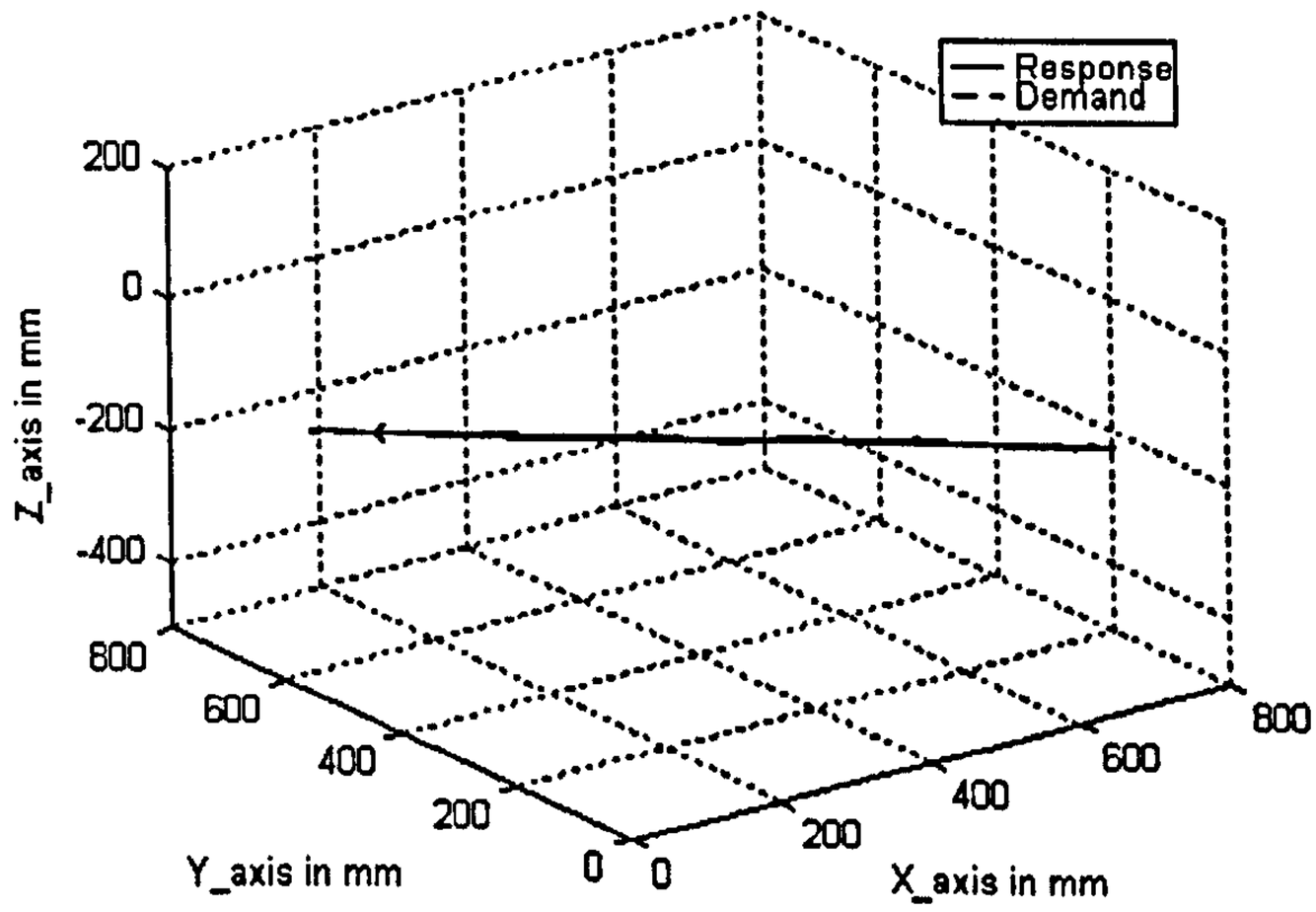


Figure 63: 3D View of the End Point Displacement for the Straight Line Trajectory

The maximum position errors for the waist, shoulder and elbow joints are 628, 1949 and 2301 encoder pulses i.e. $0.0064rad$, $0.02rad$ and $0.024rad$ respectively. The maximum errors for the waist and shoulder joint are nearly the same as for the previous controller, the maximum position error for the elbow joint is however greatly reduced and is nearly half the one obtained with the previous controller on this trajectory. At the end of the movement, the largest position error of the three motor shafts is less than 700 encoder pulses ($0.007rad$). As for the previous controller, the position errors converge to a steady state error of less than 200 encoder pulses or less than $0.002rad$ within 1 sec. The position error at the shaft of the three motors is translated to a maximum deviation of the end point of the SPRINTA in Cartesian co-ordinates, of less than $10mm$, as shown in figures 62 and 63.

6.3.2 Goalpost Test

The same parameters as used previously are used to test this controller (position and duration of the test, sampling time and interpolation method). The position demands for the shafts of the three motors are given in figure 45. The system response is shown in figures 64-66.

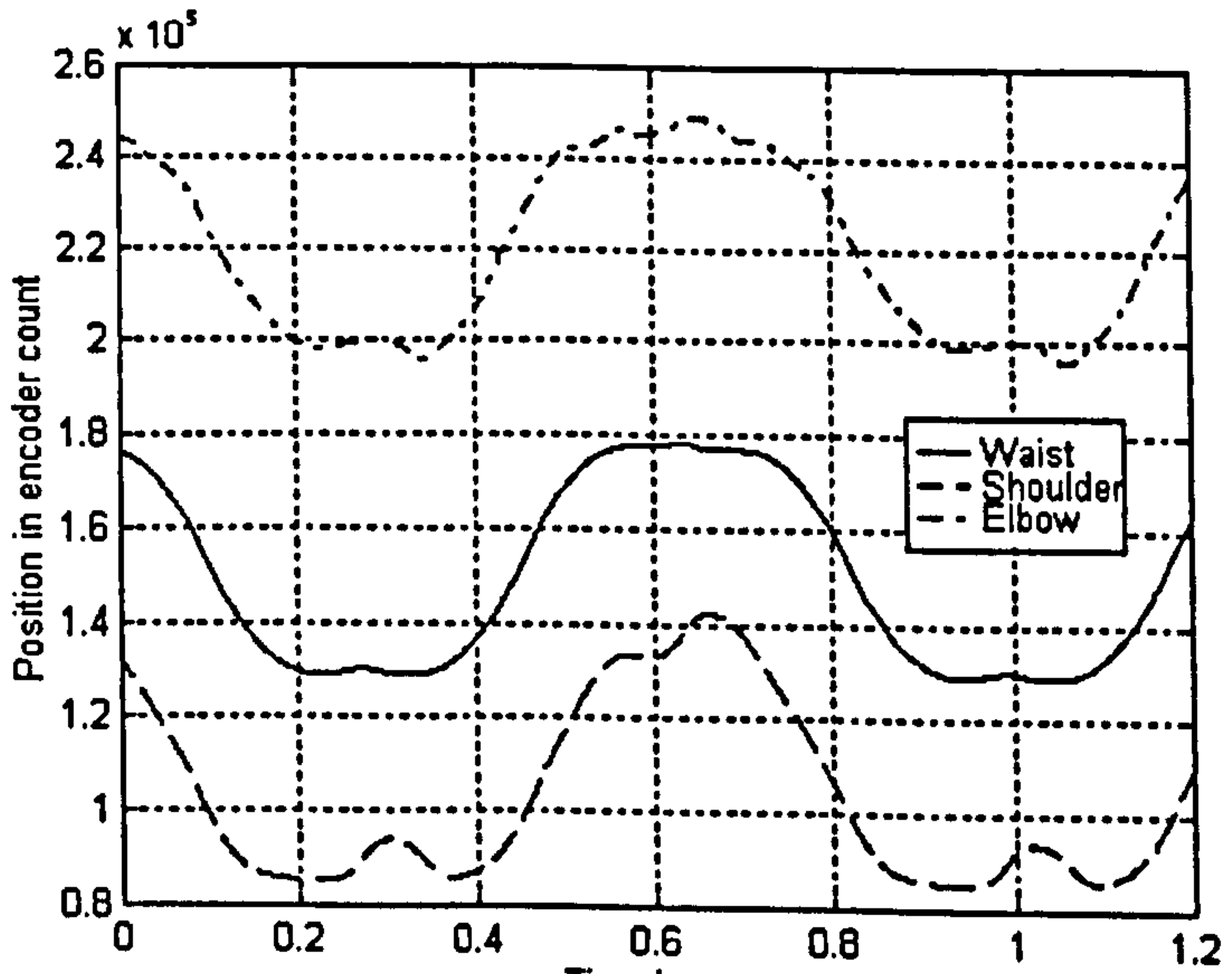


Figure 64: Response Trajectory in Motor Co-ordinates for Goalpost Test

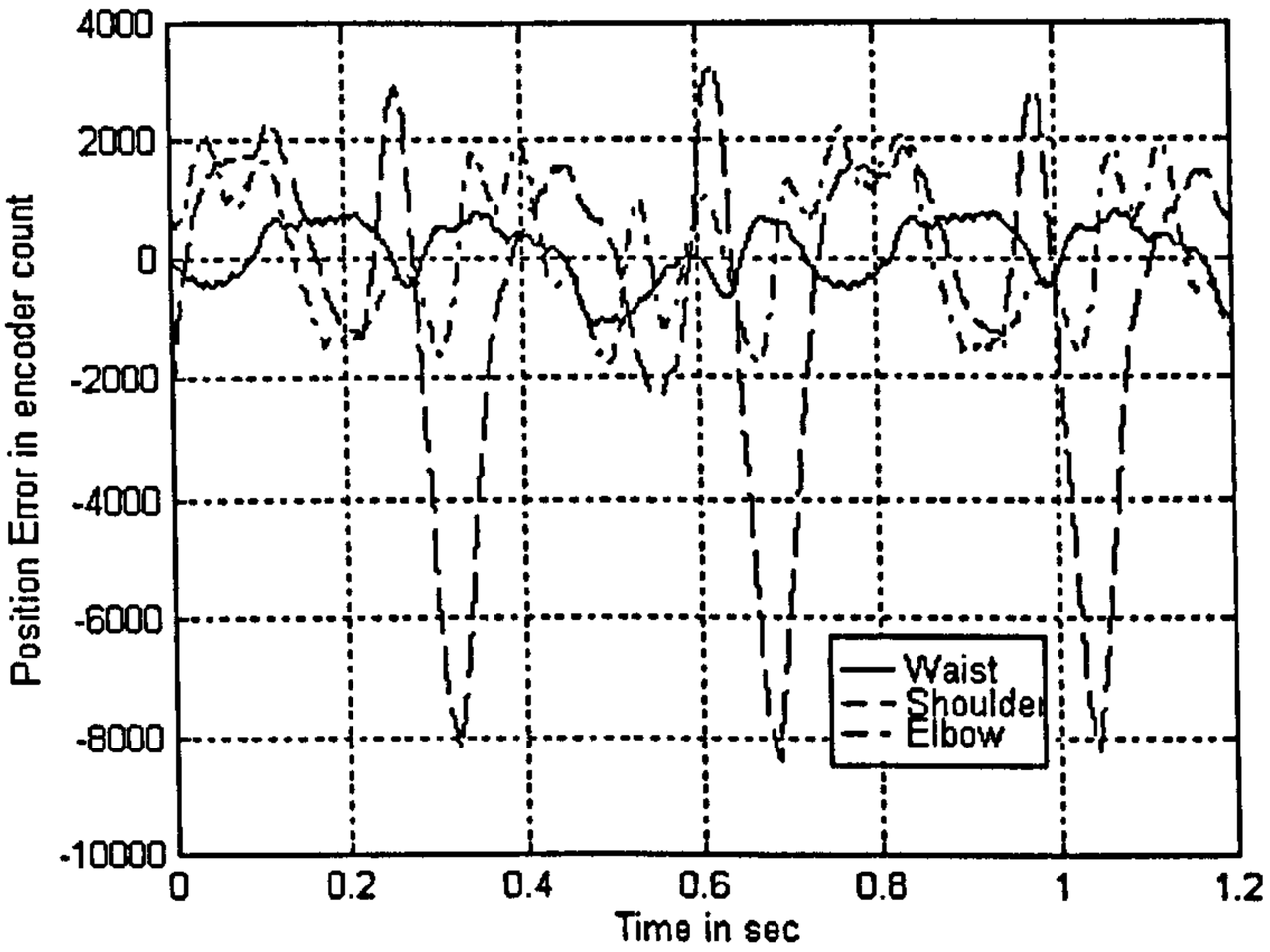


Figure 65: Position Errors in Motor Co-ordinates for the Goalpost Test

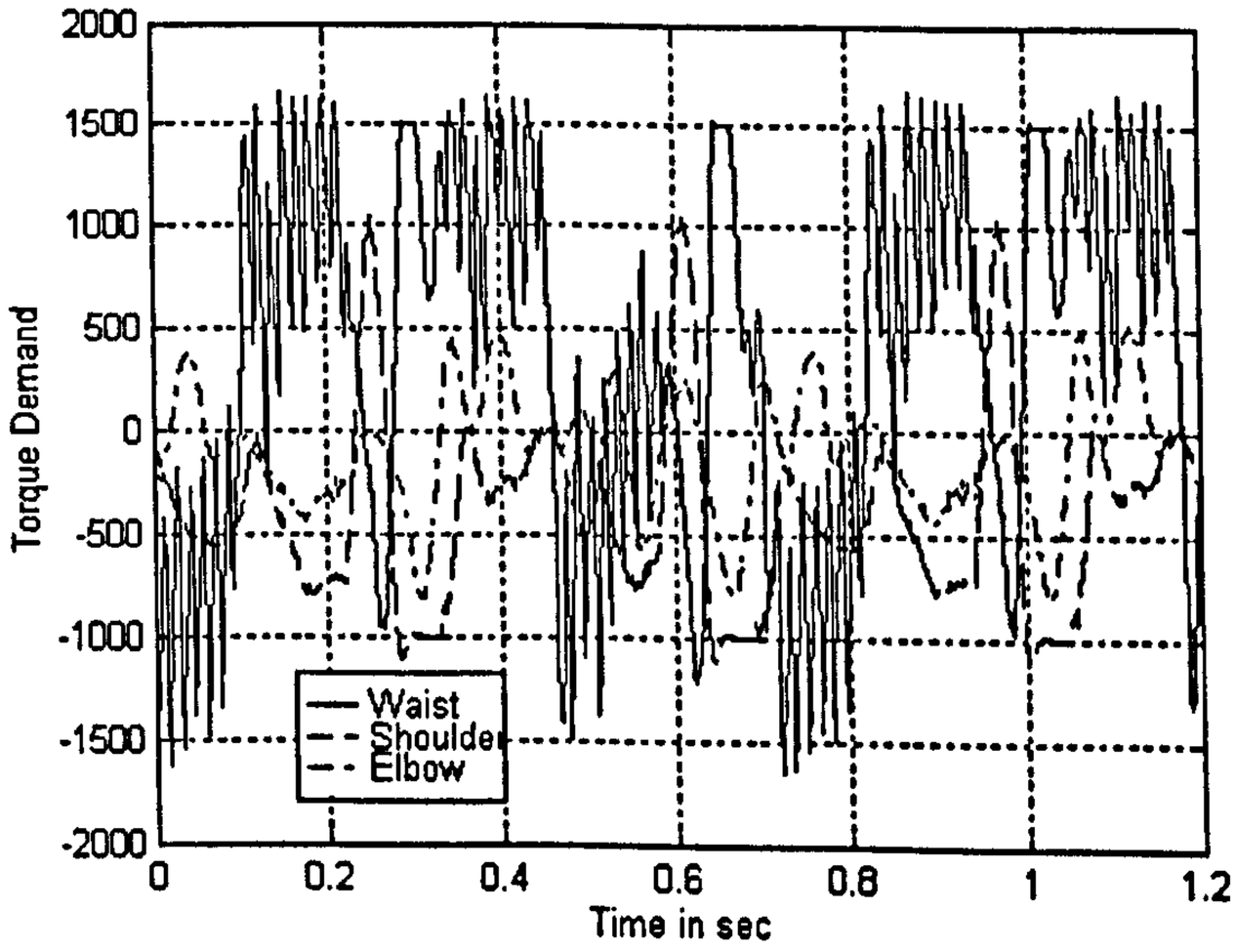


Figure 66: Torque Demands for the Goalpost Test

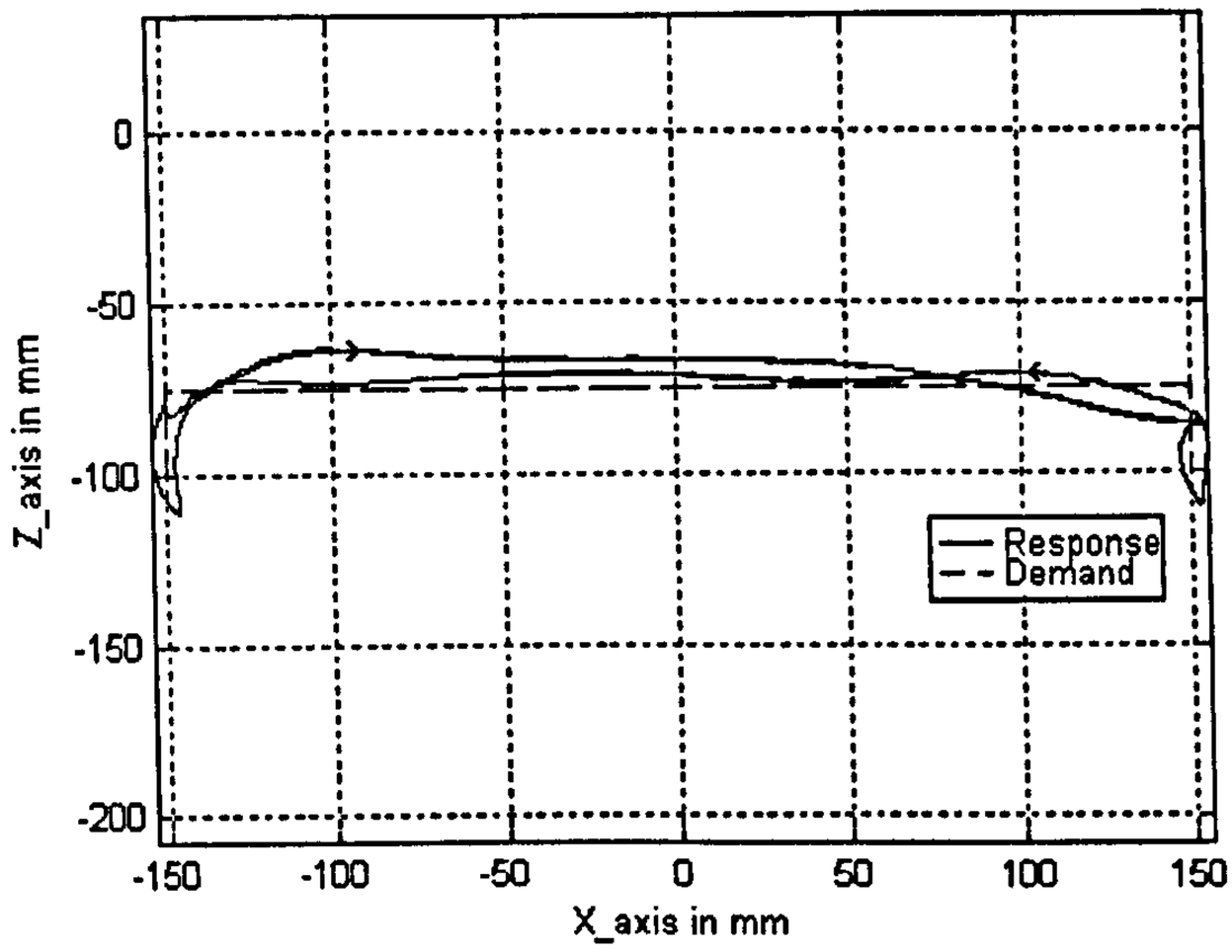


Figure 67: Front View of the End Point Displacement for the Goalpost Test

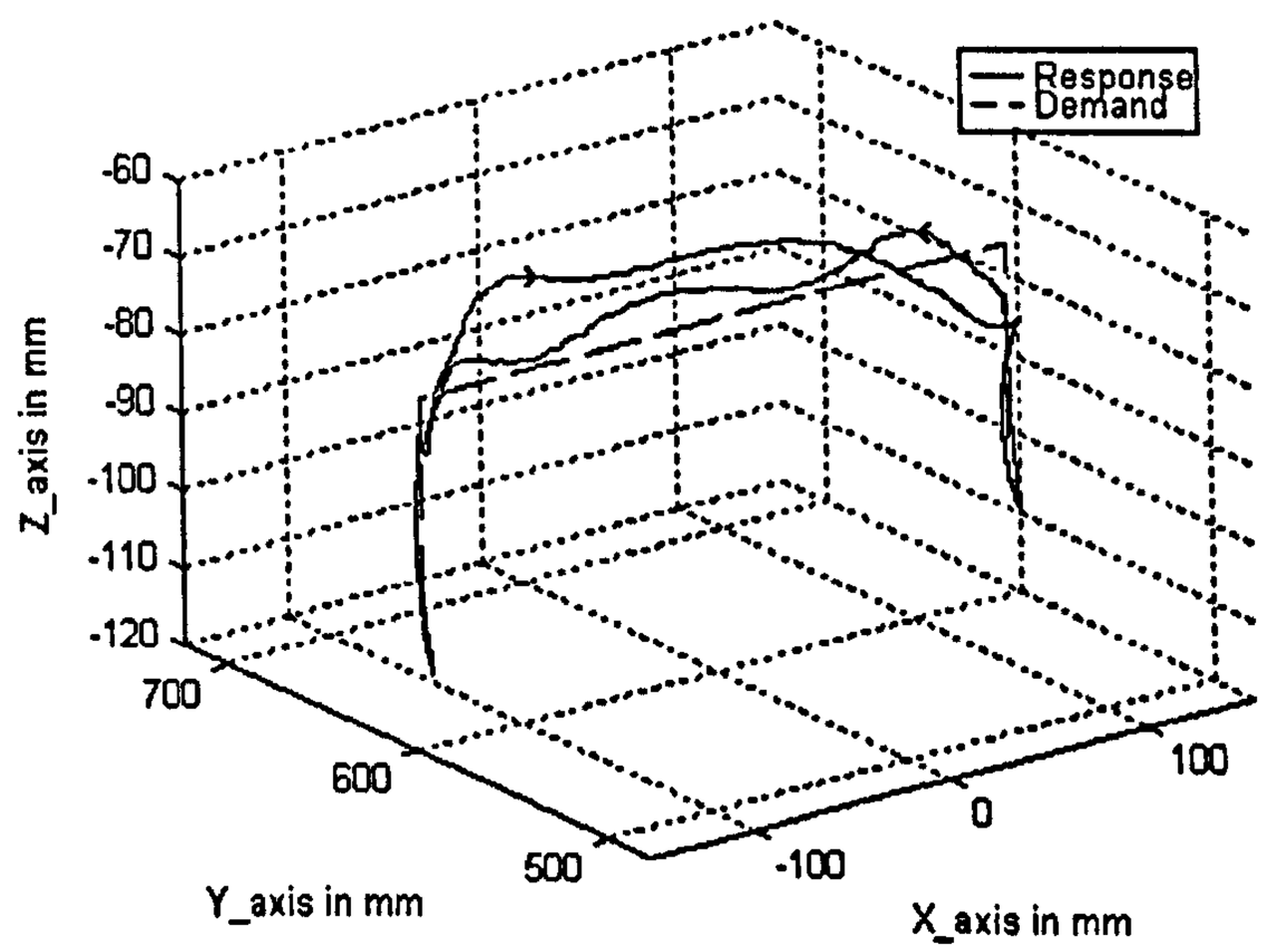


Figure 68: 3D View of the End Point Displacement for the Goalpost Test

The maximum position errors for the waist, shoulder and elbow joints are 1125, 8381 and 2122 encoder pulses or $0.012rad$, $0.086rad$ and $0.022rad$ respectively. Again the maximum position errors for the waist and shoulder joints are of the same order as those obtained with the first controller. The maximum position error for the elbow joint has however been significantly reduced, and is less than half the previous one. In Cartesian co-ordinates, the maximum error represents a deviation of the end point from the horizontal demand trajectory of less than $12mm$, as can be seen from figures 67 and 68.

