

Lorenzo Sciavicco and Bruno Siciliano

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

ABSTRACT

Redundancy represents one key towards design and synthesis of more versatile manipulators. Obstacle avoidance and limited joint range constitute two kinds of constraints which can be potentially met by a kinematically redundant manipulator. The natural scenario is the inverse kinematic problem which is certainly a crucial point for robotic manipulator analysis and control.

Based on a recently proposed dynamic solution technique, the inverse kinematic problem for redundant manipulators is solved in this paper. The kinematics of the manipulator is appropriately augmented in order to include the above mentioned constraints; the result is an efficient, fast dynamic algorithm which only makes use of the direct kinematics of the manipulator. Extensive simulation results illustrate the tracking performance for a given trajectory in the Cartesian space, while guaranteeing a collision-free trajectory and/or not violating a mechanical joint limit.

INTRODUCTION

A manipulator is termed kinematically redundant if the number of degrees of freedom (DOF's) is higher than the number of task space coordinates. The most general location of an object in the Cartesian space is completely specified by six coordinates, three for position and three for orientation. In that case a manipulator is considered redundant if it has more than six DOF's. The space of redundant solutions can be conveniently exploited to obtain a more versatile manipulator in terms of its kinematical configuration and its interaction with the environment. In particular, redundancy can be used to meet constraints on joint range availability and/or to obtain trajectories in the joint space which are collision-free in presence of obstacles along the motion. Ultimately another challenging use of redundancy is in keeping the manipulator in a configuration which makes it as dexterous as possible, that is also avoiding kinematic singularities.

This paper is based on research supported by the Ministero della Pubblica Istruzione.

The crucial point for robotic manipulator analysis, and then control synthesis, is the capability of transforming the task space coordinates into the joint space coordinates, that is solving the inverse kinematic problem. The direct kinematic problem allows one to specify in a unique straightforward manner [1] the relationship between the $(n \times 1)$ joint vector $\underline{q}$ and the $(m \times 1)$ Cartesian vector $\underline{x}$ as

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

where $\underline{f}$ is a continuous nonlinear function, whose structure and parameters are known; it associates to each $\underline{q}$ a unique $\underline{x}$, while the inverse mapping

$$\underline{q} = \underline{f}^{-1}(\underline{x}) \quad (2)$$

may have many $\underline{q}$'s associated to each $\underline{x}$ and, because of complexity of (1), is hard to express analytically.

The most direct approach for solving the inverse kinematic problem is certainly to obtain a closed-form solution to (2) [11]. This does not apply to all manipulators; a sufficient condition for obtaining a closed form solution was established by Pieper [15].

In order to overcome the drawbacks encountered in solving (1) in terms of $\underline{q}$, an alternative technique is based on the relationship between joint velocities $\dot{\underline{q}}$ and Cartesian velocities $\dot{\underline{x}}$,

$$\dot{\underline{x}} = J(\underline{q})\dot{\underline{q}} \quad (3)$$

where $J(\underline{q})$ is the Jacobian matrix $\partial \underline{f} / \partial \underline{q}$. The above relation can be inverted to provide the so-called Jacobian control method [2] for redundant manipulators

$$\dot{\underline{q}} = J^{\dagger}(\underline{q})\dot{\underline{x}} \quad (4)$$

which yields locally a minimum norm joint velocity vector in the space of least-square solutions to the left hand side of (3); the superscript $\dagger$ denotes the Moore-Penrose pseudoinverse defined as $J^{\dagger} = J^T(JJ^T)^{-1}$. The solution to (3) can be also modified by the addition of a second term to (4) as

$$\dot{\underline{q}} = J^{\dagger}\dot{\underline{x}} + (I - J^{\dagger}J)\underline{z} \quad (5)$$

The projection operator $(I - J^{\dagger}J)$ selects the components of $\underline{z}$ which are in the space of homogeneous

solutions to (3), and therefore $\underline{z}$ can be used for optimization purposes. Joint range availability is optimized in [3]. Obstacle avoidance is achieved in [4,16,17]. Conceptually similar are the solutions based on the use of generalized inverses proposed for minimization of actuator energy consumption [5], and obstacle avoidance [6]. A rather different approach is then proposed in [7]; it is essentially an iterative technique which is based on a constrained nonlinear optimization algorithm using a modified Newton-Raphson method.

