CHAPTER 3
INVERSE KINEMATICS OF
NON-REDUNDANT ROBOTS HAVING
MULTIPLE SOLUTIONS

3.1 INTRODUCTION

The inverse kinematics solution of a non-redundant industrial robot may provide multiple robot configurations all of which achieve the required goal position and orientation of the robot. An evolutionary approach based on a real-coded genetic algorithm is used to obtain the solution of the inverse kinematics problem of non-redundant robotic manipulators. All the multiple configurations obtained by this approach can be displayed using a 3D modeller developed in MATLAB for the purpose of visualization. The multiple configurations are compared on the basis of their closeness in the wrist positioning joint space to the current robot configuration. In the absence of obstacles, multiplicity resolution can be achieved by selecting the robot configuration closest to the current robot configuration in the wrist positioning joint space. Simulation experiments are carried out on a SCARA robot and a PUMA robot to illustrate the efficacy of the approach.

3.2 INVERSE KINEMATICS PROBLEM

3.2.1 KINEMATIC MODELLING

A robotic manipulator may be thought of as a set of links connected in a chain by joints, each of which usually

34
exhibits one degree of freedom. The links are numbered from 0 to n, link 0 being the fixed one and link n the end-effector. The $i^{\text{th}}$ kinematic pair, whether revolute or prismatic, couples links 'i-1' and 'i'.

Denavit and Hartenberg (1955) proposed a systematic method of establishing a coordinate system to each link of an articulated chain. An orthonormal Cartesian coordinate system $(X_i, Y_i, Z_i)$ fixed in link 'i' is established for each link at the axis of joint 'i+1' as shown in fig. 3.1. When the joint actuator activates joint 'i', the $i^{\text{th}}$ coordinate system that is fixed in link 'i' moves. The 0th coordinate system defines the base coordinate system of the robot arm. The coordinate systems are assigned as follows:

- $Z_i$ axis: Aligned with the axis of motion of the $(i+1)^{\text{th}}$ joint.
- $X_i$ axis: Established along the common perpendicular to $Z_{i-1}$ and $Z_i$, directed from the former to the latter.
- $Y_i$ axis: Axis chosen so as to make a right-handed coordinate system with $X_i$ and $Z_i$.

The Hartenberg-Denavit parameters of the link are subsequently defined as follows:

- $\theta_i$ is the joint angle between from the $X_{i-1}$ axis to the $X_i$ axis, measured in the positive sense about $Z_{i-1}$.
- $d_i$ is the $Z_{i-1}$ coordinate of the intersection between $Z_{i-1}$ and $X_i$.
- $a_i$ is the offset distance from the intersection of the $Z_{i-1}$ axis with the $X_i$ axis to the origin of the $i^{\text{th}}$ frame along the $X_i$ axis.
- $\alpha_i$ is the angle between the $Z_{i-1}$ and the $Z_i$ axis measured in the positive sense about $X_i$. 
Fig. 3.1: Link coordinate system and link parameters
The link parameters, \( a_i \) and \( \alpha_i \), determine the structure of the link and the joint parameters, \( d_i \) and \( \theta_i \), determine the relative position of neighbouring links. These parameters are shown in fig. 3.1.

The coordinate frame 'i-1' can be made to coincide with coordinate frame 'i' by performing the following successive transformations:

1. Rotate about the \( Z_{i-1} \) axis an angle of \( \theta_i \) to align the \( X_{i-1} \) axis with the \( X_i \) axis.
2. Translate along the \( Z_{i-1} \) axis a distance of \( d_i \) to bring the \( X_{i-1} \) and \( X_i \) axes into coincidence.
3. Translate along the \( X_i \) axis a distance of \( a_i \) to bring the two origins as well as the \( X \) axes into coincidence.
4. Rotate about the \( X_i \) axis an angle of \( \alpha_i \) to bring the two coordinate frames into coincidence.

The composite homogeneous 4X4 transformation matrix for the adjacent coordinate frames 'i' and 'i-1', \([i^{-1}A_i]\), known as the D-H transformation matrix (Niku, 2002) can be evaluated as

\[
[i^{-1}A_i] = \begin{bmatrix}
T_{z,d} & T_{z,\theta} & T_{x,\alpha} \\
T_{x,d} & T_{x,\theta} & T_{x,\alpha} \\
T_{y,d} & T_{y,\theta} & T_{y,\alpha} \\
T_{y,d} & T_{y,\theta} & T_{y,\alpha}
\end{bmatrix}
\]

\[
\begin{bmatrix}
\cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\
\sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\
0 & \sin \alpha_i & \cos \alpha_i & d_i \\
0 & \sin \alpha_i & \cos \alpha_i & d_i
\end{bmatrix}
\]

Using the \([i^{-1}A_i]\) matrices, the homogeneous transformation matrix \([0T_i]\) which specifies the location and orientation of the \( i^{th} \) coordinate frame with respect to the base coordinate frame is given by
\[
\begin{align*}
[\mathcal{O}_T] &= \prod_{j=1}^{i} \left[ j^{-1} A_j \right] \\
&= \phi \left( q_1, q_2, \ldots, q_m \right)
\end{align*}
\] (3.2)

where \( q_i \) is \( \theta_i \) for a revolute joint and \( d_i \) for a prismatic joint and \( m \) is the number of joint variables.

### 3.2.2 PROBLEM FORMULATION

For a robot having \( m \) joint variables, the position vector \( \{\mathcal{O}_h\} \) of the robot hand relative to the base coordinate frame can be extracted from the last column of the matrix \( [\mathcal{O}_h] \). The \( [\mathcal{O}_h] \) matrix relates the hand coordinate frame of the robot to the base coordinate frame and can be obtained through the use of DH matrices explained in section 3.2.1. For an industrial robot having intersecting last three axes, the number of degrees of freedom can be reduced to \( m-3 \) by decoupling the solution of the wrist positioning and the hand orienting joint variables (Pieper,1968). The position vector of the robot wrist \( \{\mathcal{O}_w\} \) can be thus obtained as

\[
\{\mathcal{O}_w\} = \{\mathcal{O}_h\} - \left[\mathcal{O}_{m-2}\right] \{\mathcal{O}_{m-2}\} \] (3.3)

where \( \left[\mathcal{O}_{m-2}\right] \) relates the axes of coordinate frame \( m-2 \) to the base coordinate frame and \( \{\mathcal{O}_w\} \) is a function of the vector of the wrist positioning joint variables \( \{q_1, q_2, \ldots, q_{(m-3)}\} \). The \( \{\mathcal{O}_w\} \) vector can alternately be obtained through the last column of the matrix \( \left[\mathcal{O}_{m-2}\right] \).

For an industrial manipulator working in an obstacle free environment, the performance criterion for
Multiplicity resolution can be taken as the minimization of the total joint displacement

\[ \Delta q = \| \{q\} - \{q_{\text{cur}}\} \| \]  \hspace{1cm} (3.4)

where \( \| . \| \) denotes the Euclidean distance, \( \{q_{\text{cur}}\} = \{q_{1,\text{cur}}, q_{2,\text{cur}}, ..., q_{(m-3),\text{cur}}\}^T \) represents the vector of the wrist positioning joint variables at the current configuration of the robotic manipulator and \( \{q\} = \{q_1, q_2, ..., q_{(m-3)}\}^T \) represents the unknown vector of the wrist positioning joint variables at the desired goal position of the robotic manipulator.

In order to achieve a desired position \( ^{\text{O}}P_{\text{w,des}} \) of the robot wrist, eqn. (3.3) can be used to form a constraint vector equation of the form

\[ \{^\text{O}P_{\text{w}}\} - \{^\text{O}P_{\text{w,des}}\} = \{0\} \]  \hspace{1cm} (3.5)

The limits on the wrist positioning joint variable values can be expressed as

\[ q_k^l \leq q_k \leq q_k^u \hspace{1cm} k = 1, 2, ..., m - 3 \]  \hspace{1cm} (3.6)

where \( q_k^l \) and \( q_k^u \) represent the lower and upper limits of the joint variables.

The inverse kinematics problem solution involves the solution of the non-linear transcendental vector equation given in eqn. (3.5) subject to the constraints given in eqn. (3.6). Therefore, the inverse kinematics problem can be stated as the following optimization problem
Minimize \[ \left\| \mathbf{P}_k - \mathbf{P}_{\text{des}} \right\| \]
subject to \[ q^l_k \leq q_k \leq q^u_k \quad k = 1, 2, \ldots, m - 3 \] (3.7)

The objective function of the optimization problem stated in eqn. (3.7) is multimodal in nature and has zero value at the optimal points.

The multiplicity resolution (Kalra et al., 2003a) involves the solution of the following problem

Minimize \[ \Delta q \]
subject to \[ \left\{ \mathbf{P}_k \right\} - \left\{ \mathbf{P}_{\text{des}} \right\} = \{0\} \]
\[ q^l_k \leq q_k \leq q^u_k \quad k = 1, 2, \ldots, m - 3 \] (3.8)

The solution of the optimization problem given in eqn.(3.8) involves the comparison of the total joint displacement \( \Delta q \) values at the multiple solutions of the multimodal optimization problem defined in eqn.(3.7).

