Atia, Khaled Roshdy El-Hosainy (2000) *Dynamic analysis, design and control of an industrial parallel robot.*

PhD thesis

UNIVERSITY
of
GLASGOW

Dynamic Analysis, Design and Control Of An Industrial Parallel Robot

A Thesis

Submitted in fulfilment for the degree of Doctor of Philosophy (Ph.D.)

By

Khaled Roshdy El-Hosainy Atia, M.Sc., Egypt.

Department of Mechanical Engineering,
University of Glasgow,
October 2000

List of Publications

1. Khaled R. Atia and M. P. Cartmell, " **A General Dynamic Model for a Large Scale Parallel Manipulator** ", *Robotica*, Vol. 17, PP. 675-683, 1999.

2. Khaled R. Atia and M. P. Cartmell, " **SEPA-ROBOT: A Serial-Parallel Manipulator With Singularity-Based Design** ", In: Proceedings of the IEEE International Conference on Industrial Technology, ICIT 2000, Vol. 1, PP. 49-54, Goa, India, 2000.

3. Khaled R. Atia and M. P. Cartmell, " **A New Methodology for Designing PD Controllers** ", accepted for publication in *Robotica*, August 2000.

ABSTRACT

In this work several investigations and developments have been carried out. The summary of these is as follow:

- An investigation into the applicability of the *bond-graph* methodology, using the so-called *Model Transformation Tools* software, has been undertaken to model parallel robots. This software is a novel, non-commercial, program developed at the University of Glasgow, and in addition to the standard bond graph, it contains a powerful tool called the *Hierarchical Bond Graph* for dealing with very large-scale dynamical systems. It is the first time this tool has been applied for the modelling of parallel manipulators.

- A *General Method* for modelling parallel robots using the *Hierarchical Bond-Graph* concept has been developed. The method is based on related work on the modelling of closed chain robots using the Lagrange method.

- Introduction of a new design concept to be known as *the Multi-cell Parallel Planar Manipulator*. The methodology allows for an increase in the workspace of the manipulator by increasing the number of cells without affecting the number of DOF. It can also be shown to enhance the manoeuvrability of the system.

- Application of the *multi-cell* approach to a specific 2-DOF planar parallel manipulator and recognition of the need for a general model led to the *development of a general dynamic model for the multi-cell manipulator* using the *Lagrange method.* The reason for using the Lagrange formulation is that the necessary generalisation cannot be formalised using the Bond Graph technique due to the dependency of a bond graph on the specified structure of the system being modelled.

- *Static balancing* of the new general manipulator was addressed and a *new method for balancing* has been introduced. The method reduces the number of parameters to be adjusted to only one.

- *Optimisation* of the location of the *redundant* motor in the manipulator was investigated. This was achieved through dynamic analysis of the one-cell manipulator based on different motor positions. The modelling *of the redundant actuated manipulator* was performed using the Hierarchical Bond Graph method developed for parallel robots. To the author's knowledge this is the first time *redundant actuated systems* have been modelled using the bond-graph methodology.

- The *development* of a new design for a one-cell manipulator using a *Singularity based Design* technique as proposed within this thesis. The design allows the operation of the manipulator in all known mechanical modes (Serial, Parallel, and Redundant Actuated Parallel). This enhances the versatility of the manipulator considerably. The idea was then generalised and led to the design of novel manipulators that are henceforth to be called *Parallel-Parallel manipulators*.

- Construction and manufacture of an experimental system to test the multi-cell planar parallel manipulator design. The robot has two cells and moves in the horizontal plane. The experimental system design also incorporates a high precision electronic drive for interfacing to the controlling computer.

- Investigations have been carried out into the efficiency of different non-linear control systems for application within the experimental system.

- The development of a new nonlinear, model-based, control technique based on the pre-existing computed torque and sliding mode control methods.

- The development of a new algorithm for the design of PD controllers that ensures uniform and fast dynamic responses, which are free from overshoots for all robot configurations. The technique also satisfies general stability requirements of the system.

Acknowledgements

I am indebted to the following:

- My God who has enabled and sustained me.
- Prof. M. P. Cartmell for his knowledge, guidance and criticisms. Without him the work reported in this thesis would not have been possible.
- My wife for the support and uncomplaining sacrifice.
- My parents for their encouragement and praying.
- Mr. Ian Russell and Mr. Kenneth Stevenson for their help.

Contents

List of Figures

CHAPTER 1

Figure

CHAPTER 2

Figure

CHAPTER 3

Figure

CHAPTER 4

Figure

CHAPTER 5

Figure

CHAPTER 6

Figure

CHAPTER 7

Figure

CHAPTER 8

Figure

CHAPTER 9

Figure

CHAPTER 10

Figure

List of Tables

CHAPTER 2

Table

CHAPTER 4

Table

CHAPTER 8

Table

Symbols and Abbreviations

Symbols

m	Mass
L	Length
J	Polar moment of inertia about the centre of mass
T	Kinetic energy
U	Potential energy
θ_i	Generalised co-ordinate of link i.
ω_i	Angular velocity of link i.
τ_i	Generalised force in the direction of the generalised co-ordinate.
x_j	x co-ordinates of point j.
y_j	y co-ordinates of point j.
v_j	Absolute resultant velocity of point j.
g	The gravitational acceleration vector.
$x_{c/i}$	x co-ordinate of the mass centre of the i^{th} link.
$y_{c/i}$	y co-ordinate of the mass centre of the i^{th} link.
$z_{c/i}$	z co-ordinate of the mass centre of the i^{th} link.
λ	Mass ratio.
k_a	Amplifier gain.
k_t	Motor torque constant.
b	Viscous friction coefficient.
$M(\Theta)$	Mass matrix.
$V_m(\Theta,\dot{\Theta})\dot{\Theta}$	Vector of Coriolis and Centrifugal forces.
Θ	Vector of output angular displacements.
$\dot{\Theta}$	Vector of output angular velocities.
$\ddot{\Theta}_d$	Desired acceleration vector.
$F(\dot{\Theta})$	Vector of friction forces acting at the manipulator joints.

E	Vector of servo errors.
$G(\Theta)$	Vector of gravity forces.
σ	Switching surface.
V_L	Lyapunov function.
T_d	Vector of all the perturbations and the unknown disturbances
I	Unity matrix.
e	Error.
$\widetilde{e}$	Scaled error.
t	Time.

Abbreviations

SMC	Sliding mode control.
P	Proportional control.
PD	Proportional plus derivative control.
PID	Proportional plus derivative plus integral control.
diag	Diagonal.
DOF	Degree of freedom.
RHS	Right hand side.
LHS	Left hand side.
MTT	Model Transformations Tools.

Chapter 1

Introduction

Because manufacturing versatility is an important issue in industry robots are frequently used [1]. They are ideally suited for industrial applications where working conditions typically involve high degrees of heat, noise, poisonous gases, risk of injury, boring work, and extreme physical exertion. The economical advantage of using industrial robots is down to their abilities to reduce production cost, to increase productivity, to improve product quality and to improve management control. Industrial robots are used today in many applications such as material handling, spot and arc welding operations, spray-painting, die-castings, metal forming, drilling operations and assembly operations.

In the early 1960s the first robot came into use. By the early 1980s there were more than 30000 robots working in Japan, Europe and U.S.A. The first generation of robots was unable to gather any information about its surroundings. Such robots could only perform pre-programmed motions. However, these types of robots included a wide range of designs from machines for material handling to advanced models which could perform complicated tasks such as welding and painting operations. The common feature of the robots of the first generation has been their ability to respond to instructions from a computer program without the need to change their basic set-up. Second generation of robots included all the features of first generation robots plus additional means for detailed communication with their surroundings. This communication has been accomplished by using sensors such as vision and tactile transducers.

1.1 Components of a Robot

A robot is a controlled mechanical device that replaces or partially replaces a worker in a factory. Although robots are not nearly as versatile as a human, they are much more flexible and universal than the automated machines. Externally, a robot is observed to have a *base*, an *arm*, and an *end effector*. The end effector carries a tool or a part that is usually called the workpiece. Inside the arm or the base are the *actuators* that cause the arm to move. A *control unit* is required to give instructions to the actuators. Certain *sensors* are fixed to the robot for two purposes: in order to collect information about the robot actions and its status, or for gathering information about the surrounding environment. Figure 1.1 shows the main parts of a robot.

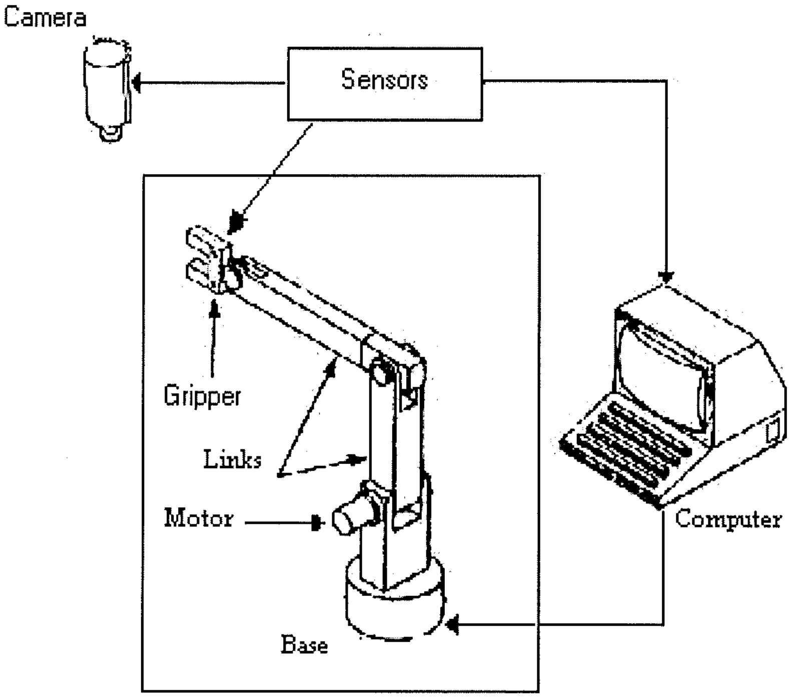


Fig. 1.1: The principal units of a robot fitted with sensors [2].

1.1.1 Arm

The arm is responsible for positioning the gripper with respect to the base. It consists of several links connected by joints that are either revolute or prismatic. The links are made rigid to support the load carried by the gripper. Usually they are hollow to reduce the weight of the arm and provide space for gearing and electrical cables. On the other hand, joints must be precisely configured to maintain accurately the arm position.

1.1.2 Wrist

A wrist is mounted on the end of the arm. The wrist, as shown in Figure 1.2, includes three joints which provide three rotational motions (Roll, Pitch and Yaw). It is important to note that it is desirable to have the wrist-roll, wrist-pitch and wrist-yaw occur at the same centre. This is because control and dexterity are both improved by such a concentric configuration.

1.1.3 End Effector

In robot terminology, the end of the wrist is the end of the arm. At this end the end effector is fixed. Typical end effectors are claws, grippers, vacuum pickups etc.

1.1.4 Actuators

Electric motors, hydraulic cylinders or pneumatic actuators can drive the robot. It is important to note that the joints of the robot can be driven directly or indirectly. In the first case and by using electric motors the joint shaft is coupled directly to the motor shaft. The disadvantage of this is that each motor forms static and dynamic loads on the lower motors. Of course this affects the load capacity of the robot. However, direct drive offers two important advantages which are better positioning accuracy and improved reliability. The first advantage is down to the absence of gear trains and, hence reduction in friction and no backlash. The improved reliability results from the decrease in the number of parts in a connection. In the indirect drive mechanical transmissions are used to connect the actuators to the joints. Despite the lack of reliability and accuracy in the indirect drives they improve the load capacity of the robot and can make its control simple.

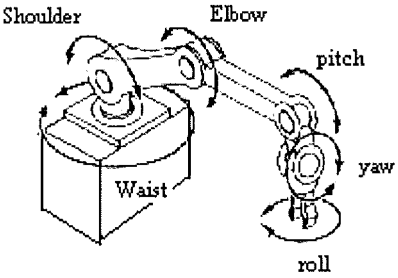


Fig. 1.2: A robot with three degrees of freedom wrist [3].

1.1.5 Control unit

The control unit contains all the necessary devices used to direct and sequence the robot links, wrist and end effector. Robots are activated by various kinds of computers, from microprocessors to minicomputers. Advanced robots require continuous processing and computations. However, this requires high-speed computers.

1.1.6 Sensors

Sensors are grouped into two classes:

- Internal sensors that measure variables within the robot structure. These can include position sensors, velocity sensors and force sensors.

- External sensors that gather information about the surroundings of the robot. These may include vision and touch sensors.

1.2 Classification of robots according to their workspace

Robots can be classified by their workspace into four types as follows,

- ***Cartesian Robots***

Cartesian robots are characterised by a small workspace, but have a high degree of rigidity. Therefore, they are suitable for machining operations. These robots can reach any part of a volume bounded by the length of the individual sections of the arm. Figure 1.3 (a) illustrates this type of robots.

- ***Cylindrical Robots***

Cylindrical robot arms consist of one revolute joint and two prismatic joints. The workspace of these arms is a cylinder. Due to the existence of the revolute joint the control of these robots is slightly more difficult than in Cartesian robots. A typical example of a cylindrical arm is shown in Figure 1.3 (b)

- ***Spherical Robots***

These robot arms have one linear joint and two revolute joints. They have a larger workspace and a lower rigidity than Cartesian and cylindrical robots. The control of these arms is more complicated than in cylindrical robots due to the revolute joints. A spherical arm can reach any point in a volume bounded by an inner and outer hemisphere. The radii of the two hemispheres correspond to the minimum and the maximum extensions of the linear part. Figure 1.3 (c) illustrates a typical example of a spherical arm.

- ***Articulated Robots***

This type of robot arm includes three revolute joints. Articulated arms are similar to human arms. Their workspace is larger than any other robot types. The control of these robots is very complicated when compared to other robot types. Figure 1.3 (d) shows a typical example of this kind of robot structures. Sometimes, a prismatic joint replaces the upper revolute joint and the axes of rotation of the other revolute joints are made vertically. This arrangement is appropriate for assembly operations. The advantage of this design is that the weights of the first two links do not affect the robot dynamics. Figure 1.2 (e) illustrates this kind of robot design.

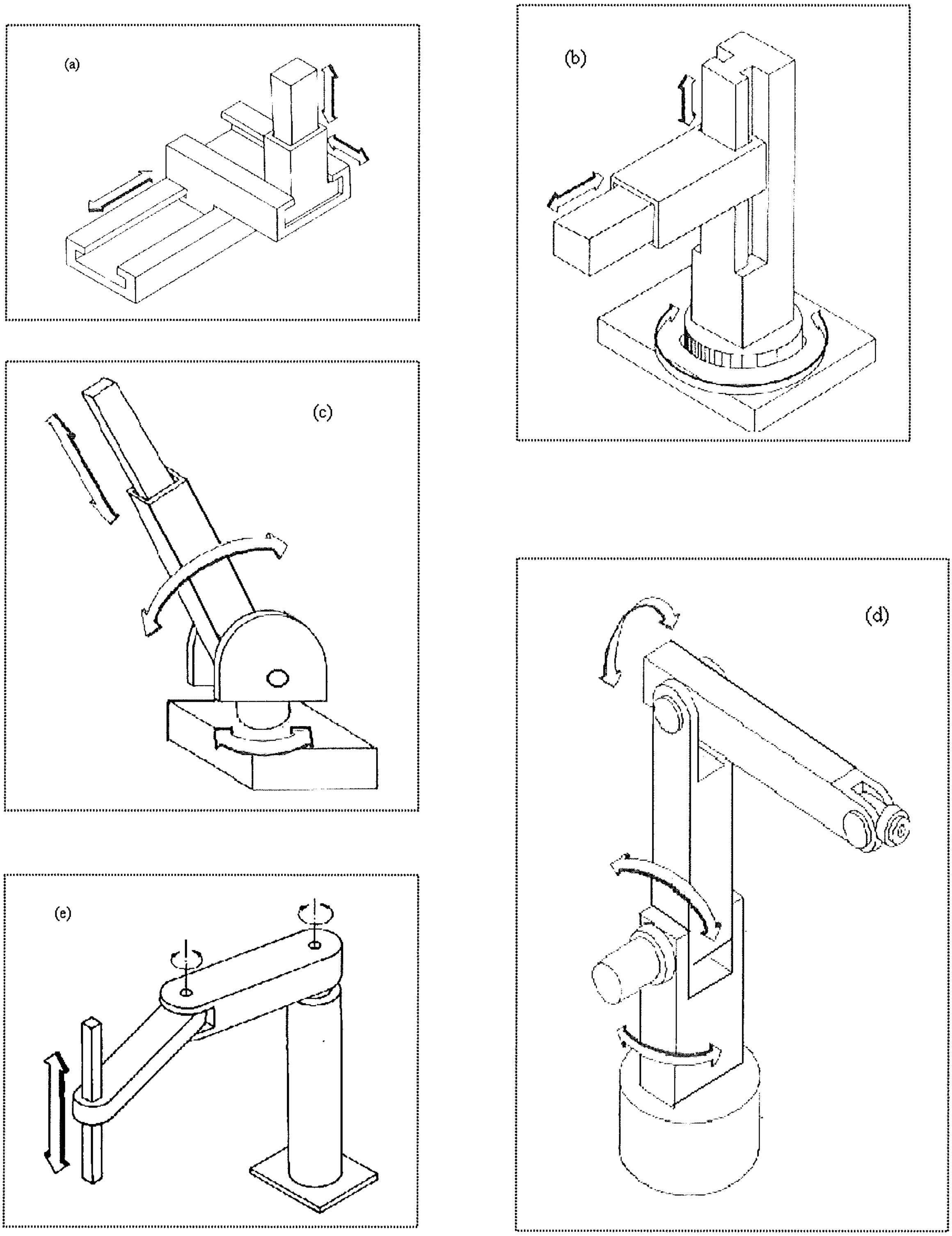


Fig. 1.3: Various kinds of robots [2],

(a) Cartesian robots, (b) Cylindrical robots, (c) Spherical robots
(d) Vertical articulated robots, (e) Horizontal articulated robots

Robots are usually open-loop mechanisms with multiple degrees of freedom (DOF). In general they have actuators at each joint, with a single DOF at the joint positions. Whilst having good operating characteristics, such as large workspace, high flexibility and manoeuvrability, typical manipulators have the disadvantages of low precision, low stiffness and low power. Additionally they are generally operated at low speeds to avoid excessive vibration and deflection. Also, the presence of gearboxes in manipulators of this sort limits their performance due to friction effects and backlash inherent in gear trains. To overcome these drawbacks the principal research efforts in the field have been centred on the development of direct-drive robots using light, strong and stiff link materials, together with small and light actuators of large power capabilities.

In recent years improvements in the performance of robots have been achieved by development of the mechanisms themselves. Considerable research (see for instance [4-15]) has been published, especially on alternatives to open-loop robots, that is, parallel robots which consist mainly of closed-loop mechanisms where the links are arranged in parallel rather than in series. They generally offer a much higher rigidity and improved accuracy because the joint errors do not accumulate as they do in serial robots. Additionally, the actuators are all mounted on the supporting frame so that they do not form a part of the dynamic load as they frequently do in serial designs. The reduced load gives a smaller mobile mass resulting in potentially faster dynamic behaviour. The manipulator can also be driven directly by actuators, usually in the form of electric motors. The reduction of friction and the elimination of gearboxes, and hence backlash, allow for fast, potentially high precision, manipulators.

In the literature it can be found that the performance of parallel robots can be further improved by using actuation redundancy (see [16-17]). A robot with actuation redundancy has more actuators than are actually needed. Therefore, the selection of actuator torques to track a prescribed trajectory becomes indeterminate. This can, however, allow the optimisation of the torque distribution among the actuators according to some chosen performance criterion.

Despite the advantages of parallel robots extensive research in the area has led to the conclusion that such manipulators are problematic due to issues of poor workspace, many singular configurations, and complex kinematic and dynamic characteristics. Because of these issues research in the area of parallel robots has been distributed between kinematic optimisation and analysis (see for example [18-25]), workspace analysis [26-27], dynamic analysis [28-32] and design analysis [33]. It is also the case that much effort has been devoted to the analysis of manipulator singularities (see for instance [34-38]) leading to a prevailing opinion which appears to support the idea of singularity avoidance in both path planning and control, as well as in mechanical design (see [39-40]).

Due to the desirable characteristics of parallel manipulators, yet the problems associated with them such as poor workspace, multiple singular configurations, and complex kinematic and dynamic characteristics, the need for practical solutions to these drawbacks is vital for the exploitation of the performance benefits. In this context the research in this thesis is directed towards the development of new techniques for enhancing the performance of planar parallel robots. To meet this main aim several investigations and developments have been carried out. Before stating the nature of these developments and investigations previous work in the area of parallel robots is highlighted in the next section.

1.3 Literature Review

1.3.1 Robot Mechanism Configurations

From the kinematic point of view robotic manipulators can be classified into two types, those known as “ Serial Robots” and those defined as “Parallel Robots”. In the former the links are arranged in a consecutive manner where one actuator is located on each link to define one DOF. Figure 1.4 shows one typical example of this sort of manipulator. Within this family of robots each link must carry the subsequent links and their associated actuators. As a result large link sizes and high motor torque capabilities are necessarily required and because of their cantilever design serial robots suffer from the drawbacks mentioned earlier. Despite these drawbacks these manipulators do display many useful and very important characteristics which are required for any robot, such as:

- Large workspace capability.
- High manoeuvrability.

In comparison with serial mechanisms parallel robots are closed-loop mechanisms in which the end effector is connected to the robot base by a number of independent chains (this number is equal to the number of DOF of the robot) which have at most two links and are actuated by a unique rotary or prismatic actuator mounted on the base. It is important to refer to the fact that the word independent here means that no segment of a chain can be linked to another chain. This structural propriety results in manipulators of compact sizes and possessing multiple degrees of freedom. Typical planar and spatial examples of these manipulators, with different DOF, are shown in Figures 1.5-1.8. Due to the fact that the actuators are all mounted on the ground, as seen in the Figures, these manipulators have very small mobile masses meaning that their dynamics are potentially very fast. Because the position and orientation of the end-effector are decided by each of the independent chains these structures do not have the problem of accumulation of actuator errors as is the case for serial robots. Therefore they can position very precisely. In many cases parallel robots overcome the defects of serial robots but suffer from certain defects that can be stated as follows:

- Small working space.
- Multiple singular configurations.
- Low manoeuvrability.

To overcome the first and third of these defects research in the area of parallel robots has been focused on the development of new designs that merge the properties of serial robots into those of parallel robots [15] leading to what are known as "Hybrid systems". The design of such systems is based on *Redundancy* in both the actuation and the kinematics. Despite significant improvements in workspace and manoeuvrability these systems can become very complicated from the point of view of control and analysis. Therefore this technique has drawbacks because it loses some of the advantages of the original parallel system, mainly because the resulting system is effectively a serial robot. Figure 1.9 shows a typical hybrid manipulator driven by hydraulic actuators. The first nine DOF of this robot manipulator result from three linearly actuated parallel robots each of 3-DOF and connected in series. In order to enlarge the workspace a sliding pair has been added at the beginning of the mechanism, and this forms the remaining DOF.

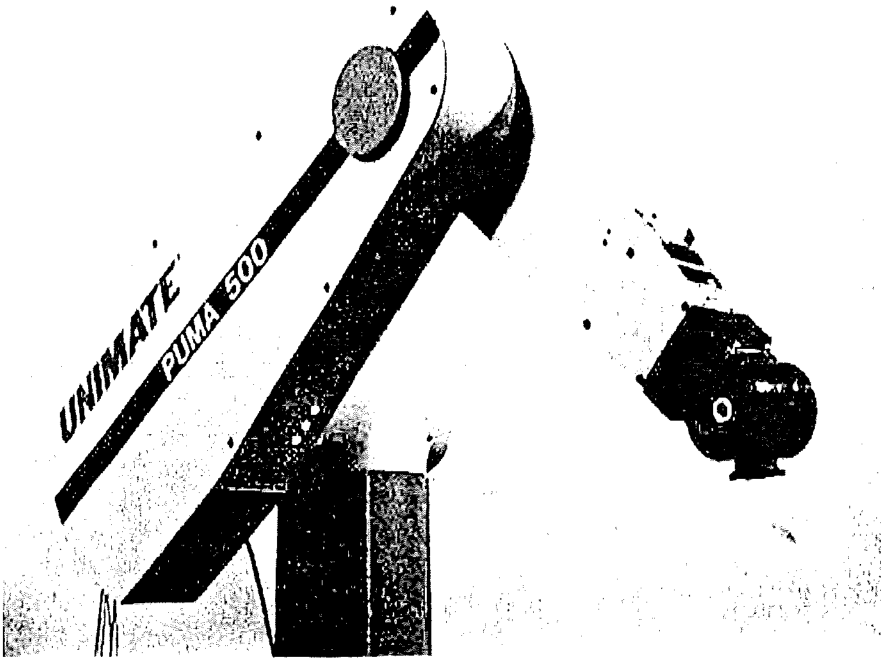


Fig. 1.4: A typical serial robot (Unimation PUMA 560, [41]).

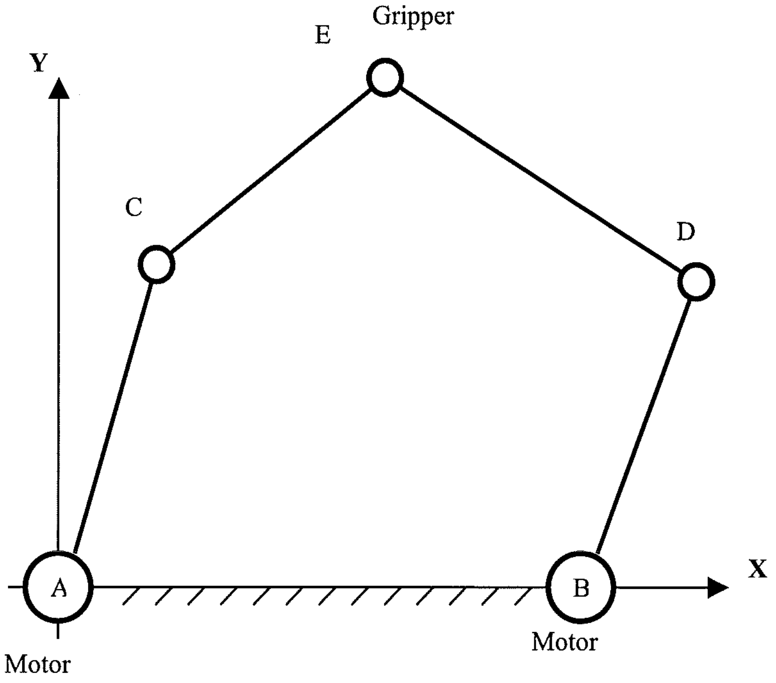


Fig. 1.5: A typical 2 DOF planar parallel manipulator (five-bar mechanism).

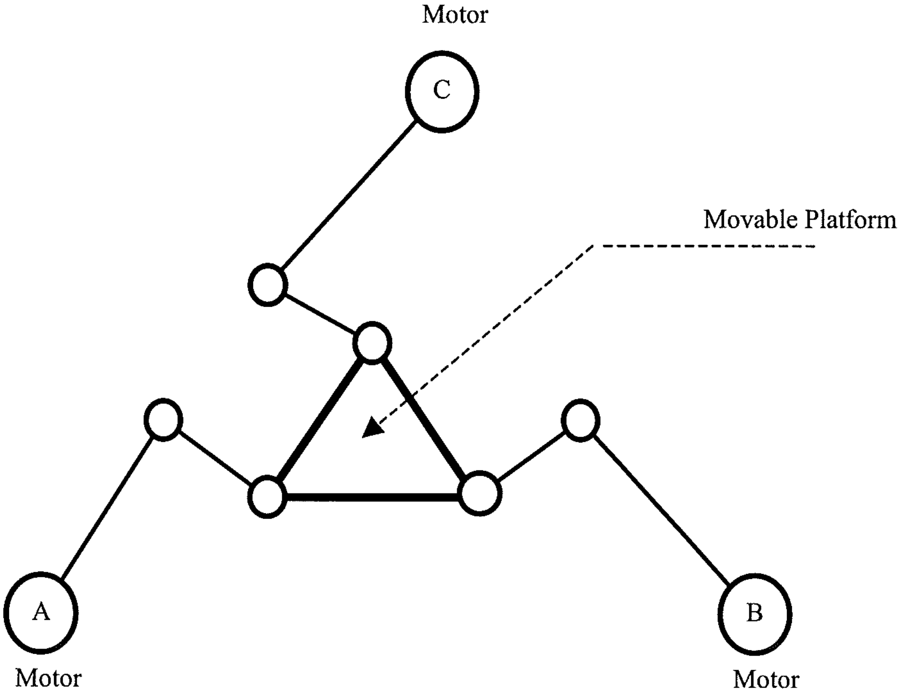


Fig. 1.6: A typical 3 DOF planar parallel manipulator.

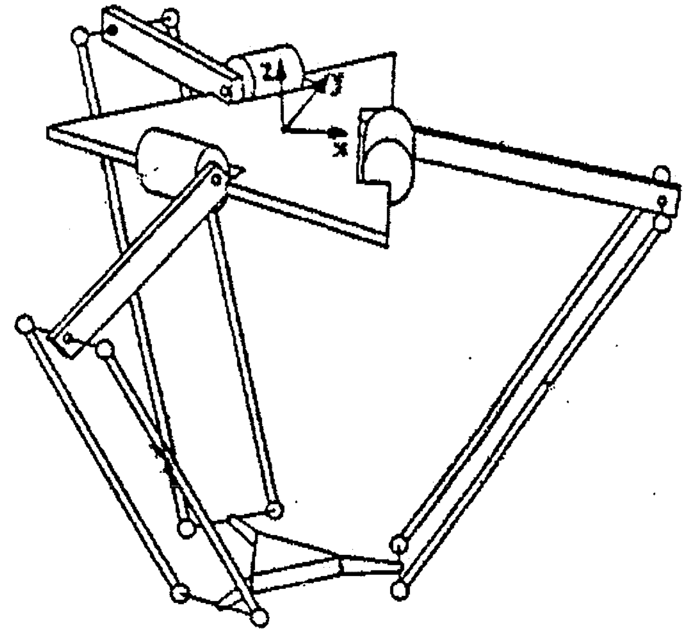


Fig. 1.7: The DELTA-580 manipulator, a 3 DOF direct-drive parallel robot [10].

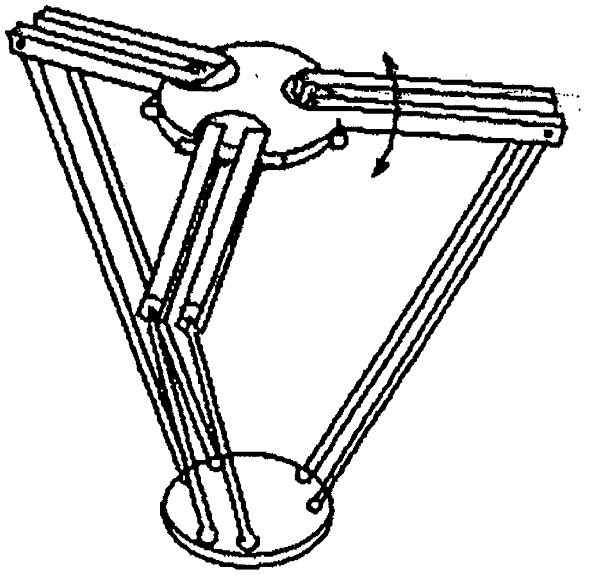


Fig. 1.8: The Hexa manipulator, a 6-DOF parallel robot [36].

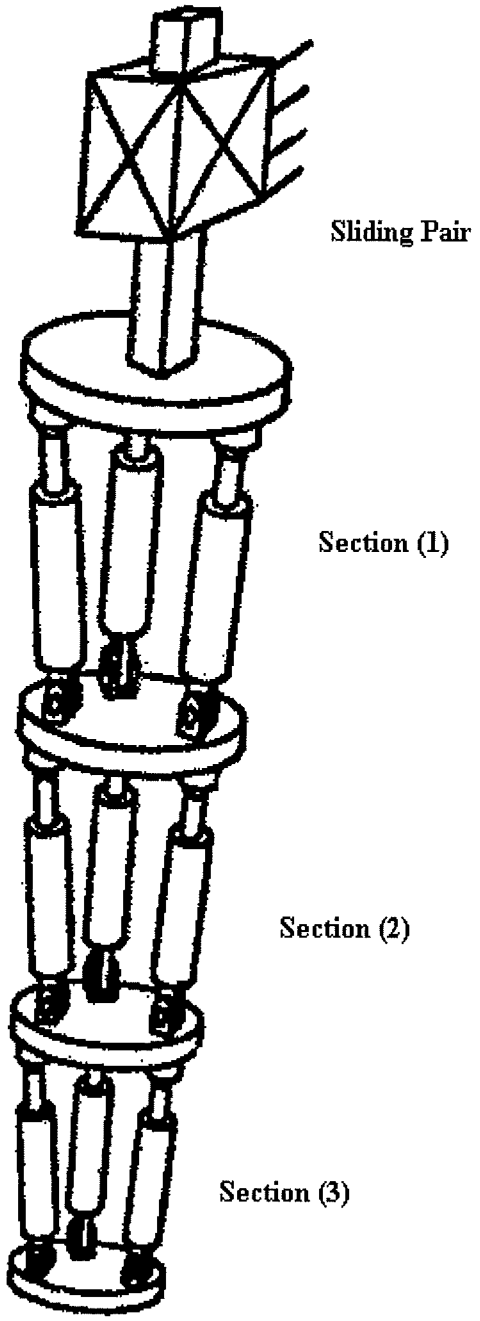


Fig. 1.9: A ten DOF hybrid serial-parallel robot [15].

1.3.2 Singularity

Singularity in parallel robots is an important issue that should be analysed irrespective of the manipulator design. Unlike in serial robots singularity in parallel robots is a structural property. The presence of singularity inevitably leads to loss of manipulator control. The analysis of singularity leads to two important classes[1] [34] [36]. These are:

- Parallel singularities.
- Serial singularities.

From the literature [34] [36] when the manipulator is in a serial singularity there is a direction along which no Cartesian velocity can be produced. The gripper loses one or more DOF and can resist one or more forces or moments without exerting any torque or force at the actuated joints. On the other hand when the manipulator is in a parallel singularity the manipulator becomes uncontrollable (the gripper of the manipulator gains one or more DOF and it cannot resist the forces or moments from one or more directions even when all actuated joints are locked). Figures 1.10 and 1.11 show these two singular configurations for a five-bar mechanism (a typical planar 2-DOF parallel robot).

There is, in fact, another case of singularity which occurs only when the parameters of the manipulator satisfy certain special conditions. In this case a finite motion of the gripper is possible, even if the actuated joints are locked, or alternatively a finite motion of the actuated joints produces no motion of the gripper [34].

The differentiation between the various types of singularities is mainly dependent on two important matrices from the velocity equations of the manipulator. The detailed procedure for obtaining these matrices, and the conditions at which singularities can occur is outlined in chapter 5. In fact, the prevailing research opinion appears to support the idea of singularity avoidance in both path planning and control, as well as in mechanical design.

[1] In the context of this research examples of singularity are here restricted to 2-DOF planar mechanisms.

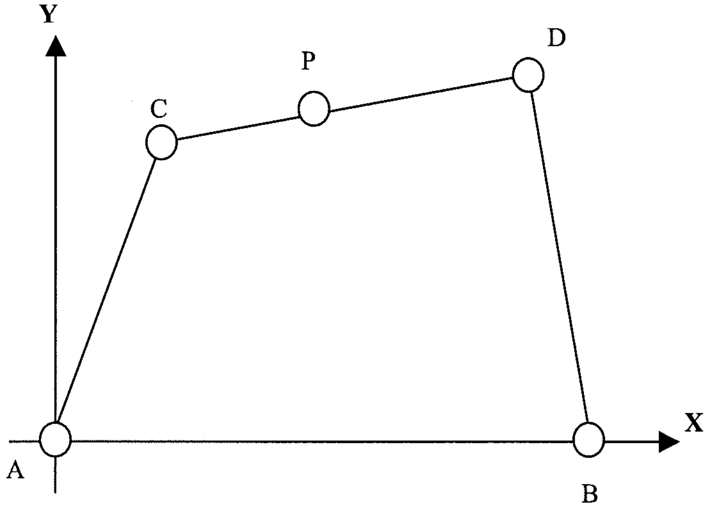


Fig. 1.10: Example of parallel singularity.

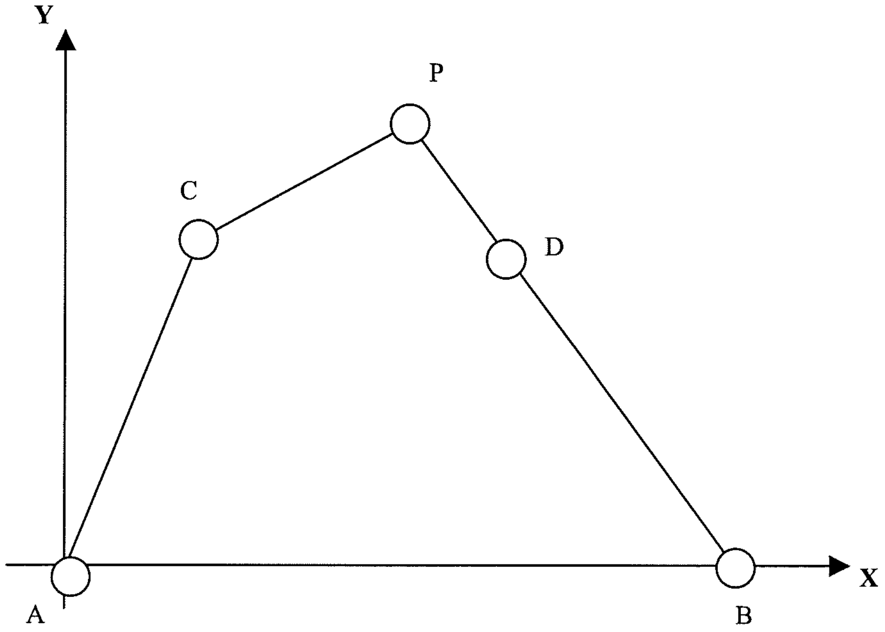


Fig. 1.11: Example of serial singularity.

1.3.3 Redundancy in Parallel Robots

Redundancy can improve the abilities and performance of parallel robots and can be applied in the following ways:

- Kinematic redundancy.
- Sensing redundancy.
- Actuation redundancy.

The first allows for the minimisation of joint speeds. Sensing redundancy can help in avoiding uncertainties in the direct kinematics or by reducing the computational effort required by adding information in order to reduce the computation time within the control loop. Finally, a manipulator with actuation redundancy has more actuators than the number of degrees of freedom that are required to define its dynamics, however this makes the selection of the motor torques (in order to follow a desired path) indeterminate. Under certain circumstances this can lead to an optimisation of the torque distribution among the actuators according to some pre-specified performance criterion. Different co-ordination schemes for actuation redundancy have been proposed, for example Nakamura and Ghodoussi [42] represented the unactuated joint angles as functions of the actuated joints and used the Jacobian matrices to parameterise the actuation redundancy. Another example is the case of Beiner [16] who transformed the robot's differential equations into algebraic relations in order to cast the problem into a standard *constrained optimisation* form.

Kock [17] removed the rotational degree of freedom of the movable platform of a 3-DOF planar manipulator and instead considered the system as a redundant 2-DOF manipulator. In this case both the kinematic and the actuation redundancy schemes were applied by adding a redundant branch and a redundant motor to the normal five-bar mechanism. In many cases the location of the redundant actuator appears to be a personal choice and there is no obviously optimum place for it to be. In this context Beiner [16] recommended the optimisation of the location of the redundant actuator in order to enhance its advantage in dynamic applications.

1.3.4 Modelling of Parallel Robots

The derivation of dynamic equations of motion for mechanical systems can be achieved using the classical concepts of vectorial mechanics [43], Newton-Euler [44], and the energy based construct of Lagrange-Euler [45] and Lagrange [46-47]. In addition to this other energy and power flow methods like the Bond-Graph approach can be used to formulate the dynamic equations [48-53]. In fact all of these methods can be applied directly to derive the equations of motion for serial manipulators without any conceptual difficulty. The choice of such methods is down to the modeller and the specific reason for which the equation of motion is derived.

For parallel robotic systems, containing closed chains in their design, the situation is different. This is because of the holonomic constraints involved in the kinematic structures of these manipulators, meaning that the previously mentioned modelling techniques are not directly applicable. One classical method for dealing with constraint cases of this sort is to introduce Lagrange Multipliers, λ, into Lagrange's equations [54-55]. The disadvantage of this technique is the difficulty in finding appropriate recursive computational schemes, likes those proposed for serial manipulators.

An alternative modelling approach for systems with closed chains was explored in the literature [42][56-59]. This approach solved the problem by reducing the closed chain to a tree structure mechanism consisting of multi-serial chains on the assumption of virtual actuation. This approach [42] can be summarised as follow:

1. A closed-link mechanism is transformed into an open-loop structure by virtually cutting the workless joint that closes the loop. This is on the assumption that virtual actuators drive the unactuated joints of the open-loop structure. This procedure directly corresponds to removing the holonomic constraints that close the loops in the manipulator.

2. The serial reduced system can then be dynamically modelled using any of the standard recursive formulations.

3. The torques of the actuated joints of the original closed mechanism can subsequently be computed from joint torques of the tree structure, by taking into consideration the constraints of the closed loop.

A complete step-by-step formulation of the dynamics of the basic cell of the manipulator under investigation using the above technique can be found in Appendix (A). This was done primarily to check the validity of the results obtained by the bond graph technique. It is important to refer to the fact that Lagrange's formulation is the most common modeling technique in references [42][56-59] and the examples presented were planar mechanisms.

Despite the above complications dynamic modelling of closed kinematic chains can be achieved by means of the standard Lagrangian technique after solving the kinematic constraints, and thus by reducing the number of position variables to the actual number of degrees of freedom. However this results in complex algebraic computations whilst deriving the equations of motion.

The bond-graph methodology is an attractive technique for modelling physical dynamic systems with the advantage that it can deal with many different application domains. The bond graph theory provides a graphical representation of power flow in dynamic systems wherein power flow is expressed by the product of two generalised variables; e (effort) and f (flow). The disadvantage of the method is that it needs a computer to handle the equations generated from the graph of the models. This is due to the large number of equations generally extracted from the model's graph. But the availability of computer software such as CAMP-G [60] and MTT [61-62] assists in this process, so that bond-graphs can be used practically for the dynamic modelling of physical systems.

Modelling of mechanical systems by means of bond-graphs begins with the derivation of positional equations dependent on the kinematic structure of the system and then after differentiation these are transformed to the velocity equations defining the system. The bond graph of the system is constructed from the system velocities. In constrained mechanical systems the derived velocities are not independent which gives rise to a set of inertia derivative causalities in the bond graph model. A bond graph with derivative causality implies the existence of an algebraic loop: the system dynamics are described by a set of first order differential equations together with a set of algebraic equations. This problem must be alleviated as the process of deriving the dynamic equations become cumbersome.

In order to solve the derivative causality problem Karnopp and Margolis [63] introduced a solution through which the derivative causality effect can be removed based on the addition of fictitious compliance elements (C-elements) to the appropriate junctions in the bond graph model. The addition of these elements assumes mechanical components of high stiffness. The method is advantageous because it not only eliminates the derivative causality problem but can also describe the joint forces, and can be used to model joint clearances [64]. The disadvantage of this method lies in the fact that the system states are increased to more than the actual number required. Zeid [64] proved that the Karnopp-Margolis method produces a formulation that is equivalent to the classical Lagrange multipliers method for the modelling of planar mechanisms. He also demonstrated that the use of R-elements (dampers) instead of the C-elements could eliminate derivative causality without increasing the number of states of the system. The formulation using these damping elements is still equivalent to the method of Lagrange multipliers. Khulif [65] removed the derivative causalities by adding artificial effort sources instead. The procedure to establish this method can be obtained from this reference.

For holonomic constrained mechanical systems Redfield [66] formulated bond graphs based on conservation of energy principles and generalized coordinates. This bond graph is a representation of the standard Lagrange equation of motion. Although the method assists in avoiding the derivative causality problem it is not practically useful because the modeller has to write the kinetic co-energy and potential energy in terms of the generalised coordinates, take the appropriate partial derivatives and then construct the bond graph model. The major disadvantage of the method lies in the fact that all constraint forces are lost and therefore other design variables must be reconstructed using free body diagrams and Newton's laws [67].

Despite the aforementioned and the recent efforts in the field of bond graphs (see for example [68-70]) it seems that there is no efficient, established, technique for modelling mechanical systems. The area is still open for new techniques that make the bond graphs modelling process more efficient, easy and systematic.

1.3.5 The Control Problem of Parallel Robots

Parallel robots have the merit of being able to move at high speed, this being enhanced by the use of direct drive motors. Change of inertia and the effect of centrifugal and Coriolis forces in robot dynamics cannot be ignored in high-speed applications and must therefore be considered in any control implementation. The computed torque[2] method ([44] and [71-73]) has been proposed for conventional serial link manipulators on the basis that nonlinearity can be cancelled out by the model itself and the resulting linear system together with the disturbances can be controlled by a simple *PD* or *PID* controller. The method needs the model to be of very high quality, and this is frequently very difficult to achieve because of the uncertainty in the model parameters.

The same idea can be applied to the control of parallel robots. However in addition to the above restriction solutions of the inverse dynamics of parallel mechanisms require considerably more computation than those for serial systems. The key to practical application is how fast the inverse dynamics can be solved.

The feed-forward method is another possible control technique that can be used in the control of parallel robots. It differs from the computed torque method in the way that the model does not interact with the feed back loop. For the sake of this research the two methods will be reviewed.

[2] Sometimes it is called Feed Back Linearisation.

- **Feed-Forward Controller**

The feed-forward controller is the simplest form of model-based control system. It uses the inverse dynamics of the robot, R^{-1}, to calculate the required torque, τ, to perform a prescribed joint trajectory ($\Theta_d, \dot{\Theta}_d, \ddot{\Theta}_d$). For controlling the uncertainty in parameter estimations and dealing with unknown dynamics a *PD* controller is used. The schematic diagram of the controller is shown in Figure 1.12. It is clear that the model-based portion is outside the servo-loop. This in fact can solve the inverse dynamics computational problem associated with parallel robots because the torques from the model can be computed off-line and then be read from memory at run time. Unfortunately the method does not provide complete decoupling. This is due to the fact that corrective torques in the servo-loop of one joint will affect all the other joints.

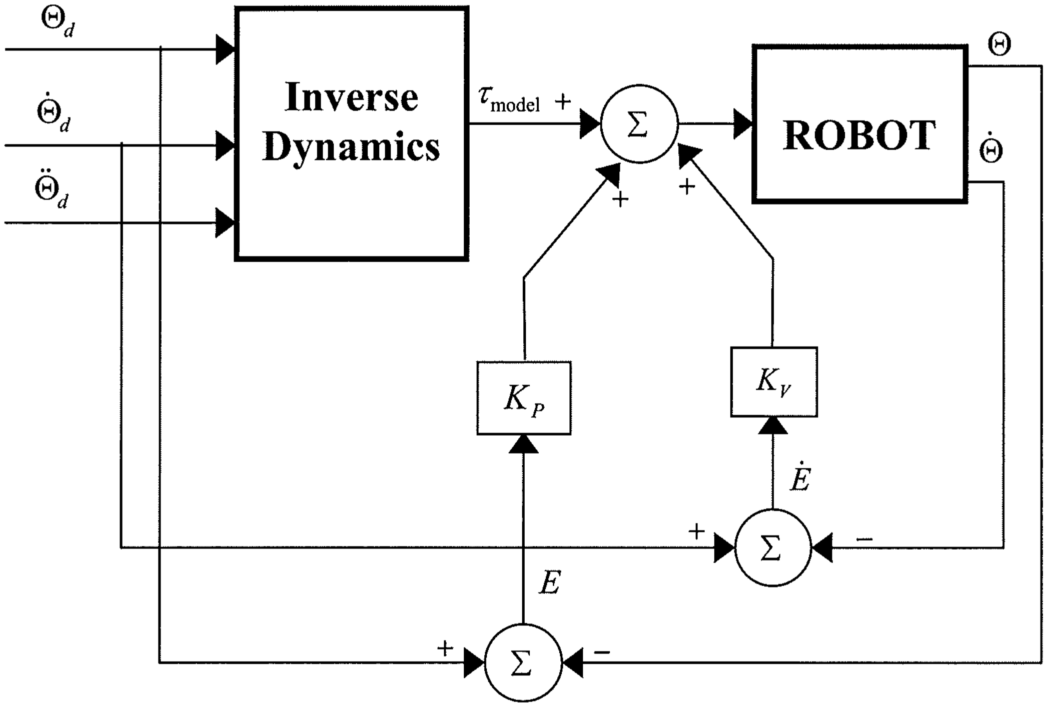


Fig. 1.12: Feed-forward control system for robot manipulators.

The following equation defines the control law,

$$\tau = \tau_{\text{model}} + K_P E + K_V \dot{E} \qquad (1.1)$$

Where E is the vector of servo errors. The terms K_P, K_V are the proportional and the derivative gain matrices of the servo-loop, respectively. Their values are chosen to satisfy stability conditions.

- **Computed Torque Controller**

The computed torque controller differs from the feed-forward controller in the way that the model-based portion interacts with the servo-loop to allow complete decoupling. The schematic representation of this controller is shown in Figure 1.13.

A realistic implementation of the computed-torque method must rely on *estimates* of the manipulator parameters rather than the exact values. Let $\tilde{M}(\Theta), \tilde{V}(\Theta,\dot{\Theta}), \tilde{G}(\Theta)$ and $\tilde{F}(\dot{\Theta})$ denote the estimated values of the manipulator inertia tensor, Coriolis and centrifugal force vector, gravity-loading vector, and frictional force vector, respectively. That is,

$$M(\Theta) = \tilde{M}(\Theta) + \Delta M(\Theta) \qquad (1.2)$$

$$V(\Theta,\dot{\Theta}) = \tilde{V}(\Theta) + \Delta V(\Theta) \qquad (1.3)$$

$$G(\Theta) = \tilde{G}(\Theta) + \Delta G(\Theta) \qquad (1.4)$$

$$F(\dot{\Theta}) = \tilde{F}(\dot{\Theta}) + \Delta F(\dot{\Theta}) \qquad (1.5)$$

The terms $\Delta M(\Theta), \Delta V(\Theta), \Delta G(\Theta)$ and $\Delta F(\dot{\Theta})$ denote the vectors of the *errors* in the estimates of the arm parameters. In order to apply the computed torque controller and therefore cancel the nonlinear dynamics of the manipulator that is defined by the following equation:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \qquad (1.6)$$

The following state feedback control law can be used,

$$\tau = \tilde{M}(\Theta)\left\{\ddot{\Theta}_d + K_V\,\dot{E} + K_P\,E\right\} + \tilde{V}(\Theta,\dot{\Theta}) + \tilde{G}(\Theta) + \tilde{F}(\dot{\Theta}) \qquad (1.7)$$

Substituting from equations 1.2-1.5 into equation 1.6 and then equating to equations 1.7 gives,

$$\ddot{E} + \left\{I - M^{-1}(\Theta)\Delta M(\Theta)\right\}\left(K_P\,E + K_V\,\dot{E}\right)$$
$$- M^{-1}(\Theta)\left\{\Delta V(\Theta,\dot{\Theta}) + \Delta G(\Theta) + \Delta F(\dot{\Theta}) + \Delta M(\Theta)\ddot{\Theta}_d\right\} = 0 \qquad (1.8)$$

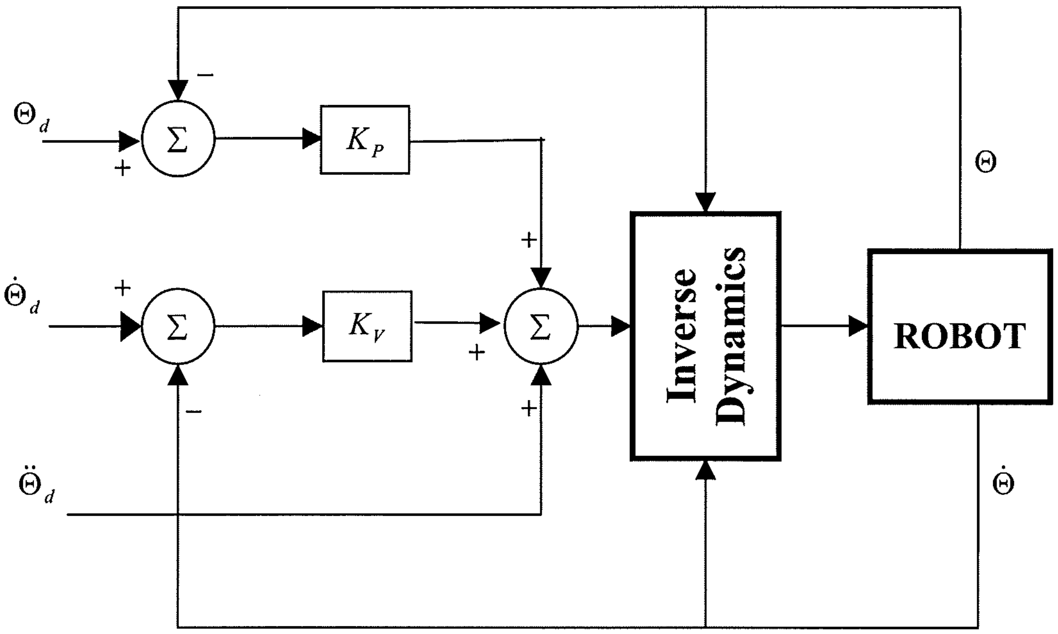


Fig. 1.13: Schematic of the computed torque Controller.

If the model is exact, equation 1.8 is reduced to the following linear second order equation that is *independent* of the arm parameters:

$$\ddot{E} + K_V \dot{E} + K_P E = 0 \qquad (1.9)$$

If the gain matrices, that is K_V and K_P, are diagonal then the closed loop equations of motion (equation 1.9) are not only linear but also *uncoupled*. The characteristic roots of equation 1.9 can be assigned to have negative real parts through the appropriate selection of the gains K_V and K_P thereby the error will approach zero asymptotically.

The major problem of the computed torque method lies in the fact that the values of the parameters in the model are often not known exactly for reasons that will be discussed in chapter 6. However, because in many applications the mass properties of the objects that the manipulator picks up are not generally known it is impossible to maintain an accurate dynamic model.

Despite the advances in control systems theory the *PID* controller is still the most commonly used controller for commercial robots. The popularity and the widespread use of this controller can be attributed to its simplicity and its generally acceptable performance characteristics. The *I* term ensures robust steady-state tracking of the step commands while the *P* and *D* terms provide desirable transient response and stability, respectively.

Recently, a new class of *PID* controller has been discussed in the literature [74-76], consisting of a nonlinear gain in cascade with a linear fixed-gain PID controller. This structure enables the controller to adapt its response based on the performance of the closed-loop control system. When the error (e) between the desired and the actual values of the controlled variable is large, the gain amplifies the error substantially to generate a large corrective action to drive the output of the system to its goal rapidly. As the error diminishes, the gain is reduced automatically to prevent large overshoot in the system response. The block diagram representation of this controller is shown in Figure 1.14. The nonlinear gain can represent any nonlinear even function of the error that is bounded in the sector $0 < K < K_{\max}$. Two partially successful functions have been proposed [74]. These are the *Sigmoidal* and the *Hyperbolic* functions and are defined as follows:

- Sigmoidal function

$$K = K_o + K_1 \left\{ \frac{2}{1 + \exp(-K_2 e)} - 1 \right\} \qquad (1.3)$$

- Hyperbolic function

$$K = K_o + K_1 \left\{ 1 - \frac{2}{\exp(K_2 e) + \exp(-K_2 e)} \right\} \qquad (1.4)$$

Where K_o, K_1 and K_2 are user-defined, positive, constants.

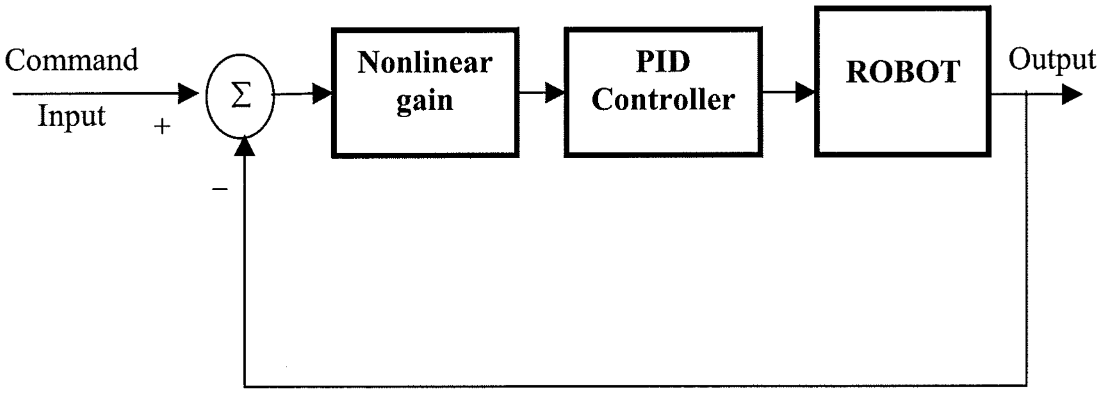


Fig. 1.14: General structure of a Nonlinear PID controller [74].

1.4 Organisation of the Thesis

The layout of the thesis is as follows:

- **Chapter 2**

In this chapter the basic concepts of bond graphs are outlined. After this the dynamics of the main cell of the parallel robot are formulated by using the standard

B*ond-Graph* methodology. The *Model Transformation Tools (MTT) software* is used to perform this task. In addition to the standard bond graphs, *MTT* contains a powerful attractive technique for modelling large-scale dynamic systems called the *Hierarchical Bond Graph.* A *general method* is discussed for modelling the parallel robot by means of the Hierarchical Bond-Graph.

- **Chapter 3**

In this chapter the novel design concept of the *Multi-cell Parallel Planar manipulator* is introduced. Application of this concept to a specific 2 DOF planar parallel manipulator is undertaken leading to the development of a general dynamic model for this multi-cell manipulator, for any configuration, using the Lagrange method. In addition to this *static balancing* of the manipulator is addressed and a proposal for balancing is introduced.

- **Chapter 4**

In chapter 4 the *optimisation* of the location of the *redundant* motor in the manipulator is investigated. This is achieved by the dynamic analysis of a one-cell manipulator using different motor locations. Modelling of the *redundant actuated manipulator* is performed using the *Hierarchical Bond Graph Method.*

- **Chapter 5**

In this chapter, a novel design for a one-cell manipulator is developed using a new *Singularity based Design* technique developed by the author of the thesis. A complete dynamical analysis of the manipulator is performed through simulation. It is important to note that the idea is generalised and leads naturally to the design of further novel manipulators, defined here by the term *Parallel-Parallel manipulators.*

- **Chapter 6**

This chapter gives details of the experimental system and the experimental rig required to study the implementation issues that arise for a planar parallel manipulator of this sort. In addition to this revision of the theoretical dynamic modes is performed in order to match the actual dynamics of the arm. The revised dynamic model is then subjected to an evaluation process achieved by comparing data collected from the experimental robot with that obtained by simulations using the dynamic model.

- **Chapter 7**

In chapter 7 investigations are carried out into the effectiveness of the feed-forward and the computed torque controllers in controlling the experimental manipulator.

- **Chapter 8**

In this chapter a relationship between the computed torque controller and the well-known *sliding mode controller* (*SMC*) is established using the *reaching law method.* After that a new design for a computed torque controller is constructed followed by theoretical and experimental evaluation of the system performance using this controller.

- **Chapter 9**

In chapter 9 investigations are carried out into the effectiveness of some nonlinear *PD* and *PID* controllers in controlling the experimental robot manipulator.

- **Chapter 10**

In this chapter a new methodology for the design of *PD* controllers is introduced followed by analysis of the system stability using the Lyapunov stability criterion. After that theoretical and experimental evaluation of the system performance under this kind of control is carried out.

- **Chapter 11**

In this final chapter the thesis is concluded and suggestions for further research are presented.

Chapter 2

Bond Graph Modelling of a 2 DOF Planar Parallel Manipulator Using the Model Transformation Tools Software

2.1 Introduction

The bond graph is a graphical construct in which all the characteristics of a physical system can be defined. In the method the physical system is divided into components, and each component has its own physical law (the so-called constitutive relationship). Connections between the components are based on the structure of the physical system. These connections are represented on the model graph by individual half arrows (*bonds*) which indicate the direction of power flow between the components. Each bond has two variables associated with it based on power; thus the two variables attached to each bond in a bond graph are the *effort* (*e*) and the *flow* (*f*). The multiplication of these two variables gives the amount of power that is carried by the bond. The effort and flow variables can be represented in any physical domain, for example the effort and flow can represent the voltage and the current in an electrical system, or the force and velocity in a mechanical system. This is, in fact, one of the advantages of using a bond graph in modelling dynamic systems within different domains. In addition to this bond graphs also highlight the structure of the system, giving a clear understanding of the interactions between the system components.

The first part of the chapter explores the definitions of the basic bond graph elements and components. Following on from this the Model Transformation Tools (MTT) [62] software is presented. After that the bond graph method is used to model a specific 2-DOF parallel manipulator using the MTT program.

2.2 Elements of Bond Graphs

2.2.1 Bonds

The primary element in any bond graph is the energy bond. It is represented by a half arrow. Each arrow carries two variables, the *effort* and the *flow*. The direction of the arrow refers to the positive direction of the flow of power. The form of the bond is seen in Figure 2.1. The flow variable is conventionally associated with the side of the bond that has the half arrow.

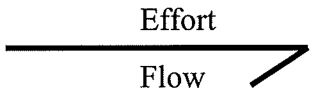


Fig. 2.1: Representation of an energy bond.

2.2.2 Variables

The meaning of the effort and flow variables in different energy domains is shown in the table 2.1 below.

Domain	Mechanical	Electrical	Hydraulic	Thermal
Effort	Force / Torque	Voltage	Pressure	Temperature
Flow	Velocity / Angular velocity	Current	Flow rate	Entropy flow rate

Table (2.1): Effort and flow variables in different energy domains.

2.2.3 Junctions

There are two types of junctions in bond graphs. The first type is the '*0-junction*' which allows the flow variables to be added algebraically. The other one is the '*1- junction*' and this allows the addition of effort variables. These junctions are generalisations of series and parallel connections in electrical systems. Table 2.2 shows the two junctions and the mathematical equations associated with them.

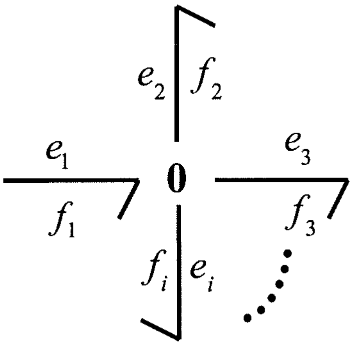
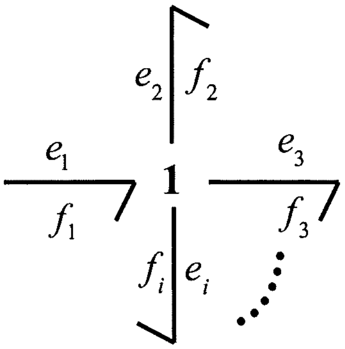
 e_2 f_2 e_1 f_1 0 e_3 f_3 f_i e_i $e_1 = e_2 = e_3 = ... = e_i$ $f_1 = f_2 + f_3 + ... + f_i$	 e_2 f_2 e_1 f_1 1 e_3 f_3 f_i e_i $e_1 = e_2 + e_3 + ... + e_i$ $f_1 = f_2 = f_3 = ... = f_i$

Table 2.2: Bond graph junctions.

2.2.4 Components

There are two basic components in bond graphs namely *One-Port* components, in which each component is connected to only one energy bond, and *Two-Port* components which are each connected to two bonds.

2.2.4.1 One-Port Components

The four main *one-port* bond graph components are the *R*-, *C*-, *I*- and *S*-components. The physical system components that are represented by these one-ports in different energy domains can be shown in Table 2.3. The physical laws governing each component determine the relationships between the power-variables associated with the bond connected to it. The physical laws that govern typical components are shown in Table 2.4, noting that linear behaviour is assumed throughout.

Component	Domain Mechanical	Electrical	Hydraulic
R	Damping (Transitional or rotational)	Resistance (Ohmic / Thermal)	Flow resistance
C	Compliance (Transitional or rotational)	Capacity	Compressibility
I	Mass / Inertia	Inductance	Fluid inertia
S	Force / Torque Velocity / Angular velocity	Voltage source / Current source	Pressure source / Flow source

Table 2.3: Physical representations of one-port components.

2.2.4.2 Two-Port Components

The only practical *two-port* components in bond graph representations are *Transformers* and *Gyrators*. They represent the physical effects of amplification or attenuation of the effort applied to them. The advantage of these two-port components is their ability to represent connections between different energy domains. As the 0- and 1-

junctions represent addition, the two-port components represent the process of multiplication. The *symbols* and the *constitutive relationships* of both the transformer and the gyrator are shown in table 2.5.

Component	Value	Symbol	Constitutive relationship
R	r	⇁ R : r	$e = f \cdot r$ or, $f = \frac{e}{r}$
C	c	⇁ C : c	$e = \frac{1}{c}\int f \cdot dt$ or, $f = c\frac{de}{dt}$
I	m	⇁ I : m	$e = m\frac{df}{dt}$ or, $f = \frac{1}{m}\int e \cdot dt$
S	-	S_f ├⇁ S_e ⇁|	$f = constant$ (Flow source) $e = constant$ (Effort source)

Table 2.4: The one-port components and their features.

Component	Gain	Symbol	Constitutive relationship
Transformer **(TF)**	k	⇁ TF : k ⇁	$e_2 = k\, e_1$ and $f_2 = \frac{1}{k} f_1$
Gyrator **(GY)**	n	⇁ GY : n ⇁	$e_2 = n\, f_1$ and $f_2 = \frac{1}{n} e_1$

Table 2.5: The two-port components and their features.

2.2.4.3 Modulated Two-Port Components

If the variables n and k in the two-port components are not constant the definition is revised to *modulated* two-port components. The modulated component has three bonds associated with it. Two of these bonds are the original energy bonds of the normal components whilst the third bond carries the modifying signal and is not an energy bond. So, there is no energy transfer through this signal. This bond is commonly called an *Activated bond.*

2.2.5 Causality

The way in which inputs and outputs are specified in bond graphs is by using *causal strokes*. The causal stroke is a short, perpendicular, line at the tail or head end of the energy bond. It indicates the direction in which the effort signal is applied. It is commonly known that the end of the bond which displays the causal stroke is the end towards which the effort is applied and that the flow is directed to the other end. One of the main features of modelling with bond graphs is that the causality is not specified when the model is being developed. The causality is specified only for the source components and it subsequently propagates through the graph. Figure 2.2 shows the directions of the effort and flow in a bond with different causal stroke locations.

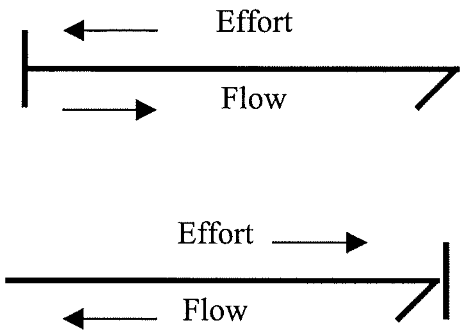


Fig. 2.2: The meaning of a causal stroke.

2.3 Model Transformation Tools (MTT)

The *MTT* program is a computer software which has the ability to take bond graph models in some graphical format and transform them to a set of useful representations such as *differential algebraic equations*, *state space equations*, *transfer functions* and *simulation codes*. The transformations in MTT can be accomplished by using software tools such as *REDUCE*, *Matlab*, *OCTAVE*, and etc., usually encapsulated in *UNIX* shell scripts. The relationships between the tools themselves are encoded in a *Make* file. This allows the modeller to specify the representation that he needs, and then all the intermediate transformations are generated automatically.

In addition to the standard bond graph, MTT also accommodates the concept of the *hierarchical* bond graph via a library of sub-systems which work in many different energy domains. The connections between these sub-systems in MTT are accomplished by using special bonds called *Named SS Components*. Further bonds, called *SS* Components, represent the inputs and the outputs of the system model. More information about MTT can be found in the manual [62].

2.4 Bond graph dynamic modelling of 2-DOF Planar Parallel Manipulator using MTT

As previously mentioned, one of the associated problems of parallel robots is their complicated dynamics, even for simple structural systems. This puts certain restrictions on the application of model-based control strategies due to the scale of the real-time computations required for the inverse dynamics. To take full advantage of parallel robots and to minimise such difficulties the manipulator shown in Figure 2.3 was chosen to be the principle mechanism to be studied. This manipulator is a special case of the conventional five-bar mechanism and has a symmetric structure with simple constraints together with links of equal lengths. In addition it has been proven that the maximum workspace for a five-bar mechanism occurs only when it exhibits this specific design [4].

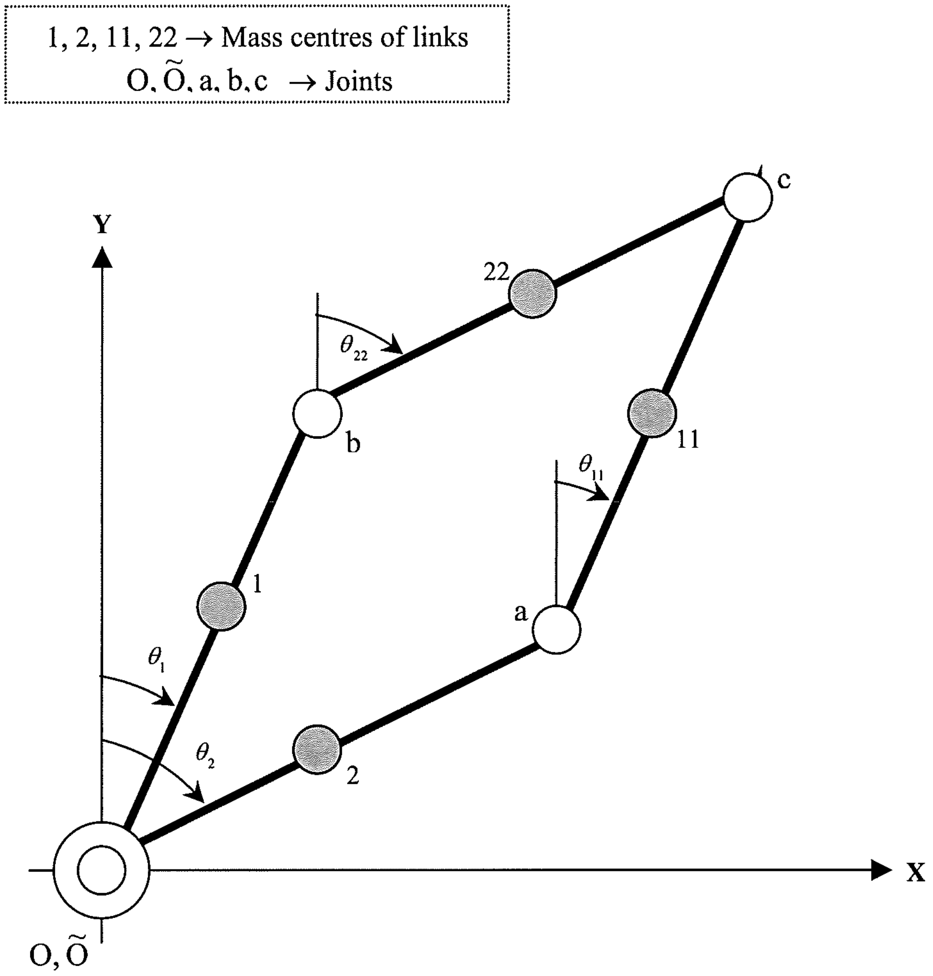


Fig. 2.3: Configuration of the, *2-DOF*, planar parallel manipulator.

The first step in the modelling process is to construct the bond graph of the manipulator. This is done by means of the following steps:

- **Assumptions**

Each link in the manipulator is assumed to be rigid with mass m, length $2L$ and polar moment of inertia about the centre of mass of the link $J = \frac{1}{3}mL^2$.

- **Solving the kinematic constraints of the mechanism**

From figure 2.3 one can deduce that:

$$\theta_1 = \theta_{11} \quad \text{and} \quad \theta_2 = \theta_{22} \qquad (2.1)$$

Differentiating with respect to time leads to,

$$\omega_1 = \omega_{11} \quad \text{and} \quad \omega_2 = \omega_{22} \qquad (2.2)$$

Where,

θ : Represents the angular displacement.

ω : Represents the angular velocity.

It is clear from Figure 2.3 that the angular displacements are measured from the Y-axis. This is due to the fact that the standard *ROD* component used in the hierarchical bond graph approach is based on this assumption. So, it is better to work to the same convention here also.

- **Defining three absolute velocities for each link**

Three absolute velocities for each link must be defined. These represent the X, Y Cartesian velocities and the angular velocity ω of the centre of mass of each link, assuming that OXY is representative of an inertial frame.

- **Link Ob (mass centre is point 1)**

$$\dot{x}_1 = \frac{d}{dt}(x_1) = \frac{d}{dt}(L\sin\theta_1) = L\omega_1\cos\theta_1 \qquad (2.3)$$

$$\dot{y}_1 = \frac{d}{dt}(y_1) = \frac{d}{dt}(L\cos\theta_1) = -L\omega_1\sin\theta_1 \qquad (2.4)$$

$$\omega_1 = \frac{d}{dt}(\theta_1) \qquad (2.5)$$

The above three velocities are represented on the bond graph for the manipulator, shown in Figure 2.4, by the three *1-junctions* x_1, y_1 and w1.

- **Link Oa (mass centre is point 2)**

Similar to the case of link Ob, the three velocities of link Oa are as follows:

$$\dot{x}_2 = \frac{d}{dt}(x_2) = \frac{d}{dt}(L\sin\theta_2) = L\omega_2\cos\theta_2 \qquad (2.6)$$

$$\dot{y}_2 = \frac{d}{dt}(y_2) = \frac{d}{dt}(L\cos\theta_2) = -L\omega_2\sin\theta_2 \qquad (2.7)$$

$$\omega_2 = \frac{d}{dt}(\theta_2) \qquad (2.8)$$

These are represented on the graph by the three 1-junctions x_2, y_2 and w2.

- **Link ac (mass centre is point 11)**

$$\dot{x}_{11} = \frac{d}{dt}(x_{11}) = \frac{d}{dt}(2L\sin\theta_2 + L\sin\theta_1) = 2L\omega_2\cos\theta_2 + L\omega_1\cos\theta_1 \qquad (2.9)$$

$$\dot{y}_{11} = \frac{d}{dt}(y_{11}) = \frac{d}{dt}(2L\cos\theta_2 + L\cos\theta_1) = -2L\omega_2\sin\theta_2 - L\omega_1\sin\theta_1 \qquad (2.10)$$

$$\omega_{11} = \omega_1 = \frac{d}{dt}(\theta_{11}) \qquad (2.11)$$

These absolute velocities are represented on the graph by the three *1-junctions* x_11, y_11 and w11.

- **Link bc (mass centre is point 22)**

Similar to the previous link, the three velocities of link bc are as follows:

$$\dot{x}_{22} = \frac{d}{dt}(x_{22}) = \frac{d}{dt}(2L\sin\theta_1 + L\sin\theta_2) = 2L\omega_1\cos\theta_1 + L\omega_2\cos\theta_2 \qquad (2.12)$$

$$\dot{y}_{22} = \frac{d}{dt}(y_{22}) = \frac{d}{dt}(2L\cos\theta_1 + L\cos\theta_2) = -2L\omega_1\sin\theta_1 - L\omega_2\sin\theta_2 \qquad (2.13)$$

$$\omega_{22} = \omega_2 = \frac{d}{dt}(\theta_{22}) \qquad (2.14)$$

These are represented by the three 1-junctions x_22, y_22 and w22

♦ **Model construction**

The bond graph model of the manipulator, Figure 2.4, is constructed from the kinematic equations 2.1-2.14 by using combinations of bonds, junctions, transformers, integrators and I-elements. On the graph each INTF component represents an integrator that is responsible for integrating the angular velocity attached to it, therefore producing the angular displacement (θ) that enters the transformer (EMTF) component via the *mod* port by using a direct bond. This displacement represents the modulation signal of the transformer whose gain, in this case, contains terms of either $Sin(\theta)$ or $Cos(\theta)$.

It is important to refer to the fact that the angular velocities of links Oa and Ob, for the sake of clarity, are repeated twice on the graph; th_1, w1 for link Ob and th_2, w2 for link Oa. These do not affect the model because direct bonds are used in the connections to their counterparts.

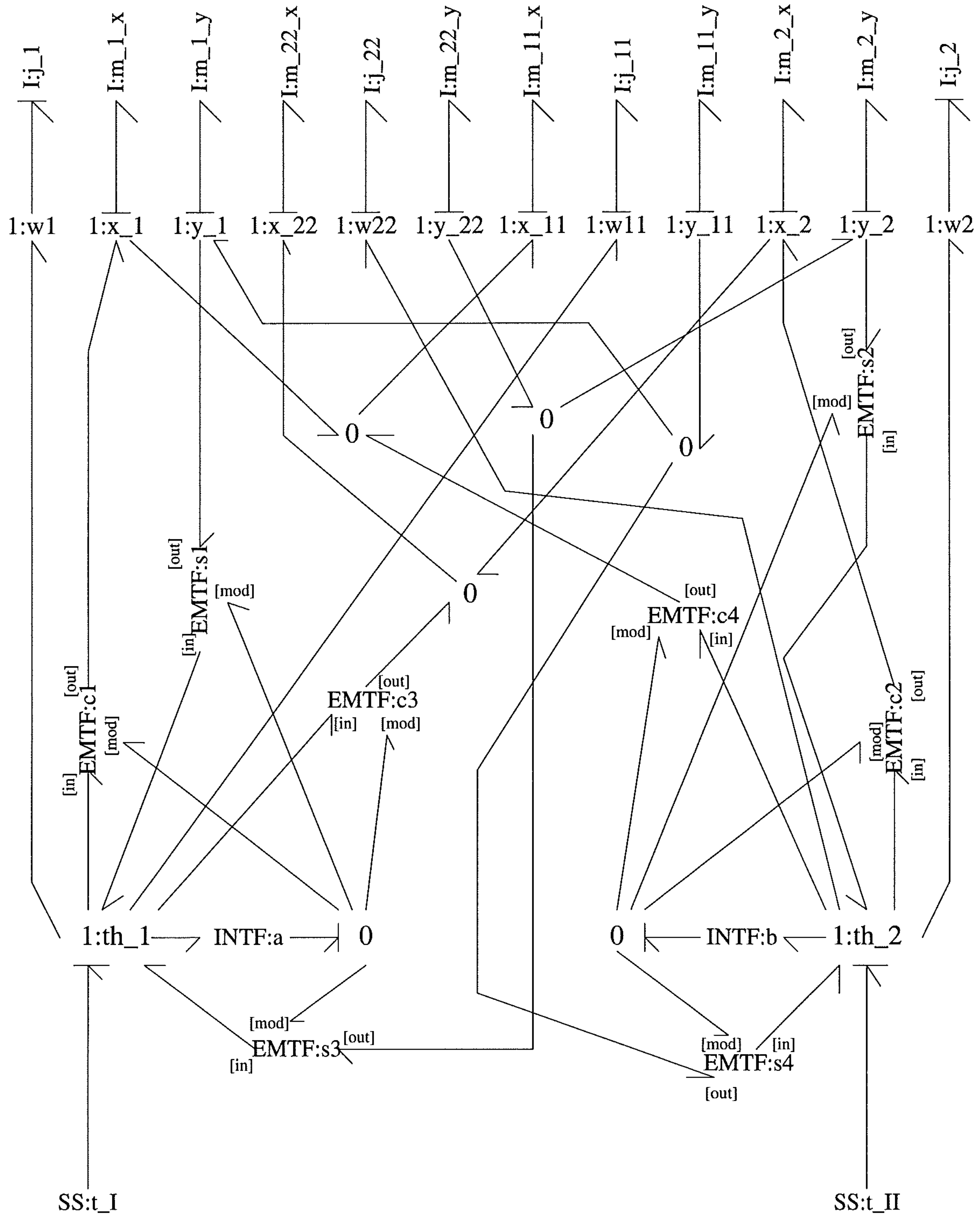


Fig. 2.4: Standard bond graph model for the parallel, 2-DOF, manipulator.

In order to complete the model of the manipulator the externally applied torques (*inputs*) and system states have to be defined on the graph. The inputs are represented by the two effort-source elements t_I and t_II. It should be noted that because in bond graphs the states result from every C- and I- element of integral causality and due the fact that each integrator has a C- element of this type (as is discussed in section 2.6) the system has two states these being the angular displacements of link Oa and of link Ob. Moreover, two other states have been chosen as the angular momenta of these two links. The definition of these two states is accomplished by assigning integral causality to the two *I-elements* that connected to the two 1-junctions representing the angular velocities of links Oa and Ob.

The second step in the bond graph modelling process using the MTT program is to write the label file of the model (Appendix B). This file is very important because it defines the characteristics of all the components in the bond graph model.

The third step in the modelling process is to extract the various representations generated by the software and this is not possible unless the bond graph of the system is causally complete. The most important representations are the structure representation and the simulation code and these can be found in Appendix (B). It is important to note that the simulation code generated by MTT has many errors in its structure meaning that it cannot be used directly in simulating the system in the Matlab program. The errors results from the fact that the software incorrectly produces capitalised letters in some words such as xInitial, MTTx1, MTTx2, MTTx3, MTTx4, MTTu1 and MTTu2.

To obtain the equation of motion in state space form the generated simulation code has to be transferred manually into the following matrix form:

$$\begin{bmatrix} 20 & 12\cos\Delta x & 0 & -12x_2\sin\Delta x \\ 12\cos\Delta x & 20 & 12x_1\sin\Delta x & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \frac{x_1}{J} \\ \frac{x_2}{J} \end{bmatrix} \qquad (2.15)$$

Where, $\Delta x = x_4 - x_3$

Equation 2.15 is not useful unless the state vector of the system is defined. This must be identified from the generated list of states (Appendix B).

The state vector of the system is as follows: $X = [j\dot{\theta}_1 \quad j\dot{\theta}_2 \quad \theta_1 \quad \theta_2]^T$

Equation 2.15 together with the above state vector gives the full information required for any control system design or analysis of the manipulator.

2.4.1 Discussion

It has been shown that twelve velocities have defined the parallel manipulator kinematics. Because these velocities are not independent the bond graph model shows many inertial (*I*-) elements with derivative causality. In the MTT program these represent the system non-states. The derivative causality in MTT does not affect the modelling process because the software has the ability to reduce differential algebraic equations into ordinary differential equations by using a special technique that is discussed by Gawthrop and Smith [50].

It has also been shown that once the constraints are solved for the parallel, 2-DOF, robot then the modelling process is systematic. In fact this is true for the system here because the constraints are very simple, and the existing MTT built-in components such as the EMTFs, are sufficient for performing the task. However, in general, solving the constraints for other simple parallel robots will inevitably lead to complicated equations that may contain functions other than simple relationships such as $L\cos, L\sin$, $2L\cos$ and $2L\sin$, and systems with these features cannot necessarily be modelled using MTT because of the lack of built-in functions.

2.5 Hierarchical bond graph modelling of the 2-DOF planar parallel manipulator using MTT

In serially driven manipulators, each link is responsible for generating one degree of freedom (two states). Hence the number of links is equal to the number of degrees of freedom of the manipulator. In fact, this feature is not relevant to systems that have closed kinematic chains in their topologies, such as parallel robots. In parallel robots the number of links can exceed the number of degrees of freedom. Because of this use of the standard bond graph technique for modelling parallel manipulators will lead to bond graph models with huge numbers of bonds and components. This feature will obviously make the graph complex and hard to follow. So for this reason the hierarchical technique of bond graph modelling is the most suitable when modelling such robots. The idea of hierarchical modelling [77] using MTT is based on a decomposition of the system into simple subsystems each of which can be considered as a component in a standard bond graph. These sub-systems may be themselves also be built hierarchically. Connecting the sub-systems together by ports allows the construction of the whole bond graph for the system. The power of a hierarchical bond graph lies in the fact that it extends the re-usability of a sub-system model, it clarifies the bond graph model and it reduces the number of bonds on the model's graph.

MTT has a library of ready-to-use subsystems for modelling mechanical manipulators that serve to reduce the time and effort required. For modelling planar robots, the ROD component represents any longitudinal link with a uniform cross-section. The bond-graph of this subsystem is shown in Figure 2.5. The component itself contains a subsystem which is the integrator INTF that integrates the angular velocity of the link and therefore producing the angular displacement needed by each modulated transform in the graph. The bond graph of this component is shown in Figure 2.6.

It is important to refer to the fact that the ROD component produces two fixed states in the form of angular momentum and angular displacement. The first state is due to the integrator (INTF) while the other state is due to the integral causality assigned to the inertial *I*- element attached to the angular velocity of the rod.

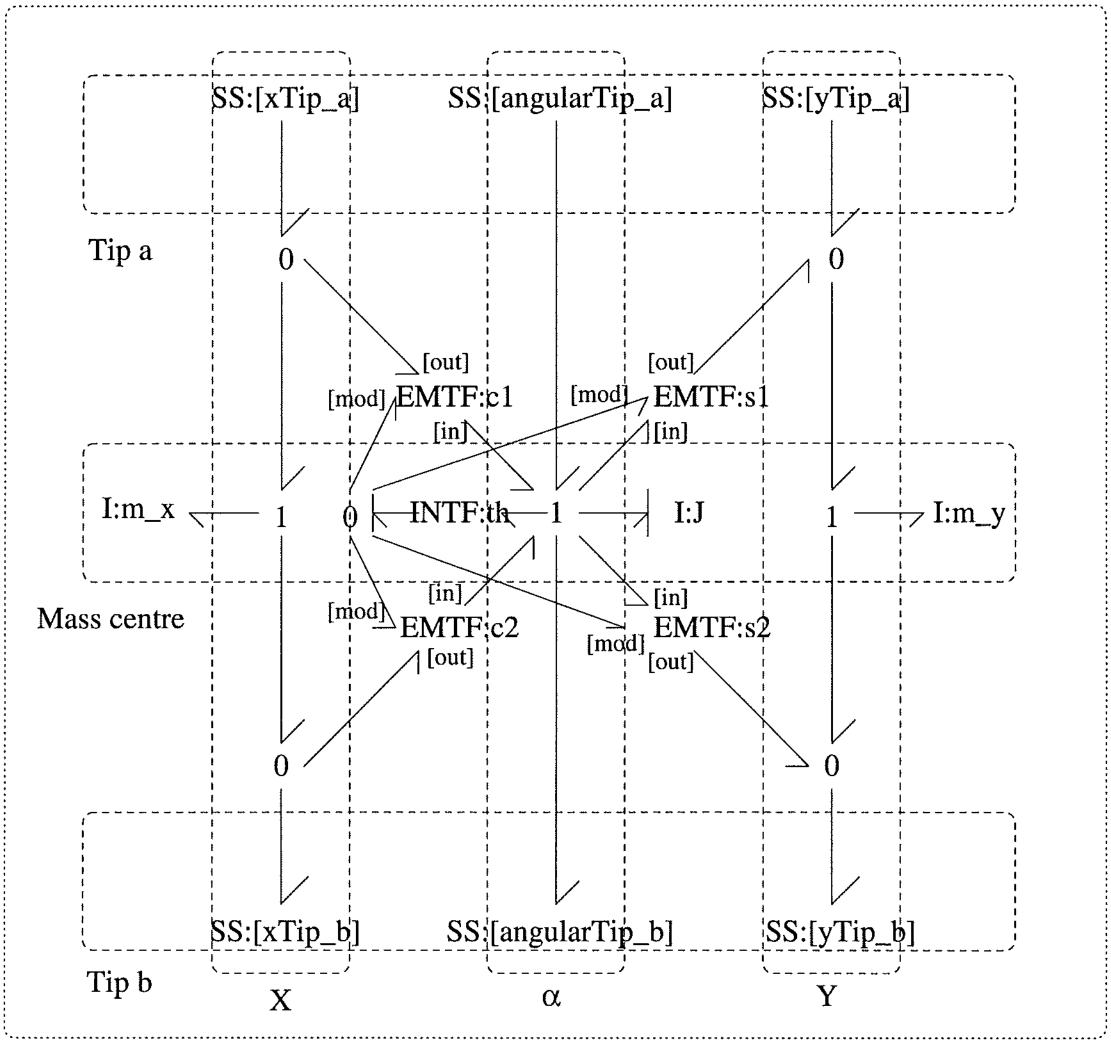


Fig. 2.5: Bond graph model for the ROD subsystem.

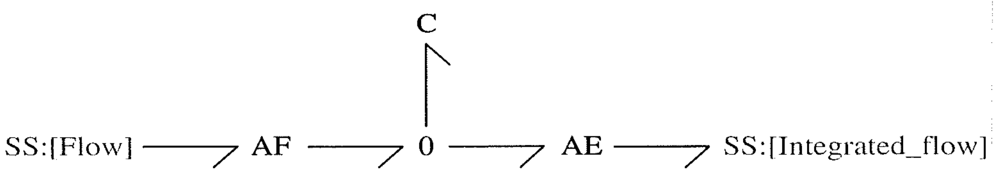


Fig. 2.6: Bond graph model for the INTF subsystem.

Figure 2.6 shows two new bond graph elements which are the AF and the AE two ports components. The general characteristics of these components and their constitutive relationships are defined in Table 2.6. It is important to refer to the fact that the gains k and n in the case of the *INTF* component are both equal to one.

Component	Gain	Symbol	Constitutive relationship
Effort amplifier **(AE)**	k	$\rightharpoondown\!\vert \quad AE:k \quad \rightharpoondown\!\vert$	$\begin{bmatrix} e_2 \\ f_1 \end{bmatrix} = \begin{bmatrix} k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ f_2 \end{bmatrix}$
Flow amplifier **(AF)**	n	$\rightharpoondown\!\vert \quad AF:n \quad \rightharpoondown\!\vert$	$\begin{bmatrix} f_2 \\ e_1 \end{bmatrix} = \begin{bmatrix} n & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}$

Table 2.6: Characteristics of *AF* and *AE* components.

During the modelling investigations for the hierarchical representation of the parallel, 2-DOF, manipulator several unavoidable problems occur. The summary of these problems is as follows:

- When constructing the bond graph model of the system, Figure 2.7, there is some confusion as to whether the power flow direction should be directed from serial chain Oac to serial chain $\tilde{O}$bc or from chain $\tilde{O}$bc to Oac.

- In directing the power flow either from the chain Oac to $\tilde{O}$bc or from $\tilde{O}$bc to Oac the MTT program indicates that the system is under-causal, meaning the failure of the modelling process. This is absolutely true because the causality considerations of both the bond graph models of the two serial chains Oac and Obc, Figures 2.8 and 2.9, lead to causality conflictions on the bonds that connect them.

- The use of the ROD component causes the system to have more states (eight) than the actual number (four). This is because each ROD produces two states which are the angular velocity and the angular displacement of the link that it represents.

- The solution of the constraints of the manipulator cannot be represented on the bond graph model due to the structure and the causality considerations of the ROD component.