The goal of this paper is to extend the inverse kinematic solution algorithm proposed in [8,9] to the case of redundant manipulators. It is shown how constraints on the joint variables and/or constraints due to obstacles in the workspace can be systematically incorporated in the solution algorithm. The properly extended direct kinematics is to be evaluated [10]. This assures that all the advantages of the dynamic solution are retained, such as continuity of the solution, drastic reduction of computation time, and generation of joint velocities at no additional cost. A case study for a simple planar four DOF's manipulator shows the performance of the proposed solution.

THE GENERAL DYNAMIC SOLUTION

The general dynamic solution technique presented in [8,9] is generalized in the following to the case of unconstrained redundant manipulators.

The inverse kinematic problem is conceived as a dynamical one in order to obtain a general solution algorithm which requires only the computation of direct kinematics (1). Let $\hat{\underline{q}}(t)$ ($n \times 1$) be a solution to (1) relative to a given Cartesian trajectory $\hat{\underline{x}}(t)$ ($m \times 1$). The following error vector $\underline{e}(t)$ can be defined between the desired trajectory $\hat{\underline{x}}(t)$ and the actual trajectory $\underline{x}(t)$ obtained from the algorithm state variables $\underline{q}(t)$,

$$\underline{e}(t) = \hat{\underline{x}}(t) - \underline{x}(t) \quad (6)$$

In order to assure the convergence of $\underline{q}(t)$ to $\hat{\underline{q}}(t)$, error dynamics is involved, i.e. via (3) (dropping the time dependence),

$$\dot{\underline{e}} = \dot{\hat{\underline{x}}} - \underline{J}(\underline{q})\dot{\underline{q}} \quad (7)$$

With the choice

$$\dot{\underline{q}} = \gamma \underline{J}^T(\underline{q})\underline{e} \quad \gamma = \alpha + (\underline{e}^T \dot{\hat{\underline{x}}}) (\underline{e}^T \underline{J} \underline{J}^T \underline{e})^{-1}, \alpha > 0 \quad (8)$$

the dynamic system of Fig. 1 assures that $\underline{e} \rightarrow 0$, and then $\underline{q} \rightarrow \hat{\underline{q}}$. This issue can be recognized by considering the error Lyapunov function $v = .5 \underline{e}^T \underline{e}$ and verifying that its time derivative is negative definite by virtue of (8) [8,9,19]. In the following it seems appropriate to give some remarks on the solution algorithm based on (8).

Remark 1. Let $\text{rank}(\underline{J}) = m$. Suppose that $\underline{e}(0) = 0$; since (8) always guarantees a null tracking error, it naturally introduces, in the neighborhood of $\underline{e} = 0$, an equivalent gain, dependent on state variables, which tends to ∞ . This point leads to the generation of a $\dot{\underline{q}}$ rich in harmonics, whose effect on $\underline{q}$, however, is cut off by the filtering nature of the integrators. Therefore, whenever also

joint velocities are needed, either additional filtering on $\dot{\underline{q}}$ must be introduced or finite tracking errors must be admitted via the purely proportional control law

$$\dot{\underline{q}} = \alpha \underline{J}^T \underline{e} \quad (9)$$

With this choice $\dot{\underline{v}}$ results negative definite only outside a region in the error space which contains $\underline{e} = 0$, that is attractive for all trajectories. Obviously the maximum tracking position error will depend directly on $\|\hat{\underline{x}}\|$, and inversely on α ; it must be underscored, however, that the steady-state ($\dot{\underline{x}} = 0$) error is identically zero [8,9,19].

An alternative inverse kinematic solution algorithm can be conceived by considering

$$\dot{\underline{q}} = \underline{J}^T (M \underline{e} + \dot{\hat{\underline{x}}}) \quad (10)$$

where M is a positive definite matrix whose eigenvalues affect the position error convergence rate to zero. The control law (10) recalls a solution similar to the pseudoinverse Jacobian control (4). It should be noted, however, that (10) seems more robust than the pure pseudoinverse Jacobian control (4), since it is obtained via a dynamic computational scheme which is inherently closed-loop.

Among the three solutions above derived, (8), (9) and (10), the proportional one (9) seems the most attractive from the computational point of view. Since by a proper choice of the feedback gain the tracking error $\underline{e}$ can be conveniently bounded, as shown in several simulations carried out [10,13,14,19,20], this solution can be adopted in the following for practical implementation of the inverse kinematic solution algorithm. Furthermore an appealing feature of the solution (9) lies in the possibility of achieving the following physical interpretation [9]. A fundamental relationship between the ($n \times 1$) vector of joint torques $\underline{\tau}$ and the Cartesian ($m \times 1$) force vector $\underline{f}$ applied at the end effector is [11]

$$\underline{\tau} = \underline{J}^T \underline{f} \quad (11)$$

It can be recognized that applying the above inverse kinematic solution algorithm along a trajectory assigned at the end effector is equivalent to regard the vector $\alpha \underline{e}$ as the elastic force vector which has to be applied at the end effector of an ideal kinematic structure, with null mass and unit viscous damping coefficient, in order to track the desired trajectory.

