SAMS, 1999, Vol. 35, pp. 45–60
Reprints available directly from the publisher
Photocopying permitted by license only

THE UNIT QUATERNION: A USEFUL TOOL FOR INVERSE KINEMATICS OF ROBOT MANIPULATORS

STEFANO CHIAVERINI [a,b,*] and BRUNO SICILIANO [a]

[a] *PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, via Claudio 21, 80125 Napoli, Italy;*
[b] *Dipartimento di Automazione, Elettromagnetismo, Ingegneria dell'Informazione e Matematica Industriale, Università degli Studi di Cassino, via G. Di Biasio 43, 03043 Cassino (FR), Italy*

(Received 10 June 1998)

Kinematic control of robot manipulators requires the computation of reference trajectories for the joint servos, through an inverse kinematics, to execute a given end-effector trajectory. A key point is concerned with end-effector orientation description which may lead to the occurrence of the so-called representation singularities. To overcome this drawback, in this paper a closed-loop inverse kinematics algorithm is presented where the unit quaternion is conveniently adopted in lieu of the Euler angles. The implementation on an industrial robot manipulator in a number of case studies is described.

Keywords: Robot manipulators; inverse kinematics; Euler angles; unit quaternion; Jacobian; Lyapunov stability

1. INTRODUCTION

Robot manipulation tasks are typically specified in terms of the position and orientation of an end-effector frame with respect to a base frame. Position is uniquely described by the Cartesian coordinates of the origin of the end-effector frame, whereas various representations of orientation exist.

*Corresponding author.

A complete description of orientation is given by the rotation matrix. Its columns are the unit vectors of the end-effector frame expressed in the base frame, yielding 9 parameters subject to 6 orthonormality constraints. On the other hand, minimal representations of orientation are defined by 3 parameters, *i.e.*, Euler angles; these have the drawback of representation singularities. Alternative descriptions of orientation in terms of 4 parameters with 1 norm constraint are respectively the axis/angle and the quaternion representation, see *e.g.* [1].

When motion is of concern, it is necessary to consider also the end-effector linear and angular velocity. With reference to the above descriptions of orientation, the relationship between the angular velocity and the time derivatives of each set of parameters should be considered. This is also important for end-effector trajectory planning.

In order to realize a kinematic control scheme, the corresponding joint trajectories to a given end-effector trajectory have to be computed; these constitute the reference inputs to the robot joint servos. Therefore, the solution to an inverse kinematics problem is required [2].

Finding an analytical solution to the inverse kinematics is possible only for nonredundant kinematic structures of simple geometry [3]. A well-established method to solve the inverse kinematics problem for general kinematic structures along a given end-effector trajectory is constituted by the closed-loop inverse kinematics (CLIK) algorithm [4] which is based on the manipulator Jacobian.

The above-mentioned problem of end-effector orientation description has an impact on the solution of the inverse kinematics problem. On one hand, whatever technique is used, the computation of a rotation matrix is mandatory since orientation descriptions based on a reduced number of parameters cannot be expressed directly as a function of the joint variables. On the other hand, the use of an inverse kinematics algorithm leads to considering a proper expression of the orientation error depending on the particular orientation description.

As opposed to previous CLIK algorithms [5, 6], this work is aimed at developing a new algorithm using the unit quaternion for describing end-effector orientation, with the advantage of avoiding singularities associated with minimal representations of orientation. The algorithmic error is extracted from the vector part of the quaternion expressing the mutual orientation between the desired and the current

end-effector frame. The rotation matrix available from the manipulator direct kinematics equation is utilized to compute the end-effector quaternion at the current joint configuration.

The implementation of the algorithm on a six-joint industrial robot manipulator with a nonspherical wrist is described by means of numerical case studies, and a comparison with an algorithm based on the Euler angles description is carried out.

2. DESCRIPTION OF ORIENTATION

The location of a rigid body in space is typically described in terms of the (3×1) position vector $\boldsymbol{p}$ and the (3×3) rotation matrix $\boldsymbol{R}$ describing the origin and the orientation of a frame attached to the body with respect to a fixed reference frame.

A minimal representation of orientation can be obtained by using a set of three Euler angles $\varphi = [\alpha\, \beta\, \gamma]^{\mathrm{T}}$. Among the 12 possible definitions of Euler angles, without loss of generality, the XYZ representation is considered leading to the rotation matrix

$$\begin{aligned} \boldsymbol{R}(\boldsymbol{\varphi}) &= \boldsymbol{R}_x(\alpha)\boldsymbol{R}_y(\beta)\boldsymbol{R}_z(\gamma) \\ &= \begin{bmatrix} c_\beta c_\gamma & -c_\beta s_\gamma & s_\beta \\ s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & -s_\alpha c_\beta \\ -c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma & c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma & c_\alpha c_\beta \end{bmatrix} \end{aligned} \tag{1}$$

where $\boldsymbol{R}_x$, $\boldsymbol{R}_y$, $\boldsymbol{R}_z$ are the matrices of the elementary rotations about three independent axes of successive frames; the notations c_ϕ and s_ϕ are the abbreviations for $\cos\phi$ and $\sin\phi$, respectively.

