Design, manufacture and control of a planar three degree-of-freedom parallel manipulator

Sophia E. S. Kounias

B. Eng.,Technological Educational Institute of Kavala, Greece, 1989

Department of Mechanical Engineering
McGill University
Montréal

August, 1993

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Engineering

ISBN 0-315-91849-7

Design manufacture and control of a 3-DOF parallel manipulator

by

Sophia E. S. Kounias

Master Thesis submitted to the
Department of Mechanical Engineering
McGill University

July 1993

To my parents
Stratis and Sonia

Table of Contents

List of Figures

List of Tables

List of Symbols

$\dot{\mathbf{c}}$:	Vector of Cartesian velocities
$\ddot{\mathbf{c}}$:	Derivative of the vector of Cartesian velocities
d:	Distance
$\mathbf{f}_i^*$:	Wrench vector
$\mathbf{f}^*$:	Generalized inertia force vector
$G_1(s)$:	Voltage transfer function
$G_2(s)$:	Torque transfer function
I_i:	Polar moment of inertia of the i^{th} link
$\mathbf{J}$:	Jacobian matrix
$\dot{\mathbf{J}}$:	Derivative of the Jacobian matrix
$\mathbf{J}_a$, $\mathbf{J}_u$:	Displacement Jacobian matrices
$\mathbf{K}_a$, $\mathbf{K}_u$:	Velocity Jacobian matrices
K_e:	Electrical constant of motor
K_d:	Damping coefficient
K_t:	Torque constant of motor
L_a:	Armature inductance
$l_{1,2,3}$:	Lengths of links
$\mathbf{M}_i$:	Mass matrix
m_i:	Mass of i^{th} link
N:	Gear ratio
$\mathbf{q}$:	9-dimensional vector
$\mathbf{q}_a$:	Vector of independent generalized coordinates
$\mathbf{q}_u$:	Vector of dependent generalized coordinates
$\mathbf{R}$:	Joint velocity Jacobian matrix
R_a:	Armature resistance
T_g:	Developed torque
T_L:	Load torque
T_f:	Friction torque

T_d:	Eddy current damping torque
T_j:	Torque necessary to accelerate the inertia
$\mathbf{T}$:	Natural orthogonal complement
$\mathbf{t}_i$:	Twist vector
$\mathbf{t}$:	Generalized twist vector
V_i:	Voltage of motor
V_e:	Rotational back e.m.f voltage
$\mathbf{v}_i$:	Two-dimensional vector
$X_{1,2}$:	Roots of kinematic equation
x_{2i}, y_{2i}:	Joint coordinates
x_C, y_C, x_D, y_D, x_F, y_F:	Joint coordinates
x_{0i}, y_{0i}:	Position of center of the base motors
$\dot{\theta}$:	Vector of joint rates
$\theta_{1,2,3}$:	Input angles of the manipulator
π_a:	Power supplied by actuators
π^*:	Power associated with the generalized inertia force
ϕ:	Orientation angle
ω:	Armature shaft velocity
Ω:	Twist constraint matrix
3-DOF:	Three-degree-of-freedom
CS1, CS2, CSA, CSB:	Comparators in chip L290
DAC:	Digital-to-analogue converter
DC:	Dirrect current
ERRV:	Motor drive signal
FTA, FTB:	Input signals to L291 from optical encoder
LSB:	Least significant bit
PM:	Permanent magnet
STA, STB:	Feedback signals for microprocessor

Abstract

This thesis presents the mechanical design of the prototype of a planar three degree-of-freedom parallel manipulator with revolute joints and the design and implementation of a closed-loop control circuit which controls the motors of the constructed manipulator.

Mechanical design here is understood as the decision-making process involved in tasks such as: material selection, structural design and drive selection. In this study the aforementioned tasks for the three degree-of-freedom parallel manipulator are described in detail and their conclusions are used for the construction of the manipulator.

Finally a motor speed/control system for the the control of the selected motors, to form a complete microprocessor-controlled DC motor servopositioning system, is designed. The final application circuit which includes additional hardware as well, is implemented, connected and tested with the motors of the constructed prototype.

In addition an introduction to parallel processing and an interface proposal are made as a first step towards the next phase of this project.

Résumé

Ce mémoire présente le concept mécanique d'un prototype planaire parallèle, à trois degrés de liberté, avec joints rotoïdes. De plus, nous y retrouvons le design et l'implantation d'un circuit à contre-réaction pour controler les moteur du manipulateur construit.

Le concept mécanique tient lieu de centre décisionnel concernant: la sélection des matériaux, des structures et la sélection des contrôles. Les étapes menant à la sélection des éléments précités sont détaillées dans l'étude, et les conclusions s'appliquent à la construction du manipulateur.

Finalement, un système de contrôle de vitesse à courrant continu et servodirectionnel est conçu, complétant le contrôle par mini-puce du moteur sélectionné également construit.

L'application finale du circuit incluant les connections entre les circuits et les résultats des tests sur le moteur du prototype construit, sont également présentés dans l'étude.

Enfin, l'introduction traitant du processeur parallèle et de l'interface constitue une première étape dans la poursuite du projet.

Acknowledgements

I am very grateful to my supervisors Prof. Louis J. Vroomen and Prof. Paul Zsombor-Murray, for their support and guidance during the course of the research. Their enthusiasm and encouragement have greatly contributed to the completion of this thesis.

I am indebted to my parents, Stratis and Sonia, for their constant financial support, confidence in my work and encouragement through out my studies.

Special thanks for their friendly support and for the many hours of discussions on technical matters (and not) go to: George Dedic, George Tewfik, Ferhan Bulca, Geoff McDowell, David Telfer and John Hayes.

In particular I am very happy to acknowledge my gratitude to my best friend Adonis Sagris for his continuous moral support, concern and enlightenment. Finally I wish to thank all my friends for constantly supporting me and in particular Soula Falagaris for helping me correct the thesis, Martine Hebert for helping me translate the abstract in French and Evan Kapetanakis for the photographs in this thesis.

Chapter 1 Introduction

1.1 Introduction

Most common robotic actuators are configured as a single open serial kinematic chain, i.e., the degree of connectivity between manipulator links is less than two. The end-effector (gripper-hand) is connected to only one link. Whether the motion between the links is provided through rotation or through prismatic actuators, the advantages of such a robot are:

- There is a certain conceptual relation between a human manipulator (arm-hand) and the robot, which is an advantage in programming.
- Complete independence (within tolerance due to construction) between all the joint actuators.
- Relatively simple to build.
- The direct and inverse kinematics and the dynamics have been analyzed for many cases.

The disadvantages of such a construction are:

- If the joint actuators are located at the link connections, the inertia of the whole robot will be large due to the mass of these actuators. This means that:
    1. The construction of the links must take the additional mass into account, i.e., rigid, heavy links,
    2. The links must be over-designed to limit the flexibility of the system,
    3. The velocity of the end-effector is limited if accuracy and precision are required,

4. If the joint actuators are gathered together, e.g., at the base plate, a means of transmission (chains, belts, cables) must be incorporated further reducing the relative accuracy and precision of the end-effector location. Even though one can design light but sufficiently strong links, the resulting flexibility often leads to possible instabilities if the arm is moved rapidly. This problem is currently under investigation i.e., Cyril [10].

Parallel manipulators have been the subject of intensive research for the past two decades. They are characterized by an architecture in which the gripper is attached to the base via several kinematic subchains leading to a configuration with multiple closed kinematic loops.

Parallel manipulators feature a number of advantages over serial manipulators, namely:

1. By allowing all their motors, or at least the heavier ones, to be fixed, i.e., mounted on the base, larger amounts of power are available whithout increasing the inertial load, thus increasing the load carrying capacity and the speed of operation,

2. By designing their links lighter, accuracy is increased and production costs are lower, and

3. By elimination of cable transmissions, accuracy and reliability are increased.

Moreover, parallel robotic architectures naturally appear in manipulation tasks requiring multi-fingered hands and in alternate tasks such as flight simulation and locomotion.

Until now the number of investigations on manipulators with closed kinematic chains is limited when compared with the ones that exist for manipulators of the serial type. The Stewart platform [51] is constructed by connecting two plates to six adjustable legs and is a six degree of freedom 6-SPS platform mechanism (S and P denote spherical and prismatic joints respectively). It was originally designed as an aircraft simulator and was also suggested for machine tool applications, space vehicle simulators, transfer machines, etc. Hoffman and McKinnon simulated aircraft motion via this type of platform by applying an SAP–VI finite element program [20]. McCallion and Truong [40] used this device as an automatic assembly table.

In his book Hunt [22] adopted the Stewart platform as a mechanism for a robotic arm. Following his idea, Fichter and McDowell [14] presented a review and some preliminary design concepts on this type of manipulator. Recently Hunt [23] undertook a systematic study of an in-parallel-actuated robotic arm in which many possible applicable parallel structures were reviewed. The Stewart platform is included.

In many instances, it may be desirable to use a parallel manipulator with a degree of freedom greater than the number of Cartesian coordinates required by a certain task, the manipulator thereby becoming redundant. Hence a manipulator can be redundant for certain tasks, even if it is nonredundant for others. Recent investigations of criteria that can be optimized over the extra degrees of freedom have been published. This includes obstacle avoidance (Maciejewski and Klein 1985 [38]), energy minimization, keeping joints within their mechanical limits (Liergeois 1977 [33]) and increasing dynamic response (Hollerbach and Suh 1985 [21], Salisbury and Abramowitz 1985 [47]).

The manipulator under study in this work, was first introduced by Hunt [23] and can be considered as a typical example of a planar multiloop mechanical system. In recent years this type of manipulators has attracted the attention of many researchers. Liergois [33], Klein and Huang [24] and Baillieul [6] among others have looked into the inverse kinematic problem of this type of manipulator, considering various types of approaches and optimization criteria. An optimum design method was presented by Gosselin and Angeles in 1988 [9]. Studies on inverse kinematics and dynamics of parallel manipulators similar to the one under study were reported by Lee and Shah [32], Stoughton and Kokkinis [29] and Ma and Angeles [37].

This study is part of a larger project which is presently being pursued in McRCIM (McGill Research Centre for Intelligent Machines). The project's main objective is to achieve real time control of the three degree-of-freedom parallel manipulator. The control of this type of mechanism differs fundamentally from that of the serial ma-

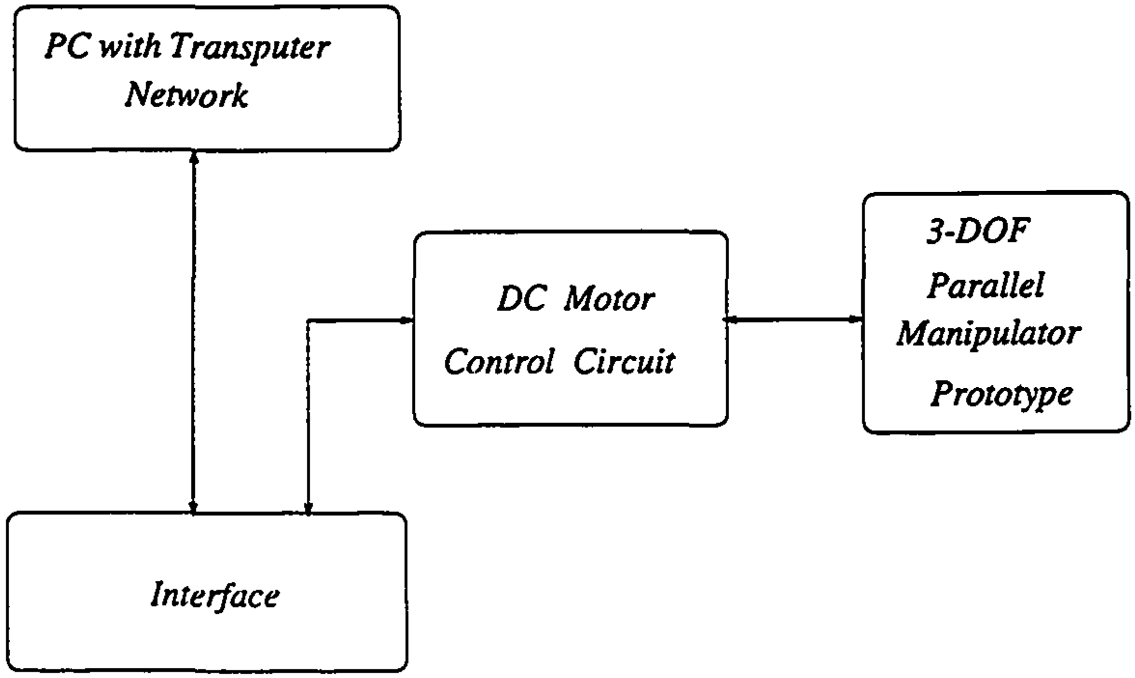


Figure 1.1: Block diagram of complete project

nipulator, hence an application of parallel processing is required. Parallel processing is achieved by introducing transputers, which are one chip microcomputers with their own memory and communication links for connecting one transputer to another.

A block diagram of the complete project is presented in Fig. 1.1. It consists of the following phases:

1. Design and manufacture of a three degree-of-freedom parallel manipulator with revolute joints,

2. Selection, design and implementation of a closed-loop control circuit for the actuators installed on the manipulator,

3. Development of a transputer program to control the manipulators trajectory and solving singularities that occur when following a specified path,

4. Design and implementation of an interface to achieve total bidirectional communication between the manipulator and the transputer network.

This study has three objectives which cover the first two phases of the project. The first objective is to design and build a prototype of the three degree-of-freedom parallel manipulator with revolute joints. The second objective is to select, install and test the proper actuators and shaft encoders of the prototype which will permit accuracy in control. The third objective is to design, implement and test a control circuit for the actuators and the encoders to achieve closed-loop control. After accomplishing these three objectives the manipulator will be ready to be controlled when the two last phases of the complete project will be terminated. These phases are presently under research by Helmy [18] and Felton [16].

1.2 Thesis overview

This thesis consists of 5 chapters.

Chapter 1 Introduction.

This chapter gives an introduction to the background and history of parallel manipulators. It includes a literature survey for the manipulator under study and a brief description of the subjects to be studied in each of the subsequent chapters, in order to provide an overview of the aforementioned objectives of the thesis.

Chapter 2 The three degree-of-freedom parallel manipulator.

In this chapter, the kinematics and dynamics of the parallel manipulator under study are presented. They have been clearly and thoroughly analyzed and described in the works of Gosselin [8] and Ma [36]. The singularities of this type of manipulator are presented. They are classified, based on their nature, into three categories which are architecture, configuration and formulation singularities.

There are several applications for this manipulator, including pick-and-place operations over a planar surface, machining of planar surfaces, mobile base for a spatial manipulator and moving platform for a terrestrial vehicle simulator.

Chapter 3 Designing the manipulator.

In Chapter 3, the mechanical design and construction of the three degree-of-freedom parallel manipulator prototype is presented.

Throughout the chapter the author explains in detail how material selection, structural design, actuator selection and shaft encoder selection were made. The chapter concludes with a description of the current prototype manufactured in the Machine Tool Laboratory of the Mechanical Engineering Department of McGill based on the previous selections.

The design philosophy differs from the one employed by Jacobsen et al. [27] in the design of the UTAH/M.I.T dextrous hand, where it was stated that ***convenience of acquisition has not been a sufficient reason for inclusion of sub-optimal components in the system.*** The emphasis of this study is closer to that of Salisbury et al. [48], where the component selection was based on financial and operational objectives.

Chapter 4 Controlling the parallel manipulator.

In this chapter, the control of the three base motors of the manipulator and the electronics involved in the hardware design of the control circuits are discussed.

The actuators, selected in the previous chapter, were permanent magnet DC motors. A mathematical model for the motors is presented and a control circuit is selected. Three chips (L290/91/92) together form a microprocessor-controlled DC motor servopositioning system that is both fast and accurate. These chips are discussed in detailed and their implementation along with additional hardware to form the final control circuit is shown. Tests of the implemented circuit are performed to prove that it operates as desired.

In addition, the control circuit is connected to the constructed manipulator and interfaced with a microcontroller and experiments are performed. The results of these experiments are presented and discussed.

At this point the objectives set at the beginning of the study are completed. The author however introduces the concept of parallel processing using the existing transputer network in McRCIM. In addition an interface proposal has been made. The interface in question is unidirectional and only open loop control can be achieved. This interface can be thought of as a first step towards the final interface design which will achieve total bidirectional communication between the manipulator plant and the transputer network.

Chapter 5 Conclusions and further research

The results of this research work are summarized in this chapter. Based on these results and the experience obtained from this research, a few suggestions for further research are proposed.

Chapter 2 The three degree-of-freedom Parallel Manipulator

2.1 Introduction

The planar three degree-of-freedom (3-DOF) parallel manipulator with revolute actuators used in this problem, was first analyzed and described by Gosselin [8]. It consists of three closed loop kinematic chains, Fig. 2.1. The advantages of such a manipulator are:

- The drive motors are fixed to the baseplate, therefore the links of the robot are lighter. In addition the motors do not contribute to the inertia of the links.
- As the end effector is controlled by three actuators, it is possible to cancel out vibrations.
- Accuracy, repeatability and velocity of the end effector can be improved.
- The load carrying capacity-vs-mass of the robot can be greatly increased.
- The calculation of the inverse kinematics becomes trivial, allowing for explicit solutions.

The disadvantages of this type of manipulator are:

- Limited workspace.
- Existence of many singularities in the workspace.
- Simultaneous control is required for all drive motors.
- Direct kinematic calculations, required for online dynamic control, are difficult.

The planar parallel manipulator represented in Fig. 2.1, has revolute joints and is composed of seven movable links and nine revolute joints. The motions of all links are limited in one plane parallel to the base. The three links connected to the base are considered as input links, while the one with three joints is the end effector, which undergoes a 3-DOF planar motion. The manipulator is driven by three motors, M_1, M_2 and M_3 which are located at the three fixed joints connecting the input links to the base. Hence, the three joints in question which are the actuated joints, are independent while the others, which are the unactuated joints, are dependent This means that once the variables associated with the former are assigned, those associated with the latter are fixed. The kinematic chains with the three closed loops are: M_1DABEM_2, M_2EBCFM_3 and M_3FCADM_1.

The manipulator studied here will be asked to arbitrarily position and orient the end-effector in the plane of motion, following a certain trajectory that will be task dependent. Hence there should not be any preferred general orientation for which the manipulator would have better properties. This requires that the manipulator should be symmetric. Therefore, the motors will be located on the vertices of an equilateral triangle and the lengths of the links will be the same for each leg, i.e.,

$$l_i = l_i' = l_i'' \qquad where: \qquad i = 1,2,3 \tag{2.1}$$

This assumption will be used throughout. In addition the distance between any two of the motors will be set equal to unity for normalization purposes.

The potential applications of this manipulator include pick-and-place operations over a planar surface, machining of planar surfaces, a mobile base for a spatial manipulator and as a moving platform for a terrestrial vehicle simulator.

2.2 Kinematics

Manipulator kinematics is the study of the relationship between joint and end-effector motions, disregarding the causality issues of these motions. The two major problems in manipulator kinematics are:

- The direct kinematics, were the motion of each actuated joint is given and the corresponding motion of the end-effector must be determined, and

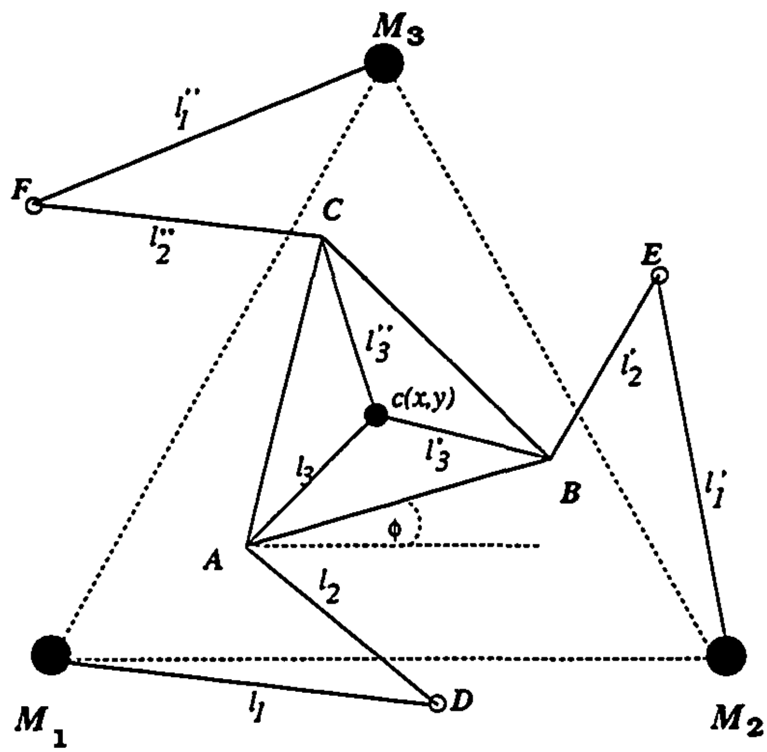


Figure 2.1: The three degree-of-freedom parallel manipulator with revolute joints

- The inverse kinematics, were the motion of the end-effector is given and the corresponding motion of each joint must be determined.

For serial manipulators, direct kinematics can be solved recursively and on-line because the relative motions of all joints are independent and given. However inverse kinematics has been a rather challenging research topic. It is now well developed as seen in the literature, i.e., McCarthy [41].

For parallel manipulators, the direct kinematics problem is not as simple as that of serial manipulators. Both the inverse and the direct kinematics are nonlinear problems, their complexities varying widely from manipulator to manipulator, depending on their architectures. In general, the direct problem is more difficult than the inverse one because of the presence of unactuated joints whose relative motions are dependent and not given.

On the other hand the inverse kinematics of parallel manipulators can be considered to have the same complexity as that of serial manipulators, because the problem can be solved independently within each individual kinematic loop and hence, the methodology of inverse kinematics of serial manipulators can be applied directly [37].

In recent years, because of their typical kinematic architectures related to parallel manipulators, Stewart platforms have attracted the attention of many researchers. Fichter [15] built several Stewart platforms and proposed their inverse kinematics and inverse dynamics models as well as a discussion on singularities. Merlet [42] studied direct kinematics and singularity problems while intensive studies have been presented on inverse kinematics and inverse dynamics by Do and Yang [12] and by Lee and Chao [31].

The kinematics of the 3-DOF planar manipulator with revolute joints, which was introduced by Hunt [22] and can be considered as a planar example of the well known Stewart platform [51], was described and analyzed in Gosselin's work [8].

Ma Ou and Angeles [37] studied the direct kinematics and dynamics that were applied to a three degree-of-freedom parallel manipulator and overcame the direct displacement problem using the method of virtual removal of kinematic constraints and a technique of four bar linkage performance evaluation to solve the problem efficiently.

2.2.1 Inverse kinematics

The problem of the inverse kinematics of the 3-DOF parallel manipulator consists of determining the angle values of the actuated joints, θ_1, θ_2, θ_3, for given values of x, y and ϕ, where x and y determine the position of the centroid of the end-effector and angle ϕ defines its orientation as shown in Fig. 2.1 and Fig. 2.2. It was shown in [1], that the solution to this problem contains eight different branches, i.e., two branches per leg since the solution for the input angles θ_1, θ_2 and θ_3 are completely uncoupled. Moreover the solution to each of these angles can be obtained from the input-output equation of a planar four bar linkage for each leg, which leads to a quadratic input-output equation, which thus contains two solutions, as shown, e.g., in Angeles and Bernier [1].

In Fig. 2.2 we can see one of the legs of the manipulator. We can see from this

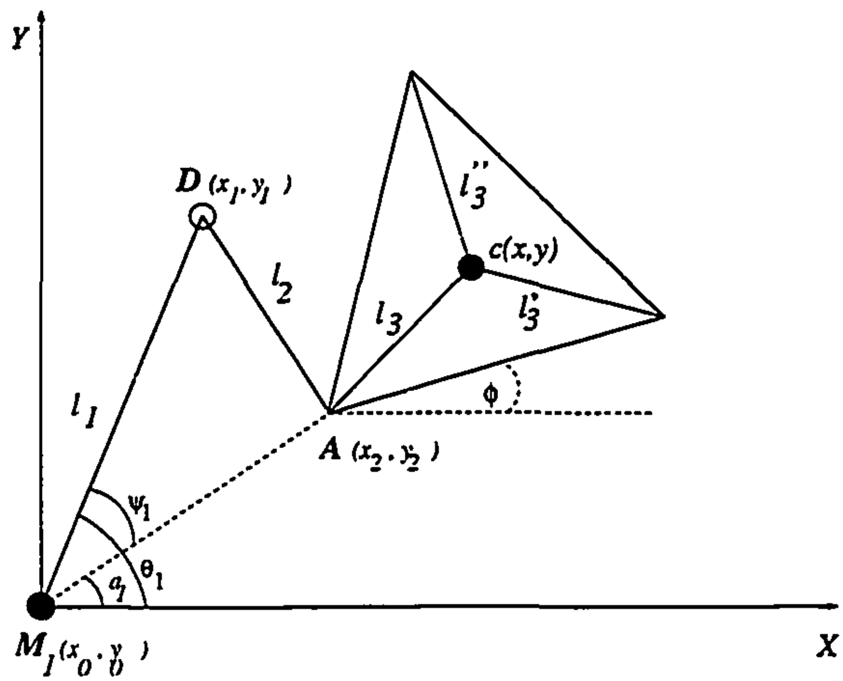


Figure 2.2: Analysis of the first leg

figure that when we have specified Cartesian coordinates (x,y,ϕ) we can consider the chain $cADM_1$ as a four bar linkage for which the position of the input link, l_3, is given. Angle θ_1 can then be computed if we use the input-output equation mentioned above.

Due to the redundant nature of the robot the inverse kinematics, i.e., the calculation of the actuating angles from the desired position, can be described through the following algebric equations:

$$\theta_i = a_i \pm \psi_i, \qquad i = 1, 2, 3 \tag{2.2}$$

where:

$$a_i = atan2(x_{2i},\ y_{2i}) \tag{2.3}$$

and:

$$\psi_i = \cos^{-2} \frac{(l_1^2 - l_2^2 + x_{2i}^2 + y_{2i}^2)}{2l_1\sqrt{x_{2i}^2 + y_{2i}^2}} \tag{2.4}$$

Angle ψ_i is being chosen on the main branch of the inverse cosine function, i.e., $0 \leq \psi \leq \pi$. Coordinates x_{2i} and y_{2i} are defined as:

$$x_{2i} = x - l_3 \cos\phi_i - x_{0i} \tag{2.5}$$

$$y_{2i} = y - l_3 \cos\phi_i - y_{0i} \tag{2.6}$$

where angles ϕ_i are given by:

$$\phi_i = (\phi + \frac{\pi}{6},\ \phi + \frac{5\pi}{6},\ \phi - \frac{\pi}{2})^T \tag{2.7}$$

and:

$$x_{0i} = (0,\ 1,\ \frac{1}{2})^T \tag{2.8}$$

$$y_{0i} = (0,\ 0,\ \frac{\sqrt{3}}{2})^T \tag{2.9}$$

are the positions of the centres of the motors.

2.2.2 Direct kinematics

As mentioned in the beginning of this section, the direct kinematic problem for parallel manipulators is more complex than the inverse problem. Based on a theorem shown by Hunt [23], that says that the solution of the direct kinematic problem for the planar three-degree-of freedom parallel manipulator leads to a maximum of six different branches, Gosselin [9] solved the direct kinematics of the manipulator under study using the following procedure:

Due to the redundant construction of the manipulator, the values x, y and ϕ can be obtained from the measured angle θ_i (Fig. 2.2)

First he calculated the position of the points D, E and F (Fig. 2.1) from the angles

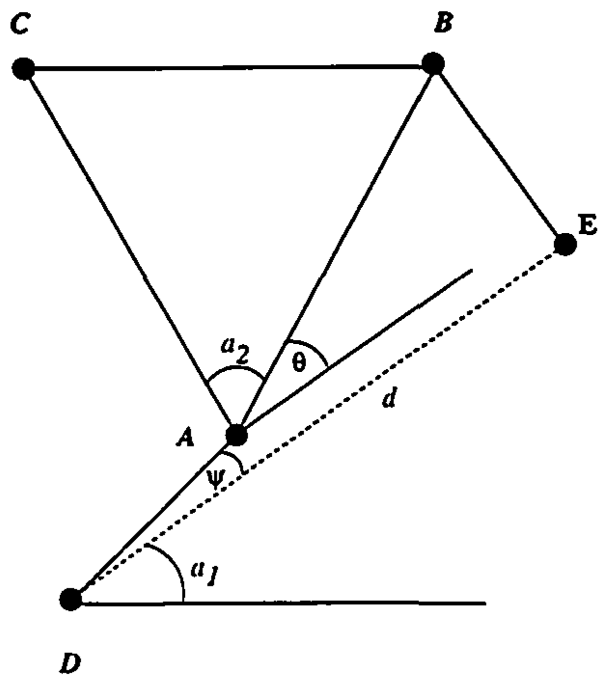


Figure 2.3: Planar four–bar linkage

θ_i and the lengths of the linkages l_i. The coordinates of point C can be expressed by the points D and E and by the angle ψ (Fig. 2.3). As Gosselin proves in his work there exists a maximum of six solutions.

The position of point C of the considered four bar linkage which is shown in Fig. 2.3, can be written as:

$$x_C = x_D + l_2 \cos(a_1 + \psi) + \sqrt{3}\, l_3 \cos(a_1 + a_2 + \theta) \tag{2.10}$$

$$y_C = y_D + l_2 \sin(a_1 + \psi) + \sqrt{3}\, l_3 \sin(a_1 + a_2 + \theta) \tag{2.11}$$

where:

$$a_1 = atan2\left[\frac{y_E - y_D}{x_E - x_D}\right] \tag{2.12}$$

$$a_2 = \frac{\pi}{3} \tag{2.13}$$

and:

$$\theta_{1,2} = 2\tan^{-2}\left[\frac{B \pm \sqrt{B^2 - AC}}{A}\right] \tag{2.14}$$

In eq. 2.14 we have :

$$A = m_1 - m_1 + (1 + m_3)\cos\psi \tag{2.15}$$

$$B = \sin\psi \tag{2.16}$$

and:

$$C = m_1 + m_2 + (m_3 - 1)\cos\psi \tag{2.17}$$

where:

$$m_1 = \frac{-d^2 - 3l_3^2}{2\sqrt{3}\,l_2\,l_3} \tag{2.18}$$

$$m_2 = \frac{d}{l_2} \tag{2.19}$$

$$m_3 = \frac{d}{\sqrt{3}\,l_3} \tag{2.20}$$

$$d = \sqrt{(x_E - x_D)^2 + (y_E - y_D)^2} \tag{2.21}$$

Therefore Gosselin ended up with a nonlinear equation that had to be solved:

$$(x_C - x_F)^2 + (y_C - y_F)^2 = l_2^2 \tag{2.22}$$

Equation 2.22 has been solved using the secant method. The range over which eq. 2.22 has real solutions is determined by the positive semidefiniteness of the quantity under the square root in eq. 2.14.

$$B^2 - AC \geq 0 \tag{2.23}$$

Which leads to, with the help of eqs. 2.15, 2.16 and 2.17:

$$(1 - m_1^2 + m_2^2) - 2(m_1 m_2 m_3) X - m_3^2 X^2 \geq 0 \tag{2.24}$$

where:

$$X = \cos\psi \tag{2.25}$$

Since the left hand side of eq. 2.24 reperesents a parabola with negative curvature everywhere, the roots of this parabola will give the limits of the range of validity of X from which the range of validity of ψ can be found. Due to the cosine function involved in eq. 2.25 it may happen that we obtain two distinct ranges of validity for angle ψ both of which should be considered. The roots of the parabola can be written as:

$$X_{1,2} = \frac{m_1 m_3 + m_2 \pm \sqrt{m_2^2 + m_3^2 + m_2^2 m_3^2 + 2 m_1 m_2 m_3}}{-m_3^2} \tag{2.26}$$

Once the range of the validity of ψ is known, we can use the secant method to obtain the solution for ψ angle.

2.2.3 Velocity Inversion

Since the inverse kinematic problem is easier to solve than the direct one the Jacobian matrix will be defined in terms of inverse transformation, i.e.,

$$\mathbf{J}\dot{\mathbf{c}} = \dot{\theta} \tag{2.27}$$

where $\dot{\mathbf{c}}$ is the vector of Cartesian velocities, given here by $\dot{\mathbf{c}} = [\dot{x}, \dot{y}, \dot{\phi}]^T$ and $\dot{\theta}$ is the vector of joint rates given here by $\dot{\theta} = [\dot{\theta}_1, \dot{\theta}_3, \dot{\theta}_3]^T$.

The Jacobian can be obtained by differentiation of equations 2.2 – 2.6 with respect to time. This leads to the following:

$$\mathbf{J} = \begin{bmatrix} \frac{a_1}{d_1} & \frac{b_1}{d_1} & \frac{c_1}{d_1} \\ \frac{a_2}{d_2} & \frac{b_2}{d_2} & \frac{c_2}{d_2} \\ \frac{a_3}{d_3} & \frac{b_3}{d_3} & \frac{c_3}{d_3} \end{bmatrix} \tag{2.28}$$

where, for i = 1, 2, 3:

$$a_i = x - x_{0i} - l_1 \, cos\theta_i - l_3 \cos\phi_i \tag{2.29}$$

$$b_i = y - y_{0i} - l_1 \, sin\theta_i - l_3 \sin\phi_i \tag{2.30}$$

$$c_i = -l_3[(y - y_{0i}) \cos\phi_i - (x - x_{0i}) \sin\phi_i] + l_1 l_3 \sin\theta_i - \phi_i \tag{2.31}$$

$$d_i = -l_1[(y - y_{0i}) \cos\theta_i - (x - x_{0i}) \sin\theta_i] - l_1 l_3 \sin\theta_i - \phi_i \tag{2.32}$$

It is obvious that the computation of the Jacobian matrix requires that the inverse kinematics problem be solved first.