6.3.3 Circle Trajectory in the Vertical Plane

The same parameters as used previously are used to test this controller (position and duration of the test, sampling time and interpolation method). The position demands for the shafts of the three motors are given in figure 51. The system response is shown in figures 69-71.

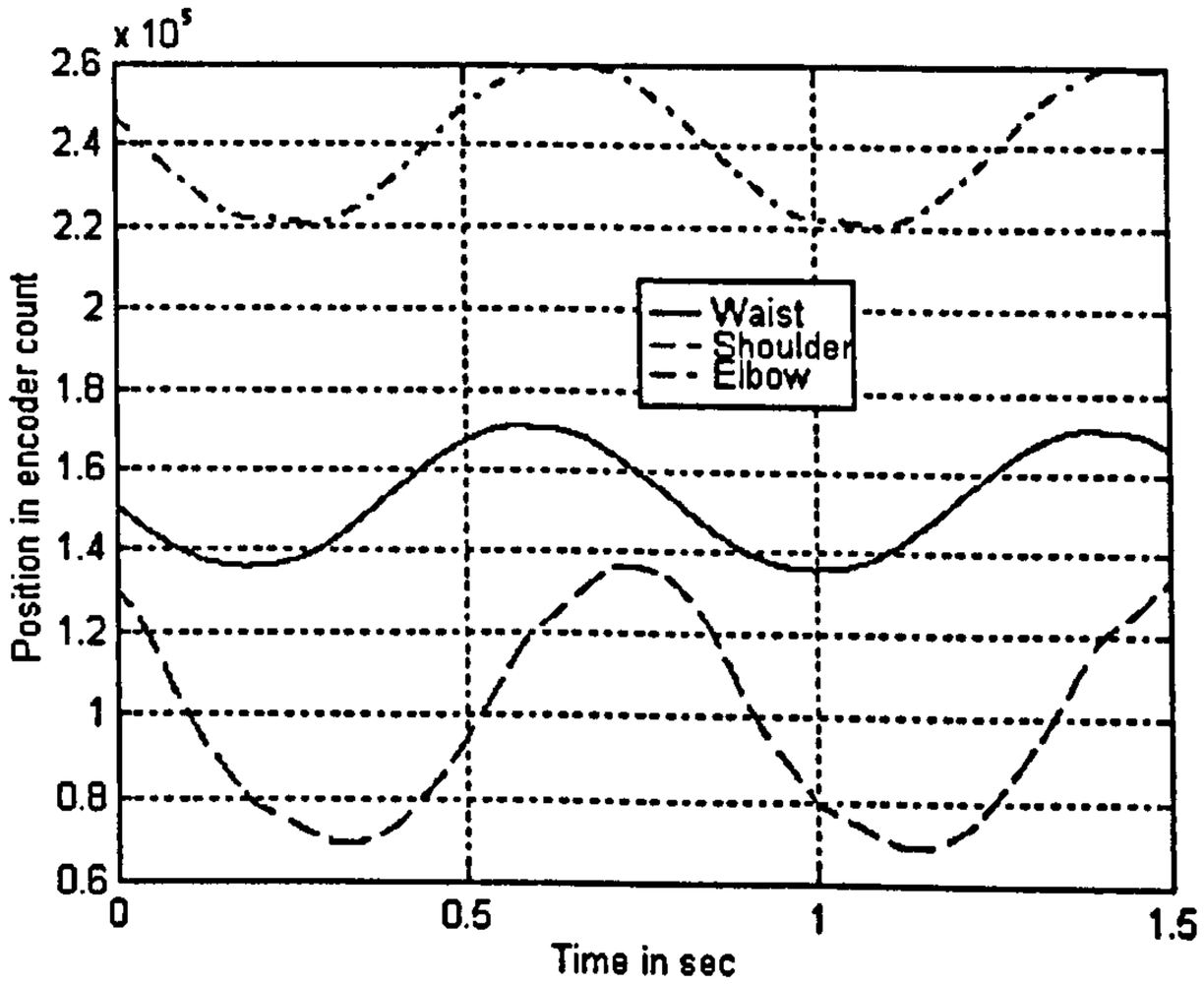


Figure 69: Response Trajectories in Motor Co-ordinates for the Circle Motion

The maximum position errors for the waist, shoulder and elbow joints are 545, 3567 and 1675 encoder pulses or $0.006rad$, $0.036rad$ and $0.017rad$ respectively. The improvement of the maximum position error with respect to the previous controller is mainly obvious on the elbow joint where the maximum position error is half the

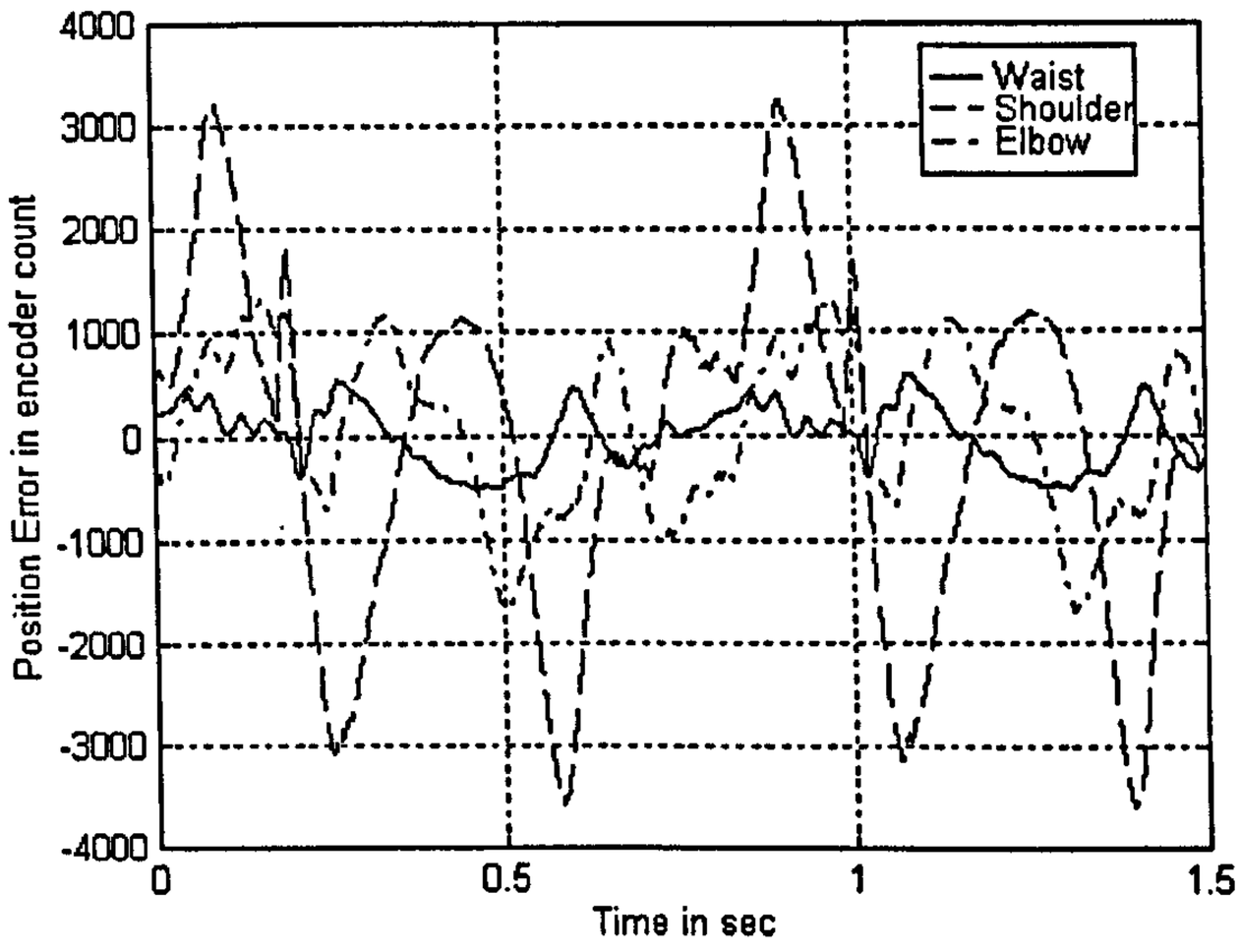


Figure 70: Position Errors in Motor Co-ordinates for the Circle Trajectory

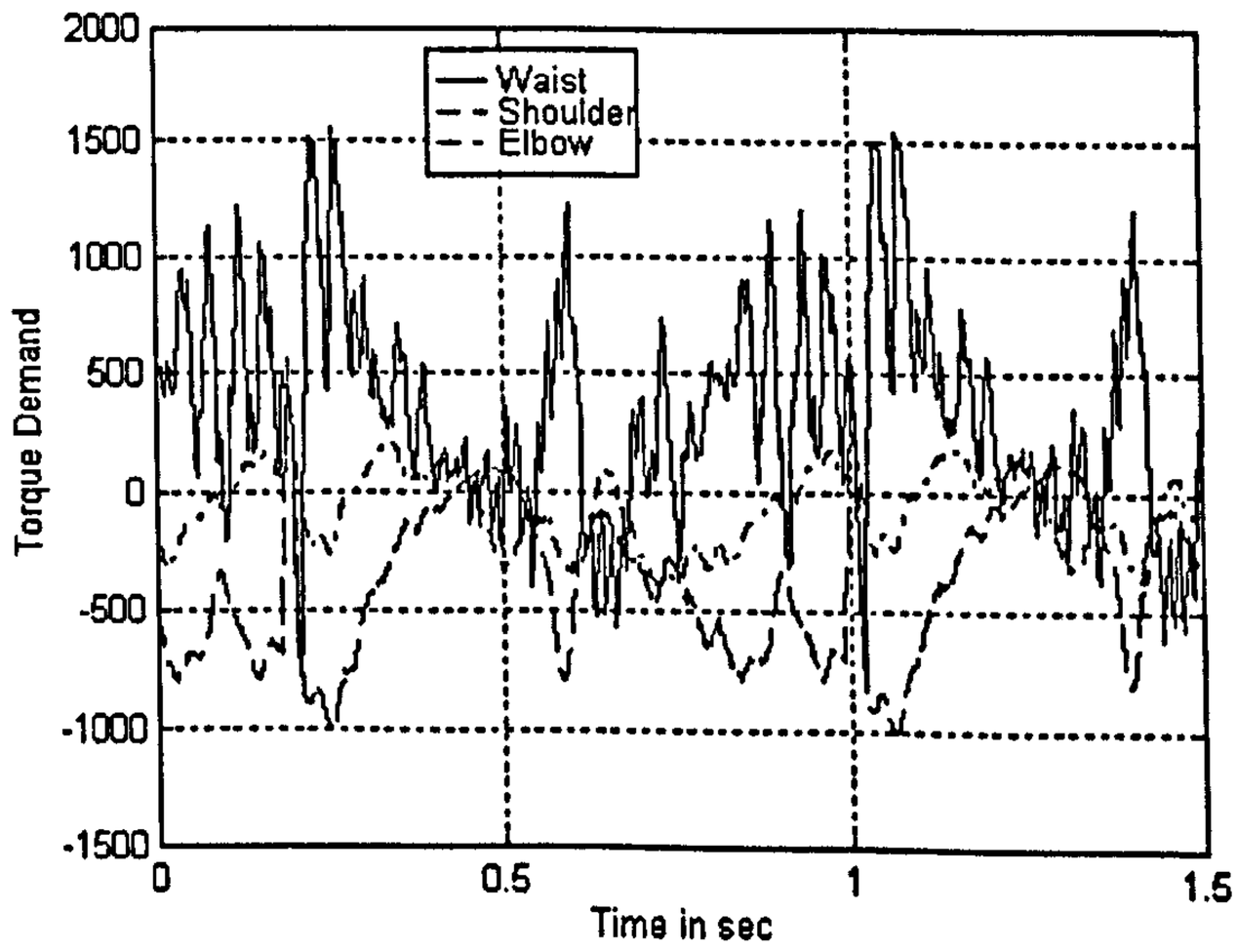


Figure 71: Torque Demands for the Circle Motion

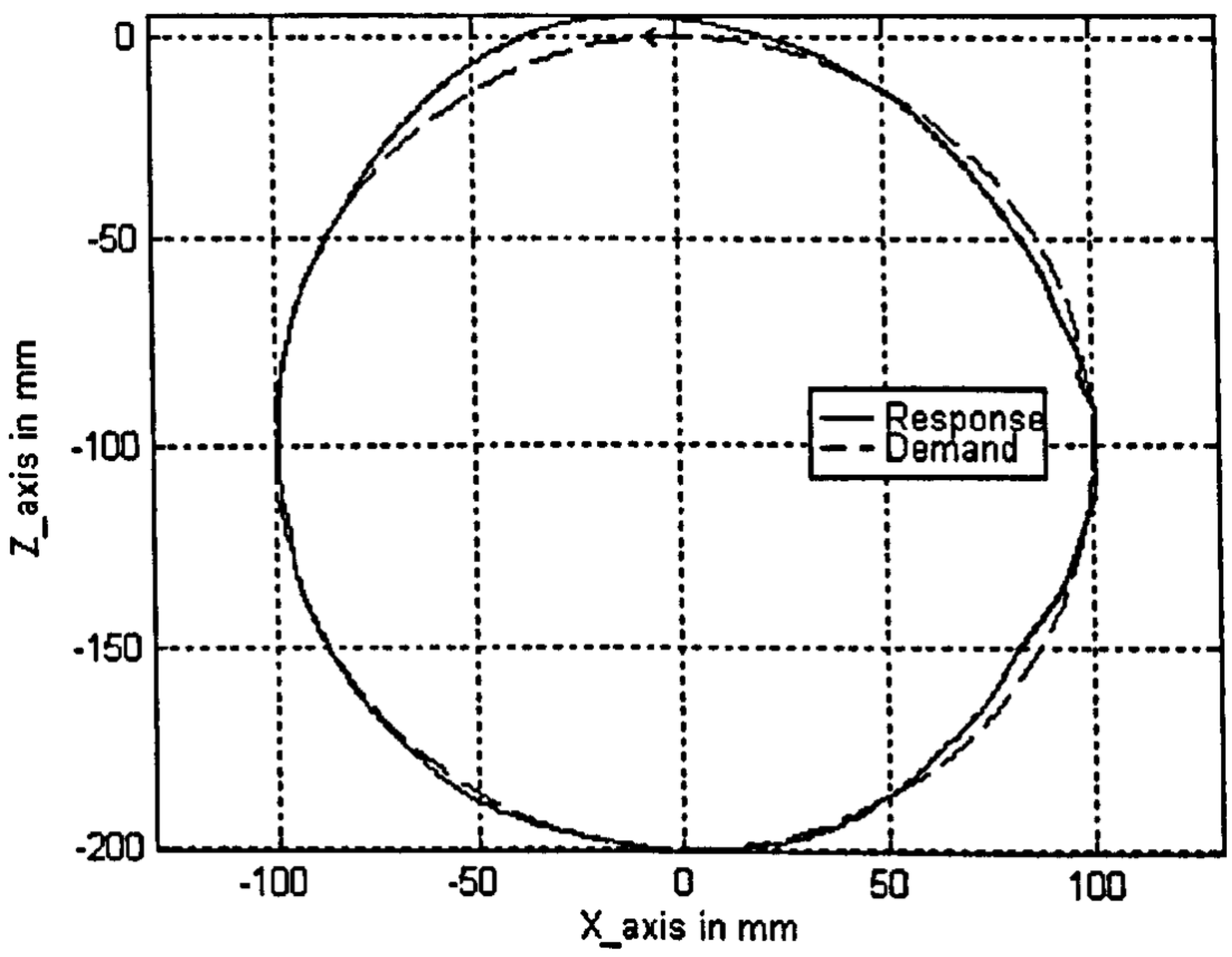


Figure 72: Front View of the End Point Displacement for the Circle Trajectory

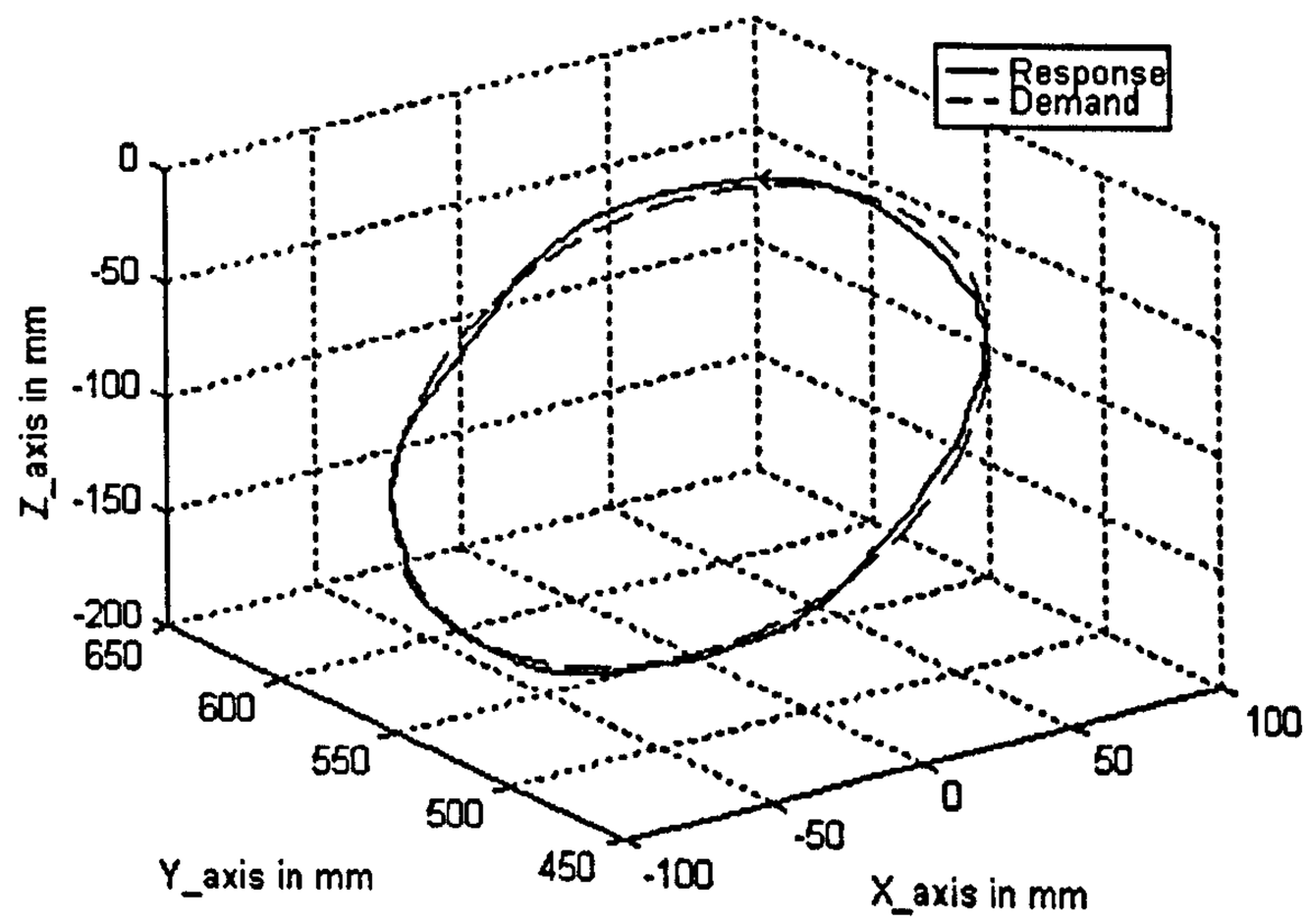


Figure 73: 3D View of the End Point Displacement for the Circle Trajectory

previous one on the same trajectory. In Cartesian co-ordinates, the maximum error represents a deviation of the end point from the demand trajectory of the circle of less than $7mm$, as can be seen from figures 72 and 73.

6.4 Appraisal of the Results

To assess the performance of the two proposed controllers, the three trajectories (straight line, goalpost test and circle) are repeated on the SPRINTA with an independent joint PID control algorithm. The control algorithm has a sampling frequency of $1milli$ sec, and all trajectories use rectilinear interpolation.

6.4.1 Straight Line Motion

The trajectory demands in motor co-ordinates are identical to the two previous straight line trajectory demand, as shown in figure 38. The system response is shown in figures 74-76.

