A General Framework for Managing Multiple Tasks in Highly Redundant Robotic Systems

Bruno Siciliano
Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125 Napoli, Italy

Jean-Jacques E. Slotine
Nonlinear Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, USA

Abstract—**The exploitation of kinematic redundancies in robotic systems may provide more dexterity and versatility in the execution of complex tasks. When functional constraint tasks are imposed in addition to the end-effector task, a task priority strategy is advisable. In this paper, we propose a general framework for managing multiple tasks in highly redundant systems. In particular, we derive joint velocity and acceleration solutions which can be used as reference input trajectories to suitable model-based controllers. We also develop a recursive implementation, and discuss the occurrence of singularities in the Jacobian associated with the generic task. Two case studies illustrate the effectiveness of the algorithm on a snake-like robot.**

I. INTRODUCTION

A manipulator is said to be kinematically redundant when more degrees of freedom than the minimum number required to execute a given task are available. This definition implies that redundancy can be established only with respect to some particular task. For instance, a six-degree-of-freedom manipulator is redundant with respect to all those five-dimensional end-effector tasks, such as arc welding, laser cutting, spray painting, for which the specification of the sixth roll angle is of no concern.

Redundant degrees of freedom can be conveniently exploited to meet a number of functional constraints which will eventually bestow more dexterity and versatility to the robot in terms of its interaction with the environment. Joint range availability [1], singularity avoidance [2], obstacle avoidance [3], compliant motion [4] are only some examples of constraint tasks which one may wish to be fulfilled along with an original end-effector task. The result is an augmented task space [5,6] to deal with for control purposes.

In order to solve the conflicting task situations, that would normally occur if the constraint tasks are specified irrespectively of the original task, the so-called task priority strategy, originally proposed in [7] and later developed in [3,8] is advisable. This imposes that the lower priority task is satisfied only in the null space of the higher priority task, the concept being extensible to multiple tasks.

This work was supported in part by Consiglio Nazionale delle Ricerche under Contract 90.00400.PF67 and in part by the National Science Foundation under Grant 8803767-MSM.

On the other hand, the implementation of robot controllers directly in the task space usually requires the solution of an inverse kinematic problem. If some joint space controller has been designed, in fact, an inverse kinematic solution is needed in order to provide the controller with the capability of using task feedback data in real-time [9]. The inverse of the manipulator Jacobian serves this purpose in the case of a nonredundant structure, whereas the solution can be obtained via the use of the generalized inverse in the redundant case.

This paper is aimed at establishing a systematic framework for managing multiple tasks in redundant systems, which is logically derived from the aforementioned task priority strategy. The generic task is assumed to be functionally expressed in terms of the robot joint variables and is in turn characterized by its Jacobian with respect to the proper joint variables.

We remark that, differently from the general formulation in [8] and from most common task space control techniques, we do not solve the inverse kinematics at the acceleration level for designing a computed-torque control, but we rather solve it at the velocity level, in order to avoid the problem of internal unstable behavior observed in [10]. The joint acceleration solutions are then derived by direct time-differentiation of the joint velocities. If both solutions are substituted in a model-based control scheme which exploits the linearity of the manipulator dynamics in terms of a set of dynamic parameters, e.g. the adaptive control scheme proposed in [11], global stable behavior can be obtained, and the above drawback can be overcome.

Remarkably, the solution is cast in a recursive fashion. This simplifies the problem of high level programming of redundant robotic systems, allowing the user to successively specify as many tasks as desired. Moreover, a number of considerations regarding the occurrence of task singularities as well as of algorithmic singularities are provided; the latter being generated by conflicts between the generic task and tasks of higher priority.

The developments are numerically illustrated for a snake-like planar robot in two case studies with different types of constraints.

II. THE MAIN RESULT

Robot control actions are naturally operated in the joint space. In most practical applications, however, it is desired to design a task space controller which uses Cartesian feedback data. This allows the robot to modify its behavior in real-time on the basis of sensory information, including e.g. obstacle avoidance, compliant motion. In order to design a task space controller, the solution of an inverse kinematic problem is needed. This poses interesting problems for a redundant manipulator.