2.2.4 Acceleration Inversion

The relationship between the joint and Cartesian accelerations was derived by differentiation of eq. 2.27 and the following was obtained:

$$\mathbf{J}\ddot{\mathbf{c}} + \dot{\mathbf{J}}\dot{\mathbf{c}} = \ddot{\theta} \tag{2.33}$$

where $\ddot{\mathbf{c}} = [\ddot{x}, \ddot{y}, \ddot{\phi}]^T$ and $\ddot{\theta} = [\ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3]^T$. The other quantities are assumed to be known from the velocity inversion. The only undefined matrix in the above equation is the derivative of the Jacobian denoted as $\dot{\mathbf{J}}$. Differentiating eqs. 2.28 to 2.32 we get:

$$\dot{\mathbf{J}} = \begin{bmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{bmatrix} \tag{2.34}$$

where, for i = 1, 2, 3:

$$A_i = \frac{d_i \dot{a}_i - a_i \dot{d}_i}{d_i^2}, \qquad B_i = \frac{d_i \dot{b}_i - b_i \dot{d}_i}{d_i^2}, \qquad C_i = \frac{d_i \dot{c}_i - c_i \dot{d}_i}{d_i^2} \tag{2.35}$$

and:

$$\dot{a}_i = \dot{x} + l_1 \dot{\theta}_i \sin\theta_i + l_2 \dot{\phi} \sin\phi \tag{2.36}$$

$$\dot{b}_i = \dot{y} - l_1 \dot{\theta}_i \cos\theta_i - l_2 \dot{\phi} \cos\phi \tag{2.37}$$

$$\begin{aligned} \dot{c}_i =& l_3 \dot{\phi}[(x - x_{oi} \cos\phi_i + (y - y_{oi}) \sin\phi_i] + l_3[\dot{x} \sin\phi_i - \dot{y} \cos\phi_i] \\ & + l_1 l_3 (\dot{\theta}_i - \dot{\phi}_i) \cos(\dot{\theta}_i - \dot{\phi}_i) \end{aligned} \tag{2.38}$$

$$\begin{aligned} \dot{d}_i =& - l_1 \dot{\theta}[(x - x_{oi} \cos\theta_i + (y - y_{oi}) \sin\theta_i] - l_1[\dot{x} \sin\theta_i - \dot{y} \cos\theta_i] \\ & + l_1 l_3 (\dot{\theta}_i - \dot{\phi}_i) \cos(\dot{\theta}_i - \dot{\phi}_i) \end{aligned} \tag{2.39}$$

This completes the acceleration inversion.

2.3 Dynamics

If we want to control a manipulator, we need to calculate actuator forces. While investigating its performance and simulating its motion, we require the dynamic equations of motion. A manipulator with closed kinematic chains, as the one under study, is a holonomic dynamic system in which the dynamics are not as easy to handle as are the dynamics of a simple open chain.

Recently several works have appeared on the direct and inverse dynamics of closed chain manipulators. Wittenburg [53] treated a closed chain mechanism as if it were cut open at one joint to virtually form a general open chain with a tree structure.

Although the final objective in the inverse dynamics problem is different, the general methods for formulating the problem are similar to those used in the direct dynamics problem. Luh and Zheng [34] and Kleinfinger and Khalil [28] set the torques applied to the passive joints to zero and the torques applied to the active ones are found by a linear system solution. Nakamura and Ghodoussi [43] derived the torques applied at the active joints by projecting the generalized torque vector of the unconstrained tree-structure system using a linear map incorporating the Jacobian of the passive joints with respect to the active joints; this is defined by an appropriate constraint map null space basis. The method was applied to a 5-bar planar linkage for robotic fingers. Angeles and Lee [2] proposed a formulation which makes use of a special basis of the null space of the velocity constraint map, defining a natural orthogonal complement of the constraint space. They applied this formulation to the 3-DOF fully parallel planar arm under study. They later gave a more detailed exposition of the method with applications to open-chain robot arms and lower pair single degree-of-freedom mechanisms [3].

Ma and Angeles [37] investigated and solved the inverse and direct dynamics problem of the three degree-of-freedom parallel manipulator under study, using the method of the natural orthogonal complement that was introduced by Angeles and Lee. In this work we will be using these results, as we will be needing them to control the manipulator later on. In the two following section we will outline Ma's and Angeles work on the inverse and direct dynamics of the manipulator.

2.3.1 Dynamics Modelling

The manipulator under study containing nine joints, naturally requires nine generalized coordinates, grouped in a 9-dimensional vector $\mathbf{q}$, to represent the kinematic relationship between individual links. Each generalized coordinate represents the rotational displacement of a joint. Because of the presence of closed kinematic loops, some generalized coordinates depend on others. Hence, $\mathbf{q}$ can be partioned as

$$\mathbf{q} = \begin{bmatrix} \mathbf{q_a} \\ \mathbf{q_u} \end{bmatrix} \tag{2.40}$$

where $\mathbf{q_a} = [q_1, q_2, q_3]^T$ consisting of independent generalized coordinates, which are associated with the actuated joints, and $\mathbf{q_u} = [q_4, ..., q_9]^T$ being a 6-dimensional vector of dependent generalized coordinates associated with the unactuated joints. These generalized coordinates are subjected to kinematic constraints which can be described by a set of holonomic constraint equations of the general form:

$$\phi(\mathbf{q_a}, \mathbf{q_u}) = \mathbf{0} \tag{2.41}$$

The number of independent constraint equations should be equal to the dimension of $\mathbf{q_u}$, i.e., six.

For the dynamic modelling of the manipulator, a 3-dimensional vector of *twist*, $\mathbf{t_i}$, and a 3 × 3 extended *mass matrix*, $\mathbf{M_i}$, of the ith link of the manipulator, for i = 1, 2, ..., 7, was first defined as follows:

$$\mathbf{t_i} = \begin{bmatrix} \omega_i \\ \dot{\mathbf{c}}_\mathbf{i} \end{bmatrix}, \qquad \mathbf{M_i} = \begin{bmatrix} I_i & 0 & 0 \\ 0 & m_i & 0 \\ 0 & 0 & m_i \end{bmatrix} \tag{2.42}$$

where m_i and I_i are the mass and the polar moment of inertia about the mass center of the ith link, respectively. Moreover a 3-dimensional vector of inertia *wrench*, $\mathbf{f_i^*}$, of the ith link was defined as:

$$\mathbf{f_i^*} = -\mathbf{M_i}\dot{\mathbf{t}}_\mathbf{i} = -\begin{bmatrix} \mathbf{I_i}\dot{\omega}_\mathbf{i} \\ \mathbf{m_i}\ddot{\mathbf{c}}_\mathbf{i} \end{bmatrix} \tag{2.43}$$

After assembling all the $\mathbf{t_i}$ vectors in a single 21-dimensional vector $\mathbf{t}$ and all the $\mathbf{f_i^*}$ vectors in a 21-dimensional vector $\mathbf{f^*}$, Ma and Angeles obtained the vectors of *generalized twist* and *generalized inertia* force of the manipulator, which are respectively:

$$\mathbf{t} = \begin{bmatrix} \mathbf{t}_1 \\ \vdots \\ \mathbf{t}_7 \end{bmatrix}, \qquad \mathbf{f}^* = \begin{bmatrix} \mathbf{f}_1^* \\ \vdots \\ \mathbf{f}_7^* \end{bmatrix} \tag{2.44}$$

It the was shown that, for any holonomic mechanical system, the following set of twist constraint equations hold:

$$\mathbf{\Omega t} = 0 \tag{2.45}$$

For the manipulator under study, $\mathbf{\Omega}$, defined as the *twist constraint* matrix, is a 18×21 dimension matrix and configuration-dependent. It was also shown that $\mathbf{t}$ is a linear transformation of the vector of generalized velocities, $\dot{\mathbf{q}}_a$, i.e.,

$$\mathbf{t} = \mathbf{T}\dot{\mathbf{q}}_a \tag{2.46}$$

From eqs. 2.45 and 2.46, it was shown that $\mathbf{\Omega T} = 0$ and hence the 21×3 matrix $\mathbf{T}$ was termed the *natural orthogonal complement* of matrix $\mathbf{\Omega}$. Vector $\mathbf{t}$ can also be expressed as a linear transformation of the vector $\dot{\mathbf{q}}$, i.e.,

$$\mathbf{t} = \mathbf{K}\dot{\mathbf{q}} \tag{2.47}$$

From eqs. 2.46 and 2.47 as well as eq. 2.41, it was derived that:

$$\mathbf{T} = \mathbf{K}_a - \mathbf{K}_u\mathbf{J}_u^{-1}\mathbf{J}_a \tag{2.48}$$

where $\mathbf{K}_a$ and $\mathbf{K}_u$ are termed velocity Jacobian matrices while $\mathbf{J}_a$ and $\mathbf{J}_u$ are *displacement Jacobian* matrices, which were defined as:

$$\mathbf{K}_a = \frac{\partial \mathbf{t}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_a}, \qquad \mathbf{K}_u = \frac{\partial \mathbf{t}(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_u} \tag{2.49}$$

$$\mathbf{J}_a = \frac{\partial \phi(\mathbf{q})}{\partial \mathbf{q}_a}. \qquad \mathbf{J}_u = \frac{\partial \phi(\mathbf{q})}{\partial \mathbf{q}_u} \tag{2.50}$$

If τ_a denotes the 3-dimensional vector of driving torques, supplied by the three actuators of the manipulator, π_a and π^* denote the power supplied by the actuators and the power associated with the generalized inertia force, respectively, then it was shown that:

$$\pi_a = \dot{\mathbf{q}}_a{}^{\mathbf{T}}\tau_a, \tag{2.51}$$

$$\pi^* = \mathbf{t}^{\mathbf{T}}\mathbf{f}^* = \dot{\mathbf{q}}_a\mathbf{T}^{\mathbf{T}}\mathbf{f}^* \tag{2.52}$$

From the conservation of energy of the whole system, the following holds:

$$\pi_a + \pi^* = 0 \tag{2.53}$$

i.e.,

$$\dot{\mathbf{q}}_a^{\mathbf{T}}\tau_a = -\dot{\mathbf{q}}_a^{\mathbf{T}}\mathbf{T}^{\mathbf{T}}\mathbf{f}^* \tag{2.54}$$

By definition, all components of $\mathbf{q}_a$ are independent and hence, the following was derived from eq. 2.54:

$$\tau_a = -\mathbf{T}^{\mathbf{T}}\mathbf{f}^* \tag{2.55}$$

which is the dynamics model of the manipulator. In this formula, the authors did not consider gravity forces. If friction is considered, the power dissipated by friction forces, torques, must be included in eq. 2.53, which leads to the following dynamics model:

$$\tau_a = -\mathbf{T}^{\mathbf{T}}(\mathbf{f}^* + \mathbf{f}^{\mathbf{f}}) \tag{2.56}$$

or

$$\tau_a = -\mathbf{T}^{\mathbf{T}}\mathbf{f}^* - \mathbf{R}^{\mathbf{T}}\tau^{\mathbf{f}} \tag{2.57}$$

where $\mathbf{f}^{\mathbf{f}}$ is a 21-dimensional vector composed of all friction wrenches exerting at each links mass center, while $\tau^{\mathbf{f}}$ is a 9-dimensional vector composed of all friction torques exerted on each joint. Moreover, $\mathbf{R}$ is a 9×3 *joint velocity Jacobian matrix*, which was defined as:

$$\mathbf{R} \equiv \frac{\partial \dot{\mathbf{q}}}{\partial \dot{\mathbf{q}}_a} = \begin{bmatrix} \mathbf{1} \\ -\mathbf{J}_u^{-1}\mathbf{J}_a \end{bmatrix} \tag{2.58}$$

where $\mathbf{1}$ is the 3×3 identity matrix.

So the final dynamics models Ma and Angeles derived are given by eqs. 2.55, 2.56 and 2.57 and are ready to be used for solving the inverse dynamics problem.

2.3.2 Direct Dynamics (Simulation)

For the formulation of the direct dynamics, let $\mathbf{M}$ denote the 21 × 21 *generalized extended mass matrix* of the manipulator, defined as:

$$\mathbf{M} = \mathrm{diag}(\mathbf{M}_1, \mathbf{M}_2, ..., \mathbf{M}_7) \tag{2.59}$$

Substituting eqs. 2.43, 2.45 and 2.46 into eq. 2.55, the equation of motion explicitly in terms of actuated joint accelerations was defined as follows:

$$\mathbf{T}^T\mathbf{M}\mathbf{T}\ddot{\mathbf{q}}_a = \tau(t) - \mathbf{T}^T\mathbf{M}\dot{\mathbf{T}}\dot{\mathbf{q}}_a \tag{2.60}$$

where t denotes time and $\dot{\mathbf{T}}$ is the derivative of matrix $\mathbf{T}$, defined as:

$$\dot{\mathbf{T}} = \dot{\mathbf{K}}_a - \dot{\mathbf{K}}_u\mathbf{J}_u^{-1}\mathbf{J}_a + \mathbf{K}_u\mathbf{J}_u^{-1}\dot{\mathbf{J}}_u\mathbf{J}_u^{-1}\mathbf{J}_a - \mathbf{K}_u\mathbf{J}_u^{-1}\dot{\mathbf{J}}_a \tag{2.61}$$

where:

$$\mathbf{K}_a = \frac{d}{dt}\left(\frac{\partial \mathbf{t}(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_a}\right), \qquad \mathbf{K}_u = \frac{d}{dt}\left(\frac{\partial \mathbf{t}(\mathbf{q},\dot{\mathbf{q}})}{\partial \dot{\mathbf{q}}_u}\right) \tag{2.62}$$

$$\mathbf{J}_a = \frac{d}{dt}\left(\frac{\partial \phi(\mathbf{q})}{\partial \mathbf{q}_a}\right), \qquad \mathbf{J}_u = \frac{d}{dt}\left(\frac{\partial \phi(\mathbf{q})}{\partial \mathbf{q}_u}\right) \tag{2.63}$$

Equation 2.60 is a typical dynamics model suitable for direct dynamics. The term which appears on the left hand side, $\mathbf{T}^T\mathbf{M}\mathbf{T}\ddot{\mathbf{q}}_a$, consists of the inertia torques which are independent of joint velocities. The first term on the right hand side, $\tau(t)$, consists of the driving torques, which are given in the direct dynamic problem. The second term, $\mathbf{T^T M \dot{T} \dot{q}_n}$, consists of the *centrifugal* and *Coriolis* torques, which are nonlinear in joint displacements and velocities, but independent from joint accelerations.

2.4 Singularities

When studying the kinematics of mechanical systems we often have to face the problem of singular configurations.

Algebraically, a singularity is defined as a special configuration in which the Jacobian matrices involved become rank deficient while, geometrically, a singularity is observed when the manipulator loses or gains extra degrees of freedom.

In the planar three degree-of-freedom parallel manipulator under study, three types of singularities occur.

2.4.1 First type of singularity

The first type corresponds to the limit of the workspace, i.e, the determinant of the Jacobian matrix tends to infinity. This condition is encountered here when one of the denominators involved in the expression of the Jacobian tends to zero.

If we define $\mathrm{K} = \mathbf{J}^{-1}$ then:

$$K = diag(d_1\,, d_2\,, d_3) \tag{2.64}$$

Also:

$$d_i \;=\; 2l_1\,(x_{2i}^2 \;+\; y_{2i}^2)^{\frac{3}{2}}\; \sin\psi_i \tag{2.65}$$

So when: $\qquad \mathbf{J} \to \infty \qquad\qquad \mathrm{K} = 0$

it is clear that this corresponds to:

$$\mathrm{d}_i = 0 \qquad\qquad \mathrm{i} = 1, 2, 3$$

which, from eq. 2.65 leads to:

$$\sin \psi_i = 0 \qquad \text{i} = 1, 2, 3$$

This type of configuration is reached whenever the links l_1 and l_2 of one of the legs are aligned. The limit of the workspace is defined by the set of points for which the quadratic equation of the inverse kinematics will lead to only one solution. This leads to the following condition in eq. 2.2:

$$\psi_i = \pm n\pi \qquad n = 0, 1, 2 \qquad i = 1, 2, 3 \tag{2.66}$$

Equation 2.66 is equivalent to $\sin \psi_i = 0$. In this type of configuration the i^{th} leg is fully extended or folded thus the set of velocities cannot be produced.

2.4.2 Second type of singularity

This type of singularity is located inside the workspace and occurs when the determinant of the Jacobian matrix tends to zero. Here the motor rates are not independent anymore and there exists a set of Cartesian velocities $\dot{\mathbf{c}}$ which are mapped into the zero vector by $\mathbf{J}$. These Cartesian velocities are then possible even when the rates of all motors are zero. These configurations can be inferred from the Jacobian matrix of the manipulator by imposing the linear independence of the columns of matrix $\mathbf{J}$, i.e:

$$k_1\, a_i + k\, b_i + k_3\, c_i = 0 \qquad i = 1, 2, 3 \tag{2.67}$$

For some real values of k_1, k_2, k_3 for which

$$|\mathbf{k}| \neq 0 \tag{2.68}$$

where:

$$\mathbf{k} = [k_1,\ k_2,\ k_3]^T \tag{2.69}$$

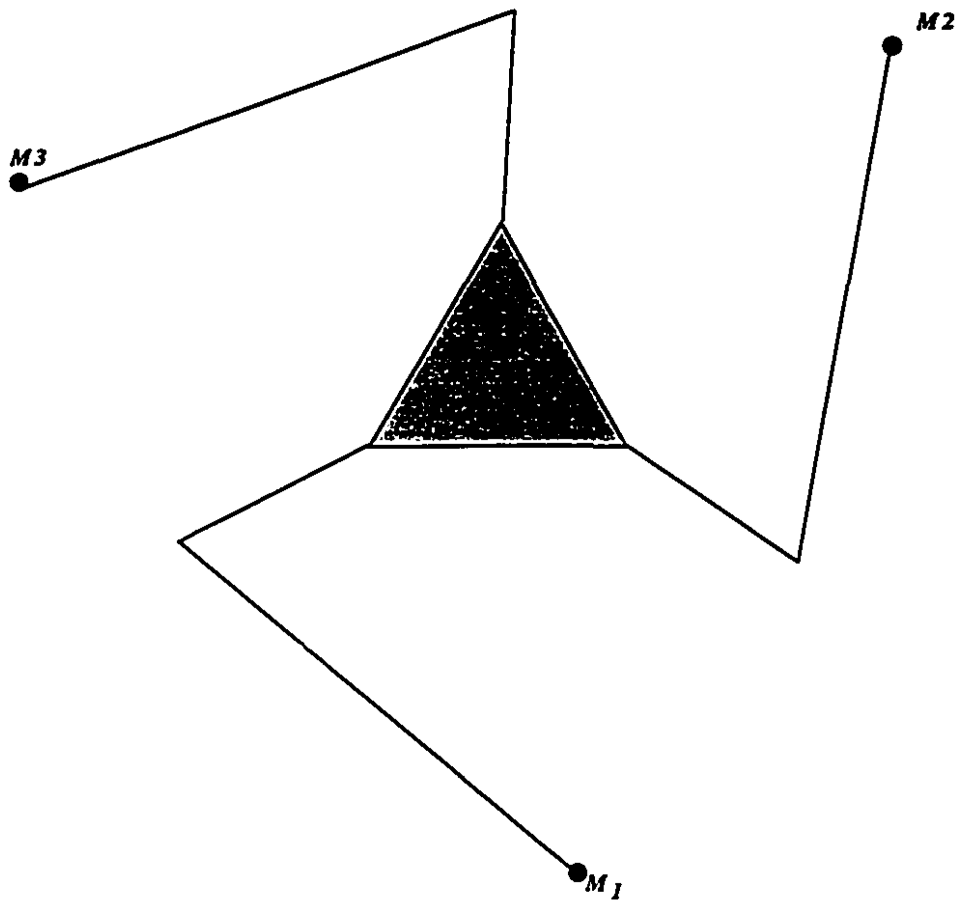


Figure 2.4: An example of the second type of singularity that occurs in the 3-DOF parallel manipulator

Examining eq. 2.67 and eqs. 2.29 - 2.32, we can see that two cases that satisfy the condition given by 2.67 can be identified:

1. This case is obtained when the lines along each of the three links of l_2 intersect at the centroid of the end-effector. In this case we have:

$$c_1 = c_2 = c_3 \tag{2.70}$$

   So equation 2.67 is satisfied if $k_1 = k_2 = 0$. A configuration of this type is shown in Fig. 2.4.

2. This case is obtained when the three links of length l_2 are parallel.

Define a set of two-dimensional vectors:

$$\mathbf{v_i} = [a_i, b_i]^T \tag{2.71}$$

where $\mathbf{v}_i$ is a vector along the two joint centers of the links of length l_2. When the three links of length l_2 are parallel, we have:

$$\mathbf{v}_1 = \pm\ \mathbf{v}_2 = \pm\, \mathbf{v}_3$$

and the second column of the Jacobian matrix **J** is a multiple of the first one.

2.4.3 Third type of singularity

This type of singularity is characterized by the indeterminacy of the Jacobian matrix. In other words, some of the quantities involved in the expression of matrix **J** take on the form $\frac{0}{0}$. According to Gosselin [8], [9], these singularities are architecture dependent and occur when choosing certain lengths for links one, two or three. Recently, however, Daniali [11] showed conclusively that the three types of singularities are quite independent of architecture.

This type of singularity was avoided in the design of the manipulator's prototype, thus it is of no interest in our study.

A brief overview of the three degree-of-freedom parallel manipulator under investigation was presented in this section. In the next chapter, the design of a prototype of such a manipulator shall be detailed.

Chapter 3 Designing the manipulator

3.1 Introduction

In this chapter, the design of the prototype of a three degree of freedom parallel manipulator will be presented. This prototype has been designed at the McGill Research Centre for Intelligent Machines and has been constructed at the Machine Tool Laboratory of the Mechanical Engineering Department of McGill University.

Mechanical design here is understood as the decision-making process involved in tasks such as:

- The material selection.
- The structural design of the robot, consisting of the design of mechanical linkages and joints capable of various movements.
- The selection of the power unit(s) that will drive the robot which can be hydraulic, pneumatic, electrical, or their combination, with or without mechanical transmissions.
- The control system selection or design, which can be of fixed or servo type. Robots with fixed control systems have fixed mechanical stops, limit switches, etc., for positioning and informing the controller. Servo-controlled robots can be either point-to-point (PTP), where specified point coordinates are under control and not the path between them, or continuous path (CP) controlled, thus achieving a smooth transition between the critical points.

The design philosophy of robots has been extensively discussed in the literature, i.e., Rodenacker [46], Pahl and Beitz [44]. Furthermore mechanical design of different robotic manipulators has been examined an analyzed by, e.g., Williams [52], who worked on the DIESTRO manipulator and Angeles and López-Cajún [4]. In this thesis, the author describes the final interaction of the mechanical design cycle for the three degree-of-freedom parallel manipulator.

It was desired to design each joint of the manipulator for a maximum rate of 2 rad/sec and a maximum acceleration of 9 rad/sec^2. As the manipulator is a prototype designed for experimental purposes, the maximum payload that the end-effector could handle was decided to be 100 g.

The volume and shape of the workspace are very important for applications since they determine the capabilities of the robot. These issues will not be discussed here as they are described and determined in detail by Gosselin [8].

It was however desired that the prototype would have a non-vanishing workspace for every angle ϕ, thus it should satisfy Gosselin's condition equations which are:

$$3(l_1 + l_2)^2 \geq (\sqrt{3}l_3 + 1)^2 \qquad (3.1)$$

and

$$3(l_1 - l_2)^2 \leq (\sqrt{3}l_3 - 1)^2 \qquad (3.2)$$

This was accomplished be choosing the right link lengths during the design.

The manipulator presently consists of three motors mounted on a base, known as the base motors. During the design the probability of further improvements was also considered. In that spirit three additional motors were also selected, called auxiliary motors. These motors were not used in this study however they were considered during the design of the prototype. They will be mounted on the unactuated joints of the end-effector of the manipulator and help to pass through singularities during the control.

As in the case of the DIESTRO design, Williams [52], and the UTAH/M.I.T hand, Jacobsen et al.[27], much of the design of the prototype is based on intuition. Many areas in robot operation are under active research. For instance, flexibility effects in manipulator design and inertia loading, Atkeson et al.[5], were not rigorously addressed by this author. However decisions regarding these subjects were made based on sound engineering judgement. Modifications will be made to the robot based on

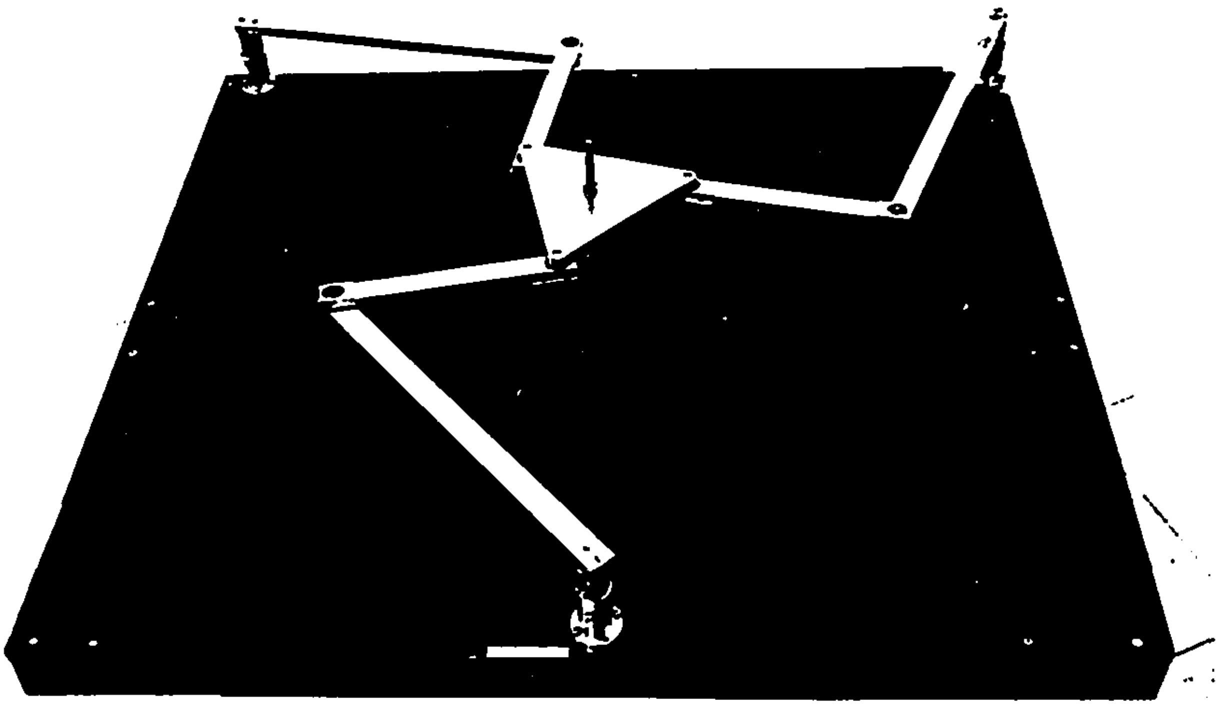


Figure 3.1: The three degree-of-freedom Parallel Manipulator prototype.

operation. A picture of the three degree-of-freedom parallel manipulator prototype is presented in Fig. 3.1.

3.2 Structural material selection

In manipulator structures, the largest deflections are induced by inertia forces. Therefor the stiffness-to-weight ratio of a link is very important, Rivin [45]. Having this in mind, the structural material to be selected must be stiff and light, so that the manipulators static deflection is reduced and its structural natural frequency is increased.

Rivin [45] discusses a number of new materials for the construction of manipulators that were not considered in this work because of there high cost, i.e., beryllium, fiber-reinforced materials etc. While ceramics are very light and temperature resistant, they have problems of brittleness, difficulty of manufacturing long links and joining them with metal parts.

Henessey [19] approached the problem of minimizing the weight of a robot using composite materials, which offer further weight reductions for certain key structural elements. The composite materials used were based on a graphite fiber embedded in an epoxy resin matrix. It was estimated that a 10 to 30 % weight reduction was realized. The problem with composite materials is the complex manufacturing process involved.

From the conventional materials the most interesting one which is widely used, is structural aluminum . It is light, stiff, available, inexpensive and highly machinable, which is also a very important criteria of selection.

Structural steel is not as light as aluminum but it has higher stiffness than aluminum and also has the advantages that were mentioned above (for structural aluminum).

The shafts that connect the links of the designed prototype are short and must be very stiff to avoid deflection. Their weight is not so important because the significant amount of the robot weight will be in the links.

From the above, the structural material of choice for the manipulator shafts was structural steel. For the links the material of choice was aluminum, as their weight plays an important role on the manipulator's performance. The most commonly available alloy that has the highest stiffness is 6061 T6. This alloy is available in a variety of shapes and sizes, therefore it was chosen for the construction of the links.

3.3 Actuators

Actuators are the devices that make robots move. There is a number of such devices in the field and we will restrict our attention to **hydraulic, pneumatic** and **electric actuators.** We will examine their advantages and disadvantages and decide which one is most suitable for the designed prototype.

3.3.1 Hydraulic actuators

Hydraulic systems make use of an incompressible fluid, oil, which is forced under pressure into a cylinder. This cylinder has a piston which moves in response to pressure on the fluid. Two kind of hydraulic actuators are available: *rotary* and *prismatic* ones. In robot applications requiring high power these are the actuators of choice and were initially very popular in manipulators.

They can produce enough force to drive joints and do not need a reduction system, as they are generally slow.

These actuators are used for robots that must move large or heavy loads. Such loads rarely need to be placed with extreme accuracy, i.e., paint spraying and gluing. Hydraulic actuators are also safe in explosive environments.

Unfortunately hydraulic systems are highly nonlinear. These nonlinearities make it more difficult to implement sophisticated and delicate control systems, unlike when using electric drives.

In addition hydraulic actuators require a great deal of equipment such as pumps, hoses and servo valves. They also tend to be messy because of fluid leakages from the connections.

This type of actuator was therefore considered unsuitable for the three degree-of-freedom parallel manipulator.

3.3.2 Pneumatic actuators

In pneumatic control systems, a compressible fluid, air, is used to drive a piston. Compressible air can be found in any workshop at low pressure, usually not exceeding

10 bars. It is simple to use and is conveyed through small and flexible pipes. As in the case of hydraulic actuators, an electrical signal controls a valve which in turn controls the flow to a cylinder. Because the fluid is non-corroding, problems of airtightness are not so important therefore low cost components can be used. The power-to-weight ratio lies between those of electrical and hydraulic systems.

Pneumatic actuators are often used for bang-bang control. They are used in simple robots and were also adapted for the UTAH/M.I.T dextrous hand, Jacobsen et al. [27].

Pneumatic actuators have the following advantages:

1. High speed and relatively high power-to-weight ratio.

2. Very low cost.

3. Simplicity of control.

4. Noncontamination of workspace (unlike an oil leak, a leak in a pneumatic system causes no mess).

There are however some disadvantages which did not suggest their selection for the manipulators drive. Basically their dynamic performance is poor in comparison with electrical servos. This poor performance is attributed to the compressibility of the fluid and to the sluggish time delay.

Operation is often noisy, especially when the compressed air is freely released into the atmosphere. In addition the system always contains a certain amount of water vapor and any resulting condensation can be difficult to eliminate and could cause damage.

3.3.3 Electrical actuators.

Electric actuators are the most popular actuators for small to medium sized manipulators. Less than one half of all commercial robots are driven by electrical energy. There are many advantages in their use:

1. The necessary energy is delivered easily in a convenient form.

2. Control is accurate, uniformly reliable and easy.

3. There are no problems of leakage or pollution.

The only disadvantage here is the weight factor, but recent improvements, as the use of rare earth magnets, have helped the torque-to-weight and torque-to-volume ratios increase.

As a result of these recent improvements, a number of motors now have much better rotational characteristics, at low speeds, that leads to even more accurate positioning. In addition electric motors are widely available in the market, in all sizes and types.

Compared with hydraulic and pneumatic actuators and having in mind that the designed manipulator requires strength and accuracy, it was concluded that electric motors best served the manipulator's power requirements.

3.3.4 Electric motor criteria selection

Theoretically any type of motor could be used, but generally only direct current motors and stepping motors are used in robotics. These were the motors that were considered.

Synchronous motors have not been used widely and will not be considered because of the difficulty in controlling them.

Three main motors, mounted on the base of the manipulator and three auxiliary ones, mounted on the plate that plays the role of the end-effector were required.

It is desirable that these motors:

- Have a large torque-to-weight ratio.
- Be capable of variable speeds.
- Be able to function at low speed and have smooth low speed rotation.
- Have smooth acceleration and deceleration.

- Have minimum rotor inertia.

In addition the three auxiliary motors must be lightweight to avoid deflection of the end-effector due to the motor weight.

3.3.5 DC Motors

DC motor makes use of the fact that a wire carrying a current in a magnetic field experiences a force. In a DC motor, the windings wrapped around a rotating armature carry the current. An arrangement of commutator segments and brushes ensures that the DC current is always in the same direction relative to the magnetic field, thus resulting in a constant force direction, i.e., torque.

The principal variation among different types of DC motors lies in the mechanism used to develop the magnetic field. As an example, in a permanent magnet DC motor, which is most used in robotics, the field is developed by permanent magnets.

Direct current motors have the important advantage of providing torque that is virtually independent of the position and speed of the motor, depending only on the field coils and armatures. If the field coil is replaced by a magnet, torque is proportional to the current in the armature and speed depends only on supply voltage. The DC motor cannot be used in positional servocontrolling without the following accessories:

1. Positional sensors.
2. Possibly tachometer generator.
3. Possibly step-down gears.

There are two large families of DC motors, the integral horse power types having power ratings of one horse power or more, and the fractional horse power motors, with power ratings of less than one horse power. Fractional horse power DC motors can be subdivided into self-exited and permanent magnet motors. Self-exited motors utilize electromagnets which are energized in conjunction with the armature, to generate the stator magnetic field, whereas the permanent magnet type requires no external power in the stator structure.

PM motors offer several advantages. Perhaps the most obvious one is that electrical power need not be supplied to generate the stator magnetic flux. Since the conversion of electrical power to mechanical power takes place in the armature winding, the power supply to the field winding results mostly in an I^2R loss (heat loss) in the winding itself. The PM motor thus simplifies power supply requirements, while at the same time it requires less cooling.

Another benefit of the PM motor is a reduced frame size for a given output power. Because of high coercive strength of permanent magnets their radial dimension is typically one fourth of that of the wound-field motor for a given air gap.

The significant advantages of PM motors over wound-field types are summarized as follows:

- Linear torque speed characteristics.
- High stall (accelerating) torque.
- No need for electrical power to generate the magnetic flux.
- A smaller frame and lighter motor for a given output power.