Remark 2. Let now $\text{rank}(\underline{J}) < m$. It could be argued that $\underline{J}^T \underline{e} = 0$ when $\underline{e}$ belongs to the null space of $\underline{J}^T$. In the light of the above equivalence, however, the null space of $\underline{J}^T$ is in those directions along which a Cartesian force applied at the end effector is completely neutralized by the mechanical constraints of the manipulator. From the implementation viewpoint, the occurrence of such situation can be detected by having $\|\dot{\underline{q}}\| = 0$ and $\|\underline{e}\| \neq 0$. Then, assuming that the target trajectory is planned consistently with the actual kinematic structure, it can be stated that always $\underline{J}^T \underline{e} \neq 0$, when $\underline{e} \neq 0$.

As it had been anticipated, only direct

kinematics ($\underline{f}, J$) is to be computed, which drastically reduces the computational burden. In actual fact, only one iteration per loop is to be intended for the digital implementation of the algorithm for trajectory tracking. The implementation of the algorithm on a single dedicated microprocessor system has been realized in the laboratory for a nonredundant manipulator [12].

It should be emphasized that the dynamic system of Fig. 1, based on the solution (9), will produce joint velocities $\dot{\underline{q}}(t)$ at no additional cost, and with a slight modification it can also generate joint accelerations $\ddot{\underline{q}}(t)$ [9]. This point is very advantageous for advanced control purposes [8].

Last but not least, it is to be mentioned that the solution algorithm here derived is completely general and manipulator independent. An even shorter number of computations are required and better performance is achievable, however, if the algorithm is customized to each particular kinematic structure, as regards the last three axes of motion at the end effector: spherical wrist [10,13,19,20], two-by-two intersecting axes [10,14,19,20], nonintersecting axes [10,19,20].

INCLUSION OF CONSTRAINTS

As previously anticipated, redundancy can be conveniently exploited to solve the inverse kinematic problem with obstacle avoidance and/or limited joint range. The occurrence of either or both of the above situations sets some constraints which can be systematically incorporated in the solution algorithm outlined in the previous section, on the condition that the task space vector in (1) is properly enlarged.

A set of task space variables that describe the configuration of the manipulator with respect to the obstacle in the workspace and/or to the limits on the joint variables can be defined. The only requirement is to express those variables in terms of the joint variables so as to obtain the augmented direct kinematics [10,19]

$$\underline{y} = \underline{f}'(\underline{q}) \quad (12)$$

where $\underline{y}$ is a $((m+v) \times 1)$ vector, with $0 \leq v \leq n-m$, which is completely specified in the task space. Then a solution to (12) can be formally obtained with a choice of the type (9) which proves very powerful, especially for on-line control purposes.

For the sake of clarity, the two cases of obstacle avoidance and limited joint range are treated separately in the following. Nonetheless they both lead to extend the task space vector as in (12), and then it will be possible to solve the inverse kinematic problem under both constraints, as demonstrated later.

Obstacle Avoidance

One of the potential advantages of a kinematically redundant manipulator is the use of the extra (redundant) DOF's to maneuver in a complex workspace and avoid contact with obstacles. The ability of the human arm to work in such environments provides a good model of this ability.

Assume that a manipulator is tracking a desired collision-free end effector trajectory in the task

space. One or more links along its kinematical structure, however, may happen to be too close to an obstacle in the workspace, and a collision is expected. Since the inverse kinematic algorithm (9) provides joint configurations which are adjacent to each other as the manipulator proceeds along the trajectory, one or more constraints need to be introduced in order to avoid the collision with the obstacle. The idea that follows is similar in some regard to the approach taken in [6] and in [16].

In the following it is supposed that obstacles are modeled as convex volumes in the 3D-space; this formalism seems effective because the actual obstructions can always be enveloped in a convex volume (the simplest is a sphere), while it is easy to compute the distance from such volumes to the links of the manipulator. For a planar mechanism one might simply adopt disks to model obstacles, as done for instance in [17].