One effective technique of solving redundancy is to extend the dimension of the original (typically, end-effector) task space by imposing a number of functional constraints of the joint variables, namely a task space augmentation [5,6]. The solution of the so-derived augmented inverse kinematic problem requires the joint variables — position, velocity and acceleration, in general — to satisfy not only the original task, but also the constraint task.

A "rough" task space augmentation, however, may generate conflicting task situations in which neither of the two tasks is executed satisfactorily. A remedy for this inconvenient is the so-called task priority strategy [3,7,8], according to which a priority between the two tasks is established beforehand, and the lower priority task produces only a self-motion which does not interfere with the higher priority task.

In the case of a highly redundant system, i.e. having many redundant degrees of freedom, it would be possible to introduce multiple constraint tasks of different nature, and then decide an order of priority between them. In the following, we present a general framework for achieving multiple tasks which is logically based on the idea of multiple task priorities.

A major difference between the proposed technique and the general formulation in [8] is explained in the following: In [8] joint acceleration solutions are derived from the second-order differential kinematics and then substituted in the robot dynamic model for designing a computed-torque control; this is also common to most task space control techniques for redundant manipulators. It was shown in [10] that this method may lead to internal unstable behavior for a redundant manipulator. Possible remedies against this incovenient were proposed respectively in [12] and [13] where control of joint velocities describing the self-motion is provided.

This inconvenient can be eliminated if one resorts to the adaptive control scheme, originally proposed in [11] for nonredundant manipulators and later extended in [14] for redundant manipulators, which guarantees global tracking convergence of end-effector trajectories. By exploiting the linearity of the manipulator dynamics in terms of a suitable set of dynamic parameters, the control law makes use of both reference velocity and acceleration inputs, besides the actual measurements. Of course, the actual joint position solution will be determined by the dynamics of the system under the above control.

Referring the reader to [11,14] for details about the adaptive control algorithm, below we focus the discussion only on redundancy issues.

Consider a generic i-th task to be characterized by the differential kinematic equation

$$\dot{\mathbf{x}}^i = \mathbf{J}^i(\mathbf{q})\dot{\mathbf{q}} \tag{1}$$

where $\mathbf{q}$ is the $(n \times 1)$ joint displacement vector, $\dot{\mathbf{x}}^i$ is the $(m_i \times 1)$ task velocity vector, and $\mathbf{J}^i$ is the $(m_i \times n)$ Jacobian matrix. In (1) it is assumed that $\sum_i m_i \le n$ so as to have, at most, a 'squared' inverse kinematic problem to solve. Also, in the remainder, the dependence of the Jacobian is omitted for notation compactness.

To be more explicit, eq. (1) refers to a set of m_i variables that can be expressed as a function of the joint variables, and then allow the definition of a meaningful Jacobian. Such task variables are used to describe different desired robot functions. For instance, a task could represent the end-effector position and/or orientation, the available joint range [1], the distance from an obstacle [3], etc. The examples in the next section will be illustrative of possible task specifications.

Proposition. Let

$$\mathbf{P}_A^i = \mathbf{I} - \mathbf{J}_A^{i\#}\mathbf{J}_A^i \tag{2}$$

be the projector (the superscript "#" denotes the generalized inverse of a matrix) onto the null space of the augmented Jacobian

$$\mathbf{J}_A^i = \begin{bmatrix} \mathbf{J}^1 \\ \mathbf{J}^2 \\ \vdots \\ \mathbf{J}^i \end{bmatrix}. \tag{3}$$

Then, the joint velocity solution

$$\dot{\mathbf{q}}^i = \dot{\mathbf{q}}^{i-1} + \tilde{\mathbf{J}}^{i\#}(\dot{\mathbf{x}}^i - \mathbf{J}^i\dot{\mathbf{q}}^{i-1}), \qquad \dot{\mathbf{q}}^1 = \mathbf{J}^{1\#}\dot{\mathbf{x}}^1, \tag{4}$$

with

$$\tilde{\mathbf{J}}^i = \mathbf{J}^i\mathbf{P}_A^{i-1}, \tag{5}$$

allows the i-th task to be executed with lower priority with respect to the previous $i-1$ task, i.e. the i-th task is executed only along those directions 'not disturbing' the $i-1$ higher priority tasks.