To overcome the problems stated above a general method for modelling the existing manipulator is introduced and which can be applied systematically when modelling manipulators with closed kinematic chains inside the MTT program. The method can be summarised by means of the following points:

➢ *Preliminary construction of the bond graph model*

In this step the arm is transformed into *two* serial chains, that is Oac and $\tilde{O}$bc, and then the chains are separately represented, hierarchically, on the graph.

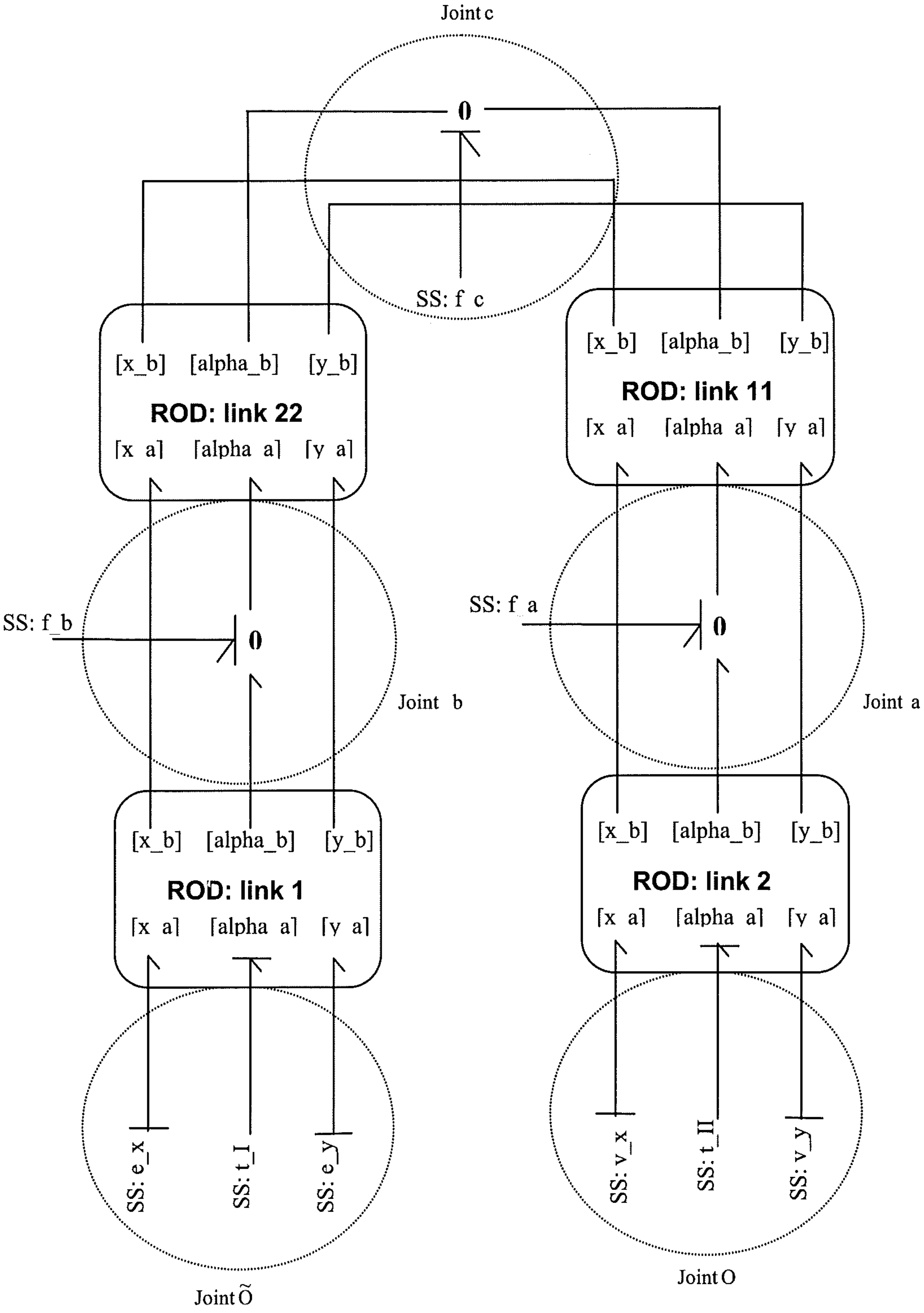


Fig. 2.7: Hierarchical bond graph model of the parallel, 2-DOF, manipulator

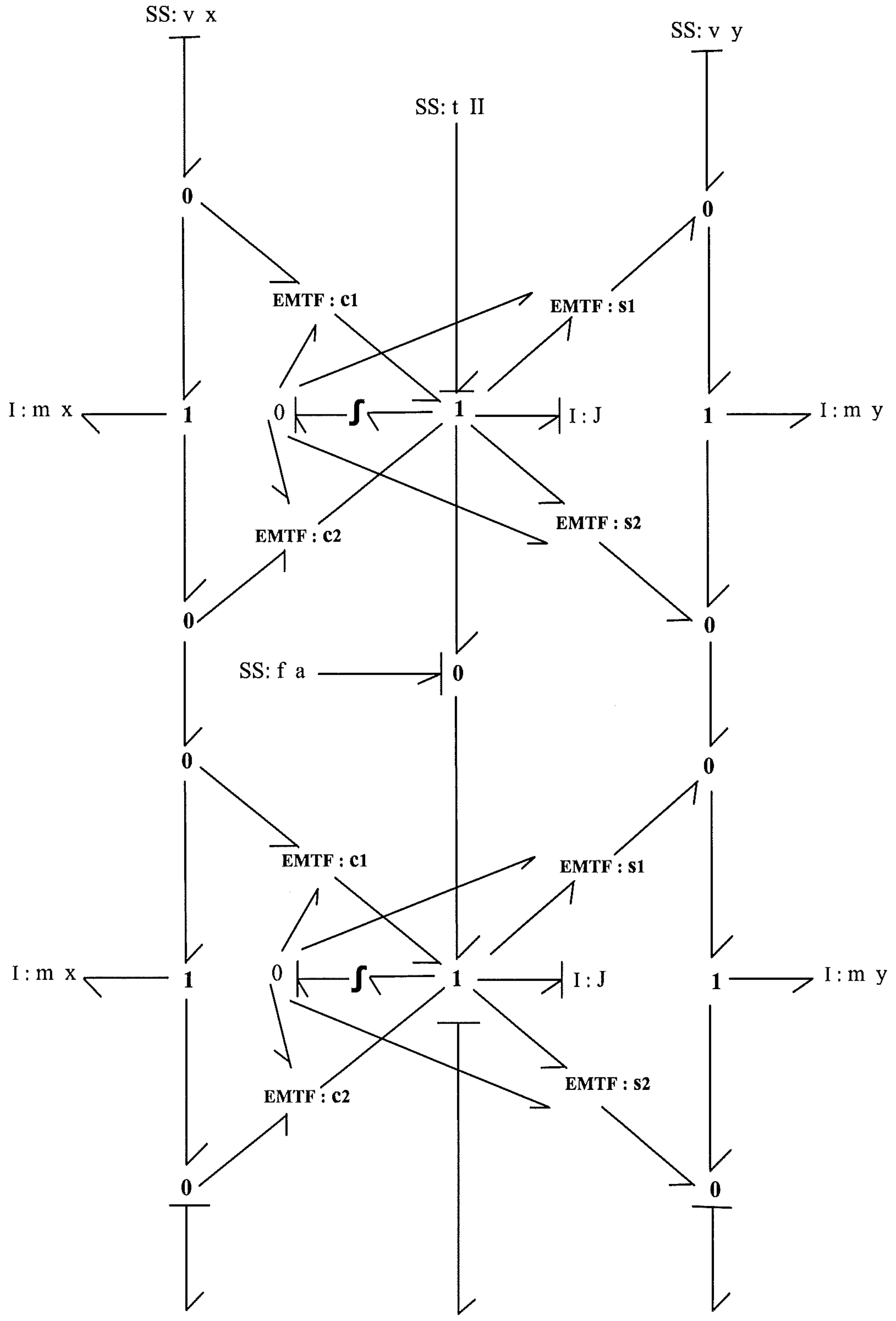


Fig. 2.8: Standard bond graph model for serial chain Oac.

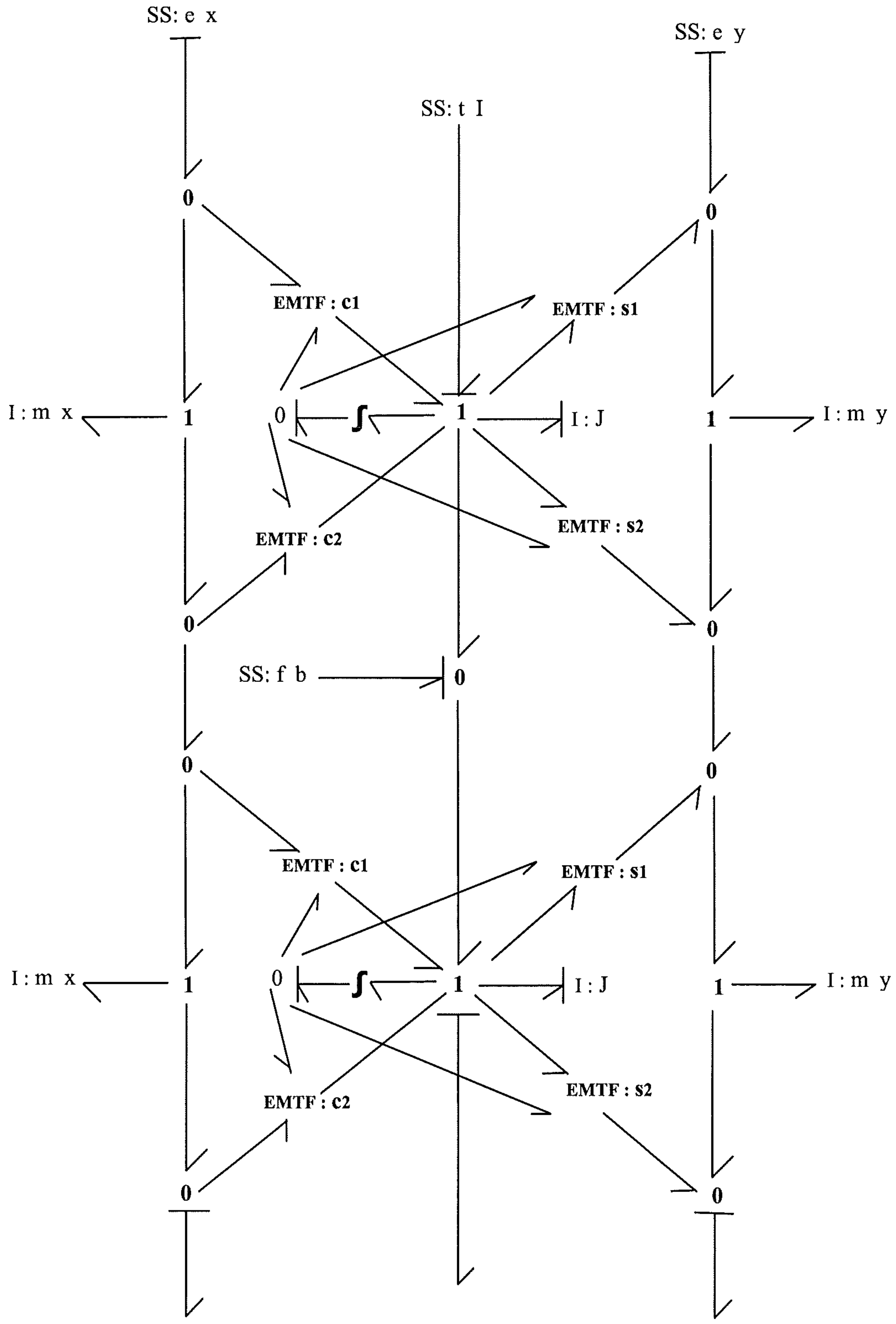


Fig. 2.9: Standard bond graph model for serial chain $\widetilde{O}bc$.

➤ *Finalising the bond graph model*

In order to connect the serial chains to finalise the model of the manipulator, Figure 2.10, the energy bonds from both chains are connected to each other by using a set of 0-junctions and then a dummy effort source is applied at each of these junctions. This process transforms the system from the under-causal state to the causal state so the system equations of motion can be obtained from the MTT program. It is important to refer to the fact that the addition of the effort sources necessarily assumes that external forces are applied to the system. To ensure that the applied effort sources have no effect on the velocity of the system zero values have been assigned to their values. The label file of the model is shown in Appendix C.

➤ *Equations of motion*

The equations of motion from the model are obtained by transforming the generated simulation code (Appendix C) manually into the state space form which in this case is as follows:

$$\begin{bmatrix} 16 & 0 & 6\cos\Delta x & -6x_3\sin\Delta x & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6\cos\Delta x & 6x_1\sin\Delta x & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 6\cos\Delta x' & -6x_7\sin\Delta x' \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6\cos\Delta x' & 6x_5\sin\Delta x' & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix}$$

$$= \begin{bmatrix} 2L(f_x\cos x_2 - f_y\sin x_2) + \tau_1 \\ \dfrac{x_1}{J} \\ 2L(f_x\cos x_4 - f_y\sin x_4) \\ \dfrac{x_3}{J} \\ 2L(f_x\cos x_6 - f_y\sin x_6) \\ \dfrac{x_5}{J} \\ 2L(f_x\cos x_8 - f_y\sin x_8) + \tau_2 \\ \dfrac{x_7}{J} \end{bmatrix} \qquad (2.16)$$

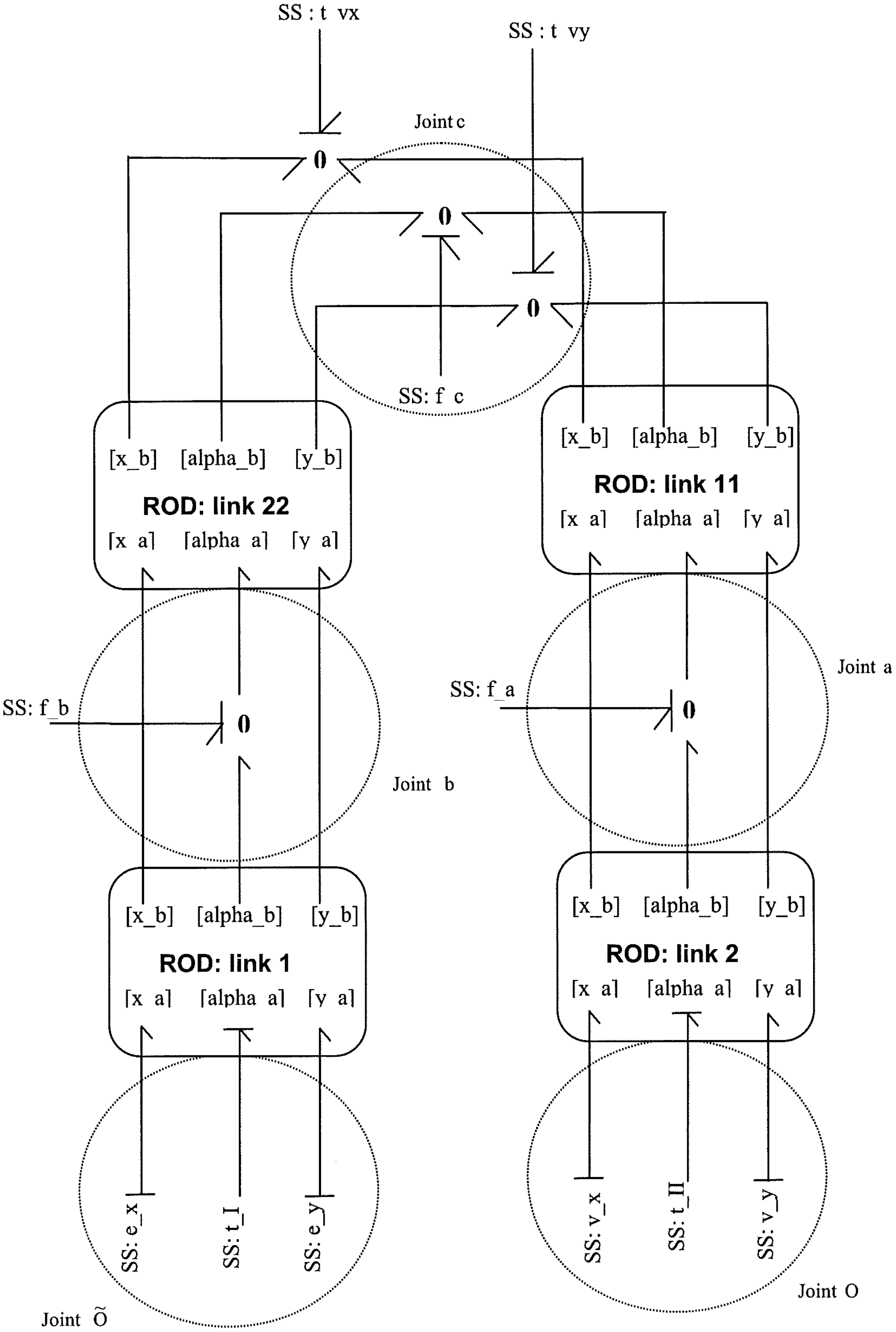


Fig. 2.10: The modified bond graph model for the parallel manipulator.

Where,

$$\Delta x = x_4 - x_2 \qquad \text{and} \qquad \Delta' x = x_8 - x_6$$

The system equations given by equation 2.16 are not final because the states are not independent and the equations contain the artificial forces, f_x and f_y which need to be removed.

- *States identification*

Because the MTT program defines the order of the of the system states automatically it is always impossible to know the relationships between the links and the states unless they are identified from the generated structure representations (Appendix C). Doing this the state vector for the manipulator is defined as follows:

$$X = [j\omega_1 \quad \theta_1 \quad j\omega_{22} \quad \theta_{22} \quad j\omega_{11} \quad \theta_{11} \quad j\omega_2 \quad \theta_2]^T$$

- *Final equations of motion*

In order to obtain the system equations of motion in its finalised form we do the following:

Recalling that the solution of the constraints of the manipulator yield,

$$\theta_1 = \theta_{11},\ \theta_2 = \theta_{22},\ \omega_1 = \omega_{11} \text{ and } \omega_2 = \omega_{22}$$

In terms of the states, the solution becomes:

$$x_2 = x_6,\ x_4 = x_8,\ x_5 = x_1 \text{ and } x_7 = x_3$$

Differentiating these with respect to time leads to,

$$\dot{x}_2 = \dot{x}_6, \dot{x}_4 = \dot{x}_8,\ \dot{x}_5 = \dot{x}_1 \text{ and } \dot{x}_7 = \dot{x}_3 \qquad (2.17)$$

Replacing the dummy forces in equation 2.16 by zeros and then substituting from 2.17 leads to the following:

$$\begin{bmatrix} 16 & 0 & 6\cos\Delta x & -6x_3\sin\Delta x & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6\cos\Delta x & 6x_1\sin\Delta x & 4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 6\cos\Delta x & -6x_3\sin\Delta x \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6\cos\Delta x & 6x_1\sin\Delta x & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \frac{x_1}{J} \\ 0 \\ \frac{x_3}{J} \\ 0 \\ \frac{x_1}{J} \\ \tau_2 \\ \frac{x_3}{J} \end{bmatrix} \tag{2.18}$$

It is important to refer here to the fact that replacing the dummy forces by zeros practically removes the holonomic constraint which connects the two serial chains.

Transforming equation 2.18 into individual equations gives:

$$16\dot{x}_1 + 6\dot{x}_3\cos\Delta x - 6x_3\dot{x}_4\sin\Delta x = \tau_1 \tag{2.19}$$

$$\dot{x}_2 = \frac{x_1}{j} \tag{2.20}$$

$$6\dot{x}_1\cos\Delta x + 6x_1\dot{x}_2\sin\Delta x + 4\dot{x}_3 = 0 \tag{2.21}$$

$$\dot{x}_4 = \frac{x_3}{j} \tag{2.22}$$

$$6\dot{x}_3\cos\Delta x - 6x_3\dot{x}_4\sin\Delta x + 4\dot{x}_1 = 0 \tag{2.23}$$

$$\dot{x}_2 = \frac{x_1}{j} \tag{2.24}$$

$$16\dot{x}_3 + 6\dot{x}_1\cos\Delta x + 6x_1\dot{x}_2\sin\Delta x = \tau_2 \tag{2.25}$$

$$\dot{x}_4 = \frac{x_3}{j} \tag{2.26}$$

Adding equations 2.19 and 2.23 leads to

$$20\dot{x}_1 + 12\dot{x}_3 \cos\Delta x - 12x_3\dot{x}_4 \sin\Delta x = \tau_1 \qquad (2.27)$$

Adding equations 2.21 and 2.25 also leads to

$$20\dot{x}_3 + 12\dot{x}_1 \cos\Delta x + 12x_1\dot{x}_2 \sin\Delta x = \tau_2 \qquad (2.28)$$

By removing repeated equations (2.24 and 2.26) and then writing equations 2.27, 2.28, 2.20 and 2.22 in matrix form it can be seen that the reduced equations of motion for the manipulator emerge, as follows:

$$\begin{bmatrix} 20 & 0 & 12\cos\Delta x & -12x_3\sin\Delta x \\ 0 & 1 & 0 & 0 \\ 12\cos\Delta x & 12x_1\sin\Delta x & 20 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \frac{x_1}{j} \\ \tau_2 \\ \frac{x_3}{j} \end{bmatrix} \qquad (2.29)$$

Taking into consideration the order of the states, it is clear that the derived equations of motion are the same as those given in section 2.4 (equation 2.15). In addition to this the results are also the same as those obtained in Appendix (A) by using the technique discussed in chapter 1.

It is important to refer here to the fact that the modelling technique here is similar to the *Serial Transformation Technique* of Appendix (A) in the way that the raw equations are belonging to the individual serial chains. The differences here lie in the fact that no artificial motors are used and the reduction of the system equations is performed by writing the dependent states in terms of the independent states instead of writing the unactuated angles in terms of the actuated angles.

2.6 Conclusions

In this chapter we have seen that Model Transformations Tools (MTT) software is able to deal with systems containing closed kinematic chains in their mechanical structure. In the standard bond graph, once the constraints are solved, the modelling process is straightforward and no further manipulation of the generated system equations is necessary. On the other hand, despite the reduction of the time and the effort during the construction of the system bond graph by use of the hierarchical tool, this approach proved to be not as straightforward.

During this chapter it was intended to show that the method developed for the hierarchical modelling of parallel robots is generally systematic and it is similar to the method described in chapter 1 in incorporating the constraints into the generated equations. By using dummy forces of zero values at the end-effector it is possible to simulate the process of transforming the parallel robot into two serial sub-chains.

Chapter 3

The Multi-Cell Parallel Manipulator: A Novel Technique for Designing Planar Parallel Robots

3.1 Introduction

In chapter one it has been discussed that one of the drawbacks of parallel manipulators is their inherently poor workspace. Clearly the optimisation of workspace is very important for all manipulator design. In the conventional 2-DOF parallel link manipulator the link lengths limit the workspace. In addition to this any increase in the link length will adversely affect the manipulator manoeuvrability. Notwithstanding that there will usually be a design limitation to some extent on the link length.

To overcome the limitations of the workspace of conventional 2-DOF parallel link manipulators the work discussed in this chapter is directed firstly to the introduction of a new design technique that allows an increase in the workspace without affecting the number of degrees of freedom of the manipulator, followed by a practical example. After that a general dynamic model for this practical example is derived using Lagrange's formulation. The model represents the manipulator in different orientations. Later in the chapter a new method for balancing this manipulator is introduced. The method actually reduces the number of variables to be adjusted to only one. Following on from this the chapter is formally concluded.

3.2 The planar multi-cell parallel Robot

If attention is initially restricted to revolute type mechanisms the conventional 2-DOF parallel manipulator can be considered as a five-bar linkage with two rotary inputs at points A and B as shown in Figure 1.5. If the distance between the two motors at A and B tends to zero, and all links have the same length, then the mechanism becomes the manipulator of chapter two (Figure 2.3). It has also been mentioned that this special case is very useful because of its simplified kinematic constraints and its workspace. If it is now considered that this linkage is a unit cell, and then several such cells are arranged in series, a multi-cell manipulator is constructed. In this way the number of degrees of freedom is not changed and the resulting workspace is equal to the number of cells multiplied by the original cell's workspace. Figure 3.1 shows a SCARA type manipulator in which this concept is applied. The manoeuvrability of this design can be optimised by proper reduction of the cell dimensions. This design also allows for the addition of multiple tools (e.g. milling or routing cutters) which may, in certain circumstances help in reducing the time required for performing tasks, for example in high-speed assembly operations.

3.3 Formulations of the general dynamic model for the multi-cell planar, 2-DOF, parallel manipulator.

3.3.1 Horizontal Orientation

The initial assumption is that the manipulator has $n+1$ cells, where n is the number of junctions in the manipulator, and a junction is defined as the point (joint) between two adjacent cells. The overall manipulator contains four links (Oa, Ob, de and ce) each having a mass m, a polar moment of inertia about the centre of mass J and a length $2L$. There are also $2n$ cross-links each with a mass $2m$, a polar moment of inertia about the centre of mass J_c and a length $4L$.

Assuming uniform rigid links with circular cross-sections yields

$$J = \tfrac{1}{3}mL^2 \qquad (3.1)$$

$$J_c = \tfrac{1}{3}\left\{2m(2L)^2\right\} = 8J \qquad (3.2)$$

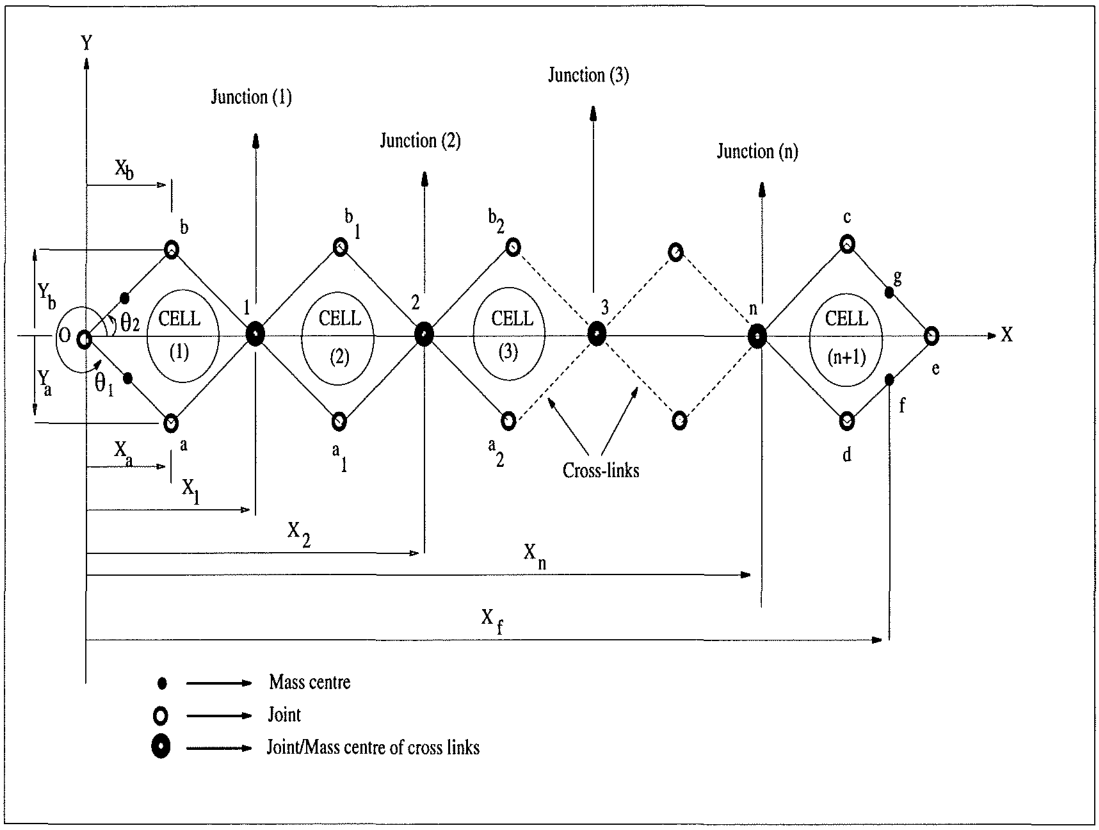


Fig. 3.1: Multi-Cell 2-DOF parallel planar manipulator, horizontal configuration.

The Lagrange formulation is used to generate the equations of motion, shown here without the usual potential energy term for orientational reasons,

$$\tau_i = \frac{d}{dt}\left(\frac{\partial T}{\partial \omega_i}\right) - \frac{\partial T}{\partial \theta_i} \qquad \text{for} \quad i = 1,2 \qquad (3.3)$$

Where,

T is the total kinetic energy of the robot.

θ_i is the generalised co-ordinate of link i.

ω_i is the angular velocity of link i.

τ_i is the generalised force in the direction of the generalised co-ordinate.

The kinetic energies of the actuated links (Oa and Ob) can be calculated as follows,

$$T_{Oa} = \tfrac{1}{2}(I_O)_{Oa}\omega_1^2 \qquad (3.4)$$

$$T_{Ob} = \tfrac{1}{2}(I_O)_{Ob}\omega_2^2 \qquad (3.5)$$

Where

T_{Oa}, T_{Ob} are the kinetic energies of links Oa and Ob, respectively.

ω_1, ω_2 are the angular velocities of link Oa and Ob, respectively.

$(I_O)_{Oa}, (I_O)_{Ob}$ are the polar moments of inertia of links Oa and Ob about point O.

The symmetry of links Oa and Ob leads to,

$$(I_O)_{Oa} = (I_O)_{Ob} = J + mL^2 = 4J \qquad (3.6)$$

Substituting from equation 3.6 in equations 3.4 and 3.5 gives

$$T_{Oa} = 2J\omega_1^2 \qquad (3.7)$$

$$T_{Ob} = 2J\omega_2^2 \qquad (3.8)$$

Now, let

x_j, y_j be the x-y co-ordinates of point j.

j be an identifier assigned to a point.

v_j be the absolute resultant velocity of point j.

Taking into considerations the solution of the constraints of the main cell outlined in chapter 2, it can be deduced, from Figure 3.1, that:

$$x_1 = 2L(\cos\theta_1 + \cos\theta_2) \qquad (3.9)$$

$$y_1 = 2L(\sin\theta_1 + \sin\theta_2) \qquad (3.10)$$

Differentiating these with respect to time leads to,

$$\dot{x}_1 = -2L(\omega_1 \sin\theta_1 + \omega_2 \sin\theta_2) \qquad (3.11)$$

$$\dot{y}_1 = 2L(\omega_1 \cos\theta_1 + \omega_2 \cos\theta_2) \qquad (3.12)$$

The absolute resultant velocity of point 1 can be calculated from the above as follows,

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2$$

$$v_1^2 = 4L^2\left\{\omega_1^2 + \omega_2^2 + 2\omega_1\omega_2\cos(\theta_2 - \theta_1)\right\} \qquad (3.13)$$

$$\alpha = \tan^{-1}\left(\frac{\dot{y}_1}{\dot{x}_1}\right)$$

Where

α is the angle between v_1 and $\dot{x}_1$.

At every junction there are two concentrated masses representing the two cross-links connected at this junction. These two masses have the same translational velocity whereas one of the cross-links has angular velocity ω_1 and the other has angular velocity ω_2.

From the above,

$$T_1 = 2\left\{\tfrac{1}{2}(2m)v_1^2\right\} + \tfrac{1}{2}J_c\omega_1^2 + \tfrac{1}{2}J_c\omega_2^2$$

$$= 2m\,v_1^2 + \tfrac{1}{2}(8J)\;\omega_1^2 + \tfrac{1}{2}(8J)\omega_2^2$$

$$= 2m\,v_1^2 + 4J\;\left(\omega_1^2 + \omega_2^2\right) \qquad (3.14)$$

Where

T_1 is the kinetic energy of the cross-links at junction 1.

Similarly, it can be deduced that

$$T_2 = 2mv_2^2 + 4J\left(\omega_1^2 + \omega_2^2\right) \qquad (3.15)$$

$$T_3 = 2mv_3^2 + 4J\left(\omega_1^2 + \omega_2^2\right) \qquad (3.16)$$

For any junction n, it can therefore stated that

$$T_n = 2mv_n^2 + 4J\left(\omega_1^2 + \omega_2^2\right) \qquad (3.17)$$

Now, if T_{tc} is the total kinetic energy of all cross-links at all the n junctions then it is possible to write

$$T_{tc} = T_1 + T_2 + T_3 + T_4 + T_5 + \cdots\cdots + T_{n-1} + T_n$$

Substituting from equations 3.14 to 3.17, the previous equation becomes

$$T_{tc} = 4Jn\left(\omega_1^2 + \omega_2^2\right) + 2m\left(v_1^2 + v_2^2 + v_3^2 + v_4^2 + v_5^2 + \cdots\cdots + v_{n-1}^2 + v_n^2\right) \tag{3.18}$$

From the physical structure of the manipulator shown in Figure 3.1 it can be deduced that

$$x_2 = 2\,x_1\ ,\ x_3 = 3\,x_1\ ,\ x_4 = 4\,x_1\ ,\ x_5 = 5\,x_1$$

leading to

$$x_n = n\,x_1 \tag{3.19}$$

Also,

$$y_2 = 2\,y_1\ ,\ y_3 = 3\,y_1\ ,\ y_4 = 4\,y_1\ ,\ y_5 = 5\,y_1$$

which leads to

$$y_n = n\,y_1 \tag{3.20}$$

Differentiating these with respect to time and calculating the absolute resultant velocity at each junction gives the following:

$$v_2 = 2\,v_1\ ,\ v_3 = 3\,v_1\ ,\ v_4 = 4\,v_1\ ,\ v_5 = 5\,v_1$$

and therefore

$$v_n = n\,v_1 \tag{3.21}$$

Now, from equation 3.18, it is possible to construct the following

$$T_{tc} = 4Jn\left(\omega_1^2 + \omega_2^2\right) + 2m\left(v_1^2 + 4\,v_1^2 + 9\,v_1^2 + 16\,v_4^2 + \cdots\cdots + n^2\,v_1^2\right)$$

$$= 4Jn\left(\omega_1^2 + \omega_2^2\right) + 2m\,v_1^2\left(1 + 4 + 9 + 16 + \cdots\cdots + n^2\right)$$

$$= 4Jn\ \left(\omega_1^2 + \omega_2^2\right) + 2m\,v_1^2\left\{\tfrac{1}{6}n(n+1)(2n+1)\right\} \tag{3.22}$$

Substituting for v_1 from equation 3.13 leads to,

$$T_{tc} = 4J\left\{ n + n(n+1)(2n+1)\right\}\left(\omega_1^2 + \omega_2^2\right) + \left\{8Jn(n+1)(2n+1)\right\}\omega_1\omega_2\cos(\theta_2 - \theta_1) \tag{3.23}$$

Assuming that there is an imaginary junction at point e, i.e. this will be the junction number $n+1$ in the system, so from equations 3.19 and 3.20,

$$x_e = x_{n+1} = (n+1)x_1 \tag{3.24}$$

and,

$$y_e = y_{n+1} = (n+1)y_1 \tag{3.25}$$

From Figure 3.1 it can be deduced that

$$\begin{aligned} x_f &= x_e - L\cos\theta_2 \\ &= 2(n+1)L\cos\theta_1 + (2n+1)L\cos\theta_2 \end{aligned} \tag{3.26}$$

And,

$$\begin{aligned} y_f &= y_e - L\sin\theta_2 \\ &= 2(n+1)L\sin\theta_1 + (2n+1)L\sin\theta_2 \end{aligned} \tag{3.27}$$

Differentiating these with respect to time gives,

$$\dot{x}_f = -2(n+1)L\omega_1\sin\theta_1 - (2n+1)L\omega_2\sin\theta_2 \tag{3.28}$$

and,

$$\dot{y}_f = 2(n+1)L\omega_1\cos\theta_1 + (2n+1)L\omega_2\cos\theta_2 \tag{3.29}$$

Calculating the absolute resultant velocity,

$$v_f^2 = \dot{x}_f^2 + \dot{y}_f^2$$

$$= 4\left(n+1\right)^2 L^2\omega_1^2 + \left(2n+1\right)^2 L^2\omega_2^2 + 4\left(n+1\right)\left(2n+1\right)L^2\omega_1\omega_2\cos\left(\theta_2-\theta_1\right) \qquad (3.30)$$

Similarly,

$$x_g = x_e - L\cos\theta_1$$

$$= 2\left(n+1\right)L\cos\theta_2 + \left(2n+1\right)L\cos\theta_1 \qquad (3.31)$$

And,

$$y_g = y_e - L\sin\theta_1$$

$$= 2\left(n+1\right)L\sin\theta_2 + \left(2n+1\right)L\sin\theta_1 \qquad (3.32)$$

Calculating the absolute resultant velocity of point g yields,

$$v_g^2 = 4\left(n+1\right)^2 L^2\omega_2^2 + \left(2n+1\right)^2 L^2\omega_1^2 + 4\left(n+1\right)\left(2n+1\right)L^2\omega_1\omega_2\cos\left(\theta_2-\theta_1\right) \qquad (3.33)$$

Next, in order to find the contributions of links de and ce the kinetic energies are defined as follows,

T_{de}, T_{ce} are the kinetic energies of links de and ce respectively.

Where,

$$T_{de} = \tfrac{1}{2}m\,v_f^2 + \tfrac{1}{2}J\,\omega_2^2 \qquad (3.34)$$

$$T_{ce} = \tfrac{1}{2}m\,v_g^2 + \tfrac{1}{2}J\,\omega_1^2 \qquad (3.35)$$

Substituting for v_f and v_g, from equations 3.30 and 3.33, in 3.34 and 3.35 yields,

$$T_{de} = 6J(n+1)^2\omega_1^2 + \tfrac{1}{2}J\left\{3(2n+1)^2+1\right\}\omega_2^2 + 6J(n+1)(2n+1)\omega_1\omega_2\cos(\theta_2-\theta_1) \tag{3.36}$$

and,

$$T_{ce} = 6J(n+1)^2\omega_2^2 + \tfrac{1}{2}J\left\{3(2n+1)^2+1\right\}\omega_1^2 + 6J(n+1)(2n+1)\omega_1\omega_2\cos(\theta_2-\theta_1) \tag{3.37}$$

Therefore from the foregoing the total kinetic energy of the system (T) can be determined, thus,

$$\begin{aligned}
T &= T_{Ob} + T_{Oa} + T_{de} + T_{ce} + T_{tc} \\
&= 2J\,\omega_1^2 + 2J\,\omega_2^2 \\
&\quad + 6J(n+1)^2\omega_2^2 + \tfrac{1}{2}J\left\{3(2n+1)^2+1\right\}\omega_1^2 + 6J(n+1)(2n+1)\omega_1\omega_2\cos(\theta_2-\theta_1) \\
&\quad + 6J(n+1)^2\omega_1^2 + \tfrac{1}{2}J\left\{3(2n+1)^2+1\right\}\omega_2^2 + 6J(n+1)(2n+1)\omega_1\omega_2\cos(\theta_2-\theta_1) \\
&\quad + 4J\left\{n + n(n+1)(2n+1)\right\}\left(\omega_1^2+\omega_2^2\right) + \left\{8Jn(n+1)(2n+1)\right\}\omega_1\omega_2\cos(\theta_2-\theta_1) \\
&= J\omega_1^2\left\{8n^3+24n^2+26n+10\right\} + J\omega_2^2\left\{8n^3+24n^2+26n+10\right\} \\
&\qquad + J\omega_1\omega_2\cos(\theta_2-\theta_1)\left\{16n^3+48n^2+44n+12\right\}
\end{aligned} \tag{3.38}$$

Now, letting

$$\xi = \left(8n^3+24n^2+26n+10\right) \tag{3.39}$$

$$\psi = \left(16n^3+48n^2+44n+12\right) \tag{3.40}$$

Substituting into equation 3.38 gives,

$$T = J\xi\omega_1^2 + J\xi\omega_2^2 + J\psi\omega_1\omega_2\cos(\theta_2-\theta_1) \tag{3.41}$$

Taking the necessary partial derivatives of equation 3.41 gives,

$$\frac{\partial T}{\partial \omega_1} = 2J\,\xi\omega_1 + J\psi\cos(\theta_2 - \theta_1)\,\omega_2 \qquad (3.42)$$

$$\frac{\partial T}{\partial \omega_2} = 2J\xi\,\omega_2 + J\psi\cos(\theta_2 - \theta_1)\,\omega_1 \qquad (3.43)$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \omega_1}\right) = 2J\xi\,\ddot{\theta}_1 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - J\psi\sin(\theta_2 - \theta_1)\omega_2\,(\omega_2 - \omega_1) \qquad (3.44)$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \omega_2}\right) = 2J\xi\,\ddot{\theta}_2 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_1 - J\psi\sin(\theta_2 - \theta_1)\omega_1(\omega_2 - \omega_1) \qquad (3.45)$$

$$\frac{\partial T}{\partial \theta_1} = J\psi\sin(\theta_2 - \theta_1)\,\omega_1\omega_2 \qquad (3.46)$$

$$\frac{\partial T}{\partial \theta_2} = -J\psi\sin(\theta_2 - \theta_1)\,\omega_1\omega_2 \qquad (3.47)$$

Applying Lagrange's equations yields,

$$\tau_1 = 2J\xi\ddot{\theta}_1 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - J\psi\sin(\theta_2 - \theta_1)\,\omega_2^2 \qquad (3.48)$$

and,

$$\tau_2 = 2J\xi\ddot{\theta}_2 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_1 + J\psi\sin(\theta_2 - \theta_1)\,\omega_1^2 \qquad (3.49)$$

Transforming equations 3.48 and 3.49 to state space form requires the following substitutions,

$$\omega_1 = x_1\ ,\ \theta_1 = x_2,\ \omega_2 = x_3 \text{ and } \theta_2 = x_4$$

Substituting in 3.48 and 3.49 yields,

$$\frac{\tau_1}{J} = 2\xi\,\dot{x}_1 + \psi\cos(x_4 - x_2)\dot{x}_3 - \psi\sin(x_4 - x_2)x_3\,\dot{x}_4 \tag{3.50}$$

and,

$$\frac{\tau_2}{J} = 2\xi\,\dot{x}_3 + \psi\cos(x_4 - x_2)\dot{x}_1 + \psi\sin(x_4 - x_2)x_1\,\dot{x}_2 \tag{3.51}$$

By writing equations 3.50 and 3.51 in matrix form we get,

$$\begin{bmatrix} 2\xi & 0 & \psi\cos\Delta x & -x_3\psi\sin\Delta x \\ 0 & 1 & 0 & 0 \\ \psi\cos\Delta x & x_1\psi\sin\Delta x & 2\xi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{\tau_1}{J} \\ x_1 \\ \frac{\tau_2}{J} \\ x_3 \end{bmatrix} \tag{3.52}$$

Where, $\Delta x = x_4 - x_2$

The state vector is given by

$$X = [\omega_1 \quad \theta_1 \quad \omega_2 \quad \theta_2]^T$$

Now, if we substitute for $n = 0$ (which means that there is only one cell) in equation 3.52, the following is obtained

$$\begin{bmatrix} 20 & 0 & 12\cos\Delta x & -12x_3\sin\Delta x \\ 0 & 1 & 0 & 0 \\ 12\cos\Delta x & 12x_1\sin\Delta x & 20 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{\tau_1}{J} \\ x_1 \\ \frac{\tau_2}{J} \\ x_3 \end{bmatrix} \tag{3.53}$$

It is clear that the result given by equation 3.53 is the same as that obtained in chapter 2 for the one-cell manipulator if it is considered that the first and the third elements of the state vector in chapter 2 are the angular momenta while here they are the angular speeds.

3.3.2 Vertical Orientation

In this case the manipulator is moving in the X - Z plane. A part of the manipulator with the new configuration is shown in Figure 3.2. The system has the same kinetic energy as when the manipulator is moving in the horizontal plane.

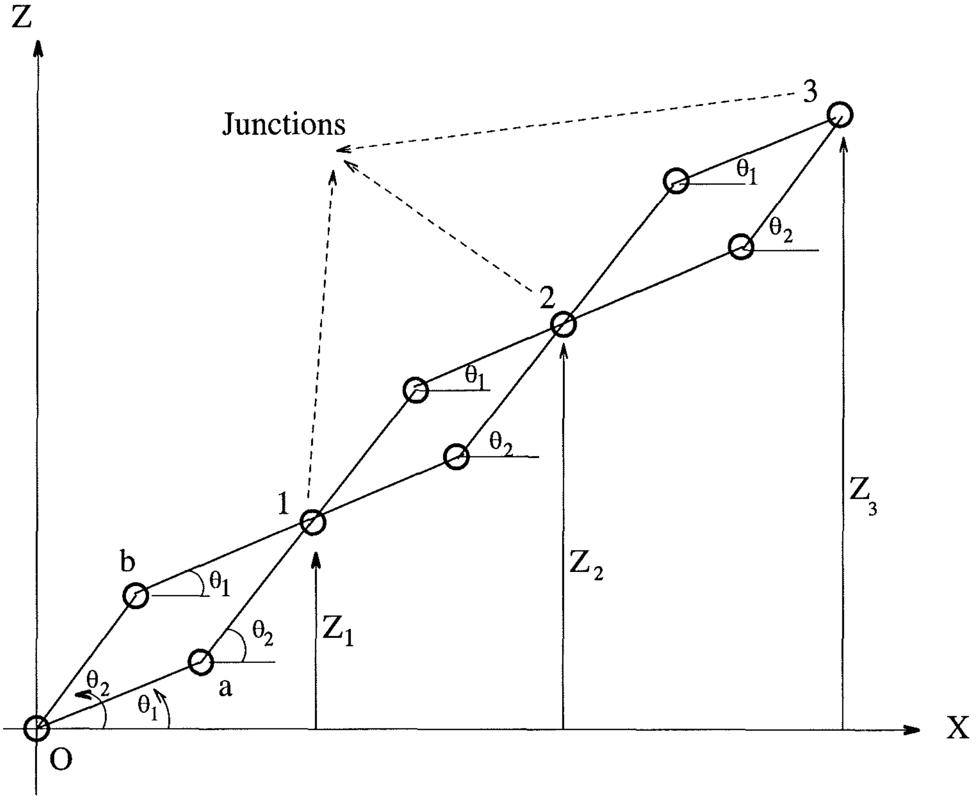


Fig. 3.2: Part of the manipulator with the new orientation.

Recalling the value of the kinetic energy T,

$$T = J\omega_1^2\left\{8n^3 + 24n^2 + 26n + 10\right\} + J\omega_2^2\left\{8n^3 + 24n^2 + 26n + 10\right\}$$
$$+ J\omega_1\,\omega_2\cos(\theta_2 - \theta_1)\left\{16n^3 + 48n^2 + 44n + 12\right\} \qquad (3.54)$$

To complete the derivation of the equations of motion it remains to find the total potential energy of the manipulator, U. The potential energy stored in the i^{th} link of the mechanism is the amount of work required to displace its centre of mass from the horizontal reference plane in the presence of gravity.

The total potential energy U stored in the system is given by [71]

$$U = -\sum_{i=1}^{k} m_i g^T C^i \qquad (3.55)$$

Where,

m_i is the mass of the i^{th} link.

g is the gravitational acceleration vector.

C^i is the vector representing the centre of mass of the i^{th} link as shown in Figure 3.3.

Now, letting

$x_{c/i}$: be the x co-ordinate of the mass centre of the i^{th} link.

$y_{c/i}$: be the y co-ordinate of the mass centre of the i^{th} link.

$z_{c/i}$: be the z co-ordinate of the mass centre of the i^{th} link.

g_0 : be the acceleration of gravity, 9.81 $m/\sec^2$.

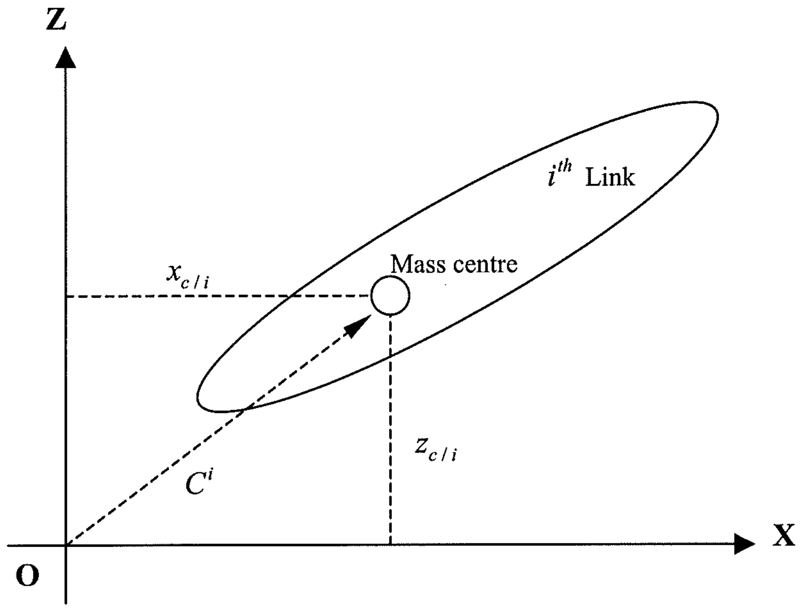


Fig. 3.3: Configuration of the i^{th} link.

Then,

$$g^T = \begin{bmatrix} 0 & 0 & g_0 \end{bmatrix}$$

$$C^i = \begin{bmatrix} x_{c/i} & y_{c/i} & z_{c/i} \end{bmatrix}^T$$

$$g^T C^i = \begin{bmatrix} 0 & 0 & g_0 \end{bmatrix} \begin{bmatrix} x_{c/i} \\ y_{c/i} \\ z_{c/i} \end{bmatrix} = g_0 z_{c/i} \qquad (3.56)$$

Substituting into equation 3.55 yields,

$$U = -\sum_{i=1}^{k} m_i g_0 z_{c/i} \qquad (3.57)$$

- Links Oa and Ob.

Letting

U_{Oa} : be the potential energy of link Oa.

U_{Ob} : be the potential energy of link Ob.

Then from equation 3.57, we have

$$U_{Oa} = -mg_0 z_{c/Oa}$$
$$U_{Ob} = -mg_0 z_{c/Ob}$$

From Figure 2.3, the following emerges

$$z_{c/Oa} = L\sin\theta_1$$
$$z_{c/Ob} = L\sin\theta_2$$

Now, the potential energies of links Oa and Ob become

$$U_{Oa} = -mg_0 L\sin\theta_1 \qquad (3.58)$$

$$U_{Ob} = -mg_0 L\sin\theta_2 \qquad (3.59)$$

- Cross-links

Recalling that each junction represents the mass centres of the connected cross-links. Hence, there are two masses at every junction, each with a value of $2m$.

Let, $U_1, U_2, U_3, \ldots\ldots\ldots\ldots, U_n$ represent the potential energies of the cross-links at junctions 1,2,3,............,n.

$$U_1 = -2(2m)g_0 z_1 = -4mg_0 z_1$$

Similarly,

$$U_2 = -4mg_0 z_2$$

$$U_3 = -4mg_0 z_3$$

$$U_4 = -4mg_0 z_4$$

Now, for any junction it is possible to write,

$$U_n = -4mg_0 z_n \qquad (3.60)$$

Letting

U_{tc} be the total potential energy of all cross-links at all the junctions.

Then, we have,

$$U_{tc} = U_1 + U_2 + U_3 + U_4 + \ldots\ldots\ldots\ldots + U_n$$

$$= -4\,m\,g_0\left\{\,z_1 + z_2 + z_3 + z_4 + \ldots\ldots\ldots\ldots + z_n\right\} \qquad (3.61)$$

Since the Z co-ordinate for any point here is the same as the Y co-ordinate in the pervious part of the chapter (where the manipulator moves in the horizontal plane), the following relationships are obtained

$$\left(\begin{array}{l} z_1 = 2L(\sin\theta_1 + \sin\theta_2) \\ z_2 = 2z_1 \\ z_3 = 3z_1 \\ z_n = nz_1 \\ z_{c/de} = z_f = 2L(n+1)\sin\theta_1 + L(2n+1)\sin\theta_2 \\ z_{c/ce} = z_g = 2L(n+1)\sin\theta_2 + L(2n+1)\sin\theta_1 \end{array}\right) \qquad (3.63)$$

Now, from equation 3.61 it is clear that

$$U_{tc} = -4mg_0\{ z_1 + 2z_1 + 3z_1 + 4z_1 + \ldots\ldots\ldots\ldots + nz_1\}$$

$$= -4mg_0 z_1\{ 1 + 2 + 3 + 4 + \ldots\ldots\ldots\ldots + n\} \qquad (3.63)$$

The term $\{ 1 + 2 + 3 + 4 + \ldots\ldots\ldots + n\}$ constitutes a sequence of the form

$$a + (a + d) + (a + 2d) + (a + 3d) + \ldots\ldots$$

With $a = d = 1$, the sum to n terms is given by

$$\tfrac{1}{2}n\{ 2a + (n-1)d\} = \tfrac{1}{2}n(n+1)$$

Substituting into equation 3.63 yields,

$$U_{tc} = -4\, mg_0 2L(\sin\theta_1 + \sin\theta_2)\tfrac{1}{2}n(n+1)$$

$$= -4\, n(n+1)Lmg_0(\sin\theta_1 + \sin\theta_2) \qquad (3.64)$$

- Links ce and de

$$U_{ce} = -m\, g_0\, z_g$$

$$= -mg_0\{ 2L(n+1)\, \sin\theta_2 + L(2n+1)\, \sin\theta_1\} \qquad (3.65)$$

and,

$$U_{de} = -mg_0 z_f$$

$$= -mg_0\{ 2L(n+1)\sin\theta_1 + L(2n+1)\sin\theta_2\} \qquad (3.66)$$

Now, from equation 3.57, the total potential energy of the manipulator can be shown to be as follows

$$U = -\sum_{i=1}^{k} m_i g_0 \, z_{c/i} = U_{Oa} + U_{Ob} + U_{tc} + U_{ce} + U_{de}$$

$$= -mg_0 L \sin\theta_1 - mg_0 L \sin\theta_2$$

$$-4n\,(n+1) L m g_0 (\sin\theta_1 + \sin\theta_2)$$

$$-mg_0 \{\, 2L(n+1)\sin\theta_2 + L(2n+1)\sin\theta_1 \}$$

$$-mg_0 \{\, 2L(n+1)\sin\theta_1 + L(2n+1)\sin\theta_2 \}$$

This gives,

$$U = -mg_0 L\phi\,(\sin\theta_1 + \sin\theta_2) \qquad (3.67)$$

Where, $\phi = (4n^2 + 8n + 4)$

The Lagrange equations for systems having potential energy, is given by

$$\tau_i = \frac{d}{dt}\left(\frac{\partial T}{\partial \omega_i}\right) - \frac{\partial T}{\partial \theta_i} - \frac{\partial U}{\partial \theta_i} \qquad \text{for} \quad i = 1,2 \qquad (3.68)$$

$$\frac{\partial U}{\partial \theta_1} = -mg_0 L\phi \cos\theta_1$$

$$\frac{\partial U}{\partial \theta_2} = -mg_0 L\phi \cos\theta_2$$

Since the total kinetic energy of the system is the same as the horizontal case, the first and the second terms of the R.H.S. of equation 3.68 can be obtained from the previous part of the chapter.

$$\tau_1 = \frac{d}{dt}\left(\frac{\partial T}{\partial \omega_1}\right) - \frac{\partial T}{\partial \theta_1} - \frac{\partial U}{\partial \theta_1}$$

$$= 2J\xi\,\ddot{\theta}_1 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - J\psi\omega_2\left(\omega_2 - \omega_1\right)\sin(\theta_2 - \theta_1)$$

$$- J\psi\omega_1\omega_2\sin(\theta_2 - \theta_1) + mg_0L\phi\cos\theta_1$$

$$= 2J\xi\,\ddot{\theta}_1 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - J\psi\omega_2^2\sin(\theta_2 - \theta_1) + mg_0L\phi\cos\theta_1 \tag{3.69}$$

$$\tau_2 = \frac{d}{dt}\left(\frac{\partial T}{\partial \omega_2}\right) - \frac{\partial T}{\partial \theta_2} - \frac{\partial U}{\partial \theta_2}$$

$$= 2J\xi\,\ddot{\theta}_2 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_1 - J\psi\omega_1\left(\omega_2 - \omega_1\right)\sin(\theta_2 - \theta_1)$$

$$+ J\psi\omega_1\omega_2\sin(\theta_2 - \theta_1) + mg_0L\phi\cos\theta_2$$

$$= 2J\xi\,\ddot{\theta}_2 + J\psi\cos(\theta_2 - \theta_1)\ddot{\theta}_1 + J\psi\omega_1^2\sin(\theta_2 - \theta_1) + mg_0L\phi\cos\theta_2 \tag{3.70}$$

Transforming the above two equations into state space form as before, the following equation of motion results

$$\begin{bmatrix} 2J\xi & 0 & J\psi\cos\Delta x & -J\psi\, x_3\sin\Delta x \\ 0 & 1 & 0 & 0 \\ J\psi\cos\Delta x & J\psi\, x_1\sin\Delta x & 2J\xi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \tau_1 - k\cos x_2 \\ x_1 \\ \tau_2 - k\cos x_4 \\ x_3 \end{bmatrix} \tag{3.71}$$

Where,

$$k = mg_0L\phi \quad \text{and} \quad \Delta x = x_4 - x_2$$

3.3.2.1 Static balancing of the manipulator

In order to perform balancing counter masses must be added to the manipulator. The traditional way to balance manipulators is to fit extensions to the links and then add counter masses to the end of these extensions [78]. In this work balancing is performed by adding a counter cell and one counter mass. The configuration of the counter cell and the location of the counter mass are shown in Figure 3.4 below.

The potential energy of the counter system (U_{cs}) is calculated as follows

- Potential energy of link *OB*

$$U_{OB} = -mg_0 z_{c/OB} = -mg_0\left(-L\sin\theta_2\right) = mg_0 L\sin\theta_2 \qquad (3.72)$$

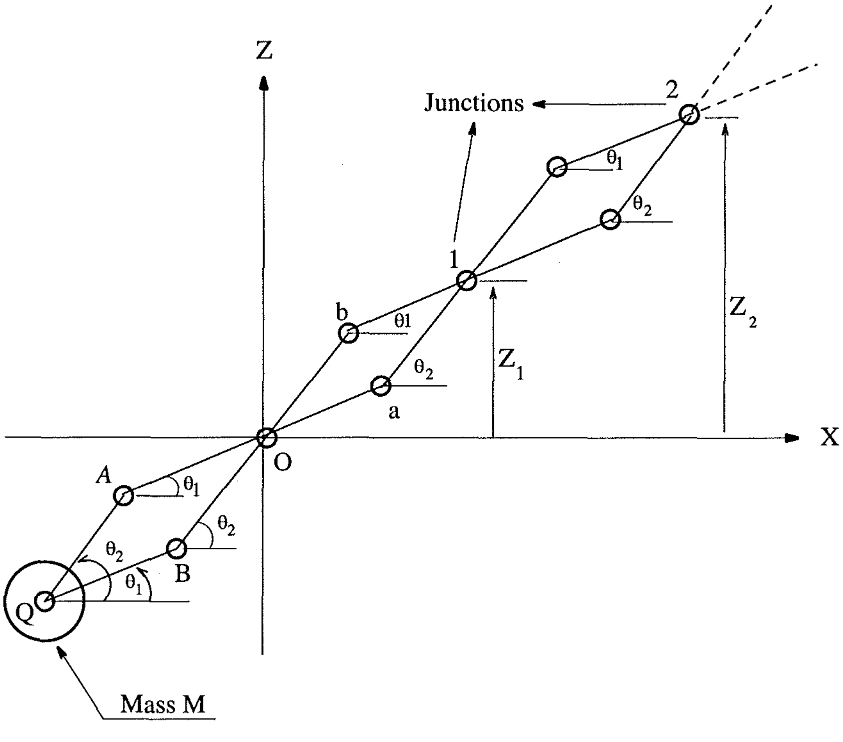


Fig. 3.4: Part of the balanced manipulator.

- Potential energy of link OA

$$U_{OA} = -mg_0\, z_{c/OA} = -mg_0(-L\sin\theta_1) = mg_0 L\sin\theta_1 \tag{3.73}$$

- Potential energy of link AQ

$$\begin{aligned} U_{AQ} &= -mg_0\, z_{c/AQ} \\ &= -mg_0(-2L\sin\theta_1 - L\sin\theta_2) = mg_0 L(2\sin\theta_1 + \sin\theta_2) \end{aligned} \tag{3.74}$$

- Potential energy of link BQ

$$\begin{aligned} U_{BQ} &= -mg_0 z_{c/BQ} \\ &= -mg_0(-2L\sin\theta_2 - L\sin\theta_1) = mg_0 L(2\sin\theta_2 + \sin\theta_1) \end{aligned} \tag{3.75}$$

- Potential energy of the counter mass (M)

$$\begin{aligned} U_M &= -Mg_0 z_Q \\ &= -Mg_0(-2L\sin\theta_1 - 2L\sin\theta_2) = 2Mg_0 L(\sin\theta_1 + \sin\theta_2) \end{aligned} \tag{3.76}$$

From which we have

$$\begin{aligned} U_{cs} &= U_{OB} + U_{OA} + U_{AQ} + U_{BQ} + U_M \\ &= mg_0 L\sin\theta_2 + mg_0 L\sin\theta_1 + mg_0 L(2\sin\theta_1 + \sin\theta_2) \\ &\qquad + mg_0 L(2\sin\theta_2 + \sin\theta_1) + 2Mg_0 L(\sin\theta_2 + \sin\theta_1) \\ &= 2mg_0 L(2+\lambda)(\sin\theta_1 + \sin\theta_2) \end{aligned} \tag{3.77}$$

Where, $\lambda = \dfrac{M}{m}$, and is defined as the mass ratio.

The condition for complete static balancing occurs if, and only if, the potential energy of the whole manipulator, including the counter system, vanishes. Mathematically this means that,

Potential energy (Manipulator) + Potential energy (Counter system) = 0

or, $$U + U_{cs} = 0 \qquad (3.78)$$

Substituting for U and U_{cs}, from equations 3.67 and 3.77 respectively, into equation 3.76 yields,

$$-mg_0 L\phi(\sin\theta_1 + \sin\theta_2) + 2mg_0 L(2+\lambda)(\sin\theta_1 + \sin\theta_2) = 0$$

Solving for ϕ yields,

$$\phi = 2(2+\lambda) \qquad (3.79)$$

Substituting for $\phi = (4n^2 + 8n + 4)$, from section 3.3.2, into the above equation gives the following condition

$$M = 2mn\,(n+2) \qquad (3.80)$$

The required counter-mass for balancing the manipulator, for any number of cells, can be obtained from equation 3.80.

Due to the presence of the counter system the kinetic energy of the manipulator is increased. So, the kinetic energy of the robot must be updated and the new equations of motion should be formulated.

If T_{cs} is defined as being the kinetic energy of the counter system, then

$$T_{cs} = T_{OA} + T_{OB} + T_{AQ} + T_{BQ} + T_M \qquad (3.81)$$

Where, the R.H.S terms are the kinetic energies of links OA, OB, AQ, BQ and the counter mass M , respectively.

$$T_{OA} = \tfrac{1}{2}(I_O)_{OA}\omega_1^2$$

Where,

$$(I_O)_{Oa} = (I_O)_{Ob} = J + mL^2 = 4J$$

This leads to

$$T_{OA} = 2J\omega_1^2 \qquad (3.82)$$

Similarly,

$$T_{OB} = 2J\omega_2^2 \qquad (3.83)$$

From Figure 3.4, it can be deduced that

$$\left(\begin{array}{l} x_{c/AQ} = -2L\cos\theta_1 - L\cos\theta_2 \\ z_{c/AQ} = -2L\sin\theta_1 - L\sin\theta_2 \\ x_{c/BQ} = -2L\cos\theta_2 - L\cos\theta_1 \\ z_{c/BQ} = -2L\sin\theta_2 - L\sin\theta_1 \\ x_Q = -2L\cos\theta_1 - 2L\cos\theta_2 \\ z_Q = -2L\sin\theta_2 - 2L\sin\theta_1 \end{array}\right) \qquad (3.84)$$

Differentiating equation 3.84 with respect to time yields,

$$\left(\begin{aligned} \dot{x}_{c/AQ} &= 2L\omega_1 \sin\theta_1 + L\omega_2 \sin\theta_2 \\ \dot{z}_{c/AQ} &= -2L\omega_1 \cos\theta_1 - L\omega_2 \cos\theta_2 \\ \dot{x}_{c/BQ} &= 2L\omega_2 \sin\theta_2 + L\omega_1 \sin\theta_1 \\ \dot{z}_{c/BQ} &= -2L\omega_2 \cos\theta_2 - L\omega_1 \cos\theta_1 \\ \dot{x}_{Q} &= 2L\omega_1 \sin\theta_1 + 2L\omega_2 \sin\theta_2 \\ \dot{z}_{cQ} &= -2L\omega_2 \cos\theta_2 - 2L\omega_1 \cos\theta_1 \end{aligned}\right) \tag{3.85}$$

By deriving the resultant velocity of the mass centre of each link of the counter cell and that of the counter mass, the following relationships are obtained

$$v_{c/AQ}^2 = L^2\left\{4\omega_1^2 + \omega_2^2 + 4\omega_1\omega_2 \cos(\theta_2 - \theta_1)\right\} \tag{3.86}$$

$$v_{c/BQ}^2 = L^2\left\{4\omega_2^2 + \omega_1^2 + 4\omega_1\omega_2 \cos(\theta_2 - \theta_1)\right\} \tag{3.87}$$

$$v_Q^2 = 4L^2\left\{\omega_1^2 + \omega_2^2 + 2\omega_1\omega_2 \cos(\theta_2 - \theta_1)\right\} \tag{3.88}$$

$$\begin{aligned} T_{AQ} &= \tfrac{1}{2} m v_{c/AQ}^2 + \tfrac{1}{2} J\omega_2^2 \\ &= 2J\left\{3\omega_1^2 + \omega_2^2 + 3\omega_1\omega_2 \cos(\theta_2 - \theta_1)\right\} \end{aligned} \tag{3.89}$$

$$\begin{aligned} T_{BQ} &= \tfrac{1}{2} m v_{c/BQ}^2 + \tfrac{1}{2} J\omega_1^2 \\ &= 2J\left\{3\omega_2^2 + \omega_1^2 + 3\omega_1\omega_2 \cos(\theta_2 - \theta_1)\right\} \end{aligned} \tag{3.90}$$

$$\begin{aligned} T_Q &= \tfrac{1}{2} m v_Q^2 \\ &= 2ML^2\left\{\omega_2^2 + \omega_1^2 + 2\omega_1\omega_2 \cos(\theta_2 - \theta_1)\right\} \end{aligned} \tag{3.91}$$

Substituting for M from equation 3.80 into 3.91 yields,

$$T_Q = 12Jn(n+2)\left\{\omega_2^2 + \omega_1^2 + 2\omega_1\omega_2\cos(\theta_2 - \theta_1)\right\} \qquad (3.92)$$

Now, from equation 3.81,

$$\begin{aligned} T_{cs} &= T_{OA} + T_{OB} + T_{AQ} + T_{BQ} + T_M \\ &= 2J\omega_1^2 + 2J\omega_2^2 + 2J\left\{3\omega_1^2 + \omega_2^2 + 3\omega_1\omega_2\cos(\theta_2 - \theta_1)\right\} \\ &\qquad + 2J\left\{3\omega_2^2 + \omega_1^2 + 3\omega_1\omega_2\cos(\theta_2 - \theta_1)\right\} \\ &\qquad\qquad + 12Jn(n+2)\left\{\omega_2^2 + \omega_1^2 + 2\omega_1\omega_2\cos(\theta_2 - \theta_1)\right\} \end{aligned}$$

This gives,

$$T_{cs} = J\left\{(12n^2 + 24n + 10)\omega_1^2 + (12n^2 + 24n + 10)\omega_2^2 + 12(2n^2 + 4n + 1)\omega_1\omega_2\cos(\theta_2 - \theta_1)\right\} \qquad (3.93)$$

Now, the new total kinetic energy of the manipulator (T_{new}) can be obtained from equations 3.54 and 3.93, as follows,

$$\begin{aligned} T_{new} &= T + T_{cs} \\ &= J\omega_1^2\left\{8n^3 + 24n^2 + 26n + 10\right\} + J\omega_2^2\left\{8n^3 + 24n^2 + 26n + 10\right\} \\ &\qquad + J\omega_1\omega_2\cos(\theta_2 - \theta_1)\left\{16n^3 + 48n^2 + 44n + 12\right\} \\ &\qquad\qquad + J\left\{(12n^2 + 24n + 10)\omega_1^2 + (12n^2 + 24n + 10)\omega_2^2\right\} \\ &\qquad\qquad\qquad + 12J(2n^2 + 4n + 1)\omega_1\omega_2\cos(\theta_2 - \theta_1) \end{aligned}$$

This yields,

$$T_{new} = J\left\{\left(8n^3 + 36n^2 + 50n + 20\right)\omega_1^2 + \left(8n^3 + 36n^2 + 50n + 20\right)\omega_2^2\right\}$$
$$+ J\left(16n^3 + 72n^2 + 92n + 24\right)\omega_1\omega_2\cos\left(\theta_2 - \theta_1\right) \qquad (3.94)$$

Now, letting

$$\tilde{\xi} = 8n^3 + 36n^2 + 50n + 20$$

and,

$$\tilde{\psi} = 16n^3 + 72n^2 + 92n + 24$$

Then,

$$T_{new} = J\left\{\tilde{\xi}\omega_1^2 + \tilde{\xi}\omega_2^2 + \tilde{\psi}\omega_1\omega_2\cos\left(\theta_2 - \theta_1\right)\right\} \qquad (3.95)$$

By comparing equations 3.41 and 3.95, it is clear that both are the same except for $\xi \equiv \tilde{\xi}$ and $\psi \equiv \tilde{\psi}$. So, the new equation of motion is the same as equation 3.52, but using the new symbols. Hence we have

$$\begin{bmatrix} 2\tilde{\xi} & 0 & \tilde{\psi}\cos\Delta x & -\tilde{\psi}x_3\sin\Delta x \\ 0 & 1 & 0 & 0 \\ \tilde{\psi}\cos\Delta x & \tilde{\psi}x_1\sin\Delta x & 2\tilde{\xi} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \frac{\tau_1}{J} \\ x_1 \\ \frac{\tau_2}{J} \\ x_3 \end{bmatrix} \qquad (3.96)$$

3.4 Conclusions

In this chapter the concept of multi-cell planar parallel manipulators has been introduced. The concept allows for an increase in the workspace without affecting the original system structure. With proper reduction of the cell dimensions the manoeuvrability of the whole manipulator can be greatly enhanced. Following on from this, a general dynamic model for the manipulator with any orientation has been derived by using the Lagrange formulation. The correctness of the model has been checked by comparing its one-cell equations of motion with the equations obtained in chapter 2 for the same mechanism using the bond graph method. After that a new method of static balancing has been introduced and which led to the reduction of the number of variables, to be adjusted, to only one parameter.

Chapter 4

Bond Graph Modelling and Analysis of a 2-DOF Planar Redundant-Actuated One-Cell Parallel Manipulator

4.1 Introduction

In addition to the advantages of parallel manipulators the performance of such mechanisms can be further improved by using actuation redundancy. A manipulator with actuation redundancy has more actuators than the number of degrees of freedom which are required to define its dynamics, however this makes the selection of the actuator torques, in order to follow a desired path, indeterminate. Under certain circumstances this feature can lead to an optimisation of the torque distribution among the actuators according to some pre-specified performance specification. In many cases the location of the redundant actuator in the robot appears to be a personal choice. In this context Beiner [16] recommended that it is important to optimise of the location of the redundant actuator in order to enhance its advantage in dynamic applications.

This chapter is dedicated to investigate the effect of the redundant motor location on the performance of a redundant one-cell parallel planar manipulator. The arm is modelled by the bond-graph method using the Model Transformation Tools (MTT) software. The equations of motion are obtained using the hierarchical method discussed in chapter 2. The analysis of the redundant system is performed in the second part of the chapter using the derived dynamic model.

4.2 Hierarchical Bond Graph Modelling of a Redundant One-Cell Parallel Manipulator

In order to optimise the location of a redundant motor in the one-cell parallel manipulator of Figure 4.1, the motors torques associated with all the possible locations of such a redundant actuator for a given task must be calculated and then compared. However, this requires a dynamic model for the manipulator at each location of that motor. To reduce the time and the effort while performing this task it was decided to go for the formulation of a general model which captures all the possible locations of the redundant actuator and then to assign zero values for the parameters of the redundant motors at all locations except the chosen one. Because the locations at joint A and joint C are the same (by symmetry) the dynamic model assumes only two redundant actuators at joint A and B, in addition to the two main motors located at joints O and $\tilde{O}$.

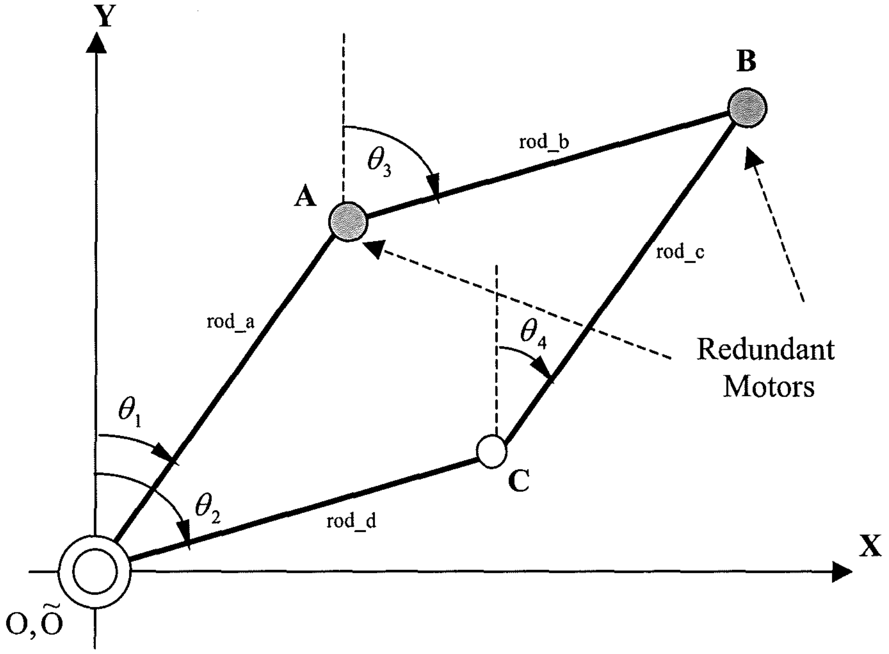


Fig. 4.1: Configuration of the redundant one-cell manipulator.

In the bond graph model of the system, Figure 4.2, the motors at O, Õ, A and B are represented by the effort sources t_I, t_II, f_2 and f_3, respectively. The masses of the redundant motors at joints A and B are represented by the inertial elements m_xx & m_yy and m_x & m_y, respectively.

It is important to mention that in the model the mass of the redundant motor between links AB and CB also includes the payload mass, and the dummy forces (t_vx and t_vy) include the external forces applied to the end-effector of the arm. The rest of the graph is similar to the bond graph model of the normal parallel one-cell manipulator discussed in chapter 2. The label file for this particular model and the generated representations such as the simulation code, list of inputs etc. are shown in Appendix (D). It is important to note that the list of states in Appendix (D) is a Matlab file, meaning that it is structurally different from those in Appendices (B) and (C). This is an additional representation by the MTT program as it was upgraded whilst this particular work was underway.

To get the raw equations of motion the simulation code is transformed to the following equations

$$k_1\dot{x}_1 + 6k_2 \cos\Delta x\, \dot{x}_3 + 6k_2 x_3 \sin\Delta x\, \dot{x}_4 = \tau_1 - \tau_3 + 2L\left(f_y \cos x_2 - f_x \sin x_2\right) \tag{4.1}$$

$$\dot{x}_2 = \frac{x_1}{J} \tag{4.2}$$

$$6k_2 \cos\Delta x\, \dot{x}_1 + 4k_3\, \dot{x}_3 - 6k_2 x_1 \sin\Delta x\, \dot{x}_2 = \tau_3 - \tau_4 + 2L\left(f_y \cos x_4 - f_x \sin x_4\right) \tag{4.3}$$

$$\dot{x}_4 = \frac{x_3}{J} \tag{4.4}$$

$$4\dot{x}_5 + 6\cos\hat{\Delta x}\, \dot{x}_7 + 6x_7 \sin\hat{\Delta x}\, \dot{x}_8 = \tau_4 + 2L\left(f_y \cos x_6 - f_x \sin x_6\right) \tag{4.5}$$

$$\dot{x}_6 = \frac{x_5}{J} \tag{4.6}$$

$$6\cos\hat{\Delta x}\, \dot{x}_5 + 16\,\dot{x}_7 - 6x_5 \sin\hat{\Delta x}\, \dot{x}_6 = \tau_2 + 2L\left(f_y \cos x_8 - f_x \sin x_8\right) \tag{4.7}$$

$$\dot{x}_8 = \frac{x_7}{J} \tag{4.8}$$

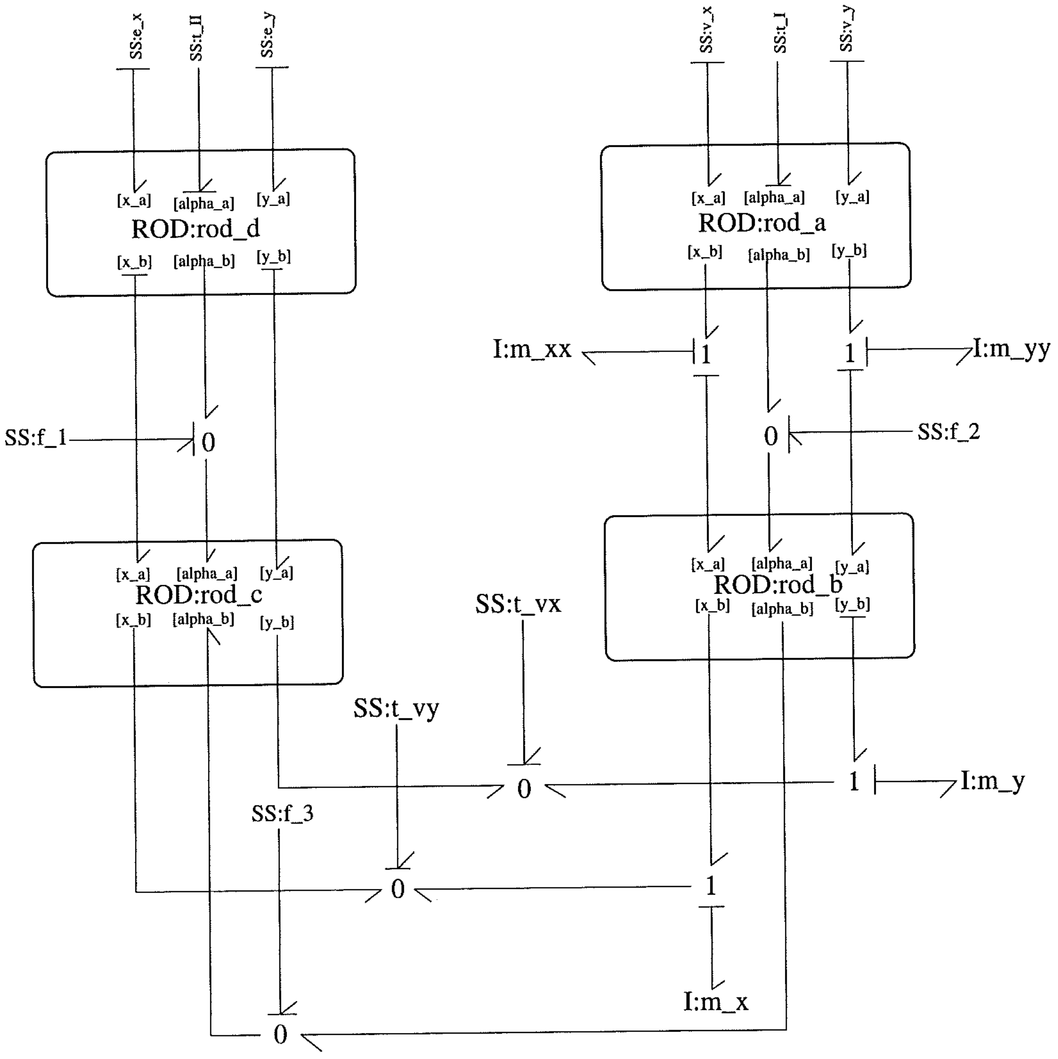


Fig. 4.2: The hierarchical bond-graph model for the redundant parallel manipulator.

Where,

$$k_1 = 16 + \frac{12}{m}\left(m_m + m_p + m_n\right)$$

$$k_2 = \frac{2\left(m_m + m_p\right)}{m} + 1$$

$$k_3 = \frac{3\left(m_m + m_p\right)}{m} + 1$$

$$\Delta x = x_2 - x_4 \quad \text{and} \quad \hat{\Delta} x = x_6 - x_8$$

and,

m_m : is the mass of the redundant motor at joint B.

m_n : is the mass of the redundant motor at joint A.

m_p : is the payload mass.

f_x: is the applied external force in the direction of the X co-ordinate.

f_y: is the applied external force in the direction of the Y co-ordinate.

m : is the link mass.

L : is the link length.

J : is the polar moment of inertia about the centre of mass of the link.

τ_1: is the torque of the motor located at joint O.

τ_2: is the torque of the motor located at joint $\tilde{\text{O}}$.

τ_3: is the torque of the redundant motor at joint A.

τ_4: is the torque of the redundant motor at joint B.

The solution of the constraints from chapter 2 gives,

$$x_6 = x_2,\ x_8 = x_4,\ x_5 = x_1,\ x_7 = x_3 \text{ and } \hat{\Delta} x = \Delta x \qquad (4.9)$$

Differentiating these with respect to time yields,

$$\dot{x}_6 = \dot{x}_2,\ \dot{x}_8 = \dot{x}_4,\ \dot{x}_5 = \dot{x}_1 \text{ and } \dot{x}_7 = \dot{x}_3 \qquad (4.10)$$

Substituting from equations 4.9 and 4.10 into equations 4.1- 4.8 leads to

$$k_1\dot{x}_1 + 6k_2\cos\Delta x\,\dot{x}_3 + 6k_2x_3\sin\Delta x\,\dot{x}_4 = \tau_1 - \tau_3 + 2L(f_y\cos x_2 - f_x\sin x_2) \tag{4.11}$$

$$\dot{x}_2 = \frac{x_1}{J} \tag{4.12}$$

$$6k_2\cos\Delta x\,\dot{x}_1 + 4k_3\,\dot{x}_3 - 6k_2x_1\sin\Delta x\,\dot{x}_2 = \tau_3 - \tau_4 + 2L(f_y\cos x_4 - f_x\sin x_4) \tag{4.13}$$

$$\dot{x}_4 = \frac{x_3}{J} \tag{4.14}$$

$$4\dot{x}_1 + 6\cos\Delta x\,\dot{x}_3 + 6x_3\sin\Delta x\,\dot{x}_4 = \tau_4 + 2L(f_y\cos x_2 - f_x\sin x_2) \tag{4.15}$$

$$\dot{x}_2 = \frac{x_1}{J} \tag{4.16}$$

$$6\cos\Delta x\,\dot{x}_1 + 16\,\dot{x}_3 - 6x_1\sin\Delta x\,\dot{x}_2 = \tau_2 + 2L(f_y\cos x_4 - f_x\sin x_4) \tag{4.17}$$

$$\dot{x}_4 = \frac{x_3}{J} \tag{4.18}$$

Removing the repeated equations and then adding equations 4.11 and 4.15 yields,

$$(k_1 + 4)\,\dot{x}_1 + 6(k_2 + 1)\cos\Delta x\,\dot{x}_2 + 6(k_2 + 1)x_3\sin\Delta x\,\dot{x}_4 =$$
$$\tau_1 + \tau_4 - \tau_3 + 4L(f_y\cos x_2 - f_x\sin x_2) \tag{4.19}$$

Adding equations 4.13 and 4.17 gives,

$$6(k_2 + 1)\cos\Delta x\,\dot{x}_1 + 4(k_3 + 4)\,\dot{x}_3 - 6(k_2 + 1)x_1\sin\Delta x\,\dot{x}_2 =$$
$$\tau_2 + \tau_3 - \tau_4 + 4L(f_y\cos x_4 - f_x\sin x_4) \tag{4.20}$$

Equations 4.19, 4.12, 4.20 and 4.14 are the reduced state equations of motion of the redundant manipulator.

The state vector is defined by

$$x_1 = J\dot{\theta}_1, x_2 = \theta_1, x_3 = J\dot{\theta}_2 \text{ and } x_4 = \theta_2 \tag{4.21}$$

Now, in order to calculate the motor torques for a given task the states of the system must enter the equations as parameters, however this requires the system equations to be written in the joint space as follows,

Substituting from 4.21 into equations 4.19 and 4.20 and rearranging yields,

$$\tau_1 + \tau_4 - \tau_3 = J(k_1 + 4)\ddot{\theta}_1 + 6J(k_2 + 1)\cos(\theta_1 - \theta_2)\ddot{\theta}_2 \\ + 6J(k_2 + 1)\sin(\theta_1 - \theta_2)\dot{\theta}_2^2 - 4L(f_y \cos\theta_1 - f_x \sin\theta_1) \qquad (4.22)$$

$$\tau_2 + \tau_3 - \tau_4 = 6J(k_2 + 1)\cos(\theta_1 - \theta_2)\ddot{\theta}_1 + 4J(k_3 + 4)\ddot{\theta}_2 \\ - 6J(k_2 + 1)\sin(\theta_1 - \theta_2)\dot{\theta}_1^2 - 4L(f_y \cos\theta_2 - f_x \sin\theta_2) \qquad (4.23)$$

Now, letting

$$u_1 = \tau_1 + \tau_4 - \tau_3 \qquad (4.24)$$

$$u_2 = \tau_2 + \tau_3 - \tau_4 \qquad (4.25)$$

We have two different cases as follows,

- Redundant motor at joint A ($\tau_4 = 0$)

Substituting for $\tau_4 = 0$ into equations 4.24 and 4.25 gives two equations in three unknowns. To solve for these three unknowns we use the pseudo-inverse approach method mentioned in reference [16]. It is important to note that this approach gives the unconstrained solution of equations 4.24 and 4.25 for the three unknowns.

Writing equations 4.24 and 4.25 in matrix form yields,

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

This leads to,

$$\tau_1 = \frac{2u_1 + u_2}{3} \qquad (4.26)$$

$$\tau_2 = \frac{u_1 + 2u_2}{3} \qquad (4.27)$$

$$\tau_3 = \frac{u_2 - u_1}{3} \qquad (4.28)$$

- Redundant motor at joint B ($\tau_3 = 0$)

Similarly, substituting for $\tau_3 = 0$ into equations 4.24 and 4.25 and writing them in matrix form yields,

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

This leads to,

$$\tau_1 = \frac{2u_1 + u_2}{3} \qquad (4.29)$$

$$\tau_2 = \frac{u_1 + 2u_2}{3} \qquad (4.30)$$

$$\tau_4 = \frac{u_1 - u_2}{3} \qquad (4.31)$$

The manipulator is forced to follow the following third order joint-space trajectory,

$$\left(\begin{array}{l} \theta_1(t) = 0.1745 + 1.0472\,t^2 - 0.349\,t^3 \\ \dot{\theta}_1(t) = 2.0944\,t - 1.047\,t^2 \\ \ddot{\theta}_1(t) = 2.0944 - 2.094\,t \\ \theta_2(t) = 0.7854 + 0.9818\,t^2 - 0.3273\,t^3 \\ \dot{\theta}_2(t) = 1.9635\,t - 0.9818\,t^2 \\ \ddot{\theta}_2(t) = 1.9635 - 1.9635\,t \end{array} \right) \qquad (4.32)$$

The initial and the final conditions are

$$\theta_1(0)=\tfrac{\pi}{18}\,(\text{rad}),\ \theta_2(0)=\tfrac{\pi}{4}\,(\text{rad}),\ \theta_1(f)=\tfrac{\pi}{2}\,(\text{rad})\ \text{and}\ \ \theta_2(f)=\tfrac{2\pi}{3}\,(\text{rad})$$

The manipulator is assumed to possess the mechanical parameter values given in Table 4.1 below. In the analysis the arm is simulated for two different redundant motor masses (Case I and Case II) and for different payloads and external forces. The simulation tests are summarised in Table 4.2. The simulations are performed inside Matlab 5.3 using script files (Appendix E).

In each test, and for each of the two simulation cases (Case I and Case II), the sum of the absolute values of the motors torques for the case where the redundant motor is at joint A is compared to its counterpart when the redundant motor is located at joint B. The simulation results obtained are shown in Figures 4.3 to 4.18.

4.3 Discussion

The results show that the location of the redundant motor only has a considerable effect on the robot performance when the payload mass is small. In this case it is preferable to locate the redundant motor near the robot-base in order to reduce the power consumption while performing tasks. Figures 4.3-4.4 of case I and Figures 4.11-4.12 of case II clarify this fact by showing that the value of the total torque is smaller in the case where the redundant motor is at joint A than when the motor is at joint B. As the payload mass gradually increases the results reveal that the effect of the location of the redundant motor decreases and this is quite obvious from Figures 4.5 and 4.13 depicting case I and case II, respectively.

In applications where the arm is in contact with its environment by applying forces the analysis shows that the effect of the redundant motor position is nearly negligible. The results of tests 4-7 for both cases (case I and case II) prove the previous point. It is important to refer here to the fact that when the arm is carrying payloads, together with also applying forces, the same result is obtained. Clearly, Figures 4.10 for case I and Figure 4.18 for case II both explain this.

From the above results, and the fact that most of the robot applications involve contact with the environment, it can be concluded that the location of the redundant motor is not important and its position should be decided upon only by the practical design considerations required for the manipulator.

Parameter	Value
L	0.125 (m)
m	0.31 (kg)
J	0.0004 kg.m^2
m_n	0.44 kg (Case I) 0.88 kg (Case II)
m_m	0.44 and 0.88 (kg)

Table 4.1: Manipulator parameters for simulations.

Test	Payload (m_p) (kg)	f_x (N)	f_y (N)
1	0	0	0
2	0.5	0	0
3	1	0	0
4	0	5	0
5	0	10	0
6	0	0	5
7	0	0	10
8	0.5	5	5

Table 4.2: Simulation tests for the manipulator.

- Case I ($m_n, m_m = 0.44\,\text{kg}$)

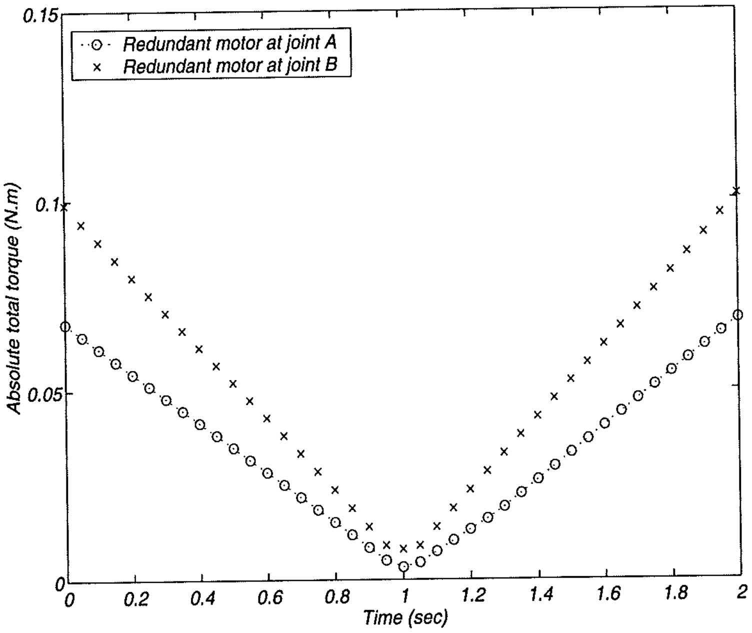


Fig. 4.3: Test 1 ($m_p = 0\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 0\,\text{N}$).

