

Solving Manipulator Redundancy with the Augmented Task Space Method using the Constraint Jacobian Transpose

Bruno Siciliano

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

Abstract

This tutorial work is aimed at surveying an effective local technique of redundancy resolution for robotic manipulators at the velocity level, namely the augmented task space method. The operational task is augmented by the addition of a proper constraint task that is intended to specify the internal motion of the arm. Conflicting task situations that lead to the occurrence of artificial singularities are tackled by resorting to the Jacobian transpose for the constraint task. Extension of the scheme to the acceleration level is discussed; the issue of null space joint velocity stabilization is particularly addressed and tested in a case study.

1. Introduction

Kinematic redundancy is adopted in robotic manipulators to achieve more dexterous and versatile motions. The redundant degrees of freedom available with respect to a given task can be exploited to generate internal joint motions that reconfigure the structure according to functional constraints [1-6]. Most of the proposed techniques solve redundancy locally, i.e. treating the manipulator Jacobian as a constant matrix evaluated at the current configuration; see [7,8] for tutorials. Global methods were proposed [9-12] to achieve optimal behavior along the whole task trajectory; even if they perform better than local methods, they are not suitable for real-time sensor-based robot control applications.

The natural way of specifying the internal joint motion is the *projected gradient* method [1,13] that achieves an iterative minimization of a configuration dependent objective function. If the manipulator is guaranteed to keep off kinematic singularities, a computationally attractive alternative is to use the *reduced gradient* method [14] that is even more efficient in approaching the local optimum.

A convenient framework to handle redundant manipulators is the *augmented task space* approach independently introduced in [15] and in [16], and later used in [17] under the name of configuration control. The idea is to augment the operational (end-effector) task with a suitable constraint task that influences the internal motion of the arm.

A typical drawback of this method, however, is the occurrence of *artificial singularities* —originally noted in [18] in a different context— of the augmented Jacobian matrix which are introduced in addition to the kinematic singularities, due to conflicts between the two tasks.

The augmented task space approach has a theoretical affinity with the approach based on *inverse kinematic functions* [19] which are defined on a singularity-free workspace; the computation of analytical functions, however, is feasible only for manipulators with a reduced number of joints.

Both the above methods enjoy the desirable property of *repeatability*, i.e. closed task paths generate closed joint paths. Nevertheless, the original technique that was aimed to overcome the non-repeatability problem [20] of the pure pseudoinverse solution method [21] is the *extended Jacobian* technique [18,22]; see also [23] for a mathematical treatment of repeatable strategies. Differently from the augmented task method, the constraint task vector is derived through the optimization of a scalar objective function that needs to be extremized at the initial joint configuration in order to propagate optimality throughout arm motion. However, similarly to the above method, a failure occurs when the extended Jacobian is singular (artificial singularity).

In order to manage conflicting task situations in an augmented task space framework, the *task priority* strategy was introduced [24] that establishes an order of priority between the operational task and the constraint task. The method is computationally more expensive than the previous ones, but remarkably gives a correct primary operational task solution as long as the task Jacobian maintains full-rank. Nonetheless, the solution is ill-conditioned close to artificial singularities due to the use of a pseudoinverse of a matrix that becomes near rank-deficient. This drawback was addressed in [3], where an effective solution was devised by treating the above matrix as singular in the neighborhood of the artificial singularity.

From the preceding discussion it should be quite clear that important issues in the resolution of manipulator redundancy are:

- efficient exploitation of redundant degrees of freedom;
- effective handling of task conflicts, i.e. avoidance of

artificial singularities;

- repeatability of the joint paths generated with an inverse kinematics algorithm;
- reduced computational burden of the solution.

A desirable method that combines the computational simplicity of the augmented task space technique with the effectiveness of the task priority scheme has been recently proposed [25,26] and is surveyed in this tutorial work. The key feature is to adopt the *Jacobian transpose* for the constraint task, so as to generate feasible solutions in the neighborhood of artificial singularities. The original idea refers back to [27,28], where the Jacobian transpose was utilized to cope with kinematic singularities of the end-effector Jacobian.

