

Figure 9.1. The PUMA 560 kinematic structure.

with the geometric and algebraic approaches taken by the previous authors.

Without computing the forward kinematics, we will illustrate how Eqs. (4.9)-(4.13) may be easily obtained. For quick reference in the following discussion, we write the equations immediately

$$t_z = s_{23} c_4 s_5 + c_{23} c_5 \quad (9.1)$$

$$p_z = a_2 s_2 + a_3 s_{23} - d_4 c_{23} \quad (9.2)$$

$$p \cdot t = (a_3 + a_2 c_3) c_4 s_5 + d_3 s_4 s_5 - c_5 (d_4 + a_2 s_3) \quad (9.3)$$

$$(p^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2)/(2a_2) = d_4 s_3 + a_3 c_3. \quad (9.4)$$

To illustrate the simplification obtained by the frame assignment described earlier and inner-product invariance under rotations, we give in detail the development of equation (9.4).

With $l_1 = l_5 = l_6 = 0$ and $l_4 = d_4 z$, Eq. (4.4) yields

$$p = R_1(R_2(R_3 l_4 + l_3) + l_2)$$

or

$$p = R_1 R_2 R_3 [l_4 + R_3^{-1} l_3 + R_3^{-1} R_2^{-1} l_2].$$

By orthogonality, the inner-product $p \cdot p$ has the same value as the inner-product of the term in brackets hence

$$p \cdot p = [l_4 + R_3^{-1} l_3 + R_3^{-1} R_2^{-1} l_2] \cdot [l_4 + R_3^{-1} l_3 + R_3^{-1} R_2^{-1} l_2].$$

The inner-product of each term in the square brackets with itself is the square of the length of that vector. For example,

$$\mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2 \cdot \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2 = \mathbf{l}_2 \cdot \mathbf{l}_2 = a_2^2 + d_2^2 = l_2^2.$$

These inner-product manipulations represent a considerable algebraic simplification that requires little or no mental effort. Further, they provide a methodology and considerable insight into how to find other algebraic reductions.

Some of the cross terms also reduce; for instance,

$$\mathbf{R}_3^{-1}\mathbf{l}_3 \cdot \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2 = \mathbf{l}_3 \cdot \mathbf{R}_2^{-1}\mathbf{l}_2.$$

Complete expansion of Eq. (9.4) and application of the reduction techniques just discussed lead to

$$(p^2 - l_4^2 - l_3^2 - l_2^2)/2 = l_4 \cdot [\mathbf{R}_3^{-1}\mathbf{l}_3 + \mathbf{R}_3^{-1}\mathbf{R}_2^{-1}\mathbf{l}_2] + l_3 \cdot \mathbf{R}_2^{-1}\mathbf{l}_2.$$

For this manipulator, vectors

$$\mathbf{l}_4 = [0, 0, d_4]^T = d_4 \mathbf{z},$$

$$\mathbf{R}_3^{-1}\mathbf{l}_3 = [a_3, d_3, 0]^T, \text{ and}$$

$$\mathbf{R}_2^{-1}\mathbf{l}_2 = [a_2, 0, 0]^T = a_2 \mathbf{x}$$

allow us to simplify the last equation to

$$(p^2 - l_4^2 - l_3^2 - l_2^2)/2 = d_4 \mathbf{z} \cdot [\mathbf{R}_3^{-1}\mathbf{l}_3 + a_2 \mathbf{R}_3^{-1}\mathbf{x}] + a_2 \mathbf{l}_3 \cdot \mathbf{x},$$

(e.g., $\mathbf{l}_4 \cdot \mathbf{R}_3^{-1}\mathbf{l}_3$ is obviously 0, which eliminates θ_4 from this equation).

Without any matrix multiplication required, we obtain the fully simplified relation involving θ_3 only:

$$(p^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2)/2 = a_2 (d_4 s_3 + a_3 c_3).$$

The last equation yields at most two solutions for θ_3 . After applying trigonometric identities for angle sums to (9.2), we get

$$p_z = a_2 s_2 + a_3 (s_2 c_3 + s_3 c_2) - d_4 (c_2 c_3 - s_2 s_3),$$

and grouping terms, we obtain

$$(a_2 + a_3 c_3 + d_4 s_3) s_2 + (a_3 s_3 - d_4 c_3) c_2 = p_z.$$

With each value of θ_3 , 2 values for θ_2 can be obtained from this last equation.

With θ_2 and θ_3 known, (9.1) and (9.3) become functions of θ_4 and θ_5 only. Although this system of 2 equations in two unknowns can theoretically be solved, its solution is not obvious. A simpler solution exists if Eqs. (4.13) to (4.16) are considered,

$$p \cdot x = R_1(R_2(R_3 l_4 + l_3) + l_2) \cdot x = p_x, \quad (9.5)$$

$$p \cdot y = R_1(R_2(R_3 l_4 + l_3) + l_2) \cdot y = p_y, \quad (9.6)$$

or

$$(d_4 s_{23} + a_3 c_{23} + a_2 c_2) c_1 + d_3 s_1 = p_x$$

$$-d_3 c_1 + (d_4 s_{23} + a_3 c_{23} + a_2 c_2) s_1 = p_y.$$

The last two Eqs. form a linear system in s_1 and c_1 and provide a unique value for θ_1 .

Eqs. (4.13) and (4.14) along with (9.1) provide a way to solve for θ_4 and θ_5 . By solving for c_5 in (9.1) and substituting in

$$t_x = c_1 c_{23} c_4 s_5 + s_1 s_4 s_5 - c_1 s_{23} c_5 \quad (9.7)$$

and

$$t_y = s_1 c_{23} c_4 s_5 - c_1 s_4 s_5 - s_1 s_{23} c_5, \quad (9.8)$$

we obtain

$$c_1 c_4 s_5 + s_1 c_{23} s_4 s_5 = t_x c_{23} + t_z c_1 s_{23} \quad (9.9)$$

$$s_1 c_4 s_5 - c_1 c_{23} s_4 s_5 = t_y c_{23} + t_z s_1 s_{23}. \quad (9.10)$$

This linear system can be solved for the products $c_4 s_5$ and $s_4 s_5$ uniquely. When s_5 is not 0, two solutions for θ_4 are then obtained by

$$\theta_4 = \text{Atan2}(s_4 s_5, c_4 s_5) \text{ or } \theta_4 = \text{Atan2}(-s_4 s_5, -c_4 s_5).$$

When $s_5=0$, joint axes z_3 and z_5 are aligned and the manipulator loses one DOF. Only the sum $\theta_4+\theta_6$ can be found by use of Eqs. (4.3) and (4.4).

With θ_4 known, the t_x and t_y equations above constitute a linear system of equations which yields a unique solution for θ_5 . The last joint variable θ_6 can then be obtained from 2 more equations from (4.3) such as the n_z and b_z equations.

This procedure will yield 8 solutions which must then be checked for joint variable range limitations. We end the discussion of the PUMA example with the observation that the forward kinematics were never determined to obtain the inverse kinematic solution!

Example 2: The GP66.

Consider the manipulator geometry with kinematic parameters given in Table 9-2. This robot arm is an existing industrial manipulator that belongs to the 11-011 class of orthogonal arms and does not allow closed-form solutions. Although, Table 8-1 specifies that the most complex arm structure within the orthogonal manipulators class 11-011 can be solved with a two-dimensional iterative technique, but the GP66 (Figure 9.2) has two consecutive pairs of intersecting axes and a prismatic joint and it can be solved with a one-dimensional iterative technique. Another reason for discussing this arm here is to show that the techniques developed in this text apply to manipulators with prismatic joints as well.

An iterative method that exactly computes the position, but approximates the orientation, was proposed for this type of geometry by Lumelsky (1984). The technique presented here differs in that it solves for both the orientation and the position with the same precision and it is applicable to a larger variety of manipulators.