The set of the Euler angles corresponding to a given rotation matrix

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2}$$

is

$$\begin{aligned} \alpha &= \operatorname{Atan}2(-r_{23}, r_{33}) \\ \beta &= \operatorname{Atan}2\left(r_{13}, \sqrt{r_{11}^2 + r_{12}^2}\right) \\ \gamma &= \operatorname{Atan}2(-r_{12}, r_{11}) \end{aligned} \tag{3}$$

with $\beta \in (-\pi/2, \pi/2)$, whereas the solution is

$$\begin{aligned} \alpha &= \text{Atan}\,2(r_{23}, -r_{33}) \\ \beta &= \text{Atan}\,2\left(r_{13}, -\sqrt{r_{11}^2 + r_{12}^2}\right) \\ \gamma &= \text{Atan}\,2(r_{12}, -r_{11}) \end{aligned} \tag{4}$$

with $\beta \in (\pi/2, 3\pi/2)$; the function Atan $2(y, x)$ computes the arctangent of the ratio y/x but utilizes the sign of each argument to determine which quadrant the resulting angle belongs to.

Solutions (3) and (4) degenerate when $\beta = \pm \pi/2$; in this case, it is possible to determine only the sum or difference of α and γ, *i.e.*,

$$\alpha \pm \gamma = \text{Atan}\,2(r_{21}, r_{22}) \tag{5}$$

where the plus sign applies for $\beta = +\pi/2$ and the minus sign applies for $\beta = -\pi/2$. These configurations are termed representation singularities. It should be remarked that each of the other Euler angles description suffers from the occurrence of two representation singularities.

The relationship between the time derivative of the Euler angles $\dot{\boldsymbol{\varphi}}$ and the body angular velocity $\boldsymbol{\omega}$ is given by

$$\boldsymbol{\omega} = \boldsymbol{T}(\boldsymbol{\varphi})\dot{\boldsymbol{\varphi}}, \tag{6}$$

where the transformation matrix $\boldsymbol{T}$ corresponding to the above XYZ description is

$$\boldsymbol{T}(\boldsymbol{\varphi}) = \begin{bmatrix} 1 & 0 & s_\beta \\ 0 & c_\alpha & -s_\alpha c_\beta \\ 0 & s_\alpha & c_\alpha c_\beta \end{bmatrix}. \tag{7}$$

It can be recognized that $\boldsymbol{T}$ becomes singular at the representation singularities $\beta = \pm \pi/2$; notice that, in these configurations, it is not possible to describe an arbitrary angular velocity with a set of Euler angles time derivatives.

To solve the representation singularity problem, an alternative description of orientation can be obtained by resorting to a four-parameter representation in terms of a rotation angle ϑ about an axis in space described by the (3×1) unit vector $\boldsymbol{r}$. The resulting rotation

matrix is

$$\boldsymbol{R}(\vartheta, \boldsymbol{r}) = c_\vartheta \boldsymbol{I} + (1 - c_\vartheta)\boldsymbol{r}\boldsymbol{r}^{\mathrm{T}} - s_\vartheta \boldsymbol{S}(\boldsymbol{r}) \tag{8}$$

where $\boldsymbol{I}$ is the (3×3) identity matrix and $\boldsymbol{S}(\cdot)$ is the matrix operator performing the cross product between two (3×1) vectors.

It is clear that $\boldsymbol{R}(-\vartheta, -\boldsymbol{r}) = \boldsymbol{R}(\vartheta, \boldsymbol{r})$, *i.e.*, a rotation by $-\vartheta$ about $-\boldsymbol{r}$ cannot be distinguished from a rotation by ϑ about $\boldsymbol{r}$; hence, the angle/axis representation is not unique. This drawback can be overcome by a different four-parameter description; namely, the Euler parameters or unit quaternion defined as

$$\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\} \tag{9}$$

where

$$\begin{aligned} \eta &= \cos\frac{\vartheta}{2} \\ \boldsymbol{\epsilon} &= \sin\frac{\vartheta}{2}\boldsymbol{r}, \end{aligned} \tag{10}$$

with $\eta \geq 0$ for $\vartheta \in [-\pi, \pi]$; η is called the scalar part of the quaternion while $\boldsymbol{\epsilon}$ is called the vector part of the quaternion.