Proof. Let

$$\dot{\mathbf{x}}^j = \mathbf{J}^j\dot{\mathbf{q}} \qquad j = 1,\ldots,i-1 \tag{6}$$

denote the differential kinematic equation of either of the previous $i-1$ tasks. Substituting a solution of the kind (4) to $\dot{\mathbf{q}}$ in (6), for the generic j-th task, gives

$$\dot{\mathbf{x}}^j = \mathbf{J}^j\dot{\mathbf{q}}^j + \mathbf{J}^j\tilde{\mathbf{J}}^{i\#}(\dot{\mathbf{x}}^i - \mathbf{J}^i\dot{\mathbf{q}}^j) \qquad j = 1,\ldots,i-1. \tag{7}$$

In (7), observe that $\mathbf{J}^j\tilde{\mathbf{J}}^{i\#} = \mathbf{O}$, which is a direct consequence of (5), since $\mathbf{P}_A^{i-1}$ is a projector onto the null space of $\mathbf{J}_A^{i-1}$ and then also onto the null space of the generic j-th task. Therefore, we have proved that the solution (4) does not alter the previous tasks of higher priority.

At this point, let us remark that the i-th task may actually not be feasible. If $\mathbf{J}^i$ is singular (*task singularity*), the i-th task cannot be satisfied, regardless of all the other tasks. If $\tilde{\mathbf{J}}^i$ is singular, without $\mathbf{J}^i$ being singular (*algorithmic singularity*), the i-th task cannot be satisfied given the previous $i-1$ tasks. The occurrence of singularities will be reprised below.

If neither $\mathbf{J}^i$ nor $\tilde{\mathbf{J}}^i$ is singular, then the i-th task is actually satisfied by using (4). Indeed, plugging (4) into (1) yields

$$\dot{\mathbf{x}}^i = \mathbf{J}^i\dot{\mathbf{q}}^{i-1} + \mathbf{J}^i\tilde{\mathbf{J}}^{i\#}(\dot{\mathbf{x}}^i - \mathbf{J}^i\dot{\mathbf{q}}^{i-1}). \tag{8}$$

Now, notice that $\mathbf{J}^i\tilde{\mathbf{J}}^{i\#}$ can be written as $\mathbf{J}^i\mathbf{P}_A^{i-1}\tilde{\mathbf{J}}^{i\#}$, recalling that the projector $\mathbf{P}_A^i$ is both symmetric ($\mathbf{P}_A^{i\mathrm{T}} = \mathbf{P}_A^i$) and idempotent ($\mathbf{P}_A^i\mathbf{P}_A^i = \mathbf{P}_A^i$). This, accounting for (5), implies that $\mathbf{J}^i\tilde{\mathbf{J}}^{i\#} = \mathbf{I}$ (since neither $\mathbf{J}^i$ nor $\tilde{\mathbf{J}}^i$ is singular), thus concluding the proof. ∎

Having derived the joint velocity solution, the joint acceleration solution can be obtained by direct differentiation of the joint velocity solution, as done also in [15]. We argue that this approach is more effective than solving for $\ddot{\mathbf{q}}$ from the second-order differential kinematic equation which can be obtained by further differentiating eq. (1) with respect to time. In this way, as already pointed out above, we ensure control of those joint velocities in the null-space that may lead to unstable behavior. Therefore, differentiating (4) with respect to time gives

$$\begin{aligned}\ddot{\mathbf{q}}^i &= \ddot{\mathbf{q}}^{i-1} + \tilde{\mathbf{J}}^{i\#}(\ddot{\mathbf{x}}^i - \dot{\mathbf{J}}^i\dot{\mathbf{q}}^{i-1} - \mathbf{J}^i\ddot{\mathbf{q}}^{i-1}) \\ &\quad + \dot{\tilde{\mathbf{J}}}^{i\#}(\dot{\mathbf{x}}^i - \mathbf{J}^i\dot{\mathbf{q}}^{i-1}), \\ \ddot{\mathbf{q}}^1 &= \mathbf{J}^{1\#}\ddot{\mathbf{x}}^1 + \dot{\mathbf{J}}^{1\#}\dot{\mathbf{x}}^1.\end{aligned} \tag{9}$$