Furthermore, the extension of the scheme to the second-order (acceleration) level is discussed. In fact, if it is desired to dynamically control a redundant manipulator in the operational space, it is necessary to compute not only the joint velocity solutions but also the acceleration solutions corresponding to the given operational motion. It is shown how to derive an augmented task space acceleration solution algorithm [29], still using the constraint Jacobian transpose, with the addition of a suitable damping term so as to avoid the undesirable effect of unstable internal motion at the velocity level observed in [30,31]. A numerical case study is developed.

2. Task Space Augmentation

Consider a manipulator with an open kinematic chain of links connected by joints. Let q denote the $(n \times 1)$ vector of joint space variables and x_O the $(m \times 1)$ vector of operational space variables, e.g. end-effector location. The direct kinematic equation can be written in the form

$$x_O = f_O(q), \quad (1)$$

where f_O is a vector-valued nonlinear function that is nonlinear for manipulators with revolute joints.

The manipulator is termed *kinematically redundant* when the number of joint variables is greater than the number of operational variables that are necessary to describe a given task. Therefore, a manipulator is intrinsically redundant when the dimension of the operational space is smaller than the dimension of the joint space ($m < n$). Redundancy is anyhow a concept relative to the task assigned to the manipulator. Even in the case of $m = n$, a manipulator can be functionally redundant if only a number of r components of operational space are of concern for the specific task, with $r < m$; in other words there does not absolutely exist a redundant manipulator, but the same manipulator can be redundant with respect to a task and non-redundant with respect to another.

Redundancy can provide the manipulator with *dexterity* and *versatility* in its motions [4-6]. The typical example is constituted by the human arm that has seven degrees of

freedom: three in the shoulder, one in the elbow and three in the wrist, without considering the degrees of freedom in the fingers [32,33]. This manipulator is intrinsically redundant; in fact, if the base and the hand position and orientation are both fixed—that requires six degrees of freedom—the elbow position can be moved thanks to the additional available degree of freedom. Then, for instance, it is possible to avoid obstacles in the workspace [3]. Further, if a joint reaches its mechanical limit, there might be other joints that allow the execution of the programmed end-effector motion [1].

The *augmented task space* approach [15,16] provides a natural framework to exploit redundancy in robotic systems. An additional constraint task is introduced by specifying a $(p \times 1)$ vector x_C as a function of the manipulator joint variables, i.e.

$$x_C = f_C(q). \quad (2)$$

Obviously, it is $p \leq n - r$ so as to constrain at most all the available redundant degrees of freedom. Augmenting eq.(1) with eq. (2) gives

$$x = \begin{pmatrix} x_O \\ x_C \end{pmatrix} = \begin{pmatrix} f_O(q) \\ f_C(q) \end{pmatrix}, \quad (3)$$

whose solution q has to satisfy the original operational task and meet the constraint task.

An analysis of eq. (3) is difficult in view of its nonlinear form that does not always allow to obtain closed-form solutions [19]. Hence, it is customary to consider the differential mapping that relates joint velocities \dot{q} to task velocities \dot{x} , i.e.

$$\dot{x} = \begin{pmatrix} J_O(q) \\ J_C(q) \end{pmatrix} \dot{q} = J(q)\dot{q}, \quad (4)$$

where J_O and J_C are respectively the $(r \times n)$ operational Jacobian matrix and the $(p \times n)$ constraint Jacobian matrix. Eq. (4) is more tractable than eq. (3) thanks to its linearity in the joint velocities. Therefore, once a task trajectory $x(t)$ is assigned, one might compute a joint velocity solution \dot{q} through a (pseudo)inverse of the augmented Jacobian matrix $J(q)$, i.e.

$$\dot{q} = J^\dagger(q)\dot{x}, \quad (5)$$

and then integrate over time—with known initial condition $q(0)$ —to find a joint trajectory solution $q(t)$. The symbol “ \dagger ” denotes the pseudoinverse of a matrix which reduces to the inverse when the matrix is square. Notice that (5) corresponds only to a local resolution of redundancy, since the Jacobian changes with the arm configuration. Further, if the whole space of redundancy is spanned, i.e. $p = n - r$, the solution (5) generates repeatable joint paths for repeatable task paths [19,23].