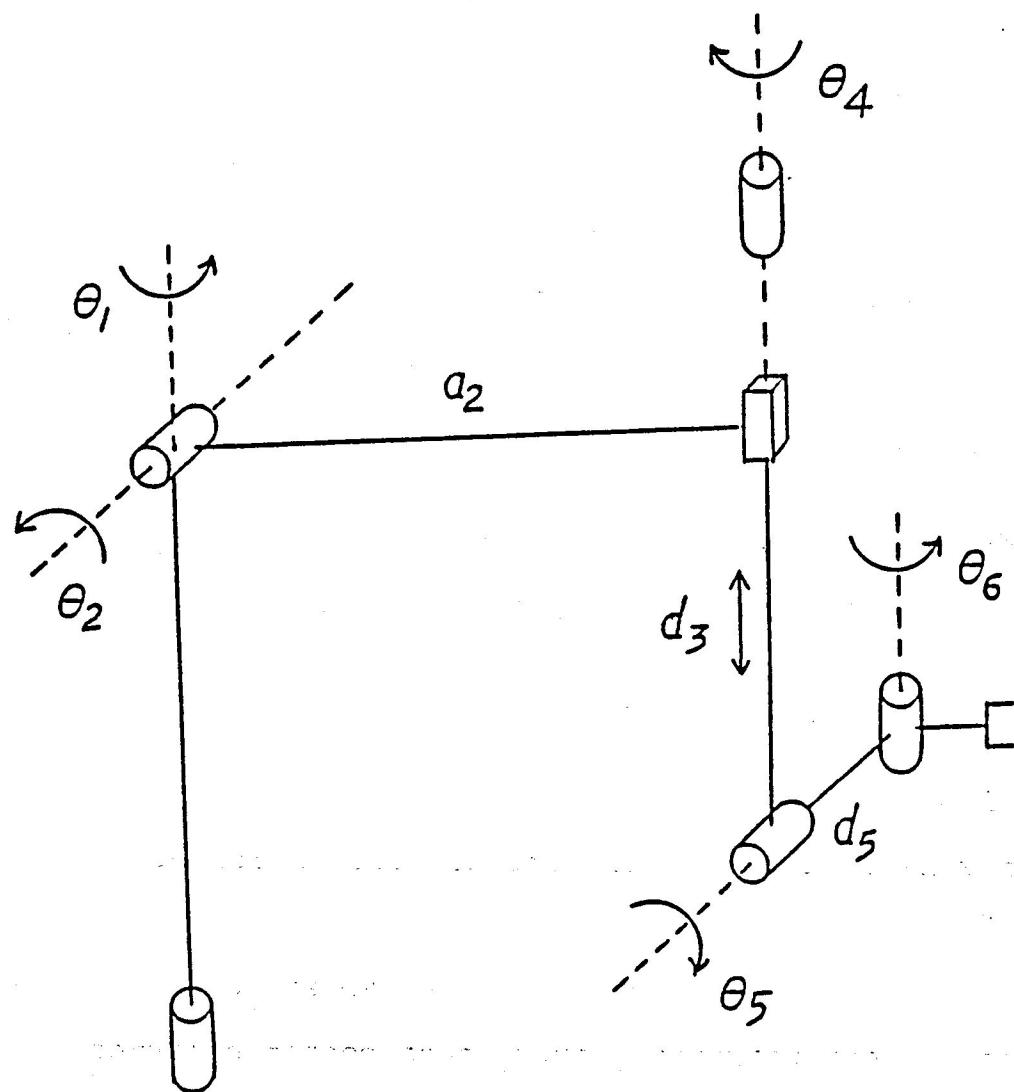


Figure 9.2. The GP66 kinematic structure.

Table 9-2. GP66 manipulator kinematic parameters.

joint	d	θ	a	α
1	0	θ_1	0	$\pi/2$
2	0	θ_2	a_2	$\pi/2$
3	d_3	0	0	0
4	0	θ_4	0	$\pi/2$
5	d_5	θ_5	0	$\pi/2$
6	0	θ_6	0	0

Joint 3 is prismatic.

We will implement an iterative technique to find the zeros of a real valued function of θ_1 . First we show that knowledge of θ_1 is sufficient to determine a solution set. Assuming a guess of θ_1 , the corresponding x and z components of ${}^1p = R_1^{-1}p$ and ${}^1t = R_1^{-1}t$ can be computed,

$${}^1p_x = p_x c_1 + p_y s_1 \quad (9.11)$$

$${}^1p_z = p_x s_1 - p_y c_1 \quad (9.12)$$

$${}^1t_x = t_x c_1 + t_y s_1 \quad (9.13)$$

$${}^1t_z = t_x s_1 - t_y c_1. \quad (9.14)$$

For this robot $l_1 = l_4 = l_6 = 0$, $R_2^{-1}l_2 = a_2x$, $l_3 = d_3z$, $l_5 = d_5z$ and $R_3 = I$. With these values, (4.4) yields

$$p = R_1 R_2 (d_5 R_4 z + d_3 z + a_2 x)$$

or after multiplication by R_1^{-1}

$${}^1\mathbf{p} = \mathbf{R}_2(d_5 \mathbf{R}_4 \mathbf{z} + d_3 \mathbf{z} + a_2 \mathbf{x}). \quad (9.15)$$

Vector ${}^1\mathbf{t}$ is given by

$${}^1\mathbf{t} = \mathbf{R}_1^{-1} \mathbf{t} = \mathbf{R}_2 \mathbf{R}_4 \mathbf{R}_5 \mathbf{z}. \quad (9.16)$$

A reduced system of equations can be obtained from the last two equations by considering the expressions of ${}^1\mathbf{p}_z$, ${}^1\mathbf{t}_z$, ${}^1\mathbf{t} \cdot {}^1\mathbf{p}$, and ${}^1\mathbf{p} \cdot {}^1\mathbf{p}$.

Computing ${}^1\mathbf{t}_z$. From Eq. (9.16) and using Eqs. (4.5) and (4.6) as necessary yields

$${}^1\mathbf{t}_z = {}^1\mathbf{t} \cdot \mathbf{z} = \mathbf{R}_5 \mathbf{z} \cdot \mathbf{R}_4^{-1} \mathbf{y}.$$

Since $\mathbf{R}_5 \mathbf{z} = [S_5, -C_5, 0]^T$ and $\mathbf{R}_4^{-1} \mathbf{y} = [S_4, 0, -C_4]^T$, the preceding equation becomes

$${}^1\mathbf{t}_z = S_4 S_5. \quad (9.17)$$

Computing ${}^1\mathbf{p}_z$. Since ${}^1\mathbf{p}_z = {}^1\mathbf{p} \cdot \mathbf{z}$, from Eq. (9.15), properties (4.5) and (4.6) and $\mathbf{R}_2^{-1} \mathbf{z} = \mathbf{y}$, we obtain

$${}^1\mathbf{p}_z = (d_5 \mathbf{R}_4 \mathbf{z} + d_3 \mathbf{z} + a_2 \mathbf{x}) \cdot \mathbf{y}$$

which is easily seen to produce

$${}^1\mathbf{p}_z = -d_5 C_4. \quad (9.18)$$

Computing ${}^1\mathbf{t} \cdot {}^1\mathbf{p}$. Eqs. (9.15), (9.16) and use of Eqs. (4.5) and (4.6) yield

$${}^1\mathbf{t} \cdot {}^1\mathbf{p} = \mathbf{t} \cdot \mathbf{p} = \mathbf{R}_5 \mathbf{z} \cdot (d_5 \mathbf{z} + d_3 \mathbf{R}_4^{-1} \mathbf{z} + a_2 \mathbf{R}_4^{-1} \mathbf{x}).$$

With $R_4^{-1}z=y$ and $R_5z \cdot d_5z=0$, this equations reduces to

$$t \cdot p = -d_3 c_5 + a_2 c_4 s_5. \quad (9.19)$$

Computing p.p. The inner product directly produces

$${}^1p \cdot {}^1p = p \cdot p = d_5^2 + d_3^2 + a_2^2 + 2 a_2 d_5 s_4 \quad (9.20)$$

without any matrix operations.

Equation (9.19) can be used to define a real function of θ_1 , i.e.,

$$f(\theta_1) = -d_3 c_5 + a_2 c_4 s_5 - t \cdot p. \quad (9.21)$$

Values of θ_1 that yield a solution to the Inverse kinematics of this manipulator must be zeros of function f. Eqs. (9.17), (9.18), and (9.20) provide a way to compute f given θ_1 .

With θ_1 known, 1p_x , 1p_z , 1t_x and 1t_z are fully determined. Equation (9.18) then yields

$$c_4 = -{}^1p_z/d_5 \quad (9.22)$$

and

$$s_4 = u_4 \text{ Trig}(c_4) \quad (9.23)$$

where $u_4 = 1$ or -1 expresses a sign ambiguity and the function Trig is defined by $\text{Trig}(x)=(1-x^2)^{1/2}$.

The prismatic variable d_3 can then be found from Eq. (9.20),

$$d_3 = (p \cdot p - a_2^2 - d_5^2 - 2 a_2 d_5 s_4)^{1/2}, \quad (9.24)$$

and from (9.17), the value of s_5 can be computed, if s_4 is not zero,

$$s_5 = {}^1t_z/s_4 \quad (9.25)$$

and

$$c_5 = u_5 \operatorname{Trig}(s_5) \quad (9.26)$$

where $u_5 = 1$ or -1 is another sign ambiguity. This additional sign ambiguity can be avoided if more equations involving θ_2 are considered. Indeed, Eq. (9.15) allows us to derive expressions for 1p_x and 1p_y as

$${}^1p_x = R_2(d_5 R_4 z + d_3 z + a_2 x) \cdot x$$

and

$${}^1p_y = R_2(d_5 R_4 z + d_3 z + a_2 x) \cdot y.$$

These last equalities yield a system of two equations that can readily be solved for s_2 and c_2 ;

$$s_2 = (d_3 {}^1p_x + k_0 p_z)/(d_3^2 + k_0^2), \quad (9.27)$$

$$c_2 = (k_0 {}^1p_x - d_3 p_z)/(d_3^2 + k_0^2), \quad (9.28)$$

where $k_0 = (a_2 + d_5 s_4)$. The value of c_5 can then be obtained from either the expression for 1t_y ,

$${}^1t_y = R_2 R_3 R_4 R_5 z \cdot y$$

which, after using properties (4.5) and (4.6), gives

$$c_5 = (tz - s_2 c_4 s_5) / c_2 \quad (9.29)$$

or from

$${}^1t_x = R_2 R_3 R_4 R_5 z \cdot x,$$

which yields

$$c_5 = (c_2 c_4 s_5 - {}^1t_x) / s_2. \quad (9.30)$$

With the computed values of d_3 , c_4 , c_5 and s_5 , the value of $f(\theta_1)$ is fully determined.

The derivative of f can also be evaluated. By differentiating (9.18) with respect to θ_1 , $dc_4/d\theta_1$ can be obtained;

$$dc_4/d\theta_1 = -{}^1p_x/d_5, \quad (9.31)$$

where we substituted $d({}^1p_z)/d\theta_1 = {}^1p_x$.