Two types of permanent magnet motors were considered in our motor selection : Stepper motors and DC torque motors.

Stepper Motors

The stepper motor is a device which translates electrical pulses into mechanical movements. The output shaft rotates through a fixed angular rotation for each incoming pulse or excitation. These pulses are controlled by a computer or a programmable controller which sends commands to an indexer which sends pulses to a drive. The driver then sends current to the stepper motor. By counting these pulses the position and speed of stepper motors can be found, without the need of positional or velocity

MOTOR CHARACTERISTICS	DC MOTORS	STEPPER MOTORS
Peak Torque	HIGH	LOW
Inertia	HIGH	HIGH
Power Rate	VERY HIGH	HIGH
Torque - speed Curve	LINEAR	NONLINEAR
Power - to - weight Ratio	HIGH	LOW

Table 3.1: Comparison of Motor Characteristics

feedback. Three main disadvantages were considered and this type of motors was therefore not chosen:

- Lack of availability of a variety of step angles. Step motors have fixed step angles, which may not always fit given applications. A change in step angle requires another motor.
- These motors are in concept synchronous motors and this implies certain disadvantages, i.e. torque depends on position, pull out torque and stability.
- The torque-to-weight ratio is low when compared with a continuous DC motor.

If we compare the peak torque of a continuous DC motor with the peak torque of a stepper motor, we will see that the peak torque of a DC motor is higher. Both peak torque and torque ripple are important in robotics specially for low speed robot operations as in the case under study.

Summarizing the advantages of the DC motors, we see that the DC motor is lighter than the stepper motor, the peak torque is higher and the torque ripple at low speed is much smaller (Table 3.1). For these reasons continuous DC motors were the choice for our manipulator.

3.3.6 More details on DC motors

In their effort to improve the performance of servo mechanism actuators, designers have continually sought to produce motors with higher torque and lower moment of inertia. As a result some basic trends in the design of DC low inertia motors appeared. One was the small diameter iron core armature, operating in very high magnetic flux levels. The more refined style of this design featured conductors bonded to the armature core surface : the so-called *slotless armature* design. This design has great mechanical strength, high torsional rigidity and is reasonably efficient; but it suffers from two principal drawbacks which disqualified them for our application:

- Inductance. As a result its electrical time constant is high; and
- The armature resistance and the torque constant are small, requiring very high armature currents and low voltage power supplies, making the transistor amplifier unnecessarily complex and expensive.

The other design trend, which has since gained almost universal acceptance, is the *moving coil* concept. The armature structure here, is supported mainly by non-magnetic materials and the active conductors are therefore moving in an air gap with a high magnetic flux density.

Motor demagnetization commonly occurs when the motors in an application require a *plug reversal*. A plug reversal means a sudden switching of the polarity of the applied voltage to achieve a rapid reversal of motor rotation, or perhaps a quick stop. When a motor is plug reversed, it sees not only the voltage applied to its terminals, but also the armature generated voltage as voltage sources. As motor reversal occurs frequently in robotics a motor must be chosen with little danger of demagnetization. Moving coil motors are generally not as susceptible to demagnetizing peak currents as iron core motors because of their significant air gap and special pole shoe design.

The moving coil motor is characterized by the absence of rotating iron on the armature. Since the conductors are operating in an air gap the armature features low electrical time constant- typically less than 0.1 ms. The absence of iron also brings about another benefit: there is no reluctance torque effect, i.e, no magnetic cogging, and the motor weight is significantly reduced.

Summarizing we can say that the DC brush commutated motor characteristics are:

- Motor torque is a linear function of motor current.
- Motor speed is a linear function of load torque when operated at a constant voltage.
- The no load speed and stall torque are directly proportional to the applied voltage.
- The motor direction of rotation is reversible by reversing the power supply polarity.
- The motors are capable of operating over a wide range of voltage speed and torque.

Brushless motors were also considered for the three degree-of-freedom parallel manipulator. The problem was the limited range of brushless motors on the market.

The designed prototype, as we mentioned in the beginning of this chapter, will very often be operating at relatively low speeds. As a result preference will be given to motors with rated lower no-load speeds, during the motor selection. In addition gearheads will be used with the selected motors so that the appropriate speed and torque values can be obtained.

3.4 Sensors

Robots need a wide range of sensors to obtain information. Sensors in general provide feedback to the controller containing information about the robot's action or its environment.

The selected sensors must be capable of detecting rotary motion and providing both velocity and position control. The sensors considered were:

- Potentiometers.
- Optical encoders and
- Resolvers.

A potentiometer is a rotational, mechanically variable resistor. Shaft rotation moves a contact across a resistor producing a voltage proportional to the angular position. Potentiometers are relatively cheap and simple and can have almost infinite resolution (Fig. 3.2); their accuracy is limited more by non-linearity and noise.

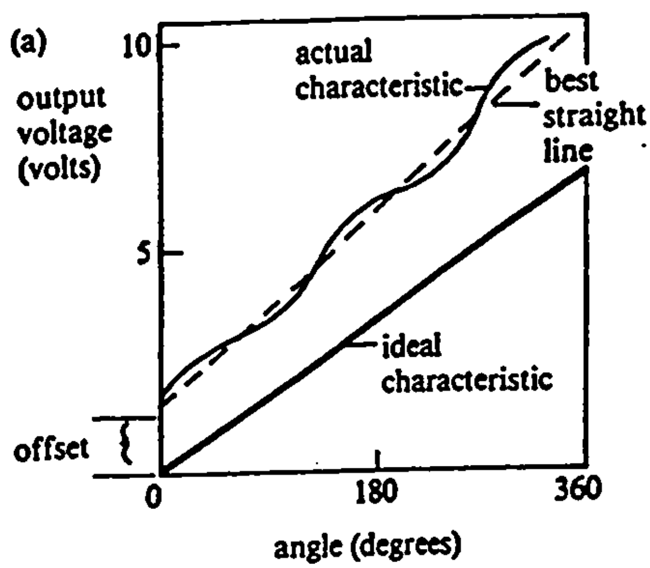


Figure 3.2: Characteristics of a potentiometer (infinite resolution)

Potentiometers are rarely used in robots. Apart from their linearity not being good enough for the highest accuracies, they need more individual calibration than other transducers. An added weakness is that the output voltage is proportional to the supply voltage, which must therefore be extremely well regulated.

Optical encoders are digital devices which use a photoelectric method to detect the movement of an internal disk. These encoders offer numerous advantages. They

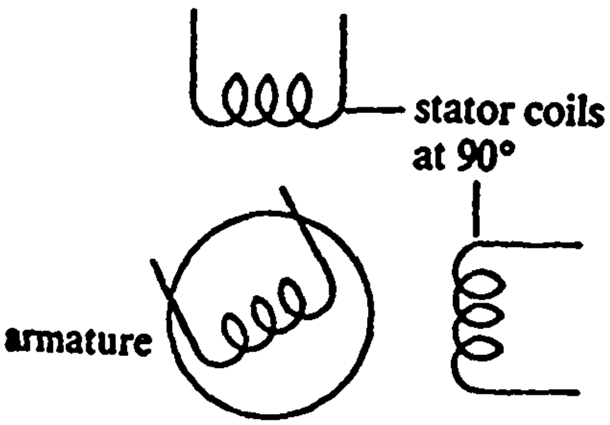


Figure 3.3: Principal operation of resolvers

are very light, have very good accuracy and in many motor types they are directly mounted, thus occupy less space than other devices.

Resolvers are transformers having two fixed windings and a rotor winding; Fig 3.3 shows the basic form. The rotor coil carries an alternating current at a few kilohertz. The amplitude of the alternating voltage induced in the static coils depends on the rotor angle. The amplitude of the voltages in the two coils is proportional to $\sin\theta$ and $\cos\theta$, respectively, where θ is the shaft angle. The signal is harder to convert into a digital measure of angle than that of other transducers, but resolvers are robust and the signal is resistant to interference. As with potentiometers, linearity is more of a limit on accuracy than resolution. Available resolvers are also heavier than optical encoders. An additional disadvantage of potentiometers and resolvers is that their measurements are analogue and susceptible to gear backlash and shaft vibration as they are usually mounted to the link shaft and not directly to the motor shaft.

Greater resolution, less weight and the ability to be mounted directly to the motor shaft were the advantages that lead us to select optical encoders for the motion control of the three degree-of-freedom parallel manipulator. In addition optical shaft encoders are non-contact thus they do not burden the system with added inertia and friction and there high encoding speed offers high noise immunity.

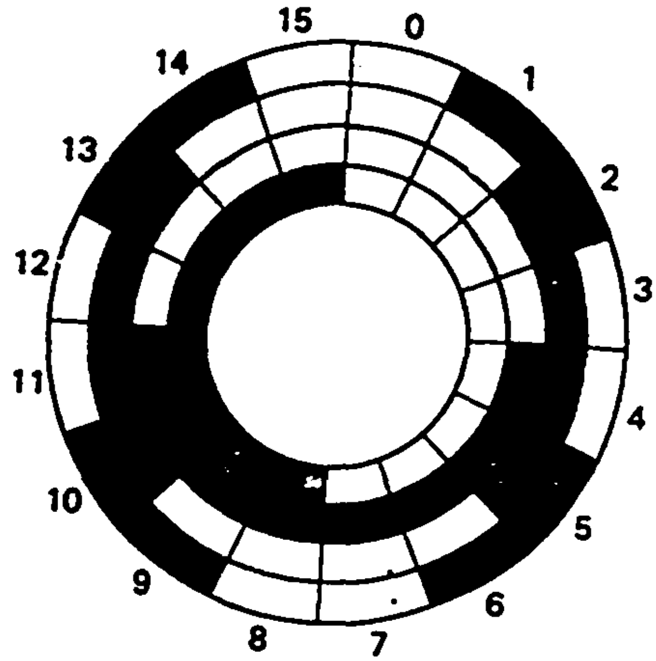


Figure 3.4: Absolute shaft encoder using Gray code

3.4.1 Optical encoders

They are two basic types of optical shaft encoders, absolute and incremental ones.

An absolute encoder gives an angular reading that corresponds to its specific position. Since encoders are sensitive to noise, absolute encoders often use a special output code called the Gray code. This code (Fig 3.4) has been designed to ensure that only a single bit changes value for each increment of the absolute encoder. Thus absolute encoders are less susceptible to errors in determining position. The drawback of absolute encoders is that they need a photocell and pattern ring on the disk for every bit in the word, so a 12 or 14 bit encoder is a complex device and it also needs a cable with many conductors and large connectors. In addition compared to incremental encoders they are heavier and much more expensive.

Incremental encoders are more widely used in robot technology. There are two types of them available: one-channel and two-channel encoders.

One-channel incremental encoders have one track of disk slots, making these encoders sufficient for velocity control but not for position control. Although they are available in small sizes, when compared with two-channel encoders their resolution is significantly less. In addition single output encoders cannot determine the direction of rotation.

Therefore most robots use two-channel encoders, also known as biphase encoders. The biphase encoder provides control signals to a counter, which records actual displacement. These count-up and count-down signals are obtained by decoding the transitions of two input signals, S_1 and S_0, produced by two LED/phototransistor pairs which straddle a disk, consisting of a single row of holes or slots on the circumference as shown in Fig 3.5. The two sensors, circumferentially separated by $2N\Pi \pm \frac{\Pi}{2}$ cycles of the land/gap sequence around the disk's periphery, yield S_1 and S_0 outputs shown in Fig. 3.6. The transition of the signals S_1 and S_0 provide displacement increments with sign, i.e., positive or negative position quanta. A transition from state 0 (00) to state 2 (10) is a forward (clockwise) rotation, while that from state 0 to state 1 (01) is a backward (counterclockwise) displacement.

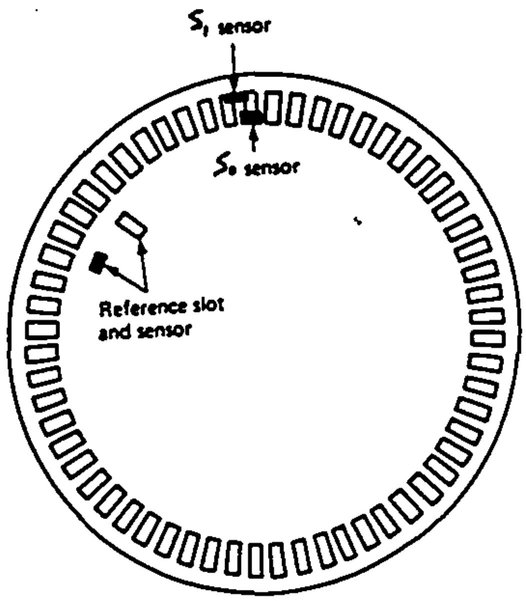
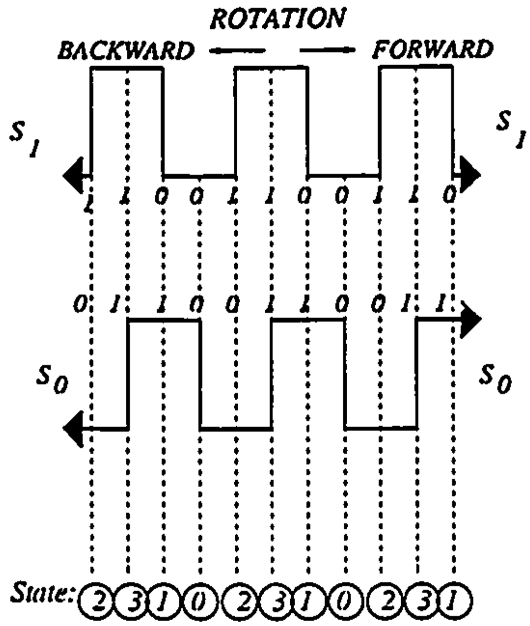


Figure 3.5: Biphase encoder hardware



State	S_1	S_0
0	0	0
1	0	1
2	1	0
3	1	1

Figure 3.6: Biphase encoder output

Since

- High resolution,
- Low weight,
- Low cost and simple implementation,

were desired for the three degree-of-freedom parallel manipulator the two channel incremental encoder was selected. Absolute encoders were discarded because of their weight and there cost while single output incremental encoders were undesirable because they were unable to produce direction information.

3.5 Manufacture

The detailed mechanical designs of each part of the three-degree-of-freedom parallel manipulator prototype, are presented in Appendix A.

In Fig. 3.7 we can see the shear and bending moment diagrams of one of the branches of the manipulator. Due to the symmetry of the design these diagrams are the same for the other branches. Because of the small link lengths and the small

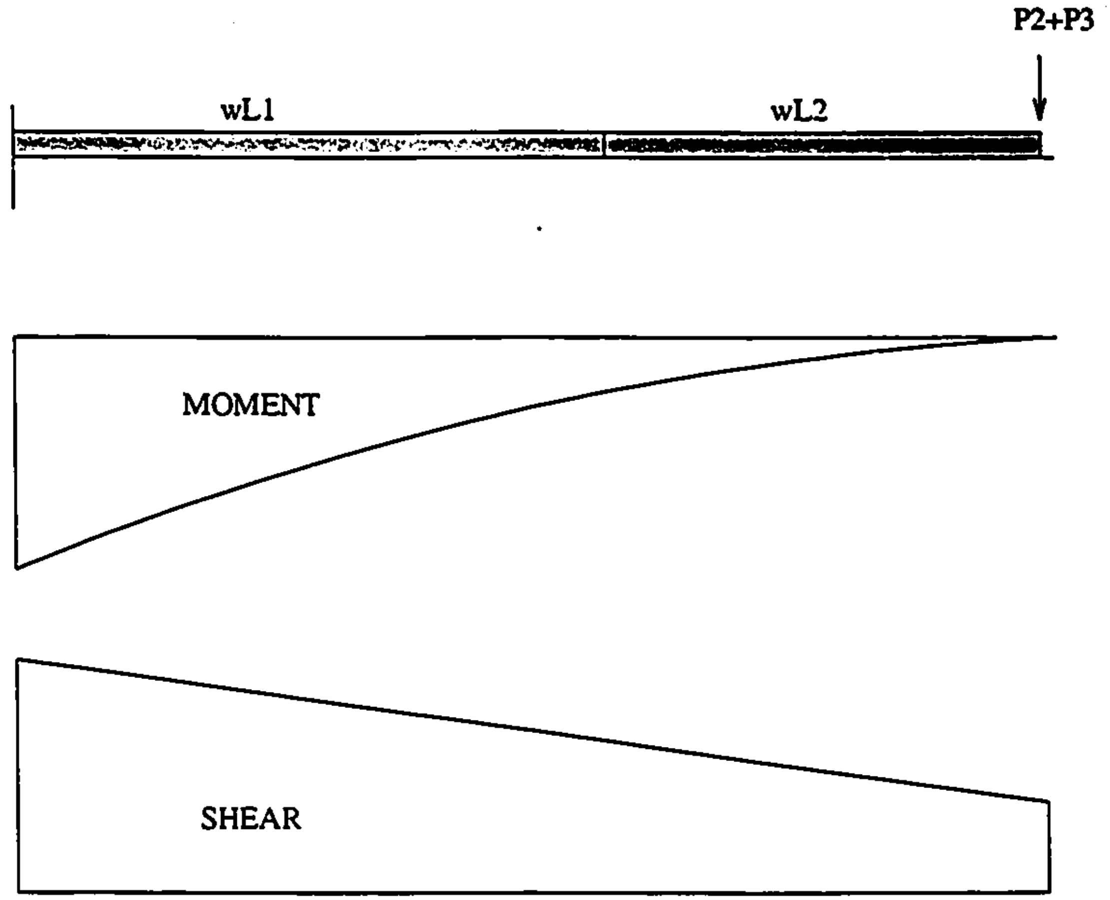


Figure 3.7: Moment and shear diagrams, of one branch of the manipulator

forces that are applied on them we assume, for these diagrams, that link1 and link2 are acting like a uniform cantilever beam with a fixed end.

The calculations of the bending moment, the shear forces , the bending stress and the deflection were based on the principle of superposition and the following formulas were used:

In the following formulas the notation that has been used is:

M_{imax} : The maximum bending moment.

V_{imax} : The maximum vertical shear.

S_{ib} : The bending stress.

I : The moment of inertia of the cross-sectional area of the beam.

c : The distance from the neutral axis to the most distant point of the cross section.

y_{imax} : The maximum deflection of the beam.

E : The modulus of Elasticity.

P_2 : Load caused from mass of the auxiliary motor.

P_3 : Load caused from mass of payload divided by three.

L : Total length of beam equal to $L_1 + L_2$.

For the calculation of the total maximum bending stress and deflection the following procedure was used:

1. For the distributed load:

$$S_{b1} = \frac{M_1 c}{I} \tag{3.3}$$

where:

$$M_{1max} = -\frac{WL}{2} \tag{3.4}$$

$$W = wL = beam\ \ weight \times gravity\ \ acceleration \times total\ \ lenght\ \ of\ \ link \tag{3.5}$$

and:

$$c = \frac{h}{2} \qquad\qquad I = \frac{bh^3}{12} \tag{3.6}$$

For the deflection in this case:

$$y_{1max} = \frac{-1}{8} \times \frac{WL^3}{EI} \tag{3.7}$$

2. For the concentrated load (P = $P_2 + P_3$):

$$S_{b2} = \frac{M_2 c}{I} \tag{3.8}$$

where:

$$M_{2max} = -P \times L \tag{3.9}$$

and c and I are as in the first case.

For the deflection here:

$$y_{2max} = \frac{-1}{3} \times \frac{PL}{3EI} \tag{3.10}$$

Finally for total stress and deflection we have:

$$S_b = S_{b1} + S_{b2} \tag{3.11}$$

and:

$$y_{max} = y_{1max} + y_{2max} \tag{3.12}$$

The values found using the above equations for the constructed manipulator were substantially below the proportional limits for aluminum, therefore the maximum deflection observed is acceptable.

The three base shafts that connect the main drive motors with the first links carry all the thrust and axial forces that are produced. This was the reason that a special support was designed for these shafts. Thrust and needle bearings were placed in the designed support to minimize the aforementioned forces.

All links were connected to each other with angular contact bearings to achieve the proper link rotations.

Due to the proper material selection, construction and size of the prototype, the bending moment does not cause any deflection in our prototype.

Data from a number of companies were evaluated for compatibility with the manipulators requirements (Appendix B). Maxon Precision Motors Inc. (Maxon Precision Motors Inc 1990 [39]) offered the motors, gearheads and encoders that were most suitable for our prototype and that were finally used.

The gearheads used with the motors are planetary gearheads, where three or more gears carry the torque instead of one as in spur gearheads. For this reason they have higher torque ratings than spur gearheads of similar physical sizes. In other words the planetary gearhead will lead to less shaft deflection and lighter motor-gearhead units.

Details of the design, component selection and construction of the three degree-of-freedom parallel manipulator were presented in this chapter. In the next chapter, Chapter 4, the control of the constructed prototype will be discussed and further analyzed.

Chapter 4 Controlling the parallel manipulator

4.1 Introduction

In this chapter the control of the DC motors of the 3 degree of freedom parallel manipulator will be discussed and further analyzed.

The control of this type of mechanism differs fundamentally from that of serial manipulators, hence an application of parallel processing is required. Parallel processing can be defined as a technique for increasing the computation speed for a task, by dividing the algorithm into several sub-tasks and allocating multiple processors to execute these sub-tasks simultaneously. To achieve parallel processing for the case under study a transputer network was introduced and the following steps were taken:

1. Installation of transputer network in a PC,

2. Programming the transputer network,

3. Interface design and implementation,

4. DC motor control hardware design and implementation.

The installation and programming of the transputer network that will be used, has already been described and analyzed by Helmy [18]. The main consideration in this work was the design, implementation and testing of the DC motor control circuits. In addition results from the performed experiments are presented. In conclusion to this chapter, an interface proposal has been made. This proposed interface can be thought of as a first step towards the final interface design which will achieve total bidirectional communication between the manipulator plant and the transputer network.

For the control of DC motors, there are two major categories: feedback or closed-loop control and open-loop control.

A typical control system has the feature that some output quantities are measured and compared with the desired output values, and the system's output is corrected by using the resulting errors from the comparison. This system is called feedback control system. A block diagram of the feedback control system is shown in Fig. 4.1.

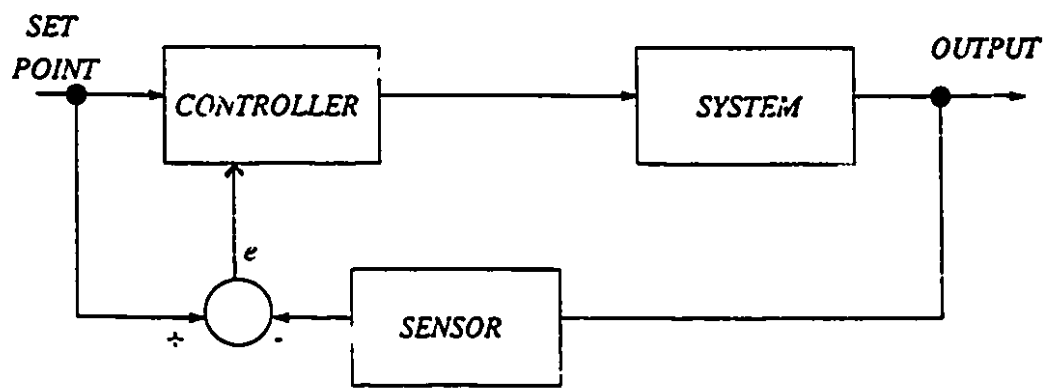


Figure 4.1: Feedback Control System

In some cases, it is also possible to control the system in an open-loop manner, where feedback is not used. Such a kind of system is presented in Fig 4.2

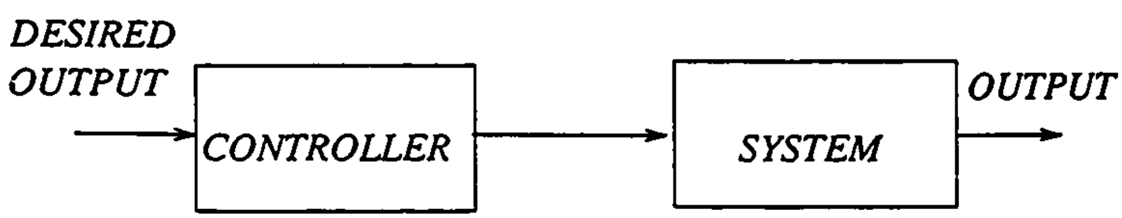


Figure 4.2: Open-loop Control System

Feedback control is widely used in robotic systems. The reason is that, by using feedback, the designer of the robotic system is often able to use inexpensive and inaccurate components, while the system is capable of achieving precise control in the presence of measuring errors and unpredictable disturbances.

4.2 Modelling of a DC motor

The actuator tested and selected for this project was a small permanent magnet (PM) DC motor. In the previous chapter we saw that PM motors posses linear speed-torque characteristics and provide high output torque over a wide range of speeds. The DC motor, along with an appropriate DC amplifier system and either position or velocity feedback, provides the power element in many guidance and control applications.

The general modeling equations of a DC motor are obtained by writing the voltage equation for the armature circuit and the torque equation. The armature has an inductance L_a and a resistance R_a. The velocity of the armature shaft is represented by ω. The electromagnetic torque provides the load torque T_L, the torque lost in friction and windage T_f, the eddy current damping torque T_d which is proportional to speed and the torque necessary to accelerate the inertia T_j.

$$\textbf{VOLTAGE}: \quad V = R_a i + L_a \frac{di}{dt} + V_e \tag{4.1}$$

where V_e is the rotational back e.m.f and is equal with:

$$V_e = K_e \times \omega \tag{4.2}$$

where K_e is the electrical constant of the motor.

At any instant of time, the developed torque, T_g, must be equal and opposite to the sum of the torques necessary to overcome friction, viscous damping, inertia and the load torque. Thus:

$$\textbf{TORQUE}: \quad T_g = T_L + T_f + K_d\omega + J\frac{d\omega}{dt} = K_t \times i \tag{4.3}$$

where $K_d\omega = T_d$ and the inertia torque is represented by the product of the moment of inertia J and the angular acceleration $\frac{d\omega}{dt}$, and J is the total moment of inertia of the rotor and the load referred to the motor shaft. K_t represents the torque constant of the motor and K_d represents the damping coefficient. Here the field flux is constant and unaffected by armature reaction.

Equations 4.1, 4.2 and 4.3 constitute a basic set of equations that model the DC motor and from which transfer functions for the DC motor operated in various modes may be obtained. Taking the Laplace transform of both sides of these equations and rearranging them leads to:

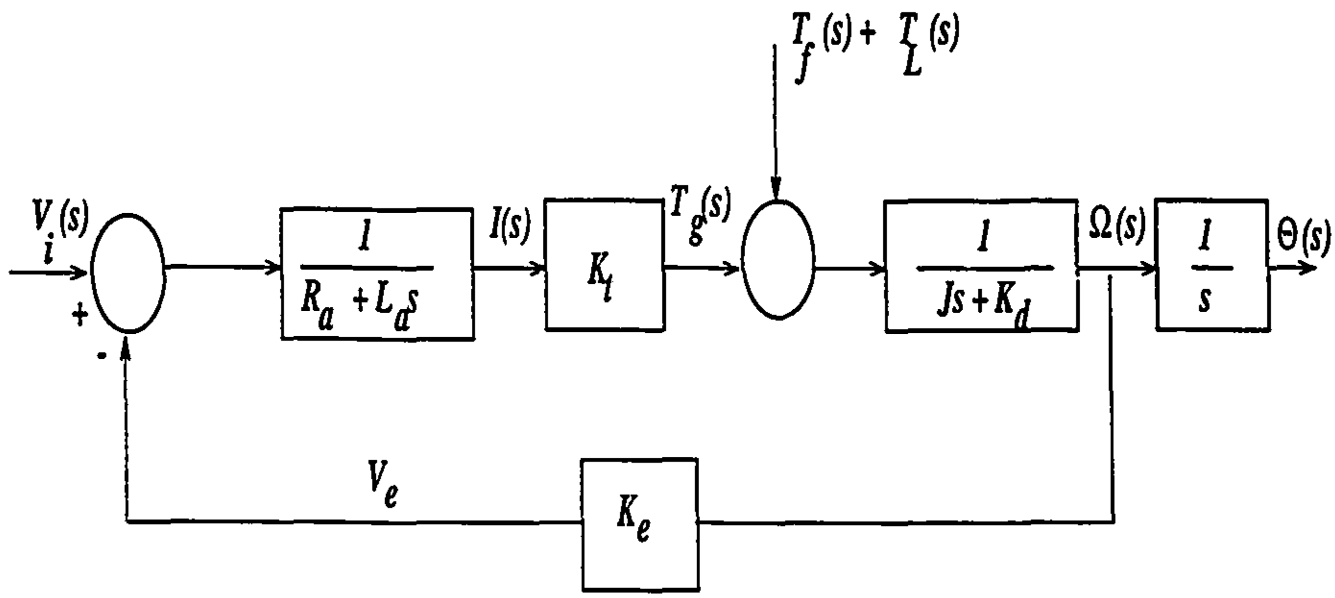


Figure 4.3: Block diagram

$$V(s) - V_e(s) = (R_a + sL_a) \times I(s) \tag{4.4}$$

$$V_e(s) = K_e\Omega(s) \tag{4.5}$$

$$T_g(s) = K_t I(s) \tag{4.6}$$

$$T_g(s) - T_L(s) - T_f(s) = (K_d + sJ)\Omega(s) \tag{4.7}$$

The block diagram representation of these basic equations is shown in Fig. 4.3.

The block diagram of Fig. 4.3 represents a two input system with the output being either the angular velocity ω or the angular position θ, or both. From Fig. 4.3 the output velocity of the system is written as:

$$\Omega(s) = G_1(s)V(s) + G_2(s)[T_f(s) + T_L(s)] \tag{4.8}$$

in which:

$$G_1(s) = \frac{\Omega(s)}{V(s)} \tag{4.9}$$

when $T_f(s) + T_L(s) = 0$ and:

$$G_2(s) = \frac{\Omega(s)}{T_f(s) + T_L(s)} \tag{4.10}$$

when $V(s) = 0$.

The voltage velocity transfer function $G_1(s)$ is:

$$G_1(s) = \frac{\Omega(s)}{V(s)} = \frac{K_t}{(L_a s + R_a)(Js + K_d) + K_t K_e} = \frac{K_m}{\alpha s^2 + \beta s + 1} \tag{4.11}$$

where:

$$K_m = \frac{K_t}{R_a K_d + K_t K_e}$$

$$\alpha = \frac{L_a J}{R_a K_d + K_t K_e}$$

$$\beta = \frac{R_a J + L_a K_d}{R_a K_d + K_t K_e}$$

Equation 4.11 represents the voltage transfer function of the motor, under the assumption that T_f and T_L are zero. This equation can also be expressed as:

$$G_1(s) = \frac{K_t}{R_a K_d(1 + \tau_e)(1 + \tau_m) + K_t K_e} \tag{4.12}$$

where:

$$\tau_e = \frac{L_a}{R_a} = electrical \;\; time \;\; constant$$

$$\tau_m = \frac{J}{K_d} = mechanical \quad time \quad constant$$

If the armature voltage is very small, the electrical time constant can be neglected and eq. 4.11 becomes:

$$G_v(s) = \frac{\Omega(s)}{V(s)} = \frac{K_t}{R_a(Js + K_d) + K_t K_e} = \frac{K_m}{\tau s + 1} \tag{4.13}$$

where:

$$\tau = \frac{R_a J}{R_a K_d + K_t K_e} \tag{4.14}$$

The load torque transfer function G_2 is given by:

$$G_2 = \frac{\Omega(s)}{T_f(s) + T_L(s)} = \frac{-\dfrac{1}{Js + K_d}}{1 + \dfrac{K_t K_e}{(Js + K_d)(L_a s + R_a)}} = \frac{-\frac{R_a}{K_t} K_m [\frac{L_a}{R_a} s + 1]}{\alpha s^2 + \beta s + 1} \tag{4.15}$$

which, if armature inductance is negligible, reduces to:

$$G_L(s) = \frac{\Omega}{T_f(s) + T_L(s)} = \frac{-\frac{R_a}{K_t} K_m}{\tau s + 1} \tag{4.16}$$

This motor must rotate a load, which consists of some inertia J_L and a constant opposing torque T_L, through a given angular trajectory θ_L. The torque/speed ratio of the motor can be modified, with modest additional loss, by coupling the motor shaft through a gear train to the load.

If N is the gear ratio, then motor torque will become:

$$T_g = (J + \frac{1}{N^2} J_L) \frac{d\omega}{dt} + \frac{1}{N} T_L \tag{4.17}$$

and motor angular position becomes:

$$\theta = N \times \theta_L \tag{4.18}$$

Given the parameters of the dc motor and specifications of the load to be driven, the required control voltage can be easily calculated. Further detailed analysis of the DC motor can be found in [30], [13] and [49].

4.3 The L290/L291/L292 DC motor speed/position control system

The DC motors mounted on the base of the manipulator are digitally controlled by the L290/91/92 DC motor speed/position control system. The L290, L291 and L292 together form a complete microprocessor - controlled DC motor servopositioning system that is both fast and accurate. These chips are primarily intended for use with a DC motor and optical encoder in the configuration shown schematically in Fig. 4.4. This system is controlled by a microprocessor, or a microcomputer, which determines the optimum speed profile for each movement and passes appropriate commands to the L291, which contains the system's D/A converter and error amplifiers. The L291 generates a voltage control signal to drive the L292 switchmode driver which powers the motor. An optical encoder on the motor shaft provides signals which are processed by the L290 tachometer converter to produce tacho voltage feedback and position feedback signals for the L291 plus distance/direction feedback signals for the control micro.

The system operates in two modes to achieve high speed and accuracy: closed loop speed control and closed loop position control. The combination of these two modes allows the system to travel rapidly towards the target position then stop precisely without oscillations.