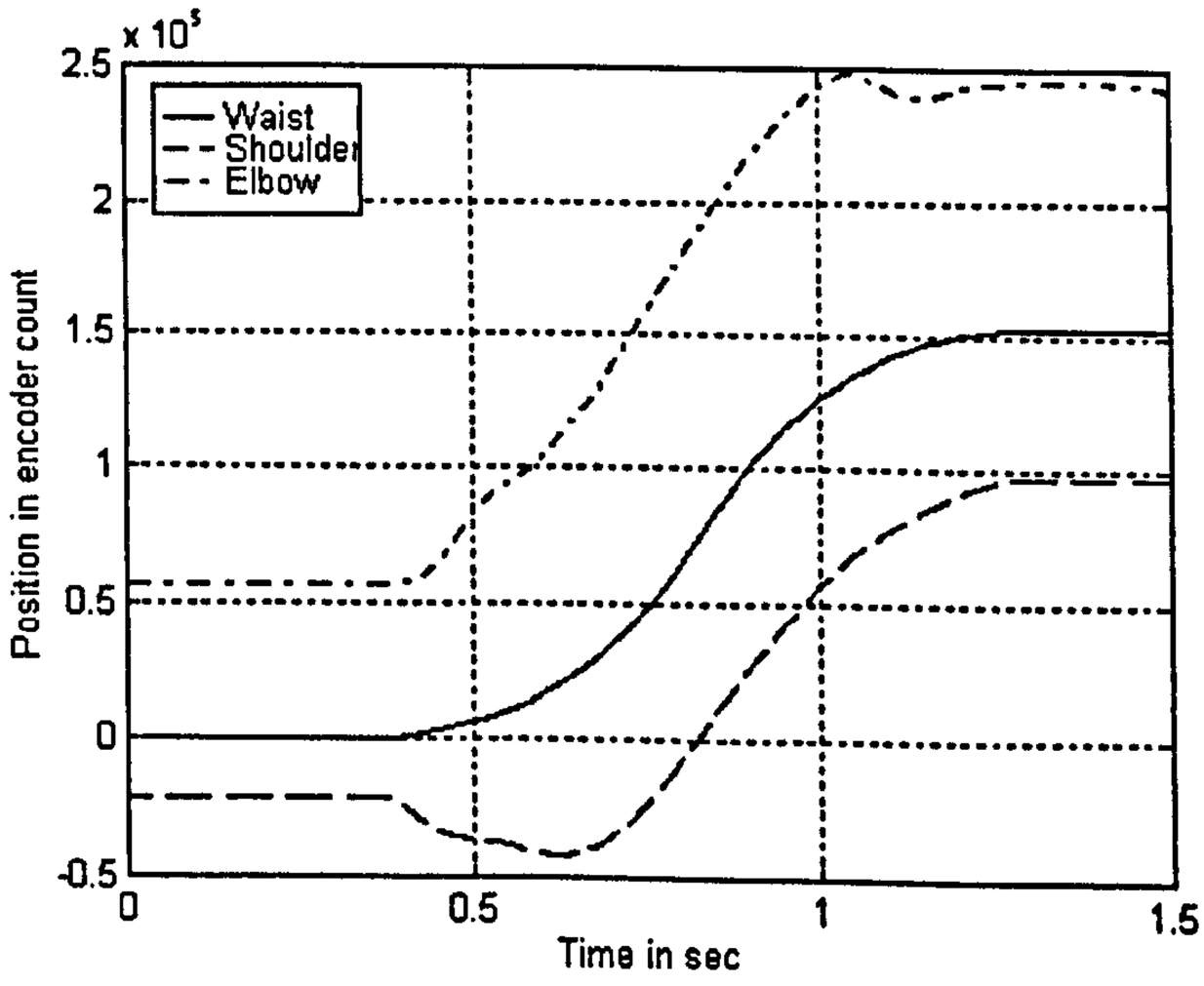


Figure 74: Response Trajectories in Motor Co-ordinates for the Straight Line Motion

The maximum position errors for the waist, shoulder and elbow joints are 969, 2661 and 10053 encoder pulses i.e. $0.0099rad$, $0.027rad$ and $0.103rad$ respectively , which represent a deviation of the end point in Cartesian co-ordinates of $39mm$ as

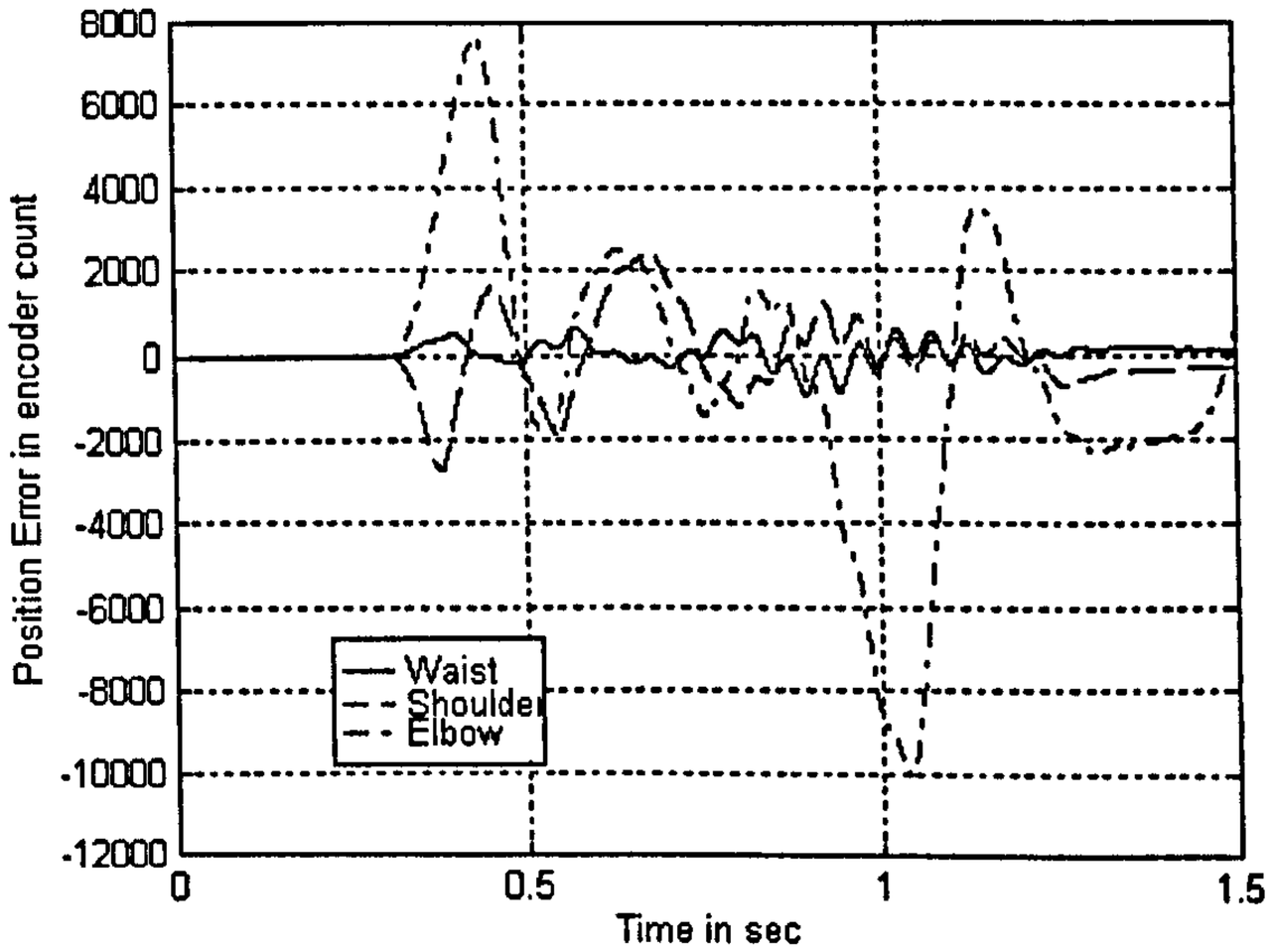


Figure 75: Position Errors in Motor Co-ordinates for the Straight Line Trajectory

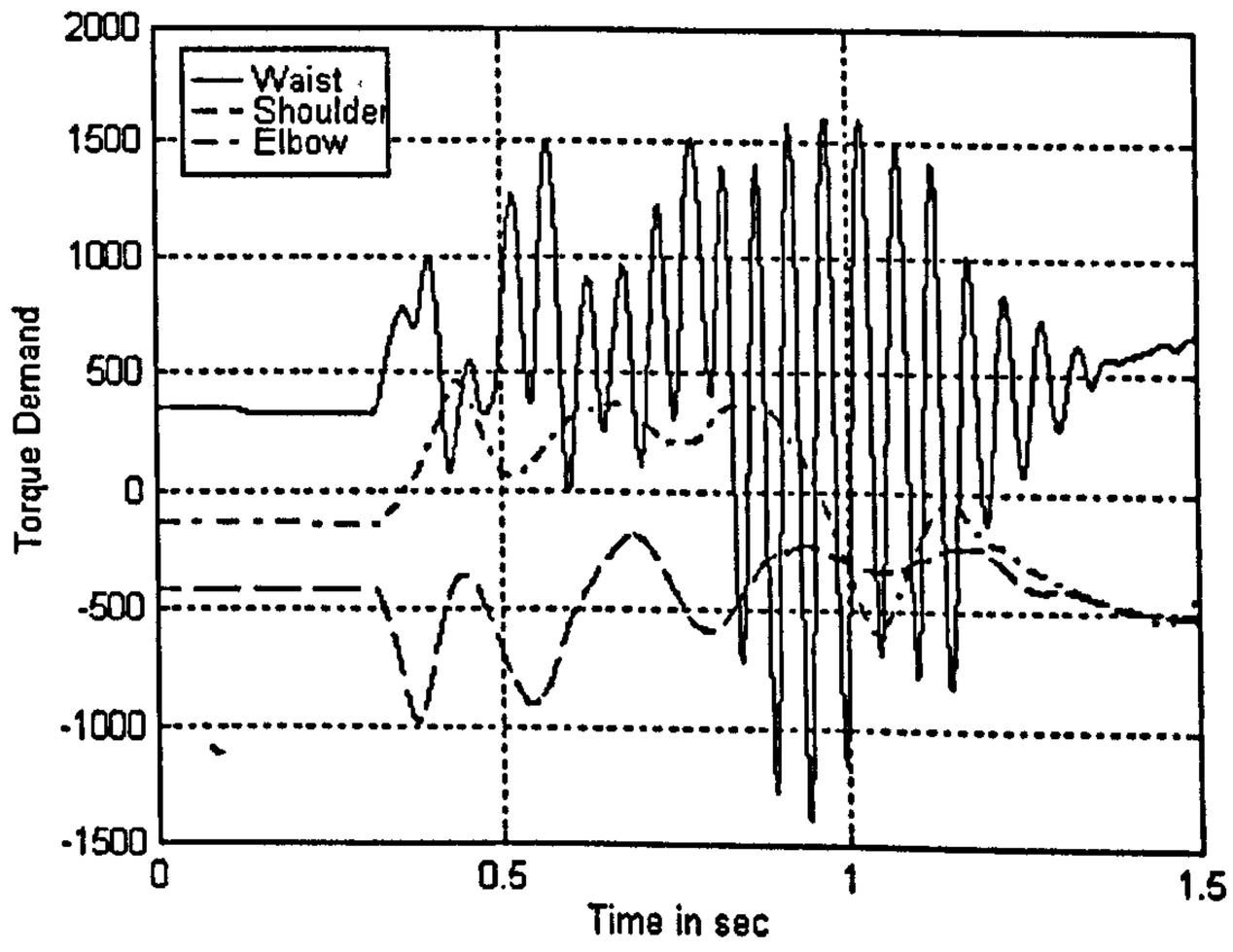


Figure 76: Torque Demands for the Straight Line Motion

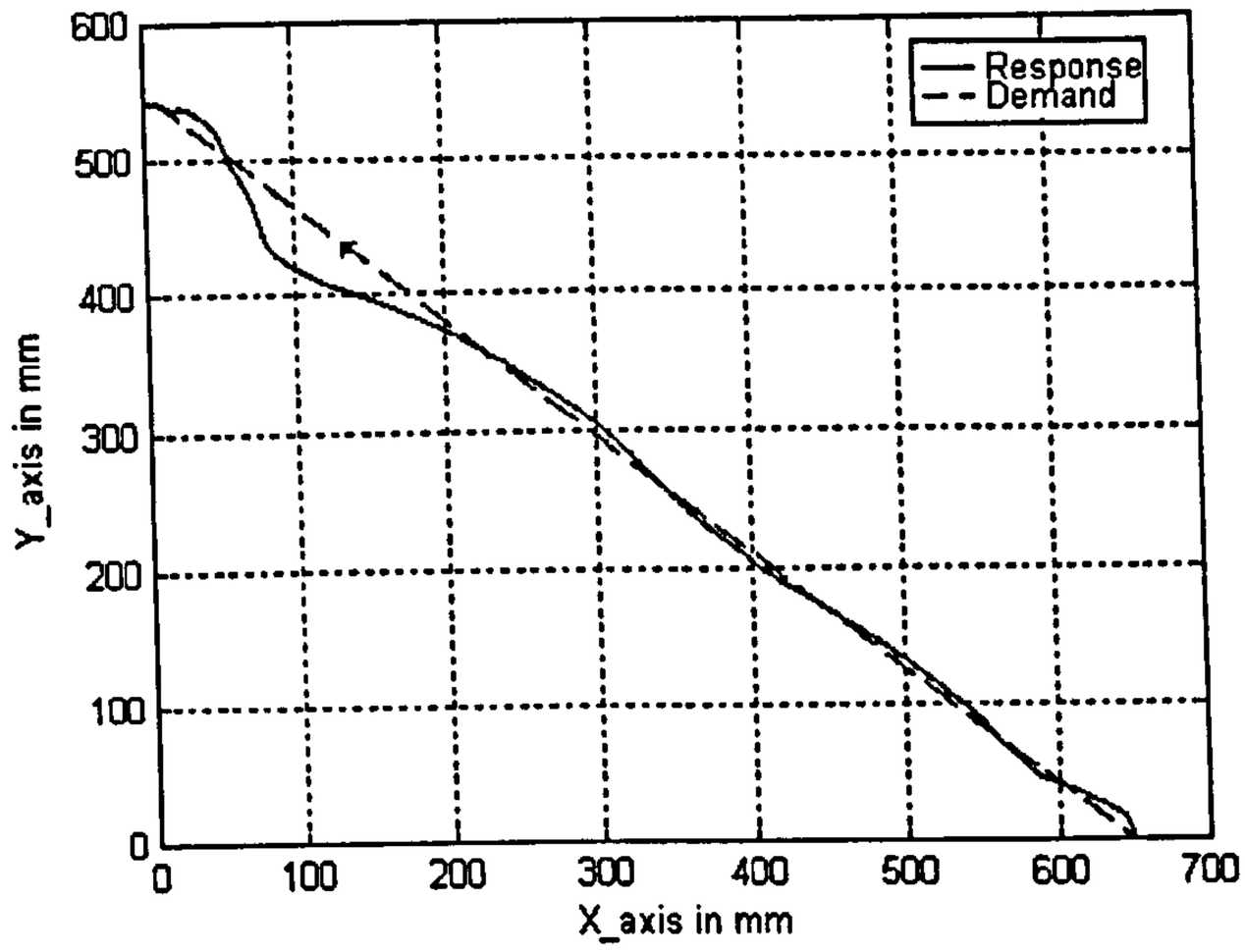


Figure 77: Top View of the End Point Displacement for the Straight Line Trajectory

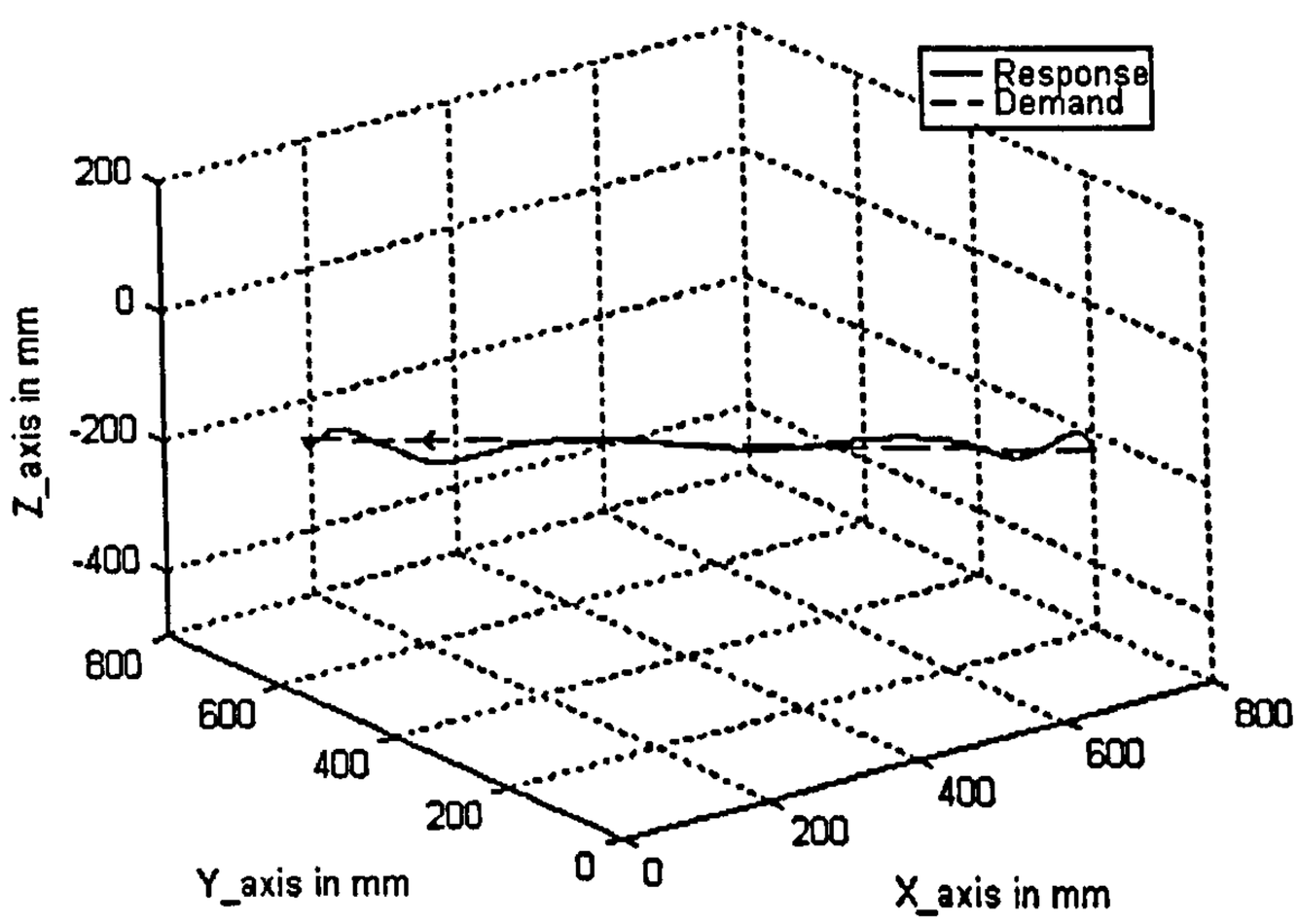


Figure 78: 3D View of the End Point Displacement for the Straight Line Trajectory

		PID	Controller 1	Controller 2
$\|E\|_{\max}$ in encoder counts	*Waist*	969	629	628
	Shoulder	2661	1936	1949
	Elbow	10053	5766	2301
$\int \|E\|$ in encoder counts.sec	*Waist*	229	228	237
	Shoulder	802	774	781
	Elbow	2708	1506	885

Table 2: Performance Comparison for the Straight Line Trajectory

can be seen from figures 77 and 78.

The performance obtained with the three controllers (model based with varying bound, model based with smooth varying bound and PID controller) are compared with each other. The maximum position error on the shaft of the motor, $|E|_{\max}$ as well as a performance index, $\int |E|$ derived from the ISO performance criteria for industrial robots [42] is used to make the comparison. The performance index is the integral of the absolute position error of the motor shaft with respect to time, and one index is used per motor. The results obtained for the three controllers on the straight line trajectory are summarised in table 2.

The table 2 clearly shows the improvement in the position errors with the two model based controllers compared to the classic independent joint PID control algorithm. A clear reduction of the maximum position error is achieved for the three motors with the model based controllers with reference to the PID controller. The improvement of the performance index $\int |E|$, is seen for the shoulder and elbow motors. The performance improvements are translated to lower end point deviation from the trajectory demands, as illustrated in figures 42, 62 and 77.

6.4.2 Goalpost Test

The trajectory demands in motor co-ordinates are identical to the two previous goalpost test demands, as shown in figure 45. The system response is shown in figures 79-81.

The maximum position errors for the waist, shoulder and elbow joints are 1942, 6826 and 10209 encoder pulses i.e. $0.02rad$, $0.07rad$ and $0.104rad$ respectively, which

