Monograph:

Zalzala, Ali. M.S. and Morris, A.S. (1990) Ultrasonic Location Measurement for Fast Robot Control: A Transputer-Based Environment. Research Report. Acse Report 380 . Dept of Automatic Control and System Engineering. University of Sheffield

Ultrasonic Location Measurement for Fast Robot Control : A Transputer-Based Environment

by:

Ali M.S. Zalzala and Alan S. Morris

Robotics Research Group,
Department of Control Engineering,
University of Sheffield,
Mappin Street,
Sheffield S1 3JD,
U.K.

Research Report # 380

January 1990

RR # 380

Abstract

The predominant form of operation for the presently available robot manipulators has been to teach the desired motion in a pre-programmed manner. However, the new generation of industrial robots are required to have a greater interactive ability for sensing the outside world, hence being able to function more satisfactorily in integrated automated environments. This paper presents an efficient position measurement device that gives the position and orientation of the robot end effector in real-time. An array of ultrasonic transmitters are employed along with multiple receivers to cover the required working area, thus avoiding the hazards of the presence of obstacles. Furthermore, a distributed form of the proposed algorithm has been developed on a multiprocessor system, which provides the equivalent joint-space configurations needed by the controller loops, while checking for the problems of redundant positions and singularities. The proposed system has been found to be able to track the desired trajectory at a frequency of (4.35 KHz), with the distributed algorithm implemented by a network of 12 T800 transputers. In addition, a case study has been constructed for the PUMA 560 robot manipulator which shows the efficiency of the system.

RR # 380

description of the electronic circuitry is shown in section 5. The principles of robot kinimatics are presented in section 6, while the measurement of orientation is described in section 7. The computational complexity of the whole procedure is assessed in section 8. Section 9 shows the full distributed structure of the device, while its practical implementation on an actual multiprocessor system is included in section 10. Practical tracking results along with real-time simulations for the PUMA 560 arm are shown in section 11. Finally, conclusions are drawn in section 12.

2. Statement of the Problem

An overview of an integrated robot system is shown in figure (1), where the task specifier has the job of directing the manipulator to some specific location in space. The trajectory planner is then activated, providing the time-history of motion required to accomplish the specified task. The parameters of motion produced by the planner, namely the joints positions, velocities and accelerations are then fed to the motion controller, producing the actuating values of torques (or forces) for each of the arm motors that will provide the planned movement of the end-effector. However, due to possible changes in the surrounding environment and/or the uncertainties of the dynamic model of the arm used by the controller, certain deviations in the end-effector motion are to be expected. Therefore, a feedback signal proportional to the deviation errors should be made available to the controller in an attempt to maintain the desired motion.

The feedback signal generator has two phases of operation, as follows :

□ Detecting the end-effector *position* and *orientation* in cartesian space.

□ Transforming the detected 3D location to its equivilant *joint-space positions* via the inverse kinematics algorithm. This is of extreme importance, since the control procedure is usually performed in the configuration-space of the robot, where constraints on the motors performance exist.

The problem with solving the inverse kinematics of a robot arm is that the outcome of the transformation is not unique. Hence, several configurations in the joint space could map on to one location in the 3D cartesian space. Therefore, the best choice amongst these *redundant* solutions must be selected. In addition, the presence of a *singularity* point would greatly affect the tracking performance.

Although the procedure of tracking the desired motion is considered to be quite a complicated job, the recent availability of fast motion planners [Zalzala and Morris 1989e, Zalzala and Morris 1989g] and controllers [Zalzala and Morris 1989i, Zalzala and Morris 1989a] make a feasable solution possible. Hence, the computational power involved in the function of the sensory position detector should be equally as fast to provide the error correction signals at a suitable control rate. Therefore, a suitable recasting of the procedures in a distributed structure would produce the ultimate solution to such a problem.

3. Ultrasonic Measurement Concepts

The main idea of distance measurements using ultrasonic waves is illustrated by figure (2), where the distance l between the transmitter and the receiver could be calculated. The required distance is given by

$$l = c \times t_l \tag{1}$$

where,

$c \equiv$ speed of sound, and

$t_l \equiv$ time required by sound to travel a distance l.

Hence, the travelling distance could be calculated once the time of propagation (t_l) and the speed of sound are known. However, the latter quantity is environment-dependent. A change in the surrounding *temperature* effects the speed of sound as follows [Dorf 1987],

$$c = 331.4 \times \sqrt{\left[\frac{T}{273}\right]} \tag{2}$$

where T is the absolute temperature of perfect gas (in Kelvin).

The effect of *humidity* should also be considered, in addition to the presence of air currents in the environment.

However, once $3D$ space is addressed, a more complicated configuration is expected, as shown in figure (3). In figure (3), the cartesian coordinates of the target point **P** are to be identified. Such a point represents the robot arm tip (or end-effector), in addition to bearing the source of ultrasonic waves. Therefore, trigonometric techniques are used to calculate the x, y and z coordinates of point **P** [Dickinson and Morris 1988]. Three ultrasonic receivers are placed at distinct locations within the coordinate system, denoted as UR_a,UR_b and UR_c in figure (3), where each of the latter two are separated from the former by a fixed distance of L_1 and L_2, respectively. The origin of the coordinate system shown is chosen as one corner of the work volume, and is defined as the *Environment Global Coordinate System,* while the *Robot Base Coordinate System* is placed in the lower centre of the volume.

Considering (eqn.1), the corresponding distances d_a,d_b and d_c are calculated, and the spatial coordinates of point **P** are found as

$$x = \frac{d_a^2 + L_1^2 - d_b^2}{2\,L_1} \tag{3}$$

$$y = \frac{d_a^2 + L_2^2 - d_c^2}{2\,L_2} \tag{4}$$

$$z = \sqrt{d_a^2 - x^2 + y^2} \tag{5}$$

It should be noted, however, that the coordinates calculated are with respect to the environment global coordinate system. These values should be transformed to the robot base coordinate system by using the translational transformation

$$\mathbf{P}_R = \mathbf{P}_E + \mathbf{P}_{ORG} \tag{6}$$

where,

$\mathbf{P}_R = (X,Y,Z)^T \equiv$ 3-D position of end-effector with respect to the robot base coordinate system,

$\mathbf{P}_E = (x,y,z)^T \equiv$ 3-D position of end-effector with respect to the environment global coordinate system, and

$\mathbf{P}_{ORG} = (X_b,Y_b,Z_b)^T \equiv$ 3-D translational vector, transfering the origin of the global coordinate system to that of the robot base system.

The robot base coordinate system is assumed to have no change of orientation over the global system. Hence, only a translational transformation is required, as indicated by (eqn.6).

The accuracy of these measurements depends heavily on the precision of both wave propagation and detection. Although this aspect of the design is beyond the scope of this work, it will be mentioned briefly in section 5 following.

4. Transducer Configurations

Since the range of transmitting and receiving angles of existing ultrasonic transducers is limited (typically 60-100 degrees for a 40KHz piezoelectric element), then the configuration shown by figure (3) would be inadequate for accurate position measurement. In the actual environment of a robot arm, the presence of any obstacles, or even a link of the arm itself in certain configurations, may cause one or more of the receivers to be hidden, thus causing an error in the measurement. In addition, the use of only one ultrasonic transmitter held at the robot tip would not be adequate to cover the whole work volume in all possible arm configurations.

Hence, in an attempt to overcome such difficulties, the use of multiple transmitters and receivers has been investigated.

4.1. Multiple Receivers

Considering a cubic work area of side L, a total of eight ultrasonic receivers have been assigned, one for each corner of the volume. Hence, twenty four combinations of receivers are possible to detect any point within the cube, where the failure of one or more of them would be tolerable. This arrangement is shown in figure (4), where ultrasonic transmitters and receivers are denoted as UT and UR_i, i=1,..,8, respectively. However, the computations described in (eqns.3-5) have to be repeated for each possible combination of these receivers, and then an average value of all measurements must be calculated as

$$q = \frac{1}{24} \sum_{i=1}^{24} q_i \qquad (7)$$

where $q \in \{ x, y, z \}$.

4.2. Multiple Transmitters

Considering an active tranmitting angle of 100 degrees, a single transmitter on the robot hand would not be adequate to cover the whole work volume. Hence, combinations of a number of transducers have been investigated. Different possible arrays of 4,5 and 7 transmitters are shown in figure (5), where the shaded spaces denote the dead areas where the waves are ineffective. Hence, using a larger array would give a better coverage of the work volume. It should be noted, however, that using a larger number of transmitters would increase the physical diameter of the array, which may become an obstacle when mounted on the robot end-effector. Therefore, a suitable size for the transmitting array should be chosen according to the needs of the application.

5. The Tracking System

The structure of the position tracking system is shown in figure (6), where all distances between the transmitting array and the eight receiving transducers can be measured. The *counter circuitry* connected to each receiver is activated once an ultrasonic burst is transmitted, and is stopped once it reaches its destination. Hence, the count will be proportional to the travelled distance [Dickinson and Morris 1988]. Once all distances have been measured, a suitable combination of them can be used to determine the required XYZ coordinates.

A practical implementation of the above system has been constructed with suitable electronic circuitary [Shahidi 1989], and an accuracy of (±1 mm) was achieved.

6. Robot Arm Kinematics

For the measured 3-D cartesian position to be of use in an integrated control system, its equivalent value in the robot N-joint space must be computed, where N denotes the number of joints on a given robot. The transformation between cartesian and joint coordinates is accomplished via the *inverse kinematics equations*. The purpose of this section is to give a brief description of these equations, and their application to the 6 revolute joint PUMA arm.

6.1. The Link Parameters

The kinematics equations describe the geometrical arm motion without regard to the forces causing motion [Craig 1986]. Since a robot manipulator consist of a series of links connected by the corresponding joints, the geometric relations between all these links must be established. These relations can be exploited by introducing the *link parameters,* which are better known as the Denavit-Hartenberg parameters [Denavit and Hartenberg 1955]. Hence, for a 6-axis revolute robot arm, the coordinate frame transformation from one joint to another is a function of joint and link parameters.

The relation between links i and $i-1$ in a chain is illustrated in figure (7), where the coordinate system of link i is attached with its origin on the joint i axis. The link parameters shown are defined as :

$d_i \equiv$ link offset, a signed distance measured along Z_{i-1} from X_{i-1} to X_i,

$a_i \equiv$ link length, measured along X_i from Z_{i-1} to Z_i,

$\alpha_i \equiv$ link twist, a signed quantity representing the angle measured about X_i from Z_{i-1} to Z_i, and

$\theta_i \equiv$ joint angle, a signed quantity measured about Z_{i-1} from X_{i-1} to X_i.

Hence, a set of these four parameters is to be provided for each link of the manipulator. If the associated joint is revolute, then θ_i would be the variable. Alternatively, d_i varies in the case of a prismatic joint. The link parameters for the PUMA 560 arm are shown in Table (1).

Table (1) : PUMA 560 Link Parameters

Link #	α_i (*degree*)	d_i (*m*)	a_i (*m*)	θ Range (*degree*)
1	-90	0.0000	0.0000	-160 to 160
2	0	0.0000	0.4318	-225 to 45
3	90	0.1505	-0.0191	-45 to 225
4	-90	0.4331	0.0000	-110 to 170
5	90	0.0000	0.0000	-100 to 100
6	0	0.0000	0.0000	-266 to 266

6.2. The Kinematics Formulation

Each coordinate system defined for each link of the manipulator is related to its neighbouring links. Hence, for links i and $i-1$ of figure (7), there exists a transformation relating both, having the following form :

$$\mathbf{T}_i^{i-1} = \left[\begin{array}{c|c} \begin{array}{c} Rotation \\ 3\times 3 \end{array} & \begin{array}{c} Translation \\ 3\times 1 \end{array} \\ \hline \begin{array}{c} Perspective \\ 3\times 1 \end{array} & \begin{array}{c} Scaling \\ 1\times 1 \end{array} \end{array}\right] \tag{8}$$

which may be defined as a basic homogeneous rotation-translation matrix

$$\mathbf{T}_i^{i-1} = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

where, $C\lambda = Cos(\lambda)$, $S\lambda = Sin(\lambda)$, $\lambda \in \{ \theta , \alpha \}$. Thus, a transformation between the base coordinate system and link N coordinate system can be defined as

$$\mathbf{T}_N^0 = \prod_{i=1}^{N} \mathbf{T}_i^{i-1} = \mathbf{T}_1^0 \, \mathbf{T}_2^1 \cdots \mathbf{T}_N^{N-1} \tag{10}$$

Considering a PUMA-like robot, the link coordinate systems can be attached as shown in figure (8a), where the relevant parameters are illustrated by the schematic diagram of figure (8b). Hence, considering the parameters of table (1), the transformation matrices between each two successive links could be defined as

$$\mathbf{T}_1^0 = \begin{bmatrix} C\theta_1 & 0 & S\theta_1 & 0 \\ S\theta_1 & 0 & -C\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$\mathbf{T}_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2C_2 \\ S\theta_2 & C\theta_2 & 0 & a_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

$$\mathbf{T}_3^2 = \begin{bmatrix} C\theta_3 & 0 & -S\theta_3 & a_3C_3 \\ S\theta_3 & 0 & C\theta_3 & a_3S_3 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

$$\mathbf{T}_4^3 = \begin{bmatrix} C\theta_4 & 0 & S\theta_4 & 0 \\ S\theta_4 & 0 & -C\theta_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{14}$$

$$\mathbf{T}_5^4 = \begin{bmatrix} C\theta_5 & 0 & -S\theta_5 & 0 \\ S\theta_5 & 0 & C\theta_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

$$\mathbf{T}_6^5 = \begin{bmatrix} C\theta_6 & -S\theta_6 & 0 & 0 \\ S\theta_6 & C\theta_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

We further define

$$\mathbf{T}_3^0 = \mathbf{T}_1^0\,\mathbf{T}_2^1\,\mathbf{T}_3^2$$

$$= \begin{bmatrix} C\theta_1C\theta_{23} & -S\theta_1 & -C\theta_1S\theta_{23} & C\theta_1C\theta_{23}a_3+C\theta_1a_2C\theta_2+S\theta_1d_3 \\ S\theta_1C\theta_{23} & C\theta_1 & -S\theta_1S\theta_{23} & S\theta_1C\theta_{23}a_3+S\theta_1a_2C\theta_2-C\theta_1d_3 \\ S\theta_{23} & 0 & C\theta_{23} & S\theta_{23}a_3+a_2S\theta_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{17}$$

as the transformation between the base system and the wrist system, while

$$\mathbf{T}_6^3 = \mathbf{T}_4^3\,\mathbf{T}_5^4\,\mathbf{T}_6^5$$

$$\begin{bmatrix} C\theta_4C\theta_5C\theta_6-S\theta_4S\theta_6 & -C\theta_4C\theta_5S\theta_6-S\theta_4C\theta_6 & -C\theta_4S\theta_5 & 0 \\ S\theta_4C\theta_5C\theta_6+C\theta_4S\theta_6 & C\theta_4C\theta_6-S\theta_4C\theta_5S\theta_6 & -S\theta_4S\theta_5 & 0 \\ S\theta_5C\theta_6 & -S\theta_5S\theta_6 & C\theta_5 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{18}$$

is the transformation between the wrist system and the end-effector.