Initially the system operates in speed control mode. A movement begins when the microcomputer applies a speed demand word to the L291, typically calling for maximum speed. At this instant the motor speed is zero so there is no tacho feedback and the system operates effectively in open loop mode. In this condition a high current

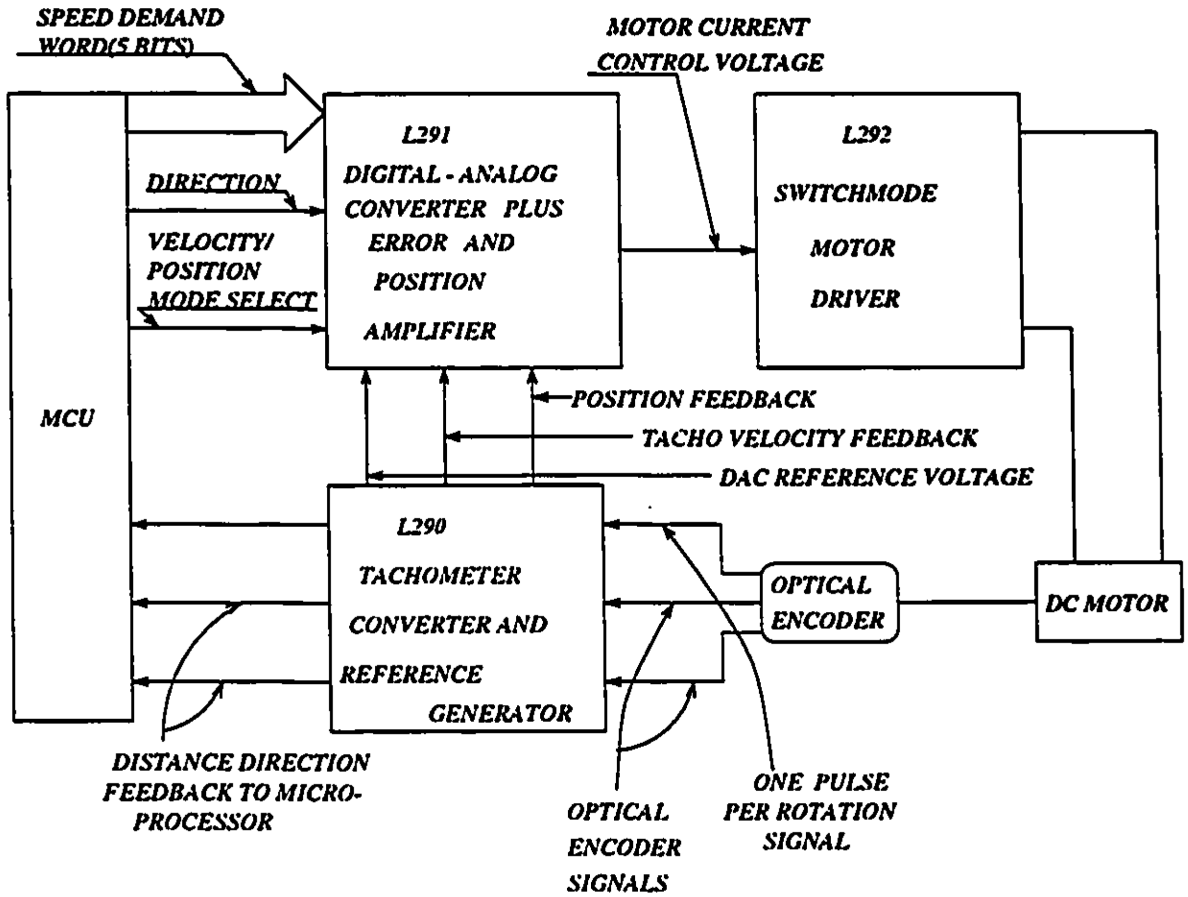


Figure 4.4: The L290/L291/L292 DC motor control system

peak - up to 2A - accelerates the motor rapidly to ensure a fast start.

As the motor accelerates the tacho voltage rises and the system operates in close loop speed mode, moving forwards the target position. The microcomputer, which is monitoring the optical encoder signals (squared by the L290), reduces the speed demand word gradually when the target position is close. Each time the speed demand word is reduced the motor is braked by the speed control loop.

Finally, when the speed code is zero and the target position extremely close, the microcomputer commands the system to switch to position mode. The motor then stops rapidly at the desired position and is held in an electronic detent.

4.3.1 Optical encoder

The optical encoder selected and used in this system was the Maxon 3416 encoder, which has 100 slots. This means that it produces 100 counts per one rotation of the encoder disk.

Light sources and sensors are mounted so that the encoder generates two quasi-sinusoidal signals with a phase difference of ± 90 deg. These signals are refered to as FTA and FTB. The frequency of these signals indicates the speed of rotation and the relative phase difference indicates the direction of rotation.

Both technical data and the external circuit implemented so that the encoder can properly operate are included in appendix C.

4.3.2 The L290 tachometer converter

The L290 tachometer converter processes the the optical encoder signals, FTA and FTB to generate a tachometer voltage, a position signal and feedback signals for the microprocessor. It also generates a reference voltage for the system's D/A converter.

Analytically the tacho generation function can be expressed as:

$$TACHO = \frac{dV_{AB}}{dt} \times \frac{FTA}{|FTA|} - \frac{dV_{AA}}{dt} \times \frac{FTB}{|FTB|} \tag{4.19}$$

In the L290 (block diagram Fig. 4.5) this function is implemented by amplifying FTA and FTB in A1 and A2 to produce V_{AA} and V_{AB}. V_{AA} and V_{AB} are differentiated by external RC networks to give the signal V_{MA} and V_{MB} which are phase shifted and proportional in amplitude to the speed of rotation. V_{MA} and V_{MB} are passed to multipliers, the second input of which are the sign of the other signal before differentiation.

The sign ($\frac{FTA}{|FTA|}$ or $\frac{FTB}{|FTB|}$) is provided by the comparators CS1 and CS2. Finally the multiplier outputs are summed by A3 to give the tacho signal. Fig. 4.6 shows the waveforms for this process.

This seemingly complex approach has three important advantages:

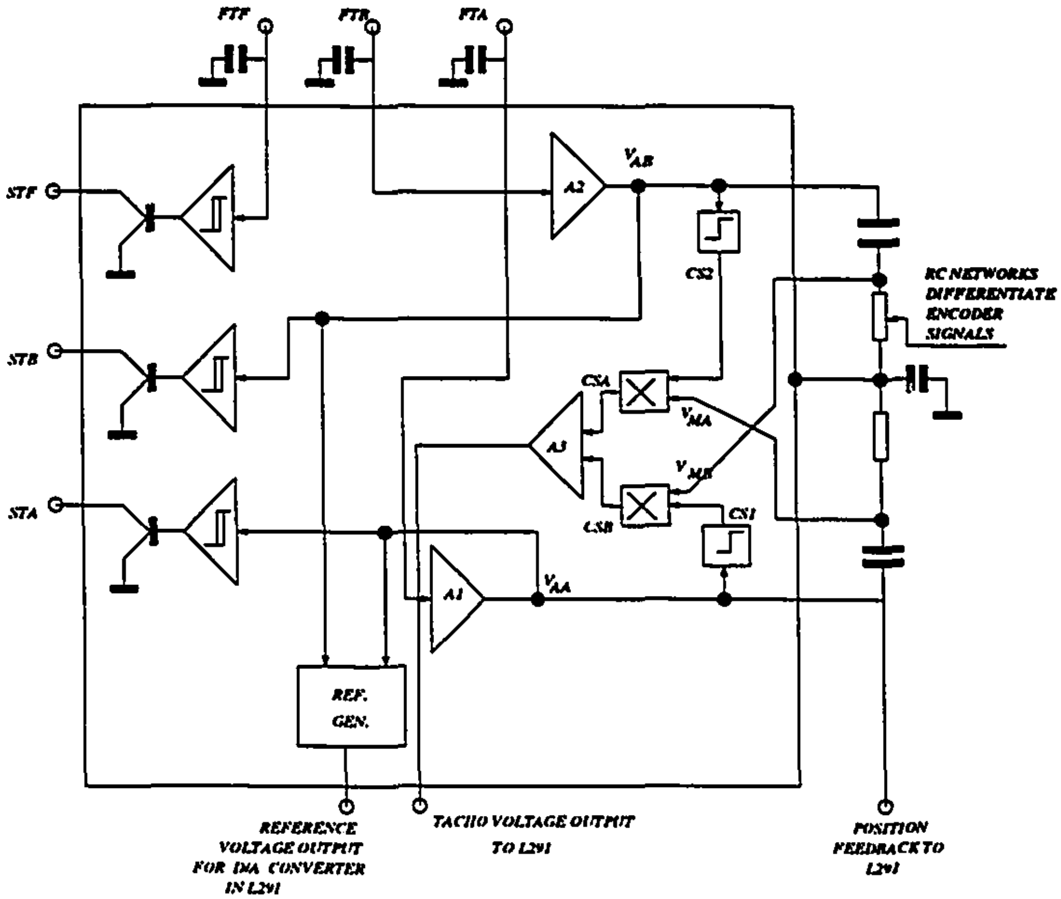


Figure 4.5: L290 Block diagram

- Since the peaks and nulls of CSA and CSB tend to cancel out, the ripple is very small.

- The ripple frequency is the fourth harmonic of the fundamental so it can be filtered easily without limiting the bandwidth of the speed loop.

- It is possible to acquire tacho information much more rapidly, giving a good response time and transient response.

Feedback signals for the microprocessor, STA and STB, are generated by squaring FTA and FTB. Position feedback for the L291 is obtained simply from the output of A1.

The L290 also generates a reference voltage for the L291's D/A converter. This reference is derived from V_{AA} and V_{AB} with the function:

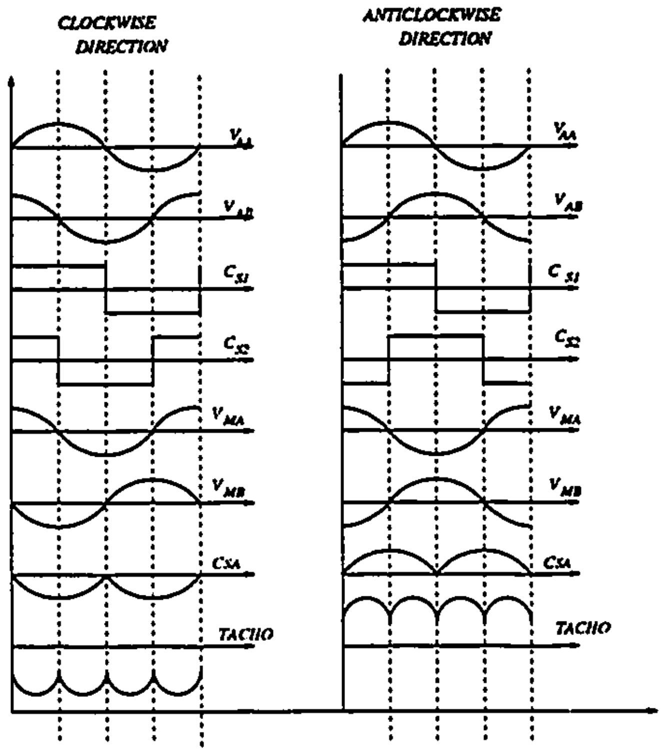


Figure 4.6: The waveforms that illustrate the generation of the tacho voltage in the L290.

$$V_{ref} \equiv |V_{AA}| + |V_{AB}| \tag{4.20}$$

Since the tacho voltage is also derived from V_{AA} and V_{AB} it follows that the system is self compensating and can tolerate variations in input levels, temperature changes and component aging with no deterioration of performance.

4.3.3 The L291 D/A converter and amplifiers

The L291 shown in Fig. 4.7, links the system to the micro and contains the system's main error amplifier plus a position amplifier which allows independen. adjustment of the characteristics of the position loop.

The L291 contains a 5 bit D/A converter accepting a natural binary code and generating a bipolar output current, the polarity of which depends on the SIGN input. The amplitude of the output current is a multiple of a reference current I_{ref}.

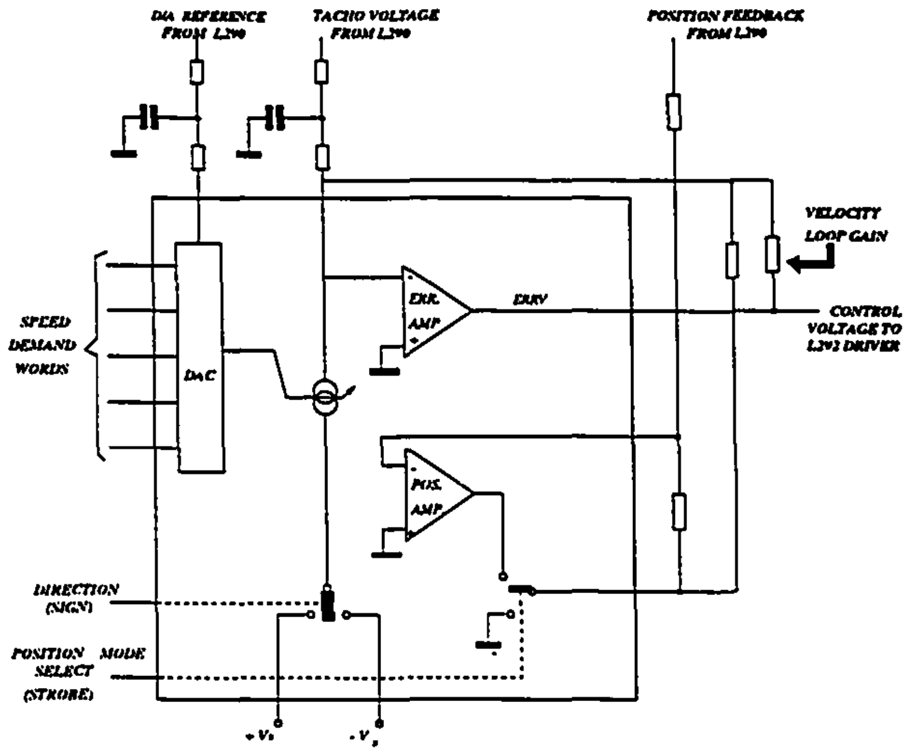


Figure 4.7: The L291 D/A converter and position amplifier

The maximum output current is:

$$I_{FS} = \pm\frac{31}{16} \times I_{ref} \tag{4.21}$$

Table 4.1 shows the value of I_o for different input codes. Note that the input bits are active low .

This D/A converter has a maximum linearity error equal to $\pm\frac{1}{2}$ LSB (or $\pm$ 1.61 % F'JLL SCALE); that guarantees its monotonicity.

The main error amplifier sums the D/A converter output and the tacho signal to produce the motor drive signal ERRV. The position amplifier is provided to allow independent adjustment of the position loop gain characteristics and is switched in/out of circuit to select the mode. The final position mode is actual speed plus position but since the tacho voltage is almost zero when position mode is selected the effect of the speed loop is negligible.

DIGITAL INPUT WORD						OUTPUT CURRENT
SIGN	SC5 MSB	SC4	SC3	SC2	SC1 LSB	I_o
L	L	L	L	L	L	$-\frac{31}{16} I_{ref}$
L	H	H	H	H	L	$-\frac{1}{16} I_{ref}$
X	H	H	H	H	H	0
H	H	H	H	H	L	$+\frac{31}{16} I_{ref}$
H	L	L	L	L	L	$+\frac{31}{16} I_{ref}$

X = DONT CARE
L = LOW
H = HIGH

Table 4.1: Values of output current for different input codes

4.3.4 The L292 switchmode motor driver

The L292 can be considered as a power transconductance amplifier which delivers a motor current proportional to the control voltage (ERRV) from the L291. It drives the motor efficiently in switchmode and incorporates an internal current feedback loop to ensure that the motor current is always proportional to the input control signal. The input signal (see block diagram in Fig. 4.8) is first shifted to produce a unipolar signal (the L292 has a single supply) and passes to the error amplifier where it is summed with the current feedback signal. The resulting error signal is used to modulate the switching pulses that drive the output stage.

External sense resistors monitor the load current, feeding back motor current information to the error amplifier via the current sensing amplifier. The L292 incorporates its own voltage reference and all the functions required for closed-loop current

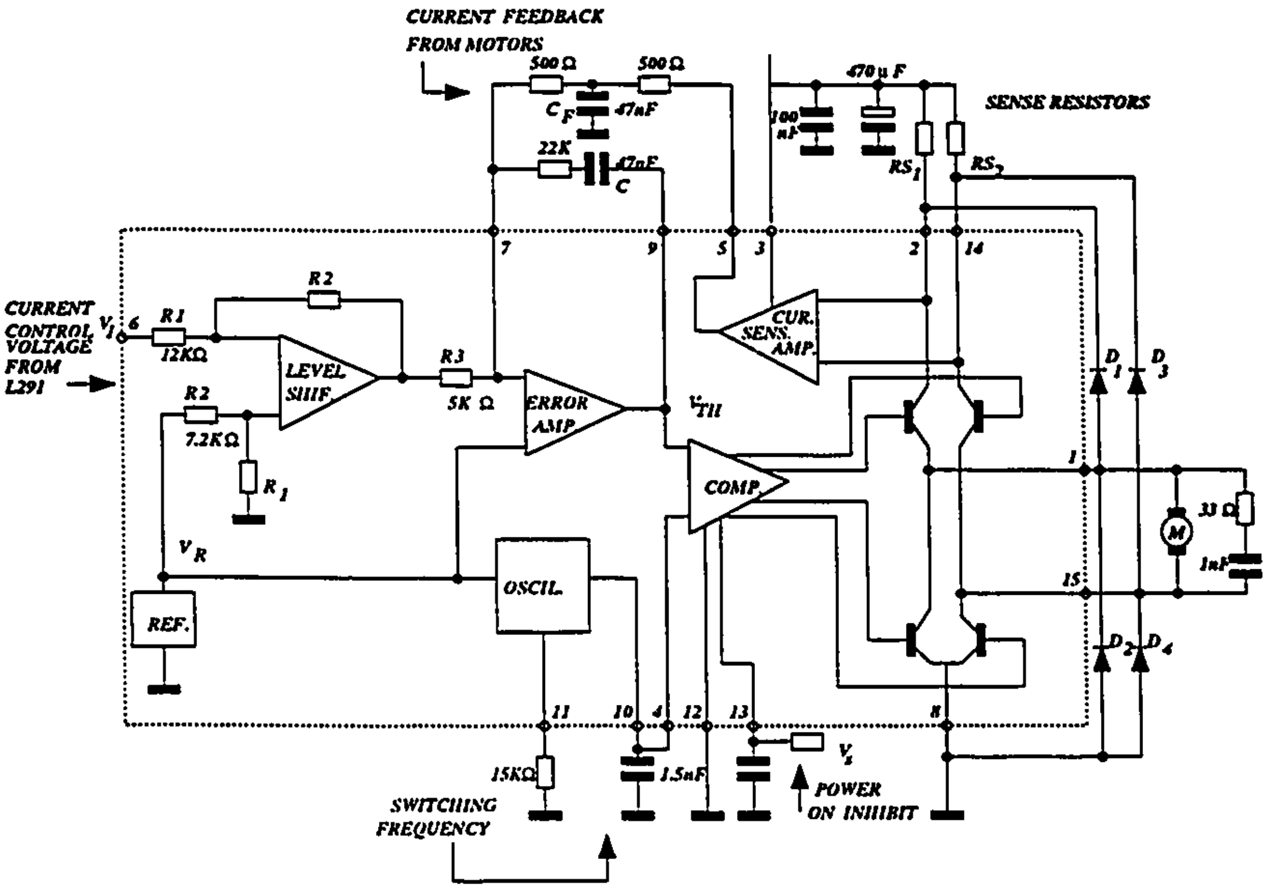


Figure 4.8: The L292 block diagram

control of the motor. Further, it features two enable inputs one of which is useful to implement a power on inhibit function.

The L292's output stage is a bridge configuration capable of handling up to 2A at 36 volts. A full bridge was chosen because it allows a supply voltage to the motor effectively twice the voltage allowed if a half bridge is used. A single supply was chosen to avoid problems associated with pump-back energy.

In a double supply configuration, such as the example in figure 4.9a, current flows for most of the time through D1 and Q1. A certain amount of power is thus taken from one supply and pumped back into the other. Capacitor C1 is charged and its voltage can rise excessively, risking damage to the associated electronics.

By contrast, in a single supply configuration like Fig. 4.9b the single supply capacitor participates in both the conduction and recirculation phases. The average

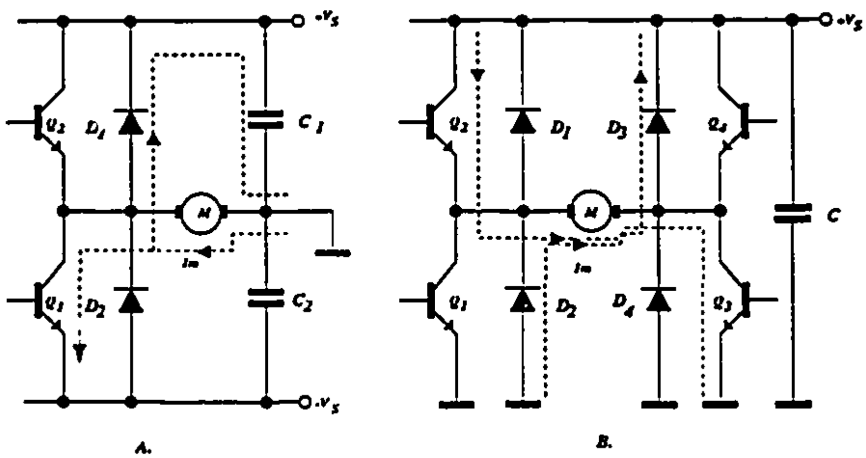


Figure 4.9: Double and single bridge supply configurations

current is such that power is always taken from the supply and the problem of an uncontrolled increase in capacitor voltage does not arise.

A problem associated with the system used in the L292 is the danger of simultaneous conduction in both legs of the output bridge which could destroy the device. To overcome this problem the comparator which drives the final stage consists of two separate comparators. Both receive the same V_t, the triangular wave from the oscillator signal, but on opposite inputs.

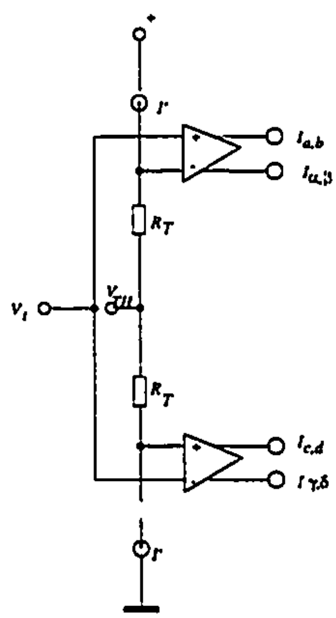


Figure 4.10: The L292's final comparator which consists actually of two comparators.

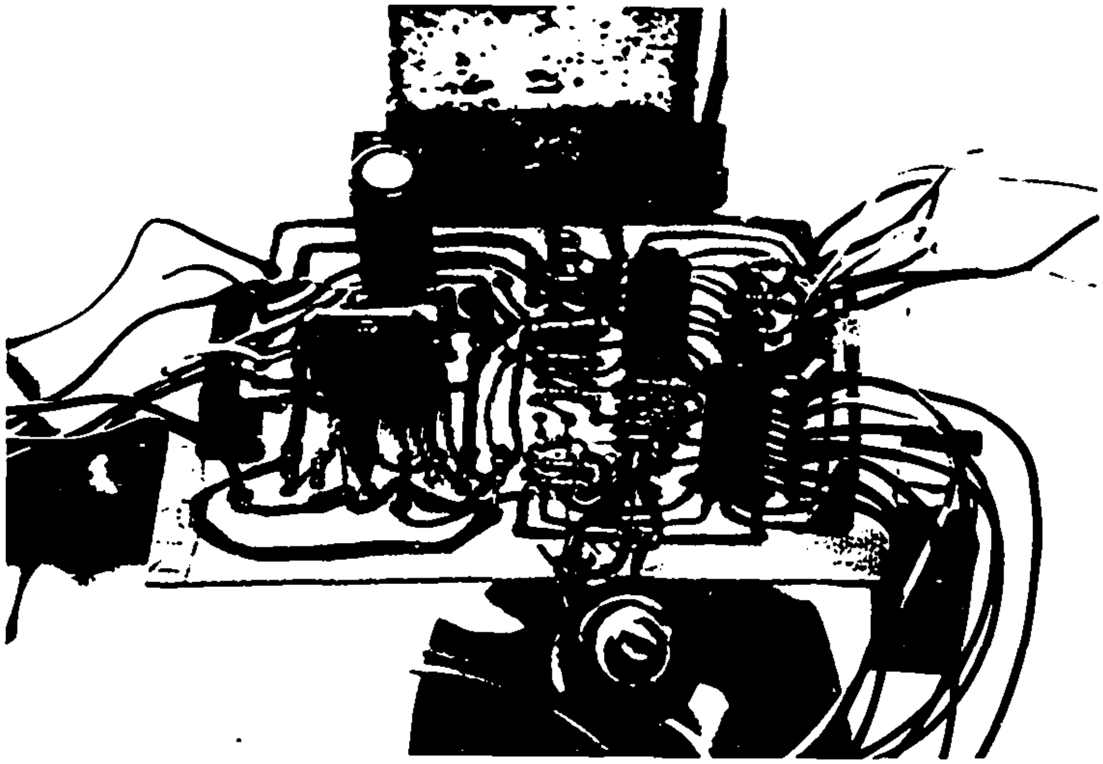


Figure 4.11: Final application circuit

The other two inputs are driven by V_{TH}, the error amplifier output, shifted by plus or minus $R_\tau I'$ (Fig. 4.10). This voltage shift when compared with V_t results in a delay in switching from one comparator to the other. Consequently there will always be a delay between switching off one leg of the bridge and switching on the other. The delay τ is a function of the integrated resistor R_τ (1.5 kΩ) and an external capacitor C17 connected to pin 10 which also fixes the oscillator frequency. The delay is given by:

$$\tau = R_\tau \times C17 \tag{4.22}$$

4.4 Final application circuit implementation and testing

The final DC motor control circuit is shown in Fig 4.11. It consists of the three chips described previously in detail and additional components that ensure security and proper operation of the control circuit. In appendix D Table D.1 indicates the values of these components and also explains their purpose.

The circuit was implemented in the Measurement Laboratory of McGill University and analogue tests were performed to ensure that it operates as expected. In these tests, high and low signals (1's and 0's) were provided to the D/A converter input data lines, the direction line and the speed/position mode select line. Depending on the signals the chip was providing, the motor was rotating as expected and the position feedback signals, coming from the L290 chip, were detected on the oscilloscope. After the test was performed, three printed circuit boards were designed and manufactured.

The next and final step of this research was to perform experiments involving the control circuit interfaced with a microprocessor and connected with a base motor of the manipulator.

4.5 The experiments

4.5.1 Introduction

The main objectives of the following experiments were to demonstrate the ability of the controller to move an actuated link of the manipulator to a specific point and back, feeding the motor with constant or variable voltage and to compare the actual motion and velocity of the link with the ideal ones. The manipulator as mentioned in the beginning of this research is symmetric. The control circuits, motors and shaft encoders used for the actuated links are identical. For these reasons only one actuated link was tested.

The control algorithm developed for the experiments was programmed into the Motorola 68HC05K1 microprocessor. This microprocessor is equipped with an In-Circuit Simulation Kit (M68HC705KICS) [35].

4.5.2 The M68HC705K In-Circuit Simulator Kit

The M68HC705K In-Circuit Simulator Kit consists of a small printed circuit board (or pod), the ICS05K in-circuit/standalone simulator software and the IASM05K assembler. This kit works with any PC and has a DOS-compatible parallel port.

The kit forms a full non-real time simulator and an I/O emulator for the 68HC05K1 device. The pod may be connected to a desired hardware and actual inputs and outputs of this hardware can be used during simulation. With the kit, the hardware and code are tested.

Assembler

The machine code is written and compiled with the assembler. Syntax errors such as unknown identiners are noted. The manual Understanding Small Microcontrollers [50] explains the manner in which the processor works and the instructions required for use. Text is easily copied or moved to other places in the program. After compiling the code, the assembler creates a file with the hexadecimal code which can be written in the memory of the processor. This file is used in the simulator.

Simulation of the microprocessor

The microprocessor is simulated by the computer and the pod. The system is verified in a stepwise manner. The simulator is able to generate fault messages other than those detected previously by the assembler, i.e, when the processor reads unutilized memory locations, the simulator will give a fault message. The processor has a RAM of 32 bytes which is used for temporary data and the stack. Careful programming is required to make optimal use of the RAM.

The ICS05K main screen consists of ten primary windows The values in the CPU (accumulator, X-register, stack pointer and program counter), the values of the I/O ports, the memory of the simulated processor, the source code and the instruction window are viewed. In the instruction window the user can communicate with the simulated processor. The user can manipulate memory places, I/O ports, breakpoints etc. In addition the simulator allows one to see the actual process occurring inside the controller (IC).

I/O Port function

There exist ten bidirectional pins which form two parallel I/O ports, port A and port B. Each I/O pin is programmable as an input or an output. The contents of the data direction registers determine the data direction for each I/O pin. Port A is an 8-bit general purpose bidirectional I/O port while port B is a 2-bit bidirectional I/O port [17].

In a typical system the L290/1/2 system is connected to the control microcomputer through nine I/O lines: seven outputs and two inputs. The outputs are all connected to the L291 D/A converter and consist of the five bit speed demand word, SIGN (which sets the direction) and the speed/position mode select line. Position feedback for the micro comes from the L290 tacho converter and consists of the signals STA, STB (the square encoder outputs).

In the following experiments port A will be providing the seven outputs to the control circuit while port B will be accepting the two inputs from the implemented circuit.

4.5.3 Control Strategy

Two different experiments are performed. In the first experiment the motor is required to move the actuated link from an initial position to a specified final position and back and the voltage provided is constant. In the other experiment the same motion is required but the voltage provided to the motor by the controller changes.

The control algorithm of the first case (which is part of the control algorithm of the next case), can be divided into four main parts: Start motion, continue until reaching desired position, reverse direction and stop when reaching initial position.

In the second case the control algorithm is subject to the following requirements:

1. Start motion with maximum speed,

2. At a certain position decrease speed and continue motion until desired position,

3. Reverse direction with maximum speed,

4. Decrease speed at a certain position,

5. Stop motion when reaching initial position.

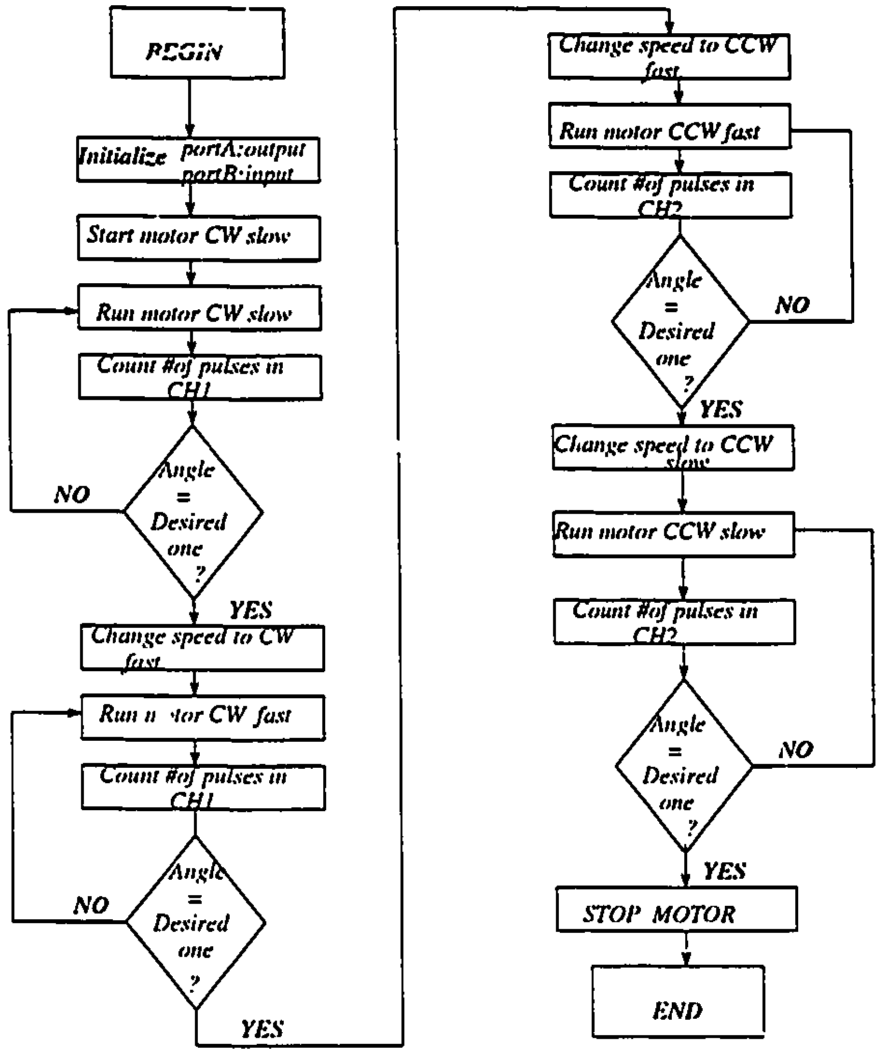


Figure 4.12: Flowchart of the control algorithm for experiment No2

In Fig. 4.12 the flowchart of this control algorithm is presented. The control algorithm is written in assembler. The listing of the two programs used in the experiments (source code and machine code) is given in Appendix E.

The experiments were setup and performed in the Measurement Laboratory of McGill University. A photograph of the setup is shown in Fig. 4.13.

In the first experiment it was desired to move the link from 0 to 45 degrees and back without changing the voltage that was provided to the control circuit via

Figure 4.13: Photograph of experiment setup

the microcomputer. The time needed for this experiment was 2.12 seconds. The ideal motion and velocity profiles of the actuated link were found using the cubic polynomial, for a single cubic segment which starts and ends at rest. This cubic polynomial is further discussed in [7].

In the second experiment the same motion was desired but the voltage provided was changing. The time needed for this experiment was 3.6 seconds. The ideal motion and velocity were calculated the same way as in the first experiment.

The following set of graphs (Figs. 4.14 to 4.19) and (Figs. 4.20 to 4.25) are the results of the actual motion and velocity, the ideal motion and velocity and their comparison for both experiments.

