

Control of Robotic Systems through Singularities

Stefano Chiaverini Lorenzo Sciavicco Bruno Siciliano

Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli "Federico II"
via Claudio 21, 80125 Napoli, Italy

Abstract

The goal of this work is to provide an overview of major control techniques that manage the occurrence of singularities for robotic systems. The common feature of these methods is a modification of the inverse differential kinematic mapping which is ill-conditioned in the neighbourhood of a singularity. The following solutions are discussed; namely, the Jacobian transpose, the Jacobian pseudoinverse, and the damped least-squares Jacobian inverse.

Introduction

At a singular configuration the linear mapping which relates the joint-space velocity vector to the task-space velocity vector through the Jacobian matrix of a given manipulator becomes rank-deficient and the solution of the inverse kinematics problem is undefined; furthermore, very high values and discontinuities of the joint-space velocity result in the neighbourhood of the singularity due, respectively, to the unfeasible components of the assigned task-space velocity vector and to sudden crossings through the singular configuration itself.

When a preprogrammed reference task-space trajectory is to be tracked, it is normally possible to plan the trajectory so that singular configurations are avoided. Alternatively, it is possible to interpolate the joint-space solution close to the singularities. On the other hand, sensory control applications —where the reference trajectory is not known in advance— demand for singularity-robust inverse kinematics algorithms to guarantee the effectiveness of the robot control system all over the manipulator's workspace.

In order to overcome singularities, the basic numerical solution to the inverse kinematics problem which can be obtained by simply inverting the direct kinematic mapping is replaced by suitably-defined mappings relating the task-space to the joint-space; these

are designed to ensure well-behaved joint-space solutions close to a singularity, while guaranteeing accurate trajectory tracking far from singularities.

The present work is aimed at surveying the most effective inverse differential kinematics techniques which allow control of robotic systems through singularities. We will focus our attention only to the singularity handling issues, while eventual kinematic redundancy will not be exploited.

A very simple inverse kinematics solution can be obtained by using the transpose of the Jacobian matrix to relate the joint-space velocity vector to the task-space location of the manipulator [1]; this corresponds to adopting an impedance control law for an ideal manipulator of simplified dynamics [2]. The approach leads to a closed-loop formulation which offers an iterative solution to the inverse kinematics problem; the solution is exact for a given constant task-space location while is approximate in the trajectory-tracking case. The solution is computationally inexpensive and performs very robust behaviour close and through singular configurations; as a drawback, optimal tuning of the algorithm is required [3] and degraded accuracy is experienced when tracking fast task-space trajectories.

A more accurate solution can be obtained by using a pseudoinverse of the Jacobian matrix to transform desired task-space trajectory into the corresponding joint-space motion [4]. This solution is defined even at singular configurations, but high joint velocities may result in the neighbourhood of singularities [5]. Compared to the transpose solution, the pseudoinverse solution is computationally demanding in view of real-time applications. An efficient procedure to compute the pseudoinverse of the Jacobian matrix can be devised by taking advantage of the kinematic analysis of the manipulator structure. In order to avoid excessive joint velocities close to singularities, the manipulator can be treated as singular in a suitably defined region around each singularity; inside the region the available extra degree(s) of freedom is used to achieve a continuous joint velocity solution [6].

Another approach to solve the inverse kinematics problem is the damped least-squares technique giving an approximate solution which is well-conditioned and defined everywhere in the manipulator's workspace [7,8]. The approach seems to be promising for real-time applications, as it implicitly removes the unfeasible motion components while it requires less computation than the previous method. A problem is to select a damping factor which results in a satisfactory trade-off between tracking accuracy and feasibility of the resulting joint-space solution. A varying damping factor gives better performance but is, in general, expensive to compute on-line [9]; simplified computation has recently been proposed [10]. Geometrical insight may prove very useful to perform the tuning of the damping factor [11].