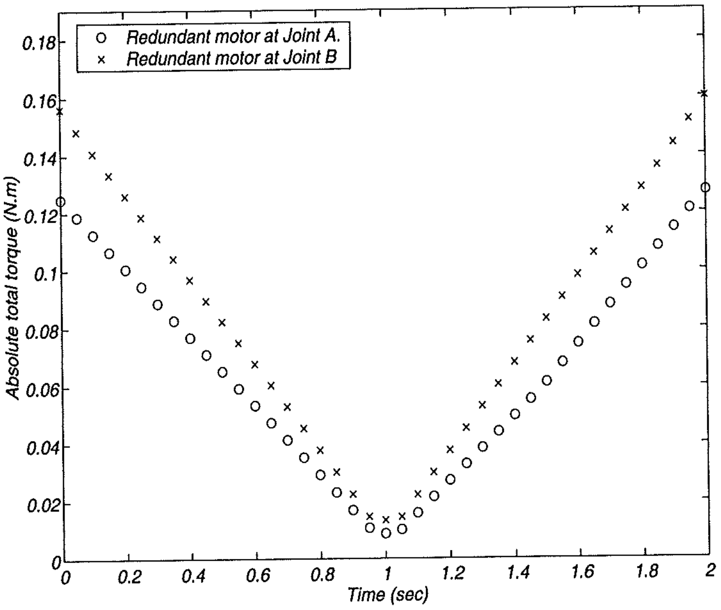


Fig. 4.4: Test 2 ($m_p = 0.5\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 0\,\text{N}$).

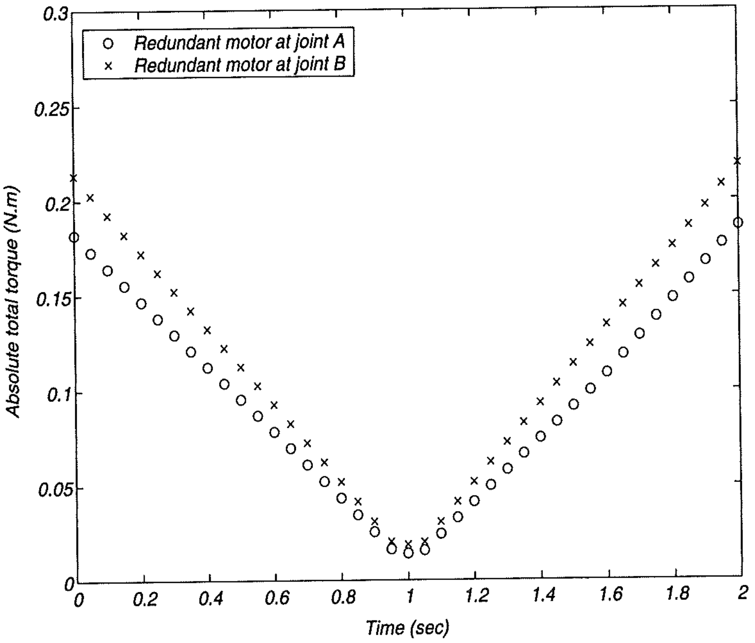


Fig. 4.5: Test 3 ($m_p = 1\,\mathrm{kg}, f_x = 0\,\mathrm{N}$ and $f_y = 0\,\mathrm{N}$).

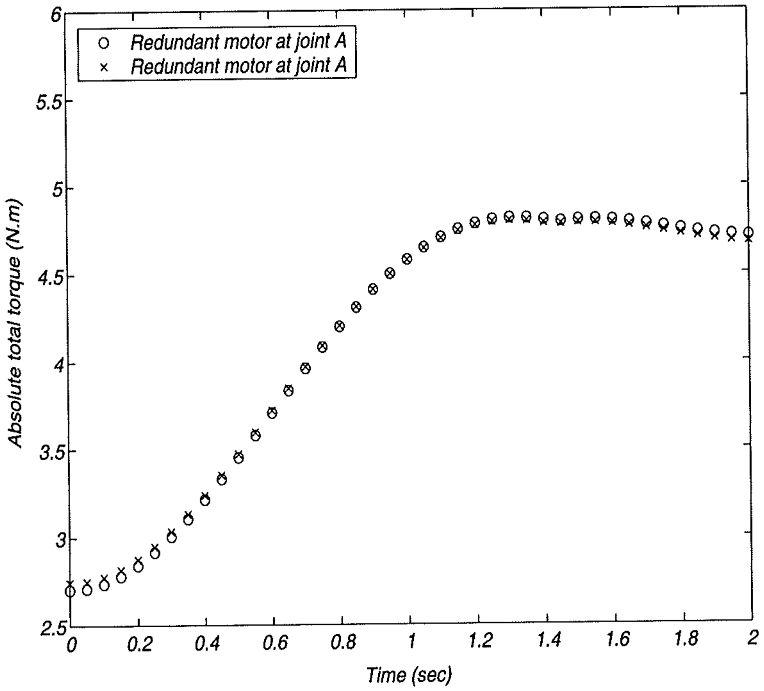


Fig. 4.6: Test 4 ($m_p = 0\,\mathrm{kg}, f_x = 5\,\mathrm{N}$ and $f_y = 0\,\mathrm{N}$).

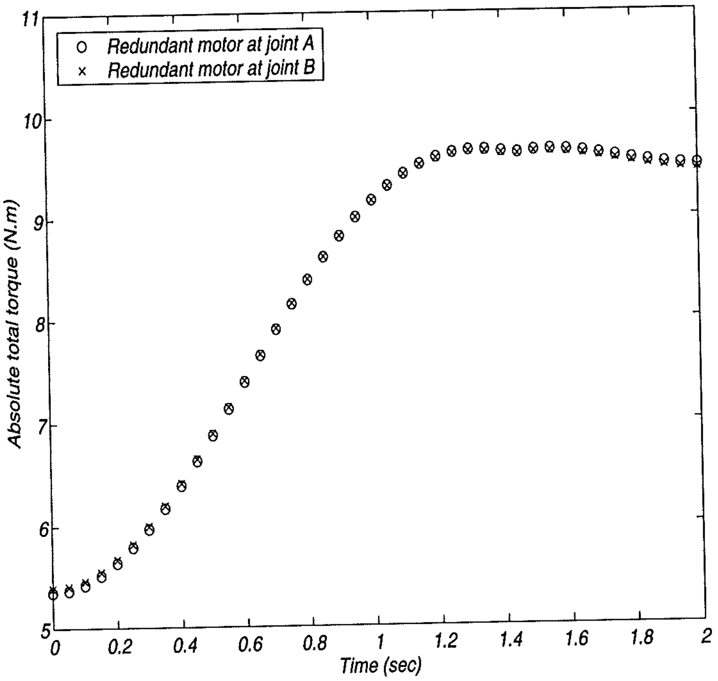


Fig. 4.7: Test 5 ($m_p = 0\,\text{kg}, f_x = 10\,\text{N}$ and $f_y = 0\,\text{N}$).

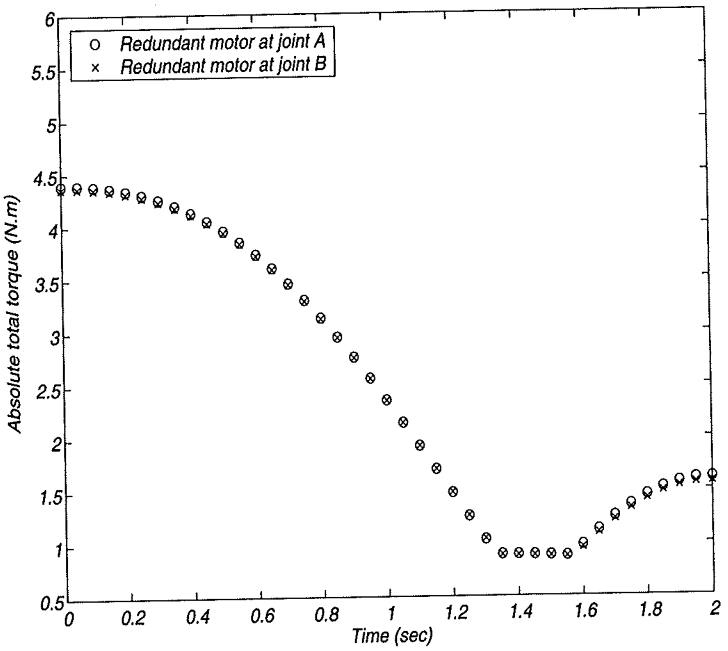


Fig. 4.8: Test 6 ($m_p = 0\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 5\,\text{N}$).

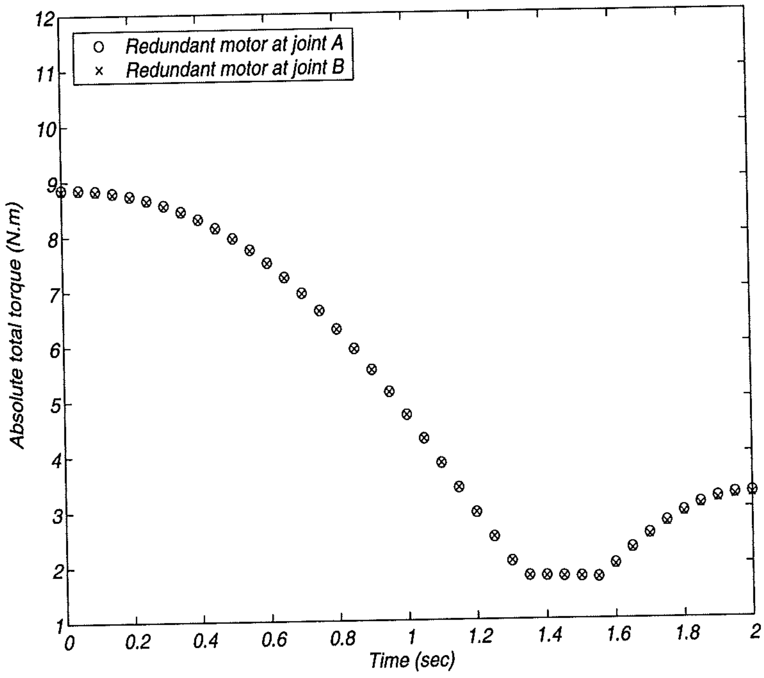


Fig. 4.9: Test 7 ($m_p = 0\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 10\,\text{N}$).

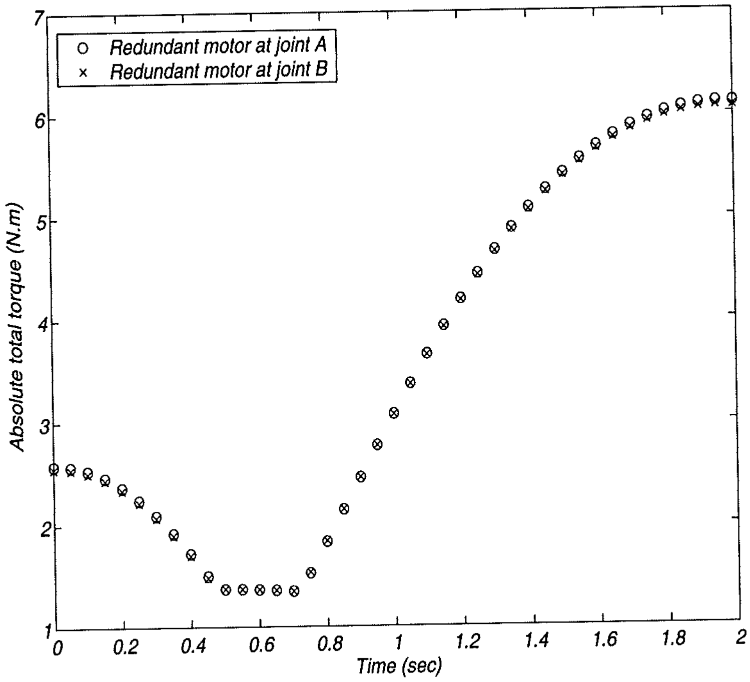


Fig. 4.10: Test 8 ($m_p = 0.5\,\text{kg}, f_x = 5\,\text{N}$ and $f_y = 5\,\text{N}$).

- Case II ($m_n, m_m = 0.88\,\text{kg}$)

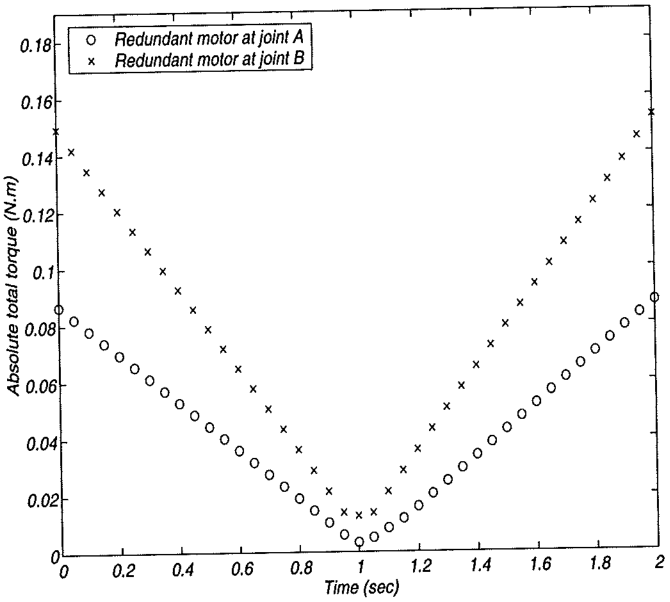


Fig. 4.11: Test 1 ($m_p = 0\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 0\,\text{N}$).

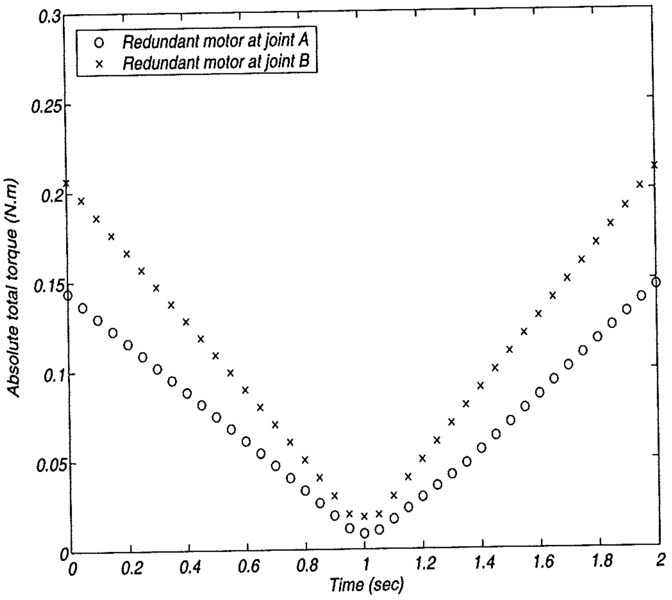


Fig. 4.12: Test 2 ($m_p = 0.5\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 0\,\text{N}$).

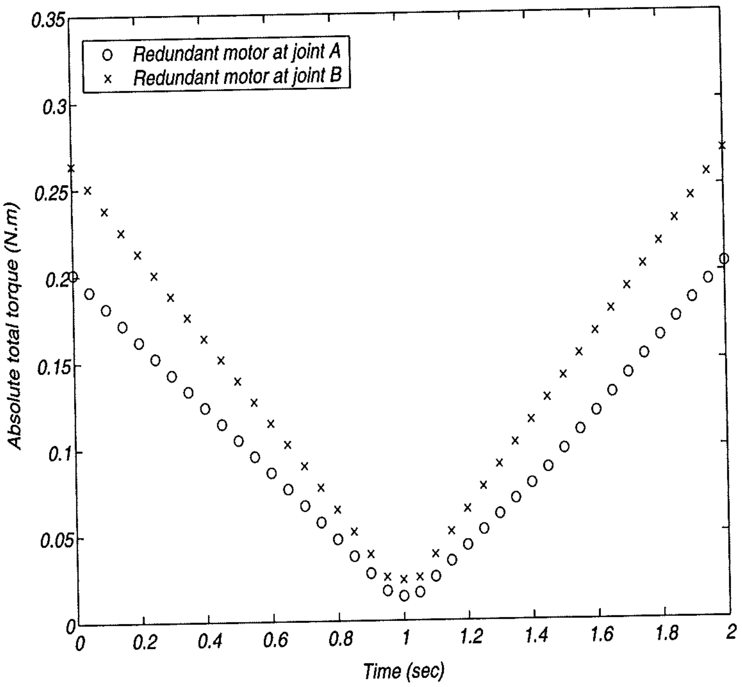


Fig. 4.13: Test 3 ($m_p = 1\,\mathrm{kg}, f_x = 0\,\mathrm{N}$ and $f_y = 0\,\mathrm{N}$).

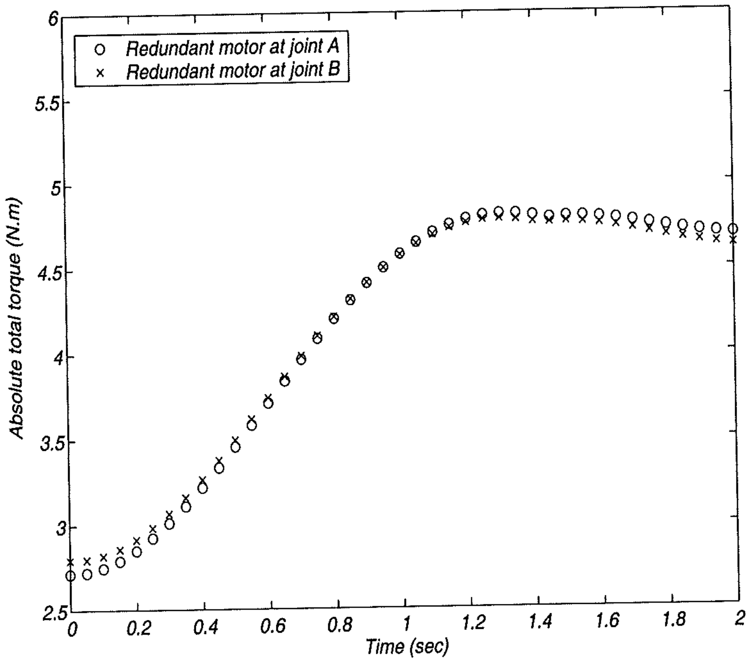


Fig. 4.14: Test 4 ($m_p = 0\,\mathrm{kg}, f_x = 5\,\mathrm{N}$ and $f_y = 0\,\mathrm{N}$).

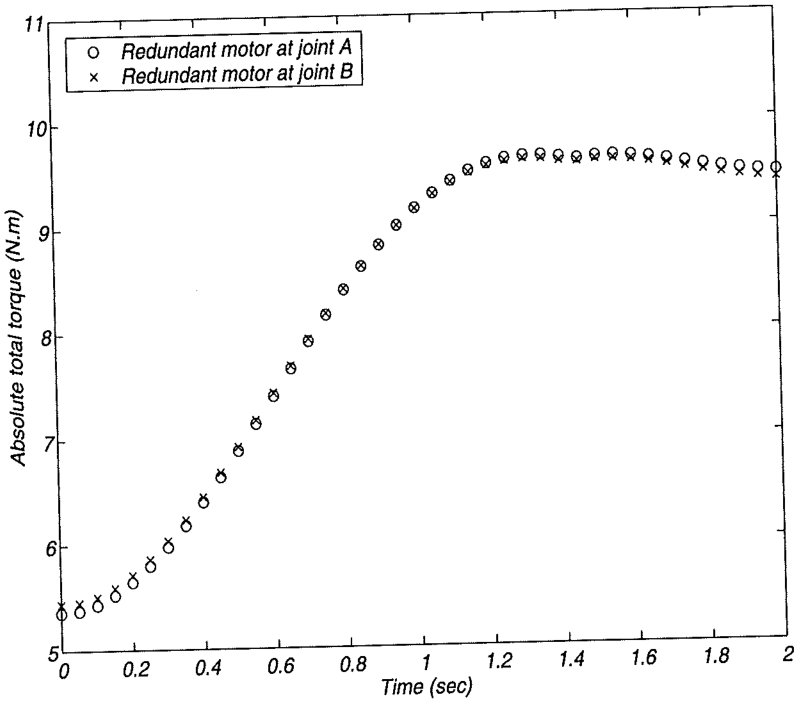


Fig. 4.15: Test 5 ($m_p = 0$ kg, $f_x = 10$ N and $f_y = 0$ N).

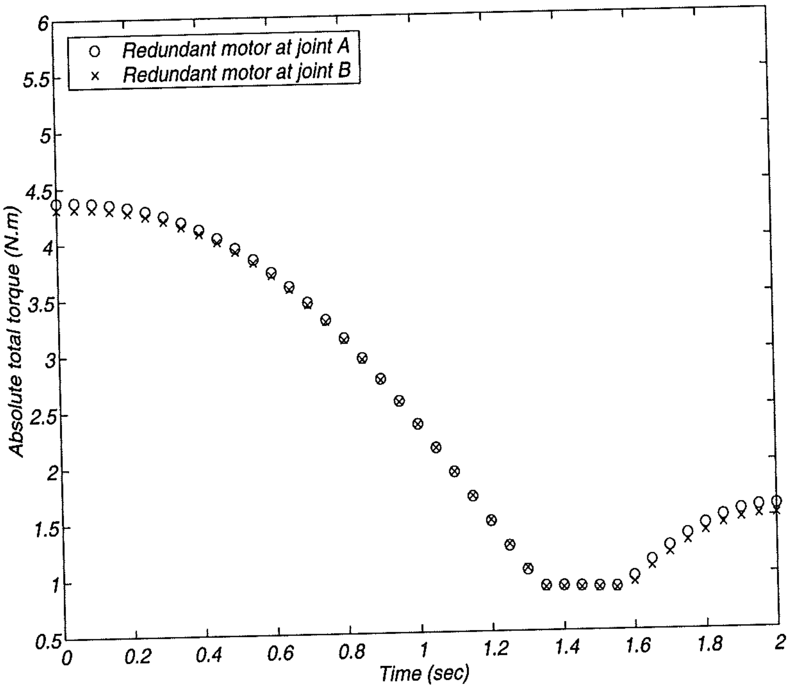


Fig. 4.16: Test 6 ($m_p = 0$ kg, $f_x = 0$ N and $f_y = 5$ N).

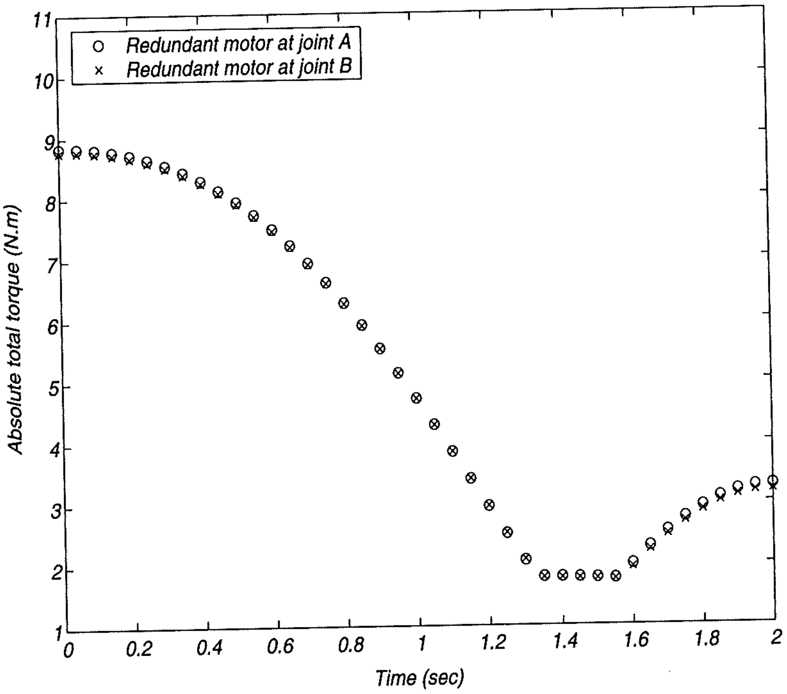


Fig. 4.17: Test 7 ($m_p = 0\,\text{kg}, f_x = 0\,\text{N}$ and $f_y = 10\,\text{N}$).

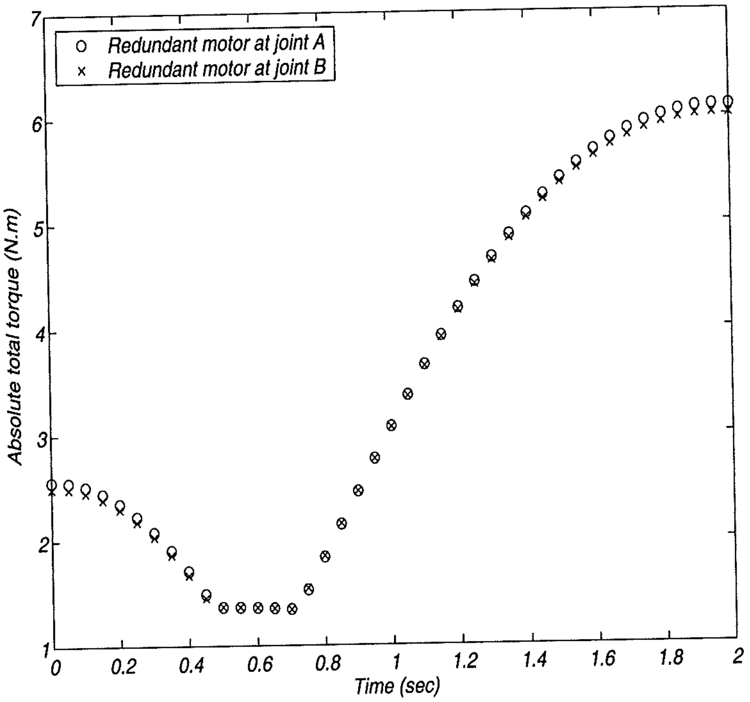


Fig. 4.18: Test 8 ($m_p = 0.5\,\text{kg}, f_x = 5\,\text{N}$ and $f_y = 5\,\text{N}$).

4.4 Conclusions

In this chapter the redundant one-cell, 2-DOF, parallel planar manipulator has been modelled using the bond-graph method via the Model Transformation Tools software (MTT). The developed method of chapter 2 has been used to perform the modelling process and to the author's knowledge it is the first time the bond-graph technique has been used in modelling redundant manipulators. The model represents the arm when carrying more than one redundant manipulator, and hence it is a general model. In addition to this the effect of the location of the redundant motor on the performance of the manipulator has also been analysed. The conclusion to the analysis recommends inserting the redundant motor as near the base as possible if the manipulator is designed only for the normal pick and place tasks, and the payloads are small. If the manipulator is designed for applications that need contact with the environment such as, machining operations, then the position of the redundant motor is not important and it can be put in any suitable location.

Chapter 5

SEPA-ROBOT: A Serial-Parallel Manipulator with Singularity-Based Design

5.1 Introduction

Despite the advantages offered by parallel manipulators such as high rigidity, high precision, high load capacity etc., extensive research in the area has led to the conclusion that such structures are problematic due to issues of poor workspace and having many singular configurations. Although the workspace issue can be manipulated and can be optimised in several ways such as the method discussed in chapter 3, the singularity issue cannot be avoided due to the structural properties of these robots. It is important to mention that this issue must be analysed for any parallel robot because of the fact that at singular configurations the control of the manipulator is inevitably lost. In fact the prevailing opinion in the research literature appears to support the idea of singularity avoidance in both path planning and control, as well as in the mechanical design of such robots (see for example [39] [40]). On the other hand, although serial robots suffer from many drawbacks such as low rigidity, low load capacity, low precision and etc., these manipulators have the characteristics of large workspace and high manoeuvrability. It is important to note that these two properties are very important for any manipulator design.

From the above facts it can be concluded that there are two contradictory manipulator designs. The parallel design avoids the drawbacks of the serial design but suffers from the lack of advantages of the serial design which are of great concern to the versatility of the robot manipulator. Due to this the work here is directed towards the development of a new manipulator which can work in different operating modes (designs). It was decided to call the manipulator "SEPA-ROBOT" because the manipulator can work as a serial robot and as a parallel robot. In addition to these two modes, the manipulator can also work as a redundant actuated parallel robot. The manipulator design here is based on singularity, meaning that the work here is rather different to that presented by other researchers. It is important to note that this design technique can open the door to building new manipulators or mechanical systems which apply the same idea. This, in fact, makes the controller design a challenging matter because there are no specific dynamics for such systems.

In addition to the above contributions the work also generalises the idea to include the design of new manipulators, defined by the term "Parallel-Parallel Robots". In these manipulators the contribution of actuation redundancy, when applied, is maximum.

The first part of the chapter is directed towards introducing the issue of singularity in the planar parallel manipulators via the analysis of the manipulator under investigation. The second part explains the manipulator design and its operation. Following on from this, and in order to evaluate the performance of the new design, three dynamic models representing the different working modes are formulated using the standard Lagrange method. After that the performance of the different modes are examined via simulation. Following this the generalisation of the idea of singularity-based design is presented together with design examples.

5.2 Singularity in Parallel Robots

In the case of the design proposed here the discussion on singularity is restricted to the mechanism shown in Figure 5.1, the one-cell manipulator of chapter 2, which is the core of the new manipulator. The mechanism has two DOF with all joints being rotary. The two main actuators are located at joints A and B.

From Figure 5.1 the position co-ordinates of the gripper are as follows,

$$x_p = 2L\cos\theta_1 + 2L\cos\theta_4 \tag{5.1}$$

$$y_p = 2L\sin\theta_1 + 2L\sin\theta_4 \tag{5.2}$$

Differentiating these with respect to time yields,

$$\dot{x}_p = -2L\sin\theta_1\,\dot{\theta}_1 - 2L\sin\theta_4\,\dot{\theta}_4 \tag{5.3}$$

$$\dot{y}_p = 2L\cos\theta_1\,\dot{\theta}_1 + 2L\cos\theta_4\,\dot{\theta}_4 \tag{5.4}$$

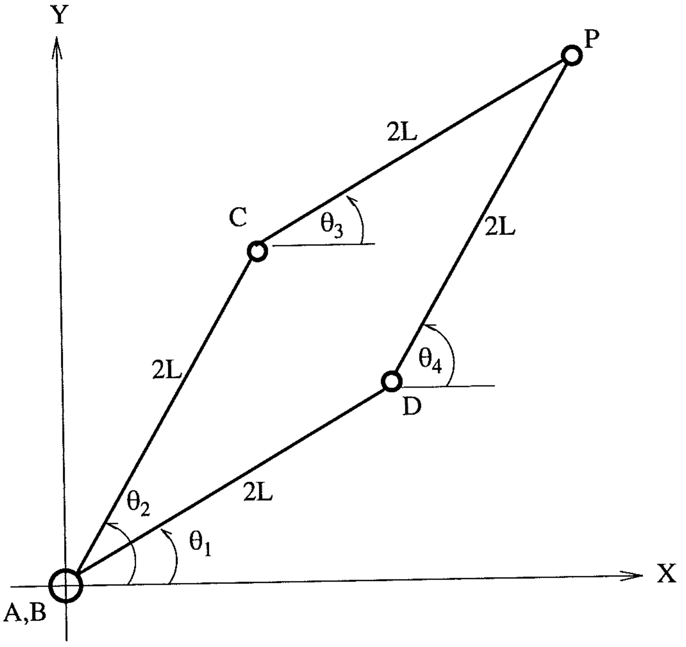


Fig. 5.1: Configuration of one-cell parallel manipulator.

Equations 5.3 and 5.4 can be written as follows,

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = \begin{bmatrix} -2L\sin\theta_1 \\ 2L\cos\theta_1 \end{bmatrix} \dot{\theta}_1 + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2L\cos\theta_4 \\ 2L\sin\theta_4 \end{bmatrix} \dot{\theta}_4 \qquad (5.5)$$

On the other hand, the position co-ordinates of the endpoint can also be defined by

$$x_p = 2L\cos\theta_2 + 2L\cos\theta_3 \qquad (5.6)$$

$$y_p = 2L\sin\theta_2 + 2L\sin\theta_3 \qquad (5.7)$$

Differentiating equations 5.6 and 5.7 with respect to time yields,

$$\dot{x}_p = -2L\sin\theta_2\,\dot{\theta}_2 - 2L\sin\theta_3\,\dot{\theta}_3 \qquad (5.8)$$

$$\dot{y}_p = 2L\cos\theta_2\,\dot{\theta}_2 + 2L\cos\theta_3\,\dot{\theta}_3 \qquad (5.9)$$

Equations 5.8 and 5.9 can be written as follows,

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = \begin{bmatrix} -2L\sin\theta_2 \\ 2L\cos\theta_2 \end{bmatrix} \dot{\theta}_2 + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2L\cos\theta_3 \\ 2L\sin\theta_3 \end{bmatrix} \dot{\theta}_3 \qquad (5.10)$$

To remove both the two idle (unactuated) joint velocities, that is $\dot{\theta}_3$ and $\dot{\theta}_4$, from equations 5.5 and 5.10 we pre-multiply both sides of these equations by $[2L\cos\theta_4 \quad 2L\sin\theta_4]$ and $[2\mathrm{L}\cos\theta_3 \quad 2L\sin\theta_3]$, respectively. Performing this yields the following two equations,

$$[2L\cos\theta_4 \quad 2L\sin\theta_4] \begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = [2L\cos\theta_4 \quad 2L\sin\theta_4] \begin{bmatrix} -2L\sin\theta_1 \\ 2L\cos\theta_1 \end{bmatrix} \dot{\theta}_1 \qquad (5.11)$$

$$\begin{bmatrix}2L\cos\theta_3 & 2L\sin\theta_3\end{bmatrix}\begin{bmatrix}\dot{x}_p \\ \dot{y}_p\end{bmatrix} = \begin{bmatrix}2L\cos\theta_3 & 2L\sin\theta_3\end{bmatrix}\begin{bmatrix}-2L\sin\theta_2 \\ 2L\cos\theta_2\end{bmatrix}\dot{\theta}_2 \qquad (5.12)$$

Now, combining equations 5.11 and 5.12 leads to

$$\begin{bmatrix}2L\cos\theta_4 & 2L\sin\theta_4 \\ 2L\cos\theta_3 & 2L\sin\theta_3\end{bmatrix}\begin{bmatrix}\dot{x}_p \\ \dot{y}_p\end{bmatrix} = \begin{bmatrix}4L^2\sin(\theta_4-\theta_1) & 0 \\ 0 & 4L^2\sin(\theta_3-\theta_2)\end{bmatrix}\begin{bmatrix}\dot{\theta}_1 \\ \dot{\theta}_2\end{bmatrix} \qquad (5.13)$$

Equation 5.13 is called the velocity equation of the manipulator and it takes the form

$$A\dot{P} = B\dot{\theta}$$

Where

$\dot{P}$: is the Cartesian velocity vector of the gripper.

$\dot{\theta}$: is the vector of the actuated joint velocities.

A : is the direct kinematic matrix.

B : is the inverse kinematic matrix.

The summary of the results in reference [34] and reference [36] states that singularity occurs whenever the determinant of A or B , or both, vanishes. When $\det(B) = 0$ the manipulator is in a serial singularity and there is a direction along which no Cartesian velocity can be produced. The gripper loses one or more DOF and it can resist one or more forces or moments without exerting any torque or force at the actuated joints. On the other hand when $\det(A) = 0$ the manipulator is in a parallel singularity and the manipulator becomes uncontrollable (the gripper of the manipulator gains one or more DOF and it cannot resist the forces or moments from one or more directions even when all actuated joints are locked). The last case occurs only when the parameters of the manipulator satisfy certain special conditions and in this case a finite motion of the gripper is possible even if the actuated joints are locked, or alternatively a finite motion of the actuated joints produces no motion of the gripper [34].

For the case discussed here the determinants of A and B are as follows,

$$\det(A) = 4L^2 \sin(\theta_3 - \theta_4) \qquad (5.14)$$

$$\det(B) = 16L^4 \sin(\theta_4 - \theta_1)\sin(\theta_3 - \theta_2) \qquad (5.15)$$

Recalling that the solution of the constraints of the mechanism of Figure 5.1 is defined by,

$$\theta_4 = \theta_2 \text{ and } \theta_3 = \theta_1$$

Then, from equations 5.14 and 5.15, we have

$$\det(A) = -4L^2 \sin(\theta_2 - \theta_1) \qquad (5.16)$$

$$\det(B) = -16L^4 \sin^2(\theta_2 - \theta_1) \qquad (5.17)$$

This means that both the determinants of A and B vanish for the same condition, which is $\theta_2 - \theta_1 = k\pi$ where $k = 0,1$ (assuming restrictions on θ_2 and θ_1). For $k = 0$, the manipulator is fully extended, as shown in Figure 5.2, and it demonstrates the definitions of both serial and parallel singularity. However, when $k = 1$ the manipulator demonstrates also both singularities, but in this case the end-effector (point P) overlaps the origin at point A and B, as shown in Figure 5.3.

5.3 Manipulator design

In the first singular configuration ($k = 0$) the gripper cannot resist any angular moment. Therefore if an additional motor is placed at joint C or D, any torque produced by this motor will rotate the two links DP and CP together. This case simulates the second link of a 2-DOF serial robot. If we also let the two main motors at A and B have the same control signal (i.e. the same torque) the links AD and BC will also rotate together as if they are the first link of the serial robot. One possible design for the manipulator is shown in Figure 5.4. In this case the arm moves in the horizontal plane and the system is quite similar to the SCARA robot. It is important to refer here to the fact that the necessary condition for the manipulator to initiate this serial mode is to bring the original manipulator to the singular configuration of Figure 5.2.

However, if the manipulator is to be operated in the redundant parallel mode the mechanism should start at a configuration other than those of Figures 5.2 and 5.3. Additionally the three actuators at A, B and C or D should have control signals (torques) that can be calculated according to the results in [16] and [42]. On the other hand if it is required to operate the manipulator in the normal parallel mode then the third motor at C or D can be removed or neglected and treated as a concentrated mass. In addition the manipulator should initiate at the same configuration as the previous mode.

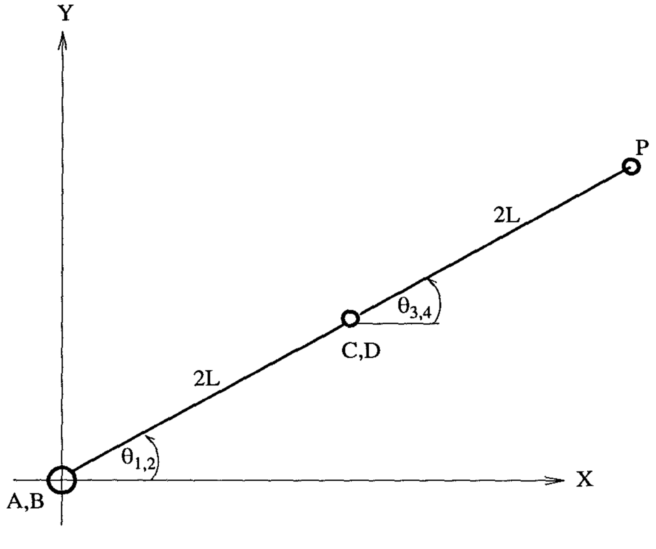


Fig. 5.2: First singular configuration.

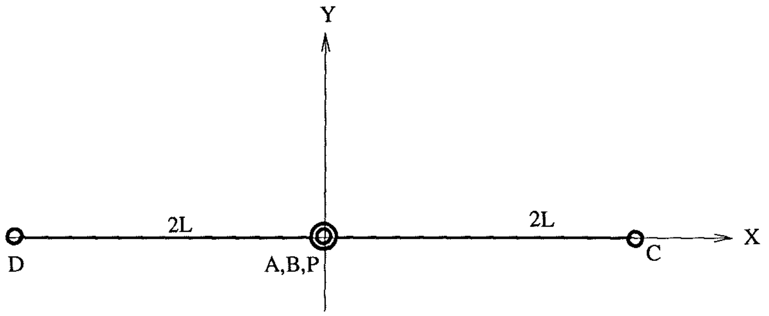


Fig. 5.3: Second singular configuration.

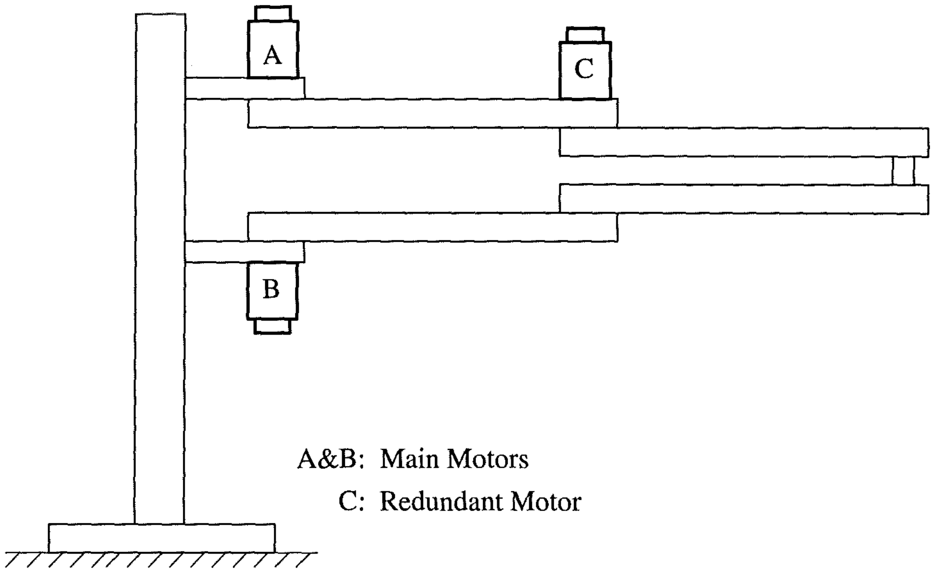


Fig. 5.4: A possible design for the new manipulator.

5.4 Dynamics of Working Modes

Since the mechanical structure of the arm changes for the different modes the dynamics of the arm necessarily differ in each case. Thus, there is no single dynamic model that represents the arm. In order to analyse the arm in each case, the Lagrange method will be used to obtain the equations of motion for each case. The links of the manipulator are assumed to be uniform rods each with a mass m, a length of $2L$ and a moment of inertia about the centre of mass of J. The mass of the redundant motor and the payload mass are represented by m_m and m_L, respectively.

- **Redundant mode dynamics**

To help in deriving the equation of motions the redundant manipulator is redrawn in Figure 5.5 where the location of the redundant motor can be shown.

$$J = \tfrac{1}{3}mL^2 \qquad (5.18)$$

$$(I_{BC})_B = (I_{AD})_A = J + mL^2 = 4J \qquad (5.19)$$

The kinetic energy of link BC is as follows,

$$T_{BC} = \tfrac{1}{2}(I_{BC})_B \dot{\theta}_2^2 = 2J\dot{\theta}_2^2 \qquad (5.20)$$

Similarly,

$$T_{AD} = \tfrac{1}{2}(I_{AD})_A \dot{\theta}_1^2 = 2J\dot{\theta}_1^2 \qquad (5.21)$$

For the mass of the redundant motor the kinetic energy is given by,

$$T_M = \tfrac{1}{2}m_m v_C^2 = \tfrac{1}{2}m_m \left(2L\dot{\theta}_2\right)^2 = 2m_m L^2\dot{\theta}_2^2 \qquad (5.22)$$

Now letting x_g, y_g, v_g and x_h, y_h, v_h be the x-y position co-ordinates and the resultant velocities of the mass centres of links CP and DP, respectively. Then we have,

$$x_g = 2L\cos\theta_2 + L\cos\theta_1$$

$$y_g = 2L\sin\theta_2 + L\sin\theta_1$$

Differentiating with respect to time and calculating the resultant velocity give,

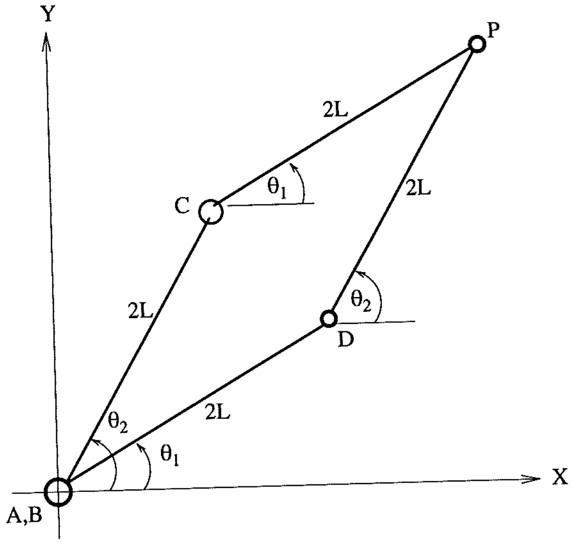


Fig. 5.5: Configuration of the redundant parallel mode.
(C: Redundant motor)

$$\dot{x}_g = -\left(2L\sin\theta_2\dot{\theta}_2 + L\sin\theta_1\dot{\theta}_1\right) \qquad (5.23)$$

$$\dot{y}_g = 2L\cos\theta_2\dot{\theta}_2 + L\cos\theta_1\dot{\theta}_1 \qquad (5.24)$$

$$v_g^2 = \dot{x}_g^2 + \dot{y}_g^2 = 4L^2\dot{\theta}_2^2 + L^2\dot{\theta}_1^2 + 4L^2\dot{\theta}_1\dot{\theta}_2\cos(\theta_2 - \theta_1) \qquad (5.25)$$

Hence, the kinetic energy of link CP is given by,

$$T_{CP} = \tfrac{1}{2}mv_g^2 + \tfrac{1}{2}J\dot{\theta}_1^2 = 6J\dot{\theta}_2^2 + 2J\dot{\theta}_1^2 + 6J\cos(\theta_2 - \theta_1)\dot{\theta}_1\dot{\theta}_2 \qquad (5.26)$$

Similarly,

$$x_h = 2L\cos\theta_1 + L\cos\theta_2$$

$$y_h = 2L\sin\theta_1 + L\sin\theta_2$$

Differentiating with respect to time and calculating the resultant velocity give,

$$\dot{x}_h = -\left(2L\sin\theta_1\dot{\theta}_1 + L\sin\theta_2\dot{\theta}_2\right) \tag{5.27}$$

$$\dot{y}_h = 2L\cos\theta_1\dot{\theta}_1 + L\cos\theta_2\dot{\theta} \tag{5.28}$$

$$v_h^2 = \dot{x}_h^2 + \dot{y}_h^2 = 4L^2\dot{\theta}_1^2 + L^2\dot{\theta}_2^2 + 4L^2\dot{\theta}_1\dot{\theta}_2\cos(\theta_2 - \theta_1) \tag{5.29}$$

The kinetic energy of link DP is then given by,

$$T_{DP} = \tfrac{1}{2}mv_h^2 + \tfrac{1}{2}J\dot{\theta}_2^2 = 6J\dot{\theta}_1^2 + 2J\dot{\theta}_2^2 + 6J\cos(\theta_2 - \theta_1)\dot{\theta}_1\dot{\theta}_2 \tag{5.30}$$

For the payload mass the kinetic energy is calculated as follows,

$$x_P = 2L\cos\theta_1 + 2L\cos\theta_2$$

$$y_P = 2L\sin\theta_1 + 2L\sin\theta_2$$

$$\dot{x}_P = -\left(2L\sin\theta_1\dot{\theta}_1 + 2L\sin\theta_2\dot{\theta}_2\right) \tag{5.31}$$

$$\dot{y}_p = 2L\cos\theta_1\dot{\theta}_1 + 2L\cos\theta_2\dot{\theta} \tag{5.32}$$

$$v_P^2 = \dot{x}_P^2 + \dot{y}_P^2 = 4L^2\dot{\theta}_1^2 + 4L^2\dot{\theta}_2^2 + 8L^2\dot{\theta}_1\dot{\theta}_2\cos(\theta_2 - \theta_1) \tag{5.33}$$

$$T_L = \tfrac{1}{2}m_L v_p^2 = 2m_L L^2\dot{\theta}_1^2 + 2m_L L^2\dot{\theta}_2^2 + 4m_L L^2\cos(\theta_2 - \theta_1)\dot{\theta}_1\dot{\theta}_2 \tag{5.34}$$

Now the total kinetic energy of the manipulator is given by

$$\begin{aligned} T &= T_{BC} + T_{AD} + T_{CP} + T_{DP} + T_L + T_M \\ &= \left(10J + 2m_L L^2\right)\dot{\theta}_1^2 + \left(10J + 2m_m L^2 + 2m_L L^2\right)\dot{\theta}_2^2 \\ &\qquad + \left(12J + 4m_L L^2\right)\cos(\theta_2 - \theta_1)\dot{\theta}_1\dot{\theta}_2 \end{aligned} \tag{5.35}$$

Letting,

$$\xi_1 = 10J + 2m_L L^2 \tag{5.36}$$

$$\xi_2 = \xi_1 + 2m_m L^2 \tag{5.37}$$

$$\xi_3 = 12J + 4m_L L^2 \tag{5.38}$$

These lead to,

$$T = \xi_1 \dot{\theta}_1^2 + \xi_2 \dot{\theta}_2^2 + \xi_3 \cos(\theta_2 - \theta_1)\dot{\theta}_1 \dot{\theta}_2 \tag{5.39}$$

Now the total virtual work is given by,

$$\delta W = \tau_A \delta(\theta_1) + \tau_B \delta(\theta_2) + \tau_C \delta(2\pi - \theta_2 + \theta_1)$$

$$= (\tau_A + \tau_C)\delta\theta_1 + (\tau_B - \tau_C)\delta\theta_2 \tag{5.40}$$

Then we have,

$$\tau_1 = \frac{\partial W}{\partial \theta_1} = \tau_A + \tau_C \tag{5.41}$$

$$\tau_2 = \frac{\partial W}{\partial \theta_2} = \tau_B - \tau_C \tag{5.42}$$

Lagrange's equation of motion,

$$\tau_i = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_i}\right) - \frac{\partial T}{\partial \theta_i} \quad \text{for i} = 1,2 \tag{5.43}$$

From equations 5.41-5.43, the equations of motion of the redundant manipulator are as follows,

$$\tau_A + \tau_C = 2\xi_1\ddot{\theta}_1 + \xi_3\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - \xi_3\dot{\theta}_2(\dot{\theta}_2 - \dot{\theta}_1)\sin(\theta_2 - \theta_1) - \xi_3\dot{\theta}_1\dot{\theta}_2\sin(\theta_2 - \theta_1)$$

$$= 2\xi_1\ddot{\theta}_1 + \xi_3\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - \xi_3\sin(\theta_2 - \theta_1)\ \dot{\theta}_2^2 \qquad (5.44)$$

$$\tau_B - \tau_C = 2\xi_2\ddot{\theta}_2 + \xi_3\cos(\theta_2 - \theta_1)\ \ddot{\theta}_1 - \xi_3\dot{\theta}_1(\dot{\theta}_2 - \dot{\theta}_1)\sin(\theta_2 - \theta_1) + \xi_3\dot{\theta}_1\dot{\theta}_2\sin(\theta_2 - \theta_1)$$

$$= 2\xi_2\ddot{\theta}_2 + \xi_3\cos(\theta_2 - \theta_1)\ \ddot{\theta}_1 + \xi_3\sin(\theta_2 - \theta_1)\ \dot{\theta}_1^2 \qquad (5.45)$$

- **Normal parallel mode without the redundant motor mass**

In this case we substitute for $m_m = 0$ in equations 5.37 and removing the torque τ_C from equations 5.44 and 5.45. Performing these operations leads to,

$$\tau_A = 2\xi_1\ddot{\theta}_1 + \xi_3\cos(\theta_2 - \theta_1)\ \ddot{\theta}_2 - \xi_3\sin(\theta_2 - \theta_1)\ \dot{\theta}_2^2 \qquad (5.46)$$

$$\tau_B = 2\xi_1\ddot{\theta}_2 + \xi_3\cos(\theta_2 - \theta_1)\ \ddot{\theta}_1 + \xi_3\sin(\theta_2 - \theta_1)\ \dot{\theta}_1^2 \qquad (5.47)$$

- **Serial mode dynamics**

The configuration of the manipulator in this case is shown in Figure 5.6.

Recalling that,

$$(I_{BC})_B = (I_{AD})_A = J + mL^2 = 4J$$

The kinetic energies of both links AD and BC are then calculated as follows,

$$T_{AD} = T_{BC} = \tfrac{1}{2}(I_{BC})_B\dot{\theta}_1^2 = 2J\dot{\theta}_1^2 \qquad (5.48)$$

Also, the kinetic energy of the motor mass at joint C is given by,

$$T_M = \tfrac{1}{2} m_m v_C^2 = \tfrac{1}{2} m_m \left(2L\dot{\theta}_1\right)^2 = 2m_m L^2 \dot{\theta}_1^2 \qquad (5.49)$$

The x-y position co-ordinates of the end-effector are as follows,

$$x_P = 2L\cos\theta_1 + 2L\cos\theta_3$$

$$y_P = 2L\sin\theta_1 + 2L\sin\theta_3$$

Differentiating with respect to time and calculating the resultant velocity give,

$$\dot{x}_P = -\left(2L\sin\theta_1\dot{\theta}_1 + 2L\sin\theta_3\dot{\theta}_3\right) \qquad (5.50)$$

$$\dot{y}_p = 2L\cos\theta_1\dot{\theta}_1 + 2L\cos\theta_3\dot{\theta}_3 \qquad (5.51)$$

$$v_P^2 = \dot{x}_P^2 + \dot{y}_P^2 = 4L^2\dot{\theta}_1^2 + 4L^2\dot{\theta}_3^2 + 8L^2\dot{\theta}_1\dot{\theta}_3\cos\left(\theta_3 - \theta_1\right) \qquad (5.52)$$

The kinetic energy of the payload mass is then given by,

$$T_L = \tfrac{1}{2} m_L v_p^2 = 2m_L L^2\dot{\theta}_1^2 + 2m_L L^2\dot{\theta}_3^2 + 4m_L L^2\cos\left(\theta_3 - \theta_1\right)\dot{\theta}_1\dot{\theta}_3 \qquad (5.53)$$

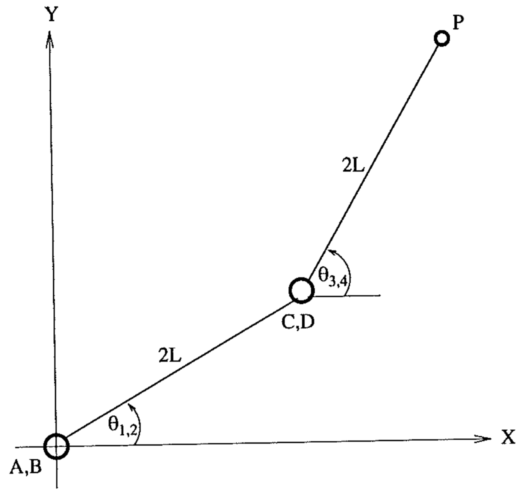


Fig. 5.6: Configuration of the manipulator serial mode.

The x-y position co-ordinates of the mass centre of link CP or DP is given by,

$$x_h = 2L\cos\theta_1 + L\cos\theta_3$$

$$y_h = 2L\sin\theta_1 + L\sin\theta_3$$

Differentiating with respect to time and calculating the resultant velocity give,

$$\dot{x}_h = -\left(2L\sin\theta_1\dot{\theta}_1 + L\sin\theta_3\dot{\theta}_3\right) \tag{5.54}$$

$$\dot{y}_h = 2L\cos\theta_1\dot{\theta}_1 + L\cos\theta_3\dot{\theta}_3 \tag{5.55}$$

$$v_h^2 = \dot{x}_h^2 + \dot{y}_h^2 = 4L^2\dot{\theta}_1^2 + L^2\dot{\theta}_3^2 + 4L^2\dot{\theta}_1\dot{\theta}_3\cos(\theta_3 - \theta_1) \tag{5.56}$$

The kinetic energies of links CP and DP are given by,

$$\begin{aligned} T_{CP} = T_{DP} &= \tfrac{1}{2}mv_h^2 + \tfrac{1}{2}J\dot{\theta}_3^2 \\ &= 6J\dot{\theta}_1^2 + 2J\dot{\theta}_3^2 + 6J\cos(\theta_3 - \theta_1)\ \dot{\theta}_1\dot{\theta}_3 \end{aligned} \tag{5.57}$$

Now the total kinetic energy of the manipulator is calculated as follows,

$$T = T_{BC} + T_{AD} + T_{CP} + T_{DP} + T_L + T_M \tag{5.58}$$

Substituting from equations 5.48, 5.49, 5.53 and 5.57 in equation 5.58 gives,

$$\begin{aligned} T = &\left(16J + 2m_L L^2 + 2m_m L^2\right)\dot{\theta}_1^2 + \left(4J + 2m_L L^2\right)\dot{\theta}_3^2 \\ &+ \left(12J + 4m_L L^2\right)\cos(\theta_3 - \theta_1)\ \dot{\theta}_1\dot{\theta}_3 \end{aligned} \tag{5.59}$$

Letting,

$$\lambda_1 = 16J + 2m_L L^2 + 2m_m L^2 \tag{5.60}$$

$$\lambda_2 = 4J + 2m_L L^2 \tag{5.61}$$

$$\lambda_3 = 12J + 4m_L L^2 \tag{5.62}$$

Substituting from 5.60-5.62 in equation 5.59 yields,

$$T = \lambda_1 \dot{\theta}_1^2 + \lambda_2 \dot{\theta}_3^2 + \lambda_3 \cos(\theta_3 - \theta_1)\dot{\theta}_1 \dot{\theta}_3 \tag{5.63}$$

Now the total virtual work is given by,

$$\begin{aligned} \delta W &= \tau_A \ \delta(\theta_1) + \tau_B \ \delta(\theta_1) + \tau_C \ \delta(\theta_3 - \theta_1) \\ &= (\tau_A + \tau_B - \tau_C) \ \delta\theta_1 + \tau_C \ \delta\theta_3 \end{aligned} \tag{5.64}$$

From equation 5.64, the generalised forces are calculated as follows,

$$\tau_1 = \frac{\partial W}{\partial \theta_1} = \tau_A + \tau_B - \tau_C \tag{5.65}$$

$$\tau_3 = \frac{\partial W}{\partial \theta_3} = \tau_C \tag{5.66}$$

From Lagrange's equation (equation 5.43), the equations of motion for this serial mode are given by,

$$\tau_A + \tau_B - \tau_C = 2\lambda_1 \ddot{\theta}_1 + \lambda_3 \cos(\theta_3 - \theta_1)\ddot{\theta}_3 - \lambda_3 \dot{\theta}_3^2 \sin(\theta_3 - \theta_1) \tag{5.67}$$

$$\tau_C = \lambda_3 \cos(\theta_3 - \theta_1)\ddot{\theta}_1 + 2\lambda_2 \ddot{\theta}_3 + \lambda_3 \dot{\theta}_1^2 \sin(\theta_3 - \theta_1) \tag{5.68}$$

5.5 Performance evaluation of the manipulator modes

In order to complete the design process it is necessary to find a suitable basis for the selection of the motors. Additionally the dynamic characteristics of the different modes under different payloads need to be evaluated. In order to do this the arm was simulated to perform a pre-specified dynamic task and the payload was gradually increased until the motors reach their maximum values of the torques. The three motors were assumed to have the same maximum torque this is because the redundant motor becomes a main motor in the serial mode. The task used for the simulations is a point-to-point fifth order joint-space trajectory connecting $\theta_{10} = 0.175$ rad, $\theta_{20} = 0.785$ rad to $\theta_{1f} = 1.571$ rad and $\theta_{2f} = 2.88$ rad in $t_f = 1\,\text{sec}$. The parameters of the manipulator are assumed to be of the values $m = 0.25\,\text{kg}$, $2L = 0.125\,\text{m}$ and $m_m = 0.44\,\text{kg}$. The maximum torque limit is chosen to be 6.5 N.m.

The trajectory equations for the links are as follows,

$$\theta_1(t) = 0.1745 + 13.963\,t^3 - 20.9445\,t^4 + 8.3778\,t^5 \tag{5.69}$$

$$\dot{\theta}_1(t) = 41.889\,t^2 - 83.778\,t^3 + 41.889\ t^4 \tag{5.70}$$

$$\ddot{\theta}_1(t) = 83.778\,t - 251.334\,t^2 + 167.556\,t^3 \tag{5.71}$$

$$\theta_2(t) = 0.7854 + 20.994\,t^3 - 31.4195\,t^4 + 12.5664\,t^5 \tag{5.72}$$

$$\dot{\theta}_2(t) = 62.8319\,t^2 - 125.6637\,t^3 + 62.8319\ t^4 \tag{5.73}$$

$$\ddot{\theta}_2(t) = 125.6638\,t - 376.9911\,t^2 + 251.3276\,t^3 \tag{5.74}$$

In the simulations, the determination of the motor torques in the case of the redundant mode is achieved by using the pseudo-inverse approach used in chapter 4. To implement this we do the following,

In equations 5.44 and 5.45, letting

$$\tau_1 = \tau_A + \tau_C \qquad (5.75)$$

$$\tau_2 = \tau_B - \tau_C \qquad (5.76)$$

Writing these two equations in matrix form gives,

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \tau_A \\ \tau_B \\ \tau_C \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \qquad (5.77)$$

Solving this equation leads to,

$$\tau_A = \frac{2\tau_1 + \tau_2}{3} \qquad (5.78)$$

$$\tau_B = \frac{\tau_1 + 2\tau_2}{3} \qquad (5.79)$$

$$\tau_C = \frac{\tau_1 - \tau_2}{3} \qquad (5.80)$$

In the case of the serial mode the two motors A and B are assumed to have the same torques. The simulations are performed using the Simulink/Matlab5.3 program, and the block diagrams for the three modes are shown in Figures 5.7, 5.8 and 5.9.

5.6 Results and Discussion

The results show that the maximum payload achieved by the redundant mode is 5.25 kg and it is 5.20 kg for the serial mode, whilst it is 5.00 kg for the non-redundant mode without the mass of the redundant motor. The joint torques for all the modes are shown in Figures 5.10, 5.11 and 5.12. From these results the redundant and the serial modes succeed in carrying higher loads without exceeding the torque limits of the motors. It is important to note that in the case of the redundant mode the gain achieved (compared to the non-redundant mode) is relatively small (5%) if compared with gains obtained by other researchers. This may be attributed to the different tasks imposed or to the different optimisation algorithms used in the determination of the motor torques, or may even be due to the differences in the manipulator designs. The first proposal agrees with the findings of Beiner [16] where smaller gains in dynamic tasks were reported for specific trajectory paths. On the other hand, the improvements in the serial mode are mainly due to the extra motor at the base (A or B). In this work the serial mode can be considered as a redundant serial robot with the redundant motor located on the ground. This is opposite to the redundant parallel mode where the extra motor forms a dynamic load on the system. In other words the help provided by the extra motor in the serial mode is 100 %. It is important to refer to the fact that this result is based mainly on the assumption that the three motors have the same maximum limits. If the maximum torque limit of the motor at C is decreased, the achieved maximum payload in the serial mode will be decreased. This is absolutely true because the motor located at C is vitally important for the serial robot, as it is one of the main motors of the arm.

Despite the improvements in the redundant parallel mode over the non-redundant one at higher loads, the results obtained at low loads were different and the non-redundant mode was found to be better than the redundant mode. As the payload increases the difference decreases until they become the same. Further increase in the payload leads to improvements in the redundant mode over the non-redundant one. We should refer here to the fact that the results obtained show that there is a limit on the gains obtained from the redundant mode. A further increase in the payload up to a level of 10 kg gives the same gain, which is 5% over the non-redundant mode.

From the design point of view the selection of the torque limits for the motors should be as follows:

1. Motor C should be selected on the basis of the applications required by the serial mode.
2. Motor A and B should be selected on the basis of the applications required by the non-redundant robot and should be enough to satisfy the serial applications.

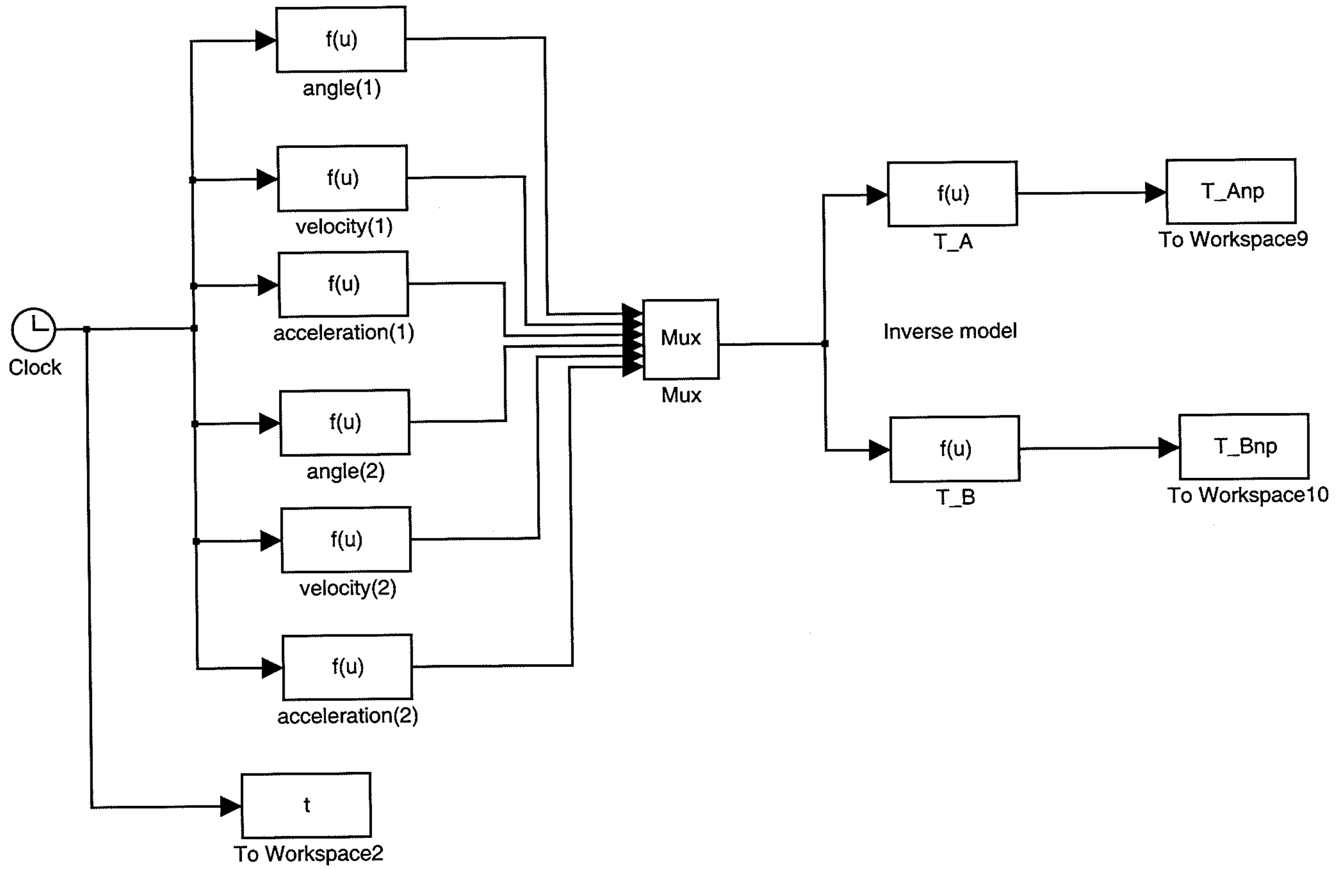


Fig. 5.7: Simulink block diagram for calculating the motors torques in the case of the normal parallel mode.

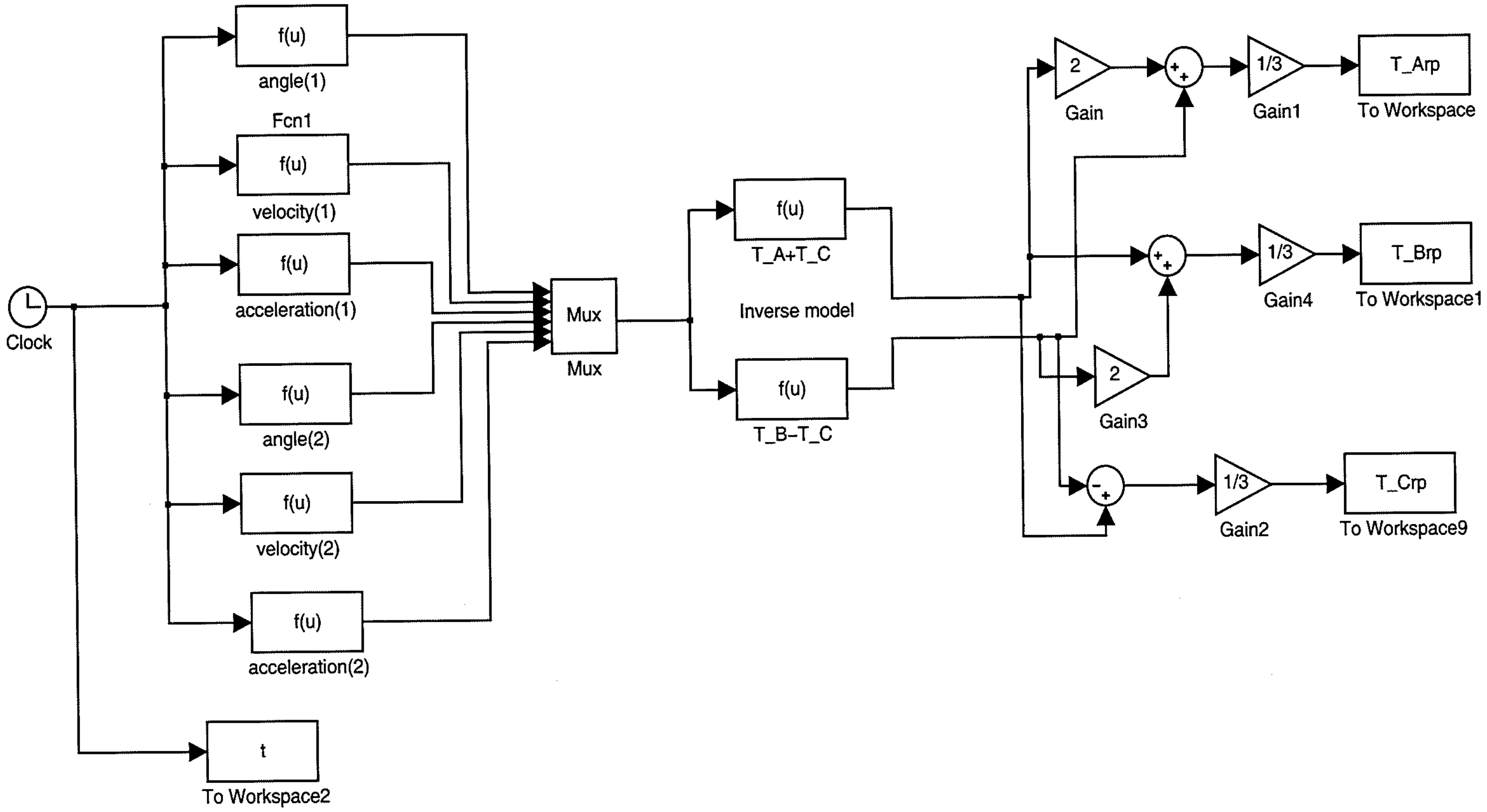


Fig. 5.8: Simulink block diagram for calculating the motors torques in the case of the redundant parallel mode.

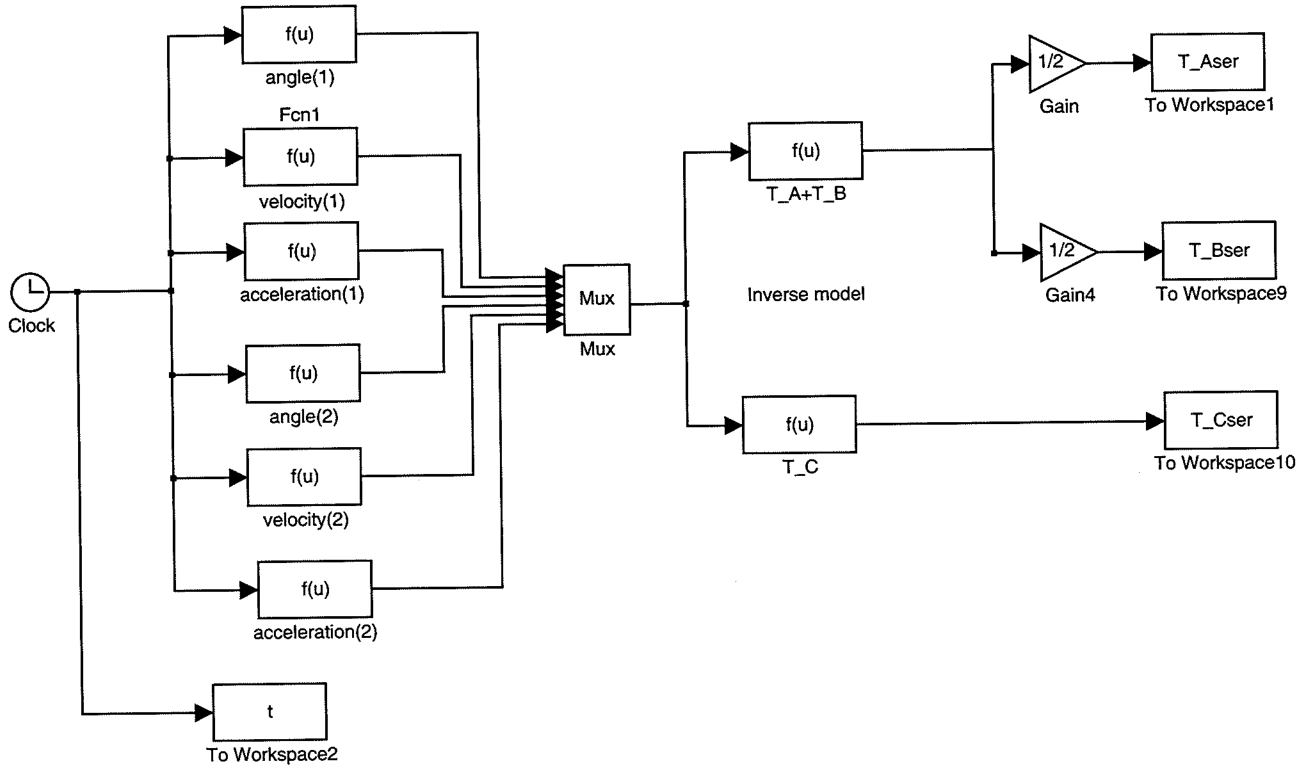


Fig. 5.9: Simulink block diagram for calculating the motors torques in the case of the serial mode.

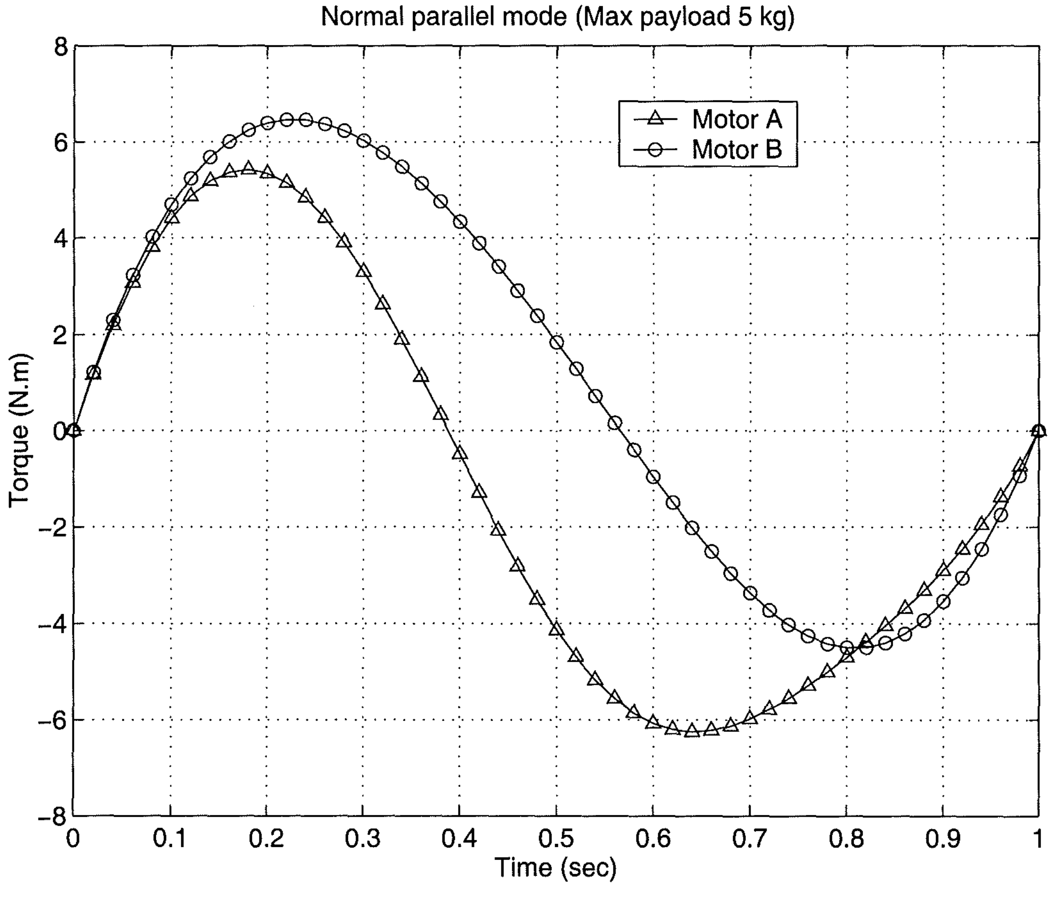


Fig. 5.10: Torque requirements for the normal parallel mode.

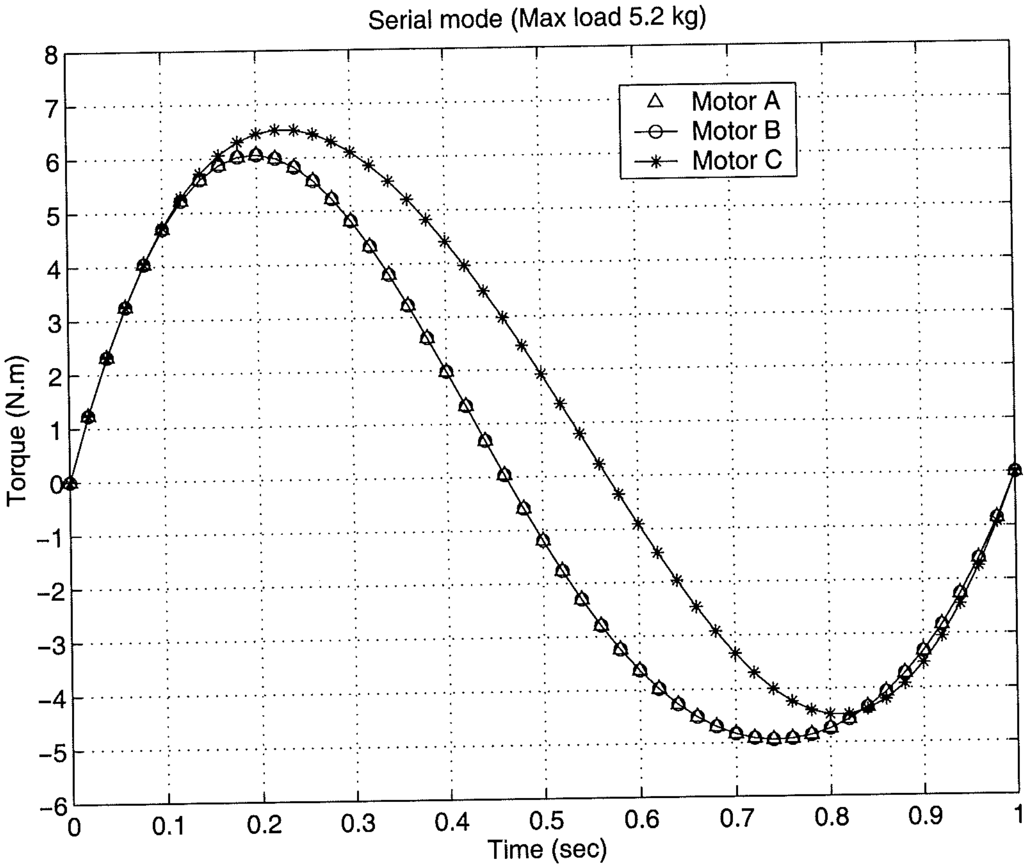


Fig. 5.11: Torque requirements for the serial mode.

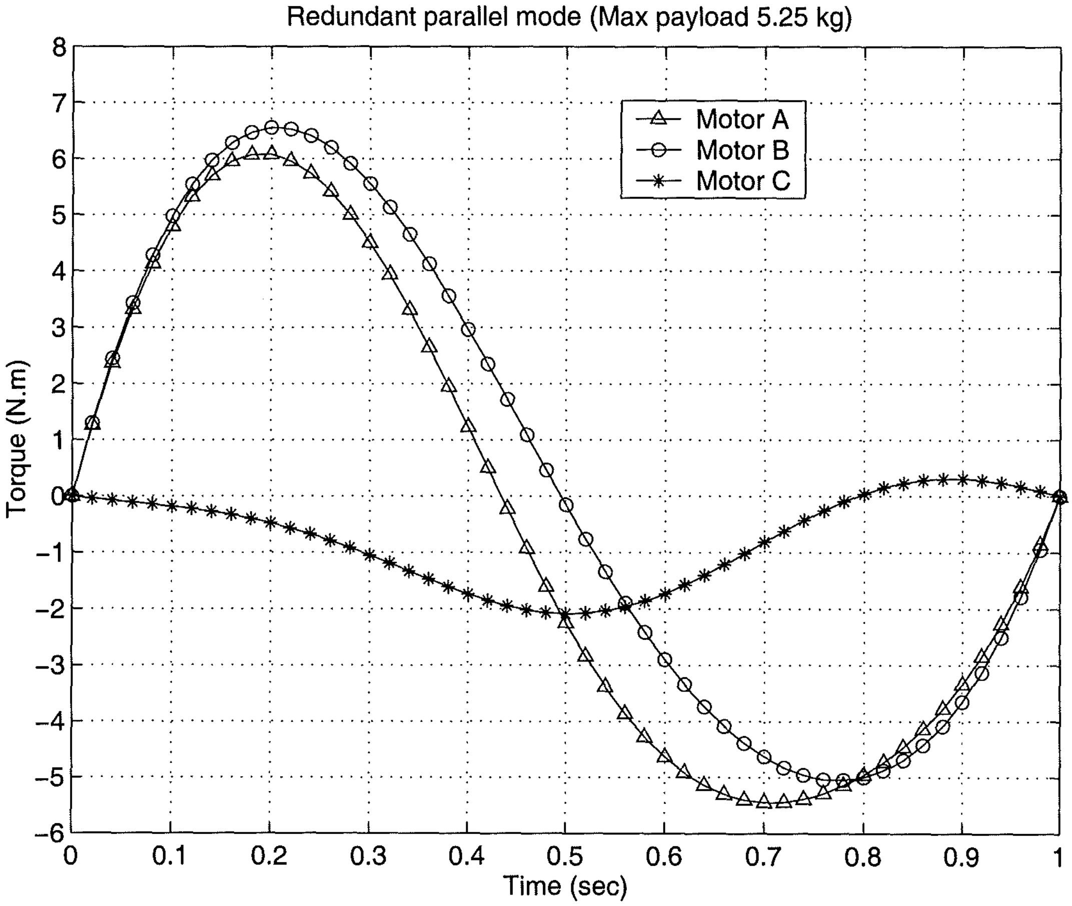


Fig. 5.12: Torque requirements for the redundant parallel mode.

5.7 Generalisation

It has been shown that singularity can be used to build mechanical structures with interesting dynamic behaviour. The idea can be extended to the design of a new class of planar manipulators which could be called parallel-parallel manipulators (Figure 5.13 and 5.14). In such manipulators the original mechanism of Figure 5.1 is used as a serial branch in the normal parallel manipulator. This allows the applications of actuation-redundancy with more than one redundant motor and with all actuators on the ground. In other words the redundant actuators do not form a dynamic load on the original system thus increasing their contribution to the manipulator without losses. Additionally changing the mode of operation of each branch (as done in section 5.3) it can allow different operating characteristics.

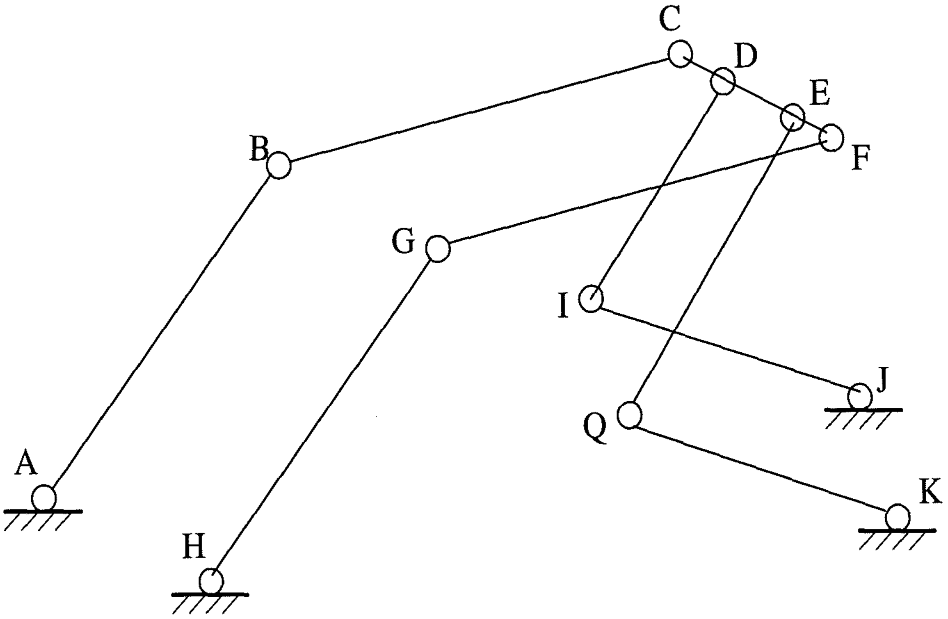


Fig. 5.13: Parallel-Parallel 2 DOF manipulator.

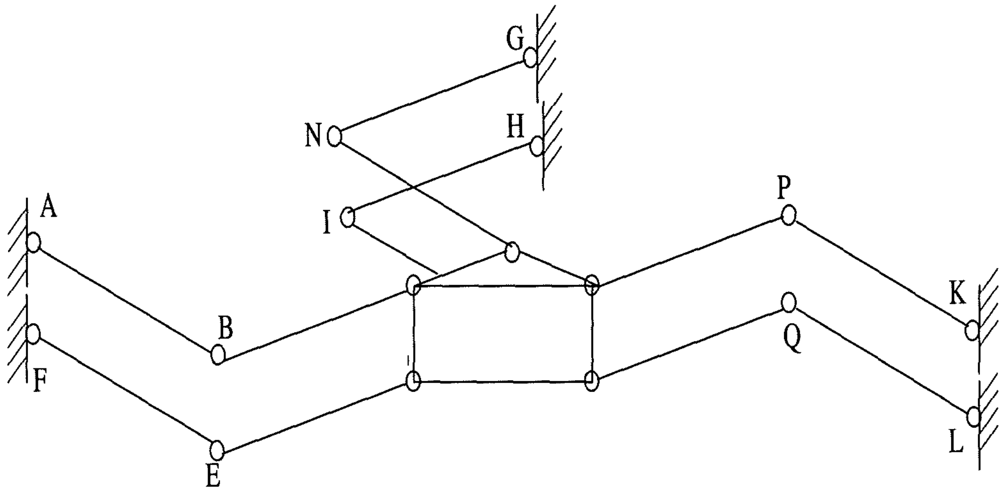


Fig. 5.14: Parallel-Parallel 3 DOF manipulator.

5.8 Conclusions

In this chapter the new concept of singularity-based design is presented and used to develop a manipulator with multiple operating modes. For tasks with large workspace and medium load capacities the serial mode can be used. For high loads the normal or redundant-actuated parallel mode can be chosen. For each mode a dynamic model is derived using the Lagrange method. The possible conditions by which the motors torques can be selected have also been defined. In addition to this, a performance analysis has been carried out for each operating mode by using simulations. The results indicate improvements in the serial mode form the viewpoint of load capacity. Lastly the idea is generalised to include the design of new planar mechanisms called parallel-parallel manipulators.

Chapter 6

Experimental Set-up and Model Validation

6.1 Introduction

Because the control problem of parallel manipulators is the main concern of the work here, and due to the importance of the issue of practical validation to any control system design, it has been decided to build an experimental parallel manipulator. The arm is an implementation of the multi-cell parallel manipulator design technique. It has two cells and is directly driven by electric motors. The manipulator moves in the horizontal plane, so as to cancel the gravity forces. The first part of the chapter is directed towards giving the details of the overall experimental robot. These include the specifications of the links, actuators, sensors, data acquisition and control software. Since, in general, the theoretical dynamic model of the robot is not always perfect, due to issues such as the uncertainty in the robot parameters, noise in the measured signals, ignoring the flexibility of joints and links, ignoring friction at the joints etc., it was decided that the next part of the chapter would be directed towards evaluating the quality of the derived dynamics of the manipulator. This is performed via a comparison of the data collected form the experimental robot with that obtained by simulations using the dynamical model. It is important to note that this process is very important for all control techniques that are structurally dependent on the dynamic model of the robot.

6.2 Specifications of the Experimental System Components

6.2.1 Manipulator (Mechanical Part)

Because the manipulator has two cells there are two cross-links and four normal links. The approximate value of the mass of each normal link is 0.1467 kg while it is

0.291 kg. for each cross-link. Figures 6.1 and 6.2 show the dimensional detail of each of these links while Figure 6.3 shows the whole robot. From Figure 6.3 it can be seen that the arm has two serial chains (up and down), each with three links, two of them being normal links and the other is a cross-link. Two joints located at the centre of the cross-link, and the end of the third link connects the two chains. The first degree of freedom (DOF) results from the angular motion of the first link of the up serial chain. It has been decided to refer to this link as the first input link. The second DOF is due to the motion of the first link of the down serial chain. We refer to this link as the second input link. Controlling the motion of these two input links controls the x-y co-ordinates of the gripper.

6.2.2 Actuators/Amplifiers

The actuators of the experimental parallel robot are direct current (DC) permanent magnet servomotors. Both motors are the same model, which is S642-1B/T supplied by Electro-Craft. This motor has a peak current demand of 16.2 A, peak torque of 3.3 N.m, stall current of 3.3 A, stall torque of 0.68 N.m and a maximum supply voltage of 60 V. The moment of inertia of the rotor is 1.3 kg.cm^2 and the motor torque constant is 0.23 N.m/A.