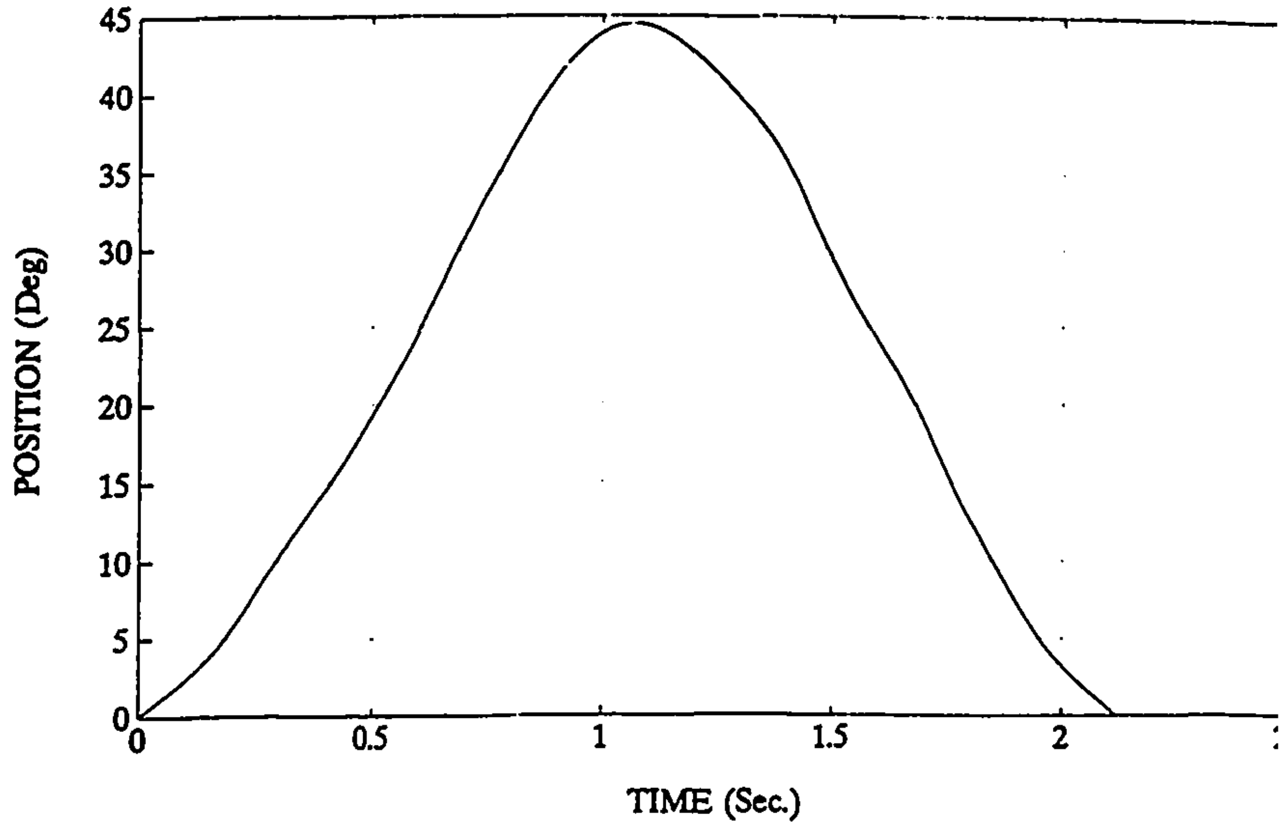


Figure 4.14: Experiment No. 1: Actual motion of the actuated joint

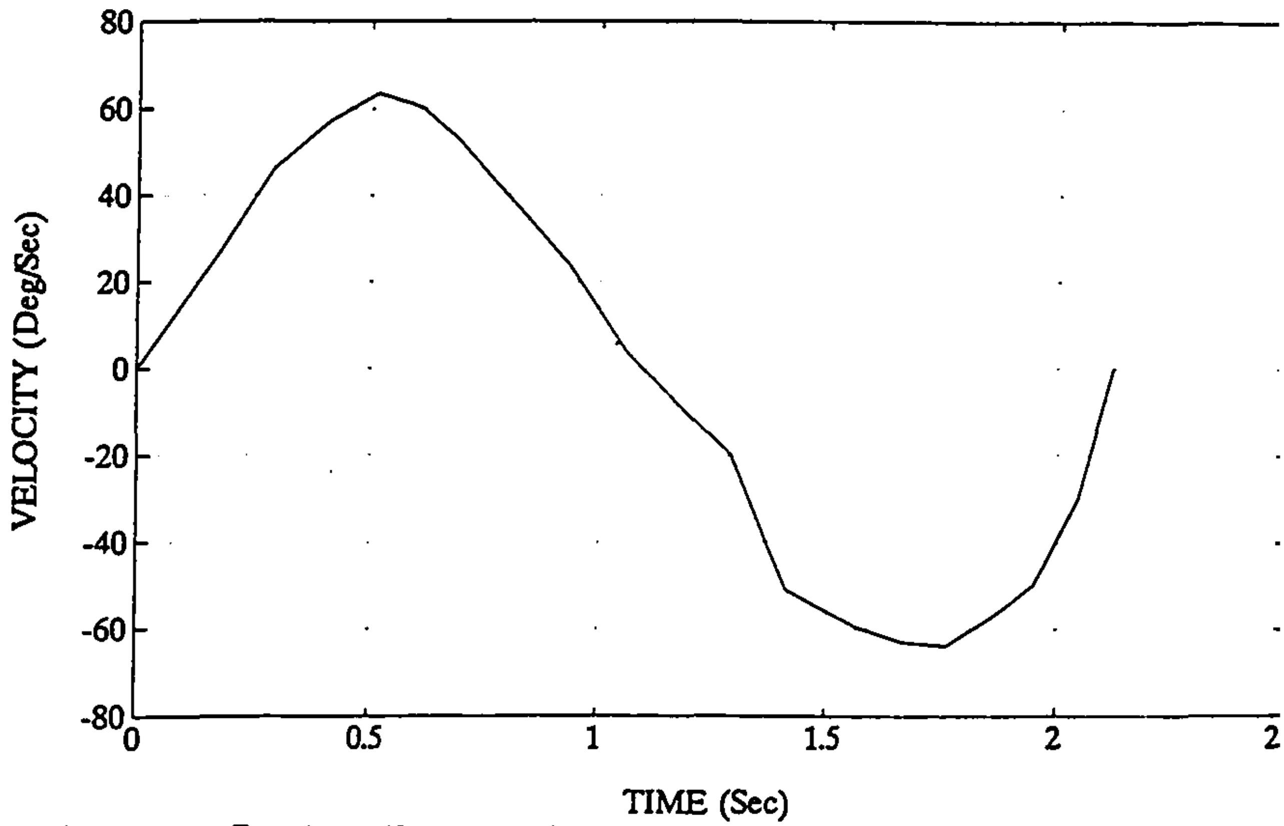


Figure 4.15: Experiment No. 1: Actual velocity of actuated joint

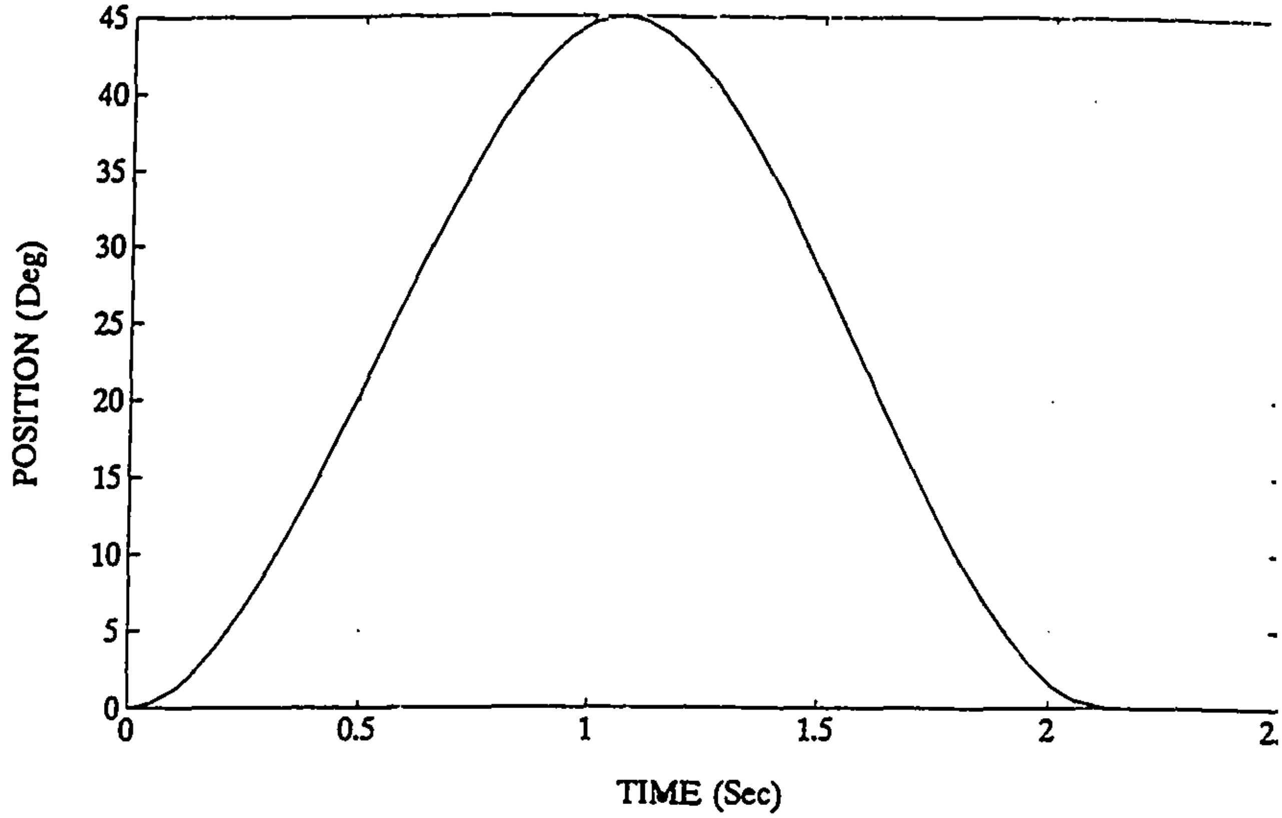


Figure 4.16: Experiment No. 1: Ideal motion of actuated joint

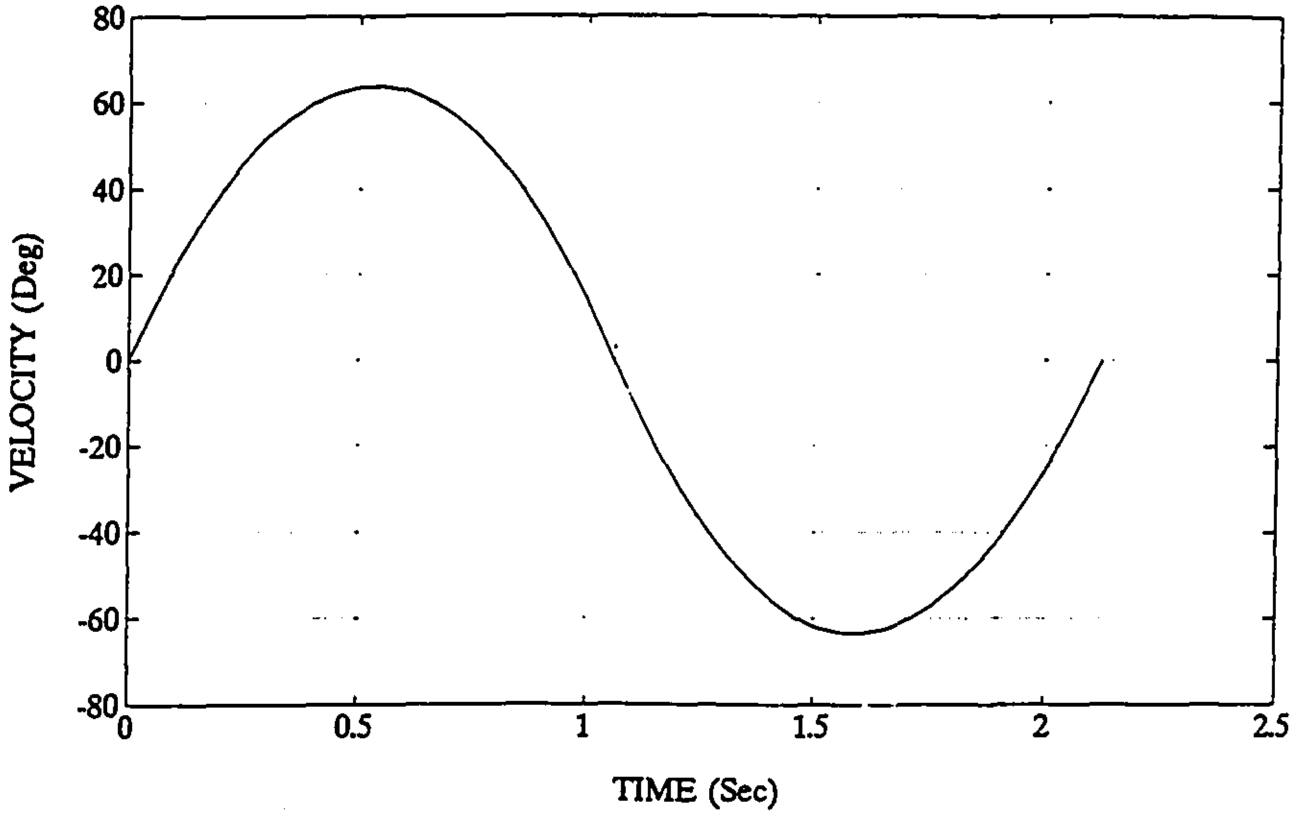


Figure 4.17: Experiment No. 1: Ideal velocity of actuated joint

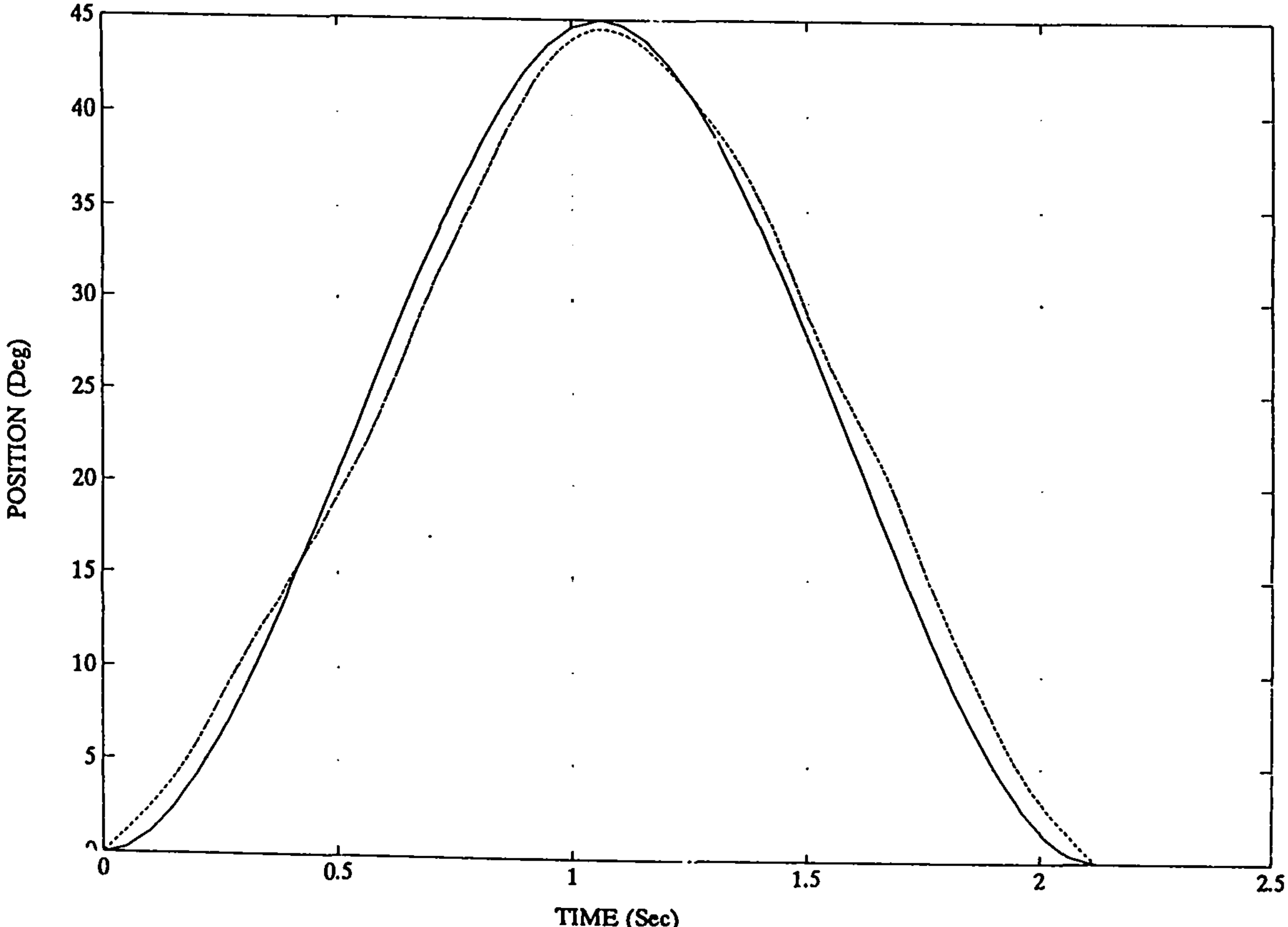


Figure 4.18: Experiment No 1: Ideal VS Actual position

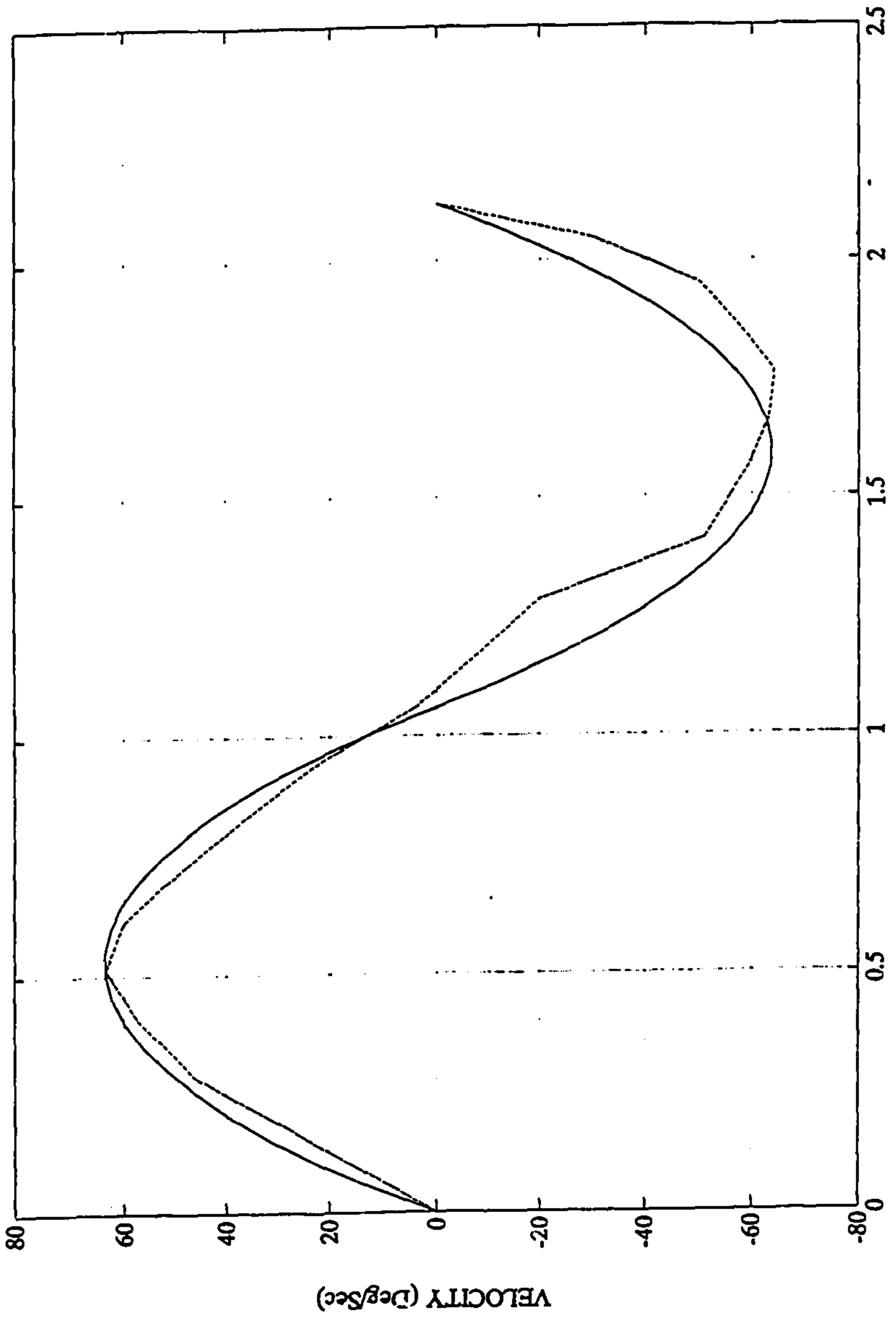


Figure 4.19: Experiment No. 1: Ideal VS Actual velocity

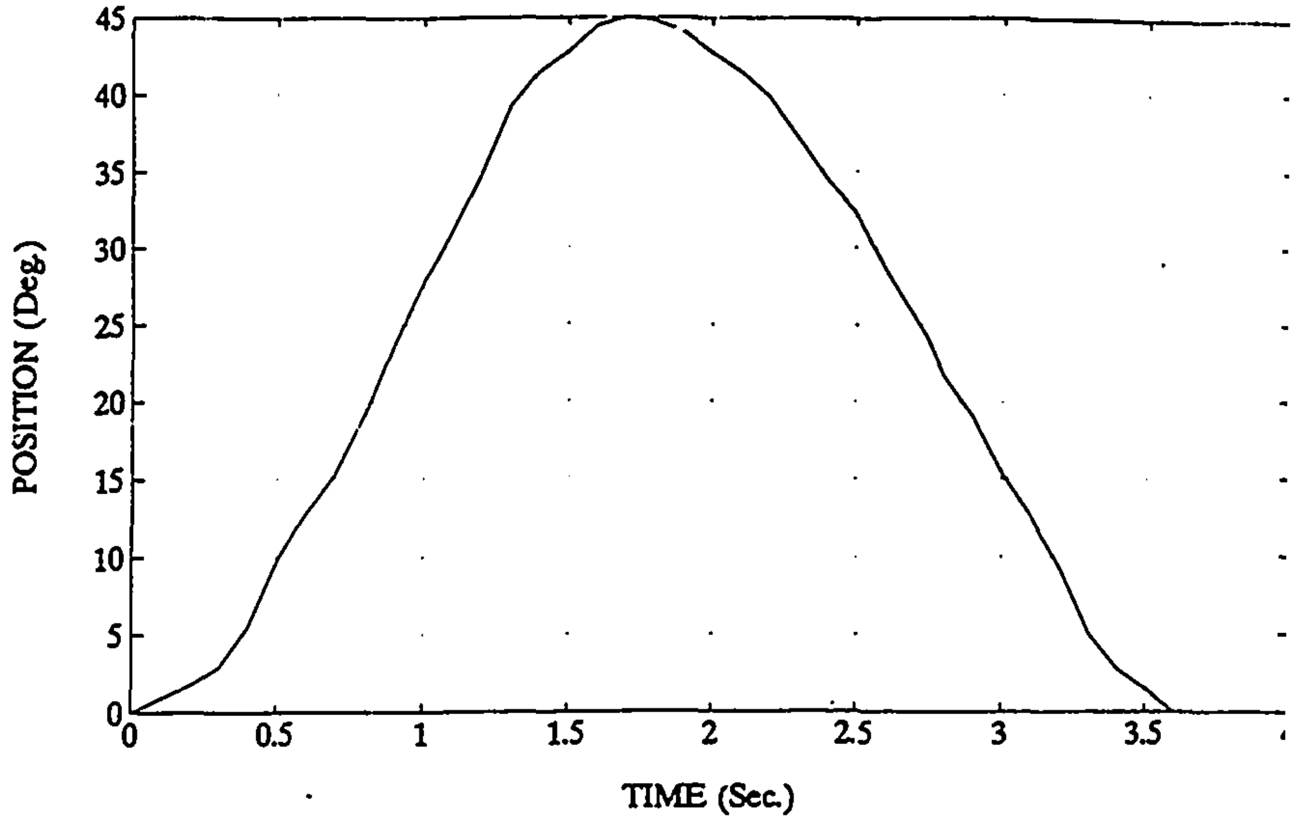


Figure 4.20: Experiment No. 2: Actual motion of the actuated joint

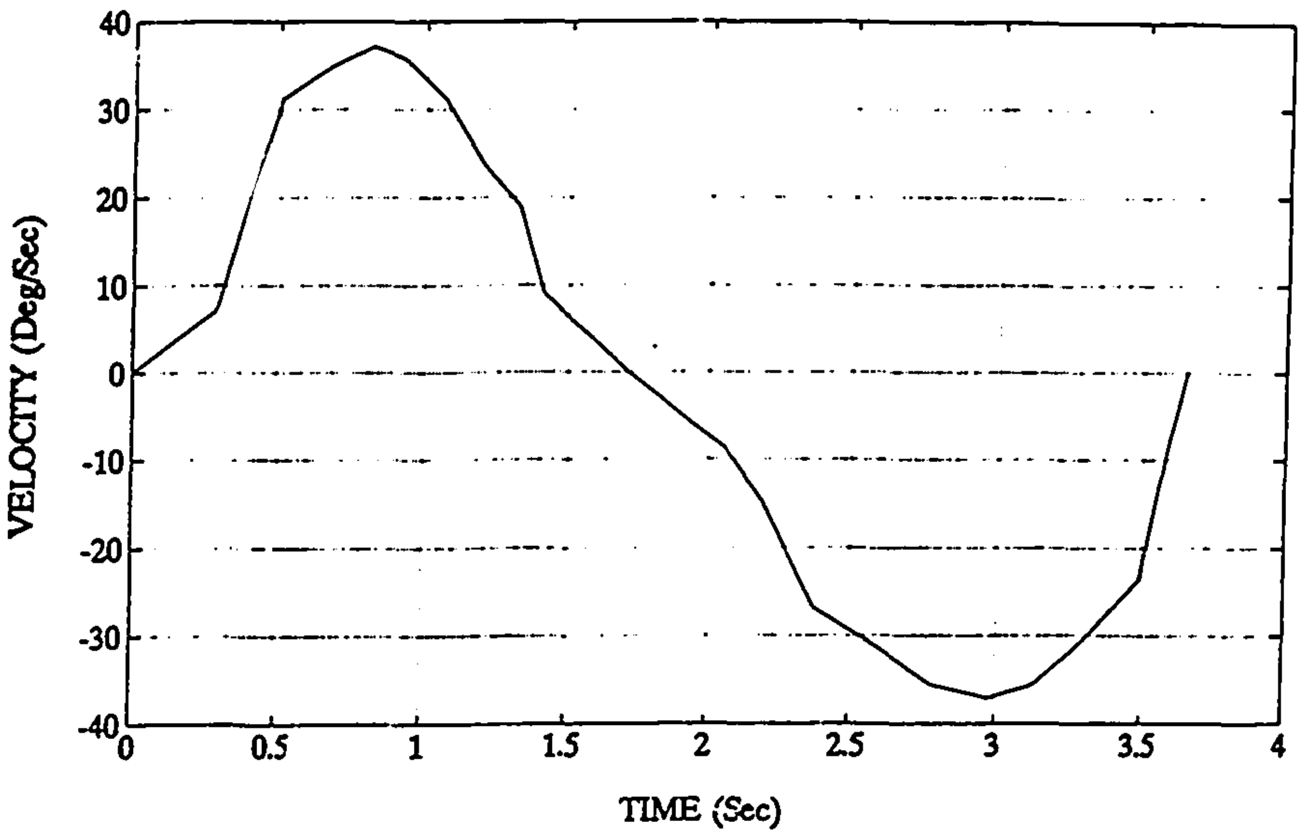


Figure 4.21: Experiment No. 2: Actual velocity of actuated joint

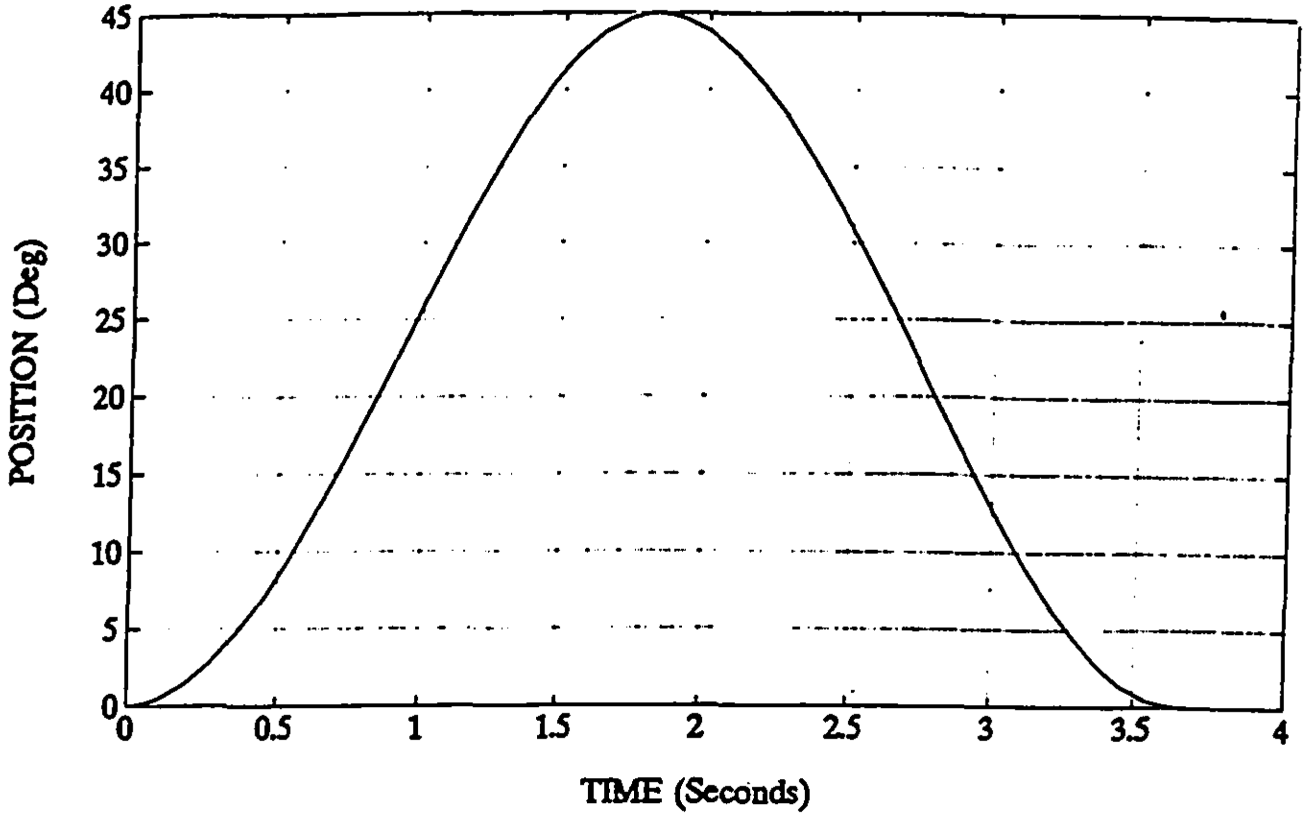


Figure 4.22: Experiment No. 2: Ideal motion of actuated joint

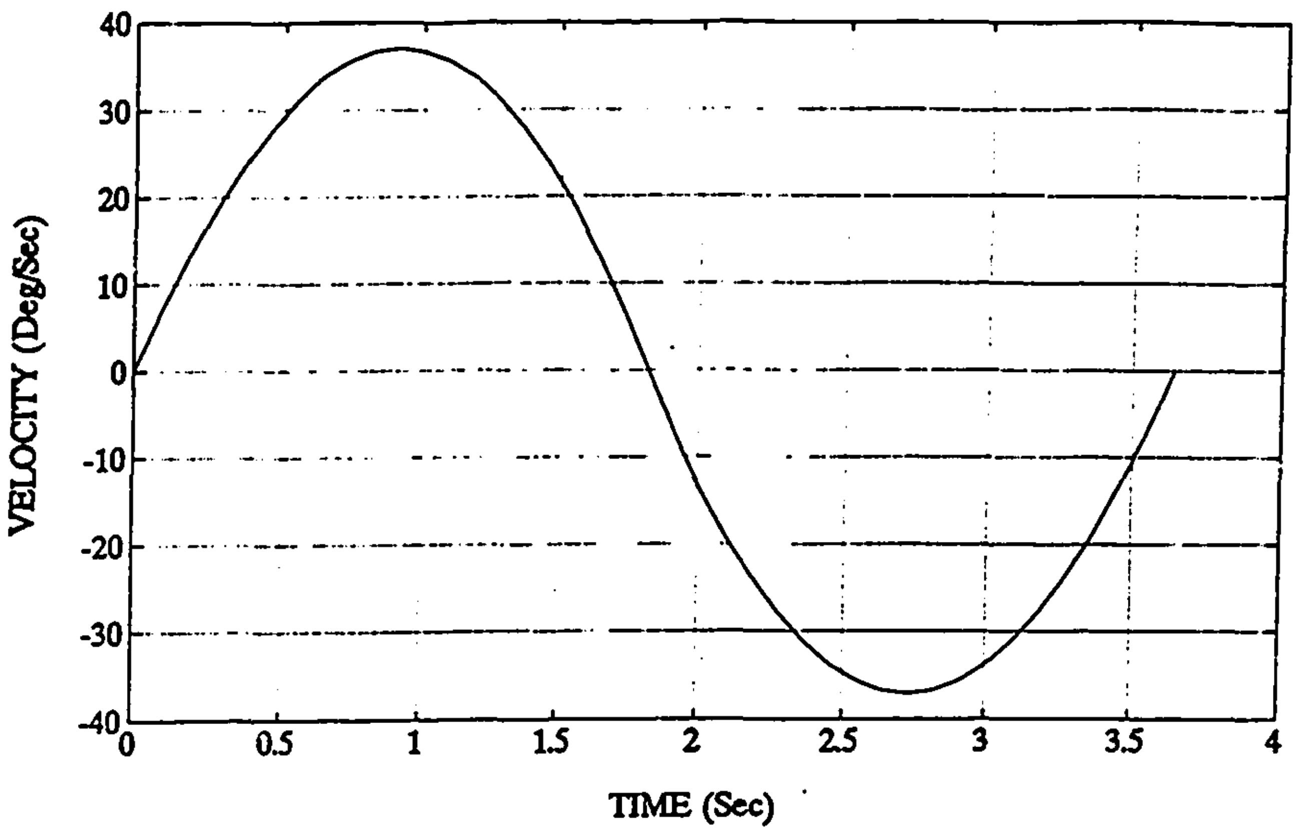


Figure 4.23: Experiment No. 2: Ideal velocity of actuated joint

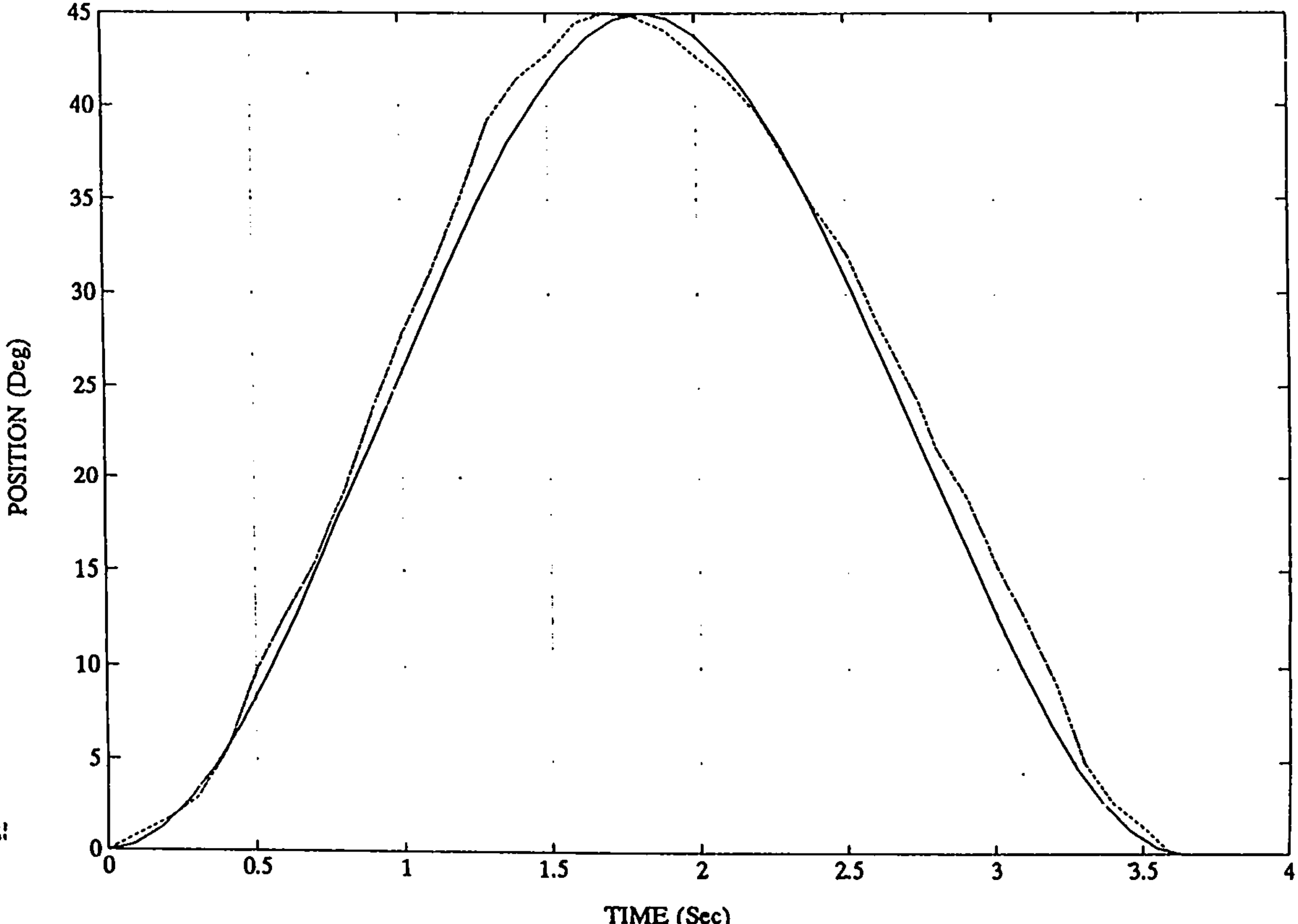


Figure 4.24: Experiment No 2: Ideal VS Actual position

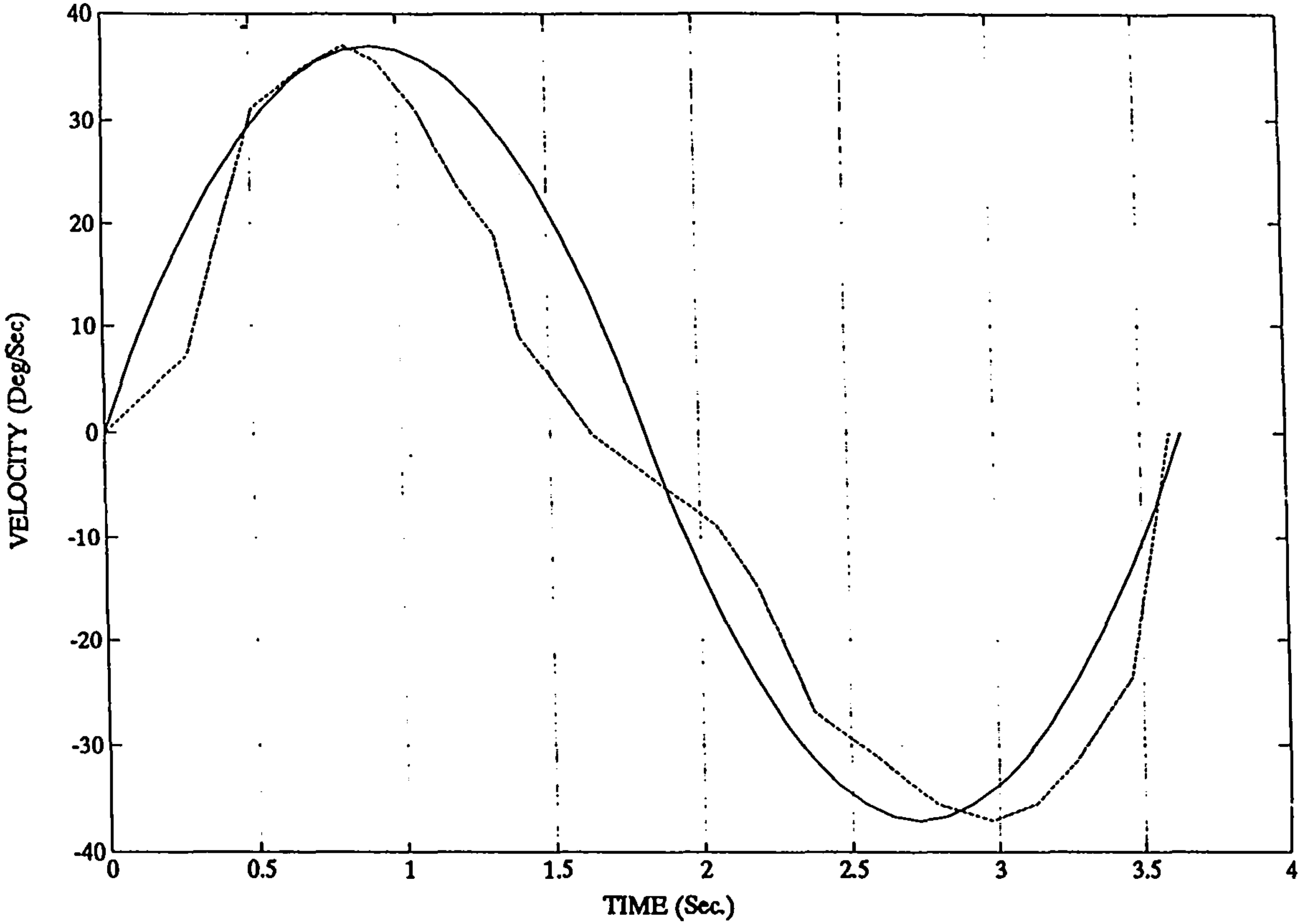


Figure 4.25: Experiment No. 2: Ideal VS Actual Velocity

In Figs. 4.18 and 4.24 both actual and desired trajectory are plotted -vs- time. In both experiments, as shown in these figures, the actual displacement follows the desired trajectory very closely. Desired versus actual velocity is shown in Figs. 4.19 and 4.25. The actual velocity was found by differentiating the actual motion. Variations from the desired velocity were expected and can be seen in the aforementioned figures. The motion of the actuated link, in these experiments, was smooth. These results can be improved if the bandwidth of the designed control circuit is improved, so that the frequency response of the system is improved.

At this point the project enters a new phase. In order to obtain real time control the prototype must now be interfaced with the existing transputer network of McRCIM to achieve parallel processing. In the next and last section of this study, an introduction to parallel processing is presented and an interface proposal is made as a first step towards the next phase.

4.6 The transputer

4.6.1 Introduction

In order to make computers faster, there are two different approaches:

- Make the components of which the computer is built faster. The problem with this approach is that industry is getting close to fundamental physical limits, which means that only diminishing returns in speed of operation may be expected from new generations of integrated circuits.

- Increase the number of processors, working in parallel. Recent advances in both semiconductor and computer technologies have enabled parallelism to be used as a viable technique to obtain high performance at a modest cost, in a wide range of computing systems. The problem here is that the communication overhead can cause a saturation or even a decrease of calculating speed as the number of processors increase too much.

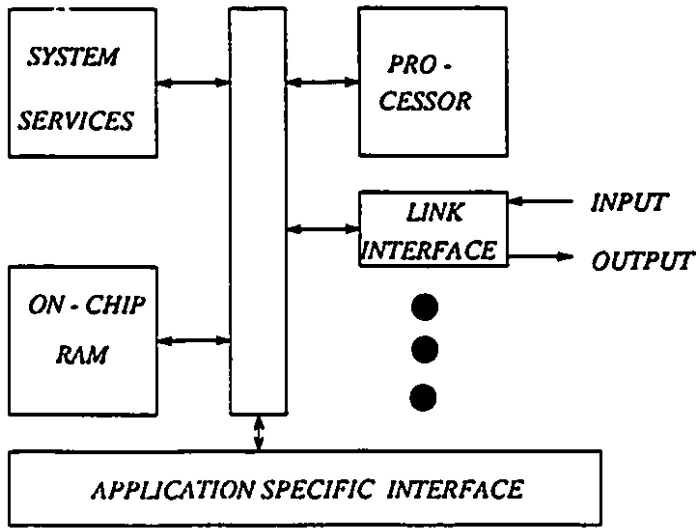


Figure 4.26: Basic transputer architecture

A solution for the problem of the communication overhead of parallel processing is found with the introduction of *transputers*. The transputer family is a family of 16 and 32 bit single-chip microcomputers that have their own memory and communicattion links for connecting one transputer to another. Thus, the transputer combines processing, memory and interconnections in a single VLSI chip. The big advantage of transputers is that the transputer architecture is developed in order to service parallel processing. because the transputers have links and on-board memory, they do not have to share a common bus or memory with other processors. This is a big advantage, because the common bus is the cause of most trouble in conventional parallel computers:

- With links there is no contention for the communication system.
- Links don't have a capacitive load penalty when transputers are added to the system.
- The communications bandwidth does not saturate as the size of the system increases.

The basic transputer architecture is shown in Fig. 4.26 .

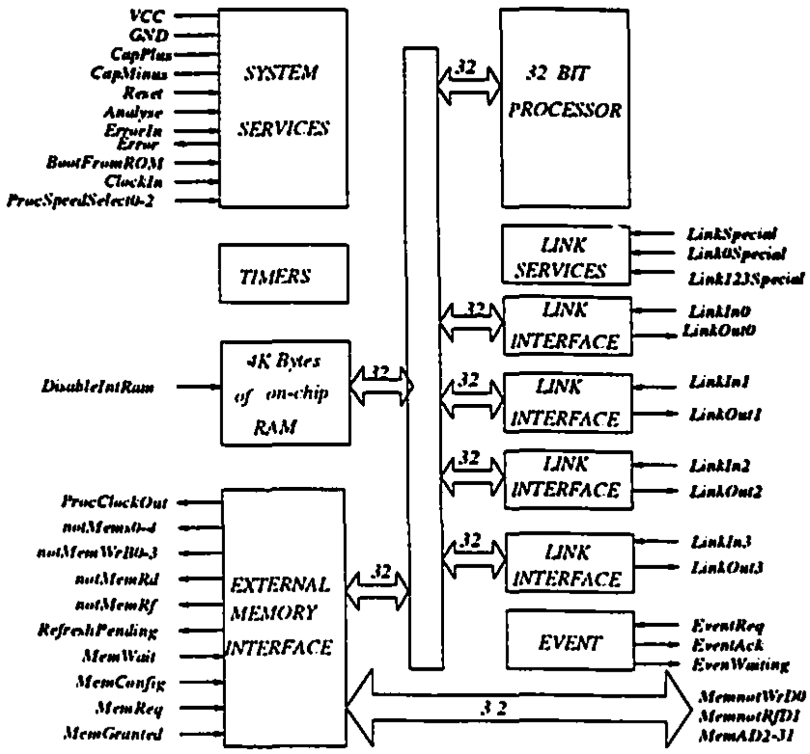


Figure 4.27: Block diagram and architecture of the T425 transputer

Transputers are designed to implement the programming language *Occam* [26] very efficiently and although they will also provide efficient implementation of most modern languages, concurrency in transputer systems is only available through Occam.

4.6.2 The IMS T425 transputer

The IMS T425 transputer (Fig. 4.27) is the one that was installed programmed and used to control the manipulator under study. It has 4Kbytes on-chip RAM for high speed processing, a configurable memory interface and four standard INMOS communication links. The instruction set achieves efficient implementation of high level languages and provides direct support for the occam model of concurrency when using either a single transputer or a network [25].

It is a 32 bit CMOS microcomputer capable of instruction rates of 30 MIPS peak and 15 MIPS sustained when operated with a 30 MHz clock. The transputer links are capable of transferring data at 5, 10 or 20 Mbits/sec, however only one link is

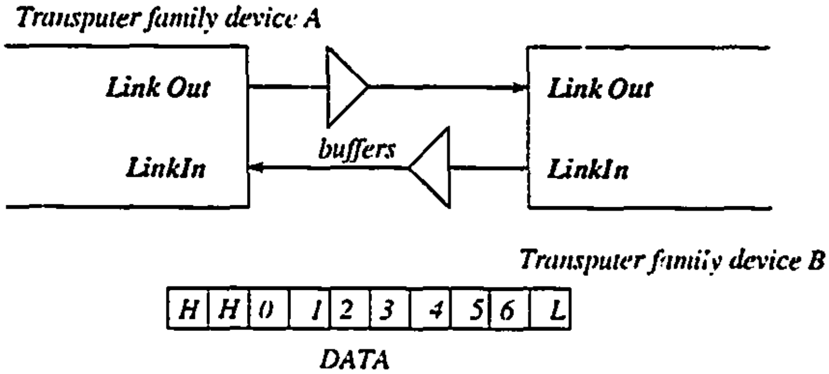


Figure 4.28: Transputer link protocol

available to transfer data to the motor controllers implemented.

The transputer link protocol is "little-endian" with the least significant bit of the least significant byte being sent out over the link first. Each byte transmitted serially is structured as a high (one) start bit followed by a one bit, followed by the eight data bits ending with a low (zero) stop bit (Fig. 4.28). Each transmission must be acknowledged by a high-low (one-zero) two bit signal.

Internally floating point numbers conform to the IEEE standard and integers are represented in sign-magnitude rather than in 2's complement form.

The existing transputer network at McRCIM (McGill Research Centre For Intelligent Machines) contains one transputer module motherboard (the B008), connected to an IBM PC AT, and two transputer modules both containing one IMS T425 transputer. One module contains 1 Mbyte memory and the other contains 2 Mbytes memory. In both modules the event channel is not connected. Also available are some IMS C011 link adaptor chips. These are used to connect the transputer network to peripheral devices.

4.6.3 The C011 link adaptor

The C011 link adaptor converts from the inmos serial link protocol to a parallel byte wide interface. It was used in the design of the transputer/DC motor controllers interface. This link adaptor can operate as either a peripheral interface or a bus interface, the former mode being of interest here (Fig. 4.29). The standard commu-

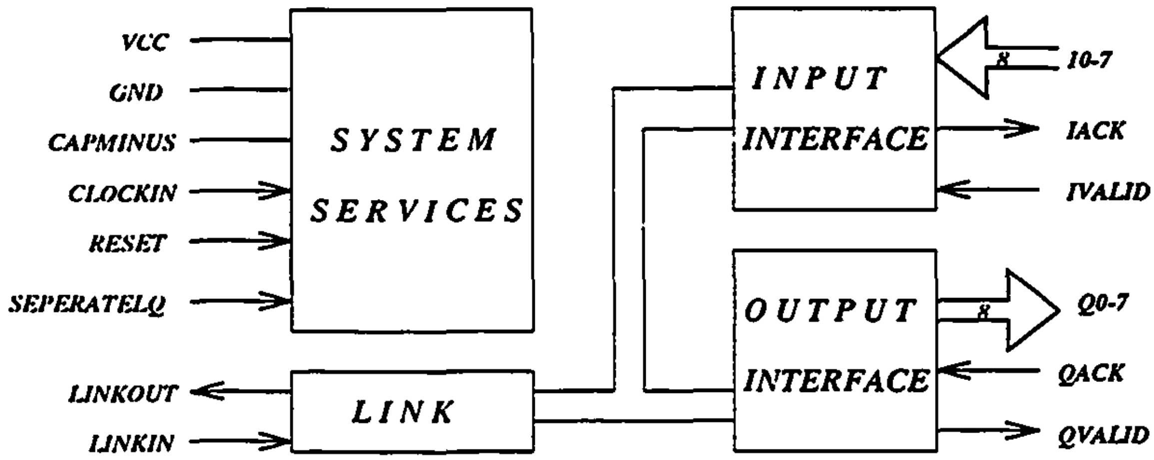


Figure 4.29: The C011 link adaptor block diagram

nication speed is 10 Mbit/sec, but the device can also operate at the higher speed of 20 Mbits/sec. The parallel input and output lines to the C011 are fully handshaken, with only the output lines being used for this interface design. The C011 is capable of bidirectional serial communication, but as a first interface design step here, it is only to receive data.

The C011 removes the start bits and the stop bit from the transmitted data and places the byte on the output lines. To perform handshaking, the C011 takes a QValid signal high, indicating to the peripheral that there is new data to be read. When the peripheral has processed the data and is ready to receive a new byte, it takes the QAck signal high causing the C011 to place the high-low acknowledge signal on the serial link, indicating to the transputer that the C011 is ready for a new byte.

4.7 Interfacing the transputer and the motor controllers

The characteristics of the transputer make it very well suited to perform calculations fast and output the results over its communication links, so they can be used by other transputers or by peripherals. A program, which calculates the required torques that have to be exerted by the three motors of the parallel manipulator manufactured, in order for the manipulator to follow a desired path, has already been written by Helmy [18] and is installed in the network.

The torque values are in the form of 64 bit floating point numbers. The program converts the results to 64 bit signed integer values so that the information can be meaningful to the D/A converters of the motor drive circuit.