Differential kinematics

A velocity $\dot{\mathbf{q}} \in \mathbb{R}^n$ in joint-space coordinates is related to the corresponding velocity $\dot{\mathbf{x}} \in \mathbb{R}^m$ in task-space coordinates through the equation

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (1)$$

where $J(q)$ is the $(m \times n)$ Jacobian matrix of the manipulator considered. In all cases of interest it is $n \geq m$; when $n > m$ the manipulator is said to be *redundant* and there exists an $(n - m)$ -dimensional subspace of R^n in which any joint-space velocity gives a null velocity in the task space.

If for some configuration $\hat{q}$ it happens that $\text{rank}(J(\hat{q})) = r < m$, the configuration is termed as *singular*. At a singular configuration the subspace of the joint velocity space which maps into the null velocity vector in the task space increases its dimension, as $\dim(\mathcal{N}(J(\hat{q}))) = n - r > n - m$. On the other hand, since $\dim(\mathcal{R}(J(\hat{q}))) = r < m$, only an r -dimensional subspace of task-space velocities can be spanned at a singularity; this subspace is the space of feasible motion for the manipulator. In general, at a singular configuration an assigned velocity vector in the task space may thus have both *feasible components*, lying in $\mathcal{R}(J(\hat{q}))$, and *degenerate components*, belonging to $\mathcal{R}^\perp(J(\hat{q}))$. It is clear that no joint velocity can provide a task-space velocity having components in a degenerate direction.

An effective tool to analyze the linear mapping from the joint velocity space into the task velocity space defined by (1) is offered by the singular value decomposition of the Jacobian matrix; this is given by

$$J = U \Sigma V^T = \sum_{i=1}^m \sigma_i u_i v_i^T \quad (2)$$

where U is the $m \times m$ matrix of the output singular vectors u_i , V is the $n \times n$ matrix of the input singular vectors v_i , and $\Sigma = \begin{pmatrix} S & O \end{pmatrix}$ is the $m \times n$ matrix whose $(m \times m)$ diagonal submatrix S contains the singular values σ_i of the matrix J . If r denotes the rank of J , the following hold:

- a) $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_m = 0$
- b) $\mathcal{R}(J) = \text{span}\{u_1, \dots, u_r\}$
- c) $\mathcal{N}(J) = \text{span}\{v_{r+1}, \dots, v_n\}$

Notice that the $m - r$ output singular vectors associated to the null singular values represent the degenerate directions in the given configuration. The singular value decomposition is continuous and well-behaved not only in singular values but also in the direction of the singular vectors; the u_i and v_i vectors will thus not change much in the neighbourhood of a singularity.

The solution of the inverse differential kinematics problem requires to find the joint velocity $\dot{q}$ associated to an assigned task-space velocity $\dot{x}$. In the remainder, even if the robotic system is redundant with respect to the given task, redundancy will not be exploited in solving the inverse kinematics.

The Jacobian transpose method

In the previous section the differential kinematic mapping from the joint velocity space into the task velocity space was described in terms of the range space and null space of the Jacobian matrix. Therefore, the inverse mapping is characterized by the range

space and null space of the transpose of the Jacobian matrix. In particular, the following well-known relations hold:

$$\mathcal{R}(\mathbf{J}) = \mathcal{N}^\perp(\mathbf{J}^T) \quad \mathcal{N}(\mathbf{J}) = \mathcal{R}^\perp(\mathbf{J}^T) \quad (3)$$

which somewhat indicate the possibility of building an inverse mapping based on the Jacobian *transpose* matrix.

On the other hand, it is also known that the Jacobian transpose describes the static mapping from the task force space into the joint torque space, i.e.

$$\mathbf{t} = \mathbf{J}^T(\mathbf{q})\mathbf{f} \quad (4)$$

where $\mathbf{t} \in \mathbb{R}^n$ is the vector of joint torques and $\mathbf{f} \in \mathbb{R}^m$ is the vector of end-effector forces. Notice that eq. (4) can be derived by application of the principle of virtual work to eq. (1). As a consequence, the relations in (3) fully characterize the inherent duality existing between the kinematic mapping and the static mapping.