The value of $ds_4/d\theta_1$ can be obtained by differentiating the Pythagorean identity,

$$d[s_4^2 + c_4^2]/d\theta_1 = d(1)/d\theta_1 = 0$$

and solving for $ds_4/d\theta_1$,

$$ds_4/d\theta_1 = -c_4 (dc_4/d\theta_1)/s_4 = c_4 {}^1p_x/(s_4 d_5). \quad (9.32)$$

Differentiating (9.20) yields

$$d(d_3)/d\theta_1 = -a_2 d_5 (ds_4/d\theta_1)/d_3.$$

Substituting from Eq. (9.32) gives

$$dd_3/d\theta_1 = -a_2 d_5 c_4 {}^1p_x/(s_4 d_5 d_3), \quad (9.33)$$

and from (9.17), $ds_5/d\theta_1$ can be obtained,

$$ds_5/d\theta_1 = [{}^1t_x - s_5 (ds_4/d\theta_1)]/s_4. \quad (9.34)$$

Once again from differentiation of the Pythagorean identity,

$$dc_5/d\theta_1 = -s_5 (ds_5/d\theta_1)/c_5 \quad (9.35)$$

and $df/d\theta_1$ is finally obtained by differentiating equation (9.21),

$$\begin{aligned} df/d\theta_1 = a_2 & [c_4 (ds_5/d\theta_1) + s_5 (dc_4/d\theta_1)] \\ & - [d_3 (dc_5/d\theta_1) + c_5 (dd_3/d\theta_1)]. \end{aligned} \quad (9.36)$$

By use of the one-dimensional Newton-Raphson iterative method, a new estimate for θ_1 is given by

$$\theta_{1,\text{new}} = \theta_1 - f(\theta_1)/(df/d\theta_1).$$

Once θ_1 is obtained to the desired accuracy, the remaining joint variables θ_2 , θ_4 , and θ_5 are then computed from the values of their sines and cosines as obtained, along with d_3 from the last iteration. A vector equation in θ_6 can be obtained from (4.3) which gives,

$$R_6 \mathbf{x} = \begin{bmatrix} c_6 \\ s_6 \\ 0 \end{bmatrix} = R_5^{-1} R_4^{-1} R_2^{-1} R_1^{-1} R \mathbf{x}. \quad (9.37)$$

This equation can be solved uniquely for θ_6 .

In practical situations, the derivative of f can be estimated by

$$df/d\theta_1 = [f(\theta_1 + \delta\theta_1) - f(\theta_1)]/\delta\theta_1,$$

where $\delta\theta_1$ is a small increment of θ_1 .

When the value of θ_1 is close enough to a solution, the complexity can be reduced by computing $df/d\theta_1$ numerically at any iteration using the values of θ_1 and $f(\theta_1)$ in the preceding iteration,

$$(df/d\theta_1)^i = (f^{i-1} - f^i)/(\theta_1^{i-1} - \theta_1^i),$$

where the superscript represents the iteration number at which the variable is computed. This saves the computational cost of Eqs. (9.31) through (9.36) and avoids the problem of special cases that occur when division by a number close to zero is needed in any of those equations.

The procedure just described was programmed to compute the joint variables for 10 equidistant points on a linear trajectory with constant orientation that will move the end-effector from the initial pose

$$\mathbf{P} = \begin{bmatrix} \sqrt{2}/2 & 0 & \sqrt{2}/2 & 1 \\ \sqrt{2}/2 & 0 & -\sqrt{2}/2 & -1/2 \\ 0 & 1 & 0 & -1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

to the position $[1/2, 1/2, 1/10]^T$. Table 9-3 shows the output of the program.

Table 9-3. GP66 trajectory points (in joint coordinates)

θ_1	θ_2	d_3	θ_4	θ_5	θ_6
-19.072	54.427	1.192	-140.114	-137.013	-121.439
-15.319	54.980	1.090	-135.196	-135.357	-125.247
-11.061	55.823	0.992	-129.853	-133.343	-129.428
-6.234	57.063	0.901	-124.100	-130.873	-134.024
-0.773	58.831	0.820	-118.000	-127.817	-139.068
5.374	61.276	0.751	-111.700	-124.006	-144.568
12.239	64.532	0.697	-105.467	-119.245	-150.474
19.805	68.657	0.662	-99.716	-113.360	-156.644
27.968	73.551	0.649	-94.958	-106.315	-162.840
36.488	78.908	0.660	-91.649	-98.352	-168.788
45.000	84.279	0.694	-90.000	-90.000	-174.278

All angles are in degrees. $a_2 = 0.36$, $d_5 = 0.19$

The maximum number of iterations needed per point was six. The guess for each point was the value of θ_1 at the preceding point. The experiment was started with a guess of -20° . Convergence at every point was obtained when 6 or more points were taken along the trajectory. Although Table 9-3 gives the joint variables to only 3 decimal places, they were computed with a precision of 10^{-5} . The program was written in C and run on an AT&T 3B2/310 desktop computer.

The kinematic inversion for the 11 points took 0.32 seconds which gives an average time per kinematic inversion of 29.1 milliseconds, clearly suitable for real-time high precision inverse kinematics.

Example 3: The OM25 Manipulator

In this example we present an orthogonal manipulator that can achieve a given end-effector pose in sixteen distinct configurations. Besides this interesting point, we discuss this manipulator to show the power of the simple iterative techniques developed in this dissertation not only as fast inverse kinematic methods, but also as interactive inverse kinematic analysis tools capable of finding all the solutions for certain manipulators. The OM25 manipulator is illustrated in Figure 9.3 and its DH-parameters are listed in Table 9-4.

Table 9-4. DH-parameters of OM25 manipulator.

Joint	d	θ	a	α
1	0	θ_1	a_1	$\pi/2$
2	0	θ_2	a_2	0
3	d_3	θ_3	0	$\pi/2$
4	0	θ_4	a_4	0
5	0	θ_5	0	$\pi/2$
6	0	θ_6	0	0

OM25 MANIPULATOR KINEMATIC STRUCTURE

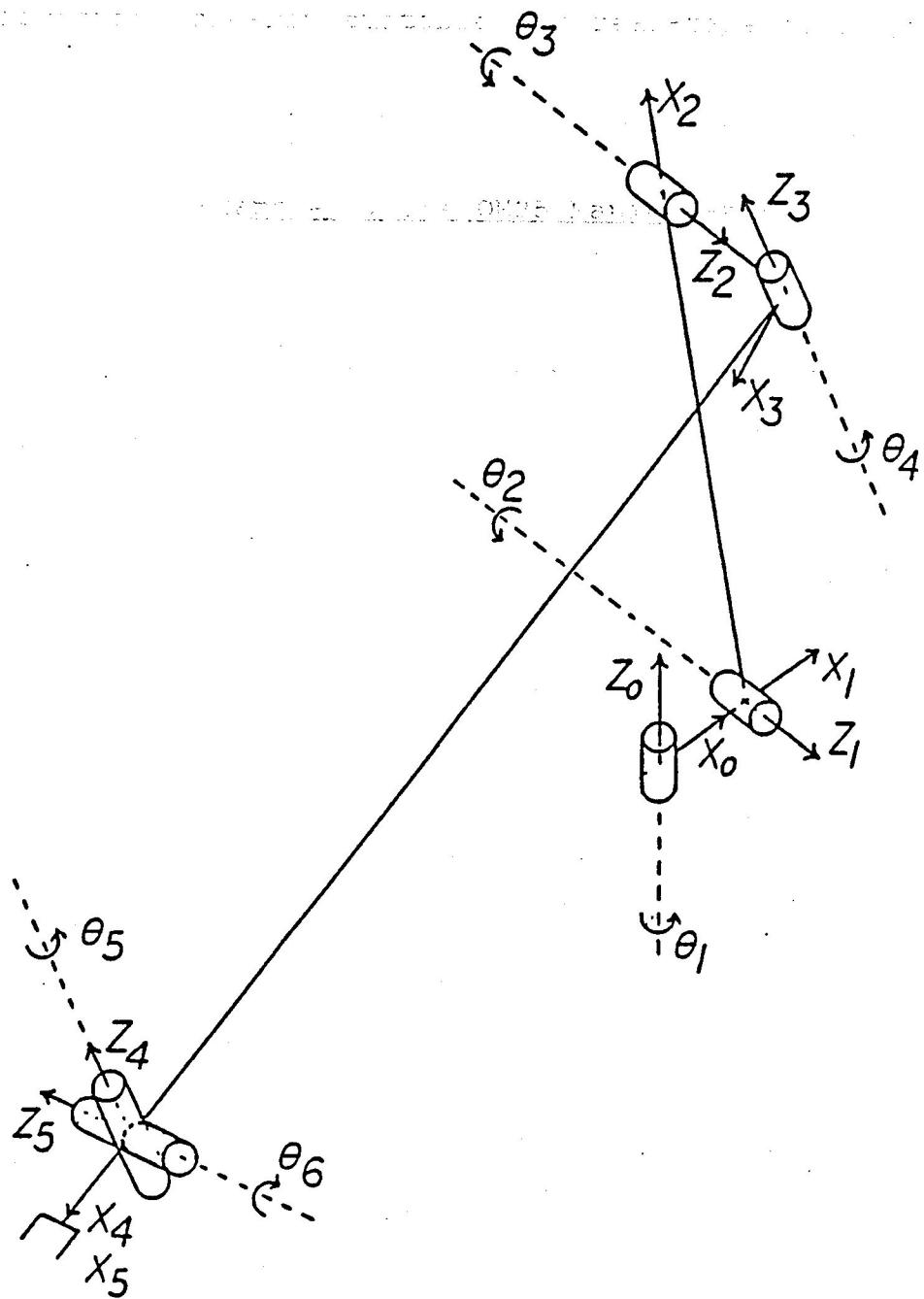


Figure 9.3. The OM25 manipulator kinematic structure.

This robot combines the following special structures identified in Chapter 5:

Case 2 is satisfied by the four-DOF segment z_1 to z_4 .

Case 5 is satisfied by the segment z_0 to z_3 and by the segment z_3 to z_5 .