It is worth remarking that, differently from the angle/axis description, a rotation by $-\vartheta$ about $-\boldsymbol{r}$ gives the same quaternion as the one associated with a rotation by ϑ about $\boldsymbol{r}$; this solves the above nonuniqueness problem. The rotation matrix corresponding to a given quaternion is

$$\boldsymbol{R}(\eta, \boldsymbol{\epsilon}) = (\eta^2 - \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon})\boldsymbol{I} + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^{\mathrm{T}} - 2\eta \boldsymbol{S}(\boldsymbol{\epsilon}). \tag{11}$$

On the other hand, the unit quaternion corresponding to a given rotation matrix (2) is

$$\begin{aligned} \eta &= \frac{1}{2}\sqrt{r_{11} + r_{22} + r_{33} + 1} \\ \boldsymbol{\epsilon} &= \begin{bmatrix} \frac{1}{2}\mathrm{sgn}(r_{32} - r_{23})\sqrt{r_{11} - r_{22} - r_{33} + 1} \\ \frac{1}{2}\mathrm{sgn}(r_{13} - r_{31})\sqrt{r_{22} - r_{33} - r_{11} + 1} \\ \frac{1}{2}\mathrm{sgn}(r_{21} - r_{12})\sqrt{r_{33} - r_{11} - r_{22} + 1} \end{bmatrix}. \end{aligned} \tag{12}$$

The relationship between the time derivative of the unit quaternion and the body angular velocity $\boldsymbol{\omega}$ is established by the so-called

propagation equation

$$\begin{bmatrix} \dot{\eta} \\ \dot{\boldsymbol{\epsilon}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\boldsymbol{\omega}^{\mathrm{T}} \\ \boldsymbol{\omega} & -\boldsymbol{S}(\boldsymbol{\omega}) \end{bmatrix} \begin{bmatrix} \eta \\ \boldsymbol{\epsilon} \end{bmatrix} \tag{13}$$

to be used with suitable initial conditions.

3. INVERSE KINEMATICS SCHEMES

The direct kinematics equation for a serial-chain robot manipulator can be written in terms of the vector $\boldsymbol{p}(\boldsymbol{q})$ and the rotation matrix $\boldsymbol{R}(\boldsymbol{q})$, where $\boldsymbol{q}$ is the $(n \times 1)$ vector of joint variables, and $\boldsymbol{p}$ and $\boldsymbol{R}$ respectively denote the position and the orientation of a frame attached to the end effector.

The differential kinematics is characterized by the relationship between the end-effector linear velocity $\dot{\boldsymbol{p}}$ and angular velocity $\boldsymbol{\omega}$ and the joint velocities $\dot{\boldsymbol{q}}$, *i.e.*,

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{14}$$

where $\boldsymbol{J}$ is the manipulator geometric Jacobian [2]. In the general case, this is a $(6 \times n)$ matrix which may be singular. When $n > 6$, kinematic redundancy exists which can be handled by suitable inverse kinematics techniques, *e.g.* [7–11]. On the other hand, when rank $(\boldsymbol{J}) < 6$, a kinematic singularity occurs and robustness of inverse kinematics solutions can be gained as in, *e.g.* [12–14]. Since the focus of the present work is on the problems related to orientation description, without loss of generality, the Jacobian is assumed to be a nonsingular square matrix.

Let $\boldsymbol{p}_d(t)$ and $\boldsymbol{R}_d(t)$ denote the desired end-effector trajectory. The inverse kinematics problem consists of computing the corresponding joint trajectories $\boldsymbol{q}(t)$, which are to be used as the reference inputs to the robot joint servos. Analytical inverse kinematics solutions can be found only for nonredundant manipulators having a simple geometry, *e.g.*, with a spherical wrist.

An effective solution which is applicable to any kinematic structure is based on the time integration of the joint velocities obtained by

inversion of (14). In order to overcome the typical numerical drift concerned with discrete-time implementation of this solution, a closed-loop inverse kinematics (CLIK) algorithm can be devised which acts upon an error characterizing the displacement between the desired and the current end-effector trajectory.

Let

$$\boldsymbol{e}_P = \boldsymbol{p}_d - \boldsymbol{p}(\boldsymbol{q}) \tag{15}$$

denote the end-effector position error. The end-effector orientation error depends on the choice of the orientation description. In the case of Euler angles, the error is simply

$$\boldsymbol{e}_{O,\mathrm{Eul}} = \boldsymbol{\varphi}_d - \boldsymbol{\varphi}(\boldsymbol{q}) \tag{16}$$

where $\boldsymbol{\varphi}_d$ and $\boldsymbol{\varphi}$ denote the desired and current quantities, respectively. Notice that the computation of $\boldsymbol{\varphi}(\boldsymbol{q})$ requires the extraction of the Euler angles from the end-effector rotation matrix $\boldsymbol{R}$ *via* the formulae (3) or (4).