An explicit expression of the time-derivative of $\tilde{\mathbf{J}}^{i\#}$ can be written as

$$\dot{\tilde{\mathbf{J}}}^{i\#} = -\tilde{\mathbf{J}}^{i\#}\dot{\tilde{\mathbf{J}}}^i\tilde{\mathbf{J}}^{i\#} + (\mathbf{I} - \tilde{\mathbf{J}}^{i\#}\tilde{\mathbf{J}}^i)\dot{\tilde{\mathbf{J}}}^{i\mathrm{T}}(\tilde{\mathbf{J}}^i\tilde{\mathbf{J}}^{i\mathrm{T}})^{-1} \tag{10}$$

where the pseudoinverse $\tilde{\mathbf{J}}^{i\#} = \tilde{\mathbf{J}}^{i\mathrm{T}}(\tilde{\mathbf{J}}^i\tilde{\mathbf{J}}^{i\mathrm{T}})^{-1}$ has been used, under the assumption that $\tilde{\mathbf{J}}^i$ is full-rank. A more involved expression results if a generalized inverse is used, when $\tilde{\mathbf{J}}^i$ is not full-rank. Alternatively, $\dot{\tilde{\mathbf{J}}}^{i\#}$ may be computed by numerical differentiation.

Furthermore, the time-derivative of $\tilde{\mathbf{J}}^i$ can be compactly written as

$$\dot{\tilde{\mathbf{J}}}^i = -\mathbf{J}^i(\mathbf{Q}_A^{i-1} + \mathbf{Q}_A^{i-1\mathrm{T}}) + \dot{\mathbf{J}}^i\mathbf{P}_A^{i-1} \tag{11}$$

where

$$\mathbf{Q}_A^i = \mathbf{J}_A^{i\#}\dot{\mathbf{J}}_A^i\mathbf{P}_A^i. \tag{12}$$

At this point, a number of considerations are in order concerning the occurrence of singularities in the matrices $\mathbf{J}^i$ and $\tilde{\mathbf{J}}^i$.

Remark 1. In case of a task singularity ($\mathbf{J}^i$ is singular) or an algorithmic singularity ($\tilde{\mathbf{J}}^i$ is singular), the algorithm does not get "stuck" and keeps satisfying the feasible tasks.

In both these cases, the problem of transition into and out of singularities remains, and excessive joint velocities may occur depending on the programmed task trajectory. To overcome this drawback, the generalized inverse should rather be a damped least-squares inverse [16,17] that allows to limit joint velocities in the neighborhood of singularities at the expenses of small tracking errors. To taper this behavior, the minimum singular value of the relevant Jacobian can be utilized, as proposed in [18].

Remark 2. If $\mathbf{J}^i$ is singular, the subsequent tasks are not affected.

This can be demonstrated by observing that, if $\mathbf{J}^i$ is singular, the dimension of the null space of $\mathbf{J}_A^i$ is not decreased.

Remark 3. If $\tilde{\mathbf{J}}^i$ is singular, the subsequent tasks are not affected.

This can be easily recognized by noticing that $\mathbf{J}^i$ is not used in the derivation!

It can be concluded that the general framework based on equations (4) and (9) is computationally advantageous in force of the recursive property. An even more efficient implementation can be obtained by decomposing the matrices of the kind $\mathbf{J}$ into $\mathbf{RS}$, where $\mathbf{S}$ is composed of orthogonal rows and $\mathbf{R}$ is triangular, from which $\mathbf{J}^\#$ can be simply computed as $\mathbf{S}^{\mathrm{T}}\mathbf{R}^{-1}$, e.g. in [14]. Alternatively, a filter-like procedure to avoid differentiation of the Jacobian pseudoinverse can be pursued [19].

Overall robot programming is considerably simplified by the above solution. In fact, the addition of any extra task variable can be easily managed, and conflicting situations with the previous tasks can be handled in a systematic manner.