The chosen amplifier is the MAX 250 unit supplied by Electro-Craft. It is a single axis four quadrant pulse-width modulation (PWM) drive for permanent magnet DC brush motors. The PWM frequency is 20 kHz. The drive can be operated in both velocity and torque control modes. The bandwidth of the drive, in the case of torque mode, is up to 2 kHz. The mean current value is 2.2 A while it is 5.6 A for the peak value. These are the default values of the amplifier, and removing certain resistors inside the drive motherboard allows one to vary them. The input control voltage to the amplifier must be within ± 10 V.

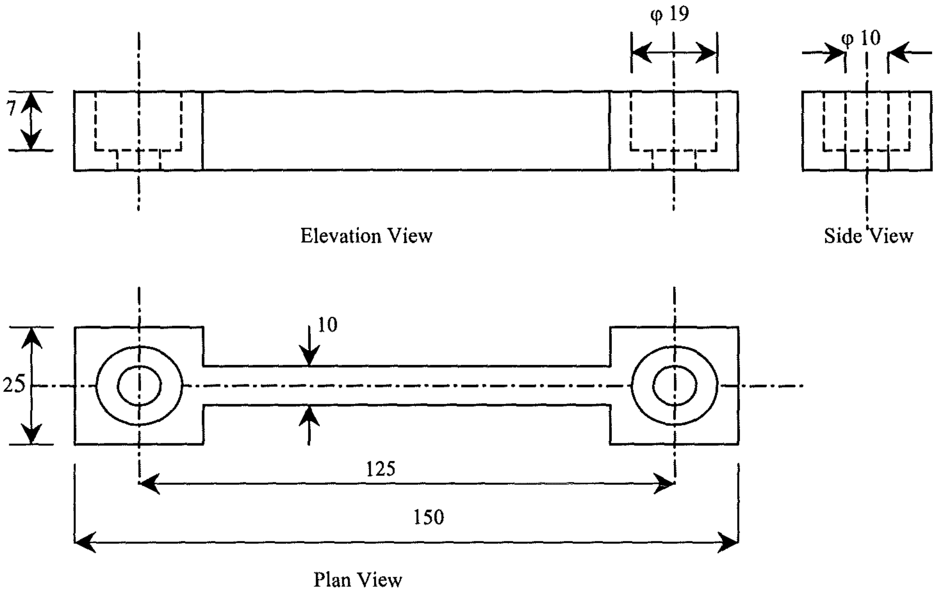


Fig. 6.1: Schematic diagram of the normal link.

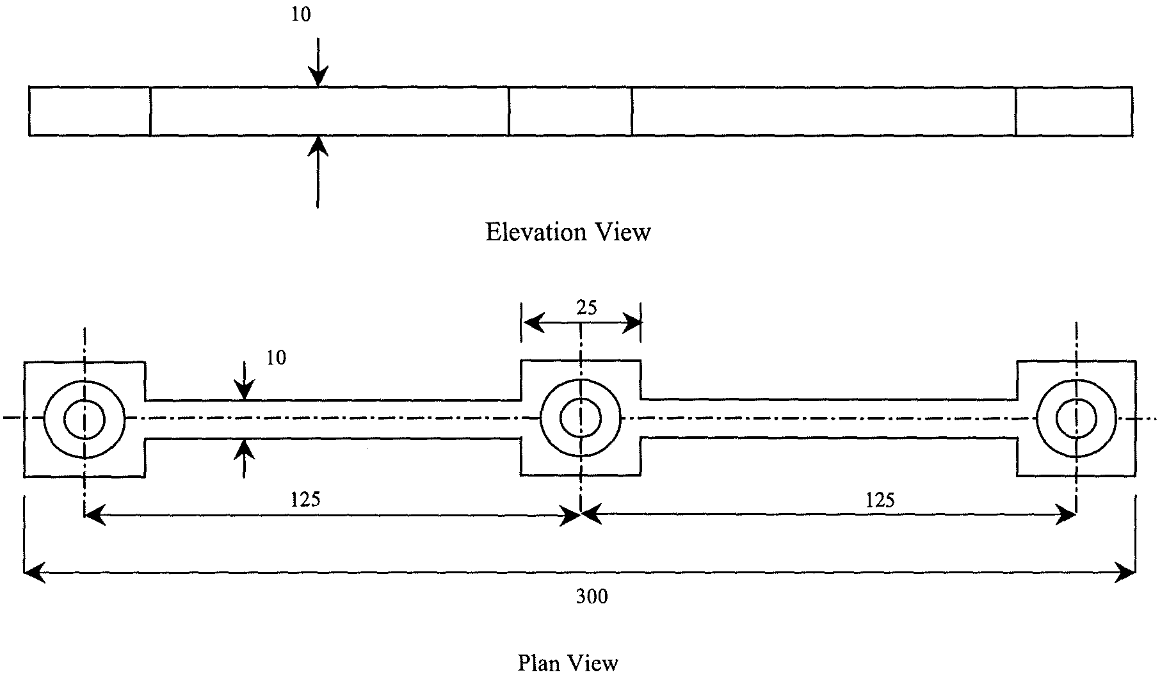


Fig. 6.2: Schematic diagram of the cross-link.

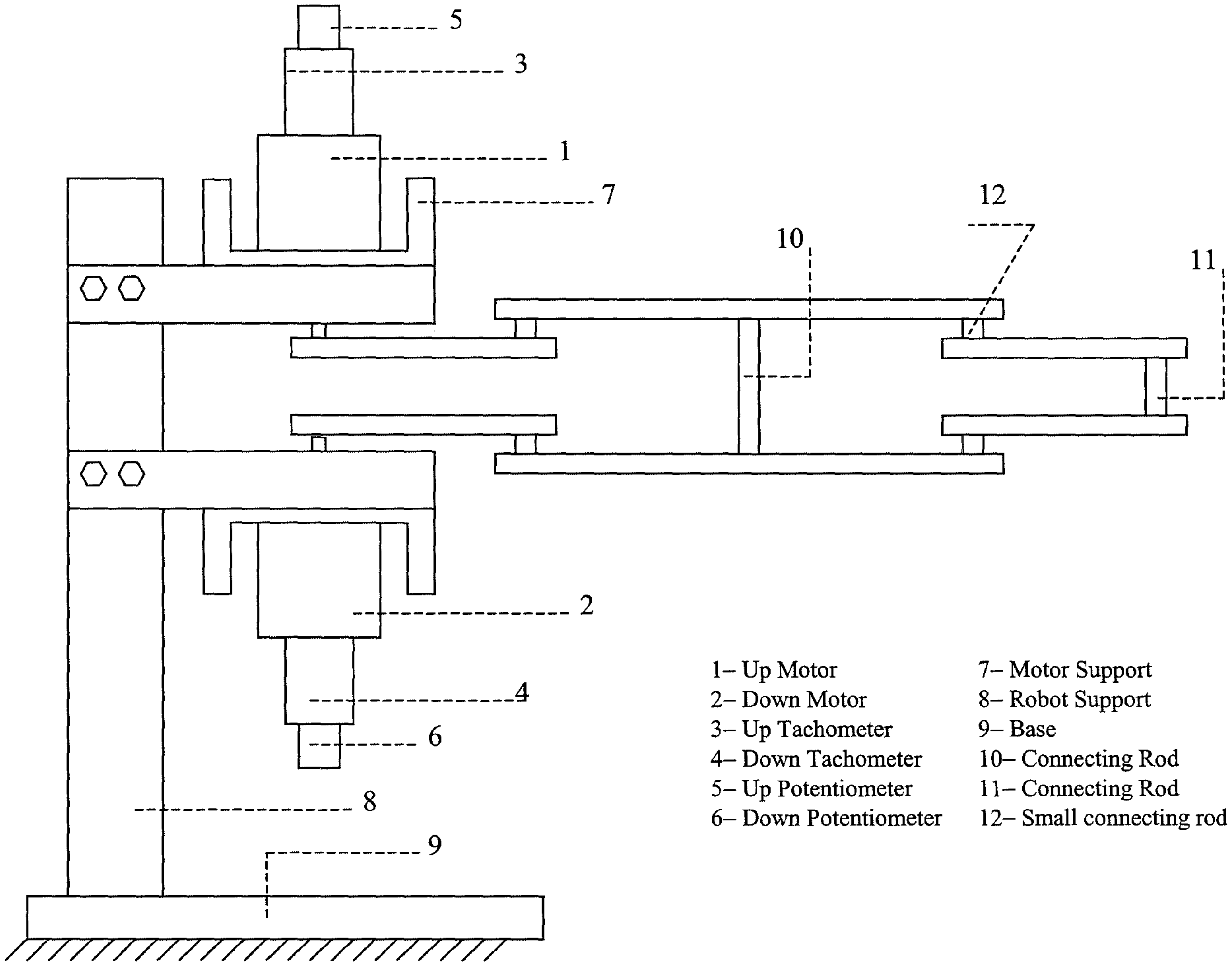


Fig. 6.3: Schematic diagram of the experimental 2-DOF, SCARA type, parallel robot.

6.2.3 Sensors

6.2.3.1 Position Sensors

For position measurements, and for reducing the cost of the overall system, it was decided to use potentiometers. The potentiometers are one turn, conductive, plastic servo potentiometers supplied by the RS Company. The maximum permissible wiper-current is 0.01 A.

6.2.3.2 Velocity Sensors

The angular velocities of the input links are measured using tachometers mounted integrally within the motors housings. To remove high frequency noise due to the amplifiers' current signals, the signals from the tachometers are filtered. This is performed digitally inside the software using a first order low-pass Butterworth filter with a cut-off frequency of 20 Hz. The tachometer constant is 0.014 V/r.p.m.

6.2.4 Control Software

The control software is the *Real Time Toolbox*, which is a package for connecting MATLAB and SIMULINK to the real world. It adds on the capability of acquiring data in real time, immediately processing it by MATLAB commands or a SIMULINK model, and then sending it back to the outside world. All the basic commands of the Real Time Toolbox have both graphical and command-line interface. From the Real Time Toolbox SIMULINK block library, three blocks are of main concern in the control of the experimental robot. The first one is called the *Adapter*, this being a special block that loads the hardware driver but does not perform any actions during the simulations. The Adapter has no inputs and no outputs. The other two blocks are responsible for sending and receiving the data to and from the robot. These are the *RT-Out* and *RT-In* blocks, respectively.

6.2.5 Computer

The computer is a basic specification Pentium 100 MHz PC with 16-mega bytes of RAM.

6.2.6 Communications

The communications between the control program and the robot is via one AD 512 data acquisition card supplied by Humusoft. The card is designed for standard data acquisition and control applications and optimised for use with the Real Time Toolbox of MATLAB. The AD 512 contains a 100 kHz 12 bit A/D converter with sample/hold circuit, four software selectable input ranges and 8 channel input multiplexers, 2 independent double buffered 12 bit D/A converters and an 8 bit digital output port. It is important to note that the $\pm 10\,\mathrm{V}$ range was chosen for both the input and the output channels. Because the driver of the card maps the selected voltage range into the range – 1 to 1 the control designer must be sure that the control signal from the software to the amplifiers is within this range.

The configuration of the overall actual system is shown in Figure 6.4. The manipulator itself is shown in Figure 6.5.

6.3 Actual Arm Dynamics

To reduce the expected gap between the theoretical and the actual robot dynamics, an attempt was made to take into account all the small features that exist in the actual system, but not in the theoretical system. These include the two connecting rods which connect the two serial chains as shown in Figure 6.3, the small connecting rods appeared in the same graph, and the masses of the bearings. These were treated as concentrated masses when refining the mathematical model.

Applying the above, the derived system dynamics (actual) are given by the following two equations,

$$\tau_1 = J\left(162\,\ddot{\theta}_1 + 142\cos(\theta_2 - \theta_1)\ddot{\theta}_2 - 142\sin(\theta_2 - \theta_1)\dot{\theta}_2^2\right) \qquad (6.1)$$

$$\tau_2 = J\left(162\,\ddot{\theta}_2 + 142\cos(\theta_2 - \theta_1)\ddot{\theta}_1 + 142\sin(\theta_2 - \theta_1)\dot{\theta}_1^2\right) \qquad (6.2)$$

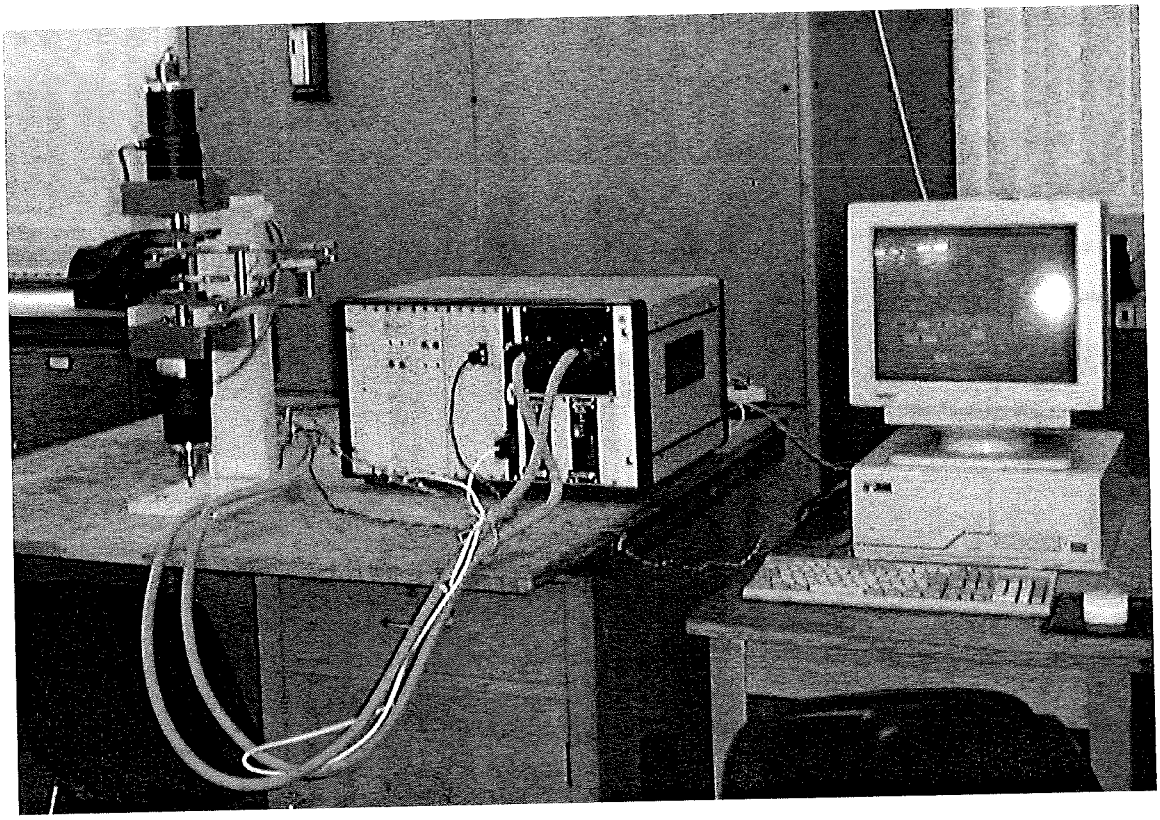


Fig. 6.4: View of the experimental system.

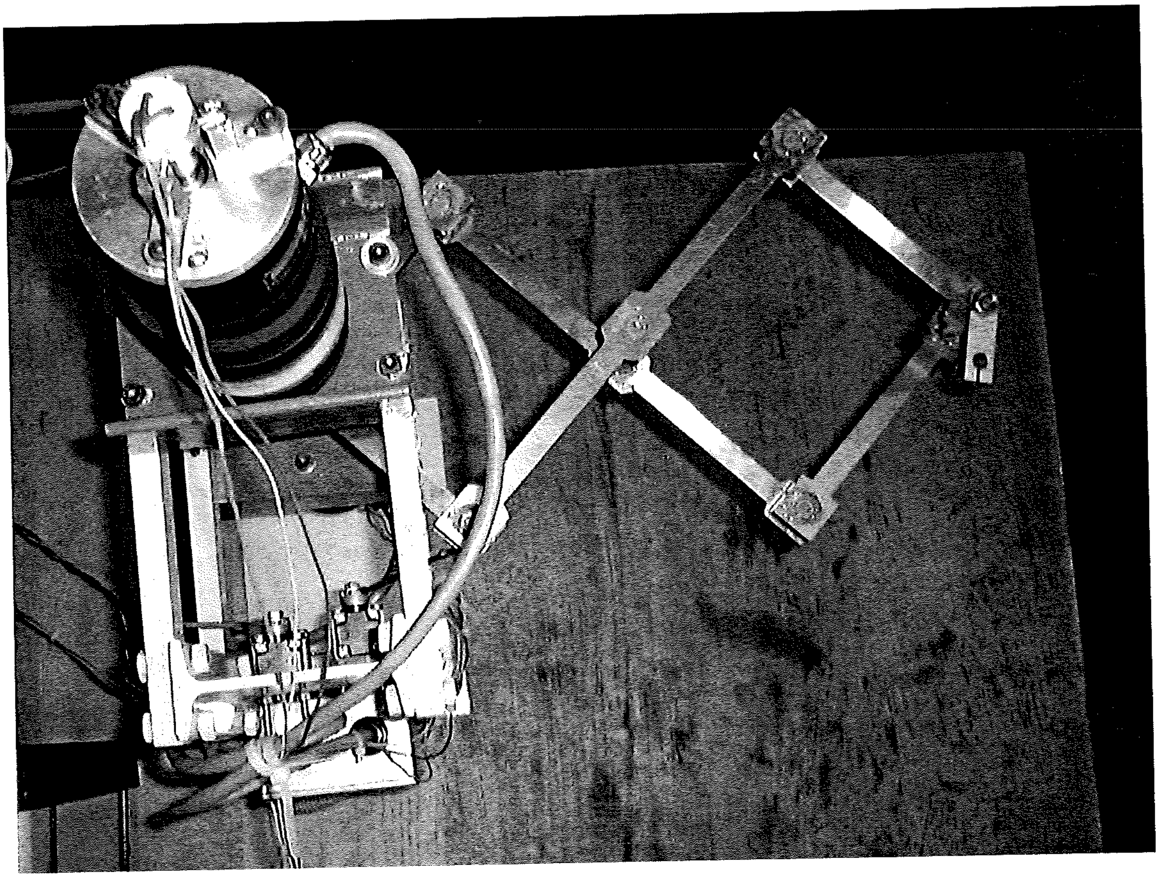


Fig. 6.5: View of the system manipulator.

Where,

$J = 4\times10^{-4}$ kg.m^{2}, this being the calculated moment of inertia of each normal link.

Because the control signals from the computer (PC) to the motors are in volts and not in torque units, equations 6.1 and 6.2 have been modified to represent the system dynamics in terms of the control voltages (u_1 and u_2) as follows,

$$u_1 = \frac{J}{k_a k_t}\left\{162\,\ddot{\theta}_1 + 142\cos\left(\theta_2 - \theta_1\right)\ddot{\theta}_2 - 142\sin\left(\theta_2 - \theta_1\right)\dot{\theta}_2^2\right\} \qquad (6.3)$$

$$u_2 = \frac{J}{k_a k_t}\left\{162\,\ddot{\theta}_2 + 142\cos\left(\theta_2 - \theta_1\right)\ddot{\theta}_1 + 142\sin\left(\theta_2 - \theta_1\right)\dot{\theta}_1^2\right\} \qquad (6.4)$$

Where,

$k_a = 1.2$ A/V, being the measured gain of each amplifier.

$k_t = 0.23$ N.m/A, being the motor torque constant.

We should refer here to the fact that in equations 6.3 and 6.4 the subscript 1 is referring to the up motor, or the up input link, and the subscript 2 is referring to the down motor, or the down input link.

6.3.1 Friction Model

To improve the actual model and because some dry friction in the experimental system was encountered, a friction model has been added to the dynamic equations. The actual dynamic model after incorporating the friction part is represented by the following equations,

$$u_1 = \frac{J}{k_a k_t}\left\{162\,\ddot{\theta}_1 + 142\cos\left(\theta_2 - \theta_1\right)\ddot{\theta}_2 - 142\sin\left(\theta_2 - \theta_1\right)\dot{\theta}_2^2\right\} + f_1\,\mathrm{sgn}\left(\dot{\theta}_1\right) + b_1\dot{\theta}_1 \qquad (6.5)$$

$$u_2 = \frac{J}{k_a k_t}\left\{162\,\ddot{\theta}_2 + 142\cos\left(\theta_2 - \theta_1\right)\ddot{\theta}_1 + 142\sin\left(\theta_2 - \theta_1\right)\dot{\theta}_1^2\right\} + f_2\,\mathrm{sgn}\left(\dot{\theta}_2\right) + b_2\dot{\theta}_2 \qquad (6.6)$$

Here f_1 and f_2 are the dry friction forces associated with both the up and the down degrees of freedom. The viscous friction coefficients are represented by b_1 and b_2 and the function sgn represents the signum function.

The above friction definition is a simplified model that considers friction forces associated with each DOF rather than with each joint. This is due to the difficulty in identifying the friction parameters of each of the unactuated joints. In order to determine the parameter f_1, the down motor was switched off and a gradually increasing voltage signal was sent to the up motor. The value of the voltage at which the arm began its motion was considered to represent the dry friction force, in volts, acting on the up DOF. Multiplying this voltage value by the amplifier gain and motor torque constant gave the value of $f_1 = 0.09$ N.m. The value of $f_2 = 0.1$ N.m was obtained by using the same procedure, but in this case the up motor was switched off. The values of the viscous friction coefficients were chosen to give the best simulation results ($b_1 = b_2 = 0.01$ N.m.sec/rad), in the context of minimised error. These values are numerically small and would therefore not have masked other modelling errors of significance.

6.4 Model Validation

The validation of the mathematical model has been achieved by comparing data obtained from the experimental manipulator with simulations using the system dynamics represented by equations 6.5 and 6.6. The actual data from the experimental system, and that from the simulations, were obtained using a simple independent joint proportional controller applied to the manipulator. During the experiments and simulations the arm was forced to follow a specified trajectory which was represented by a sine wave function. We should refer here to the fact that the sampling rate during the experiments was 200 Hz. This is the minimum value that could be obtained from the Real Time Toolbox. The block diagrams used during the simulations and the experiments are shown in Figures 6.6 and 6.7, respectively.

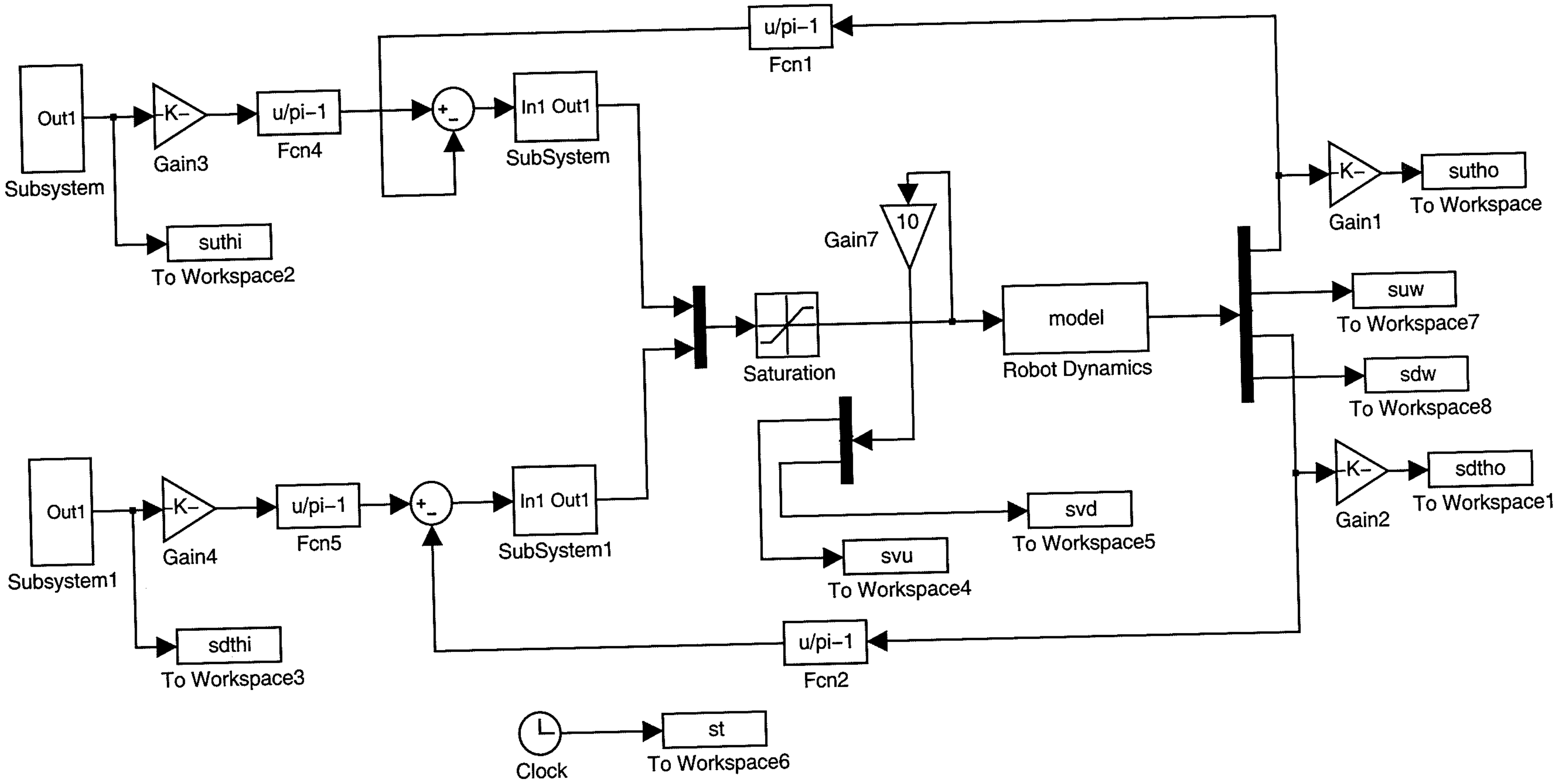


Fig. 6.6: SIMULINK block diagram for simulations.

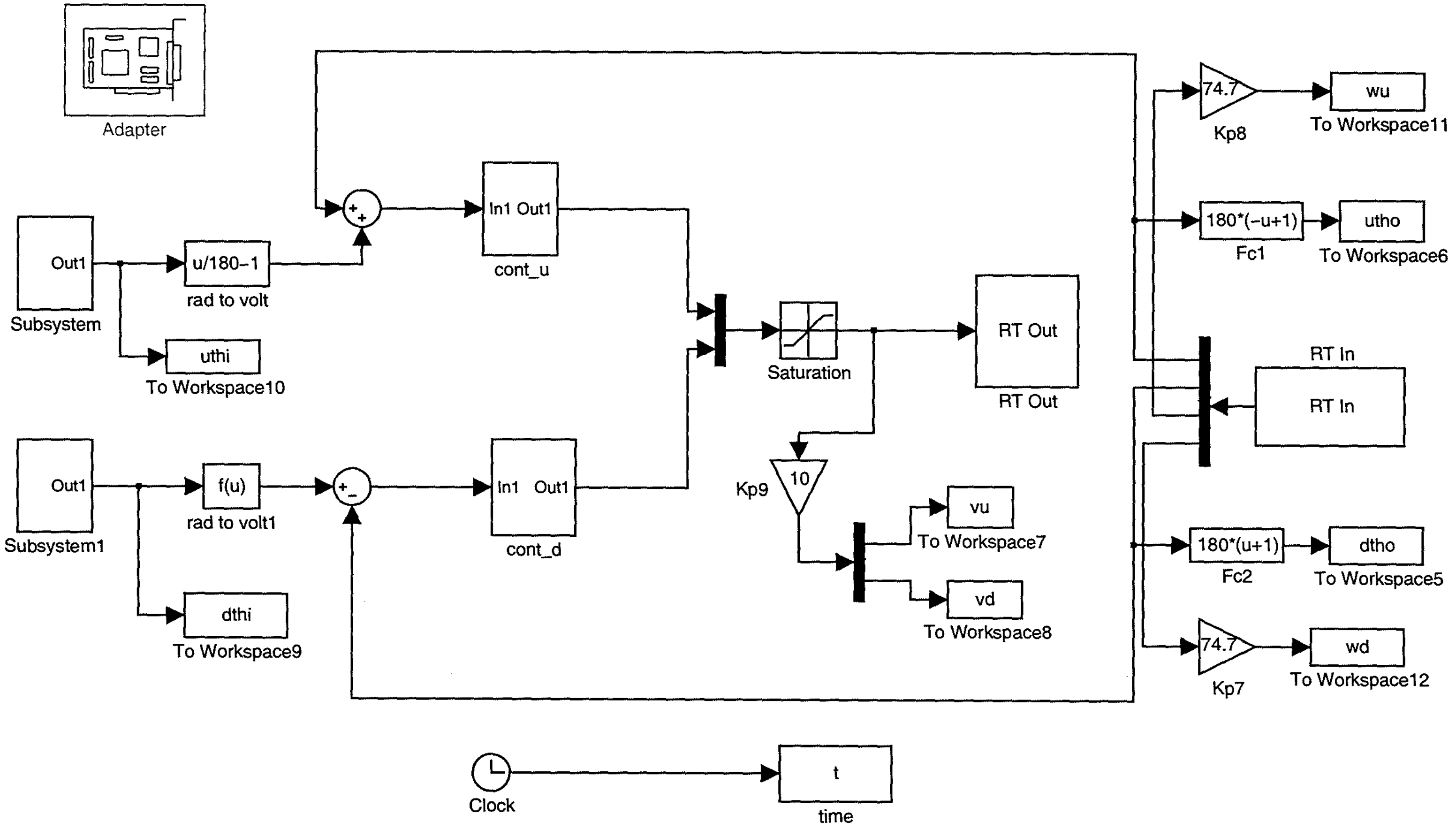


Fig. 6.7: SIMULINK block diagram for experiments.

The results in this case for the system responses and the angular velocities of each input link are shown in Figures 6.8-6.11. The results show a little difference between the actual system and the theoretical model. This is an expected result due to the following factors,

- The uncertainty in the model parameters.
- The fact that the exact friction model is unknown.
- The noise in the measured signals.
- The unknown dynamics of the electrical components, and the effects of the sampling rate which potentially has a great impact on the system.
- Ignoring the flexibility of the links and the joints.
- Ignoring the manufacturing errors in the arm.

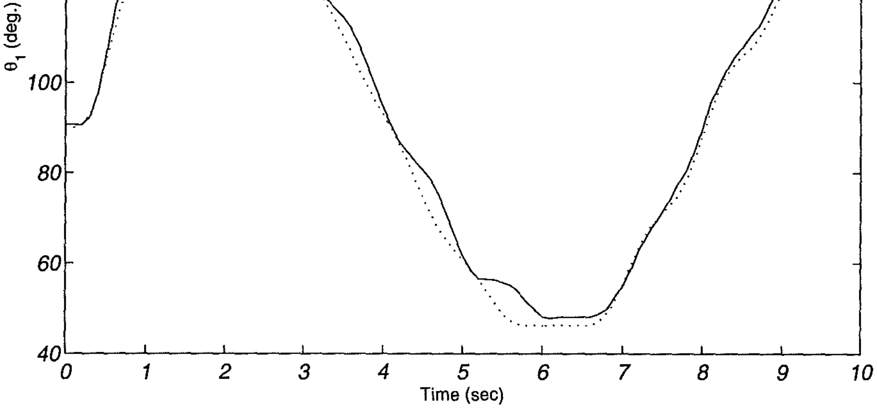


Fig. 6.8: Angular position of the first input link (*P Control*).

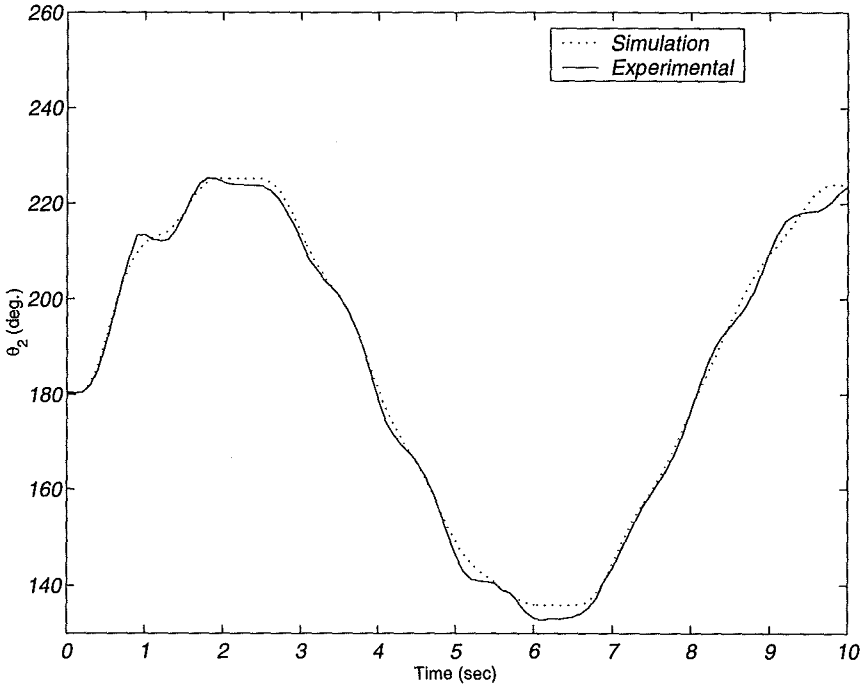


Fig. 6.9: Angular position of the second input link (P Control).

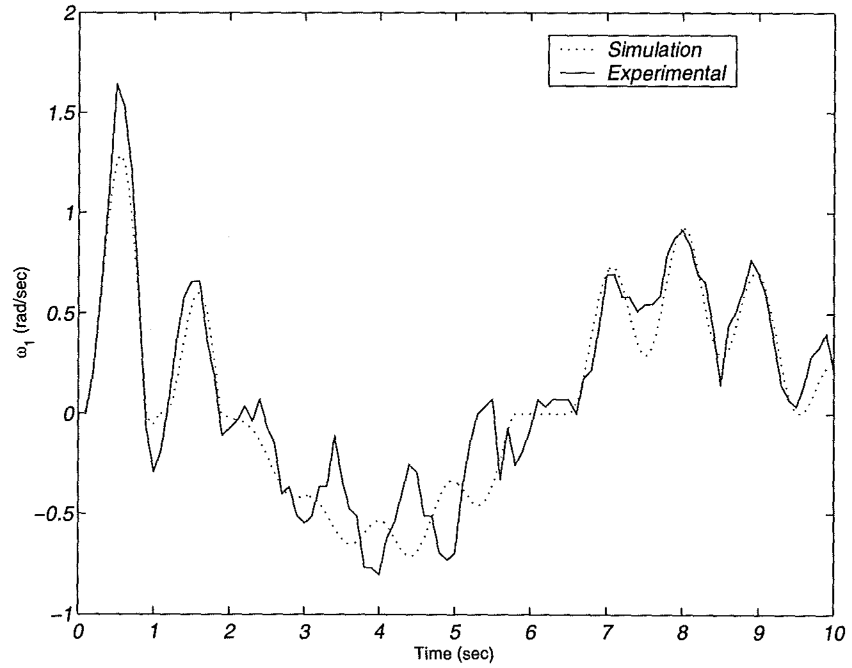


Fig. 6.10: Angular speed of the first input link (*P control*).

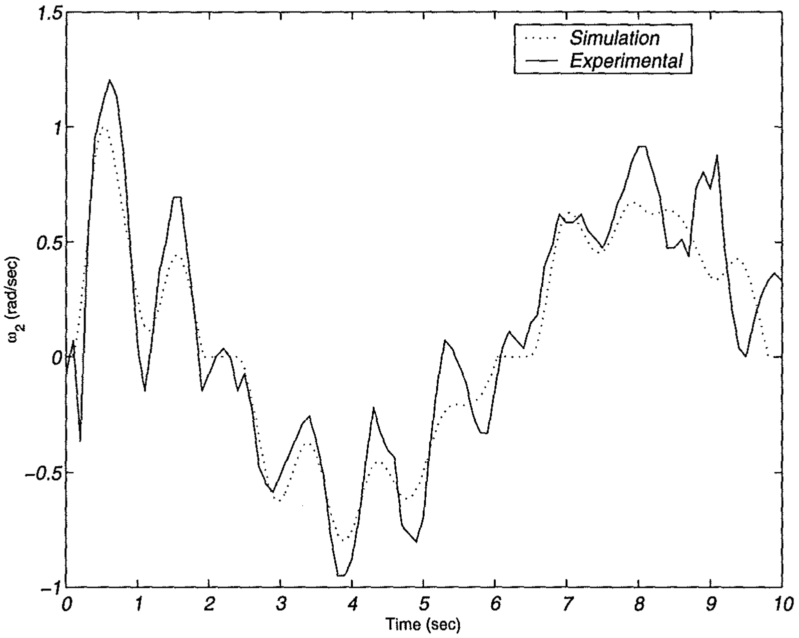


Fig. 6.11: Angular speed of the second input link (*P control*).

It is important to refer here to the fact that despite there being no effect on the validation process due to the nature of the experiments (open loop or closed loop) it was decided to use only closed loop experiments in order to prevent damage of the system due to the case of singularity which may occur as a result of uncontrollable signals to the arm during the open loop experiments. To make sure that the type of the controller has no effect on the results, the preceding experiments and simulations have been repeated, but this time a simple independent joint proportional and derivative controller has been used. The results in this case are shown in Figures 6.12-6.15. From these diagrams it can be seen that the results are still essentially the same as in the case of the proportional controller.

6.4 Conclusions

In this chapter the experimental system has been discussed in detail. The specifications of the system components, that is the *actuators*, *sensors*, *links*, *amplifiers*, *computer*, *data acquisitions* and the *control software*, have been given. Following this the theoretical dynamic model has been modified in order take into account features that are in the experimental system and not in the mathematical dynamical equations. In addition to this a simplified friction model has been added to the equations to improve the model. The quality of the derived dynamic equations has been checked out by comparing data collected from the experimental system with data from simulations. In the experiments and the simulations the robot has been subjected to a simple proportional controller. It was preferable to use closed loop experiments in order to avoid the state of singularity that the arm may be brought into due to uncontrollable signals in the open loop experiments. The results of this comparison show little difference between the actual and the theoretical model. The results are acceptable, as the dynamic model is not perfect due to the reasons mentioned earlier in the chapter. To prove that the controller has no effect on the validation results the previous experiments and simulations have been repeated by using a simple linear proportional and derivative controller. No significant difference has been found between these results and those obtained from the proportional controller.

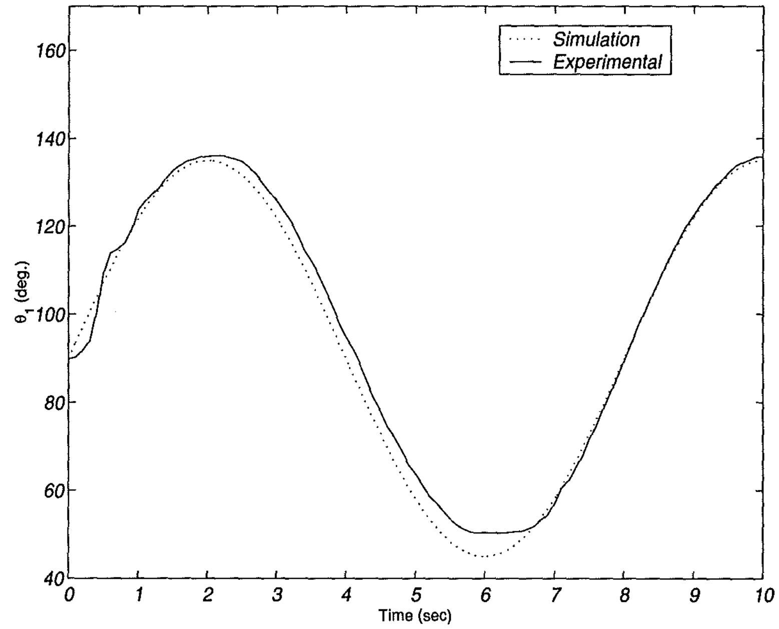


Fig. 6.12: Angular position of the first input link (*PD Control*).

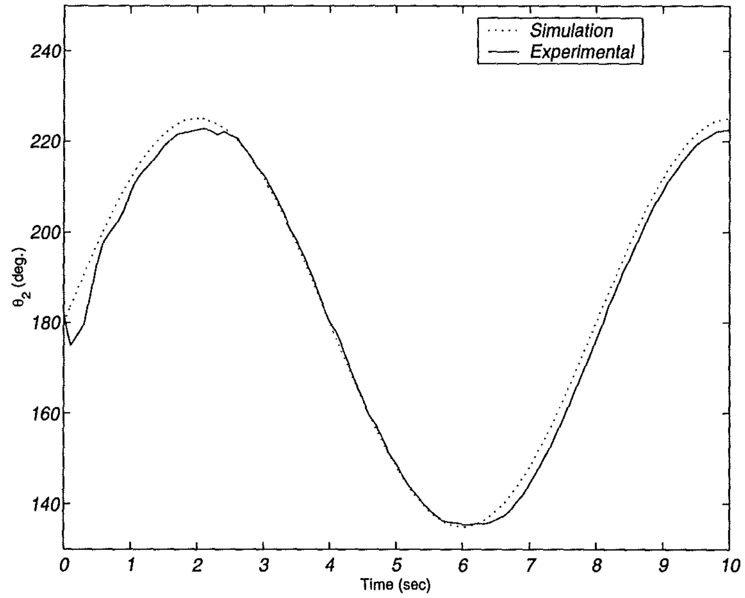


Fig. 6.13: Angular position of the second input link (*PD Control*).

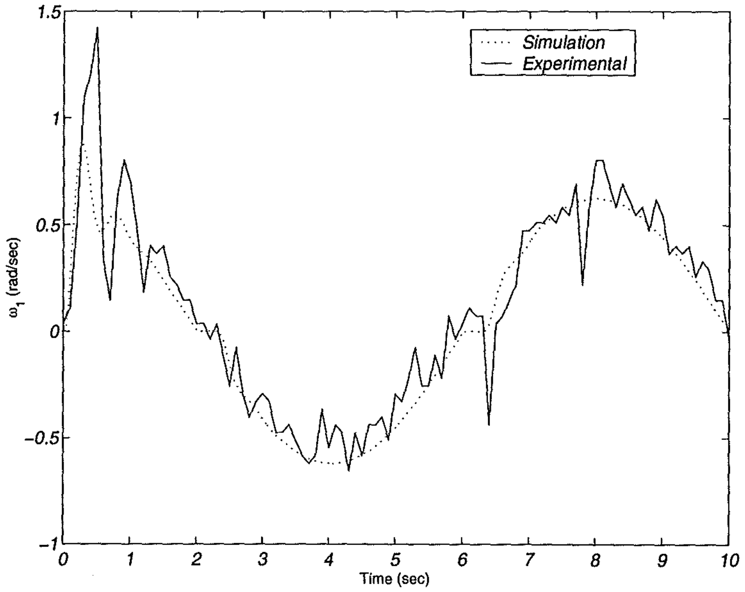


Fig. 6.14: Angular speed of the first input link (*PD Control*).

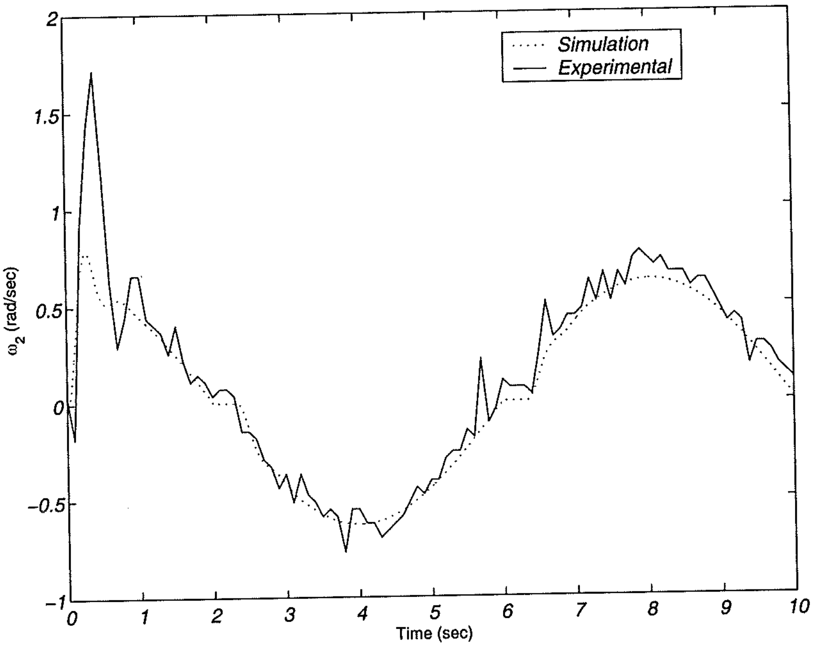


Fig. 6.15: Angular speed of the second input link (*PD Control*).

Chapter 7

Inverse Model-Based Control

7.1 Introduction

In this chapter the inverse model-based control techniques reviewed in chapter one, namely the computed torque and the feed-forward controllers, are applied to the experimental parallel manipulator. This is in order to examine the effectiveness of these methods in controlling the parallel robot and to examine any practical problems that could have arisen while using the Real-Time Toolbox in such a complicated control task. Because of the unactuated joints in the robot, and the difficulty in identifying their friction parameters or their friction models, it was decided to add an integral control action to the servo part (PD controller) of the control systems mentioned above.

7.2 Experiments

In the experiments the arm is forced to follow a prescribed trajectory represented by the following,

$$\theta_{1d}(t)=\frac{\pi}{2}-\frac{\pi}{4}\cos(\omega t) \qquad \text{(rad)}$$

$$\theta_{2d}(t)=\pi-\frac{\pi}{4}\cos(\omega t) \qquad \text{(rad)}$$

Two values for ω have been chosen: $\omega=\frac{\pi}{4}$ and $\frac{\pi}{2}$ (rad/s). The signals from the tachometers have been filtered using a first order Butterworth analogue filter with a cut-off frequency of 20 Hz. The minimum sampling-rate that can be achieved to perform the experiments is 100 Hz. The chosen Simulink solver was the ODE 45 routine. In order to compensate for the unknown friction torques at the joints an integral term has been added to the PD servo controllers. Practically this is desirable because the integral term builds up to remove any steady-state error.

7.2.1 Feed-Forward Control

The control law in this case is given by the following equation,

$$\tau_C = \tau_{\text{model}} + K_P E + K_V \dot{E} + K_I \int E dt \qquad (7.1)$$

Here, τ_{model} is the vector of control signals, in volts, calculated from the dynamic model. Because the gains in the above equation are essentially the same as for the independent joint control, they have been chosen by trial and error to give the maximum desirable performance. This was achieved by using simulations and then implementing the obtained values for the experimental arm. Because of the uncertainty in the friction parameters the gains are slightly different from the real ones. The practical values of these gains are as follows,

$$K_P = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, K_V = \begin{bmatrix} 0.08 & 0 \\ 0 & 0.05 \end{bmatrix} \text{and } K_I = \begin{bmatrix} 2.5 & 0 \\ 0 & 2.5 \end{bmatrix}$$

It is important to refer here to the fact that in the control law stated above E and $\dot{E}$ are in V and Vs^{-1}. The Simulink block diagram of the control system is shown in Figure 7.1.

7.2.2 Computed-Torque Control

The control law is given by the following equation:

$$\tau_C = M(\Theta)\left(\ddot{\Theta}_d + K_P E + K_V \dot{E} + \int E dt\right) + V_m\left(\Theta, \dot{\Theta}\right)\dot{\Theta} \qquad (7.2)$$

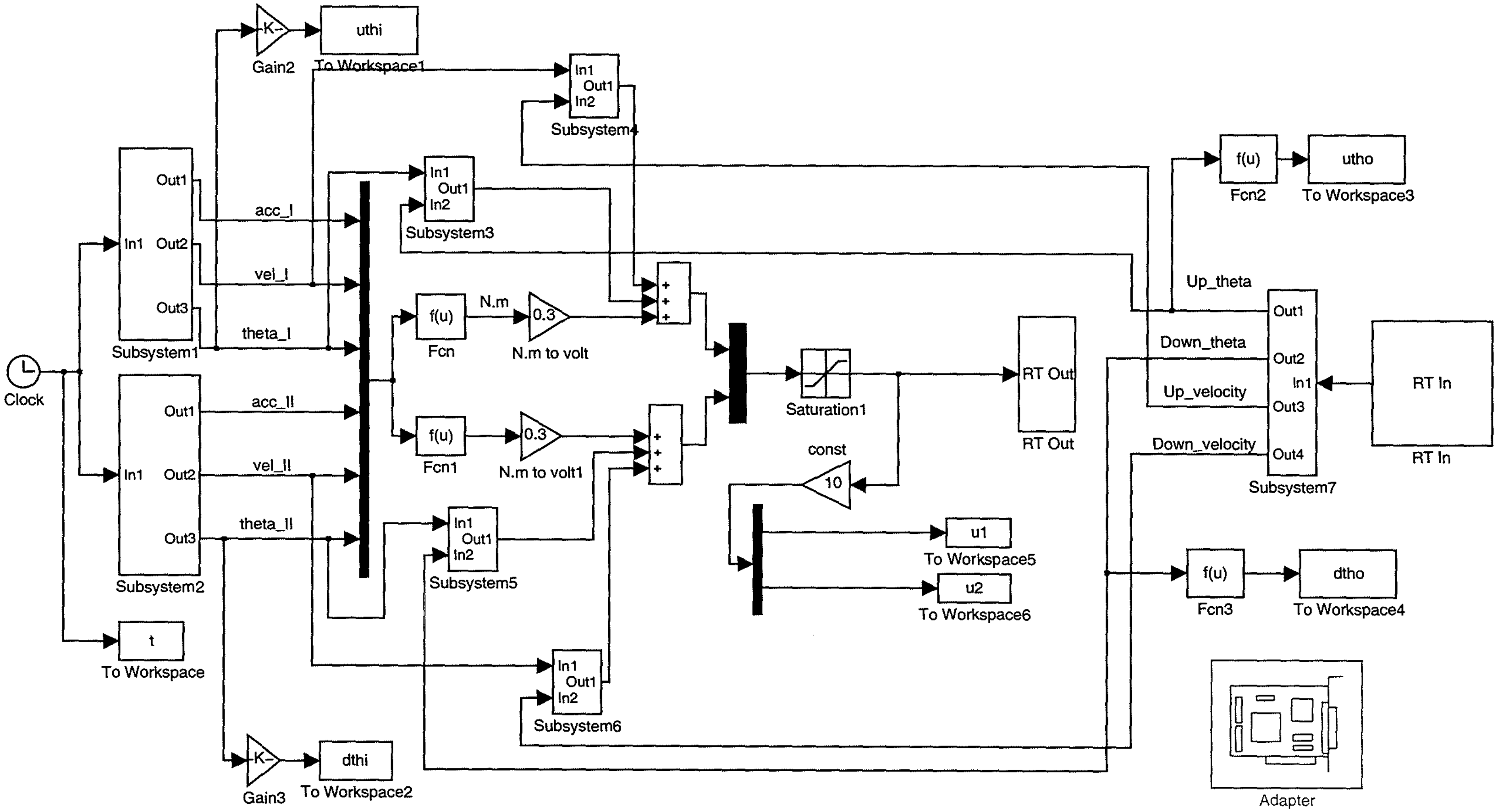


Fig. 7.1: Simulink block diagram for the feed-forward controller.

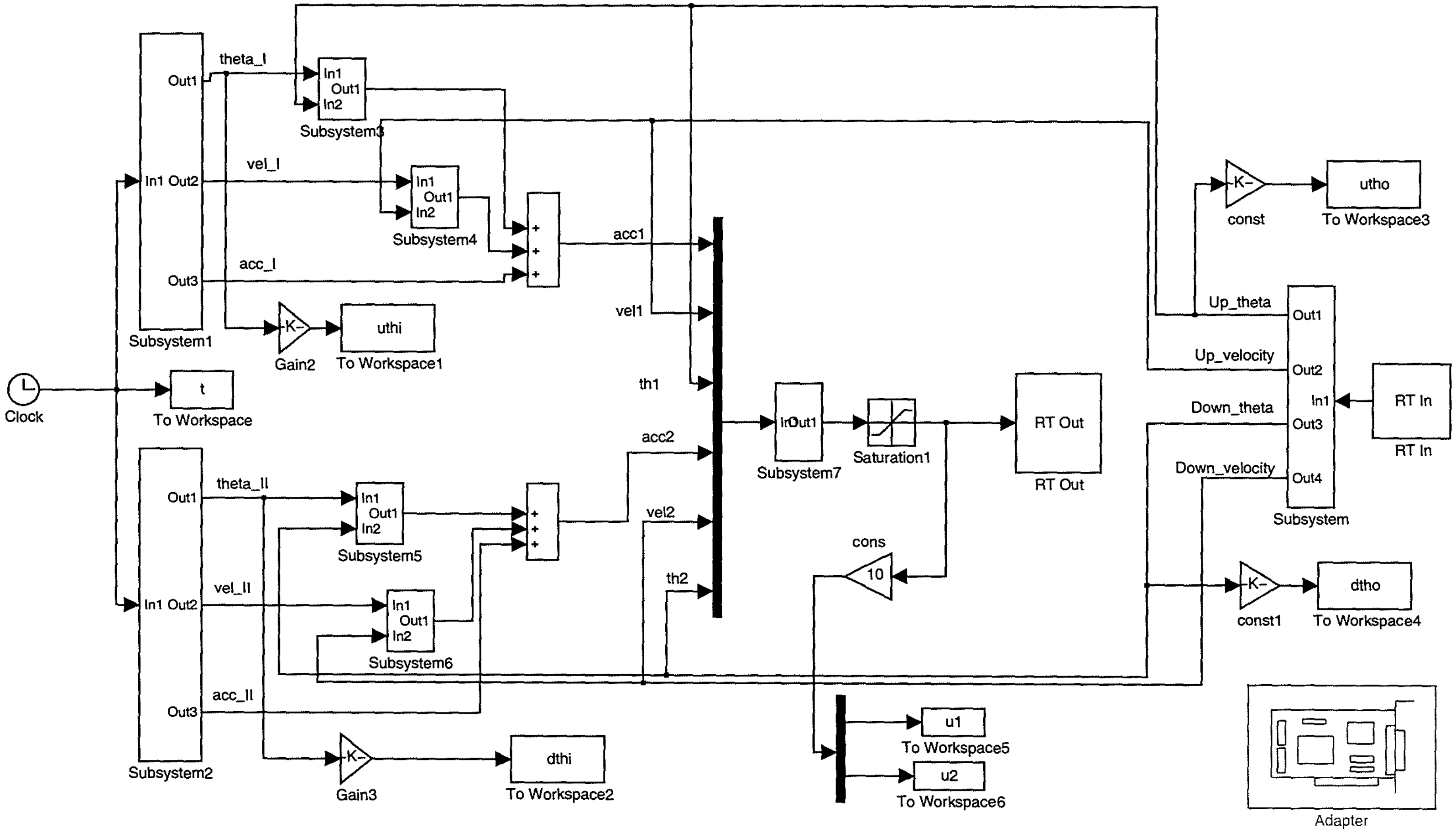


Fig. 7.2: Simulink block diagram for the computed–torque controller.

Where $M(\Theta)$ is the mass matrix calculated form the model, $V_m(\Theta,\dot{\Theta})\dot{\Theta}$ is the vector of Coriolis and Centrifugal forces, Θ is the vector of output angular displacements, $\dot{\Theta}$ is the vector of output angular velocities and $\ddot{\Theta}_d$ is the desired acceleration vector which is the second time derivative of the desired trajectory.

Due to the existence of the integral term the theoretical (assuming an exact model) system equation in the error space, that is equation 1.9, becomes as follows,

$$\ddot{E} + K_V\dot{E} + K_P E + K_I\int E dt = 0 \qquad (7.3)$$

In this case, the system order is increased to order six. Because the gain matrices K_V, K_P and K_I are diagonal, the error equation for the i^{th} degree of freedom can be written as follows,

$$\ddot{e}_i + k_{V_i}\dot{e}_i + k_{P_i}e + k_{I_i}\int e_i\, dt = 0 \qquad (7.4)$$

In order to ensure that the system is stable the gains are chosen according to the Routh-Hurwitz criterion [79]. Using equation 7.4, and this criterion, the gain should satisfy the following conditions,

$$k_{V_i}, k_{I_i} > 0 \text{ and } k_{V_i}k_{P_i} - k_{I_i} > 0 \qquad (7.5)$$

The above conditions ensure that the elements of the first column of the Routh array are all positive. The following values represent the chosen gains,

$$K_P = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, K_V = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix} \text{and } K_I = \begin{bmatrix} 80 & 0 \\ 0 & 80 \end{bmatrix}$$

Following on from this simulations are carried out to check the system performance. It is necessary to refer here to the fact that, here, E is in radians and $\dot{E}$ is in radians per second. The Simulink block diagram of the control system is shown in Figure 7.2.

7.3 Results and Discussion

7.3.1 Feed-Forward Control

In the case of the feed-forward controller and for $\omega = \frac{\pi}{2}$, the results are shown in Figures 7.3-7.8. For $\omega = \frac{\pi}{4}$, the results are shown in Figures 7.10-7.14. The results show moderately good tracking performance in the case of relatively slow motion. However the performance is not good in either case at the points where the system changes direction. This is due to stiction friction. On the other hand the results also show that noise is dominant at slow motion.

7.3.2 Computed-Torque Control

The results in this case, and for $\omega = \frac{\pi}{2}$, are shown in Figures 7.15-7.20. For $\omega = \frac{\pi}{4}$, the results are shown in Figures 7.21-7.26. Again the results of relatively slow motion are better than those for faster motion. This is attributed to the limited sampling rate that is obtainable. Here the results show superior tracking performance than in the case of feed-forward control. In addition to this the control law is able here to compensate for the unknown friction. The apparently poor performance of the feed-forward control is attributed to the fact that the method does not provide complete decoupling (the corrective torques in the feed-back control of one joint perturb all other joints).

7.4 Disturbance Rejection

To examine the ability of the two inverse-model based control techniques to reject disturbances, the previous experiments have been repeated after assigning initial conditions to the arm. The initial positions of the two input links are as follows:

$$\theta_1(0) = 25 \quad \text{(deg.)} \qquad\qquad \theta_2(0) = 105 \quad \text{(deg.)}$$

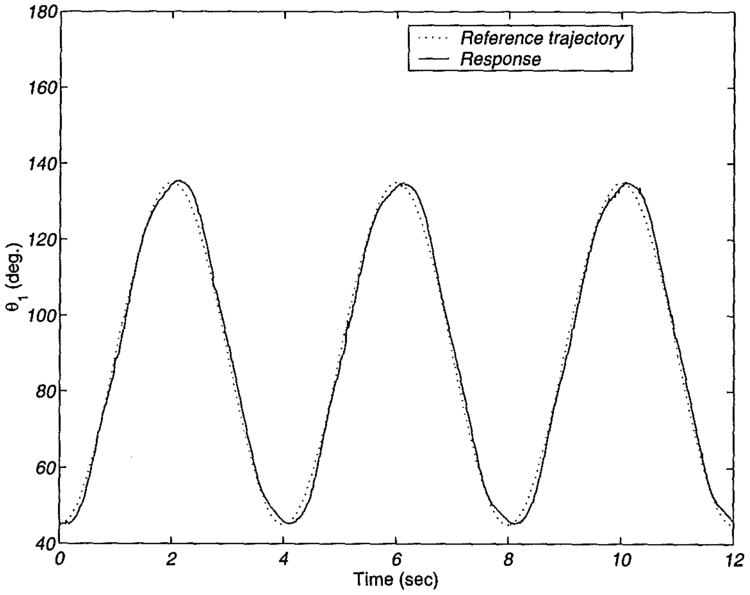


Fig. 7.3: Angular position of the first input link (Feed-Forward and $\omega = \frac{\pi}{2}$)

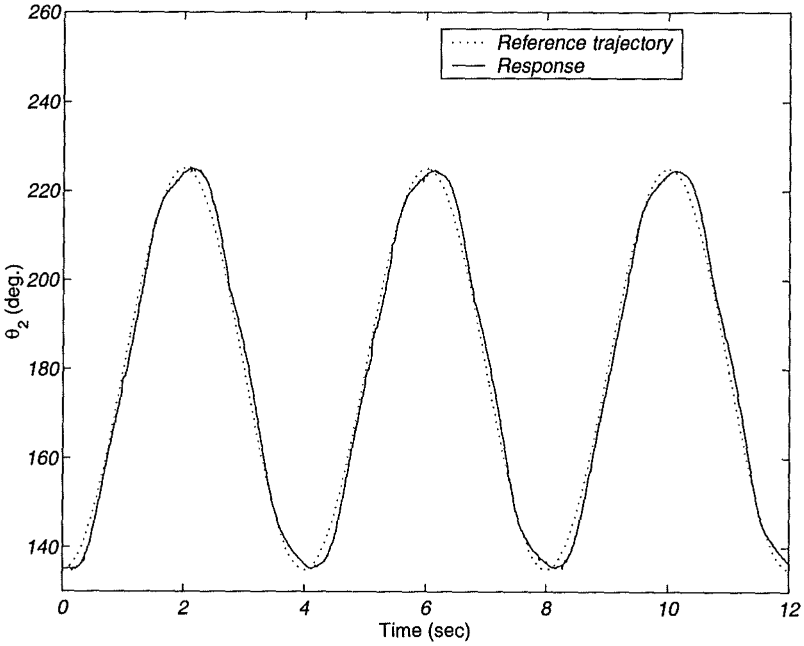


Fig. 7.4: Angular position of the second input link (Feed-Forward and $\omega = \frac{\pi}{2}$)

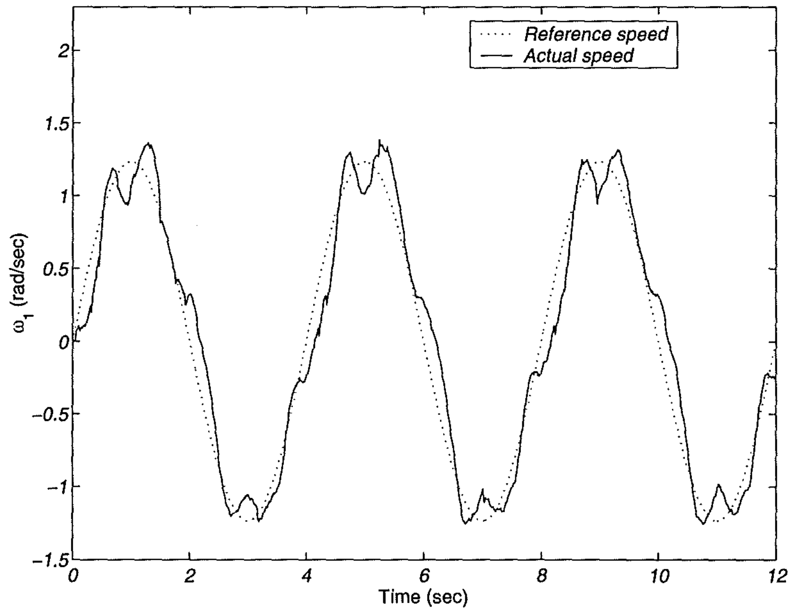


Fig. 7.5: Angular speed of the first input link (Feed-Forward and $\omega = \frac{\pi}{2}$)

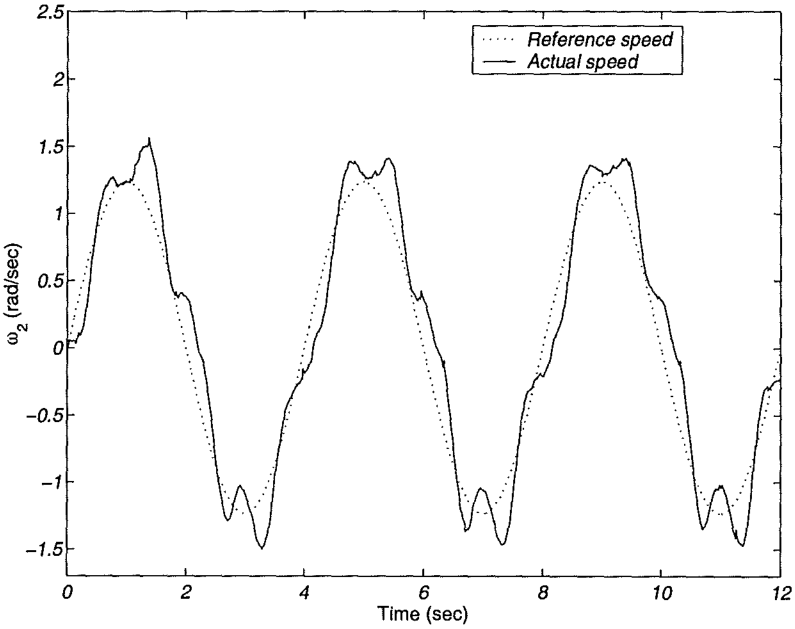


Fig. 7.6: Angular speed of the second input link (Feed-Forward and $\omega = \frac{\pi}{2}$)

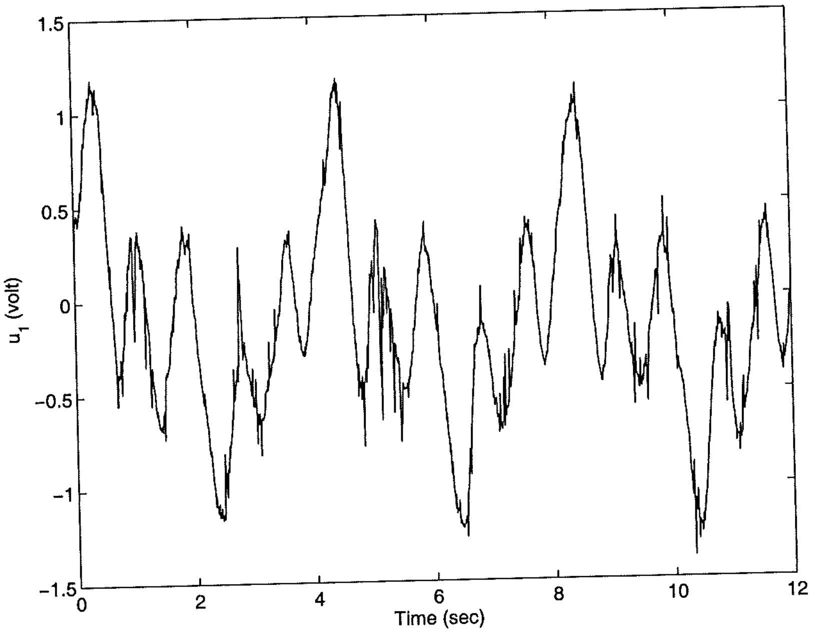


Fig. 7.7: Control signal of the first input link (Feed-Forward and $\omega = \frac{\pi}{2}$)

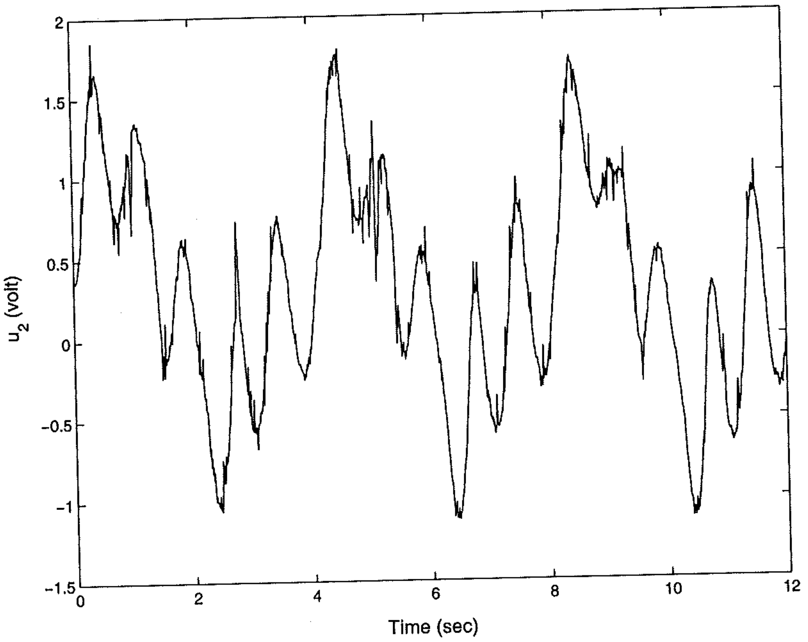


Fig. 7.8: Control signal of the second input link (Feed-Forward and $\omega = \frac{\pi}{2}$)

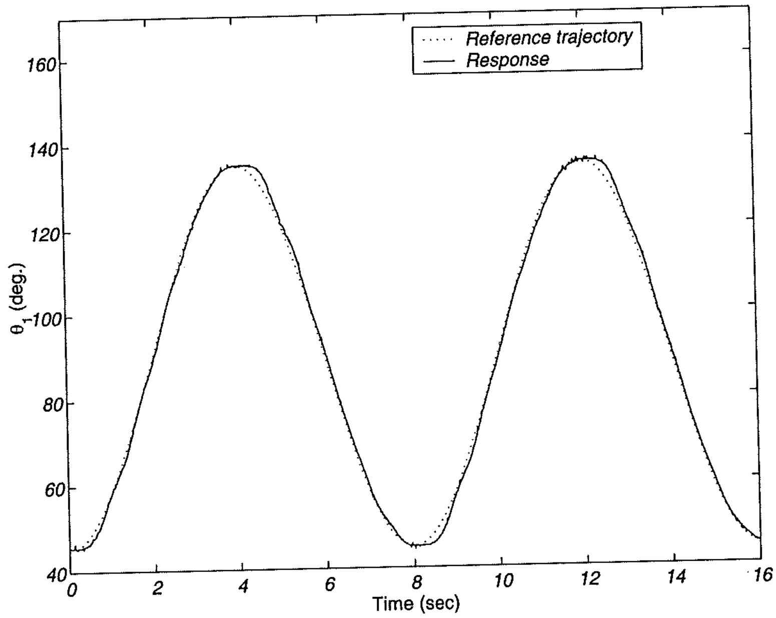


Fig. 7.9: Angular position of the first input link (Feed-Forward and $\omega = \frac{\pi}{4}$)

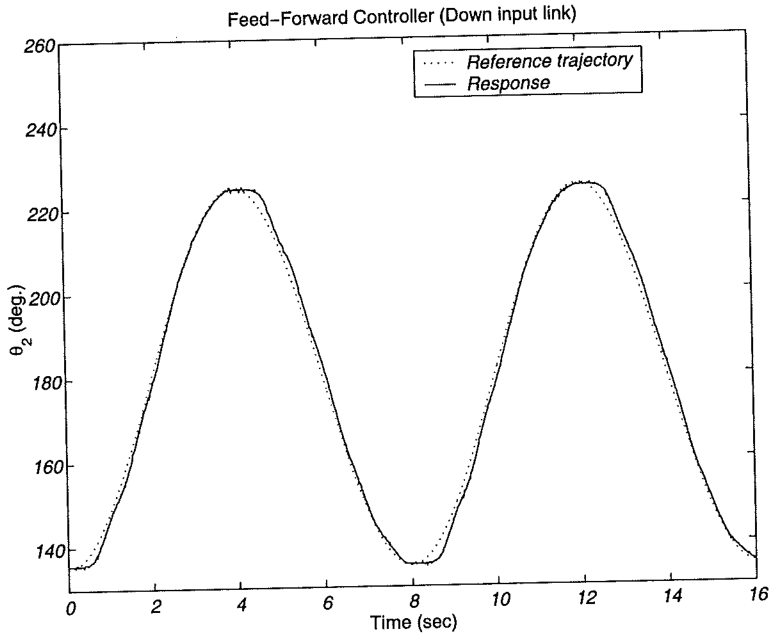


Fig. 7.10: Control signal of the second input link (Feed-Forward and $\omega = \frac{\pi}{4}$)

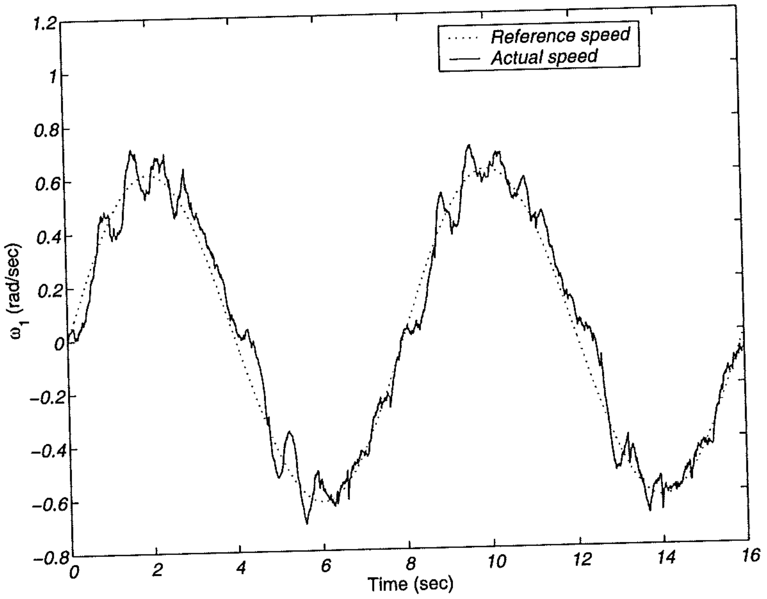


Fig. 7.11: Angular speed of the first input link (Feed-Forward and $\omega = \frac{\pi}{4}$)

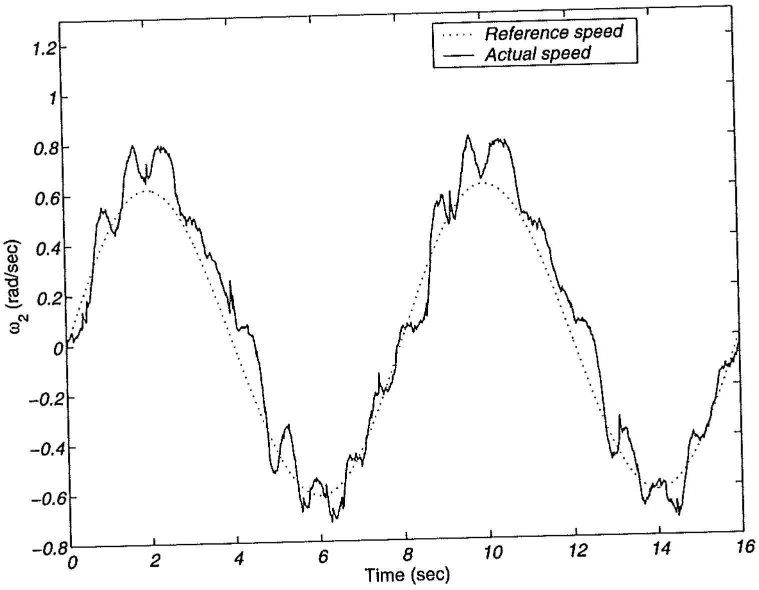


Fig. 7.12: Angular speed of the second input link (Feed-Forward and $\omega = \frac{\pi}{4}$)

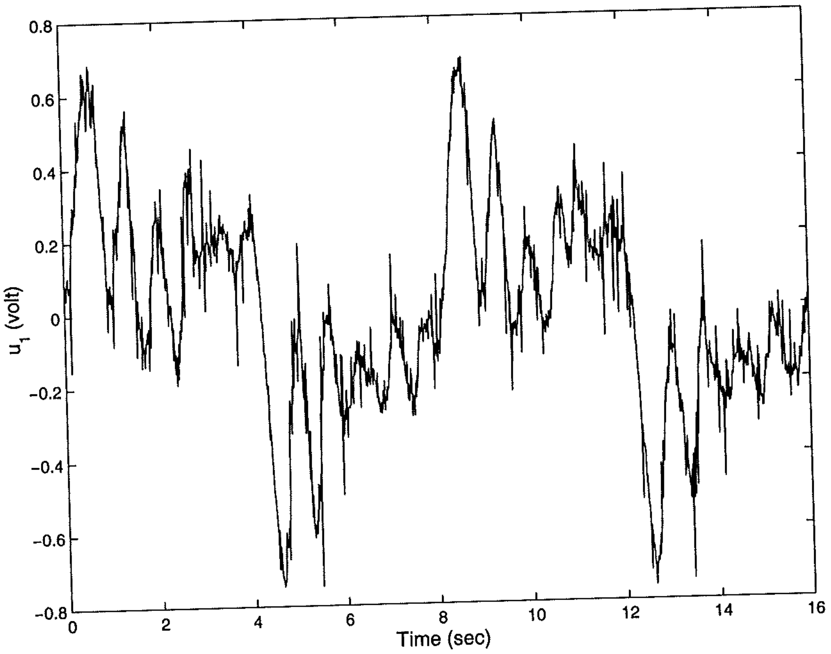


Fig. 7.13: Control signal of the first input link (Feed-Forward and $\omega = \frac{\pi}{4}$)

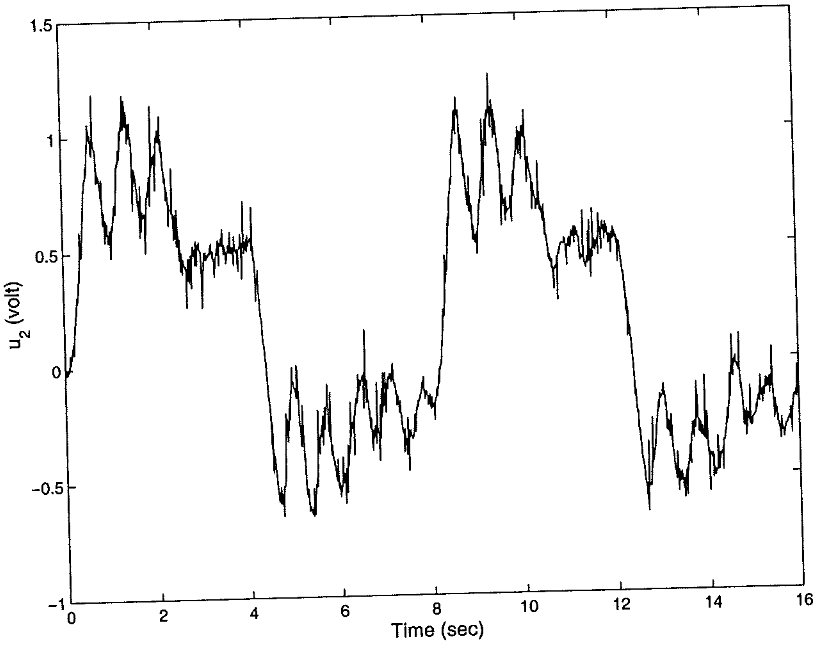


Fig. 7.14: Control signal of the second input link (Feed-Forward and $\omega = \frac{\pi}{4}$)

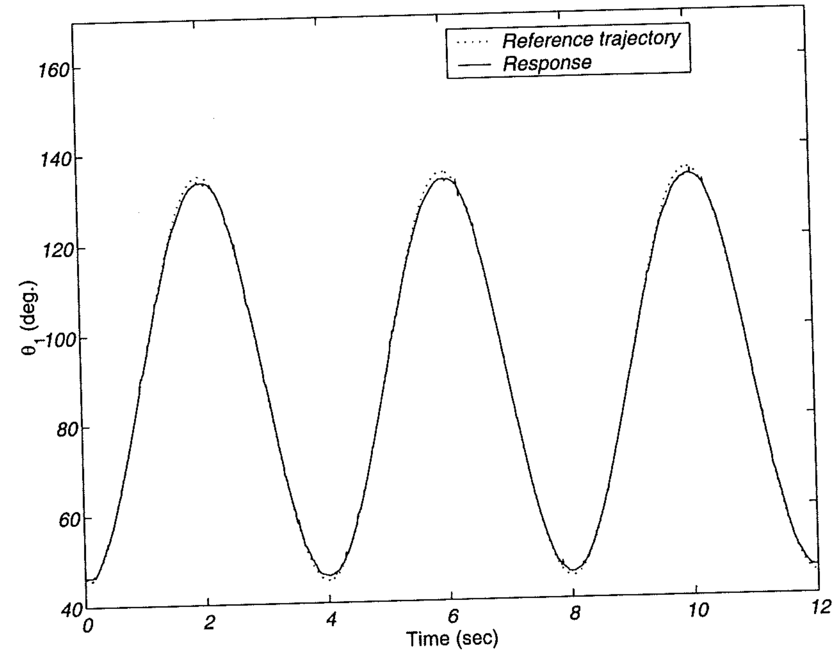


Fig. 7.15: Angular position of the first input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

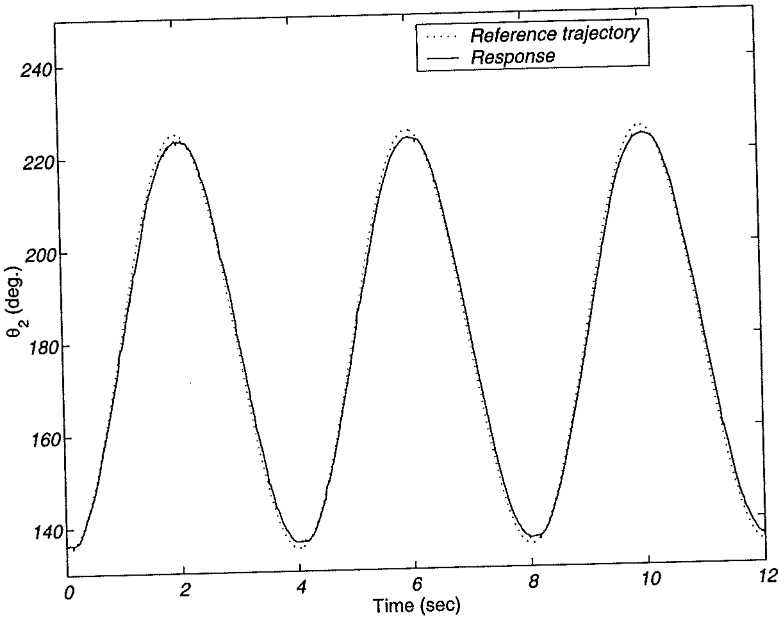


Fig. 7.16: Angular position of the second input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

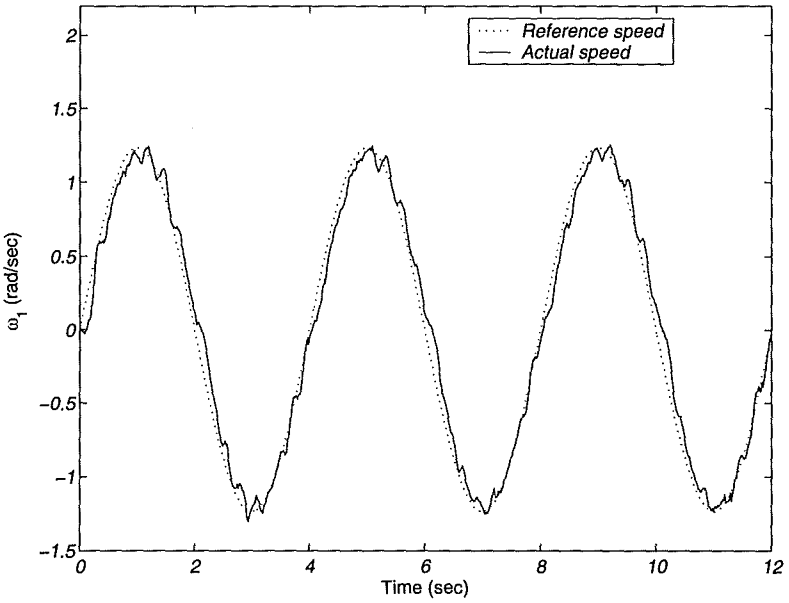


Fig. 7.17: Angular speed of the first input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

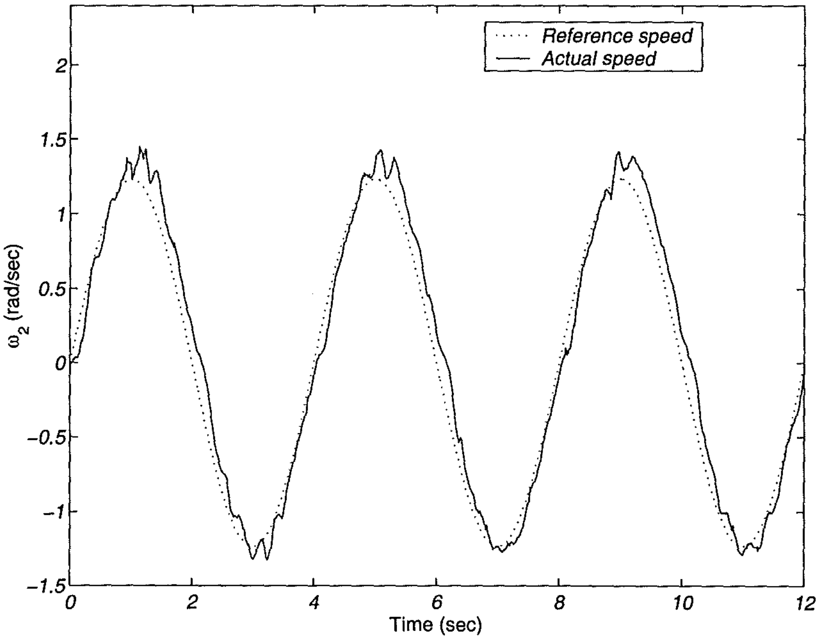


Fig. 7.18: Angular speed of the second input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

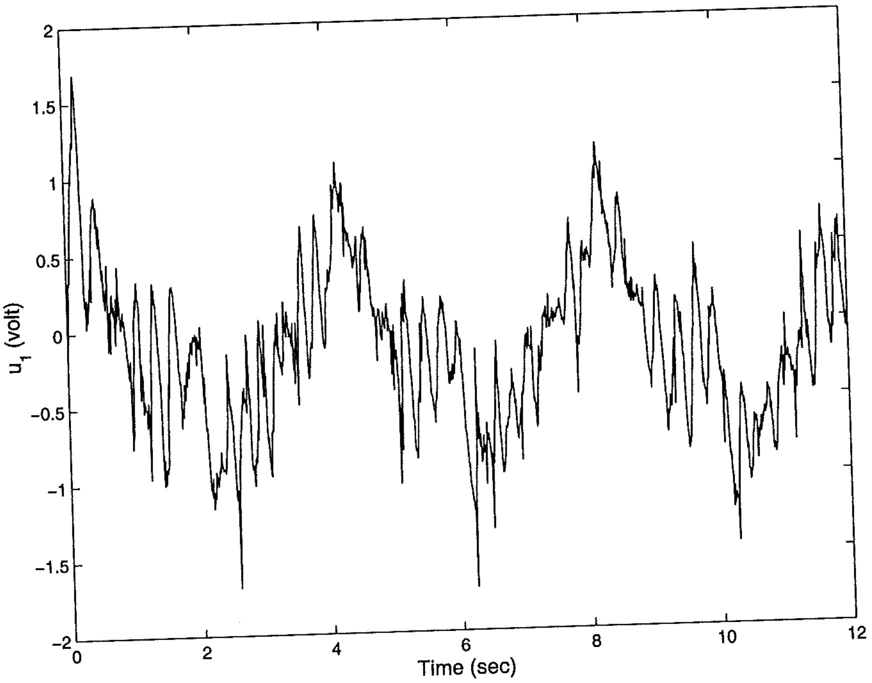


Fig. 7.19: Control signal of the first input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

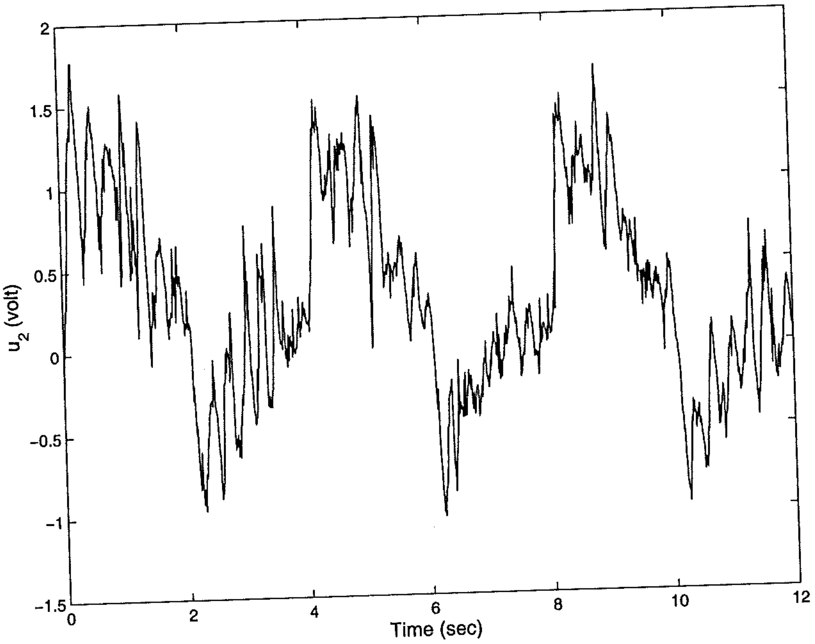


Fig. 7.20: Control signal of the second input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

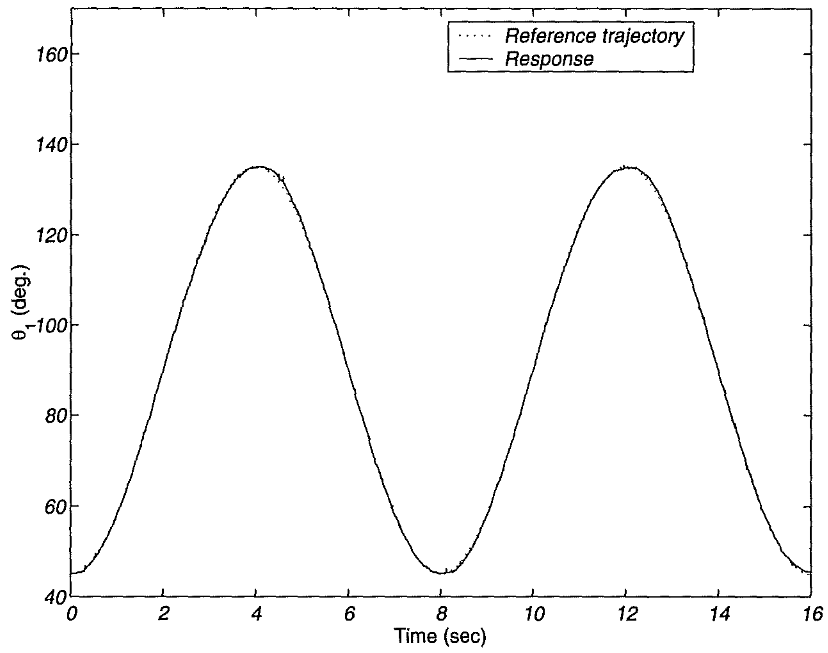


Fig. 7.21: Angular position of the first input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

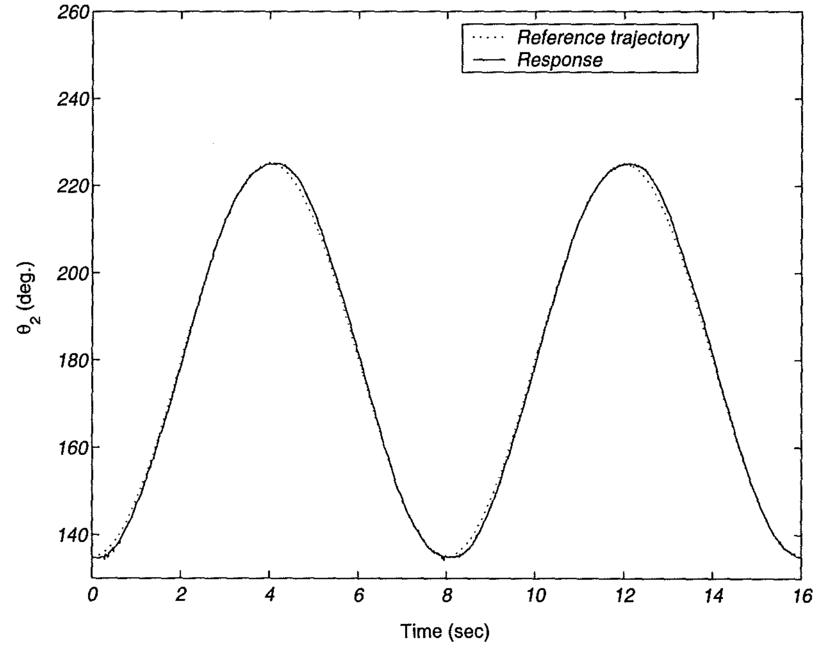


Fig. 7.22: Angular position of the second input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

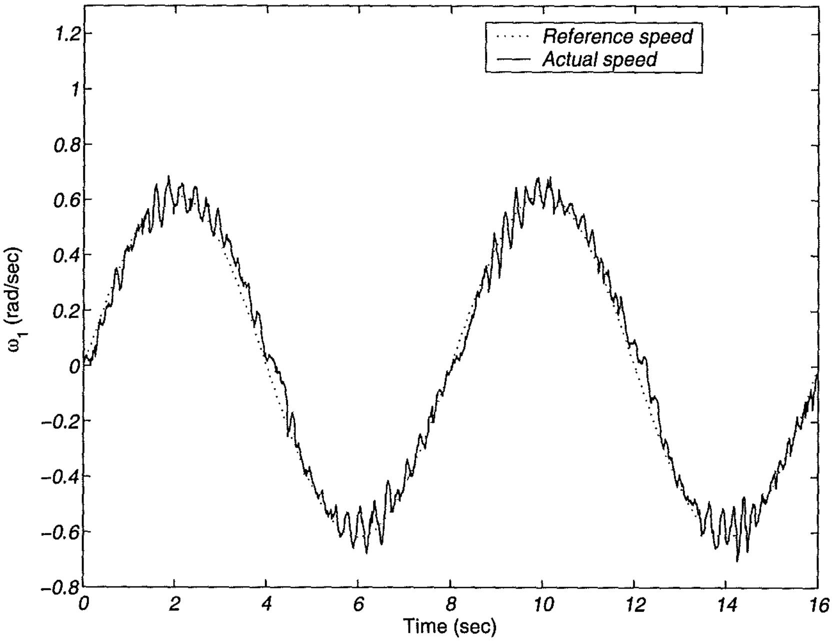


Fig. 7.23: Angular speed of the first input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

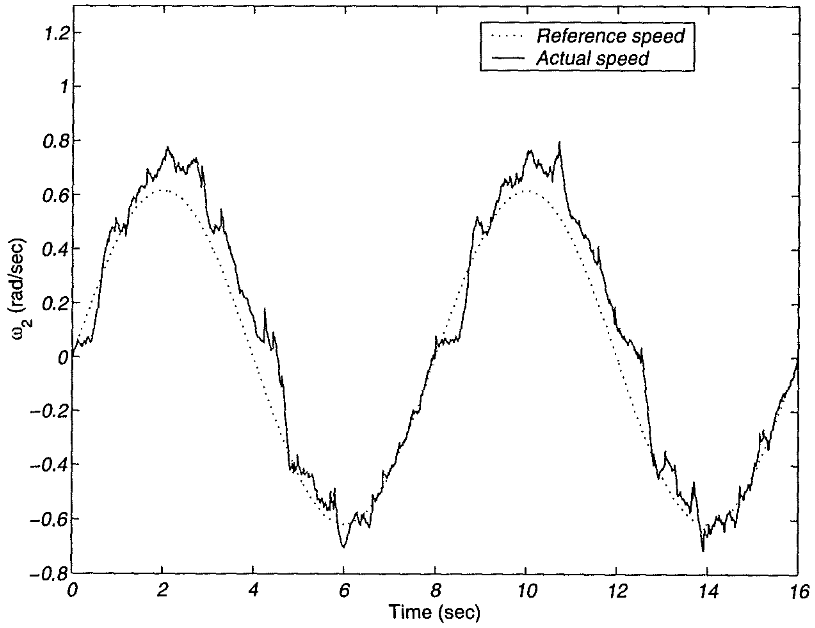


Fig. 7.24: Angular speed of the second input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

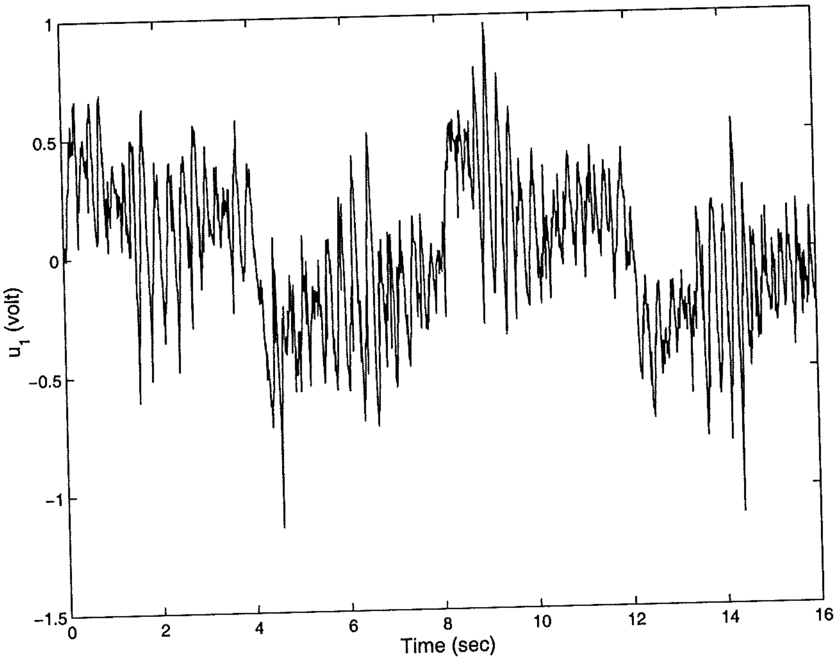


Fig. 7.25: Control signal of the first input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

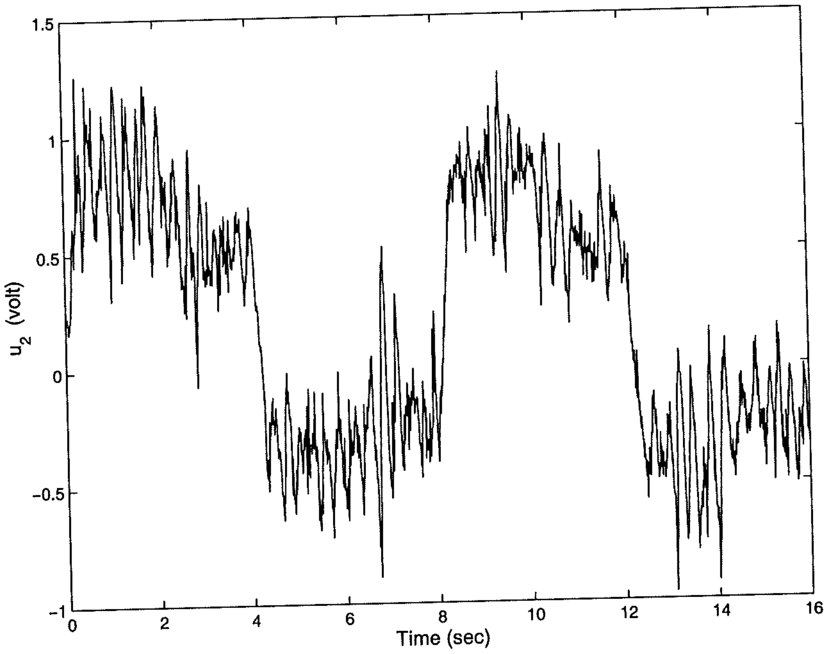


Fig. 7.26: Control signal of the second input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

7.4.1 Results (Feed-Forward Control)

The results for the case of $\omega = \frac{\pi}{2}$ are shown in Figures 7.27-7.32 while for $\omega = \frac{\pi}{4}$ the results are shown in Figures 7.33-7.38. The results obtained show that the controller is able to reject the disturbances represented by the initial conditions of the arm. Despite this result there is still an important drawback which lies in the fact that the system does not converge to its origin quickly. The graphs of the angular speed clearly show this drawback. The performance after rejecting the disturbance is the same as in the previous section.

