Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator

Stefano Chiaverini, Bruno Siciliano, *Senior Member, IEEE*, and Olav Egeland, *Member, IEEE*

Abstract—**The goal of this paper is to present experimental results on the implementation of the damped least-squares method for the six-joint ABB IRb2000 industrial robot manipulator. A number of inverse kinematics schemes are reviewed which allow robot control through kinematic singularities. The basic scheme adopts a damped least-squares inverse of the manipulator Jacobian with a varying damping factor acting in the neighborhood of singularities. The effect of a weighted damped least-squares solution is investigated to provide user-defined accuracy capabilities along prescribed end-effector space directions. An on-line estimation algorithm is employed to measure closeness to singular configurations. A feedback correction error term is introduced to ensure algorithm tracking convergence and its effect on the joint velocity solution is discussed. Computational aspects are discussed in view of real-time implementation of the proposed schemes. Experimental case studies are developed to investigate manipulator performance in the case of critical end-effector trajectories passing through and near the shoulder and wrist singularities of the structure.**

I. Introduction

KINEMATIC SINGULARITIES have long been recognized as causing one of the most serious problems in programming and control of robotic manipulators. It is well-known that when a manipulator is at—or is in the neighborhood of—a singular configuration, severe restrictions may occur on its motion.

Avoiding or reducing the effects of singularities has been an attractive topic which has captured the attention of many researchers in robotics during the last decade. A remarkable number of methods and/or algorithms aimed at computing well-behaved or robust inverse kinematics solutions have been proposed in the literature, and many papers in the past have presented simulation results.

Close to a kinematic singularity, the usual inverse differential kinematics solutions based on Jacobian (pseudo-) inverse [1] become ill-conditioned, and this is experienced in the form of very high joint velocities [2] and large control deviations [3]. Nonetheless, when a pre-programmed reference end-effector trajectory is to be tracked, it is possible either to interpolate in joint coordinates close to singular configurations [4] or to plan motions so that singularities are avoided.

On the other hand, in real-time and sensory control of robotic manipulators, the reference trajectory is not known *a priori* and some remedies must be taken in order to counteract the unexpected occurrence of singularities. The same kind of problem is encountered in joy-stick control of a robot if the operator attempts to lead the robot through—or nearby—a singularity using end-effector motion increments.

As anticipated above, several techniques have been devised to handle kinematic singularities. In some techniques [5]–[6] the degenerate directions in the end-effector space associated with a given singularity are identified and the end-effector velocity components along those directions in a suitable neighborhood of the singularity are eliminated. Other techniques are based on a modification of the exact inverse differential kinematics mapping by resorting to approximate mappings that offer robustness to singularities at the expense of reduced tracking accuracy [7]–[9]. A recent overview of algorithmic schemes for robotic systems through singularities can be found in [10].

The effectiveness of the above schemes was extensively proven by means of theoretical analysis and simulation results. Nevertheless, we believe that both further theoretical refinements of the solutions and practical aspects related to implementation on real industrial manipulators are worthy of investigation.

In the framework of our joint research project, following the earlier satisfactory experimental results obtained on the five-joint ABB Trallfa TR400 manipulator [11], [12], in this paper we investigate the performance of a number of schemes for control in singular configurations on the six-joint ABB IRb 2000 manipulator. Due to mechanical joint limits, this arm does not have the typical elbow singularity, which in any case is less troublesome than the singularities experienced inside the robot workspace. These are indeed the shoulder and wrist singularities, and they both occur for the given robot arm.

The basic inverse kinematics solution is derived using a damped least-squares inverse [7], [8] of the end-effector Jacobian which attempts to trade off accuracy against feasibility of the solution. Also, a weighted damped least-squares inverse is adopted [7] where the use of proper weighting allows shaping of the solution along given end-effector space directions [13]. Singular value decomposition [14] of the

Manuscript received April 4, 1993; revised November 1993. This paper was recommended by Associate Editor D. Repperger. This paper was supported in part by Consiglio Nazionale delle Ricerche under Contract 93.00905.PF67, and by ABB Robotics.

S. Chiaverini and B. Siciliano are with the Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, via Claudio 21, 80125 Napoli, Italy.

O. Egeland is with the Institutt for Teknisk Kybernetikk, Norges Tekniske Høgskole, O.S. Bragstads plass 8, 7034 Trondheim, Norway.

IEEE Log Number 9401416.

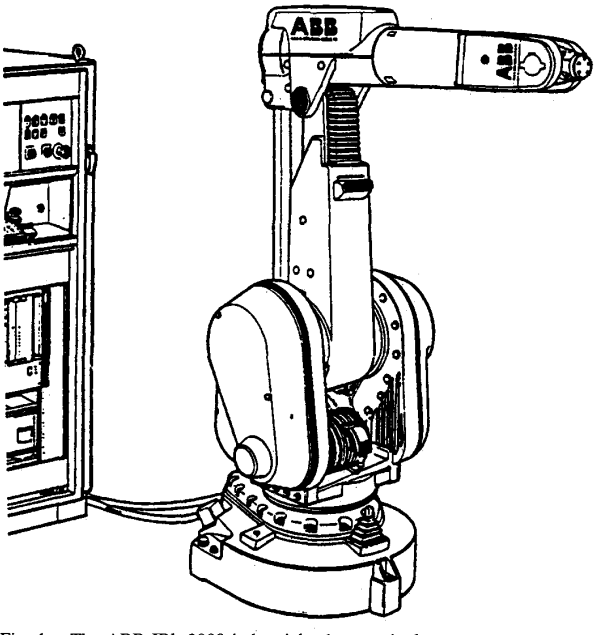


Fig. 1. The ABB IRb 2000 industrial robot manipulator.

Jacobian matrix is invoked to analyze the features of the method. A suitable criterion is used to compute varying damping and weighting factors. In this respect, to know good estimates of the smallest singular value, which gives a measure of closeness to singularity [15]–[16], is recognized to yield considerable performance enhancements. A recently proposed estimation algorithm of the two smallest singular values is utilized [17] which is originated from the numerical filtering method proposed in [18]. Further, the introduction of a feedback correction term to avoid numerical drift instabilities [19]–[21] is proposed and its effects on solution accuracy and feasibility are discussed.

In view of experimental testing, numerical implementation aspects of the solution are discussed in the paper in terms of flop requirements for the various inverse kinematics schemes.

A description of the laboratory set-up is provided. Experimental case studies are illustrated with the manipulator passing nearby either a single (wrist) or a double (wrist and shoulder) singularity. The influence of the weighting factor and feedback correction term on solution accuracy is extensively tested out, and the performance of the various solutions is compared.

The paper is organized as follows. Section II gives the kinematic description of the manipulator. The features of the inverse kinematics schemes based on the damped least-squares inverse are reviewed in Section III, while its implementation aspects are discussed in Section IV. Section V is devoted to present the experimental results obtained with numerical case studies. Conclusions are drawn in a final section.

II. Kinematics

The ABB IRb 2000 is a six-revolute-joint manipulator manufactured by ABB Robotics. The manipulator is shown in Fig. 1, while its Denavit-Hartenberg parameters (Craig's convention [22]) and mechanical joint limits are reported in Table I. The inner three joints are in the same arrangement as in an elbow manipulator, while the outer three joints constitute the spherical wrist commonly used in industrial robots.

TABLE I
ABB IRb 2000 Robot Manipulator: Denavit-Hartenberg Parameters; Joint Range and Velocity Limits

Link	ϑ [rad]	d [m]	a [m]	α [rad]
1	$\pi/2$	0	0	0
2	$\pi/2$	0	0	$\pi/2$
3	$\pi/2$	0	0.710	0
4	0	0.850	0.125	$\pi/2$
5	0	0	0	$\pi/2$
6	0	0.100	0	$\pi/2$

Joint	Working range [rad]	Max speed [rad/s]
1	$-0.99 \div +0.99$	2.01
2	$-0.85 \div +0.85$	2.01
3	$-2.72 \div -0.49$	2.01
4	$-3.43 \div +3.43$	4.89
5	$-2.00 \div +2.00$	5.24
6	$-3.14 \div +3.14$	5.24

Let $\mathbf{q}$ denote the (6×1) joint vector. The (6×6) Jacobian matrix $\mathbf{J}$ relates the joint velocity vector $\dot{\mathbf{q}}$ to the (6×1) end-effector velocity vector ν through the mapping

$$\nu = \begin{pmatrix} \dot{\mathbf{p}} \\ \omega \end{pmatrix} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{1}$$

where $\dot{\mathbf{p}}$ and ω represent end-effector linear and angular velocities, respectively.