The torque values calculated are such that the motors are in rest position (stopped) when zero torque is indicated, when positive torque is indicated the motors rotate clockwise and finally when the indicated torque is negative the motors rotate counterclockwise. Under constant load, the torque exerted by the motors can be varied by altering the speed of the motors. Therefore the torque values can be used as the speed word inputs of the D/A converters. As the torque values are signed integers, only the most significant bit is different for positive and negative numbers. Therefore the sign of the torque values can be used as the sign bit to the D/A converter. The speed word input to the D/A converters can be taken from the next five most significant bits of the torque values. If the motor is set up for clockwise rotation with positive current, then the six most significant bits of the torque values can be inverted and send directly to the L291's as input.

Sixty-four bit signed integers span a large range (-2^{63} to $+2^{63} - 1$). If the six most significant bits of the torque values are taken as inputs for the motor controllers, this would mean that the torque value would have to reach a magnitude of 2^{58} before the motor would start to move. It becomes obvious that the program has to scale magnitude of the torque values between $2^{58} - 1$ (stop) and $2^{63} - 1$ (full speed) for the data send to the motor controllers to be meaningful.

On the parallel manipulator are installed three base motors and thus the transputer calculates three torque values. In order to simplify the interfacing hardware and not slow down data transmission, the torque values are sent sequentially for motor 1, motor 2 and motor 3 and *always* in that order. By not having motor addresses, the need for address lines in the interface design is removed at the expense of fixing the design for three and only three motors. The hardware of the interface is synchronized with the transputer calculations by being reset at start up by the transputer. To remain synchronized, the transputer must always output three torque values of the proper magnitude and in the proper order, regardless of the number of motors that

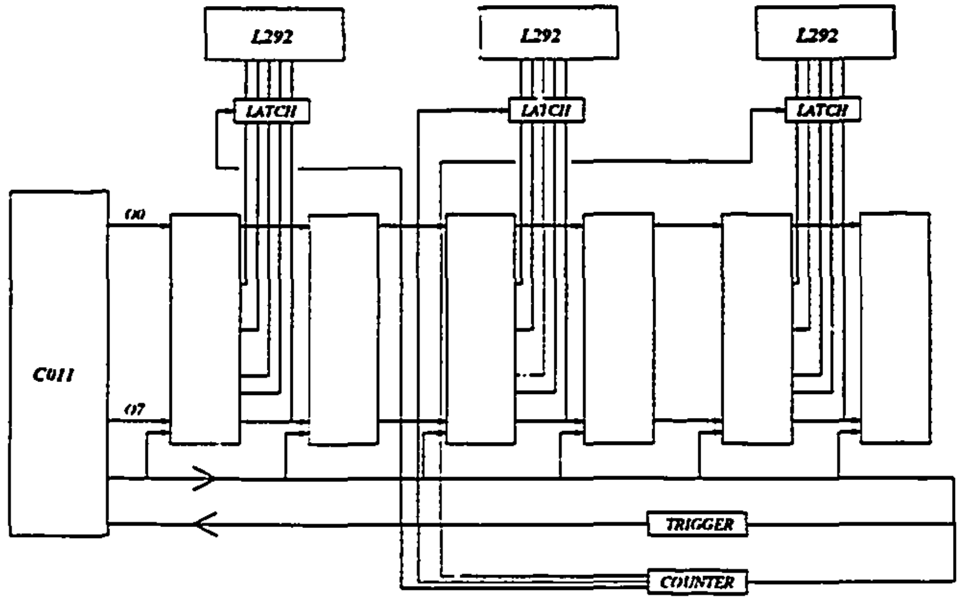


Figure 4.30: Block diagram of interface

need to be updated.

As the link protocol of the transputer calls for the least significant bit to be transmitted first, the hardware interface must wait until all of the 3×64 bits of the torque values are received before the data to the D/A converters can be updated.

4.7.1 Shift registers with output latches

The first obvious solution is to use one large shift register (or many smaller ones in line) with enough bits to hold the three torque values coming out of the transputer. While this is a possible solution after one byte is taken from the C011, it would require eight clock cycles to make space for the next incoming byte. A design were all eight bits could be shifted in one clock cycle presents a significant improvement. While the design can no longer be conceptually thought of as one large shift register, the actual number of bits needed to store the torque values in shift registers does not increase. If the hardware interface is conceptualized as a matrix of data bits, then it does not matter how the actual shift registers are oriented, as long as the locations of the bits are known and the bits needed for the D/A converter are available.

The shift registers used in the design of this interface are twenty-four TI 74LS594's.

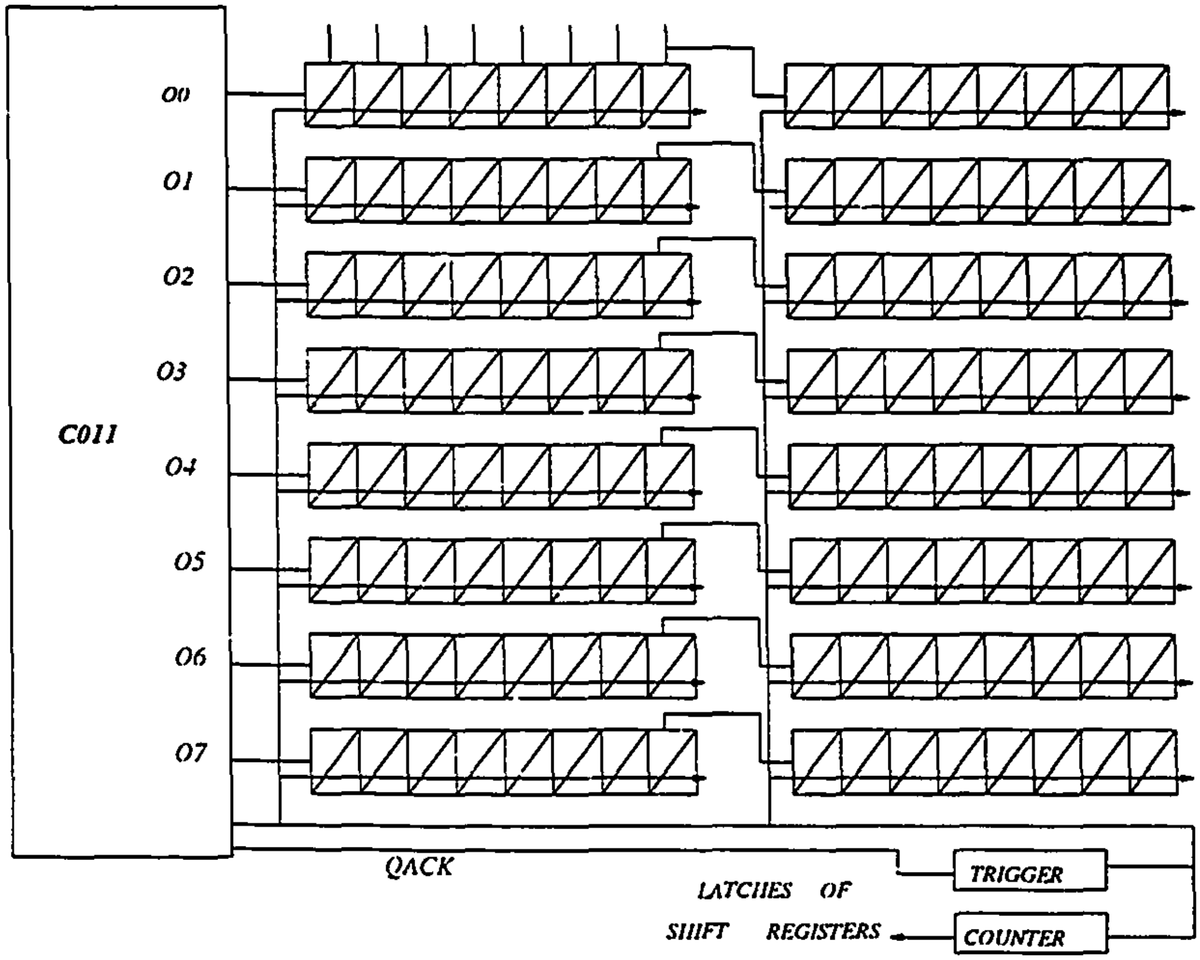


Figure 4.31: Block diagram of the 8bit serial in / parallel out shift registers with output latches

These are 8 bit serial-in, parallel-out shift registers with a storage register that can be used to latch the output. Latching the output is very important in sending data to the motor controllers. The bits in the interface are only meaningful to the motor controllers after three torque values have been received. If the output was not latched and updated only with every 3×64 bits, then the data to the controllers would change with every byte received and would be meaningless.

The shift registers are oriented such that each output line of the C011 is a serial input into a register. The QValid signal of the C011 is used as a clock input to shift the bits through the register. The design can be though of as eight shift registers in parallel with the torque values snaking across the registers Fig. 4.31.

4.7.2 Counters

Because the data seen by the D/A converters can only be updated after three torque values are received from the transputer, the interface has to keep track of the number of bytes received. Three sets of 64 bits are sent by the transputer corresponding to 24 sets of eight bits transferred from the C011 to the shift registers. The interface must then include a counter which increments with each byte received, counts 24 bytes and sends a signal to load the storage registers with the new data in the shift registers, updating the data seen by the D/A converters of the motor controllers.

The counter is made up of Dual D-type flip-flops (TI 7474) with supporting logic to count from 0 to 23 Fig. 4.32. The QValid signal from the C011 is used to increment the count. The Boolean equations governing the count sequence are given by:

$$a(t+T) = \bar{a}(t) \tag{4.23}$$

$$b(t+T) = b(t) \oplus a(t) \tag{4.24}$$

$$c(t+T) = c(t) \oplus (b(t) \cdot a(t)) \tag{4.25}$$

$$d(t+T) = \bar{e}(t) \cdot (d(t) \oplus (c(t) \cdot b(t) \cdot a(t))) \tag{4.26}$$

$$e(t+T) = e(t) \oplus ((e(t) + d(t)) \cdot c(t) \cdot b(t) \cdot a(t)) \tag{4.27}$$

The outputs of the flip-flops are fed to a NOR gate so that when the count is zero (after the twenty fourth byte is received), the output of the NOR gate causes the storage registers to be loaded with the data from the shift registers. Loading the storage registers on a zero count is also useful in the event of a reset, it causes a stop signal to be sent to the motor controllers.

4.7.3 QAck signal generator

After a byte has been moved from the C011 to the shift registers, the QAck signal must be sent to the C011 informing it that the shift registers are ready to receive another byte. This is accomplished using a TI 74123 dual monostable multivibrator

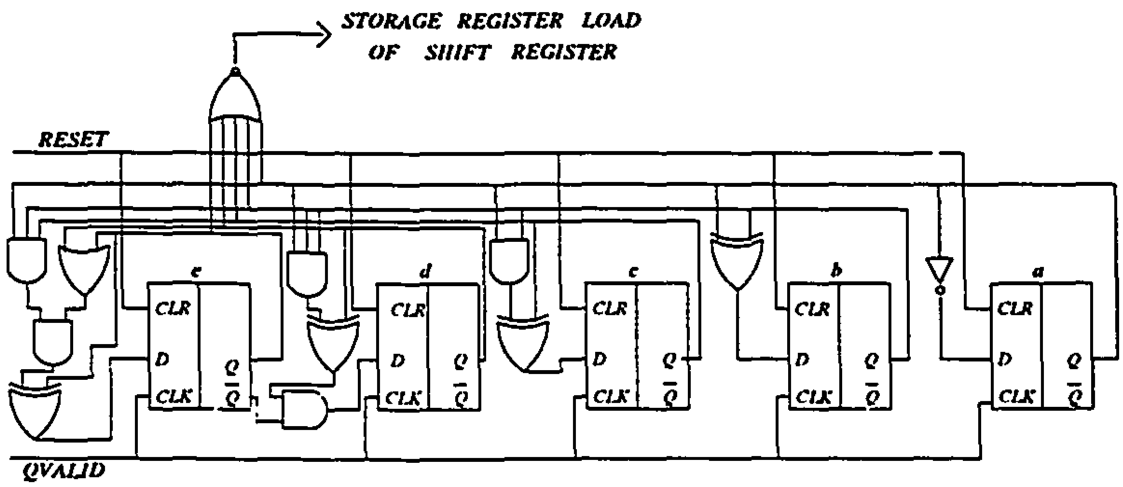


Figure 4.32: Block diagram of the Dual D-type flip-flops and supporting logic forming the counter

Fig. 4.33. The QValid signal from the C011 triggers half of the 74123. The length of this pulse is determined by the time needed to shift the data through the shift registers, increment the counter and shift information into the storage registers if necessary. The falling edge of the first pulse triggers the second half of the 74123 and the output pulse is used as the QAck signal sent to C011.

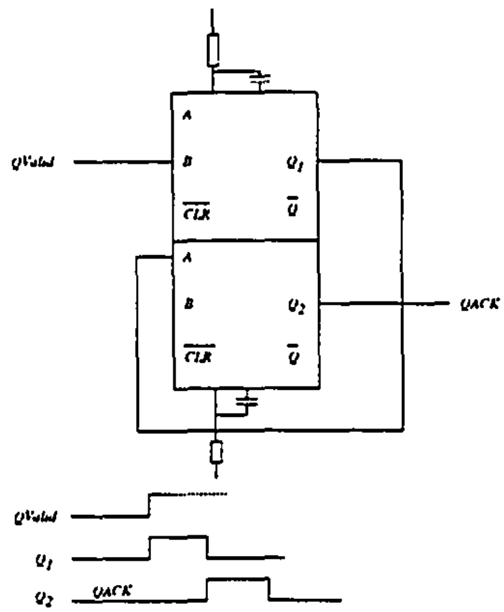


Figure 4.33: Block diagram of the 74123 used to generate the QAck signal

4.7.4 Other hardware design

A standard TTL output can drive up to ten TTL loads. In this interface design there are several signals that are split and sent to a larger number of chips. QValid is used as an input to the shift register, the counter and the QAck trigger. The reset signal clears the storage registers, the shift registers and the counter. The output of the NOR gate is used to load eighteen storage registers with the contents of the shift registers. In order for these signals to be strong enough to drive all these loads, the signals need to be passed through a TI 74367 bus driver. The 74367 outputs can be used to drive up to forty TTL loads. If more than this is needed, the original signal can be brought to two input pins and then the two output pins will be able to drive eighty TTL loads, a gain of seventy eight.

All the supporting logic in the design (inverters, AND, OR, XOR and NOR gates) are standard TI 74xx series TTL logic.

Chapter 5 Conclusions

This thesis presents the design philosophy employed for the construction of the prototype of a three degree-of-freedom parallel manipulator with revolute joints and the design and implementation of a closed-loop control circuit which would be able to control the motors of the constructed manipulator.

A prototype was designed at McRCIM (McGill Research Center for Intelligent Machines) and constructed in the Machine Tool Laboratory of McGill. The length of the links was chosen such that it would facilitate the verification of Gosselin's [8] condition equations for a non-vanishing workspace for every angle ϕ. The design conclusions were that three geared DC motors with optical encoders for position and velocity control, mounted on the base of the manipulator and coupled directly to the three base joint shafts of the prototype, represented the optimum actuator configuration. In addition for further improvement in controlling the prototype, three auxiliary geared motors with optical encoders were also selected and special design considerations were made to enable these motors to be mounted on joint shafts (connected to the end-effector) in the future. The ideal link material, for the workspace and static load specifications, was aluminum while the ideal material for the joint shafts, which represented only a small amount of the robot weight and deflection was structural steel.

