N88-17268

AN OPTIMAL RESOLVED RATE LAW FOR KINEMATICALLY REDUNDANT MANIPULATORS

McDonnell Douglas Astronautics Co. - Engineering Services
16550 Space Center Blvd., Houston, Tx. 77062

B. J. Bourgeois

ABSTRACT

The resolved rate law for a manipulator provides the instantaneous joint rates required to satisfy a given instantaneous hand motion. When the joint space has more degrees of freedom than the task space the manipulator is kinematically redundant and the kinematic rate equations are underdetermined. These equations can be locally optimized, but the resulting pseudo-inverse solution has been found to cause large joint rates in some cases. In this paper a weighting matrix in the locally optimized (pseudo-inverse) solution is dynamically adjusted to control the joint motion as desired. Joint reach limit avoidance is demonstrated in a kinematically redundant planar arm model. The treatment is applicable to redundant manipulators with any number of revolute joints and to non-planar manipulators.

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], and 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. The matrix that results from differentiating the kinematic equations is 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 spatial 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 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 is not enough information to solve for the joint rates needed to move the hand. In general, there are an infinite number of 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 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 a modified pseudo-inverse technique is used as a control law with joint reach avoidance for the redundant manipulator. This law can be derived by optimizing the sum of the weighted squares of the joint rates. 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.

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:

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

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:

$$\Delta x = [J]\,\Delta\theta \qquad (2)$$

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.

$$v = dx/dt = [J]\,w \qquad (3)$$

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 equation 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.

$$w = [J]^{-1}\,v \qquad (4)$$

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. To solve the set of equations, introduce an objective function to be minimized subject to the constraint equations:

$$v_{6x1} = J_{6x7}\,w_{7x1} \qquad (5)$$

A weighted function of joint angles, proposed by Whitney [1] is:

$$Z = 1/2\,(a_{11}\,w_1^2 + \ldots\; a_{nn}\,w_n^2) \qquad (6)$$

or in matrix notation:

$$Z = 1/2\; w^T A\, w \qquad (7)$$

Using the method of Lagrangian multipliers the solution is:

$$w = A^{-1} J^T (J A^{-1} J^T)^{-1} v \qquad (8)$$

When the weighting matrix is not used, or set to identity, the solution is:

$$w = [\,J^T (J J^T)^{-1}\,]\;v \qquad (9)$$

The expression in brackets in equation 9 is the Moore-Penrose pseudo-inverse of the Jacobian for the underdetermined system of equations 5 [4],[9].

The constraint on the weighting matrix A is such that the function Z must be non-negative for all values of w [11]. This condition will be satisfied by considering only diagonal weighting matrices with positive values. For the case of the seven jointed manipulator described above this resolved rate law is

$$\underset{7x1}{w} = \underset{7x7}{A^{-1}}\;\underset{7x6}{J^T}\;(\underset{6x7}{J}\;\underset{7x7}{A^{-1}}\;\underset{7x6}{J^T})^{-1}\;\underset{6x1}{v} \qquad (10)$$

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

IMPLEMENTATION

The control law expressed by equation 8 was implemented into a simulation of a Kinematically Redundant Planar Manipulator (KRPM). The KRPM model has a task space composed of 3 degrees of freedom: 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. This was done to form a direct analogy with the 7 jointed manipulator operating in a task space of 6 degrees of freedom. The 4 jointed KRPM is shown in Figure 1.

The KRPM model was simulated in FORTRAN on an HP9000 desktop 32 bit super micro computer. 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.

Implementation of the Control Law

The optimal RRL (equation 8) 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.

$$[RRL] = \underset{4x4}{A^{-1}} \; \underset{4x4}{J^{T}} \; (\underset{4x3}{J} \; \underset{3x4}{A^{-1}} \; \underset{4x4}{J^{T}})^{-1}_{4x3} \quad (11)$$

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 (initial θ) and through the forward kinematics the resulting hand state (initial x) is computed. Then an input is made to indicate the desired final hand state for the arm (final x). The difference between the two hand states (Δx) is found.

$$\Delta x = x_{final} - x_{initial} \quad (12)$$

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 11 and then the desired joint angle step Δθ is computed.

$$\Delta\theta = [RRL] \, \Delta x/s \quad (13)$$

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 (Δx). 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.