In configurations where $\mathbf{J}$ has full rank, any end-effector velocity can be attained. When $\mathbf{J}$ is rank deficient, i.e. $\text{rank}(\mathbf{J}) = r$, $r < 6$, constraints on the feasible end-effector velocity occur and the manipulator is said to be at a singular configuration or at a *singularity*.

The ABB IRb 2000 manipulator has a simple kinematic structure and its singularities are well-understood. We have:

- If $a_3c_3 + d_4s_3 = 0$,[1] the elbow is stretched out and the manipulator is in the so-called *elbow* singularity; this does *not* correspond to a reachable configuration of the manipulator, due to the mechanical joint range for q_3, and then is of no interest.
- If the wrist point lies on the axis of joint 1, its position cannot be changed by a rotation of q_1 and the manipulator is in the so-called *shoulder* singularity (Fig. 2(a)); from simple geometrical relationship, it can be found that the shoulder singularity occurs when

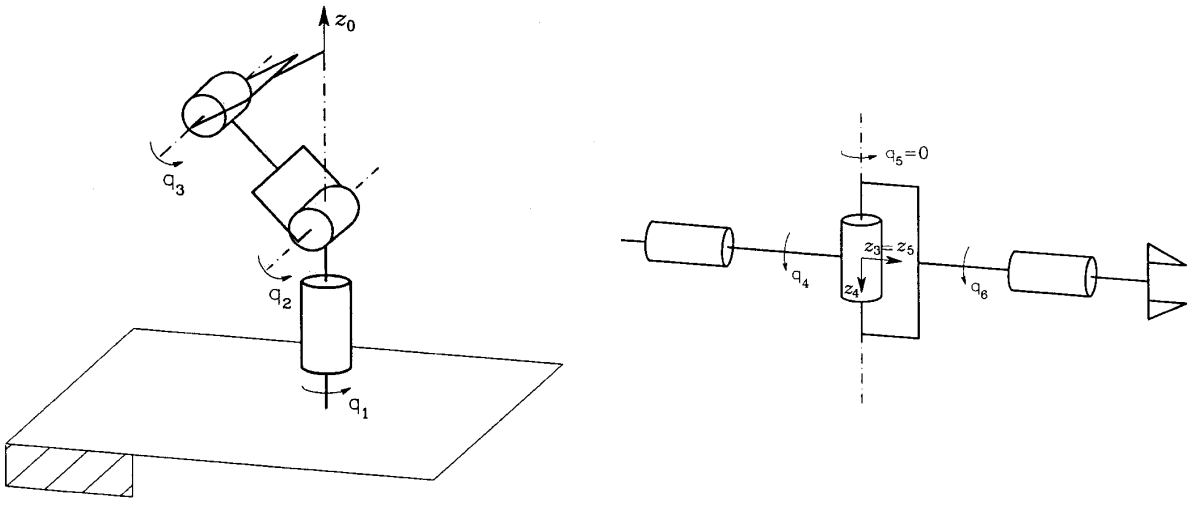
$$a_2s_2 + a_3c_{23} + d_4s_{23} = 0. \tag{2}$$

- If

$$q_5 = 0, \tag{3}$$

the two roll axes of the wrist are aligned and the manipulator is in the so-called *wrist* singularity (Fig. 2(b)).

[1] Conventional abbreviations have been used for sine and cosine of pertinent angles.



(a) (b)

Fig. 2. (a) Shoulder singularity. (b) Wrist singularity.

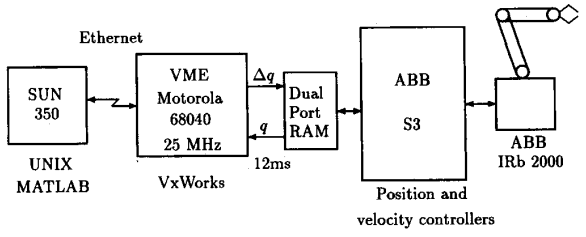


Fig. 3. Block diagram of laboratory set-up.

III. Damped Least-Squares Inverse Kinematics

It is well-known that the control system of a robotic manipulator operates in the joint space generating the driving torques at the joint actuators. The reference trajectory for the joint control servos is to be generated via kinematic inversion of the given end-effector trajectory; when the arm is at—or close to—a singularity, large joint velocities may occur or degenerate directions may exist where end-effector velocity is not feasible. Therefore, the control system of a robotic manipulator should be provided with the capability of handling singularities. The following two subsections review the inverse kinematics solution based on the damped least-squares method with estimate of the smallest singular value of the Jacobian matrix. Then, the user-defined accuracy strategy is adopted in the framework of a weighted damped least-squares solution and a new criterion for choosing the weighting matrix is proposed for handling the wrist singularity of manipulators with a spherical wrist. Finally, a feedback correction technique is proposed with suitable shaping of the feedback gain matrix.

A. Basic Damped Least-Squares Scheme

An effective strategy that allows motion control of manipulators in the neighborhood of kinematic singularities is the *damped least-squares* technique originally proposed in [7]–[8]. The method corresponds to solving the equation

$$\mathbf{J}^{\mathrm{T}}(\mathbf{q})\nu = \left(\mathbf{J}^{\mathrm{T}}(\mathbf{q})\mathbf{J}(\mathbf{q}) + \lambda^2\mathbf{I}\right)\dot{\mathbf{q}} \tag{4}$$

in lieu of (1); in (4), $\lambda \geq 0$ is the damping factor and $\mathbf{I}$ is the (6×6) identity matrix. It can be easily shown that the solution to (4) can be formally written as

$$\dot{\mathbf{q}} = \left(\mathbf{J}^{\mathrm{T}}(\mathbf{q})\mathbf{J}(\mathbf{q}) + \lambda^2\mathbf{I}\right)^{-1}\mathbf{J}^{\mathrm{T}}(\mathbf{q})\nu. \tag{5}$$

Notice that, when $\lambda = 0$, (1) becomes identical to (4) and the damped least-squares solution reduces to a regular matrix inversion which is ill-conditioned close to a singularity.

It is important to point out that solutions to (4) satisfy the condition

$$\min_{\dot{\mathbf{q}}} \left(\|\nu - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}\|^2 + \lambda^2\|\dot{\mathbf{q}}\|^2\right) \tag{6}$$

which evidences the possibility of trading off accuracy against feasibility of the joint velocity required to generate the given end-effector velocity. Therefore, it is essential to select suitable values for the damping factor: Small values of λ give accurate solutions but low robustness to the occurrence of singular and near-singular configurations. Large values of λ result in low tracking accuracy even when a feasible and accurate solution would be possible.