Having obtained the values of the wrist positioning joint variables through the solution of the optimization problems given in eqns.(3.7) and (3.8), the hand orienting joint variables can be calculated from the following matrix equation

\[
\begin{bmatrix}
[\mathbf{n}]^{-3} \mathbf{R}_m \\
[\mathbf{o}]^{-3} \mathbf{R}_m - 3
\end{bmatrix}
= \begin{bmatrix}
\mathbf{n}_x & \mathbf{o}_x & \mathbf{a}_x \\
\mathbf{n}_y & \mathbf{o}_y & \mathbf{a}_y \\
\mathbf{n}_z & \mathbf{o}_z & \mathbf{a}_z
\end{bmatrix}
\] (3.9)

where \([\mathbf{n}]^{-3} \mathbf{R}_m\) and \([\mathbf{o}]^{-3} \mathbf{R}_m - 3\) can be extracted from \([\mathbf{n}]^{-3} \mathbf{R}_m\) and 
\([\mathbf{o}]^{-3} \mathbf{R}_m - 3\) respectively. The vectors \( \mathbf{\hat{n}} = \{n_x, n_y, n_z\}^T \)
\( \mathbf{\hat{o}} = \{o_x, o_y, o_z\}^T \) and \( \mathbf{\hat{a}} = \{a_x, a_y, a_z\}^T \) represent the known normal,
orientation and approach vectors of the hand with respect to the base coordinate frame. Further, \([\mathbf{o}]^{-3} \mathbf{R}_m - 3\) is a
function of the wrist positioning joint variables vector \( \{q_1, q_2, \ldots, q_{m-3}\}^T \) which is known from the solution of the optimization problem given in eqn. (3.8). The right hand side of eqn. (3.9) is thus known.

The limits on the hand orienting joint variables can be expressed as

\[
q^l_k \leq q_k \leq q^u_k \quad k = m - 2, m - 1, m \quad (3.10)
\]

To calculate the hand orienting joint variables, eqn. (3.9) can be solved through the following optimization problem

\[
\begin{align*}
\text{Minimize} & \quad \frac{1}{2} \sum_{i=1}^{3} \sum_{j=1}^{3} (r_{ij} - r'_{ij})^2 \\
\text{subject to} & \quad q^l_k \leq q_k \leq q^u_k \quad k = m - 2, m - 1, m
\end{align*}
\]

where \( r_{ij} \) and \( r'_{ij} \) are the elements of the left hand side and right hand side of the matrix eqn. (3.9).

### 3.3 DETAILS OF EVOLUTIONARY APPROACH

The evolutionary approach based on a real-coded GA, described in chapter 2, is used as the optimization strategy for the solution of the inverse kinematics problem stated in eqns. (3.7), (3.8) and (3.11). The general basic mechanics of this approach has been described in chapter 2. In this section, the problem-specific parameters of the approach have been discussed.

**INDIVIDUAL REPRESENTATION**

For a robotic manipulator, the individual in a population would be represented by the real number vector of the joint variables \( \{q_1, q_2, \ldots, q_{m-3}\}^T \) for the solution of the wrist positioning problem and by the vector
for the solution of the hand orienting problem.

**INITIALIZATION**

An initial population of robot configurations is generated by random sampling from the variable search space. The goal is to select individuals spread over the entire search space, respecting the limits of joint motion specified by eqns. (3.6) and (3.10).

**EVALUATION AND SELECTION**

The fitness of an individual is a measure of how good a solution it provides to the optimization problems stated in eqns. (3.7), (3.8) and (3.11). The initialization procedure, along with appropriate recombination and mutation strategies described in sections 2.3.3 and 2.3.4, ensures that all individuals in the population satisfy the limits of joint motion given in eqns. (3.6) and (3.10). Optimization problems given in eqns. (3.7) and (3.11) thus reduce to unconstrained multimodal optimization problems.

The individuals providing the solutions of the optimization problem given in eqn. (3.7) are referred to as 'feasible solutions' since they satisfy the constraints of the problem stated in eqn. (3.8). All other possible individuals are referred to as 'infeasible solutions'. The fitness function to be minimized for the solution of the wrist positioning joint variables is therefore defined as

\[
\text{Fitness} = \begin{cases} 
\|q^* - q_{\text{des}}\| & \text{for infeasible solutions} \\
\Delta q & \text{for feasible solutions}
\end{cases}
\]

(3.12)
In the current work, the binary tournament selection operator is used. For the solution of the optimization problem given in eqn. (3.7), fitness is evaluated according to its definition for infeasible solutions given in eqn. (3.12). The fitness values associated with the ‘feasible solutions’, obtained through the solution of the optimization problem given in eqn. (3.7), are subsequently evaluated according to the fitness definition given in eqn. (3.12). Comparison of these fitness values provides a means for multiplicity resolution.

For the subsequent solution of the hand orienting joint variables through the optimization problem given in eqn. (3.11), the fitness function may be defined as

\[
\text{Fitness} = \sum_{i=1}^{N} \sum_{j=1}^{N} \left( r_{ij} - r'_{ij} \right)^2
\]  

Two niching strategies are independently used in this work along with the tournament selection operator to maintain solutions around the multiple optima. Oei et al. (1991) suggested a niching approach wherein a niche size parameter \( n \) is defined by the user. For each of the two solutions participating in a tournament, the niche count of the \( i \)th individual, \( n_{ci} \), can be evaluated as

\[
n_{ci} = \sum_{j=1}^{N} \text{Sh}(d_{ij})
\]  

Here \( N \) represents the total number of individuals in the population, \( d_{ij} \) represents the Euclidean distance between the \( i \)th and \( j \)th individual and \( \text{Sh}(d_{ij}) \) represents the sharing function values for the \( i \)th individual. The latter is evaluated as

\[
\text{Sh}(d_{ij}) \text{ evaluated as}
\]
\[ Sh(d_{ij}) = 1 - \left( \frac{d_{ij}}{\sigma_{share}} \right)^{a''}, \quad \text{if } d_{ij} \leq \sigma_{share} \]  
\[ = 0, \quad \text{otherwise} \] (3.15)

The parameter \( a'' \) is taken as 1 and the \( \sigma_{share} \) parameter is given by

\[ \sigma_{share} = \frac{\sqrt{\sum_{k=1}^{n} (q_k^u - q_k^l)^2}}{2\sqrt{r}} \] (3.16)

In eqn. (3.16), \( n \) represents the number of variables in the problem and \( r \) represents the number of niches in the search space. The tournament selection is carried out as follows:-

If the niche count of both solutions is less than \( n' \), the one with the better fitness wins; otherwise, the one with the smaller niche count wins.

Thus, if one solution is overly crowded and the other is not, the second solution is chosen.

Deb and Agrawal (1999) proposed a niching approach where a maximum of 's' individuals are sought as a partner for the selection of an individual 'i'. If any of the 's' individuals lies within the \( \sigma_{share} \) parameter defined in eqn. (3.16), that solution competes with solution 'i' and the one with a better fitness value wins. If none of the 's' individuals is found to lie within \( \sigma_{share} \) of solution 'i', the latter is automatically declared selected.

The niching strategies proposed by Oei et al. (1991) and Deb and Agrawal (1999) are hereafter referred to as niching strategy 1 and niching strategy 2 respectively.
A deviation measure is suggested by Deb and Kumar (1995) to calculate the deviation of the distribution of the population from an ideal distribution for a multimodal problem. This deviation measure is defined as

\[ \psi = \sqrt{\sum_{k=1}^{r+1} \left( \frac{n_k - \bar{n}_k}{\hat{n}_k} \right)^2} \]  

(3.17)

where \( \bar{n}_k \) and \( \hat{n}_k \) are the expected value of number of solutions and the standard deviation of the number of solutions near the \( k^{\text{th}} \) optimal solution in the population respectively. In eqn. (3.17), \( n_k \) represents the actual number of solutions in the vicinity of the \( k^{\text{th}} \) optimal solution in the population. The parameters \( \bar{n}_k \) and \( \hat{n}_k \) are defined as

\[ \bar{n}_k = \frac{N}{r+1} \quad k = 1, 2, ..., r \]

(3.18)

\[ \bar{n}_k = 0 \quad k = r + 1 \]

\[ \hat{n}_k = \sqrt{\bar{n}_k \left( 1 - \frac{\bar{n}_k}{N} \right)} \quad k = 1, 2, ..., r \]

(3.19)

\[ \hat{n}_k = \sqrt{\frac{1}{r+1} \sum_{k=1}^{r+1} \bar{n}_k^2} \quad k = r + 1 \]

In the computer simulations carried out in this work, all solutions around the optimal points for which the position error of the wrist lies within a specified value are counted as the number of solutions near the corresponding optimum.