Hence,

$$\mathbf{T}_6^0 = \mathbf{T}_3^0\,\mathbf{T}_6^3 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

where,

$\mathbf{n} \equiv$ unit normal vector,

$\mathbf{s} \equiv$ unit slide vector,

$\mathbf{a} \equiv$ unit approach vector,

$\mathbf{p} \equiv$ position vector,

and their elements are defined as :

$$n_x = C\theta_1C\theta_{23}C\theta_4C\theta_5C\theta_6-C\theta_1C\theta_{23}S\theta_4S\theta_6-C\theta_6C\theta_5S\theta_1S\theta_4-S\theta_6S\theta_1C\theta_4-C\theta_1S\theta_{23}S\theta_5C\theta_6 \tag{20}$$

$$n_y = S\theta_1C\theta_{23}C\theta_4C\theta_5C\theta_6-S\theta_1C\theta_{23}S\theta_4S\theta_6+C\theta_6C\theta_5C\theta_1S\theta_4+S\theta_6C\theta_1C\theta_4-S\theta_1S\theta_{23}S\theta_5C\theta_6 \tag{21}$$

$$n_z = S\theta_{23}C\theta_4C\theta_5C\theta_6-S\theta_{23}S\theta_4S\theta_6+C\theta_{23}S\theta_5C\theta_6 \tag{22}$$

$$s_x = -C\theta_1C\theta_{23}C\theta_4C\theta_5S\theta_6-C\theta_1C\theta_{23}S\theta_4C\theta_6+S\theta_6C\theta_5S\theta_1S\theta_4-C\theta_6S\theta_1C\theta_4+C\theta_1S\theta_{23}S\theta_5S\theta_6 \tag{23}$$

$$s_y = -S\theta_1C\theta_{23}C\theta_4C\theta_5S\theta_6-S\theta_1C\theta_{23}S\theta_4C\theta_6-S\theta_6C\theta_5C\theta_1S\theta_4+C\theta_6C\theta_1C\theta_4+S\theta_1S\theta_{23}S\theta_5S\theta_6 \tag{24}$$

$$s_z = -S\theta_{23}C\theta_4C\theta_5S\theta_6-S\theta_{23}S\theta_4C\theta_6-C\theta_{23}S\theta_5S\theta_6 \tag{25}$$

$$a_x = S\theta_5S\theta_1S\theta_4-C\theta_1C\theta_{23}C\theta_4S\theta_5-C\theta_1S\theta_{23}C\theta_5 \tag{26}$$

$$a_y = -S\theta_1C\theta_{23}C\theta_4S\theta_5-S\theta_5C\theta_1S\theta_4-S\theta_1S\theta_{23}C\theta_5 \tag{27}$$

$$a_z = C\theta_{23}C\theta_5-S\theta_{23}C\theta_4S\theta_5 \tag{28}$$

$$p_x = C\theta_1C\theta_{23}a_3-C\theta_1S\theta_{23}d_4+C\theta_1a_2C\theta_2+S\theta_1d_3 \tag{29}$$

$$p_y = S\theta_1C\theta_{23}a_3-S\theta_1S\theta_{23}d_4+S\theta_1a_2C\theta_2-C\theta_1d_3 \tag{30}$$

$$p_z = C\theta_{23}d_4+S\theta_{23}a_3+a_2S\theta_2 \tag{31}$$

which would relate the end-effector coordinate system to the base coordinate system. As mentioned in (eqn.8), the left-upper 3×3 matrix of $\mathbf{T}_6^0$ is defined as the *rotational matrix*, $\mathbf{R}_{\phi\gamma\psi}$, and is given in terms of the Euler angles as

$$\mathbf{R}_{\phi\gamma\psi} = \begin{bmatrix} \mathbf{n} & \mathbf{s} & \mathbf{a} \end{bmatrix} = \mathbf{R}_{z,\phi}\,\mathbf{R}_{u,\gamma}\,\mathbf{R}_{w,\psi}$$

$$= \begin{bmatrix} C\phi & -S\phi & 0 \\ S\phi & C\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\gamma & -S\gamma \\ 0 & S\gamma & C\gamma \end{bmatrix} \begin{bmatrix} C\psi & -S\psi & 0 \\ S\psi & C\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{32}$$

Hence,

$$\begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} = \begin{bmatrix} C\phi C\psi - S\phi C\gamma S\psi & -C\phi S\psi - S\psi C\gamma C\psi & S\phi S\gamma \\ S\phi C\psi + C\phi C\gamma S\psi & -S\phi S\psi + C\phi C\gamma C\psi & -C\phi S\gamma \\ S\gamma S\psi & S\gamma C\psi & C\gamma \end{bmatrix} \tag{33}$$

which represents a rotation by an angle ϕ about the Z-axis, followed by a rotation γ angle about the rotated X-axis, then a rotation ψ angle about the rotated Z-axis. The resultant change in the coordinate frame is best shown by figure (9).

In formulating (eqn.19), intermediate product results could be defined as a set of matrices $\mathbf{U}_i$, where

$$\mathbf{U}_6 = \mathbf{T}_6^5 \tag{34}$$

$$\mathbf{U}_5 = \mathbf{T}_5^4 \,.\, \mathbf{U}_6 \tag{35}$$

$$\mathbf{U}_4 = \mathbf{T}_4^3 \,.\, \mathbf{U}_5 \tag{36}$$

$$\mathbf{U}_3 = \mathbf{T}_3^2 \,.\, \mathbf{U}_4 \tag{37}$$

$$\mathbf{U}_2 = \mathbf{T}_2^1 \,.\, \mathbf{U}_3 \tag{38}$$

and,

$$\mathbf{U}_1 = \mathbf{T}_1^0 \,.\, \mathbf{U}_2 \tag{39}$$

which would be useful in formulating the inverse kinematics equations, as will be shown in the next section.

6.3. The Inverse Kinematics Transformation for a PUMA-like Arm

The problem of *Inverse Kinematics (IK)* is concerned with computing the joint variables for a specific motion, given the position and orientation of the robot end-effector (i.e. the transformation matrix $\mathbf{T}_6^0$). The relevant formulation is to be derived breifly in the following for a PUMA-like manipulator with 6 revolute joints. Thus, the following function is required

$$\boldsymbol{\theta} = IK \left[\begin{array}{c} \mathbf{P} \\ --- \\ \mathbf{E} \end{array} \right] \tag{40}$$

or,

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix} = IK \left[\begin{bmatrix} p_x \\ p_y \\ p_z \\ --- \\ \phi \\ \gamma \\ \psi \end{bmatrix} \right] \tag{41}$$

where IK denotes the inverse kinematics procedure, and

$\boldsymbol{\theta} = (\theta_1, \theta_2, ..., \theta_6)^T \equiv$ values for each joint of the arm,

$\mathbf{P} = (p_x, p_y, p_z)^T \equiv$ robot tip position coordinates in 3-D space, and

$\mathbf{E} = (\phi, \gamma, \psi)^T \equiv$ Euler angles, representing the robot tip orientation in 3-D space.

The solution of the inverse kinematics problem can be divided into two sub-problems. First, the first three joints are found which would position the wrist at a specific point in space, then the final three joints are calculated, thus providing the correct orientation of the end-effector. This is due to the possibility of decoupling the wrist and end-effector mechanisms in the 6 DOF revolute PUMA-like arm considered.

In solving this problem, a computationally efficient algorithm initially formulated by [Paul and Zhang 1986] has been used. Although other kinematics formulations can be found in the literature [Paul,Shimano and Mayer 1981, Fu,Gonzalez and Lee 1987, Craig 1986], the above mentioned procedure is of more, if not equal, computational efficiency, which is the main requirement in the ULMD system. In this approach the arm joint values are computed by solving the following set of matrix equations

$$\mathbf{T}_6^0 = \mathbf{U}_1 \tag{42}$$

$$(\mathbf{T}_1^0)^{-1} \cdot \mathbf{T}_6^0 = \mathbf{U}_2 \tag{43}$$

$$(\mathbf{T}_2^1)^{-1} \cdot (\mathbf{T}_1^0)^{-1} \cdot \mathbf{T}_6^0 = \mathbf{U}_3 \tag{44}$$

$$(\mathbf{T}_3^2)^{-1} \cdot (\mathbf{T}_2^1)^{-1} \cdot (\mathbf{T}_1^0)^{-1} \cdot \mathbf{T}_6^0 = \mathbf{U}_4 \tag{45}$$

$$(T_4^3)^{-1} \cdot (T_3^2)^{-1} \cdot (T_2^1)^{-1} \cdot (T_1^0)^{-1} \cdot T_6^0 = U_5 \quad (46)$$

$$(T_5^4)^{-1} \cdot (T_4^3)^{-1} \cdot (T_3^2)^{-1} \cdot (T_2^1)^{-1} \cdot (T_1^0)^{-1} \cdot T_6^0 = U_6 \quad (47)$$

each in turn, where the operator $(.)^{-1}$ denotes matrix inversion. The required formulae for computing all joint values are included in the following two sections.

6.3.1. Solution for the First 3 Joints

- Considering the first joint of the arm (waist), then

$$\theta_1 = ATAN2\ (p_y,p_x) + \text{asin} \left[\frac{d3}{k_1} \right] \quad (48)$$

where

$$k_1 = \sqrt{p_x^2 + p_y^2} \quad (49)$$

In the above, asin denotes the arc sine function, and $ATAN2(x,y))$ is a four-quadrants version of $ATAN(\frac{x}{y})$, which is employed for the correct evaluation of the arc tangent function [Fu,Gonzalez and Lee 1987]. In addition, another value of θ_1 can be defined as

$$\theta'_1 = ATAN2\ (p_y,p_x) + \pi - \text{asin} \left[\frac{d3}{k_1} \right] \quad (50)$$

where the value of θ_1 implies having the arm's shoulder in a *right* position, while θ'_1 occures with a *left* position.

- Also, for the second joint of the arm (shoulder),

$$\theta_2 = ATAN2\ (p_z,k_2) + k_4 \quad (51)$$

where,

$$k_2 = C\theta_1 p_x + S\theta_1 p_y \quad (52)$$

$$k_3 = k_3^2 + p_z^2 \quad (53)$$

$$k_4 = \text{acos} \left[\frac{a_2^2 - d_4^2 - a_3^2 + k_3}{2\ a_2 \sqrt{k_3}} \right] \quad (54)$$

and acos denotes the arc cosine function. This value of θ_2 implies an *above* elbow position for a *right* shoulder, while a *below* position could be defined as

$$\theta'_2 = ATAN2\ (p_z,k_2) - k_4 \quad (55)$$

- The third joint of the arm (elbow) is defined as

$$\theta_3 = ATAN2\ (a_3,d_4) - ATAN2\ (k_5,k_6) \tag{56}$$

where,

$$k_5 = C\theta_2 k_2 + S\theta_2 p_z - a_2 \tag{57}$$

$$k_6 = S\theta_2 p_z - S\theta_2 k_2 \tag{58}$$

6.3.2. Solution of the Last 3 Joints (Wrist)

- The 4th joint of the arm is given by

$$\theta_4 = ATAN2\ (k_7,k_9) \tag{59}$$

where,

$$k_7 = C\theta_1 a_y - S\theta_1 a_x \tag{60}$$

$$k_8 = C\theta_1 a_x + S\theta_1 a_y \tag{61}$$

$$k_9 = C\theta_{23} k_8 + S\theta_{23} a_z \quad , \theta_{23} = \theta_2 + \theta_3 \tag{62}$$

The above solution is correct if $S\theta_5<0$, alternatively if $S\theta_5>0$ the following is computed

$$\theta'_4 = ATAN2\ (-k_7,-k_9) \tag{63}$$

- The 5th joint is computed by

$$\theta_5 = ATAN2\ (k_{10},k_{11}) \tag{64}$$

where,

$$k_{10} = S\theta_5 = -C\theta_4 k_9 - S\theta_4 k_7 \tag{65}$$

$$k_{11} = C\theta_5 = -S\theta_{23} k_8 + C\theta_{23} a_z \tag{66}$$

- Finally, for the last joint,

$$\theta_6 = ATAN2\ (k_{12},k_{13}) \tag{67}$$

where,

$$k_{12} = -k_{11}k_{18} - k_{10}k_{16} \tag{68}$$

$$k_{13} = -\ S\theta_4 k_{17} - C\theta_4 k_{15} \tag{69}$$

and,

$$k_{14} = C\theta_1 s_x + S\theta_1 s_y \quad (70)$$

$$k_{15} = S\theta_1 s_x - C\theta_1 s_y \quad (71)$$

$$k_{16} = -S\theta_{23} k_{14} + C\theta_{23} s_z \quad (72)$$

$$k_{17} = C\theta_{23} k_{14} + S\theta_{23} s_z \quad (73)$$

$$k_{18} = C\theta_4 k_{17} - S\theta_4 k_{15} \quad (74)$$

6.4. Difficulties in Solution

6.4.1. Redundancy

Although the inverse kinematics formulation given in previous sections yields correct results, the results are not unique. Consequently, such an algorithm does not provide a one-to-one mapping of any point in the cartesian space to its equivalent joint space. Therefore, for the PUMA robot, certain points in the cartesian space could be reached by up to *eight* different joint configurations. These *redundancies* in the joint values are best illustrated by the block diagram of figure (10). Hence, a choice has to be made as to the best set of joint values to use. It should be noted that due to the limitations on the joint ranges, some of these alternatives may not be accessable. One solution to such a problem is to have the correct configuration of the arm stored before performing the required motion. Hence, at each control cycle, the joint values nearest to the previously detected set are chosen to be correct. This approach would require initiating the robot motion from a specific location within the work volume. In addition, the correct value of joint 1 of the arm can be checked by computing

$$k_{19} = ATAN2(p_y, p_x) - \frac{\pi}{2} \quad (75)$$

for which $k_{19}<0$ implies a right shoulder, and $k_{19}>0$ a left one. Considering a right shoulder, the correct value of θ_2 can be determined by

$$k_{20} = ATAN2(p_y, k_2) \quad (76)$$

where $k_{20}<0$ yields a down elbow, while $k_{20}>0$ an above elbow. As for the correct value of θ_4, then having $S\theta_5<0$ (defined by eqn.66) leads to (eqn.59), while $S\theta_5>0$ makes (eqn.63) the correct choice.