To gain more insight into the features of solution (5), the singular value decomposition [14] of the Jacobian matrix is

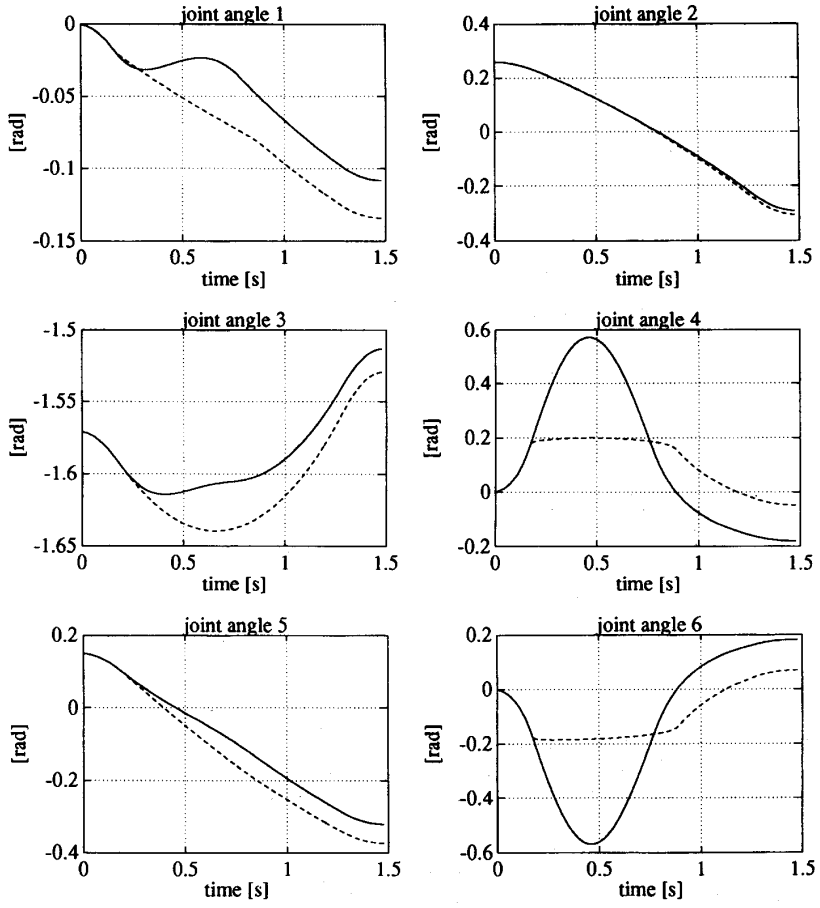


Fig. 4. Joint angles for Trajectory 1 without feedback correction; damped least-squares scheme (solid), weighted damped least-squares scheme (dashed).

helpful, that is

$$\mathbf{J} = \sum_{i=1}^{6} \sigma_i \mathbf{u}_i \mathbf{v}_i^{\mathrm{T}} \tag{7}$$

where $\mathbf{v}_i$ ($\mathbf{u}_i$) are the input (output) singular vectors, and σ_i are the singular values ordered so that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, with r being the rank of $\mathbf{J}$.

On the basis of (7), the damped least-squares solution (5) can be rewritten as

$$\dot{\mathbf{q}} = \sum_{i=1}^{6} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v}_i \mathbf{u}_i^{\mathrm{T}} \nu. \tag{8}$$

It is clear that the components for which $\sigma_i \gg \lambda$ are little influenced by the damping factor since

$$\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \approx \frac{1}{\sigma_i}. \tag{9}$$

On the other hand, when a singularity is approached, the smallest singular value tends to zero while the associated component of the solution is driven to zero by the factor σ_i/λ^2; this progressively reduces the joint velocity associated with the near-degenerate components of the commanded velocity ν .

The damping factor λ determines the degree of approximation introduced with respect to the pure least-squares solution; then, using a constant value for λ may turn out to be inadequate for obtaining good performance over the entire manipulator workspace. An effective choice is to adjust λ as a function of some measure of closeness to the singularity at the current configuration of the manipulator; to this purpose manipulability measures [7], [23] or estimates of the smallest singular value can be adopted. Remarkably, currently available microprocessors even allow real-time computation of full singular value decomposition [24]–[25].

A singular region can be defined on the basis of the estimate of the smallest singular value of $\mathbf{J}$; outside the region the exact solution is used, while inside the region a configuration-varying damping factor is introduced to obtain the desired approximate solution. The factor must be chosen so that continuity of joint velocity $\dot{\mathbf{q}}$ is ensured in the transition at the border of the singular region. We have selected the damping factor according to the following law [13]:

$$\lambda^2 = \begin{cases} 0 & \text{when } \hat{\sigma}_6 \geq \varepsilon \\ \left(1 - \left(\frac{\hat{\sigma}_6}{\varepsilon}\right)^2\right) \lambda_{\max}^2 & \text{otherwise,} \end{cases} \tag{10}$$

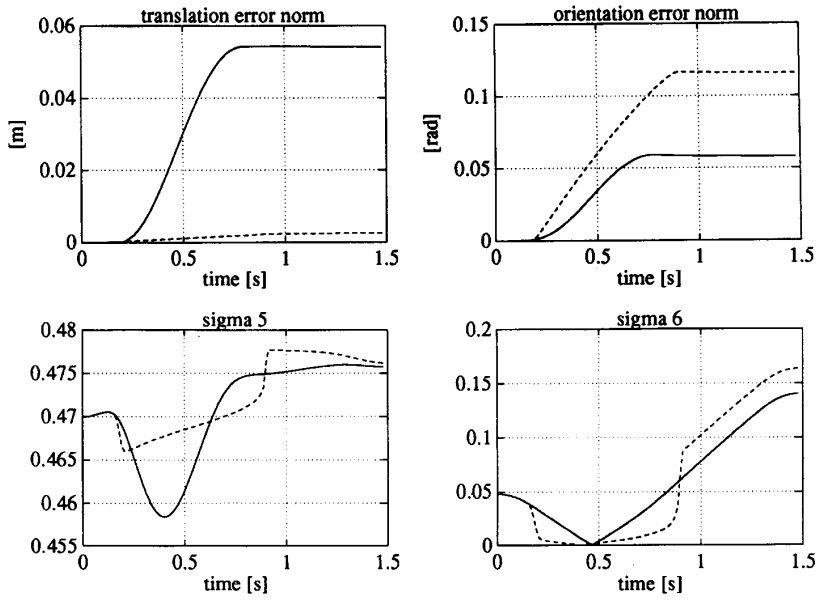


Fig. 5. Translation and orientation error norms and estimates of the two smallest singular values for Trajectory 1 without feedback correction; damped least-squares scheme (solid), weighted damped least-squares scheme (dashed).

where $\hat{\sigma}_6$ is the estimate of the smallest singular value, and ε defines the size of the singular region; the value of $\lambda_{\max}$ is at user's disposal to suitably shape the solution in the neighborhood of a singularity.

Equation (10) requires computation of the smallest singular value. In order to avoid a full singular value decomposition, we resort to recursive numerical techniques to find an estimate of the smallest singular value to be used in (10).

B. Estimate of the Smallest Singular Value

An estimate of the smallest singular value can be computed according to the recursive algorithm in [18]. The algorithm is described below with a slight modification as the last input singular vector $\mathbf{v}_6$ is estimated instead of the output singular vector $\mathbf{u}_6$.

Suppose that an estimate $\hat{\mathbf{v}}_6$ of the last input singular vector is available, so that $\hat{\mathbf{v}}_6 \approx \mathbf{v}_6$ and $\|\hat{\mathbf{v}}_6\| = 1$. This estimate is used to compute the vector $\hat{\mathbf{v}}_6'$ from

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I})\hat{\mathbf{v}}_6' = \hat{\mathbf{v}}_6. \tag{11}$$

Then the square of the estimate $\hat{\sigma}_6$ of the smallest singular value can be found as

$$\hat{\sigma}_6^2 = \frac{1}{\|\hat{\mathbf{v}}_6'\|} - \lambda^2 \tag{12}$$

while the estimate of $\mathbf{v}_6$ is updated using

$$\hat{\mathbf{v}}_6 = \frac{\hat{\mathbf{v}}_6'}{\|\hat{\mathbf{v}}_6'\|}. \tag{13}$$

The above estimation scheme is based on the assumption that $\mathbf{v}_6$ is slowly rotating, which is normally the case. However, if the manipulator is close to both the wrist and shoulder singularities, the vector $\mathbf{v}_6$ will instantaneously rotate 90° if the two smallest singular values cross. The estimate of the smallest singular value will then track σ_5 initially, before $\hat{\mathbf{v}}_6$ again converges to $\mathbf{v}_6$. This problem was recognized in [18]. As previously noted, the ABB IRb 2000 does not have an elbow singularity due to joint limits, so at most two of the singular values can become zero.

