N89-10094

MP5321 P.20

An Optimal Resolved Rate Law for Kinematically Redundant Manipulators

B. J. Bourgeois

McDonnell Douglas Astronautics Co.
- Engineering Services

May 1987

An Optimal Resolved Rate Law for Kinematically Redundant Manipulators

B. J. Bourgeois
McDonnell Douglas Astronautics Co. - Engineering Services

ABSTRACT - The resolved rate law for a manipulator provides the instantaneous joint rates required to satisfy a given instantaneous hand motion. When the number of degrees of freedom in the task space is the same as the number of degrees of freedom in the joint space the Jacobian matrix is square and the resolved rate law is easily determined for non-singular configurations. When the joint space has more degrees of freedom than the task space the manipulator is kinematically redundant and the kinematic rate equations are under-determined. In this paper an objective function is optimized with respect to the n kinematically redundant rate equations to provide an optimal resolved rate law that can be tuned to control the joint motion in a variety of ways. This law is used in an iterative algorithm to find joint angle solutions to the inverse nonlinear kinematic equations. The behaviour of the optimal resolved rate law is demonstrated and investigated in a 4 degree of freedom kinematically redundant planar arm model. A weighting matrix is used in the resolved rate law to avoid reach limits during the trajectory to a desired hand state. The treatment is applicable to manipulators with any number of revolute joints.

1.0 INTRODUCTION

The resolved rate law for a manipulator converts the instantaneous hand rates into instantaneous joint rates [1]. This allows the joints to be simultaneously commanded to move the hand with a desired instantaneous translational and rotational velocity. The space that the hand moves in is called the task space [2], which is usually composed of 6 degrees of freedom. The space mapped out by the joint angles is called the joint space. The mathematical relationship between the task space and the joint space defines the resolved rate law for the manipulator.

The kinematic equations express the hand state in terms of the manipulator joint angles and are usually very nonlinear. It is usually straight-forward, but tedious to write down the kinematic equations that express the hand state in terms of the joint angles. It is easy to differentiate the kinematic equations to arrive at an expression for the hand translational and rotational rates in terms of the joint rates. The matrix that results is sometimes called the Jacobian matrix for the manipulator. The kinematic equations for most manipulators are very nonlinear and generally cannot be inverted to solve for the joint angles in terms of the hand parameters [2].

The Shuttle Remote Manipulator System (SRMS) has six joints and the end-effector operates in a six degree of freedom space (three spacial and three angular). This is a convenient design because the number of degrees of freedom in the joint space and in the task space are the same; the Jacobian matrix is square and can be easily inverted, except in specific configurations where the Jacobian matrix is singular. When these singularities are avoided, the inverted Jacobian is a valid resolved rate law for the SRMS because it transforms a desired end-

effector translational and rotational velocity into joint rate commands. The joint rate commands can be issued periodically to the six servo motors to accomplish an end-effector motion as desired.

The simple resolved rate law described above is used in the SRMS flight software to drive the manual and the automatic modes. For these modes there is a requirement to move the hand coordinate system (or some coordinate system rigidly associated with the hand system) from one state to another with translation along a relatively straight path and rotation about a constant vector. Much effort has been spent finding such paths that are free from encounters with joint reach limits.

When a manipulator has more joints than the number of degrees of freedom in the task space it is said to be kinematically redundant [3]. The Jacobian matrix for a kinematically redundant manipulator is not square and cannot be directly inverted to arrive at an easy resolved rate law. There are more joint variables to solve for than there are kinematic equations. There is not enough information to solve for the joint rates needed to move the hand. In general, there may be an infinite number ways to move the joints in unison to provide the desired hand motion for the kinematically redundant manipulator [4],[5].

It is essential to arrive at some sort of a resolved rate law in order to control or simulate a manipulator. Several methods have been introduced to arrive at adequate resolved rate laws for the kinematically redundant manipulator. One approach is to add specific constraints on the manipulator so that the kinematic equations can be solved. A more general approach is to minimize or maximize an objective function subject to the kinematic constraint equations. These methods have been investigated in several papers to study iterative solutions to the kinematically redundant constraint equations [1],[3-7].

In this paper an optimal control law with a weighting matrix is derived using the Moore-Penrose pseudo-inverse for general manipulators with various dimensions in task space and joint space. The behavior of the control law is demonstrated and investigated using a kinematically redundant planar arm simulation. Several algorithms are introduced and evaluated for dynamically adjusting the weighting matrix during the trajectory for the purpose of avoiding joint reach limits.

2. PROBLEM FORMULATION

The resolved rate law for a manipulator is derived from the kinematic equations. For a manipulator with n joints and a hand operating in a task space of m dimensions, the m kinematic equations are of the form:

$$x = \{x_k\} = \{f_k(\theta_1, \theta_2, \ldots \theta_n)\} \quad k=1,m \tag{2.1}$$

where x is the vector containing the task space coordinates and θ is the vector of joint angles. If each joint is moved by a small amount, $\Delta\theta$, then the movement of the hand in the task coordinates, Δx, is found in the differential of the kinematic equations:

$$\Delta x = [J]\, \Delta\theta \tag{2.2}$$

ORIGINAL PAGE IS
OF POOR QUALITY

where J is the Jacobian matrix [8], composed of the partial derivatives of the functions f with respect to each of the joint angles. Similarly, the kinematic rate equations are found by differentiating the kinematic equations with respect to time.

$$v = dx/dt = [J]\, w \tag{2.3}$$

where v is the hand velocity vector expressed in the task coordinates and w is the vector of joint rates. The resolved rate law is found by solving the kinematic rate equations (2.3) for the joint rates (w) in terms of the hand velocity (v). In the case where the task space and the joint space have the same number of dimensions (m=n) the Jacobian matrix is square and the resolved rate law is easily found.

$$w = [J]^{-1} v \tag{2.4}$$

When the determinant of the Jacobian is zero, the manipulator is in a physical singularity and cannot supply motion in all of the dimensions of the task space. In mathematical terms the joint space does not span the task space when the arm is in a singularity.

