

ON THE USE OF REDUNDANCY IN ROBOT KINEMATIC CONTROL

Lorenzo Sciacvico , Bruno Siciliano and Pasquale Chiacchio

Dipartimento di Informatica e Sistemistica
Università di Napoli
Via Claudio 21, 80125 Napoli, Italy

Abstract

Redundancy offers a potential over nonredundancy in robot design and control in terms of greater motion/force manipulability. The solution to the kinematic control problem is crucial for an effective use of redundancy. The redundant degree-of-freedom's (DOF's) can be exploited to meet additional motion/force constraints, such as dexterity, task compatibility, obstacle avoidance, and limited joint range. The augmented task space approach is adopted and a recently established computational method to solve the constrained inverse kinematic problem is proposed. Finally, a case study for a snake-like robot operating in a constrained environment is developed and simulation results are included.

Introduction

The design of redundant manipulator geometries has been lately receiving a considerable interest among roboticists. The key point is the potential advantage offered by a redundant structure over a nonredundant structure, in terms of increased motion/force manipulability. Several indices of this ability have been recently established in the literature [1-7]. In this scenario the solution to the robot kinematic control problem appears to be the drawback to overcome for an effective use of redundant manipulators. In fact, the extra degree-of-freedom's gained with a redundant design may concur to meet additional constraints on the solution to the inverse kinematic problem. Typical constraints are obstacle avoidance [1-3], limited joint range [4], dexterity measures [5,6], and task compatibility indices [7]. This issue is discussed in this paper and the augmented task space approach is followed [8,9], that is the task space which determines the number of redundant DOF's is suitably augmented to include the above kinds of constraints. The solution to the constrained inverse kinematic problem is provided by a recently established algorithm which avoids any matrix inversion [10,11]. An interesting case study is worked out. This concerns a snake-like planar robot (many DOF's) operating in a constrained environment; it is shown how the end-effector can automatically be positioned inside while avoiding link collision with the boundaries of the environment. Simulation results illustrate the effectiveness of the proposed approach.

Direct Kinematics and Statics

For any manipulator with known geometrical dimensions, the direct kinematic equation specifies the relation between the $(n \times 1)$ joint vector $\underline{q}$ and the $(m \times 1)$ task vector $\underline{x}$ as [12]

$$\underline{x} = \underline{f}(\underline{q}) \quad (1)$$

where $\underline{f}$ is a continuous nonlinear function which associates to each $\underline{q}$ a unique $\underline{x}$.

Differentiating eq. (1) with respect to time yields the relation between the joint velocity vector $\dot{\underline{q}}$ and the task velocity vector $\dot{\underline{x}}$, through the $(m \times n)$ Jacobian matrix $J(\underline{q}) = \partial \underline{f} / \partial \underline{q}$ [12], i.e.

$$\dot{\underline{x}} = J(\underline{q})\dot{\underline{q}}. \quad (2)$$

The direct static equation for a manipulator specifies the relation between the $(m \times 1)$ end-effector force vector $\underline{y}$ and the $(n \times 1)$ joint torque vector $\underline{\tau}$, through the transpose of the Jacobian matrix [12], i.e.

$$\underline{\tau} = J^T(\underline{q})\underline{y}. \quad (3)$$

If the manipulator is redundant with respect to a certain task, it is $m < n$. Assuming that the Jacobian matrix has full rank m for almost all $\underline{q}$'s, one has $(n - m)$ redundant DOF's available. If for some $\underline{q}$ the Jacobian has rank less than m , the manipulator is said to be at a singular configuration. In this configuration the manipulator loses its ability to move along or rotate about some direction of the space, meaning that its manipulability is reduced [12].

Definition of Constraints

As discussed above, redundancy can be conveniently exploited to meet additional motion/force constraints, so as to obtain greater manipulability in terms of manipulator kinematical configuration and interaction with the environment. To this purpose, the human arm constitutes a tangible model of this ability [1].