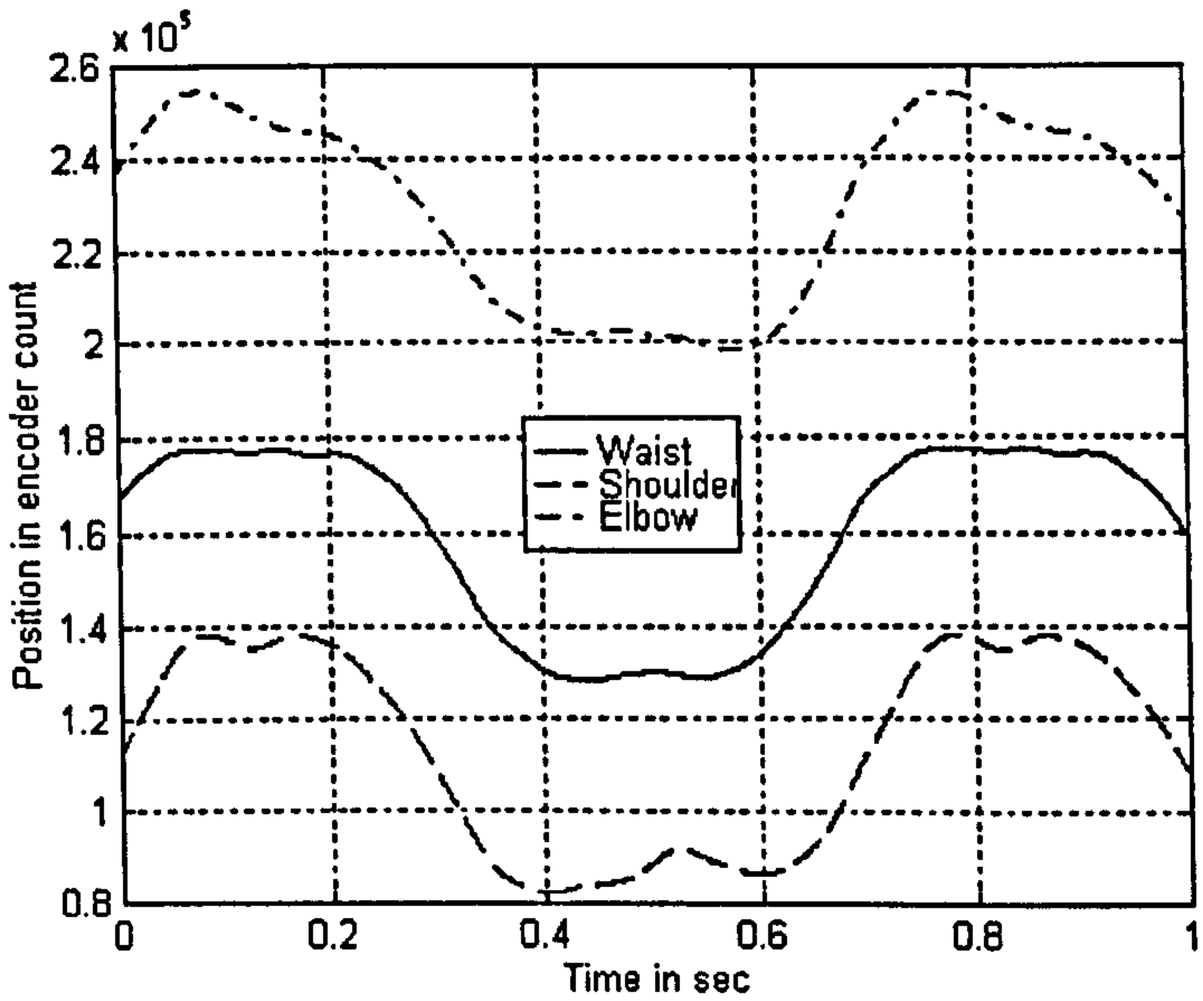


Figure 79: Response Trajectories in Motor Co-ordinates for the Goalpost Test

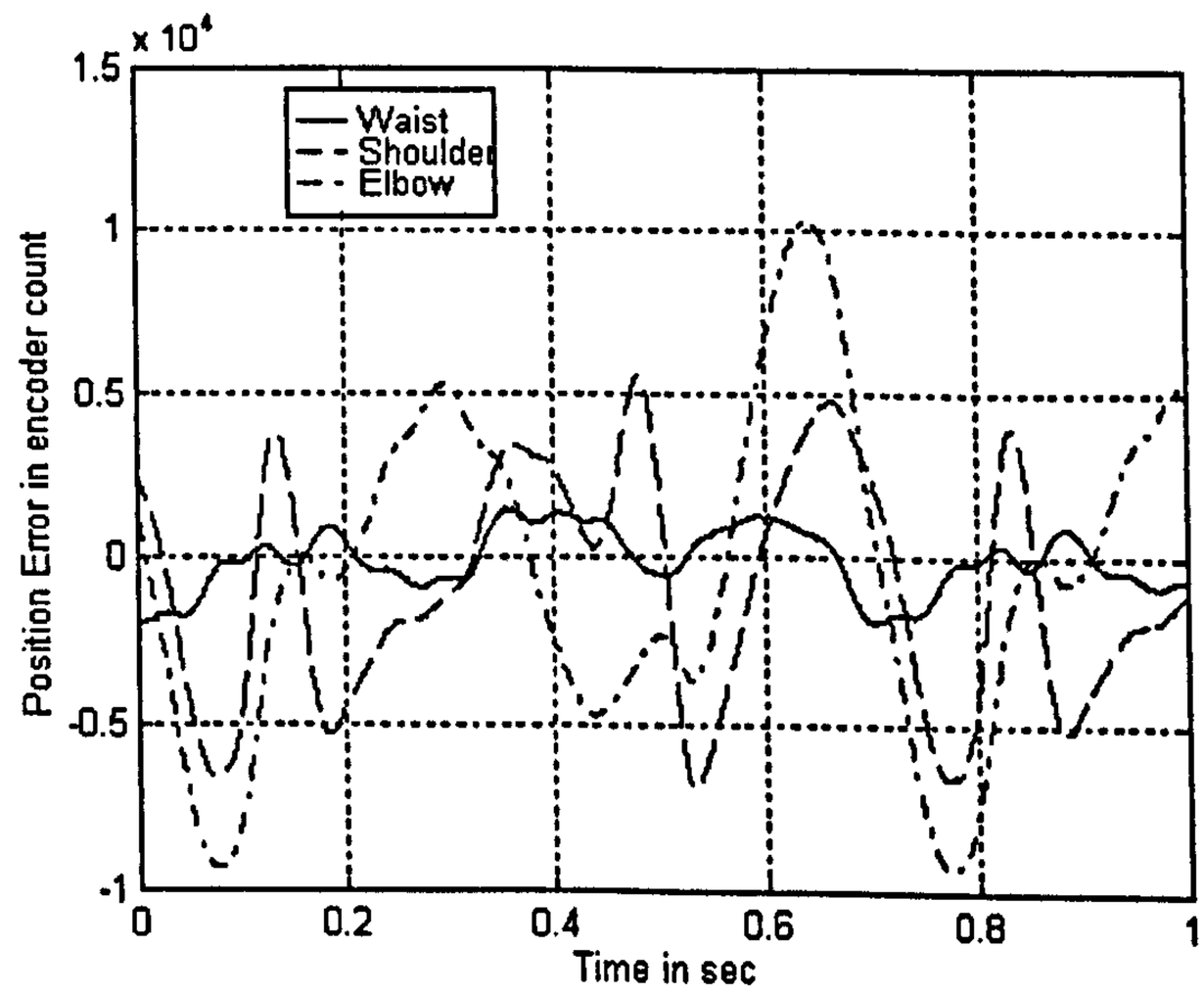


Figure 80: Position Errors in Motor Co-ordinates for the Goalpost Test

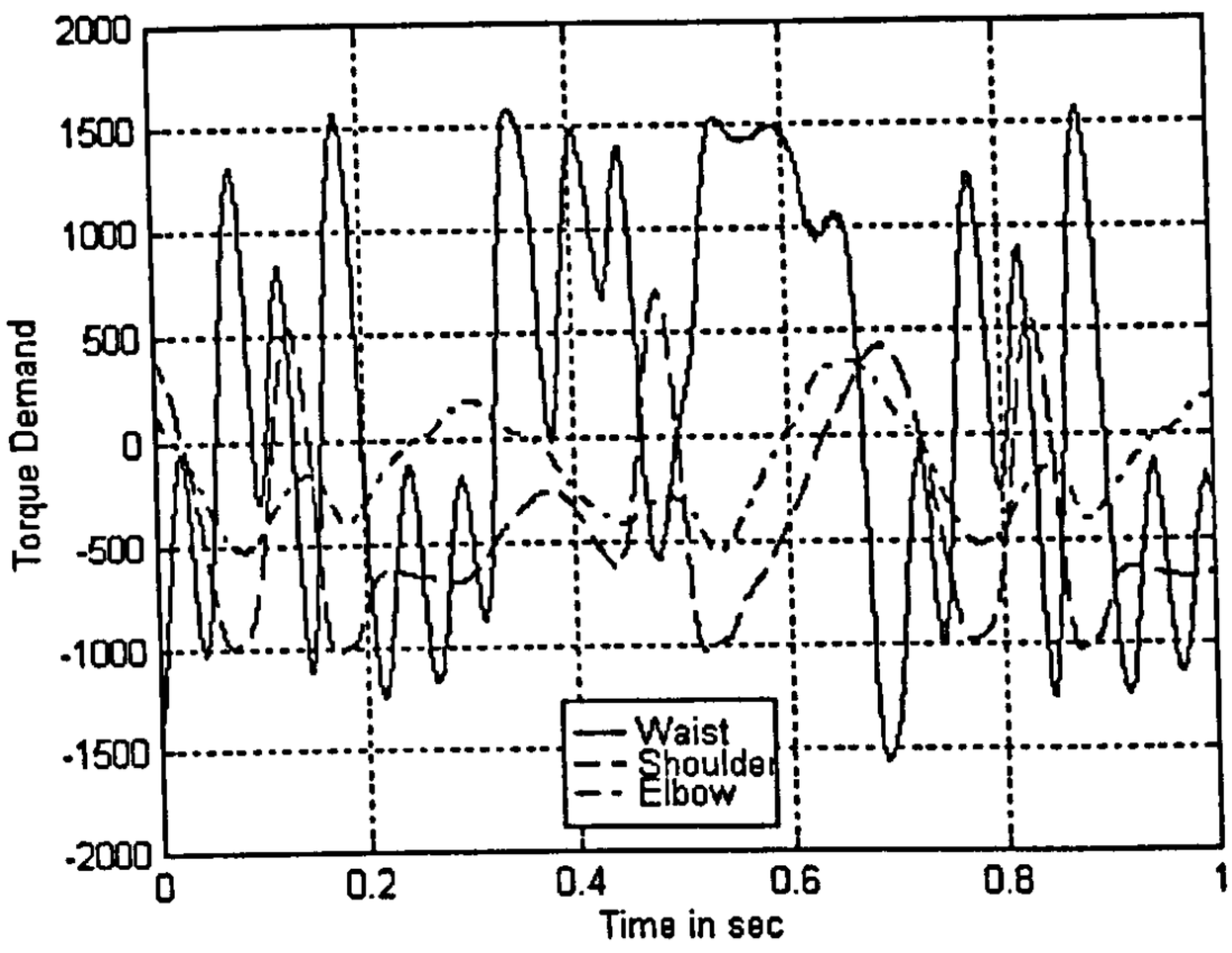


Figure 81: Torque Demands for the Goalpost Test

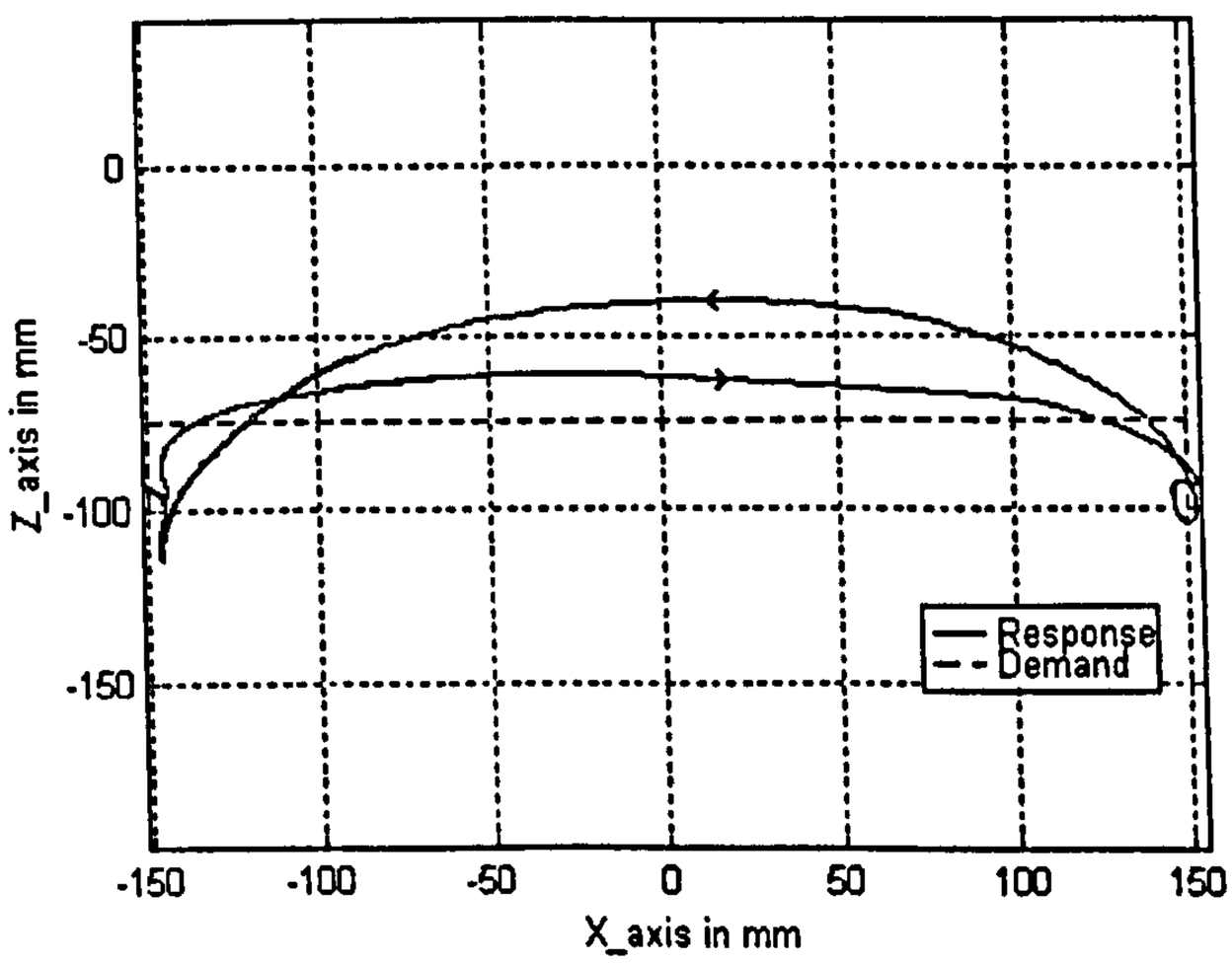


Figure 82: Front View of the End Point Displacement for the Goalpost Test

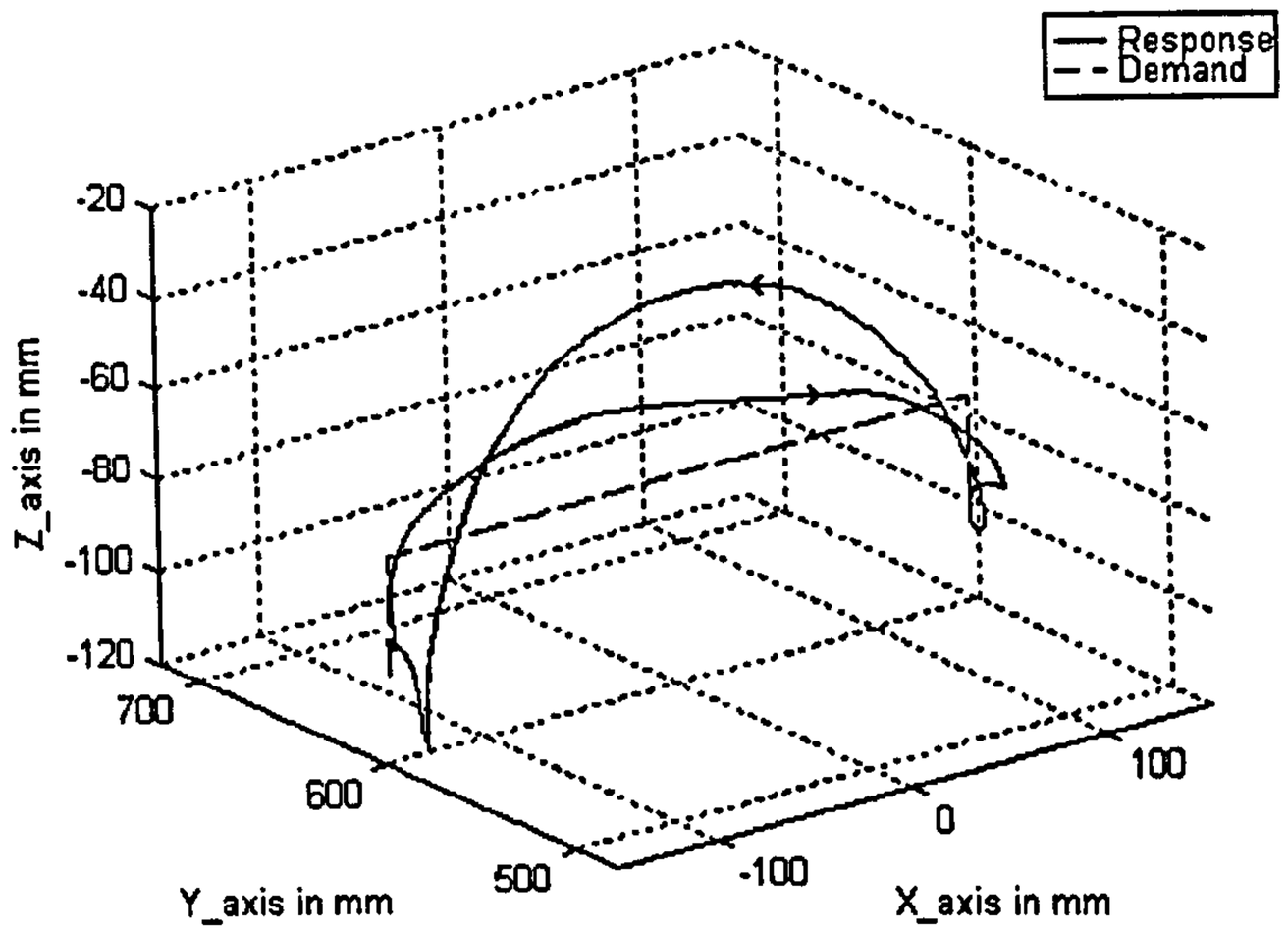


Figure 83: 3D View of the End Point Displacement for the Goalpost Test

		PID	Controller 1	Controller 2
$\|E\|_{\max}$ in encoder counts	*Waist*	1942	1135	1125
	Shoulder	6826	9339	8381
	Elbow	10209	6020	2122
$\int \|E\|$ in encoder counts.sec	*Waist*	550	352	333
	Shoulder	2110	1735	1491
	Elbow	2784	1714	693

Table 3: Performance Comparison for the Goalpost Test

represent a deviation of the end point in Cartesian co-ordinates of $35mm$ from the horizontal trajectory demands as can be seen from figures 82 and 83.

As for the straight line trajectory, the performance obtained for the goalpost test for the three controllers are summarised in table 3. The performance index $\int |E|$, and the maximum position error of the shaft of the motors $|E|_{\max}$ are again used. Note that the integral of the position errors is performed over one full cycle of the goalpost test, i.e. forward and return movement.

From table 3, the performance of the two model based controllers are superior to that of the independent joint PID controller. The benefit of the two model based

controllers is apparent on the performance index $\int |E|$. In terms of maximum position error $|E|_{\max}$, net improvements are obtained for the waist and elbow motor. The differences in performance between Controller 1 (model based controller with varying bounds) and Controller 2 (model based controller with smooth varying bounds) are mainly noticeable on the elbow motor. The overall improvement of the error results in a end point displacement closer to the demand position, figures 49, 67 and 82.

Remark: From figures 47, 65 and 80 showing the position error, it appears that the errors for each of the three motors oscillate. This is due to the cyclic nature of the trajectory, the endpoint has to move twice 'up and down' in the vertical plane for one complete cycle of the goalpost test. Further, the data presented covers 1.2 sec and the cycle time is 0.7 sec, from one cycle to the next it can be observed that the position errors are similar. This similarity highlights the high level of repeatability of the SPRINTA robot which is demonstrated by G. Vines [100].

6.4.3 Circle Trajectory in the Vertical Plane

The trajectory demands in motor co-ordinates are identical to the two previous circle trajectory demands, as shown in figure 51. The system response is shown in figures 84-86.

The maximum position errors for the waist, shoulder and elbow joints are 1392, 3792 and 3996 encoder pulses i.e. $0.014rad$, $0.039rad$ and $0.041rad$ respectively, which represent a deviation of the end point in Cartesian co-ordinates of $9mm$ from the trajectory demands as can be seen from figures 87 and 88.

The performance obtained by the three controllers (PID, model based with varying bounds, and model based with smooth varying bounds) are summarised in table 4. The performance index $\int |E|$, and the maximum position error of the shaft of the motors $|E|_{\max}$ are again the criteria used to assess the performances. Note that the integral of the position errors is performed over one full circle movement, which represents a duration of 0.8 sec.