A crucial issue for the Jacobian matrix (pseudo)inversion is the occurrence of rank deficiencies. In the case of eq. (4),

these are imputable not only to kinematic singularities of the operational Jacobian $J_O(q)$ and to constraint singularities of the constraint Jacobian $J_C(q)$, but also to the singularities of the augmented Jacobian $J(q)$. In other words, even if both J_O and J_C are non-singular, J may be singular. This happens when the rows of J_C become linearly dependent on the rows of J_O , indicating that the constraint task is in conflict with the operational task at the current configuration q . In technical terms, if $\text{rank}(J_O) = r$ and $\text{rank}(J_C) = p$, then $\text{rank}(J) = r + p$ if and only if $\mathcal{R}(J_O^T) \cap \mathcal{R}(J_C^T) = \{\emptyset\}$, where $\mathcal{R}(\cdot)$ denotes the range space [26].

In this case, the manipulator is said to be in an *artificial singularity*, and no feasible solution for \dot{q} exists unless $\dot{x} \in \mathcal{R}(J)$. Actually, the constraint task is often chosen to keep the manipulator off kinematic singularities; thus, the occurrence of an artificial singularity is really an undesirable effect from a practical viewpoint.

An effective way to handle the conflicting task situations is offered by the *task priority strategy* [24], that assigns different priorities to the operational task and the constraint task and ensures the correct execution of the task with higher priority. Instead of solving eq. (4) as in (5), the joint velocity solution is computed as—dropping the dependence on q —

$$\dot{q} = J_O^\dagger \dot{x}_O + (I - J_O^\dagger J_O)(J_C(I - J_O^\dagger J_O))^\dagger (\dot{x}_C - J_C J_O^\dagger \dot{x}_O), \quad (6)$$

where I denotes the $(r \times r)$ identity matrix. The operator $(I - J_O^\dagger J_O)$ projects the secondary velocity contribution on the null space $\mathcal{N}(J_O)$, guaranteeing correct execution of the primary operational task which is then unaffected by the constraint task. Obviously, if desired, the order of priority can be switched, e.g. in an obstacle avoidance task when an obstacle comes to be along the end-effector path.

Solution (6) can be simplified to [3]

$$\dot{q} = J_O^\dagger \dot{x}_O + (J_C(I - J_O^\dagger J_O))^\dagger (\dot{x}_C - J_C J_O^\dagger \dot{x}_O), \quad (7)$$

since the operator $(I - J_O^\dagger J_O)$ is both hermitian and idempotent.

The above task priority solutions, however, solve the problem of artificial singularities only in part, because both (6) and (7) still involve the computation of the pseudoinverse of the matrix $J_C(I - J_O^\dagger J_O)$ which is rank-deficient at an artificial singularity. In technical terms, if $\text{rank}(J_O) = r$ and $\text{rank}(J_C) = p$, then $\text{rank}(J_C(I - J_O^\dagger J_O)) = p$ if and only if $\mathcal{R}(J_C^T) \cap \mathcal{R}(J_O^T) = \{\emptyset\}$ [26], which is the same condition as for the above augmented Jacobian matrix. This in turn reveals that, when a pseudoinverse of the matrix $J_C(I - J_O^\dagger J_O)$ exists, the task priority solution (6) (or (7)) becomes just a computationally simpler expression of the pure augmented task solution (5), and still the problem remains in the neighborhood of artificial singularities.

A possible remedy to tackle the above inconvenience is to use a damped least-squares solution [34,35] in connection with the matrix $J_C(I - J_O^\dagger J_O)$ in such a way that the errors due to damping will purely affect the secondary constraint task directions. It is anticipated, however, that the computational requirements of such a solution might be impractical for on-line implementation of the technique [36].