If the robot is required to move in a cluttered environment, obstacle avoidance and limited joint range represent two types of constraints to account for in the trajectory planning and inverse kinematics solving. Assume that a manipulator is tracking a desired collision-free

end-effector trajectory. One or more links may happen to be too close to an obstacle in the workspace, and a collision is expected (Fig. 1). Obstacles can be supposed to be modeled as convex volumes. According to the approach proposed in [2,3], the distance (ld 1) between the link candidate to a collision (point p_0) and the obstacle (point c) has to be computed. This distance has to be maintained at a value greater than a given threshold [10]. Similarly, in case a joint limit q_i is encountered [4], the current q_i has to be braked and kept far enough from the limit [10].

The constraints which characterize the manipulator dexterity can be nicely defined by following the kineto-static formulation given in [7]. There the manipulator is viewed as a mechanical transformer with joint velocity and torque as input and task velocity and force as output. From eq. (2), it can be said that at a given configuration q , the unit sphere in R^n defined by

$$\dot{q}^T \dot{q} \leq 1 \quad (4)$$

maps into the velocity ellipsoid in R^m defined by

$$\dot{x}^T (J(q)J^T(q))^{-1} \dot{x} \leq 1. \quad (5)$$

Similarly from eq. (3), it can be said that at a given configuration q , the unit sphere in R^n defined by

$$\tau^T \tau \leq 1 \quad (6)$$

maps into the force ellipsoid in R^m defined by

$$y^T (J(q)J^T(q)) y \leq 1. \quad (7)$$

It can be easily proved that the principal axes (eigenvectors) of the velocity and force ellipsoids coincide, whereas the axis lengths (eigenvalues) are in inverse proportions [7].

The volume of the velocity ellipsoid (5) can be used as an effective means for singularity avoidance. The manipulability measure at a configuration q is defined as [5]

$$w(q) = [\det J(q)J^T(q)]^{\frac{1}{2}} \quad (8)$$

and is proportional to the above volume. At a singular configuration q it is obviously $w(q) = 0$. A dynamic manipulability measure is also defined in [6] which is related to the inertia matrix of the manipulator.

The force ellipsoid is at the basis of the definition of the condition number [1]

$$\kappa(q) = \frac{\lambda[J(q)J^T(q)]}{\Lambda[J(q)J^T(q)]} \quad (9)$$

where Λ and λ respectively denote the largest and the smallest eigenvalues of the matrix JJ^T . While a determinant going to zero marks the presence of a singularity as in (8), the actual value of the determinant is not a practical measure of the degree of ill-conditioning. Instead, the condition number gives a measure of closeness of the force ellipsoid to a sphere. At an isotropic configuration q it is $\kappa(q) = 1$, that is the manipulator will exert end-effector forces of equal magnitude in all directions. Also, on the basis of the above duality between force and velocity

ellipsoids, an isotropic configuration will present end-effector velocities of equal magnitude in all directions.

While the previous two measures quantify the natural (dexterous) configuration of a manipulator, different indices have been recently proposed in [7] to quantify the compatibility of a configuration with respect to a given task. In other words, it seems appropriate to distinguish between fine manipulation tasks, where accurate control of small velocity/force is required, and coarse manipulation tasks, where exertion of large velocity/force is required. Therefore, the force transmission ratio along a given direction u can be defined as the distance from the center to the surface of the force ellipsoid along u

$$\alpha(q) = [u^T (J(q)J^T(q)) u]^{-\frac{1}{2}} \quad (10)$$

and dually the velocity transmission ratio along a direction v as

$$\beta(q) = [v^T (J(q)J^T(q))^{-1} v]^{-\frac{1}{2}}. \quad (11)$$

It follows that the best direction for effecting velocity (maximum β) is also the best direction for controlling force (minimum α). Dually, the best direction for effecting force (maximum α) is also the best direction for controlling velocity (minimum β). The two indices (10) and (11) can be also combined into one compatibility index $c = (1/\alpha\beta)$ as a quantitative measure of good control (exertion) compatibility in a given direction.