When the manipulator has fewer joints than dimensions in the task space (n<m) the system is overdetermined and the resolved rate law may be found by using the pseudo-Jacobian method. For a 5 jointed manipulator acting in a task space of 6 dimensions, there will be 6 kinematic equations but only 5 joint variables.

$$\underset{6x1}{v} = \underset{6x5}{J}\ \underset{5x1}{w} \tag{2.5}$$

The brackets around the Jacobian have been dropped to simplify the notation. The resolved rate law for such a manipulator can be derived by pre-multiplying by the Jacobian as follows:

$$J^T v = J^T J\, w \tag{2.6}$$

$$w = (J^T J)^{-1} J^T v \tag{2.7}$$

$$\underset{5x1}{w} = [\ (\underset{5x6}{J^T}\ \underset{6x5}{J})^{-1}\ \underset{5x6}{J^T}\ ]\ \underset{6x1}{v} \tag{2.8}$$

The expression within the brackets is the Moore-Penrose pseudo-inverse of the full rank rectangular (6x5) Jacobian for the overdetermined system of equations 2.5 [9].

For a kinematically redundant manipulator (n>m) the Jacobian matrix is not square and the system of equations is underdetermined. For a 7 jointed manipulator operating in a task space of 6 dimensions there are 6 kinematic equations and 7 joint variables. The Jacobian matrix is a 6 by 7 matrix. There are several approaches which have been used to solve this underdetermined set of equations. The approach taken here is to

introduce an objective function to be minimized subject to the constraint equations:

$$\underset{6x1}{v} = \underset{6x7}{J}\ \underset{7x1}{w} \tag{2.9}$$

An obvious objective function to consider is:

$$Z = \frac{1}{2}\left(w_1^2 + w_2^2 + w_3^2 \ldots + w_n^2 \right) \tag{2.10}$$

When this function is minimized, the solution results in the least amount of instantaneous motion in all joints. Using the method of Lagrangian multipliers (L) the following n+m equations result [11].

$$w = J^T L \tag{2.11}$$

$$J w = v \tag{2.12}$$

Solve equation 2.11 for the Lagrangian multipliers and use equation 2.12 to substitute v for Jw.

$$L = \left(J J^T \right)^{-1} J w = \left(J J^T \right)^{-1} v \tag{2.13}$$

Substituting this expression for the Lagrangian multipliers gives the resolved rate law.

$$w = \left[J^T \left(J J^T \right)^{-1} \right] v \tag{2.14}$$

It is interesting to compare this result with that of the pseudo-inverse result of equation 2.7. The expression in brackets in equation 2.14 is the Moore-Penrose pseudo-inverse of the Jacobian for the underdetermined system of equations (2.9) [4],[9].

A more interesting objective function [1] has the form:

$$Z = \frac{1}{2}\left(a_{11} w_1^2 + a_{22} w_2^2 + \ldots a_{nn} w_n^2 \right) \tag{2.15}$$

or in matrix notation:

$$Z = \frac{1}{2} w^T A w \tag{2.16}$$

The constraint on the weighting matrix A is such that the function Z is non-negative for all values of w [11]. This condition will be satisfied by considering only diagonal weighting matrices with positive values. The resolved rate law that results from optimizing this objective func-

tion can be found by following the procedure used in equations 2.11 through 2.14.

$$w = A^{-1} J^{T} (J A^{-1} J^{T})^{-1} v \qquad (2.17)$$

This result also appears in [1]. For the case of the seven jointed manipulator described above this resolved rate law is

$$\underset{7x1}{w} = \underset{7x7}{A^{-1}} \underset{7x6}{J^{T}} (\underset{6x7}{J} \underset{7x7}{A^{-1}} \underset{7x6}{J^{T}})^{-1} \underset{6x1}{v} \qquad (2.18)$$

where the dimensions of each matrix and vector have been indicated for clarity. The weighting matrix A is of special interest. It can be used to control the motion of the joints by dynamically changing the values of the diagonal components during the trajectory.

3.0 IMPLEMENTATION

The control law expressed by equation 2.17 was implemented into a kinematically redundant planar manipulator model. The planar arm model was developed for the purpose of studying the motion of the redundant manipulator. This model is easy to work with because the task space is confined to a flat plane and is adequate for studying the behavior of the control law.

3.1 The Kinematically Redundant Planar Manipulator (KRPM)

The KRPM model has a task space composed of 3 degrees of freedom as shown in figure 1. The task space is composed of X, Z and P (pitch) directions for the hand, and all joints are pitch joints. The number of pitch joints (n) can be specified from 3 to 10. For this study, n was set to 4 so that there is only 1 redundant joint except where noted otherwise. This was done to form a direct analogy with the 7 jointed manipulator operating in a task space of 6 degrees of freedom. The kinematic equations for the 4 jointed planar arm are shown below. The usual abbreviated notation is used to simplify the algebra.

$c1 = \cos(\theta_1)$
$c12 = \cos(\theta_1 + \theta_2)$
$c123 = \cos(\theta_1 + \theta_2 + \theta_3)$
$c1234 = \cos(\theta_1 + \theta_2 + \theta_3 + \theta_4)$
similarly for s1, s12, s123, and s1234 using sines
L_i = Length of boom i (3.1)

ORIGINAL PAGE IS
OF POOR QUALITY

The kinematic equations are now easily written.

$$X = L_1c1 + L_2c12 + L_3c123 + L_4c1234 \tag{3.2}$$

$$Z = -L_1s1 - L_2s12 - L_3s123 - L_4s1234 \tag{3.3}$$

$$P = \theta_1 + \theta_2 + \theta_3 + \theta_4 \tag{3.4}$$

The Jacobian for this arm can be found analytically by differentiating the above equations.

$$\begin{vmatrix} -L1s1 - L2s12 & -L2s12 - L3s123 & -L3s123 & -L4s1234 \\ -L3s123 - L4s1234 & -L4s1234 & -L4s1234 & \\ -L1c1 - L2c12 & -L2c12 - L3c123 & -L3c123 & -L4c1234 \\ -L3c123 - L4c1234 & -L4c1234 & -L4c1234 & \\ 1 & 1 & 1 & 1 \end{vmatrix} \tag{3.5}$$

The Jacobian for the planar arm can be greatly simplified by considering the special case where all of the boom lengths are set to unity. The following abbreviations are also convenient.

$$\begin{aligned} c14 &= c1 + c12 + c123 + c1234 \\ c24 &= c12 + c123 + c1234 \\ c34 &= c123 + c1234 \\ &\text{similarly for s14, s24, and s34} \end{aligned} \tag{3.6}$$

The Jacobian for the special case can now be conveniently expressed as follows.

$$J = \begin{vmatrix} -s14 & -s24 & -s34 & -s1234 \\ -c14 & -c24 & -c34 & -c1234 \\ 1 & 1 & 1 & 1 \end{vmatrix} \tag{3.7}$$

The KRPM model was simulated in FORTRAN on an HP9000 desktop 32 bit super micro computer. The model simulates the kinematics of any planar arm with 3 task dimensions (m=3) and any number of pitch joints (n) and booms of independent lengths. The Jacobian is computed numerically using a recursive vector form [1] to allow the simulation of various arms types. The attributes of the manipulator are described in a data file. Various manipulators may be represented by changing the data file.