7.4.2 Results (Computed-Torque Control)

The results of the fast trajectory are shown in Figures 7.39-7.44. In the case of the slow trajectory the results are shown in Figures 7.45-7.50. Here, the performance is very much better than that of the feed-forward control because the convergence to the origin is fast. On the other hand the results of the low speed trajectory show better tracking accuracy than those of the fast trajectory. It is important to note that the disturbance rejection property should be fast for any control system. However this does depend on the sampling rate of the control signals to the actuators.

7.5 Conclusions

In this chapter the two inverse model-based control laws (Feed-forward and Computed torque) have been applied to the experimental parallel robot. The results show that both methods are good at controlling the robot, but the computed-torque controller has been proven to be much better than the feed-forward controller. In both methods the tracking accuracy is very good at slow motion. On the other hand, despite the absence of friction mechanisms incorporated in the dynamic model used in both these control methods a compensation for friction is obtained from the integral action in the servo control part. The best compensation is reported for the case of the computed-torque control. In the disturbance rejection test the computed torque controller has been proved to be better than the feed-forward controller.

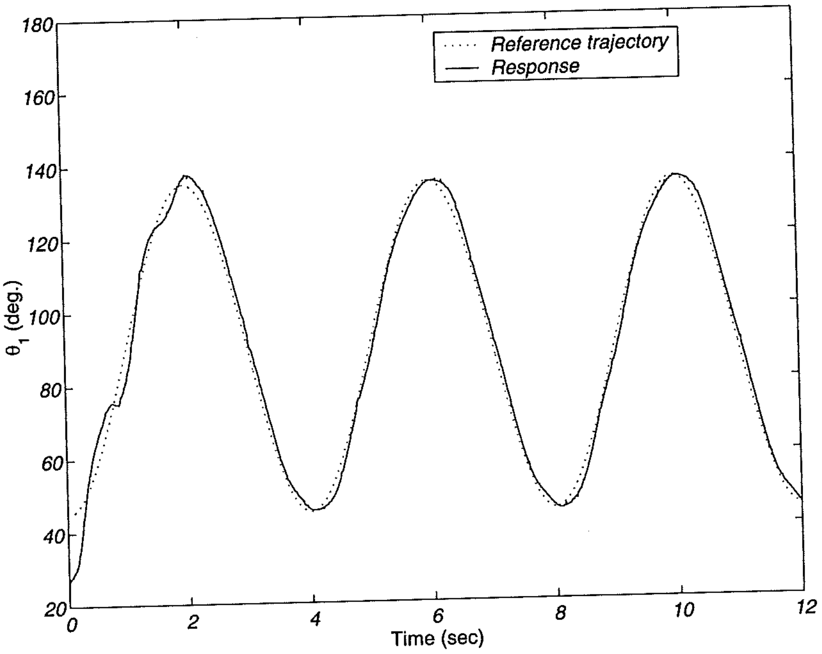


Fig. 7.27: Angular position of the first input link (Feed-Forward and $\omega = \frac{\pi}{2}$).

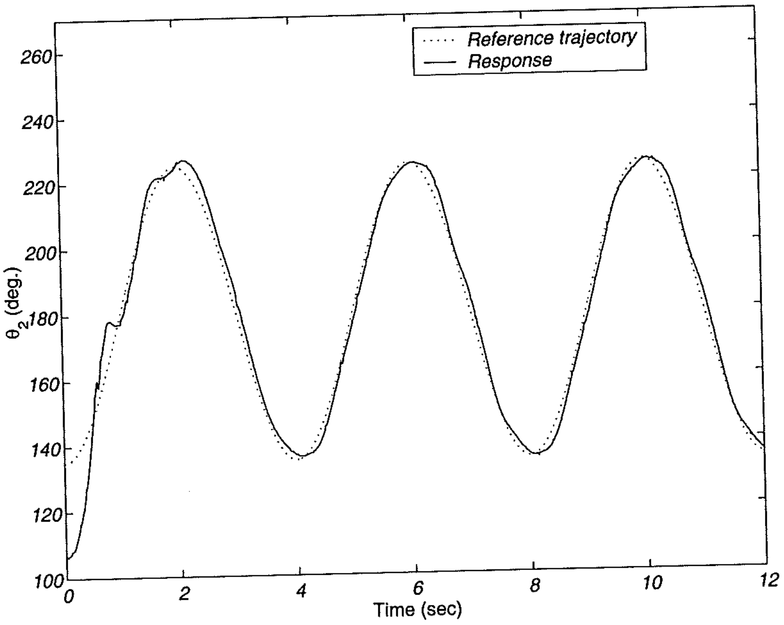


Fig. 7.28: Angular position of the second input link (Feed-Forward and $\omega = \frac{\pi}{2}$).

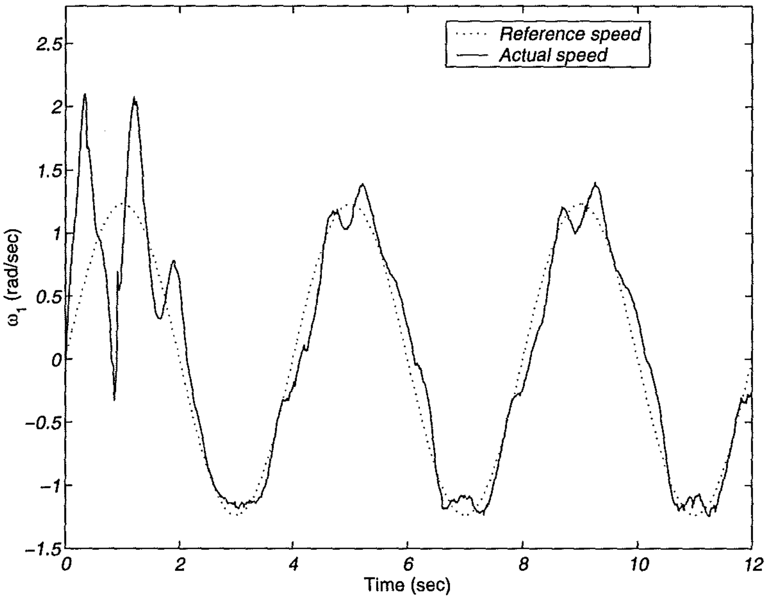


Fig. 7.29: Angular speed of the first input link (Feed-Forward and $\omega = \frac{\pi}{2}$).

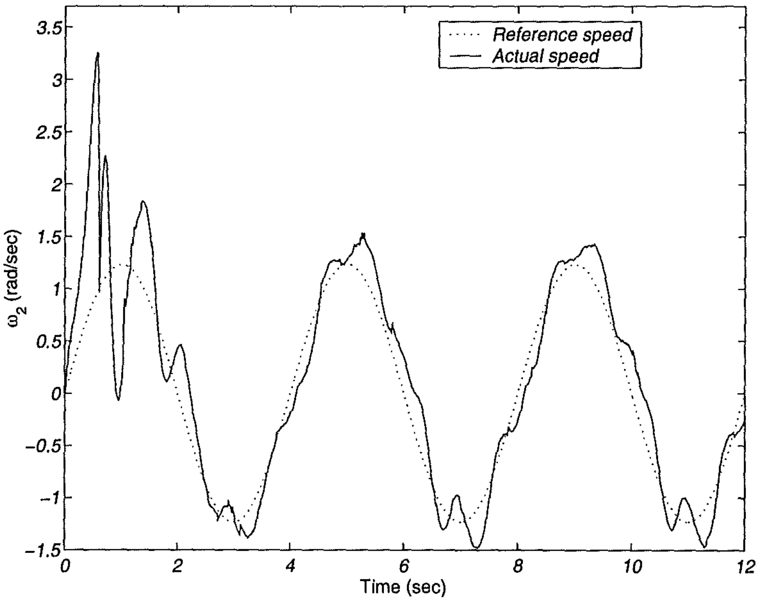


Fig. 7.30: Angular speed of the second input link (Feed-Forward and $\omega = \frac{\pi}{2}$).

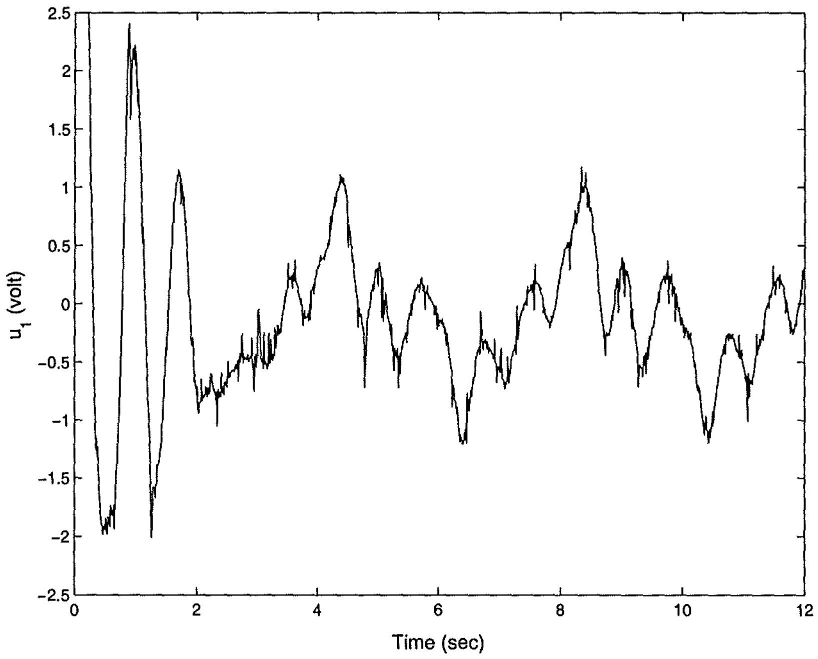


Fig. 7.31: Control signal of the first input link (Feed-Forward and $\omega = \frac{\pi}{2}$).

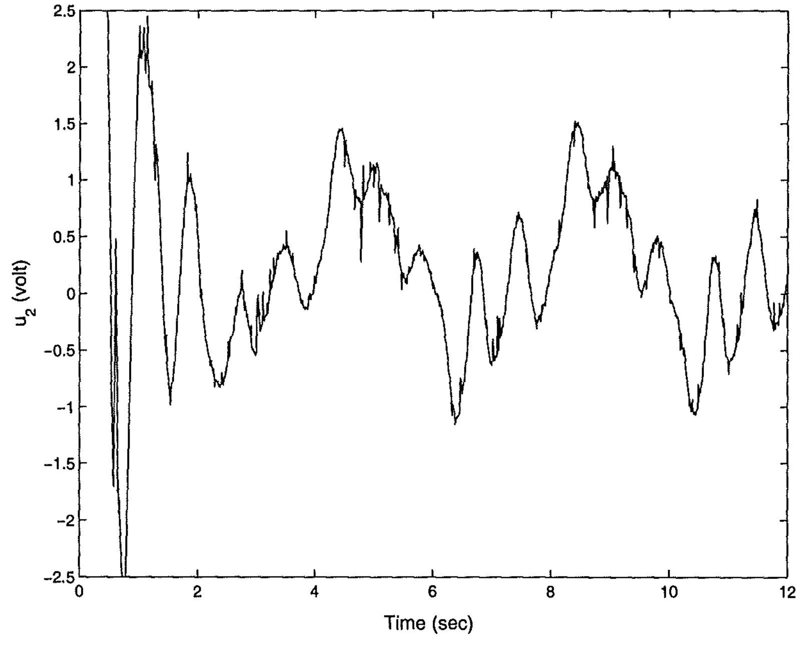


Fig. 7.32: Control signal of the second input link (Feed-Forward and $\omega = \frac{\pi}{2}$).

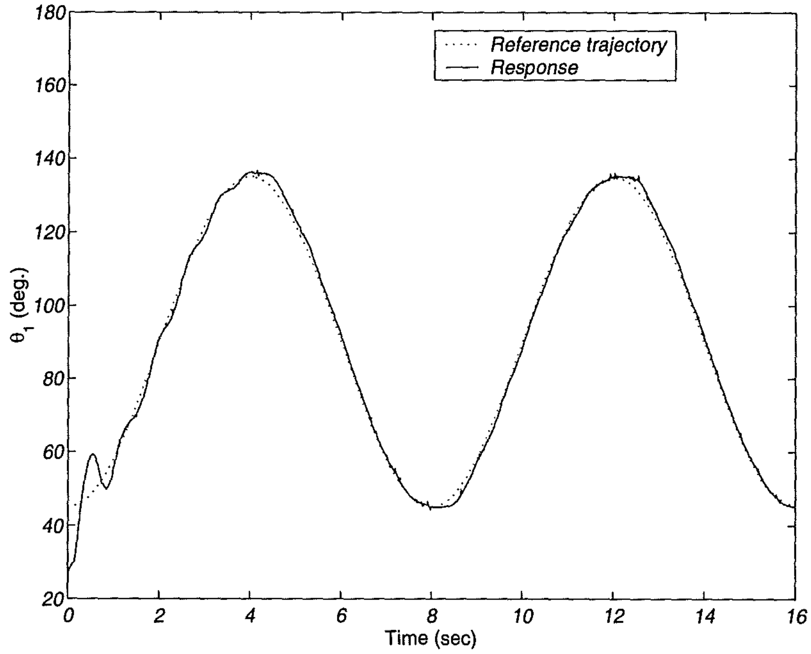


Fig. 7.33: Angular position of the first input link (Feed-Forward and $\omega = \frac{\pi}{4}$).

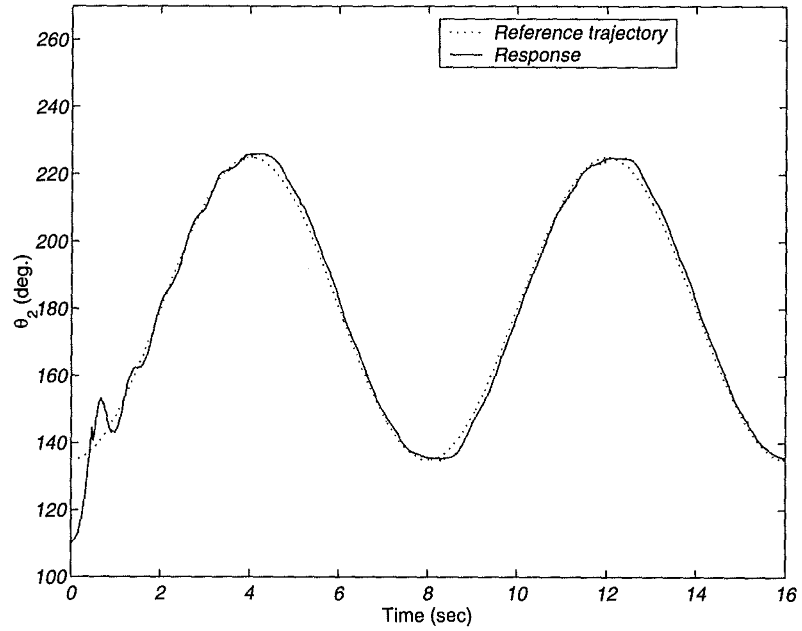


Fig. 7.34: Angular position of the second input link (Feed-Forward and $\omega = \frac{\pi}{4}$).

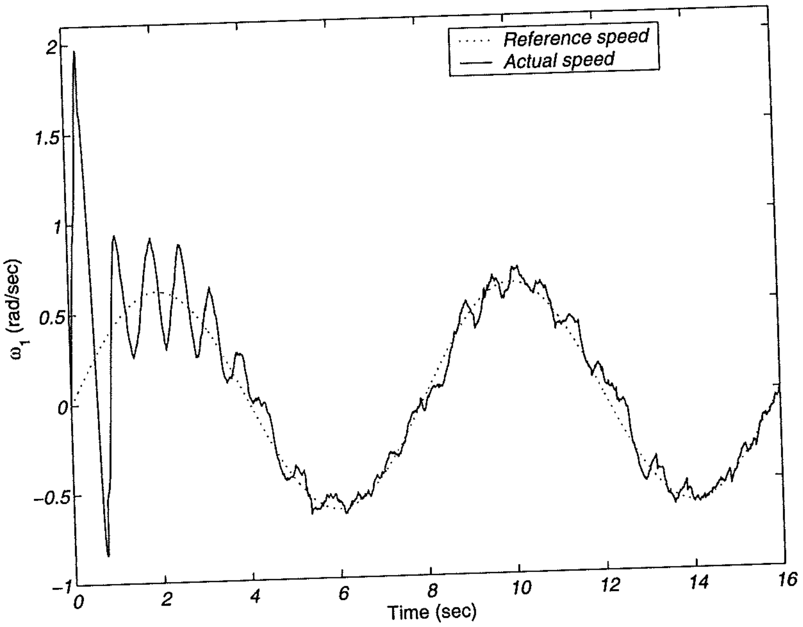


Fig. 7.35: Angular speed of the first input link (Feed-Forward and $\omega = \frac{\pi}{4}$).

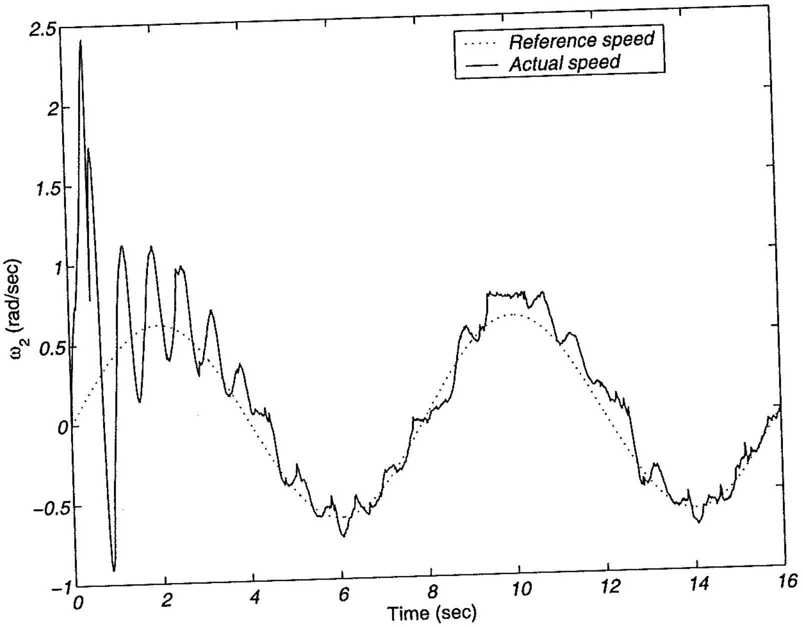


Fig. 7.36: Angular speed of the second input link (Feed-Forward and $\omega = \frac{\pi}{4}$).

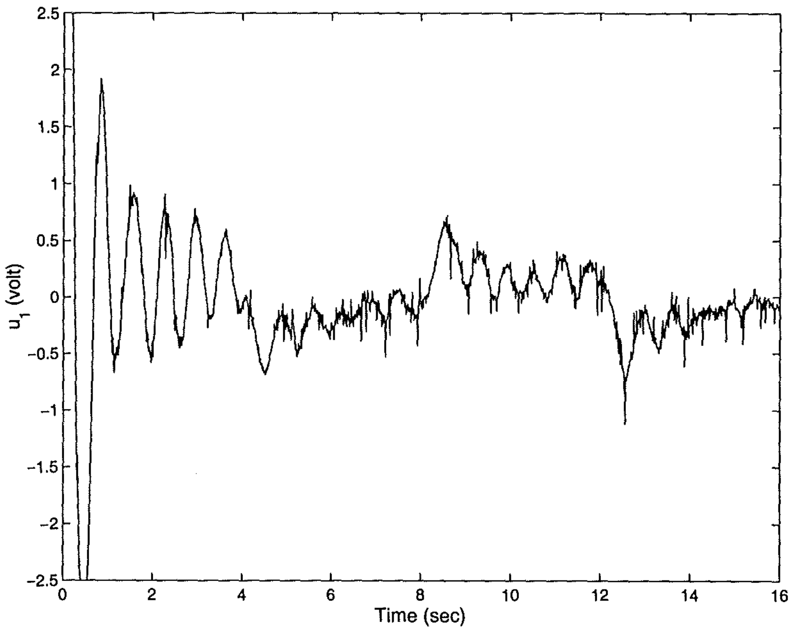


Fig. 7.37: Control signal of the first input link (Feed-Forward and $\omega = \frac{\pi}{4}$).

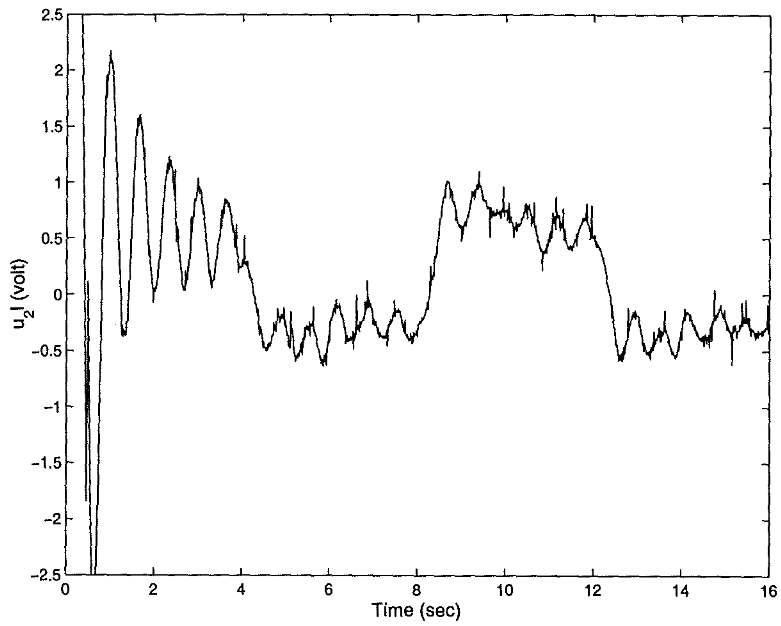


Fig. 7.38: Control signal of the second input link (Feed-Forward and $\omega = \frac{\pi}{4}$).

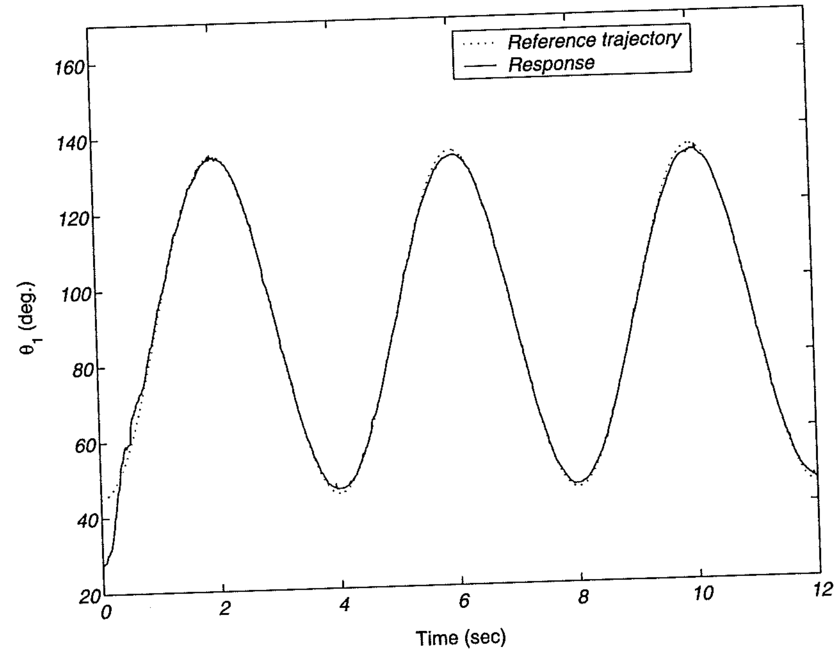


Fig. 7.39: Angular position of the first input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

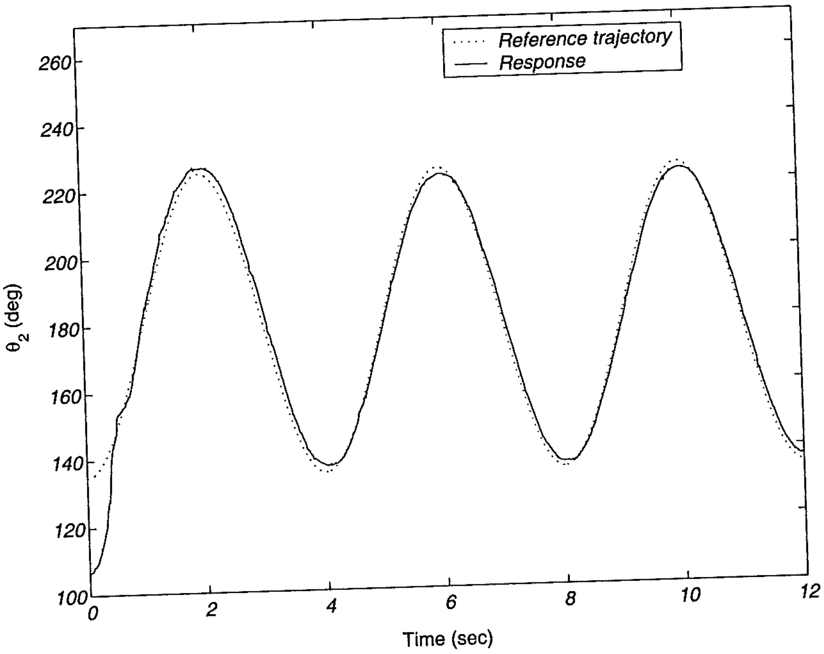


Fig. 7.40: Angular position of the second input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

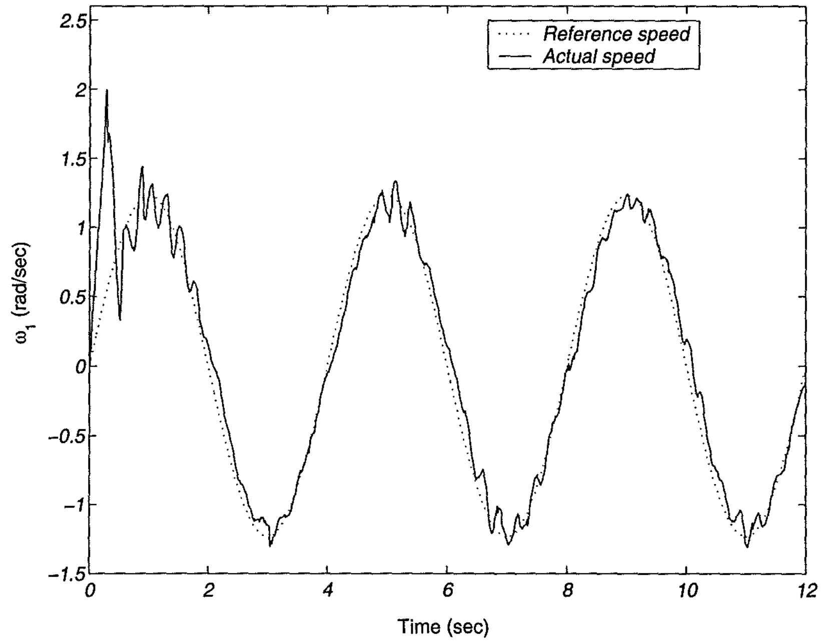


Fig. 7.41: Angular speed of the first input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

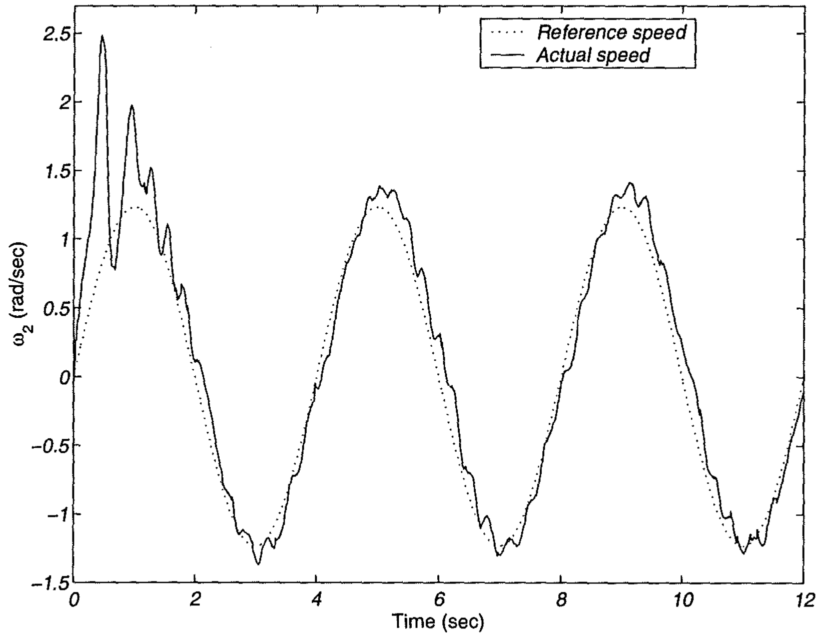


Fig. 7.42: Angular speed of the second input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

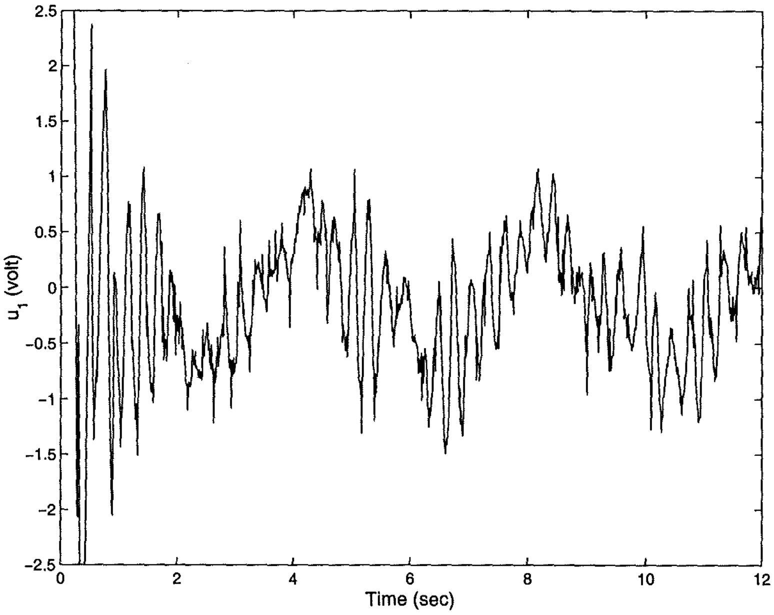


Fig. 7.43: Control signal of the first input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

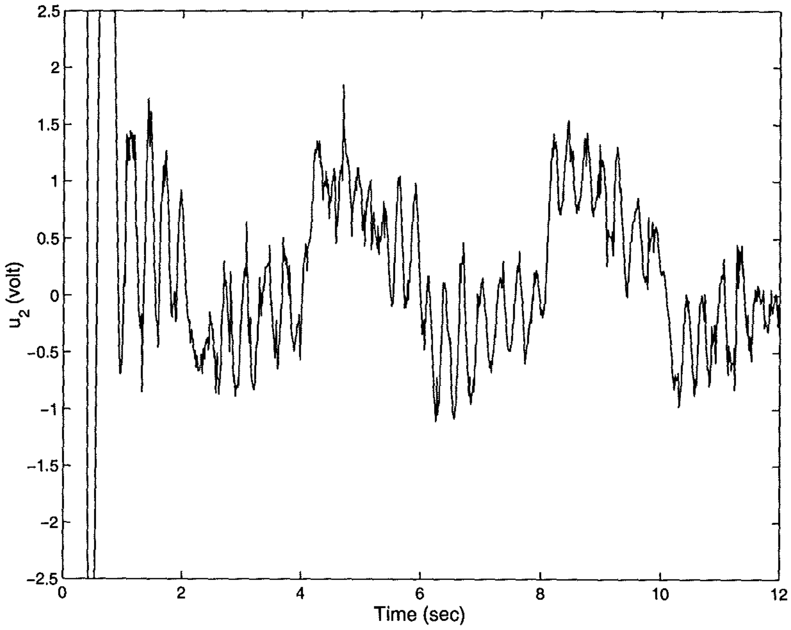


Fig. 7.44: Control signal of the second input link (Computed-Torque and $\omega = \frac{\pi}{2}$).

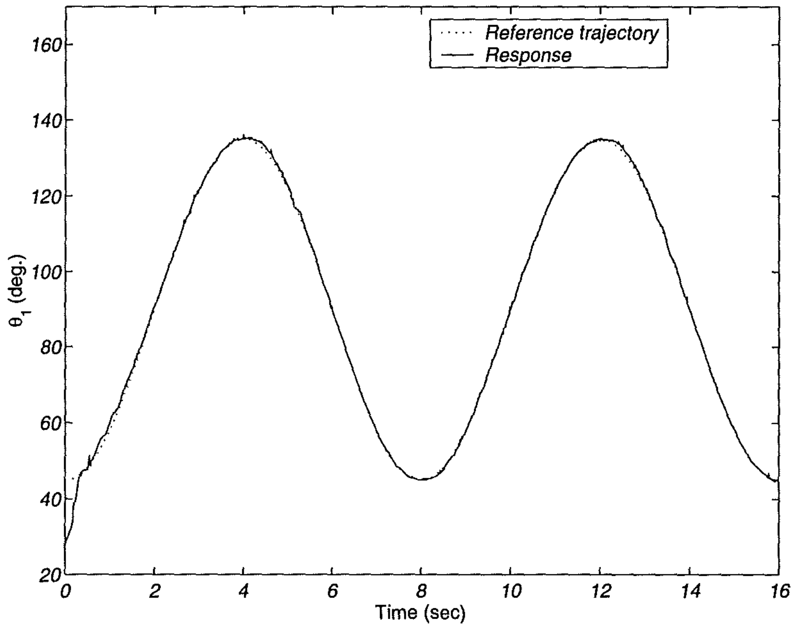


Fig. 7.45: Angular position of the first input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

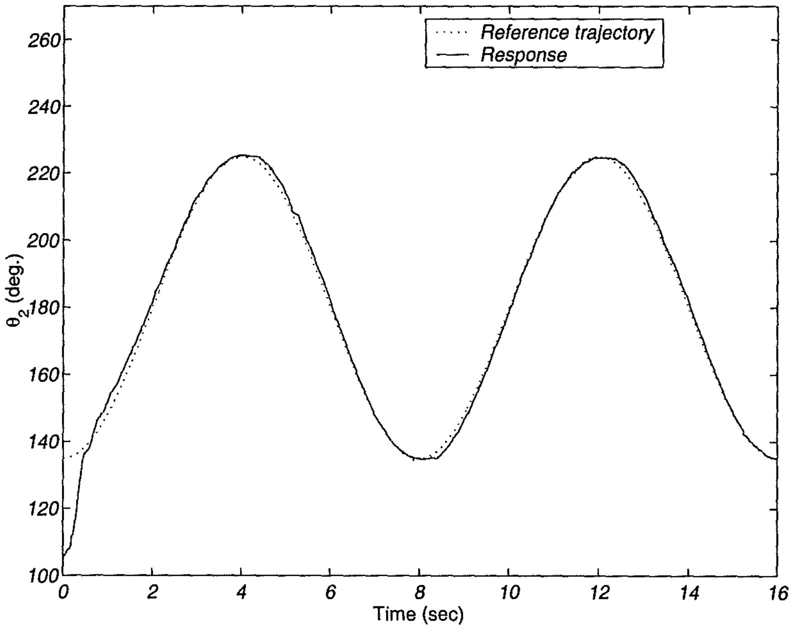


Fig. 7.46: Angular position of the second input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

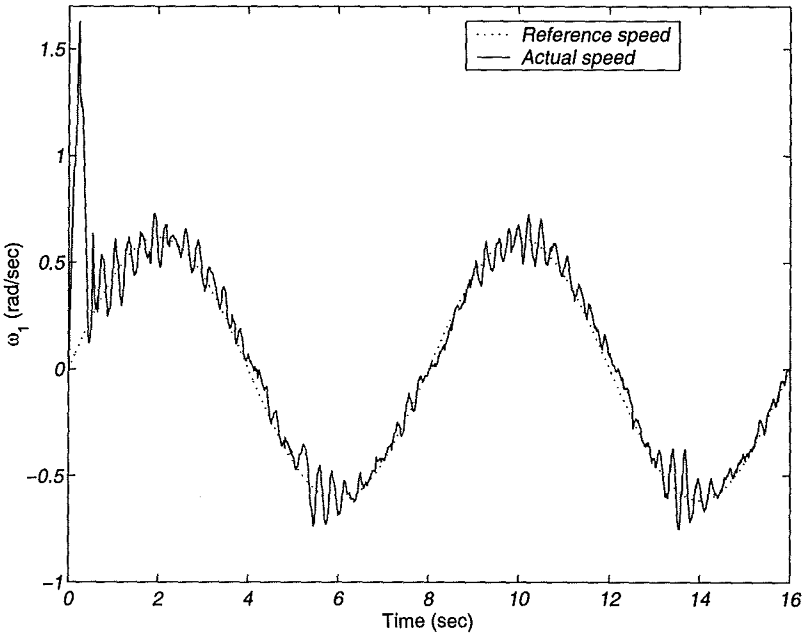


Fig. 7.47: Angular speed of the first input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

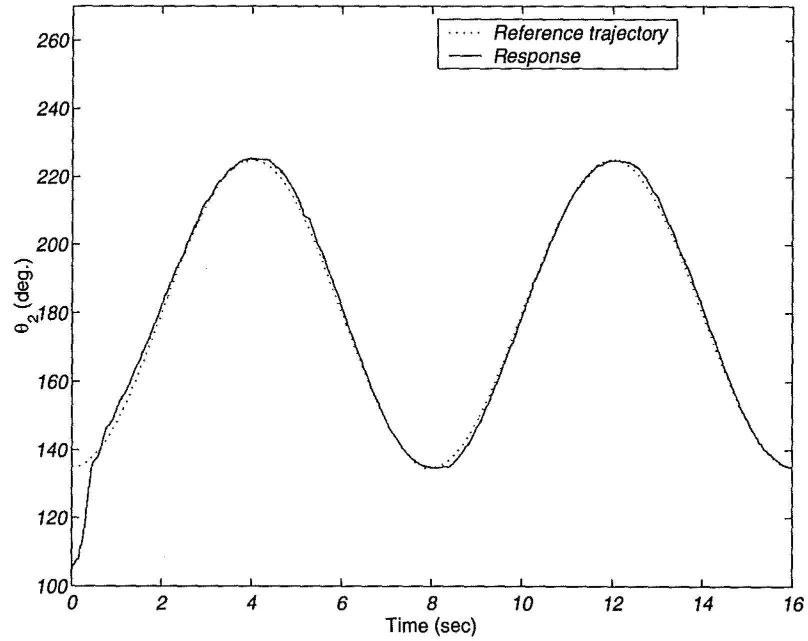


Fig. 7.48: Angular speed of the second input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

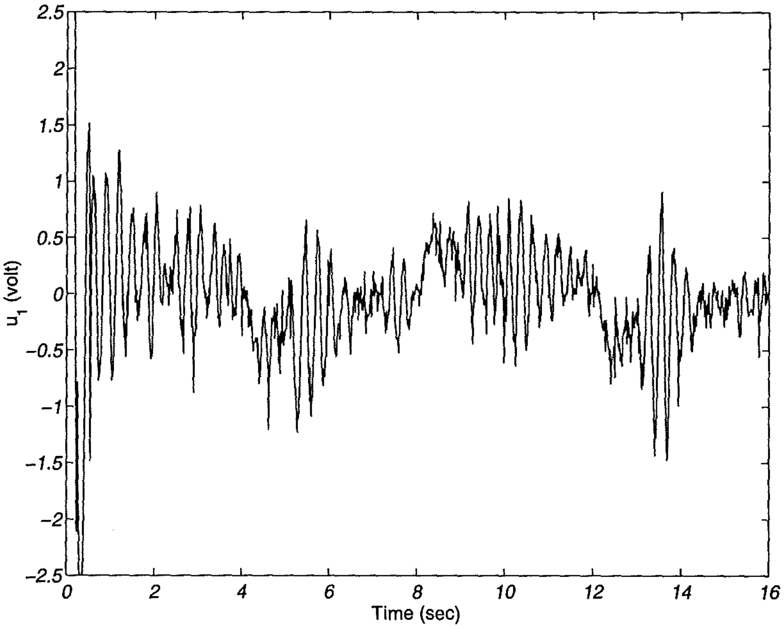


Fig. 7.49: Control signal of the first input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

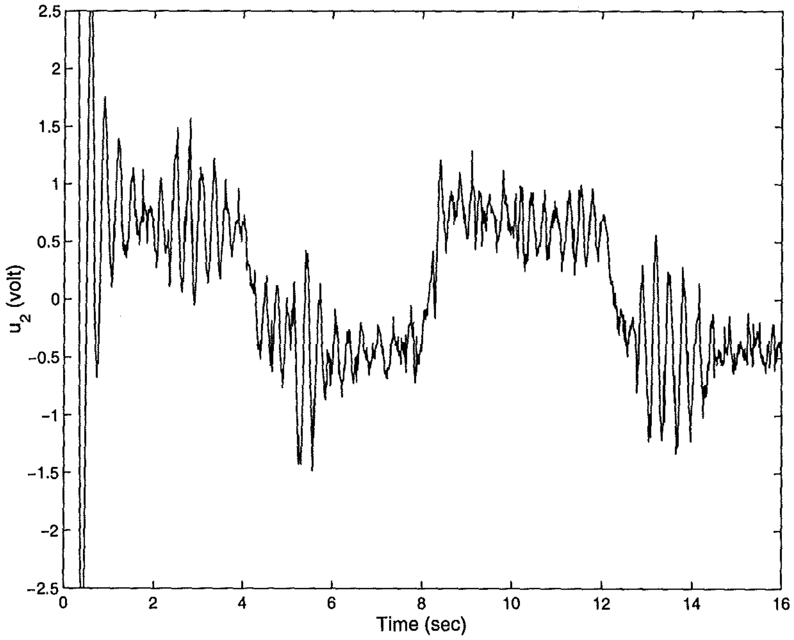


Fig. 7.50: Control signal of the second input link (Computed-Torque and $\omega = \frac{\pi}{4}$).

Chapter 8

On the design of computed-torque controllers for sliding mode behaviour

8.1 Introduction

Nonlinear model-based control systems such as the computed-torque controllers are good techniques for robot control when the parameters of the model are known and the control computer is sufficiently able to compute the inverse dynamics in a reasonable time. This fact is equally applicable to both serial and parallel robots. Using the wealth of identifications techniques designed for serial structures can assist in solving the problem of identifying the unknown parameters of these robots. In fact their structural property is the key in the development of these techniques, where individual links parameters and friction terms of the joints can be identified separately. For parallel robots the problems are complicated. Neither the links parameters and their joint friction terms can be individually identified, nor can the inverse dynamics be calculated in real time. In addition to this the existence of many spherical joints in such spatial mechanisms makes the situation worse when identifying the friction parameters. It is important to note that the only solution to these problems is to use robust control systems. These are normally good techniques in the presence of uncertainties (see for example [80-82]).

Sliding mode control (SMC) is one of the most important methodologies for nonlinear control. This is not only because of its robustness but also due to its invariance (under certain conditions) with respect to system perturbations and external disturbances [83]. This chapter is organised as follows in order to realise the aim of designing a robust computed-torque controller. Firstly the computed-torque method is briefly reviewed. Secondly the sliding mode control system technique is presented. Following on from this a relationship between the SMC system and the computed-torque control method is established. Following this the latest trials of other research efforts to overcome the difficulties associated with the sliding mode control systems are presented, followed by derivations to explain their outcomes. After that the effort is directed towards the design of a new robust controller. Following this simulations are carried out using the SEPA robot in order to validate the control law, and also to examine the behaviour of all robot designs under this controller. For practical validation the controller is implemented on the existing experimental parallel robot.

8.2 Computed-torque method for robot manipulators

The equation of motion of an n-DOF rigid manipulator is given by

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \qquad (8.1)$$

Where,

$M(\Theta)$:	is the inertia matrix with dimensions $n \times n$.
$V(\Theta,\dot{\Theta})$:	is an $n \times 1$ vector of Coriolis and centrifugal forces.
$F(\dot{\Theta})$:	is an $n \times 1$ vector of friction forces acting at the manipulator joints.
$G(\Theta)$:	is an $n \times 1$ vector of gravity forces.
Θ :	is an $n \times 1$ vector of joint positions.
τ :	is an $n \times 1$ vector of joint torques supplied by the actuators.

In order to control the manipulator of equation 8.1 using the computed torque method, the control law, as discussed in chapter 1, is given by the following equation [44],

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + K_v\dot{E} + K_pE\right) + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \qquad (8.2)$$

Where K_v and K_p are $n \times n$ diagonal matrices of positive and constant elements on the diagonal. The vector of the servo errors is defined as ($E = \Theta_d - \Theta$), where Θ_d and Θ represent the vectors of the desired and the actual outputs respectively.

If the model is exact and the elements of K_p and K_v are represented by k_{Pi} and k_{Vi} respectively, the system response can represent either of the following three cases,

I- Critically damped response ($k_{Vi}^2 = 4k_{Pi}$).

II- Over-damped response ($k_{Vi}^2 > 4k_{Pi}$).

III- Under-damped response ($k_{Vi}^2 < 4k_{Pi}$).

It is important to note that the above conditions result from the solution of the error equation of the system as defined by equation 1.9 (chapter 1).

Sometimes an integral term is added to equation 8.2 so that the system has no steady-state error. In this case, equation 8.2 becomes as follows,

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + K_v\dot{E} + K_pE + K_i\int Edt\right) + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \qquad (8.3)$$

Where K_i is a diagonal matrix with dimension $n \times n$ and with positive, and constant, elements on the diagonal. In fact the addition of such an integral term has a side effect because, as discussed in chapter 7, it raises the order of the system. To reduce this effect, the elements of the K_i matrix are always kept small [84].

8.3 Sliding mode control system (SMC) technique

The aim of the sliding mode control system technique is to design a control law that forces the system to behave in a prescribed manner irrespective of the original system dynamics and the uncertainty in its parameters or the external disturbances. The predefined system behaviour is represented by a stable differential equation usually of order less than the order of the original system. This differential equation is defined as follows,

$$\sigma(X) = 0 \qquad (8.4)$$

Where X represents the state vector of the system that is defined as $X^T = [E^T, \dot{E}^T]$.

The set of all X such that $\sigma(X) = 0$ is a $(2n-1)$-dimensional subspace or "hyper-plane" in R^{2n} which is normally called the switching surface [71]. In fact, for nonlinear systems in general the design of such switching surfaces is itself a research area and there are several design procedures that can do the job (see for example [83]). For robotic manipulators three alternatives for switching surfaces have been reported in the literature [85] and they are as follows,

I- $\sigma = E$

II- $\sigma = \dot{E} + \Lambda E$

III- $\sigma = \ddot{E} + F_1 \dot{E} + F_0$

Where Λ, F_1 and F_0 are diagonal matrices of positive elements on the diagonal.

It is important to note that the second alternative is the common switching function (see for example [80-81] [86-88]). The simplest case occurs when using this alternative, and when $n = 1$. In this case the matrix Λ degenerates to a scalar value a and the switching surface corresponds to a line through the origin with a slope of $(-a)$, as seen in Figure 8.1.

The switching surface divides the state space into two parts. When the states of the system satisfy $\sigma(X) < 0$, then we are on one side of the switching surface and the control law has one form. On the other hand when $\sigma(X) > 0$ we are on the other side and the control law has another form. Here the controller changes structure in order to maintain the system on the sliding surface. On the switching surface the system is said to be in the sliding mode. In the sliding mode ($\sigma(X) = 0$) the dynamics of the system simplify substantially. If the switching surface is chosen to be the second alternative, mentioned earlier, then the system dynamics reduce to,

$$\dot{E} + \Lambda E = 0 \quad (8.5)$$

Clearly, the tracking errors are independent of the robot parameters and external disturbances. Because Λ is diagonal the tracking error for the i^{th} DOF is given by the following equation,

$$e_i(t) = e_i(0)\exp(-a_i t) \quad (8.6)$$

Where, $e_i(0)$ represents the initial value of the error of the i^{th} DOF and a_i represent its associated element in the matrix Λ.

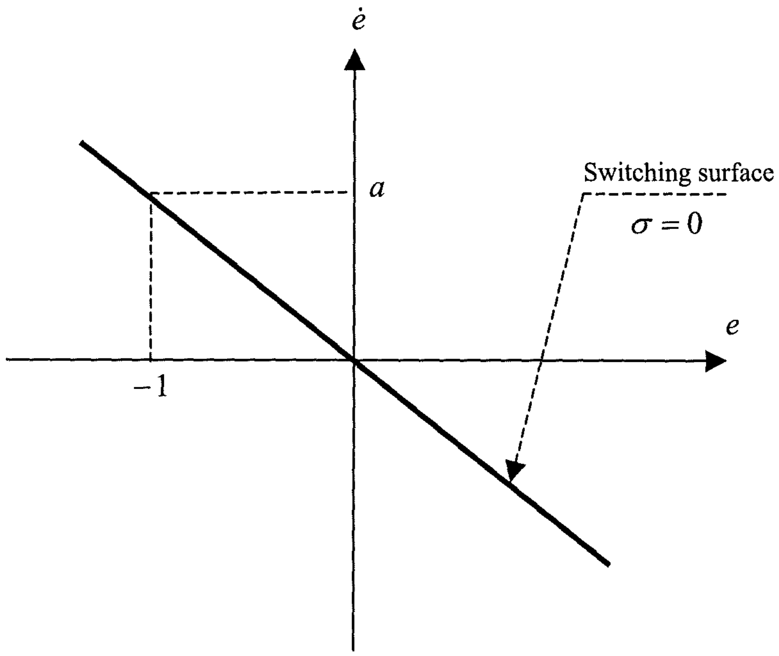


Fig. 8.1: The simplest case of a switching surface.

From equation 8.6, it is clear that the time-rate at which the error is vanishing can be controlled through the choice of the elements of Λ.

After choosing the appropriate switching surface, the next step in the SMC design technique is the formulation of the overall control law which drives the state of the system towards the sliding surface in a finite time and then maintains the state on this surface once it has been reached, despite the perturbations. It is relevant to note that the dynamics of the system in the first stage are called the *reaching mode dynamics*. The conceptual components of the SMC strategy for these two stages are Lyapunov control and bang-bang control, respectively.

By using Lyapunov control the designer should only verify the following condition,

$$\frac{1}{2}\frac{d}{dt}\left\{\sigma^T(X)\sigma(X)\right\}<0 \qquad (8.7)$$

The left hand side of equation 8.7 represents the time derivative of the following Lyapunov function,

$$V_L=\frac{1}{2}\sigma^T(X)\sigma(X) \qquad (8.8)$$

Equation 8.8 is a Lyapunov function in the sense that it is continuously differentiable, $V_L>0$ and $V_L=0$, if and only if, $\sigma(X)=0$. The main problem with the above design procedure lies in the fact that it does not provide any clear idea about the structure of the control law, meaning that the designer must look for the suitable control which satisfies equation 8.7. This is in fact a very complicated task especially for multi-input multi-output nonlinear systems.

8.3.1 The Reaching law method for the design of SMC systems

The reaching law method is a systematic design procedure for the design of SMC systems [83]. In addition to its simplicity the method has two advantages. The first is that by using the reaching law in the design of the control law the system becomes invariant to the perturbations and the external disturbances during the reaching stage. The proof of this issue can be found in reference [83]. The second advantage is that the use of the reaching law gives a clear idea about the nature of the system dynamics during the reaching phase (a missing property when using the Lyapunov criterion).

***Definition* 8.1** [83]: The reaching law is a differential equation which specifies the dynamics of a switching function $\sigma(X)$.

The practical general form of the reaching law is given by the following equation,

$$\dot{\sigma} = -QSgn(\sigma) - KH(\sigma) \qquad (8.9)$$

Where, Q and K are diagonal matrices of dimension $m \times m$, with m representing the number of inputs of the system, and having positive elements on the diagonal. The vectors $Sgn(\sigma)$ and $H(\sigma)$ are given by

$$Sgn(\sigma) = [\,\mathrm{sgn}(\sigma_1),....,\mathrm{sgn}(\sigma_m)\,]^T \qquad (8.10)$$

$$H(\sigma) = [\,h_1(\sigma_1),.......,h_m(\sigma_m)\,]^T \qquad (8.11)$$

Where the conditions $\sigma_i\, h_i(\sigma_i) > 0$ and $h_i(0) = 0$ should be satisfied.

Three practical special cases for the reaching law have been reported [83] and they are as follows,

$$\dot{\sigma}(X) = -QSgn(\sigma) \qquad (8.12)$$

$$\dot{\sigma}(X) = -QSgn(\sigma) - K\sigma \qquad (8.13)$$

$$\dot{\sigma}_i = -q_i|\sigma_i|^{\alpha} Sgn(\sigma_i) \qquad (8.14)$$

Where, $0 < \alpha < 1$ and $i = 1$ to m.

From the choice of the parameters in the differential equation for the reaching law the rate at which the system moves towards the sliding surface can be controlled. In order to understand this issue let us multiply both sides of the general form of the reaching law by $\sigma^T(X)$ as follows,

$$\sigma^T \dot{\sigma} = -\sigma^T Q\, Sgn(\sigma) - \sigma^T KH(\sigma) \qquad (8.15)$$

The left hand side of equation 8.15 is the time-rate of the Lyapunov function defined by equation 8.8 and the right hand side satisfies the reaching condition defined by equation 8.7. It is now obvious that the reaching speed can be adjusted by proper choice of Q, K and $H(\sigma)$.

The formulation of the control law by using the reaching law method requires the robot dynamics to be written as follows,

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) + T_d \qquad (8.16)$$

Where T_d represents the vector of all the perturbations and the unknown disturbances.

In order to explain how the control law can be formulated by the reaching law method the reaching law of equation 8.12 is chosen as an example. In addition to this the following sliding surface is chosen

$$\sigma = \Lambda E + \dot{E} \qquad (8.17)$$

Taking the time derivative of the above equation gives

$$\dot{\sigma} = \Lambda \dot{E} + \ddot{E} \qquad (8.18)$$

Rewriting the system equation of motion (equation 8.16) in the error-space yields,

$$\ddot{E} = \ddot{\Theta}_d - M^{-1}(\Theta)\left(\tau - V(\Theta,\dot{\Theta}) - G(\Theta) - F(\dot{\Theta}) - T_d\right) \qquad (8.19)$$

Substituting from equation 8.19 into equation 8.18 gives,

$$\dot{\sigma} = \Lambda\dot{E} + \ddot{\Theta}_d - M^{-1}(\Theta)\left(\tau - V\left(\Theta,\dot{\Theta}\right) - G(\Theta) - F\left(\dot{\Theta}\right) - T_d\right) \qquad (8.20)$$

Now, equating equations 8.20 and 8.12 and solving for the control law τ yields,

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + \Lambda\dot{E} + QSgn(\sigma)\right) + V\left(\Theta,\dot{\Theta}\right) + G(\Theta) + F\left(\dot{\Theta}\right) + T_d \qquad (8.21)$$

The existence of the unknown term T_d in equation 8.21 represents the only difficulty when deriving the SMC law by the reaching law method. It is important to note that if this problem is not treated properly, the reaching time may be increased (i.e. the reaching speed may be decreased). The following derivation explains this issue.

Suppose that the unknown term T_d is assigned a known quantity N. By doing this the control law of equation 8.21 becomes as follows,

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + \Lambda\dot{E} + QSgn(\sigma)\right) + V\left(\Theta,\dot{\Theta}\right) + G(\Theta) + F\left(\dot{\Theta}\right) + N \qquad (8.22)$$

Now, substituting from equation 8.22 into equation 8.20 which describes the time derivative of the switching function, yields

$$\begin{aligned}\dot{\sigma} = \Lambda\dot{E} + \ddot{\Theta}_d - M^{-1}(\Theta)\Big\{&M(\Theta)\left(\ddot{\Theta}_d + \Lambda\dot{E} + QSgn(\sigma)\right)\\ &+ V\left(\Theta,\dot{\Theta}\right) + G(\Theta) + F\left(\dot{\Theta}\right) + N - V\left(\Theta,\dot{\Theta}\right) - G(\Theta) - F\left(\dot{\Theta}\right) - T_d\Big\}\end{aligned}$$

$$= -QSgn(\sigma) + M^{-1}(\Theta)\left(-N + T_d\right) \qquad (8.23)$$

Equation 8.23 shows that the actual reaching law, and of course the associated reaching speed, is strongly dependent on the quantity $\left(-N + T_d\right)$. If the value of N does not match the unknown disturbances T_d, clearly the reaching speed will be affected.

However, in order to solve the problem Gao and Hung [83] assume a priori knowledge of the upper and the lower bounds of T_d, and according to this the value of N was chosen. The author's manipulations have led to

$$N = M(\Theta)\left\{\left(\frac{F_u - F_l}{2}\right) Sgn(\sigma) - \left(\frac{F_u + F_l}{2}\right)\right\} \qquad (8.24)$$

Where the vectors F_u and F_l represent the upper and the lower values of the vector T_d, respectively.

Equation 8.24 in fact is equivalent to the following logical function

$$N = \begin{Bmatrix} -M(\Theta)F_l & \text{if} \ \ \sigma > 0 \\ -M(\sigma)F_u & \text{if} \ \ \sigma < 0 \end{Bmatrix} \qquad (8.25)$$

Now, substituting from equation 8.24 into equation 8.22 yields,

$$\tau = M(\Theta)\left\{\ddot{\Theta}_d + \Lambda\dot{E} + QSgn(\sigma) + \left(\frac{F_u - F_l}{2}\right) Sgn(\sigma) - \left(\frac{F_u + F_l}{2}\right)\right\}$$

$$+ V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \qquad (8.26)$$

It is important to note that equation 8.26 is practical only if $F_u > 0$ and $F_l < 0$. This because of the fact that if $F_u > 0$, $F_l > 0$ and $F_u = F_l = \overline{F}$ then the above control law will contain a constant term defined by $(-M(\Theta)\overline{F})$. This is unacceptable because the control law in this case does not depend on the system state. Despite this problem Gao and Hung [83] did not refer to this issue explicitly in their work. They used $F_u > 0$ and $F_l < 0$, together with $|F_u| = |F_l| = \text{f}$, in their example. If this is the case then the value of N reduces to

$$N = M(\Theta)\text{f}\ Sgn(\sigma) \qquad (8.27)$$

and the control law takes the following form

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + \Lambda\dot{E} + \tilde{Q}Sgn(\sigma)\right) + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \quad (8.28)$$

Where,

$$\tilde{Q} = Q + \text{f} \quad (8.29)$$

It is important to mention that the robust term in equation 8.28, that is $\tilde{Q}Sgn(\sigma)$, is entirely normal in any SMC law, meaning that equation 8.28 is the only practical form of the control law by the reaching law method. In fact the term $\tilde{Q}Sgn(\sigma)$ represents the actual reaching speed of the system whose original speed is defined by $\tilde{Q}Sgn(\sigma)$, (equation 8.6). In other words, the actual reaching law may be written as

$$\dot{\sigma}_{Actual} = -\tilde{Q}Sgn(\sigma) \quad (8.30)$$

8.4 Derivation of the relationship between the computed-torque and the sliding mode control systems

***Proposition* 8.1**: The computed-torque controller, either with or without an integral control action, is a sliding mode controller but with an invalid reaching law.

Proof.

By comparing the SMC law defined by equation 8.28 with the computed-torque law of equation 8.2 it may be considered that

$$K_V \equiv \Lambda \ \& \ K_p E \equiv QSgn(\sigma)$$

From these it can be concluded that the computed-torque controller is actually a sliding mode controller with the following features

$$\sigma = \dot{E} + K_v E \quad \& \quad \dot{\sigma} = \text{-}K_p E$$

The reaching law here is invalid because it does not satisfies the reaching law definition.

The above result is also true for the case of the computed-torque controller which has integral control action (equation 8.3), but in this case the invalid reaching law is given by

$$\dot{\sigma} = \text{-}K_p E - K_i \int E \, dt$$

8.5 General difficulties associated with the SMC systems

Chattering is the basic obstacle preventing sliding mode control (SMC) from gaining wider use in practical applications. This phenomenon is generally undesirable since it involves extremely high control activity, and it may also excite any neglected high-frequency dynamics. For the sake of clarity chattering results from the robustness term in the control law that contains the discontinuous signum function. The main assumption in the design of SMC is that the control has to be switched infinitely fast from one value to another. This is practically difficult to achieve due to the time delays accompanied with digital implementations and the limited bandwidth of actuators. Generally, it originates from any nonideal switching at the sliding surface. Slotine and Sastry [89] introduced one common approach for chattering reduction. The approach involves introducing a boundary layer around the switching surface as shown in Figure 8.2, and then using a continuous control within this layer. By using this method the signum function in the control law is replaced by the following saturation function

$$Sat(\sigma,\phi) = \left\{ \begin{array}{ll} \dfrac{\sigma}{\phi} & \text{if } |\sigma| < \phi \\ Sgn(\sigma) & \text{otherwise} \end{array} \right\} \tag{8.31}$$

Where ϕ represents the boundary layer thickness.

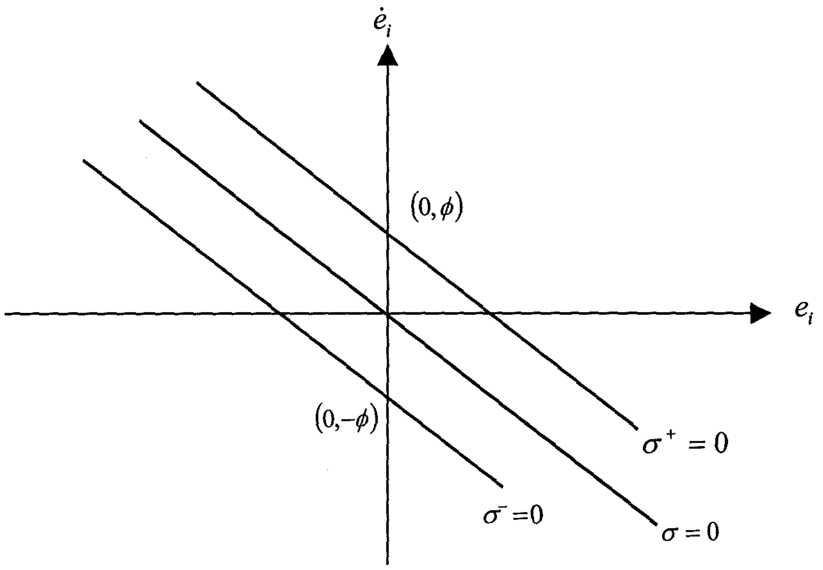


Fig. 8.2: Schematic diagram of the boundary layer in the case of simple sliding surface.

It is immediately obvious from Figure 8.2 that there are two switching surfaces σ^+, σ^- instead of the original surface $\sigma = 0$. In addition, equation 8.31 assumes that the continuous control is represented by linear interpolation between the control-efforts defined on the two sliding surfaces σ^+, σ^-. Despite this, Slotine and Sastry [89] claim that any continuous interpolation between these two extreme control values is sufficient. Although the methodology does reject the high frequency chattering, the tracking accuracy is degraded by an amount linearly proportional to the size of the boundary layer.

The mechanism by which the high frequency chattering is rejected is not explained in the forgoing discussion. To clarify the process one assumes that the vector N does not match T_d in the reaching law of equation 8.23. Then, replacing the signum function by the saturation function in this law leads to,

$$\dot{\sigma} = -\left(\frac{Q}{\phi}\right)\sigma + M^{-1}(\Theta)\left(-N + T_d\right) \quad (8.32)$$

Equation 8.32 shows that the variable σ, which is a measure of the distance to the sliding surface, is the output of a stable first-order filter whose input (to first order) is the quantity defined by $M^{-1}(\Theta)(-N+T_d)$. Consequently chattering is eliminated, as long as the unmodelled high-frequency dynamics are not excited [89]. The additional filtration of perturbations through equation 8.17 results in the degradation of the tracking error. It is important to note that the existence of the two filters led Slotine [86] to make a claim for the balancing of the their break-frequencies to fix the best attainable tracking precision. To take the advantages of the system bandwidth, the boundary layer was made to vary with time in the work by Slotine and Coetsee [90]. For the sake of brevity nearly all of the work in the area of chattering reduction use the same approach, and the main efforts are directed towards introducing different continuous control laws inside the boundary layer to reduce the degradation in the tracking error (see for example [87-88] and [91-92].

Goa and Hung [83] mentioned that using the power rate reaching law defined by equation 8.14 could reduce chattering. In fact they stated that the absence of the term $QSgn(\sigma)$ from this reaching law is the reason for chattering elimination. Obviously this is not entirely logical because chattering results from the existence of the signum function, and the presence of that function in the power rate law confirms that comment. In fact the power reaching law reduces the amplitude of chattering because the gain associated with the signum function has its lowest value (zero) on the sliding surface. Hence it can be concluded that using nonlinear gains in the reaching law can reduce chattering.

Before closing the discussion of this section one thing needs clarifying, that being the outcome of the aforementioned efforts in removing chattering. The control law of the system when it is inside the boundary layer clearly explains this outcome. Replacing the signum function in equation 8.28 by the saturation function leads to

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + \Lambda\dot{E} + \frac{\tilde{Q}}{\phi}\sigma\right) + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \quad (8.33)$$

Substituting from equation 8.17 into the above control law gives

$$\tau = M(\Theta)\left(\ddot{\Theta}_d + \left(\Lambda + \frac{\tilde{Q}}{\phi}\right)\dot{E} + \frac{\tilde{Q}\Lambda}{\phi}E\right) + V(\Theta,\dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \qquad (8.34)$$

The control law of equation 8.34 has exactly the same structure as the computed-torque control law defined by equation 8.2. Hence it can be concluded that the methodology of the boundary layer tries (without intention) to implement the computed-torque control when the system is inside the boundary layer. On the other hand the outcome of recent efforts to reduce the degradation of the tracking error inside the boundary layer by the use of perturbation estimation [87] has led to a computed-torque law with integral control action. In order to justify this issue let us first look at the derived control law given in that reference. The following relation defines the law

$$\tau = M(l)\left(\ddot{l}_d + \lambda\dot{E} + K_r\frac{\sigma}{\phi} + K_p\sigma + K_pH\int\sigma\, dt\right) + C_c\,\dot{l} + G(l) \qquad (8.35)$$

Where $\ddot{l}_d$ represents the vector of the required acceleration, $M(l)$ is the mass matrix and λ, K_r, K_p, H are positive definite matrices. The vectors of gravity and friction forces are the $G(l)$and $C_c\dot{l}$ terms, respectively. The vector of linear displacement (the actuators are linear) and the boundary layer thickness are represented by l andϕ, respectively. It is important to note that in reference [87] the contribution of the Coriolis and centrifugal forces was neglected. In addition to this the sliding surface was given by

$$\sigma = \lambda E + \dot{E} \qquad (8.36)$$

Now, substituting from equation 8.36 into 8.35 yields,

$$\tau = M(l)\left\{\ddot{l}_d + \lambda\dot{E} + K_r\frac{(\dot{E} + \lambda E)}{\phi} + K_p(\dot{E} + \lambda E) + K_pH\int(\dot{E} + \lambda E)dt\right\} + C_c\dot{l} + G(l)$$

$$= M(l)\left\{\ddot{l}_d + \left(\frac{K_r}{\phi} + K_P\right)\dot{E} + \left(K_P H + \left(\frac{K_r}{\phi} + K_P\right)\lambda\right)E + K_P H\lambda \int E\, dt\right\} + C_c\dot{l} + G(l)$$

(8.37)

Therefore from equation 8.37 it is clear that the outcome is a computed-torque controller with an integral term.

8.6 The design of a new robust, computed-torque, controller

In section 8.4 it has been shown that the computed-torque controller is a sliding mode controller, but with an invalid reaching law. Hence in order to enhance the behaviour of the computed-torque controller the reaching law must be repaired. In order to achieve this aim the following reaching law is introduced,

$$\dot{\sigma}_i = -k_i\sigma_i - q_i|\sigma_i|^{\alpha} Sgn(\sigma_i) \quad (8.38)$$

Where, $0 < \alpha < 1$.

By using the above reaching law, together with the sliding surface of equation 8.17, the new control law (by using the reaching law method) is given by

$$\tau = M(\Theta)\left\{\ddot{\Theta}_d + (K + \Lambda)\dot{E} + (K\Lambda)E + \Psi(\sigma)\right\} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) + N \quad (8.39)$$

Where,

$$\Psi^T(\sigma) = \left\lfloor q_1|\sigma_1|^{\alpha_1}\operatorname{sgn}(\sigma_1), q_2|\sigma_2|^{\alpha_2}\operatorname{sgn}(\sigma_2), \ldots\ldots, q_n|\sigma_n|^{\alpha_n}\operatorname{sgn}(\sigma_n)\right\rfloor$$

Because N is no more than a discontinuous term it is ignored in the control law on the assumption that the term $\Psi(\sigma)$ includes it. The control law becomes as follows after doing this,

$$\tau = M(\Theta)\left\{\ddot{\Theta}_d + (K + \Lambda)\dot{E} + (K\Lambda)E + \Psi(\sigma)\right\} + V(\Theta, \dot{\Theta}) + G(\Theta) + F(\dot{\Theta}) \quad (8.40)$$

The quality of the new control law is assessed in the following two sections through simulations and practical implementation. First, simulations have been carried out using the SEPA robot to ensure the effectiveness of the control law to all robot designs. Following this the experimental parallel robot will be subjected to the control law to identify any practical issue which may arise.

8.7 Simulations

Assuming that the SEPA manipulator of chapter 5 is moving in the horizontal plane and using the values $m = 1$ kg, $2L = 1\,\text{m}$ and $m_m = 0.5\,\text{kg}$ where m_m is the mass of the actuator at joint C, the mode dynamics (without friction at the joints) are as follows,

- ***Normal Parallel Mode (without*** m_m)

$$\begin{bmatrix}\tau_A \\ \tau_B\end{bmatrix} = \begin{bmatrix} a_{11} & a_{12}\cos(\theta_2-\theta_1) \\ a_{21}\cos(\theta_2-\theta_1) & a_{22}\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix} -b_1\sin(\theta_2-\theta_1)\dot{\theta}_2^2 \\ b_2\sin(\theta_2-\theta_1)\dot{\theta}_1^2\end{bmatrix} \tag{8.41}$$

Where,

$$a_{11} = a_{22} = 1.66 \text{ and } a_{12} = a_{21} = b_1 = b_2 = 0.996.$$

- ***Redundant Parallel Mode***

$$\begin{bmatrix}\tau_A + \tau_C \\ \tau_B - \tau_C\end{bmatrix} = \begin{bmatrix} a_{11} & a_{12}\cos(\theta_2-\theta_1) \\ a_{21}\cos(\theta_2-\theta_1) & a_{22}\end{bmatrix}\begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix} -b_1\sin(\theta_2-\theta_1)\dot{\theta}_2^2 \\ b_2\sin(\theta_2-\theta_1)\dot{\theta}_1^2\end{bmatrix} \tag{8.42}$$

Where,

$$a_{11} = 1.66, a_{22} = 2.16 \text{ and } a_{12} = a_{21} = b_1 = b_2 = 0.996$$

- ***Serial Mode***

$$\begin{bmatrix} \tau_A + \tau_B - \tau_C \\ \tau_C \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12}\cos(\theta_3 - \theta_1) \\ a_{21}\cos(\theta_3 - \theta_1) & a_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} -b_1\sin(\theta_3 - \theta_1)\dot{\theta}_3^2 \\ b_2\sin(\theta_3 - \theta_1)\dot{\theta}_1^2 \end{bmatrix} \quad (8.43)$$

Where,

$$a_{11} = 3.156, a_{22} = 1.164 \text{ and } a_{12} = a_{21} = b_1 = b_2 = 0.996$$

8.7.1 Simulations (Case I)

In order to examine the robustness against uncertainties in the robot parameters, the controller will be designed using the following values

$$a_{11} = 2.1587, a_{22} = 1.82 \text{ and } a_{12} = a_{21} = b_1 = b_2 = 0.7$$

The first and the second value represent the averages of those of the three modes. The last value is chosen randomly. This is performed in order to simulate the system when its parameters are not known exactly. Using the above values the errors in the estimate of the parameters in the various modes are shown in Table 8.1. In addition to this the following gains are used during the simulations.

$$K = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix}, \Lambda = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad \& \quad q_1 = q_2 = 3 \quad \text{and} \quad \alpha = .7$$

The links are commanded to track the following trajectories with $\omega = \frac{\pi}{4}$ Hz,

$$\theta_1(t) = \tfrac{\pi}{2} - \tfrac{\pi}{4}\cos(\omega t) \quad (rad) \quad \& \quad \theta_2(t) = \theta_3(t) = \pi - \tfrac{\pi}{4}\cos(\omega t) \quad (rad)$$

The assumed initial values for the links are: $\theta_1(0) = 30^\circ$ and $\theta_2(0) = \theta_3(0) = 99^\circ$. The simulations are performed using the *SIMULINK* block diagram shown in Figure 8.3. The maximum simulation time step is 0.01 second. The tracking errors of the various links in the manipulator modes are shown in Figures 8.4-8.6 and their control signals are shown in Figure 8.7-8.9. The phase planes are shown in Figures 8.10-8.12.

8.7.2 Simulations (Case II)

To further show the robustness of the above control law the previous simulations have been repeated, but with the assumption that no information about the term $V(\Theta,\dot{\Theta})$ is available. In this case, the control law of equation 8.40 becomes as follows,

$$\tau = M(\Theta)\left\{\ddot{\Theta}_d + (K+\Lambda)\dot{E} + (K\Lambda)E + \Psi(\sigma)\right\} \qquad (8.44)$$

The tracking errors of the links in the various modes are shown in Figures 8.13-8.15. The motors control signals and the phase planes are shown in Figures 8.16-8.18 and Figures 8.19-8.21, respectively.

8.7.3 Simulations (Case III)

This part is intended to examine the effect of the speed of the input trajectory on the tracking errors. In order to do this the first simulations are repeated, assuming no initial errors. The initial conditions are $\theta_1(0)=45^\circ$ and $\theta_2(0)=\theta_3(0)=135^\circ$. Two different values for the speed are used ($\omega=\frac{\pi}{8}$ Hz and $\omega=\frac{\pi}{2}$ Hz). The first is called the slow speed and the other is called the fast speed. The tracking error results in the various mechanical modes are shown in Figures 8.22-8.27.

Parameter	Serial Mode	Normal Parallel Mode	Redundant Parallel Mode
a_{11}	- 31.0 %	+ 30 %	+ 30 %
a_{12}	+ 22.7 %	+ 22.7 %	+ 22.7 %
a_{21}	+ 22.7 %	+22.7%	+ 22.7 %
a_{22}	+ 56.4 %	+ 9.6 %	- 15 %
b_1	+ 22.7 %	+ 22.7 %	+ 22.7 %
b_2	+ 22.7 %	+ 22.7 %	+ 22.7 %

Table 8.1. Approximate values of the error in the estimate of the robot parameters in the various modes.

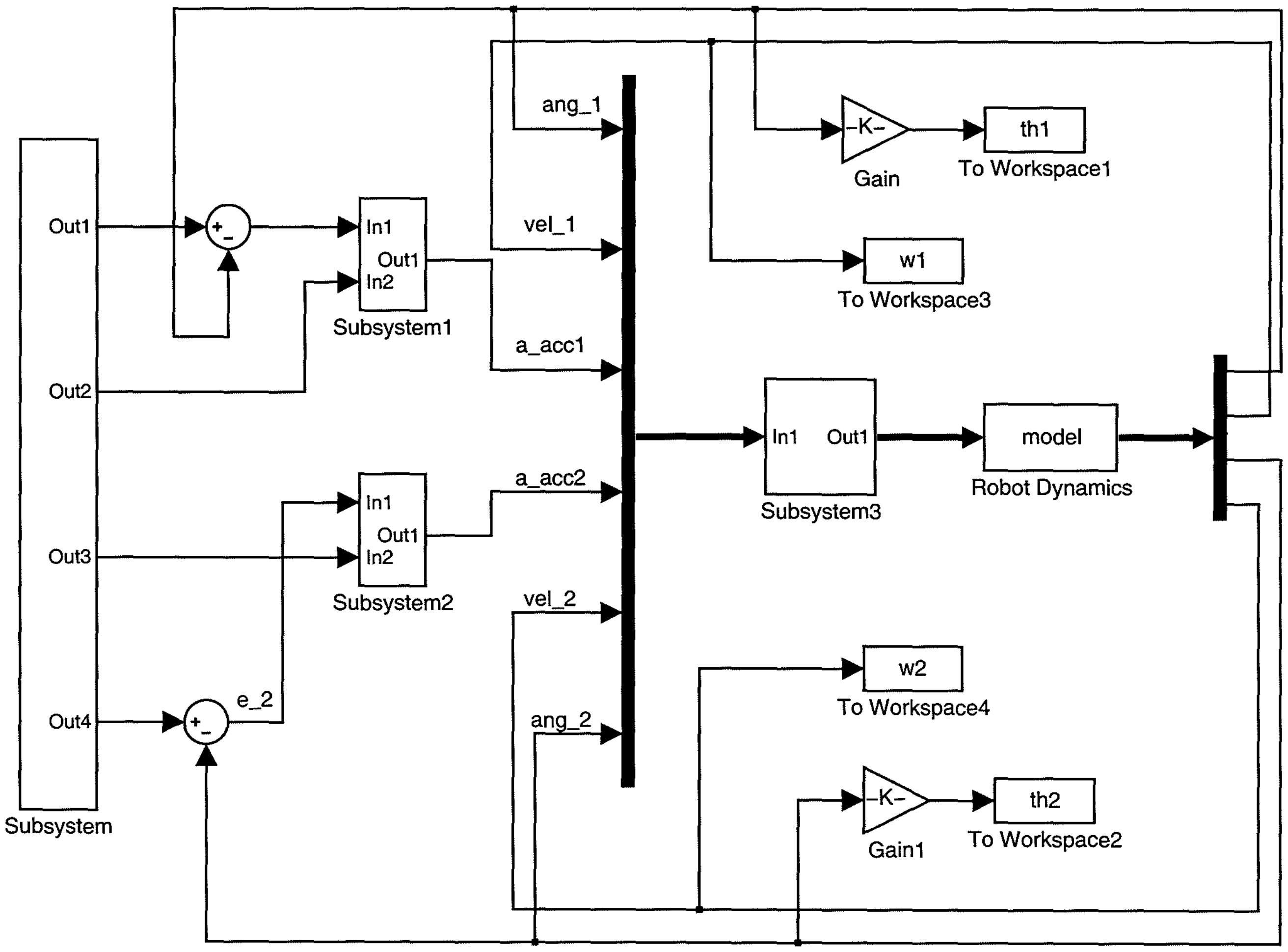


Fig. 8.3: Simulink block diagram for simulation purposes.

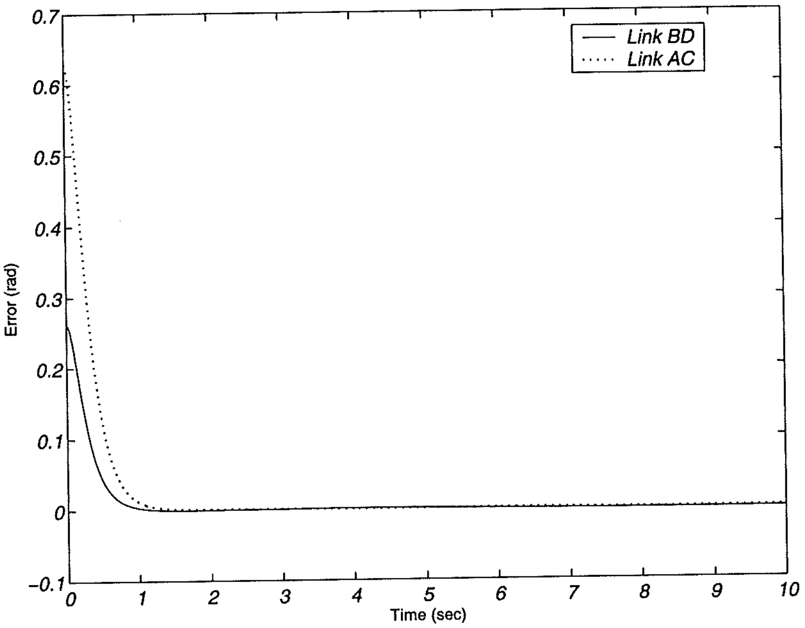


Fig. 8.4: The tracking errors in the case of the *Normal Parallel Mode (Case I)*.

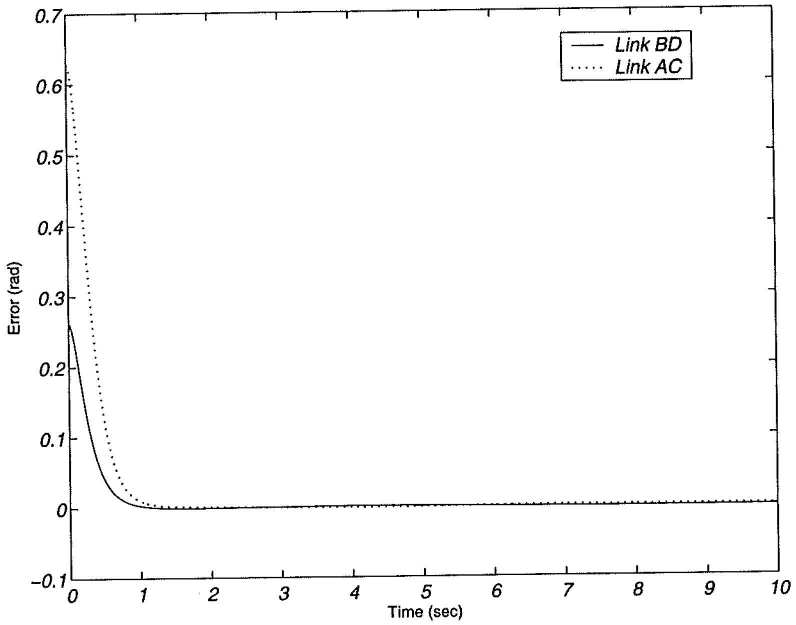


Fig. 8.5: The tracking errors in the case of the *Redundant Parallel Mode (Case I)*.