3. Constraint Jacobian Transpose Method

A well-established method to solve the inverse kinematics of robotic manipulators is based on the use of the Jacobian transpose in lieu of the Jacobian (pseudo)inverse [27,28]. Even if limited tracking errors occur, the former has two basic advantages over the latter: it is computationally cheaper and may work also at singularities.

This method can be applied directly to the augmented task Jacobian in (4), leading to the solution

$$\dot{q} = J^T(q) K e, \quad (8)$$

where K is a suitable positive-definite symmetric (diagonal) matrix that weighs the task tracking error $e = x_d - x$, being x_d and x respectively the desired and actual task vectors. A simple Lyapunov argument shows that the error is ultimately bounded along the trajectory $x_d(t)$ —the larger the elements of K , the smaller the norm of e —and is driven asymptotically to zero at steady-state ($\dot{x}_d = 0$). Notice that, in the case of $p = n - r$, also solution (8) enjoys the repeatability property of solution (5); to the purpose, it is sufficient to observe that $\mathcal{R}(J^T) \equiv \mathcal{R}(J^\dagger)$.

In spite of the simplicity of solution (8), problems may occur at an artificial singularity. Specifically, when $K e \in \mathcal{N}(J^T)$ with $e \neq 0$, it is $\dot{q} = 0$ and the algorithm may in principle get stuck. Then, depending on the task directions specified by x_d , the algorithm will guarantee convergence of the sole components of $K e$ outside $\mathcal{N}(J^T)$.

In order to discriminate between the task directions provided by x_O and x_C , the task priority concept illustrated above can be adopted to modify solution (8) appropriately. In detail, assuming that higher priority is given to the operational task, a solution to (4) can be devised in the form [25]

$$\dot{q} = J_O^T K_O e_O + (I - J_O^\dagger J_O) J_C^T K_C e_C, \quad (9)$$

with obvious meaning of the quantities K_O , K_C , e_O , e_C . Notice that solution (9) can be thought as obtained by taking the transpose of the modified augmented Jacobian matrix

$$\begin{pmatrix} J_O \\ J_C(I - J_O^\dagger J_O) \end{pmatrix},$$

where the constraint Jacobian is inherently projected on the null space of the operational Jacobian.

The algorithm based on (9) ensures boundedness of the operational tracking error e_O , independently of the constraint task. Further, if $\text{rank}(J_C) = p$ and $\mathcal{R}(J_O^T) \cap \mathcal{R}(J_C^T) = \{\emptyset\}$, also the constraint tracking error is bounded. The proof goes through a Lypaunov argument [26] and is omitted here.

At this point, it is worth noticing that solution (9) requires computation of the pseudoinverse of the operational Jacobian in any case. This suggests using a solution of the kind (5) for the operational task, but still preserving the use of the Jacobian transpose for the constraint task [26], i.e.

$$\dot{q} = J_O^\dagger(\dot{x}_O + K_{Oe}e_O) + (I - J_O^\dagger J_O)J_C^T K_{Ce}e_C. \quad (10)$$

The algorithm based on (10) gives the same performance as before, but in addition the operational tracking error is null, provided that $e_O(0) = 0$. Actually the first term of (10) represents an effective modification of the pure pseudoinverse solution that avoids the typical problem of numerical drift, thanks to the presence of the feedback correction term $K_{Oe}e_O$.

The essential feature of solution (10) is the use of the constraint Jacobian transpose which makes it to be preferred to the pure task priority solution (7), apart from the presence of the feedback correction term for the operational task.

When a constraint task is specified independently of the operational task, there is no guarantee that the augmented Jacobian remains full-rank along the entire task path and incompatibility between the two tasks may arise. The avoidance of the (pseudo)inversion of the matrix $J_C(I - J_O^\dagger J_O)$ allows the algorithm to work even at an artificial singularity. In technical terms, if $\text{rank}(J_C) = p$ but $\mathcal{R}(J_O^T) \cap \mathcal{R}(J_C^T) \neq \{\emptyset\}$, when $J_C^T K_{Ce}e_C \in \mathcal{R}(J_O^T)$, the second term of solution (10) vanishes with $e_C \neq 0$; the higher-priority operational task path is still tracked ($e_O = 0$) but the errors for the lower-priority constraint task can be tolerated.