The Inverse Kinematic Algorithm

It is well known that the solution of the inverse kinematic problem, i.e. given x solve eq. (1) for q , is fundamental in robot manipulator control. In case of redundant manipulators, the approach is based on the inversion of the mapping (2). It can be shown that the general solution to (2) is given by

$$\dot{q} = J^+(q)\dot{x} + [I - J^+(q)J(q)]\dot{q}_0 \quad (12)$$

where J^+ is an $(n \times m)$ generalized inverse of matrix J , I is the $(n \times n)$ identity matrix, and $\dot{q}_0$ is an $(n \times 1)$ arbitrary joint velocity vector. It is worth noticing that the solution (12) composes of the least-square term of minimum norm [12] plus the homogeneous term created by the projection operator $(I - J^+J)$ which selects the components of $\dot{q}_0$ in the null space of the mapping J . Hence, the vector $\dot{q}_0$ can be used to "locally" optimize [8] some additional criterion based on the constraints defined in the previous section [1-7].

A rather different approach to the inverse kinematic problem for redundant manipulators is given by a recently proposed general solution algorithm which is based on a dynamic reformulation of the problem [13,14]. The method is summarized in the following.

Let $\hat{q}$ be a solution to (1) relative to a given end-effector location $\hat{x}$. A task error vector e can be defined between the reference vector $\hat{x}$ and the actual vector x computed from the current joint vector q via eq. (1),

$$e = \hat{x} - x. \quad (13)$$

Here the explicit dependence on time is not shown. However, differentiating with respect to time yields

$$\dot{\underline{e}} = \dot{\underline{x}} - J(q)\dot{q}. \quad (14)$$

It can be proved [13,14] that the choice

$$\dot{q} = KJ^T(q)\underline{e} \quad (15)$$

ensures that the tracking error is bounded and the positional error ($\underline{x} = 0$) is zero, by means of a suitable selection of the positive definite matrix K . The resulting closed-loop scheme is illustrated in Fig. 2. Attractive features of the scheme are that only direct kinematic functions (f, J) must be computed and the joint velocity vector $\dot{q}$ is inherently generated by the algorithm. The solution (15) apparently serves as inverse kinematic solution for a general unconstrained redundant manipulator. It will be shown in the following section how constraints can be embedded in the solution.

Task Space Augmentation

As discussed above, a redundant arm offers the possibility of setting a number of constraints on its manipulation, which make the structure more versatile. The task augmentation approach is an alternative strategy to the usual methods based on the use of generalized inverses, see eq. (12), for which the constraints are incorporated into the homogeneous solution term [1-7]. Shortcomings of the usual method are that repetitive motions in the task space need not involve repetitive paths in the joint space [8,9], and kinematic singularities are not avoided in any practical sense, since the joint velocity are minimized only instantaneously [8,9].

The main idea of the task augmentation approach is to impose a number of functional constraints in terms of the joint vector, expressed in the same base frame associated with the manipulator direct kinematic equation (1), as [8]

$$\underline{0} = \underline{h}(q) \quad (16)$$

where $\underline{h}(q)$ is an $(r \times 1)$ vector, with $r \leq (n - m)$, so as cover at most all redundant DOF's. Therefore, the augmented kinematic equation results

$$\begin{bmatrix} \underline{x} \\ \underline{0} \end{bmatrix} = \begin{bmatrix} \underline{f}(q) \\ \underline{h}(q) \end{bmatrix} \quad (17)$$

whose joint vector solution q not only places the end-effector at the desired location $\underline{x}$, but also meets the required constraints, specified by $\underline{h}$ in implicit form. Also, differentiating both sides of (12) with respect to time provides the mapping of the joint velocity vector into the augmented task velocity vector, i.e.

$$\begin{bmatrix} \dot{\underline{x}} \\ \underline{0} \end{bmatrix} = \begin{bmatrix} J(q) \\ \frac{\partial \underline{h}(q)}{\partial q} \end{bmatrix} \dot{q} \quad (18)$$

where the matrix

$$J_a(q) = \begin{bmatrix} J(q) \\ \frac{\partial \underline{h}(q)}{\partial q} \end{bmatrix} \quad (19)$$

is called the augmented Jacobian matrix.