Case 7 is satisfied by the segment z_1 to z_4 .

Case 9 is satisfied by the segment z_2 to z_5 .

The structure of this arm, although simple enough to allow the use of a one-dimensional iterative technique (as seen in Chapter 7), still does not allow closed form solutions.

Once again we will implement an iterative technique based on finding the zeros of a real valued function of θ_1 .

With θ_1 known (initial estimate), the vectors

$${}^1t = R_1^{-1}t = \begin{bmatrix} c_1 t_x + s_1 t_y \\ t_z \\ s_1 t_x - c_1 t_y \end{bmatrix} \quad (9.39)$$

and

$${}^1p = R_1^{-1}p = \begin{bmatrix} c_1 p_x + s_1 p_y \\ p_z \\ s_1 p_x - c_1 p_y \end{bmatrix} \quad (9.40)$$

are fully determined. Expressions for 1t_z , 1p_z , ${}^1t \cdot {}^1p$, and ${}^1p \cdot {}^1p = p \cdot p$ are given in terms of the remaining joint angles by

$$-c_{45} = {}^1t_z \quad (9.41)$$

$$a_4 S_4 + d_3 = {}^1 p_z \quad (9.42)$$

values of a_4 and d_3 are obtained.

$$a_4 S_5 + S_{45} (a_2 C_3 + a_1 C_{23}) = t \cdot p \quad (9.43)$$

$$\begin{aligned} a_4 C_4 (a_2 C_3 + a_1 C_{23}) + d_3 a_4 S_4 + a_1 a_2 C_2 = \\ [p \cdot p - a_1^2 - a_2^2 - a_4^2 - d_3^2]/2. \end{aligned} \quad (9.44)$$

Expressions for ${}^1 t_x$, ${}^1 t_y$, ${}^1 p_x$, and ${}^1 p_y$ provide the additional equations,

$$C_{23} S_{45} = {}^1 t_x \quad (9.45)$$

$$S_{23} S_{45} = {}^1 t_y \quad (9.46)$$

$$a_4 C_4 C_{23} + a_2 C_2 + a_1 = {}^1 p_x \quad (9.47)$$

$$a_4 C_4 S_{23} + a_2 S_2 = {}^1 p_y \quad (9.48)$$

If the function f is defined from equation (9.43), then for a given value of θ_1 , the corresponding values of S_5 , S_{45} , C_3 and C_{23} must be computed before

$$f(\theta_1) = a_4 S_5 + S_{45} (a_2 C_3 + a_1 C_{23}) - t \cdot p \quad (9.49)$$

can be evaluated.

Computing f

Given θ_1 , the components of ${}^1 t$ and ${}^1 p$ are computed from Eqs (9.39) and (9.40). The value of C_{45} and S_{45} are then given by Eq. (9.41)

$$C_{45} = -({}^1 t_x)$$

and

$$S_{45} = u_{45} \operatorname{Trig}(C_{45})$$

where u_{45} is a sign ambiguity and $\text{Trig}(x) = \sqrt{1-x^2}$.

Equation (9.45) and (9.46) yield

$$c_{23} = {}^1t_x / s_{45} \quad (9.50)$$

$$s_{23} = t_z / s_{45} \quad (9.51)$$

respectively, when s_{45} is not zero. In the special case where s_{45} has an extremely small absolute value, $\theta_{45} = 0$ or π and the current value of θ_1 will allow closed form solving of the remaining joint variables and $f(\theta_1)$ can still be computed. Next, s_4 and c_4 are computed from equation (9.42),

$$s_4 = ({}^1p_z - d_3) / a_4 \quad (9.52)$$

$$c_4 = u_4 \text{ Trig}(s_4), \quad (9.53)$$

where u_2 is another sign ambiguity, and c_2 and s_2 are obtained from (9.47) and (9.48) respectively,

$$c_2 = ({}^1p_x - a_4 c_4 c_{23} - a_1) / a_2 \quad (9.54)$$

$$s_2 = (p_z - a_4 c_4 s_{23}) / a_2. \quad (9.55)$$

Finally the values of θ_3 and θ_5 are computed as

$$\theta_3 = \text{atan2}(s_{23}, c_{23}) - \text{atan2}(s_2, c_2)$$

and

$$\theta_5 = \text{atan2}(s_{45}, c_{45}) - \text{atan2}(s_4, c_4)$$

and $f(\theta_1)$ can be computed. A few points are noteworthy in this derivation. First, since the above procedure computes f from a guess θ_1 and not a value of θ_1 that corresponds to a true solution set, the values of sine and cosine of any angle computed from 2 different equations (such as (9.45) and (9.46) for θ_{23} or (9.47) and (9.48) for θ_2) will in general not satisfy the Pythagorean identity (4.17).

An alternate possibility is to use one of the two equations to solve for one trigonometric function, either sine or cosine, compute the other from equation (4.17) and use the second equation to avoid a sign ambiguity only. For example, instead of using Eq. (9.46), we can compute s_{23} from

$$s_{23} = \text{Sign}(t_z/s_{45}) \text{ Trig}(c_{45}) \quad (9.56)$$

to insure compatibility with identity (4.17).

As discussed in Chapters 6 and 7, the ability to compute the function $f(\theta_1)$ proved sufficient to provide a practical one-dimensional inverse kinematic algorithm for the manipulator of Figure 9.3.

The sign ambiguities u_4 and u_{45} above give four possible combinations that must all be tried in the search for a root of the function f . Once a root of f is found, the values of θ_1 , θ_4 and θ_5 are known along with a value for c_3 from the last iteration.

A one-dimensional Newton-Raphson algorithm was programmed to find the zeros of the function f defined in Eq. (9.49) with $d_3 = 0.2$, $a_1 = 0.3$, $a_2 = 1$, $a_4 = 1.5$. Once a value of θ_1 for which $f(\theta_1)=0$ was found, the corresponding solution set was completed and checked for consistency by verifying that the solution set satisfies the expression for t_y as obtained from Eq. (7.1). This simple test proved effective in filtering out extraneous solutions for this particular experiment. A more involved consistency verification procedure may be required for different manipulators. However, Computing the complete forward kinematics from Eq. (7.1) and verifying all pose elements constitutes a worst case condition.

Inverse Kinematic Solution Search Algorithm

The one-dimensional inverse kinematic method just described was programmed in pascal on a personal microcomputer. A simple search algorithm was then implemented by selecting regularly spaced values of the initial estimate of θ_1 within the interval $[0, 2\pi)$ with the problem pose

$$P = \begin{bmatrix} -0.760117 & -0.641689 & 0.102262 & -1.140175 \\ 0.133333 & 0 & 0.991071 & 0 \\ -0.635959 & 0.766965 & 0.085558 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9.57).$$

This search program was run for each of the four possible combinations of values for the sign ambiguities u_{45} and u_4 ((1,1), (1,-1), (-1,1), and (-1,-1)).

The 16 solution sets found are listed in Table 9-5. This result is of importance because it provides the first tangible proof that a six-DOF manipulator can actually achieve a given pose with 16 different configurations.

Some manipulators can achieve a particular end-effector pose in an infinite number of configurations when two or more joint axes coincide. Such degenerate conditions, where the manipulator loses degrees of freedom, force the manipulator jacobian to become singular. To verify that the manipulator under discussion is non-degenerate at each of the sixteen configurations found, we compute the determinant of the manipulator Jacobian.

The symbolic Jacobian of a 6-DOF manipulator has its simplest expression in frame 3 (mid-frame) (Renaud, 1980a).

For the robot described here, the mid-frame Jacobian 3J , computed using tables given in Doty (1987), is

$${}^3J = \begin{bmatrix} 0.2 C_{23} & S_3 & 0 & 0 & 1.5 S_4 & 0 \\ -0.3-C_2 & 0 & 0 & 0 & -1.5 C_4 & 0 \\ 0.2 S_{23} & -C_3 & 0 & 0 & 0 & -1.5 C_5 \\ S_{23} & 0 & 0 & 0 & 0 & S_{45} \\ 0 & 1 & 1 & 0 & 0 & -C_{45} \\ -C_{23} & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (9.58)$$

Table 9-5. OM25 manipulator configurations for pose P of equation (9.57).

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	d_J
1	0.000	107.458	112.460	-7.662	0.000	0.000	1.310
2	0.000	107.458	-67.540	-172.338	180.000	180.000	1.310
3	88.670	-176.682	-178.394	-63.284	157.829	139.944	-0.800
4	88.670	-176.682	1.606	-116.716	22.171	-40.056	-0.800
5	113.841	4.741	-179.093	-55.954	-63.659	-42.463	-1.256
6	113.841	4.741	0.907	-124.046	-116.341	137.537	-1.256
7	168.703	-104.205	146.556	-16.393	-170.903	98.216	0.803
8	168.703	-104.205	-33.444	-163.607	-9.097	-81.784	0.803
9	180.000	107.458	-147.375	-7.662	-164.675	180.000	0.732
10	180.000	107.458	32.625	-172.338	-15.325	0.000	0.732
11	-120.748	173.066	-178.472	31.328	-146.087	142.605	-0.717
12	-120.748	173.066	1.528	148.672	-33.913	-37.395	-0.717
13	-96.292	-5.766	-179.142	38.477	51.922	-39.631	-1.441
14	-96.292	-5.766	0.858	141.523	128.078	140.369	-1.441
15	-11.768	-105.495	-114.490	1.243	6.408	-79.398	1.318
16	-11.768	-105.495	65.510	178.757	173.592	100.602	1.318

The determinant of the manipulator Jacobian, d_J , is independent of the frame of expression and can be easily obtained from matrix 3J :

$$d_J = 1.5 \{ c_3 s_{45} [s_4 (0.3 + c_2) - 0.2 c_{23} c_4] \\ - s_3 s_{23} c_4 (1.5 c_5 + 0.2 s_{45}) \}. \quad (9.59)$$

The values of d_J , listed in Table 9-5, prove that all sixteen solutions found correspond to non-degenerate configurations of the OM25 robot arm.