An extension of the scheme in [18] has recently been proposed in [17] by estimating not only the smallest but also the second smallest singular value. Assume that the estimates $\hat{\mathbf{v}}_6$ and $\hat{\sigma}_6$ are available and define the matrix

$$\mathbf{H} = \mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I} - (\hat{\sigma}_6^2 + \lambda^2)\hat{\mathbf{v}}_6\hat{\mathbf{v}}_6^{\mathrm{T}}. \tag{14}$$

With this choice, the second smallest singular value of $\mathbf{J}$ plays in

$$\mathbf{H}\hat{\mathbf{v}}_5' = \hat{\mathbf{v}}_5 \tag{15}$$

the same role as σ_6 in (11) and then will provide a convergent estimate of $\hat{\mathbf{v}}_5$ to $\mathbf{v}_5$ and $\hat{\sigma}_5$ to σ_5 .

At this point, suppose that $\hat{\mathbf{v}}_5$ is an estimate of $\mathbf{v}_5$ so that $\hat{\mathbf{v}}_5 \approx \mathbf{v}_5$ and $\|\hat{\mathbf{v}}_5\| = 1$. This estimate is used to compute $\hat{\mathbf{v}}_5'$ from (15). Then, an estimate of the square of the second smallest singular value of $\mathbf{J}$ is found from

$$\hat{\sigma}_5^2 = \frac{1}{\|\hat{\mathbf{v}}_5'\|} - \lambda^2 \tag{16}$$

and the estimate of $\mathbf{v}_5$ is updated using

$$\hat{\mathbf{v}}_5 = \frac{\hat{\mathbf{v}}_5'}{\|\hat{\mathbf{v}}_5'\|}. \tag{17}$$

On the basis of this modified estimation algorithm, crossing of singularities can be effectively detected; also, by switching the two singular values and the associated estimates $\hat{\mathbf{v}}_5$ and $\hat{\mathbf{v}}_6$, the estimation of the smallest singular value will be accurate even when the two smallest singular values cross [17].

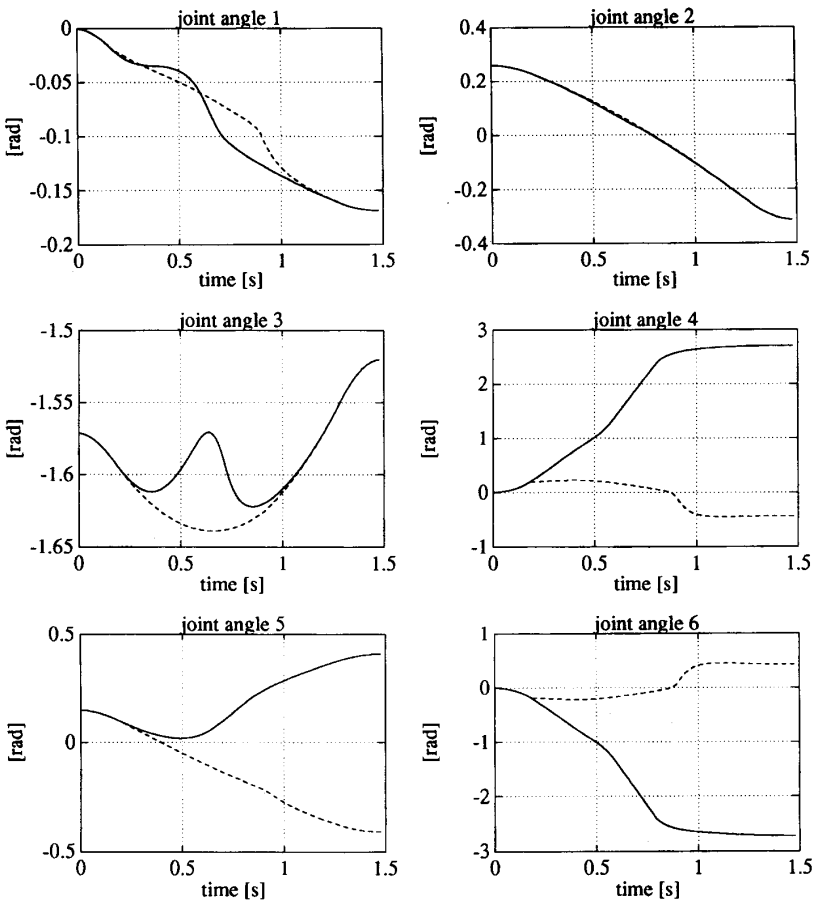


Fig. 6. Joint angles for Trajectory 1 with feedback correction; damped least-squares scheme (solid), weighted damped least-squares scheme (dashed).

C. User-Defined Accuracy

The above damped least-squares method achieves a compromise between accuracy and robustness of the solution. This is performed without specific regard to the components of the particular task assigned to the manipulator's end-effector. The *user-defined accuracy* strategy introduced in [13] based on the weighted damped least-squares method in [7] allows to discriminate between directions in the end-effector space where higher accuracy is desired and directions where lower accuracy can be tolerated. This is the case, for instance, of spot welding or spray painting in which the tool angle around the approach direction is not essential to the fulfillment of the task.

Let a weighted end-effector velocity vector be defined as

$$\tilde{\nu} = \mathbf{W}\nu \tag{18}$$

where $\mathbf{W}$ is the (6×6) task-dependent weighting matrix taking into account the anisotropy of the task requirements. Substituting (18) into (1) gives

$$\tilde{\nu} = \tilde{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} \tag{19}$$

where $\tilde{\mathbf{J}} = \mathbf{W}\mathbf{J}$. It is worth noticing that if $\mathbf{W}$ is full-rank, solving (1) is equivalent to solving (19), but with different conditioning of the system of equations to solve. This suggests to select only the strictly necessary weighting action in order to avoid undesired ill-conditioning of $\tilde{\mathbf{J}}$.

Equation (19) can be solved using the weighted damped least-squares technique [7], that is

$$\tilde{\mathbf{J}}^{\mathrm{T}}(\mathbf{q})\tilde{\nu} = \left(\tilde{\mathbf{J}}^{\mathrm{T}}(\mathbf{q})\tilde{\mathbf{J}}(\mathbf{q}) + \lambda^2\mathbf{I}\right)\dot{\mathbf{q}}. \tag{20}$$

Again, the singular value decomposition of the matrix $\tilde{\mathbf{J}}$ is helpful, i.e.

$$\tilde{\mathbf{J}} = \sum_{i=1}^{6} \tilde{\sigma}_i \tilde{\mathbf{u}}_i \tilde{\mathbf{v}}_i^{\mathrm{T}} \tag{21}$$

and the solution to (20) can be written as

$$\dot{\mathbf{q}} = \sum_{i=1}^{6} \frac{\tilde{\sigma}_i}{\tilde{\sigma}_i^2 + \lambda^2} \tilde{\mathbf{v}}_i \tilde{\mathbf{u}}_i^{\mathrm{T}} \tilde{\nu}. \tag{22}$$

It is clear that the singular values $\tilde{\sigma}_i$ and the singular vectors $\tilde{\mathbf{u}}_i$ and $\tilde{\mathbf{v}}_i$ depend on the choice of the weighting matrix $\mathbf{W}$. While this has no effect on the solution $\dot{\mathbf{q}}$ as long as $\tilde{\sigma}_6 \gg \lambda$, close to singularities where $\tilde{\sigma}_r \ll \lambda$, for some $r < 6$, the solution can be shaped by properly selecting the matrix $\mathbf{W}$.