At this point, it is clear that all kinds of constraints defined above can be suitably described by the vector $\underline{h}$ in (17). Further, the algorithm based on eq. (15) can be successfully adopted to solve the inverse kinematic problem for constrained redundant manipulators, on condition that the matrix J is suitably replaced by the matrix J_a defined in (19). The inclusion of constraints on obstacle avoidance and limited joint range is described in [10], whereas the inclusion of constraints on dexterity measures is reported in [11]. As regards the latter point, in [8] it has been noticed that the augmented Jacobian J_a in (19) will present not only the same singularities as those of the end-effector Jacobian J in (2), but also other singularities due to the addition of the r rows. If the augmented task space error happens to span the null space of J_a , the solution of type (15) gets stuck and no further improvement of the manipulability measure is possible, along with the fact that a position error may occur. In other words, in order to guarantee that the additional constraints are exactly met, one should ensure that the vector $\underline{h}$ in (17) is obtained by projecting the given constraints onto the null space of the end-effector Jacobian J [8].

A Case Study

One of the potential advantages of a kinematically redundant manipulator is the use of the redundant DOF's to manoeuvre in a complex workspace and avoid contact with obstacles. Assume that a manipulator is tracking a desired collision-free end effector trajectory in the task space. If one or more links along its kinematical structure happen to be too close to an obstacle in the workspace, a collision can be expected; then one or more constraints need to be introduced in order to avoid collisions with the obstacle.

In order to present the performance of the proposed inverse kinematic algorithm for constrained redundant manipulator, a case study is developed in the following. A planar seven DOF's manipulator is considered. The manipulator task consists of reaching a desired end effector position, located inside a constrained environment which is constituted by a torus. The manipulator is assumed to start from a joint configuration which places the end effector outside the torus.

The insertion strategy imposes that once the joints (that is their cartesian positions) enter the torus, they are constrained to belong to a geometric locus that assures the absence of collisions between manipulator links and torus inner walls. The chosen locus is a circumference of radius $(r+R)/2$, concentric with the two circumferences of radii r and R which mark the torus boundaries in the horizontal plane.

As the degree of redundancy is equal to five, not more than five constraints can be added to define the augmented task space. In view of the described task, forcing to zero the distance between cartesian joint position and the geometrical locus can be assumed as an attractivity constraint; it can be written as:

$$h_i(q) = \|x_c - f_i(q)\| = 0, \quad i=2,8 \quad (20)$$

$$q = (q_1, \dots, q_{i-1})$$

where x_c represents the torus center coordinates, and $f_i(q)$ is the direct kinematic function which characterizes the cartesian position of the i -th joint. In the planar case it can be written as:

$$f_i(q) = \begin{bmatrix} f_{ix}(q) \\ f_{iy}(q) \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{i-1} l_j c_{1j} \\ \sum_{j=1}^{i-1} l_j s_{1j} \end{bmatrix} \quad (21)$$

where $c_{1k} = \cos(q_1 + \dots + q_k)$ and $s_{1k} = \sin(q_1 + \dots + q_k)$. In relation with the considered structure, the subscript $i=8$ indicates the end effector.

Denoting with p_i the position of the i -th joint in a cartesian reference frame, the augmented kinematic equation is

$$\begin{bmatrix} p_i \\ 0 \end{bmatrix} = \begin{bmatrix} f_i(q) \\ h(q) \end{bmatrix} \quad (22)$$

where $h(q) = \delta \cdot h_i(q)$ is a vector whose maximum dimension is five and whose components are $h_i(q)$; it is defined by suitably selecting five attractivity constraints. The scalar δ , will be 1 or 0 depending if the corresponding constraint is active or not.

It must be emphasized that an insertion strategy which "pulls" the end effector inside the torus (by assigning a trajectory to p_8) and successively activates the attractivity constraints on the joints 7,6,5... is problematic; it easily happens that the position constraints on the end effector and the attractivity constraints on the other joints induce contrasting actions on the joint variables; in other words, given the complexity of both the task and the structure of the manipulator, algorithmic singularities can frequently occur, which are due to the augmented rows in (22). When such a situation arises, the position error grows and instability can appear because of the discrete implementation of the inversion algorithm (15).