**RECOMBINATION AND MUTATION**

The recombination operator should create children solutions within the range of the joint variables given in
eqns. (3.6) and (3.10). The spread factor of the SBX operator, described in section 2.3.3, is therefore evaluated using eqns. (2.5), (2.6) and (2.7).

In order to find the optimal solutions within the multiple optimum basins efficiently, a speciation method which restricts mating between individuals in a basin is employed in this work. To implement mating restriction for the SBX recombination operator, the mate for an individual is chosen as follows:

The Euclidean distance between the two individual solutions is computed. If the distance is closer than a parameter $\sigma_{\text{mating}}$ they participate in the crossover operation as parents, otherwise another individual is chosen at random and their distances are computed. This process is continued until a suitable mate is found or all population members are exhausted in which case a random individual is chosen as a mate. In all simulations of this study, the $\sigma_{\text{mating}}$ parameter is kept the same as the value of $\sigma_{\text{share}}$ used for the niching method.

The mutation operator should also create a solution within the range of the joint variables given in eqns. (3.6) and (3.10). The parameter $\delta$, used in the mutation operator described in section 2.3.4, is therefore evaluated using eqns. (2.10) and (2.11).

### 3.4 3D ROBOT MODELLER

MATLAB is a powerful environment for a wide range of mathematical functions and graphical presentation that is available on a range of computer platforms. The core functionality can be extended by application specific toolboxes. Analysis work and robot representation for
purpose of graphical visualization in the current work has been carried out in MATLAB. A robotics toolbox developed by Corke (1996) provides many functions that are required in robotics including functions in the area of robot kinematics. However, it does not address the issues of multiplicity resolution of non-redundant robots and the collision avoidance of redundant robots which are the main objectives of the current work. Another major drawback of the toolbox is that the visual presentation of the robot is through line segments. Moreover, these line segments do not necessarily correspond to robot links but join the origins of sequential link coordinate frames. Though such an approach eliminates the need for additional detailed geometric data of the robot, the visualization of the robot is poor. A three-dimensional model of the robot through the use of MATLAB graphic functions is therefore desirable. The aim of the current MATLAB robot representation is to provide enough details for ease of visualization. However, it does not provide the power of geometric solid modellers of commercial CAD packages.

3.4.1 PRIMITIVES USED FOR LINK CONSTRUCTION

In the current work, primitives are used to construct the links of the robot. Links which cannot be constructed through these primitives are constructed by specifying their geometric data directly. The primitive geometric parameters and the robot link geometric data are defined in terms of the link coordinate frames.
The main primitives used in the robot model developer are:

1. Cylinder - The syntax of the cylinder primitive is as follows:

```cyl(base_centre_coordinate1, base_centre_coordinate2, base_radius, coordinate3_min, coordinate3_max, theta_start, theta_end, link_no, jt_variable, jt_type, index)
```

where

i) base_centre_coordinate1, base_centre_coordinate2 are the coordinates of the centre of the cylinder base in coordinate directions lying parallel to the cylinder base

ii) base radius is the cylinder radius

iii) coordinate3_min and coordinate3_max are the coordinates of the end-points of the cylinder axis in the third coordinate direction

iv) theta_start and theta_end describe the angular extent of the cylinder base (they are used to create partial cylinders; values of 0 and 2π generate a complete cylinder)

v) link_no is the link number of which the primitive is a part

vi) jt_variable is an array representing the joint variables of links 1 through ‘link_no’

vii) jt_type is an array specifying the type of joints (revolute or prismatic) from joint 1 through joint ‘link_no’

viii) index is a character representing the link coordinate axis (x,y,z) along which the cylinder axis lies.
The parameters pertaining to primitive geometric data for the cylinder primitive are illustrated in figs. 3.2(a) through 3.2(c).

2. Polyhedron - The syntax of the polyhedron primitive is as follows:

\[
polyhedron(mid\_xcoordinate, mid\_ycoordinate, length, parallel1, parallel2, zmin, zmax, link\_no, \text{jt}\_variable, \text{jt}\_type)
\]

where

i) mid\_xcoordinate, mid\_ycoordinate are the x and y-coordinates of the centre of the closer polyhedron face

ii) length is the linear dimension of the polyhedron in the link x-direction

iii) parallel1 and parallel2 are the linear dimensions of the polyhedron end faces in the local y-direction

iv) zmin and zmax define the height of the polyhedron in the link z-direction

The parameters link\_no, \text{jt}\_variable and \text{jt}\_type have the same meaning as is described for the cylinder primitive. The parameters pertaining to primitive geometric data for the polyhedron primitive are illustrated in fig. 3.3.

An additional "build" function is used to construct link parts which cannot be constructed using the above primitives. This function has the following syntax:

\[
\text{Build}(x\_face1, y\_face1, z\_face1, x\_face2, y\_face2, z\_face2, link\_no, \text{jt}\_variable, \text{jt}\_type)
\]
Fig. 3.2(a): Cylinder primitive generated through
\texttt{cyl(0, 0, 10, -50, -40, \pi, 2\pi, link_no, jt_variable, jt_type,'z')}
Fig. 3.2(b): Cylinder primitive generated through
\[ \text{cyl}(0,0,10,-20,-10,0,2\pi, \text{link_no, jt_variable, jt_type,'y'}) \]
Fig. 3.2(c): Cylinder primitive generated through
cyl(0,0,10,-30,-20, \pi, 2\pi,
link_no, jt_variable,
jt_type,'x')
Fig. 3.3: Polyhedron primitive generated through
polyhedron(-40,0,350,60,100,-53,
53,link_no, jt_variable,
jt_type)
where $x_{\text{face1}}, y_{\text{face1}}, z_{\text{face1}}, x_{\text{face2}}, y_{\text{face2}}, z_{\text{face2}}$ are vectors containing the x, y and z-coordinates of the vertices of faces 1 and 2 of the part. The generated part is the same as would be formed through a sweep from face 1 to face 2. The parameters link_no, jt_variable and jt_type have the same meaning as is described for the cylinder primitive. The parameters pertaining to geometric data for the build function are illustrated in fig. 3.4.

Fig. 3.5 shows a simple link of a robotic manipulator modelled using the above mentioned primitives.

For the actual link generation in the robot base coordinate frame, the link geometric data provided through primitive parameters or otherwise is internally transformed from the link coordinate frame to the robot base coordinate frame through the following transformation

$$[O_{o}] = [^{0}_{T_{i}}][O_{i}] \quad i = 1, 2, ..., m \quad (3.20)$$

where $[O_{i}]$ and $[O_{o}]$ represent the link geometric data in the link coordinate frame and base coordinate frame of the robot respectively and $[^{0}_{T_{i}}]$ represents the 4X4 homogeneous transformation matrix defining the $i^{th}$ link coordinate frame in terms of the robot base coordinate frame. This matrix can be evaluated using eqn.(3.2).

### 3.4.2 ROBOT MODEL GENERATION

The methodology of the robot model generation is explained through the flow chart given in fig. 3.6.

Figs. 3.7 and 3.8 show a SCARA robot and a PUMA robot with coordinate frames attached to the links. The geometric details of the three dimensional robot models developed in the current work, for purpose of illustration of simulation
Fig. 3.4: Part generated through
build(x1, y1, z1, x2, y2, z2, link_no, jt_variable, jt_type)
where x1={67 67 67 67 67 67 67 67 67 67 67}, y1={-35 -35 -55
-122 -122 -55 -35 -35 -75 -75
-35}, z1={53 85 85 50 -50 -85
-67}, y2=y1, z2=z1.
Fig. 3.5: Link generated through
polyhedron(0, 0, 432, 164, 299, 52, 149, link_no, jt_variable, 
jt_type),
cyl(0,0,82,52,149,-\pi/2, 
\pi/2,link_no, jt_variable, 
jt_type,z'),
cyl(-432, 0, 149.5, 52, 149, 
\pi/2,3\pi/2, link_no, jt_variable, 
jt_type,z')
Fig. 3.6: Methodology of robot model generation

Establish link coordinate system for robot links

Can all links be constructed using primitives?

Yes

Define primitive parameters for all links

No

1. Define primitive parameters wherever possible
2. Define parameters for build function

Define DH parameters of robot

Transform data to robot base coordinate frame

Synthesize robot
Fig. 3.7: SCARA robot
Fig. 3.8: PUMA robot
results, are given in sections A.1 and A.2 respectively. Figs. 3.9 and 3.10 show the models of the robots generated by MATLAB using the proposed methodology.