A motor speed/position control system was selected for the control of the DC motors of the constructed prototype. This system consists of three chips, namely: L290, L291 and L292, which together form a complete microprocessor - controlled DC motor servopositioning system which is both fast and accurate. The final application circuit which includes additional hardware, was implemented and connected with the base actuators of the prototype. Analogue tests were performed to prove that the circuit would function as expected. In addition experiments were setup and executed and results of the manipulator's actual motion an velocity were displayed. These results were compared with the desired motion and velocity profiles for both experiments and it was found that they followed them closely.

This project now enters a new phase. In order to obtain real time control, the prototype must now be interfaced with the existing transputer network of McRCIM to achieve parallel processing. An introduction to parallel processing was presented and finally an interface proposal was made as a first step towards the next phase.

In the next phase of this project the following extensions could be made and are suggested:

- Mount the auxiliary motors on the prototype and extend the existing transputer software so as to activate them in order to pass through singularities; and
- Design and implement a bidirectional interface to accomplish the desired communication between the transputer network and the prototype.

Bibliography

[1] Angeles, J. and Bernier A., "*A general method of four-bar linkage mobility analysis*". ASME Journal of Mechanisms, Transmissions and Automation in Design, Vol. 109, No. 2, pp. 197-203, 1987.

[2] Angeles, J. and Lee, S., "*The formulation of dynamical equations of holonomic mechanical systems using a natural orthogonal complement*". Trans. ASME, Journal of Applied Mechanics, Vol. 55, No. 1, pp. 243-244, 1988.

[3] Angeles, J. and Lee, S., "*The modelling of holonomic mechanicak systems using a natural orthogonal complement*". Transaction of CSME, Vol. 13, No. 4, pp. 81-89, 1989.

[4] Angeles, J. and López-Cajúun, C. S., "*The dexterity Index of Serial type Robotic Manipulators*". ASME Trends and Development in Mechanisms, Machines and Robotics, September 24-28, Kissimmee, Florida, Vol. 3, pp. 79-84, 1988.

[5] Atkeson, C. G., An, C. H. and Hollerbach, J. M., "*Estimation of Inertial Parameters of Manipulator Loads and Links*". The International Journal of Robotics Research, Vol. 15, No. 3, pp. 101-109, 1986.

[6] Baillieul, J., "*Kinematic programming alternatives for redundant manipulators*". Proceedings 1985 International Conference in Robotics and Automation, Silver Spring, pp. 337-345.

[7] Craig, J. J., "*Introduction to Robotics, Mechanics and Control*". Addison-Wesley Publishing, 1986.

[8] Gosselin C., "*Kinematic Analysis, Optimization and Programming of Parallel Robotic Manipulators*". Ph.D Thesis, McGill University, June 1988.

[9] Gosselin, C. and Angeles, J., "*The optimum kinematic design of a planar three degree of freedom parallel manipulator*". Trans. ASME, Journal of Mechanisms, Transmission and Automation in Design, 1988, Vol. 110, pp. 35-41.

[10] Cyril, X., Angeles, J. and Misra, A. K., "*FLEXLINK: A sofware package for the dynamic simulation of flexible manipulators*". Proceedings of 1989 ASME International Computers in Engineering Conference, Anaheira, CA, July 30-Aug. 3, pp. 453-459, 1989.

[11] Daniali, H. R., Zsombor-Murray, P. J. and Angeles, J., *"On singularities of planar parallel manipulators"*, CIM Rpt. 93-05-20, 7pp.

[12] Do, D. Q. W. and Yand, D. C. H., *"Inverse dynamics analysis of a platform type of robot"*. Journal of Robotic Systems, Vol. 5, No. 3, pp. 210-227, 1988.

[13] Electrocraft Corporation, *"D.C. Motors, Speed Controls, Servo Systems"*. MINN., 1972.

[14] Fichter, E. F and McDowell, E. D, *"A Novel-Design for a robot arm"*. Advances in computer Technology, an ASME Publication, 1980, pp. 250-256

[15] Fichter, E. F., *"A Stewart platform-based manipulator: general theory and practical consruction"*. The International Journal of Robotic Research, Vol. 5, No. 2, pp. 157-182, 1986.

[16] Felton, P., *"The design of a transputer interface"*, internal report, McGill 1992.

[17] *"HC05 M68HC705K1 Technical Data"*, Motorola.

[18] Helmy, A., *"Introduction to parallel processing and the kinematics and dynamics of a three degree of freedom parallel manipulator"*, M.Eng. Thesis, McGill University, April 1993.

[19] Hennessey, M. P., Priebe, J. A., Paul, C. H. and Grommes, R. J., *"Design of a lightweight Robotic Arm and Controller"*. IEEE Conference on Robotics and Automation, Raleigh, North Carolina, pp. 779-785, 1987.

[20] Hoffman, R. and McKinnon, M. C., *"Vibration modes of an Aircraft Simulator Motion System"*. Proceedings of the 5th World Congress for the theory of Machines and Mechanisms, an ASME Publication, 1979, pp. 611-616.

[21] Hollerbach, J. M. and Suh, K.C, *"Redundancy resolution of manipulators through torque optimazation"*. Proceedings 1985 IEEE International Conference, Robotics and Automation, Silver Spring, IEEE Computer Society Press, pp. 1016-1022.

[22] Hunt K.H, *"Kinematic Geometry of Mechanisms"*. Oxford University Press, London 1978.

[23] Hunt, K.H, *"Structural kinematics of In-Parallel Actuated Robot arms"*. ASME Journal of Mechanisms Transmissions and Automation in Design, Vol. 105, Dec. 1983, pp. 705-712.

[24] Klein, C. A. and Huang, C. H., *"Review of pseudoinverse control for use with kinematically redundant manipulators"*. IEEE Trans. Sys., Man, pp. 245-250.

[25] Inmos, *"The T425 Transputer data book"*, Inmos, 1989.

[26] Inmos, "Occam Manual". Inmos, 1989.

[27] Jacobsen, S. C., Iversen, E. K., Knutti, D. F., Johnson, R. T. and Biggers, K. B., *"Design of the UTAH/M.I.T. Dextrous Hand : Work in Progress"*. The International Journal of Robotics Research, Vol. 3, No. 4, pp. 21-50, 1984.

[28] Kleinfinger, J. F. and Khalil, W., *"Dynamic modelling of robots with closed kinematic chains"*. Submitted to the International Journal of Robotics and Automation, 1986.

[29] Kokkinis, T., *"Dynamic hybrid control of cooperating robots by non-linear inversion"*. Journal of Robotics, 1989, Vol. 5, No. 4, pp. 359-368.

[30] Kuo, B. C. and Tal, J., *"D.C Motors and Control Systems"*. SRL Publishing Company, 1978.

[31] Lee, K. M. and Chao, A., *"On the analysis of a three degree of freedom manipulator"*. International Journal of Robotics and Automation, Vol. 3, No. 2, pp. 90-96, 1988.

[32] Lee, K. and Shah, D. K., *"Kinematic analysis of a three degree of freedom in-parallel actuated manipulator"*. IEEE, Journal of Robotics and Automation, 1988, Vol. 4, No. 3, pp. 354-360.

[33] Liergeois, A., *"Automatic supervisory control for the configuration and behaviour of multibody mechanisms"*. IEEE Trans. Sys., 1977, Man, Cyber. SMC-7(12), pp. 842-868.

[34] Luh, J. Y. S. and Zhang, Y. F., *"Computation of input generelized forces for robots with closed kinematic chain mechanisms"*. IEEE Journal of Robotics and Automation, Vol. 2, pp. 95-103, 1985.

[35] *"M68HC705K1CS In-Circuit Simulator User's Manual"*. Motorola CSIC Microcontroller Division.

[36] Ma, O., *"Mechanical Analysis of Parallel Manipulators with Simulation, Design and Control applications"*. Ph.D. Thesis, McGill University, May 1991.

[37] Ma, O. and Angeles, J., *"Direct kinematics and dynamics of a planar parallel manipulator"*. Proceedings ASME Design and Auto. Conference, 1989, September, Montreal, Vol. 3, pp. 313-320.

[38] Macierjewski, A. A. and Klein C. A., *"Obstacle avoidance for kinematically redundant manipulators in dynamically varying envirc-ments"*. In Robotics Research: The Second International Symposium, eds. H. Hanafusa and H. Inoue, 1985, pp. 155-162. Cambridge MIT Press.

[39] *"Maxon precision motors"*, edition 1990-1991.

[40] McCallion, H. and Truong, P. D., *"The analysis of a 6-dof workstation for Mechanised Assembly"*.Proceeding of the 5th World Congress for the theory of machines and Mechanisms, an ASME Publication, 1979, pp. 611-616.

[41] McCarthy, J. M. (Editor), *"Kinematics of robot manipulators"*. A collection of papers published in the International Journal of Robotics Research, M.I.T. Press, Cambridge, MA, 1987.

[42] Merlet, J-P., *"Les Robots Paralleles"*. Hermes, Paris, 1990.

[43] Nakamura, Y. and Ghodoussi, M., *"Dynamic computation of closed-link robot mechanisms with nonredundant and redundant actuators"*. IEEE Trans. on Robotics and Automation, Vol. 5, No. 3, pp. 294-302, 1989.

[44] Pahl, G. and Beitz, W., *"Engineering design"*. Springer-Verlag, Berlin, 1977.

[45] Rivin, E. I., *"Mechanical design of robots"*. McGraw Hill, 1988.

[46] Rodenaker, W. G.," *Methodishes Konstuieren"*. Springer-Verlag, Berlin, 1970.

[47] Salisbury, J. K. and Abramowitz, J. D, *"Design and control of a redundant mechanism for small motion"*. Proceedings, 1985, IEEE Conference of Robotics and Automation, March 25-28, 1985, St.Louis.

[48] Salisbury, J. K., Townsend, W., Eberman, B. and Dipietro, D., *"Preliminary Design of a Whole-Arm Manipulation System (WAMS)"*. IEEE Conference on Robotics and Automation, Philadelphia, PA, pp. 254-260, 1988.

[49] Say, M. G. and Taylor, E. O., *"Direct Current Machines"*. John Wiley and Sons, New York, 1980.

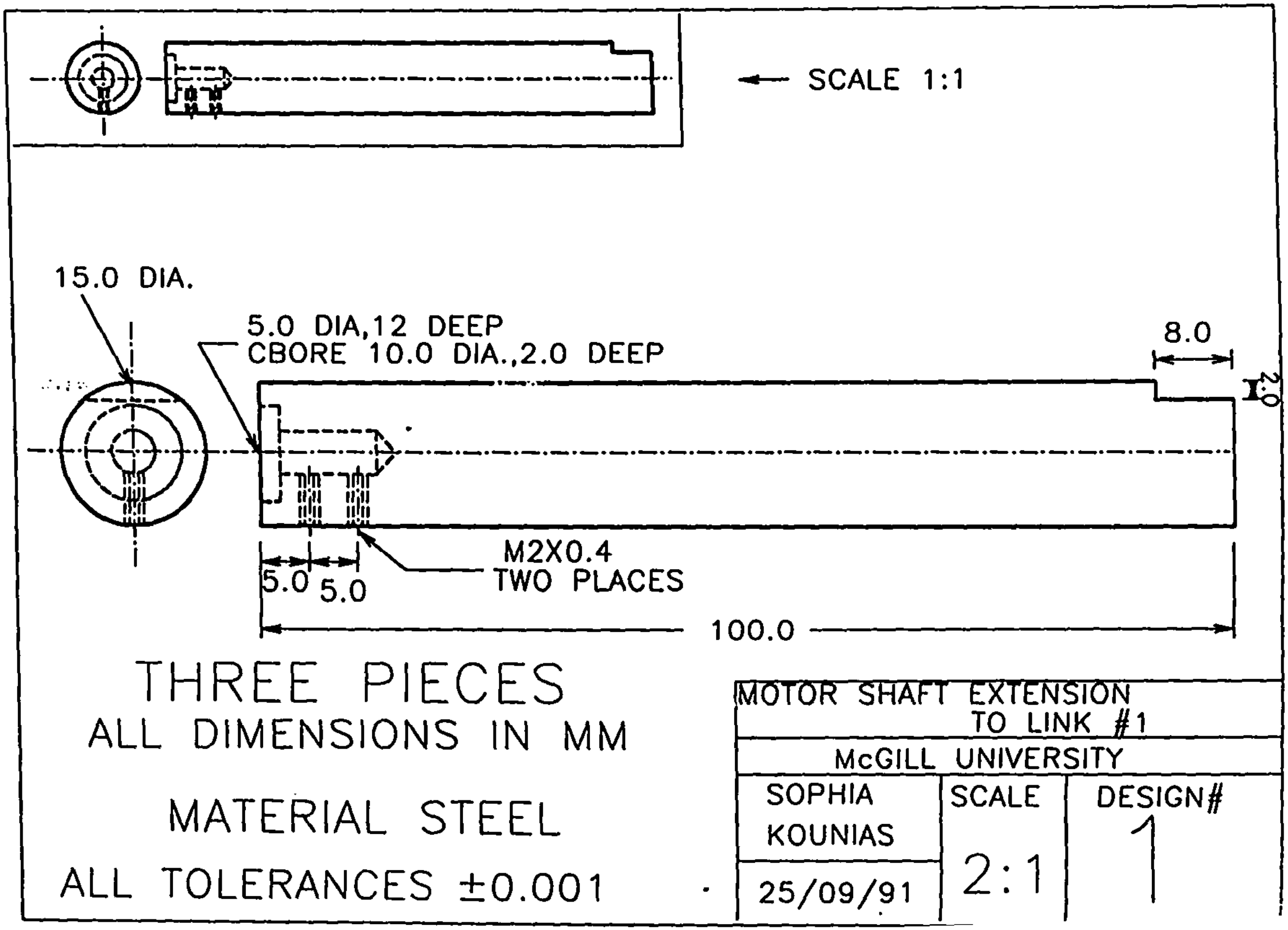
[50] Sibigtroth, J. M., *"Understanding small microcontrollers"*. Motorola CSIC Microcontroller Division.

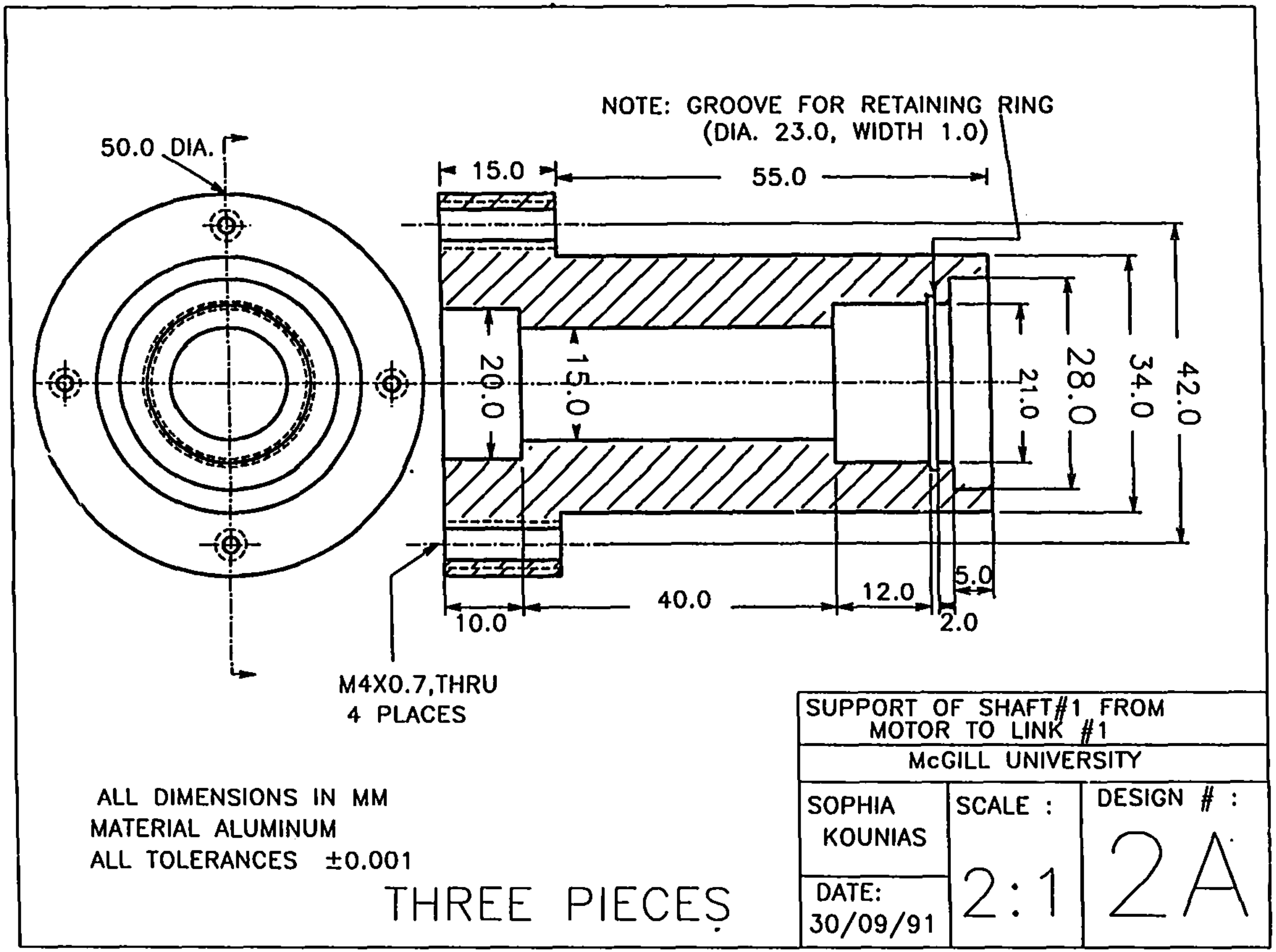
[51] Stewart, D., "*A Platform with six degrees of freedom*". Proceedings of the Institution of Mechanical Engineers, Vol. 180, Part 1, No. 15, 1965-66, pp. 371-386.

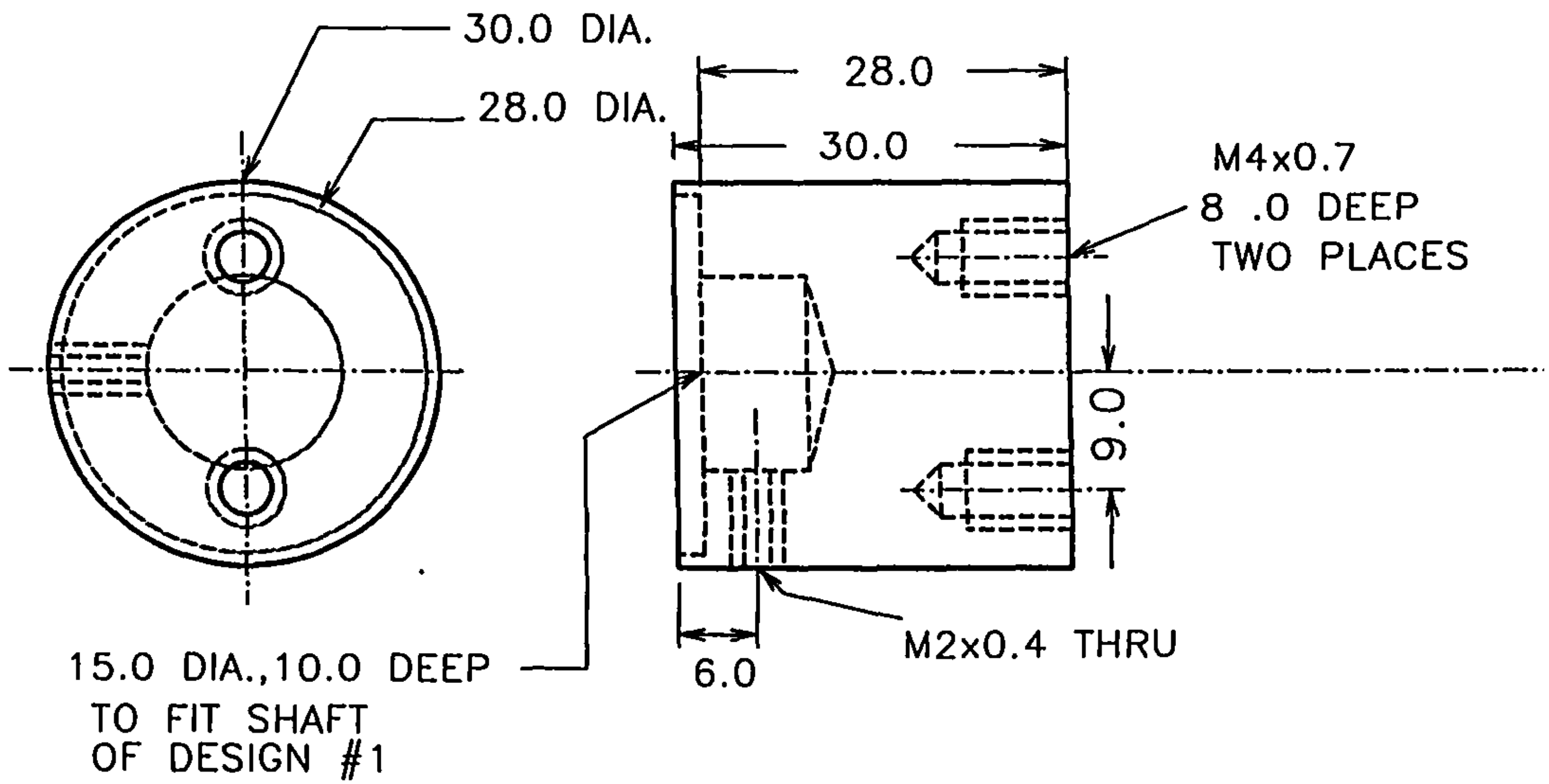
[52] Williams, O., "*Kinematics and Design of Robotic Manipulators with Complex Architectures*". M.Eng. Thesis, McGill University, May 1989.

[53] Wittenburg, J., "*Dynamics of Systems of Rigid Bodies*". B. G. Teubner, Stuttgart, Germany, 1977.

Appendix A Mechanical design of the 3-DOF parallel manipulator


SCALE 1:1
15.0 DIA.
5.0 DIA,12 DEEP
CBORE 10.0 DIA.,2.0 DEEP
8.0
2.0
M2X0.4
TWO PLACES
5.0
5.0
100.0
THREE PIECES
ALL DIMENSIONS IN MM
MATERIAL STEEL
ALL TOLERANCES ±0.001
MOTOR SHAFT EXTENSION TO LINK #1
McGILL UNIVERSITY
SOPHIA KOUNIAS
SCALE
2:1
DESIGN#
1
25/09/91


NOTE: GROOVE FOR RETAINING RING
(DIA. 23.0, WIDTH 1.0)
50.0 DIA.
15.0
55.0
20.0
15.0
21.0
28.0
34.0
42.0
10.0
40.0
12.0
5.0
2.0
M4X0.7,THRU
4 PLACES
ALL DIMENSIONS IN MM
MATERIAL ALUMINUM
ALL TOLERANCES ±0.001
THREE PIECES
SUPPORT OF SHAFT#1 FROM
MOTOR TO LINK #1
McGILL UNIVERSITY
SOPHIA
KOUNIAS
DATE:
30/09/91
SCALE :
2:1
DESIGN # :
2A