It can be assumed that a link has avoided a convex obstacle if its minimum distance from the obstacle is greater than a preplanned threshold distance. If all links satisfy this condition there is no reason for modifying the current solution along the trajectory, and the dynamic solution algorithm (9) will select one of the ∞^{n-m} possible configurations, depending on the initial joint configuration.

On the other hand, if the distance between one of the links and an obstacle becomes less than the threshold, the current solution is to be modified. It is understood that at most $n - m$ constraints of such type can be activated. One might also think of setting several values of thresholds in correspondence of each link of the manipulator. This can be done accounting for the type of sensors used to detect the above distances, such as proximity sensors, video cameras etc...

In order to illustrate the technique here proposed, assume first that a single link is involved in a possible collision with a single obstacle. It is understood that such pair varies as the manipulator's end effector tracks the desired trajectory of motion. Let then $\hat{d}_o$ be the threshold distance and $\underline{c}$ denote the position vector of the point of interest on the obstacle. A point at minimum distance from the link to the obstacle can be determined [16]; let $\underline{p}_o$ indicate the position vector of this so-called obstacle avoidance point. Both vectors $\underline{c}$ and $\underline{p}_o$ are defined with respect to the same base frame in which the direct kinematics of the manipulator is expressed. See Fig. 2 for a planar example. Another important remark is in order; the position of the minimum distance point moves, as the manipulator moves about the obstacle, i.e. $\underline{p}_o$ has to be dynamically recomputed along the trajectory. If the distance $\|\underline{d}_o\|$ between the two points, where $\underline{d}_o = \underline{p}_o - \underline{c}$, becomes less than the threshold distance $\hat{d}_o$, there is a danger of collision, and the joint velocities, which represent the control inputs to the system of Fig. 1, need to be modified accordingly to the new constraint activated. This can be accomplished as follows. In analogy with the error definition between reference and actual task variables (6), define the error

$$e_o = .5(\hat{d}_o^2 - \underline{d}_o^T \underline{d}_o) \quad (13)$$

Differentiating (13) with respect to time gives

$$\dot{\mathbf{e}}_o = \dot{\mathbf{d}}_o - \frac{\mathbf{d}_o^T}{\mathbf{d}_o} \dot{\mathbf{c}} - \mathbf{j}_{do}^T \dot{\mathbf{q}} \quad (14)$$

with

$$\mathbf{j}_{do}^T = \frac{\mathbf{d}_o^T}{\mathbf{d}_o} \mathbf{j}_{po}^T \quad (15)$$

where $\mathbf{j}_{po}$ is the Jacobian matrix $\partial \mathbf{p}_o / \partial \mathbf{q}$ of the obstacle avoidance point [16]. If the error dynamics (7) is opportunely extended by (14), a control $\dot{\mathbf{q}}$ can be obtained in the same formal way as in (9). To this purpose, $\mathbf{e}$ will indicate the extended error vector $((m+1) \times 1)$ in the Cartesian space, whose last component is e_o defined in (13), and $\mathbf{J}$ will denote the extended Jacobian matrix $((m+1) \times n)$, whose last row is $\mathbf{j}_{do}^T$ defined in (15). In this way the motion of those DOF's which determine the position of the obstacle point $\mathbf{p}_o$ is braked, preventing the link of interest from approaching the obstacle. As a matter of fact, a link which is candidate for a collision is forced to move tangentially around the imaginary sphere with center at $\mathbf{c}$ and of radius $\mathbf{d}_o$. It is to note that in (14) both the cases of moving obstacle ($\dot{\mathbf{c}} \neq 0$) and varying threshold distance ($\dot{\mathbf{d}}_o \neq 0$) have been considered. In what follows it is assumed that $\dot{\mathbf{c}} = 0$ and $\dot{\mathbf{d}}_o = 0$, without loss of generality.

At this extent, it seems quite straightforward to extend this technique to other pairs of points interested to a collision (an obstacle and an obstacle avoidance point along the structure of the manipulator, respectively) up to eventually cover the $n - m$ redundant DOF's. Discussion of when an obstacle constraint can be released will be provided after presenting the overall solution.