In sum, solution (10) constitutes a nice trade-off between the computational simplicity of the augmented Jacobian transpose solution (8) and the effectiveness of the task priority solution (7). The savings in computation comes at the cost of non-null tracking errors for the constraint task, in view of the use of the transpose. It may be observed, however, that the constraint is often constant over time ($\dot{x}_C = 0$); then the actual errors will be smaller, and it can be concluded that the above savings is worthwhile. A number of case studies that demonstrate the effectiveness of the proposed solution method can be found in [25,26].

4. Extension to Acceleration Resolution

All the above schemes solve redundancy at the velocity level. In order to dynamically control a redundant manipulator in the operational space, it is necessary to compute not only the joint velocity solutions but also the acceleration solutions to the given operational space motion

trajectory. The second-order kinematics can be obtained by further differentiating eq. (4), i.e.

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q}. \quad (11)$$

At this point, it would be quite natural to solve (11) for the joint accelerations by regarding the second term on the right-hand side as associated to \ddot{x} . Thus, the (pseudo)inverse solution corresponding to (5) is

$$\ddot{q} = J^\dagger(q)(\ddot{x} - \dot{J}(q, \dot{q})\dot{q}) \quad (12)$$

that can be integrated with respect to time —with known initial conditions $q(0), \dot{q}(0)$ — to find $\dot{q}(t)$ and $q(t)$. Then the set (q, \dot{q}, \ddot{q}) can be fed into the inverse dynamics computation for task space control purposes [37]. A similar acceleration solution can be devised for the task priority strategy that corresponds to the velocity solution (7), i.e.

$$\ddot{q} = J_O^\dagger \ddot{y}_O + (J_C(I - J_O^\dagger J_O))^\dagger (\ddot{y}_C - J_C J_O^\dagger \ddot{y}_O) \quad (13)$$

with

$$\ddot{y}_O = \ddot{x}_{Od} - \dot{J}_O \dot{q} + K_{DO} \dot{e}_O + K_{PO} e_O \quad (14)$$

$$\ddot{y}_C = \ddot{x}_{Cd} - \dot{J}_C \dot{q} + K_{DC} \dot{e}_C + K_{PC} e_C, \quad (15)$$

so as to include also proportional-derivative feedback correction terms for the operational and constraint tasks.

One shortcoming of the above procedure is that solving redundancy at the acceleration level may generate internal instability of joint velocities [30]. The occurrence of this phenomenon, which in fact sets kinetic limitations on the use of redundancy [38], is basically related to the instability of the zero dynamics [39] of the second-order system described by (11) under solutions of the kind (12). A technique that overcomes the above drawback was proposed in [31] but was too computationally demanding, since it was based on the symbolic expression of the derivative of the Jacobian pseudoinverse.

Nonetheless, acceleration solutions can be computed by symbolic differentiation of velocity solutions so as to inherit all the properties of a first-order solution and then avoid the above inconvenience of joint velocity instability [40]. Further insight into the relationship between first-order and second-order methods can be found in [41].

In the framework of the augmented task space method with constraint Jacobian transpose, the second-order solution corresponding to (10) with stabilization of internal joint velocities is [29]

$$\ddot{q} = J_O^\dagger \ddot{y}_O + (I - J_O^\dagger J_O)(J_C^T(K_{DC} \dot{e}_C + K_{PC} e_C) - K_V \dot{q}) \quad (16)$$

with \ddot{y}_O as in (14); K_V is a suitable positive-definite matrix that is used for the damping term in the null space of the solution that provides well-behaved arm motion.