The joint velocity algorithmic solution is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}_A^{-1}(\boldsymbol{q}) \begin{bmatrix} \dot{\boldsymbol{p}}_d + \boldsymbol{K}_P \boldsymbol{e}_P \\ \dot{\boldsymbol{\varphi}}_d + \boldsymbol{K}_O \boldsymbol{e}_{O,\mathrm{Eul}} \end{bmatrix} \tag{17}$$

where $\boldsymbol{K}_P$ and $\boldsymbol{K}_O$ are suitable positive definite matrix gains, and $\boldsymbol{J}_A$ is the analytical Jacobian [2] that is related to the geometric Jacobian by the relationship

$$\boldsymbol{J}_A = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{O} \\ \boldsymbol{O} & \boldsymbol{T}^{-1}(\boldsymbol{\varphi}) \end{bmatrix} \boldsymbol{J} \tag{18}$$

with $\boldsymbol{T}$ as in (7) and $\boldsymbol{O}$ is the (3×3) null matrix. Note that $\boldsymbol{J}_A$ is singular at a representation singularity, where it is not possible to describe an arbitrary set of Euler angles time derivatives with a set of joint velocities.

In view of (15) and (16), substituting (17) into (14) gives

$$\dot{\boldsymbol{e}}_P + \boldsymbol{K}_P \boldsymbol{e}_P = \boldsymbol{0} \tag{19}$$

$$\dot{\boldsymbol{e}}_{O,\mathrm{Eul}} + \boldsymbol{K}_O \boldsymbol{e}_{O,\mathrm{Eul}} = \boldsymbol{0}. \tag{20}$$

It is easy to see that system (19) is exponentially stable, implying that $\boldsymbol{e}_P$ converges to zero. Likewise, system (20) is exponentially stable as long as no representation singularity occurs.

The block scheme of the resulting CLIK algorithm using the Euler angles is shown in Figure 1.

In order to devise an inverse kinematics scheme based on the unit quaternion, a suitable end-effector orientation error shall be defined. Let $\mathcal{Q}_d = \{\eta_d, \boldsymbol{\epsilon}_d\}$ and $\mathcal{Q} = \{\eta, \boldsymbol{\epsilon}\}$ represent the unit quaternions associated with $\boldsymbol{R}_d$ and $\boldsymbol{R}$, respectively. The mutual orientation can be expressed in terms of the unit quaternion $\Delta\mathcal{Q} = \{\Delta\eta, \Delta\boldsymbol{\epsilon}\}$ where

$$\Delta\eta = \eta\eta_d + \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon}_d \tag{21}$$

$$\Delta\boldsymbol{\epsilon} = \eta\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon} - \boldsymbol{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}. \tag{22}$$

It can be recognized that $\Delta\mathcal{Q} = \{1, \boldsymbol{0}\}$ if and only if $\boldsymbol{R}$ and $\boldsymbol{R}_d$ are aligned, and thus it is sufficient to consider $\Delta\boldsymbol{\epsilon}$ to express and end-effector orientation error in the form

$$\boldsymbol{e}_{O,\mathrm{Quat}} = \eta(\boldsymbol{q})\boldsymbol{\epsilon}_d - \eta_d\boldsymbol{\epsilon}(\boldsymbol{q}) - \boldsymbol{S}(\boldsymbol{\epsilon}_d)\boldsymbol{\epsilon}(\boldsymbol{q}). \tag{23}$$

Notice that the explicit computation of $\eta(\boldsymbol{q})$ and $\boldsymbol{\epsilon}(\boldsymbol{q})$ is not possible, but it requires the intermediate computation of the rotation matrix $\boldsymbol{R}(\boldsymbol{q})$ that is available from the manipulator direct kinematics; then, the unit quaternion can be extracted using the formulae (12).

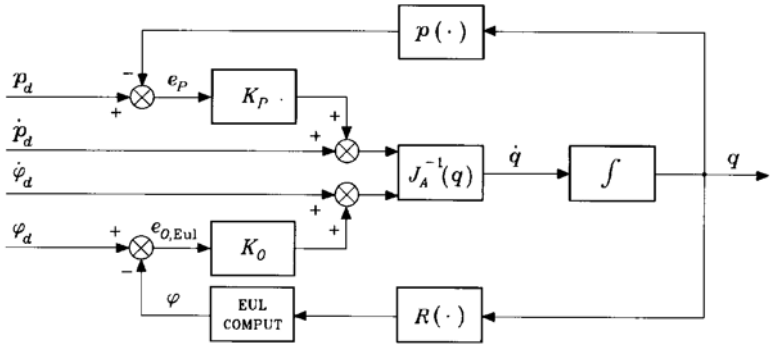

FIGURE 1 CLIK algorithm using the Euler angles.

With the unit quaternion-based orientation error (23), the joint velocity algorithmic solution is given by

$$\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q}) \begin{bmatrix} \dot{\boldsymbol{p}}_d + \boldsymbol{K}_P \boldsymbol{e}_P \\ \boldsymbol{\omega}_d + \boldsymbol{K}_O \boldsymbol{e}_{O,\text{Quat}} \end{bmatrix}. \tag{24}$$

It is worth pointing out that in (24), differently from the previous Euler angles based algorithm in (17), the geometric Jacobian appears in lieu of the analytical Jacobian.