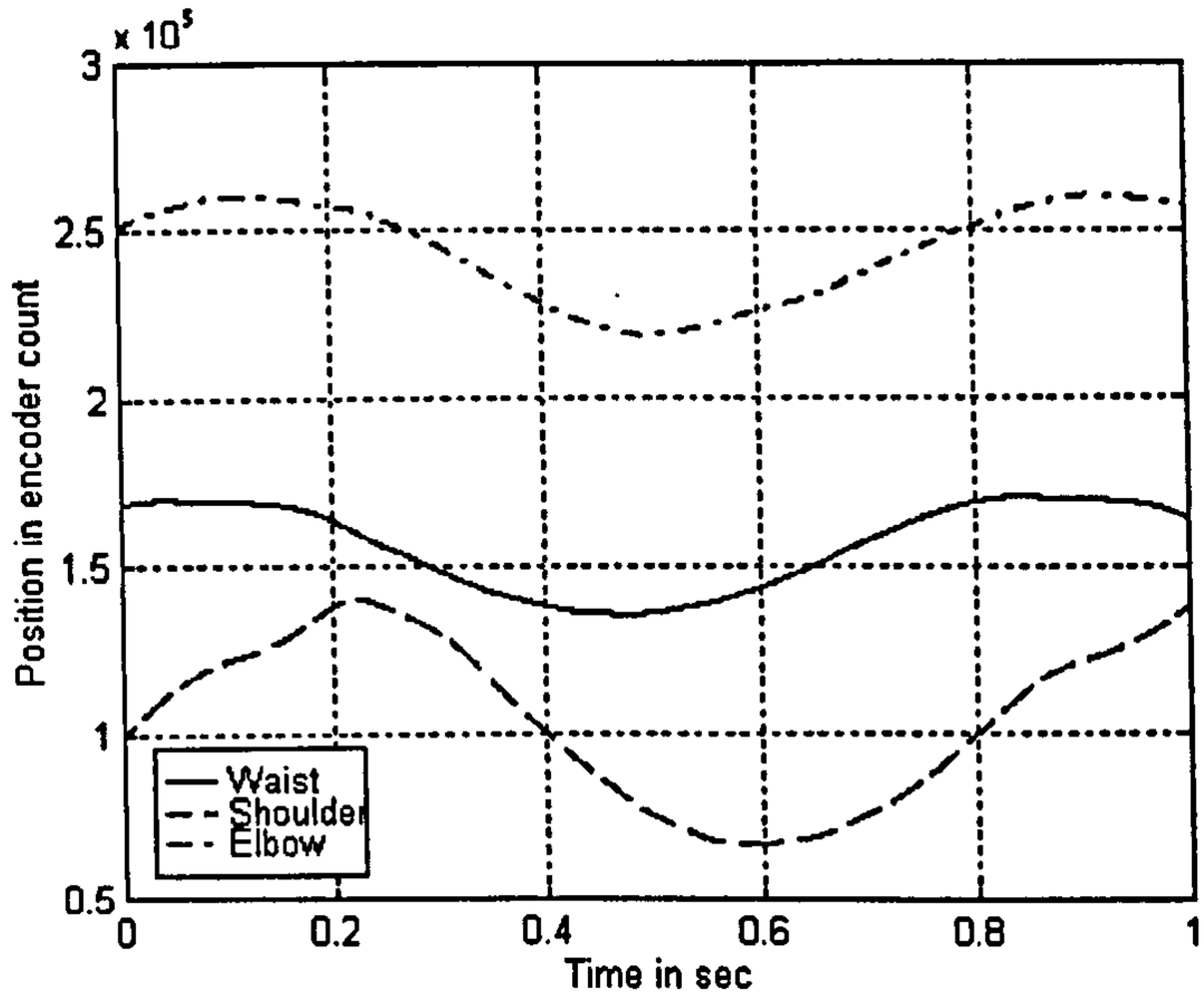


Figure 84: Response Trajectories in Motor Co-ordinates for the Circle Motion

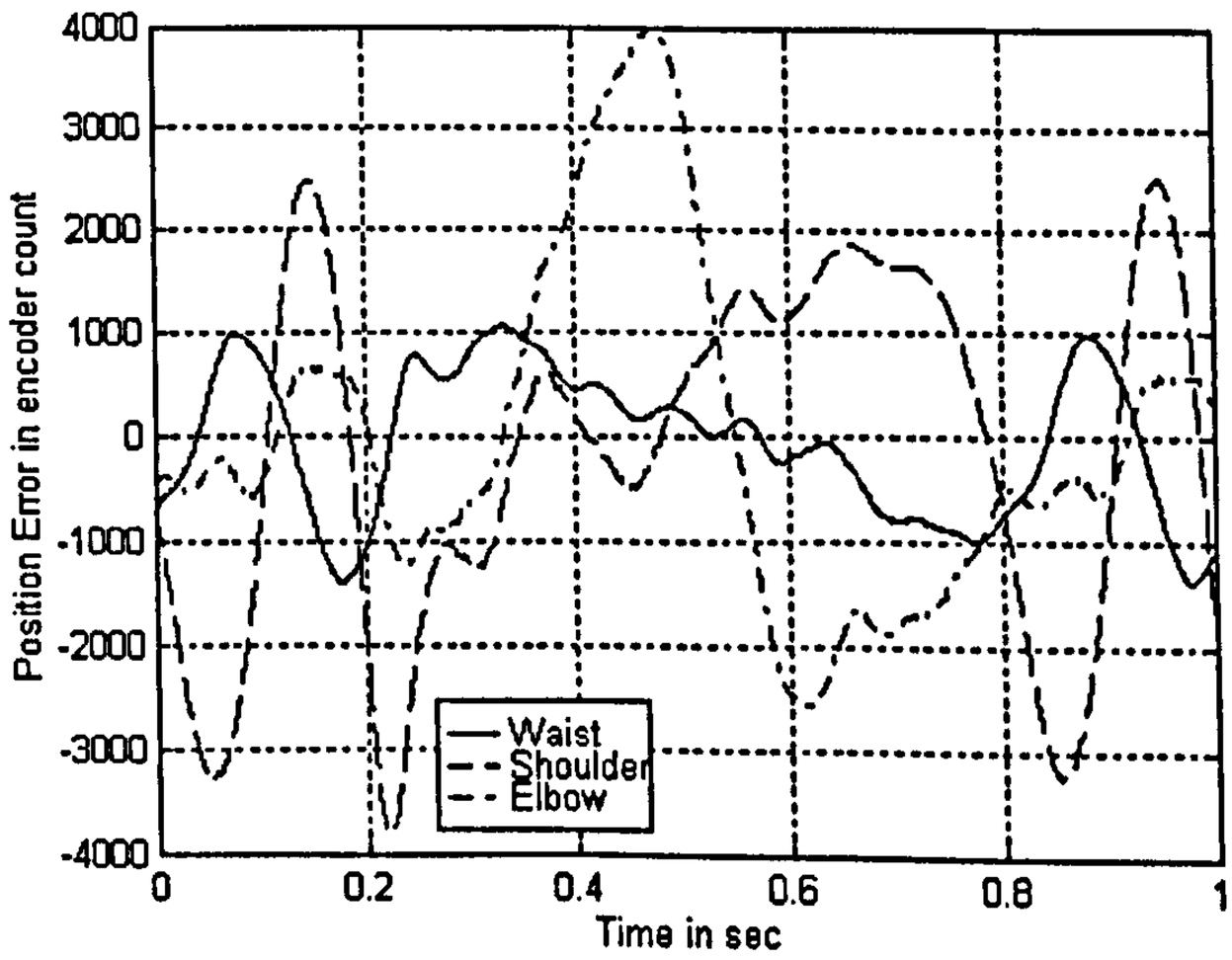


Figure 85: Position Errors in Motor Co-ordinates for the Circle Trajectory

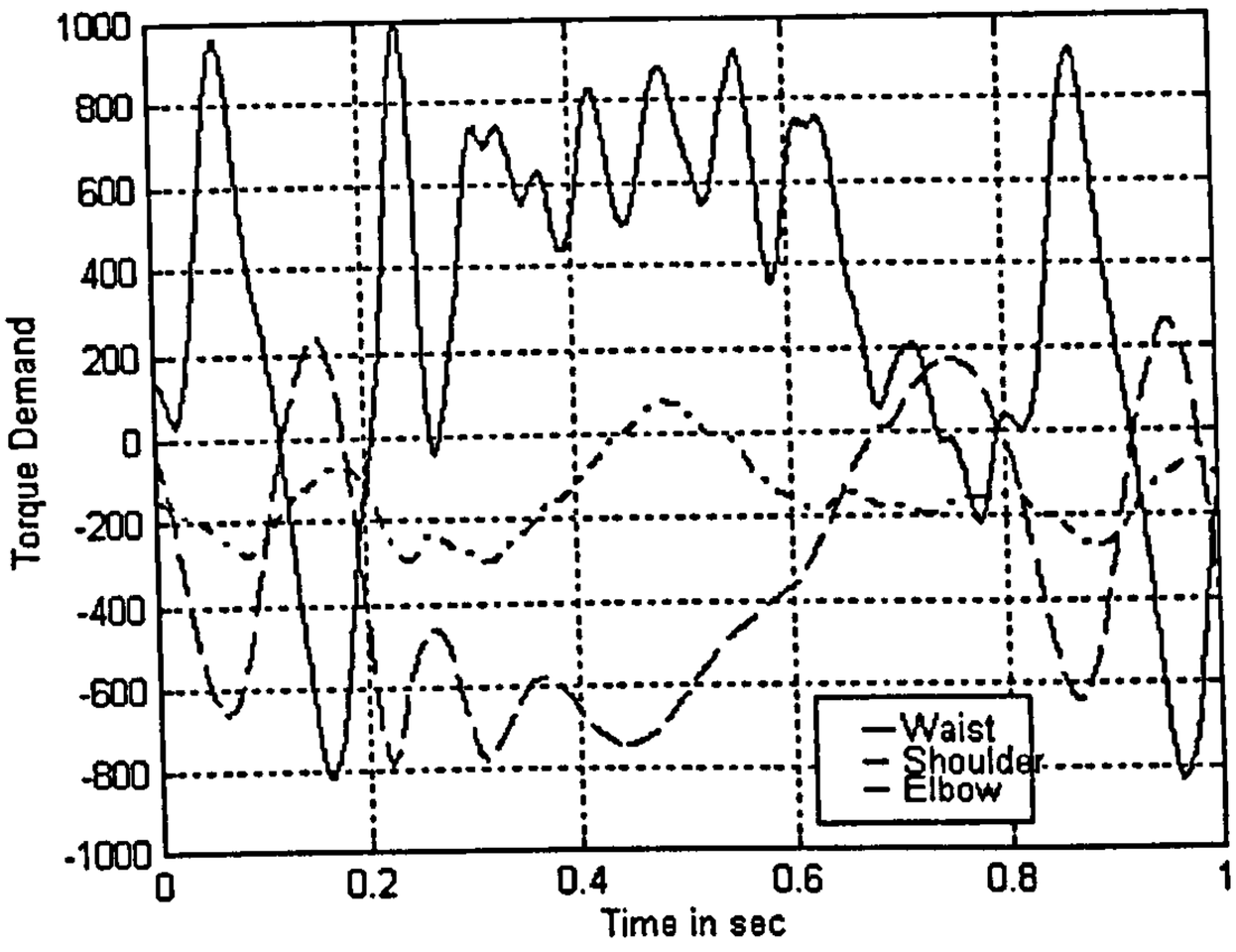


Figure 86: Torque Demands for the Circle Motion

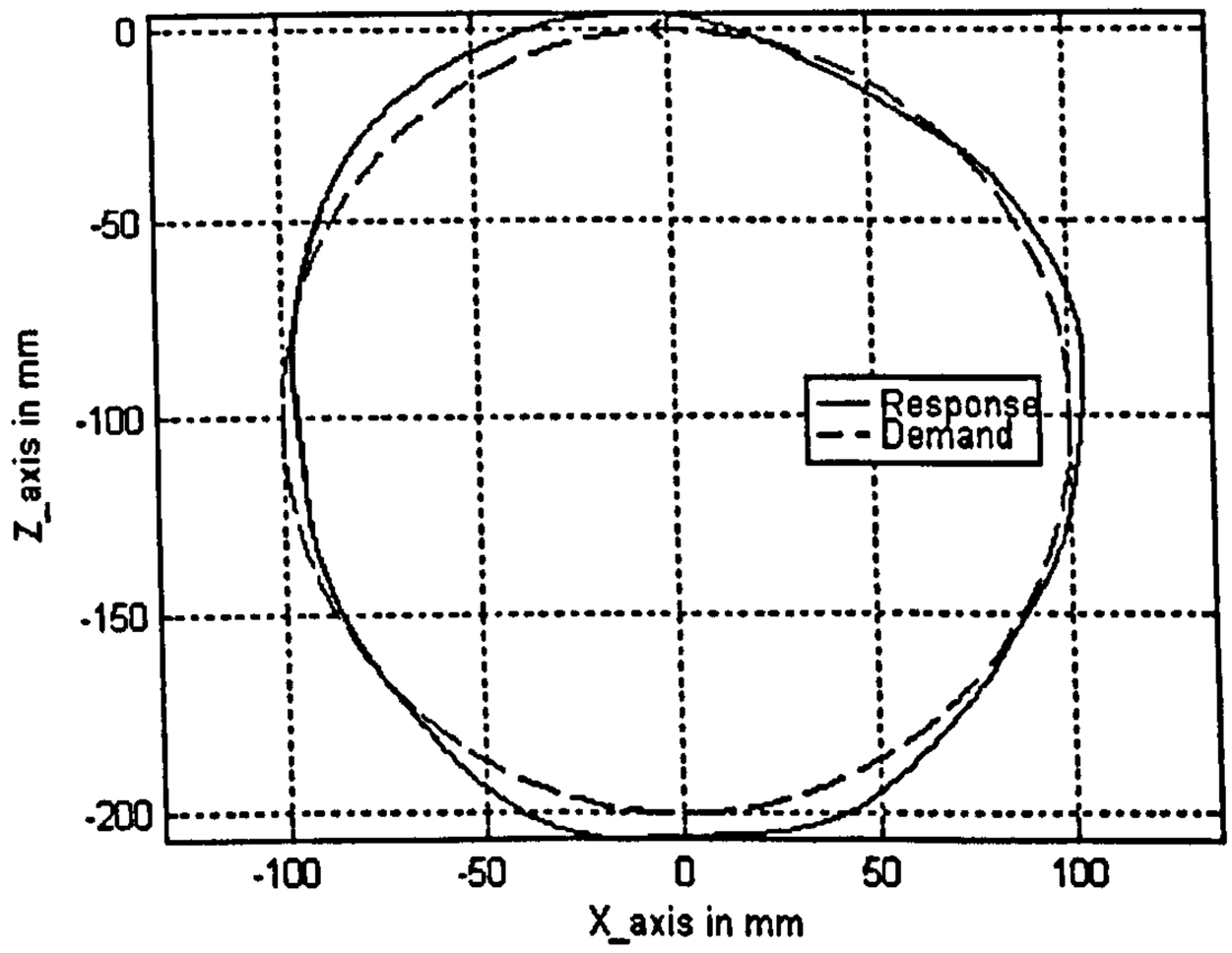


Figure 87: Front View of the End Point Displacement for the Circle Trajectory

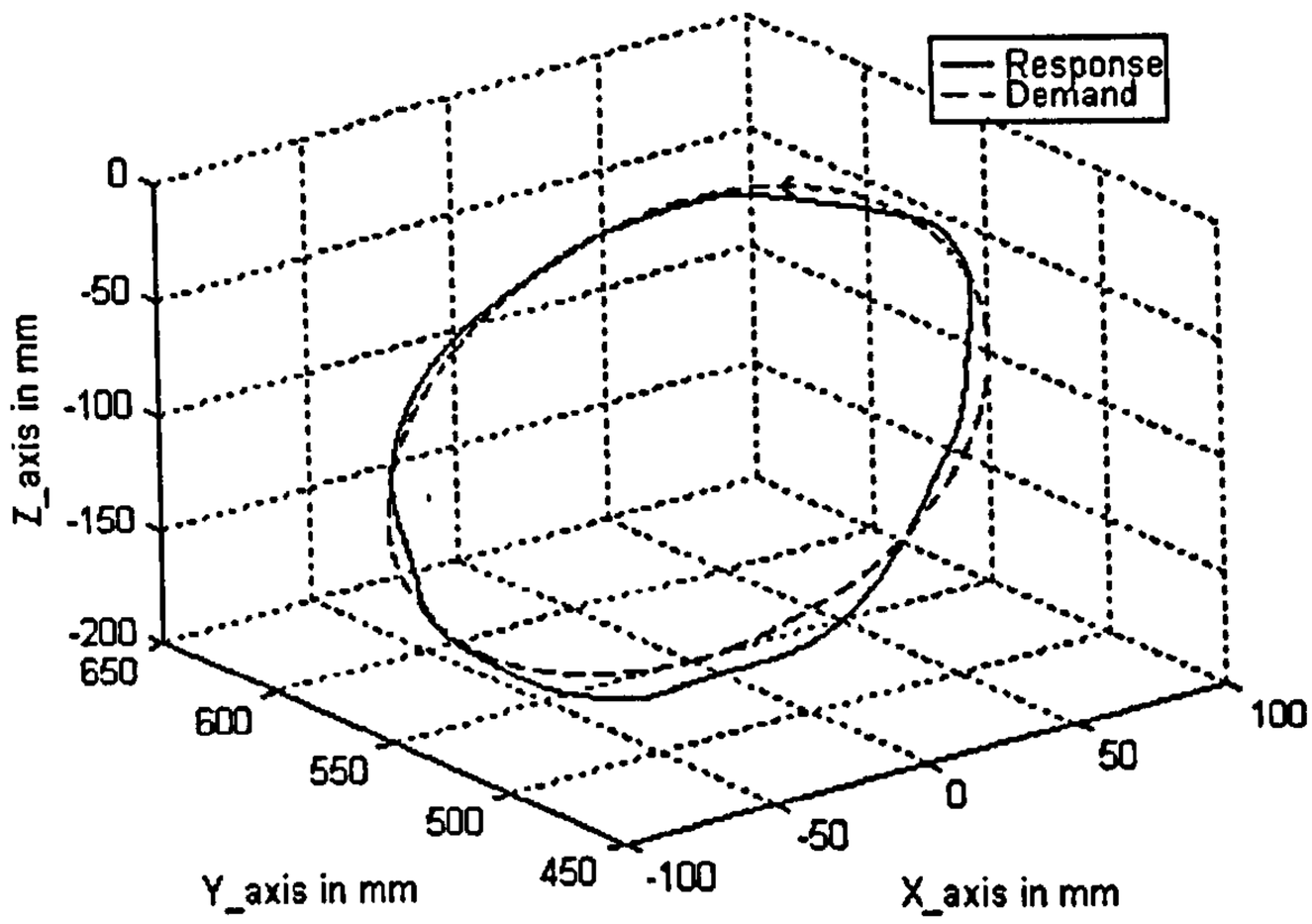


Figure 88: 3D View of the End Point Displacement for the Circle Trajectory

		PID	Controller 1	Controller 2
$\|E\|_{\max}$ in encoder counts	*Waist*	1392	668	545
	Shoulder	3792	3998	3567
	Elbow	3996	3369	1675
$\int \|E\|$ in encoder counts.sec	*Waist*	446	228	192
	Shoulder	1047	1099	1024
	Elbow	1108	775	558

Table 4: Performance Comparison for the Circle Trajectory

According to table 4, the performance of the two model based controllers are similar to those of the independent joint control PID for the circle trajectory. Performance improvement can however be observed on the waist and elbow joint for $\int |E|$ and $|E|_{\max}$. This improvement mainly concerns Controller 2 compared to the performance of the PID controller. In Cartesian co-ordinates, the improvement of the position error of the shaft of the motors results in a more accurate displacement of the end point of the SPRINTA robot, as can be seen from figures 72 and 87.

6.4.4 Torque Demand Assessment

As mentioned in section 2.4, each motor drive is configurated so that it responds to the command data with a torque proportional to the maximum torque available. This maximum torque is not constant across the whole range of operating speeds. At the nominal point of operation the maximum torque available for the waist and shoulder motors is $4Kgf.m$ or $39N.m$, and for the elbow motor the maximum torque is $1Kgf.m$ or $10N.m$. For the figures showing the torque demands, the maximum torque in all cases is given by 2047. So for the waist and shoulder motors ± 2047 is equivalent to $\pm 39N.m$ and for the elbow motor it is equivalent to $\pm 10N.m$.

The torque representations for the three trajectories for Controller 1, figures 41, 48 and 54, and for Controller 2, figures 61, 66 and 71 is made of higher frequencies than the torque demands for the PID controller, figures 76 81 and 86. This is explained by the discontinuous nature of the controllers. Even though the VSC algorithm has been designed so that the chattering phenomenum which is the cause of the high frequencies contained in the torque, is kept to a minimum via bound variation. The bandwidth of the torque resulting from the VSC algorithm is larger than the one for the PID controller. This larger bandwidth of the torque demands do not however cause any problems of operation for the SPRINTA, as the frequencies are below any resonance frequencies of the SPRINTA and there is no resulting vibration.

7 CONCLUSION

The development of control algorithms for robot manipulators has reached a mature stage, where theoretical bases for the devised algorithms are well established. Among these algorithms are the model-based, which use the non-linear dynamic equations of the manipulator as a feedforward term for the controller. Such controllers are claimed to perform better than those which do not use the kinetic description of the robot in the controller design.

This thesis shows how the dynamic equations of the robot manipulator can be used to design a robust controller and how the robot properties can be used to improve the robust part of the controller.

7.1 Summary of Findings

Robot manipulators are highly non-linear systems, traditionally controlled by assuming that the joints are independent of each other and interactions between them are considered as disturbances. This assumption is valid when there is a large reduction ratio between the actuator and the joint, and when the required performance is not too demanding; in this case PID-like controllers are generally sufficient. The requirements for robot manipulators nowadays however can no longer be satisfied with such an approach. This has lead to the development of new types of controllers that take into account the joint couplings. In the model based methodology [3], the dynamic equations of the robot are integrated into the controller as a feedforward term. The implementation of such equations results in an overall linearized system. The linearization is efficient as long as the model used perfectly matches the robot system [51]. In practical applications, such an ideal situation is not obtained, since the adopted model has some parameter uncertainties as well as unmodeled structures. To deal with that, the controller must be made adaptive or robust so that it copes with the uncertainties and/or mismatches.

Variable structure control with sliding mode is a robust control methodology that can deal with large disturbances [99]. It has gained popularity in recent years, because of its versatile nature and inherent robustness properties. The first reported

application of variable structure control with sliding mode to robotics is due to Young [104]. In the initial control scheme used by Young [104], a hierarchical methodology [99] is adopted for the controller design. This slows down the response of the system and does not take advantage of the robot properties. Slotine [86] proposes the use of a computed torque feedforward term to linearize the overall system and compensate the uncertainties of the feedfoward term with a VSC element. The design procedure is however cumbersome and the resulting control laws are highly conservative, which leads to significant chattering. Bailey *et al.* [12] propose to design a controller by using a Lyapunov function, so that the controller design is simpler and systematic. The chattering problem still occurs but in a reduced form. Bailey *et al.* tackle this by using a saturation function instead of the pure discontinuous function. Many other authors have proposed controller designs that use a model based structure with a VSC regulator and adaptive versions have also been proposed. The resulting controller structures are usually very similar, and all address the chattering problem after the controller design has been completed.

7.2 The Devised Controller and its Implementation

The most significant limitation with robot control algorithms employing VSC with sliding mode is the chattering phenomenon, which can damage the overall electro-mechanical structure, as well as reduce the motion performance. The general approach to overcome this is to replace the switching function by a smoother one after the original design has been completed. The design of the proposed controller starts in a similar way as the design methodology proposed by Bailey [12]. A pseudo-energy Lyapunov function is selected and a general model-based structure for the controller is assumed. A number of assumptions are made on the resulting uncertainties and mismatches between the controller and the actual robot model. Each mismatch is analysed individually and the bound for each is established. It is found that the control effort from the VSC element and the chattering can be greatly reduced by indexing the bounds with the position and velocity errors. The bounding results from the interpretation of the physical nature of the uncertainties and mismatches.

Simulations confirm that the chattering can be eradicated by bounds indexing,

however it was observed that a position offset occurs and an integral term has to be added to the control law. The algorithm is successfully implemented in this form on a SCARA robot [66].

The control algorithm implemented on the SPRINTA has the same structure as the proposed controller. A kinetic model is used as a feedforward term and a VSC element with varying bounds is used to overcome the uncertainties and mismatches. The main feature of the SPRINTA is the non-linear transmission which greatly complicates the dynamic equations and makes the use of these equations as a feedforward term ineffectual. The equations have to be simplified so that they can be calculated within a reasonable sampling interval. A high level of inaccuracies and mismatches result from these simplifications, so further emphasis is putted on the VSC regulator. The phase plane of the tracking error has to be decomposed into zones, and similarly the workspace of the SPRINTA is divided into regions in order to give a further degree of adjustment for the bounds. Two variants of the controllers have been successfully implemented and tested in the SPRINTA. In the first one, the bounds are adjusted with respect to the tracking error and the workspace position, in a discontinous manner, whilst the adjustment is made in a smooth way for the other one. The performance obtained with the two controllers for three different trajectories are superior to those of an independent joint PID controller.

7.3 Concluding Remarks

The design approach has demonstrated how the system properties and physics can be used to modify the controller for the system under control. Further, this specific fitting of the controller to the control problem has lead to the indexing of the bound of the VSC regulator with the position and velocity errors, which greatly helps to reduce the chattering. Allowing the variation of the bounds for VSC brings new degrees of freedoom to the design as well as a new perspective. Chattering is reduced and so is the stress put on the system under control. Systems which may not have been controlled with VSC techniques because of the chattering effect, can now be controlled with VSC which has varying bounds.

7.4 Further Work

The proposed controller has been designed for the purpose of controlling a robot manipulator, however the methodology employed as well as the general form of the controller can be applied to a large class of mechanical systems, for which a dynamic model can be derived and used as feedforward element. The inverted pendulum or the ball and beam for example can be a next step for the implementation and experimentation for the controller. The application of the controller to these two systems will constitute excellent test cases to investigate further applications to real systems.

In this work, the parameters and the order of variation of the bounds are tuned experimentally, so investigations on the tuning of these terms could be conducted to derive a systematic approach to selecting suitable values. Similarly, during the experimental implementation extra degrees of freedom in tuning the bounds are introduced, i.e. Δ, Γ and Ψ. These extra degrees of freedom are derived and tuned by trial and error methods. Research can be conducted to assess the effect of each of the extra parameters (Δ, Γ and Ψ) on the control law and on the type of trajectories. It is observed that some values provide excellent response for the straight line test but not for the circle test or vice versa. In the work presented the tuning is done to give acceptable responses for all types of trajectory.

The proposed controller is tested on a SCARA robot with no modification to the control law and with little effort. On the contrary, the implementation on the SPRINTA requires the addition of extra degrees of adjustment and a fair amount of time before it works satisfactorily. The investigation of the impact of the non-linear transmission on the control strategy can be another area of research.

The proposed controller is assessed against an independent joint PID controller, and comparisons with other types of controller could be conducted to have a better view of the performance of the proposed controller.

In the proposed scheme, the VSC only acts as a regulator. There are many control algorithms where VSC is the dominant element of the control strategy, and the introduction of bound variation gives extra degrees of freedom to the design and potentially a better response as it can reduce the control effort. The extension

of varying bounds idea to other types of VSC and their assessment is a potential development route for the work presented.

Upgrading the controller board of the SPRINTA would enable it to run more complex control algorithms, since the actual controller does not make full use of the potential of the robot. The inverse dynamic model used to generate the feedforward term can be improved by incorporating more of the robot dynamics. Also other types of controller could be tested on the SPRINTA such as fuzzy logic or adaptive controllers.