6.4.2. Degeneracy

When one joint of the manipulator is at, or near, a singularity point, its position is said to degenerate, where the arm loses one or more degrees of freedom. Such degeneracy points are usually encountered at the edges of the work space, but could also be located within it. Thus, the accuracy of tracking the trajectory would be greatly affected. Such a problem is usually tackeled in the *trajectory planning phase,* where the specified time-history of motion would be chosen such that singularity points are avoided.

7. Measuring the End-Effector Orientation

For the control scheme to be efficient, the tracking system should provide both position and orientation of the robot hand. Thus, in addition to measuring the 3-D position as was shown in section 3, the orientation, represented by the vector of Euler angles **E** defined within (eqn.41) should be found. Then, the transformation of (eqn.31) can be solved completely, providing all 6 joint variables.

It has been argued that the last three joint values of the robot can be measured directly, since minimal error can be expected due to the zero length of links 5 and 6. Hence, no deformation can be expected, resulting in highly accurate measurements. Nevertheless, a method for calculating the Euler angles using ultrasonic measurements will be introduced. The choice is then left to the user's needs.

The formulation is based on using the rotational matrix defined in (eqn.33) to relate two coordinate systems with coincident origins. Refering to figure (11), the point **P** could be related to both coordinates systems C (XYZ) and $\hat{C}$ $(\hat{X}\hat{Y}\hat{Z})$ by a rotational matrix $\mathbf{R}_{\hat{C}}^{C}$ as

$$\mathbf{P}_{\hat{C}} = \mathbf{R}_{\hat{C}}^{C}\, \mathbf{P}_{C} \qquad (77)$$

or,

$$\begin{bmatrix} p_{\hat{x}} \\ p_{\hat{y}} \\ p_{\hat{z}} \end{bmatrix} = \begin{bmatrix} C\phi C\psi - S\phi C\gamma S\psi & -C\phi S\psi - S\psi C\gamma C\psi & S\phi S\gamma \\ S\phi C\psi + C\phi C\gamma S\psi & -S\phi S\psi + C\phi C\gamma C\psi & -C\phi S\gamma \\ S\gamma S\psi & S\gamma C\psi & C\gamma \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \qquad (78)$$

where $\mathbf{P}_C$ and $\mathbf{P}_{\hat{C}}$ denotes the point location within frames C and $\hat{C}$, respectively.

The wrist design of a PUMA-like manipulator is better known as a *spherical wrist,* whose joint axes intersect at a common point. Such a design greatly simplifies the kinematic analysis, since it allows the decomposition of both positioning and orientation. Hence, considering the fact that all three coordinate systems of links 4, 5 and 6 coincide at their origins in the robot considered (see figure (8)), then what is needed is two measurements, both with respect to the base coordinate system. The first would provide the end-effector position when all wrist joints are frozen (P_C), while the second is measured when joint variables θ_4, θ_5 and θ_6 are in action ($P_{\hat{C}}$). Therefore, the difference between both locations is purly rotational, and the relation of (eqn.78) can be applied. Solving (eqn.78) gives the Euler angles which define the change in orientation with respect to the base coordinate system.

As far as the ultrasonic measurements are concerned, the second location could be detected by using a second, though identical, set of transducers. However, a different frequency has to be used to distinguish between both locations. Some practical difficulties would arise at this stage, as for the value of this second frequency compared to the 40 KHz used originally. A frequency of 20 KHz may cause certain interference, while going for the MHz range would introduce the problem of relatively small spread angle.

In a practical implementation, the first array of transmitters is placed on the arm's wrist, providing the position of the origin of the coincident wrist assembley coordinate frames with respect to the base frame by a vector $\mathbf{Q} = (p_x,p_y,p_z)^T$. Hence, the values of the first 3 joints of the arm can be determined using $\mathbf{Q}$ and (eqns.48-58).

In the remainder of this section, a method for evaluating the rotational matrix $\mathbf{R}_{\hat{C}}^{C}$ given two measured points $P_1 = (x_1,y_1,z_1) = P_C$ and $P_2 = (x_2,y_2,z_2) = P_{\hat{C}}$ will be introduced. The rotational matrix presented in (eqn.33) employs a combination of rotations about the principle axis, as was indicated by figure (9). However, these rotations could be expressed as one rotation about an arbitrary axis, $\mathbf{r}$, representing a unit vector, $r=(r_x,r_y,r_z)^T$, as shown in figure (12). Thus, the corresponding rotational matrix could be expressed [Fu,Gonzalez and Lee 1987] as

$$\mathbf{R}_{\mathbf{r},\gamma} = \begin{bmatrix} r_x^2V\gamma+C\gamma & r_xr_yV\gamma-r_zS\gamma & r_xr_zV\gamma+r_yS\gamma \\ r_xr_yV\gamma+r_zS\gamma & r_y^2V\gamma+C\gamma & r_yr_zV\gamma-r_xS\gamma \\ r_xr_zV\gamma-r_yS\gamma & r_yr_zV\gamma+r_xS\gamma & r_z^2V\gamma+C\gamma \end{bmatrix} \tag{79}$$

where $C\gamma = Cos(\gamma)$ and $V\gamma = 1\text{-}Cos(\gamma)$. This is known as the *equivalent-angle-axis representation* [Craig 1986], where the coordinate frame C is rotated about the origin by an angle γ, producing the new system $\hat{C}$. Hence, determining the coordinates of the unit vector $\mathbf{r}$ and the value of the angle γ seperating vectors $\mathbf{P}_1$ and $\mathbf{P}_2$ would completely define the required rotational matrix.

The two required points P_1 and P_2 are determined as follows. The point P_1 is measured directly by placing a second array of transmitters on the arm's end-effector, thus providing its position when all 6 joints are in action. The second point, P_2, is required when joints 4, 5 and 6 are frozen (i.e. $\theta_4=\theta_5=\theta_6=0$). However, since the position of the origin of the sixth coordinate frame has been already measured by the vector **Q**, then the position Q can be transformed to account for the additional length of the tool while maintaining the condition of a frozen wrist. Such a procedure would provide the values of P_2. This situation is illustrated in figure (13), which shows the three non-colinear points Q, P_1 and P_2.

The coordinates of P_2 with respect to the base coordinate system is defined as

$$\mathbf{P}_2 = \mathbf{T}_6^0 \, \mathbf{T}_{tool}^6 \, \mathbf{P}_2^6 \tag{80}$$

where the elements of $\mathbf{T}_6^0$ are computed from (eqns.20-31) considering $\theta_4=\theta_5=\theta_6=0$ along with the values of θ_1, θ_2 and θ_3 computed earlier, while the translational transformation

$$\mathbf{T}_{tool}^6 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_{tool} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{81}$$

accounts for the tool (or end-effector) attached to the arm. expressing P_2 in the 6th frame as $P_2^6 = (0,0,0,1)^T$ (i.e. no rotation), then (eqn.80) yields

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_{tool} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{82}$$

then

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} p_x + a_x l_{tool} \\ p_y + a_y l_{tool} \\ p_z + a_z l_{tool} \end{bmatrix} \tag{83}$$

Hence, the additional deviation in the coordinates of point $Q = (p_x, p_y, p_z)$ measured earlier is due to the precence of the tool when the wrist assembley is frozen.

To construct the equivilant-angle-axis representation, a translational transformation is applied to the origin of the wrist system defined by **Q**, moving it to the base coordinate system, as shown in figure (14). Hence, the new coordinates of P_1 and P_2 are

$$\overline{\mathbf{P}}_1 = \mathbf{P}_1 - \mathbf{Q} \tag{84}$$

$$\overline{\mathbf{P}}_2 = \mathbf{P}_2 - \mathbf{Q} \tag{85}$$

Refering to figure (14), the normal vector, **R**, to the plane formed by the three non-colinear points $\overline{P}_1$, $\overline{P}_2$, and the origin O can be determined using analytical geometry. Thus, the cross product of vectors $\overline{\mathbf{P}}_1$ and $\overline{\mathbf{P}}_2$ yields :

$$\overline{\mathbf{P}}_1 \times \overline{\mathbf{P}}_2 = \mathbf{R} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \overline{x}_1 & \overline{y}_1 & \overline{z}_1 \\ \overline{x}_2 & \overline{y}_2 & \overline{z}_2 \end{vmatrix} \tag{86}$$

or,

$$\mathbf{R} = (\overline{y}_1\overline{z}_2 - \overline{z}_1\overline{y}_2)\,\mathbf{i} + (\overline{z}_1\overline{x}_2 - \overline{x}_1\overline{z}_2)\,\mathbf{j} + (\overline{x}_1\overline{y}_2 - \overline{y}_1\overline{x}_2)\,\mathbf{k}$$

$$= (R_x\,,\,R_y\,,\,R_z) \tag{87}$$

Then, the unit normal vector **r** is found by normalising the coordinates of **R** as

$$r_q = \frac{R_q}{\sqrt{R_x^2+R_y^2+R_z^2}} \tag{88}$$

where $q \in \{x,y,z\}$. In addition, the angle γ could be defined as

$$Sin\,(\,\gamma\,) = \frac{|\mathbf{R}|}{|\overline{\mathbf{P}}_1|\;|\overline{\mathbf{P}}_2|} = \frac{\sqrt{R_x^2+R_y^2+R_z^2}}{\sqrt{\overline{x}_1^2+\overline{y}_1^2+\overline{z}_1^2}\;\sqrt{\overline{x}_2^2+\overline{y}_2^2+\overline{z}_2^2}} \tag{89}$$

where $|\overline{\mathbf{P}}_1| = |\overline{\mathbf{P}}_2| = l_{tool}$.

Considering the outcome of (eqns.88,89), the rotational matrix of (eqn.79) is completely defined, where

$$C\gamma = \sqrt{1-S\gamma^2} \tag{90}$$

and,

$$V\gamma = 1-C\gamma \tag{91}$$

Nevertheless, the rotational matrix calculated determines the change in the end-effector orientation due to the variations in the last three joint angles. The complete rotational matrix would be

$$\mathbf{R}_{\mathbf{T}_6^0} = \mathbf{R}_{\mathbf{T}_3^0}\,.\,\mathbf{R}_{r,\gamma} \tag{92}$$

where $\mathbf{R}_{\mathbf{T}_6^0}$ and $\mathbf{R}_{\mathbf{T}_3^0}$ represent the upper-left 3×3 rotational matrices of $\mathbf{T}_6^0$ and $\mathbf{T}_3^0$, respectively. In should be noted that the inverse kinematics formulation presented in section 6.3 requires only the second and third columns of $\mathbf{R}_{\mathbf{T}_6^0}$ (i.e. vectors **s** and **a**), therefore only these two columns are computed.

8. The Computational Complexity

In recent reserch, emphasis has been placed on the need for fast control algorithms. Since the introduced ULMD system is proposed for real-time applications, it must be able to provide the necessary information for the robot control loops within the specified control cycle. In this section, the computational requirements of all ULMD procedures will be assessed, and the result will be reported for the PUMA 560 manipulator. The complexity of computations is represented by the algorithms employed in the system, namely the *Position Detection,* the *Orientation Detection* and the *Inverse Kinematics* procedures.

8.1. The Position Detection Procedure (*PD*)

Due to the precence of multiple receivers, as was shown earlier by figure (4), a total of 24 possible combinations of 3-receivers sets can be distinguished. Although some of these combinatioins may not be applicable due to the presence of obstacles and/or disruptions, we will consider here the worst case of having all receivers active. The *worst* case mentioned hereafter is in terms of the computetional requirements. Thus, while having all receivers active would be considered as the *best* case as far as a good measurment in concerned, that might prove to be quite time consuming computationally.

The computation of a single *PD* procedure is defined by (eqn.3-5), and is shown in the first column of table (2), while the total complexity for all 24 sets of receivers is included in the second column of the table. It should be mentioned that certain simplifications in (eqn.3-5) are possible, because of the constant values of distances L_1 and L_2 during the whole application.

Table (2) : Complexity of the PD Procedure			
Floating-Point Operation	Number of operations		
	Single Procedure	All 24 Combinations	
		Detailed	Total
(+ or -)	6	144 + 3 ‡ +23 *	170
(*)	5	120 + 8 †	128
(/)	2	48 + 8 † + 1 *	57
($\sqrt{\ }$)	1	24	24

† Computing *d*'s for all 8 receivers
‡ Computing $\mathbf{P}_R$ (eqn.6)
* Averaging values (eqn.7)

8.2. The Orientation Detection Procedure (*OD*)

Since the calculation of the Eulerian angles requires the knowledge of two points in space, the second point has to be measured according to the same procedure described in section 8.1 (i.e. a second PD procedure). Therefore, in addition to the computations of table (2), which represents the case of having all 8 receivers active, the complexity of the equations of section 7 should be accommodated for as well. This latter is shown in table (3), in addition to the overall complexity required by the *OD* procedure.

Table (3) : Complexity of the OD Procedure

Floating-Point Operation	Number of operations			
	Second PD Procedure	Rotational Matrix		Total
		Detailed	Total	
(+ or -)	170	11 † + 7 ‡ + 16 *	34	204
(*)	128	5 + 10 + 33	48	176
(/)	57	0 + 4 + 0	4	61
($\sqrt{}$)	24	2 + 2 + 0	4	28
Sin	-	2 + 0 + 0	2	2

† Computing $\overline{P_1}$, $\overline{P_2}$ (eqns.83-85)
‡ Computing $\mathbf{R}_{r,\gamma}$
* Computing $\mathbf{R}_{T_6^0}$ (eqn.92)

8.3. The Inverse Kinematics Procedure (*IK*)

The complexity of the *IK* algorithm will be exploited for all joints of the manipulator, considering all combinations of values due to the *redundancies* in the arm configuration. The floating-point operations required to compute the joint values are shown in table (4).