3.5 SIMULATION EXPERIMENTS

3.5.1 SCARA ROBOT

Simulation experiments were performed on the SCARA robot shown in fig. 3.7. The DH parameters of the robot are given in Table 3.1 and the limits on the first two joint variables $q_1$ and $q_2$ in radians are:

$$-2.01 \leq q_1 \leq 2.01$$
$$-2.53 \leq q_2 \leq 2.53$$

The multi-modal nature of the inverse kinematics problem for this robotic manipulator is due to the existence of multiple configurations which can result in the same wrist position of the robot. These configurations can be referred to as 'Left' and 'Right' configurations respectively. The third joint variable corresponding to the prismatic joint of the robot is used for the vertical positioning of the robot wrist. Its value remains the same for the two configurations of the robot that result in the same wrist position.

Using eqns. (3.1) and (3.2) the homogeneous transformation matrix from the base coordinate frame to the coordinate frame attached to the third link is given by

$$[0_{T_3}] = [0_{A_1}] [1_{A_2}] [2_{A_3}]$$

$$= \left[ \begin{array}{cccc} C_2C_1 - S_2S_1 & -C_2S_1 - S_2C_2 & a_4C_1 + a_2C_{12} & 0 \\ S_2C_2 + C_1S_1 & -S_2S_1 + C_1C_2 & a_4S_1 + a_2S_{12} & 0 \\ 0 & 0 & 1 & d_1 + d_2 + q_3 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

(3.21)
Fig. 3.9: SCARA robot model generated in MATLAB environment
Fig. 3.10: PUMA robot model generated in MATLAB environment
Table 3.1: DH parameters of SCARA robot

<table>
<thead>
<tr>
<th>Link number</th>
<th>$\theta_i$</th>
<th>$d_i$</th>
<th>$a_i$</th>
<th>$\alpha_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$q_1$</td>
<td>400 mm</td>
<td>580 mm</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>$q_2$</td>
<td>91 mm</td>
<td>470 mm</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>$q_3$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
The position vector of the robot wrist, \( \{^0P_w\} \), obtained from the last column of \( ^0T_3 \), is given by

\[
\{^0P_w\} = \begin{bmatrix} a_1C_1 + a_2C_{12} \\ a_1S_1 + a_2S_{12} \\ d_1 + d_2 + q_3 \end{bmatrix}
\] (3.22)

where \( C_i = \cos(q_i) \), \( S_i = \sin(q_i) \), \( C_{ij} = \cos(q_i + q_j) \) and \( S_{ij} = \sin(q_i + q_j) \).

The inverse kinematics problem for the SCARA robot can thus be stated as

\[
\text{Minimize} \quad \begin{bmatrix} a_1C_1 + a_2C_{12} \\ a_1S_1 + a_2S_{12} \\ d_1 + d_2 + q_3 \end{bmatrix} - \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}
\]
subject to \( q_k^l \leq q_k \leq q_k^u \quad k = 1, 2, 3 \) (3.23)

where \( X, Y \) and \( Z \) are the numerical values of the Cartesian coordinates at the desired wrist position of the robot. Since the vertical positioning for this robot is achieved through the third joint variable independently and this variable can be calculated through a single subtraction, the three-variable problem stated in eqn. (3.23) can be reduced to the following two-variable problem for obtaining the values of the first two joint variables

\[
\text{Minimize} \quad \begin{bmatrix} a_1C_1 + a_2C_{12} \\ a_1S_1 + a_2S_{12} \end{bmatrix} - \begin{bmatrix} X \\ Y \end{bmatrix}
\]
subject to \( q_k^l \leq q_k \leq q_k^u \quad k = 1, 2 \) (3.24)

To fix the population size of the real-coded GA the GA was run, with different populations sizes, using niching strategy 2 to obtain the multiple inverse kinematics solutions of the SCARA robot corresponding to
wrist position X=600mm, Y=400 mm and Z=130mm. Fifty runs were carried out, with different initial populations, for each value of population size. The value of the third joint variable was taken as -361mm which results in a z-position of 130 mm. Cross-over probability equal to 0.9, SBX distribution index $\eta$ equal to 50, s/N ratio equal to 0.4 and $t_{\max}$ (for purpose of calculation of mutation probability) equal to 800 were used in these runs.

Table 3.2 shows the values of the accuracy of the solutions and the minimum and mean deviation measure of the best population distribution obtained during 50 runs of the real-coded GA with different population sizes. This table shows that with a population size of 80 the GA evaluated the multiple configurations of the SCARA robot, with a wrist positioning error of less than 1mm, in 47 out of 50 runs. The success rate of 94% for a population size of 80 is considerably greater than the success rate for a population size of 40 whereas it is slightly better than the success rate for a population size of 60. Hence, a population size of 80 is used for obtaining the multiple inverse kinematics solutions of the SCARA robot.

Fig. 3.11 shows the variation of the deviation measure defined in eqn.(3.17) with generations using population sizes of 40, 60 and 80. These variations correspond to the runs for which the minimum value and mean value from 50 to 100 generations of the deviation measure have the least value. Table 3.2 and fig. 3.11 show that the minimum and mean value of deviation measure of the best distribution of population members around the multiple configurations of the SCARA robot are least for a population size of 80. The minimum value of deviation measure for this population size is 0 and the mean value from 50 to 100 generations is 0.25.
Table 3.2: Variation of accuracy of solutions and minimum and mean deviation measure of best distribution, for the SCARA robot inverse kinematics, with change of population size (position of wrist (600mm, 400mm, 130mm), cross-over probability = 0.9, SBX distribution index = 50, s/N = 0.4)

<table>
<thead>
<tr>
<th>Population size</th>
<th>Number of runs for which positioning error of configurations evaluated by GA is</th>
<th>Details of best distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Between 0 and 1 mm</td>
<td>Between 1 and 2 mm</td>
</tr>
<tr>
<td>40</td>
<td>30</td>
<td>13</td>
</tr>
<tr>
<td>60</td>
<td>44</td>
<td>6</td>
</tr>
<tr>
<td>80</td>
<td>47</td>
<td>3</td>
</tr>
</tbody>
</table>
Fig. 3.11: Variation of deviation measure, defined in eqn. (3.17), with generations obtained during SCARA robot inverse kinematics solution using population sizes of 40, 60 and 80.
The effect of s/N ratio, used in niching strategy 2, on the accuracy of solutions was studied by running the real-coded GA for different s/N values. The cross-over probability, SBX distribution index $\eta$ and $t_{max}$ (for purpose of calculation of mutation probability) were kept fixed at the values used for determining the population size and the population size was kept fixed at 80. Fifty runs were carried out, with different initial populations, for each value of s/N ratio. Table 3.3 shows the values of the accuracy of the solutions and the minimum and mean deviation measure of the best population distribution obtained during 50 runs of the real-coded GA with different s/N ratios. With a s/N ratio of 0.4, the real-coded GA evaluated the multiple configurations of the SCARA robot, with a wrist positioning error of less than 1mm, in 47 out of 50 runs. It can be seen from Table 3.3 that the performance of the real-coded GA with a s/N ratio of 0.4 is significantly better than its performance with a s/N ratio of 0.2 whereas there is no significant change in the performance when the s/N ratio is changed to 0.6 or 0.8. A s/N ratio of 0.4 is used for obtaining the multiple inverse kinematics solutions of the SCARA robot since large values of s/N ratio makes the implementation computationally expensive.

Fig. 3.12 shows the variation of the deviation measure defined in eqn.(3.17) with generations using s/N ratios of 0.2, 0.4, 0.6 and 0.8. These variations correspond to the runs for which the minimum value and mean value from 50 to 100 generations of the deviation measure have the least value. Table 3.3 shows that the minimum and mean value of deviation measure of the best distribution of population members around the multiple configurations of the SCARA robot are least for a s/N
Table 3.3: Variation of accuracy of solutions and minimum and mean deviation measure of best distribution, for the SCARA robot inverse kinematics, with change of s/N ratio of niching strategy 2 (position of wrist (600mm, 400mm, 130mm), cross-over probability = 0.9, SBX distribution index = 50, population size = 80)

<table>
<thead>
<tr>
<th>s/N</th>
<th>Number of runs for which positioning error of configurations evaluated by GA is</th>
<th>Details of best distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Between 0 and 1mm</td>
<td>Between 1 and 2 mm</td>
</tr>
<tr>
<td>0.2</td>
<td>42</td>
<td>8</td>
</tr>
<tr>
<td>0.4</td>
<td>47</td>
<td>3</td>
</tr>
<tr>
<td>0.6</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>0.8</td>
<td>48</td>
<td>1</td>
</tr>
</tbody>
</table>
Fig. 3.12: Variation of deviation measure, defined in eqn. (3.17), with generations obtained during SCARA robot inverse kinematics solution using s/N ratios of 0.2, 0.4, 0.6 and 0.8.
ratio of 0.4. The minimum value of deviation measure for this s/N ratio is 0 and the mean value from 50 to 100 generations is 0.25.