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. A ten step iteration results in a reasonably straight hand trajectory, but eighty steps were required for very good results.

BEHAVIOR OF THE OPTIMAL RESOLVED RATE LAW

The ability of the rate law to handle the redundancy of the arm was demonstrated by driving the arm to the same hand state from various starting configurations. This also serves as an inverse solution to the kinematics, by providing several possible joint sets that satisfy the requested hand state. In Figure 2 the end-effector was commanded to the state X=3, Z=0, and Pitch= 0 (3,0,0) from four 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 11 equal to identity. This is the equivalent of the pseudo-inverse solution.

The Effect of the Weighting Matrix

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 3 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 (pseudo-inverse), the final value of the second joint is undesirable (Figure 3a). 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 noticeably better (Figure 3b). Joint 2 moved less from start to finish than in Figure 3a. 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 3a the final condition of the arm is not desirable 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 3d, the final situation is much more desirable, because joint 2 has more freedom to move around in the neighborhood of the final end-effector state.

REACH AVOIDANCE ALGORITHMS

The behavior of the pseudo-inverse of equation 9 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 is a complicated treatment. A simpler approach is taken here which makes use of the weighting matrix A in the RRL of equation 11.

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 desirable 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 repeatedly 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 algorithm 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.

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 mid-range 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 simultaneously.

To test the ability to avoid a single joint reach limit, a case was taken 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. In the trajectory that was selected 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 4 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. This is generally not desirable because the loss of motion in this joint removes the redundancy of the arm. In methods 2 and 3, the joint moved back out of the reach zone of 10 degrees from the reach limit. Methods 2 and 3 allow the joint to maintain motion.

In Figure 5a 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 5b).

In the case shown in Figure 6a joint 3 exceeds a -160 degree limit and then goes past -180. With reach avoidance (Figure

6b) joint 2 swings out dramatically to allow joint 3 to avoid its reach limit.

SUMMARY

The pseudo-inverse Jacobian with a weighting matrix has been implemented with reach avoidance capability into a simulation of a kinematically redundant planar manipulator. This optimal resolved rate law has been demonstrated in the planar model and reach avoidance has been mostly successful with this model by dynamically adjusting the components of the weighting matrix during maneuvers. Three reach avoidance algorithms were tested. The locally optimized resolved rate law has been improved by incorporating joint reach avoidance.

REFERENCES

1. Whitney, D. E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," J. of Dynamic Systems, Measurement and Control," Trans. ASME, Vol. 92, Dec. 1972, pp. 303-309.
2. Craig, J. J., Introduction to Robotics, Addison-Wesley, 1986.
3. Benhabib, B., Goldenberg, A.A., and Fenton, R. G., "A Solution to the Inverse Kinematics of Redundant Manipulators," J. of Robotic Systems, Vol. 2, No. 4, 1985, pp. 373-385.
4. Klein, C.A., and Huang, C. "Review of Pseudoinverse Control for Use with Kinematically Redundant Manipulators," IEEE Trans. Systems, Man, and Cybernetics, Vol SMC-13, No. 3, March/Apr 1983.
5. Oh, S., Orin, D., Bach, M., "An Inverse Kinematic Solution for Kinematically Redundant Robot Manipulators," J. of Robotic Systems, Vol. 1, No. 3, 1984, pp. 235-249.
6. Goldenberg, A. A., Benhabib, B., Fenton, R.G., "A Complete Generalized Solution to the Inverse Kinematics of Robots", IEEE Journal of Robotics and Automation, Vol RA-1, No. 1, March, 1985, pp. 14-20.
7. Benati, M., Morasso, P., Tagliasco, V., "The Inverse Kinematic Problem for Anthropomorphic Manipulator Arms," J. of Dynamic Systems, Measurement, and Control, Vol. 104, March, 1982, pp. 110-113.
8. Paul, R. P., Robot Manipulators: Mathematics, Programming and Control, MIT Press., Cambridge, MA, 1981.
9. Forsythe, Malcom, and Moler, Computer Methods for Mathematical Computation, Prentice-Hall, NJ, 1977.
10. Whitney, D.E. "Optimum Stepsize Control for Newton-Raphson Solution of Nonlinear Vector Equations," IEEE Transactions on Automation Control, Vol. AC-14, No. 5, Oct., 1969, pp.572-74.
11. Zahradnik, Raymond L., Theory and Techniques of Optimization For Practicing Engineers, Barnes & Nobles, NY, 1971.