3.2 Implementation of the Control Law

The optimal RRL (equation 2.17) was implemented into the KRPM arm model by first adapting the dimensions to those of the planar arm. The task space contains 3 dimensions, and the number of joints (n) may be 4 or greater. The RRL for the 4 jointed KRPM is defined as follows with dimensions shown.

$$[RRL]_{4x4} = A^{-1}_{4x4} J^{T}_{4x3} \left(J_{3x4} A^{-1}_{4x4} J^{T}_{4x3} \right)^{-1} \qquad (3.8)$$

The simulation iterates through the RRL to drive the hand to a desired state. The flow of the calculation is as follows. The arm is positioned at a valid set of joint angles ($\theta_{initial}$) and through the forward kinematics the resulting hand state ($x_{initial}$) is computed. Then an input is made to indicate the desired final hand state for the arm (x_{final}). The difference between the two hand states (Δx) is found.

$$\Delta x = x_{final} - x_{initial} \qquad (3.9)$$

The number of steps to take during the trajectory (s) can be selected. The vector Δx is divided into s steps. The RRL is computed using equation 3.8 and then the desired joint angle step $\Delta\theta$ is computed.

$$\Delta\theta = [RRL]\ \Delta x/s \qquad (3.10)$$

The joint angles are then updated by adding the changes in joint angles. This procedure is repeated for steps 2 through s, maintaining the same step length in distance (X and Z) and in rotation but with an adjustment in the vector direction ($|\Delta x|$). After the last step is taken (step number s), a Newton-Raphson (NR) iteration is automatically invoked to trim up the final hand state to within a tolerance of the desired hand state. This method does not consider the joint rates of the manipulator. This simulation progresses by taking steps.

3.3 Step Size

A study was performed to determine the step size needed to provide hand motion along a straight line. Good results are measured by inspecting the path that the end-effector describes. The ideal trajectory should be a straight line. Several trajectories were tested while varying the number of steps between 1 and 80.

In figure 2 the arm is commanded from the joint angle state at A to the hand state at B. When a Newton-Raphson (NR) iteration is used (s=1) the first step actually goes the wrong way, to point 1 in figure 2a. When the NR iteration is completed, the arm ends up at the joint angle state shown at B. In figure 2b a two step iteration (s=2) is shown. The first step is a NR half-step and the second step is the remaining half step with an adjustment in direction. A full step NR iteration occurs between 2 and B resulting in a different final joint state than in 2a. A ten step iteration (2e) results in a reasonably straight hand trajectory, but eighty steps are required for very good results (2f).

In figure 3a the arm was commanded from the joint state of (90,0, -90,0) at A to the hand state of (3,0,0) at B with a small step size arriving at the joint configuration at B. This trajectory was then reversed by commanding from the joint angles at point B to the original hand state at A (2,-2,0) for various step sizes. Notice that for large

step sizes (figures 3b and 3c), the arm does not return to the original joint configuration of (90,0,-90,0) in figure 3a. As the step size decreases (a larger number of steps), the reverse trajectory converges upon the perfect joint state of (90,0,-90,0). This is expected because the trajectory that minimizes the motion of the joints in one direction should also minimize the motion in the reverse direction also. This procedure is considered a validation of the implementation of the RRL. These results illustrate the importance of taking small step sizes while seeking a practical joint angle solution for a redundant arm using the iterative inverse method.

3.4 Iterative Inverse Solutions

Several investigators have used modified NR algorithms to find inverse kinematic solutions for manipulators [3],[5],[6],[10]. All of these approaches are aimed at finding a quick joint angle solution with large step sizes in joint space, causing the hand of the manipulator to take an unpredictable and unrealistic path. This poses two problems when dealing with a physical manipulator. When the robot is actually commanded from the initial joint angles to the hand state through a resolved rate law, a different set of joint angles is likely to result than the one found by the iterative inverse. This effect is best illustrated in figure 2 where the joint angles at the final configuration in 2a are very different from the joint angles at the end of figure 2e. The joint angle solution is not very useful if the manipulator cannot be commanded to it. Secondly, when large steps in joint space are made there is more chance of violating the joint reach limits.

If the iterative inverse for the redundant arm constrains the hand to follow a straight path, rotate about a constant vector and checks for joint reach limits during the iteration then a solution arrived at will be a feasible one. This requires having a knowledge of the RRL that will drive the manipulator and taking small steps in the iteration.

4.0 BEHAVIOUR OF THE OPTIMAL RESOLVED RATE LAW

The behaviour of the optimal resolved rate law in the KRPM model is demonstrated in this section. Several trajectories were run to illustrate the ability of the control law to handle the redundancy of the kinematics and to study the effects of the weighting matrix on the joint motion.

The ability of the control law to handle the redundancy of the arm was demonstrated by driving the arm to the same hand state from various starting configurations. It also serves as an inverse solution to the kinematics, by providing several possible joint sets that satisfy the requested hand state. In figure 4 the end-effector was commanded to the state X=3, Z=0, and Pitch= 0 (3,0,0) from six different initial joint configurations. In each case the end-effector ends up at the final state of (3,0,0), but with different final joint angles. This simple test demonstrates the ability of the control law to drive the KRPM to different final joint states for a given end-effector state. The final joint angles are dependent on the initial joint angles.

The above maneuvers were performed with the weighting matrix A in

equation 3.8 equal to identity. This is the equivalent of having no weighting matrix (equation 2.14). The effect of the weighting matrix on the motion was demonstrated by running the same trajectory with various values of one of the components of the A matrix and observing the effects on the motion of the corresponding joint.

In figure 5 the arm was commanded to the end-effector state of (2,0,0) at B from the joint angle state (90,-90,0,0) at A. When the weighting matrix is not used (A is identity), the final value of the second joint is undesireable (fig. 5a). When a value of 2.0 is used for A(2,2) and 1.0 for all other diagonal components of A, the final position of joint 2 is noticably better (fig. 5b). Joint 2 moved less from start to finish than in figure 5a. The joint moves progressively less from start to finish as A(2,2) is increased. The remaining pitch joints have moved more to make up for the loss of mobility in joint 2, thus resulting in a more desirable overall final arm configuration.