Real-coded GAs using niching strategies 1 and 2 have been compared on the basis of their abilities to provide the multiple configurations of the SCARA robot (Kalra et al., 2004). The parameter $n'$ of niching strategy 1 was varied and the real-coded GA was run with fifty different initial populations for each value of parameter $n'$. The values of cross-over probability, SBX distribution index $\eta$, $t_{\text{max}}$ (for purpose of calculation of mutation probability) and population size were the same as those for the GA implementation using niching strategy 2. Table 3.4 shows the accuracy of the solutions and the minimum and mean deviation measure of the best population distribution obtained for different values of parameter $n'$ of niching strategy 1. The table shows that when $n'$ is 50, the real-coded GA evaluated the multiple configurations of the SCARA robot, with a positioning error of less than 1mm, in 47 out of 50 runs. The success rate of 94% is considerably greater than the success rate for $n'$ values of 40 and 45. The success rate for a $n'$ value of 50 is the same as the success rate of the real-coded GA using niching strategy 2 with s/N ratio equal to 0.4.

Fig. 3.13 shows the variation of the deviation measure defined in eqn. (3.17) with generations using $n'$ values of 40, 45 and 50. These variations correspond to the runs for which the minimum value and mean value from 50 to 100 generations of the deviation measure have the least value. Table 3.4 shows that the minimum and mean value of deviation measure of the best distribution of population members around the multiple configurations of
Table 3.4: Variation of accuracy of solutions and minimum and mean deviation measure of best distribution, for the SCARA robot inverse kinematics, with change of \( n' \) of niching strategy 1 (position of wrist(600mm,400mm,130mm), cross-over probability=0.9, SBX distribution index = 50, population size=80)

<table>
<thead>
<tr>
<th>( n' )</th>
<th>Number of runs for which positioning error of configurations evaluated by GA is</th>
<th>Details of best distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Between 0 and 1 mm</td>
<td>Between 1 and 2 mm</td>
</tr>
<tr>
<td>40</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>45</td>
<td>31</td>
<td>19</td>
</tr>
<tr>
<td>50</td>
<td>47</td>
<td>3</td>
</tr>
</tbody>
</table>
Fig. 3.13: Variation of deviation measure, defined in eqn. (3.17), with generations obtained during SCARA robot inverse kinematics solution using $n^*$ parameter values of 40, 45 and 50.
the SCARA robot are least for a \( n' \) of 50. The minimum value of deviation measure for this \( n' \) value is 0 and the mean value from 50 to 100 generations is 1.43. The corresponding values obtained for the GA implementation using niching strategy 2, with \( s/N \) ratio of 0.4, are 0 and 0.25. The mean values of deviation measure for all runs of the real-coded GA, using niching strategy 1 with \( n' \) value of 50 and niching strategy 2 with \( s/N \) ratio of 0.4, are given in Appendix B.

From these results, it can be inferred that real-coded GA implementations using niching strategies 1 and 2 are able to evaluate the multiple inverse kinematics solutions of the SCARA robot for the same percentage of runs. Real-coded GA using niching strategy 2, that is, the niching strategy proposed by Deb and Agrawal (1999) was used in all subsequent simulation experiments of the SCARA robot. The following control parameters were thereafter used for the GA: Population size = 80, Cross-over probability = 0.9, SBX distribution index \( \eta = 50 \), \( t_{max} \) (for purpose of calculation of mutation probability) = 800, \( s/N \) ratio for niching strategy 2 = 0.4.

Several simulation experiments were conducted, each corresponding to the same initial and different final configuration of the SCARA robotic manipulator. The initial configuration corresponds to the joint variable vector \( (0 \ 0)^T \) radians. The final configurations are specified through the x-coordinate, y-coordinate and z-coordinate of the robot wrist. The value of the third joint variable in all the experiments is taken as -361mm which results in a z-position of 130 mm.

Table 3.5 shows the values of the joint variables evaluated by the real-coded GA using niching strategy 2.
Table 3.5: Results of simulation experiments performed on SCARA robot

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Position of wrist (coordinates in mm)</th>
<th>$q_1$ (in rad)</th>
<th>$q_2$ (in rad)</th>
<th>Positioning error of wrist (in mm)</th>
<th>Fitness value of feasible solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(600,400,130)</td>
<td>-0.1201</td>
<td>1.6380</td>
<td>0.69</td>
<td>1.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.2960</td>
<td>-1.6404</td>
<td>0.50</td>
<td>2.09</td>
</tr>
<tr>
<td>2</td>
<td>(400,-600,130)</td>
<td>-1.6907</td>
<td>1.6383</td>
<td>0.45</td>
<td>2.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.2741</td>
<td>-1.6401</td>
<td>0.48</td>
<td>1.66</td>
</tr>
<tr>
<td>3</td>
<td>(350,350,130)</td>
<td>-0.1066</td>
<td>2.1804</td>
<td>0.14</td>
<td>2.18</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.6783</td>
<td>-2.1807</td>
<td>0.44</td>
<td>2.75</td>
</tr>
<tr>
<td>4</td>
<td>(-100,700,130)</td>
<td>0.9903</td>
<td>1.6762</td>
<td>0.16</td>
<td>1.95</td>
</tr>
<tr>
<td>5</td>
<td>(650,-450,130)</td>
<td>-1.2358</td>
<td>1.4449</td>
<td>0.49</td>
<td>1.90</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0257</td>
<td>-1.4475</td>
<td>0.41</td>
<td>1.45</td>
</tr>
</tbody>
</table>
The positioning errors of the wrist, for the values of the joint variables obtained using this GA implementation, are also given in the table. Table 3.5 shows that the real-coded GA is able to evaluate the multiple configurations of the SCARA robot at different wrist locations, within a positioning error of 1mm, in all the simulation experiments. For the simulation experiment no. 4, only one possible configuration of the robot exists due to the limits on the joint variables. The real-coded GA evaluates only this configuration even when the number of configurations is input as two. If the geometric approach given in Fu et al. (1987) was used, one out of two calculated configurations would have to be discarded by the user after observing the limits on the joint variables.

The fitness values of the feasible multiple solutions evaluated by the real-coded GA using niching strategy 2, $A_q$, for the simulation experiments are given in Table 3.5 and can be used for multiplicity resolution. Figs. 3.14(a) through 3.14(d) show the distribution of individuals in the population at the initial, two intermediate and generation number 50 of the real-coded GA for simulation experiment no. 1. For this simulation experiment, the robot configuration described by the joint variable vector $\{-0.1201\ ,\ 1.6\ ,\ 3\ ,\ 8\ ,\ 0\}_{\rm\ radians}$ would be preferred over the configuration described by the joint variable vector $\{1.2960\ ,\ -1.6404\}_{\rm\ radians}$ to achieve the wrist position $(600\, \text{mm},\ 400\, \text{mm},\ 130\, \text{mm})$ since it involves a smaller total joint displacement. The distribution of individuals in the population at the initial generation and generation numbers 5, 10 and 70 of the real-coded GA using niching strategy 2, for simulation experiment no. 5, is shown in figs. 3.15(a) through 3.15(d).
Fig. 3.14: Distribution of individuals in population at (a) initial generation (b) generation no. 5 (c) generation no. 10 (d) generation no. 50 of simulation experiment no. 1 of SCARA robot.
Fig. 3.15: Distribution of individuals in population at (a) initial generation (b) generation no. 5 (c) generation no. 10 and (d) generation no. 70 of simulation experiment no. 5 of SCARA robot
The 3D modeller developed in MATLAB and described in section 3.4 was used to visualize the multiple configurations obtained through the inverse kinematics solution. Fig. 3.16 shows the SCARA robot with the wrist at position (600mm, 400mm, 130mm) in both the 'Right' and 'Left' configurations respectively.

3.5.2 PUMA ROBOT

3.5.2.1 SOLUTION OF WRIST POSITIONING JOINT VARIABLES

Simulation experiments were also performed on the PUMA robot shown in fig. 3.8. The DH parameters of this robot are given in Table 3.6.

The limits on the joint variables $q_1$ through $q_6$ in radians are as under:

\[-2.79 \leq q_1 \leq 2.79\]
\[-3.93 \leq q_2 \leq 0.79\]
\[-0.79 \leq q_3 \leq 3.93\]
\[-1.92 \leq q_4 \leq 2.97\]
\[-1.75 \leq q_5 \leq 1.75\]
\[-4.64 \leq q_6 \leq 4.64\]

The multi-modal nature of the inverse kinematics problem for this robotic manipulator is due to the existence of multiple configurations which can result in the same wrist position of the robot. These configurations can be referred to as 'Left and above arm', 'Left and below arm', 'Right and above arm' and 'Right and below arm' configurations respectively.