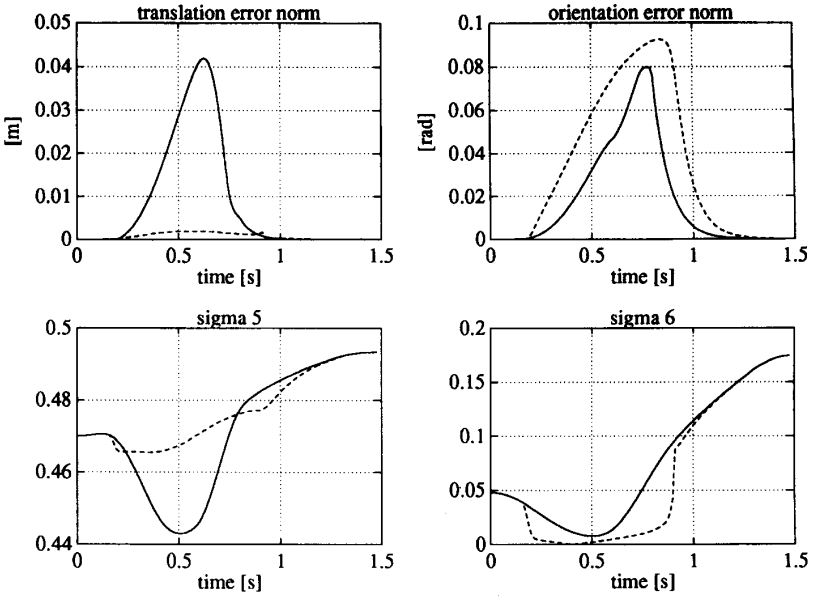


Fig. 7. Translation and orientation error norms and estimates of the two smallest singular values for Trajectory 1 with feedback correction; damped least-squares scheme (solid), weighted damped least-squares scheme (dashed).

For the typical elbow geometry with spherical wrist, it is worthwhile to devise a special handling of the wrist singularity which is difficult to predict at the planning level in the end-effector space. It can be recognized that, at the wrist singularity, there are only two components of the angular velocity vector that can be generated by the wrist itself. The remaining component might be generated by the inner joints, at the expense of loss of accuracy along some other end-effector space directions though. For this reason, lower weight should be put on the angular velocity component that is infeasible to the wrist. For the ABB IRb 2000, this is easily expressed in the frame attached to link 4; let $\mathbf{R}_4$ denote the rotation matrix describing orientation of this frame with respect to the base frame, so that the infeasible component is aligned with the x-axis. We propose then to choose the weighting matrix as

$$\mathbf{W} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_4 \mathrm{diag}\{w, 1, 1\} \mathbf{R}_4^{\mathrm{T}} \end{pmatrix}. \tag{23}$$

Similarly to the choice of the damping factor as in (10), the weighting factor w is selected according to the following expression:

$$(1-w)^2 = \begin{cases} 0 & \text{when } \hat{\sigma}_6 \geq \varepsilon \\ \left(1 - \left(\frac{\hat{\sigma}_6}{\varepsilon}\right)^2\right)(1 - w_{\min})^2 & \text{otherwise} \end{cases} \tag{24}$$

where $w_{\min} \geq 0$ is a design parameter [13] and [26].

D. Feedback Correction

The above inverse kinematics solutions are expected to suffer from typical numerical drift, when implemented in discrete time. In order to avoid this drawback, a feedback correction term [19]–[21] can be keenly introduced by replacing the end-effector velocity ν by

$$\nu_d + \mathbf{K}\mathbf{e} \tag{25}$$

where the subscript "d" denotes the desired reference end-effector velocity, $\mathbf{K}$ is a positive definite—usually diagonal—(6 × 6) matrix, and $\mathbf{e}$ expresses the error between the desired and actual end-effector location. The error $\mathbf{e}$ can be represented by [27]

$$\mathbf{e} = \begin{pmatrix} \mathbf{e}_t \\ \mathbf{e}_o \end{pmatrix} = \begin{pmatrix} \mathbf{p}_d - \mathbf{p} \\ \frac{1}{2}(\mathbf{n} \times \mathbf{n}_d + \mathbf{s} \times \mathbf{s}_d + \mathbf{a} \times \mathbf{a}_d) \end{pmatrix} \tag{26}$$

where the translation error is given by the (3×1) vector $\mathbf{e}_t$ and the orientation error is given by the (3×1) vector $\mathbf{e}_o$. The end-effector position is expressed by the (3×1) position vector $\mathbf{p}$ while its orientation by the (3×3) rotation matrix $\mathbf{R} = (\mathbf{n}\ \mathbf{s}\ \mathbf{a})$, with $\mathbf{n}$, $\mathbf{s}$, $\mathbf{a}$ being the unit vectors of the end-effector frame.

It is important to notice that, in the neighborhood of a singularity, end-effector errors typically increase along the near-degenerate components of the given end-effector velocity and convergence is slowed down [21]. Therefore, we propose to shape the action of the feedback correction term around the singularities using $\mathbf{K} = \varrho \mathbf{K}_0$, where $\mathbf{K}_0$ is a constant matrix and ϱ is a varying factor to be properly adjusted.

We have found that it is important to have $\varrho = 0$ inside the singular region defined by $\sigma_6 \leq \varepsilon$. Indeed, if a velocity is assigned along a near-degenerate direction and a nonzero gain is used for the feedback correction term, the error $\mathbf{e}$ will eventually approach zero; however, the resulting joint velocities may cause the manipulator to reach the joint limits. Outside the singular region, interpolation is used to achieve a smooth solution and full value to the gains ($\varrho = 1$) is set when far enough from the singularity. In our experience, interpolation had to be performed when $\varepsilon < \sigma_6 < 4\varepsilon$ using a quadratic type function, i.e.

$$\varrho = \begin{cases} 0 & \text{when } \sigma_6 \leq \varepsilon \\ \frac{(\sigma_6 - \varepsilon)^2}{(3\varepsilon)^2} & \text{when } \varepsilon < \sigma_6 < 4\varepsilon \\ 1 & \text{otherwise.} \end{cases} \tag{27}$$

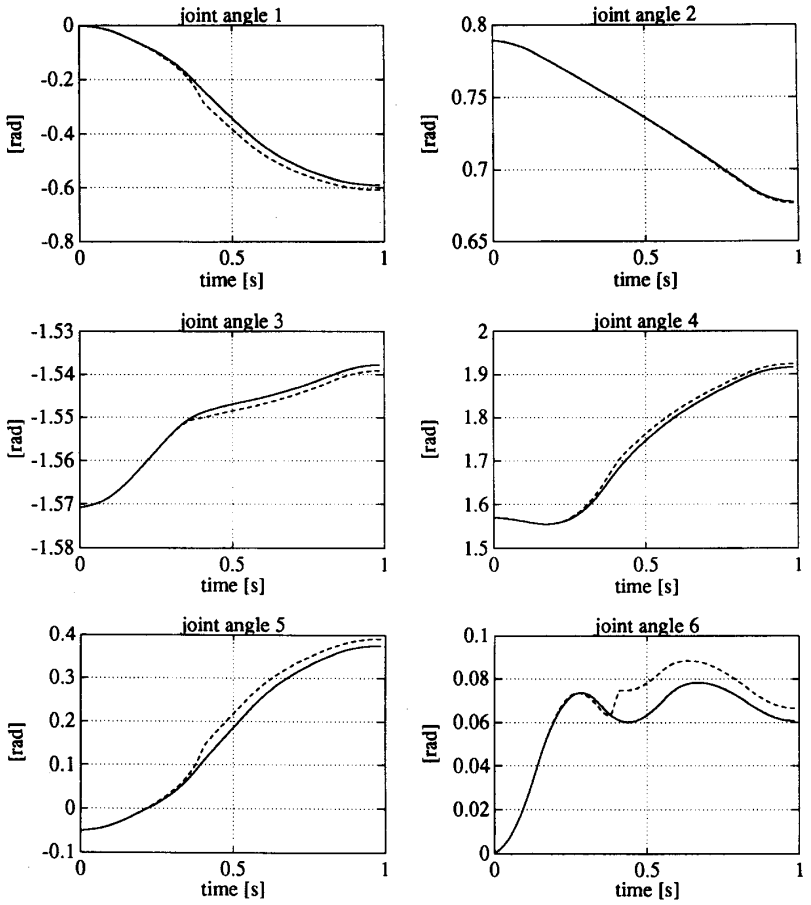


Fig. 8. Joint angles for Trajectory 2; case a) (solid), case b) (dashed).