In view of (15) and (23), substituting (24) into (14) gives (19) and

$$\boldsymbol{\omega}_d - \boldsymbol{\omega} + \boldsymbol{K}_O \boldsymbol{e}_{O,\text{Quat}} = \boldsymbol{0}. \tag{25}$$

It should be observed that now the orientation error equation is not homogeneous in $\boldsymbol{e}_{O,\text{Quat}}$ since it contains the end-effector angular velocity error instead of the time derivative of the orientation error.

To study stability of system (25), consider the positive definite Lyapunov function [15]

$$V = (\eta_d - \eta)^2 + (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon})^{\mathrm{T}} (\boldsymbol{\epsilon}_d - \boldsymbol{\epsilon}). \tag{26}$$

In view of the quaternion propagation (13), the time derivative of V along the trajectories of systems (25) is given by

$$\dot{V} = -\boldsymbol{e}_{O,\text{Quat}}^{\mathrm{T}} \boldsymbol{K}_O \boldsymbol{e}_{O,\text{Quat}} \tag{27}$$

which is negative definite, implying that $\boldsymbol{e}_{O,\text{Quat}}$ converges to zero.

The block scheme of the resulting CLIK algorithm using the unit quaternion is shown in Figure 2.

4. CASE STUDIES

The CLIK algorithms using the Euler angles and the unit quaternion, respectively, have been tested on the industrial robot Comau SMART-3 S available in the PRISMA Lab (Fig. 3). This is a six-revolute-joint manipulator having an anthropomorphic geometry with both shoulder

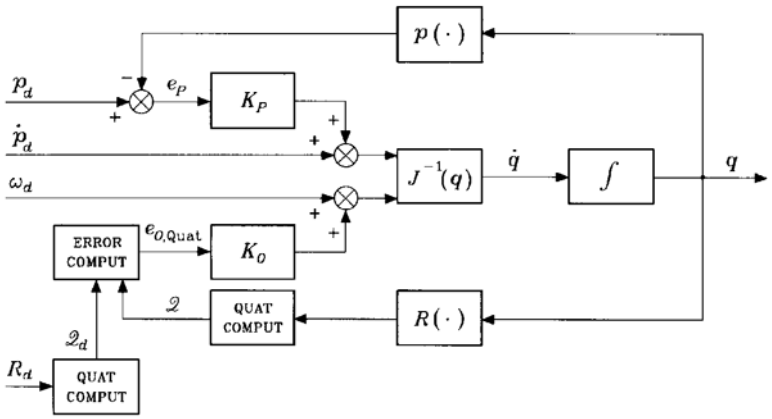

FIGURE 2 CLIK algorithm using the unit quaternion.

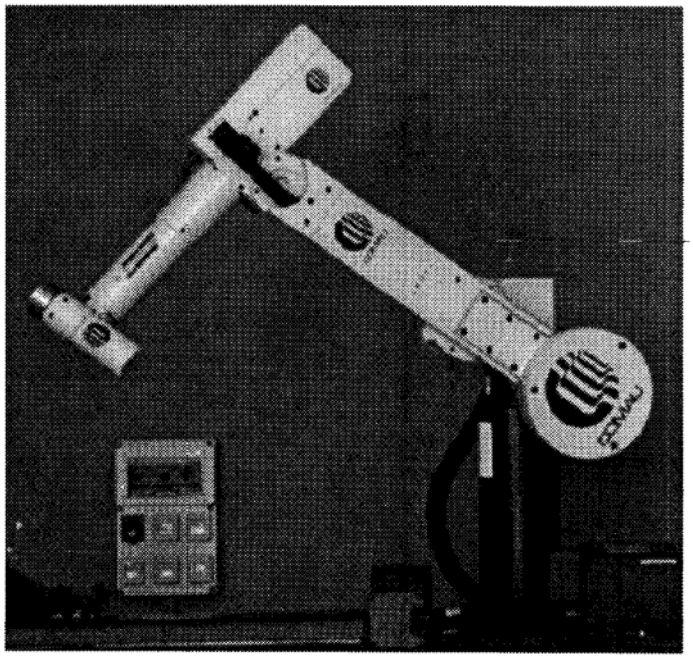

FIGURE 3 Comau SMART-3 S industrial robot.