Using eqns. (3.1) and (3.2) the homogeneous transformation matrix from the base coordinate frame to the coordinate frame attached to the sixth link can be obtained. The position vector of the robot hand, $\{^6p_T\}$, can be
Fig. 3.16: (a) Right and (b) left configurations of SCARA robot with wrist at position (600mm, 400mm, 130mm)
Table 3.6: DH parameters of PUMA robot

<table>
<thead>
<tr>
<th>Link number</th>
<th>$\theta_i$</th>
<th>$d_i$ (in mm)</th>
<th>$a_i$ (in mm)</th>
<th>$\alpha_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$q_1$</td>
<td>0</td>
<td>0</td>
<td>$-90^\circ$</td>
</tr>
<tr>
<td>2</td>
<td>$q_2$</td>
<td>149.09</td>
<td>431.8</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>$q_3$</td>
<td>0</td>
<td>-20.32</td>
<td>$90^\circ$</td>
</tr>
<tr>
<td>4</td>
<td>$q_4$</td>
<td>433.07</td>
<td>0</td>
<td>$-90^\circ$</td>
</tr>
<tr>
<td>5</td>
<td>$q_5$</td>
<td>0</td>
<td>0</td>
<td>$-90^\circ$</td>
</tr>
<tr>
<td>6</td>
<td>$q_6$</td>
<td>56.25</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
extracted from this homogeneous transformation matrix and is given by

\[
\mathbf{h}_x = C_i \left( d_6 (C_{23}C_4 + S_{23}C_5) + S_{23}d_4 + a_3C_{23} + a_2C_2 \right) \\
- S_i \left( d_6S_3 + d_4 \right) \\
\mathbf{h}_y = S_i \left( d_6 (C_{23}C_4 + S_{23}C_5) + S_{23}d_4 + a_3C_{23} + a_2C_2 \right) + C_i \left( d_6S_3 + d_4 \right) \\
\mathbf{h}_z = d_i \left( C_{23}C_2 - S_{23}C_4 \right) + C_{23}d_4 - a_3S_{23} - a_2S_2
\] (3.25)

The position vector of the robot wrist, \(\mathbf{P}_w\), can be obtained using eqn. (3.3) as

\[
\mathbf{P}_w = \begin{pmatrix}
C_i(a_2C_{23} + a_3C_2) - d_4S_2 \\
S_i(a_2C_{23} + a_3C_2) + d_4C_1 \\
d_4C_{23} - a_3S_{23} - a_2S_2
\end{pmatrix}
\] (3.26)

The inverse kinematics problem for the wrist positioning of the PUMA robot can now be stated as

Minimize

\[
\begin{pmatrix}
C_i(a_2C_{23} + a_3C_2) - d_4S_2 \\
S_i(a_2C_{23} + a_3C_2) + d_4C_1 \\
d_4C_{23} - a_3S_{23} - a_2S_2
\end{pmatrix} - \begin{pmatrix}
X \\
Y \\
Z
\end{pmatrix}
\]

subject to \(q_k^l \leq q_k \leq q_k^u\) \(k = 1, 2, 3\)

To fix the population size of the real-coded GA the GA was run, with different population sizes, using niching strategy 2 to obtain the multiple inverse kinematics solutions of the PUMA robot corresponding to wrist position \(X=540\,\text{mm}, Y=210\,\text{mm}\) and \(Z=260\,\text{mm}\). Fifty runs were carried out, with different initial populations, for each value of population size. Cross-over probability equal to 0.9, SBX distribution index \(\eta\) equal to 150, s/N ratio equal to 0.4 and \(t_{max}\) (for purpose of calculation of mutation probability) equal to 800 were used in these runs. Table 3.7 shows the values of the accuracy of the solutions and the minimum and mean deviation measure of
Table 3.7: Variation of accuracy of solutions and minimum and mean deviation measure of best distribution, for the PUMA robot inverse kinematics, with change of population size (position of wrist (540mm, 210mm, 260mm), cross-over probability = 0.9, SBX distribution index = 150, s/N = 0.4).

<table>
<thead>
<tr>
<th>Population size</th>
<th>Number of runs for which the number of configurations evaluated by the evolutionary approach, within a positioning error of 2 mm, is</th>
<th>Details of best distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Four</td>
<td>Three</td>
</tr>
<tr>
<td>120</td>
<td>4</td>
<td>41</td>
</tr>
<tr>
<td>150</td>
<td>32</td>
<td>18</td>
</tr>
<tr>
<td>180</td>
<td>32</td>
<td>18</td>
</tr>
</tbody>
</table>
the best population distribution obtained during 50 runs of the real-coded GA with different population sizes. This table shows that with a population size of 150 the GA evaluated all the multiple configurations of the PUMA robot, with a wrist positioning error of less than 2 mm, in 32 out of 50 runs. The success rate of 64% for a population size of 150 is considerably greater than the success rate for a population size of 120 whereas it is the same as the success rate for a population size equal to 180. A large population size makes the implementation computationally expensive. Hence, a population size of 150 is used for obtaining the multiple inverse kinematics solutions of the PUMA robot.

Fig. 3.17 shows the variation of the deviation measure defined in eqn.(3.17) with generations using population sizes of 120, 150 and 180. These variations correspond to the runs for which the minimum value and mean value from 150 to 300 generations of the deviation measure have the least value. Table 3.7 and fig. 3.17 show that the minimum and mean value of deviation measure of the best distribution of population members around the multiple configurations of the PUMA robot are least for a population size of 150. The minimum value of deviation measure for this population size is 1.74 and the mean value from 150 to 300 generations is 4.05.

The effect of s/N ratio, used in niching strategy 2, on the accuracy of solutions was studied by running the real-coded GA for different s/N values. The cross-over probability, SBX distribution index $\eta$ and $t_{\text{max}}$ (for purpose of calculation of mutation probability) were kept fixed at the values used for determining the population size and the population size was kept fixed at 150. Fifty runs each were carried out, with different initial populations, for
Fig. 3.17: Variation of deviation measure, defined in eqn. (3.17), with generations obtained during PUMA robot inverse kinematics solution using population sizes of 120, 150 and 180.
various s/N ratios. Table 3.8 shows the values of the accuracy of the solutions and the minimum and mean deviation measure of the best population distribution obtained during 50 runs of the real-coded GA with different s/N ratios. With a s/N ratio of 0.4, the real-coded GA evaluated the four configurations of the PUMA robot, with a wrist positioning error of less than 2mm, in 32 out of 50 runs. It can be seen from Table 3.8 that with a s/N ratio of 0.4 the real-coded GA gives a significantly better performance in obtaining the multiple configurations of the PUMA robot than with s/N ratios of 0.2, 0.6 or 0.8. A s/N ratio of 0.4 is therefore used for obtaining the multiple inverse kinematics solutions of the PUMA robot.

Fig. 3.18 shows the variation of the deviation measure defined in eqn. (3.17) with generations using s/N ratios of 0.2, 0.4, 0.6 and 0.8. These variations correspond to the runs for which the minimum value and mean value from 150 to 300 generations of the deviation measure have the least value. Table 3.8 and fig. 3.18 show that the minimum and mean value of deviation measure of the best distribution of population members around the multiple configurations of the PUMA robot are least for a s/N ratio of 0.4. The minimum value of deviation measure for this s/N ratio is 1.74 and the mean value from 150 to 300 generations is 4.05.

Real-coded GAs using niching strategies 1 and 2 were again compared on the basis of their abilities to provide the multiple configurations of the PUMA robot. The parameter $n^*$ of niching strategy 1 was varied and the real-coded GA was run with thirty different initial populations for each value of $n^*$. The values of cross-over probability, SBX distribution index $\eta$, $t_{\text{max}}$ (for purpose of
Table 3.8: Variation of accuracy of solutions and minimum and mean deviation measure of best distribution, for the PUMA robot inverse kinematics, with change of s/N ratio of niching strategy 2 (position of wrist (540mm, 210mm, 260mm), cross-over probability=0.9, SBX distribution index = 150, population size=150)

<table>
<thead>
<tr>
<th>s/N ratio</th>
<th>Number of runs for which the number of configurations evaluated by the evolutionary approach, within a positioning error of 2 mm, is</th>
<th>Details of best distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Four</td>
<td>Three</td>
</tr>
<tr>
<td>0.2</td>
<td>25</td>
<td>23</td>
</tr>
<tr>
<td>0.4</td>
<td>32</td>
<td>18</td>
</tr>
<tr>
<td>0.6</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>0.8</td>
<td>23</td>
<td>26</td>
</tr>
</tbody>
</table>
Fig. 3.18: Variation of deviation measure, defined in eqn. (3.17), with generations obtained during PUMA robot inverse kinematics solution using s/N ratios of 0.2, 0.4, 0.6 and 0.8
calculation of mutation probability) and population size were the same as the values taken for the GA implementation using niching strategy 2. Table 3.9 shows the accuracy of the solutions and the minimum and mean deviation measure of the best population distribution for different values of parameter \( n' \) of niching strategy 1. The table shows that with \( n' \) values of 45 and 50, the GA evaluated all the multiple configurations of the PUMA robot, with a wrist positioning error of less than 2mm, in 1 out of 30 runs. However, considering the evaluations of a subset of all the multiple configurations, a \( n' \) value of 45 performs better than a \( n' \) value of 50. The success rate of the real-coded GA using niching strategy 1 with a \( n' \) value of 45 is, however, much smaller than the success rate of the real-coded GA using niching strategy 2 with a \( s/N \) ratio of 0.4.

