Robotica (2003) volume 21, pp. 137–142. © 2003 Cambridge University Press
DOI: 10.1017/S0263574702004678 Printed in the United Kingdom

A contribution to minimum-time task-space path-following problem for redundant manipulators

Francesco Basile* and Pasquale Chiacchio**

** Dipartimento di Ingegneria Elettrica e dell'Informazione, Università degli Studi di Salerno, Via Ponte Don Melillo 1, 84084 Fisciano (SA) (Italy)*
*** Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli (Italy).*

(Received in Final Form: October 4, 2002)

SUMMARY
The minimum-time path-following problem for non-redundant manipulator has already been analyzed and a number of efficient solution algorithms have been proposed. These algorithms can be applied also to redundant manipulators if the path is assigned in the joint space. If the path is assigned in the task space instead, how to exploit kinematic redundancy to reduce the execution time is still an open problem. In this paper it is proposed to follow a heuristic approach to choose between different solutions to the inverse kinematics problem that underlies the minimum-time path-following control problem. The aim is to obtain a joint-space path which configures the manipulator so as to improve acceleration/deceleration capabilities of the end-effector. This is obtained by penalizing the motion of joints with higher inertia-to-torque ratios. Numerical results are presented for an "easy-to-understand" three degree-of-freedom planar manipulator involved in two-dimensional paths.

KEYWORDS: Redundant manipulators; Path following; Task-space; Joint-space path.

1. INTRODUCTION

The minimum-time path-following control problem is of great importance for industrial applications of robotic manipulators. In fact, industrial manipulators are usually intended to perform repetitive tasks and so execution time minimization is equivalent to a productivity increase. The speed and hence the productivity of an industrial manipulator are limited by the trajectory assigned which could require dynamic efforts that are not achievable because of the limited joint torques.

The first study of a minimum-time control problem was done by Hollerbach[1] who developed a time-scaling property of manipulator dynamics; given a trajectory in the joint space, the corresponding joint torques/forces were calculated and, if they were beyond their limits, time scaling was used and the new torques/forces were calculated without recomputing the inverse dynamics. Two independent solutions to the minimum-time control problem were proposed by Shin and McKay[2] and Bobrow *et al.*;[3] once a joint-space path is parametrized with a single scalar variable, the nonlinear dynamics and actuator constraints are transformed into constraints on the acceleration along the path, reducing the system to a second-order one with nonlinear bounds on the input. The two solutions differ for the algorithm used to find the switching points. A quite similar solution was proposed by Pfeiffer and Johanni[4] where the focus was concentrated on the geometrical interpretation of the torque/force limits. Slotine and Yang[5] presented a computationally more efficient technique to solve the problem based on the fact that characteristic switching points can be found in a straightforward manner. Chen and Desrochers[6] showed that for this problem one and only one joint torque is always in saturation at every finite time interval, if the assigned joint-space path is not the one that solves the point-to-point minimum-time problem.

The output of these algorithms are the joint torques to be fed to the manipulator in order to follow the path. This is an open-loop approach to the problem and thus the variation of parameters or modeling errors could lead to unpredictable behavior. In Dahl and Nielsen[7] a closed-loop scheme which eventually modifies the manipulator's behavior to ensure exact path tracking was devised. Experimental results for an industrial manipulator were reported by Dahl.[8] Time optimal motions in the presence of dynamic singularities, i.e. when one of the equation of motions reduces to velocity constraints, were investigated by Shiller and Lu[9] and by Shiller.[10]

All the above algorithms rely on the fact that a joint-space path is available as a parametrized curve twice differentiable with respect to the path parameter, so that also the joint velocities and accelerations can be written as functions of the time derivatives of the parameter. When the given path is in the task space,[3–5] an inverse kinematics solution is needed. This can be found (not always in a simple manner) for non-redundant arms whilst for redundant arms the solution is not unique and could not exist in a closed form.