In figure 5a the final condition of the arm is not disireable because joint 2 could be very near a reach limit, thus restricting any future movement after arrival at the desired end-effector state. In figure 5e, the final situation is much more desireable, because joint 2 has more freedom to move around in the neighborhood of the final end-effector state.

In this section it has been demonstrated that the weighting matrix can be used to discourage the motion of particular joints. It seems reasonable to use this information to maintain the joints away from their respective joint reach limits. This is a very desireable goal in robotic control, but is limited to redundant manipulators.

5.0 REACH AVOIDANCE ALGORITHMS

The behaviour of the pseudo-inverse of equation 2.14 has been reported to be peculiar in some cases [4]. The peculiarity has been associated with joint reach limit violations during certain tasks such as a closed path or cyclic motion. If reach avoidance logic is incorporated into the RRL, these problems may be resolved. One method of incorporating reach avoidance into the RRL is to include the upper and lower joint limits as a constraint in the optimization algorithm [6], which may become a complicated treatment. A simpler approach is taken here which makes use of the weighting matrix A in the RRL of equation 3.8.

In the previous section the effect of the weighting matrix on the arm motion was illustrated. It was shown that the redundant arm can be controlled to arrive at different final joint angles, some more desireable than others, and yet satisfy the same hand state. With these two findings, it is evident that the arm can be driven to arrive at various final joint angles as desired by controlling the weighting matrix during the trajectory.

The components of the weighting matrix (diagonal) must be computed from pass to pass according to some driving requirements. Examples of driving requirements are obstacle avoidance, joint reach limit avoidance, or some mechanical or electrical criteria. For this study, the goal is reach limit avoidance.

Three algorithms were implemented for evaluation. The first algo-

ORIGINAL PAGE IS
OF POOR QUALITY

rithm is the simplest: when any joint is within a tolerance of a reach limit then the component of the weighting matrix for that joint is set to a large value (ABIG), otherwise the component is set to unity. ABIG will be a design constant that may be varied to provide the desired performance.

The second algorithm is similar to the first, but has the following requirement. If a joint is moving away from its reach limit then the weighting matrix component for that joint is set back to unity. This encourages the joint to move away from the tolerance zone.

The third algorithm does not use the tolerance test. The value of the weighting matrix component for each joint is scaled from 1 at its midrange value to ABIG at either of its joint reach limits. Also, as in algorithm number 2, if the joint is moving toward its midrange value then the value of the weighting matrix component is set to unity. This algorithm is designed to encourage each joint to stay near the midrange value.

Each of the above three algorithms were implemented into the KRPM model described previously and tested until validated. The algorithms were then used to study a single joint encountering a reach limit and two joints encountering reach limits simultaineously.

To test the ability to avoid a single joint reach limit, take the case where the start and end of a trajectory are known to be valid end-effector states, but a reach limit is encountered without reach avoidance. The trajectory shown in figure 4d was used for this test where the third joint begins at -90 degrees, reaches -106 degrees, and ends at -87 degrees. Suppose that the limit for this joint is at -100 degrees. Figure 6 shows the trace of the third joint with no reach avoidance and for each of the three reach limit avoidance algorithms using a value of 100 for Abig. Each of the algorithms successfully avoided the imposed reach limit. In method 1 the joint angle does not move back out of the tolerance zone of 10 degrees. In methods 2 and 3, the joint moved back out of the reach zone of 10 degrees from the reach limit.

The trajectories with and without reach avoidance are shown in figure 7. Notice that with reach avoidance the first joint moves faster during the first few interations (figures 7b, c, and d) than without reach avoidance (figure 7a).

In figure 8a the arm was commanded from the joint state of (90,0, -135,90) to the hand state of (-.1,-2,90) causing two joints, joint 3 and 4, to approach reach limits. With reach avoidance both joint positions are improved in the final configuration (figure 8b).

In the case shown in figure 9a joint 3 exceeds a -160 degree limit and then goes past -180. With reach avoidance (figure 9b) joint 2 swings out dramatically to allow joint 3 to avoid its reach limit.

6.0 SUMMARY

The pseudo-inverse Jacobian with a weighting matrix has been derived as a resolved rate law for the kinematically redundant manipulator. The resolved rate law has been demonstrated with a kinematically redundant planar manipulator model. Reach avoidance has been mostly successfull with this model by dynamically adjusting the components of

the ng matrix during a maneuver. In some extreme cases the reach limit not avoidable. The locally optimized resolved rate law has been ved by incorporating joint reach avoidance. Reach avoidance has used in the inverse seeking solution to arrive at feasible solut The need has been demonstrated for using small steps sizes in trative inverse seeking algorithm for the purpose of arriving at usoint angle solutions for trajectory planning.

work could be aimed at studying reach avoidance techniques duririal tasks such as cyclic motion. A similar exercise should be ped with a manipulator operating in 6 degrees of freedom as the behavould be different than for the planar arm which operates in a limitne with all pitch joints.

ACKNCENTS

The wishes to express appreciation to Barry Rogers and Susan Rogerthe development of the KRPM model and to Richard Theobald for his ging support and final review of the paper.

7.0 RCES

[1] . Whitney, "The Mathematics of Coordinated Control of stic Arms and Manipulators," *J. of Dynamic Systems, ement and Control,"* Trans. ASME, Vol. 92, pp. 303-309, Dec.

[2] aig, *Introduction to Robotics*, Addison-Wesley, 1986.

[3] nhabib, A.A. Goldenberg, and R.G. Fenton, "A Solution to the Kinematics of Redundant Manipulators," *J. of Robotic*, 2(4), 373-385 (1985).

[4] Klein and C. Huang, "Review of Pseudoinverse Control for Use inematically Redundant Manipulators," *IEEE Trans. Systems, d Cybernetics*, Vol SMC-13, No. 3, March/Apr 1983.

[5] D. Orin, and M. Bach, "An Inverse Kinematic Solution for lically Redundant Robot Manipulators," *J. of Robotic*, 1(3), 235-249 (1984).

[6] Goldenberg, B. Benhabib, and R.G. Fenton, "A Complete ized Solution to the Inverse Kinematics of Robots", *IEEE of Robotics and Automation*, Vol RA-1, No. 1, pp14-20, 1985.