Limited Joint Range

Conceptually similar is the activation of a mechanical constraint on a joint variable for the inverse kinematic scheme of Fig. 1. The idea that follows is partially inspired by [5]. Assume that a joint variable q_i is kinematically constrained between two constant extremal values q_{imin} and q_{imax} ,

$$q_{imin} \leq q_i \leq q_{imax} \quad (16)$$

If the joint variable approaches either of the limits while the manipulator's end effector is tracking the preplanned trajectory, the solution given in (9) is to be modified. As done above for obstacle avoidance, a threshold distance $\hat{\mathbf{d}}_q$ can be defined with the intent that if the distance of the current q_i from either of the two limits becomes less than $\hat{\mathbf{d}}_q$, the control (9) needs to be modified. To this purpose define the error

$$\mathbf{e}_q = \hat{\mathbf{d}}_q - \mathbf{d}_q \quad (17)$$

where either $\mathbf{d}_q = q_i - q_{imin}$ or $\mathbf{d}_q = q_{imax} - q_i$, depending on which limit is involved; note that $\hat{\mathbf{d}}_q$ has to be dynamically recomputed along the trajectory. Differentiating (17) with respect to time gives

$$\dot{\mathbf{e}}_q = \dot{\hat{\mathbf{d}}}_q - \mathbf{u}_i^T \dot{\mathbf{q}} \quad (18)$$

where

$$\mathbf{u}_i^T = (0 \ 0 \ \dots \ \underset{(i)}{+1} \ \dots \ 0) \quad (19)$$

the sign + applying for q_{imin} and the sign - applying for q_{imax} . Therefore, it seems natural to extend the error dynamics (7) by (18), quite in the same way as it has been done for obstacle avoidance. The extended error vector will include an additive component due to $\mathbf{e}_q$ in (17), and correspondingly the extended Jacobian matrix will contain a row given by $\mathbf{u}_i^T$ in (19). In this way the motion of that DOF which was approaching the limit is opportunely braked and kept on the threshold. As above, if the threshold distance is chosen constant, $\dot{\hat{\mathbf{d}}}_q = 0$.

Finally, as in the above case, the technique can be extended to account for other mechanical constraints of this type, up to eventually cover the $n - m$ redundant DOF's. The possibility of releasing a joint limit constraint and the occurrence of both type of constraints will be discussed in the following subsection.

The Overall Solution

On the basis of the results of the two previous subsections, the overall dynamic solution to the inverse kinematic problem for constrained redundant manipulators is established in the following. To this purpose the task space vector $\mathbf{x}$ is enlarged into the $((m+v) \times 1)$ vector $\mathbf{y}$, as anticipated in (12),

$$\mathbf{y} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}_o \\ \mathbf{x}_q \end{bmatrix} \quad (20)$$

where $\mathbf{x}_o$ is the $(k \times 1)$ vector whose components are the quantities of the type $\mathbf{d}_{oq}$ defined in (13) for each active obstacle constraint, and $\mathbf{x}_q$ is the $(r \times 1)$ vector whose components are the quantities of the type $\mathbf{d}_q$ defined in (17) for each active joint limit constraint. Correspondingly the task space reference vector becomes

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathbf{x}} \\ \hat{\mathbf{x}}_o \\ \hat{\mathbf{x}}_q \end{bmatrix} \quad (21)$$

with obvious meaning of the vectors $\hat{\mathbf{x}}_o$ ($k \times 1$) and $\hat{\mathbf{x}}_q$ ($r \times 1$). It must be remarked that

$$v = k + r \leq n - m \quad (22)$$

so as to activate at most $n - m$ constraints on obstacle avoidance (in number of k) and joint range availability (in number of r). The control is then derived in the same formal way as in (9), i.e.

$$\dot{\mathbf{q}} = \alpha \mathbf{J}_{e-y}^T \mathbf{e}_y \quad (23)$$

where

$$\mathbf{J}_e = \begin{bmatrix} \mathbf{J} \\ \mathbf{J}_{do} \\ \mathbf{U} \end{bmatrix} \quad (24)$$

is the extended Jacobian matrix which includes,

besides the end effector Jacobian matrix J , the Jacobian matrix J_{d0} ($k \times n$) whose rows j_{d0} are defined in (15), and the matrix U ($r \times n$) whose rows are defined in (19), and

$$\underline{e}_y = \begin{bmatrix} \underline{e} \\ \underline{e}_0 \\ \underline{e}_q \end{bmatrix} \quad (25)$$

is the extended error vector in the task space which includes, besides the end effector error vector $\underline{e}$, the error vector $\underline{e}_0$ ($k \times 1$) whose components are defined in (13), and the error vector $\underline{e}_q$ ($r \times 1$) whose components are defined in (17). Finally in (23) $\alpha > 0$ is a feedback gain which determines the convergence rate of $\underline{e}_y$.

Under the control (23), the end effector trajectory $\hat{\underline{x}}(t)$ is still tracked as in (9), and the redundant DOF's are conveniently used to avoid collision with the obstacles present in the environment and/or to avoid the generation of joint trajectories which are not kinematically feasible.