Conversely, a more efficient strategy has been chosen, which "pushes" the manipulator inside the torus: the joints are successively driven inside the torus and then the attractivity constraints are activated simultaneously relaxing the relative positional constraint. Moreover the attractivity constraint on the i -th joint is supposed to be satisfied by only varying the $(i-1)$ th joint variable. Thus the algorithmic singularities are automatically avoided because the augmented Jacobian matrices will result block-diagonal.

This procedure will be described following a list of sequential subtasks, each of one brings a joint external to the torus in the point $\bar{P}$ (see fig.3) inside the torus and activates the attractivity constraint on the joint inside the torus.

subtask	positional constraint	attractivity constraint
1	$p_8(q_6, q_7)$	none
2	$p_7(q_5, q_6)$	$h_8(q_7)$
3	$p_6(q_4, q_5)$	$h_8(q_7), h_7(q_6)$
...	...	...
6	$p_3(q_1, q_2)$	$h_8(q_7), h_7(q_6), h_6(q_5),$ $h_5(q_4), h_4(q_3)$

The notation $p_j(q_{i-1}, q_{i-2})$ means that the j -th joint trajectory tracking is obtained using only the $(j-1)$ th and the $(j-2)$ th joint variables; with $p_j(q_{i-1})$ is denoted that the j -th joint is forced to belong to the geometric locus only by using the $(j-1)$ th joint variable.

During the execution of the generic subtask, the augmented kinematic function is:

$$\begin{bmatrix} p_i(t) \\ 0 \end{bmatrix} = \begin{bmatrix} f_i(q_1, \dots, q_{i-1}) \\ h_{i+1}(q_1, \dots, q_i) \\ \vdots \\ h_8(q_1, \dots, q_7) \end{bmatrix} \quad (23)$$

The associated jacobian matrix is obtained by differentiating the (23) with respect to the joint variables chosen to satisfy each constraint. It results:

$$J = \text{diag} \{ J_i, \partial h_{i+1} / \partial q_i, \dots, \partial h_8 / \partial q_7 \} \quad (24)$$

where $J_i = \partial f_i / \partial q_{i-1}, \partial q_{i-2}$ represents the jacobian matrix relative to the positioning of the i -th joint. From (23) and (24), it can be argued that the dimensionality of the algorithm progressively increases from 2 to 7, in passing from subtask 1 to 6. In each subtask algorithmic singularities cannot occur at all, due to the diagonal structure of the associated jacobian matrix; the kinematic singularities, relative to the positioning task (that is rank $J_i < 2$) can be avoided by suitably locating the manipulator with respect to the torus.

In fig.3a the starting configuration is depicted. The manipulator with 7 equal links of length 1.236 m, is drawned up. The initial conditions on the joints are: $q_1(0) = 0^\circ$, $q_2(0) = 175^\circ$, $q_3(0) = -175^\circ$, $q_4(0) = 175^\circ$, $q_5(0) = -175^\circ$, $q_6(0) = 175^\circ$, $q_7(0) = -175^\circ$.

The torus, with an inner radius $r = 1.75$ m and an outer radius $R = 2.25$ m has center $x_c = 0.618, 4$. in the cartesian frame used in describing the kinematic of the manipulator. The frame origin is assumed to be on the manipulator base.

In Fig.3b an intermediate situation in the execution of subtask 3 is depicted.

In Fig.3c the final configuration of the manipulator is shown; the end effector has reached a point diametrically opposite to $\bar{P}$.

In Fig.4 the joint variables are plotted, corresponding to the accomplishment of the insertion task.

The "extraction" of the manipulator from the torus can be obtained with a sequence of suitable subtasks following a strategy inverse with respect to that one previously described.

Conclusions

A new solution to the inverse kinematic problem for redundant manipulators has been presented. It is based on a closed-loop algorithm whose convergence is assured by choosing a control function which only involves the computation of the direct kinematics of the manipulator.