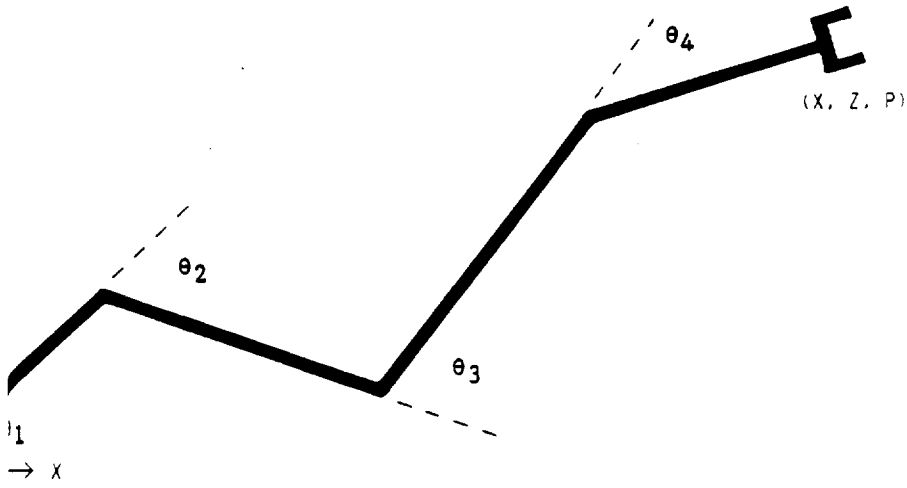
[7] hati, P. Morasso, and V. Tagliasco, "The Inverse Kinematic for Anthropomorphic Manipulator Arms," *J. of Dynamic, Measurement, and Control*, Vol. 104, pp. 110-113, March

[8] Paul, *Robot Manipulators: Mathematics, Programming and* , MIT Press., Cambridge, MA, 1981.

[9] e, Malcom, and Moler, *Computer Methods for Mathematical tion*, Prentice-Hall, NJ, 1977.

[10] , D.E. "Optimum Stepsize Control for Newton-Raphson Solution linear Vector Equations," *IEEE Transactions on Automation* , Vol. AC-14, No. 5, pp.572-74, Oct. 1969.

[11] ik, Raymond L., *Theory and Techniques of Optimization For ing Engineers*, Barnes & Nobles, NY, 1971.



; 1. A kinematically redundant planar manipulator (KRPM) 'our joints and a task space of X, Z, and Pitch (P).

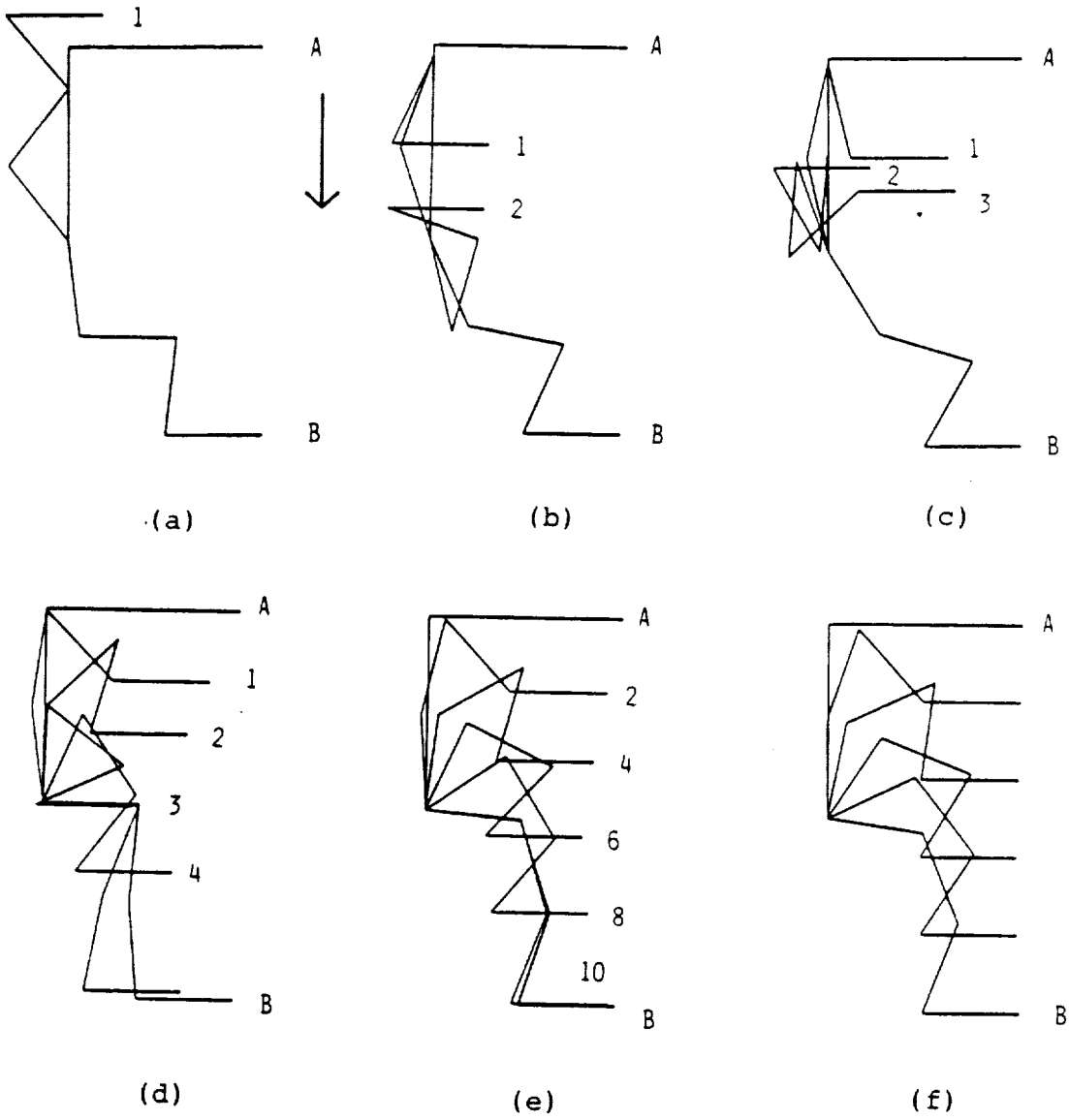


Figure 2. When a full step is taken from A to B, the first step (indicated by a "1") goes the wrong way. As smaller step are taken from A to B by dividing the path into 2 (b), 3(c), 5(d), 10(e), and 80 (f) equal steps before taking full steps to get to B, the trajectory becomes straight and each step goes in the correct direction.