To solve the inverse kinematic problem for a redundant arm iterative algorithms based on the unweighted pseudo-inverse of the Jacobian matrix can be used,[11] eventually put in the form of a closed-loop algorithm.[12] If no constraint is imposed, these algorithms give the solution that guarantees the local minimization of joint velocity norm. This solution, however, does not guarantee that the resulting joint-space path is the one that actually solves the minimum-time control problem for a redundant arm.

https://doi.org/10.1017/S0263574702004678 Published online by Cambridge University Press

A suitable constraint should then be included in the inverse kinematic solution so as to exploit redundancy in order to reduce the execution time for a given task-space path. The inclusion of an additional constraint in the inverse kinematic problem can be easily made with the closed-loop form algorithm presented in reference [13]. Chiacchio[14] proposed to use the projection of the dynamic manipulability ellipsoid as a constraint in the inverse kinematic problem in order to improve acceleration/deceleration capabilities of the end effector along the specified taskspace path.

In this paper an inverse kinematic algorithm is proposed which uses a weighted psedoinverse of the manipulator Jacobian matrix in order to locally improve acceleration/deceleration capabilities of the manipulator along the path. The chosen weighted pseudoinverse penalizes the motion of joints suffering from higher inertia-to-torque ratios. Since the inverse kinematics solution is only locally optimal, a scalar parameter is introduced which must be tuned by trial-and-error in order to find the best achievable solution with the proposed method. The technique is appealing in that its computationa1 burden is almost equivalent to other established algorithms and the tuning of only one scalar parameter is required; thus, it is well suited for the off-line computing which is typical of minimum-time problems.

The paper is organized as follow: Section 2 reviews the minimum-time control problem. Section 3 presents the method to improve the minimum-time control algorithm for redundant manipulator. Section 4 deals with the implementation details and discusses case studies which show the effectiveness of the proposed technique. Conclusions are drawn in Section 5.

2. THE MINIMUM-TIME PATH-FOLLOWING PROBLEM

In this section the minimum-time control problem solution as presented in the references [2–5] is briefly reviewed.

Denote the joint displacements and the applied joint torques/forces with the $n \times 1$ vectors $\boldsymbol{q}$ and $\boldsymbol{\tau}$, respectively, so that the dynamics of an n-degree-of-freedom rigid robotic arm can be written as

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{q}}^{\mathrm{T}}\boldsymbol{C}(\boldsymbol{q})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{H}(\boldsymbol{q})$ is the $n \times n$ positive-definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q})$ is the $n \times n \times n$ tensor for the centripetal and Coriolis torques, and $\boldsymbol{g}(\boldsymbol{q})$ is the $n \times 1$ vector of gravitational torques.

A path for the manipulator can be given either in the joint space or in the task space. It is supposed that a path in the joint space can be written as a continuous parametrized curve that never retraces itself

$$\boldsymbol{q} = \boldsymbol{q}(\lambda) \tag{2}$$

where $\lambda \in [0, \lambda_{max}]$ is a single scalar parameter. Time derivatives of $\boldsymbol{q}$ along the path can be expressed as

$$\dot{\boldsymbol{q}} = \frac{d\boldsymbol{q}}{d\lambda}\dot{\lambda}, \tag{3}$$

$$\ddot{\boldsymbol{q}} = \frac{d\boldsymbol{q}}{d\lambda}\ddot{\lambda} + \frac{d^2\boldsymbol{q}}{d\lambda^2}\dot{\lambda}^2. \tag{4}$$

If the path is assigned in the task space it is supposed that it can be written as

$$\boldsymbol{p} = \boldsymbol{p}(\lambda) \tag{5}$$

where $\boldsymbol{p}$ is a $m \times 1$ vector, $m \le n$. If the direct kinematics of the manipulator is written as $\boldsymbol{p} = \boldsymbol{k}(\boldsymbol{q})$ and $\dot{\boldsymbol{p}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$, it is necessary to have the inverse kinematics written, in the case $m = n$, as

$$\boldsymbol{q} = \boldsymbol{k}^{-1}(\boldsymbol{p}(\lambda)), \tag{6}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\frac{d\boldsymbol{p}}{d\lambda}\dot{\lambda}, \tag{7}$$