In view of solving the inverse kinematics problem, the mapping (4) can be conveniently adopted to devise an iterative solution algorithm [2]. In detail, let

$$\mathbf{e} = \mathbf{x}_d - \mathbf{x} \quad (5)$$

denote the difference between an assigned task-space position $\mathbf{x}_d$ and the task-space position $\mathbf{x}$ that can be reconstructed from the current joint space solution. This difference can be used to construct an elastic force vector $\mathbf{K}\mathbf{e}$ —with $\mathbf{K}$ positive definite and usually constant, diagonal—which has to be applied at the end effector of a ‘virtual’ manipulator, with the same kinematics as the actual manipulator but with null mass and unit viscous damping, in order to drive its end-effector position $\mathbf{x}$ to the given position $\mathbf{x}_d$.

Upon these premises, one may adopt an impedance control law for the virtual manipulator which in turn will provide the inverse kinematics solution for the actual manipulator. The sought law then results in

$$\dot{\mathbf{q}} = \mathbf{J}^T(\mathbf{q})\mathbf{K}\mathbf{e}. \quad (6)$$

A simple Lyapunov argument can be used to prove the stability of the closed-loop feedback virtual system and then the convergence of the inverse kinematics algorithm [1]; notice that $\mathbf{J}^T\mathbf{K}$ determines the convergence rate.

If the assigned end-effector position is time-varying, that is one is interested to inverting a task-space trajectory into a corresponding joint-space trajectory, two possibilities exist for the application of solution (6): Get a significant sampling of the given trajectory, let the algorithm solve for each point, and then interpolate between solutions in the joint space. Let the algorithm run on-line along the given trajectory, thus directly generating the joint-space trajectory. In both cases, inversion errors must be tolerated: In the former, those are due to the interpolation. In the latter, those descend from the ‘dynamic’ formulation of the problem, i.e. from the tracking properties of the algorithm.

As a matter of fact, the on-line solution is more appealing for control purposes. In that case, it can be recognized that the problem of algorithm tracking performance becomes crucial; the matrix $\mathbf{J}^T(\mathbf{q})\mathbf{K}$ and the given velocity $\dot{\mathbf{x}}_d$ are the relevant quantities

affecting performance. In particular, in practical discrete-time implementation of the algorithm, an upper bound exists on the norm of $\mathbf{J}^T \mathbf{K}$ which varies with the current configuration.

An enhancement of the algorithm can be achieved by rendering the matrix $\mathbf{J}^T \mathbf{K}$ less sensitive to variations of joint configuration along the trajectory; this is accomplished by choosing a configuration-dependent $\mathbf{K}$ which compensates for variations of $\mathbf{J}$, for instance see [3].

At a kinematic singularity $\hat{\mathbf{q}}$, when $\mathbf{K}\mathbf{e} \in \mathcal{N}(\mathbf{J}^T(\hat{\mathbf{q}}))$ with $\mathbf{e} \neq \mathbf{0}$, it is $\dot{\mathbf{q}} = \mathbf{0}$ and the algorithm may in principle get 'stuck'. It can be easily shown, however, that such an equilibrium point is unstable as long as the time evolution of $\mathbf{x}_d$ drives $\mathbf{K}\mathbf{e}$ outside $\mathcal{N}(\mathbf{J}^T)$. This result is not surprising since, in force of the relations in (3), vectors belonging to the subspace of the degenerate components cannot be accommodated by both the virtual and the actual manipulator.

Further insight into the features of this solution can be gained by considering the singular value decomposition of the Jacobian transpose; from (2) it is

$$\mathbf{J}^T = \sum_{i=1}^m \sigma_i \mathbf{v}_i \mathbf{u}_i^T \quad (7)$$

which reveals a continuous, smooth behaviour of the solution close and through singular configurations.