Table (4) : Comlexity of the IK Procedure								
Joint Value	Floating Point Operations †							
	(+ or -)	(*)	(/)	$\sqrt{}$	*Sin*	Tan^{-1}	Sin^{-1}	Cos^{-1}
θ_1	5	2	1	1	-	1	1	-
θ_2	8	10	1	2	1	2	-	1
θ_3	5	5	-	1	1	2	-	-
θ_4	5	10	-	-	-	2	-	-
θ_5	2	4	-	-	-	1	-	-
θ_6	7	14	-	-	-	1	-	-
Total	32	45	2	4	2	9	1	1

† Including check of configuration (eqns.75,76)

Certain simplifications were assumed in computing all three procedures, which reduced a significant part of the computations, as follows:

• Computations involving purely the parameters of the arm (table (1)) are made off-line, since such parameters are constant and unlikely to be changed during motion.

• In computing the trigonometric functions *Sin* and *Cos* of an angle θ, the sine function is computed first since it is less expensive, then the cosine function is evaluated as

$$Cos(\theta) = \sqrt{1-Sin^2(\theta)} \tag{93}$$

which is again less expensive than evaluating $Cos(\theta)$ directly. In addition, compound angle addition formulae were used to evaluate $C\theta_{23}$ and $S\theta_{23}$.

• The Computations of θ_1 and θ_2 are totally independant, and both redundant values of each have to be calculated. However, this would yield four possible values for θ_3, as was indicated in figure (10). Therefore, a decision has to be made on the correct choice of θ_1 and θ_2 before evaluating θ_3, thus saving 75% of the computation time. The same method applies for joint values θ_5 and θ_6.

• Certain computations which are required more than once by the same, or other, equation have been distinguished and hence are executed only once.

A better practical evaluation could be shown if the computational complexity of the ULMD system is expressed in terms of the CPU access time of the processing computer. Thus, adding up the computations of the involved procedures, the total execution time required on a single T800 transputer could be found, as shown in table (5).

Table (5) : Total Complexity of the ULMD System

Joint Value	Floating Point Operations							
Floating Point Operation	(+ or -)	(*)	(/)	$\sqrt{}$	Sin	Tan^{-1}	Sin^{-1}	Cos^{-1}
Number	406	349	120	56	4	9	1	1
CPU Time (μsec)	142.1	192.0	102.0	358.4	67.2	197.1	22.2	21.3
Total Time (msec)	1.10							

The contribution of each of the PD, OD and IK procedures in the above execution time is shown in table (6).

Table (6) : Execution Time of Different Procedures

Procedure	CPU Time (μsecs)
PD	332.0
OD	432.9
IK	337.5

Although the above execution time of (1.10 msecs) is good considering the capabilities of the T800 transputer, it is quite inadequate compared to the recently developed algorithms and structures for fast robot control [Zalzala and Morris 1989a]. The ULMD system must be fast enough to be of use in the control feedback loop. In addition, to ensure the correct measurement of the end-effector position and orientation, several samples are usually required for each control cycle, which would put a further burden on the system. This will be illustrated in the practical evaluation of section 11. Therefore, a faster system has to be defined, where the principles of distributed processing are applied to the described computational procedures, in an attempt to cut down the execution time.

9. The Distributed Formulation

Concurrency in the ULMD system can be exploited at two distinct levels:

□ Procedure Level : where the PD, OD and IK procedures could be executed in a pipelined method, where the total execution time would be the longest of all three.

□ Measurement Level : where measuring the position and orientation of the robot hand could be executed concurrently for each of the 48 combinations of receivers.

In this section, the above levels of concurrency are defined, and their corresponding jobs are constructed, making it possible to implement the ULMD on a multiprocessor system.

9.1. Pipelined Processing of the Computational Procedures

The following pipeline stages are defined, where the overall execution time would be that of the slowest:

- Stage 1 : The coordinate measurements for both the PD and OD procedures for all possible combinations of the receivers.
- Stage 2 : Averaging the measurements supplied by stage 1, thus providing points Q and P_1.
- Stage 3 : Computing the first 3 joint values of the PUMA arm (section 6.3.1).
- Stage 4 : Computing the rotational matrix required for the determination of the other 3 joints (section 7).
- Stage 5 : Computing the last 3 joint values of the PUMA arm (section 6.3.2).

9.2. Parallelism in the Location Measurements

As mentioned earlier, two points in the 3-D space have to be measured, where 24 possible sets of receivers could be used for each. Hence, all 48 calculations could be performed in parallel, once the count associated with each of the 16 receivers is known. However, certain calculations could be shared between sets having a common receiver, as shown in figure (15). Therefore, assigning a single processor to each corner of the cubic volume would accommodate for up to six sets of combinations. This would cut down the expected communications overhead between a large array of 48 processors, especially when limited links are available for each (e.g. the T800 transputer).

According to the levels of concurrency discussed above, three types of processing units have been constructed as follows:

• A Measurement Unit (*MU*) : which has the function of accessing the data provided by a set of four receivers, which should have a similar configuration to that of figure (15). According to the validity of these data values, three (or less) combinations of receivers could be established, yielding the same number of certain coordinates sets. The MU would also perform the averaging procedure on the sets of data produced, hence sending only one set of data to the next unit. This *local* averaging would cut down the communications burden and lead to a more efficient performance.

• An Averaging Unit (AU) : Collecting the resultant measurements from each of the *MU*s, and perform an averaging procedure on the appropriate sets of coordinate values (eqn.8), producing the two detected positions in 3-D space. The output of this unit is two 3×1 vectors of floating-point (fp) numbers.

• An Initial Joint calculations Unit (IJC) : Using one of the two points produced by the *AU* to compute the first three joint angles, hence the word *initial.* The output of this unit is the second point passed by the *AU*, and the set of values

$$v = \{ \theta_1, \theta_2, \theta_3, C\theta_1, S\theta_1, C\theta_{23}, S\theta_{23} \} \quad (94)$$

that is, a total of 10 fp numbers.

• A Rotational Unit (RU) : Utilising the output of the *IJU* to compute the rotational matrix, where only two columns are passed on to the next unit, along with the set v, hence a total of 13 fp numbers.

• A Final Joint calculation Unit (FJU) : Computing the last three joint angles, and providing all six values to the robot controller.

The computational complexities associated with each of the above units are included in table (7).

Unit	Number of Floating Point Operations *							
	(+ or -)	(*)	(/)	$\sqrt{}$	Sin	Tan^{-1}	Sin^{-1}	Cos^{-1}
MU	36 + 12 †	30	12 + 6 †	6	-	-	-	-
AU	42	-	6	-	-	-	-	-
IJU	18	17	2	4	2	5	1	1
RU	34	48	4	4	2	-	-	-
FJU	14	28	-	-	-	2	-	-

Table (7) : Comlexity of the Distributed Units

* Worst case (all 8 receivers active).
† Local averaging.

The above processing modules should be placed on different processors if the pipelined structure is to be effective. Thus, the following architecture is proposed for the multiprocessor system.

10. Constructing the Multiprocessor System

10.1. The T800 Transputer Network

Considering the processing units designated in the previous section, a practical implementation of the ULMD system is carried out on an actual multiprocessor system. The INMOS T800 transputer [INMOS 1988a] has been adapted as the main block for such a system.

Each of the eight *MU* modules (one for each corner of the work volume) is placed on a single transputer, as shown in figure (16). Once each *MU* gets its data from the appropriate receivers, an exchange of data is made between all units simultaniously, providing the necessary combined sets. Calculations are then performed within each processor, where a maximum of six PD procedures would be executed. A local averaging procedure is then performed on each of the MUs, one for each 3 sets of data produced for both point locations, thus providing one set for both positions.

Each of the other computational modules is placed on one processor as follows. The sets of two coordinate values are then collected by another processor, accommodating the *AU* module, where the corresponding data sets are averaged for each point of the two detected. The two measured points are then passed to the 10th transputer containing the *IJU*, performing the inverse kinematics for the first 3 joints. The *RU* module is then activated, calculating the rotational matrix. Its result is passed to the neighbouring *FJU* module which performs the rest of the inverse kinematic procedure, producing the overall joint values. The transputer accommodating the *FJU* is the one connected to the host computer, hence providing an interface to the robot controler.

When the first point is detected, the performance of the processing units would be sequencial in the five layers, where the first layer contains the eight transputers with the *MUs* on, while the other transputers holding the *AU*, *IJU*, *RU* and *FJU* compose the other four layers, respectively. However, in measuring other points, these five layers act as a five-stage pipeline, hence cutting down the computational requirements to that of the largest.

10.2. Considering the Communications

As indicated by figure (16), extensive communications are required between the different MU modules and other processing units. In a practical multiprocessor system, such an overhead must be accounted for if a true assessment of the design is sought.

The communications overhead for all five stages of the pipeline are shown in table (8), along with the required execution time on the links of the transputers. The numbers shown for stage #1 are for the case of all receivers being active. It should be noted, however, that some data has to be passed through an intermediate transputer to reach its destination. This is due to the inherent 4-link limitation in the transputer hardware design. Due to the ability of the T800 to perform computations and access its links simultaniously, the communications burden of stages 2 through 5 of the pipeline are covered by the computations undertaken by their respective processing units.

Table (8) : ULMD Execution Time *				
Pipeline Stage #	Communications		Computation Time	Total
	Float Numbers	Time		
MUs	3	16.32	87.00	103.32
AU	18	97.92	19.80	117.72 †
IJU	3	16.32	229.55	229.55 †
RU	5	27.20	100.90	100.90 †
FJU	7	38.08	64.10	64.10 †

† Communications executed on two links and in parallel with computations.
* All time values in μseconds.

The results shown in table (8) indicate the function of the ULMD system within a period of (0.230 millisecs), where the third stage of the pipeline (the IJU procedure) dominates the total computations. Hence, a device could provide the inputs for the controller at a rate of (4.35 KHz), which is quite fast compared to today's function of the PUMA 560 robot at a rate of (35.7 Hz) [Fu,Gonzalez and Lee 1987]. This speed would allow for several samples to be taken within each control cycle, thus enhansing the measurement accuracy.

10.3. Further Levels of Parallelism

It is noted from the results of table (8) that the bottleneck in the computations is the inverse kinematics procedure executed by the IJU. Hence, a further simplification of this unit would produce an even better performance. However, the inverse kinematics formulation is inherently dependent, where the computations of a certain joint depend on the values of the proceeding ones. Thus, a parallel formulation would involve distributing each of the joint equations on several processin elements. This was found to be extremely impractical in the case of the T800 transputer, since the communications requirements in such a design would overcome the computations performed by each processor, thus rendering the procedure as inefficient, and cost-effective. In addition, the limited number of links available for the transputer adds to the problem. Therefore, employing a multi-transputer system to implement such a distributed structure is concluded to be inefficient, and cost-effective.

11. Experimental Performance of the Position Measurement Device

In this section, an experimental evaluation of the position measurement device is presented. The position of the robot end-effector is measured in real-time, where the appropriate software is written for an IBM-XT compatible computer running under MS-DOS, and using the C programming language. With this sequential implementation, the measurement of a single point in 3-D space and further computing its corresponding joint values requires 0.77 seconds of CPU time. However, several samples are taken, and then averaged to produce a properly acceptable correct result. Using an averaging rate of 10 measurements, the control cycle required a total of 6.96 seconds. Such an execution time is totally unacceptable in real-time robot control, and illustrates clear ly the need for the distributed pipelined implementation presented in the previous sections. This tracking performance is compared against that of the transputer network in table (9).

Table (9) : Tracking Performance †

Tracking Environment	Averaging Rate	
	1 Sample	10 Samples
IBM-XT-Compatible	0.77	6.96
T800 Network ‡	0.00023	0.0023

† All values in seconds.
‡ Real-time simulation.

The PC monitoring output is shown in figure (18).

The PUMA 560 robot was programmed to move its tip in a specific cartesian path, for which the equivalent joint values are computed. The tracking results are shown in figure (17) for both the measured cartesian paths and the calculated joint paths.

In addition, the activity of each of the 8 receivers has been monitored during motion, and are reported in figure (19). It can be shown from the results of figure (19) that receivers 3, 6 and 7 have been disturbed during tracking by the arm blocking the ultrasonic waves. Hence, if the measurements were to be based upon their results, significant errors were to occur. This shows the significance of the multi-transducers configuration proposed, where accurate measurements could be accomplished by making use of other active receivers.

12. Conclusion

An ultrasonic location measurement device has been presented for the fast control of robot manipulators. A special configuration of the robot work volume has been designed to ensure proper and accurate detection of the ultrasonic waves. In addition, the measurement of both the position and orientation of the robot end-effector is considered, making it possible to compute the equivilant joint values via the inverse kinematics algorithm. The electronic circuit designed provides an accuracy of (± 1 mm) when tracking the desired trajectory. By recasting the computations in a distributed pipelined method employing a network of 10 T800 transputers, the location of the robot end-effector can be measured, and the equivilant joint coordinates provided, within a period of (0.343 msecs). The presented system promises to be of extreme benefit for implementing fast control algorithms for intelligent, sensory-based, robot manipulators.

References

[Dorf 1987].
International Encyclopaedia of Robotics: Applications and Automation, (Wiley, New York, 1987).

[INMOS 1988a].
INMOS LTD., "IMS T800 Architecture", Technical Note #6, (1988) .

[Brady,Hollerbach,Johnson,Lozano-Perez and Mason 1982].
BRADY, J. M., HOLLERBACH, J. M., JOHNSON, T. L., LOZANO-PEREZ, T., AND MASON, M. T., *Robot Motion : Planning and Control*, (MIT Press, 1982).

[Craig 1986].
CRAIG, J. J., *Introduction to Robotics : Mechanics and Control*, (Addison-Wesley, 1986).

[Denavit and Hartenberg 1955].
DENAVIT, J. AND HARTENBERG, R. S., "A Kinematic Notation for Lower-pair Mechanisms Based on Matrices ", *J. App. Mech.*, Vol. 77, pp. 215-221, (1955).

[Dickinson and Morris 1988].
DICKINSON, M. AND MORRIS, A. S., "Co-ordinate Determination and Performance Analysis for Robot Manipulators and Guided Vehicles ", *IEE Proceedings, Part-A*, Vol. 135, No. 2, pp. 95-98, (1988).