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}^{-1}\left[\frac{d\boldsymbol{p}}{d\lambda}\ddot{\lambda} + \frac{d^2\boldsymbol{p}}{d\lambda^2}\dot{\lambda}^2 - \left(\boldsymbol{J}^{-1}\frac{d\boldsymbol{p}}{d\lambda}\right)^T \boldsymbol{R}\left(\boldsymbol{J}^{-1}\frac{d\boldsymbol{p}}{d\lambda}\right)\dot{\lambda}^2\right], \tag{8}$$

where $\boldsymbol{R}$ is the Hessian matrix of the vector function $\boldsymbol{k}(\boldsymbol{q})$.

Substituting (2–4) or (6–8) in (1) yields equations of the form[5]

$$a_i(\lambda)\ddot{\lambda} + b_i(\lambda)\dot{\lambda}^2 + c_i(\lambda) = \tau_i \quad i = 1, \ldots, n \tag{9}$$

that is a set of n scalar equations in the pararmeter λ. The joint actuator limits can be written as

$$\tau_{i,min} \le \tau_i \le \tau_{i,max} \quad i = 1, \ldots, n. \tag{10}$$

Combining (9) and (10), one obtains

$$\tau_{i,min} \le a_i(\lambda)\ddot{\lambda} + b_i(\lambda)\dot{\lambda}^2 + c_i(\lambda) \le \tau_{i,max} \qquad i = 1, \ldots, n. \tag{11}$$

At this point, a few algorithms have been proposed. In the following we describe the one proposed by Shin and McKay.[2]

Step 1: Draw the maximum velocity curve in the space of the states λ and $\dot{\lambda}$, that is the curve connecting the points beyond which one of the equations (11) cannot be satisfied for any value of $\dot{\lambda}$. The admissible region is delimited by this curve and by the lines $\lambda = 0$, $\lambda = \lambda_{max}$ and $\dot{\lambda} = 0$.
Step 2: Start at $\lambda = 0$, $\dot{\lambda} = 0$ and construct a trajectory with the maximum acceleration value until this curve leaves the admissible region or goes past $\lambda = \lambda_{max}$.
Step 3: Starting at $\lambda = \lambda_{max}$, $\dot{\lambda} = 0$, do as in Step 2, proceeding backward and using the maximum deceleration value.
Step 4: If the two curves intersect, then stop.
Step 5: Else, find along the maximum velocity curve a point from which an accelerating trajectory remains below the maximum velocity curve.
Step 6: Construct a decelerating trajectory backwards from that point until it intersects an accelerating trajectory. This gives a switching point.
Step 7: Construct an accelerating trajectory from that point until it intersects the decelerating trajectory of Step 2, and

https://doi.org/10.1017/S0263574702004678 Published online by Cambridge University Press

then the procedure terminates, or it leaves the admissible region, and then go to Step 5.

Now the execution time can be computed as

$$\int_0^{\lambda max} (1/\dot{\lambda}) d\lambda \tag{12}$$

on the resulting trajectory in the plane of λ, $\dot{\lambda}$.

3. REDUNDANT MANIPULATOR

The established algorithms work well also for redundant manipulators if the path to follow is assigned in the joint space as in (2). If a task-space path is given the algorithms rely on the fact that an inverse kinematics solution exists, so that equations (6–8) can be written or, at least, numerically computed.

If the manipulator is redundant, it is well known that the inverse kinematic solution is not unique and it could not exist in a closed form. In reference [4] it is proposed to use an iterative numerical algorithm based on the unweighted pseudo-inverse of the Jacobian matrix: A sufficiently large number of points on the path is taken, then they are inverted to find joint space points, and finally interpolation techniques are used to obtain a jointspace path. However, such a technique utilizes an inverse kinematics solution that only guarantees locally the minimum norm of joint velocities. If minimum-time algorithms are executed on the resulting joint space path, there is no certainty that the obtained trajectory is the optimal one. This is because the kinematics of a redundant manipulator is underspecified and thus it has been chosen only as one of the possible solutions with no criteria.