The Jacobian pseudoinverse method

The most natural approach to solve differential kinematics is based on the inversion of the mapping (1) using [4]

$$\dot{\mathbf{q}} = \mathbf{J}^T(\mathbf{q}) (\mathbf{J}(\mathbf{q}) \mathbf{J}^T(\mathbf{q}))^{-1} \dot{\mathbf{x}}. \quad (8)$$

Notice that when $n = m$ solution (8) simplifies to $\dot{\mathbf{q}} = \mathbf{J}^{-1} \dot{\mathbf{x}}$. Solution (8) presents two major limitations: It is not defined at a singular configuration. In the neighbourhood of a singularity, it gives an exact solution that may result in high joint velocities.

For this reason, an improved solution is obtained using the *pseudoinverse* of the Jacobian. The pseudoinverse $\mathbf{J}^\dagger$ of $\mathbf{J}$ is a unique matrix satisfying the conditions [12]

$$\mathbf{J}^\dagger \mathbf{J} \mathbf{a} = \mathbf{a} \quad \forall \mathbf{a} \in \mathcal{R}^\perp(\mathbf{J}) \quad (9)$$

$$\mathbf{J}^\dagger \mathbf{b} = \mathbf{0} \quad \forall \mathbf{b} \in \mathcal{R}^\perp(\mathbf{J}) \quad (10)$$

$$\mathbf{J}^\dagger(\mathbf{a} + \mathbf{b}) = \mathbf{J}^\dagger \mathbf{a} + \mathbf{J}^\dagger \mathbf{b} \quad \forall \mathbf{a} \in \mathcal{R}(\mathbf{J}) \quad \forall \mathbf{b} \in \mathcal{R}^\perp(\mathbf{J}). \quad (11)$$

Solution (8) is then modified into

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger(\mathbf{q}) \dot{\mathbf{x}} \quad (12)$$

and satisfies the condition

$$\min_{\dot{\mathbf{q}}} \|\dot{\mathbf{q}}\| \quad (13)$$

of all $\dot{\mathbf{q}}$ that fulfill

$$\min_{\dot{\mathbf{q}}} \|\dot{\mathbf{x}} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}\|. \quad (14)$$

Although solution (12) is defined even for singular configurations, high joint velocities will still result in the neighbourhood of singularities [5]. This happens because eq. (12) is equivalent to eq. (8) at non-singular configurations, while it discontinuously offers an approximate solution at singularities; the latter aspect is a source of additional problems since it implies discontinuous solutions in joint space. In detail, by using (12), first the set of all joint velocities ensuring best accuracy is determined via (14) and then the minimum norm joint velocity in that set is chosen via (13) as the solution. Since a minimum norm criterion is used after the accuracy requirement has been satisfied, it is difficult to guarantee feasibility of the solution. This issue becomes crucial when the manipulator attains near-singular configurations; in these cases, indeed, the Jacobian matrix is full-rank but ill-conditioned. As a consequence, an exact solution is possible but very large joint velocities are required if the assigned $\dot{\mathbf{x}}$ has components along directions which become degenerate at the singularity.

In the framework of the singular value decomposition, with reference to eq. (2), the pseudoinverse solution (12) can be written in the form

$$\dot{\mathbf{q}} = \sum_{i=1}^m \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^T \dot{\mathbf{x}}. \quad (15)$$

When a singularity is approached, the smallest singular value tends to zero; this makes the solution very sensitive to the component of the commanded velocity in the $\mathbf{u}_m^T$ direction which thus requires a large joint velocity to be performed. On the other hand, at the singularity the direction $\mathbf{u}_m^T$ becomes degenerate and joint velocity discontinuity is experienced if a non-null task velocity is specified in that direction.

In order to exploit the potential of solution (12), a systematic and efficient procedure to compute the pseudoinverse of the Jacobian matrix is needed as well as the continuity of the joint-space solution must be ensured.