When $e_c = 0$, solution (16) guarantees exponential stability of joint velocities in the null space of the operational Jacobian J_O . Notice that, in the case of a constant constraint task ($\dot{x}_{Cd} \equiv 0$), the added contribution in (16) serves as a regularizing term that ensures positive definiteness of the matrix $J_C^T K_{DC} J_C + K_V$ premultiplying \dot{q} . A simple case study is carried out for a planar arm with three revolute joints and unitary link lengths; absolute joint coordinates were used to simplify direct kinematics computation. The end-effector trajectory to be tracked is cyclic and is described by

$$x_{Od}(t) = \begin{pmatrix} 1 + \sin(\pi t) \\ 1 + \cos(\pi t) \end{pmatrix}.$$

The initial arm configuration is chosen such that $e_O(0) = 0$ but $\dot{e}_O(0) \neq 0$. The actual algorithms are discrete-time versions of the continuous-time solutions presented above and were simulated over two complete cycles with a 2nd order Runge-Kutta integration method at 5 msec sampling time.

In order to show the limitations of solving redundancy at the acceleration level, the resolved acceleration solution with feedback correction term (13) but without null space contribution (no constraint) is tested first, using $K_{PO} = \text{diag}\{100, 100\}$ and $K_{DO} = \text{diag}\{20, 20\}$. The resulting stroboscopic motion of the arm is sketched in Fig. 1. The joint trajectories (Fig. 2) reveal the build-up of unstable velocities near the completion of the second cycle. Note also the non-repeatability of the joint paths after the completion of the first cycle.

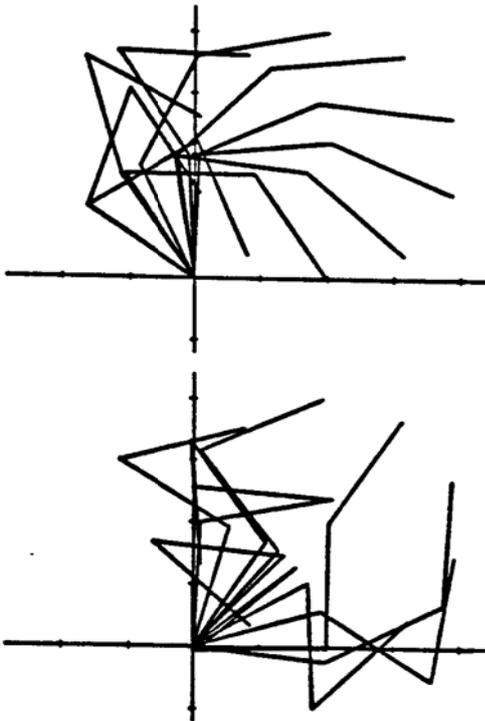


Fig. 1. Stroboscopic motion of the arm with the pure resolved acceleration solution: first cycle (above) and second cycle (below).