Is it well-known that to exploit a manipulator's redundancy it is possible to add a term in the null space of the Jacobian as

$$\frac{d\boldsymbol{q}}{d\lambda} = \boldsymbol{J}^{\dagger} \frac{d\boldsymbol{q}}{d\lambda} + \alpha (\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}) \nabla_q h \tag{13}$$

where α is a scalar gain, $\boldsymbol{J}^{\dagger} = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{J}^{\mathrm{T}})^{-1}$ is the unweighted pseudoinverse of $\boldsymbol{J}$, the matrix $\boldsymbol{I} - \boldsymbol{J}^{\dagger}\boldsymbol{J}$ is a projector in the null space of $\boldsymbol{J}$ and h is a scalar function to be maximized. In the minimum-time problem a suitable function h should be found to represent acceleration capabilities of the end-effector (as it has been proposed in reference [14]).

An alternative way is to resort to a weighted pseudoinverse of the Jacobian rmatrix

$$\boldsymbol{J}_W^{\dagger} = \boldsymbol{W}^{-1}\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{J}\boldsymbol{W}^{-1}\boldsymbol{J}^{\mathrm{T}})^{-1} \tag{14}$$

where $\boldsymbol{W}$ is a symmetric, positive-definite, eventually configuration-dependent matrix. The differential inverse kinematics can then be written as

$$\frac{d\boldsymbol{q}}{d\lambda} = \boldsymbol{J}_{\mathrm{W}}^{\dagger} \frac{\mathrm{d}\boldsymbol{p}}{d\lambda}. \tag{15}$$

The problem is now to find a suitable $\boldsymbol{W}$. Consider the simplified dynamic model

$$\boldsymbol{\tau} = \boldsymbol{H} \frac{d\boldsymbol{q}}{d\lambda} \ddot{\lambda} \tag{16}$$

obtained from (1) and (4) neglecting velocity-dependent and gravitational terms.

Symmetric and constant bounds on the joint torques are considered here; if the bounds were not symmetric a suitable transformation can be performed. Therefore, it is possible to define the $2n$ bounding inequalities for the joint torques

$$|\tau_i| \le \tau_i^{\max}, \quad i = 1, \ldots, n \tag{17}$$

In the general case when the bounds of each joint torque are not equal, it is useful to scale them to have normalized joint torques, that is

$$\tilde{\boldsymbol{\tau}} = \boldsymbol{L}^{-1}\boldsymbol{\tau}, \tag{18}$$

where $\boldsymbol{L} = \mathrm{diag}(\tau_1^{\max}, \ldots, \tau_n^{\max})$ is the scaling matrix. At this point, the inequalities (17) can be written in a compact form as

$$||\tilde{\boldsymbol{\tau}}||_{\infty} \le 1 \tag{19}$$

where $||\tilde{\boldsymbol{\tau}}||_{\infty} = \max(|\tilde{\tau}_1| \quad, \ldots, |\tilde{\tau}_n|)$. Equation (19) defines a hypercube in the space of normalized joint torques.

The simplified dynamic model (16) becomes

$$\tilde{\boldsymbol{\tau}} = \boldsymbol{L}^{-1}\boldsymbol{H} \frac{d\boldsymbol{q}}{d\lambda} \ddot{\lambda}. \tag{20}$$

For a given $\frac{dq}{d\lambda}$, and in turn for a given direction of $\tilde{\boldsymbol{\tau}}$, the module of $\tilde{\boldsymbol{\tau}}$ linearly depends on the parameter acceleration $\ddot{\lambda}$. The maximum and minimum values (equal and opposite) of $\ddot{\lambda}$ are obtained when the normalized torques are on their bound:

$$\tilde{\boldsymbol{\tau}}_{\mathrm{bound}} = \boldsymbol{L}^{-1}\boldsymbol{H} \frac{d\boldsymbol{q}}{d\boldsymbol{\lambda}} \ddot{\lambda}_{max} \tag{21}$$

from which it can be concluded that

$$\ddot{\lambda}_{max}^2 = \frac{\tilde{\boldsymbol{\tau}}_{\mathrm{bound}}^{\mathrm{T}} \tilde{\boldsymbol{\tau}}_{\mathrm{bound}}}{\frac{dq^{\mathrm{T}}}{d\lambda} [(\boldsymbol{L}^{-1}\boldsymbol{H})^{\mathrm{T}} \boldsymbol{L}^{-1} H] \frac{dq}{d\lambda}} \tag{22}$$