[Dunkin and Thaler 1986].
DUNKIN, W. M. AND THALER, G. J., "Ultrasonic Position Sensing for a Sentry Robot and/or a Robot Manipulator Arm", In *2nd. IERE Int. Conf. on Real-Time Control of Electromechanical Systems*, (1986) .

[Fu,Gonzalez and Lee 1987].
FU, K. S., GONZALEZ, R. C., AND LEE, C. S. G., *Robotics : Control, Sensing, Vision and Intelligence*, (McGraw Hill,New York, New York, 1987).

[Gilby and Parker 1984].
GILBY, J. H. AND PARKER, G. A., "Robot Arm Position Measurement Using Laser Tracking Techniques", In *Proc. 7th British Robot Association Annual Conf.*, (1984) .

[Lau,Hocken and Haynes 1985].
LAU, K., HOCKEN, R., AND HAYNES, L., "Robot Performance Measurements Using Automatic Laser Tracking Techniques ", *Robotics and Computer-Integrated Manufacturing*, Vol. 2, pp. 227-36, (1985).

[Paul,Shimano and Mayer 1981].
PAUL, R. P., SHIMANO, B., AND MAYER, G. E., "Kinematic Control Equations for Simple Manipulators ", *IEEE Trans. Syst., Man, Cyber.*, Vol. SMC-11, pp. 449-55, (1981).

[Paul and Zhang 1986].
PAUL, R. P. AND ZHANG, H., "Computationally Efficient Kinematics for Manipulators with Spherical Wrists Based on the Homogeneous Transformation Representation ", *Int. J. Robotics Research*, Vol. 5, No. 2, pp. 32-44, (1986).

[Porrill,Pollard,Pridmore,Bowen,Mayhew and Frisby 1988].
PORRILL, J., POLLARD, S. B., PRIDMORE, T. P., BOWEN, J. B., MAYHEW, J. E. W., AND FRISBY, J. P., "TINA: The Sheffield AIVRU Vision System", AIVRU memo #27, (AI Vision Research Unit, Sheffield University, United Kingdom,1988) .

[Shahidi 1989].
SHAHIDI, F., "Ultrasonic Tracking of Robot End-Effector Position", M.Sc. Thesis, (Dept. Control Eng., Univ. of Sheffield,1989) .

[Zalzala and Morris 1989a].
ZALZALA, A. M. S. AND MORRIS, A. S., "A Distributed Pipelined Architecture of the Recursive Lagrangian Equations of Motion for Robot Manipulators with VLSI Implementation", Research Report #353, (Dept. Control Eng.,Univ. of Sheffield, UK,1989) .

[Zalzala and Morris 1989e].
ZALZALA, A. M. S. AND MORRIS, A. S., "An On-Line Minimum-Time Trajectory Generator for Intelligent Robot Manipulators", In *Proc. IMC Sixth Conference on Advanced Manufacturing Technology*, (Dublin City University,Dublin,Ireland,1989) , pp. 1114-31.

[Zalzala and Morris 1989g].
ZALZALA, A. M. S. AND MORRIS, A. S., "A Fast Trajectory Tracker for Intelligent Robot Manipulators", In *Proc. Int. Conf. Applications of Transputers*, (University of Liverpool,Liverpool,UK,(to appear),1989) .

[Zalzala and Morris 1989i].
ZALZALA, A. M. S. AND MORRIS, A. S., "Distributed Robot Control on a Transputer Network ", *IEE Proceedings, Part-E*, (submitted for publication), (1989).

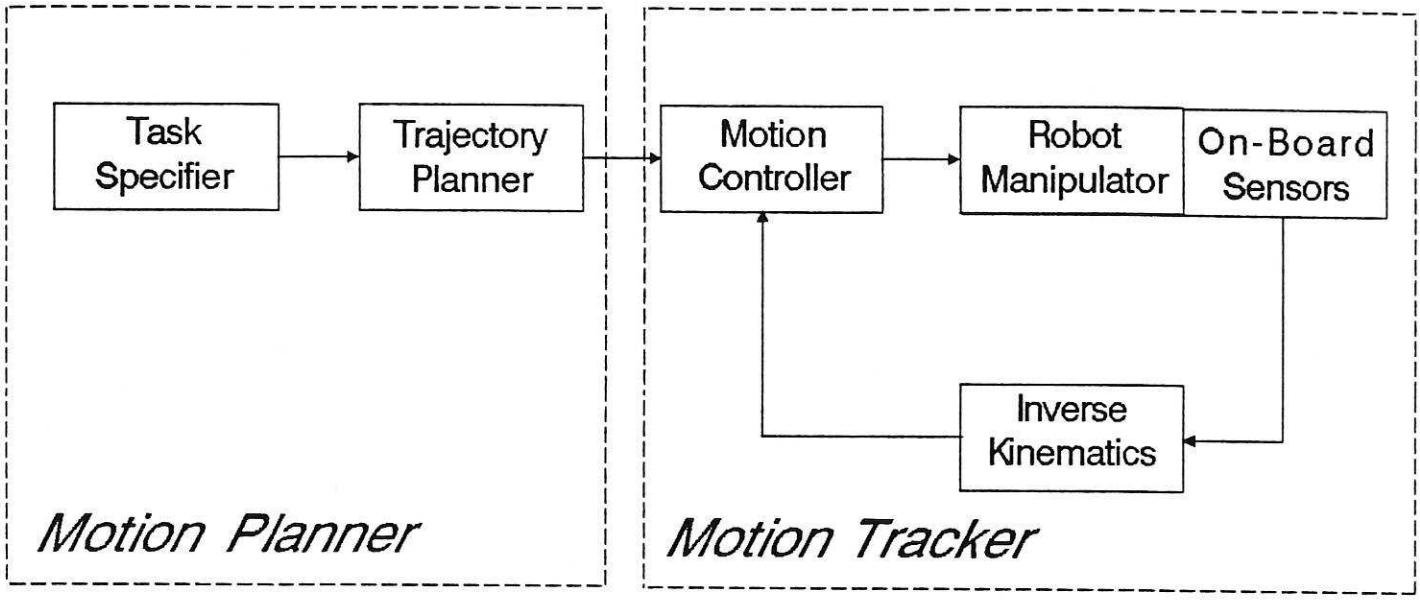


Figure (1) : The Robot Integrated System

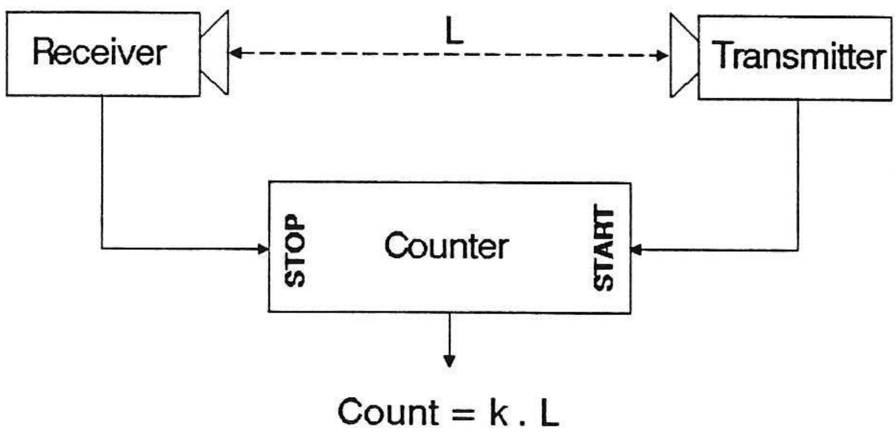


Figure (2)