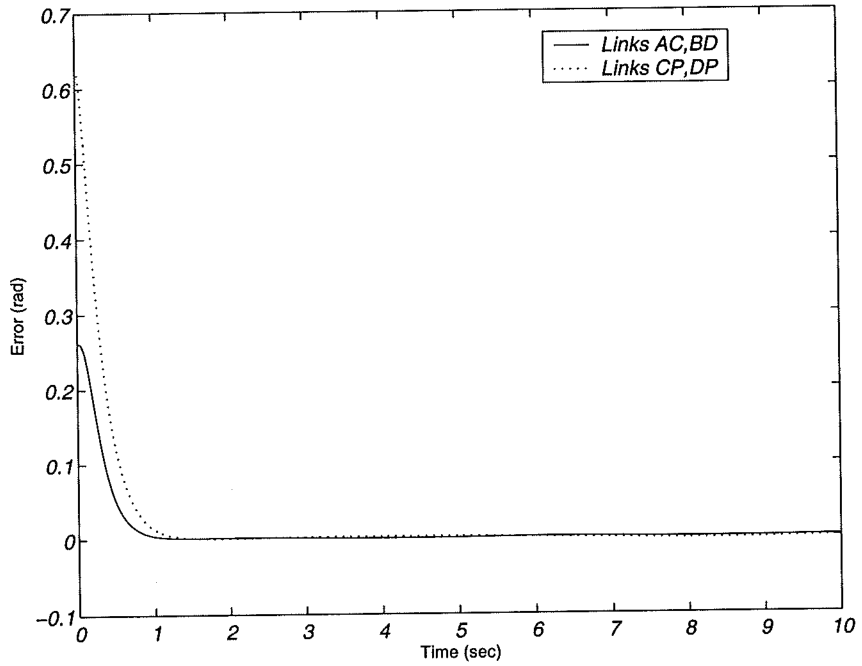


Fig. 8.6: The tracking errors in the case of the *Serial Mode (Case I)*.

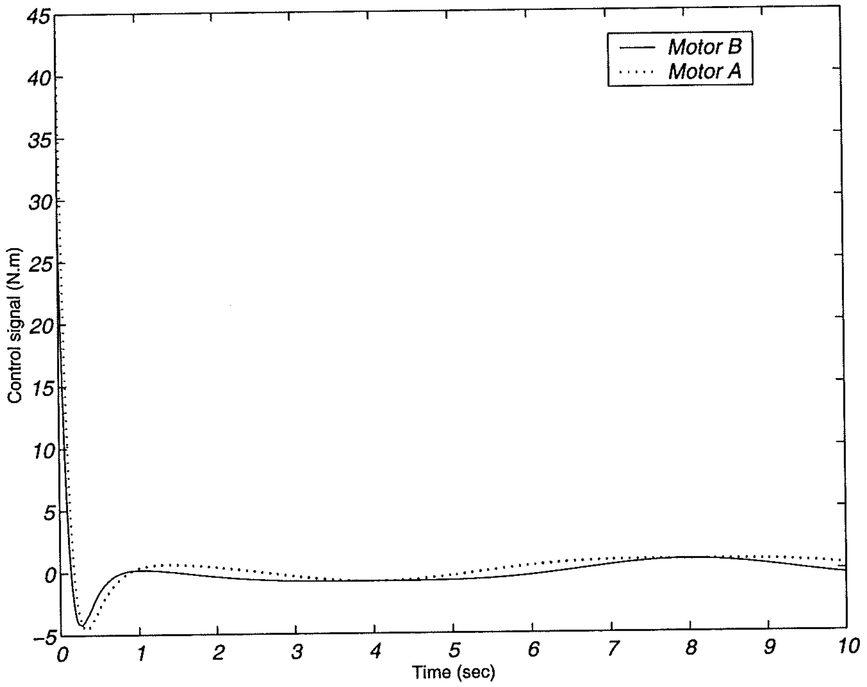


Fig. 8.7: Control torques in the case of the *Normal Parallel Mode (Case I)*.

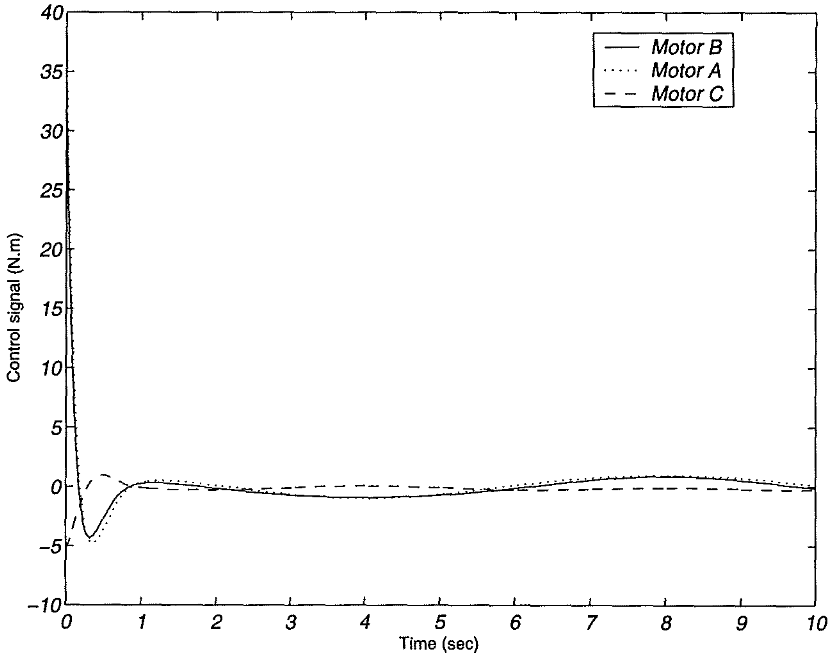


Fig. 8.8: Control torques in the case of the *Redundant Parallel Mode (Case I)*.

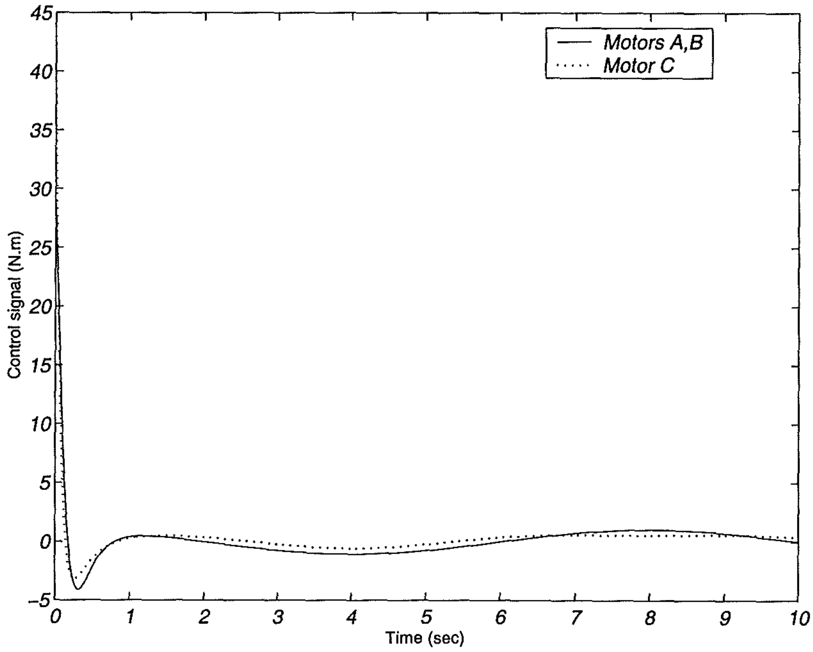


Fig. 8.9: Control torques in the case of the *Serial Mode (Case I)*.

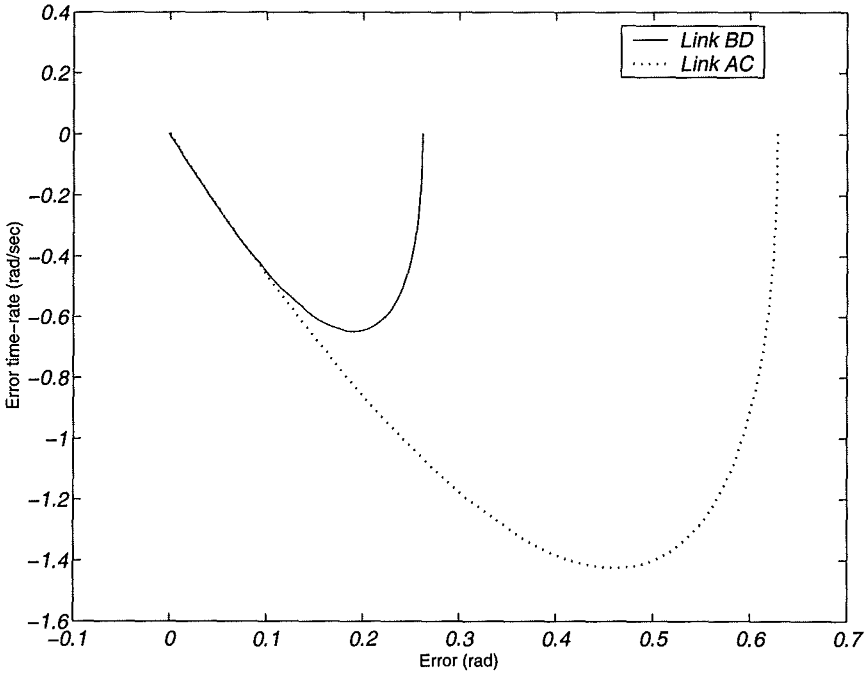


Fig. 8.10: Phase plane in the case of the *Normal Parallel Mode (Case I)*.

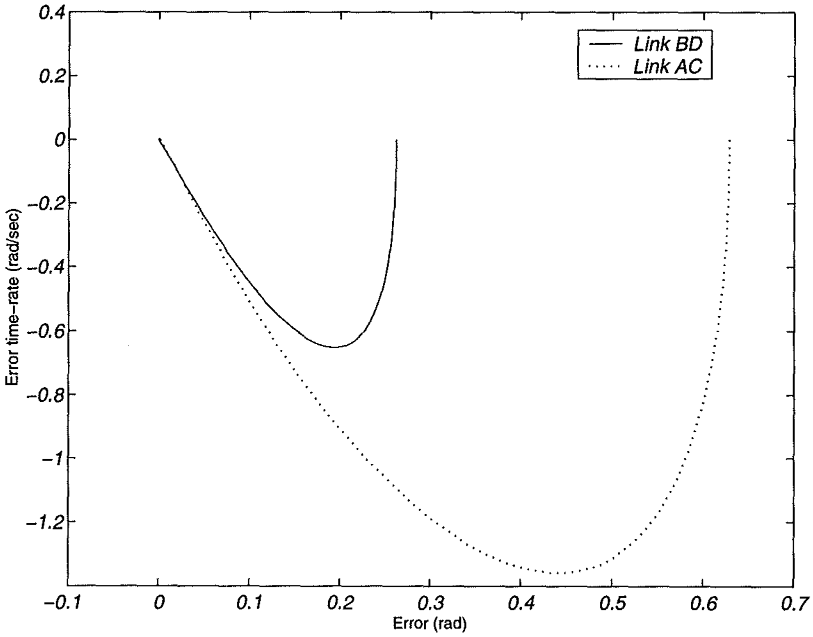


Fig. 8.11: Phase plane in the case of the *Redundant Parallel Mode (Case I)*.

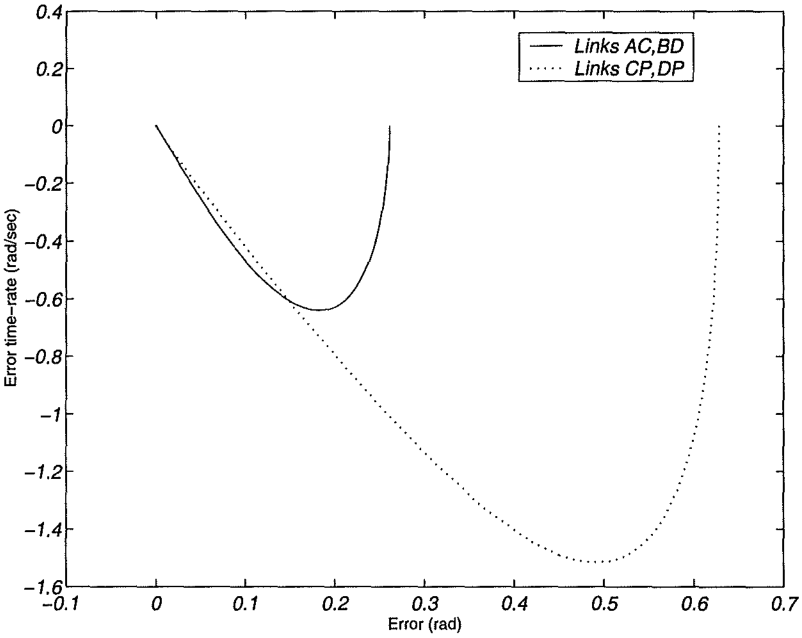


Fig. 8.12: Phase plane in the case of the *Serial Mode (Case I)*.

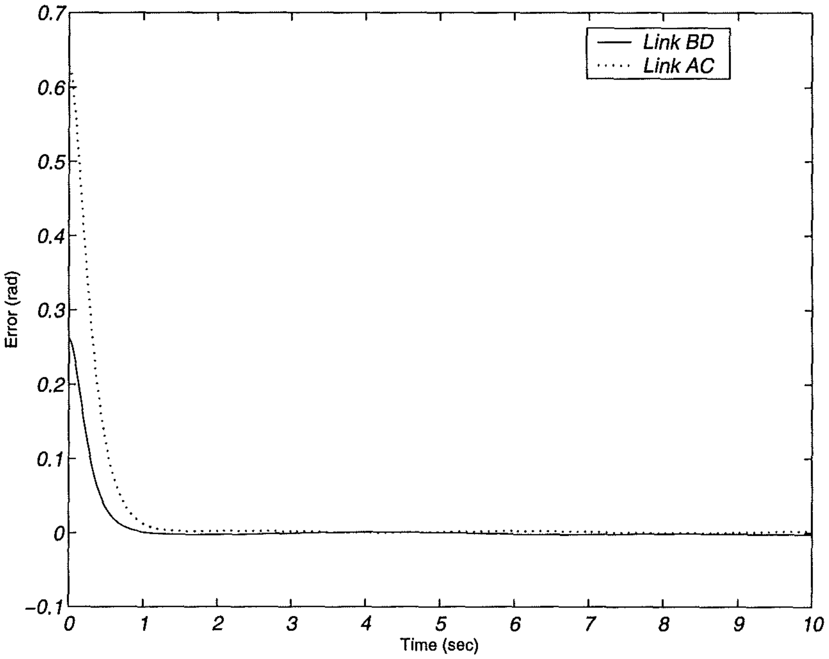


Fig. 8.13: The tracking errors in the case of the *Normal Parallel Mode (Case II)*.

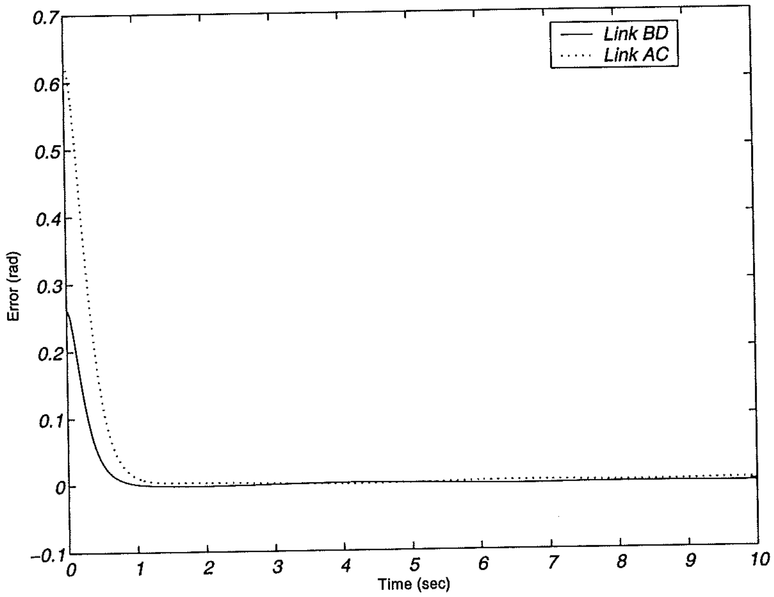


Fig. 8.14: The tracking errors in the case of the *Redundant Parallel Mode (Case II)*.

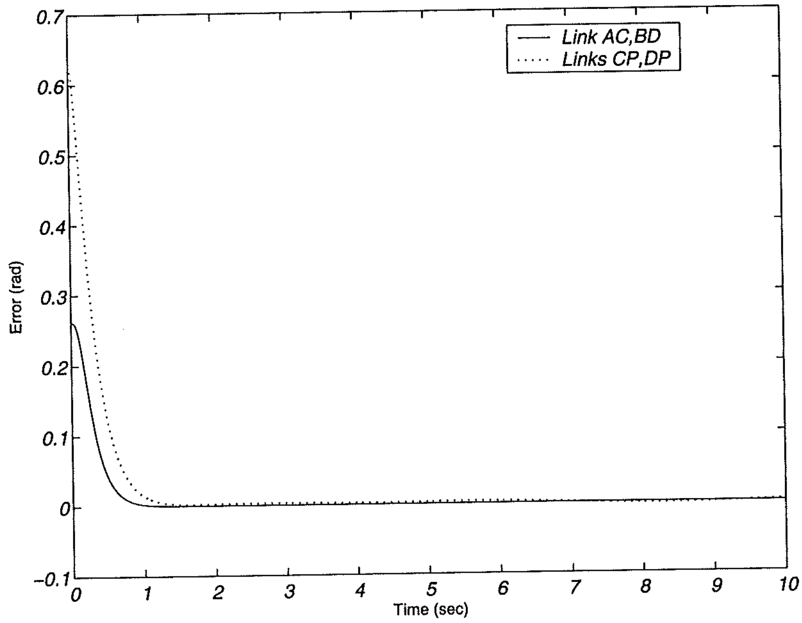


Fig. 8.15: The tracking errors in the case of the *Serial Mode (Case II)*.

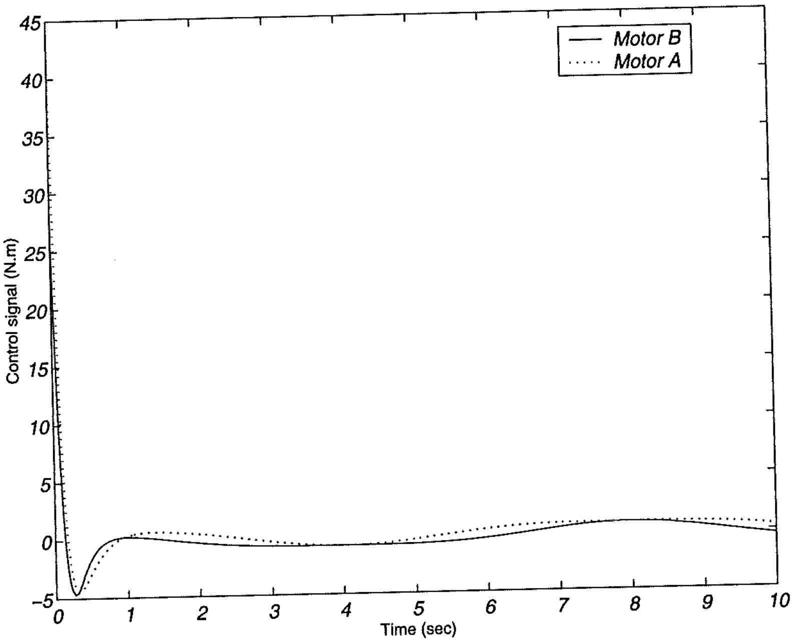


Fig. 8.16: Control torques in the case of the *Normal Parallel Mode (Case II).*

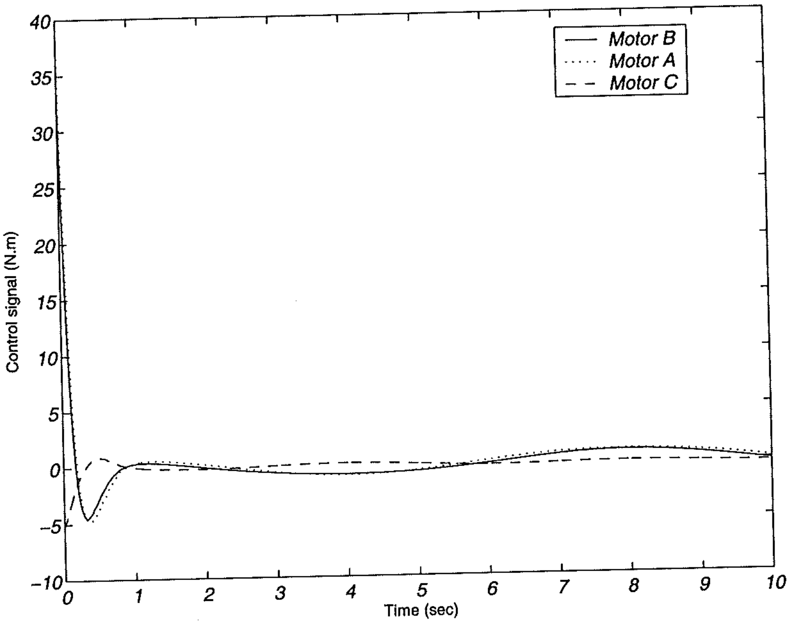


Fig. 8.17: Control torques in the case of the *Redundant Parallel Mode (Case II).*

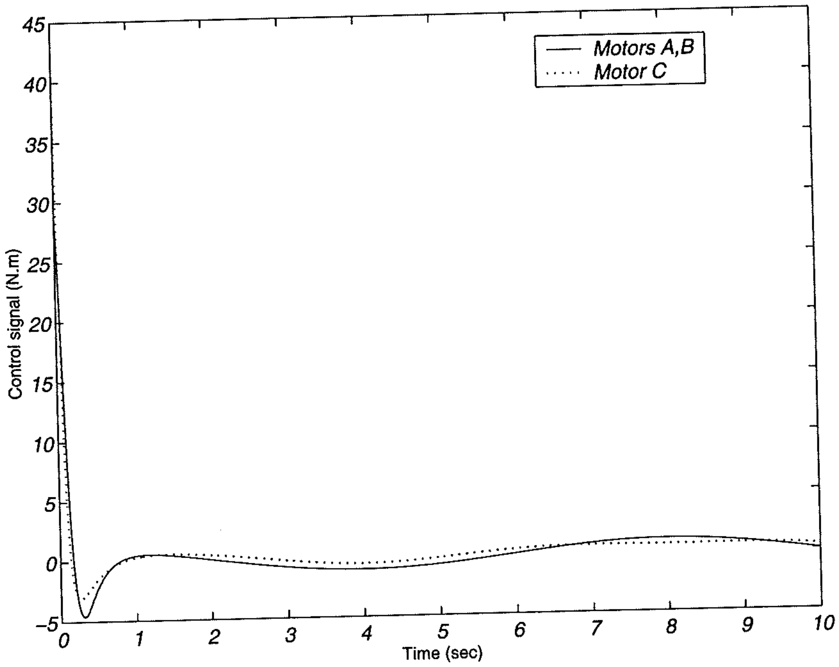


Fig. 8.18: Control torques in the case of the *Serial Mode (Case II)*.

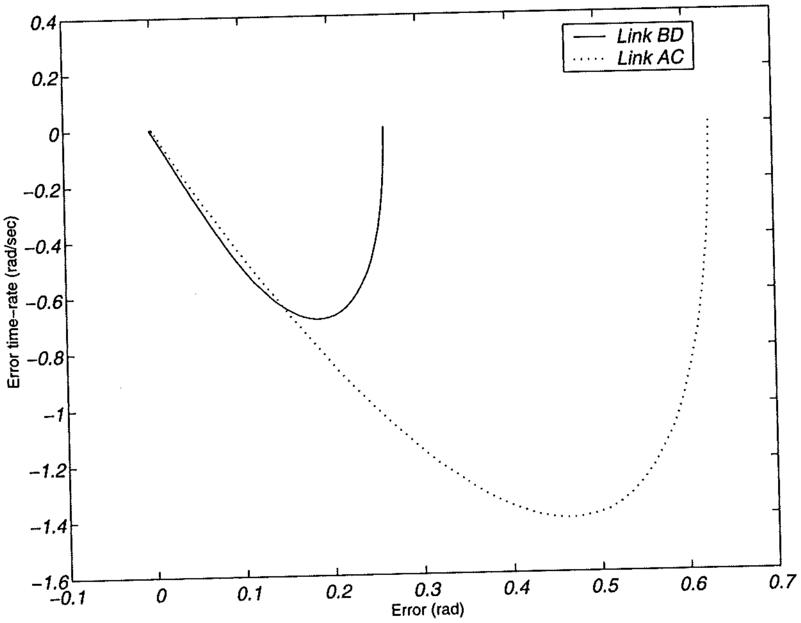


Fig. 8.19: Phase planes in the case of the *Normal Parallel Mode (Case II)*.

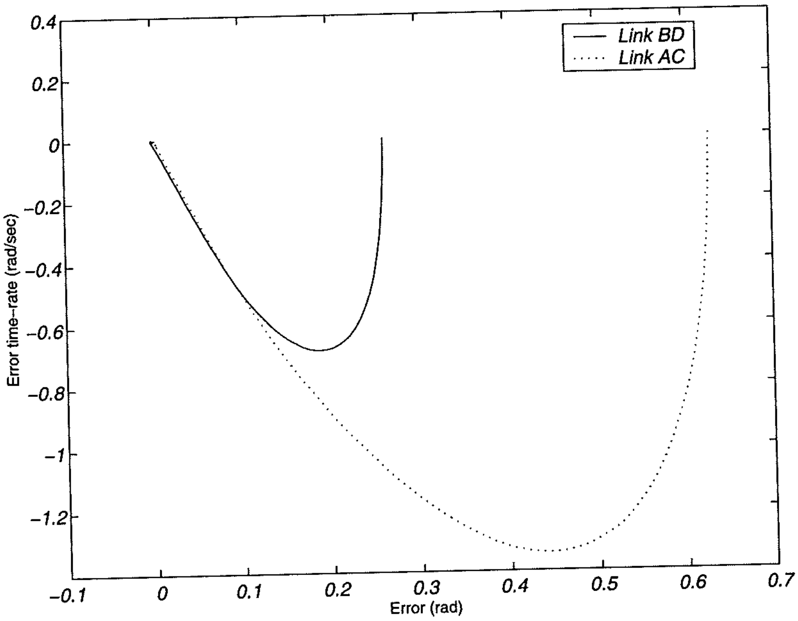


Fig. 8.20: Phase plane in the case of the *Redundant Parallel Mode (Case II)*.

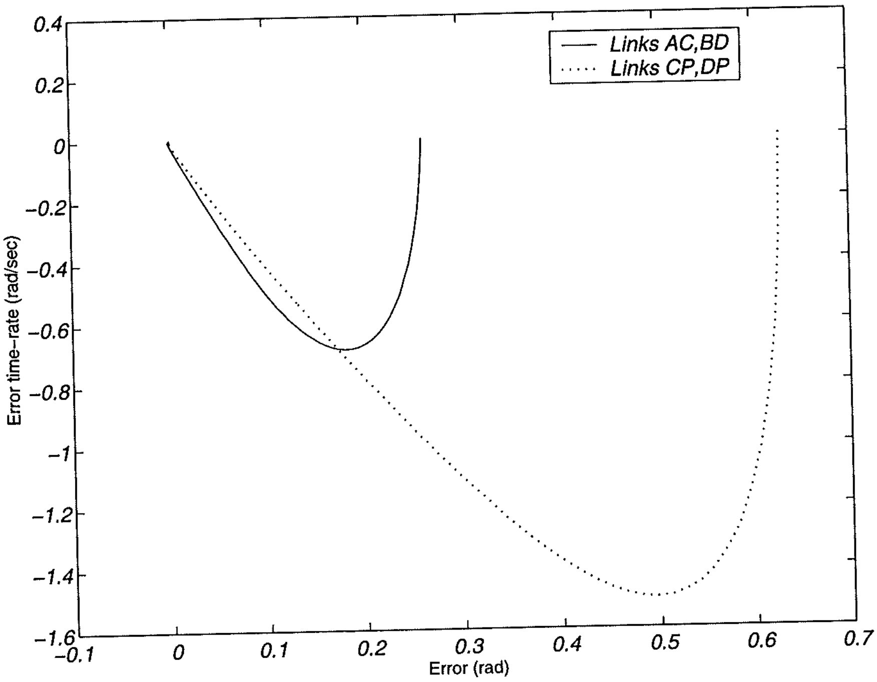


Fig. 8.21: Phase plane in the case of the *Serial Mode (Case II)*.

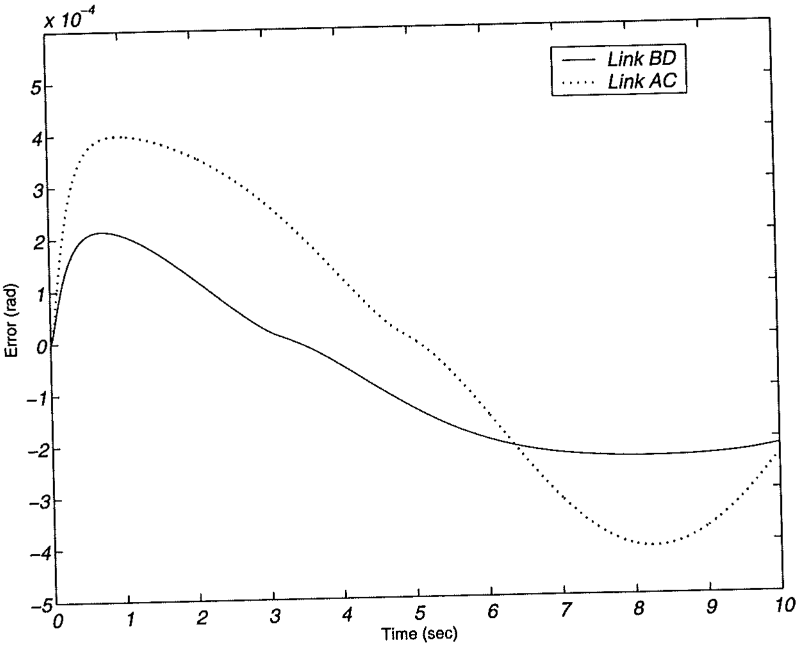


Fig. 8.22: The tracking errors in the case of the *Normal parallel mode* (slow-speed trajectory).

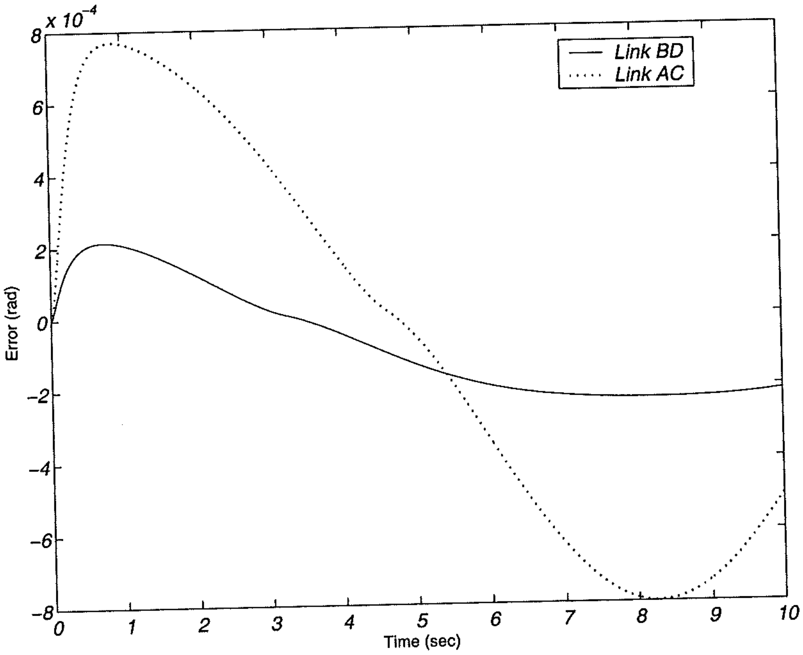


Fig. 8.23: The tracking errors in the case of the *redundant parallel mode* (slow-speed trajectory).

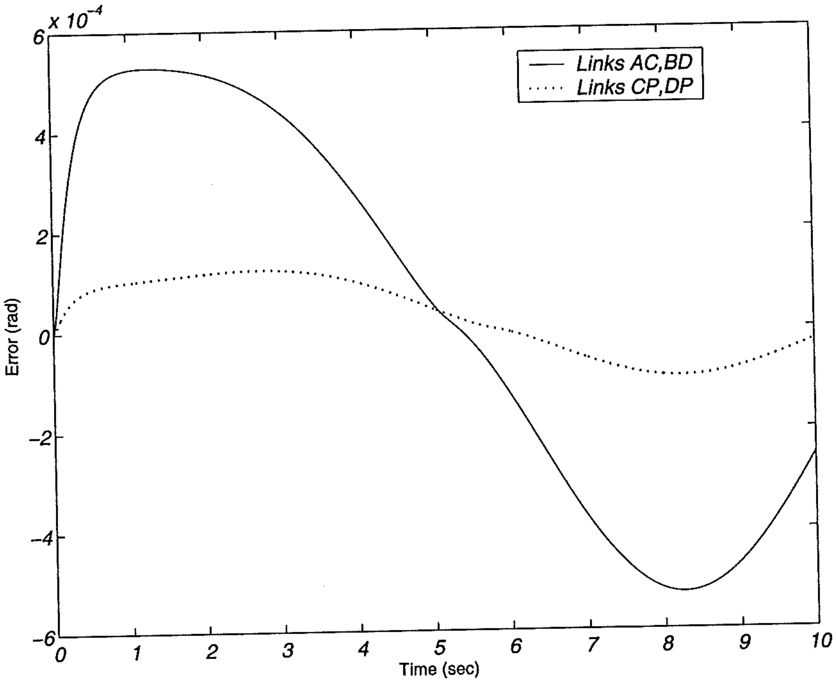


Fig. 8.24: The tracking errors in the case of the *Serial mode* (slow-speed trajectory).

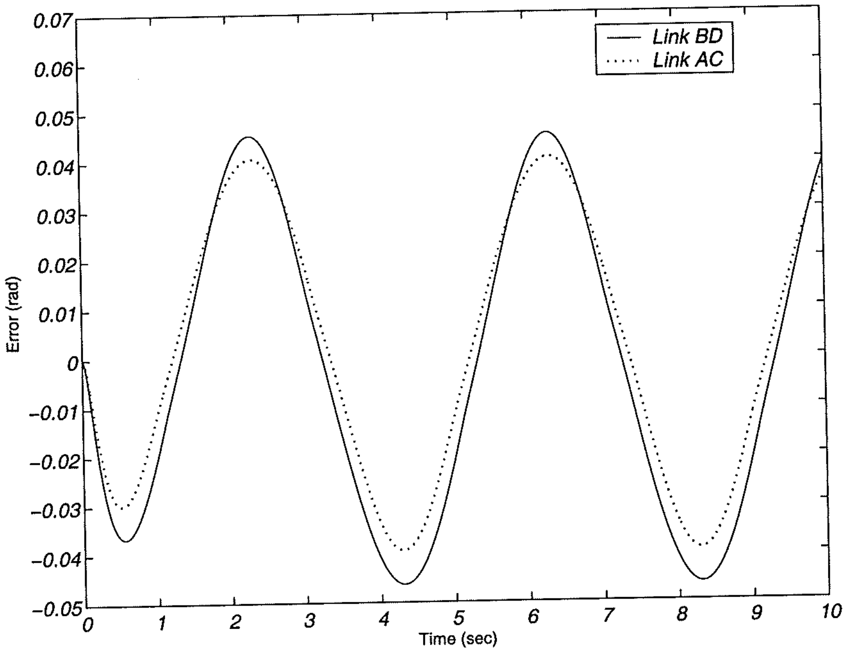


Fig. 8.25: The tracking errors in the case of the *Normal parallel mode* (fast-speed trajectory).

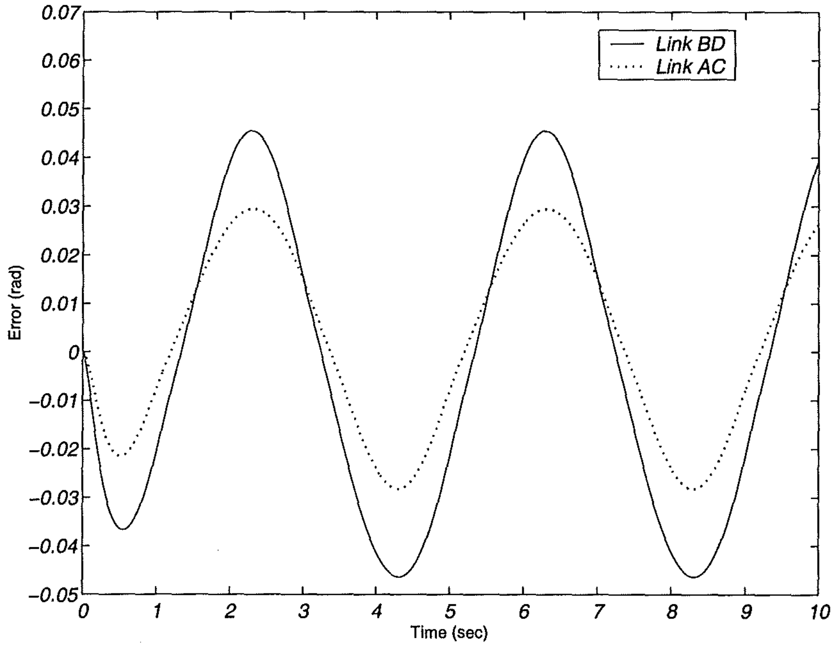


Fig. 8.26: The tracking errors in the case of the *redundant parallel mode* (fast-speed trajectory).

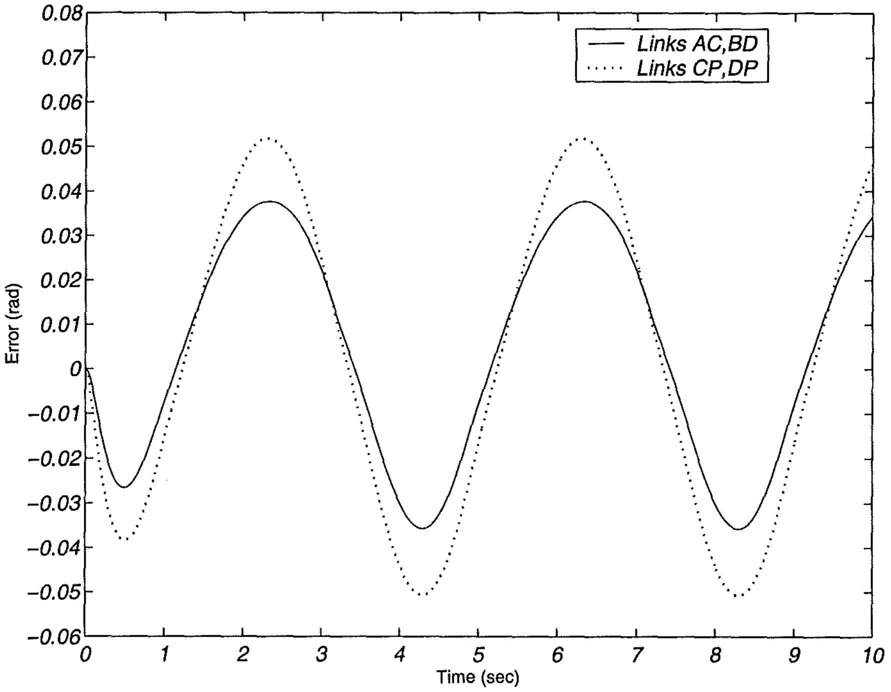


Fig. 8.27: The tracking errors in the case of the *Serial mode* (fast-speed trajectory).

8.8 Experiments

The following gains are used during the experiments,

$$K = \begin{bmatrix} 9 & 0 \\ 0 & 8 \end{bmatrix}, \Lambda = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix} \quad \& \quad q_1 = 5, q_2 = 8 \text{ and } \alpha = .7$$

The links are commanded to track the same trajectories as in the simulations but with three different speeds ($\omega = \frac{\pi}{8}, \frac{\pi}{4}$ and $\frac{\pi}{2}$ Hz). The *SIMULINK* block diagram is shown in Figure 8.28. The tracking errors and the associated control signals for the slow speed trajectory are shown in figures 8.29 and 8.30. The results of the medium and the fast speed trajectories are shown in Figures 8.31-8.34. The sampling rate is the same as in the previous chapter (100 Hz). In addition the tachometer signals are filtered using a first order Butterworth filter with a cut-off frequency of 50 Hz.

8.9 Discussion

On examining the results of the first two cases of simulations it can be found that for all robot designs the controller is able to track the specified trajectories with a very small error (nearly zero) despite the perturbations given to the robot parameters and the unknown dynamics. Besides this no chattering is reported in the control signals due to the reaching law used here. In addition, the system is able to reach the sliding surface in a finite time with little differences among the various dynamic modes due to the different level of perturbations and the contribution levels of the unknown dynamics.

In the last case of the simulations the results show that for the slow-speed trajectory the average maximum tracking error of the various mechanical designs is 0.00055 radians, approximately. Moving to the high-speed trajectory, the average obtained is 0.05 radians, which means that the tracking error is degraded by a very high percentage. It is important to note that this result is quite normal, as slope of the trajectory should be bounded due to the limited bandwidth of the system.

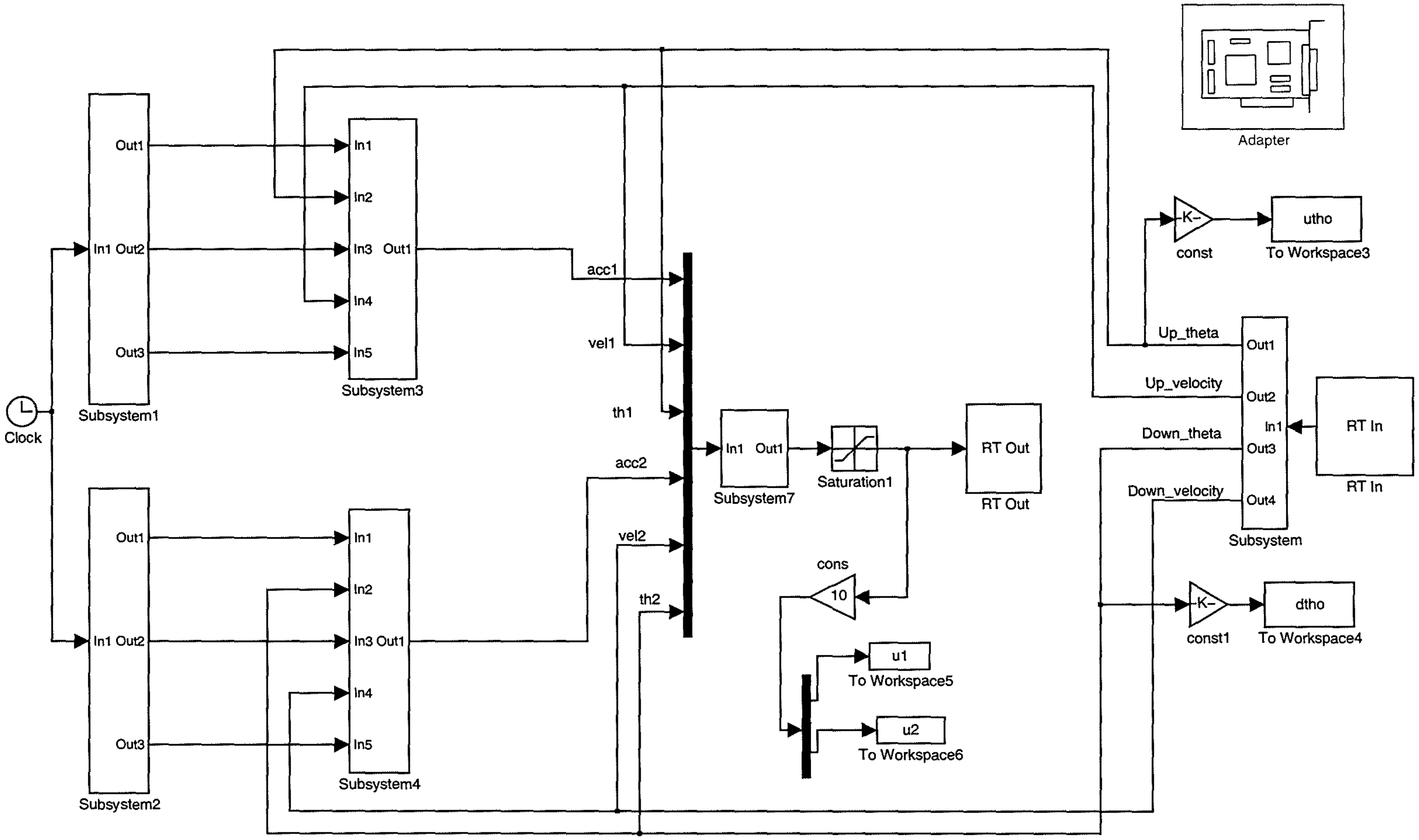


Fig. 8.28: Simulink block diagram for experiments.

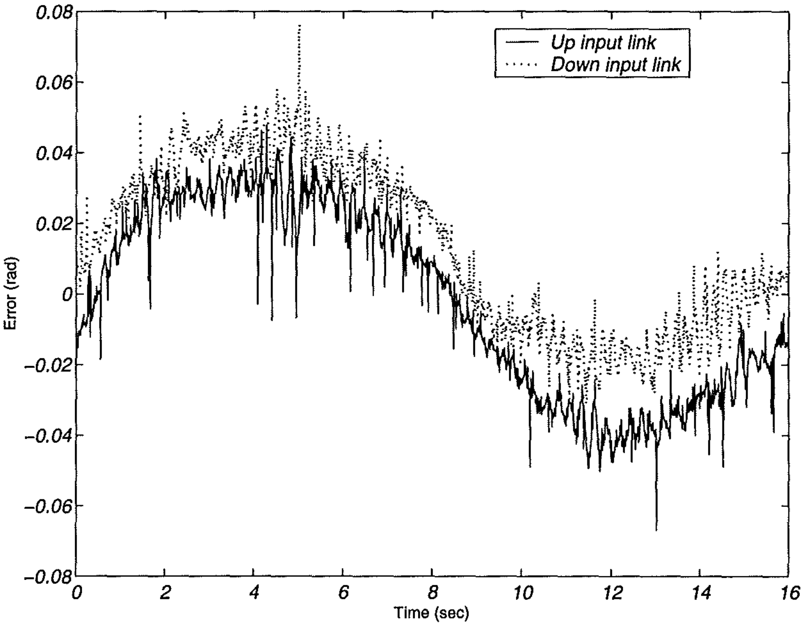


Fig. 8.29: The tracking errors in the case of the slow speed trajectory.

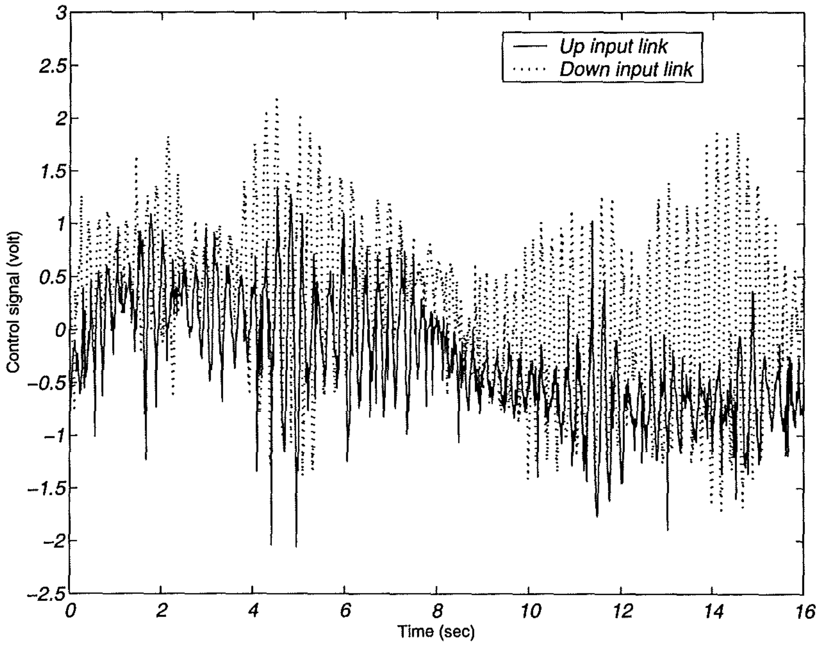


Fig. 8.30: The control signals in the case of the slow speed trajectory.

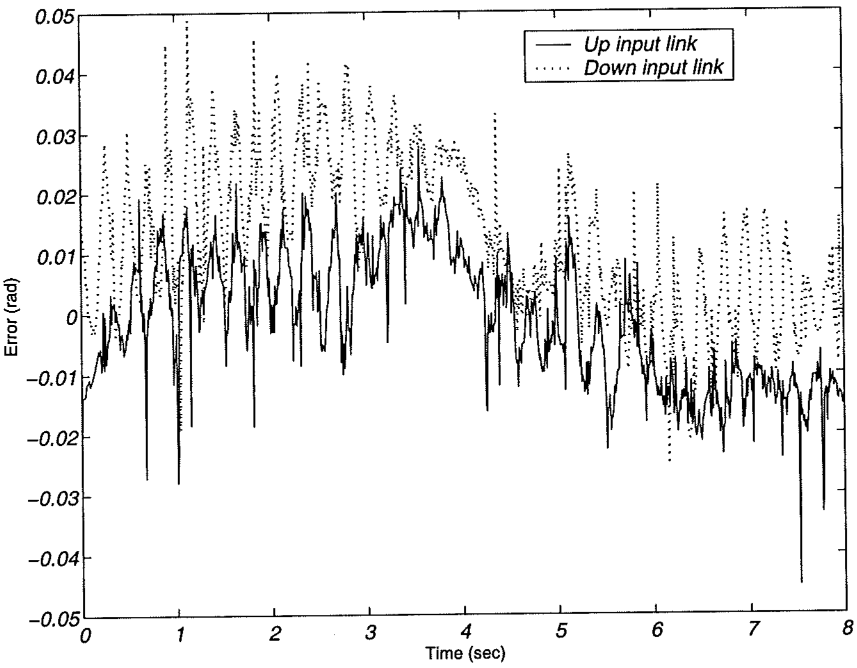


Fig. 8.31: The tracking errors in the case of the medium speed trajectory.

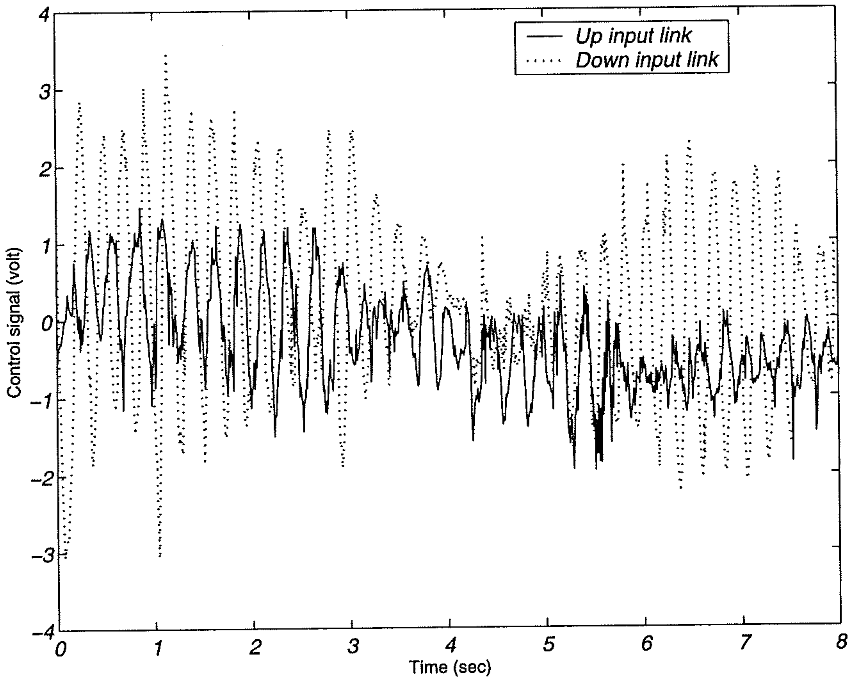


Fig. 8.32: The control signals in the case of the medium speed trajectory.

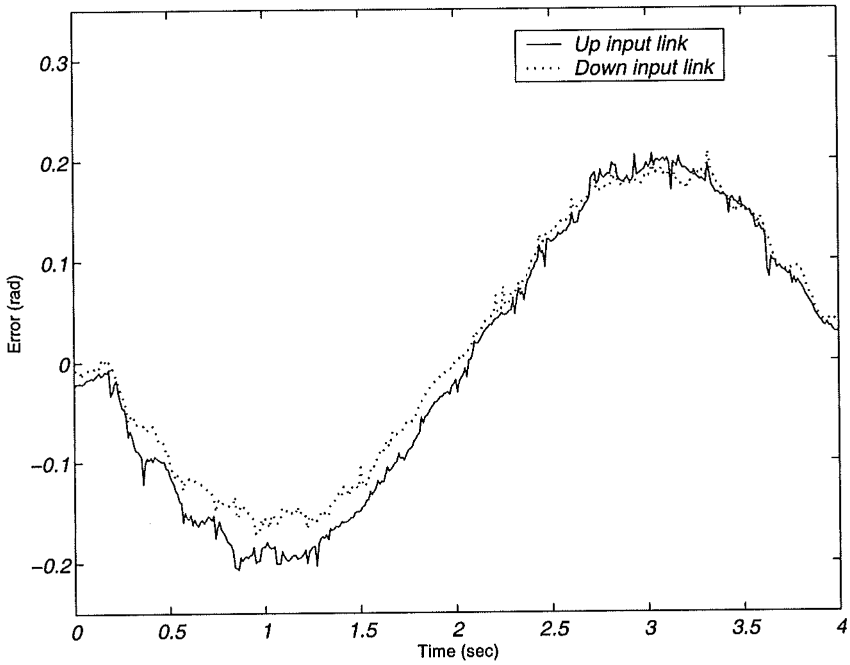


Fig. 8.33: The tracking errors in the case of the fast speed trajectory.

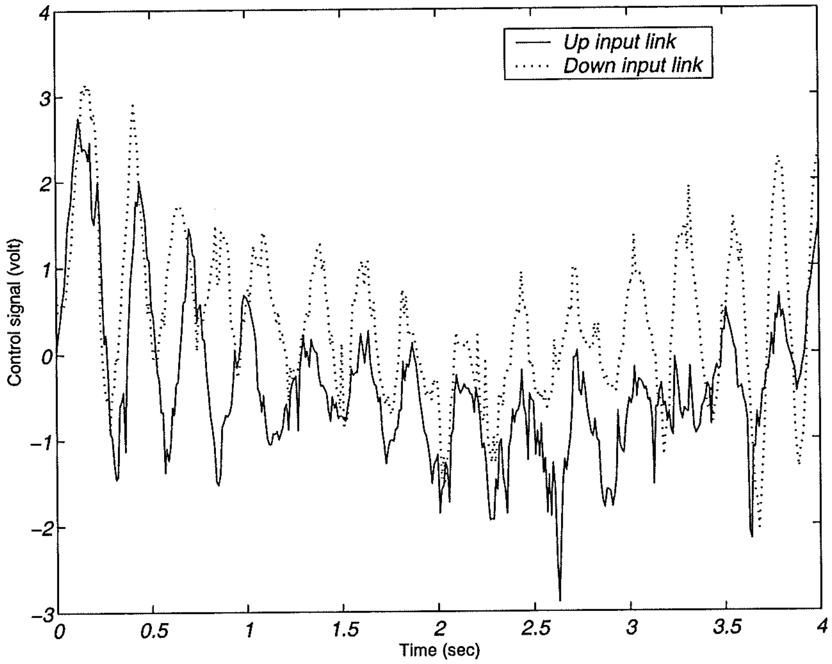


Fig. 8.34: The control signals in the case of the fast speed trajectory.

The limited sampling time and the noise in the actual system give a maximum tracking error of (0.05) radians, approximately, together with a little chattering. In fact, the most important factor is the sampling time which is very low compared with the bandwidth of the parallel robot. It is important to refer here to the fact that the control law used here does not contain any friction models. Besides this no information was available about the dynamics of the motors and their amplifiers. Despite these problems the controller is still able to keep the tracking error within a reasonable limit. The aforementioned results concern the slow and the medium speed trajectories. The results of the high-speed trajectory agree with those obtained by simulations. The error here is increased to (0.2) radians.

8.10 Conclusions

In this chapter it has been proven that the computed torque controllers are sliding mode controllers (SMC) but with unsuitable reaching dynamics. It has also been shown that chattering elimination in the conventional SMC through the boundary layer method leads to a normal computed torque control law. In addition to this it has been shown that the use of perturbation estimations to reduce the degradation of the error due to the use of the boundary layer method leads to a computed torque controller with an integral term. Following this a new robust computed torque controller has been designed based on the reaching law method. Simulations have been carried out in order to examine the robustness of the controller against the uncertainties in the robot parameters and the unknown dynamics. The SEPA robot has been used to ensure the effectiveness of the controller for the three designs tested. For practical examinations, experiments have been carried out on the existing parallel robot. It has been found that the theoretical results agree with those of the experiments. The most effective parameter found is the sampling rate which should be high enough to ensure low tracking errors.

Chapter 9

Evaluation of Some Nonlinear PID and PD Control Systems

9.1 Introduction

It is well known that a very simple PD controller, with gravity compensation, is an adequate solution for set-point control of robot manipulators to satisfy the stability requirement [71] [84]. The practical significance of this control technique lies in the fact that it requires no detailed knowledge of the manipulator dynamics except the gravity-loading vector. In practice, an integral action is added (PID controller) to reject constant perturbations at the cost of a reduced system bandwidth [93]. In fact, with the existing control structure it is impossible to select fixed gains which can critically damp the response to disturbances for all configurations of a given robot system. Average gains are always chosen which approximate critical damping at the centre of the manipulator workspace [44]. This inevitably results in overshoot at other positions within the manipulator workspace.

To cope with the non-uniform dynamic performance of the PID controller, Seraji [74-75] suggests using a sector-bounded nonlinear gain in cascade with the linear PID controller. This gain represents an even function of the error to give high outputs at high inputs and vice versa. From the author's point of view the gain allows a large corrective action when the error is large. As the error diminishes the gain is reduced in order to prevent large overshoots in the response. Following this, stability analysis of the P, PD, PI, and PID controllers have been performed using the Popov criterion after assuming linear dynamics for the robot with restrictions to single-input single-output. This restriction does not guarantee the stability of real systems. It is important to refer to the fact that the method is restricted only to systems with output feedback (velocity feedback is not allowed). In addition, no simulations have been performed to clarify the idea. Furthermore, examining the range of the variation of the gains, one can find that it is very small, which means no significant effect on the system performance.

Armstrong [76] used a nonlinear PD controller with P and D gains, each with two terms for force control. The first term represents the smaller control gain that is kept constant. The second term consists of a higher constant gain multiplied by a switching function that controls its application to the system. From the author's perspective the modulation of the P gain increases the damping while the modulation of the D gain shortens the rise time. In that work both gains are modulated with the larger values applied at large errors and the smaller values at low errors. The method is also restricted to linear systems with single-input single-output. The idea is different from the foregoing in that it can be applied to systems of output and velocity feedback.

A quick review of the research efforts in the area of nonlinear PD or PID control shows that one can find that it has all centred on one idea; the use of variable gains to improve the system response. This is a very old technique and the examination of early textbooks [94] will shows that it is really based on the idea of the nonlinear servomechanism as built by J. B. Lewis and described in [94]. The idea was to have a positioning system whose damping is negative, or at least very small, when the error is large. This tends to ensure a more rapid response to large errors than in the corresponding linear system, and this desirable effect was accomplished by using a nonlinear velocity feedback dependent on the absolute value of the error. Although the system response was improved for step inputs, instability was reported for double pulse inputs.

In this chapter, the nonlinear PID controller of Seraji [74-75] is evaluated experimentally. Following this, a nonlinear PD controller with a modulated P and D gain is designed and then experimentally evaluated. The aim here is to evaluate these specific nonlinear control methods in application to real systems.

9.2 Evaluation of the PID controller of Seraji

For the i^{th} degree of freedom the control law is given by the following equation,

$$u_i = k_{P_i}\tilde{e}_i + k_{V_i}\dot{\tilde{e}}_i + k_{I_i}\int \tilde{e}_i dt \qquad (9.1)$$

Where k_P, k_V, k_I are constant gains and $\tilde{e}$ represents the scaled error (the error e multiplied by the nonlinear gain) as defined by the following equation,

$$\tilde{e}_i = \left\{1 - \frac{1}{\exp(0.5e_i) + \exp(-0.5e_i)}\right\} e_i \qquad (9.2)$$

Here, the nonlinear gain is represented by a Hyperbolic function. The following practical gain values are used during the experiments,

$$k_P = 3, k_V = 1, \text{and } k_I = 0.3$$

The performance of the system under the application of the control law of (9.1) is compared to its counterpart under the application of a linear control law. Two different sets for the gains of the linear law have been chosen and are as follows,

(Case I) $k_P = 3, k_V = 1, \text{and } k_I = 0.3$

(Case II) $k_P = 2.25, k_V = 0.75, \text{and } k_I = 0.225$

The gain values of the second case result from multiplying the average of the extreme values of the output of the nonlinearity by the gains of the first case. This is done to enhance the evaluation process that is extended to include the P, PI, PD controllers, and leads to an examination of the system stability under these control laws. The robot, under all the various kinds of control laws, is subjected to two types of inputs. The first is a step input and the second is a sine function trajectory. It is necessary to refer here to the fact that due to the simplicity of the control laws it is possible to fix the sampling rate to the value of 200 Hz; this being twice the sampling rate that has been used in chapters 7 and 8. The Simulink solver is the ODE 45 routine. All information from the experiments, such as the responses, angular speeds, control signals, is extracted and is exhibited for full examination of the system performance.

9.2.1 Results

- **Step Input**

The results of the P and the PI control (Figures 9.1–9.12) for $\theta_1,\theta_2,\omega_1,\omega_2,u_1$ and u_2 show that the nonlinear control tends to be more stable than the linear alternative. Generally, all are stable due to the existence of friction in the system. Under linear control, the damped oscillations are always symmetric about the desired goal. The damped oscillations in the nonlinear control exhibit a very different behaviour. Under the linear PI control (case II), Figures 9.7-9.12, the oscillations have pushed the arm to the singularity state. This is explained on the graph by the disappearance of the oscillations after $t = 7\,\text{sec}$. It is necessary to note that the disappearance of the oscillations occurs because the arm has been stopped manually to prevent damage to the system due to singularity.

The results of the PD and the PID control (Figures 9.13–9.24) $\theta_1,\theta_2,\omega_1,\omega_2,u_1$ and u_2 show no significant differences between the nonlinear and the linear control. The linear PID controller with higher gains exhibits reasonable performance against the existing stiction-friction. This is logical behaviour because the high gains allow enough control torque at low errors to force the system to converge to its goal. This cannot be achieved by the nonlinear control because the gains are drastically decreased at low errors.

- **Sine input**

The results of the P and PI control (Figures 9.25-9.36) $\theta_1,\theta_2,\omega_1,\omega_2,u_1$ and u_2 show that the linear control, especially the one with higher gains, allows better tracking than the non-linear control. The PD and the PID results (Figures 9.37-48) show that the linear control of high gains is the best in tracking the sine trajectory. However, there is no big difference between the linear and the non-linear P, PI, PD and the PID control.

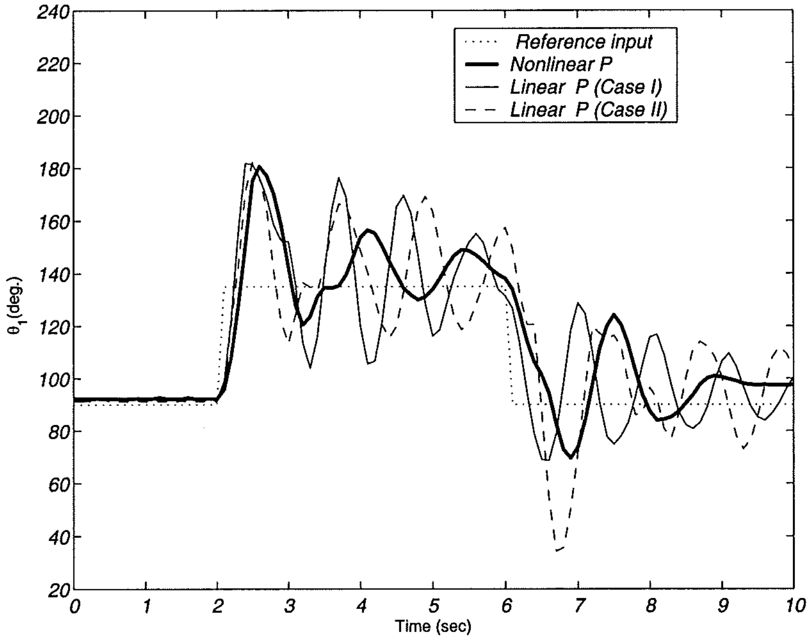


Fig. 9.1: Angular position of the first input link (P control).

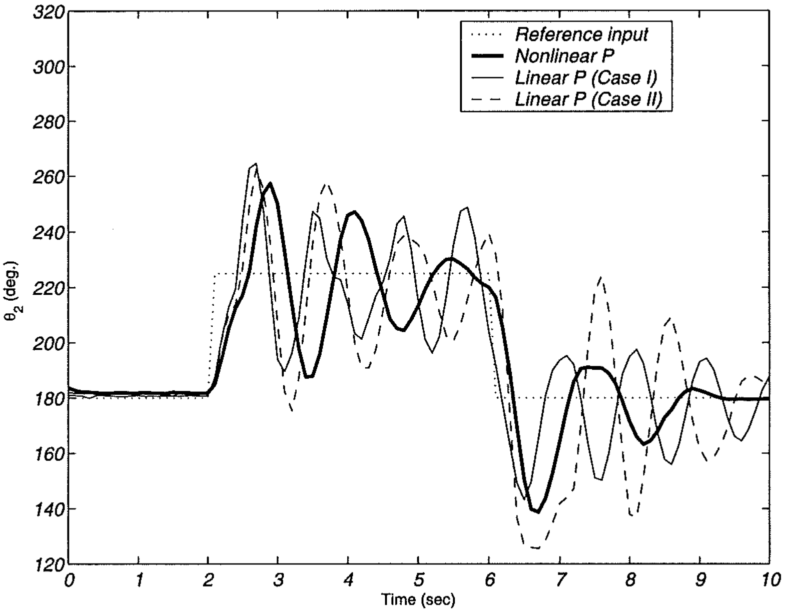


Fig. 9.2: Angular position of the second input link (P control).

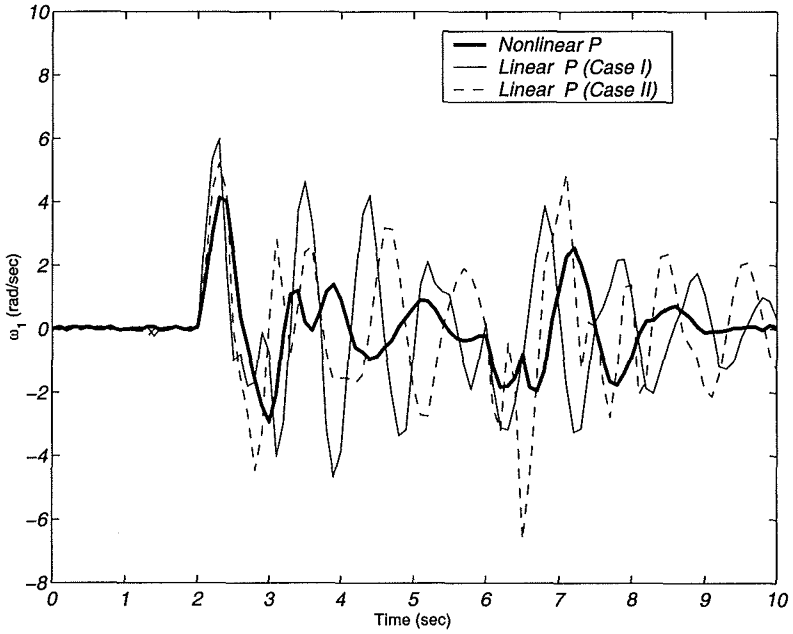


Fig. 9.3: Angular speed of the first input link (P control).

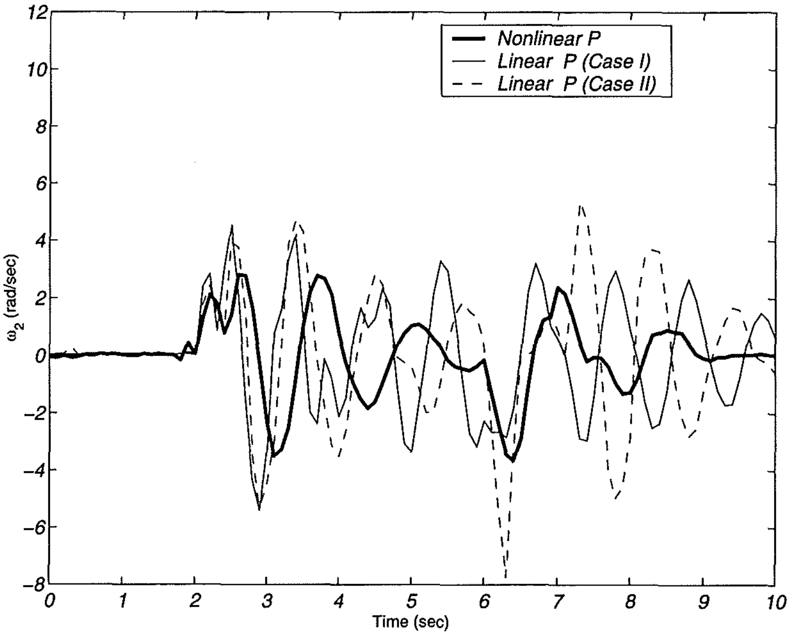


Fig. 9.4: Angular speed of the second input link (P control).

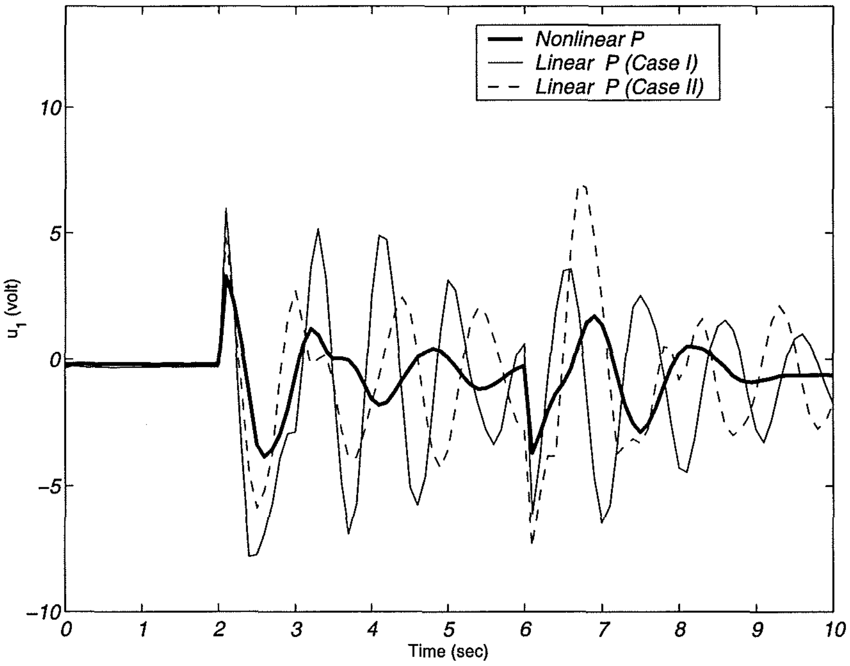


Fig. 9.5: Control signal of the first input link (P control).

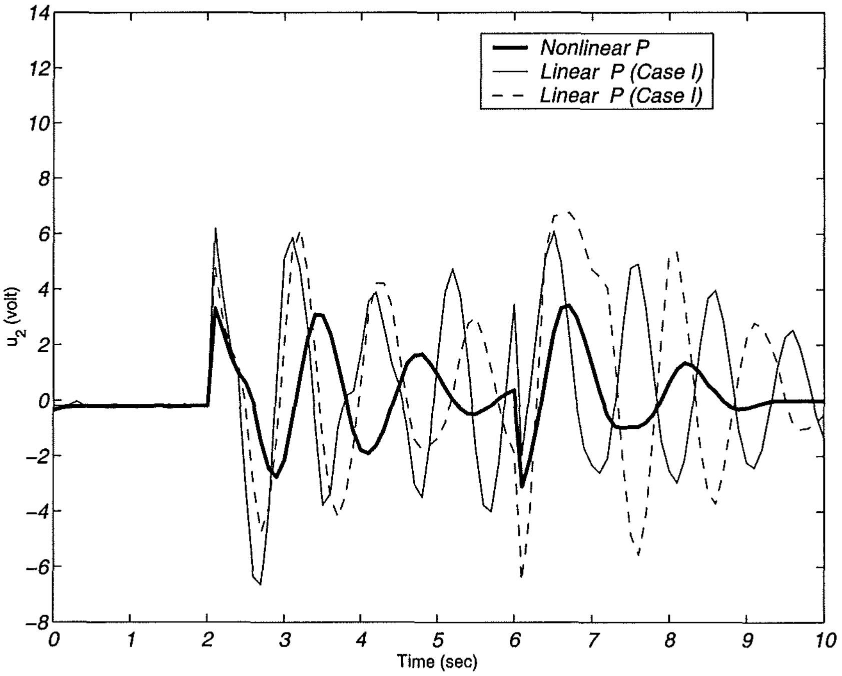


Fig. 9.6: Control signal of the second input link (P control).

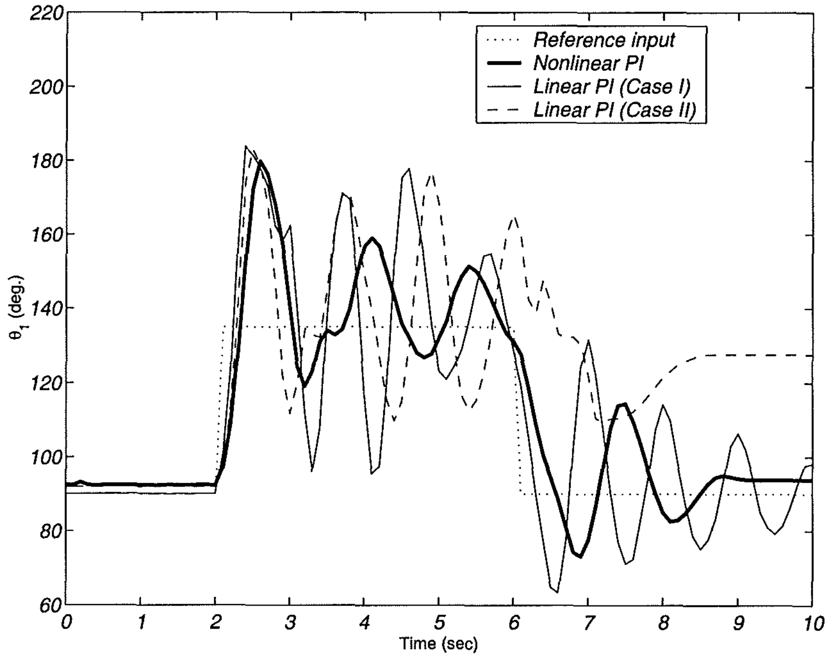


Fig. 9.7: Angular position of the first input link (PI control).

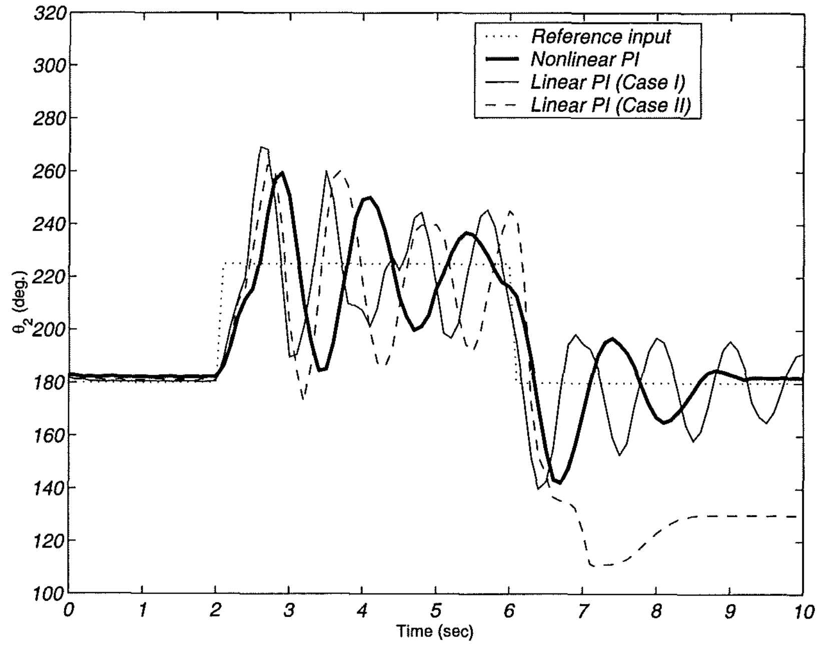


Fig. 9.8: Angular position of the second input link (PI control).

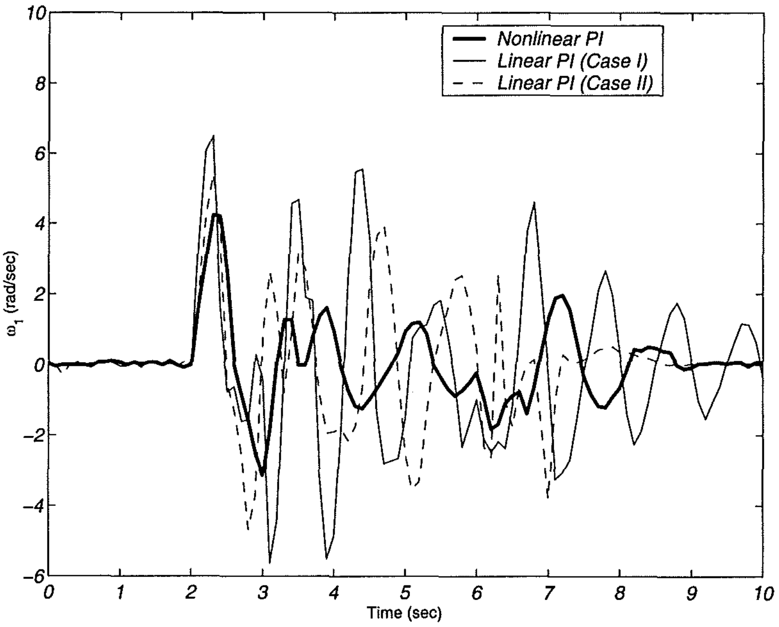


Fig. 9.9: Angular speed of the first input link (PI control).

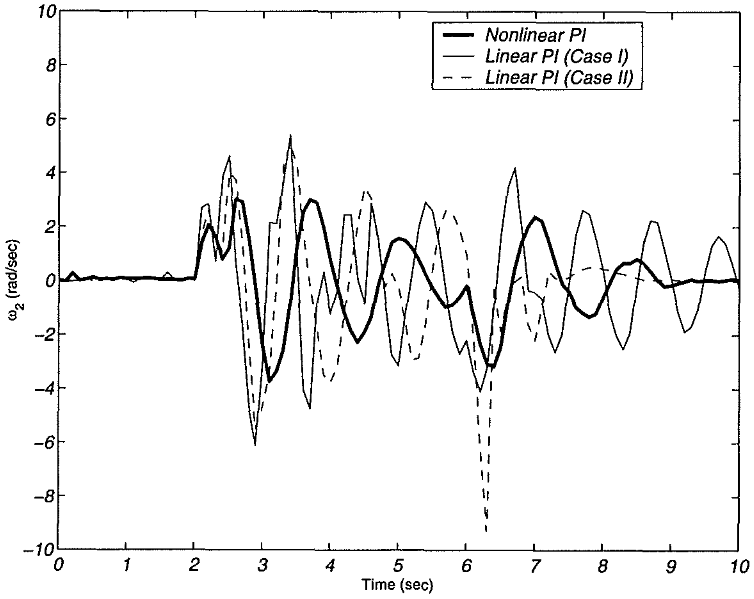


Fig. 9.10: Angular speed of the second input link (PI control).

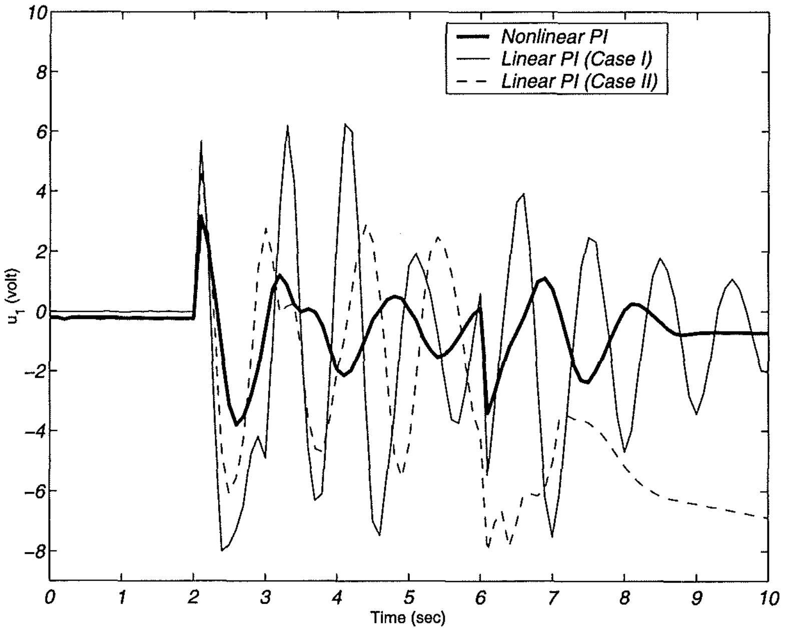


Fig. 9.11: Control signal of the first input link (PI control).

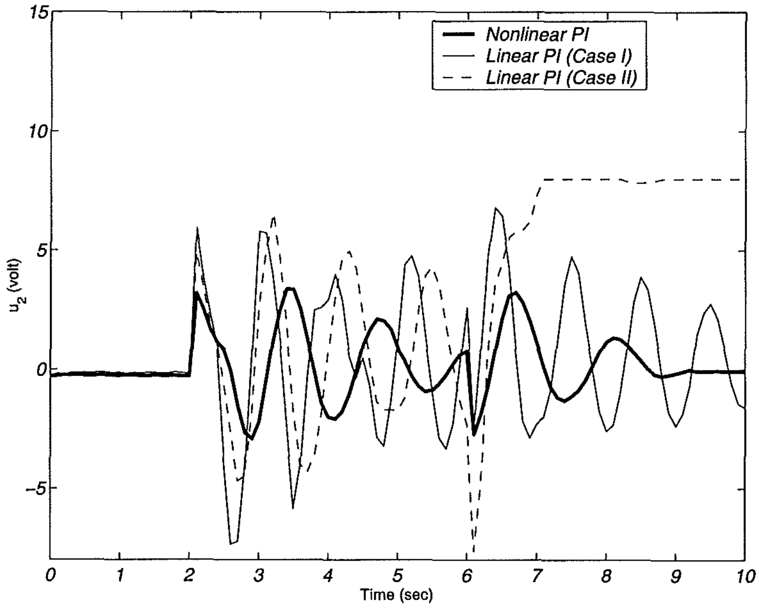


Fig. 9.12: Control signal of the second input link (PI control).

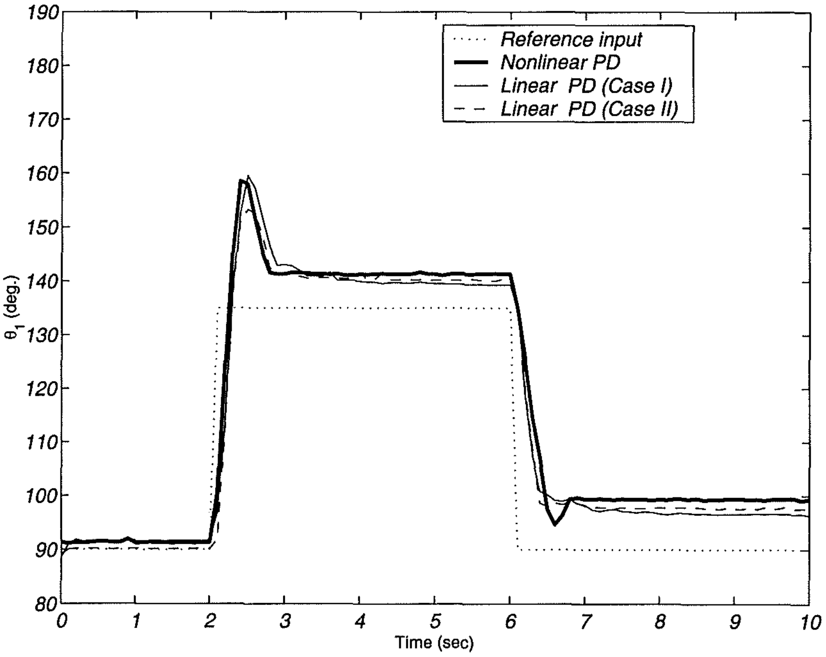


Fig. 9.13: Angular position of the first input link (PD control).

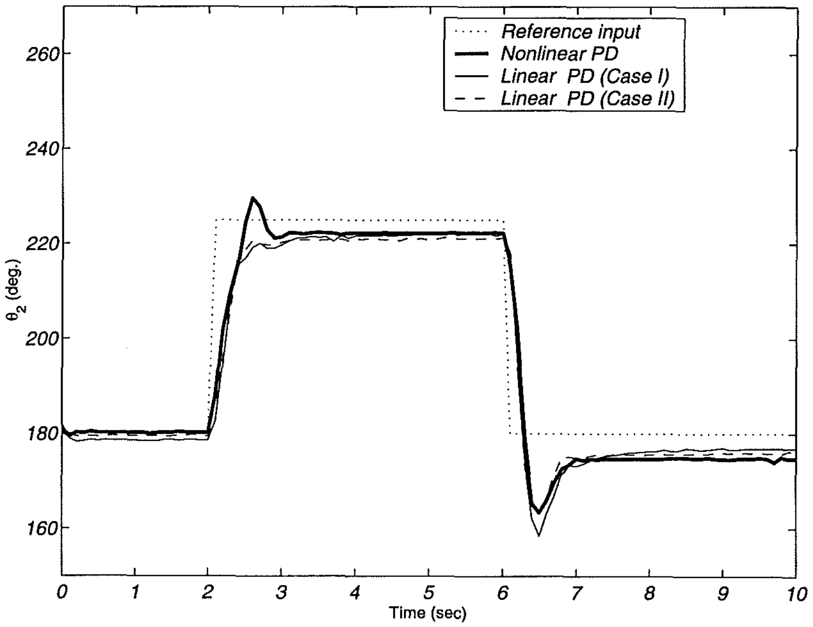


Fig. 9.14: Angular position of the second input link (PD control).

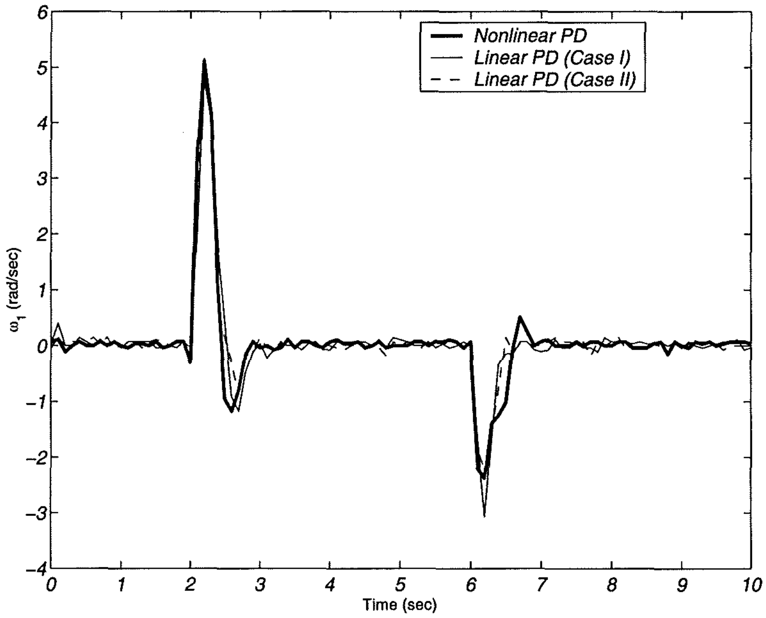


Fig. 9.15: Angular speed of the first input link (PD control).

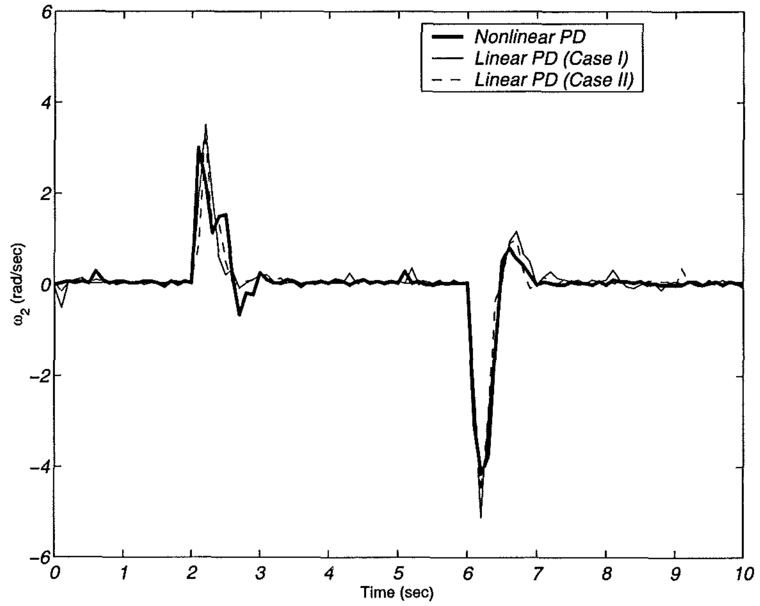


Fig. 9.16: Angular speed of the second input link (PD control).

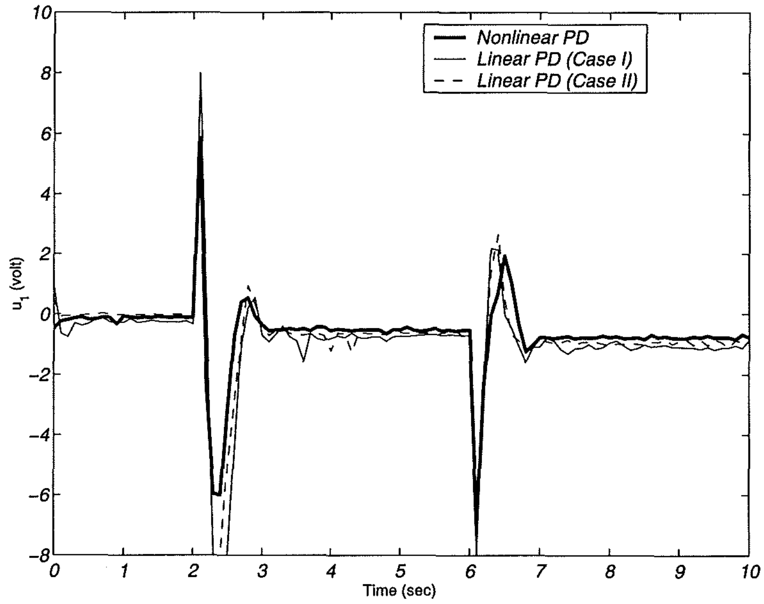


Fig. 9.17: Control signal of the first input link (PD control).