At the numerator of (22) there is the Euclidean norm of the maximum torques which depends on the direction of $\tilde{\boldsymbol{\tau}}_{max}$ and then on $\frac{dq}{d\lambda}$. Maximization of (22) with respect to $\frac{dq}{d\lambda}$ is thus difficult.

If a more conservative assumption is made, that is the available torques belong to the hypersphere

$$\tilde{\boldsymbol{\tau}}^{\mathrm{T}} \tilde{\boldsymbol{\tau}} \le 1, \tag{23}$$

https://doi.org/10.1017/S0263574702004678 Published online by Cambridge University Press

equation (22) simplifies to

$$\ddot{\lambda}^2_{max} = \frac{1}{\frac{dq^{\mathrm{T}}}{d\lambda}\left[(\boldsymbol{L}^{-1}\boldsymbol{H})^{\mathrm{T}}\,\boldsymbol{L}^{-1}\boldsymbol{H}\right]\frac{dq}{d\lambda}}. \tag{24}$$

To maximize (12) it is sufficient to minimize its denominator between all the $\frac{dq}{d\lambda}$ which satisfy the differential kinematics $\boldsymbol{J}\frac{dq}{d\lambda} = \frac{dp}{d\lambda}$. The problem is then a linear static optimization one with quadratic objective function

$$\min\left(\frac{d\boldsymbol{q}^T}{d\lambda}\boldsymbol{Q}\frac{d\boldsymbol{q}}{d\lambda}\right) \text{ with } \boldsymbol{J}\frac{d\boldsymbol{q}}{d\lambda} = \frac{d\boldsymbol{p}}{d\lambda} \tag{25}$$

where $\boldsymbol{Q} = (\boldsymbol{L}^{-1}\boldsymbol{H})^T\boldsymbol{L}^{-l}\boldsymbol{H}$.

The solution to (25) is in terms of the weighted pseudoinverse of the manipulator Jacobian

$$\frac{d\boldsymbol{q}}{d\lambda} = \boldsymbol{J}^{\dagger}_{\mathrm{Q}}\frac{d\boldsymbol{p}}{d\lambda} \tag{26}$$

with $\boldsymbol{J}^{\dagger}_{\mathrm{Q}}$ computed as in (14). By analyzing the structure of matrix $\boldsymbol{Q}$ it can be concluded that solution (26) penalizes the joints which have a bigger inertia-to-maximum torque ratio. This is particularly evident when matrix $\boldsymbol{H}$ is diagonal, since matrix $\boldsymbol{Q}$ is diagonal too with elements $\boldsymbol{Q}_{ii} = (H_{ii}/\tau_i^{\max})^2$.

So far it has been proposed a criterion to invert the points on the task-space path in order to find a joint-space path where to perform minimum-time path-following algorithms. However, many assumptions have been made: Dynamic model without velocity-dependent and gravitational terms; bounds on the joint torques expressed as in (23); kinematic inversion performed on each task-space point via (26) which only locally solves (25). There is no guarantee, indeed, that the algorithm gives the best trajectory; it is worth to investigate, however, if it could lead to better results that the use of the unweighted pseudoinverse.

To the purpose, introduce a scalar parameter γ and consider the solution to (25)

$$\frac{d\boldsymbol{q}}{d\lambda} = \boldsymbol{J}^{\dagger}_{Q^\gamma}\frac{d\boldsymbol{p}}{d\lambda} \tag{27}$$

with the weight matrix $\boldsymbol{Q}^{\gamma}$ defined as $\exp(\gamma\log(\boldsymbol{Q}))$.