The first problem can be faced by suitably transforming the Jacobian matrix as

$$\mathbf{PJ} = \begin{pmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{pmatrix} \quad (16)$$

where $\mathbf{P}$ is an $(m \times m)$ transformation matrix such that $\mathbf{J}_1$ is an $(r \times n)$ matrix of full rank, and $\mathbf{J}_2$ is the $((m-r) \times n)$ matrix resulting from the above transformation. Once $\mathbf{J}_1$ has been derived, it is straightforward to compute the pseudoinverse of the Jacobian as [13]

$$\mathbf{J}^\dagger = \mathbf{J}_1^T (\mathbf{J}_1 \mathbf{J}_1^T \mathbf{J} \mathbf{J}_1^T)^{-1} \mathbf{J}_1 \mathbf{J}^T. \quad (17)$$

The problem thus remains to find the transformation matrix as simply as possible. It can be recognized that $\mathbf{P}$ is a projector onto a base of the task velocity space, whose first r rows span the subspace of feasible motion and last $(m-r)$ rows represent directions of motion that linearly depend on the first.

Geometrical insight into the kinematic structure is helpful to derive, case by case, the required projector. For typical manipulators it is possible to identify classes of

singular configurations which allow the determination of $\mathbf{P}$ in a systematic manner; to this end, it is convenient to analyze the singularity in a suitable link-fixed frame. This has been demonstrated, for instance, for a six-degree-of-freedom PUMA-like geometry with zero offsets [6]: For the three well-known types of singularities (wrist, elbow, and shoulder) it turns out that the projectors can be expressed in terms of the two rotation matrices from the base frame to the frames of links 5 and 2, respectively. In these cases, the computation of $\mathbf{P}$ is inexpensive as those matrices are already available for Jacobian computation.

Secondly, the problem of continuity of the joint-space solution can be tackled by treating the manipulator as singular in a suitably defined region around each singularity; this implies that only r task space directions are accomplished. Thus, $(n-r)$ degrees of freedom are available in the region, of which $(n-m)$ are the usual redundant degrees of freedom—that are of no interest for the present work—and $(m-r)$ are the extra degrees of freedom generated by regarding the manipulator as if it were singular. The latter are exploited to ensure the continuity of the joint velocity solution.

In order to accomplish the above, the computation of the pseudo inverse is carried out as in (17), using a matrix $\tilde{\mathbf{J}}_1$ which has the same structure in the whole region as in the singularity. Let $\tilde{\mathbf{J}}_1$ denote such matrix; the modified pseudoinverse is computed as

$$\mathbf{J}^- = \tilde{\mathbf{J}}_1^T \left(\tilde{\mathbf{J}}_1 \mathbf{J}^T \tilde{\mathbf{J}}_1^T \right)^{-1} \tilde{\mathbf{J}}_1 \mathbf{J}^T. \quad (18)$$

Continuity can be accomplished by interpolating between the singularity and a border point $\mathbf{q}_b$. Inside the region, the solution (12) is modified into

$$\dot{\mathbf{q}} = \mathbf{J}^-(\mathbf{q})\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}^-(\mathbf{q})\mathbf{J}(\mathbf{q})) \dot{\mathbf{q}}_0 \quad (19)$$

where $(\mathbf{I} - \mathbf{J}^-\mathbf{J})$ is a projector onto $\mathcal{N}(\mathbf{J}_1)$ which approximates $\mathcal{N}(\mathbf{J})$ inside the region but coincides with it at the singularity.

In the case of a single singularity, a simple choice for $\dot{\mathbf{q}}_0$ is

$$\dot{\mathbf{q}}_0 = \alpha \mathbf{J}^\dagger(\mathbf{q}_b) \dot{\mathbf{x}} \quad (20)$$

where α is the interpolation factor which is zero at the singularity and unity at the border of the region. In the case of multiple singularities, one interpolation factor has to be used for each singularity to ensure continuity when the manipulator transits from one restricted region to another region associated with a singularity of different dimension; a continuous solution could be obtained using multiple border points, but this gives a complex solution with increased computation. In a PUMA-like structure, the wrist singularity is of primary concern since it is difficult to predict at the trajectory planning level. In this case, one may think of interpolating only for that singularity, tolerating instead discontinuities due to elbow and shoulder singularities; in fact, those singularities are naturally characterized in the task space and thus they can be avoided during the planning [6].