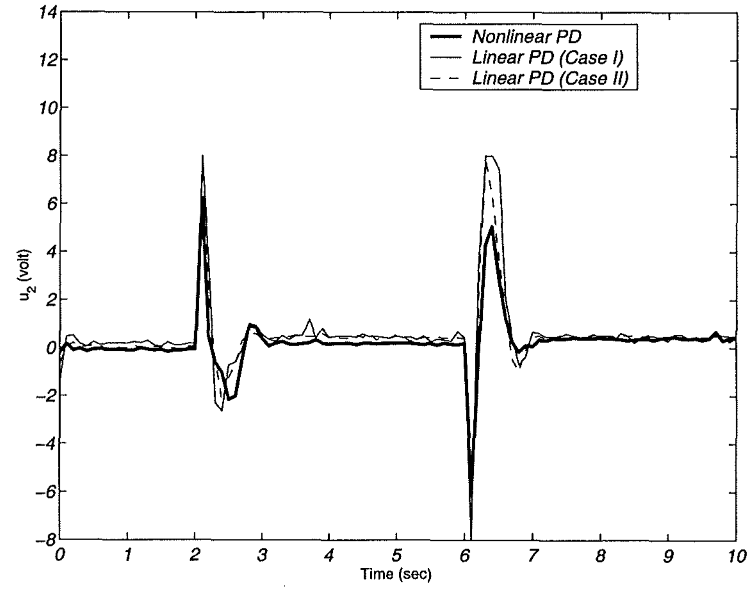


Fig. 9.18: Control signal of the second input link (PD control).

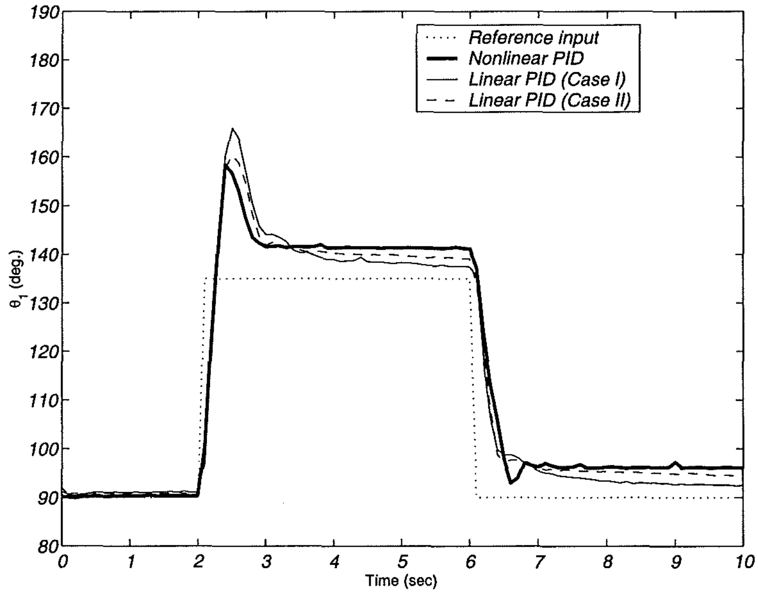


Fig. 9.19: Angular position of the first input link (PID control).

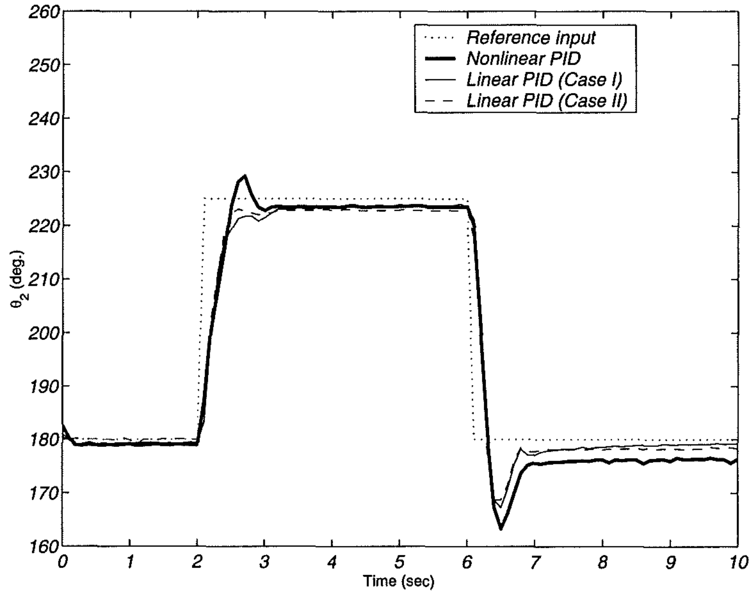


Fig. 9.20: Angular position of the second input link (PID control).

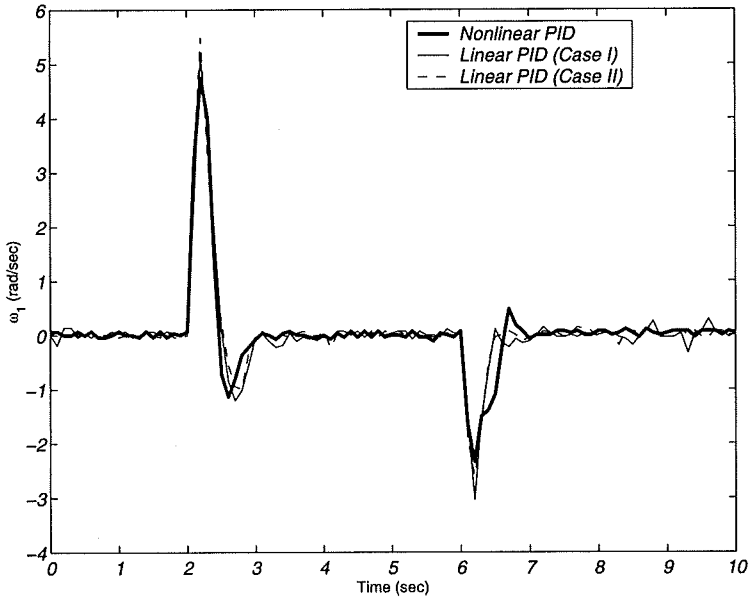


Fig. 9.21: Angular speed of the first input link (PID control).

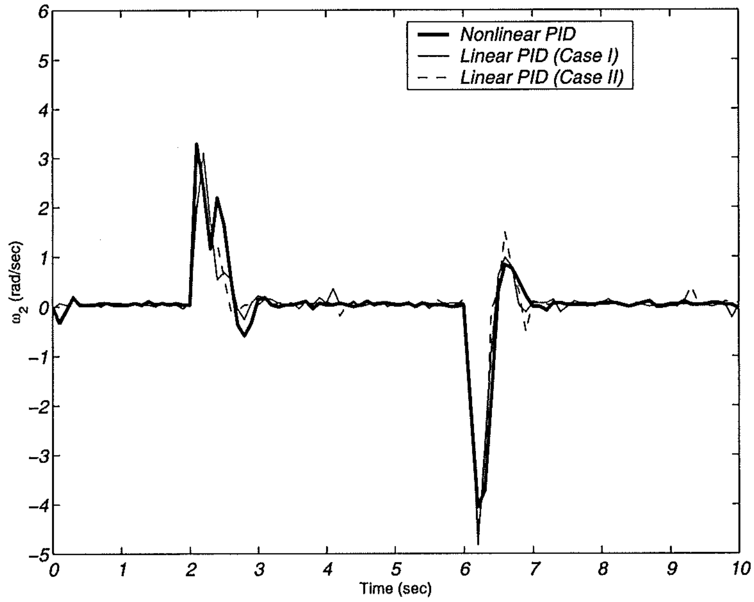


Fig. 9.22: Angular speed of the second input link (PID control).

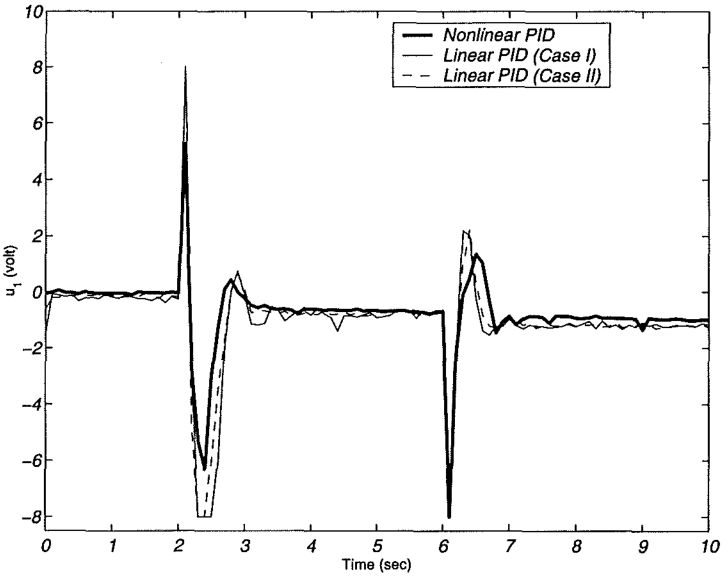


Fig. 9.23: Control signal of the first input link (PID control).

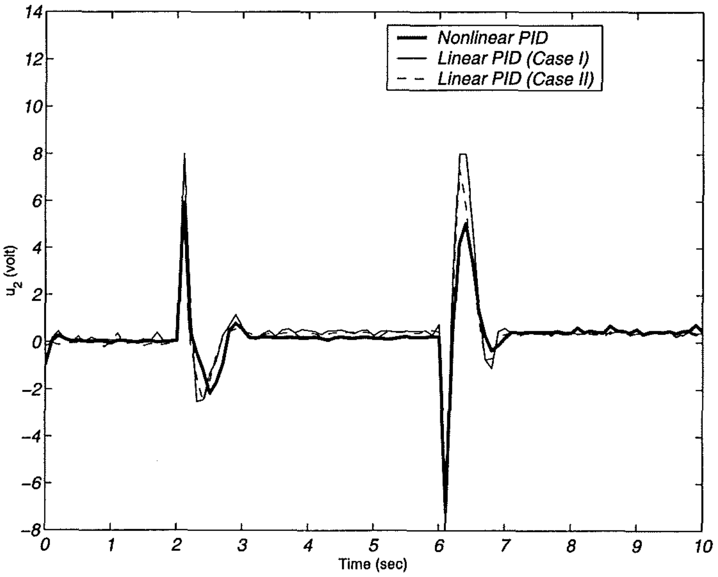


Fig. 9.24: Control signal of the second input link (PID control).

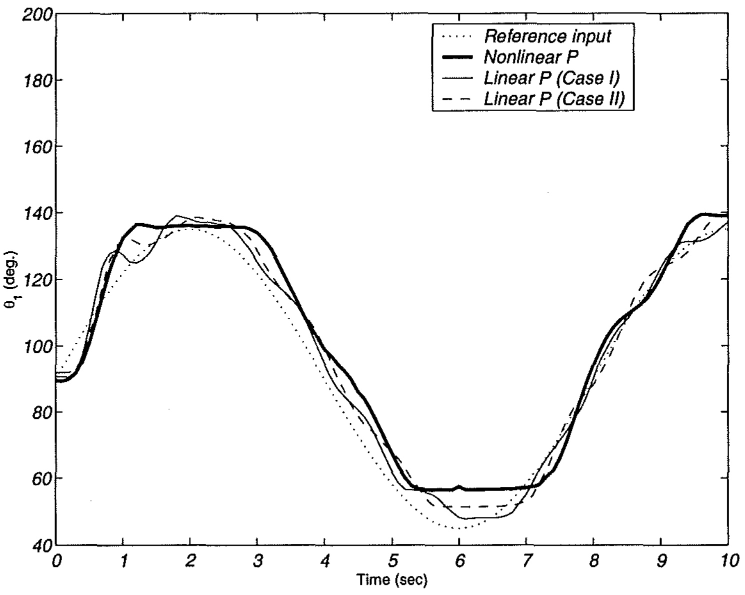


Fig. 9.25: Angular position of the second input link (P control).

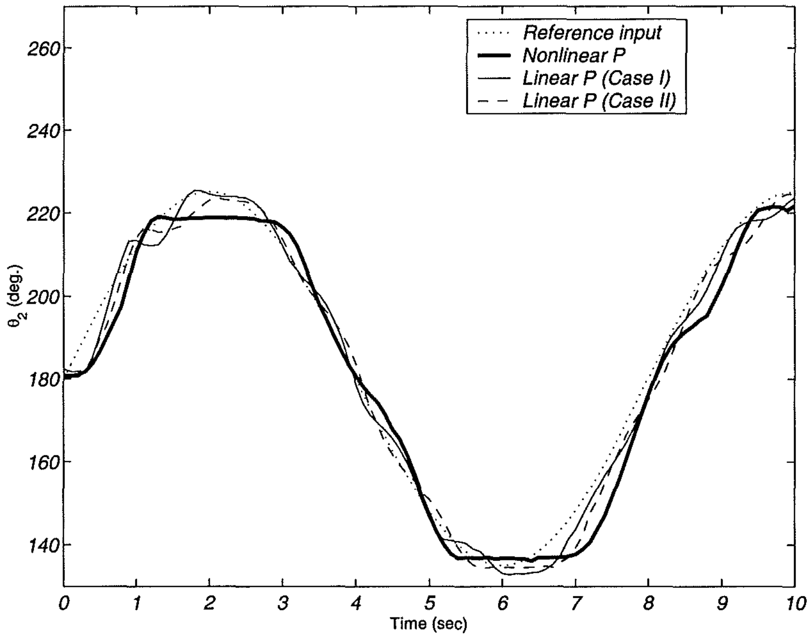


Fig. 9.26: Angular position of the second input link (P control).

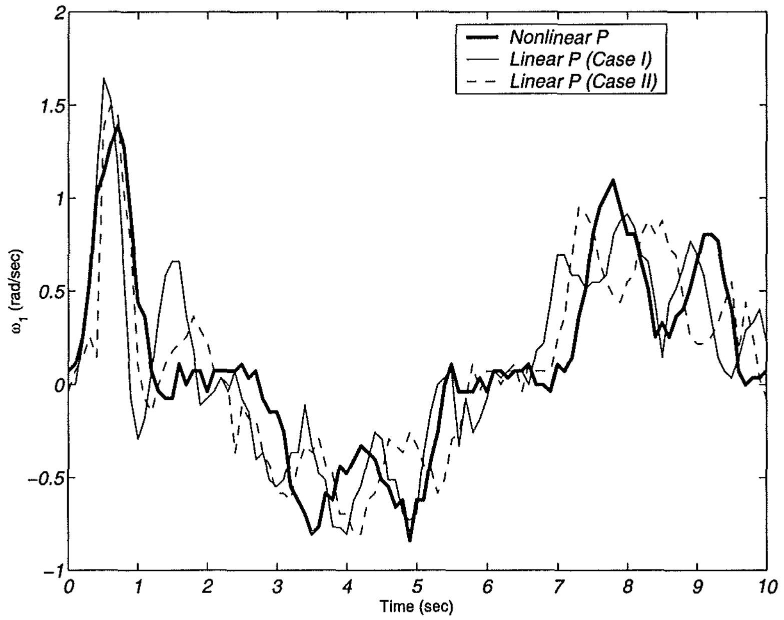


Fig. 9.27: Angular speed of the first input link (P control).

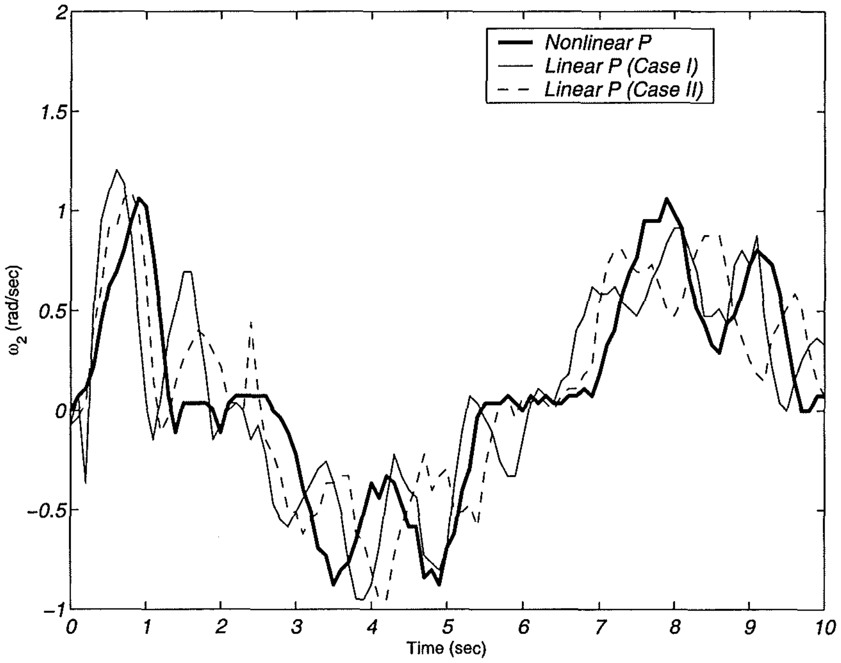


Fig. 9.28: Angular speed of the second input link (P control).

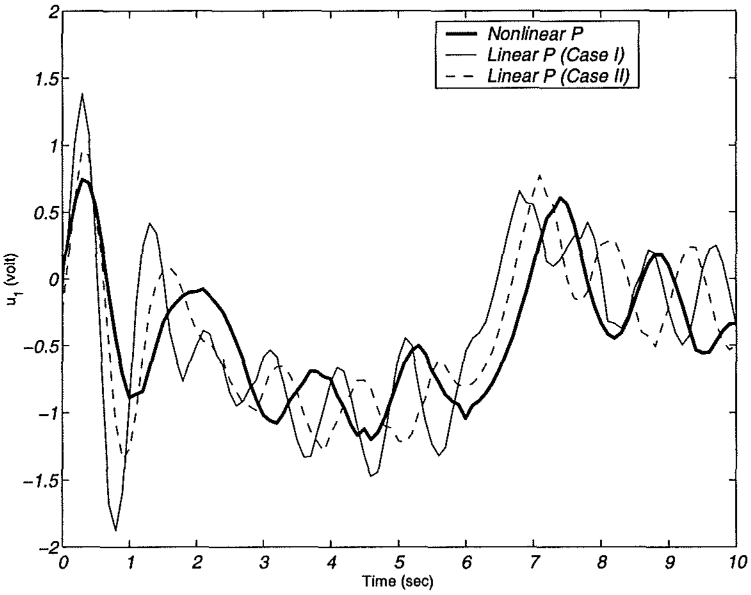


Fig. 9.29: Control signal of the first input link (P control).

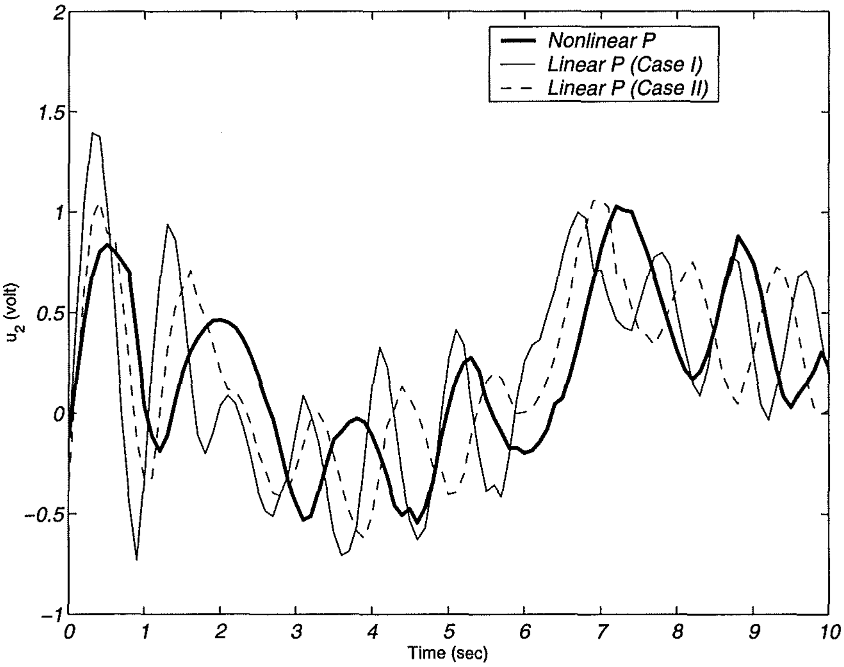


Fig. 9.30: Control signal of the second input link (P control).

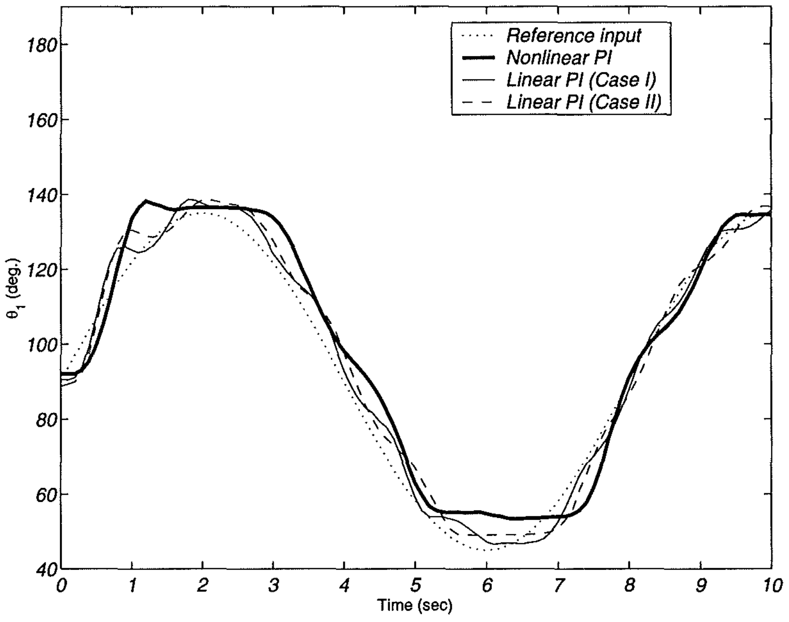


Fig. 9.31: Angular position of the second input link (PI control).

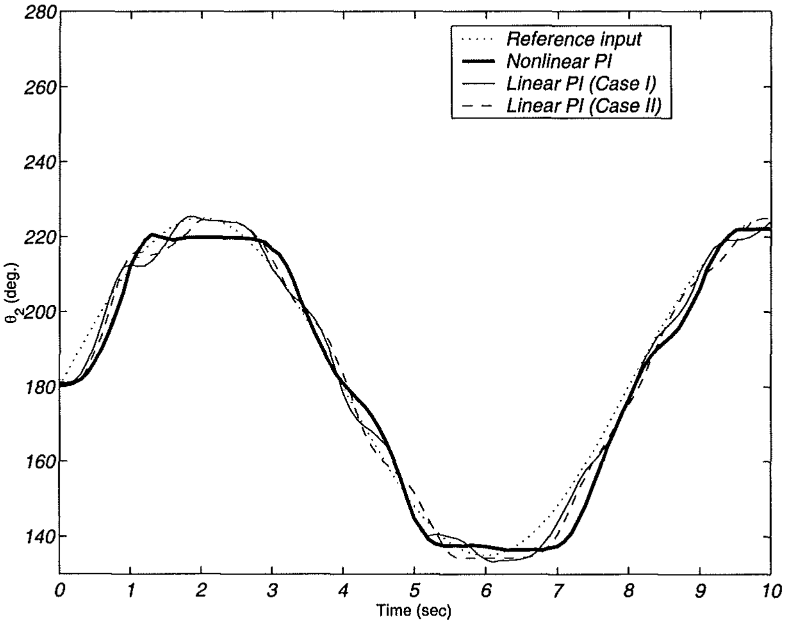


Fig. 9.32: Angular position of the second input link (PI control).

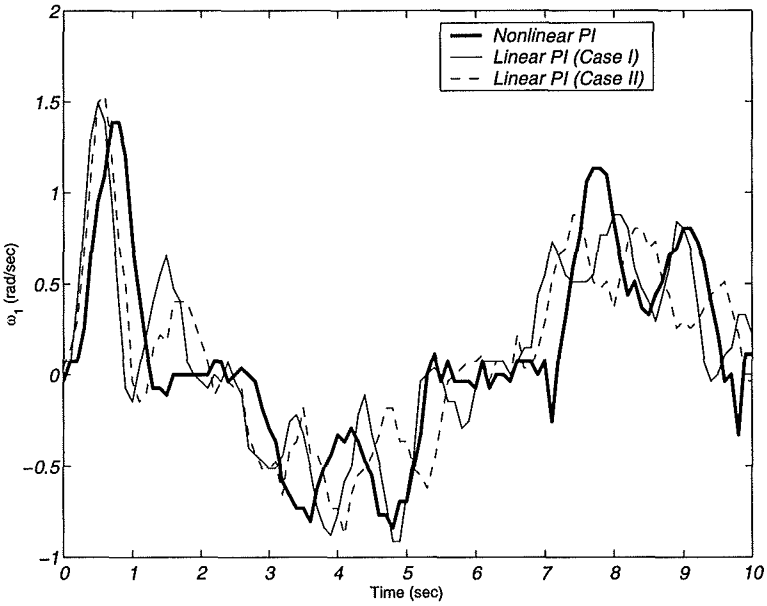


Fig. 9.33: Angular speed of the first input link (PI control).

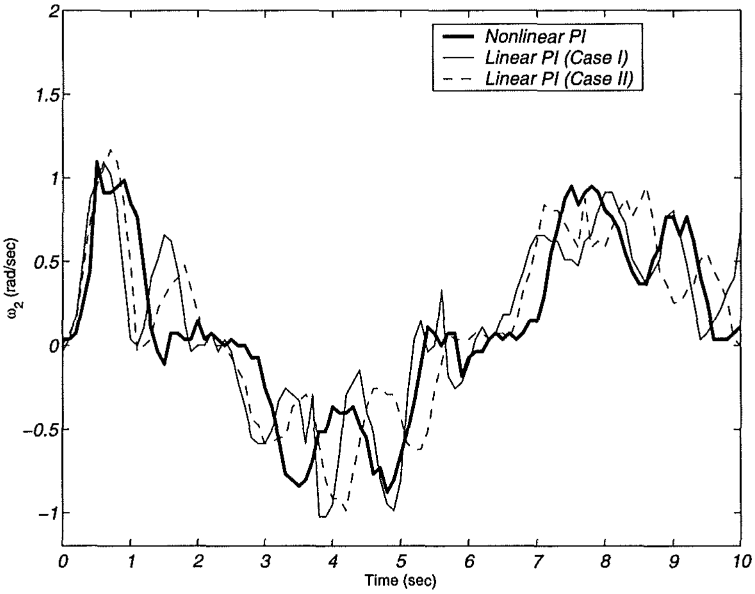


Fig. 9.34: Angular speed of the second input link (PI control).

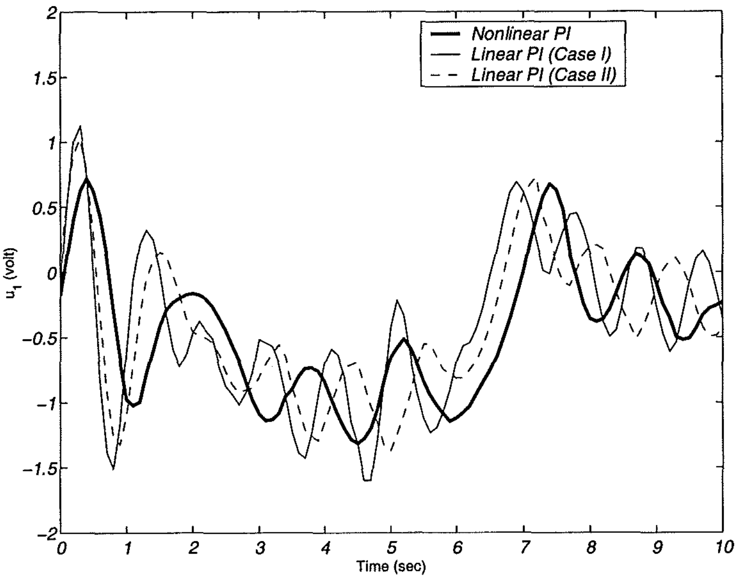


Fig. 9.35: Control signal of the first input link (PI control).

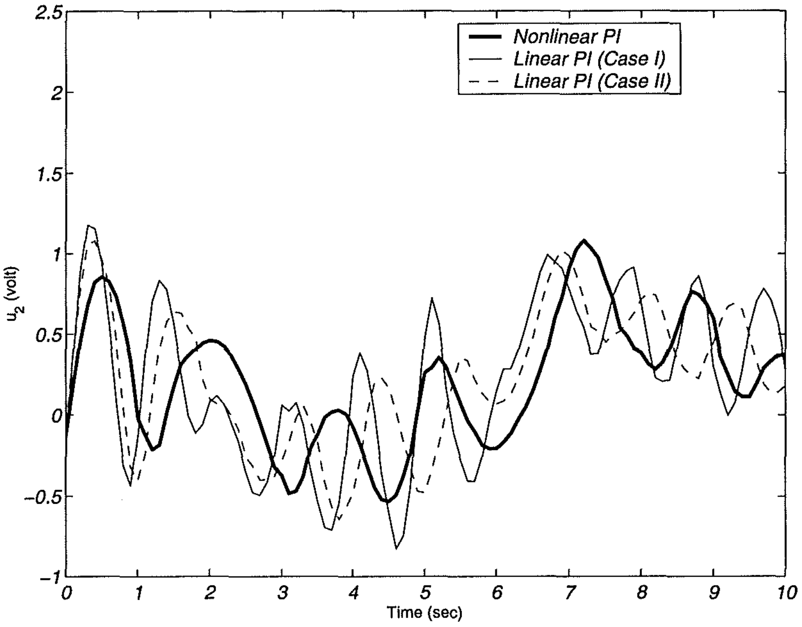


Fig. 9.36: Control signal of the second input link (PI control).

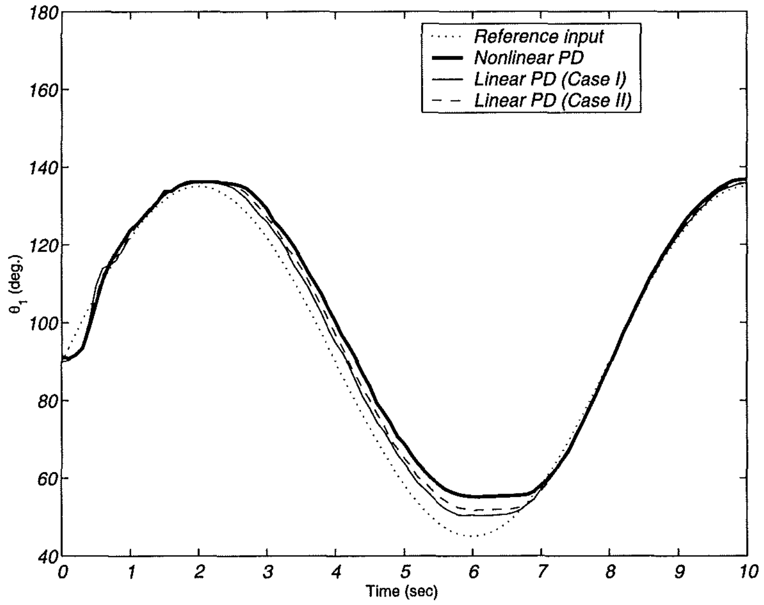


Fig. 9.37: Angular position of the first input link (PD control).

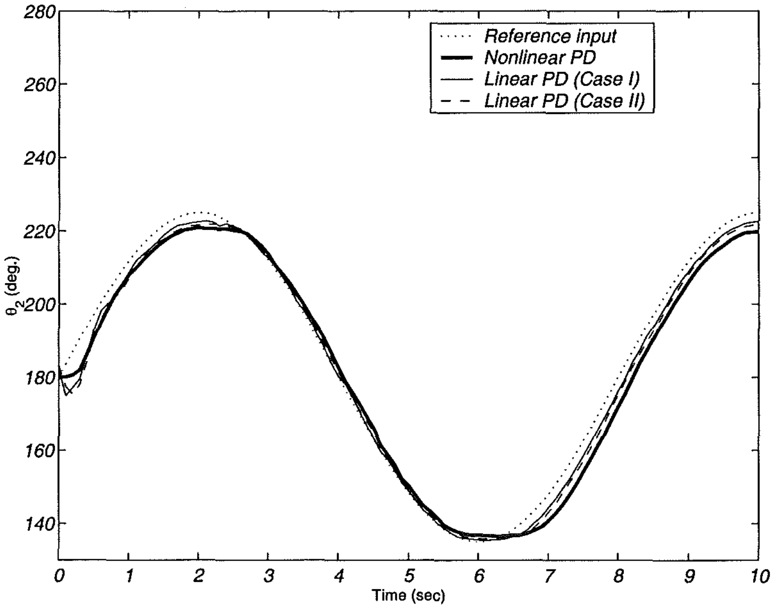


Fig. 9.38: Angular position of the second input link (PD control).

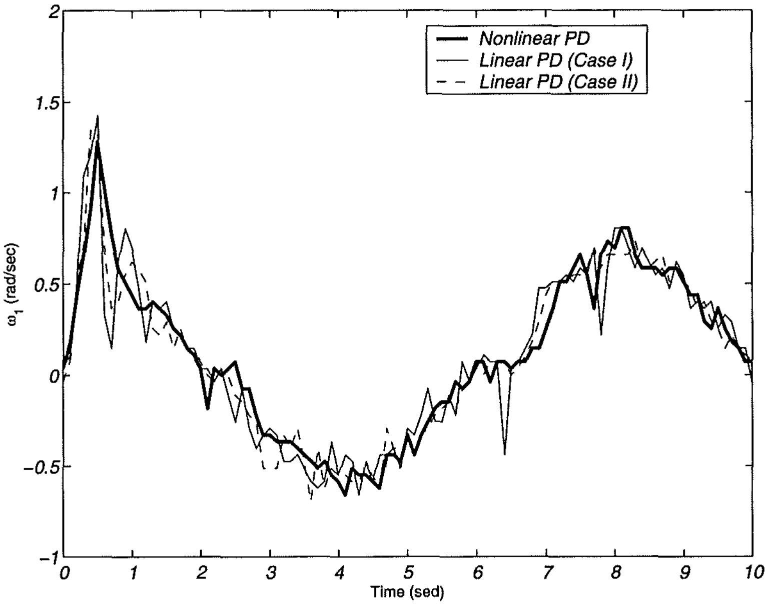


Fig. 9.39: Angular speed of the first input link (PD control).

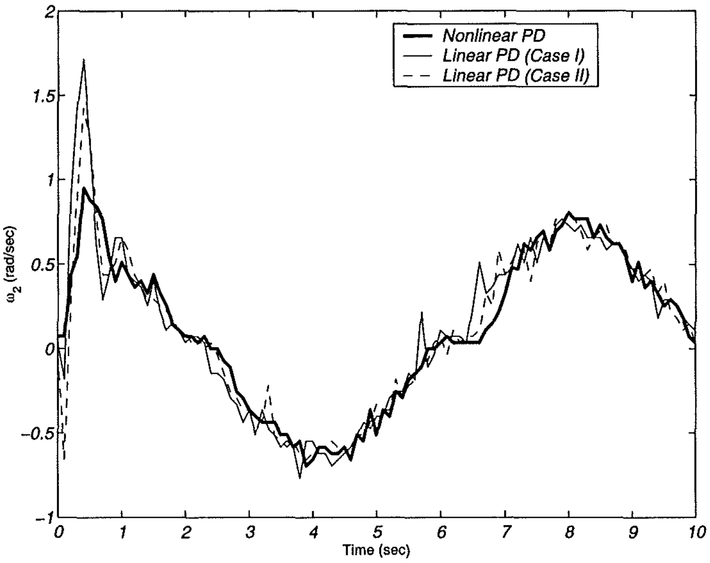


Fig. 9.40: Angular speed of the second input link (PD control).

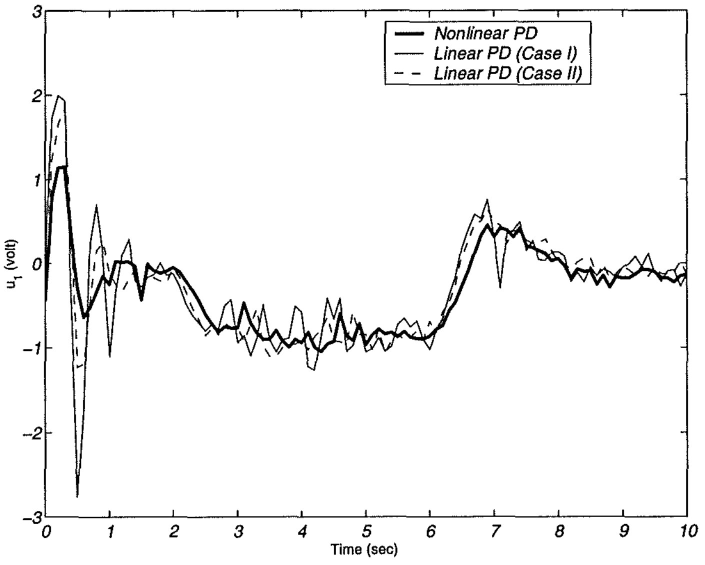


Fig. 9.41: Control signal of the first input link (PD control).

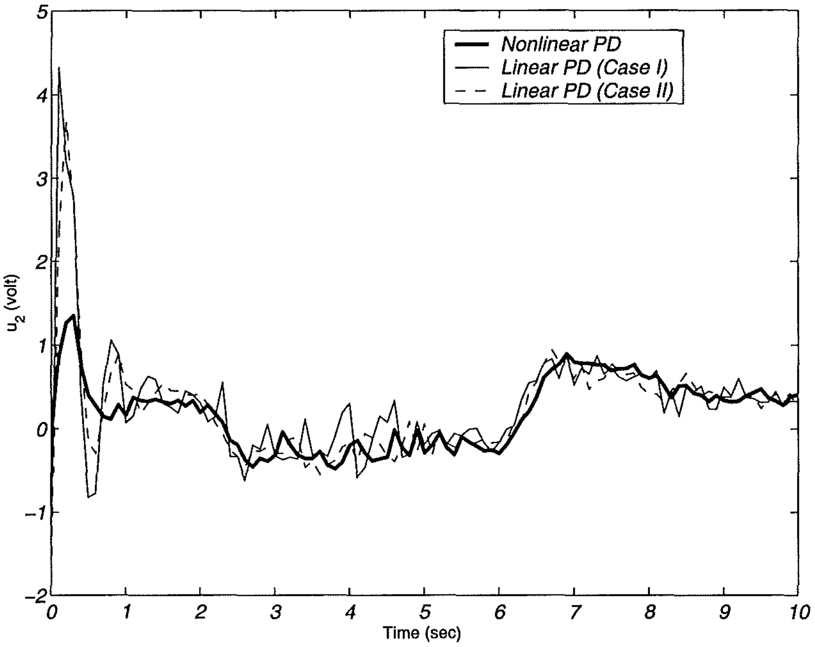


Fig. 9.42: Control signal of the second input link (PD control).

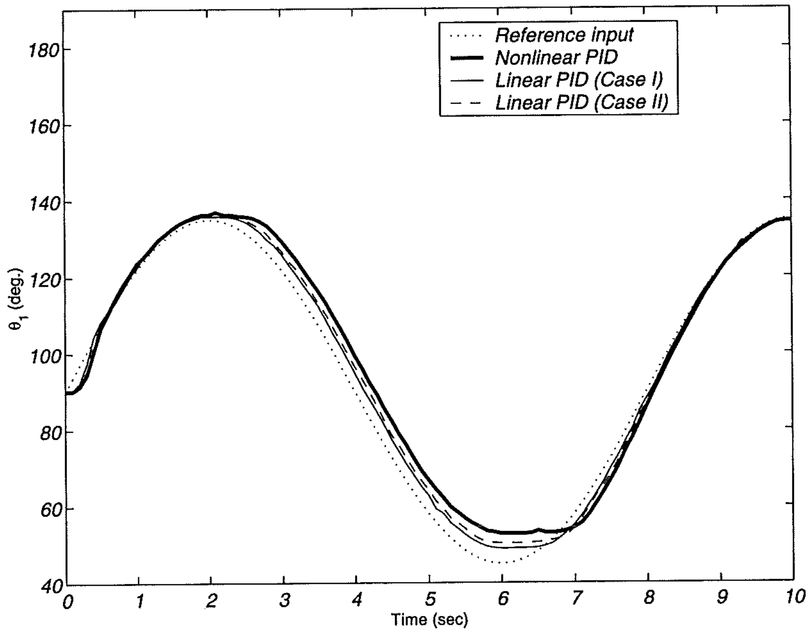


Fig. 9.43: Angular position of the second input link (PID control).

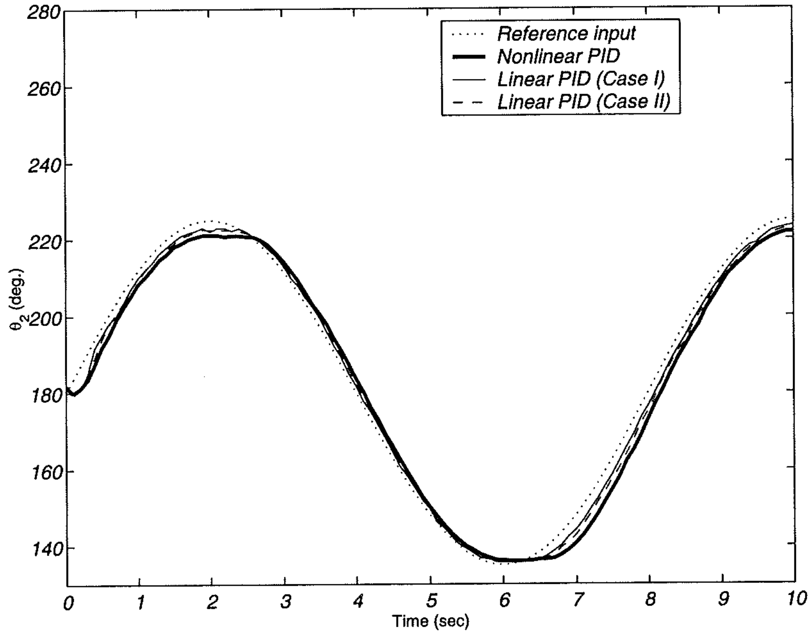


Fig. 9.44: Angular position of the second input link (PID control).

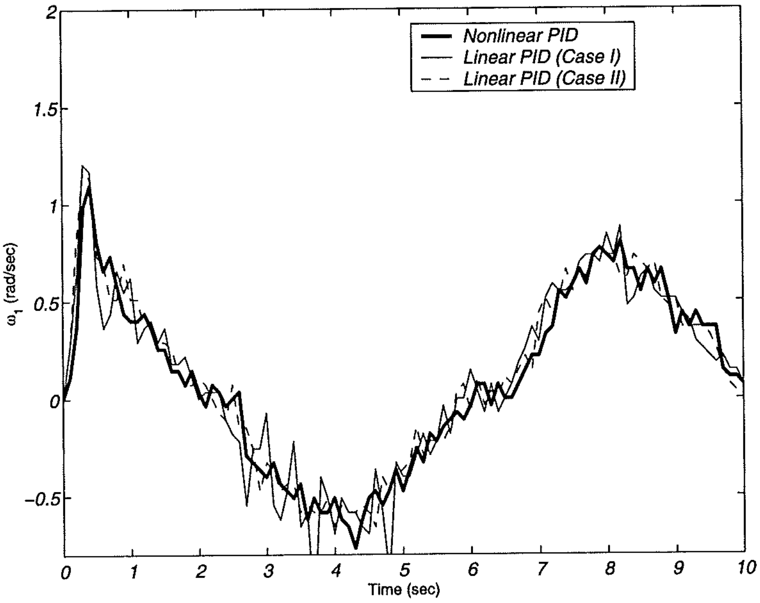


Fig. 9.45: Angular speed of the first input link (PID control).

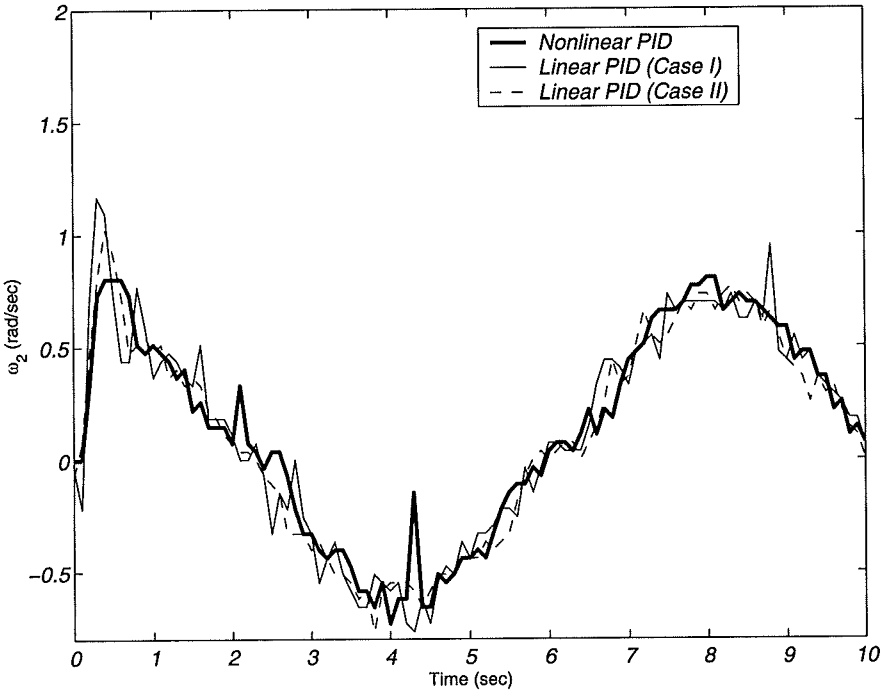


Fig. 9.46: Angular speed of the second input link (PID control).

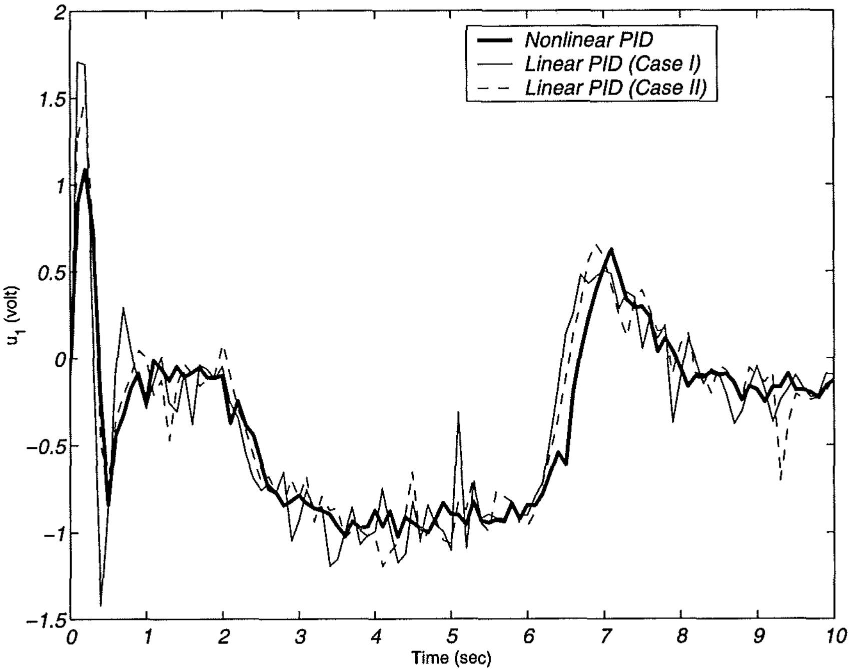


Fig. 9.47: Control signal of the first input link (PID control).

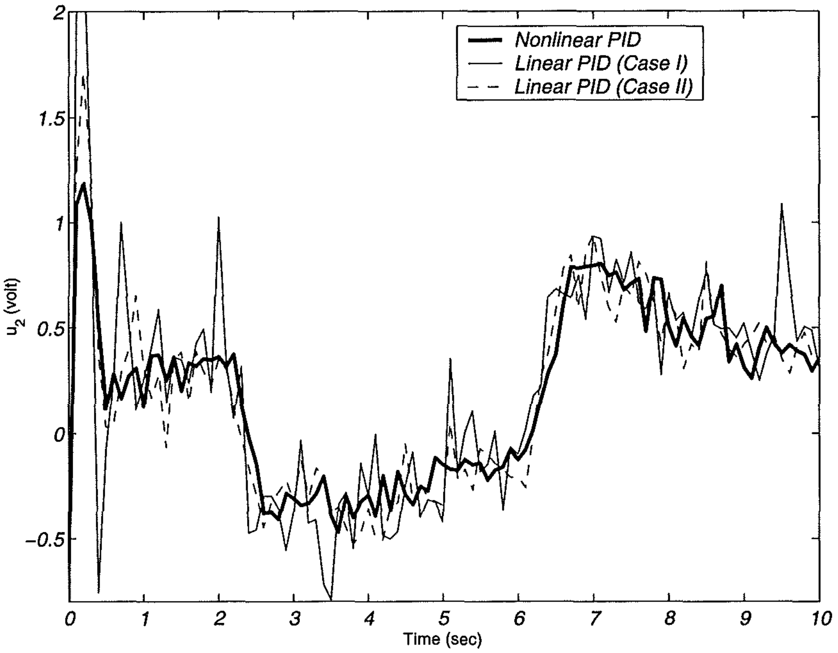


Fig. 9.48: Control signal of the second input link (PID control).

The idea proposed by Seraji [74-75] was to have a uniform dynamic response of the system with very little overshoot using the nonlinear gain. To examine the possibility of doing this the robot was subjected to two step-inputs of different magnitudes. The performance of the system under the nonlinear PD and PID controls is evaluated. This is for conciseness and also because they are the most practical controllers. The results (Figures 9.49–9.52) show that there is not any uniformity in the system's dynamic response. In addition, overshoot is always present and cannot be prevented. Generally, as expected, the proposed controller of Seraji is just one possible idea. There is not any theoretical technique behind it. It is not enough to have gains (that depend only on the error) of variable magnitudes in the control law to achieve the objectives.

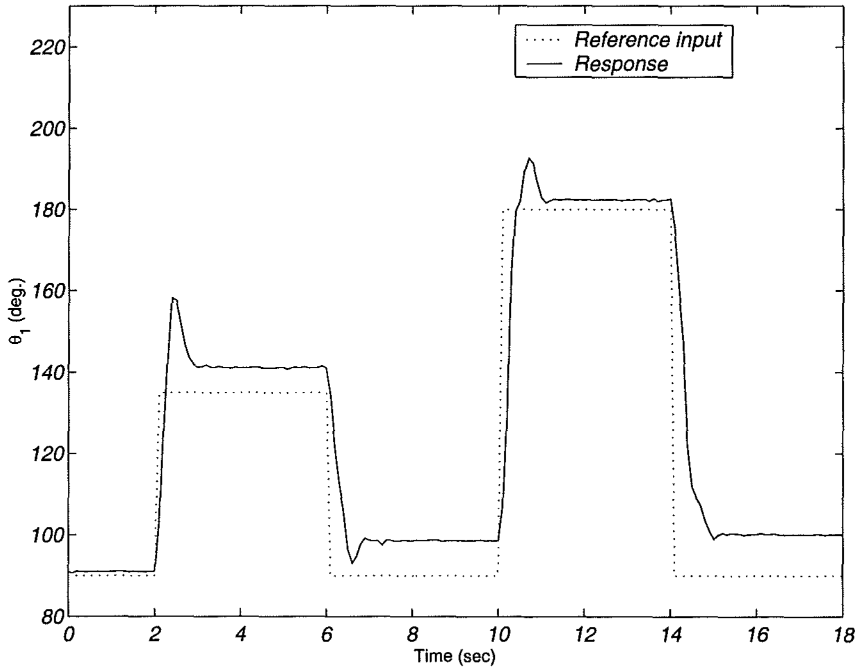


Fig. 9.49: Angular position of the first input link (PD control).

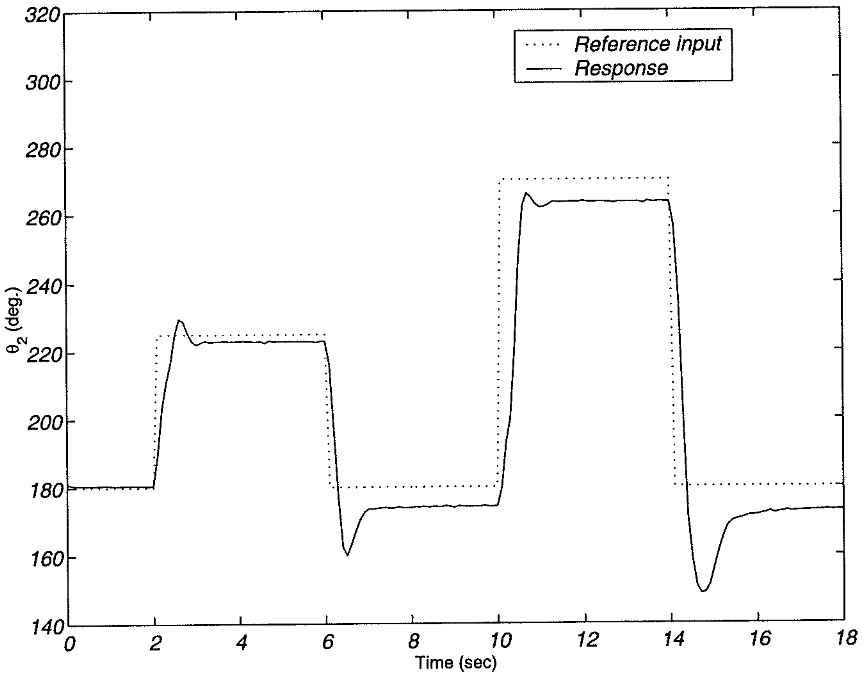


Fig. 9.50: Angular position of the second input link (PD control).

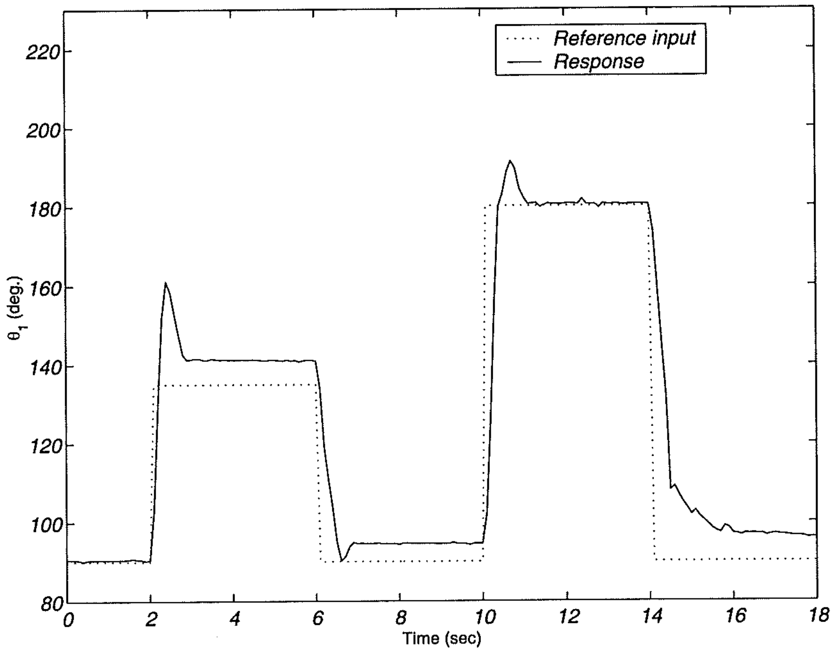


Fig. 9.51: Angular position of the first input link (PID control).

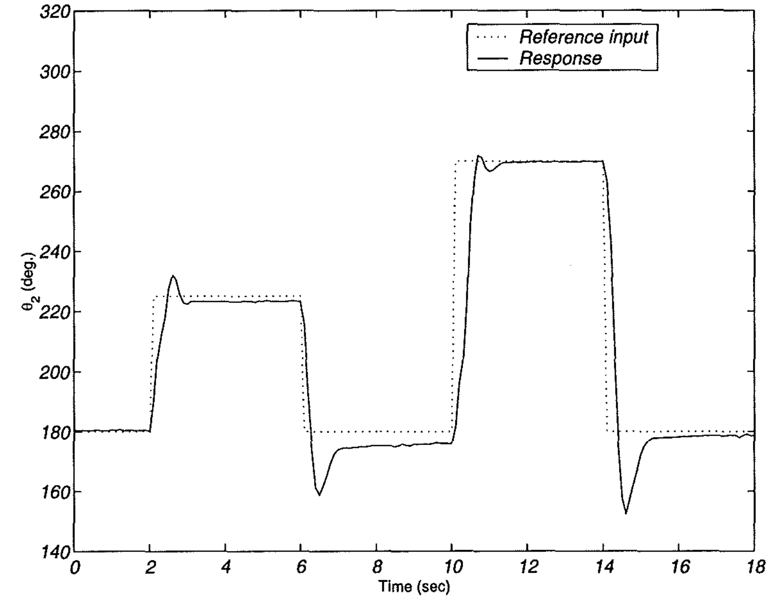


Fig. 9.52: Angular position of the second input link (PID control).

9.3 Design and Evaluation of a nonlinear PD controller with modulated P and D gains

For the i^{th} DOF, the proposed nonlinear PD control law is given by the following equation,

$$u_i = \left\{ k_{0i} + k_{1_i} |e_i| \right\} e_i - \left\{ b_{0_i} - b_{1_i} |e_i| \right\} \dot{\theta}_i \tag{9.3}$$

Here, k_0 represents the minimum proportional gain, while b_0 is the maximum derivative gain. The other gains are user defined positive constants that control the amount of variations in the P and D gains. At large errors the proportional gain is automatically amplified and the derivative gain is reduced. These actions allow for rapid rise times. As the error diminishes, the P gain is reduced to its minimum value while the D gain is increased to its maximum magnitude. This yields a maximum braking force near the goal in an attempt to prevent large overshoots.

Now, the task is to examine the performance of the robot under the application of the above control law. To do this, experiments have been carried out using the following practical gain values,

$$k_0 = 3, k_1 = 4, b_0 = 20 \text{ and } b_1 = 5$$

Because the controller is designed for set point control, in the experiments the system is only subjected to double step-inputs of different magnitudes. This is for the examination of the uniformity of the dynamic response of the system. It is necessary to refer here to the fact that we are still able to reach the same sampling rate as in the previous section (200 Hz).

9.3.1 Results

Under this specific input, the results in Figures 9.53–9.58 show an improvement in the system rise time at the cost of overshooting due to the saturation in the control effort. To alleviate this, the variations in the gains can be decreased, but this immediately leads to degradation in the system rise time. As seen in the results, the problem of the stiction- friction is still there. In addition there is not any uniformity in the response and for every step input there is a different performance.

9.4 Conclusions

In this chapter it has been shown that the idea of using nonlinear gain in cascade with the conventional linear control makes little impact on the system stability for some control formulations such as P and PI. The system under these nonlinear control laws has proven to be more stable than under the linear laws. For other types of control (PD and PID), the system performance is still the same as for the linear case. The non-linear PD controller with modulated P and D gains has improved the system rise time at the cost of saturating the control. Because of this saturation overshoots always exist. Any attempt to alleviate such overshoots will be at the cost of degrading the system speed of response. It is important to note that the two controllers discussed herein failed to compensate for input variations, and yield non-uniform responses.

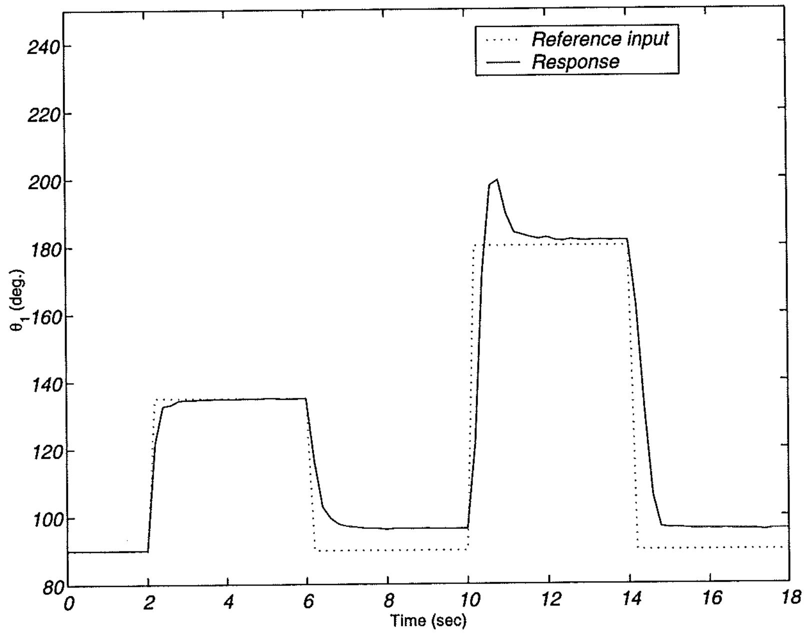


Fig. 9.53: Angular position of the first input link (Nonlinear PD control).

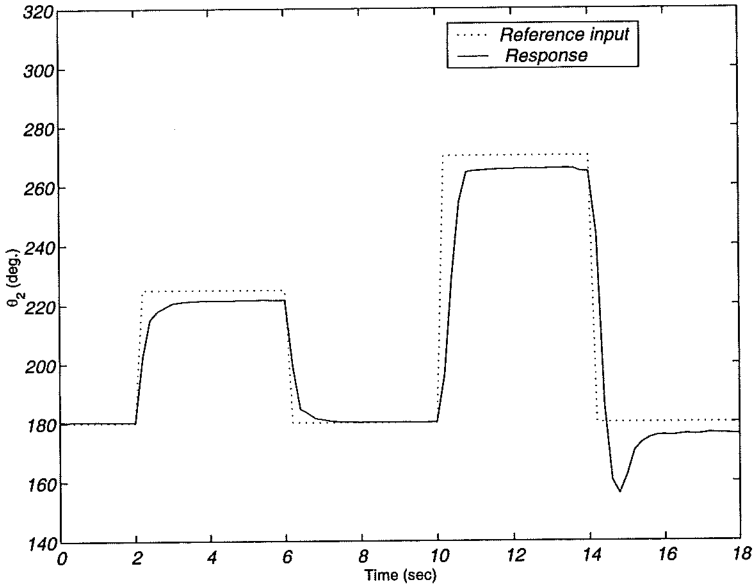


Fig. 9.54: Angular position of the second input link (Nonlinear PD control).

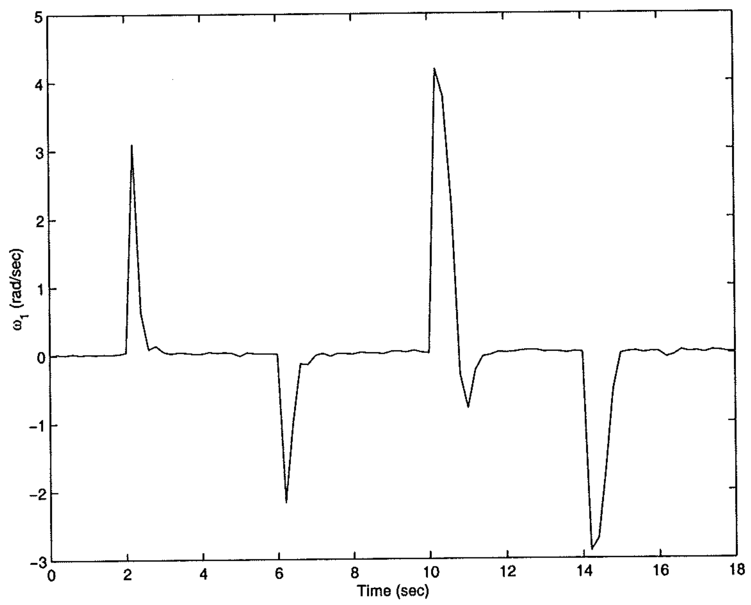


Fig. 9.55: Angular speed of the first input link (Nonlinear PD control).

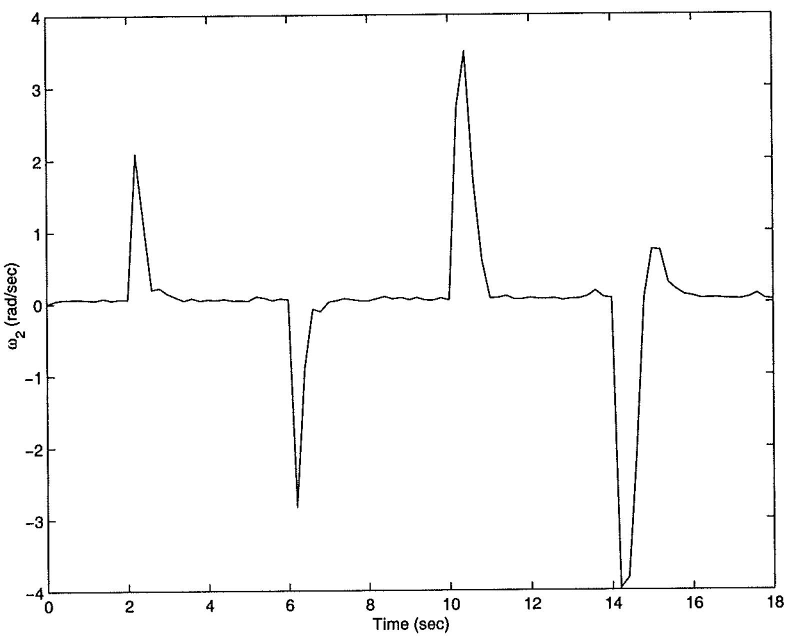


Fig. 9.56: Angular speed of the second input link (Nonlinear PD control).

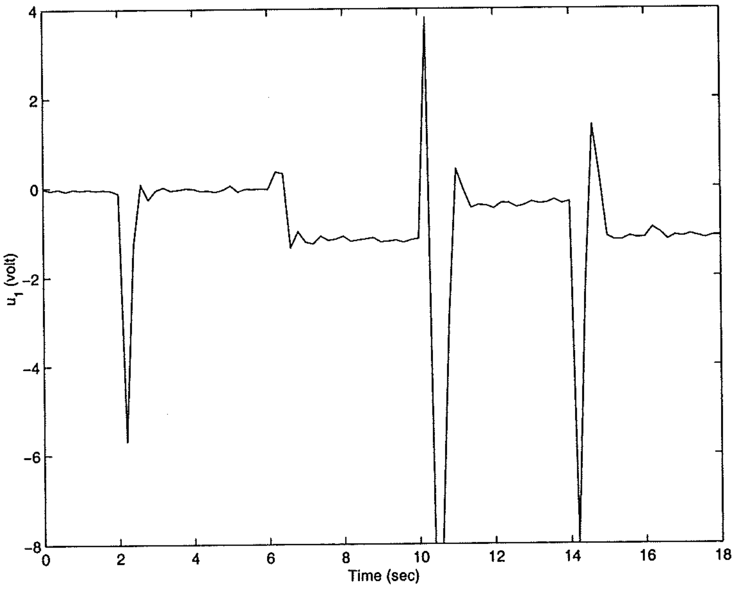


Fig. 9.57: Control signal of the first input link (Nonlinear PD control).

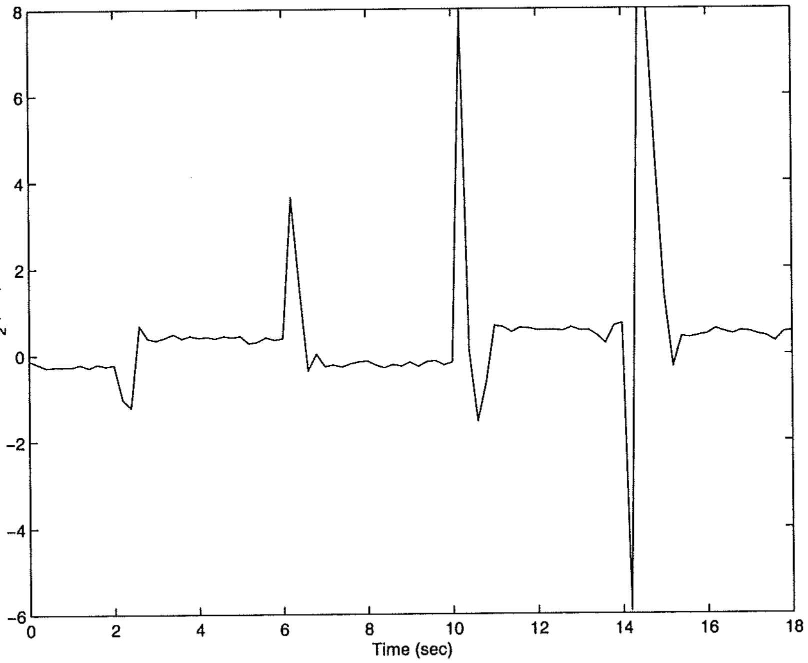


Fig. 9.58: Control signal of the second input link (Nonlinear PD control).

Chapter 10

On the design of high-performance PD Controllers: A New Methodology

10.1 Introduction

In the previous chapter it has been shown that the available linear and nonlinear PD or PID controllers are unable to critically damp the system response to disturbances for all robot configurations. Due to this, overshoot is always present and cannot be prevented unless the system is severely damped. From an economic point of view this solution can be strongly unacceptable. On the other hand overshooting may be permissible to some degree in certain systems but it is prohibited in the case of parallel robots as this may bring the robot arm to one of its singular configurations, thereby causing damage to the system. In this sense the work here introduces a new methodology for the design of PD controllers to ensure fast system response and no overshoot for all robot configurations, i.e., it gives uniform and fast dynamic performance of the system. In addition to this the method is also able to satisfy the necessary stability requirement of nonlinear systems. For the purposes of theoretical assessment simulations have been carried out on the SEPA robot to ensure its validation for all robot designs. For practical validation experiments have been carried out on the existing parallel robot.

10.2 Theoretical Background and Problem Statement

Under standard assumptions the dynamics of a rigid robotic manipulator with either prismatic or revolute joints can be described by the following equation [95],

$$\tau_c = M(\Theta)\ddot{\Theta} + V_m(\Theta,\dot{\Theta})\dot{\Theta} + F(\dot{\Theta}) + G(\Theta) \qquad (10.1)$$

Here, Θ is a vector of generalised co-ordinates, $M(\Theta)$is a symmetric, positive definite inertia matrix, $V_m(\Theta,\dot{\Theta})\dot{\Theta}$ is a vector of centrifugal and Coriolis forces, $G(\Theta)$ is a vector of gravity forces, $F(\dot{\Theta})$ is a vector of friction forces, and τ_c is a vector of control forces or torques.

With $F(\dot{\Theta}) = 0$, global asymptotic stability of the closed loop system is ensured by the following PD control [84] [93],

$$\tau_c = K_P E - K_V \dot{\Theta} + G(\Theta) \qquad (10.2)$$

Where K_P and K_V are constant, diagonal, gain matrices of dimensions that depend on the mechanical design of the robot manipulator. The vector E is the regulation error with respect to the reference input Θ_d which is constant here.

Proper choice of the elements of the gain matrices can critically damp the system response to a specific input. Once this input is changed the system exhibits either sluggish or oscillatory response depending on the input values. This represents a non-uniform dynamic response that is undesirable. Overshoot is the most critical issue that arises due to the non-uniform dynamic response. This problem must be avoided for robotic systems, especially in the case of parallel robots that have many singular configurations inside their workspace. The concern is always to avoid the state of singularity which can cause damage to the manipulator, as the control is lost.

To ensure uniform dynamic performance for robotic systems, the next section introduces a novel PD control design technique that guarantees this. In addition, the controller allows fast system responses with no overshoot. Besides, global asymptotic stability is guaranteed for robotic manipulators that are defined by equation 10.1.

10.3 Controller design

To prevent the robotic system defined by equation 10.1 (under the control of equation 10.2) from overshooting its desired input, the response of the i^{th} degree of freedom must be kept inside the second or the fourth quarter of the phase plane of Figure 10.1. However, this is not a strong condition for the prevention of overshoot. That condition is where the response of all degrees of freedom is maintained inside either of the two shaded areas in the second and fourth quadrants of Figure 10.1. The line which divides the second and the fourth quadrants is defined by the following equation,

$$\phi_i = \lambda_i e_i - \dot{\theta}_i = 0 \qquad (10.3)$$

where, λ_i is a gain representing the slope of the line.

After specifying the above line, and because we seek the response to be in the aforementioned shaded areas, the distance from the error axis to the line can be used as a parameter to control the value of the derivative gain. To implement this, the control law of equation 10.2 will take the following nonlinear form,

$$\tau_c = K_P E - \Omega(E,\dot{\Theta})K_V\dot{\Theta} + G(\Theta) \qquad (10.4)$$

where, $\Omega(E,\dot{\Theta}) = diag[1+\exp(-k_1\psi_1), 1+\exp(-k_2\psi_2), \ldots\ldots, 1+\exp(-k_n\psi_n)]$

and, $\psi_i = \lambda_i|e_i| - |\dot{\theta}_i|$.

Here, *diag* means diagonal, k_i is a user-defined, constant, positive gain and n is the number of degrees of freedom of the robot.

In the control law stated above the elements of K_V should be chosen to be small so as to represent the smallest derivative gain. The motivation for using this controller can now be discussed qualitatively. When the system is far away from the line defined by equation 10.3 the exponential term is reduced drastically causing the derivative term in the control law to be at its lowest value. This ensures rapid response in the early stages. As time passes the system reaches the specified line and the exponential term becomes equal to one, resulting in an increase in the value of the damping term. When the system overshoots the line the damping is increased by a very large amount causing the system to converge to the line. It is important to refer here to the fact that, despite this potential overshoot, we are still inside the desirable areas that are defined by the second and the fourth quadrants of the phase plane. This process is repeatedly applied to any trajectory on the phase plane. This inevitably results in the desired uniform dynamic performance of the system.

The next section examines the stability of the system defined by equation 10.1 under the control law of equation 10.4.

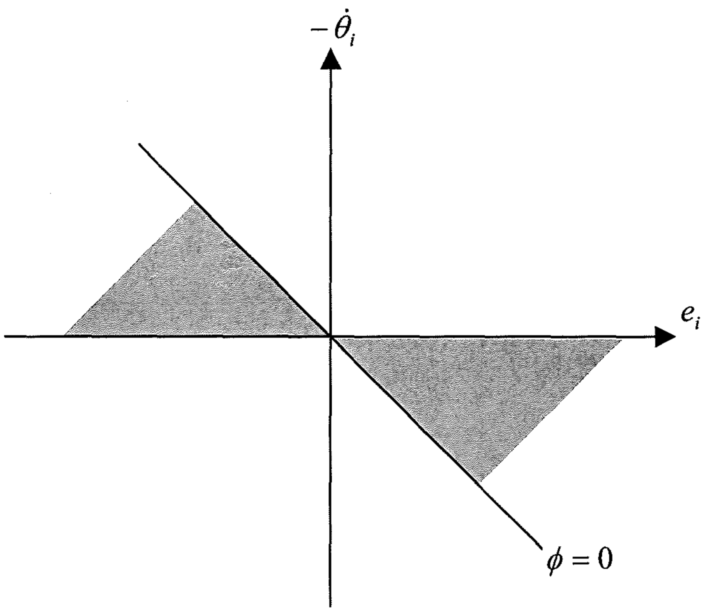


Fig. 10.1: Phase plane for the i^{th} DOF.

10.4 Stability analysis

Assuming only viscous friction, equation 10.1 becomes as follows,

$$\tau_c = M(\Theta)\ddot{\Theta} + V_m(\Theta,\dot{\Theta})\dot{\Theta} + B\dot{\Theta} + G(\Theta) \qquad (10.5)$$

where, B is a diagonal matrix with positive definite gains representing the viscous friction coefficients.

Theorem

The closed-loop system defined by equations (10.5) and (10.4) is globally, asymptotic, stable with respect to E and $\dot{E}$:

$$E, \dot{E} \to 0 \quad \text{as} \quad t \to \infty$$

Proof:

Consider the following scalar Lyapunov function,

$$V = \frac{1}{2}\dot{\Theta}^T M(\Theta)\dot{\Theta} + \frac{1}{2}E^T K_P E \qquad (10.6)$$

Differentiating with respect to time yields,

$$\dot{V} = \frac{1}{2}\dot{\Theta}^T \dot{M}(\Theta)\dot{\Theta} + \dot{\Theta}^T M(\Theta)\ddot{\Theta} + E^T K_P \dot{E} \qquad (10.7)$$

Substituting from equations 10.4 and 10.5 into equation 10.7 gives,

$$\dot{V}_L = \frac{1}{2}\dot{\Theta}^T \dot{M}(\Theta)\dot{\Theta} + \dot{\Theta}^T \left\{K_P E - \Omega(E,\dot{\Theta})K_V\dot{\Theta} - V_m(\Theta,\dot{\Theta})\dot{\Theta} - B\,\dot{\Theta}\right\} - E^T K_P \dot{\Theta}$$

Due to the skew symmetry of $\left(\frac{1}{2}\dot{M}(\Theta) - V_m(\Theta,\dot{\Theta})\right)$ [95], [84] and $\dot{\Theta} = -\dot{E}$ for set-point control, the time derivative of the Lyapunov function is reduced to the following equation,

$$\dot{V} = -\dot{\Theta}^T \Omega(E,\dot{\Theta}) K_V \, \dot{\Theta} - \dot{\Theta}^T B \, \dot{\Theta} \qquad (10.8)$$

Since K_V,B and Ω are positive definite, $\dot{V}_L \leq 0$ is satisfied and the system is globally stable.

10.5 Simulations

To examine the uniformity in the dynamic responses of all robotic systems under the application of this control law, simulations have been carried out using the SEPA robot. The manipulator parameters are the same as those of chapter 8. The arm dynamics are defined by equations 8.41, 8.42 and 8.43.

The design parameters of the controller (for all dynamic modes) are as follows,

$$K_p = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}, \; K_V = \begin{bmatrix} 4.5 & 0 \\ 0 & 4.5 \end{bmatrix}, \; \lambda_1 = \lambda_2 = 2, \text{ and } \; k_1 = k_2 = 2$$

To examine the effectiveness of the new control law in rejecting all disturbances, the arm has been subjected to a set of step inputs of different magnitudes. It is important to refer here to the fact that the motor torques in the serial and redundant modes are calculated in the same way as defined in chapter 5 and chapter 8. In addition, because friction only increases the stability of the system it is ignored in the simulation. The simulations have been performed using *Simulink*.

10.5.1 Simulation Results

The simulation results, in the form of system responses and motors control signals, for the *normal parallel mode*, the *redundant parallel mode* and the *serial mode* are shown in Figures 10.2-10.5, 10.6-10.10 and 10.11-10.14, respectively. The results show that the new controller is able to reject all the disturbances with the same effectiveness. The phase planes for the various operating modes are shown in Figures

10.15-10.20, and explain the uniformity of the system responses to all step inputs. In addition the controller is able to achieve fast system responses with no overshoots. The fast response results from the modulation of the derivative gain as shown in Figure 10.21.

Despite the fact that fixed design parameters have been used for the controller in all the dynamic modes the controller has succeeded in dealing with the differences between the dynamic models. This helps to underpin the robustness properties of the PD controller with gravity compensation [93].

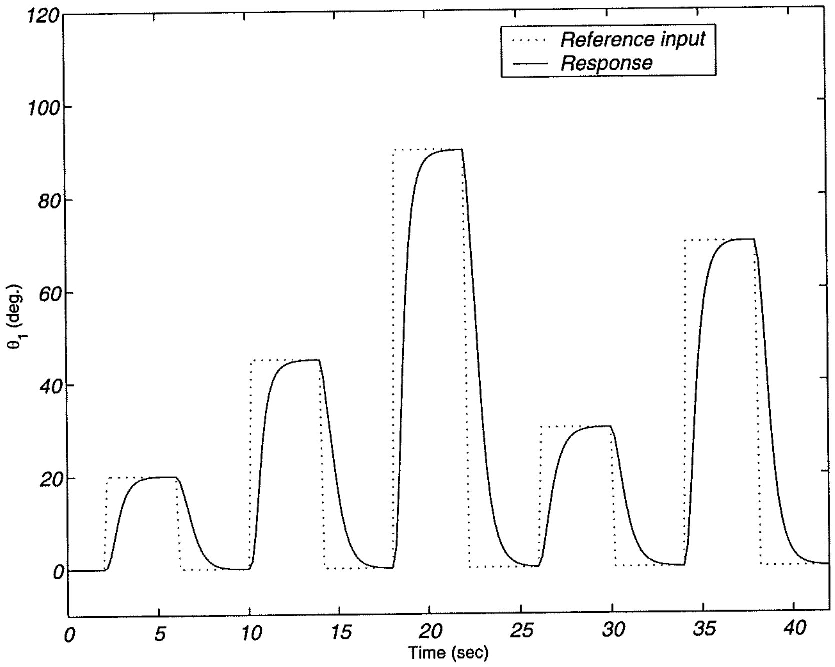


Fig. 10.2: Angular position of link BD (*Normal Parallel Mode*).

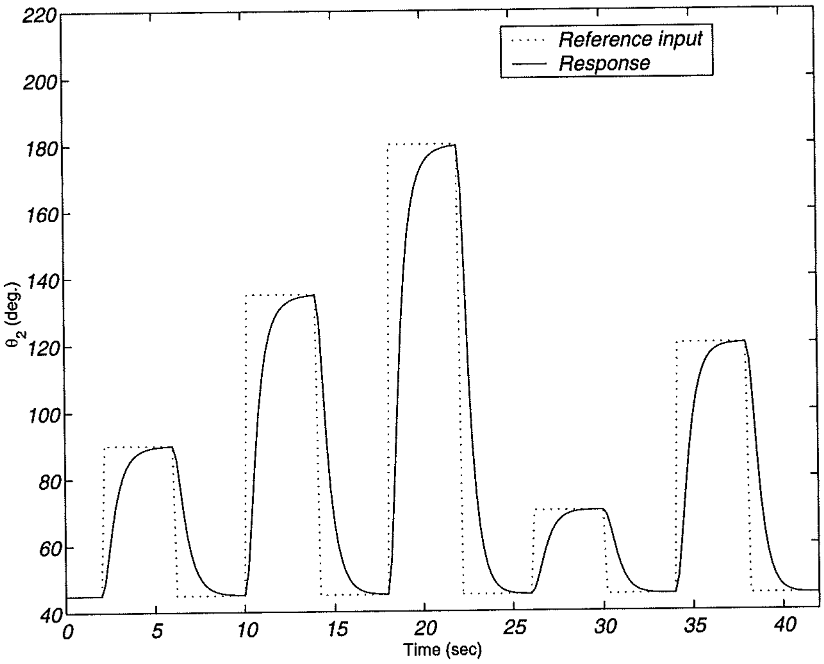


Fig. 10.3: Angular position of link AC (*Normal Parallel Mode*).

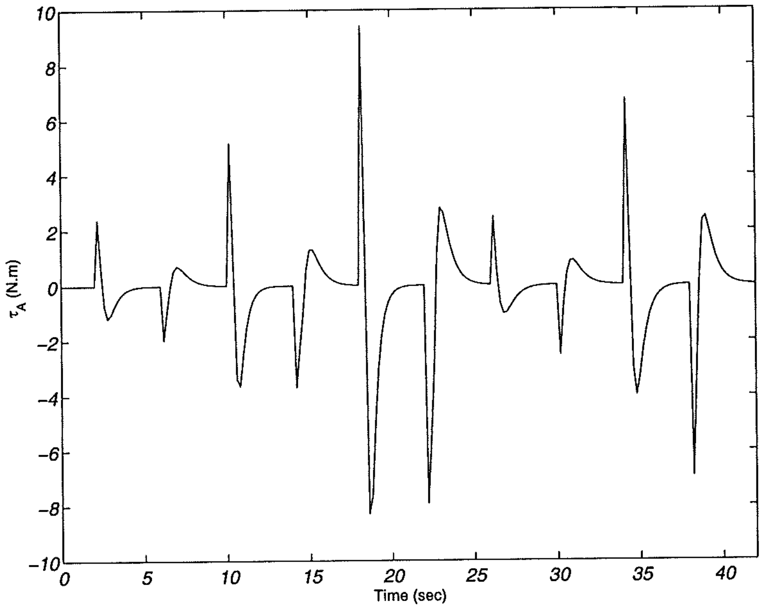


Fig. 10.4: Control signal of motor A (*Normal Parallel Mode*).

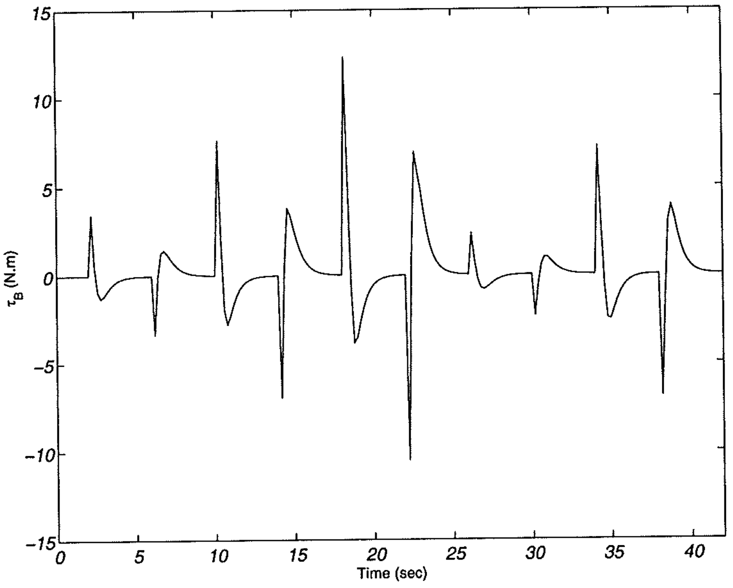


Fig. 10.5: Control signal of motor B (*Normal Parallel Mode*).

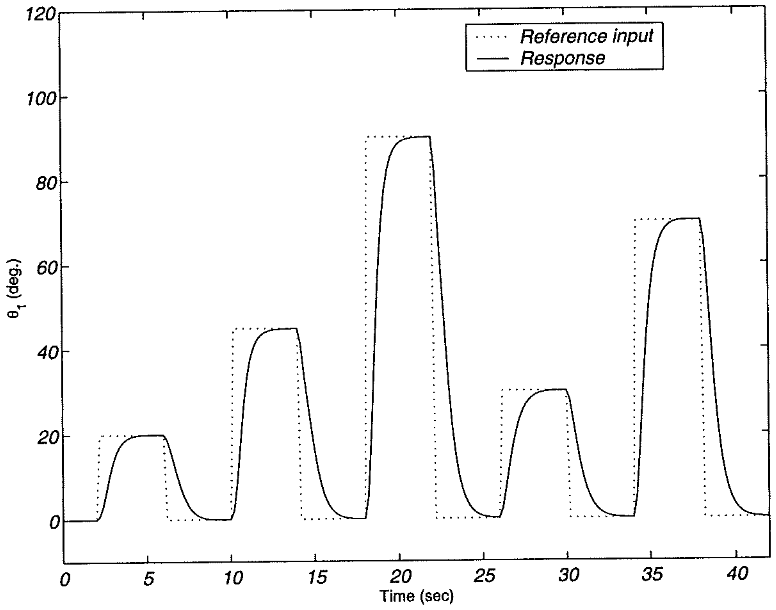


Fig. 10.6: Angular position of link BD (*Redundant Parallel Mode*).

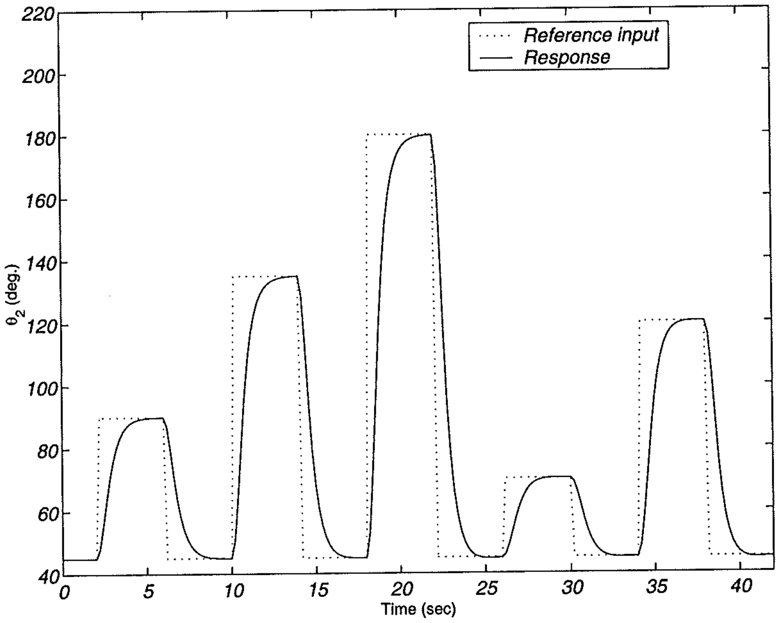


Fig. 10.7: Angular position of link AC (*Redundant Parallel Mode*).

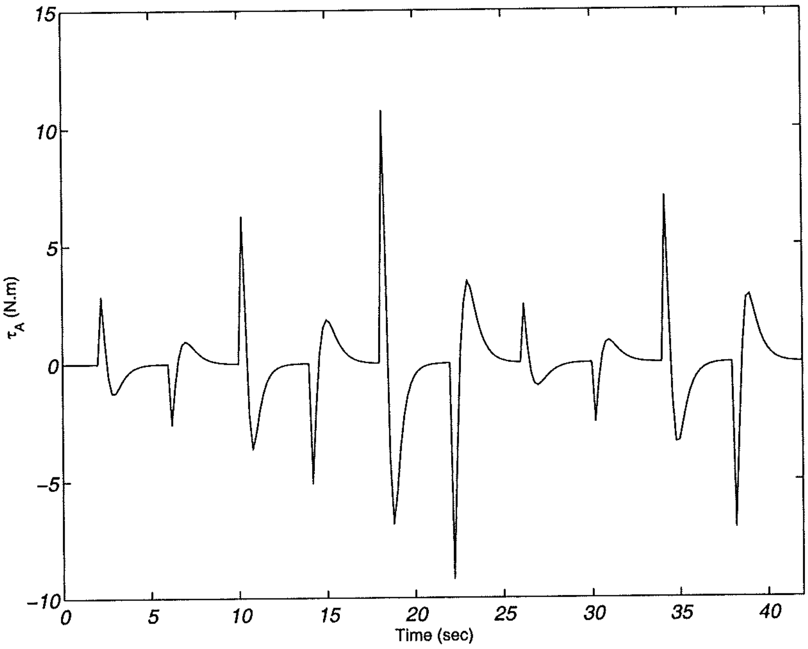


Fig. 10.8: Control signal of motor A (*Redundant Parallel Mode*).