Last but not least, the end-effector position task need not always be the highest priority task, especially in the case of highly redundant systems as discussed in this paper. For a constrained motion task, for instance, line contact with the surface may be of primary concern [14].

The design of whole-arm manipulation systems [20], where all the available manipulation surfaces of the robot are employed, would pose even more interesting issues about the assignment of task priorities. In that case, in fact, a suitably modified Jacobian in lieu of the end-effector Jacobian is to be considered in order to ensure point/line contact between the links and the manipulated object, thus keenly exploiting the available redundancies.

III. CASE STUDIES

A seven-degree-of-freedom snake-like planar arm is considered to develop two case studies. All links are assumed to be of unit length. For positioning tasks, this arm has five degrees of redundancy and thus is suitable to illustrate the effectiveness of the proposed algorithm. Therefore, the high priority task is described by (adopting absolute joint coordinates)

$$\mathbf{x}^1 = \begin{bmatrix} \sum_{j=1}^{7} c_j \\ \sum_{j=1}^{7} s_j \end{bmatrix}$$

where the shorthand notation $s_j = \sin\phi_j$, $c_j = \cos\phi_j$, with $\phi_j = q_1 + \ldots + q_j$, is used.

As the present work is not focused on the control aspects, we restrict the case studies to the pure inverse kinematic problem. In other words, without loss of generality, we assume to consider a virtual manipulator with unitary inertia matrix, and operating in the absence of gravity and friction. Of course, the application of the adaptive algorithm to the actual manipulator will guarantee global tracking convergence in spite of dynamic parameter variations [21].

The motion of the arm in the different cases is illustrated in the following which will clearly evidence the performance of the solution algorithm with multiple task priorities. The sampling time in all simulations is 1 msec.

In a first case study, the tip trajectory is a straight line from $(3,2)$ to $(3,-0.5)$ to be executed in a time of 2 sec. The initial joint configuration is $(\pi, -\pi/2, 0, -\pi/2, 0, 0, 0)$. A disc of radius 0.3 located at $(2.5, 0)$ is present which obstructs the motion of the outer link of the arm. Without any obstacle avoidance constraint, the arm comes into collision with the disc, as anticipated (Fig. 1).

A constraint task with lower priority is then introduced as [22]

$$\mathbf{x}^2 = \frac{1}{2}(2.5 s_7 - \sum_{j=1}^{7} s_{7-j})^2$$

with $\mathbf{x}_d^2 = 0.3$. Fig. 2 shows the motion of the arm surrounding the obstacle, and still tracking the tip trajectory satisfactorily in force of the task priority.

In a second case study, the tip trajectory is a circle of radius 1 with center at $(4,0)$ to be executed in a time of 1 sec. The initial joint configuration is $(\pi/2, 0, -\pi/2, 0, 0, -\pi/2, 0)$. It is desired to maintain the downward orientation, but the natural motion of the arm, without constraint, clearly does not meet this goal (Fig. 3).

On the contrary, if a constraint task with lower priority is introduced simply as

$$\mathbf{x}^2 = \sum_{j=1}^{7} q_j$$

with $\mathbf{x}_d^2 = -\pi/2$, the arm keeps pointing downward while tracking the circle (Fig. 4).

Successively, a yet lower priority constraint is added in the form of

$$\mathbf{x}^3 = \frac{1}{2}\sum_{j=1}^{7} q_j^2$$

with $\mathbf{x}_d^3 = 0$, which makes the inverse kinematics try to mimic a "flexible beam" clamped at the desired endpoint [14]. This also helps keeping the arm away from the joint limits as well as from singular configurations. And indeed, the postures attained by the arm, while tracking the circle and pointing downward, are typical of a more dexterous motion with respect to above (Fig. 5).

IV. CONCLUSION

A general framework for managing multiple tasks with different priorities in highly redundant robotic systems has been developed. The recursive form of the joint velocity and acceleration solutions makes the resulting algorithm attractive for implementation of task space model-based robot controllers, e.g. the adaptive control. The approach considerably simplifies the problem of high level programming of redundant robotic systems, allowing the user to specify as many tasks as desired and relieving her/him from managing possibly conflicting task situations, which are, instead, resolved in a completely automatic manner by the recursive algorithm. The theoretical results have been confirmed by the numerical simulations of two case studies on a snake-like robot, with an obstacle avoidance constraint and with an orientation constraint plus a dexterity constraint.