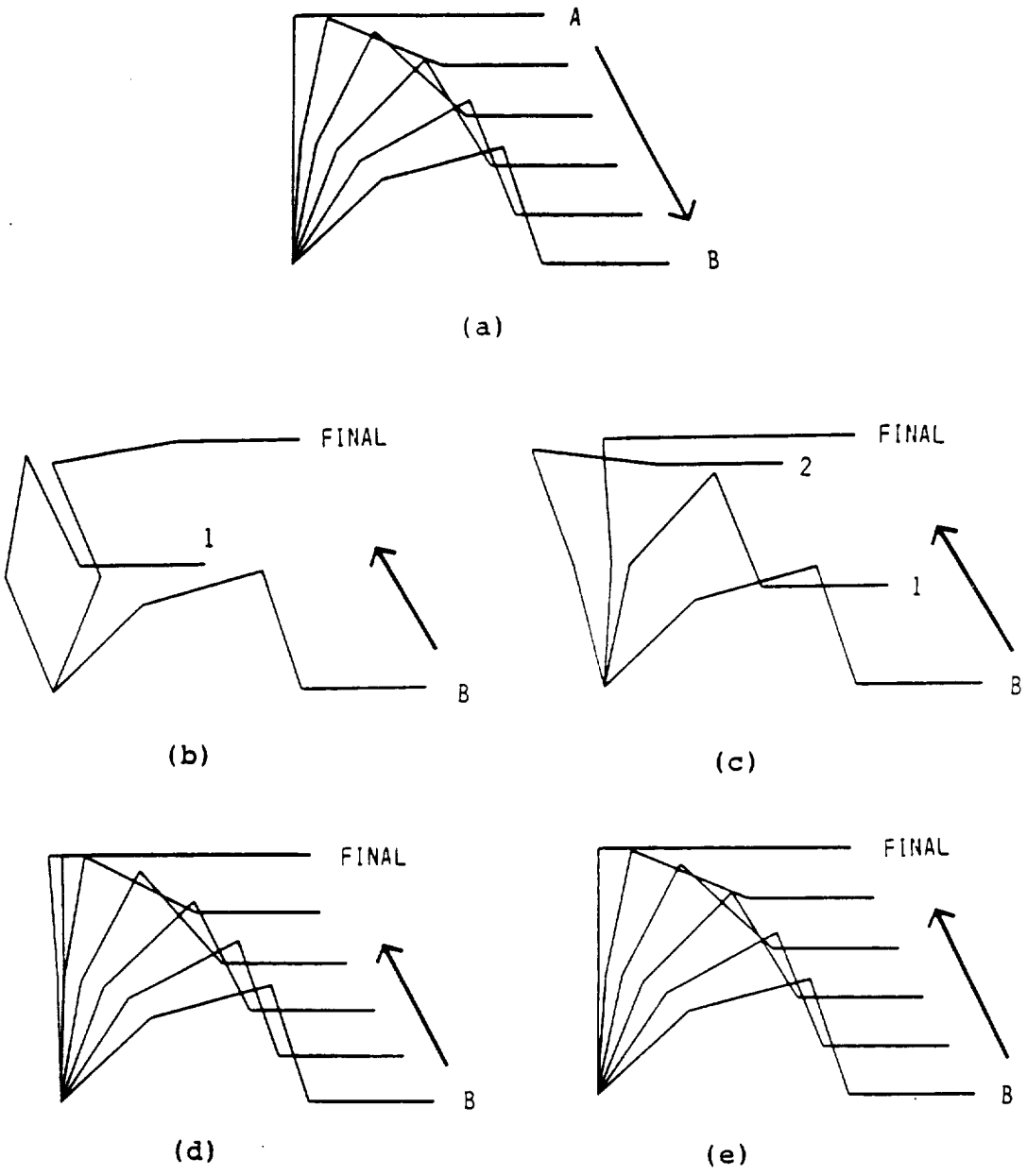


Figure 3. The arm was moved from A to B taking small steps (a). Stepping back to A with a full step iteration does not achieve the same joint angles as A. As smaller steps are taken as shown in the 2 step (b), 5 step (c), and 40 step (d) iterations the original configuration of A in (a) is approached.

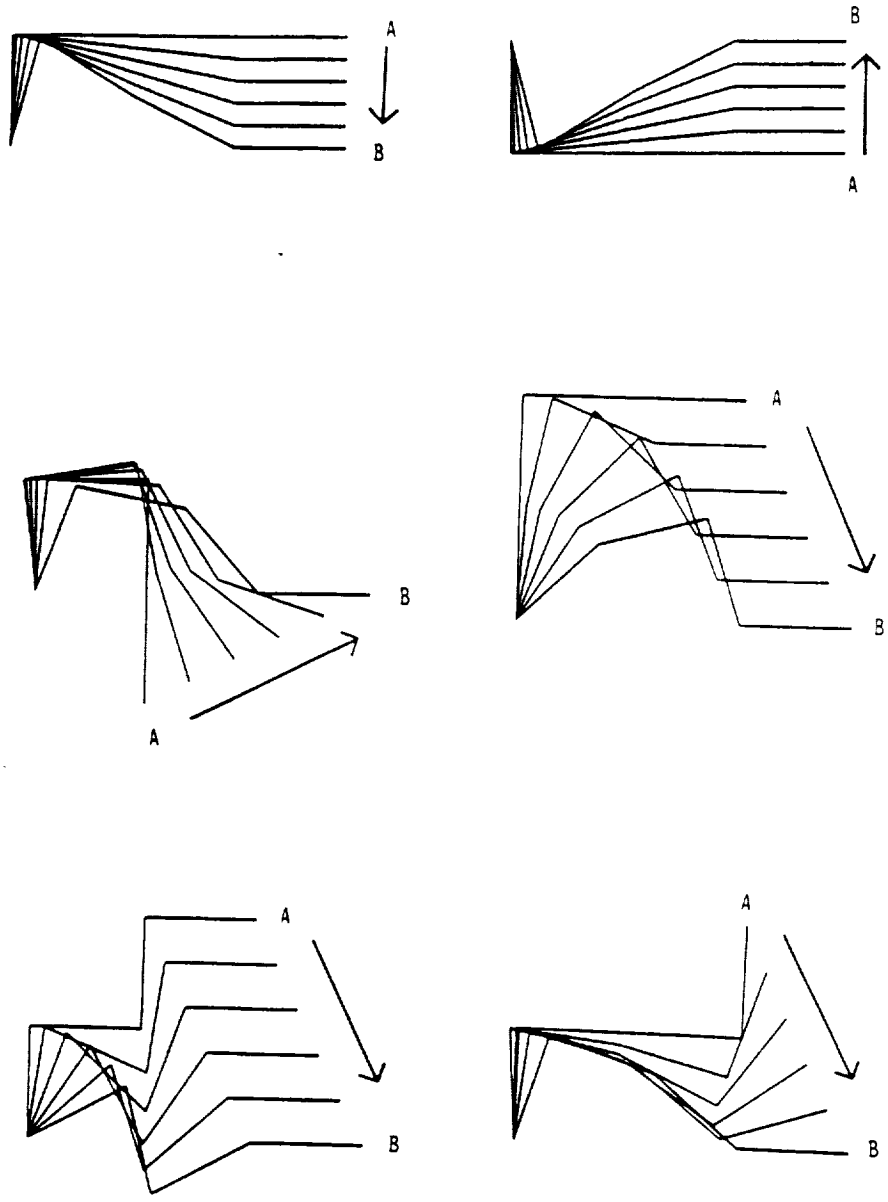
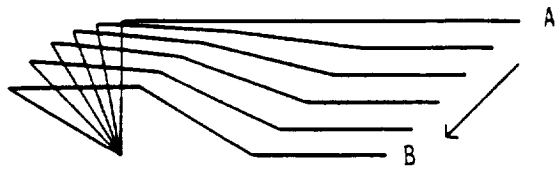
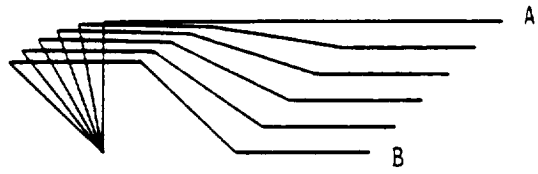