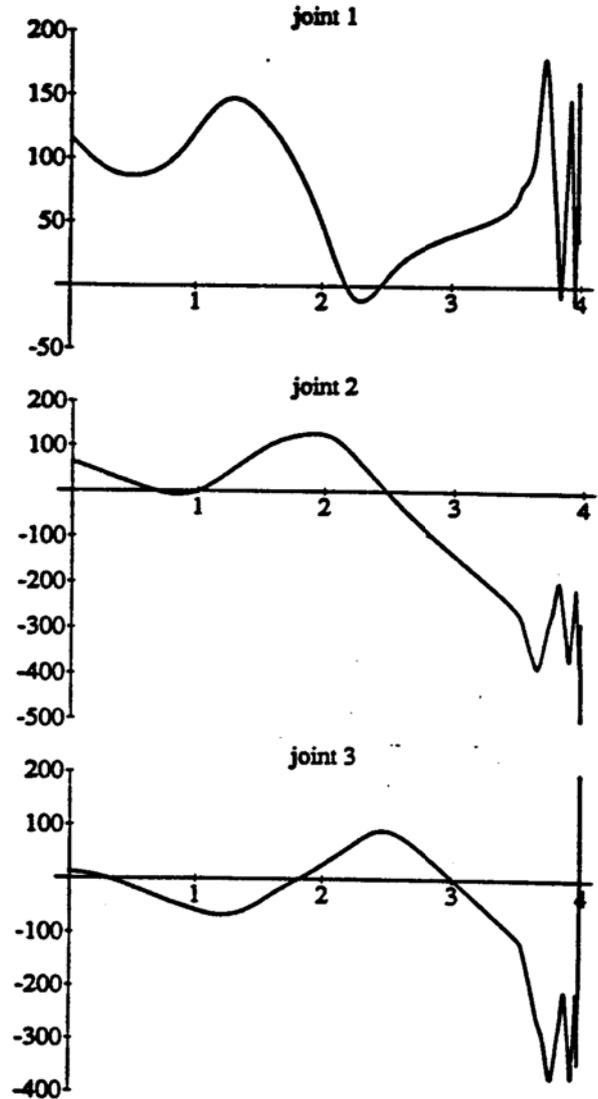


Fig. 2. Joint trajectories (in degrees) with the pure resolved acceleration solution.

Next, the stable augmented task space (16,14) is tested with the constraint task specified as the typical manipulability measure [42]

$$x_C = \sin^2(q_2 - q_1) + \sin^2(q_3 - q_2)$$

with $x_{Cd} = 2$; the gains are $K_{PO} = \text{diag}\{100, 100\}$, $K_{DO} = \text{diag}\{20, 20\}$, $K_V = \text{diag}\{40, 40, 40\}$, $k_{PC} = 1000$, $k_{DC} = 5$. The overall motion of the arm is now smoother, and is repeatable after the first cycle completion (Fig. 3); the joint trajectories (Fig. 4) demonstrate the expected stability property and the time history of the constraint task (Fig. 5) shows a satisfactory motion which, differently from the above resolved acceleration solution, is well far from kinematic singularities.

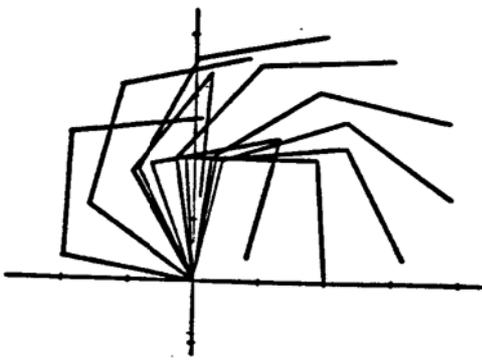


Fig. 3. Stroboscopic motion of the arm with the stable augmented task space solution.

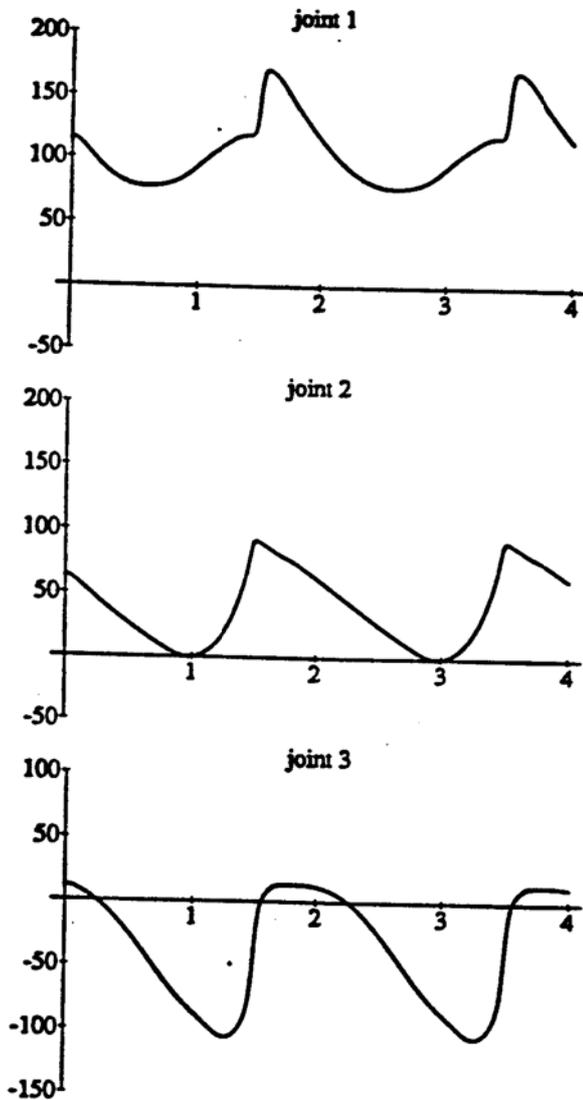


Fig. 4. Joint trajectories (in degrees) with the stable augmented task space solution.