REFERENCES

[1] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. Sys. Man Cybernet.*, vol. 7, pp. 868–871, 1977.

[2] J. Baillieul, J. M. Hollerbach, and R. W. Brockett, "Programming and control of kinematically redundant manipulators," *Proc. 23rd IEEE Conf. Decision Control*, Las Vegas, NV, pp. 768–774, Dec. 1984.

[3] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. Robot. Res.*, vol. 4, no. 3, pp. 109–117, 1985.

[4] N. Hogan, "Impedance control: An approach to manipulation," *ASME J. Dynam. Sys. Measurement Control*, vol. 107, pp. 1–24, 1985.

[5] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *Proc. 1985 IEEE Int. Conf. Robot. Automat.*, St. Louis, MO, pp. 722-728, Mar. 1985.

[6] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE J. Robot. Automat.*, vol. 4, pp. 403–410, 1988.

[7] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Analysis and control of articulated robot arms with redundancy," *Prepr. 8th IFAC World Congr.*, Kyoto, Japan, vol. 14, pp. 78–83, Aug. 1981.

[8] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 1, pp. 3–15, 1987.

[9] J. Y. S. Luh, M. W. Walker, and R. P. C. Paul, "Resolved acceleration control of mechanical manipulators," *IEEE Trans. Auto. Control*, vol. 25, pp. 468–474, 1980.

[10] J.M. Hollerbach and K.C. Suh, "Local versus global torque optimization of redundant manipulators," *Proc. 1987 IEEE Int. Conf. Robot. Automat.*, Raleigh, NC, pp. 619–624, Mar./Apr. 1987.

[11] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 3, pp. 49–59, 1987.

[12] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *Proc. 1988 IEEE Int. Conf. Robot. Automat.*, Philadelphia, PA, pp. 183–187, Apr. 1988.

[13] Z.R. Novaković and B. Siciliano, "A new second-order inverse kinematics solution for redundant manipulators," *Prepr. 2nd Int. Workshop Advances Robot Kinematics*, Linz, A, Sept. 1990.

[14] G. Niemeyer and J.-J. E. Slotine, "Computational algorithms for adaptive compliant motion," *Proc. 1989 IEEE Int. Conf. Robot. Automat.*, Scottsdale, AZ, pp. 566–571, May 1989.

[15] K. Kazerounian and Z. Wang, "Global versus local optimization in redundancy resolution of robotic manipulators," *Int. J. Robot. Res.*, vol. 7, no. 5, pp. 3–12, 1988.

[16] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME J. Dynam. Sys. Measurement Control*, vol. 108, pp. 163–171, 1986.

[17] C.W. Wampler, "Manipulator inverse kinematic solutions based on damped least-squares solutions," *IEEE Trans. Sys. Man Cybernet.*, vol. 16, pp. 93–101, 1986.

[18] A.A. Maciejewski and C.A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Robot. Sys.*, vol. 5, pp. 527–552, 1988.

[19] G. Niemeyer and J.-J.E. Slotine, "Adaptive Cartesian control of redundant manipulators," *Proc. 1990 Amer. Control Conf.*, San Diego, CA, pp. 234–241, May 1990.

[20] K. Salisbury, "Whole arm manipulation" *4th Int. Symp. Robot. Res.*, R.C. Bolles and B. Roth (Eds.), MIT Press, Cambridge, pp. 184–189, 1988.

[21] J.-J.E. Slotine and W. Li, "Adaptive manipulator control: A case study," *IEEE Trans. Auto. Control*, vol. 33, pp. 995-1003, 1988.

[22] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy," *Proc. 1986 IEEE Int. Conf. Robot. Automat.*, San Francisco, CA, pp. 1698–1704, Apr. 1986.