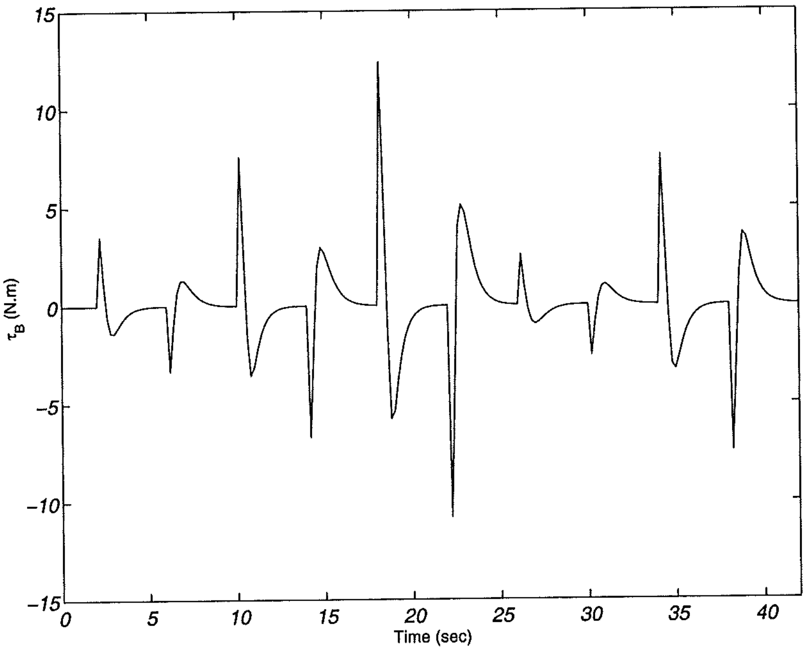


Fig. 10.9: Control signal of motor B (*Redundant Parallel Mode*).

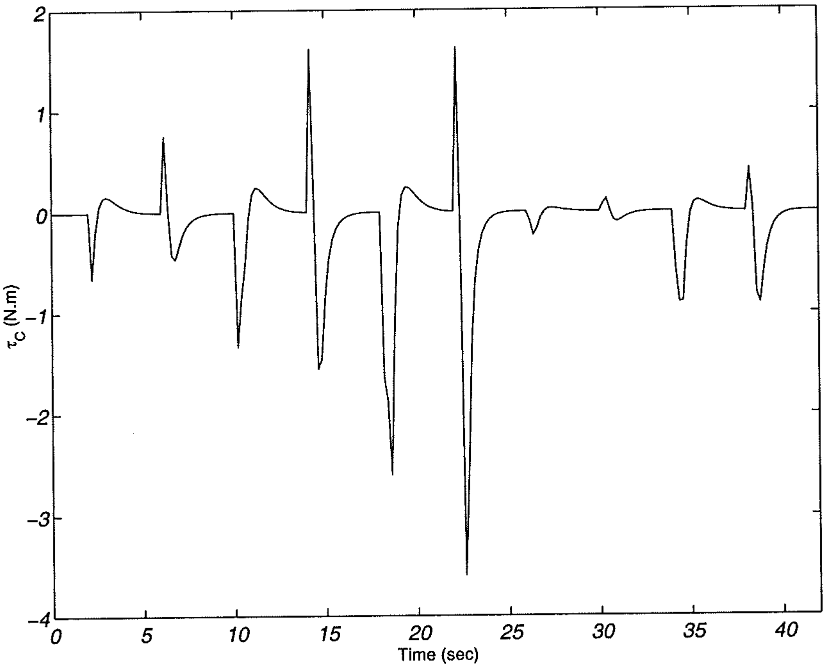


Fig. 10.10: Control signal of motor C (*Redundant Parallel Mode*).

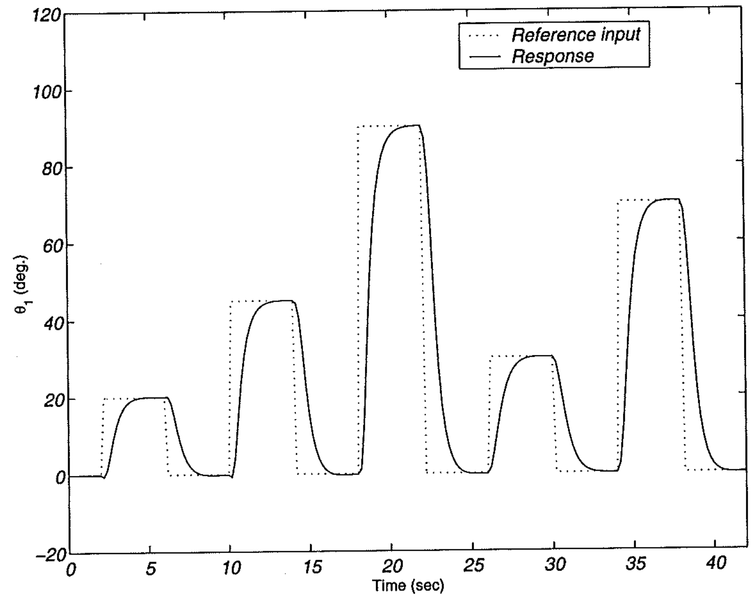


Fig. 10.11: Angular position of links BD and AC (*Serial Mode*).

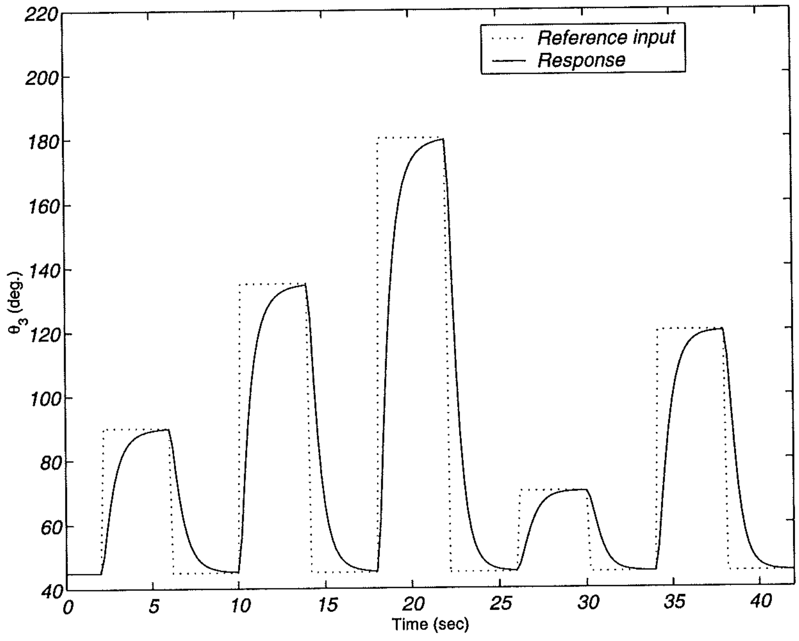


Fig. 10.12: Angular position of links DP and CP (*Serial Mode*).

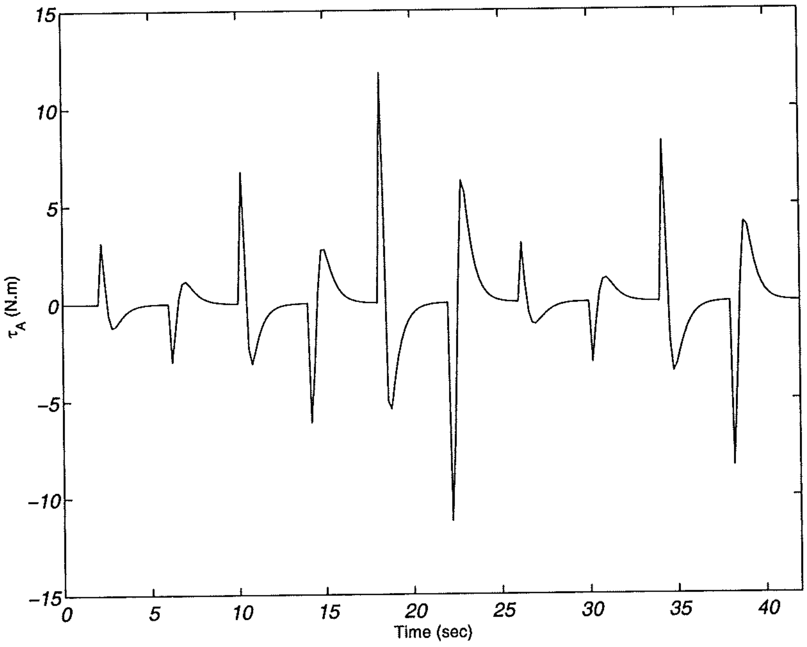


Fig. 10.13: Control signal of motor A and B (*Serial Mode*).

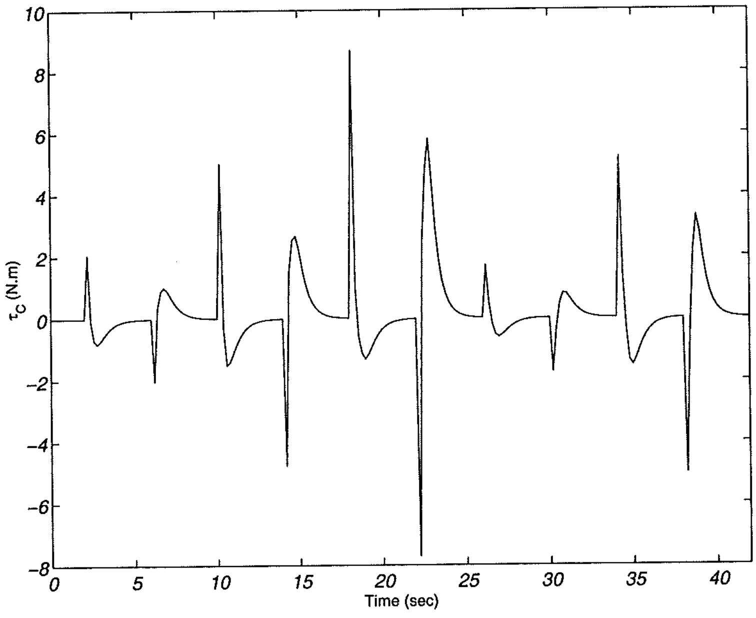


Fig. 10.14: Control signal of motor C (*Serial Mode*).

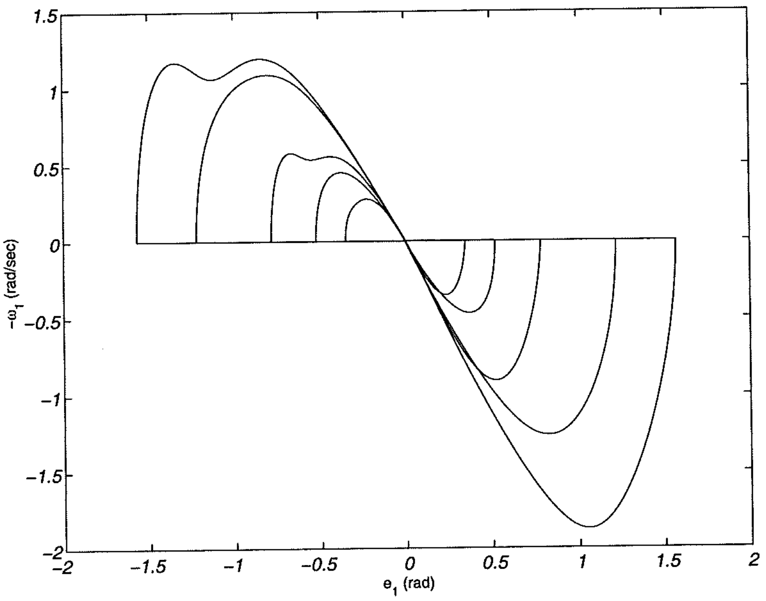


Fig. 10.15: Phase plane of link BD (*Normal Parallel Mode*).

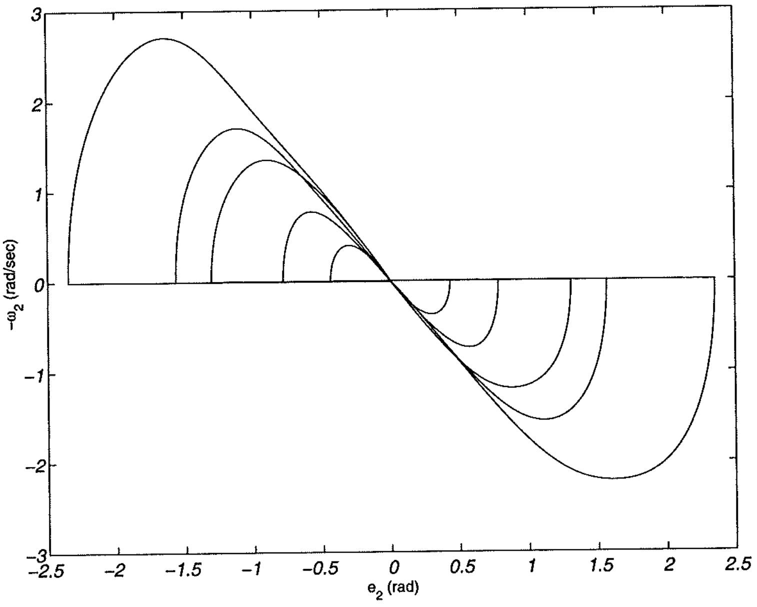


Fig. 10.16: Phase plane of link AC (*Normal Parallel Mode*).

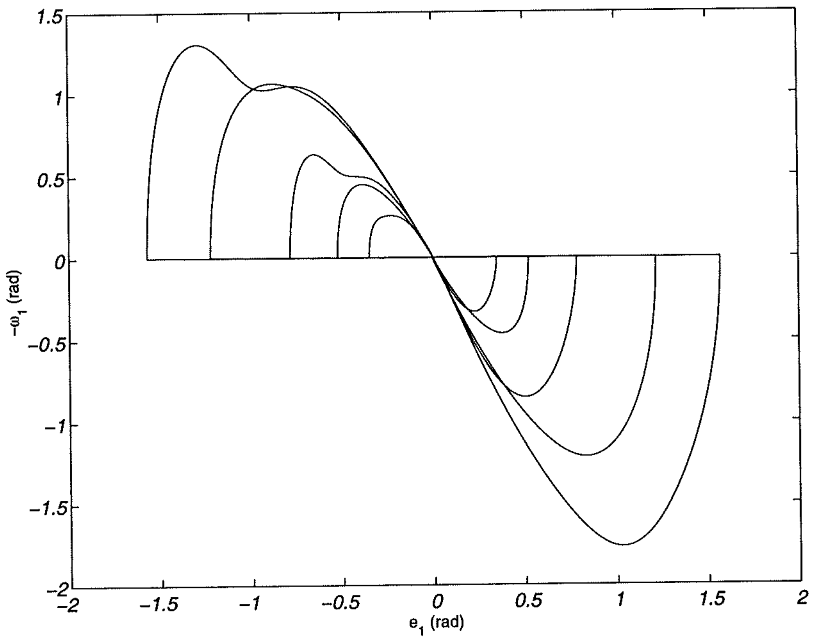


Fig. 10.17: Phase plane of link BD (*Redundant Parallel Mode*).

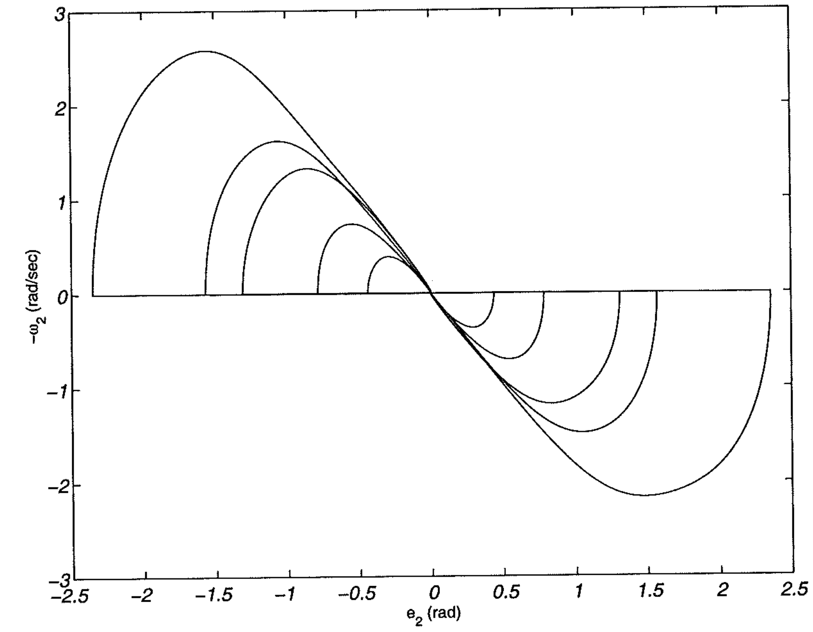


Fig. 10.18: Phase plane of link AC (*Redundant Parallel Mode*).

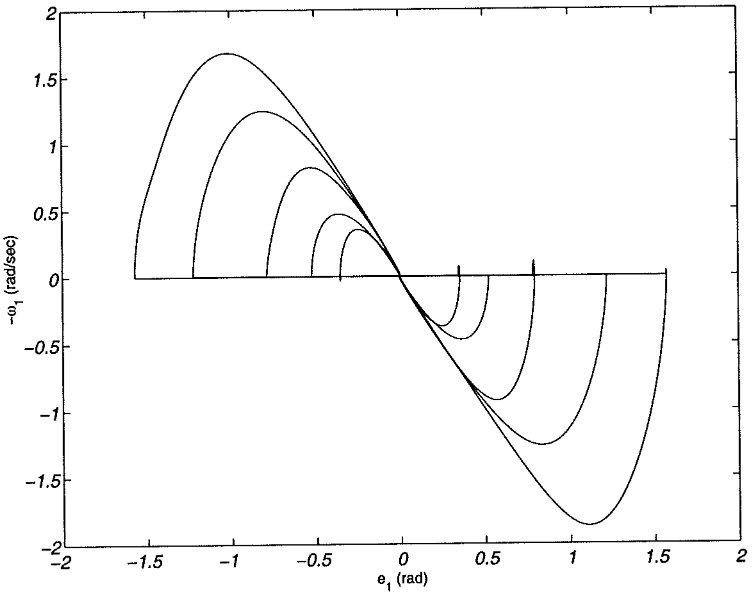


Fig. 10.19: Phase plane of links AC and BD (*Serial Mode*).

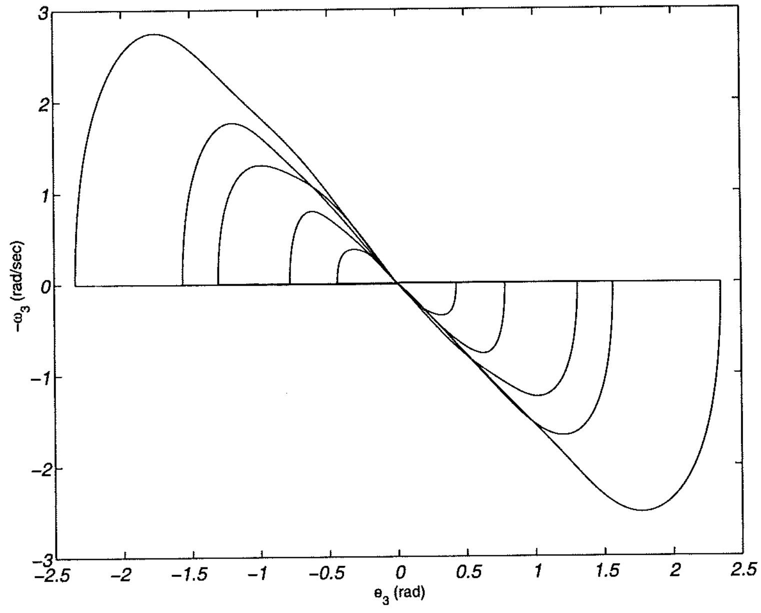


Fig. 10.20: Phase plane of links DP and CP (*Serial Mode*).

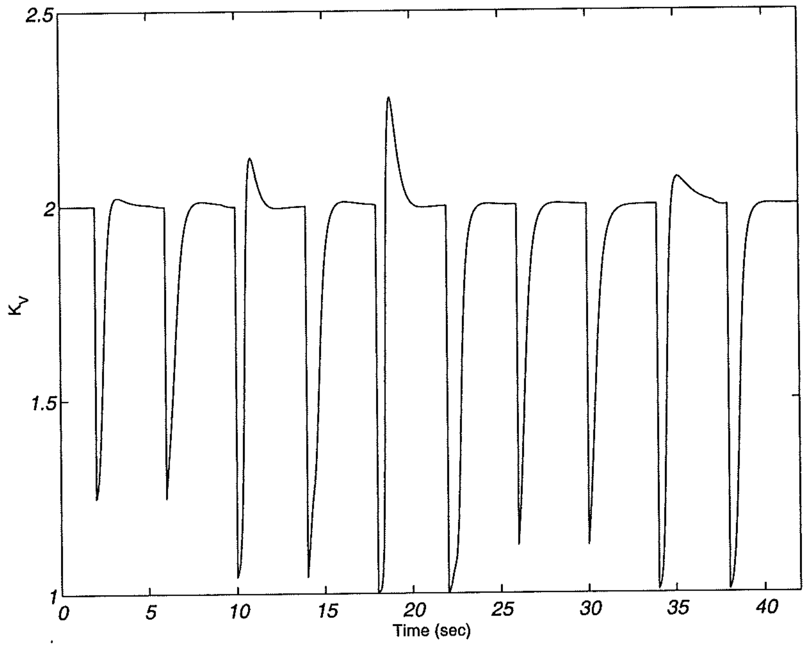


Fig. 10.21: The time-history of the derivative gain.

10.6 Experiments

In the experiments the design parameters of the controller are as follows,

$$K_P = \begin{bmatrix} 5 & 0 \\ 0 & 3 \end{bmatrix},\ K_V = \begin{bmatrix} 14 & 0 \\ 0 & 14 \end{bmatrix},\ \lambda_1 = \lambda_2 = 0.025 \text{ and } k_1 = k_2 = 40$$

The sampling rate is 200 Hz. The Simulink solver is the ODE 45 routine.

10.6.1 Experimental Results

The results of the experiments (Figures 10.22-10.27) display the same characteristics as the simulations. Due to the presence of dry friction the arm is unable to converge to its origin. However, this can be overcome by adding an integral term to the control law. Generally the results obtained are much better than the experimental results of chapter 8. The results of this chapter exhibit no overshoots, no control saturation, uniform system performance, and fast dynamic responses. In this sense the control method has achieved its goals.

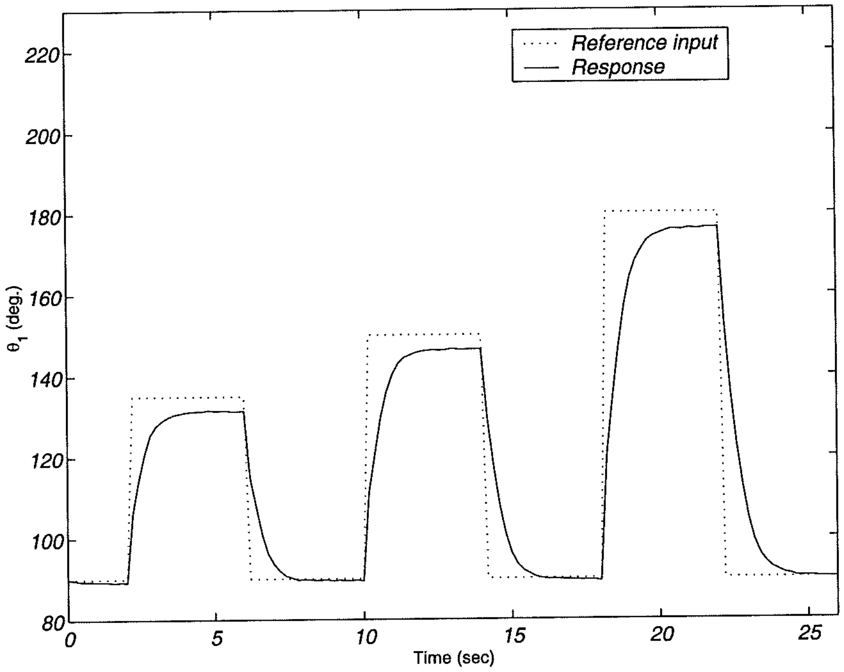


Fig. 10.22: Angular position of the first input link.

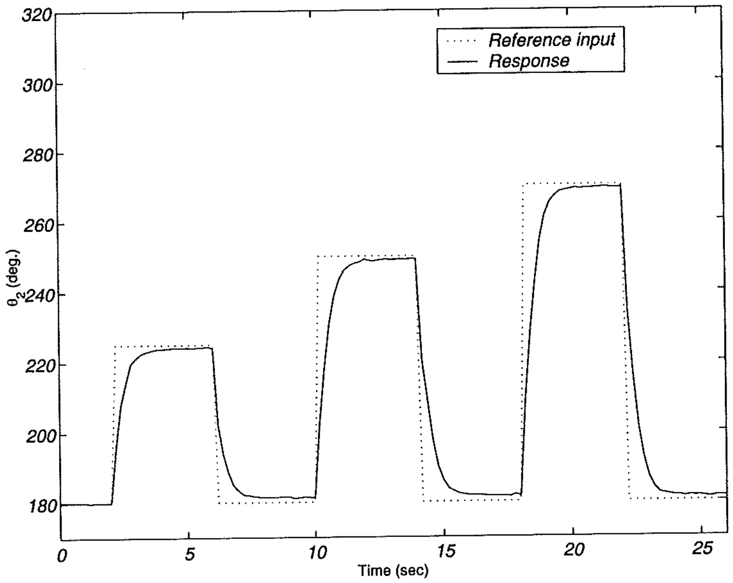


Fig. 10.23: Angular position of the second input link.

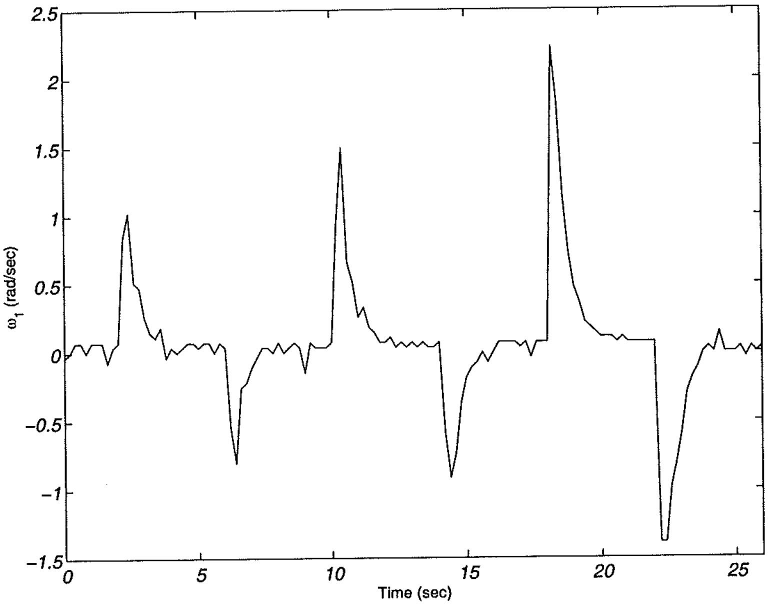


Fig. 10.24: Angular speed of the first input link.

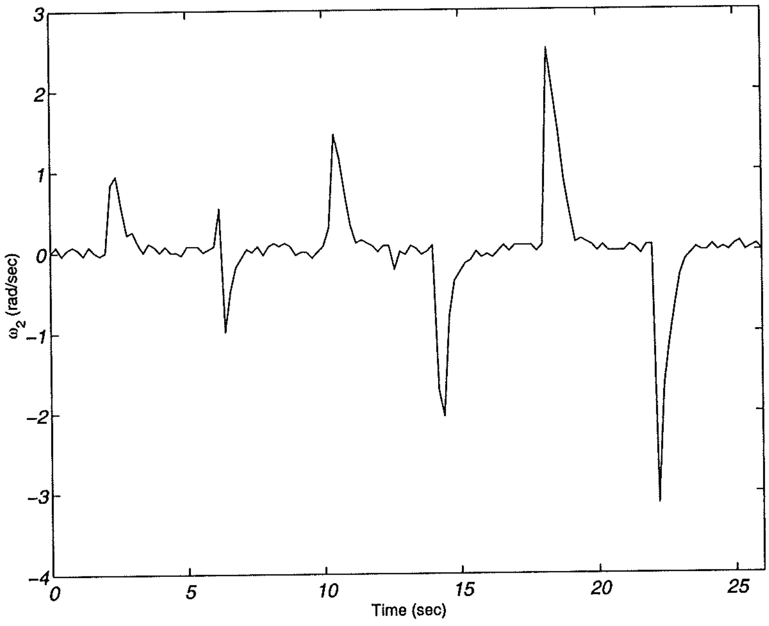


Fig. 10.25: Angular speed of the second input link.

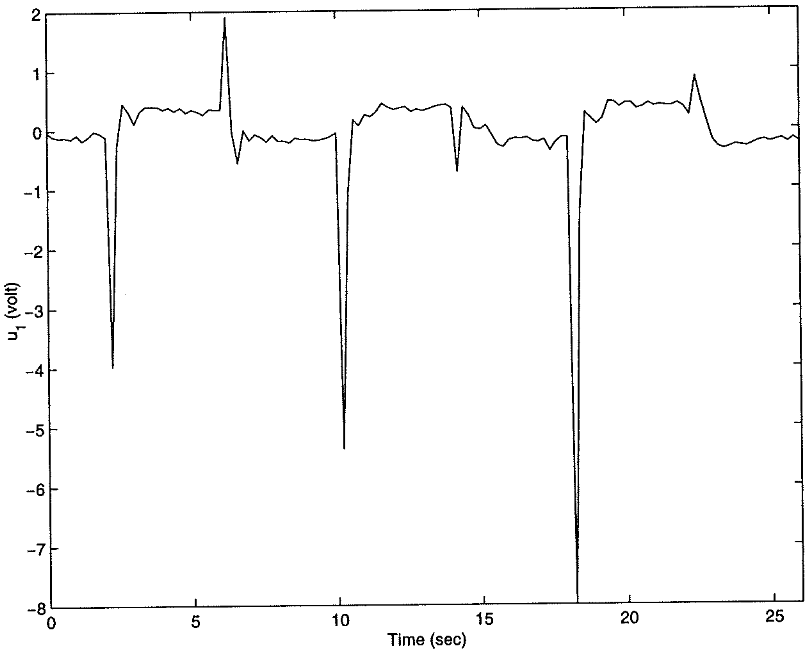


Fig. 10.26: Control signal of the first input link.

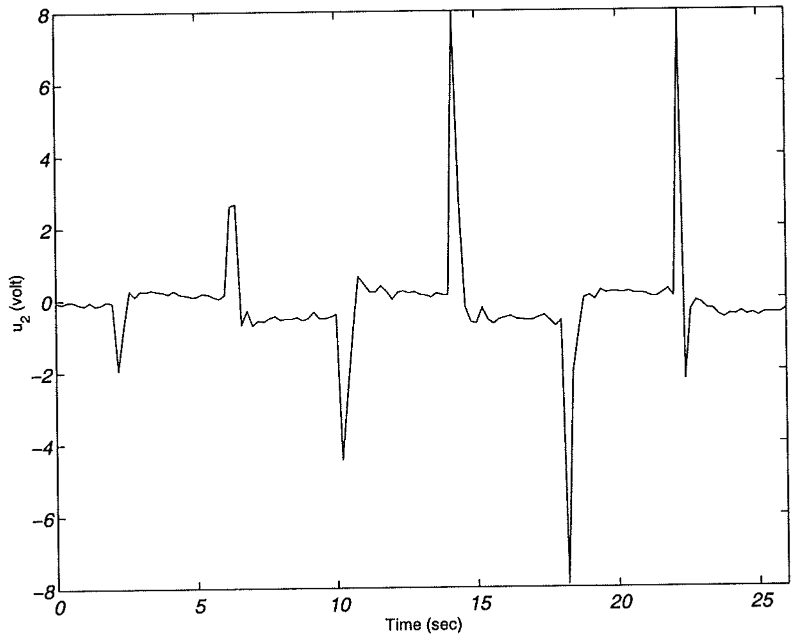


Fig. 10.27: Control signal of the second input link.

10.7 Conclusions

In this chapter a new technique has been introduced for the design of PD controllers. The design allows fast system responses without overshoots. In addition the system exhibits a uniform dynamic performance under the new control law. These facts have been theoretically and experimentally proven. The results obtained are valid for all robotic systems. Stability requirements have been fulfilled for the nonlinear system. Unlike other PD control strategies the new control law prevents any control saturation. As a result of these good results this new PD control law can be easily added to any nonlinear model-based control, as a servo controller, to enhance its performance.

Chapter 11

Conclusions

In order to improve the performance of planar parallel robots several investigations and developments have been carried out. The outcome of these is as follows,

i. In chapter 2 it has been shown that the *Model Transformations Tools* (MTT) software is able to model parallel robots. In the standard bond graph, once the constraints are solved, the modelling process is straightforward and no further manipulation of the generated system equations is necessary. On the other hand, despite the reduction of the time and the effort during the construction of the system bond graph by use of the hierarchical tool, this approach has been proven to be not as straightforward. To alleviate this a new method has been developed for the hierarchical modelling of parallel robots.

ii. In chapter 3 the concept of multi-cell planar parallel manipulators has been introduced. The concept allows for an increase in the workspace without affecting the original system's structure. By using the Lagrange formulation a general dynamic model for a manipulator applying this technique with any orientation has been derived. After that a new method of static balancing has been introduced and which led to the reduction of the number of variables, to be adjusted, to only one parameter.

iii. In chapter 4 the redundant one-cell, 2-DOF, parallel planar manipulator has been modelled using the bond-graph method by using the Model Transformation Tools software (MTT). The model simulates the arm while carrying more than one redundant manipulator, and hence it is a general model. The effect of the location of the redundant motor on the performance of the manipulator has also been analysed. The conclusion of the analysis recommends inserting the redundant motor as near to the base as possible if the manipulator is designed only for the normal pick and place dynamic tasks and the payloads are small. If the manipulator is designed for applications that need contact with the environment such as machining operations then the position of the redundant motor is not important and it can be put in any suitable location.

iv. In chapter 5 the new concept of singularity-based design has been presented and used to develop a manipulator with multiple operating modes. For each mode a dynamic model has been derived using the Lagrange method. The possible conditions by which motor torques can be selected have also been defined. In addition to this, a performance analysis has been carried out for each operating mode by using simulations. The results indicate improvements in the serial mode from the viewpoint of load capacity. After the analysis the idea has been generalised to include the design of new planar mechanisms called parallel-parallel manipulators.

v. In chapter 6 the experimental system has been discussed in detail. Following this the theoretical dynamic model has been modified in order take into account features that are in the experimental system and not in the mathematical dynamic equations. In addition to this a simplified friction model has been added to the equations to improve the model. The quality of the derived dynamic equations has been verified by comparing data collected from the experimental system with data from simulations. The results of this comparison show little difference between the actual and the theoretical model. The results are acceptable as the dynamic model is not always perfect.

vi. In chapter 7 the two inverse model-based control laws (Feed-forward and Computed torque) have been applied to the experimental parallel robot. The results show that both methods are good at controlling the robot, but the computed-torque controller has been shown to be much better than the feed-forward controller. In both methods the tracking accuracy is very good for slow motion. On the other hand despite that there is no friction model incorporated in the dynamic model used in both of the above control methods, success was apparent in compensating for friction due to the integral action in the servo control part. The best compensation is reported for the case of the computed-torque control. In the disturbance rejection test the computed torque controller has been shown to be better than the feed-forward controller.

vii. In chapter 8 it has been proven that the computed torque controllers are sliding mode controllers (SMC) but with unsuitable reaching dynamics. On the other hand it has also been proven that chattering elimination in the conventional SMC through the boundary layer method leads to a normal computed torque control law. In addition, it has been shown that the use of perturbation estimations to reduce the degradation of the error due to the use of the boundary layer method leads to a computed torque controller with an integral term. Following this a new robust computed torque controller has been designed based on the reaching law method. Simulations have been carried out in order to examine the robustness of the controller against the uncertainties in the robot parameters and the unknown dynamics. The SEPA robot has been used to ensure the effectiveness of the controller to all mechanical designs. For practical examination experiments have been carried out on the existing parallel robot. It is found that the theoretical results agree very closely with those of the experiments.

viii. In chapter 9 it has been shown that the idea of using nonlinear gain in cascade with the conventional linear control makes little impact on the system stability for some control formulations such as P and PI. The system under these nonlinear control laws has proven to be more stable than under the linear laws. For other types of control (PD and PID), the system performance remains the same as for the linear case. The non-linear PD controller with modulated P and D gains has improved the system rise time at the cost of saturating the control. Due to this overshoots always exist. Any attempt to alleviate such overshoots will be at the cost of degrading the system speed of response. It is important to note that the two controllers discussed herein failed to compensate for input variations and yield non-uniform responses.

ix. In chapter 10 a new technique has been introduced for the design of PD controllers. The design allows fast system responses without overshoots. In addition, the system exhibits a uniform dynamic performance under the new control law. These facts have been theoretically and experimentally proven. The results obtained are valid for all robotic systems. Stability requirements have been fulfilled for the nonlinear system.

Future work

The following points are proposed for further investigations,

- Development of a multi-DOF SEPA robot.

Since robotic manipulation is done in three-dimensional space, a multi-DOF robot is very desirable in the sense that it can move its tool to both an arbitrary position and an arbitrary orientation within its workspace.

- Force control of parallel robots.

Force control requires higher-speed of response than position control, especially when rigidity of manipulators, force sensors and objects is high. In serial manipulators, however, it is difficult to shorten the response time due to their heavy weights. The fast dynamic behaviour of parallel manipulators can solve this issue.

- Bond-graph modelling of multi-DOF parallel robots.

The systematic nature of bond graph method, especially the hierarchical technique, can help reducing the time and the effort in modelling parallel robots with multiple degrees of freedom. A very difficult task would arise if link flexibility were to be considered during the modelling process. This, however, requires designing new ROD component that helps making the process easy and systematic.

References

1- K. Ting and G. Tsai," Mobility and synthesis of five-bar programmable linkages", The 9th Applied Mechanism Conference, Missouri, PP. III.1-III.8.

2- M. Shoham," A Textbook of Robotics 1", Eshed Robotec Ltd., 1982.

3- A. J. Critchlow," Introduction to Robotics", Macmillan Publishing Company, 1985.

4- D. McCloy, " Some comparisons of serial-driven and parallel-driven manipulators", *Journal of Robotica*, Vol. 8, No. 4, PP. 355-362, 1990.

5- M. Uchiyama, "Structures and characteristics of parallel manipulators", *Journal of Advanced Robotics*, Vol. 8, No. 6, PP. 545-557, 1994.

6- H. Funabshi, "In parallel actuated mechanisms as a new robotic mechanism", *Journal of Advanced Robotics*, Vol. 8, No. 6, PP. 535-544, 1994.

7- M. Tanaka, " Large-scale framed structure as parallel mechanism with hyper-redundancy", *Journal of Advanced Robotics*, Vol. 8, No. 6, PP. 573-587, 1994.

8- K. Hunt, " Structural kinematics of In-Parallel-Actuated Robot-Arms", *Journal of Mechanisms, Transmission, and Automation in design*, Vol. 105, No. 4, PP. 705-712, 1983.

9- J. Merlet, " Parallel manipulators: state of the art and perspectives", *Journal of Advanced Robotics*, Vol. 8, No. 6, PP. 589-596, 1994.

10- K. Miller:"Experimental Verification of Modeling of DELTA Robot by Direct Application of Hamilton's Principle", In: Proceeding of the IEEE International conference on Robotics and automation, PP. 532-537, 1995.

11- T. Ropponen and T. Arai," Accuracy Analysis of a Modified Stewart Platforms", In: Proceeding of the IEEE International conference on Robotics and automation, PP. 521-525, 1995.

12- S. Kawamura and others," Development of an Ultrahigh Speed Robot FALCON using wire Drive system", In: Proceeding of the IEEE International conference on Robotics and automation, PP. 215-220, 1995.

13- M. Lee," Design of High Stiffness Machining Arm Using Double Parallel Mechanisms", In: Proceeding of the IEEE International conference on Robotics and automation, PP. 234-240, 1995.

14- B. Shusheng, and others," Accuracy Analysis of the Serial-Parallel Micromotion Manipulator." In: Proceeding of the IEEE International conference on Robotics and automation, PP. 2258-2263, 1997.

15- Z. Mingyang and others," Development of a Redundant Robot manipulator Based on three DOF Parallel Platform", In: Proceeding of the IEEE International conference on Robotics and automation, PP. 221-226, 1995.

16- L. Beiner, "Redundant actuation of a closed-chain manipulator", *Journal of Advanced Robotics*, Vol. 11, No. 3, PP. 233-245, 1997.

17- S. Kock and W. Schumacher, " A Parallel x-y manipulator with actuation redundancy for high-speed and active-stiffness applications", In: Proc. of the IEEE Int. Conference on Robotics and Automation, Vol. 3, PP. 2295-2300, 1998.

18- K. Han, W. Chung, and Y. Youm," Local structuralization for the Forward Kinematics of Parallel Manipulators Using Extra Sensor Data.", In: Proceeding of IEEE Int. conference on Robotics and automation, PP. 514-520, 1995.

19- K. Lee and D. Shah," Kinematic Analysis of a Three Degree of Freedom In-Parallel Actuated Manipulator." IEEE *Journal of Robotics and automation*, Vol. 4, No. 3, PP. 354-360, June 1988.

20- G. Pennock and D. Kassner, "Kinematic Analysis of Planar Eight-Bar Linkage: Application to a Platform-Type Robot", Trans. of ASME, *Journal of Mechanical Design*, Vol. 114, No. 1, PP. 87-95, March 1992.

21- C. Gosselin and J. Angeles, " A Global Performance Index for the Kinematic Optimisation of Robotic Manipulators", Trans. of ASME, *Journal of Mechanical Design*, Vol. 113, PP. 220-226, September 1991.

22- G. Feng and others, " A Physical Model of the Solution Space and the Atlas of the Reachable Workspace for 2-DOF parallel Planar Manipulators", *Journal of Machine Theory*, Vol. 31, No. 2, PP. 173-184, 1996.

23- G. Feng and others, " Distribution of Some Parameters in Physical Model of the Solution Space of 2-DOF parallel Planar Manipulators", *Journal of Machine Theory*, Vol. 30, No. 6, PP. 811-817, 1995.

24- F. Gao, X. Liu and W. Gruver, " The Global Conditioning Index in the Solution Space of Two Degree of Freedom Planar parallel Manipulators", In: Proceeding of the IEEE Int. Conference on Systems, Man and Cybernetics, Vol. 5, PP. 4055-4058, 1995.

25- H. Ozaki and others, " The Atlas of the Payload Capability For Design of 2-DOF Planar Parallel Manipulators", In: Proceeding of the IEEE Int. Conference on Systems, Man and Cybernetics, Vol. 2, PP. 1483-1488, 1996.

26- V. Kumar, " Characterization of Workspaces of Parallel Manipulators", Trans. Of ASME, *Journal of Mechanical Design*, Vol. 114, PP. 368-375, September 1992.

27- J. Merlet, C. Gosselin and N. Mouly, " Workspaces of Planar Parallel Manipulators", *Journal of Machine Theory*, Vol. 33, No. 1/2, PP. 7-20, 1998.

28- A. Rao and A. Jagadeesh, " Structure-Based Dynamic characteristics of Planar Linkages Including Platfrom-Type Robots", *Journal of Robotic Systems*, Vol. 14, No. 8, PP. 621-629, 1997.

29- A. Fattah, J. Angles, and A. Misra," Dynamics of a 3-DOF Spatial Parallel Manipulator with Flexible Links", In: Proceeding of the IEEE Int. conference on Robotics and automation, PP. 627-632, 1995.

30- K. Lee and D. Shah," Kinematic Analysis of a Three Degree of Freedom In-Parallel Actuated Manipulator." IEEE *Journal of Robotics and automation*, Vol. 4, No. 3, PP 361-367, June 1988.

31- A. Codourey, " Dynamic Modelling of Parallel Robots for Computed-Torque Control Implementation", Int. *Journal of Robotics Research*, Vol. 17, No. 12, PP. 1325-1336, July 1998.

32- R. Ben-Horin, M. Shoham and S. Djerassi, " Kinematics, dynamics and construction of planarly actuated parallel robot", *Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 14, PP. 163-172, 1998.

33- K. Youcef-Toumi, " Analysis and Design of Manipulators With Decoupled and Configuration-Invariant Inertia Tensors Using Remote Actuation", Trans. of ASME, *Journal of Dynamic Systems, Measurements and Control*, Vol. 114, PP. 204-212, June 1992.

34- C. Gosselin and J. Wang, " Singularity loci of planar parallel manipulators with revolute actuators", *Journal of Robotics and Autonomous Systems*, Vol. 21, No. 4, PP. 377-398, 1997.

35- F. Gao and W. Gruver, " Performance Evaluation Criteria for Analysis and Design of Robotic Mechanisms", In: Proceedings of the Int. Conference on Advanced Robotics, ICAR, PP. 879-884, Monterey, CA, 1997.

36- D. Chablat and P. Wenger, " Working modes and aspects in fully parallel manipulators", In: Proc. of the 1998 IEEE Int. Conf. on Robotics and Automation, Vol. 3, PP. 1964-1969, 1998.

37- C. Gibson, D. Marsh, and Y. Xiang," Singular Aspects of General Planar Motions with Two Degrees of Freedom", Int. *Journal of Robotics Research*, Vol. 17, No. 10, PP. 1068-1080, Oct. 1998.

38- C. Collins and G. Long," The Singularity Analysis of an In-Parallel Hand Controller for Force-Reflected Teleportation", IEEE Transaction on *Robotics and automation*, Vol. 11, No. 5, PP. 661-669, Oct. 1995.

39- D. Nenchev and M. Uchiyama, " PARA-ARM: A Five–Bar Parallel Manipulator with Singularity-Perturbed Design", *Journal of Mechanism and Machine Theory*, Vol. 33, No. 5, PP. 453-462, 1998.

40- D. Nenchev and M. Uchiyama, " Dynamic Analysis of Parallel-Link Manipulators Under the Singularity-Consistent Formulation", IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, 1996, Osaka, Japan.

41- P. Taylor," Robotic Control", Macmillan Education, 1990.

42- Y. Nakamura, and M. Ghodoussi, "Dynamics Computation of Closed-Link Robot Mechanisms with Redundant and Non-Redundant Actuators." IEEE Transaction on *Robotics and automation*, Vol. 5, No. 3, PP. 294-302, June 1989.

43- B. Mcinnis and Chen-Kang Liu, "Kinematics and Dynamics in Robotics: A Tutorial Based Upon classical Concepts of Vectorial Mechanics", IEEE *Journal of Robotics and automation*, Vol. RA-2, No. 4, PP. 181-186, Dec. 1986.

44- J. Craig, " Introduction to Robotics", Addison Wesley, 1986.

45- Chang-Jin Li," A New Lagrangian Formulation of Dynamics for Robot Manipulators." *Journal of Dynamic System, Measurement, and Control*, Vol. 111, PP. 559-567, Dec. 1989.

46- You-Liang Gu and Nan K. Loh," Dynamic modeling and Control by Utilizing an Imaginary Robot Model", IEEE *Journal of Robotics and automation*, Vol. 4, No. 5, PP. 532-540, Oct. 1988.

47- T. Khalil," Design and Control of a Robot arm by Computer." Unpublished MSc Thesis, Zagazig University, Egypt, 1994.

48- P. Wellstead," Introduction to Physical Modelling." Academic Press, 1979.

49- D. Karnopp, and D. Margolis," System Dynamics: A Unified Approach." Wiley Interscience Publication, 1990.

50- P. Gawthrop, and L. Smith," Metamodelling: For Bond Graphs and Dynamic systems." Prentice Hall, London, 1996.

51- R. Wynn," Bond Graph Model Based Control of Robotic Manipulators." Unpublished Ph.D. Thesis, University of Glasgow, 1993.

52- J. Blundell," Bond Graphs for Modeling Engineering Systems." John Wiley & Sons, 1982.

53- R. Fotsu-Ngwompo, S. Scararda, and D. Thomasset " Bond Graph Methodology for the Design of an Actuating System: application to two link Manipulator", International Conference on Man, Machine, and Cybernetics, PP. 2478-2483,1997.

54- R. Kankaanranta, and H. Koivo," Dynamics and Simulation of Compliant motion of a Manipulator." IEEE *Journal of Robotics and automation*, Vol. 4, No. 2, PP. 163-173, April 1988.

55- E. Haug," Computer Aided Kinematics and Dynamics of Mechanical Systems", Vol. 1, Allyn and Bacon, London, 1889.

56- R. Gunawardana and F. Ghorbel, " PD Control of Closed-Chain Mechanical Systems: An Experimental Study", In: Proceeding of the 5th IFAC Symposium, Vol. 1, PP. 79-84, Nantes, France, Sept. 97.

57- L. Menini, A. Tornamble and L. Zaccarian," Modelling and Control of an Under-Actuated Closed Kinematic Chain." IEE Proceeding On Control Theory Applications, Vol. 145, No. 1, PP. 1-8, January 1998.

58- J. Luh and Y. Zheng," Computation of Input Generalized Forces for Robotics with Closed Kinematic Chain Mechanism." IEEE *Journal of Robotics and automation*, Vol. RA-1, No. 2, PP. 95-103, June 1985.

59- J. Murray, and G. Lovell," Dynamic Modelling of Closed-Chain Robotics Manipulators and Implications for Trajectory Control.", IEEE Transaction on *Robotics and automation*, Vol. 5, No. 4, PP. 522-528, August 1989.

60- J. Granda and J. Reus," New Developments in Bond Graph Modeling Software Tools: The Computer Aided Modelling Program 'CAMP-G' and 'Matlab', International Conference on Man, Machine, and Cybernetics, PP. 1542-1547, 1997.

61- P. Gawthrop," MTT: Model Transformation Tools." In: Proceeding of the International Conference on Bond Graph Modelling and Simulation, PP. 197-202, Las Vegas, January 1995.

62- P. Gawthrop," MTT: Model Transformation Tools. Online www home Page, 1997, URL: http: //mtt.sourceforge.net/.

63- D. Karnopp and D. Margolis," Analysis and Simulation of Planar mechanism systems using Bond Graphs." *Journal of Mechanical Design*, Vol. 101, PP. 187-191, April 1979.

64- A. Zeid," Bond Graph Modeling of Planar Mechanisms with Realistic Joint Effects." *Journal of Dynamic Systems, Measurement, and Control,* Vol. 111, PP. 15-23, March 1989.

65- R. Redfield, "A Bond Graph Representation of Lagrange's Equations", In: Proceedings of the ASME Dynamic Systems and Control Division, Vol. 57-1, PP. 431-437,1995.

66- R. Redfield, "A Synthesis of Design and Lagrangian Bond Graphs", In: Proceedings of the ASME Dynamic Systems and Control Division,Vol. 58, PP. 715-720, 1996.

67- Y. Khulief, "Dynamic Analysis of Mechanisms Using Constrained Lagrangian Bond Graphs", In: Proceedings of the ASME Design Engineering Division, Vol. 47, PP. 59-65, 1992.

68- D. Karnopp, "Understanding Multibody Dynamics Using Bond Graph Representations", *Journal of Franklin Institute,* Vol. 334B, No. 4, PP. 631-642, 1997.

69- W. Favre and S. Scavarda, "Bond Graph Representation of Multibody Systems with Kinematic Loops", *Journal of Franklin Institute,* Vol. 335B, No. 4, PP. 643-660, 1998.

70- J. Jang and C. Han, "Proposition of a Modeling Method for Constrained Mechanical Systems Based on the Vector Bond Graph", *Journal of Franklin Institute,* Vol.335B, No. 3, PP. 451-469, 1998.

71- R. Schilling, " Fundamentals of Robotics Analysis and Control", Prentice Hall, 1990.

72- S. Tadokoro, " Control of parallel mechanisms", *Journal of Advanced Robotics,* Vol. 8, No. 6, PP. 559-571, 1994.

73- P. Baines and J. Mills, "Feedback Linearized Joint Torque Control of a Geared, DC Motor Driven Industrial Robot", The Int. *Journal of Robotics Research,* Vol. 17, No. 2, PP. 169-192, February 1998.

74- H. Seraji, "A New Class of Nonlinear PID Controller", In: Proceedings of the 5th IFAC Symposium, Vol. 1, PP. 65-71, Nantes, France, September 1997.

75- H. Seraji, "Nonlinear and Adaptive Control of Force and Compliance in Manipulators", The Int. *Journal of Robotics Research,* Vol. 17, No. 5, PP. 467-484, 1998.

76- B. Armstrong, "Nonlinear PD Control with Incomplete State Knowledge: Damping without Derivative", In: Proceedings of the 5th IFAC Symposium, Vol. 1, PP. 73-78, Nantes, France, September 1997.

77- P. Gawthrop," Bond Graphs, Symbolic algebra and modelling of complex systems", Technical Report, Centre for systems and Control, Glasgow University, Feb. 1998.

78- M. Jean and C. Gosselin, "Static balancing of planar parallel manipulators", In: Proc. of the IEEE Int. conference on Robotics and Automation, vol. 4, PP. 3732-3737, 1996.

79- E. Umez-Eronini, " System dynamics and control ", Brooks/Cole Publishing Company, 1999.

80- M. Fujita and N. Tanaka, " A New Type of Robust Tracking Control of Robot Manipulators Based on Generalized SP-D Control Scheme ", In: Proceedings of the 1997 IEEE Int. Conference on Robotics and Automation, PP. 2371-2376, 1997.

81- A. Jaritz and M. Spong, " An Experimental Comparison of Robust Control Algorithms on a Direct Drive Manipulator ", IEEE Transactions on *Control Systems Technology*, Vol. 4, No. 6, PP. 627-640, 1996.

82- M. Spong, " On the Robust Control of Robot Manipulators ", IEEE *Transactions on Automatic Control*, Vol. 37, No. 11, PP. 1782-1786, 1992.

83- W. Gao and J. C. Hung, " Variable Structure Control of Nonlinear Systems: A New approach ", IEEE *Transactions on Industrial Electronics*, Vol. 40, No. 1, PP. 45-55, February 1993.

84- J. Craig, " Adaptive Control of Mechanical Manipulators ", Addison Wesley, 1988.

85- Y. Stepanenko and Chun-Yi SU, " Variable structure control of robot manipulators with nonlinear sliding manifolds ", Int. *Journal of Control*, Vol. 58, No. 2, PP. 285-300, 1993.

86- J. Slotine, " Sliding controller design for nonlinear systems ", Int. *Journal of Control*, Vol. 40, No. 2, PP. 421-434, 1984.

87- Nag-In Kim, Chong-Won Lee and P. Chang, " Sliding mode control with perturbation estimation: application to motion control of parallel manipulator ", *Journal of Control Eng. Practice*, Vol. 6, No. 11, PP. 1321-1330, 1998.

88- H. Elmami and N. Olgac, " Sliding mode control with perturbation estimation (SMCPE): a new approach ", Int. *Journal of Control*, Vol. 56, No. 4, PP. 923-941, 1992.

89- J. Slotine and S. Sastry, " Tracking control of nonlinear systems using sliding surfaces, with applications to robot manipulators ", Int. *Journal of Control*, Vol. 38, No. 2, PP. 465-492, 1983.

90- J. Slotine and J. Coetsee, " Adaptive sliding controller synthesis for non-linear systems ", Int. *Journal of Control*, Vol. 43, No. 6, PP. 1631-1651, 1986.

91- P. Kachroo and M. Tomizuka, " Chattering Reduction and Error Convergence in the Sliding-Mode Control of a Class of Nonlinear Systems ", IEEE *Transactions on Automatic Control*, Vol. 41, No. 7, PP. 1063-1068, 1996.

92- Z. Lu, S. Kawamura and A. Goldenberg, " An Approach to Sliding Mode-Based Impedance Control ", IEEE *Transactions On Robotics and Automation*, Vol. 11, No. 5, PP. 754-759, 1995.

93- H. Sage, M. Mathelin and E. Osterg, " Robust control of robot manipulator: a survey ", Int. *Journal of Control*, Vol. 72, No. 16, PP. 1498-1522, 1999.

94- D. Graham and D. McRuer, " Analysis of Nonlinear Control Systems ", John Wiley and Sons, Inc., 1961.

95- L. Freidovich and A. Pervozvanski, " Some Estimates of Performance for PID-like Control of Robotic Manipulators ", In: Proceedings of the 5th IFAC Symposium, Vol. 1, PP. 85-90, 1997.

Appendix A

Formulation of the dynamic equations of the One-cell parallel manipulator by using the method of Serial transformation

A.1 Introduction

The alternative approach for the formulation of the dynamic equations of motion of the systems with closed chains solves the problem by reducing the system to a tree structure mechanism. This tree consists of multi-serial chains and it was assumed that it is virtually actuated. To check the results obtained with the bond graph technique, this appendix is directed to the modelling of the one-cell parallel planar manipulator using the above alternative method. The formulation of the dynamic equations for the serial sub-chains is performed using the standard Lagrange method.

A.2 Derivation of the dynamic equations of motion for the one-cell parallel planar manipulator

To apply the method of serial transformation to the manipulator, we first transform the arm into two sub-serial chains by virtually cutting the workless joint that closes the loop. This step directly corresponds to removing the holonomic constraints that close the loop in the manipulator. Figure 1.1 shows the manipulator after performing the above step. It is assumed that the resulting two sub-serial chains are virtually driven.

The second step in the formulation is to derive the dynamics of the serial sub-chains Oac and Obc.

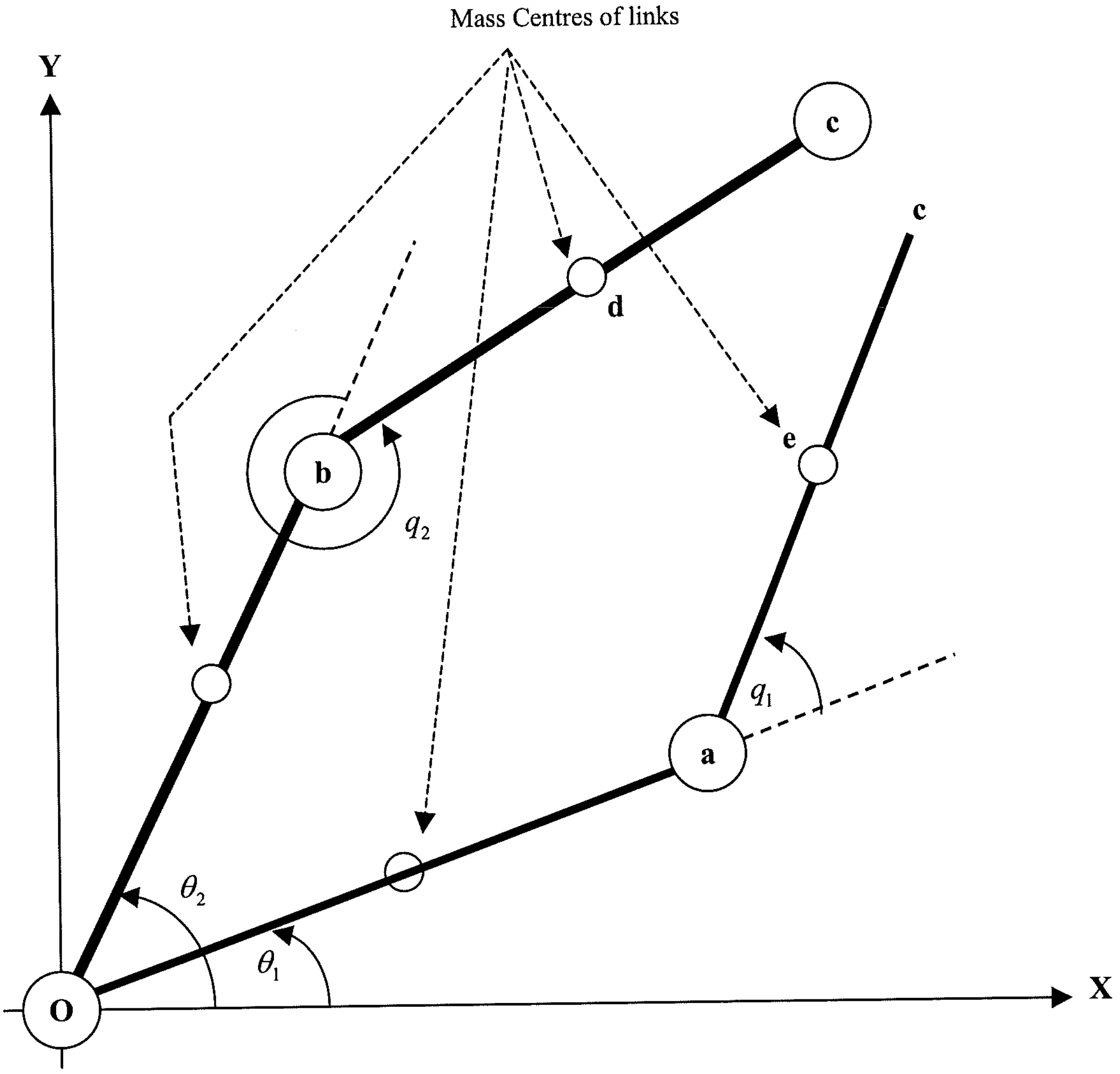


Fig. 1.1: Configuration of the one-cell parallel planar manipulator after its serial transformation.

A.2.1 Dynamics of serial sub-chain Oac

Recalling that each of the four links of the manipulator is of length $2L$, mass m and polar moment of inertia about centre of mass J.

Assuming uniform rigid links with circular cross-sections yields,

$$J = \tfrac{1}{3}mL^2 \qquad (1.1)$$

$$\left(I_{Ob}\right)_O = \left(I_{Oa}\right)_O = J + mL^2 = 4J \qquad (1.2)$$

Hence, the kinetic energy of link Oa is as follows:

$$T_{Oa} = \tfrac{1}{2}\left(I_{Oa}\right)_O\dot{\theta}_1^2 = \tfrac{1}{2}(4J)\dot{\theta}_1^2 = 2J\dot{\theta}_1^2 \qquad (1.3)$$

The position co-ordinates of the mass centre of link ac (point e) are given by,

$$x_e = 2L\cos\theta_1 + L\cos(\theta_1 + q_1) \qquad (1.4)$$

$$y_e = 2L\sin\theta_1 + L\sin(\theta_1 + q_1) \qquad (1.5)$$

Differentiating with respect to time and calculating the resultant velocity give,

$$\dot{x}_e = -2L\sin\theta_1\,\dot{\theta}_1 - L\sin(\theta_1 + q_1)\left(\dot{\theta}_1 + \dot{q}_1\right) \qquad (1.6)$$

$$\dot{y}_e = 2L\cos\theta_1\,\dot{\theta}_1 + L\cos(\theta_1 + q_1)\left(\dot{\theta}_1 + \dot{q}_1\right) \qquad (1.7)$$

$$v_e^2 = 4L^2\dot{\theta}_1^2 + L^2\left(\dot{q}_1^2 + \dot{\theta}_1^2 + 2\,\dot{q}_1\,\dot{\theta}_1\right) + 4L^2\dot{\theta}_1\left(\dot{q}_1 + \dot{\theta}_1\right)\cos q_1$$

$$= \left(5 + 4\cos q_1\right)L^2\dot{\theta}_1^2 + L^2\dot{q}_1^2 + \left(2 + 4\cos q_1\right)L^2\,\dot{\theta}_1\dot{q}_1 \qquad (1.8)$$

Hence, the kinetic energy of link ab is as follows:

$$T_{ab} = \tfrac{1}{2}mv_e^2 + \tfrac{1}{2}J\left(\dot{q}_1 + \dot{\theta}_1\right)^2$$
$$= J\left(8 + 6\cos q_1\right)\dot{\theta}_1^2 + 2J\dot{q}_1^2 + J\left(4 + 6\cos q_1\right)\dot{q}_1\dot{\theta}_1 \qquad (1.9)$$

So, the total kinetic energy of this serial sub-chain is given by:

$$T = T_{Oa} + T_{ac} \qquad (1.10)$$

Substituting from equations 1.3 and 1.9 into equations 1.10 leads to,

$$T = J\left(10 + 6\cos q_1\right)\dot{\theta}_1^2 + 2J\dot{q}_1^2 + J\left(4 + 6\cos q_1\right)\dot{q}_1\dot{\theta}_1 \qquad (1.11)$$

Lagrange's equations of motion:

$$\tau_i = \frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_i}\right) - \frac{\partial T}{\partial \theta_i} \quad \text{for i} = 1{,}2 \qquad (1.12)$$

$$\frac{\partial T}{\partial \dot{\theta}_1} = J\left(20 + 12\cos q_1\right)\dot{\theta}_1 + J\left(4 + 6\cos q_1\right)\dot{q}_1 \qquad (1.13)$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\theta}_1}\right) = J\left(20 + 12\cos q_1\right)\ddot{\theta}_1 + J\left(4 + 6\cos q_1\right)\ddot{q}_1$$
$$-12J\dot{q}_1\dot{\theta}_1\sin q_1 - 6J\sin q_1\,\dot{q}_1^2 \qquad (1.14)$$

$$\frac{\partial T}{\partial \theta_1} = 0 \qquad (1.15)$$

$$\frac{\partial T}{\partial \dot{q}_1} = 4J\dot{q}_1 + J\left(4 + 6\cos q_1\right)\dot{\theta}_1 \qquad (1.16)$$

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_1}\right) = 4J\ddot{q}_1 + J\left(4 + 6\cos q_1\right)\ddot{\theta}_1 - 6J\sin q_1\,\dot{\theta}_1\dot{q}_1 \qquad (1.17)$$

$$\frac{\partial T}{\partial q_1} = -6J\sin q_1\,\dot{\theta}_1^2 - 6J\sin q_1\,\dot{q}_1\dot{\theta}_1 \qquad (1.18)$$

Applying Lagrange's equations gives,

$$\tau_{Oa} = J\left\{(20+12\cos q_1)\ddot{\theta}_1 + (4+6\cos q_1)\ddot{q}_1 - 12\sin q_1\,\dot{\theta}_1\dot{q}_1 - 6\sin q_1\,\dot{q}_1^2\right\} \qquad (1.19)$$

$$\tau_{ac} = J\left\{4\ddot{q}_1 + (4+6\cos q_1)\ddot{\theta}_1 + 6\sin q_1\,\dot{\theta}_1^2\right\} \qquad (1.20)$$

Where, τ_{Oa} and τ_{ac} are the generalised virtual torques associated with links Oa and ac respectively.

A.2.2 Dynamics of serial sub-chain Obc

In a similar way the virtual generalised torques associated with links Ob and bc can be represented as follows,

$$\tau_{Ob} = J\left\{(20+12\cos q_2)\ddot{\theta}_2 + (4+6\cos q_2)\ddot{q}_2 - 12\sin q_2\,\dot{\theta}_2\dot{q}_2 - 6\sin q_2\,\dot{q}_2^2\right\} \qquad (1.21)$$

$$\tau_{bc} = J\left\{4\ddot{q}_2 + (4+6\cos q_2)\ddot{\theta}_2 + 6\sin q_2\,\dot{\theta}_2^2\right\} \qquad (1.22)$$

The third step in the formulation of the dynamic equations of the original parallel manipulator is to incorporate the constraints of the manipulator. This step is performed by firstly writing the idle joint generalised co-ordinates in terms of the actuated co-ordinates. After that we derive the transformation matrix (w) that relates the original closed chain generalised torques to those of the serial sub-chains.

From the original configuration of the one-cell parallel manipulator we can deduce that

$$q_1 = \theta_2 - \theta_1 \qquad (1.23)$$

$$q_2 = 2\pi - (\theta_2 - \theta_1) \qquad (1.24)$$

The transformation matrix is defined as follows,

$$w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ \dfrac{\partial q_1}{\partial \theta_1} & \dfrac{\partial q_1}{\partial \theta_2} \\ \dfrac{\partial q_2}{\partial \theta_1} & \dfrac{\partial q_2}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (1.25)$$

Now, if we let τ_1^C and τ_2^C be the generalised torques associated with links Oa and Ob of the original closed chain manipulator then we have the following relation

$$\begin{bmatrix} \tau_1^C \\ \tau_2^C \end{bmatrix} = w^T \cdot \begin{bmatrix} \tau_{Oa} \\ \tau_{Ob} \\ \tau_{ac} \\ \tau_{bc} \end{bmatrix} \qquad (1.26)$$

Which leads to,

$$\tau_1^C = \tau_{Oa} - \tau_{ac} + \tau_{bc} \qquad (1.27)$$

$$\tau_2^C = \tau_{Ob} + \tau_{ac} - \tau_{bc} \qquad (1.28)$$

Substituting from equations 1.19, 1.20, 1.21 and 1.22 into equations 1.27 and 1.28, respectively, we get the reduced equations of motion of the parallel planar manipulator, as follows,

$$\tau_1^C = J\left\{(20+12\cos q_1)\ddot{\theta}_1 + (4+6\cos q_1)\ddot{q}_1 - 12\sin q_1\,\dot{\theta}_1\dot{q}_1 - 6\sin q_1\,\dot{q}_1^2\right\}$$
$$- J\left\{4\ddot{q}_1 + (4+6\cos q_1)\ddot{\theta}_1 + 6\sin q_1\,\dot{\theta}_1^2\right\}$$
$$+ J\left\{4\ddot{q}_2 + (4+6\cos q_2)\ddot{\theta}_2 + 6\sin q_2\,\dot{\theta}_2^2\right\} \qquad (1.29)$$

Substituting from equations 1.23 and 1.24 and rearranging leads to,

$$\tau_1^C = 20J\ddot{\theta}_1 + 12J\cos(\theta_2-\theta_1)\ddot{\theta}_2 - 12J\sin(\theta_2-\theta_1)\dot{\theta}_2^2 \qquad (1.30)$$

By means of a similar approach we also get the following,

$$\tau_2^C = 20J\ddot{\theta}_2 + 12J\cos(\theta_2-\theta_1)\ddot{\theta}_1 + 12J\sin(\theta_2-\theta_1)\dot{\theta}_1^2 \qquad (1.31)$$

These results are the same as the results obtained in chapter three if we put the number of cells to equal to zero.

Assuming that the state vector is defined as $X^T = \lfloor J\dot{\theta}_1 \quad \theta_1 \quad J\dot{\theta}_2 \quad \theta_2 \rfloor$. Then, by transforming the equations of motion into the state space form, we get the following matrix form that validates the results obtained by the bond graph method:

$$\begin{bmatrix} 20 & 0 & 12\cdot\cos\Delta x & -12\cdot\sin\Delta x\cdot x_3 \\ 0 & 1 & 0 & 0 \\ 12\cdot\cos\Delta x & 12\cdot\sin\Delta x\cdot x_1 & 20 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix} = \begin{bmatrix}\tau_1\\ \dfrac{x_1}{j}\\ \tau_2\\ \dfrac{x_3}{j}\end{bmatrix}$$

Where, $\Delta x = x_4 - x_2$.

Appendix B

Label file and List of Various representations Generated by MTT

B.1 Label File

```
%SUMMARY unit_cell
%DESCRIPTION <Detailed description here>
%% Label file for system 2-DOF Parallel Manipulator (model_lbl.txt)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%
% %% $Id$
% %% $Log$
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
%       a comment (ie starting with %)
%       Component-name    CR_name    arg1,arg2,..argn
%       blank
% Component type EMTF
        c1            lcos    flow,l
        c2          lcos    flow,l
        c3          lcos    flow,n % n=2L
        c4          lcos    flow,n
        s1          lsin    flow,l
        s2          lsin    flow,l
        s3          lsin    flow,n
        s4          lsin    flow,n

% Component type I
        j_1         lin     flow,j
        j_11        lin     flow,j
        j_2         lin     flow,j
        j_22        lin     flow,j
        m_11_x      lin     flow,m
        m_11_y      lin     flow,m
        m_1_x       lin     flow,m
        m_1_y       lin     flow,m
        m_22_x      lin     flow,m
        m_22_y      lin     flow,m
        m_2_x       lin     flow,m
        m_2_y       lin     flow,m

% Component type SS
        t_I         external        external
        t_II        external        external
```

B.2 List of Generated Representations

- **List of Inputs/Outputs**

List of inputs-outputs for system 2-DOF Parallel Manipulator			
	Component	System	Repetition
1	t_I	2-DOF Parallel Manipulator	1
2	t_II	2-DOF Parallel Manipulator	1

- **List of Non-States Elements**

List of Non-States for system 2-DOF Parallel Manipulator			
	Component	System	Repetition
1	j_11	2-DOF Parallel Manipulator	1
2	j_22	2-DOF Parallel Manipulator	1
3	m_11_x	2-DOF Parallel Manipulator	1
4	m_11_y	2-DOF Parallel Manipulator	1
5	m_1_x	2-DOF Parallel Manipulator	1
6	m_1_y	2-DOF Parallel Manipulator	1
7	m_22_x	2-DOF Parallel Manipulator	1
8	m_22_y	2-DOF Parallel Manipulator	1
9	m_2_x	2-DOF Parallel Manipulator	1
10	m_2_y	2-DOF Parallel Manipulator	1

- **List of States**

List of States for system 2-DOF Parallel Manipulator			
	Component	System	Repetition
1	j_1	2-DOF Parallel Manipulator	1
2	j_2	2-DOF Parallel Manipulator	1
3	a (th_1)	2-DOF Parallel Manipulator	1
4	b (th_2)	2-DOF Parallel Manipulator	1

- **Simulation Code**

```
function [sys,x0] = model_cse(t,x,u,flag,xInitial);
%Constrained-state eqns in Simulab form for system 2-DOF Parallel Manipulator
%File model_cse.m
%Generated by MTT
if nargin<4; flag=0; end;
if (abs(flag) == 1) | (abs(flag) == 3);
% Set up the State variables;
MTTx1 = x(1);
MTTx2 = x(2);
MTTx3 = x(3);
MTTx4 = x(4);
% Set up the Input variables;
MTTu1 = u(1);
MTTu2 = u(2);
end;
if abs(flag) == 1        %state derivative;
mtte = zeros(4,4);
```

```
ans2=sin(mttx3)^2*m*n^2+2*j;
ans1=2*cos(mttx3)^2*l^2*m+cos(mttx3)^2*m*n^2+2*sin(mttx3)^2*l^2*m+ans2;
mtte(1,1)=ans1/j;
mtte(1,2)=(2*l*m*n*(cos(mttx3)*cos(mttx4)+sin(mttx3)*sin(mttx4)))/j;
mtte(1,4)=(2*l*m*mttx2*n*(-cos(mttx3)*sin(mttx4)+cos(mttx4)*sin(mttx3)))/j;
mtte(2,1)=(2*l*m*n*(cos(mttx3)*cos(mttx4)+sin(mttx3)*sin(mttx4)))/j;
ans2=sin(mttx4)^2*m*n^2+2*j;
ans1=2*cos(mttx4)^2*l^2*m+cos(mttx4)^2*m*n^2+2*sin(mttx4)^2*l^2*m+ans2;
mtte(2,2)=ans1/j;
mtte(2,3)=(2*l*m*mttx1*n*(cos(mttx3)*sin(mttx4)-cos(mttx4)*sin(mttx3)))/j;
mtte(3,3)=1;
mtte(4,4)=1;
mttedx(1,1)=mttu1;
mttedx(2,1)=mttu2;
mttedx(3,1)=mttx1/j;
mttedx(4,1)=mttx2/j;
sys = mtte\mttedx;
elseif abs(flag) == 3   %outputs;
mtty(1,1)=mttx1/j;
mtty(2,1)=mttx2/j;
sys = mtty;
elseif abs(flag) == 0   %structure;
        sys = [4,0,2,2,0,0];
        if nargin<5; xinitial = zeros(4,1); end;
        x0  = xinitial;
end;
```

Appendix C

Label file and List of Various representations Generated by MTT for the Hierarchical Model

C.1 Label File

```
%SUMMARY unit_cell
%DESCRIPTION <Detailed description here>
%% Label file for system unit-cell (h-model_lbl.txt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
% %% Version control history
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
% %% $Id$
% %% $Log$
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%

%% Each line should be of one of the following forms:
%       a comment (ie starting with %)
%       Component-name      CR_name       arg1,arg2,..argn
%       blank


% Component type ROD
link_1    none    l;l;j;m
link_22   none    l;l;j;m
link_11   none    l;l;j;m
link_2    none    l;l;j;m
```

```
% Component type Zero flow source
v_x    internal 0
e_x    internal 0
v_y    internal 0
e_y    internal 0

% Component type effort source
t_I    external external
t_II   external external

% Component type effort source
t_vx   f_x   internal  % Dummy force = 0
t_vy   f_y   internal  % Dummy force = 0

% Component type Zero effort source
f_a    0  internal
f_b    0  internal
f_c    0  internal
```

C.2 List of Generated Representations

- **List of Inputs/Outputs**

List of inputs/outputs for system Unit-Cell			
	Component	System	Repetition
1	t_I	Unit-Cell	1
2	t_II	Unit-Cell	1

- **List of Non-States Elements**

List of Non-States for system Unit-Cell			
	Component	system	repetition
1	m_x	Unit-Cell-link_1	1
2	m_y	Unit-Cell-link_1	1
3	m_x	Unit-Cell-link_22	1
4	m_y	Unit-Cell-link_22	1
5	m_x	Unit-Cell-link_11	1
6	m_y	Unit-Cell-link_11	1
7	m_x	Unit-Cell_link_2	1
8	m_y	Unit-Cell-link_2	1

- **List of States**

List of States for system Unit-Cell			
	Component	system	repetition
1	J	Unit-Cell-link_1	1
2	Mtt3	Unit-Cell-link_1_th	1
3	J	Unit-Cell-link_22	1
4	Mtt3	Unit-Cell-link_22_th	1
5	J	Unit-Cell-link_11	1
6	Mtt3	Unit-Cell-link_11_th	1
7	J	Unit-Cell-link_2	1
8	Mtt3	Unit-Cell-link_2_th	1

♦ Simulation Code

```
function [sys,x0] = h_mod_cse(t,x,u,flag,xInitial);
%Constrained-state eqns in Simulab form for system unit-cell
%File fourlink_cse.m
%Generated by MTT
if nargin<4; flag=0; end;
if (abs(flag) == 1) | (abs(flag) == 3);
% Set up the State variables;
MTTx1 = x(1);
MTTx2 = x(2);
MTTx3 = x(3);
MTTx4 = x(4);
MTTx5 = x(5);
MTTx6 = x(6);
MTTx7 = x(7);
MTTx8 = x(8);
% Set up the Input variables;
MTTu1 = u(1);
MTTu2 = u(2);
end;
if abs(flag) == 1        %state derivative;
E = zeros(8,8);
E(1,1)=(5*cos(mttx2)^2*l^2*m+5*sin(mttx2)^2*l^2*m+j)/j;
E(1,3)=(2*l^2*m*(cos(mttx2)*cos(mttx4)+sin(mttx2)*sin(mttx4)))/j;
E(1,4)=(2*l^2*m*mttx3*(-cos(mttx2)*sin(mttx4)+cos(mttx4)*sin(mttx2)))/j;
E(2,2)=1;
E(3,1)=(2*l^2*m*(cos(mttx2)*cos(mttx4)+sin(mttx2)*sin(mttx4)))/j;
E(3,2)=(2*l^2*m*mttx1*(cos(mttx2)*sin(mttx4)-cos(mttx4)*sin(mttx2)))/j;
E(3,3)=(cos(mttx4)^2*l^2*m+sin(mttx4)^2*l^2*m+j)/j;
E(4,4)=1;
E(5,5)=(cos(mttx6)^2*l^2*m+sin(mttx6)^2*l^2*m+j)/j;
E(5,7)=(2*l^2*m*(cos(mttx6)*cos(mttx8)+sin(mttx6)*sin(mttx8)))/j;
E(5,8)=(2*l^2*m*mttx7*(-cos(mttx6)*sin(mttx8)+cos(mttx8)*sin(mttx6)))/j;
E(6,6)=1;
E(7,5)=(2*l^2*m*(cos(mttx6)*cos(mttx8)+sin(mttx6)*sin(mttx8)))/j;
E(7,6)=(2*l^2*m*mttx5*(cos(mttx6)*sin(mttx8)-cos(mttx8)*sin(mttx6)))/j;
E(7,7)=(5*cos(mttx8)^2*l^2*m+5*sin(mttx8)^2*l^2*m+j)/j;
E(8,8)=1;
mttedx(1,1)=2*cos(mttx2)*f_y*l-2*sin(mttx2)*f_x*l+mttu1;
mttedx(2,1)=mttx1/j;
mttedx(3,1)=2*cos(mttx4)*f_y*l-2*sin(mttx4)*f_x*l;
mttedx(4,1)=mttx3/j;
mttedx(5,1)=2*cos(mttx6)*f_y*l-2*sin(mttx6)*f_x*l;
mttedx(6,1)=mttx5/j;
mttedx(7,1)=2*cos(mttx8)*f_y*l-2*sin(mttx8)*f_x*l+mttu2;
mttedx(8,1)=mttx7/j;
sys = E\Edx;
elseif abs(flag) == 3   %outputs;
mtty(1,1)=mttx1/j;
mtty(2,1)=mttx7/j;
sys = mtty;
elseif abs(flag) == 0   %structure;
        sys = [8,0,2,2,0,0];
        if nargin<5; xinitial = zeros(8,1); end;
        x0  = xinitial;
end;
```

Appendix D

Label file and List of Various representations Generated by MTT for the Hierarchical Model of the Redundant Parallel Robot

D.1 Label File

```
%SUMMARY unit_cell
%DESCRIPTION <Detailed description here>
%% Label file for system Redundant one-cell (model_lbl.txt)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
% %% Version control history
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
% %% $Id$
% %% $Log$
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%
%% Each line should be of one of the following forms:
%       a comment (ie starting with %)
%       Component-name     CR_name       arg1,arg2,..argn
%       blank
%
rod_a      none      l;l;j;m
rod_b      none      l;l;j;m
rod_c      none      l;l;j;m
rod_d      none      l;l;j;m
%
v_x      internal  0
e_x      internal  0
v_y      internal  0
e_y      internal  0
%
```

```
t_I        external    external
t_II       external    external

%
m_x       lin     flow,m_m
m_y       lin     flow,m_m
m_xx      lin     flow,n_n
m_yy      lin     flow,n_n

%
f_1       0   internal
f_2       external internal
f_3       external internal
t_vx      f_x internal
t_vy      f_y internal
```

D.2 List of Generated Representations

- **List of Inputs, outputs, states and non-states**

```
function [input_name,output_name,state_name] = G_R_M_struc
%% Structure file (G_R_M_struc.m)
%% Generated by MTT at Tue Feb  9 13:27:56 GMT 1999
input_name = [
'G_R_M_t_I'
'G_R_M_t_II'
'G_R_M_f_2'
'G_R_M_f_3'
];
nonstate_name = [
'G_R_M_rod_a_m_x'
'G_R_M_rod_a_m_y'
'G_R_M_rod_b_m_x'
'G_R_M_rod_b_m_y'
'G_R_M_rod_c_m_x'
'G_R_M_rod_c_m_y'
'G_R_M_rod_d_m_x'
'G_R_M_rod_d_m_y'
'G_R_M_m_x'
'G_R_M_m_y'
'G_R_M_m_xx'
'G_R_M_m_yy'
];
output_name = [
'G_R_M_t_I'
'G_R_M_t_II'
];
state_name = [
'G_R_M_rod_a_J'
'G_R_M_rod_a_th_mtt3'
'G_R_M_rod_b_J'
'G_R_M_rod_b_th_mtt3'
'G_R_M_rod_c_J'
'G_R_M_rod_c_th_mtt3'
'G_R_M_rod_d_J'
'G_R_M_rod_d_th_mtt3'
];
```

- **Simulation Code**

```
function [sys,x0] = G_R_M_cse(t,x,u,flag,xInitial);
%Constrained-state eqns in Simulab form for system G_R_M
%File G_R_M_cse.m
%Generated by MTT
if nargin<4; flag=0; end;
if (abs(flag) == 1) | (abs(flag) == 3);
% Set up the State variables;
MTTx1 = x(1);
MTTx2 = x(2);
MTTx3 = x(3);
MTTx4 = x(4);
MTTx5 = x(5);
MTTx6 = x(6);
MTTx7 = x(7);
MTTx8 = x(8);
% Set up the Input variables;
MTTu1 = u(1);
MTTu2 = u(2);
MTTu3 = u(3);
MTTu4 = u(4);
end;
if abs(flag) == 1        %state derivative;
mtte = zeros(8,8);
ans3=4*sin(mttx2)^2*l^2*m_m+4*sin(mttx2)^2*l^2*n_n+j;
ans2=4*cos(mttx2)^2*l^2*n_n+5*sin(mttx2)^2*l^2*m+ans3;
ans1=5*cos(mttx2)^2*l^2*m+4*cos(mttx2)^2*l^2*m_m+ans2;
mtte(1,1)=ans1/j;
ans3=2*sin(mttx2)*sin(mttx4)*m_m;
ans2=cos(mttx2)*cos(mttx4)*m+2*cos(mttx2)*cos(mttx4)*m_m+sin(mttx2)*sin(mttx4)
*m+ans3;
ans1=2*l^2*ans2;
```

```
mtte(1,3)=ans1/j;
ans3=2*cos(mttx4)*sin(mttx2)*m_m;
ans2=-cos(mttx2)*sin(mttx4)*m-
2*cos(mttx2)*sin(mttx4)*m_m+cos(mttx4)*sin(mttx2)*m+ans3;
ans1=2*l^2*mttx3*ans2;
mtte(1,4)=ans1/j;
mtte(2,2)=1;
ans3=2*sin(mttx2)*sin(mttx4)*m_m;
ans2=cos(mttx2)*cos(mttx4)*m+2*cos(mttx2)*cos(mttx4)*m_m+sin(mttx2)*sin(mttx4)
*m+ans3;
ans1=2*l^2*ans2;
mtte(3,1)=ans1/j;
ans3=-2*cos(mttx4)*sin(mttx2)*m_m;
ans2=cos(mttx2)*sin(mttx4)*m+2*cos(mttx2)*sin(mttx4)*m_m-
cos(mttx4)*sin(mttx2)*m+ans3;
ans1=2*l^2*mttx1*ans2;
mtte(3,2)=ans1/j;
ans2=4*sin(mttx4)^2*l^2*m_m+j;
ans1=cos(mttx4)^2*l^2*m+4*cos(mttx4)^2*l^2*m_m+sin(mttx4)^2*l^2*m+ans2;
mtte(3,3)=ans1/j;
mtte(4,4)=1;
mtte(5,5)=(cos(mttx6)^2*l^2*m+sin(mttx6)^2*l^2*m+j)/j;
mtte(5,7)=(2*l^2*m*(cos(mttx6)*cos(mttx8)+sin(mttx6)*sin(mttx8)))/j;
mtte(5,8)=(2*l^2*m*mttx7*(-cos(mttx6)*sin(mttx8)+cos(mttx8)*sin(mttx6)))/j;
mtte(6,6)=1;
mtte(7,5)=(2*l^2*m*(cos(mttx6)*cos(mttx8)+sin(mttx6)*sin(mttx8)))/j;
mtte(7,6)=(2*l^2*m*mttx5*(cos(mttx6)*sin(mttx8)-cos(mttx8)*sin(mttx6)))/j;
mtte(7,7)=(5*cos(mttx8)^2*l^2*m+5*sin(mttx8)^2*l^2*m+j)/j;
mtte(8,8)=1;
mttedx(1,1)=2*cos(mttx2)*f_y*l-2*sin(mttx2)*f_x*l+mttu1-mttu3;
mttedx(2,1)=mttx1/j;
mttedx(3,1)=2*cos(mttx4)*f_y*l-2*sin(mttx4)*f_x*l+mttu3-mttu4;
mttedx(4,1)=mttx3/j;
mttedx(5,1)=2*cos(mttx6)*f_y*l-2*sin(mttx6)*f_x*l+mttu4;
mttedx(6,1)=mttx5/j;
mttedx(7,1)=2*cos(mttx8)*f_y*l-2*sin(mttx8)*f_x*l+mttu2;
mttedx(8,1)=mttx7/j;
sys = mtte\mttedx;
elseif abs(flag) == 3   %outputs;
mtty(1,1)=mttx1/j;
mtty(2,1)=mttx7/j;
sys = mtty;
elseif abs(flag) == 0   %structure;
        sys = [8,0,2,4,0,0];
        if nargin<5; xinitial = zeros(8,1); end;
        x0  = xinitial;
end;
```

Appendix E

Matlab Script files for Calculating the Motor torques for Case I of the Redundant Actuated Manipulator

E.1 Script file for Case I (redundant motor at joint A)

```
% Case I
%Motor at joint A
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m_p=0;    % payload mass
 fx=0;
 fy=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%
  m=0.31; %Link mass;
m_m=0;   % mass of redundant motor at joint B;
m_n=0.44; % mass of redundant motor at joint A;
l=0.125;  %Link half-Length
J=0.0004; %moment of inertia about link centre of mass
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1=16+12*(m_m+m_p+m_n)/m;
k2=2*(m_m+m_p)/m+1;
k3=3*(m_m+m_p)/m+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=(0:.05:2);
th1=0.1745+1.0472*t.*t-0.349*t.*t.*t;          % Theta (1);
dth1=2.0944*t-1.047*t.*t;                       % Theta_dot (1);
ddth1=2.0944-2.094*t;                           % Theta_dot_dot (1);
th2=0.7854+0.9818*t.*t-0.3273*t.*t.*t;         % Theta (2);
dth2=1.9635*t-0.9818*t.*t;                      % Theta_dot (2);
ddth2=1.9635-1.9635*t;                          % Theta_dot_dot (2);
ans=-4*l*(fy*cos(th1)-fx*sin(th1));
```

```
u1=J*((k1+4)*ddth1+6*(k2+1)*cos(th2-th1).*ddth2-6*(k2+1)*sin(th2-
th1).*dth2.*dth2)+ans;
ans1=-4*l*(fy*cos(th2)-fx*sin(th2));
u2=J*(6*(k2+1)*cos(th2-th1).*ddth1+4*(k3+4)*ddth2+6*(k2+1)*sin(th2-
th1).*dth2.*dth2)+ans1;
it1=(2*u1+u2)/3;
it2=(u1+2*u2)/3;
it3=(u2-u1)/3;
ta=abs(it1)+abs(it2)+abs(it3);
save itest1.mat t ta;
```

E.2 Script file for Case I (redundant motor at joint B)

```
% Case I
%Motor at joint B
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m_p=0;    % payload mass
 fx=0;
 fy=0;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%
  m=0.31; %Link mass;
m_m=0.44; % mass of redundant motor at joint B;
m_n=0; % mass of redundant motor at joint A;
l=0.125;  %Link half-Length
J=0.0004; %moment of inertia about link centre of mass
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
k1=16+12*(m_m+m_p+m_n)/m;
k2=2*(m_m+m_p)/m+1;
k3=3*(m_m+m_p)/m+1;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t=(0:.05:2);
th1=0.1745+1.0472*t.*t-0.349*t.*t.*t;              % Theta (1);
dth1=2.0944*t-1.047*t.*t;                           % Theta_dot (1);
ddth1=2.0944-2.094*t;                                % Theta_dot_dot (1);
th2=0.7854+0.9818*t.*t-0.3273*t.*t.*t;             % Theta (2);
dth2=1.9635*t-0.9818*t.*t;                          % Theta_dot (2);
ddth2=1.9635-1.9635*t;                              % Theta_dot_dot (2);
ans=-4*l*(fy*cos(th1)-fx*sin(th1));
u1=J*((k1+4)*ddth1+6*(k2+1)*cos(th2-th1).*ddth2-6*(k2+1)*sin(th2-
th1).*dth2.*dth2)+ans;
ans1=-4*l*(fy*cos(th2)-fx*sin(th2));
u2=J*(6*(k2+1)*cos(th2-th1).*ddth1+4*(k3+4)*ddth2+6*(k2+1)*sin(th2-
th1).*dth2.*dth2)+ans1;
```

```
iit1=(2*u1+u2)/3;
iit2=(u1+2*u2)/3;
iit3=(u1-u2)/3;
tb=abs(iit1)+abs(iit2)+abs(iit3);
save iitest1.mat t tb;
```