The aim of this research was to design a controller for the purpose of fast trajectory tracking. More and more robots are required to execute operations where they have to interact with the environment. The control of the amount of force at the endpoint effector is then very important. The research and development of algorithms that can sense the amount of force at the end effector and feedback the information to the controller so as to regulate it is a promising area, and will open the door to new applications for robots (tele-operated robot, co-operation between robots, rehabilitation robots, ...).

At present, only general purpose trajectories are implemented on the SPRINTA, and research to determine the type of trajectories and applications for which the robot demonstrates greatest potential could be conducted so as to identify possible markets for the robot.

8 REFERENCES

[1] C. Abdallah, D. Dawson, P. Dorato and M. Jamshidi, 'Survey of Robust Control for Rigid Robots', IEEE Control System Magazine, Feb 1991.

[2] Adept Technology Inc., 1212 Bordeaux Drive, Sunnyvale, CA 94089, 'Adept-One Manipulator System: Product Description and Specifications', Aug. 1985.

[3] C.H. An, C.G. Atkeson and J.M. Hollerbach, 'Model-Based control of a robot manipulator', The MIT Press, 1988.

[4] C.H. An, C.G. Atkeson J.D. Griffiths and J.M. Hollerbach 'Experimental Evaluation of Feedforward and Computed Torque Control' , IEEE Trans. Rob. and Auto., Vol.5, No.3, Mars 1989.

[5] H. Asada and T. Kanade, 'Design of a Direct-Drive Mechanical Arm', Tech. Report CMU-RI-TR-81-1, Robotics Institute, Carnegie Mellon Univ., 1981.

[6] H. Asada, T. Kanade and I. Takeyama, 'Control of a Direct-Drive Arm', Transactions of the ASME Journal of dynamic Systems, Measurement, and Control, Vol.105, September 1983.

[7] H. Asada and J.J.E. Slotine, 'Robot Analysis and Control', Wiley-Interscience, 1986.

[8] H. Asada, K. Youcef-Toumi, 'Analysis and Design of a Direct-Drive Arm With a Five-Bar-Link Parallel Drive Mechanism', Trans. of ASME, Jour. Dyn. Syst. Meas. and Cont., Vol. 106, pp. 225-230, Sept. 1984.

[9] H. Asada, K. Youcef-Toumi, 'Direct-Drive Robots: Theory and Practice', The MIT Press, 1987.

[10] K.J. Åström and B. Wittenmark, 'Adaptive Control', Addison-Wesley, 1989.

[11] A. Balestrino, G. De Maria and L. Sciavicco, 'An Adaptive Model Following Control for Robotic Manipulators', Transactions of the ASME Journal of dynamic Systems, Measurement, and Control, Vol. 105, pp. 143-151, September 1983.

[12] E. Bailey and A. Arapostathis, 'Simple sliding mode control scheme applied to robot manipulators', Int. J. of Control, Vol. 45, No. 4, pp.1197-1209, 1987.

[13] E.A. Barbashin, V. A. Tabueva and R. M. Eidinov, 'On the stability of a Variable-Structure Control System when the Sliding Conditions are Violated', Automation and Remote Control, Vol. 24, No.7, pp.810-816, July 1963.

[14] E.A. Barbashin and E. I. Gerashchenko, 'A Design Principle for Stabilisation Systems', Automation and Remote Control, Vol. 26, No.6, pp.989-997, June 1965.

[15] H. Berghuis, 'Model-Based Robot Control : from Theory to Practice', Ph.D. Thesis, University of Twente, 1993.

[16] M.A. Bermant and S. V. Emel'yanov, 'On a Method of Controlling a Certain Class of Objects with Varying Parameters', Automation and Remote Control, Vol. 24, No.5, pp.568-579, May 1963.

[17] C. Bissell, ' A. A. Andronov and the Development of Soviet Control Enginecring', IEEE Control System, Vol. 18, No. 1, Feb. 1998.

[18] G. Buja and A. Souliaev, 'A Variable-Structure Controller', IEEE Trans. Automatic Control, Vol. 33, No. 2, pp.206-209, Feb. 1988.

[19] Yi-Feung Chen, Tsutomo Mita and Shinji Wakui, 'A New and Simple Algorithm for Sliding Mode Trajectory Control of the Robot Arm', IEEE Trans. Automatic Control, Vol. 35, No. 7, pp.828-829, July 1990.

[20] N.P. Chironis, 'Mechanisms and Mechanical Devices Sourcebook', McGraw-Hill, ISBN 0-07-010918-4, 1991.

[21] P.I. Corke , 'High-Performance Visual Closed-loop Robot Control', Ph.D. Thesis, University of Melbourne, July 1994.

[22] P.I. Corke , 'A Robotics Toolbox for MATLAB', IEEE Robotics and Automation Magazine, Vol.3, No.1, Mars 1996.

[23] J.J. Craig, 'Adaptive control of mechanical manipulators', Addison-Wesley, Reading, Ma, USA, 1988.

[24] J.J. Craig, 'Introduction to Robotics : Mechanics and Control', second edition, Addison-Wesley, 1989.

[25] S. Davis and D. Chen, 'High Performance Brushless DC Motors for Direct Drive Robot Arm', Jour. Powerconversion & Intelligent Motion, Vol. 11, No. 8, pp 34-38, Aug. 1985.

[26] A. Datta and M.T. Ho, 'Computed Torque Control of Rigid Robots with Improved Transient Performance', Proc. of the American Control Conference, San Francisco, CA, pp.1418-1422, June 1996.

[27] S. Dubowsky, 'A Perspective of the advancement of Robotic Systems during the past 15 years', SYROCO'97, Nantes, Sept. 1997.

[28] S.V. Emel'yanov and V. A. Taran,'Stabilization of Variable-Structure Automatic Control Systems by means of Inertial Elements with Variable Time Constants', Automation and Remote Control, Vol. 25, No.6, pp.790-794, June 1964.

[29] J. Erschler, F. Roubellat and J. P. Vernhes, 'Automation of a Hydroelectric Power Station Using Variable-Structure Control System', Automatica, Vol. 10, pp.31-36, 1974.

[30] A.F. Filippov, 'Differential equations with Discontinuous right hand-sides', Mathematicheskii Sbornik,Vol. 51, no.1, 1960. American Mathematical Society Translation 42, pp.199-231, 1964.

[31] I. Flügge-Lotz and M.D. Maltz, 'Altitude Stabilization using a Contactor Control System with a Linear Switching Criterion', Automatica, Vol. 2, pp.255-274, 1965.

[32] J.B. Gamble and N.D. Vaughan, 'Comparison of Sliding Mode Control with State feedback and PID Control Applied to a Proportional Solenoid Valve',

Transactions of the ASME Journal of dynamic Systems, Measurement, and Control, Vol.118, September 1996.

[33] W. Gao and J. C. Hung, 'Variable Structure Control of Non-linear Systems: A New Approach', IEEE Trans. on Indust. Electr. Vol. 40, No. 1, Feb. 1993.

[34] E. Gilbert and I.J. Ha, 'An approach to non-linear feedback control with applications to robotics', IEEE Trans. Syst. Man, Cybern., Vol.SMC-14, pp.879-884, 1984.

[35] R. Gorez, 'Sliding mode as a rational approach to robot controller design', Proc. Modelling and Control of Mechanical Systems, ed. A. Astolfi, D.J.N. Limebeer, C. Melchiorri, A. Tornambè & R.B. Vinter, Imperial College Press, 17-27 June1997.

[36] R. Gorez, G. Antonelli and C. Ganseman, 'Design of PID Robot Controllers Via VSS Approach', Proc. 5th. Symposium on Robot Control (SYROCO'97), Nantes, France, pp. 137-144, Sept. 1997.

[37] J. Guldner and V. Utkin, 'Sliding Mode Control for Gradient Tracking and Robot Navigation Using Artificial Potential Fields', IEEE Trans. Robotics and Automation, Vol. 11, No. 2, pp.247-254, April 1995.

[38] S.R. Habibi and R.J. Richards, 'Sliding mode control of an electrically powered industrial robot', IEE Proc.-D, Vol. 139, No. 2, pp. 207-225, 1992.

[39] R.S. Hartenberg and Denavit, 'A Kinematic notation for lower pair mechanisms based on matrices', ASME Journal of Applied Mechanics, Vol.77, pp.215-221, June 1955.

[40] R.L. Huston and F.A. Kelly,' The Development of Equations of Motion of Single-Arm Robots', IEEE Trans. SMC, Vol.12, 1982.

[41] Intel Corporation, Pipers Way, Swindon, Wilts, SN3 1RJ, ' $80960KB$ Datasheet', Dec. 1989, Order number: $270565-004$.

[42] ISO 9283:1991, 'Manipulating industrial robots: performance criteria and related testing methods, Path Characteristics', 1991.

[43] U. Itkis, 'Control System of Variable Structure', John Wiley and Sons, 1976.

[44] R. Johansson, 'System Modeling and Identification', Prentice Hall, 1993.

[45] T.R. Kane and D.A. Levison, 'The use of Kane's Dynamical Equations in Robotics', Int. J. Rob. Res., Vol. 2, 1983.

[46] H. Kawasaki, T. Bito and K. Kanzaki, 'An Efficient Algorithm for the Model-Based Adaptive Control of Robotic Manipulators', IEEE Transactions on Robotics and Automation, Vol.12, No.3, pp.496-501, 1991.

[47] H. Kazerooni and S. Kim, 'On the Design and Construction of Direct-Drive Robots', Trans. ASME, Jour. Engineering for Industry, Vol. 112, pp 197-201, May 1990.

[48] R. Kelly, 'Adaptive computed torque plus Compensation Control for Robot Manipulators', Mech. Mach. Theory, Vol.25, No.2, pp.161-165, 1990.

[49] R. Kelly and R. Salgado, 'PD Control with Computed Feedforward of Robot Manipulators : A Design Procedure', IEEE Trans. on Robotics and Automation, Vol.10, No.4, 1994.

[50] R. Kelly, V. Santibanez and F. Reyes, 'On Saturated-Proportional Derivative Feedback with Adaptive Gravity Compensation of Robot Manipulators', Int. J. Adapt. Cont. and Sig. Proc., Vol.10, pp.465-479, 1996.

[51] P.K. Khosla and T. Kanade, 'Real-Time Implementation and Evaluation of Computed-Torque Scheme', IEEE Transactions on Robotics and Automation, Vol.5, No.2, April 1989.

[52] Y.D. Landau, ' Adaptive Control', Ed. Dekker, 1979.

[53] C.S.G. Lee, B.H. Lee and R. Nigam,' Development of the generalised d'Alembert equations of motion for mechanical manipulators', Proc. 6th IFAC Conf. Dec. and Cont., 1982.

[54] Dai Gil Lee, Ki Soo Kim and Yoon Keun Kwak, ' Manufacturing of a SCARA type Direct-Drive Robot with Graphite Fibre Epoxy Composite Material', Robotica, Vol. 9, pp 219-229, 1991.

[55] F.L. Lewis, C.T. Abdallah and D.M. Dawson, 'Control of Robot Manipulators', Macmilan Publishing Company, 1993.

[56] F. L. Lewis, K. Liu and A. Yesildirek, 'Neural Net Robot Controller with Guaranteed Tracking Performance', IEEE Trans. Neural Networks, Vol.6, No. 3, pp.703-713, May 1995.

[57] Q. Li, S.K. Tso and W. J. Zhang, 'Trajectory tracking control of robot manipulators using neural-network-based torque compensator', Proc Instn Mech Engrs, Vol.212, Part I, pp.361-372, 1998.

[58] J.Y.S. Luh, 'Conventional Controller Design for Industrial Robots - A Tutorial', IEEE Trans. Syst. Man, Cybern., Vol.SMC-13, pp.298-316, 1983.

[59] J.Y.S. Luh, M.W. Walker and R.P.C. Paul, 'One-Line Computational Scheme for Mechanical Manipulators', ASME Journal of dynamic Systems, Measurement, and Control, Vol.102, June 1980, pp.69-76.

[60] G Maliotis, 'A hybrid model reference adaptive control/computed torque control scheme for robotic manipulators', Proc. Instn Mech. Engrs, Vol.205, pp.215-221, 1991.

[61] G.A. Medrano-Cerda, E.E. Elduklin and M. Cetin, 'Balancing and attitude control of double and triple inverted pendulums', Trans. Inst. M.C., Vol.17, No.3, 1995.

[62] K. Miller, 'Experimental Verification of Modelling of DELTA Robot dynamics by Direct Application of Hamilton's Principle', IEEE International Conference on Robotics and Automation, 1995.

[63] Y.I. Neimark, 'Oscillatory Processes (ramp action) in Relay Automatic Control Systems', Automation and Remote Control, No.1, pp.29-35, 1957.

[64] P. Nigrowsky and P. J. Turner, 'Hybrid Controller Algorithms for the purpose of fast Tracking of a Non-linearly Actuated Robotic Arm', Proc. of 3rd Int. ICSC Symp. on IIA and SOCO, pp. 127-132, Genova, Italy, June 1999.

[65] P. Nigrowsky and P. J. Turner, 'Variable Structure Control of Robot Manipulators with Varying Bounds', Proc. of 7th IEEE Mediterranean Conf. on Control and Automation, MED'99, pp. 2285-2291, Haifa, Israel, June 1999.

[66] P. Nigrowsky, A. Visioli and P. J. Turner, 'Practical Evaluation of V.S.C. with Varying Bounds of Robot Manipulator', Proc. of 2nd Inter. Conf. on Recent Advances in Mechatronics, ICRAM'99, pp. 62-67, Istanbul, Turkey, May 1999.

[67] Nippon Seiko K.K., Europe: Middlesex House, 29-45, High Street, Edgware, Middlesex. HA8 7DH, 'Megatorque Motor System User's Manual', fourth edition, May 1990, Document number: $C20003 - 004$.

[68] R. Ortega, 'Passivity-Based control of Euler-Lagrange systems: applications to robots, AC motors and power converters', Proc. Modelling and Control of Mechanical Systems, ed. A. Astolfi, D.J.N. Limebeer, C. Melchiorri, A. Tornambè & R.B. Vinter, Imperial College Press, 17-27 June1997.

[69] R. Ortega and M.W. Spong, 'Adaptive Motion Control of Rigid Robots : a Tutorial', Automatica, Vol.25, No.6, pp.877-888, 1989.

[70] S.K. Pal, 'Direct Drive High Energy Permanent Magnet Brush and Brushless DC Motors for Robotic Applications', In IEE Colloquium in Robot Actuators, Savoy Place, London, Digest No. 1991/146, Oct. 1991.

[71] R.P. Paul, 'Robot Manipulators : Mathematics, Programming, and Control', Cambridge, Massachusetts : MIT Press, 1981.

[72] R.M. Peters, 'Speed Variation with Elliptical Wheels', In Mechanisms 1973, pp. 1-6, The Institute of Mechanical Engineers, ISBN 0-85298-316-6, 1974.

[73] R.J. Richards, D.S. Reay,'The real-time application of VSS controllers: implementation on elements of a 'direct-drive' arm a *Syke* robot and a *PUMA* robot', Trans. Inst. MC, Vol.14, No.3, pp. 201-210, 1992.

[74] R.J. Richards, D.S. Reay,'The real-time application od Variable Structure System (VSS) controllers: an experimental comparative study on a simple robot arm', Trans. Inst. MC, Vol.13, No.4, pp. 139-152, 1991.

[75] P. Rocco, 'Stability of PID Control for Industrial Robot Arm', IEEE Trans. on Robotics and Automotion, Vol.12, No.4, 1996.

[76] N. Sadegh, R. Horowitz, 'An Exponentially Stable Adaptive Control Law For Robot Manipulators', IEEE Transactions on Robotics and Automation, Vol.6, No.4, pp491-496, 1990.

[77] H. Seraji, 'Decentralized Adaptive Control of Manipulators :Theory, Simulation, and Experimentation', IEEE Transactions on Robotics and Automation, Vol.5, No.2, pp.183-201, 1989.

[78] H. Sira-Ramirez, 'Dynamical sliding mode control strategies in the regulation of non-linear chemical processes', Int. J. Control, Vol. 56, pp.1-21, 1992.

[79] H. Sira-Ramirez, M. Zribi and S. Ahmad, 'Dynamical sliding mode control approach for vertical flight regulation in the helicopters', IEE Proc. Control Theory Appl., Vol. 1, No. 1, pp. 19-24, Jan. 1994.

[80] J.J.E. Slotine, 'Sliding controller design for non-linear systems', Int. J. Control, Vol.40, No.2, pp.421-434, 1984.

[81] J.J.E. Slotine, 'The Robust Control of Robot Manipulators', Int. J. of Robotics Research, Vol. 4, No. 2, pp.49-64, 1985.

[82] J.J.E. Slotine, 'Putting Physics in Control-The example of Robotics', IEEE Control Systems Magazine, Dec. 1988.

[83] J.J.E. Slotine and W. Li, 'On the Adaptive Control of Robot Manipulators', International Journal of Robotics Research, Vol. 6, No.3, pp.49-59, 1987.

[84] J.J.E. Slotine and W. Li, 'Composite Adaptive Control of Robot Manipulators', Automatica, Vol.25, No.4, pp.509-519, 1989.

[85] J.J.E. Slotine and W. Li, 'Applied Non-linear Control', Prentice Hall, 1991.

[86] J.J.E. Slotine and S.S. Sastry, 'Tracking control of non-linear systems using sliding surfaces, with application to robot manipulators', Int. J. Control, Vol.38, No.2, pp.465-492, 1983.

[87] M.W.Spong, J. De Schutter, H. Bruyninckx and J.T.Y. Wen, 'Control of Robots and Manipulators', The Control Handbook, ed. W.S. Levine, CRC/IEEE Press, 1995.

[88] S. Spurgeon and C. Edward, 'Non-linear control of a 175 kW gas Burner', IEE Jour. Computing & Control Eng., pp.29-34, Feb. 1995.

[89] M. Takegaki and S. Arimoto, 'A new Feedback Method for Dynamic Control of Manipulators', Transactions of the ASME Journal of dynamic Systems, Measurement, and Control, Vol.102, pp.119-125, June 1981.

[90] K. Takase, T. Hasegawa and T. Suehiro, 'Design and Control of a Direct Drive Manipulato', Proc. of the Int. Symp. on Design and Synthesis, Tokyo, Japan, pp. 333-338, July 1984.

[91] P. Tomei, 'Adaptive PD Controller for robot Manipulators', IEEE Trans. Robotics and Automation, Vol.7, pp.565-570, 1991. R. K elly, 'Commcnts on : Adaptive PD ...', IEEE Trans. Robotics and Automation, Vol.9, pp.117-119, 1993.

[92] Y.Z. Tsypkin, 'A. A. Andronov and Automatic Control theory', Automation and Remote Control, No.5, pp.697-700, May 1974.

[93] P. J. Turner, P. Nigrowsky and G. Vincs, 'Improved Performance for Robot Manipulators via Transmission Re-Design', To be published in Robotica, 1999.

[94] S.G. Tzafestas and P.A. Prokopiou, 'Compensation of Teleoperator Modelling Uncertainties with a Sliding Mode Controller', Robotics & Computer-Integrated Manufacturing, Vol. 13, No. 1, pp.9-20, 1997.

[95] V.I. Utkin, 'Equations of the slipping regime in discontinuous systems, I ', Automation and Remote Control, No.12, pp.1897-1907, 1971.

[96] V.I. Utkin, 'Equations of the slipping regime in discontinuous systems, II ', Automation and Remote Control, No.2, pp.211-219, 1972.

[97] V.I. Utkin, 'Variable Structure Systems with Sliding Mode', IEEE Transactions on Automatic Control, Vol. ac-22, No.2, April 1977.

[98] V.I. Utkin, 'Sliding Mode Control and its Applications to Variable Struture Systems', Mir : Moscow, 1978.

[99] V.I. Utkin, 'Sliding Modes in Control Optimization', Springer-Verlag, 1992.

[100] G. Vines, 'Linear and Non-linear Direct Transmissions : A New Approach to the Design of Robot Joint Transmissions', Ph.D. Thesis, Elect. Dept., Brunel University, 1996.

[101] M.W. Walker and D.E. Orin, 'Efficient dynamic Computer Simulation of Robotic Mechanisms', Journal of Dyn. Syst., Meas., and Cont., Vol.104, Sept. 1982.

[102] R. Welburn, ' Ultra High Torque Motor System for Direct Drive Robotics', In Robot 8, Detroit, pp. 19-71, June 1984.

[103] Kai S. Yeung and Yon P. Chen, 'A New Controller Design for Manipulators Using the Theory of Variable Structure Systems', IEEE Trans. on Automatic Control, Vol.33, No.2, 1988.

[104] K.K.D. Young, 'Controller design for a manipulator using the theory of variable structure systems' , IEEE Trans. Syst. Man Cybern., Vol. SMC-8, No.2, pp.101-109, 1978.

[105] K. K. D. Young, 'Design of Variable Structure Model-Following Control System', IEEE Trans. Automatic Control., Vol. AC-23, No.6, pp.1079-1085, 1978.

[106] Chee-Fai Yung and Shih-Duen Lin,' New Smooth Approximation of Variable Struture Systems with Application to Tracking Control', Proc. 33rd Conf. Decision and Control, Lake Buena Vista, FL, Dec. 1994.

[107] A.S.I. Zinober, O.M.E. El-Ghezawi and S.A. Billings, 'Multivariable variable-struture adaptive model-following control systems', IEE Proc., Vol.129, Pt.D, No.1, pp.6-12, 1982.

8 APPENDIX A : THE DENAVIT-HARTENBERG CONVENTION

Any robot can be described kinematically by giving the value of four quantities for each link. Two describe the link itself, and two describe the link's connection to a neighboring link. In the usual case of a revolute joint, θ_i is called the joint variable, and the other three quantities would be fixed link parameters. For prismatic joints, d_i is the joint variable and the other three quantities are fixed link parameters. The definition of mechanisms by means of these quantities is a convention usually called the Denavit-Hartenberg notation [39]. Other methods of describing mechanisms are available.

The Denavit-Hartenberg method is a matrix method that systematically assigns co-ordinate systems to each link of an articulated chain. In the case of a revolute joint, the axis of this joint i is aligned with Z_{i-1}. The X_{i-1} axis is directed along the normal from Z_{i-1} to Z_i and for intersecting axes is parallel to $Z_{i-1} \times Z_i$. The link and joint parameters may be summarised with figure 89.