The redundant degree of freedom's can be exploited to meet additional motion/force constraints, such as manipulability, task compatibility and obstacle avoidance. The direct kinematics can be appropriately augmented by including the chosen mentioned constraints; the result is an efficient algorithm which enables the designer to make the most of redundant manipulator versatility.

The problem of fitting a 7 DOF's planar manipulator in a torus has been considered; it has been shown that, by conveniently modelling the insertion strategy, the solution algorithm give rise to joint variables trajectories which produce collisions-free motions inside the torus.

References

- [1] C.A. Klein, "Use of redundancy in the design of robotic systems," 2nd Int. Symp. Rob. Res., Kyoto, Japan, Aug. 1984.
- [2] A.A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," Int. J. Rob. Res., 4, 3, 1985.
- [3] M. Kircanski and M. Vukobratovic, "Trajectory planning for redundant manipulators in the presence of obstacles," 5th CISM-IFTOMM Ro.Man.Sy, Udine, Italy, June 1984.
- [4] M. Vukobratovic and M. Kircanski, "A dynamic approach to nominal trajectory synthesis for redundant manipulators," IEEE Trans. Syst., Man, Cybern., 14, 4, 1984.
- [5] T. Yoshikawa, "Manipulability of robotic mechanisms," Int. J. Rob. Res., 4, 2, 1985.
- [6] T. Yoshikawa, "Dynamic manipulability of robot manipulators," J. Rob. Syst., 2, 1, 1985.
- [7] S.L. Chiu, "Control of redundant manipulators for task compatibility," IEEE Int. Conf. Rob., Autom., Raleigh, NC, March/April 1987.
- [8] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," IEEE Int. Conf. Rob., Autom., St. Louis, MO, March 1985.
- [9] J. Baillieul et al., "Kinematically redundant robot manipulators," Workshop Space Telerob., JPL, Pasadena, CA, Jan. 1987.
- [10] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," IEEE J. Rob., Autom., to be published, 1988.
- [11] L. Sciavicco and B. Siciliano, "An inverse kinematic solution algorithm for dexterous redundant manipulators," 3rd Int. Conf. Adv. Rob., Versailles, France, Oct. 1987.
- [12] J.J. Craig, Introduction to Robotics: Mechanics and Control. Reading, MA: Addison-Wesley, 1986.
- [13] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," IEEE Trans. Syst., Man, Cybern., 16, 4, 1986.

- [14] A. Balestrino, G. De Maria, L. Sciavicco, and B. Siciliano, "An algorithmic approach to coordinate transformation for robotic manipulators," Advanced Robotics, to be published, 1988.

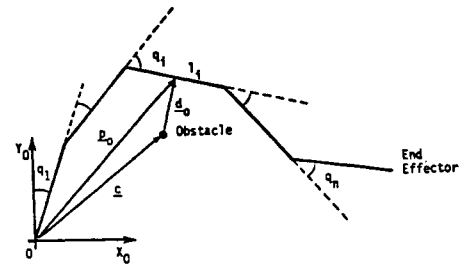


Fig. 1 - Geometry of a planar manipulator showing the interaction with the obstacle.

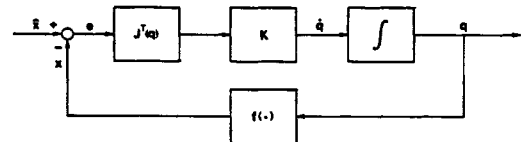


Fig. 2 - The general closed-loop inverse kinematic scheme.

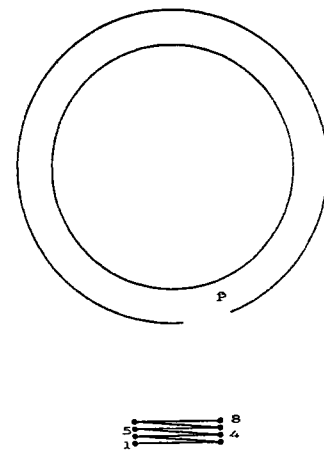


Fig. 3a - Case study: starting configuration.