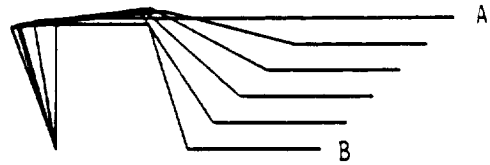
Figure 4. The KRPM is commanded to the same hand state of 3,0,0 at B from six different initial joint angle states at A to demonstrate the behaviour of the RRL.



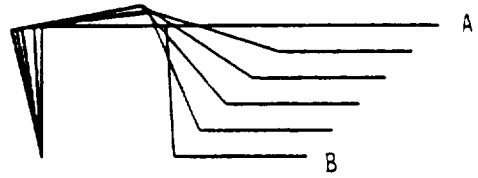
(a)



(b)



(c)



(d)

Figure 5. Effects of the weighting matrix on the trajectory are shown for values of A(2,2) set at 1 (a), 2 (b), 10 (c), and 100 (d).

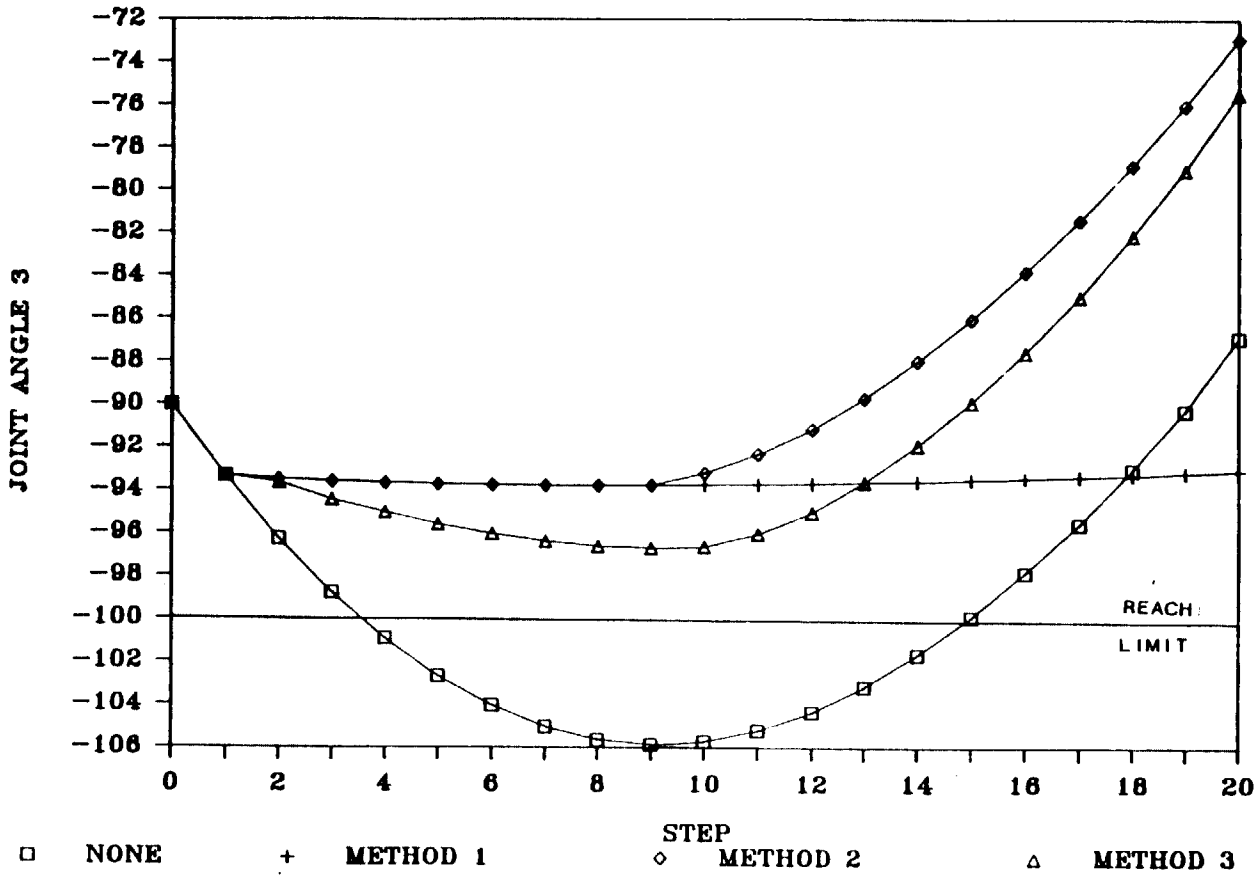


Figure 6. The effects of reach avoidance on joint 3 during a command from a joint state of 90,0,-90,0 to a hand state of 3,0,0. The reach limit is successfully avoided for each of the 3 reach avoidance algorithms.