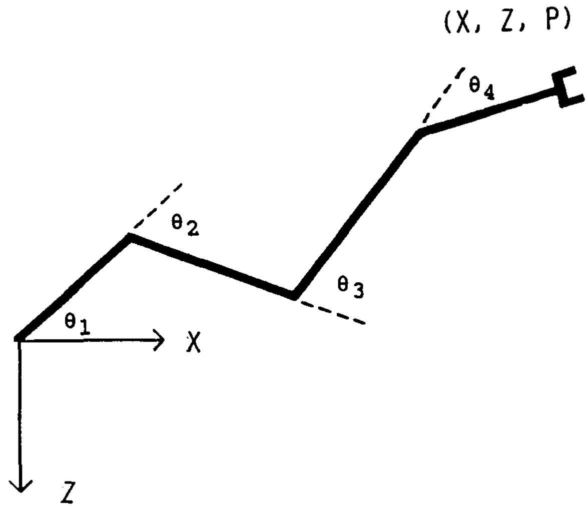


Figure 1 - A kinematically redundant planar manipulator with four joints and a task space of X, Z, and Pitch (P).

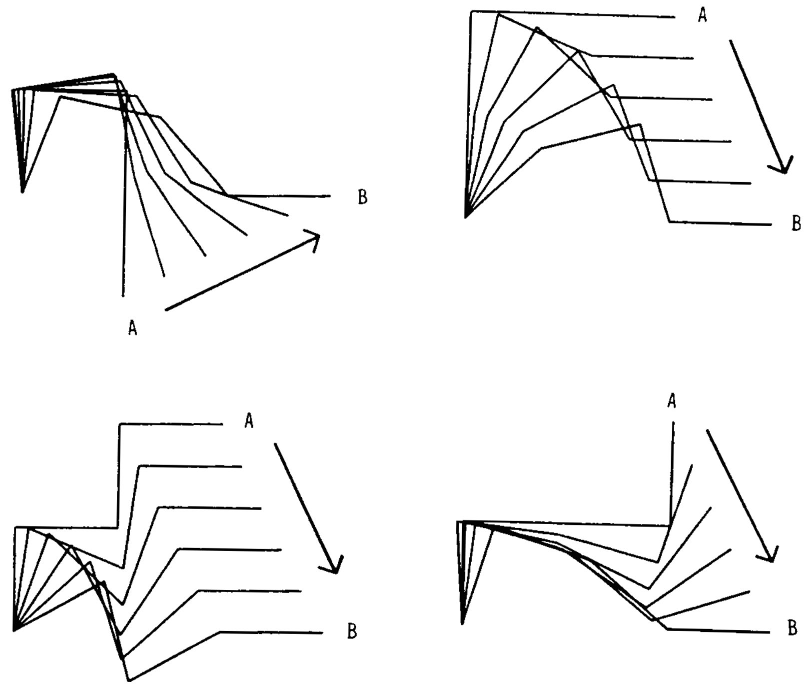


Figure 2 - The KRPM is commanded to the same hand state of (3,0,0) at B from four different initial joint angle states at A to demonstrate the behavior of the RRL.