When $\gamma = 0$ solution (27) is the one which uses the unweighted pseudoinverse of the Jacobian, when $\gamma = 1$ equation (27) coincides with (26).

4. IMPLEMENTATION DETAILS AND CASE STUDIES

To show the effectiveness of the proposed technique, it has been implemented for a three degree-of-freedom planar manipulator rnoving in the horizontal plane. The dynamic parameters of the manipulator's links in SI units are reported in the table below where ℓ_{i} is the length, ℓ_{ci} is the distance of the center of mass from the joint axis, m_i is the mass of the i-th link.

Link	ℓ_{i}	ℓ_{ci}	m_i
1	1.0	0.5	4.0
2	0.8	0.4	2.0
3	0.5	0.25	0.6

The link inertia has been set as $I_i = m_i\ell_i^2/12$, $i = 1,2,3$. The bounds on the joint torques are

$$\boldsymbol{\tau}_{\max} = -\boldsymbol{\tau}_{\min} = \begin{bmatrix} 50 \\ 10 \\ 2 \end{bmatrix}$$

The manipulator has been set in the initial configuration

$$\boldsymbol{q} = \begin{bmatrix} \frac{\pi}{2} \\ -\frac{\pi}{2} \\ -\frac{\pi}{2} \end{bmatrix}$$

and a number of paths have been considered which require different motions. On each path 300 points have been considered; for each point a kinematic inversion, using (27) a closed-loop inverse kinematic algorithm[13] has been performed with γ varying from 0 to 1.2 in step of 0.05. For each γ the obtained joint-space path consists of interpolating cubic splines, that give sufficient small interpolating error. Finally, the algorithm proposed by Shin and McKay[2] has been applied and the execution times computed.

All the tested paths are plotted in Figure 1 together with the initial configuration of the manipulator.

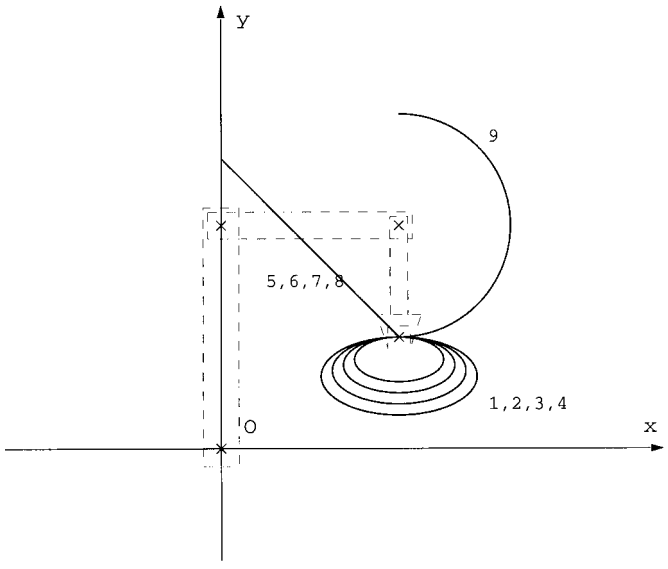


Fig. 1. Test paths together with a manipulator's initial configuration.

https://doi.org/10.1017/S0263574702004678 Published online by Cambridge University Press

The first four paths are ellipses of different dimensions. This is aimed to test the technique on similar paths of growing length so as to evaluate the influence of the local scope of the solution (27). The ellipses' equations are

$$\boldsymbol{p}(\lambda) = \begin{bmatrix} c_x + r_x \sin(2\pi\lambda) \\ c_y + r_y \cos(2\pi\lambda) \end{bmatrix}$$

with the following parameters

Path	c_x	r_x	c_y	r_y
1	0.8	0.20	0.400	0.10
2	0.8	0.25	0.375	0.15
3	0.8	0.30	0.350	0.20
4	0.8	0.35	0.325	0.25

The execution times as a function of the parameter γ are reported in Figure 2. Execution times when the unweighted pseudoinverse is used ($\gamma = 0$) are $t_1 = 0.8020$, $t_2 = 0.8982$, $t_3 = 0.9870$, $t_4 = 1.0709$. It can be noticed that by using the weighted pseudoinverse solution improvements are obtained for all the paths.