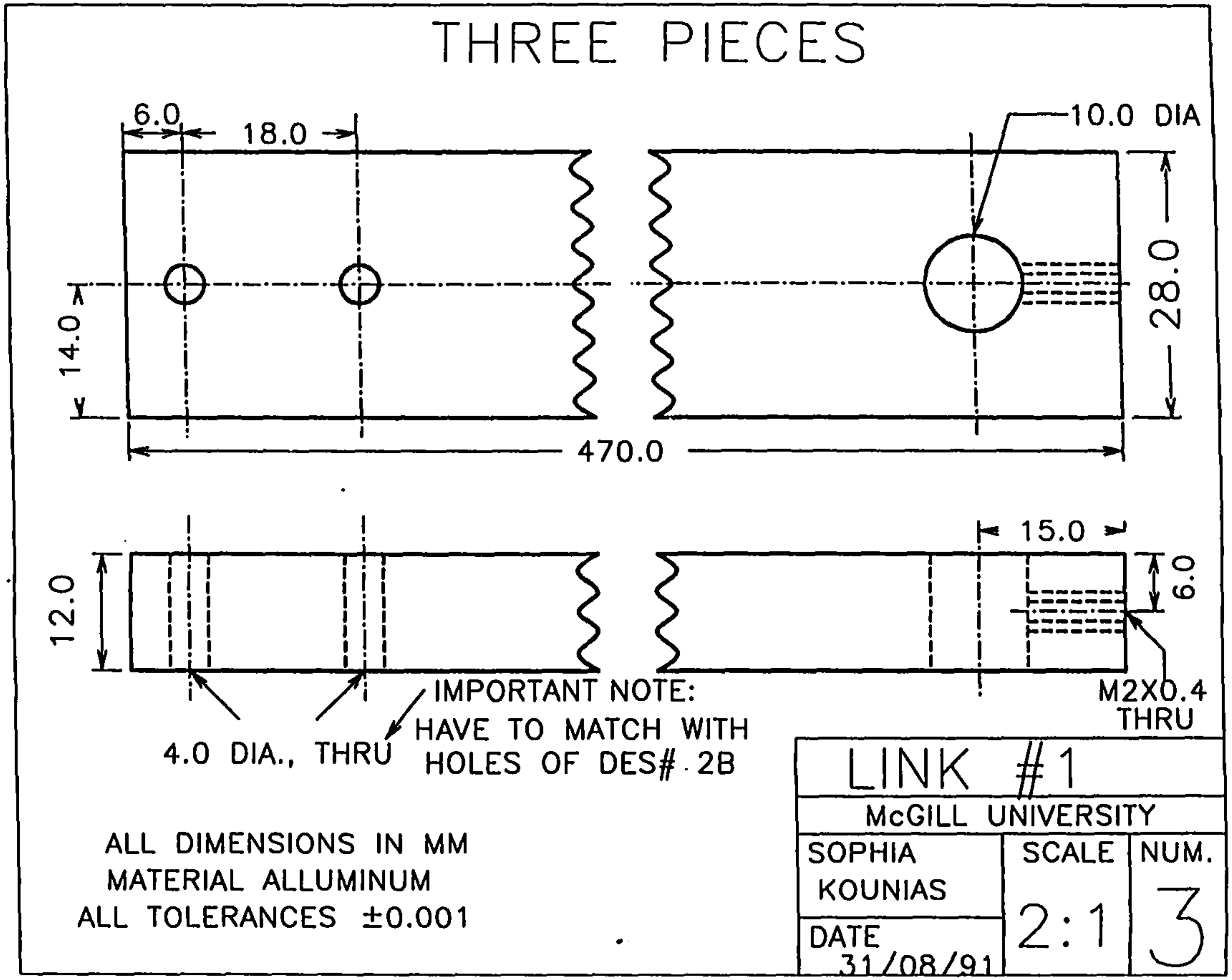
THREE PIECES

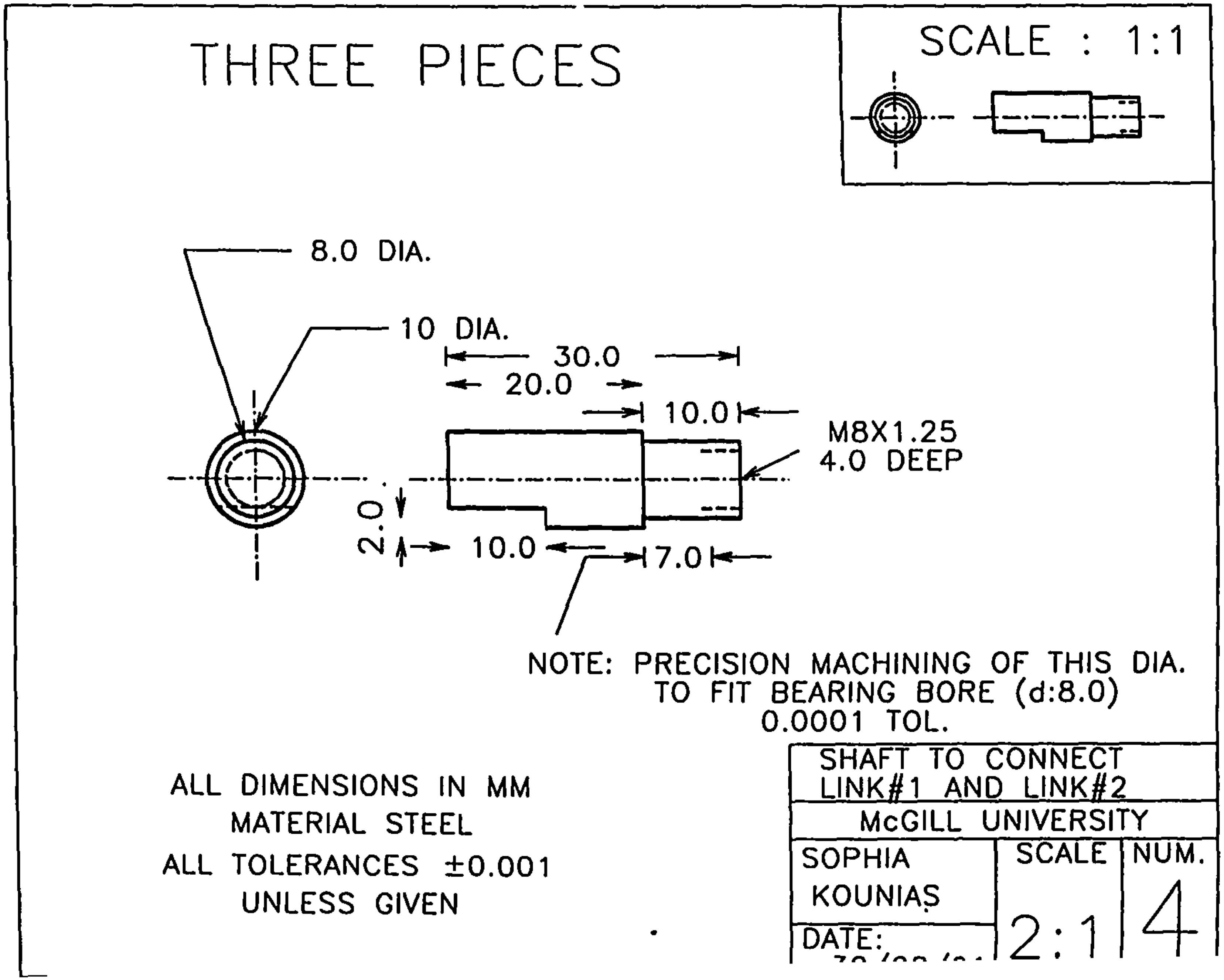
ALL DIMENSIONS IN MM

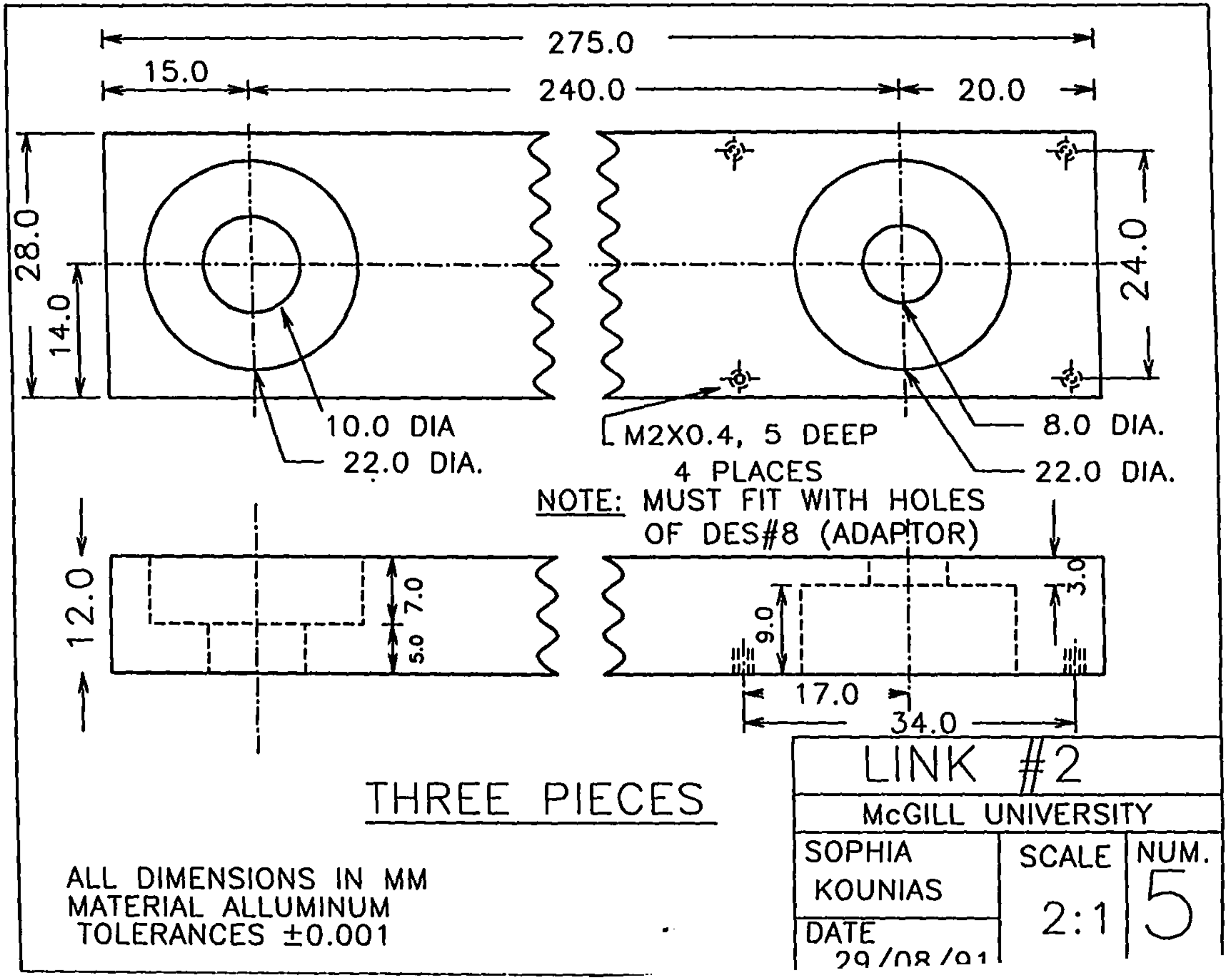
MATERIAL ALLUMINUM

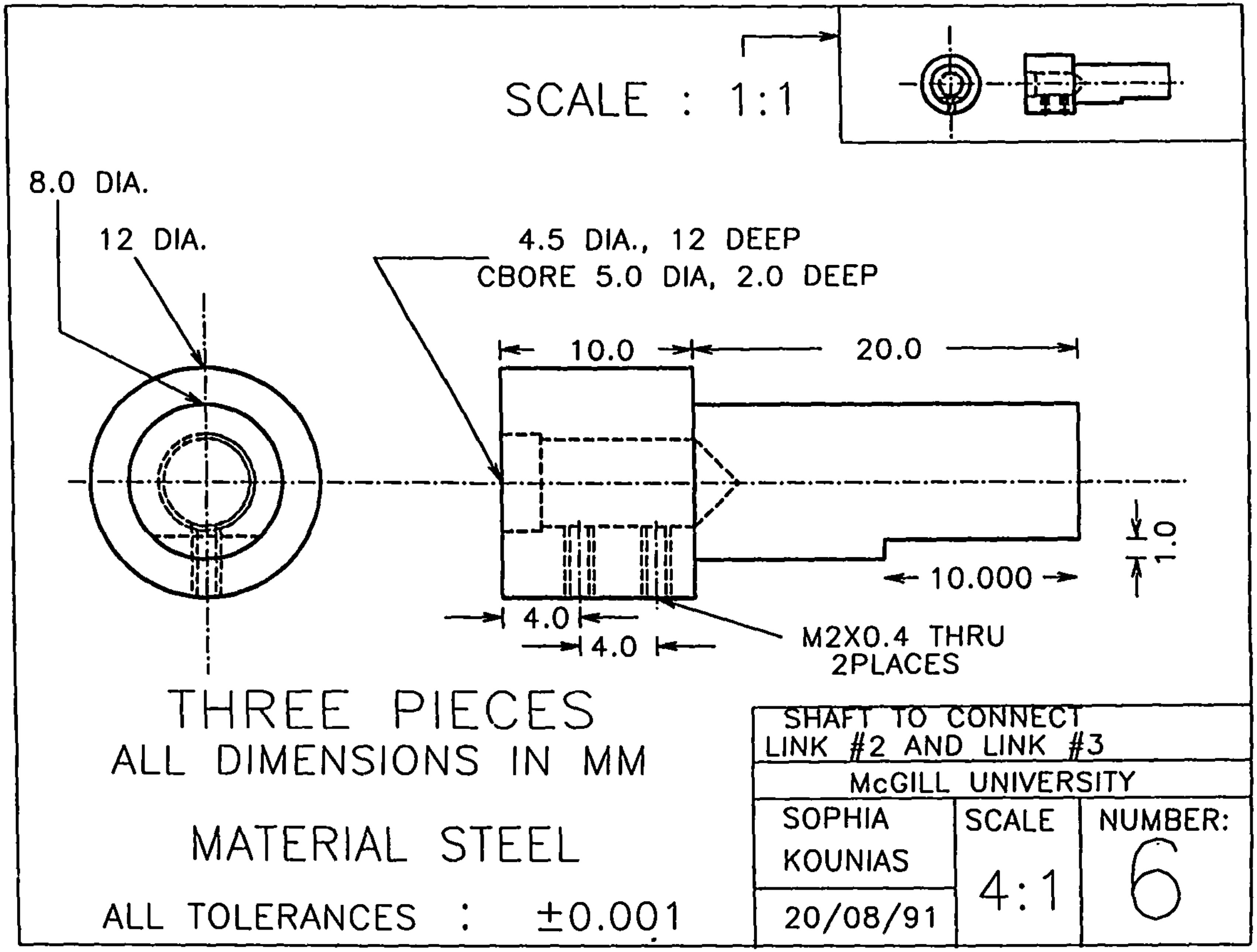
ALL TOLERANCES ±0.001

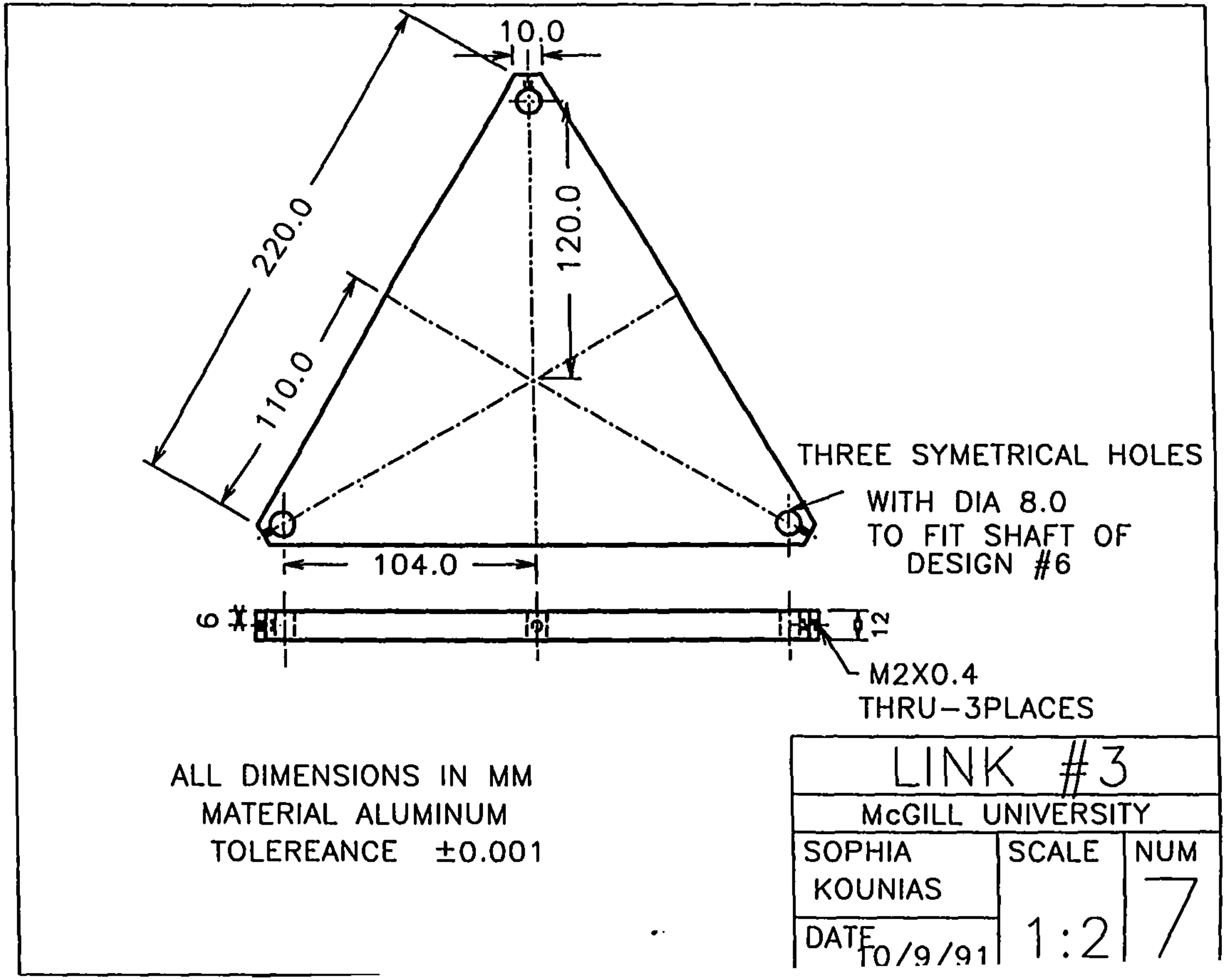
SUPPORT OF SHAFT#1 THAT CONNECTS IT WITH LINK#1		
McGILL UNIVERSITY		
SOPHIA KOUNIAS	SCALE	NUMBER:
30/09/91	2:1	2B

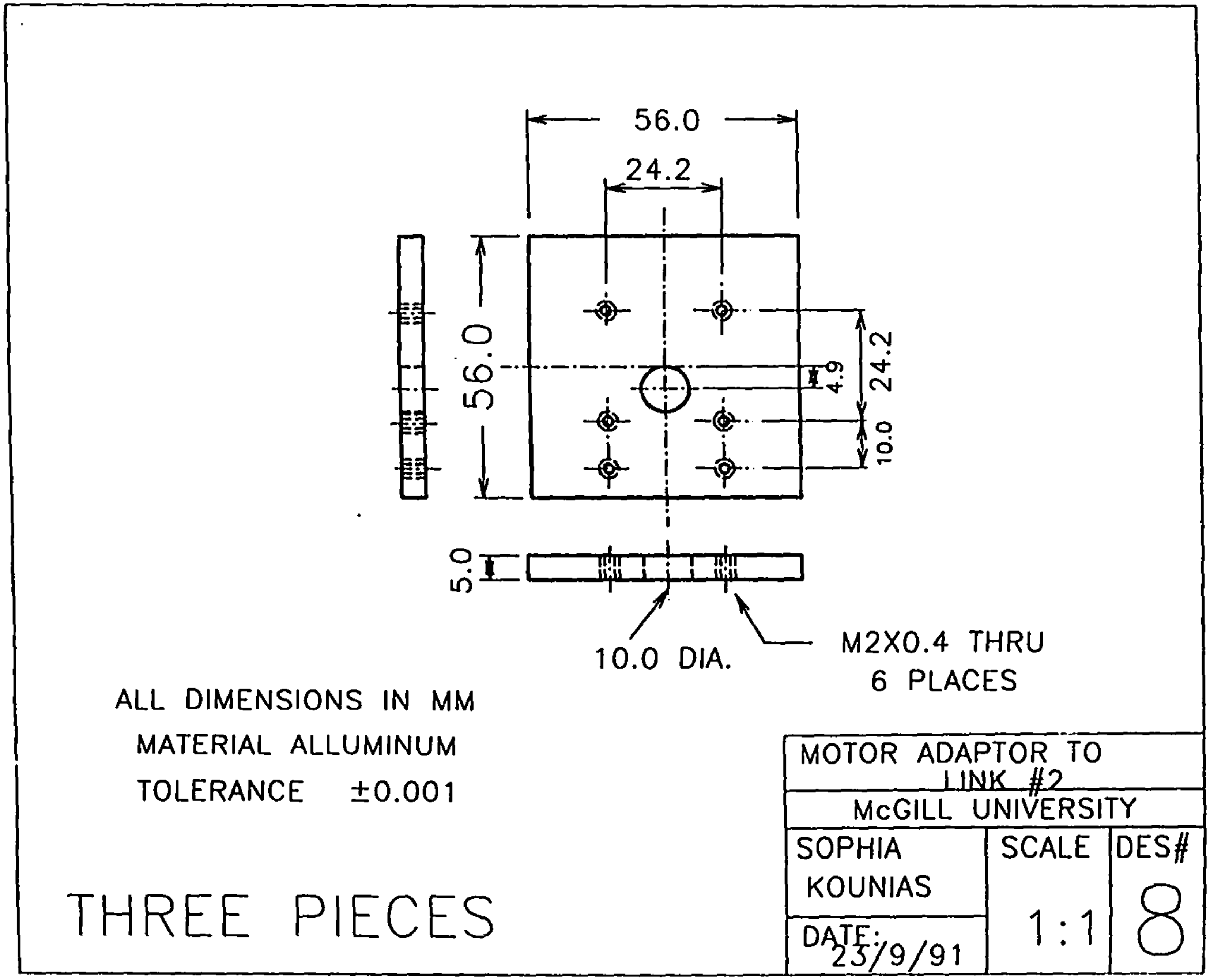

THREE PIECES
6.0
18.0
10.0 DIA
14.0
28.0
470.0
15.0
12.0
6.0
M2X0.4
THRU
4.0 DIA., THRU
IMPORTANT NOTE:
HAVE TO MATCH WITH
HOLES OF DES# 2B
LINK #1
McGILL UNIVERSITY
SOPHIA
KOUNIAS
SCALE
NUM.
2:1
3
DATE
31/08/91
ALL DIMENSIONS IN MM
MATERIAL ALLUMINUM
ALL TOLERANCES ±0.001


THREE PIECES
SCALE : 1:1
8.0 DIA.
10 DIA.
30.0
20.0
10.0
M8X1.25
4.0 DEEP
2.0
10.0
7.0
NOTE: PRECISION MACHINING OF THIS DIA.
TO FIT BEARING BORE (d:8.0)
0.0001 TOL.
ALL DIMENSIONS IN MM
MATERIAL STEEL
ALL TOLERANCES ±0.001
UNLESS GIVEN
SHAFT TO CONNECT
LINK#1 AND LINK#2
McGILL UNIVERSITY
SOPHIA
KOUNIAS
DATE:
SCALE
2:1
NUM.
4


275.0
15.0
240.0
20.0
28.0
14.0
24.0
10.0 DIA
22.0 DIA.
M2X0.4, 5 DEEP
4 PLACES
NOTE: MUST FIT WITH HOLES
OF DES#8 (ADAPTOR)
8.0 DIA.
22.0 DIA.
12.0
7.0
5.0
9.0
3.0
17.0
34.0
THREE PIECES
ALL DIMENSIONS IN MM
MATERIAL ALLUMINUM
TOLERANCES ±0.001
LINK #2
McGILL UNIVERSITY
SOPHIA
KOUNIAS
DATE
29/08/91
SCALE
2:1
NUM.
5


SCALE : 1:1
8.0 DIA.
12 DIA.
4.5 DIA., 12 DEEP
CBORE 5.0 DIA, 2.0 DEEP
10.0
20.0
1.0
10.000
4.0
4.0
M2X0.4 THRU
2PLACES
THREE PIECES
ALL DIMENSIONS IN MM
MATERIAL STEEL
ALL TOLERANCES : ±0.001
SHAFT TO CONNECT
LINK #2 AND LINK #3
McGILL UNIVERSITY
SOPHIA
KOUNIAS
SCALE
4:1
NUMBER:
6
20/08/91


10.0
220.0
120.0
110.0
104.0
6
12
THREE SYMETRICAL HOLES
WITH DIA 8.0
TO FIT SHAFT OF
DESIGN #6
M2X0.4
THRU-3PLACES
ALL DIMENSIONS IN MM
MATERIAL ALUMINUM
TOLEREANCE ±0.001
LINK #3
McGILL UNIVERSITY
SOPHIA
KOUNIAS
SCALE
NUM
DATE 10/9/91
1:2
7


56.0
24.2
56.0
4.9
24.2
10.0
5.0
10.0 DIA.
M2X0.4 THRU
6 PLACES
ALL DIMENSIONS IN MM
MATERIAL ALLUMINUM
TOLERANCE ±0.001
THREE PIECES
MOTOR ADAPTOR TO LINK #2
McGILL UNIVERSITY
SOPHIA KOUNIAS
SCALE
1:1
DES#
8
DATE: 23/9/91

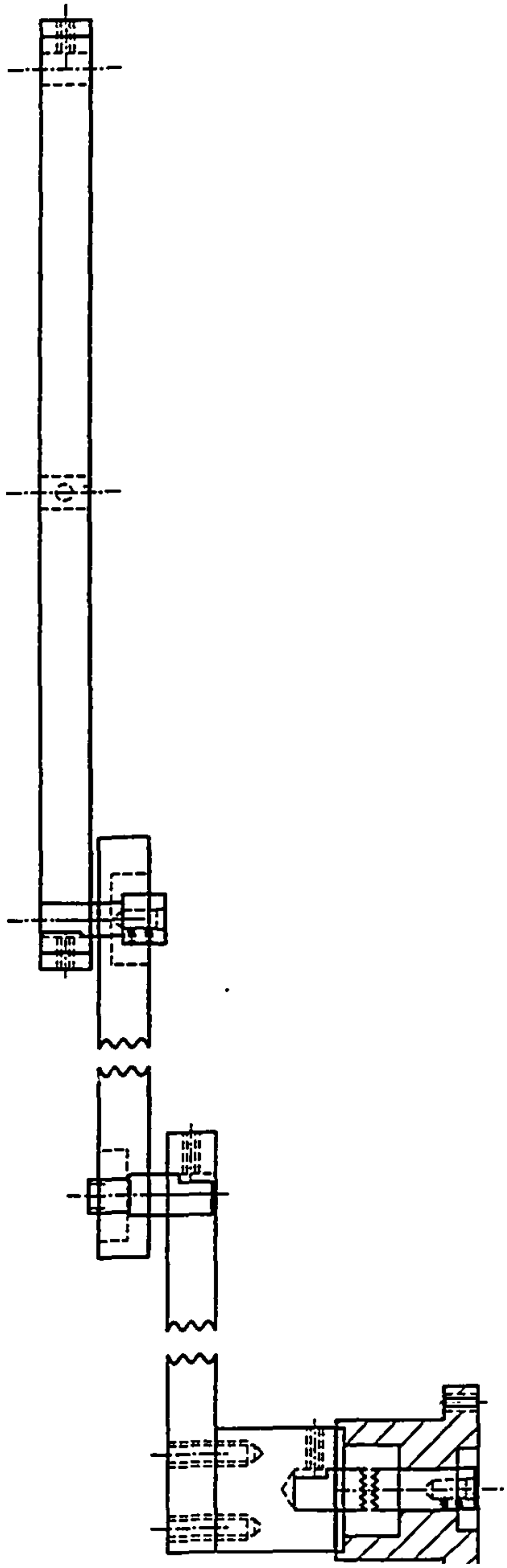


Figure A.1: Assembly design of one arm of the prototype at maximum extension/singularity

Appendix B List of Companies for actuator selection

In this appendix, a list of companies that were eliminated because their motors were unsuitable for the three degree-of-freedom parallel manipulator are mentioned. The following motor companies were eliminated because their products deviated too from the design criteria.

- Baldor Electric
- Molor Motor and Coil
- Motronics Corp
- RAE Corporation
- Yaskawa Corp of America

Detailed information on the following companies was sought. Comparing the data of their motors with the Maxon precision motor data, it was concluded that Maxon precision motors were smaller and more suitable for our application.

- Clifton Precision
- Panasonic
- PMI
- Pittman
- Motor Technology Inc.
- Apcor
- Motor Search Co.
- Inertial Motors Corporation

- Autocontrol Corporation
- Novatronics of Canada Ltd
- Precilec, France.

Appendix C Optical encoder technical data and test circuit

Technical Data

Supply Voltage: 5V ± 5%

Output Signal: TTL compatible

Output Current I_L at C_1, C_2: 16mA

Counts per turn: 100

Number of Channels: 2

Phase shift: 90deg ± 20deg

Max. operating Frequency: 100 kHz

Inertia of Encoder Disk: 0.12 gcm^2

Important points

- Encoder disk protected against dust
- Motor shaft and encoder disk are rigidly bounded
- Encoder does not require bearings
- Motor with preloaded bearings
- Output signal: $U_{high} = V_{cc} - I_L \times 10k\Omega$
- ON/OFF relationship to be trimmed using two external potentiometers

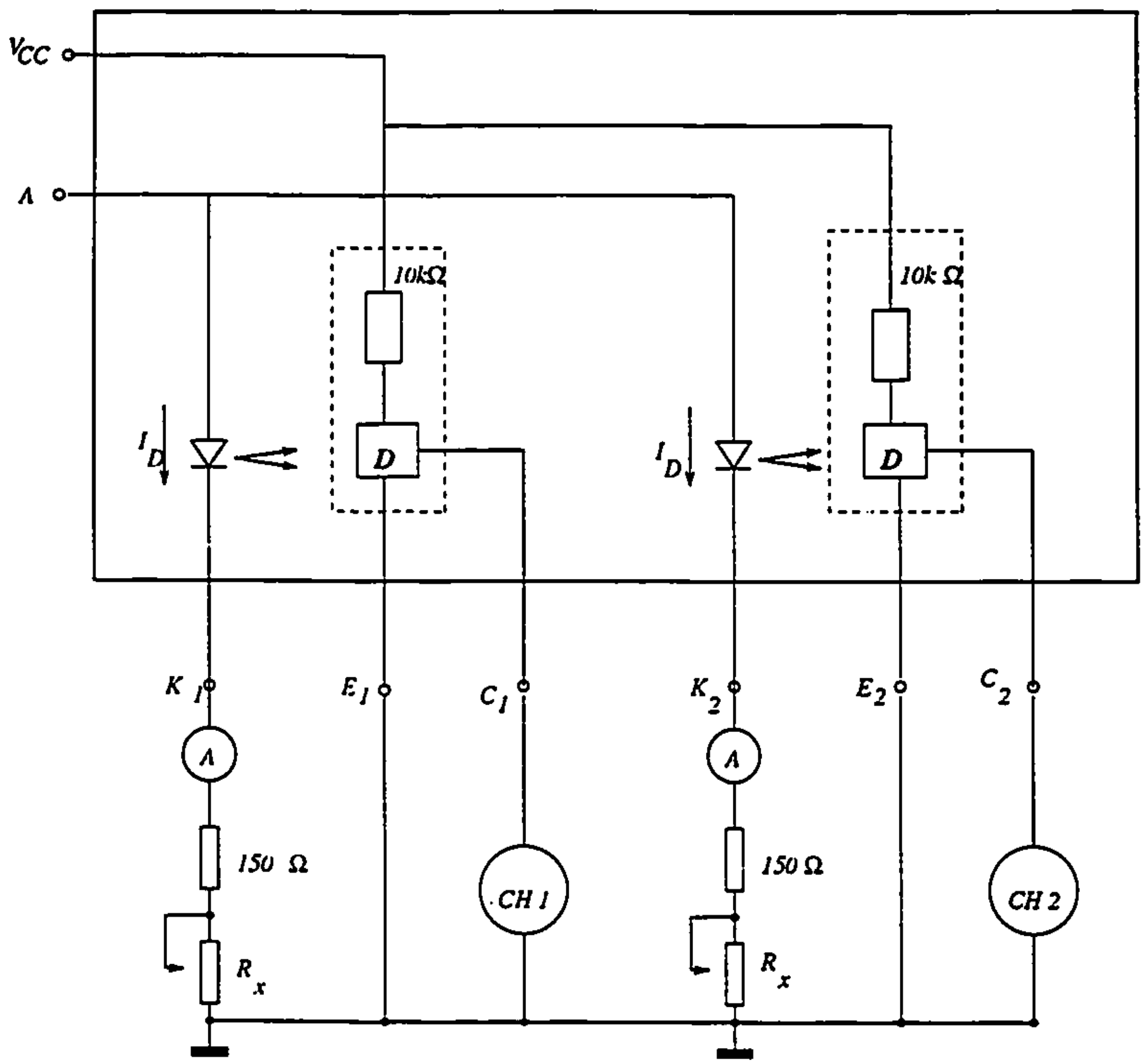


Figure C.1: Testing circuit for the encoder

Appendix D Values of components of control circuit

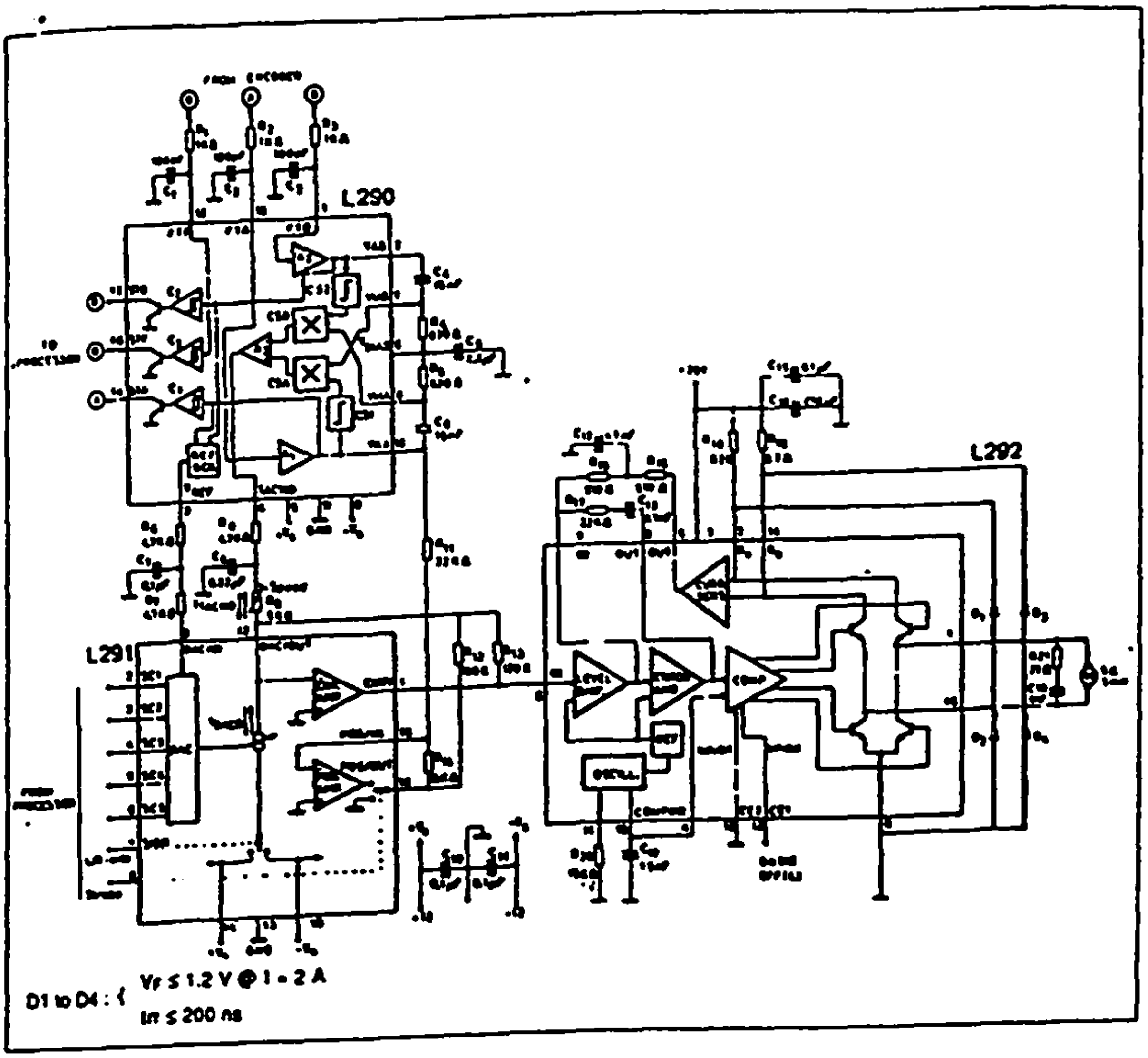


Figure D.1: Final control circuit diagram

Component	Value	Purpose
$R_1 R_2 R_3$	1 KΩ	To filter the noise on the enc. signal
$R_4 R_5$	1 KΩ	Differentiator network
R_6	5.6 KΩ	Set the D/A input current
R_7	6.8 KΩ	Set the D/A input current
R_8	3.3 KΩ	To set the motor speed
R_9	5 KΩ	To adjust the motor speed
R_{11}	22 KΩ	Set the position loop gain.
R_{12}	100 KΩ	Set the position loop gain.
R_{13}	120 KΩ	Set the speed loop gain.
R_{14}	15 KΩ	Set the position loop gain.
$R_{15} R_{16}$	560 Ω	To filter the feedback current
R_{17}	12 KΩ	Set the gain of the err. ampilfier
$R_{18} R_{19}$	2.2 Ω	Set the transconductance value of the L292
R_{20}	15 KΩ	Set the oscillator frequency
R_{21}	33 Ω	Compensation network
$C_1 C_2 C_3$	100 pF	To filter the noise on the enc. signal
$C_4 C_6$	15 nF	Differentiator network
C_5	2.2 μF	By-pass capacitor
C_7	0.1 μF	Low pass filter for D/A input current
C_8	0.22 μ	low pass filter for tacho signal
C_{10} C_{11}	0.1 μF	Supply by-pass capacitor
C_{12}	47nF	Filter the feedback current
C_{13}	47 nF	To set the gain of err. amplifier
C_{15}	0.1 μF	Supply by-pass capacitor
C_{16}	470 μF	Supply by-pass capacitor
C_{17}	1.5 nF	Set the oscillator frequency
C_{18}	1 nF	Compensation network
$D_1 D_2 D_3 D_4$	1 A	Recirculation diodes.

Table D.1: Values of additional components of control circuit

Appendix E Source and machine code of experiment

```
                *************************************************
                *                EXPERIMENT #1                  *
                *This program is to be programmed in a M68HC05 micro-  *
                *processor. The processor must provide voltage to the  *
                *controller of the parallel manipulator, move the      *
                *actuated link 45 degrees, after detecting the 45 Deg. *
                *movement make the link change direction and move back *
                *to the initial position.                              *
                *************************************************
                *Initializing memory position and input-output ports *
                *************************************************
0200            ROM EQU $200
0200                ORG ROM
0200 A6FF           LDA #$FF     ; INITIALIZE PORTA AS OUTPUTS
0202 B704           STA DDRA
0204 A600           LDA #0       ; INITIALIZE PORTB AS INPUTS
0206 B705           STA DDRB
                *************************************************
                *               MAIN PROGRAM                    *
                *************************************************
0208 A63E       MAIN LDA #$3E   ;Start motor with maximum Clockwise speed
020A B700           STA PORTA
020C CD021C         JSR CH1      ;go to subroutine CH1
020F A600           LDA #$00     ;change motor to maximum Counterclockwise
                                                                   speed
0211 B700           STA PORTA
0213 CD022B         JSR CH2      ;go to subroutine CH2
0216 A6FF           LDA #$FF     ;Stop motor
0218 B700           STA PORTA
021A 20EC           BRA MAIN  ;start again motion
                *************************************************
                *Subroutines CH1,CH2 detect 45 degrees of motion   *
                *counting the rising edges of the shaft encoders outputs*
                *************************************************
021C AE18       CH1 LDX #$18; Load x register with #$18
021E 0101FD     LOOP1 BRCLR 0,PORTB,LOOP1 ; If port b is low goto loop1
0221 5A               DECX ;Decrement x
0222 2706             BEQ DONE ; If x = 0 goto  DONE
0224 0001FD     LOOP2 BRSET 0,PORTB,LOOP2 ; If portb is high goto loop2
0227 CC021E           JMP LOOP1 ; goto loop1
022A 81         DONE  RTS ; return to program
022B AE18       CH2   LDX #$18
022D 0101FD     LOOP1B BRCLR 0,PORTB,LOOP1B
0230 5A               DECX
0231 2706             BEQ DONEB
0233 0001FD     LOOP2B BRSET 0,PORTB,LOOP2B
0236 CC022D           JMP LOOP1B
0239 81         DONEB  RTS
03F8                    ORG VECTORS
03F8 0200               DW ROM
03FA 0200               DW ROM
03FC 0200               DW ROM
03FE 0200               DW ROM
```

Symbol Table

CH1	021C
CH2	022B
DONE	022A
DONEB	0239
LOOP1	021E
LOOP1B	022D
LOOP2	0224
LOOP2B	0233

```
                  *******************************************************
                  *                 EXPERIMENT #2                         *
                  *This program is to be programmed in a M68HC05 micro-   *
                  *processor. The processor must provide voltage to the   *
                  *controller of the parallel manipulator, move the       *
                  *actuated link 22.5 Deg, after detecting the 22.5 Deg.  *
                  *movement change speed, continue movement until reaching
                  *45 Deg, change direction and after 22.5 Deg change     *
                  *speed until reaching initial position and stop.        *
                  ******************************************************
                  *Initializing memory position and input-output ports *
                  ******************************************************
0200              ROM EQU $200
0200                  ORG ROM
0200 A6FF             LDA #$FF   ; INITIALIZE PORTA AS OUTPUTS
0202 B704             STA DDRA
0204 A600             LDA #0     ; INITIALIZE PORTB AS INPUTS
0206 B705             STA DDRB
                  ***************************************************
                  *          MAIN PROGRAM                           *
                  ***************************************************

0208 A620         MAIN LDA #$20 ; Start motion with min. clockwise speed.
020A B700             STA PORTA
020C CD022A           JSR CH1; goto subroutine CH1
020F A63E             LDA #$3E  ;Change speed with maximum Clockwise speed
0211 B700             STA PORTA
0213 CD022A           JSR CH1    ;go to subroutine CH1
0216 A600             LDA #$00   ;change speed to maximum Counterclockwise
0218 B700             STA PORTA
021A CD0239           JSR CH2    ;go to subroutine CH2
021D A620             LDA #$20 ; Change speed to minimum Counterclockwise
021F B700             STA PORTA
0221 CD0239           JSR CH2 ; Goto subroutine CH2
0224 A6FF             LDA #$FF   ;Stop motor
0226 B700             STA PORTA
0228 20DE             BRA MAIN  ;start again motion
                  ***************************************************
                  *Subroutines CH1,CH2 detect 22.5 degrees of motion    *
                  *counting the rising edges of the shaft encoders outputs*
                  ***************************************************
022A AE0C         CH1 LDX #$C; Load x register with #$C
022C 0101FD       LOOP1 BRCLR 0,PORTB,LOOP1 ; If port b is low goto loop1
022F 5A                 DECX ;Decrement x
0230 2706               BEQ DONE ; If x = 0 goto  DONE
0232 0001FD       LOOP2 BRSET 0,PORTB,LOOP2 ; If portb is high goto loop2
0235 CC022C             JMP LOOP1 ; goto loop1
0238 81           DONE  RTS ; return to program
0239 AE0C         CH2   LDX #$C
023B 0101FD       LOOP1B BRCLR 0,PORTB,LOOP1B
023E 5A                  DECX
023F 2706                BEQ DONEB
0241 0001FD       LOOP2B BRSET 0,PORTB,LOOP2B
0244 CC023B              JMP LOOP1B
0247 81           DONEB  RTS

03F8                        ORG VECTORS
03F8 0200                   DW ROM
03FA 0200                   DW ROM
```

```
03FC 0200          DW ROM
03FE 0200          DW ROM
```

Symbol Table

```
CH1        022A
CH2        0239
DONE       0238
DONEB      0247
LOOP1      022C
LOOP1B     023B
LOOP2      0232
LOOP2B     0241
MAIN       0208
ROM        0200
```