IV. Numerical Implementation Aspects

The algorithmic computational load of the above inverse kinematics schemes plays a significant role in view of real-time implementation on the industrial manipulator. Therefore, the following analysis is devoted to the numerical aspects of the solution.

The direct kinematics was computed with an algorithm based on the standard technique proposed in [1]. Care was taken to avoid unnecessary computation, like products by zeros or ones, or matrix elements that are not required to achieve the final result. The overall computation of position and orientation of the end-effector and Jacobian matrix was thus reduced to 127 flops. The number of flops is the total number of floating point operations as defined in [14] and can be found e.g. in MATLAB using the flop function.

Equation (4) was solved by computing

$$\mathbf{y} = \mathbf{J}^{\mathrm{T}}\nu \tag{28}$$

and solving

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I})\dot{\mathbf{q}} = \mathbf{y} \tag{29}$$

for $\dot{\mathbf{q}}$, using the Cholesky decomposition

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I}) = \mathbf{G}^{\mathrm{T}}\mathbf{G} \tag{30}$$

where $\mathbf{G}$ is an upper triangular matrix of proper dimensions. The Cholesky decomposition was computed with a C function based on algorithm 4.2.2 in [14]; the number of flops was 91. The linear system (29), decomposed as in (30), was solved using algorithms 3.1.1 and 3.1.2 in [14], requiring 36 flops each.

Furthermore, the matrix $\mathbf{J}^{\mathrm{T}}\mathbf{J}$ is symmetric, and accordingly only the upper triangular part of the matrix needs to be computed. The computation of $\mathbf{J}^{\mathrm{T}}\nu$ in (28) and $\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I}$ in (30) required 303 flops. Then, the total amount of computation involved to solve (4) for a given $\mathbf{J}$ was 466 flops.

As for the estimate of the smallest singular value, solution of (11) using the available Cholesky factor $\mathbf{G}$ given in (30) required additional 72 flops. Further, the computation of the norm $\|\hat{\mathbf{v}}_6'\|$ used in (12)–(13) involved 12 flops, while the computation of $\hat{\sigma}_6^2$ in (12) required 2 flops. In total, this gave 92 flops.

Noticing that

$$\mathbf{H}^{-1} = (\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I})^{-1} - \frac{1}{(\sigma_6^2 + \lambda^2)}\mathbf{v}_6\mathbf{v}_6^{\mathrm{T}} \tag{31}$$

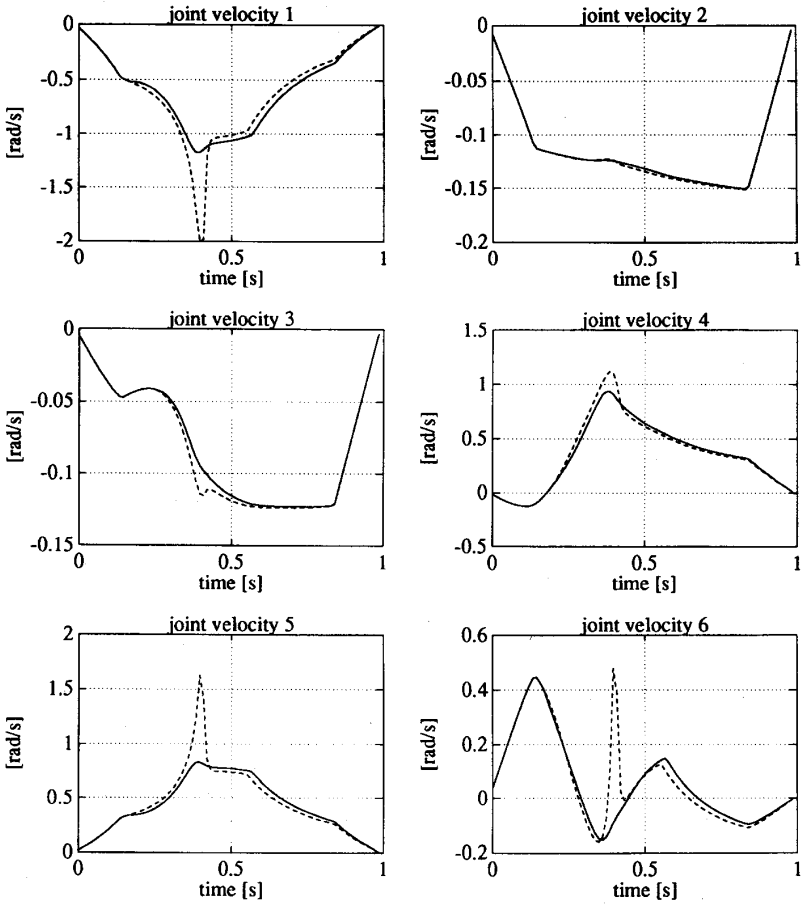


Fig. 9. Joint velocities for Trajectory 2; case (**a**)) (solid), case (**b**)) (dashed).

computation of $\hat{\mathbf{v}}_5'$ from (15) can be achieved again with the Cholesky factor $\mathbf{G}$ available from (30). This was done by first solving

$$(\mathbf{J}^{\mathrm{T}}\mathbf{J} + \lambda^2\mathbf{I})\mathbf{z} = \hat{\mathbf{v}}_5 \tag{32}$$

for $\mathbf{z}$, and then computing

$$\hat{\mathbf{v}}_5' = \mathbf{z} - \hat{\mathbf{v}}_6'(\hat{\mathbf{v}}_6^{\mathrm{T}}\hat{\mathbf{v}}_5). \tag{33}$$

Finally, (16)–(17) were computed. The total computational load for the estimate of the second smallest singular value was thus 115 flops.

V. Experimental Results

A. Laboratory Set-Up

The experiments were run on an ABB IRb 2000 robot manipulator. The original joint servos of the S3 industrial control system were used, and an interface was established at the joint increment level. This allowed implementation of a two-stage control strategy, that is an inverse kinematics module based on the foregoing damped least-squares solution providing the reference inputs to the manipulator joint servos. This was done in cooperation with ABB Robotics who slightly modified the S3 control system by adding an interface board with a dual-port RAM, as well as software modules to establish the required communication protocol. The resulting communication facilitated the transfer of joint variables and increments at a sampling time of 12 (ms), and in addition it allowed for remote initialization and activation of the ABB S3 system.

The inverse kinematics were computed on a 25 (MHz) Motorola 68040 VME board which communicated with the ABB S3 control system through the dual-port RAM. The software was developed on a SUN 350 workstation, and the executable code was downloaded to the 68040 board and executed using VxWorks. A block diagram showing the interconnection between major components of the laboratory set-up is sketched in Fig. 3.

The program system which was executed on the Motorola 68040 was written in C, and consisted of two activities; an interactive activity containing initialization and user interface, and the real-time controller activity. The real-time controller activity had the following steps:

- input joint angles,
- calculate joint increments,
- output joint increments,
- store data for logging.

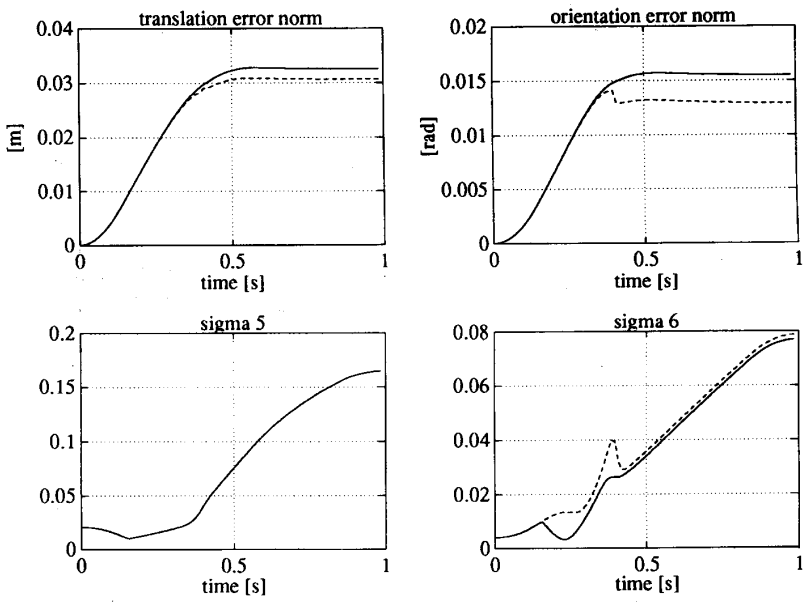


Fig. 10. Translation and orientation error norms and estimates of the two smallest singular values for Trajectory 2; case (**a**)) (solid), case (**b**)) (dashed).