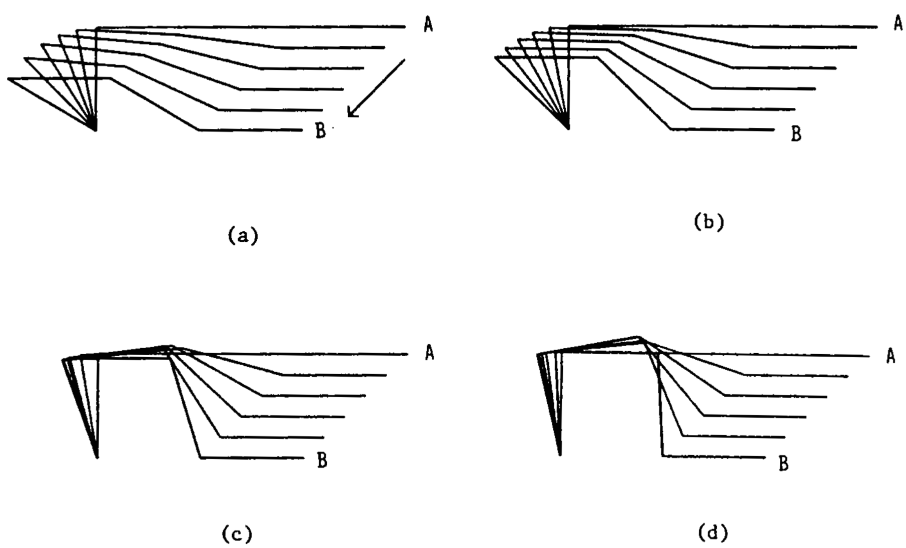


Figure 3 - Effects of the weighting matrix on the trajectory are shown for values of A(2,2) set at 1 (a), 2 (b), 10 (c), and 100 (d).

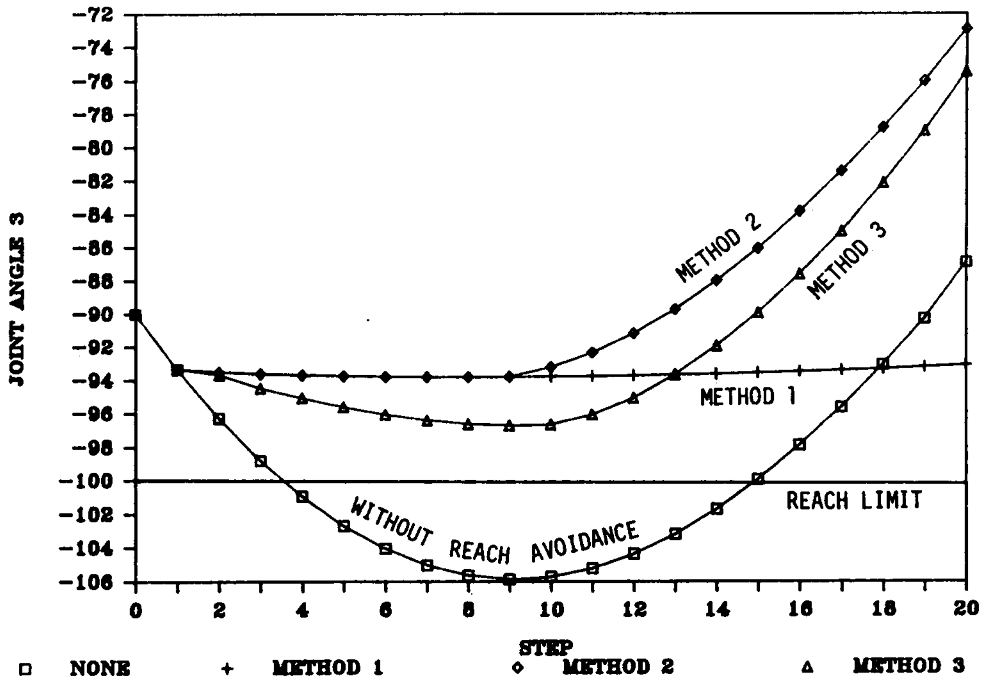


Figure 4 - The effects of reach avoidance on joint 3 are shown during a command from a joint state of (90,0,-90,0) to a hand state of (3,0,0). The reach limit is successfully avoided for each of the 3 reach avoidance algorithms.