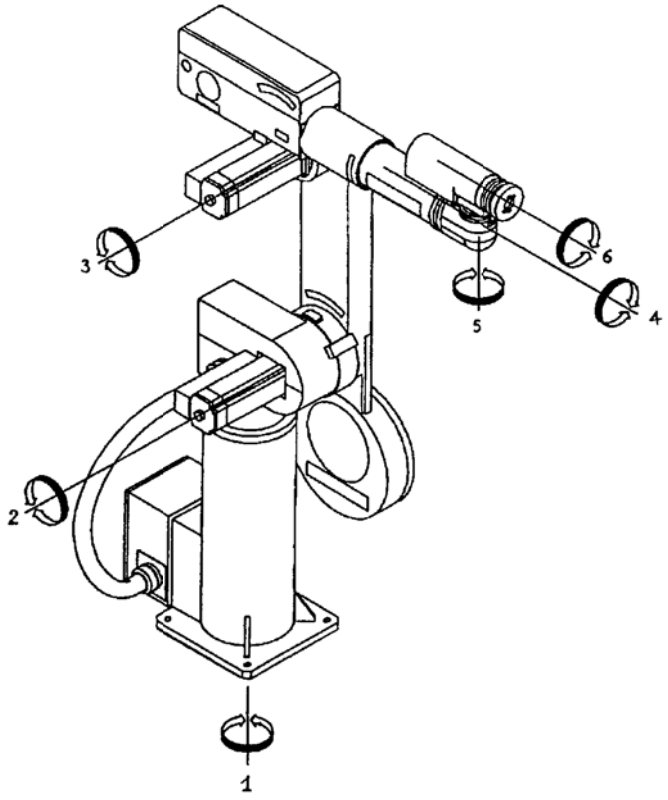

FIGURE 4 Kinematic structure of Comau SMART-3 S robot manipulator.

and elbow offset and a non-spherical wrist. A sketch of the kinematic structure is depicted in Figure 4. The manipulator direct kinematics equation and geometric Jacobian can be determined on the basis of the well-known Denavit-Hartenberg convention.

As a first case study, the following task has been considered. The manipulator is placed at the initial joint configuration

$$\boldsymbol{q} = [\pi/2 \quad -\pi/4 \quad 0 \quad 0 \quad -\pi/2 \quad 0]^{\mathrm{T}}$$

corresponding to the desired initial end-effector position and orientation

$$\boldsymbol{p}_{di} = [0.1130 \quad 1.2057 \quad 0.1930]^{\mathrm{T}}$$
$$\boldsymbol{\varphi}_{di} = [-\pi/4 \quad 0 \quad -\pi/2]^{\mathrm{T}}.$$

The desired final end-effector position and orientation are

$$\boldsymbol{p}_{df} = [0.7130 \quad 0.8457 \quad 0.1930]^{\mathrm{T}}$$
$$\boldsymbol{\varphi}_{df} = [-\pi/3 \quad 0 \quad -\pi/2]^{\mathrm{T}}.$$

The initial and final path coordinates have been interpolated using 5th-order polynomial time laws such that null initial and final velocities and accelerations are obtained. The duration of the motion is 2 s. Notice that the orientation trajectory is generated using the Euler angles for both algorithms so as to test them on the same desired trajectory; to this purpose, $\boldsymbol{R}_d$ is computed from $\boldsymbol{\varphi}_d$ *via* (1) at each time instant, and then $\mathcal{Q}_d$ is extracted from $\boldsymbol{R}_d$.

The algorithms have been implemented in discrete time using Euler integration rule with 0.001 s time interval. The gains appearing in both (17) and (24) have been set to:

$$\boldsymbol{K}_P = \begin{bmatrix} 1000 & 0 & 0 \\ 0 & 1000 & 0 \\ 0 & 0 & 1000 \end{bmatrix}$$
$$\boldsymbol{K}_O = \begin{bmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{bmatrix}.$$

The computation to extract the quaternion from a given rotation matrix has been performed by using the efficient algorithm in [16] based on the solution (12).

The results are displayed in Figures 5 and 6 in terms of the time history of the joint angles, velocities, norms of end-effector position and orientation errors. It can be recognized how for both schemes the joint motion is smooth, while the tracking performance is excellent; further, the errors vanish at steady state.

The second case study is aimed at analyzing the behaviour of the above two CLIK algorithms when a representation singularity occurs

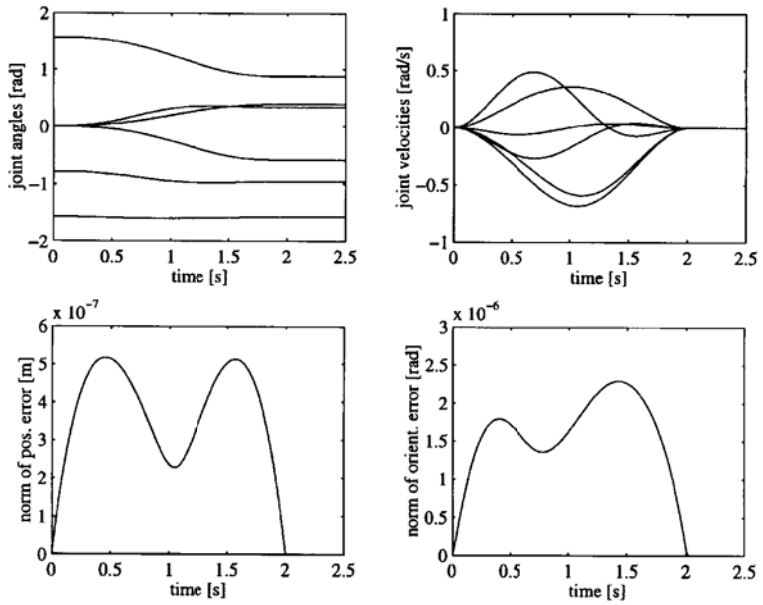

FIGURE 5 Results of discrete-time implementation of the Euler angles-based CLIK algorithm in the first case study.