Figure 9.4 shows photographs of a computer graphics simulation of the OM25 manipulator in the sixteen configurations listed in Table 9-5. Figure 9.3 is a hand drawing of this manipulator in configuration 1 of Table 9-5 (i.e. corresponding to the first solution set) with all link frames clearly indicated and, to help differentiate between solutions with common values of θ_1 and θ_2 , we have attempted to indicate the direction of axis vectors z_1 , z_3 , z_4 , and z_5 on the photographs. The position and orientation of the end-effector and the base frame (as shown on Fig. 9.3) are the same for all sixteen representations of Figure 9.4.

It is also interesting to note that this large number of solutions can be realized by an orthogonal manipulator with a fairly simple geometry. Finally, this example shows that the techniques developed in this dissertation can be used to implement simple and efficient inverse kinematic analysis tools such as the search algorithm just discussed

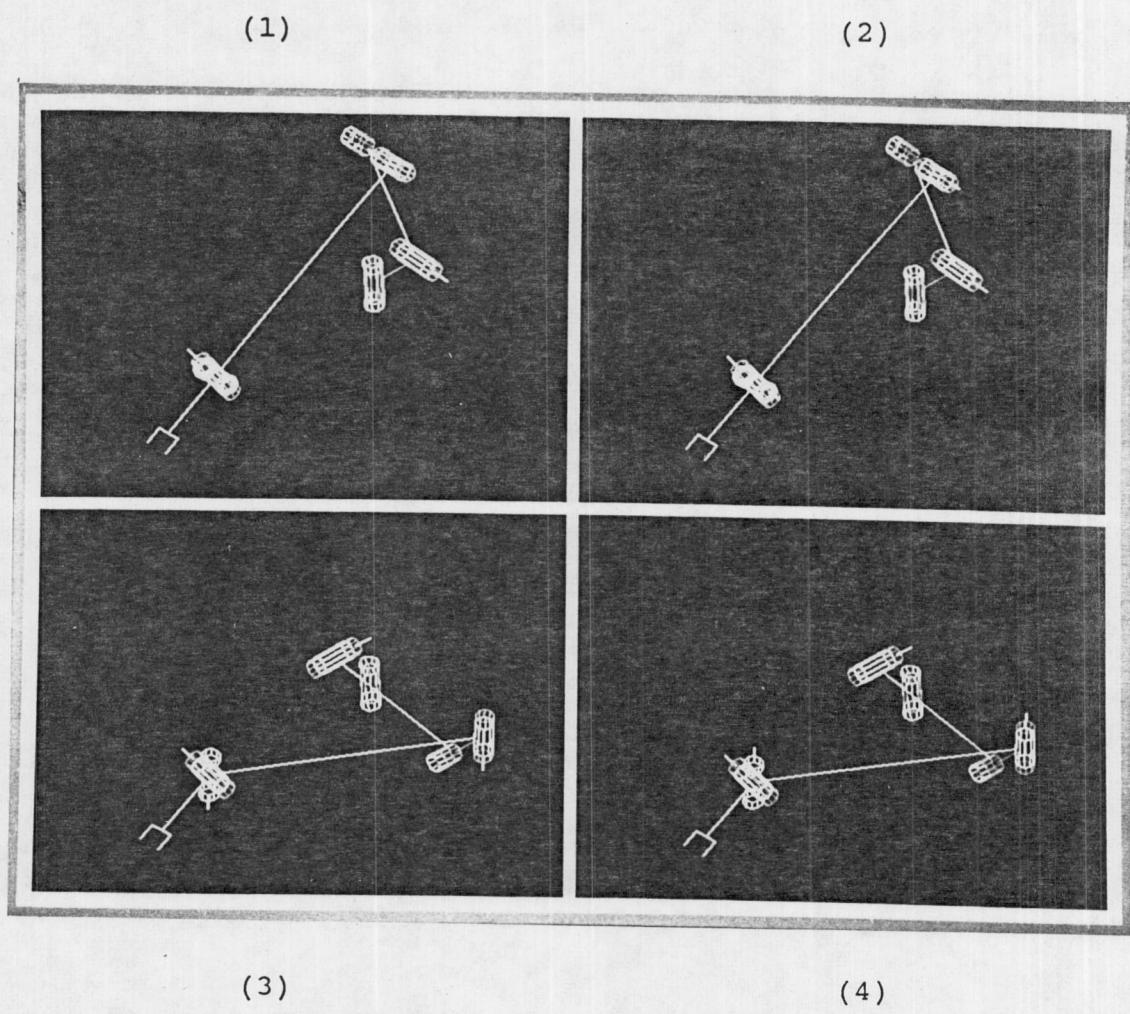
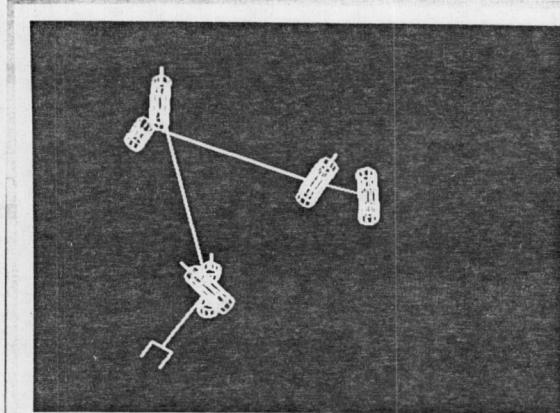
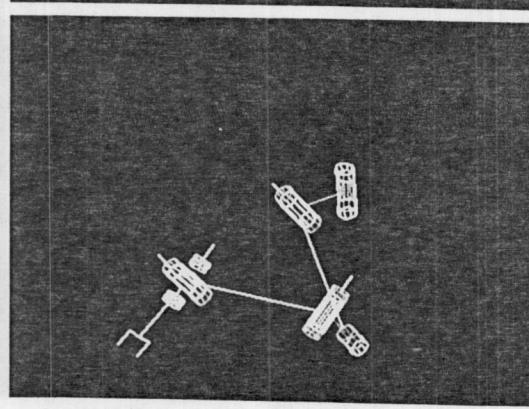
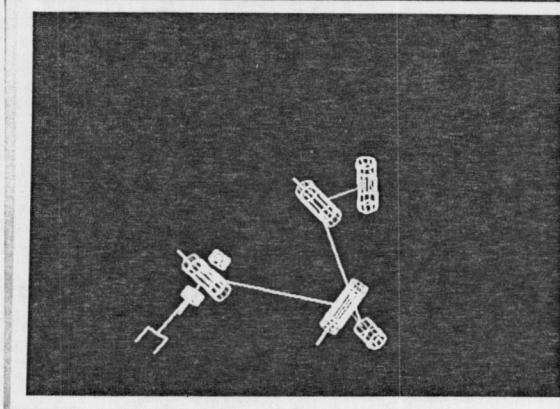
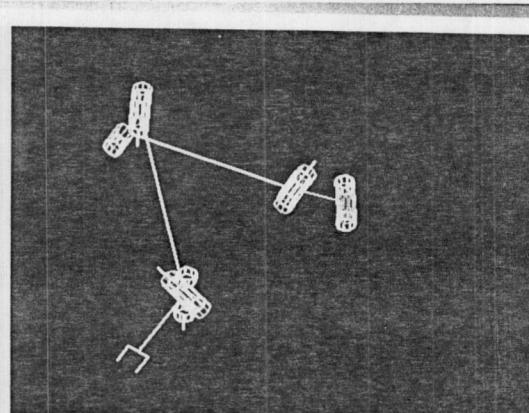


Figure 9.4. Computer simulation of the sixteen configurations of Table 9-5.

(5)



(6)

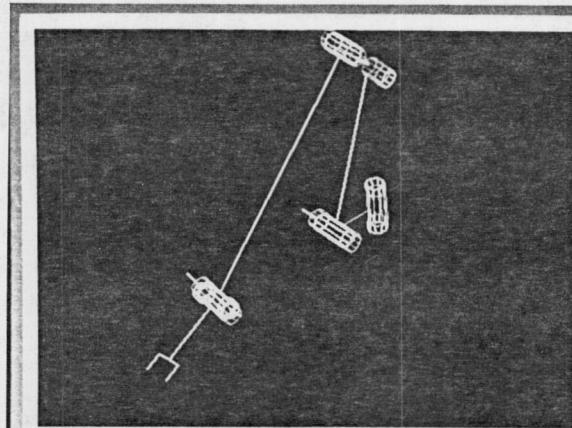


(7)

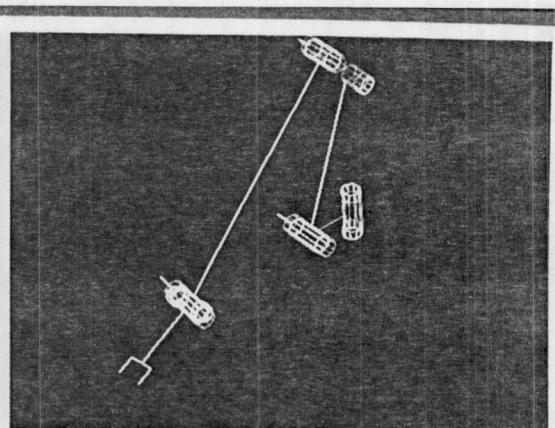
(8)