The damped least-squares Jacobian inverse method

Another method which is especially directed to overcoming the problem of control through singularities makes use of the *damped least-squares inverse* of the Jacobian matrix for solving the inverse differential kinematics problem [7,8]. The method corresponds to using

$$\dot{\mathbf{q}} = \mathbf{J}^T(\mathbf{q}) (\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}) + \lambda^2 \mathbf{I})^{-1} \dot{\mathbf{x}} \quad (21)$$

in lieu of solution (12); in (21) $\lambda \in \mathbb{R}$ is the damping factor. It is easily seen that when λ is zero, solutions (8) and (21) become identical.

Solution (21) satisfies the condition

$$\min_{\dot{\mathbf{q}}} \|\dot{\mathbf{x}} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}\|^2 + \lambda^2 \|\dot{\mathbf{q}}\|^2 \quad (22)$$

which, differently from (13,14), accounts for both accuracy and feasibility in choosing the joint-space velocity required to produce the given task-space velocity. In this regard, it is essential to select a suitable value for the damping factor; small values of λ give accurate solutions but low robustness to the occurrence of singular and near-singular configurations, high values of λ result in low tracking accuracy even where a feasible and accurate solution would be possible.

Resorting to the singular value decomposition (2), the damped least-squares solution (21) can be rewritten as

$$\dot{\mathbf{q}} = \sum_{i=1}^m \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v}_i \mathbf{u}_i^T \dot{\mathbf{x}}. \quad (23)$$

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

$$\frac{\sigma_i}{\sigma_i^2 + \lambda^2} \approx \frac{1}{\sigma_i}. \quad (24)$$

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 required to achieve near-degenerate components of the commanded task velocity. In comparison with the previous pseudoinverse method, solutions (12) and (21) behave identically as long as the singular values are significantly larger than the damping factor.

The damping factor serves as an index for the minimum singular value to establish whether the current configuration can be treated as near-singular. Further, it determines the degree of approximation introduced with respect to the pure least-squares pseudoinverse solution. Therefore, an optimal choice for λ requires consideration of the smallest non-null singular value experienced along the whole trajectory and of the minimum damping needed to ensure feasible joint velocities. The use of a configuration-varying damping factor is advisable to achieve good performance in the entire manipulator's workspace.

The natural choice is to adjust λ as a function of some measure of closeness to a singularity at the current configuration; the closer the arm is to the singularity, indeed, the higher the need is for damping. One proposal has been brought up in [7] as to use

the manipulability measure to determine an appropriate value for the damping factor at each configuration; in order to avoid unnecessary damping far from the singularity, a threshold value of manipulability is assigned above which a pure pseudoinverse is used. The manipulability measure alone, however, does not constitute an absolute measure of closeness to a singularity; this is otherwise provided by the minimum singular value of the Jacobian matrix. A solution that tunes the damping factor according to numerical estimates of the minimum singular value has been developed in [9]; a selective filtering of the directions defined by the singular vectors is also accomplished by isolating the contribution pertaining to the minimum singular value, which has the advantage of removing velocity transformation errors along the feasible directions. A recent study [10] has shown that the singular value decomposition is considerably simplified when applied to the manipulator Jacobian matrix, and this appears promising in view of an on-line computation of the minimum singular value.

An effective solution has been proposed in [11] which attempts to combine the advantages of both the above techniques; namely, the analytical simplicity of a closed-form adjusting law for the damping factor as in [7] and the correctness of shaping the damping factor according to an estimate of the minimum singular value as in [9]. The key of the solution is to exploit kinematical insight and then define a region inside which a very simple estimate of the minimum singular value is devised to be used in the damped least-squares inverse; for instance, in the case of the wrist singularity, the angle itself causing the singularity has been recognized as a good estimate of the minimum singular value [11].

Conclusion

In the above sections we have reviewed —what we believe— the most relevant methods for solving the differential kinematics of a manipulator in the neighbourhood of singular configurations.