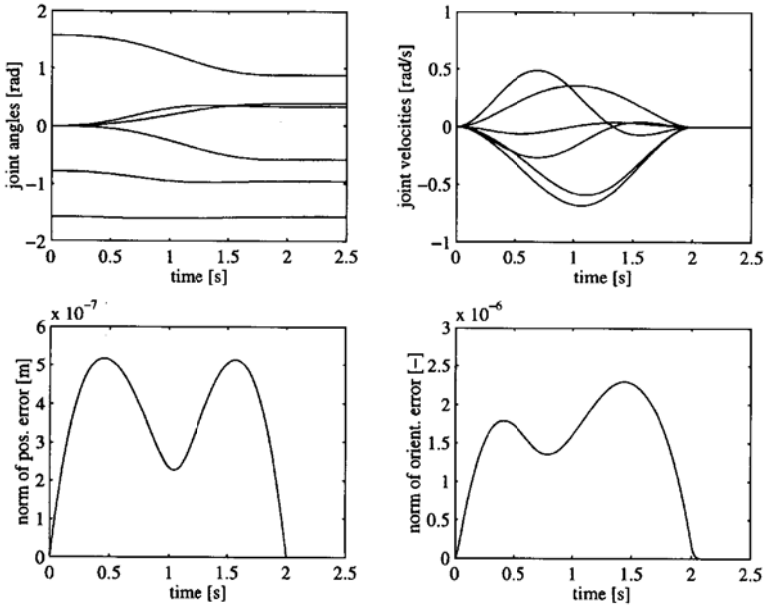

FIGURE 6 Results of discrete-time implementation of the unit quaternion-based CLIK algorithm in the first case study.

along the desired end-effector trajectory. The manipulator is placed at the same initial joint configuration, whereas the desired final end-effector position and orientation are

$$\boldsymbol{p}_{df} = [0.7130 \quad 0.8457 \quad 0.1930]^{\mathrm{T}}$$
$$\boldsymbol{\varphi}_{df} = [\pi/4 \quad \pi \quad -\pi/2]^{\mathrm{T}}.$$

The desired initial and final path coordinates have been interpolated like in the first case study.

The results are displayed in Figures 7 and 8 in terms of the time history of the determinant of the transformation matrix in (7), joint velocities, norms of end-effector position and orientation errors. It can be recognized how, in the neighbourhood of the representation singularity, the scheme based on the Euler angles generates very large values of joint velocities which ultimately cause incorrect inverse kinematics solutions thereafter. On the other hand, the scheme based on the unit quaternion does not suffer at all from the representation

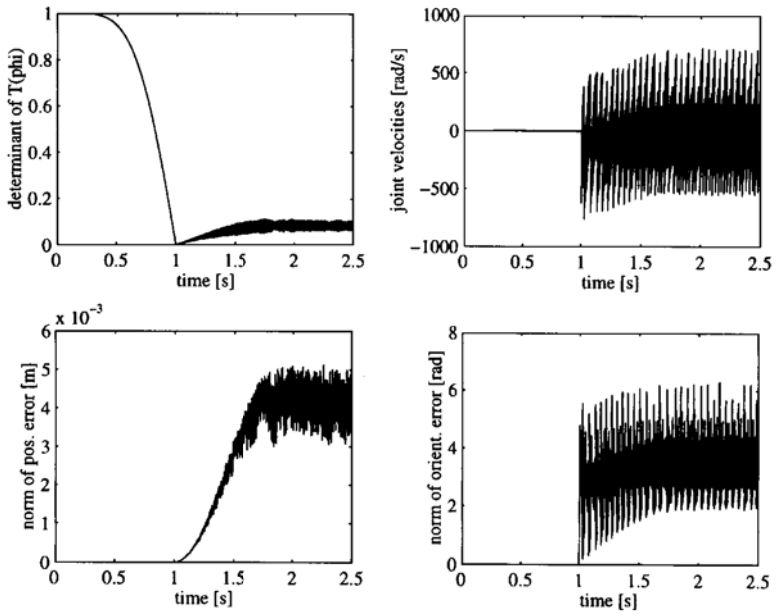

FIGURE 7 Results of discrete-time implementation of the Euler angles-based CLIK algorithm in the second case study.