Fig. 3.19 shows the variation of the deviation measure defined in eqn. (3.17) with generations using \( n' \) values of 35, 40, 45 and 50. These variations correspond to the runs for which the minimum value and mean value from 150 to 300 generations of the deviation measure have the least value. Table 3.9 shows that the minimum and mean value of deviation measure of the best population distribution have comparable values for \( n' \) values of 45 and 50. The minimum value of deviation measure is 0.42 and 0.57 and the mean value from 150 to 300 generations is 4.78 and 4.57 for \( n' \) values of 45 and 50 respectively. The corresponding values for the real-coded GA using niching strategy 2, with a \( s/N \) ratio of 0.4, are 1.74 and 4.05. The mean values of deviation measure for all runs of the real-coded GA, using niching strategy 1 with \( n' \) value of 45
Table 3.9: Variation of accuracy of solutions and minimum and mean deviation measure of best distribution, for the PUMA robot inverse kinematics, with change of n’ of niching strategy 1 (position of wrist (540mm,210mm,260mm), cross-over probability=0.9, SBX distribution index = 150, population size=150)

<table>
<thead>
<tr>
<th>n’ value</th>
<th>Number of runs for which the number of configurations evaluated by the evolutionary approach, within a positioning error of 2 mm, is</th>
<th>Details of best distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Four</td>
<td>Three</td>
</tr>
<tr>
<td>35</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>45</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>50</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Fig. 3.19: Variation of deviation measure, defined in eqn. (3.17), with generations obtained during PUMA robot inverse kinematics solution using n* parameter values of 35, 40, 45 and 50.
and niching strategy 2 with s/N ratio of 0.4, are given in Appendix B.

On the basis of the success rates, it can be inferred that the performance of the real-coded GA using niching strategy 2 is better than that of the real-coded GA using niching strategy 1 for evaluating the multiple inverse kinematics solutions of the PUMA robot. Hence, the real-coded GA using niching strategy 2, that is, the niching strategy proposed by Deb and Agrawal (1999) is used in all subsequent simulation experiments of the PUMA robot.

The following control parameters were thereafter used for the GA: Population size = 150, Cross-over probability = 0.9, SBX distribution index $\eta = 150$, $t_{\text{max}}$ (for purpose of calculation of mutation probability) = 800, s/N ratio for niching strategy 2 = 0.4. Several simulation experiments were conducted, each corresponding to the same initial and different final configuration of the PUMA robotic manipulator. The initial configuration corresponds to the joint variable vector $\{n/2 \ 0 \ n/2\}^T$ radians. The final configurations are specified through the x-coordinate, y-coordinate and z-coordinate of the robot wrist.

Table 3.10 shows the values of the joint variables obtained using the geometric approach presented in Fu et al. (1987) and the real-coded GA using niching strategy 2. The minimum positioning errors of the wrist, for the values of the joint variables obtained using this GA implementation, are also given in the table. Table 3.10 shows that the maximum value of the positioning error is 1.91mm. The results of Table 3.10 show that the real-coded GA using niching strategy 2 is able to evaluate the
Table 3.10: Results of simulation experiments performed on PUMA robot

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Position of wrist</th>
<th>Results of geometric approach</th>
<th>Results of real-coded GA using niching strategy 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(600,149.09,200)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>q₁ (in rad)</td>
<td>q₂ (in rad)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>-1.0745</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.4313</td>
<td>0.1153</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2.6545</td>
<td>-2.0668</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2.6545</td>
<td>2.7103</td>
</tr>
<tr>
<td>2</td>
<td>(500,240,230)</td>
<td>0.1754</td>
<td>-1.2425</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1754</td>
<td>0.4294</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2.4219</td>
<td>-1.8991</td>
</tr>
<tr>
<td>3</td>
<td>(540,210,260)</td>
<td>0.1106</td>
<td>-1.2132</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.1106</td>
<td>0.3437</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2.5105</td>
<td>-1.9284</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2.5105</td>
<td>2.7979</td>
</tr>
<tr>
<td>4</td>
<td>(180,400,400)</td>
<td>-1.4947</td>
<td>-1.6169</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1.4947</td>
<td>0.0769</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.3405</td>
<td>-1.5247</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.3405</td>
<td>3.0647</td>
</tr>
<tr>
<td>5</td>
<td>(-180,400,-200)</td>
<td>1.6468</td>
<td>-0.5643</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.6468</td>
<td>1.4671</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0.8011</td>
<td>-2.5773</td>
</tr>
</tbody>
</table>
possible configurations of the PUMA robot at different wrist locations, within a positioning error of 2mm, in all the simulation experiments. For the simulation experiment no. 5, only two possible configurations of the robot exist due to the limits on the joint variables. The real-coded GA evaluates only two configurations even if the number of multiple configurations input by the user is four. In the geometric approach, two out of four calculated configurations have to be discarded by the user after observing the limits on the joint variables. Further, the geometric approach evaluates all the joint variables in the range \([-n \, n]\). The values of the evaluated joint variables may thus seem to lie outside the actual joint variable limits as can be seen for the values of \(q_2\) in the fourth configuration of simulation experiment no. 1, the value of \(q_3\) in the first configuration of simulation experiment nos. 2, 3, 4 and 5 and the values of \(q_2\) and \(q_3\) in the fourth configuration of simulation experiment nos. 2, 3 and 4. However, these values lie within the joint variable limits based on the link coordinate frame assignment of the PUMA robot shown in fig. 3.8 considering the fact that a joint variable value \(+\beta\) and \((-2n - \beta)\) are equivalent. The real-coded GA evaluates the joint variable values within the joint variable limits correctly in these instances.

The fitness values of the feasible multiple solutions evaluated by the real-coded GA using niching strategy 2, \(\Delta q\), for the simulation experiments are given in Table 3.10 and can be used for multiplicity resolution. Figs. 3.20(a) through 3.20(d) show the distribution of individuals in the population at the initial, two intermediate and generation number 260 of the real-coded GA for simulation experiment no. 1. For this simulation experiment, the
Fig. 3.20: Distribution of individuals in population at (a) initial generation (b) generation no. 20 (c) generation no. 100 and (d) generation no. 260 of simulation experiment no. 1 of PUMA robot (o-optimal points, x- population members)
robot configuration described by the joint variable vector 
\((-0.0014 \ 0.4304 \ 0.1152)\) radians would be preferred over the other three possible configurations to achieve the wrist position (600mm, 149.09mm, 200mm) since it involves a smaller total joint displacement.

The 3D modeller developed in MATLAB and described in section 3.4 was used to visualize the multiple configurations obtained through the inverse kinematics solution. Fig. 3.21 shows the PUMA robot with the wrist at position (600mm, 149.09mm, 200mm) in all possible configurations.

### 3.5.2.2 SOLUTION OF WRIST POSITIONING AND HAND ORIENTING JOINT VARIABLES

A simulation experiment was carried out wherein the desired hand matrix of the PUMA robot was taken as

\[
\begin{bmatrix}
-0.0470 & -0.7892 & -0.6124 & 540 \\
0.6597 & 0.4346 & -0.6124 & 210 \\
0.7500 & -0.4330 & 0.5000 & 260 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  

(3.28)

The method of obtaining the wrist position from the hand matrix for the PUMA robot is given in Annexure C. Using eqn. (C.1), the desired position of the robot wrist corresponding to the hand matrix given in eqn. (3.28) is \(\{^W_{\omega,\text{des}}\} = \{574.45 \ 244.45 \ 231.87\}\) mm. The solution of the wrist positioning joint variables is obtained using the procedure discussed in section 3.5.2.1.

Using eqns. (3.1) and (3.2) to evaluate \(\begin{bmatrix}^C_T \end{bmatrix}\) and \(\begin{bmatrix}^C_T \end{bmatrix}\) and extracting the appropriate rotation matrices, we may
Fig. 3.21: Multiple configurations of PUMA robot at wrist position (600mm, 149.09mm, 200mm)
obtain the expressions for $[^{0}R_{9}]$ and $[^{3}R_{6}]$ of the PUMA robot. These rotation matrices are as under:

$$[^{0}R_{9}] = \begin{bmatrix} C_{1}C_{23} & -S_{1} & C_{1}S_{23} \\ S_{1}C_{23} & C_{1} & S_{1}S_{23} \\ -S_{23} & 0 & C_{23} \end{bmatrix}$$

$$[^{3}R_{6}] = \begin{bmatrix} C_{4}C_{5}C_{6} - S_{4}S_{6} & -C_{4}C_{5}S_{6} - S_{4}C_{6} & C_{4}S_{6} \\ S_{6}C_{5} + C_{4}S_{6} & -S_{4}C_{5}S_{6} + C_{4}C_{6} & S_{4}S_{5} \\ -S_{6}C_{5} & S_{6}S_{5} & C_{5} \end{bmatrix}$$

(3.29)

These rotation matrices are substituted in eqn. (3.9) to set up the optimization problem given in eqn. (3.11) for obtaining the hand orienting joint variables of the PUMA robot.