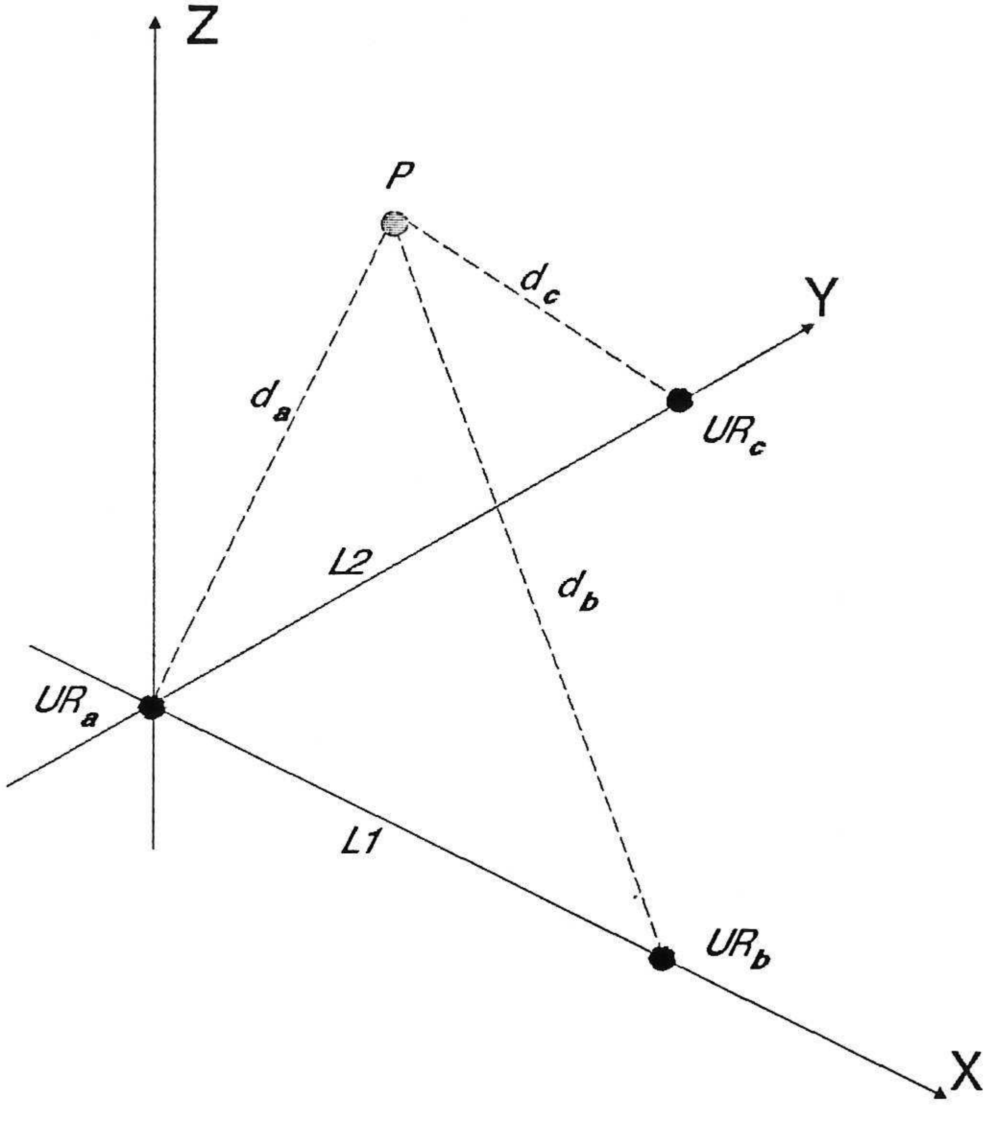


Figure (3) : 3-D Position Measurement

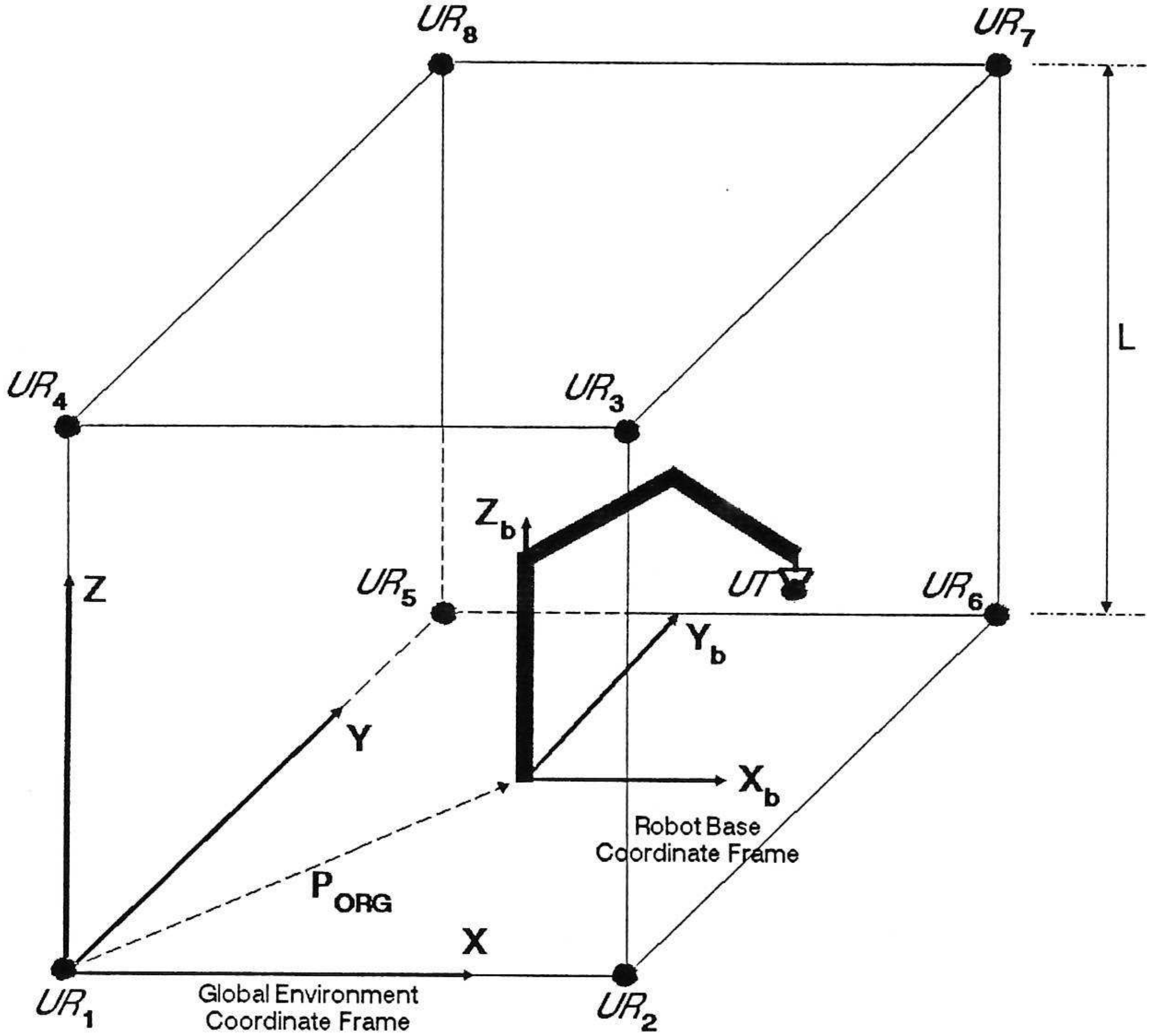
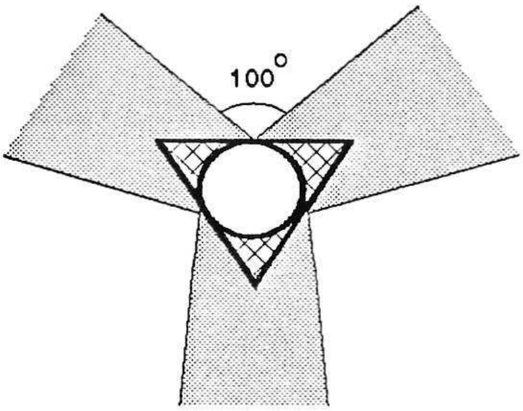
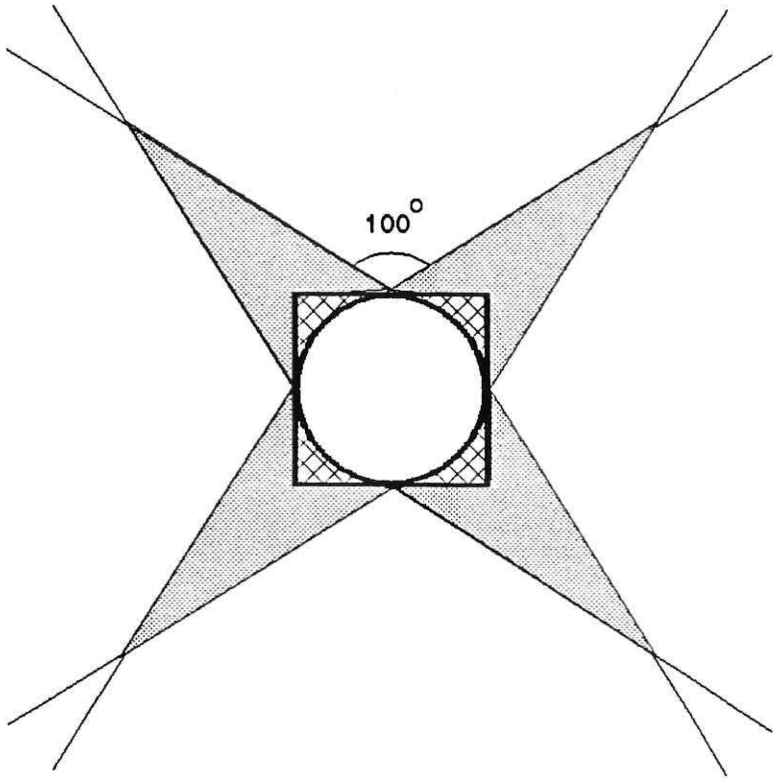