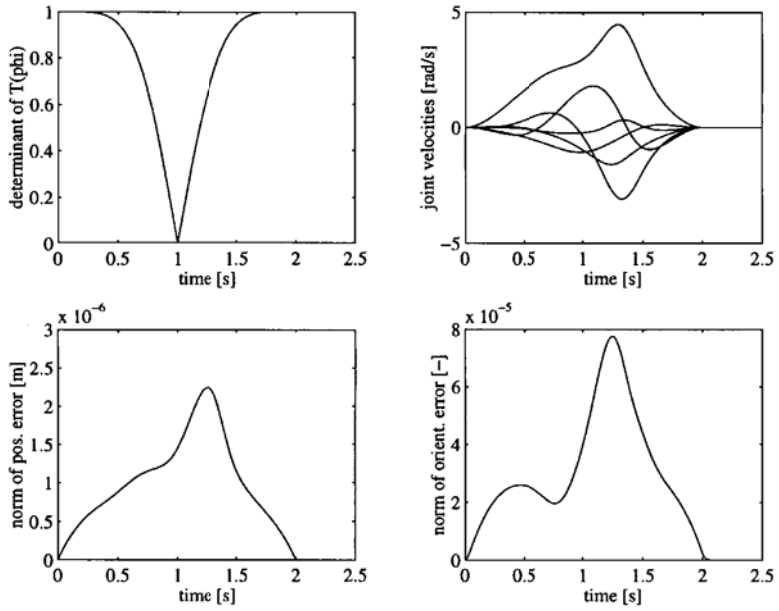

FIGURE 8 Results of discrete-time implementation of the unit quaternion-based CLIK algorithm in the second case study.

singularity in the desired orientation trajectory, and the tracking and steady-state performance remains as good as for the first case study.

5. CONCLUSION

An inverse kinematics scheme using the unit quaternion has been presented to solve the inverse kinematics problem for a robot manipulator along a given end-effector position and orientation trajectory. Convergence of the orientation error has been shown by resorting to a Lyapunov argument. A comparison with an inverse kinematics scheme using the Euler angles has been developed. The results of a numerical implementation on a six-joint industrial robot manipulator have demonstrated that the performance is comparable for the two schemes except when a representation singularity is encountered along the end-effector orientation trajectory, in which case the Euler angles-based algorithm fails.

Acknowledgements

This work was supported in part by *MURST* and in part by *ASI*.

References

[1] Roberson, R. E. and Schwertassek, R. (1988). *Dynamics of Multibody Systems*, Springer-Verlag, Berlin.
[2] Sciavicco, L. and Siciliano, B. (1996). *Modeling and Control of Robot Manipulators*, McGraw-Hill, New York.
[3] Pieper, D. L. (1968). The kinematics of manipulators under computer control, memo. AIM 72, Stanford Artificial Intelligence Laboratory.
[4] Siciliano, B. (1990). A closed-loop inverse kinematic scheme for on-line joint-based robot control, *Robotica*, **8**, 231–243.
[5] Balestrino, A., De Maria, G., Sciavicco, L. and Siciliano, B. (1988). An algorithmic approach to coordinate transformation for robotic manipulators, *Advanced Robotics*, **2**, 327–344.
[6] Chiacchio, P. and Siciliano, B. (1989). A closed-loop Jacobian transpose scheme for solving the inverse kinematics of nonredundant and redundant wrists, *J. of Robotic Systems*, **6**, 601–630.
[7] Liégeois, A. (1977). Automatic supervisory control of the configuration and behavior of multibody mechanisms, *IEEE Trans. on Systems, Man, and Cybernetics*, **7**, 868–871.
[8] Maciejewski, A. A. and Klein, C. A. (1985). Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments, *Int. J. of Robotics Research*, **4**(3), 109–117.
[9] Nakamura, Y., Hanafusa, H. and Yoshikawa, T. (1987). Task-priority based redundancy control of robot manipulators, *Int. J. of Robotics Research*, **6**(2), 3–15.
[10] Chiacchio, P., Chiaverini, S., Sciavicco, L. and Siciliano, B. (1991). 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.
[11] Chiaverini, S. (1997). Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators, *IEEE Trans. on Robotics and Automation*, **13**, 398–410.
[12] Nakamura, Y. and Hanafusa, H. (1986). Inverse kinematic solutions with singularity robustness for robot manipulator control, *ASME J. of Dynamic Systems, Measurement, and Control*, **108**, 163–171.
[13] Wampler, C. W. (1986). Manipulator inverse kinematic solutions based on damped least-squares solutions, *IEEE Trans. on Systems, Man, and Cybernetics*, **16**, 93–101.
[14] Chiaverini, S., Siciliano, B. and Egeland, O. (1994). Review of the damped least-squares inverse kinematics with experiments on an industrial robot manipulator, *IEEE Trans. on Control Systems Technology*, **2**, 123–134.
[15] Yuan, J.S.-C. (1988). Closed-loop manipulator control using quaternion feed-back, *IEEE J. of Robotics and Automation*, **4**, 434–440.
[16] Shepperd, S. W. (1978). Quaternion from rotation matrix, *AIAA J. of Guidance and Control*, **1**, 223–224.