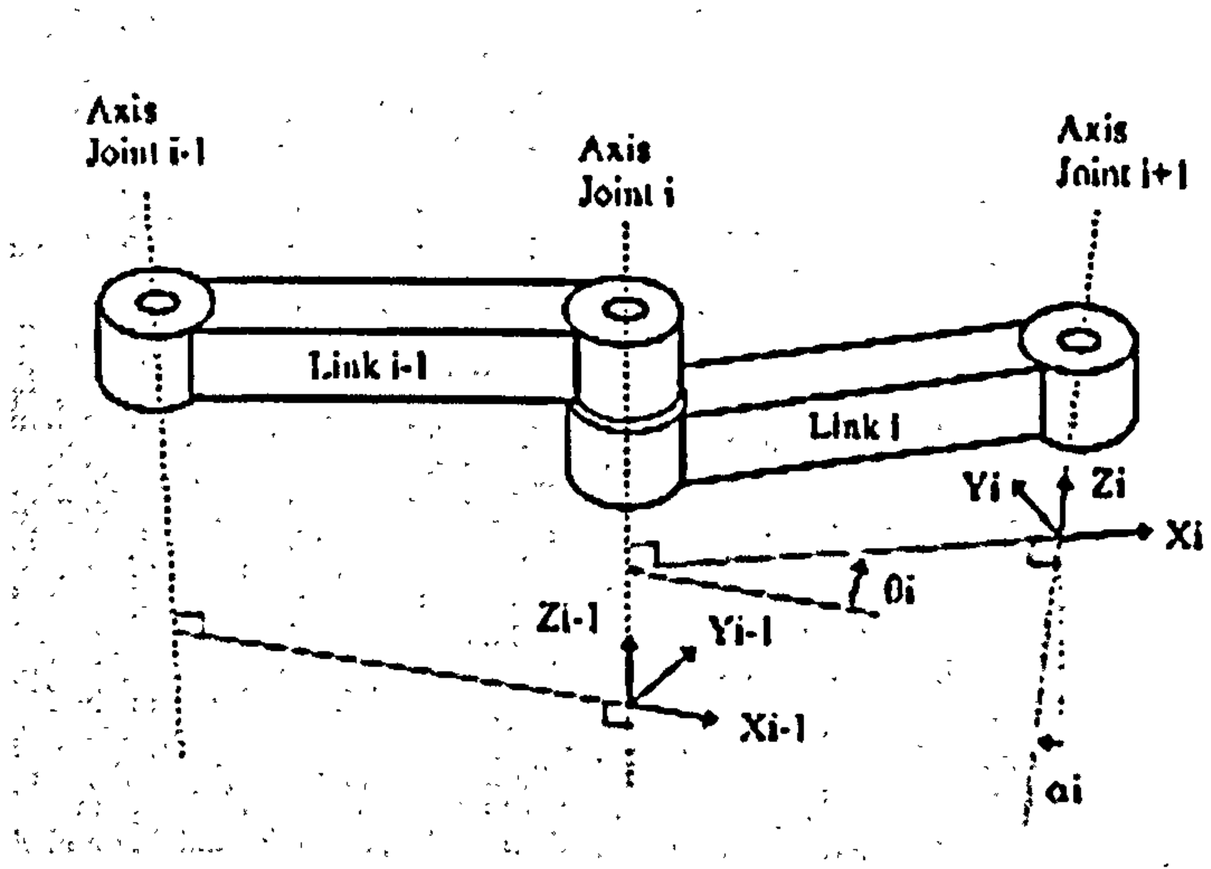


Figure 89: The Denavit-Hartenberg notation

Link length a_i : the offset distance between the Z_{i-1} and Z_i axes along the X_i axis.

Link twist α_i : the angle from the Z_{i-1} axis to the Z_i axis about the X_i axis.

Link offset d_i : the distance from the origin of frame $i-1$ to the X_i axis along the Z_{i-1} axis.

Joint angle θ_i : the angle between the X_{i-1} and X_i axes about the Z_{i-1} axis.

The Denavit-Hartenberg representation results in a 4×4 homogeneous transformation matrix

$$
{}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{99}
$$

representing each link's co-ordinate frame with respect to the previous link's co-ordinate system; that is

$$
{}^0T_i = {}^0T_{i-1} \cdot {}^{i-1}A_i \tag{100}
$$

where ${}^0T_i$ is the homogeneous transformation describing the link's co-ordinate frame i with respect to the world co-ordinate system 0.

9 APPENDIX B :MODELLING OF THE SPRINT

The initial step in the determination of the dynamic equations is to identify the inertia, mass, C.O.G. and length of each element and to have an exact description of the transmission. The length of the element can be obtained directly from measurement on the robot and the schematic. The masses for some of the robot parts were available from G. Vines [100]. To determine the missing masses and the inertia and C.O.G., a C.A.D. method is used. Some assumptions are made during this phase i.e. the system is considered to be made of five major parts, namely the forearm, the forearm gimbal drive (gimbal drive for the elbow joint), the humerus, the humerus gimbal drive (gimbal drive for the shoulder joint) and the torso. The base of the manipulator is not considered since it does not move and so does not affect the system dynamics.

9.1 Determination of Mass, Inertia and C.O.G.

The forearm , figure 90 constitutes only the last link of the manipulator. It is assumed to be made entirely of aluminum. The general shape as well as the dimensions are respected as much as possible. However, fixing holes for screws, nuts and washers are neglected. Tapped holes are replaced by plain surfaces since screws are mounted onto it. The frame chosen for the determination of the C.O.G. is located at the joint, the X-axis is directed along the link, the Y-axis is pointing upward, and the Z-axis is directed along the joint axis. The values obtained for the forearm are :

Volume of the forearm

$$\text{Volume} = 0.00022047 m^3$$

The forearm mass

$$\text{mass} = \text{Volume} \cdot \text{Density(aluminium)}$$

$$M_{forearm} = 0.00022047 \cdot 2699 = 0.595kg \tag{101}$$

Location of the forearm C.O.G.

$$\begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix} = \begin{bmatrix} 0.0389 \\ 0.0009 \\ 0.0 \end{bmatrix} m \tag{102}$$

Inertia of the forearm at the C.O.G.

$$I_{forearm} = \begin{bmatrix} 0.0007 & 0.0 & 0.0 \\ 0.0 & 0.0074 & 0.0 \\ 0.0 & 0.0 & 0.0073 \end{bmatrix} kg.m^2 \tag{103}$$

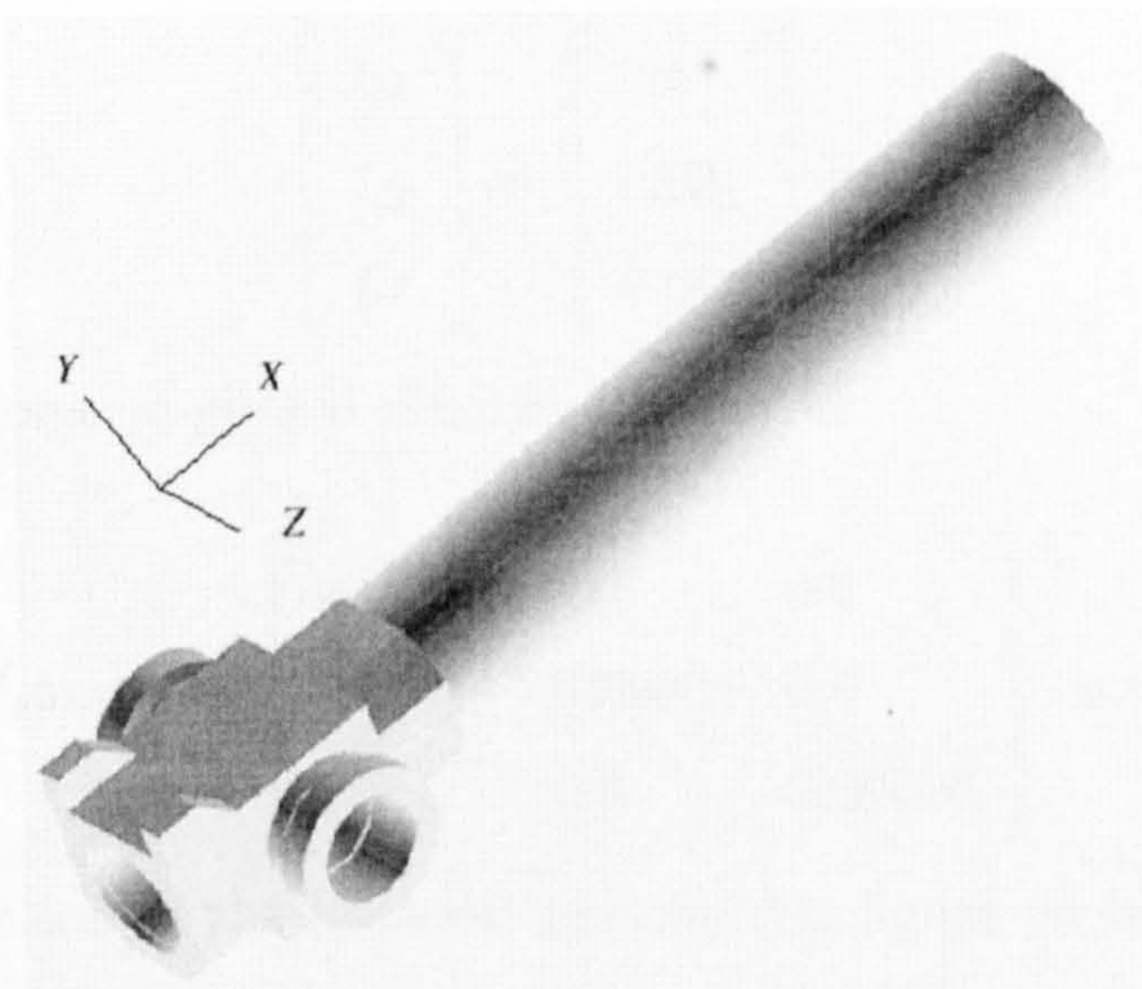


Figure 90: Forearm

The forearm gimbal drive, figure 91 represents the complete transmission system for the forearm except for the two transmission linkages which are considered in the humerus analysis. Although some of the parts have specific movement with respect to the others, the whole transmission is considered as a unique block. Nevertheless the shape of each element is respected. The frame is located at the base of the transmission system, with the X-axis pointing on the right side, the Y-axis coming

to the front, and the Z-axis along the joint axis. The system is considered to be made only of aluminum.

Volume of the forearm gimbal drive

$$\text{Volume} = 0.000076m^3$$

The forearm gimbal drive mass

$$\begin{aligned} \text{Mass} &= \text{Volume} \cdot \text{Density(aluminium)} \\ M_{gim-for} &= 0.000076 \cdot 2699 = 0.204kg \end{aligned} \tag{104}$$

Location of the forearm gimbal drive C.O.G.

$$\begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.03 \\ 0.004 \end{bmatrix} m \tag{105}$$

Inertia of the forearm gimbal drive at the C.O.G.

$$I_{gim-for} = \begin{bmatrix} 0.00006 & 0.0 & 0.0 \\ 0.0 & 0.00003 & 0.0 \\ 0.0 & 0.0 & 0.00006 \end{bmatrix} kg.m^2 \tag{106}$$

The determination of the humerus parameters, figure 92 is done in two parts. For the first part, only the elements made of aluminium are considered. The two transmission linkages for the forearm are considered as fixed elements of the humerus because of the small amplitude of their movements and their relatively small sizes so that the contribution of their displacement to the dynamics can be neglected. In the second part of the determination, the steel elements are considered. Such partitioning of the procedure is due to the presence of the large proportion of steel elements, i.e. the bearing and mounting elements. As for the forearm, the frame is placed at the joint, with the X-axis directed along the humerus axis, the Y-axis pointing upward and the Z-axis directed along the joint axis. The values are:

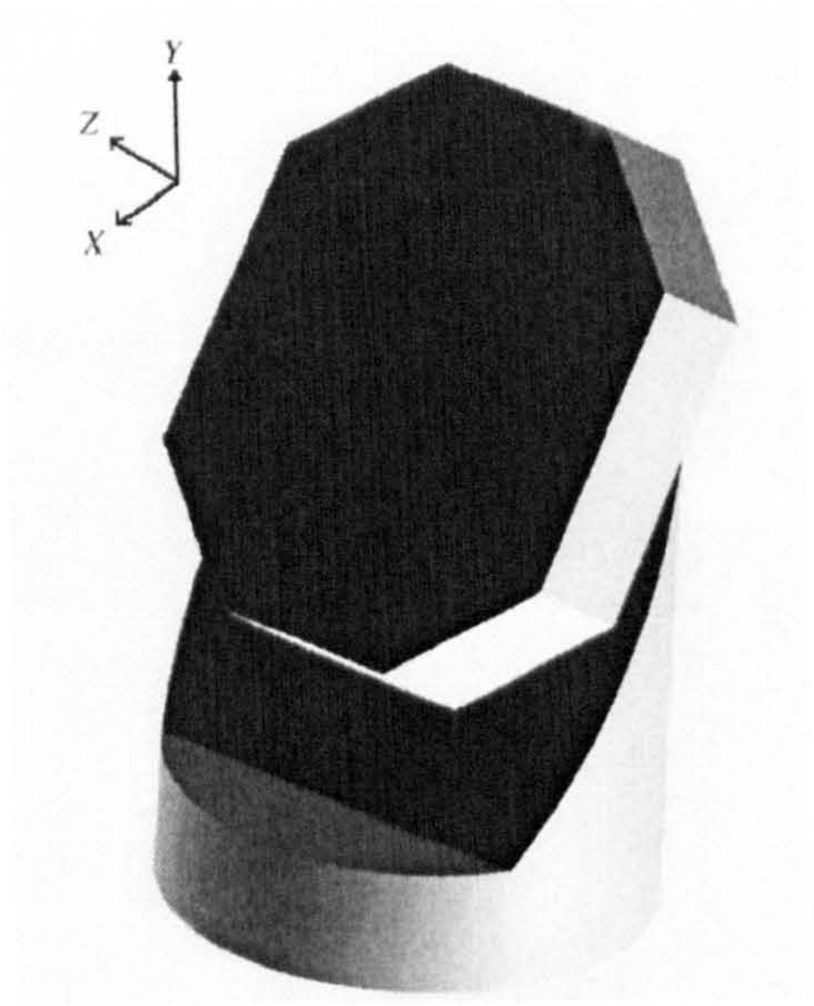


Figure 91: Gimbal drive transmission of the forearm

Volume of the humerus without the bearings

$$\text{Volume} = 0.001026m^3$$

The humerus mass without the bearings

$$\begin{aligned} \text{Mass} &= \text{Volume} \cdot \text{Density(aluminium)} \\ M_{hum-without-bearing} &= 0.001026 \cdot 2699 = 2.77kg \end{aligned} \tag{107}$$

Volume of the bearings

$$\text{Volume} = 0.0002623m^3$$

The bearing mass

$$\begin{aligned} \text{Mass} &= \text{Volume} \cdot \text{Density(Steel)} \\ M_{bearing} &= 0.000263 \cdot 5870 = 1.54kg \end{aligned} \tag{108}$$

The humerus mass

$$M_{humerus} = 2.77 + 1.54 = 4.77kg \tag{109}$$

Location of the humerus C.O.G.

$$\begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix} = \begin{bmatrix} 0.12 \\ 0.0 \\ 0.0 \end{bmatrix} m \tag{110}$$

Inertia of the humerus at the C.O.G.

$$I_{humerus} = \begin{bmatrix} 0.23 & -0.061 & 0.0 \\ -0.061 & 0.7 & 0.0 \\ 0.0 & 0.0 & 0.66 \end{bmatrix} kg.m^2 \tag{111}$$

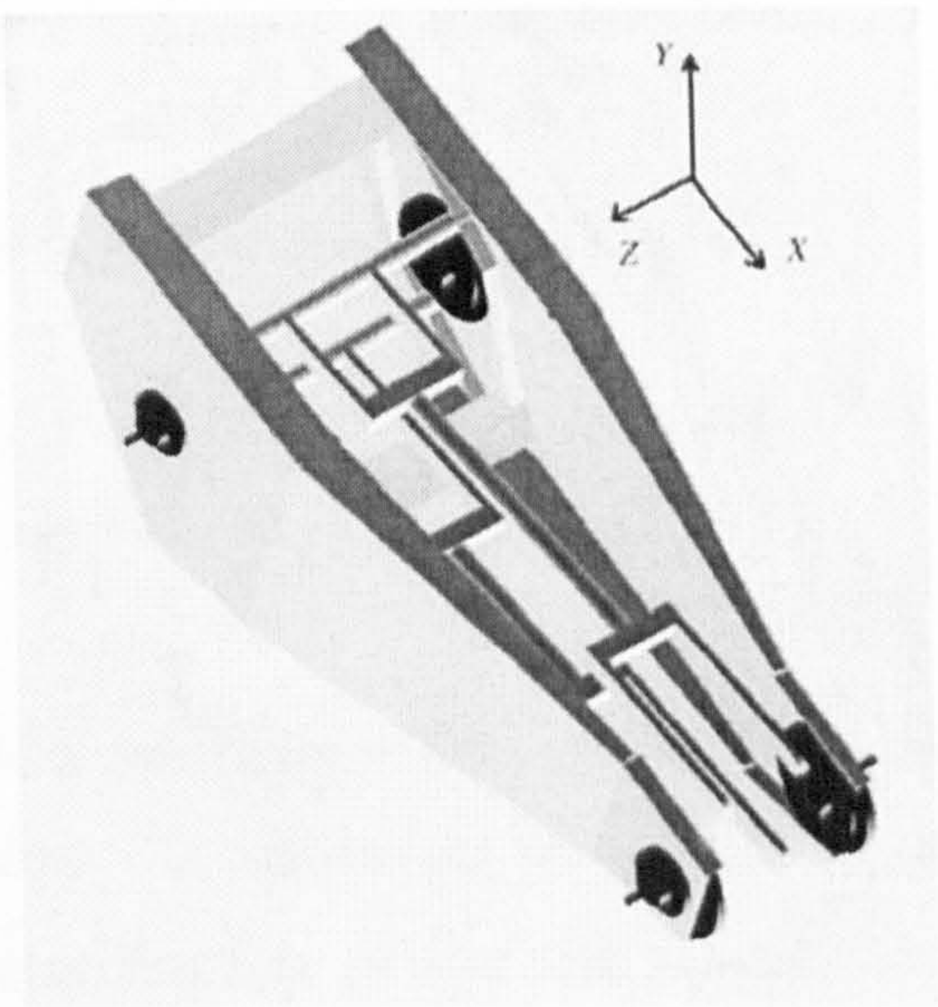


Figure 92: Humerus

The humerus gimbal drive, figure 93 consists only of the truncated cylinder, the other part of the transmission is considered as part of the torso. This assumption is justified by the fact that the second element of the humerus gimbal drive rotates with the torso, and its displacement with respect to the torso is small. The frame is located in the middle and at the base of the truncated cylinder, with the X-axis

from the front, the Y-axis pointing upward and the Z-axis directed to the left. The block is considered to be made entirely of aluminum.

Volume of the humerus gimbal drive

$$\text{Volume} = 0.000857m^3$$

The humerus gimbal drive mass

$$\begin{aligned} \text{Mass} &= \text{Volume} \cdot \text{Density(aluminium)} \\ M_{gim-hum} &= 0.000857 \cdot 2699 = 2.31kg \end{aligned} \tag{112}$$

Location of the humerus gimbal drive C.O.G.

$$\begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix} = \begin{bmatrix} -0.035 \\ 0.048 \\ 0.0 \end{bmatrix} m \tag{113}$$

Inertia of the humerus gimbal drive at the C.O.G.

$$I_{gim-hum} = \begin{bmatrix} 0.056 & -0.0115 & 0.0 \\ -0.0115 & 0.06 & 0.0 \\ 0.0 & 0.0 & 0.0377 \end{bmatrix} kg.m^2 \tag{114}$$

The torso, figure 94 is the element that supports the humerus link and accommodates its transmission. As mentioned previously, part of the humerus gimbal drive transmission is considered as part of the humerus because both parts rotate together, and that the amplitude of movement of the humerus gimbal drive is negligible with respect to the torso. The frame is located at the base and in the middle of the torso. The X-axis is coming from the front, the Y-axis is pointing upward and the Z-axis is directed to the left. The torso is considered to be made entirely of aluminum.