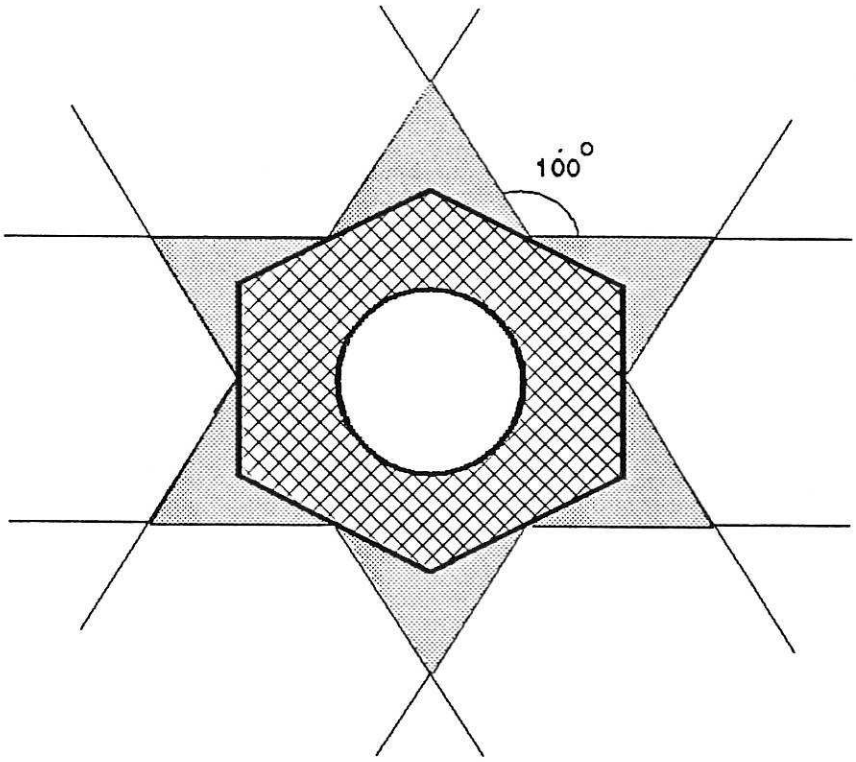
Figure (4) : Multiple Receivers



A. 4-Transmitters Array



B. 5-Transmitters Array



C. 7-Transmitters Array

Figure (5) : Multiple Transmitters

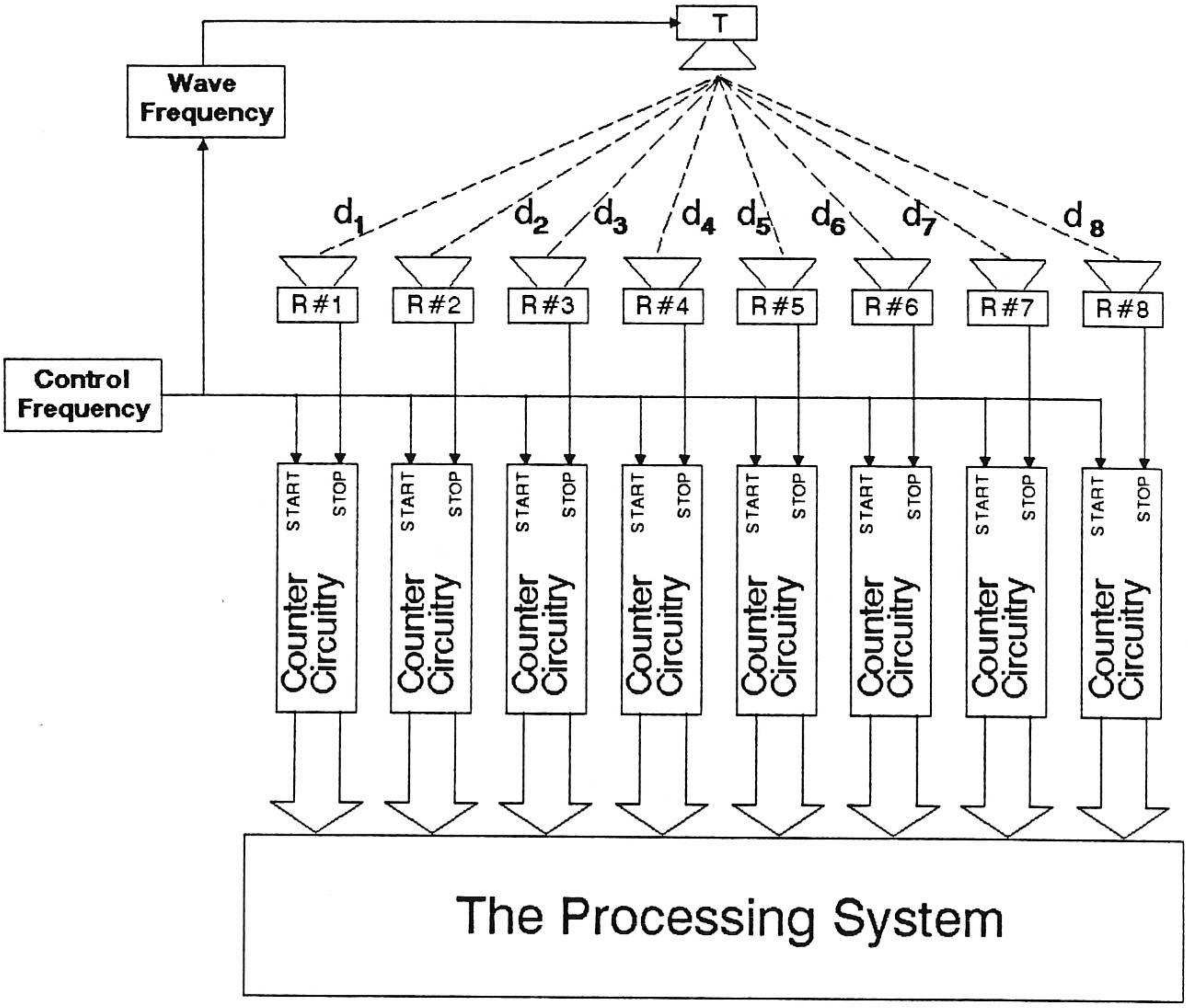


Figure (6) : the Tracking System

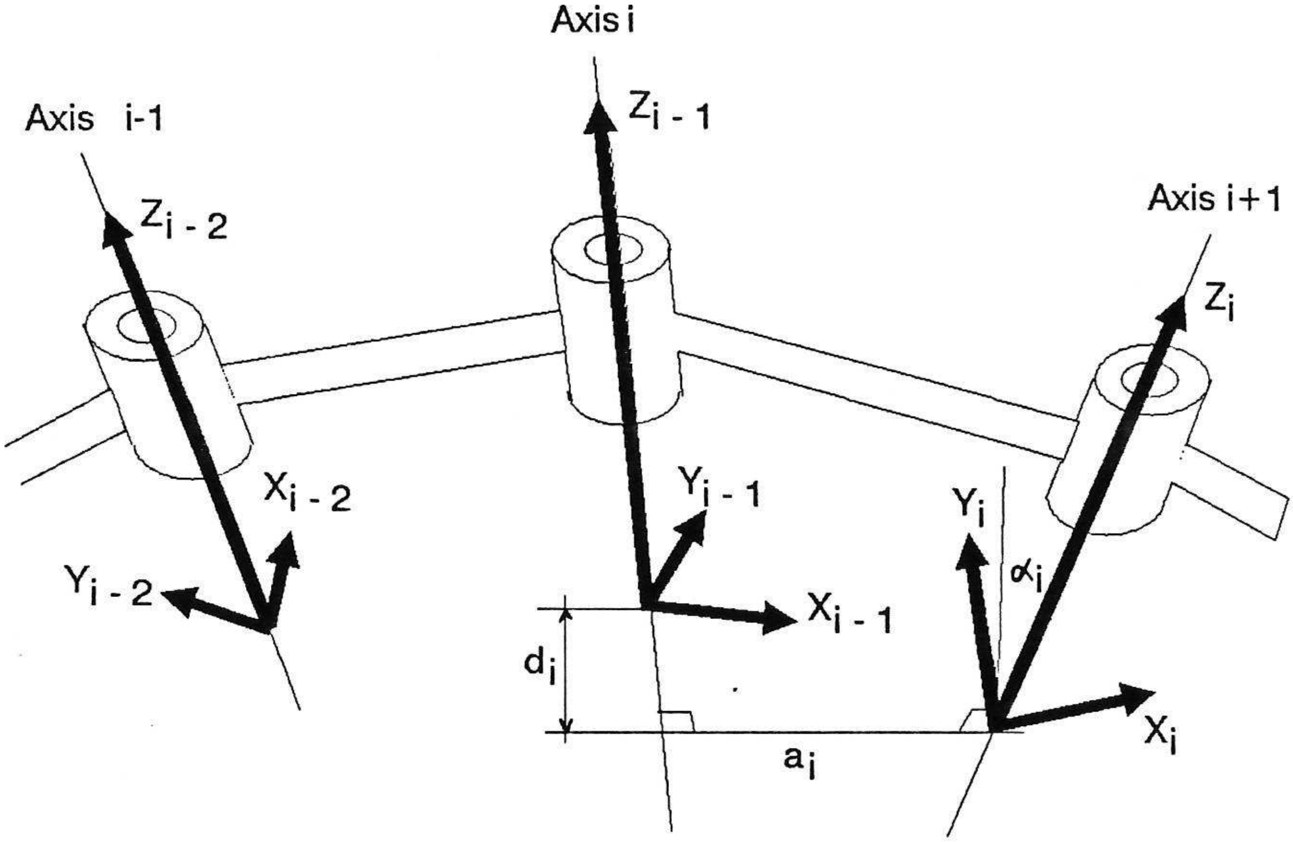


Figure (7) :
Geometric Relations Between Links

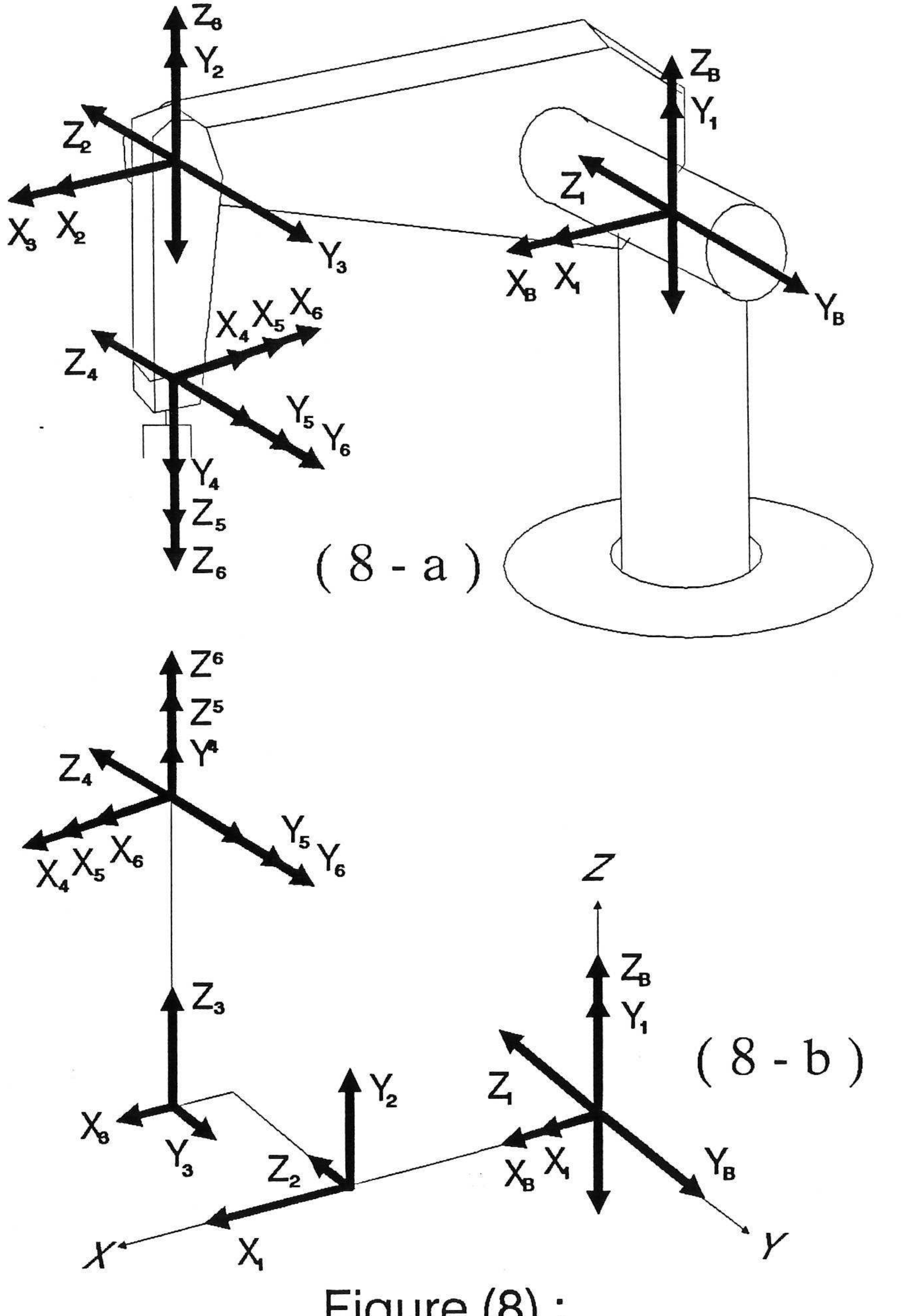


Figure (8) :
Frames Assignment for a PUMA-like Arm

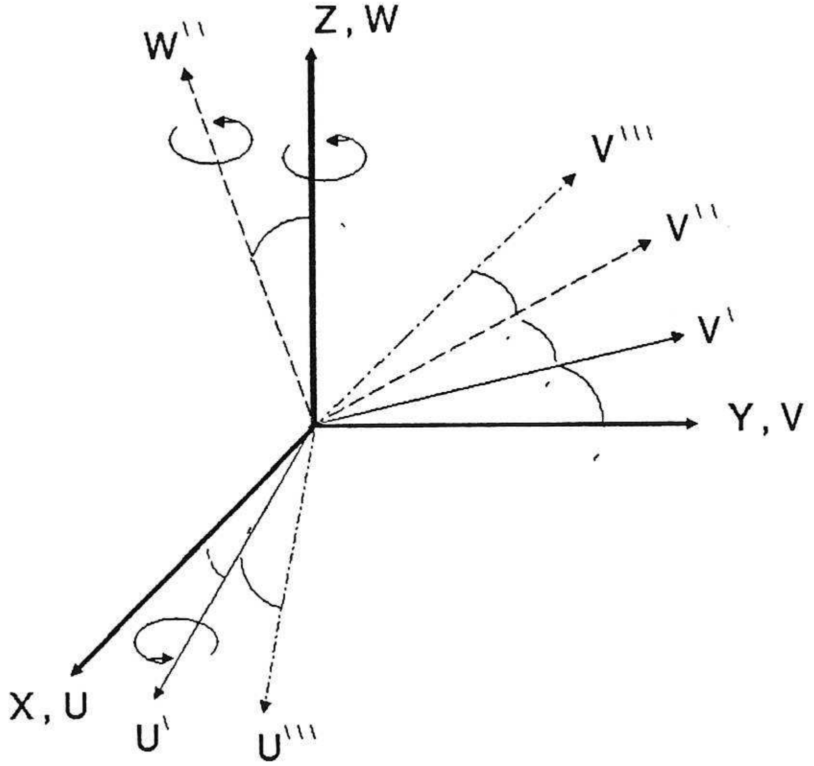


Figure (9) : Eulerian Angles System

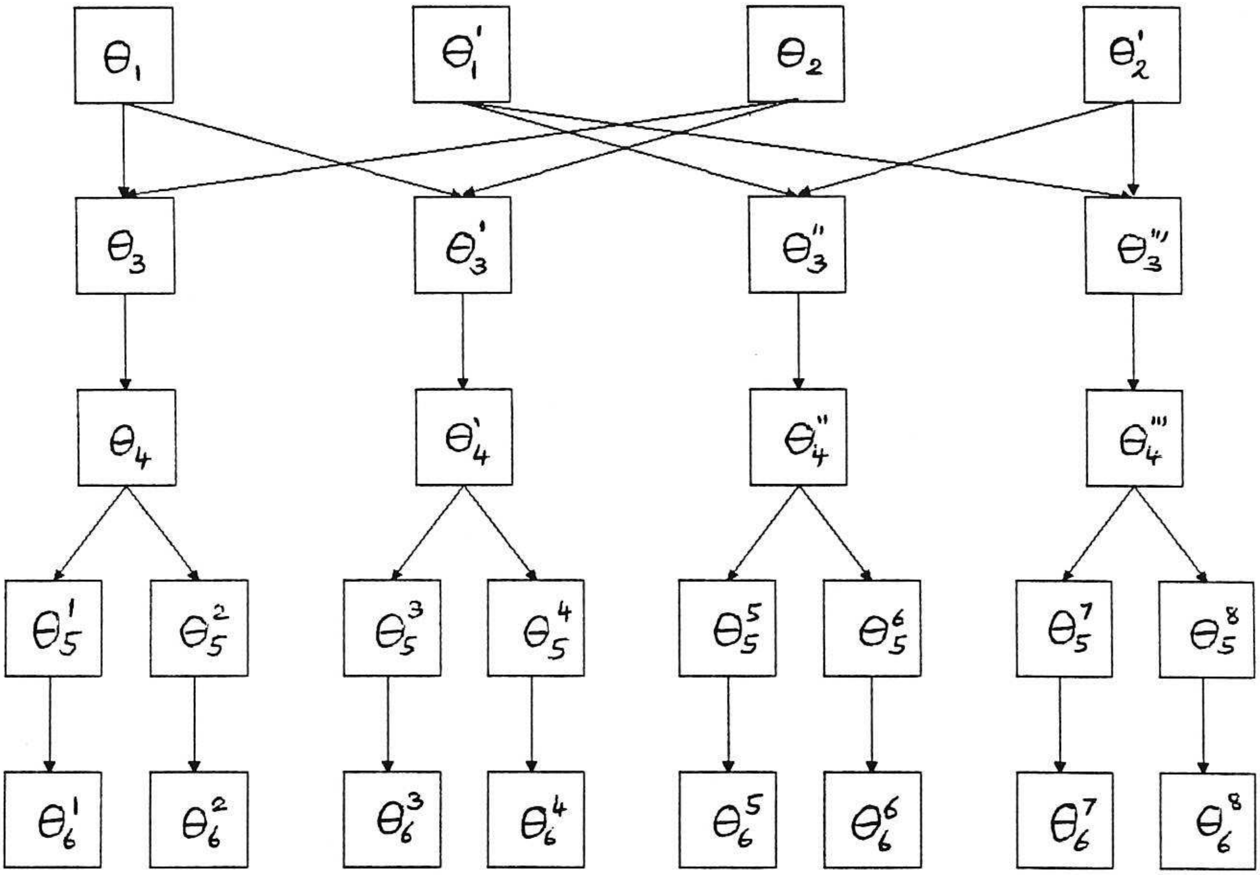


Figure (10) :
Redundancies in the PUMA Configuration

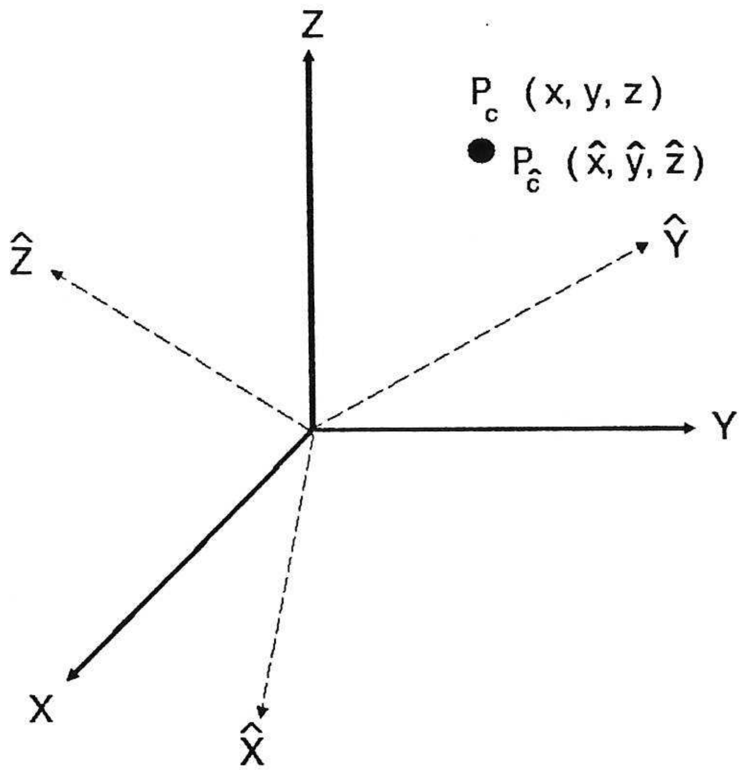


Figure (11) :
One Point Within Two Coinciding Frames

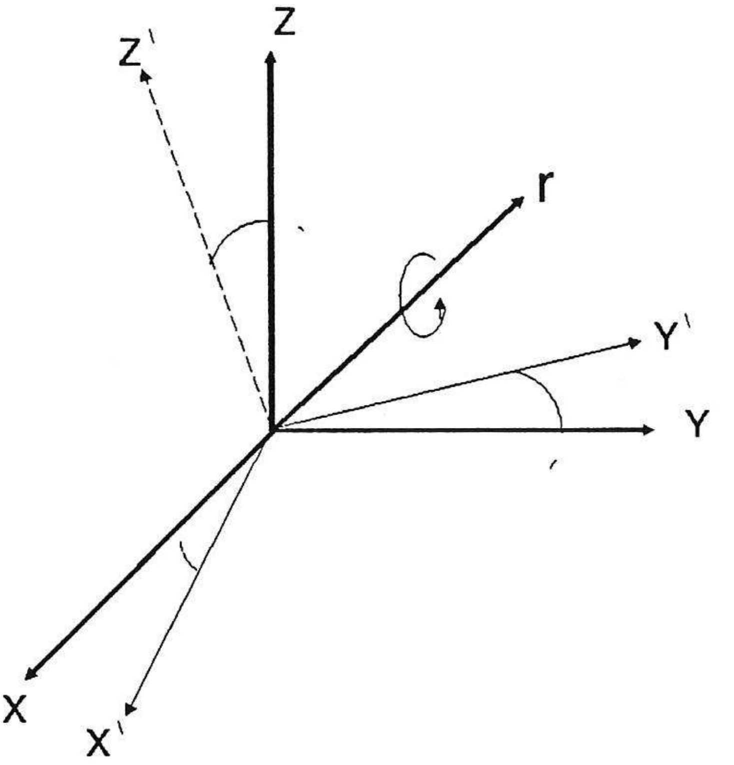


Figure (12):Rotation About an Arbitrary-Axis

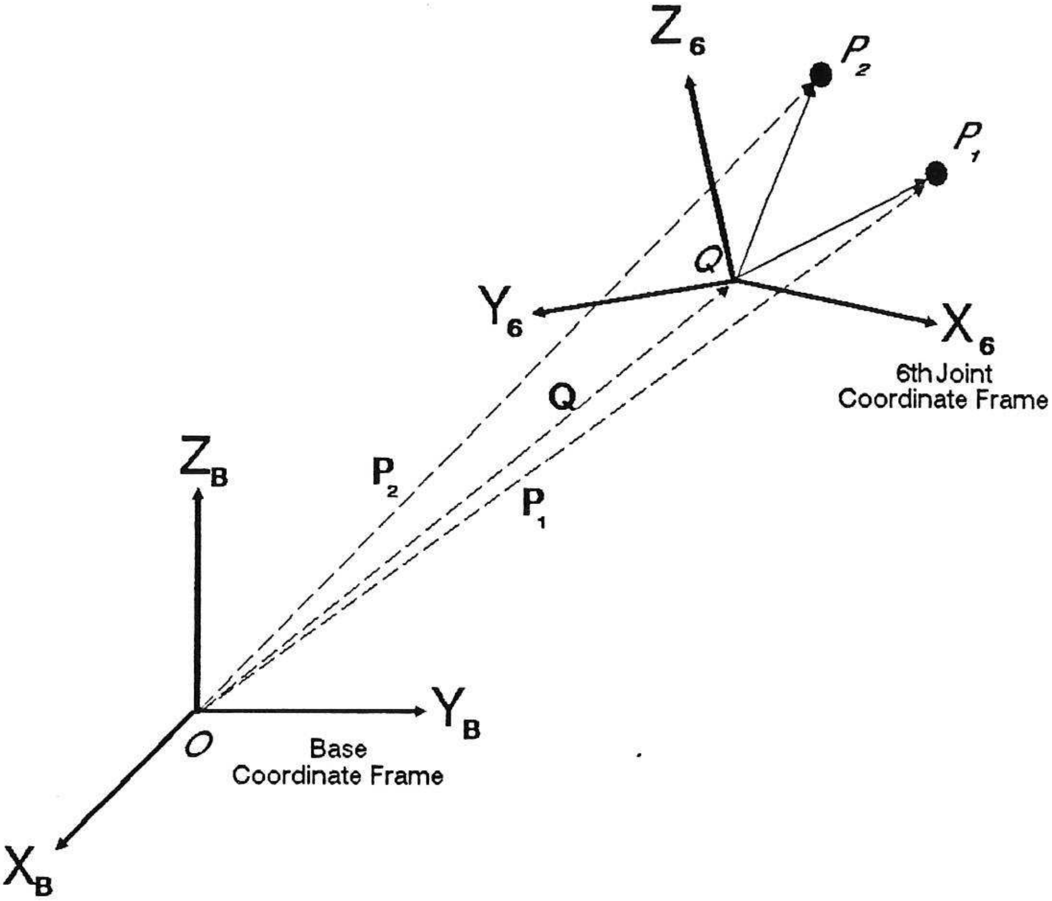


Figure (13) :
The Determination of
Three non-colinear Points in Space

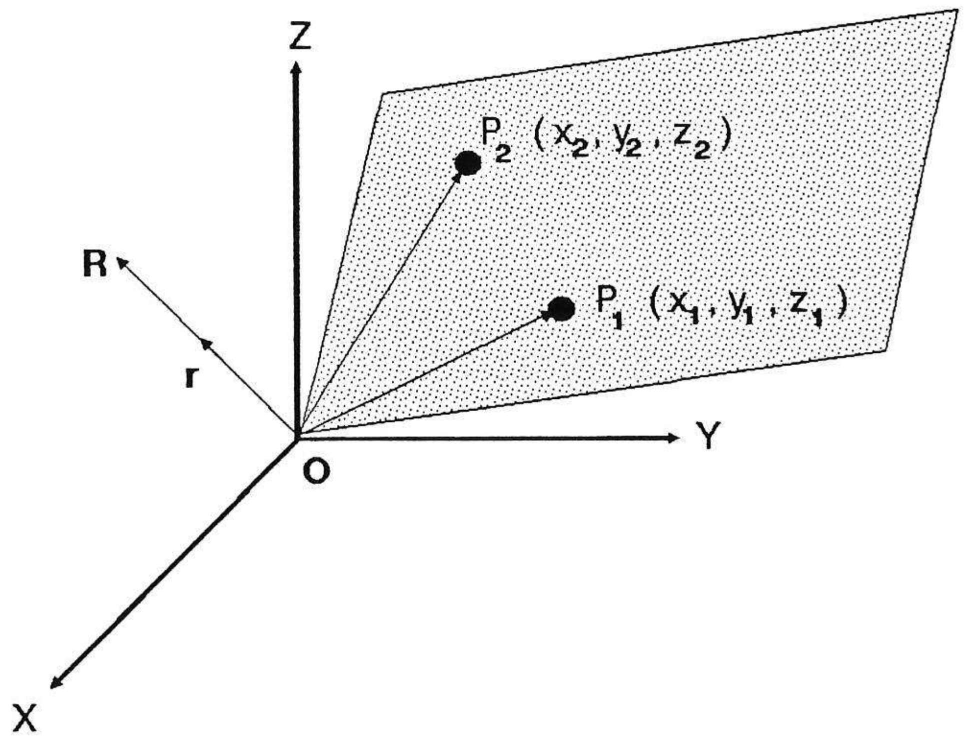