Table 3.11(a) shows the values of the wrist positioning joint variables evaluated by the geometric approach given in Fu et al. (1987) and the real-coded GA using niching strategy 2. The positioning errors of the wrist, for the values of the joint variables obtained using this GA implementation, are also given in the table. Table 3.11(a) shows that the maximum value of the positioning error is 1.63 mm. The geometric approach evaluates all the joint variables in the range $[-\pi \pi]$. The values of the joint variables evaluated may thus seem to lie outside the actual joint variable limits as can be seen for the value of $q_{2}$ in the fourth configuration. However, this value lies within the joint variable limits based on the link coordinate frame assignment of the PUMA robot shown in fig. 3.8 considering the fact that a joint variable value 2.7825 radians and -3.5007 radians are equivalent. The real-coded GA evaluates the joint variable value within the joint variable limits correctly in this instance.
### Table 3.11: Results of simulation experiments performed on PUMA robot

**(a) Solution of wrist positioning joint variables for hand matrix given in eqn. (3.28)**

<table>
<thead>
<tr>
<th>Configuration number</th>
<th>Results of geometric approach</th>
<th>Results of GA implementation using niching strategy 2</th>
<th>Positioning error of wrist (in mm)</th>
<th>Fitness value of feasible solutions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$q_1$ (in rad)</td>
<td>$q_2$ (in rad)</td>
<td>$q_3$ (in rad)</td>
<td>$q_1$ (in rad)</td>
</tr>
<tr>
<td>1</td>
<td>0.1612</td>
<td>-1.0897</td>
<td>3.0630</td>
<td>0.1608</td>
</tr>
<tr>
<td>2</td>
<td>0.1612</td>
<td>0.3591</td>
<td>0.1724</td>
<td>0.1606</td>
</tr>
<tr>
<td>3</td>
<td>-2.4981</td>
<td>-2.0519</td>
<td>0.1724</td>
<td>-2.4995</td>
</tr>
<tr>
<td>4</td>
<td>-2.4981</td>
<td>2.7825</td>
<td>3.0630</td>
<td>-2.4978</td>
</tr>
</tbody>
</table>

**Configuration number**

1, 2, 3, 4

### (b) Solution of hand orienting joint variables for configuration no. 2 and 4

<table>
<thead>
<tr>
<th>Configuration number</th>
<th>Results of geometric approach</th>
<th>Results using niching strategy 2</th>
<th>Absolute percentage error (in %) of $q_4$, $q_5$, $q_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$q_4$ (in rad)</td>
<td>$q_5$ (in rad)</td>
<td>$q_6$ (in rad)</td>
</tr>
<tr>
<td>2</td>
<td>0.5324</td>
<td>-1.4959</td>
<td>0.8252</td>
</tr>
<tr>
<td></td>
<td>-2.6092</td>
<td>1.4959</td>
<td>-2.3164</td>
</tr>
<tr>
<td>4</td>
<td>-3.0183</td>
<td>-1.4812</td>
<td>0.5838</td>
</tr>
</tbody>
</table>
Figs. 3.22(a) through 3.22(d) show the distribution of individuals in the population at the initial, two intermediate and generation number 240 of the real-coded GA during the solution of the wrist positioning joint variables for achieving the hand matrix given in eqn. (3.28).

For the hand matrix given in eqn. (3.28), the desired hand orientation is achievable only for the second and fourth configurations of wrist positioning. Corresponding to these configurations, the solution of the hand orienting joint variables obtained using the geometric approach presented in Fu et al. (1987) and the real-coded GA using niching strategy 2 are also given in Table 3.11(b). For these wrist positioning configurations, only one of the two possible hand orienting configurations of the robot exist due to the limits on the joint variables. The real-coded GA evaluates only this hand orienting configuration even if the number of hand orienting configurations provided as input by the user is two. In the geometric approach one out of the two calculated configurations has to be discarded by the user after observing the limits on the joint variables. Using Table 3.11(b), it can be seen that the maximum error in the value of the hand orienting variables is 0.41%. The real-coded GA is thus able to evaluate both the wrist positioning and hand orienting joint variables of the PUMA robot for all possible configurations. Fig. 3.23 shows all possible configurations of the PUMA robot corresponding to the hand matrix given in eqn.(3.28). These configurations are displayed using the 3D modeller developed in MATLAB.

The fitness values of the feasible multiple solutions evaluated by the real-coded GA using niching strategy 2, $Aq$, for the simulation experiment are also
Fig. 3.22: Distribution of individuals in population at (a) initial generation (b) generation no. 20 (c) generation no. 50 and (d) generation no. 240 during solution of wrist positioning joint variables for achieving hand matrix given in eqn. (3.28) (o-optimal points, x-population members)
Fig. 3.23: Multiple configurations of PUMA robot corresponding to hand matrix given in eqn. (3.28)
given in Table 3.11 and can be used for multiplicity resolution. The initial configuration corresponds to the wrist positioning joint variable vector \( \{n/2 \ 0 \ n/2\}^T \) radians. For this simulation experiment, the robot configuration described by the wrist positioning joint variable vector \( \{0.1606 \ 0.3607 \ 0.1697\}^T \) radians would be preferred over the other possible configuration to achieve the hand matrix given in eqn. (3.28) since it involves a smaller total joint displacement for the wrist positioning joint variables. The joint variable vector for the required hand matrix is thus \( \{0.1606 \ 0.3607 \ 0.1697 \ 0.5346 \ -1.4923 \ 0.8244\}^T \) radians which corresponds to the configuration shown in fig. 3.23(a).

3.6 CONCLUSIONS

An evolutionary approach based on a real-coded GA has been used to evaluate multiple inverse kinematics solutions of non-redundant industrial robots. Simulation experiments were carried out on a SCARA robot and a PUMA robot. A robot 3D modeller developed in MATLAB was used to display the multiple configurations of the robot evaluated using the evolutionary approach. Based on the simulation experiments carried out in section 3.5, the following conclusions can be drawn:-

1. An evolutionary approach based on a real-coded genetic algorithm is able to solve the inverse kinematics problem of non-redundant robotic manipulators. The approach provides the multiple configurations of the robotic manipulator considering the restrictions imposed by the joint limits. An acceptable distribution of population members around the multiple inverse kinematics solutions is also obtained. The total joint displacement, associated
with these multiple robot configurations, for the wrist positioning joint variables can be used for multiplicity resolution of the robotic manipulator. The evolutionary approach is therefore appropriate for obtaining the multiple inverse kinematics solutions of a robotic manipulator and subsequently achieving a multiplicity resolution. The evolutionary approach has the following advantages over the geometric approach:

(i) The geometric approach uses geometric heuristics to take advantage of the special structure of the manipulator for finding the joint variables. The evolutionary approach is straightforward and does not require the user to use geometric heuristics or specify arm configuration indicators. Only the forward kinematic equations of the robot are used which are simple to develop.

(ii) The evolutionary approach evaluates the multiple robot configurations correctly even when some of the configurations are not possible due to the limits on the joint variables. In the geometric approach all the configurations have to be calculated and some of these configurations may have to be discarded by the user after observing the limits on the joint variables.

(iii) Further, the evolutionary approach evaluates the joint variables within the joint variable limits correctly in all instances for a particular link coordinate frame assignment scheme. In the geometric approach certain joint variable values may seem to lie outside the actual joint variable limits and have
to re-interpreted considering the fact that a joint variable value $+\beta$ and $-(2n-\beta)$ are equivalent.

2. Two niching strategies for the tournament selection operator, along with the SBX search operator and a parameter-based mutation operator, are compared on the basis of their ability to provide the multiple solutions to the problem of robot inverse kinematics. On the basis of success rates in the simulation experiments carried out on two industrial robots, it is observed that the evolutionary approach based on a real-coded GA and using the niching strategy proposed by Deb and Agrawal (1999) performs better than the approach using the niching strategy suggested by Oei et al. (1991).

3. The success rate of the evolutionary approach based on the real-coded genetic algorithm is 94\% for the inverse kinematics problem of the SCARA robot which is an acceptable value. However, the success rate of the approach decreases with the increase in the number of degrees of freedom of the robot. The success rate has a value of 64\% for the inverse kinematics problem of the PUMA robot. The success rate for the SCARA robot corresponds to the percentage of runs for which the evolutionary approach is able to evaluate the inverse kinematics solutions within a wrist positioning error of 1\text{mm}. The corresponding value of wrist positioning error for the PUMA robot is 2\text{mm}.

4. The models of both the SCARA and the PUMA robot can be generated using the modelling primitives developed in MATLAB for link construction. The generated models require as input the values of the joint variables.
All the possible multiple configurations of the robots, evaluated using the evolutionary approach, can therefore be easily displayed for the purpose of visualization.