The Jacobian transpose method utilizes the differential kineto-static mapping to solve the inverse kinematics problem for a given task-space position. For this reason, it necessitates of a feedback correction term in a closed-loop setting. The advantage is that it provides a computationally cheap solution which is also robust to the occurrence of singularities, although the tracking performance is inherently lower than any exact inverse differential kinematics solution.

The Jacobian pseudoinverse method and the damped least-squares Jacobian inverse method provide a possibly exact inversion of the differential mapping; this implies, however, that, if a joint position solution is desired, a feedback correction is needed to eliminate numerical drift arising from open-loop integration of the joint velocity solution. On the other hand, they are both computationally more demanding than the Jacobian transpose method; this point must be considered in sensory-based real-time applications.

Two counteracting parameters are at issue when evaluating the performance of either of the above two methods; namely, accuracy versus robustness of the solution. In this respect, it can be recognized that the pseudoinverse originates from the desire of finding an accurate solution to the differential kinematic mapping, which though needs

to be relaxed in order to get satisfactory behaviour in the neighbourhood of singularities. On the contrary, the damped least-squares inverse arises from the necessity of devising a singularity-robust inverse solution to the differential kinematic mapping, which though needs to be properly tuned in order to increase its accuracy.

From the discussion in this work, it has emerged that there exist effective refinements of both the pseudoinverse and the damped least-squares inverse solutions which allow to trade accuracy off robustness. To this end, it is often decisive to perform a case-by-case analysis by taking advantage of geometrical insight into the kinematic structure under investigation.

Acknowledgements

This paper reports research work supported by *Ministero dell'Università e della Ricerca Scientifica e Tecnologica* under 40% funds.

References

- [1] A. Balestrino, G. De Maria, and L. Sciavicco, "Robust control of robotic manipulators," *9th IFAC World Congress*, Budapest, H, July 1984.
- [2] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE J. Robotics and Automation*, RA-4, 403-410, 1988.
- [3] P. Chiacchio and B. Siciliano, "Achieving singularity robustness: An inverse kinematic solution algorithm for robot control," *IEE Int. Work. Robot Control: Theory and Application*, Oxford, GB, pp. 149-156, Apr. 1988.
- [4] D.E. Whitney, "The mathematics of coordinated control of prosthetic arms and manipulators," *Trans. ASME J. Dynamic Systems, Measurement, and Control*, 303-309, 1972.
- [5] C.A. Klein and C.H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Systems, Man, and Cybernetics*, SMC-13, 245-250, 1983.
- [6] S. Chiaverini and O. Egeland, "A solution to the singularity problem for six-joint manipulators," *1990 IEEE Int. Conf. Robotics and Automation*, Cincinnati, OH, pp. 644-649, May 1990.
- [7] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *Trans. ASME J. Dynamic Systems, Measurement, and Control*, 108, 163-171, 1986.
- [8] C.W. Wampler, "Manipulator inverse kinematic solutions based on damped least-squares solutions," *IEEE Trans. Systems, Man, and Cybernetics*, SMC-16, 93-101, 1986.
- [9] A.A. Maciejewski and C.A. Klein, "Numerical filtering for the operation of robotic manipulators through kinematically singular configurations," *J. Robotic Systems*, 5, 527-552, 1988.
- [10] A.A. Maciejewski and C.A. Klein, "The singular value decomposition: Computation and application to robotics," *Int. J. Robotics Research*, 8(6), 63-79, 1989.

- [11] O. Egeland, M. Ebdrup, and S. Chiaverini, "Sensory control in singular configurations — Application to visual servoing," *IEEE Int. Work. Intelligent Motion Control*, Istanbul, TR, pp. 401–405, Aug. 1990.
- [12] L. Zadeh and C.A. Desoer, *Linear System Theory*, New York, Mac-Graw-Hill, 1963, p. 577.
- [13] D.Q. Mayne, "On the calculation of pseudoinverses," *IEEE Trans. Automatic Control*, 204–205, Apr. 1969.