Figure (14) :

The equivilant angle-axis representation

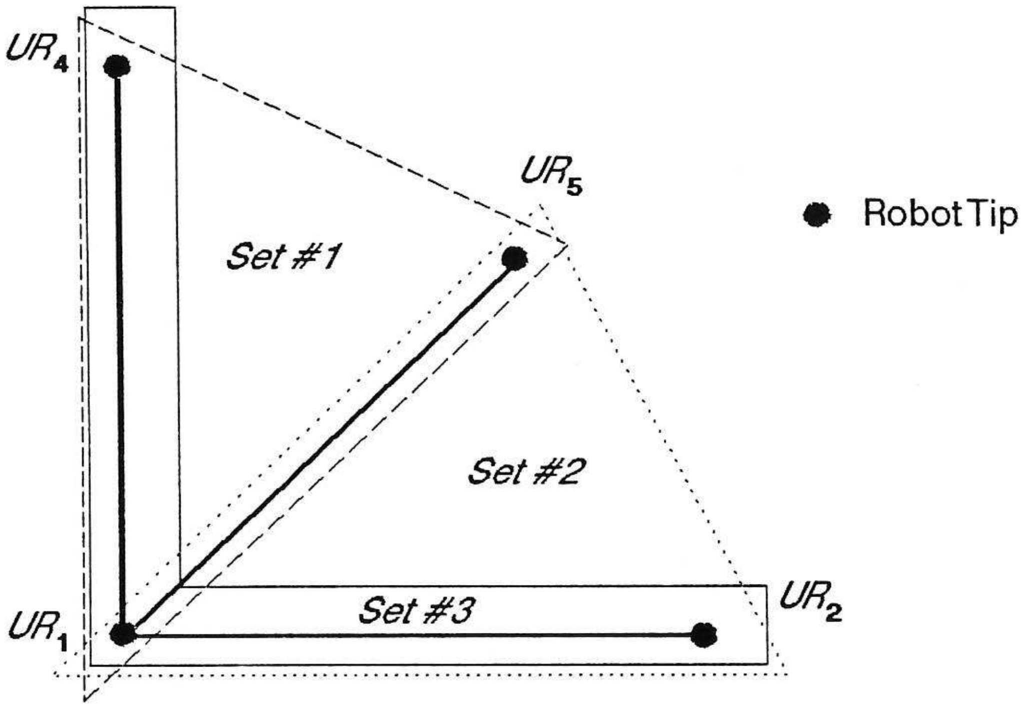


Figure (15) :
One-Corner Combinations

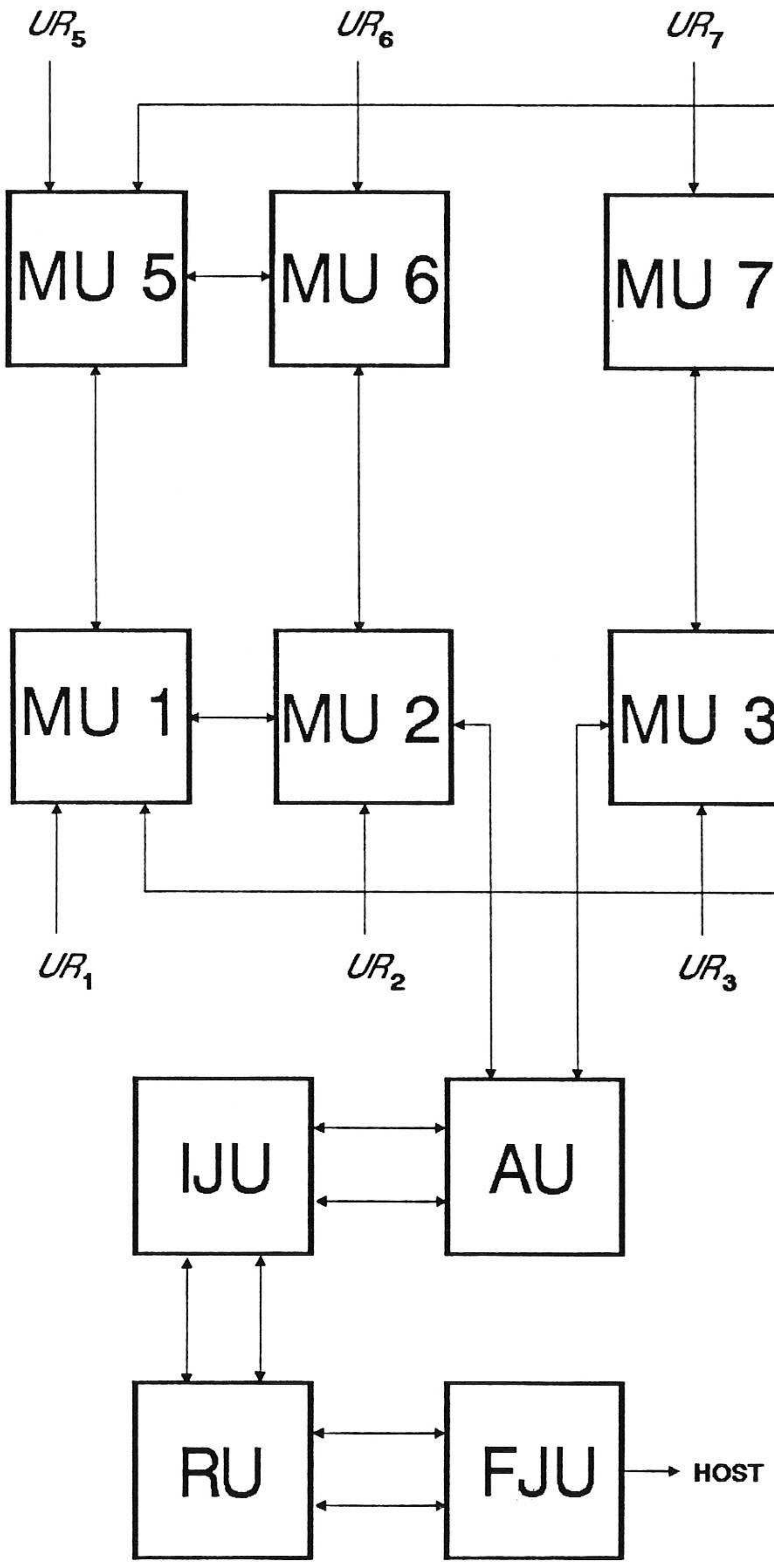
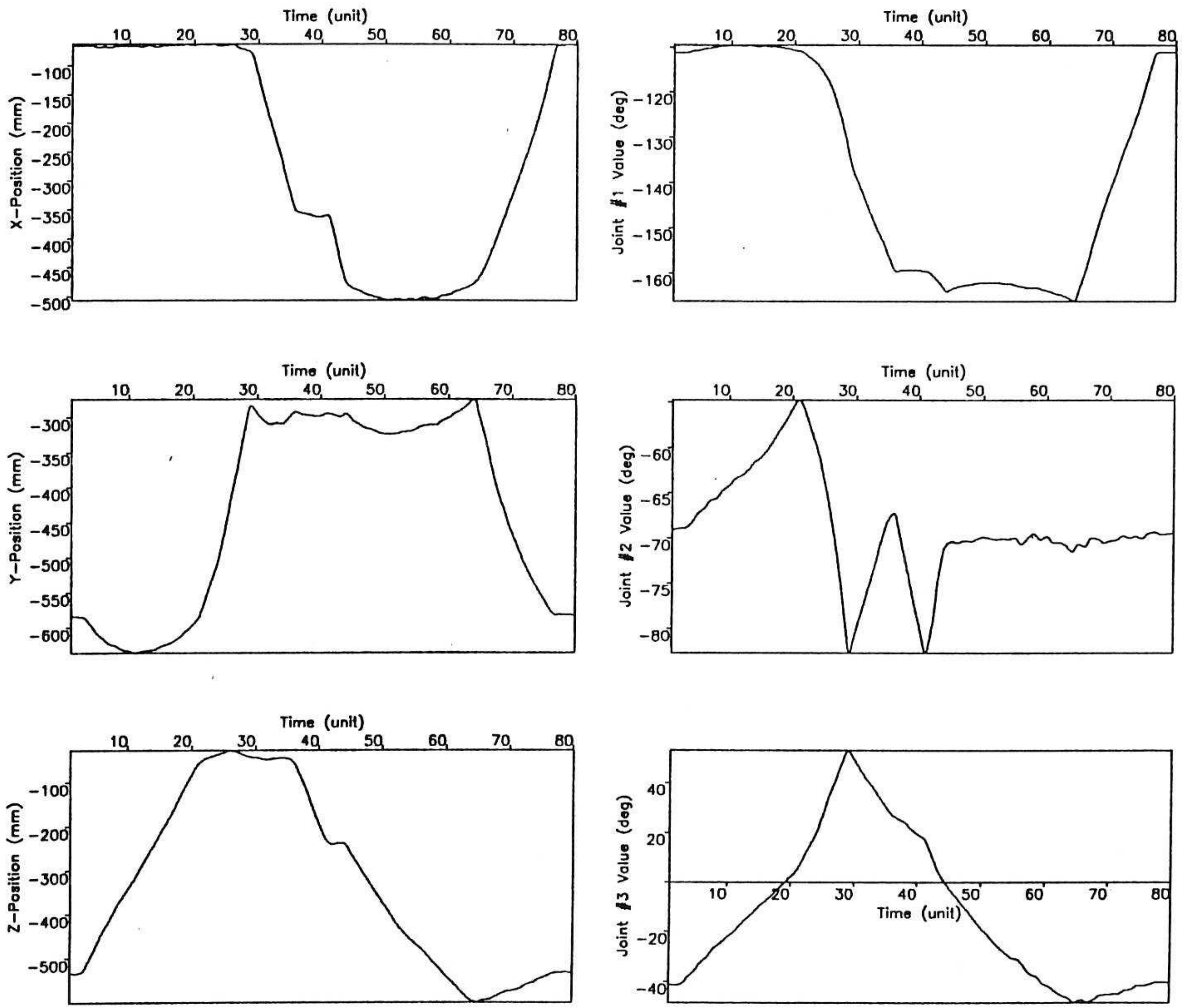


Figure (16): The T800 Transputer Network



Measured Cartesian Paths Calculated Joint Paths

– UPMD Tracking Results –

Figure (17)

*** The Ultrasonic Position Measurement System ***

Receivers Data

Rcvr #1	Rcvr #2	Rcvr #3	Rcvr #4	Rcvr #5	Rcvr #6	Rcvr #7	Rcvr #8
1955	3086	3716	2829	2678	3555	4122	3406

End-Effecter Positions (mm)

	X-Position	Y-Position	Z-Position
C1 >	689	930	844
C2 >	615	950	828
C3 >	606	938	769
C4 >	691	914	769
C5 >	715	863	820
C6 >	631	881	840
C7 >	638	873	749
C8 >	702	846	757
Selected Values	-420	-180	-277

Joint Positions (degs)

Joint #1 =	-176.16
Joint #2 =	-86.60
Joint #3 =	14.80
Joint #4 =	0.00
Joint #5 =	0.00
Joint #6 =	0.00

Tcpu = 7.0300 *** HIT ANY KEY TO TERMINATE *** Count = 1

Figure (18)