Figure 9.4--Continued

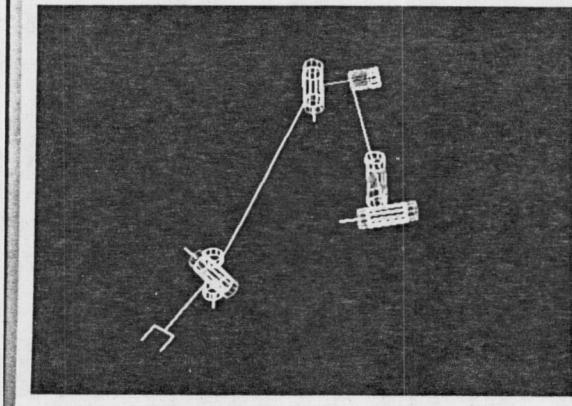
(9)



(10)



(11)



(12)

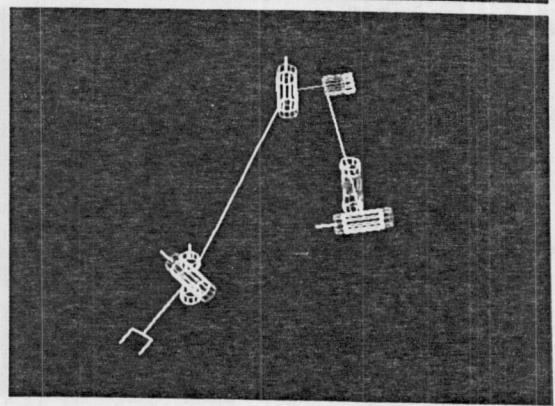
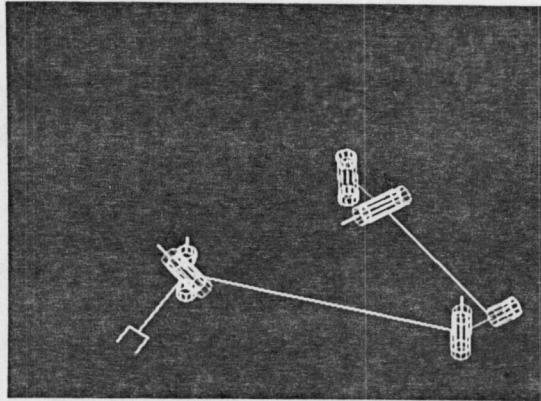
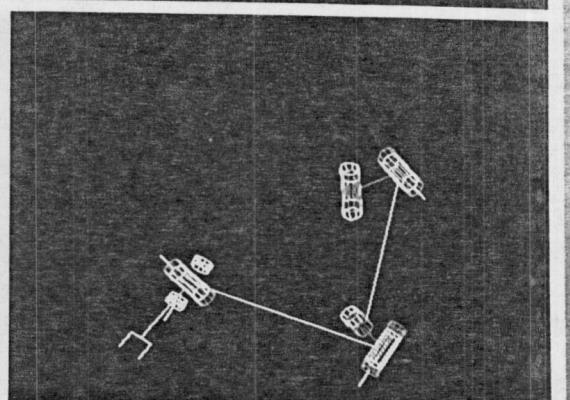
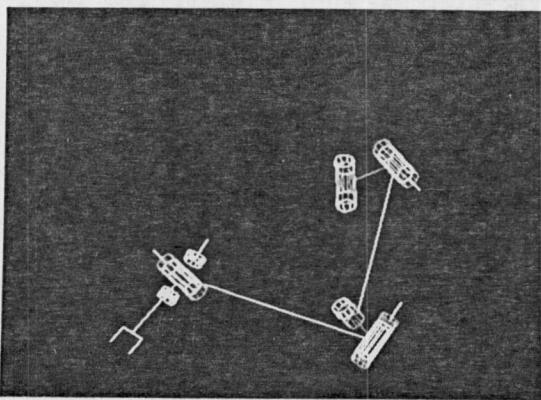
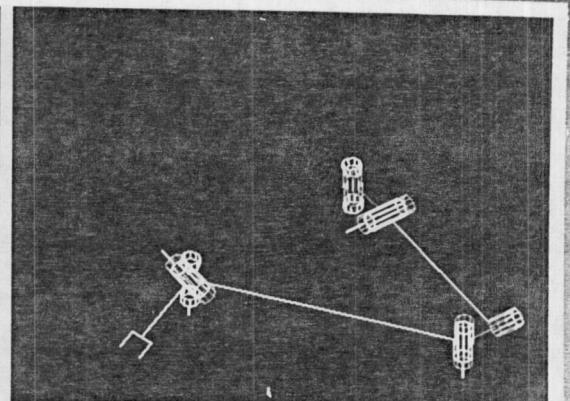


Figure 9.4--Continued

(13)



(14)



(15)

(16)

Figure 9.4--Continued

which has found all the solutions to the inverse kinematics problem of the OM25 manipulator.

Example 4: OM37 Manipulator

The most kinematically complex orthogonal manipulator class is the 11-111 (octal 37) class described by the DH-parameters of Table 9-6 in which no two consecutive joint axes are parallel. In this example we will consider a manipulator with the arbitrarily chosen kinematic parameters

$$d_2 = 0.2, d_3 = 0.1, d_4 = 0.3, d_5 = 0.4$$

$$a_1 = 0.5, a_2 = 1.0, a_3 = 1.2, a_4 = 0.5, a_5 = 0.2,$$

so that there are no two consecutive axes intersecting either.

The general two-dimensional Newton-Raphson method, described in Chapter 7 for a general six-DOF manipulator, was programmed on a personal microcomputer and run with the manipulator just described and the Cartesian end-effector pose

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1/2 \\ 0 & 1 & 0 & -1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9.60)$$

Table 9-6. DH-parameters of OM37 Manipulator.

Joint	d_i	θ_i	a_i	$\alpha_i(\text{rd})$
1	0	θ_1	a_1	$\pi/2$
2	d_2	θ_2	a_2	$\pi/2$
3	d_3	θ_3	a_3	$\pi/2$
4	d_4	θ_4	a_4	$\pi/2$
5	d_5	θ_5	a_5	$\pi/2$
6	0	θ_6	0	0

The solution set

$$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) =$$

$$(-1.28^\circ, 0^\circ, -52.48^\circ, 180^\circ, 51.20^\circ, 180^\circ)$$

was found in 3 iterations with an initial estimate of $(0, 0)$ for (θ_1, θ_2) and an accuracy of 10^{-5} . The same solution set was obtained in 6 iterations from an initial estimate of $(-10^\circ, 10^\circ)$ with the same precision.

A second solution set given by

$$(109.07^\circ, -168.21^\circ, 112.36^\circ, -16.44^\circ, -137.00^\circ, -163.92^\circ)$$

was found with the initial estimate $(115^\circ, -175^\circ)$ in 5 iterations with the same accuracy.

Example 5: A General Geometry 6-DOF Manipulator

The previous examples were all about orthogonal manipulators. Here we examine a non-orthogonal manipulator with no special structure as described by the kinematic parameters of Table 9-7 and the Cartesian end-effector pose

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1/2 \\ 0 & 1 & 0 & -1/2 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (9.61)$$

Table 9-7. DH-parameters of a non-orthogonal manipulator.

Joint	d	θ	a	α
1	0	θ_1	1.0	35°
2	0.2	θ_2	1.2	65°
3	0.3	θ_3	0.75	70°
4	0.5	θ_4	1.0	27°
5	0.4	θ_5	0.68	120°
6	0	θ_6	0	$\dots 0^\circ$

The two-dimensional Newton-Raphson method of Chapter 7 was programmed to find a solution set.

With the initial estimate $(\theta_1, \theta_2) = (80^\circ, -40^\circ)$, the solution set

$$(73.43^\circ, -32.01^\circ, -56.52^\circ, 145.42^\circ, 114.03^\circ, -179.5^\circ)$$

was found in 4 iterations with an accuracy of 10^{-5} . This result is typical of the efficiency of the two-dimensional Newton-Raphson inverse kinematics technique.

CHAPTER 10 CONCLUSION AND FUTURE WORK

This research has addressed the inverse kinematics problem of non-redundant all-revolute robot manipulators. We first showed how a convenient choice of manipulator frames and proper use of inner-product invariance of rotation transformations can be used to easily reduce the inverse kinematic problem to four simple equations independent of two of the joint variables (Doty, 1986). For four-axis robots, the reduced system of equations can always be solved in closed-form and at most two inverse kinematic solution sets can be found. We have determined that, in general, a 4-DOF arm will yield a unique solution and we identified ten special four-DOF structures for which two solutions can be found.

The same ten 4-DOF special structures, when part of a five-axis robot manipulator are sufficient to insure closed-form solutions for the 5-DOF structure. Otherwise, we have shown that the inverse kinematic problem reduces to finding the zeros of a real-valued function of one joint variable. The most kinematically complex 5-DOF arm can therefore be solved by a fast one-dimensional iterative technique such as the regular Newton-Raphson or secant method.

In the analysis of six-degree-of-freedom manipulators, we identified three major classes of 6-DOF arms according to their kinematic complexity. First, the class of closed-form arms in which we find all 6-DOF open kinematic chains with three adjacent joint axes intersecting at a common point (Pieper, 1968) or with three parallel adjacent joint axes. We discovered that the 6-DOF arms with three pairs of two parallel axes and those with two intersecting axes following or preceding two pairs of parallel joint axes had closed-form solutions as well.

The next class contains all 6-DOF manipulators that do not allow closed-form solutions but are such that knowledge of one joint variable is sufficient to obtain a complete solution set in closed-form. We determined that all 6-DOF arms that include one of the ten special 4-DOF structures discussed earlier were in this second class. The inverse kinematics problem for these manipulators reduces to finding the zeros of a real valued-function. A one-dimensional Newton-Raphson or other iterative method can be used to solve the inverse kinematics problem for these robots.