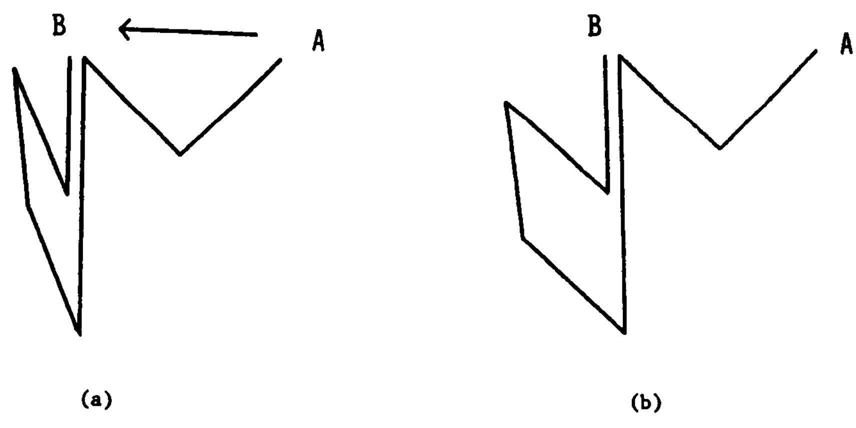
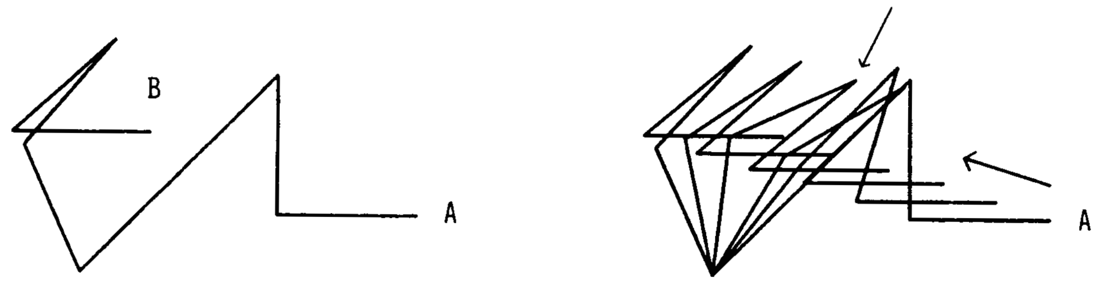
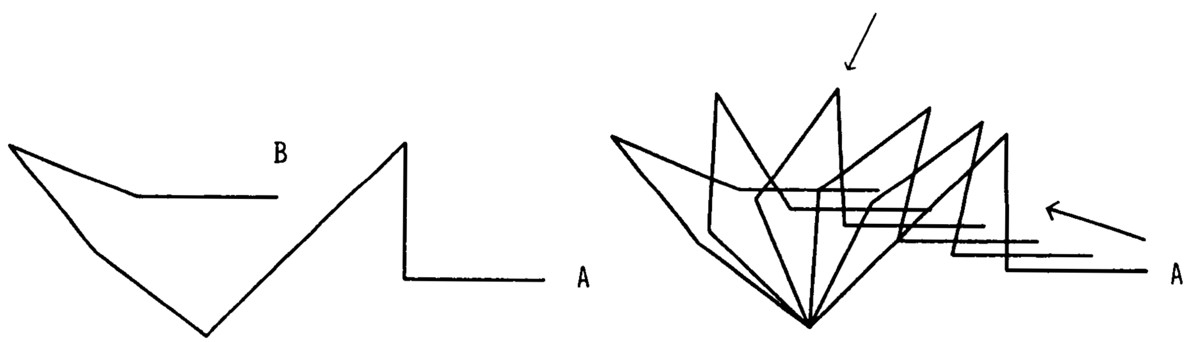


Figure 5 - Reach limit avoidance demonstration for the case of two joints violating reach limits. In (a) joints 3 and 4 violate reach limits of -160 and 160. With reach avoidance (b) both reach limits are avoided simultaneously.



(a)



(b)

Figure 6 - 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.