Although (23) is the basis of the inverse kinematic solution proposed here, proper decision making by a higher control level is equally important, if not crucial, to successful operation of the algorithm. This higher control level should be in charge of trajectory planning and activation + inactivation of constraints. To this purpose the following considerations seem appropriate.

The activation of a constraint is simply performed by detecting that one of the threshold distances (obstacle or joint limit) is violated. The introduction of constraints naturally sets restrictions on joint angles, which contribute to reduce the manipulator reachability workspace. A conflict with the desired trajectory $\hat{\underline{x}}(t)$, originally planned for the end effector, may then arise. In the limit one may wind up in a "deadlocked" situation where further movement is not possible and the system is overconstrained, for instance in a complex environment with multiple obstacles. The occurrence of such situation can be recognized, at algorithmic level, by the fact that the extended error vector $\underline{e}$ in (25) enters the null space of the extended Jacobian J_e . In this case it results $\| \dot{\underline{q}} \| = 0$ and $\| \underline{e} \| \neq 0$, and the inverse kinematic algorithm must be aborted. This statement can be justified on the basis of the previous equivalence outlined for the unconstrained case.

It should be emphasized, however, that such situation may have not been properly caused by a complex environment, but rather determined by the fact that no constraint has been released at all. Consequently the option of inactivating a constraint should be introduced, in the sense that a constraint could be released when the end effector trajectory "naturally" drives the structure away from the constraint itself. This feature can be performed at algorithmic level as follows. Remind that the solution algorithm based on (23) guarantees only a bounded, but not null, tracking error. Therefore, the criterion for inactivating a constraint can be set up in the same way as it is done for activating a constraint, based on the evaluation of proper distance errors. Thus it is assumed that a constraint is inactivated if the corresponding distance error becomes negative.

It might be argued that the convergence of the solution algorithm as one or more constraints are either activated or inactivated does not strictly follow from the convergence properties of the configurations of the system, before and after either the enlargement or the reduction of the error space. This statement is theoretically correct, but from the engineering point of view, several simulation results have shown that the algorithm correctly runs for cases of practical interest, thus encouraging the adoption of the technique here proposed.

A CASE STUDY

In order to test the performance of the proposed dynamic inverse kinematic solution for constrained redundant manipulators, a simple case study is developed in the following. A planar four DOF manipulator of the type of Fig. 2 is considered. The link lengths have been chosen as $l_1 = l_2 = l_3 = l_4 = .3$ m. A straight line reference trajectory is commanded from $\underline{\hat{p}}(0) = (.5598 \ - .15)^T$ [m] to $\underline{\hat{p}}(T) = (.8 \ -.2)^T$ [m] with $T = 3$ s, where $\underline{\hat{p}}$ denotes the desired end point position vector; the velocity profile is the typical trapezoidal one (acceleration + cruise + deceleration) with $\| \dot{\underline{\hat{p}}} \|_{\max} = .12$ m/s. The manipulator is assumed to start from a joint configuration which places the end point $\underline{p}(0)$ at $\underline{\hat{p}}(0)$. Remind that the resulting joint trajectories will depend upon the initial joint configuration, since the algorithm progresses with continuity along the given trajectory. The initial conditions of the joint variables are $\underline{q}(0) = (180 \ -30 \ -90 \ -30)^T$ [$^\circ$].

Two constraints of the type previously described have been introduced. An obstacle has been located at $\underline{c} = (.7 \ -.28)^T$ [m], and a mechanical limit on the third joint is $q_3 \leq -80^\circ$.

Four different sets of simulations have been carried out. Figs. 3 through 5 illustrate the results. A sampling time of 1 ms has been chosen; this is seen to be sufficient to evaluate the direct kinematic functions reported in the Appendix, using a single dedicated microprocessor system with floating point multiply unit [12]. The end point tracking error has been computed as the norm of the end point position error vector $\underline{e} = \underline{\hat{p}} - \underline{p}$.

In the first set of results the unconstrained control (9) has been applied with $\alpha = 1000$; to this end in [12] it is shown how discretizing the algorithm leads to the optimal choice of the feedback gain as the inverse of the sampling time. It is seen that the manipulator would violate both constraints while it tracks the end point reference trajectory. As a matter of fact the end point tracking error (Fig. 3) is small, but the fourth link is involved in a collision with the obstacle (Fig. 4) and the third joint variable exceeds -80° (Fig. 5).