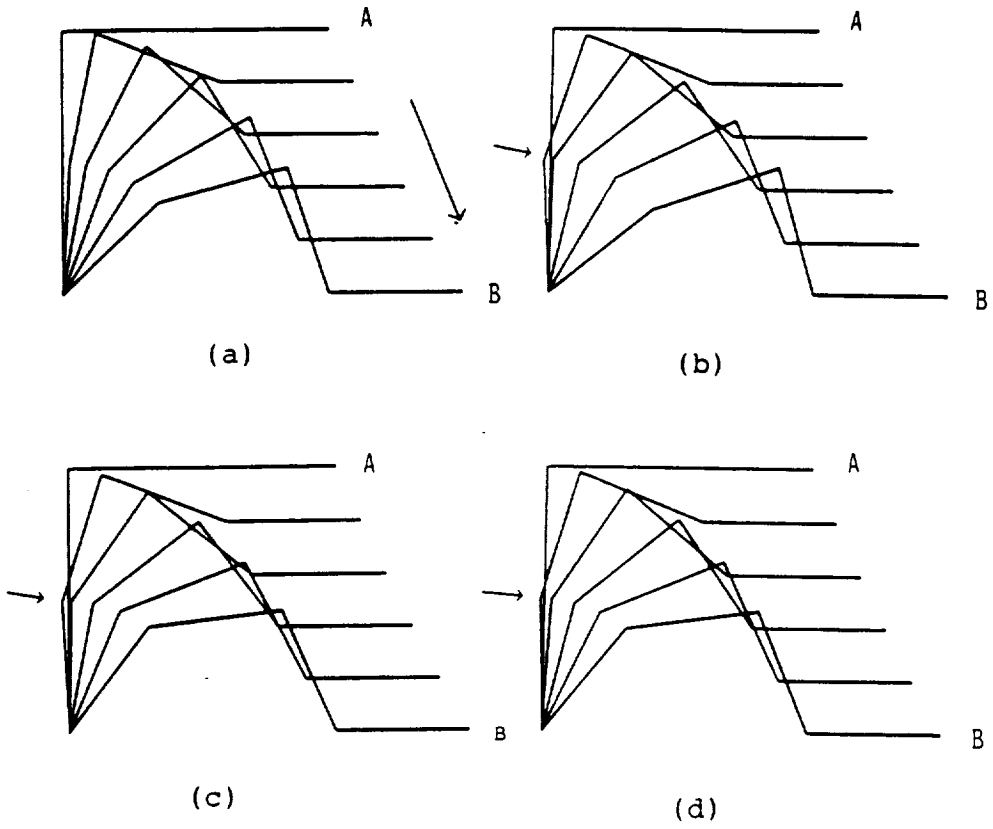


Figure 7. The trajectories are shown for the case of Figure 6 for no reach avoidance (a) where joint 3 violates a limit of -100 degrees, and for each of the reach avoidance algorithms: method 1 (b), method 2 (c), and method 3 (d) where the reach limit is avoided.

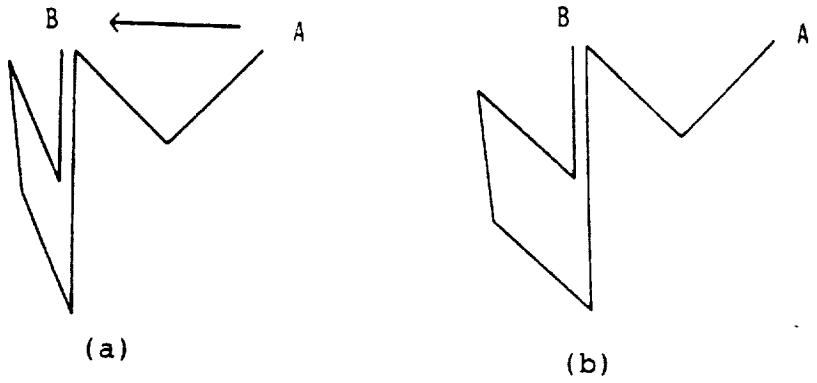


Figure 8. Reach limit avoidance demonstration for the case of two joints violating reach limits. In (a) joints 3 and 4 violate reach limits of -160 and 160. With reach avoidance (b) both reach limits are avoided simultaneously.

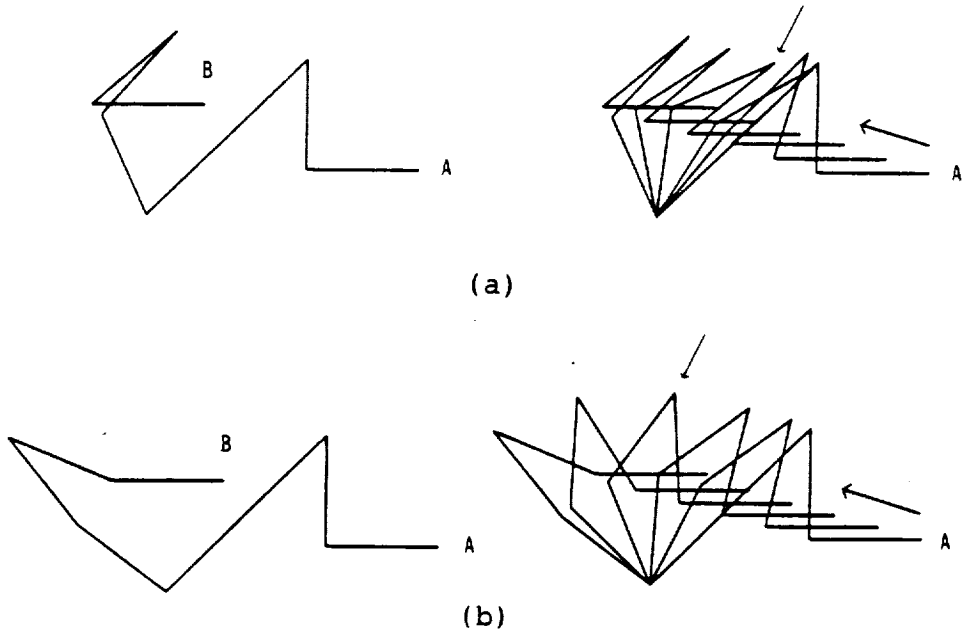


Figure 9. In (a) joint 3 exceeds -180 degrees. This problem is avoided in (b) when reach avoidance is used. The intermediate arm positions are shown to the right.