The third class of six-degree-of-freedom robots contains all the manipulators that do not fall into the two preceding classes. For the most kinematically complex six-DOF robot manipulators, the inverse kinematics problem reduces to solving a nonlinear system of only two equations in only two of the joint variables. Therefore, the robots

in this third class can still be solved using a fast two-dimensional iterative technique. This two-dimensional method still constitutes a large reduction in computational complexity with respect to the usual six-dimensional iterative techniques found in the literature. The division of the set of all 6-DOF manipulators into the three classes just discussed is illustrated in Figure 10.1.

In Chapter 8, we provide a formal definition of orthogonal manipulators (Doty 1986). This type of manipulator is important since it includes almost all existing industrial robot manipulators. We show that a structural partition of all orthogonal manipulators into twenty four classes is possible but orthogonal manipulators can also be classified in terms of kinematic complexity. Out of the twenty four partitions found, twelve can always be solved in closed form, three classes are such that their most kinematically complex elements can be solved with a one-dimensional iterative method and the remaining nine classes of orthogonal manipulators have elements for which a two-dimensional iterative technique may be required.

In Chapter 9, we analyze the inverse kinematics of five manipulators. The PUMA 560 inverse kinematics is discussed to illustrate the utility of the new inverse kinematic approach even for arms that allow closed-form solutions. The GP66, another existing industrial robot, that does not allow closed form solutions is analyzed for two major

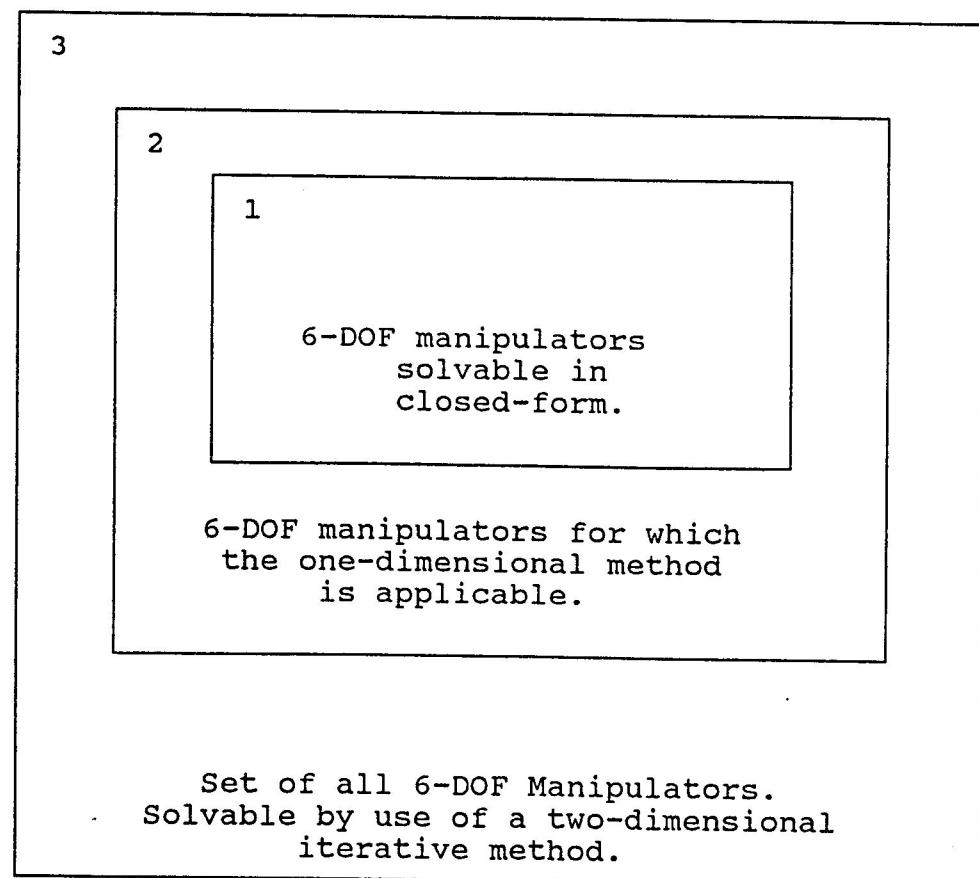


Figure 10.1. Subdivision of 6-DOF manipulators in terms of inverse kinematic techniques

reasons. First, to show that the one-dimensional Newton-Raphson algorithm can perform kinematic inversions in real-time, 29.1 milliseconds per kinematic inversion on an AT&T 3B2/310 desktop computer (this figure drops to 18 milliseconds on an AT&T 3B15 computer). The second reason is the presence of a prismatic joint. Although this dissertation was only concerned with all-revolute manipulators, the GP66 robot example shows that the techniques developed herein are applicable to manipulators with prismatic joints as well.

In example 3, we discussed an orthogonal manipulator of simple geometry, yet not simple enough to allow closed-form solutions. The OM25 robot illustrates the use of the one-dimensional technique as an off-line analysis tool. By interactively varying the robot parameters and the end-effector pose parameters in search of a maximum number of inverse kinematic solutions, the manipulator and the end-effector pose of Chapter 9, example 3 were discovered. If the work of Lee and Liang (in press) puts an upper bound of sixteen on the number of possible inverse kinematic solutions of six-DOF robots, the OM25 manipulator with the sixteen solutions found establishes 16 as the least upper bound on the number of solutions to the inverse kinematics problem.

The iterative techniques described in this dissertation have several advantages over other existing numerical

techniques. They do not require the computation of the manipulator Jacobian or its inverse nor do they require computation of the forward kinematics at any time before convergence is achieved. When the initial estimate of the variable is reasonably close to a solution (typically within 5 degrees), convergence is achieved within six iterations even with accuracies better than 10^{-5} which makes real-time high precision inverse kinematics a reality. Finally, the algorithms simplicity is such that they can be programmed and executed very rapidly even on a personal microcomputer for any manipulator with at most six degrees of freedom. This programming simplicity makes the one- and two-dimensional iterative techniques suitable for use as off-line interactive inverse kinematic tools as well.

The disadvantages of the inverse kinematics methods developed here are those inherent to iterative algorithms. No theoretical guarantee of convergence can be given and the number of iterations required for convergence depends highly on the initial estimate of the variables. A disadvantage over the homotopy map method (Tsai and Morgan 1984) is that convergence to only one solution is possible, again a common disadvantage of iterative techniques. Searching for more than one solution requires trying various estimates of the variables.

One problem that the methods developed in this text share with Tsai and Morgan's homotopy map method (1984) is

that the algorithm may converge to values that do not correspond to true inverse kinematic solution sets. This requires the addition of a post-convergence method for checking the consistency of the solutions to filter-out extraneous ones. Although this problem has not been bothersome in practice, it is one that must be addressed. A trade-off between removal of extraneous solutions and computation cost exists. One desirable future improvement on the one- and two-dimensional techniques will be the development of fast techniques for the elimination of extraneous solutions.

Another future research area is in the establishment of necessary structural conditions of 6-DOF manipulators for the applicability of the one-dimensional technique or the existence of closed-form solutions. The structural conditions established in Chapters 5 and 7 are sufficient but not necessary. There may be other such geometry conditions to be discovered.

APPENDIX SOLVING FOR θ_2

In the inverse kinematic solution for four-degree-of-freedom robot manipulators described in Chapter 4, after solving for θ_1 and θ_3 , we said that the angle θ_2 can be obtained by solving one of the two linear systems of equations in S_2 and C_2 given by Eqs. (5.23) and (5.24) or Eqs. (5.25) and (5.26). In this Appendix, we show that θ_2 can always be computed from one of those two systems when the 4-DOF arm is not in a degenerate configuration.

The absolute value of the determinant of the system of Eqs. (5.23) and (5.24), D_1 is given by

$$D_1 = (\sigma_2 d_3 - \tau_2 a_3 s_3)^2 + (a_2 + a_3 c_3)^2 \quad (A1)$$

and, in absolute value, the determinant of the system of Eqs. (5.24) and (5.25) yields

$$D_2 = (\sigma_2 \tau_3 + \tau_2 \sigma_3 c_3)^2 + (\sigma_3 s_3)^2. \quad (A2)$$

A value of θ_2 is obtained as long as D_1 and D_2 are not simultaneously equal to zero. However, both determinants equal zero requires that the four conditions

$$\sigma_2 d_3 - \tau_2 a_3 s_3 = 0 \quad (A3)$$

$$a_2 + a_3 c_3 = 0 \quad (A4)$$

$$\sigma_2 \tau_3 + \tau_2 \sigma_3 C_3 = 0 \quad (A5)$$

$$\sigma_3 S_3 = 0 \quad (A6)$$

be simultaneously satisfied. We now examine the consequences of having all four conditions satisfied. Condition (A6) requires that $\sigma_3=0$ or $S_3=0$.

Case 1: $\sigma_3=0$ and $S_3 \neq 0$, then $\tau_3=\pm 1$ and from condition (A5), we see that σ_2 must be equal to 0 (and $\tau_2=\pm 1$). Condition (A3) then requires that $a_3=0$ but $\sigma_3=a_3=0$ means that joint axes 3 and 4 coincide and the arm is degenerate.

Case 2: $\sigma_3=S_3=0$, then $C_3=\pm 1$ and, as in the previous case, $\sigma_2=0$. Condition (A4) then becomes $a_2+a_3=0$ if $C_3=1$, or $a_2-a_3=0$ if $C_3=-1$. When $C_3=1$ ($\theta_3=0$), then $a_2+a_3=0$ means that $a_2=a_3=0$ since both link lengths a_2 and a_3 are non-negative numbers. Therefore, this case means that $\sigma_2=a_2=0$ and $\sigma_3=a_3=0$ so that joint axes 2, 3 and 4 coincide and the arm is degenerate.