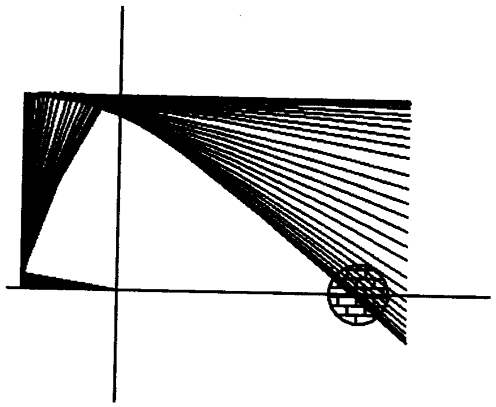
Fig. 1. The snake-like arm tracks the desired rectilinear path *but* collides with the obstacle.

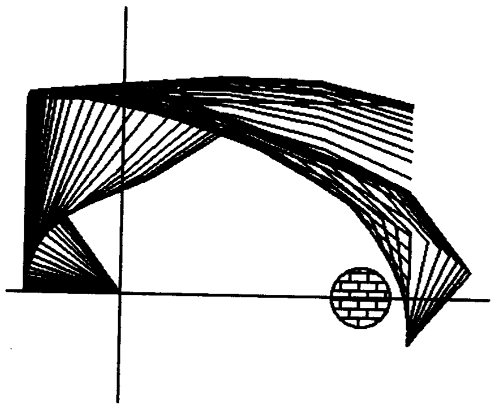
Fig. 2. The snake-like arm tracks the desired rectilinear path *and* avoids the obstacle.

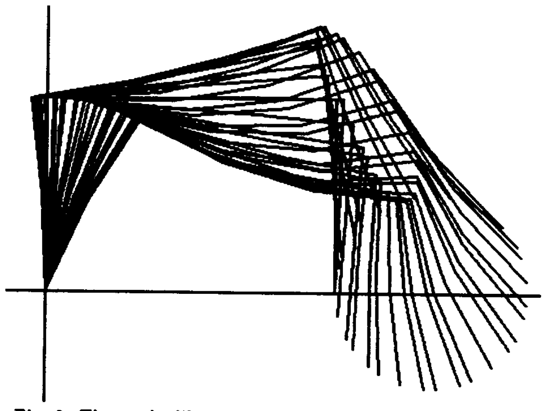
Fig. 3. The snake-like arm tracks the desired circular path *but* looses the desired downward orientation.

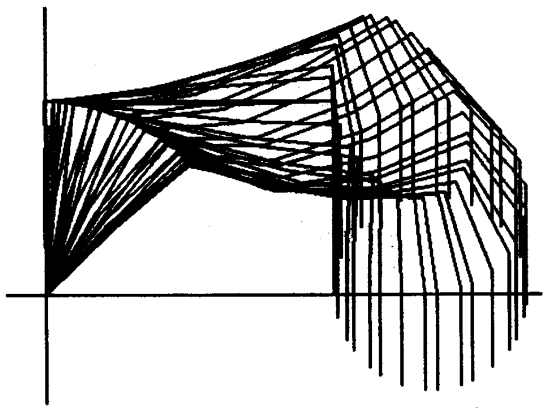

Fig. 4. The snake-like arm tracks the desired circular path *and* keeps the desired downward orientation.

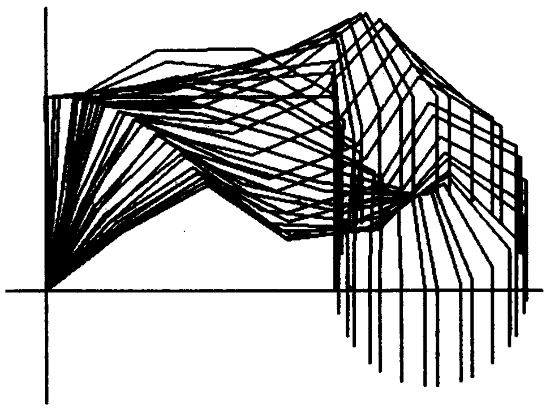

Fig. 5. The snake-like arm tracks the desired circular path *and* keeps the desired downward orientation *and* attains more dexterous configurations.