Then, four straight path are tested with different lengths. They have been chosen since they are very challenging for a manipulator with rotational joints. The path equations

$$\boldsymbol{p}(\lambda) = \begin{bmatrix} 0.8 - \beta\lambda \\ 0.5 + \beta\lambda \end{bmatrix}$$

with the following parameters

Path	β
5	0.1
6	0.2
7	0.4
8	0.8

The execution times as a function of the parameter γ are reported in Figure 3. Again improvements are obtained with respect to the unweighted pseudoinverse and they are bigger than before; in one case there is an improvement of rmore than 25%.

The final path is a half-circular path which would require only motion of the third link. The path is

$$\boldsymbol{p}(\lambda) = \begin{bmatrix} 0.8 + 0.5 \sin(\pi\lambda) \\ 1 - 0.5 \cos(\pi\lambda) \end{bmatrix}$$

Also in this case the use of a weighted pseudoinverse results in an advantage (Figure 4).

5. CONCLUSIONS

The minimum-time path-following control problem for a redundant manipulator when a task-space path is assigned,

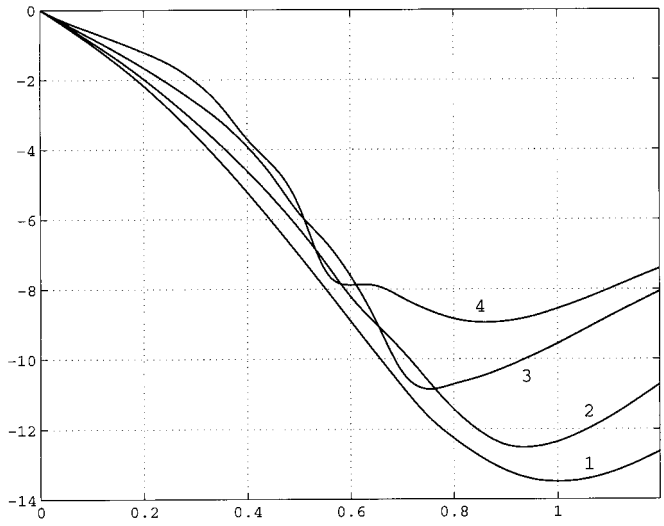


Fig. 2. Percentage variation of the execution time versus γ for elliptical paths 1,2, 3, 4.

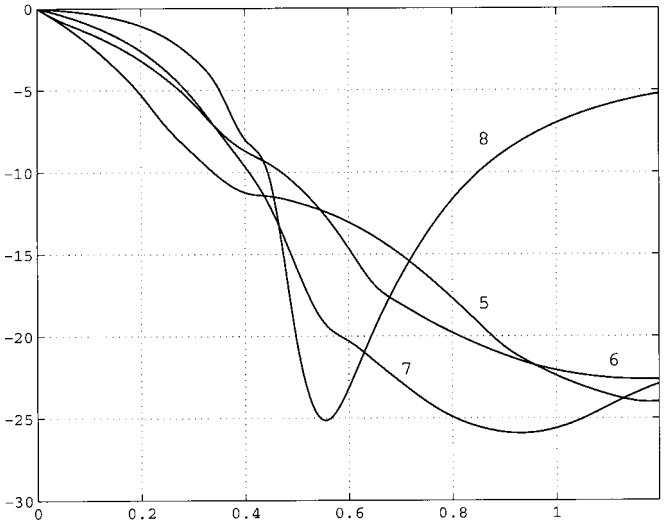


Fig. 3. Percentage variation of the execution time versus γ for straight paths 5, 6, 7, 8.