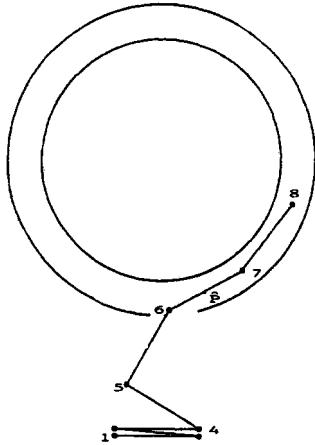


Fig. 3b - Case study: intermediate configuration during the execution of subtask 3.

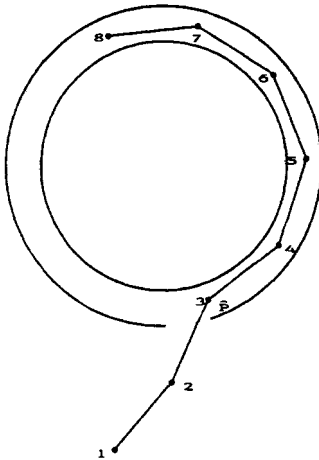


Fig 3c - Case study: final configuration.

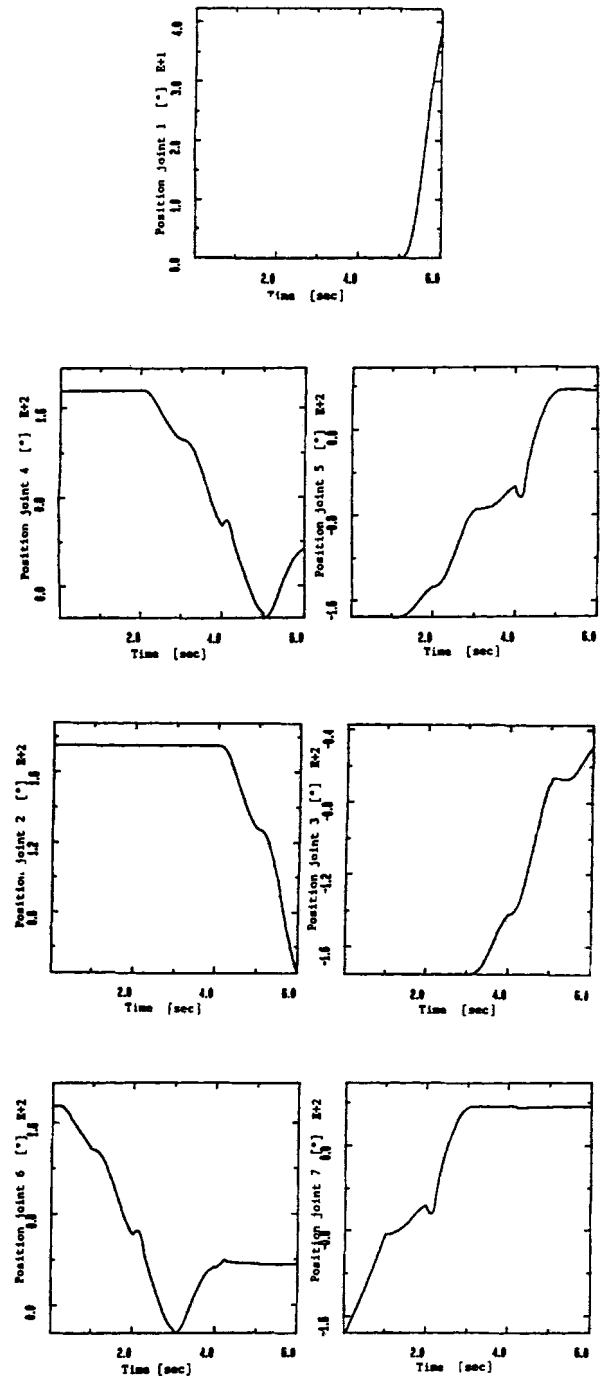


Fig. 4 - Joint variable trajectories.