Volume of the torso

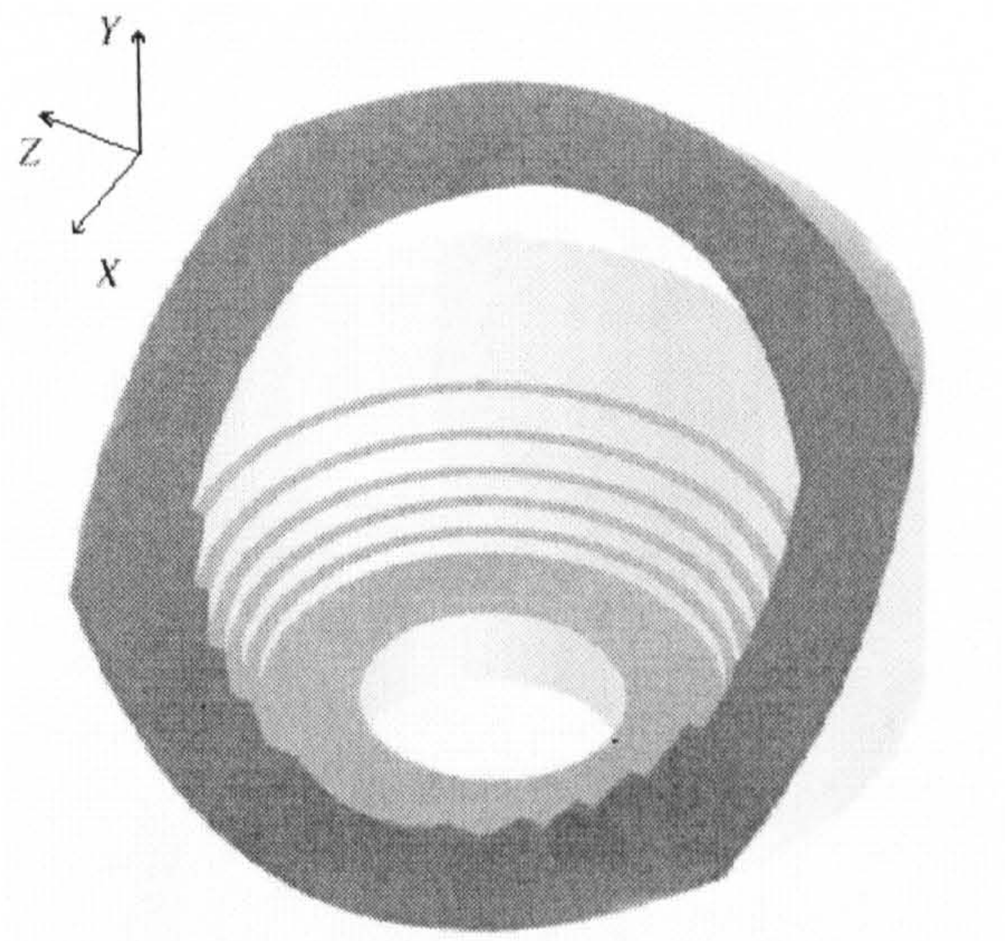


Figure 93: Gimbal drive transmission of the humerus

$$\text{Volume} = 0.00136m^3$$

The torso mass

$$\begin{aligned} \text{Mass} &= \text{Volume} \cdot \text{Density(aluminium)} \\ M_{torso} &= 0.00136 \cdot 2699 = 3.67kg \end{aligned} \tag{115}$$

Location of the torso C.O.G.

$$\begin{bmatrix} Lx \\ Ly \\ Lz \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.078 \\ 0.0 \end{bmatrix} m \tag{116}$$

Inertia of the torso at the C.O.G.

$$I_{torso} = \begin{bmatrix} 0.306 & 0.0 & 0.0 \\ 0.0 & 0.1835 & 0.0 \\ 0.0 & 0.0 & 0.21 \end{bmatrix} kg.m^2 \tag{117}$$

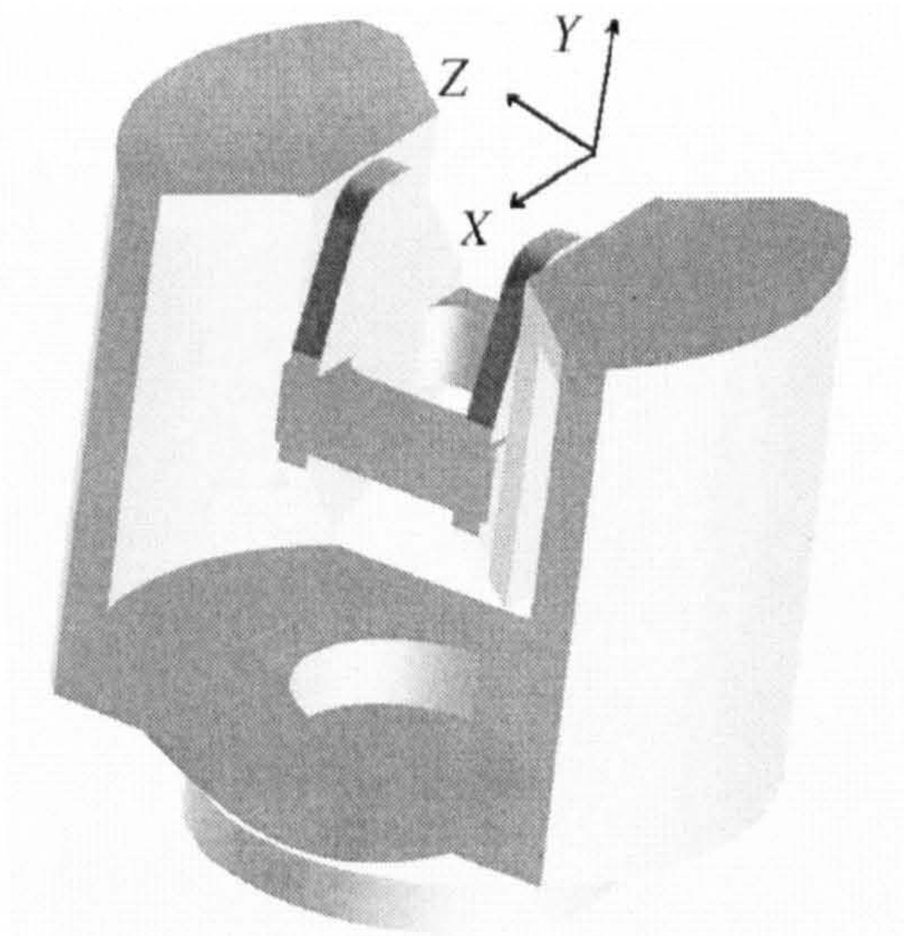


Figure 94: Torso

9.2 The Transmission Laws

The transmission element does not produce a standard 1 to 1 ratio as for direct drive nor even 1 to n as for a classic reduction transmission. The transmission elements for the humerus and forearm links are based on the sinusoidal reciprocator, see figure 1. The input-output relation for this type of transmission is relatively complex. By suitably selecting the coefficient of the transmission however it can provide a high reduction ratio while the manipulator is in a position demanding high torque (horizontal position of the link), and low or no reduction when the arm moves to a less demanding position (vertical position of the link).

The SPRINTA has three joints which are individually actuated, however the location of the motors and their transmission arrangement make the input angle of the forearm gimbal drive equal to the difference of the three motor shaft angles. Similarly, the input angle of the humerus gimbal drive is equal to the difference

between the humerus and waist motor angles.

The transmission law for the elbow is

$$\theta_{out-for} = \theta_{of-out-for} + \tan^{-1}\left(-\tan\theta_{c-for} \cdot \cos(\theta_{of-in-for} + \theta_3 - \theta_1)\right) \qquad (118)$$

where :

$\theta_{out-for}$ is the output angle for the forearm gimbal drive,

$\theta_{of-out-for} = \left(\frac{-13\cdot\pi}{36}\right)$ is the offset angle at the output,

$\theta_{c-for} = \left(\frac{11\cdot\pi}{36}\right)$ is the gradient of the non-linear transmission, and

$\theta_{of-in-for} = \left(\frac{8\cdot\pi}{180}\right)$ is the offset angle at the input.

θ_1, θ_2, and θ_3 are respectively the positions of the waist, shoulder and elbow motor shafts.

The transmission law for the shoulder is

$$\theta_{out-hum} = \theta_{of-out-hum} + \tan^{-1}\left(-\tan\theta_{c-hum} \cdot \cos(\theta_{of-in-hum} + \theta_2 - \theta_1)\right) \qquad (119)$$

where :

$\theta_{out-hum}$ is the output angle for the humerus gimbal drive,

$\theta_{of-out-hum} = \left(\frac{\pi}{4}\right)$ is the offset angle at the output,

$\theta_{c-hum} = \left(\frac{\pi}{4}\right)$ is the gradient of the non-linear transmission, and

$\theta_{of-in-hum} = \left(\frac{2\cdot\pi}{180}\right)$ is the offset angle at the input.

θ_1, and θ_2 are respectively the positions of the waist and shoulder motor shafts.

9.3 Kinematic Parameters of the SPRINTA

The kinematics represent the basic formulation for the derivation of the dynamics. They give the analytical relationship between the positions of the joints and the end point position and orientation. The joint position is defined as an angular position (radian or degree) in a local reference frame, while the end point position

and orientation are defined in the general reference frame (Cartesian and angular location). There are two fundamental aspects associated with the study of robot kinematics, the direct kinematic and the inverse kinematic solutions. The former is concerned with the general formulation of the end point location in terms of the positions of the joints, while the second represents the transformation of a desired position into joint positions. If the first solution is important to derive the location of the end point, the second solution is vital since the desired end point trajectory is specified in workspace co-ordinates. Both solutions involve the determination of the transformation matrix and the rotation matrix, which locate the co-ordinate of a link frame with respect to a previous link frame. To simplify the robot study, all manipulator descriptions are given with respect to a fixed reference frame, located at the base. Intermediate frames are also used recursively to bring the link positions and orientations to the reference frame. The conventional Denavit-Hartenberg frame assignment method is used as shown in figure 95.

The rotation matrix ${}^{i-n}R_i$, defines the orientation of link frame i with respect to the link frame $i-n$. For the SPRINTA, ${}^0R_3$, ${}^0R_2$ and ${}^0R_1$ are the rotation matrices that give the orientation of frames for links 3, 2 and 1 with respect to the base frame 0. These matrices are used for trajectory generation and for the determination of the inverse dynamics in the Lagrangian method.

$$
{}^0R_3 = \begin{bmatrix}
\cos(\theta_1)\cdot\cos(\theta_{out-hum}+\theta_{out-for}) & -\cos(\theta_1)\cdot\sin(\theta_{out-hum}+\theta_{out-for}) & \sin(\theta_1) \\
\sin(\theta_1)\cdot\cos(\theta_{out-hum}+\theta_{out-for}) & -\sin(\theta_1)\cdot\sin(\theta_{out-hum}+\theta_{out-for}) & -\cos(\theta_1) \\
\sin(\theta_{out-hum}+\theta_{out-for}) & \cos(\theta_{out-hum}+\theta_{out-for}) & 0
\end{bmatrix} \tag{120}
$$

$$
{}^0R_2 = \begin{bmatrix}
\cos(\theta_1)\cdot\cos(\theta_{out-hum}) & -\cos(\theta_1)\cdot\sin(\theta_{out-hum}) & \sin(\theta_1) \\
\sin(\theta_1)\cdot\cos(\theta_{out-hum}) & -\sin(\theta_1)\cdot\sin(\theta_{out-hum}) & -\cos(\theta_1) \\
\sin(\theta_{out-hum}) & \cos(\theta_{out-hum}) & 0
\end{bmatrix} \tag{121}
$$

$$ {}^{0}R_1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) \\ \sin(\theta_1) & 0 & -\cos(\theta_1) \\ 0 & 1 & 0 \end{bmatrix} \tag{122} $$

The position vector ${}^{i-n}P_i$ provides the Cartesian position of link frame i with respect to the previous link frame $i-n$.

$$ {}^{2}P_3 = \begin{bmatrix} l_3 \cdot \cos(\theta_{out-for}) \\ l_3 \cdot \sin(\theta_{out-for}) \\ 0 \end{bmatrix} \tag{123} $$

$$ {}^{1}P_2 = \begin{bmatrix} l_2 \cdot \cos(\theta_{out-hum}) \\ l_2 \cdot \sin(\theta_{out-hum}) \\ 0 \end{bmatrix} \tag{124} $$

$$ {}^{0}P_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix} \tag{125} $$

where :

l_3, l_2 and l_1 are the lengths of the common normals between the joint axes for links 3, 2 and 1 respectively.

$l_3 = 0.3m$, $l_2 = 0.4m$ and $l_1 = 0.565m$.

For the SPRINTA, the position vectors ${}^{2}P_3$, ${}^{1}P_2$ and ${}^{0}P_1$ define the positions of the link frames with respect to the previous link frame, because their position with respect to the base frame is not needed for determination of the inverse dynamics. As for the rotation matrices, the position vectors may be defined with respect to the base frame. The method is relatively straight forward using the 4×4 homogeneous transformation matrix ${}^{i-1}A_i$, equation (126).

$$ {}^{i-1}A_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}P_i \\ 0 & 1 \end{bmatrix} \tag{126} $$

Another important aspect of the system kinematics is the differential kinematics.

This gives the relationship between the velocities of the joints and the corresponding end-point linear and angular velocity. This relation for the end-effector velocities can be written as a linear function of the velocities of the joints using the Jacobian matrix. Two aspects are involved in the Jacobian matrix, the rotational and translational Jacobian matrix. In this case, we are only interested in the rotational Jacobian matrix since it is required for the determination of the inverse dynamics via the Lagrangian method.

Ja_3, Ja_2 and Ja_1 are the Jacobian rotation matrices for the frames of links 3, 2 and 1 with respect to the base frame 0.

$$Ja_3 = \begin{bmatrix} 0 & \sin(\theta_1) & \sin(\theta_1) \\ 0 & -\cos(\theta_1) & -\cos(\theta_1) \\ 1 & 0 & 0 \end{bmatrix} \tag{127}$$

$$Ja_2 = \begin{bmatrix} 0 & \sin(\theta_1) & 0 \\ 0 & -\cos(\theta_1) & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{128}$$

$$Ja_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{129}$$

It interesting to note that the rotational Jacobian matrices involve only the variable θ_1, which is due to the specific structure of the manipulator.

9.4 Dynamic Parameters of the SPRINTA

The dynamics are concerned with the general relationship relating the forces and torques acting on the manipulator and the accelerations, velocities and positions of the joints. As in the study of the kinematics, dynamics gives rise to two interpretations, i.e. the inverse dynamics and the forward dynamics. The inverse dynamics enable the expression of the torque at each joint in terms of the accelerations, velocities and positions of the joints. This relationship is very useful for the mechanical

Figure 95: Schematic representation of the SPRINTA

design of the structure and the selection of the actuators. Its use is also significant for the implementation of the control strategy. The forward dynamics provide the formulations of the accelerations, velocities and positions of the joints in terms of the torque at each joint. This relation is useful for simulation since it represents the dynamic model of the manipulator.

The following study is only concerned with the inverse dynamics problem because it may be used as a starting point for the formulation of a controller. Due to the complexity of the equations and their size, only the methodology with the general equations will be given in the remaining analysis.

The determination of the inverse dynamics via the Lagrange method, equation (1), requires knowledge of the kinetic and potential energies.

9.4.1 Kinetic Energy

Kinetic energy is the result of the velocity of an object, and this movement may be translational, rotational or both. In the case of robot manipulators, the rotational and translational velocities of the links contribute to the kinetic energy.

The translational kinetic energy is obtained from the link linear velocity at the C.O.G. in the X, Y and Z direction referred at the base. The initial link does not

have any linear velocity because it is a rotational joint.

Position of the humerus C.O.G. with respect to the base frame :

$$X_{humerus} = 0.12 \cdot \cos(\theta_{out-hum}) \cdot \cos(\theta_1) \quad (130)$$

$$Y_{humerus} = 0.12 \cdot \cos(\theta_{out-hum}) \cdot \sin(\theta_1) \quad (131)$$

$$Z_{humerus} = 0.565 + 0.12 \cdot \sin(\theta_{out-hum}) \quad (132)$$

where the terms are defined in equation (119).

Position of the forearm C.O.G. with respect to the base frame :

$$X_{forearm} = [0.4 \cdot \cos(\theta_{out-hum}) + 0.04 \cdot \cos(\theta_{out-for})] \cdot \cos(\theta_1) \quad (133)$$

$$Y_{forearm} = [0.4 \cdot \cos(\theta_{out-hum}) + 0.04 \cdot \cos(\theta_{out-for})] \cdot \sin(\theta_1) \quad (134)$$

$$Z_{forearm} = 0.565 + 0.4 \cdot \sin(\theta_{out-hum}) + 0.04 \cdot \sin(\theta_{out-for}) \quad (135)$$

where the terms are defined in equations (119) and (118).

Linear velocity of the humerus :

$$(V_{humerus})^2 = \left(\frac{dX_{humerus}}{dt}\right)^2 + \left(\frac{dY_{humerus}}{dt}\right)^2 + \left(\frac{dZ_{humerus}}{dt}\right)^2 \quad (136)$$

Linear velocity of the forearm :

$$(V_{forearm})^2 = \left(\frac{dX_{forearm}}{dt}\right)^2 + \left(\frac{dY_{forearm}}{dt}\right)^2 + \left(\frac{dZ_{forearm}}{dt}\right)^2 \quad (137)$$

The kinetic energy for the humerus is given by :

$$K^{linear}_{humerus} = \frac{1}{2} \cdot M_{humerus} \cdot (V_{humerus})^2 \quad (138)$$

The kinetic energy for the forearm is given by :

$$K_{forearm}^{linear} = \frac{1}{2} \cdot M_{forearm} \cdot (V_{forearm})^2 \tag{139}$$

The rotational kinetic energy is derived by considering the angular velocity acting on the inertia terms. In a general manner the rotational kinetic energy may be obtained as follow :

$$K_{link\ i}^{rotation} = \frac{1}{2} \cdot W_i^T \cdot {}^0R_i \cdot I_i \cdot {}^0R_i^T \cdot W_i \tag{140}$$

$$W_i = Jai \cdot \begin{bmatrix} \dot{\theta}_1 \\ ... \\ \dot{\theta}_n \end{bmatrix}$$

where W_i is the angular velocity vector of link i with reference to the base frame:

For the humerus :

$$K_{humerus}^{rotation} = \frac{1}{2} \cdot W_{humerus}^T \cdot {}^0R_2 \cdot I_{humerus} \cdot {}^0R_2^T \cdot W_{humerus} \tag{141}$$

$$W_{humerus} = Ja2 \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_{out-hum} \\ \dot{\theta}_{out-for} \end{bmatrix}$$

For the forearm :

$$K_{forearm}^{rotation} = \frac{1}{2} \cdot W_{forearm}^T \cdot {}^0R_3 \cdot I_{forearm} \cdot {}^0R_3^T \cdot W_{forearm} \tag{142}$$

$$W_{forearm} = Ja3 \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_{out-hum} \\ \dot{\theta}_{out-for} \end{bmatrix}$$

The derivation of the rotational kinetic energy for the transmission elements is similar, but their specific location means that they are only subject to the rotation

along the Z axis of the base frame.

For the torso :

$$K_{torso}^{rotation} = \frac{1}{2} \cdot W_{torso}^{T} \cdot {}^{0}R_1 \cdot I_{torso}^{*} \cdot {}^{0}R_1^{T} \cdot W_{torso} \tag{143}$$

$$W_{torso} = Ja1 \cdot \begin{bmatrix} \dot{\theta}_1 \\ 0 \\ 0 \end{bmatrix}$$

For the humerus gimbal drive :

$$K_{gim-hum}^{rotation} = \frac{1}{2} \cdot W_{gim-hum}^{T} \cdot {}^{0}R_1 \cdot I_{gim-hum}^{*} \cdot {}^{0}R_1^{T} \cdot W_{gim-hum} \tag{144}$$

$$W_{gim-hum} = Ja1 \cdot \begin{bmatrix} \dot{\theta}_2 - \dot{\theta}_1 \\ 0 \\ 0 \end{bmatrix}$$

For the forearm gimbal drive :

$$K_{gim-for}^{rotation} = \frac{1}{2} \cdot W_{gim-for}^{T} \cdot {}^{0}R_1 \cdot I_{gim-for}^{*} \cdot {}^{0}R_1^{T} \cdot W_{gim-for} \tag{145}$$

$$W_{gim-for} = Ja1 \cdot \begin{bmatrix} \dot{\theta}_3 - \dot{\theta}_1 \\ 0 \\ 0 \end{bmatrix}$$

I_{torso}^{*}, $I_{gim-hum}^{*}$ and $I_{gim-for}^{*}$ mean that the previously calculated inertia matrices have to be rearranged, so that the system frame of the robot manipulator and those of the transmission inertia match. Since the rotation takes place only along the vertical axis, the Y inertia terms and the Z inertia term have to be swapped around.

The total kinetic energy is given by :

$$K_{total} = K_{humerus}^{rotation} + K_{forearm}^{rotation} + K_{torso}^{rotation} + K_{gim-hum}^{rotation} + K_{gim-for}^{rotation} \tag{146}$$

9.4.2 Potential Energy

The potential energy is the result of the gravitational field acting on the C.O.G. of the links. Its determination is simpler as it involves only the static contribution of the link masses.

Using the $Z_{humerus}$ position equation of the humerus, equation (132). The potential energy can be derived as follows :

$$P_{humerus} = M_{humerus} \cdot g \cdot Z_{humerus} \tag{147}$$

For the forearm :

$$P_{forearm} = M_{forearm} \cdot g \cdot Z_{forearm} \tag{148}$$

where g is the gravitational acceleration, $9.81 m.\sec^{-2}$.

The total potential energy equals :

$$P_{total} = P_{humerus} + P_{forearm} \tag{149}$$

The final part in the determination of the inverse dynamics implies the calculation of the Lagrange operator, equation (2) and the application of the Lagrangian method, equation (1).

Remark 2 The complexity of the equations does not appear in the APPENDIX, because they have deliberately been kept simple and readable. The derivation of the inverse dynamics has been performed with MATHEMATICA®, following the above procedure.

10 APPENDIX C : DYNAMIC EQUATIONS OF THE 2 D.O.F. PLANAR MANIPULATOR

The robot manipulator used for the simulation is identical to the one used by Asada *et al.* [7, p. 229] and Gao *et al.* [33]. The geometry of the arm is shown in figure (96).

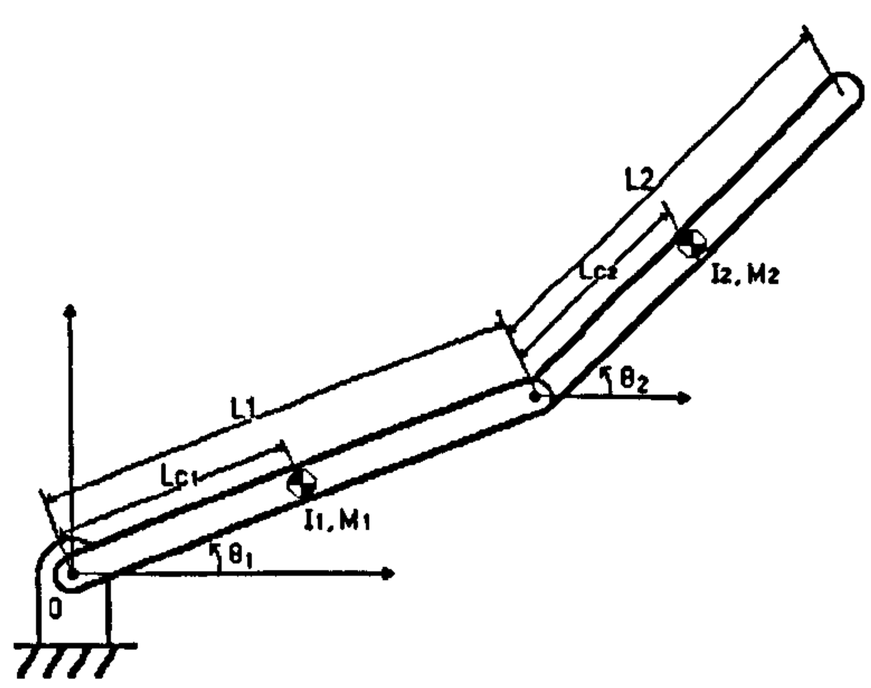


Figure 96: Geometry of the 2 D.O.F.'s planar Manipulator

The arm dynamics are described by the general equation (150).

$$T = M(\theta)\,\ddot{\theta} + C(\theta,\dot{\theta})\,\dot{\theta} + G(\theta) \tag{150}$$

where

$$M(\theta) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{151}$$

$$M_{11} = M_1.L_{c1}^2 + I_1 + M_2.(L_1^2 + L_{c2}^2 + 2.L_1.L_{c2}.\cos\theta_2) + I_2$$

$$M_{12} = M_{21} = M_2.L_1.L_{c2}.\cos\theta_2 + M_2.L_{c2}^2 + I_2$$

$$M_{22} = M_2.L_{c2}^2 + I_2$$

$$C(\theta,\dot{\theta})\,\dot{\theta} = \begin{bmatrix} C_{122}.\,\dot{\theta}_2^{\,2} + (C_{112}+C_{121}).\,\dot{\theta}_1\,.\,\dot{\theta}_2 \\ C_{211}.\,\dot{\theta}_1^{\,2} \end{bmatrix} \tag{152}$$

$$C_{111} = C_{222} = C_{212} + C_{221} = 0$$

$$C_{122} = -M_2.L_1.L_{c2}.\sin\theta_2$$

$$C_{112} + C_{121} = -2.M_2.L_1.L_{c2}.\sin\theta_2$$

$$C_{211} = M_2.L_1.L_{c2}.\sin\theta_2$$

and

$$G\,(\theta) = \begin{bmatrix} M_1.g.L_{c1}.\cos\theta_1 + M_2.g.\,(L_{c2}.\cos(\theta_1+\theta_2) + L_1.\cos\theta_1) \\ M_2.g.L_{c2}.\cos(\theta_1+\theta_2) \end{bmatrix} \tag{153}$$

The kinetic parameters are defined as follows :

	Link 1	Link 2
Inertia I_i (kg.m^2)	0.8	0.2
Mass M_i (kg)	20	10
Link Length L_i (m)	1	1
Center of Mass L_{ci} (m)	0.5	0.5

and with the gravitational acceleration, $g = 9.81\ m.\sec^{-2}$.