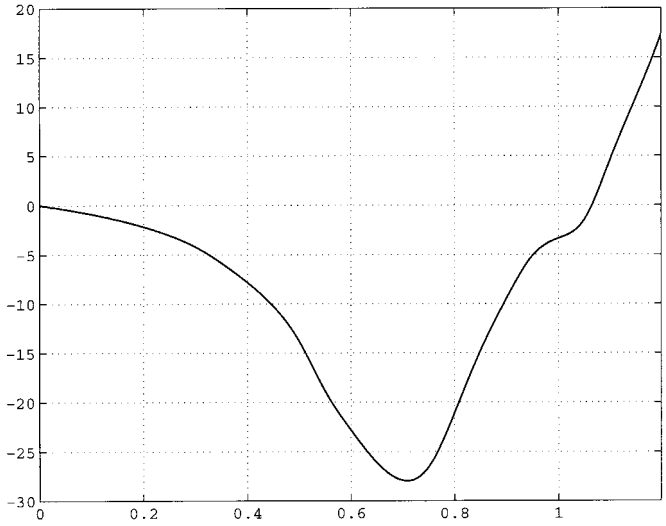


Fig. 4. Percentage variation of the execution time versus γ for the semi-circular path.

https://doi.org/10.1017/S0263574702004678 Published online by Cambridge University Press

has been investigated. It has been shown that existing algorithms for nonredundant manipulators cannot be simply extendend to handle kinematic redundancies. The inverse kinematic problem that underlies the minimum-time path-following problem has been solved by resorting to a suitable weighted pseudoinverse of the manipulator Jacobian matrix. This leads to a penalized motion of the joints with a higher inertia-to-torque ratio. A number of case studies have been presented for a three degree-of-freedom planar manipulator on two-dimensional paths. The obtained results have confirmed that the proposed technique improves path-execution times with respect to previous methods which propose the use of the unweighted pseudoinverse.

ACKNOWLEDGMENTS

This work was supported by *Ministero dell'Università e della Ricerca Scientifica e Tecnologica.*

References

1. J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," *ASME J. Dyn. Sys., Meas., Contr.* **106**, 102–106 (1984).
2. K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Trans. on Automatic Control* **AC–30**, 531–541 (1985).
3. J. E. Bobrow, S. Dubowsky and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *Int. J. Robotics Research* **4**, No. 3, 3–17 (1985).
4. F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE J. of Robotics and Automation* **RA–3**, 115–123 (1987).
5. J.-J. E. Slotine and H. S. Yang, "Improving the efficiency of time-optimal path following algorithms," *IEEE Trans. on Robotics and Automation* **5**, 118–124 (1989).
6. Y. Chen and A. A. Desrochers, "Structure of rminimum-time control law for robotic manipulators with constrained path," *IEEE Trans. on Robotics and Antomation* **6**, 388–393 (1990).
7. O. Dahl and L. Nielsen, "Torque-limited path following by on-line trajectory time scaling," *IEEE Trans. on Robotics and Automation* **6**, 554–561 (1994).
8. O. Dahl, "Path-constrained robot control with limited torques – Experimental Evaluation," *IEEE Trans. on Robotics and Automation* **10**, 658–669 (1997).
9. Z. Shiller and H.-H. Lu, "Computation of path constrained time optimal motions with dynamic singularities," *ASME J. Dyn. Sys., Meas., Contr.* **114**, 34–40 (1992).
10. Z. Schiller, "On singular time-optimal control along specified paths," *IEEE Trans. on Robotics and Automation* **10**, 561–566 (1993).
11. D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. on Man-Machine Systems* **MMS–10**, 47–53 (1969).
12. L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE Jour. of Robotics and Automation* **RA–4**, 303–3l0 (1988).
13. P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. of Robotics Research* **10**, 410–425 (1984).
14. P. Chiacchio, "Exploiting redundancy in minimum-time path following robot control," *Proc. of 1990 American Control Conf.* San Diego, CA (1990) pp. 2313–2318.

https://doi.org/10.1017/S0263574702004678 Published online by Cambridge University Press