The interactive user interface contained initialization of the S3 control system and of the communication between the ABB controller and the real-time activity. Further it contained initialization of kinematic parameters and a menu-driven system for adjusting kinematic parameters, specifying trajectory data, and selecting the algorithm for the inverse kinematics solution. In addition, a function for transferring logged data to the SUN system was included. The logged data was subsequently written to a file in MATLAB format which allowed for postprocessing and plotting in the SUN UNIX environment.

A simple solution to the problem of communication between the real-time and interactive activities was achieved by including both activities in the same C program. The interactive activity was the main program, while the real-time activity was called from a function which was started by a timer interrupt every 12 (ms). The data transfer between the two activities was performed through global data structures containing parameters for the inverse kinematics algorithms, trajectory data, logged data and logical variables for algorithm selection.

B. Case Studies

In the following we present two case studies to demonstrate the application of the above schemes to real-time kinematic control of an industrial manipulator. It is worth mentioning that extensive simulation of the system has been performed to foresee the behavior of the manipulator under the proposed control. These results, however, are not reported here for brevity, since they closely match those obtained in the experiments.

Trajectory 1: A reference trajectory through the wrist singularity was studied. The initial configuration was

$$\mathbf{q} = (0 \quad \pi/12 \quad -\pi/2 \quad 0 \quad 0.15 \quad 0)^{\mathrm{T}} \quad \text{rad}.$$

An increment $\Delta\mathbf{p} = (0.18 \; 0.45 \; -0.45)^{\mathrm{T}}$ [m] was interpolated in end-effector coordinates using linear segments with parabolic blends [28]. The blend time was 0.2 s, and the total time of the trajectory was 1.5 s. The resulting cruise velocity between 0.2 s and 1.3 s was approximately 0.5 [m/s]. The wrist singularity was encountered after approximately 0.6 s.

The damping factor was computed from (10) with $\varepsilon = 0.04$ and $\lambda_{\max} = 0.04$. The estimate $\tilde{\sigma}_6$ of the smallest singular value was computed using (11)–(17). The initial estimates of the singular values and the singular vectors were found using the singular value decomposition in MATLAB.

The basic damped least-squares scheme (4) was used first; then, for comparison, the same trajectory was tested with the weighted damped least-squares scheme based on (20). The weighting matrix was chosen as in (23)–(24) with $w_{\min} = 0.1$. The results are shown in Figs. 4 and 5 for both the damped least-squares and weighted damped least-squares schemes, without feedback correction term. In both cases the joint velocities were feasible with peak values of approximately 2 rad/s. Without weighting, the norm of the translation error at final time t_f was $\|\mathbf{e}_t(t_f)\| = 0.055$ [m], while the orientation error norm was $\|\mathbf{e}_o(t_f)\| = 0.06$ rad. With weighting, the errors were $\|\mathbf{e}_t(t_f)\| = 0.0025$ [m] and $\|\mathbf{e}_o(t_f)\| = 0.12$ rad.

This result clearly demonstrates the advantage of using weighting since our main concern was to achieve accuracy in translation. In fact, weighting resulted in a reduction of the translation error by a factor of approximately 20, while the orientation error was increased only by a factor of two. The effect of weighting on the smallest singular values is seen from Fig. 5. The solution is shaped so that the translational components of $\tilde{\mathbf{u}}_6$ are small; this is achieved at the expense of a significant reduction of $\tilde{\sigma}_6$ around the singularity compared to the corresponding σ_6 in the non-weighted case.

The same experiments were repeated with a feedback correction term according to (25) and (26) with $\mathbf{K} = \text{diag}\{12 \ldots 12\}$. The results are shown in Figs. 6 and 7. In this case the joint velocities were higher. In fact, with the damped least-squares solution, joint velocities 4 and 6 saturated between 0.6–0.8 s at 5 rad/s and -5 rad/s, respectively. With the introduction of weighting, the joint velocities were feasible with peak values less than 5 rad/s. Thanks to the feedback correction term, the end-effector error $\mathbf{e}$ converged to zero after leaving the singular region (see Fig. 7). This resulted in a reorientation of joints 4 and 6 by $\pm\pi$ in the non-weighted case, which reflects the fact that large increments in joint angles may result from small end-effector increments when the manipulator is close to a singularity. Remarkably, Fig. 7 reveals also that outside the singular region $\tilde{\sigma}_6$ tends to σ_6; obviously, this was not the case in the previous experiments without feedback correction (see Fig. 5).

Trajectory 2: A trajectory involving both the shoulder and wrist singularities was studied. The initial configuration was

$$\mathbf{q} = (0 \quad 0.7893 \quad -\pi/2 \quad \pi/2 \quad -0.05 \quad 0)^{\mathrm{T}} \quad \text{rad.}$$

An increment $\Delta\mathbf{p} = (0.1 \ 0.1 \ 0)^{\mathrm{T}}$ [m] was interpolated in end-effector coordinates using linear segments with parabolic blends. The blend time was 0.15 s, and the total time of the trajectory was 1.0 s. The resulting cruise velocity between 0.15 s and 0.85 s was approximately 0.166 [m/s]. The wrist singularity was encountered after approximately 0.22 s.

Also for this trajectory, the damping factor was computed from (10) with $\varepsilon = 0.04$ and $\lambda_{\max} = 0.04$. The basic damped least-squares solution (4) was used. The estimate $\hat{\sigma}_6$ of the smallest singular value was found: **1)** by computing both $\hat{\sigma}_5$ and $\hat{\sigma}_6$ from (11)–(17), and **2)** by computing $\hat{\sigma}_6$ from (11)–(13). As above, the initial estimates of the singular values and the singular vectors were found using the singular value decomposition in MATLAB.

The results without feedback correction are shown in Fig. 8–10. The damped least-squares scheme performs well also in this case. The norm of the translation error at final time t_f was $\|\mathbf{e}_t(t_f)\| = 0.03$[m], while the norm of the orientation error was $\|\mathbf{e}_o(t_f)\| = 0.015$ rad.

In case **a)** the crossing of the two smallest singular values associated with the wrist and shoulder singularity was successfully detected at 0.15 s and 0.37 s, and an accurate estimate $\hat{\sigma}_6$ of the smallest singular value was found. This gave satisfactory damping around the wrist singularity. The resulting joint velocities had peak values less than 1.2 rad/s. In case **b)** the crossing of the two smallest singular values caused the estimate $\hat{\sigma}_6$ to track σ_5, and the wrist singularity appearing at 0.22 s was not detected. This resulted into a low damping factor around the wrist singularity, and high joint velocities were experienced; in particular, the velocity of joint 1 saturated at -2 rad/s. Incidentally, the final errors were a little smaller in case **b)** since the incorrect estimate of σ_6 produced lower damping on the solution throughout the singular region.

VI. Conclusion

Damped least-squares schemes for kinematic control of robot manipulators have been successfully tested in a number of experimental case studies.

The basic damped least-squares scheme, offering efficient singularity handling capabilities, has been extended to include weighting which further provides user-defined accuracy capabilities. A feedback correction term to avoid numerical drifts of the discrete-time implementation of the inverse differential kinematics has been adopted, and its effects on the manipulator behavior with both the above schemes have been studied. Finally, the estimate of the second smallest singular value of the Jacobian matrix has been implemented and its improved potential over the scheme using the estimate of the sole smallest singular value has been demonstrated for a trajectory involving both the shoulder and wrist singularities.