In the second set of results only the obstacle constraint has been activated as soon as $e_0 = \hat{d}_4^2 - d_4^2$ becomes greater than zero; the threshold distance $\hat{d}_4$ has been set to .03 m. It is seen that the end point tracking performance remains satisfactory (Fig. 3) and the obstacle is avoided (Fig. 4) in the sense that $\| \underline{\hat{d}}_4 \|$, after going below $\hat{d}_4$, tends asymptotically to this value, as proven in theory. The constraint on the third joint variable,

however, is still violated (Fig. 5).

In the third set of results only the joint limit constraint has been activated as soon as $e = \hat{d}_{q_3} - d_{q_3}$ becomes greater than zero; the threshold distance $\hat{d}_{q_3}$ has been chosen as 2° . Similarly to the previous case the end point tracking is maintained (Fig. 3) and the third joint variable is braked against passing the limit (Fig. 5) in the sense that q_3 , after going above -82° , tends asymptotically to this value, as anticipated in theory.

Finally both constraints have been activated at proper instants of time. In this particular case it can be recognized (Fig. 5) that the joint limit constraint has been released, as the end point trajectory naturally drives q_3 away from the limit, due to the activation of the obstacle constraint meanwhile. It is seen that the end point tracking is maintained (Fig. 3) and the obstacle constraint is asymptotically met (Fig. 4).

CONCLUSIONS

This paper has presented a solution to the inverse kinematic problem for constrained redundant manipulators. The resulting algorithm is based on a dynamic reformulation of the problem leading to a closed loop scheme whose stability is assured by choosing a control which only involves the computation of the direct kinematics of the manipulator. The only requirement is to systematically enlarge the direct kinematics in order to include the constraints which can be potentially met by a kinematically redundant manipulator, such as obstacle avoidance and limited joint range. It has been proven, and shown by an example, that the constraints, once active, can be successfully met. The end point of the manipulator keeps tracking the desired trajectory and the resulting joint trajectories avoid the collision with an obstacle and/or avoid violation of a mechanical limit on a joint variable. Some issues regarding the activation and the inactivation of constraints have been finally addressed. The possibility of including constraints on the manipulator's dexterity is currently being investigated [18]. Future research efforts will be likely dedicated to investigate the application of the technique here presented to particular redundant geometries, such as maintenance robots operating in plasma vessels.

REFERENCES

- [1] J. Denavit and R.S. Hartenberg, "A kinematic notation for lower-pair mechanisms based on matrices," ASME J. Appl. Mech., vol. 22, pp. 215-221, June 1955.
- [2] D.E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," Trans. ASME J. Dyn. Syst., Meas., Contr., vol. 94, pp. 303-309, Dec. 1972.
- [3] A. Liegeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," IEEE Trans. Syst., Man, Cybern., vol. SMC-7, no. 12, pp. 868-871, Dec. 1977.
- [4] C.A. Klein, "Use of redundancy in the design of robotic systems," Proc. 2nd Int. Symp. Robotics Research, Kyoto, Japan, Aug. 1984.
- [5] M. Vukobratovic and M. Kircanski, "A dynamic approach to nominal trajectory synthesis for redundant manipulators," IEEE Trans. Syst., Man, Cybern., vol. SMC-14, no. 4, pp. 580-586, July/Aug. 1984.
- [6] M. Kircanski and M. Vukobratovic, "Trajectory planning for redundant manipulators in the presence of obstacles," Proc. 5th CISM-IFTOMM Ro.Man.Sy, Udine, Italy, June 1984.
- [7] A.A. Goldenberg, B. Benhabib and R.G. Fenton, "A complete generalized solution to the inverse kinematics of robots," IEEE J. Robotics, Automation, vol. 1, no. 1, pp. 14-20, Mar. 1985.
- [8] A. Balestrino, G. De Maria and L. Sciavicco, "Robust control of robotic manipulators," Proc. 9th IFAC World Congress, Budapest, Hungary, July 1984.
- [9] W.A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," Proc. 23rd IEEE Conf. Dec., Contr., Las Vegas, NV, Dec. 1984.
- [10] L. Sciavicco and B. Siciliano, "Solving the inverse kinematic problem for robotic manipulators," Prep. 6th CISM-IFTOMM Ro.Man.Sy, Cracow, Poland, Sept. 1986.
- [11] R.P. Paul, Robot Manipulators: Mathematics, Programming, and Control, MIT Press, Cambridge, MA, 1981.
- [12] G. De Maria and R. Marino, "A discrete algorithm for solving the inverse kinematic problem for robotic manipulators," Proc. '85 Int. Conf. Advanced Robotics, Tokyo, Japan, Sept. 1985.
- [13] A. Balestrino, G. De Maria, L. Sciavicco and B. Siciliano, "A novel approach to coordinate transformation for robotic manipulators," IEEE Trans. Syst., Man, Cybern., publication pending, 1986.
- [14] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," IEEE Trans. Syst., Man, Cybern., vol. SMC-16, no. 4, pp. 550-559, July/Aug. 1986.
- [15] D. Pieper, The Kinematics of Manipulators Under Computer Control, Ph.D. Thesis, Stanford University, 1968.
- [16] A.A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," Int. J. Robotics Research, vol. 4, no. 3, pp. 109-117, Fall 1985.
- [17] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy," Proc. 1986 Int. Conf. Robotics, Automation, San Francisco, CA, Apr. 1986.