When $C_3=-1$ ($\theta_3=\pi$), then $a_2-a_3=0$ and $a_2=a_3$. Since $\sigma_3=\sigma_2=0$, axes 2, 3, and 4 are parallel. The values $a_2=a_3$ and $\theta_3=\pi$ force joint axes 2 and 4 to be aligned, thereby forcing the arm in a degenerate configuration.

Case 3: $S_3=0$ and $\sigma_3 \neq 0$. This case occurs when $\theta_3=0$ or when $\theta_3=\pi$. When $\theta_3=0$, then $C_3=1$ and condition (A4) yields $a_2=a_3=0$ as in the previous case. Condition (A3) yields $\sigma_2=0$ or $d_3=0$.

Since $a_2=0$, the case $\sigma_2=0$ means that joint axes 2 and 3 are aligned and the arm is degenerate. Assuming $d_3=0$ and

$\sigma_2 \neq 0$, Condition (A5) gives $\sigma_2 \tau_3 + \tau_2 \sigma_3 = 0$ or $\sin(\alpha_2 + \alpha_3) = 0$. The values $\alpha_2 + \alpha_3 = 0$ or π , and $d_3 = a_2 = a_3 = 0$ mean that joint axes 2 and 4 are aligned so the arm is degenerate.

When $\theta_3 = \pi$, then $C_3 = -1$ and Condition (A4) forces $a_2 = a_3$ while Condition (A5) now becomes $\sigma_2 \tau_3 - \tau_2 \sigma_3 = 0$ or $\sin(\alpha_2 - \alpha_4) = 0$. The combination of values $(\alpha_2 - \alpha_4) = 0$, $a_2 = a_3$, and $\theta_3 = \pi$ again force axis 4 to align with axis 2 which puts the arm in a degenerate configuration.

Since the determinants D_1 and D_2 are simultaneously zero only when the manipulator is degenerate, we can conclude that a unique value of θ_2 can always be computed as indicated in Chapter 5 when the arm is non-degenerate.

REFERENCES

- Angeles, J. 1985. "On the numerical solution to the inverse kinematics problem." *Int. J. Robotics Res.*, 4(2): 21-37.
- Angeles, J. 1986. "Iterative kinematic inversion of general five-axis robot manipulators." *Int. J. Robotics Res.*, 4(4): 49-70.
- Craig, J.J. 1986. "Introduction to robotics mechanics and control." Addison-Wesley, Reading, Massachusetts.
- Denavit, J., and Hartenberg, R.S. 1955. "A kinematic notation for lower-pair mechanisms based upon matrices." *ASME J. Appl. Mech.*, 77: 215-221.
- Doty, K. L. 1986. "Machine Intelligence Lab. report MIL (10.86.1)." Electrical Engineering Dept., University of Florida, Gainesville, Florida.
- Doty, K. L. 1987. "Tabulation of the symbolic midframe Jacobian of a robot manipulator." *Int. J. Robotics Res.*, 6(4): 85-97.
- Duffy, J. 1980. "Analysis of mechanisms and robot manipulators." John Wiley, New York.
- Duffy J., and Crane, C. 1980. "A displacement analysis of the general spatial 7-link, 7R mechanisms." *Mech. Mach. Theory*, 15(3): 153-159.
- Duffy J., and Rooney, J. 1975. "A foundation for a unified theory of analysis of spatial mechanisms." *Trans. ASME, J. Engr. Industry*, 97(4, Series B): 1159-1164.
- Featherstone, R. 1983. "Position and velocity transformations between robot end-effector coordinates and joint angles." *Int. J. Robotics Res.*, 2(2): 35-40.
- Fu, K. S., Gonzalez, R. C., and Lee, C. S. 1987. "Robotics control, sensing, vision and intelligence." McGraw-Hill, New York.

- Goldenberg, A.A., Benhabib, B. and Fenton, R.G. 1985. "A complete generalized solution to the inverse kinematics of robots." IEEE J. of Robotics and Auto. RA-1: 1.
- Goldenberg, A.A. and Lawrence, D.L. 1985. "A generalized solution to the inverse kinematics of robotic manipulators." IEEE J. of Dynamic Systems, Measurement and Control, 107: 103.
- Hollerbach, J.M., and Sahar, G. 1983. "Wrist-partitioned, inverse kinematic accelerations and manipulator dynamics." Int. J. Robotics Res., 2(4): 61-76.
- Kazerounian, K. 1987. "On the numerical inverse kinematics of robotic manipulators." Trans. ASME J. Mech., Trans., and Auto. in Des., 109: 8-13.
- Lee, C.S.G., and Ziegler, M. 1984. "A geometric approach in solving the inverse kinematics of PUMA robots." IEEE Trans. Aerospace and Electronic Systems, AES-20(6): 695-706.
- Lee H. Y. and Liang C. G. (in press). "Displacement analysis of the general spatial 7-Link 7-R mechanism." Mech. Mach. Theory.
- Linares, J., and Page, A. 1984. "Position analysis of special mechanisms." Trans. ASME J. Mech., Trans., and Auto. in Des., 106: 252-255.
- Low, K.H., and Dubey, R.N. 1986. "A comparative study of generalized coordinates for solving the inverse kinematics problem of a 6R robot manipulator." Int. J. Robotics Res., 5(4): 69-88.
- Lumelsky, V.J. 1984. "Iterative coordinate transformation procedure for one class of robots." IEEE Trans. Sys., Man and Cyber., SMC-14(3): 500-505.
- Manseur, R., and Doty, K.L. 1988. "A fast algorithm for inverse kinematic analysis of robot manipulators." Int. J. Robotics Res., 7(3): 52-63.
- Manseur, R. and Doty, K.L. (in press). "A robot manipulator with 16 real inverse kinematic solution sets." Int. J. Robotics Res.
- Paul, R.P. 1981. "Robot manipulators: mathematics, programming and control." MIT Press, Cambridge, Massachusetts.

- Paul, R.P. and Zhang, H. 1986. "Computationally efficient kinematics for manipulators with spherical wrists based on the homogeneous transformation representation." Int. J. Robotics Res. 5(2): 32-44.
- Pieper, D.L. 1968. "The kinematics of manipulators under computer control." Ph.D. Dissertation, Stanford University, Stanford, California.
- Renaud, M. 1980a. "Calcul de la matrice jacobienne nécessaire à la commande coordonnée d'un manipulateur." Mech. Mach. Theory, 15(2): 81-91.
- Renaud, M. 1980b. "Contribution à la modélisation et à la commande dynamique des robots manipulateurs." These de Docteur d'Etat, Université Paul Sabatier de Toulouse (Sciences), Toulouse, France.
- Roth, B., Rastegar, J., and Sheinman, V., 1973. "On the design of computer controlled manipulators." On the theory and practice of robots and manipulators 1. Springer-Verlag, New York, 1: 93-113.
- Takano, M. 1985. "A new effective solution for inverse kinematic problem (synthesis) of a robot arm with any type of configuration." J. of the Faculty of Engineering. The University of Tokyo (B). 38(2): 107-135.
- Tsai, L.W. and Morgan, A.P. 1984. "Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods." ASME Paper 84-DET-20. ASME Design Engineering Technical Conference, Cambridge, Massachusetts.
- Uicker, J.J., Jr., Denavit, J. and Hartenberg, R.S. 1964. "An iterative method for the displacement analysis of spatial mechanisms." ASME J. Appl. Mech. 31(Series E): 309-314.
- Wu, C. H., and Paul, R. P., 1982. "Resolved motion force control of robot manipulators." IEEE Trans. Sys. Man. and Cyber., SMC-12(3): 266-275.

BIOGRAPHICAL SKETCH

Mr. Rachid Manseur was born in Algiers, Algeria, on February 17, 1954. He received a licence-es-sciences in mathematics from the University of Algiers on July, 1976.

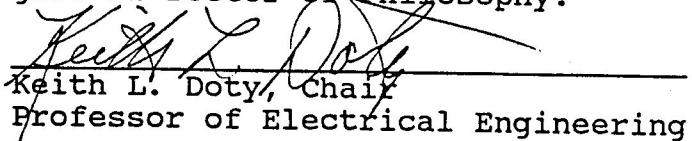
In August 1977, he entered a graduate program in electrical engineering at the University of Houston where he obtained a Master of Science degree in May, 1980.

From July 1980 to October 1983, Rachid Manseur worked as a research engineer for the Algerian National Oil Company and taught computer engineering courses at the ENITA Algerian School for Engineers.

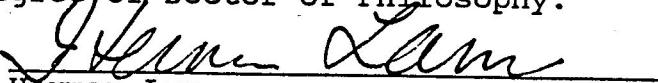
In Fall of 1984, He was admitted to the Electrical Engineering Department of the University of Florida, where he has been a teaching assistant in the Mathematics Department and in the Electrical Engineering Department while actively conducting research at the Machine Intelligence Laboratory.

Rachid Manseur is married and has two children. He intends to join the faculty of the Electrical Engineering Department of the University of Florida in Fall of 1988.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Keith L. Doty, Chair
Professor of Electrical Engineering

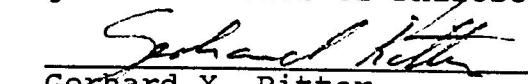
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Herman Lam
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Jose C. Principe
Associate Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Gerhard X. Ritter
Professor of Computer and Information Sciences

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Ralph G. Selfridge
Professor of Computer and Information Sciences

This dissertation was submitted to the Graduate Faculty
of the College of Engineering and to the Graduate School and
was accepted as partial fulfillment of the requirements for
the degree of Doctor of Philosophy.

August 1988

Dean, College of Engineering

Dean, Graduate School