Remarkably, all the above refinements of the basic damped least-squares solution have been implemented on industrially available hardware indicating that enhanced real-time kinematic control of robot manipulators through singularities is definitely possible.

References

[1] D. E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," *ASME J. Dyn. Syst., Meas. Control*, vol. 94, pp. 303–309, 1972.

[2] C. A. Klein and C. H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. on Syst., Man., Cyber.*, vol. 13, pp. 245–250, 1983.

[3] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE J. of Robot.Automat.*, vol. 4, pp. 43–53, 1988.

[4] R. H. Taylor, "Planning and execution of straight line manipulator trajectories," *IBM J. Res. and Dev.*, vol. 23, pp. 424–436, 1979.

[5] E. W. Aboaf and R. P. Paul, "Living with the singularity of robot wrists," *in Proc. 1987 IEEE Int. Conf. on Robot.and Automat.*, Raleigh, NC, pp. 1713–1717, Mar.–Apr. 1987.

[6] S. Chiaverini and O. Egeland, "A solution to the singularity problem for six-joint manipulators," *in Proc. 1990 IEEE Int. Conf. on Robot.Automat.*, Cincinnati, OH, pp. 644–649, May 1990.

[7] Y. Nakamura and H. Hanafusa, "Inverse kinematic solution with singularity robustness for robot manipulator control," *ASME J. Dyn. Syst., Meas., Control*, vol. 108, pp. 163–171, 1986.

[8] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. on Syst., Man, Cyber.*, vol. 16, pp. 93–101, 1986.

[9] P. Chiacchio and B. Siciliano, "Achieving singularity robustness: An inverse kinematic solution algorithm for robot control," *in IEE Int. Work. on Robot Control: Theory App.*, Oxford, GB, pp. 149–156, Apr. 1988.

[10] S. Chiaverini, "Inverse differential kinematics of robotic manipulators at singular and near-singular configurations," *in 1992 IEEE Int. Conf. on Robot.Automat.—Tutorial on 'Redundancy: Performance Indices, Singularities Avoidance and Algorithmic Implementations'*, Nice, France, pp. 2.1–2.9, May1992.

[11] O. Egeland, M. Ebdrup and S. Chiaverini, "Sensory control in singular configurations—Application to visual servoing," *in Proc. IEEE Int. Wkshp. on Intelligent Motion Control*, Istanbul, TR, pp. 401–405, Aug. 1990.

[12] S. Chiaverini, B. Siciliano and O. Egeland, "Robot control in singular configurations—Analysis and experimental results," *2nd Int. Symp. on Experimental Robotics*, Toulouse, France, June1991, in *Experimental Robotics II*, Lecture Notes in Control and Information Sciences 190, R. Chatila and G. Hirzinger (Eds.), Springer-Verlag, Berlin, Germany, pp. 25–34, 1993.

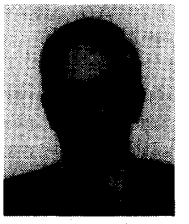
[13] S. Chiaverini, O. Egeland and R. K. Kanestrøm, "Achieving user-defined accuracy with damped least-squares inverse kinematics," *in Proc. 5th Int. Conf. on Adv. Robotics*, Pisa, I, pp. 672–677, June 1991.

[14] G. H. Golub and C. F. Van Loan, "*Matrix Computations*," 2nd ed., Baltimore, MD: Johns Hopkins University Press, 1989.

[15] C. A. Klein and B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int. J. Robot. Res.*, vol. 6, no. 2, pp. 72–83, 1987.
[16] I. Spangelo, J. R. Sagli and O. Egeland, "Bounds on the largest singular value of the manipulator Jacobian," *IEEE Trans. on Robot.Automat.*, vol. 9, pp. 93–96, 1993.
[17] S. Chiaverini, "Estimate of the two smallest singular values of the Jacobian matrix: Application to damped least-squares inverse kinematics," *J. Robotic Syst.*, vol. 10, no. 8, pp. 991–1008, 1993.
[18] A. A. Maciejewski and C. A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Robotic Syst.*, vol. 5, pp. 527–552, 1988.
[19] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Trans. on Syst., Man, Cyber.*, vol. 16, pp. 550–559, 1986.
[20] Y. T. Tsai and D. E. Orin, "A strictly convergent real-time solution for inverse kinematics of robot manipulators," *J. of Robotic Syst.*, vol. 4, pp. 477-501, 1987.
[21] C. W. Wampler and L. J. Leifer, "Applications of damped least-squares methods to resolved-rate and resolved-acceleration control of manipulators," *J. Dyn. Syst., Meas., and Control*, vol. 110, pp. 31–38, 1988.
[22] J. J. Craig, *Intro. to Robotics: Mechanics and Control*, 2nd Ed., Reading, MA: Addison-Wesley, 1989.
[23] T. Yoshikawa, "Translational and rotational manipulability of robotic manipulators," *in Proc. 1990 Amer. Control Conf.*, San Diego, CA, pp. 228–233, May 1990.
[24] A. A. Maciejewski and C. A. Klein, "The singular value decomposition: Computation and its applications to robotics," *Int. J. Robot.Res.*, vol. 8, no. 6, pp. 63–79, 1989.
[25] A. A. Maciejewski and J. M. Reagin, "A parallel algorithm and architecture for the control of kinematically redundant manipulators," *in Proc. 1992 IEEE Int. Conf. on Robot.and Automat.*, Nice, F, pp. 488–493, 1992.
[26] S. Chiaverini, O. Egeland and R. K. Spangelo, "Weighted damped least-squares in kinematic control of robotic manipulators," *Advanced Robotics*, vol. 7, pp. 201–218, 1993.
[27] J. Y. S. Luh, M. W. Walker and R. P. C. Paul, "Resolved-acceleration control of mechanical manipulators," *IEEE Trans. on Automat. Control*, vol. 25, pp. 468–474, 1980.
[28] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, New York, NY: John Wiley, 1989.

Stefano Chiaverini received the Laurea and the Research Doctorate degrees in electronics engineering at the University of Naples, Italy in 1986 and 1990, respectively.

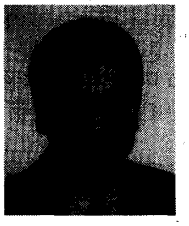
He became Research Associate in 1990 and Assistant Professor in 1993 at the Department of Computer and Systems Engineering of the University of Naples. From January to June 1989 he was a Visiting Scientist at the Robotics Laboratory of the German Aerospace Research Establishment (DLR) in Oberpfaffenhofen, Germany. His research interests include manipulator inverse kinematics techniques, redundant manipulator control, cooperative robot manipulation, and force/position control of manipulators.



Bruno Siciliano (M'91–SM'94) received the Laurea and the Research Doctorate degrees in electronics engineering at the University of Naples, Italy in 1982 and 1987, respectively.

He became Research Associate in 1989 and Associate Professor of Industrial Robotics in 1992 at the Department of Computer and Systems Engineering of the University of Naples. From the fall of 1985 to the spring of 1986 he was a Visiting Scientist at the School of Mechanical Engineering of the Georgia Institute of Technology, Atlanta, GA. His research interests include manipulator inverse kinematics techniques, redundant manipulator control, modeling and control of lightweight flexible arms, cooperative robot manipulation, and force/position control of manipulators.

Dr. Siciliano is an Associate Editor of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, Associate Editor of the *ASME Journal of Dynamic Systems, Measurement, and Control* and an editorial board member of *Robotica.*



Olav Egeland (S'85–M'86) received the Siv.Ing. degree in electrical engineering in 1984 and the Dr.Ing. degree in automatic control in 1987, both from the Norwegian Institute of Technology, Trondheim, Norway.

He became Assistant Professor in 1987 and Professor of Robotics in 1989 at the Division of Engineering Cybernetics of the Norwegian Institute of Technology. In the academic year 1988/89, he was a Visiting Scientist at the Robotics Lavboratory of the German Aerospace Research Establishment (DLR) in Oberpfaffenhofen, Germany. His research interests include kinematics, dynamics and control of manipulators, underwater vehicles, surface vessels, and flexible structures.