- [18] L. Sciavicco and B. Siciliano, "An inverse kinematic solution algorithm for dexterous redundant manipulators," '87 Int. Conf. Advanced Robotics, Versailles, France, Oct. 1987, subm. for pres.
- [19] B. Siciliano, Solution Algorithms to the Inverse Kinematic Problem for Manipulation Robots (in Italian), Ph.D. Thesis, Università di Napoli, Dec. 1986.
- [20] G. De Maria, L. Sciavicco and B. Siciliano, "A general solution algorithm to coordinate transformation for robotic manipulators," Proc. '85 Int. Conf. Advanced Robotics, Tokyo, Japan, Sept. 1985.

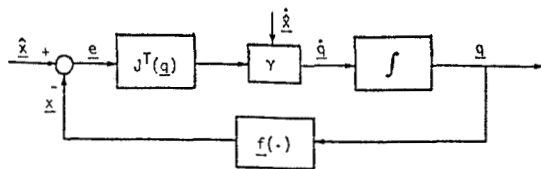


Fig. 1. The general inverse kinematics scheme.

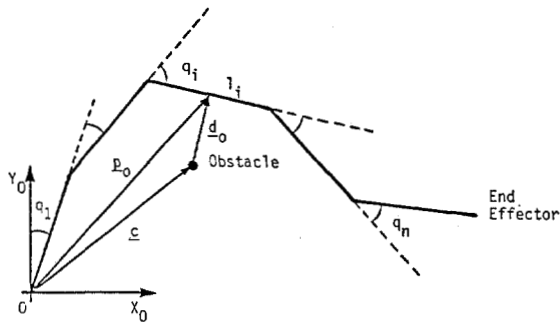


Fig. 2. Geometry of a planar arm showing the point nearest to the obstacle.

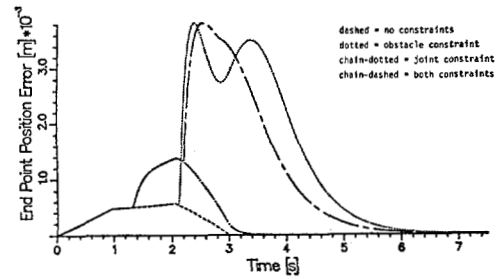


Fig. 3. End point position tracking errors.

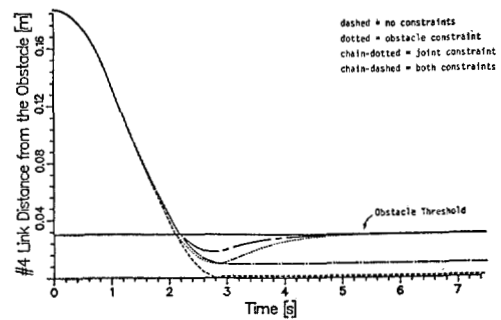


Fig. 4. 4th Link distances from the obstacle.

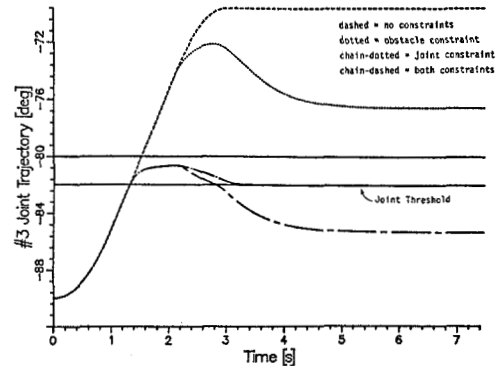


Fig. 5. 3rd Joint trajectories.