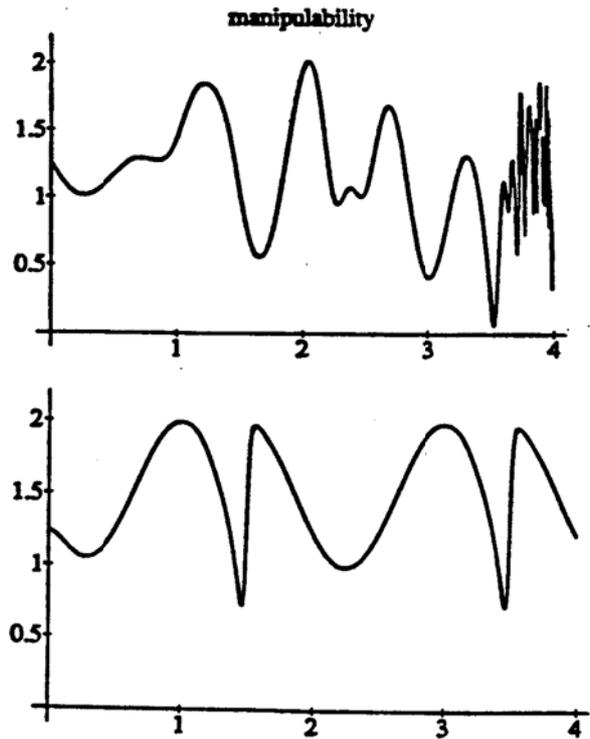


Fig. 5. Manipulability measure: pure resolved acceleration solution (above) and stable augmented task space solution (below).

5. Conclusion

The augmented task space approach has been presented as an effective method to solve kinematic redundancy both at the velocity and the acceleration level. The task priority strategy is the congenial way to handle occurrence of conflicting task situations between the operational task and the constraint task that cause artificial singularities of the augmented Jacobian. The adoption of the Jacobian transpose for the constraint task has been argued to be a good solution that inherits both the effectiveness of the operational Jacobian null space projection, at the basis of the task priority strategy, and the singularity-robustness and cheapness of the Jacobian transpose computation. A case study has demonstrated the satisfactory performance of the acceleration solution scheme which can provide also stabilization of internal joint velocities.

Acknowledgments

This work was supported by *Consiglio Nazionale delle Ricerche* under contract n. 91.01963.PF67. The material presented benefited in various ways from the contributions of P. Chiacchio, S. Chiaverini, A. De Luca, O. Egeland, Z. Novaković, G. Oriolo, J.R. Sagli, and L. Sciavicco.

References

- [1] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mecha-

- nisms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 7, pp. 868-871, 1977.
- [2] J. Baillieul, J.M. Hollerbach, and R.W. Brockett, "Programming and control of kinematically redundant manipulators," *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, NV, pp. 768-774, 1984.
 - [3] A.A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. of Robotics Research*, vol. 4, no. 3, pp. 109-117, 1985.
 - [4] C.A. Klein and B.E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *Int. J. of Robotics Research*, vol. 6, no. 2, pp. 72-83, 1987.
 - [5] J. Angeles, M. Habib, and C. Lopez-Cajun, "Efficient algorithms for the kinematic inversion of redundant robot manipulators," *Int. J. of Robotics and Automation*, vol. 3, pp. 106-116, 1988.
 - [6] R.V. Mayorga, B. Ressa, and A.K.C. Wong, "A dexterity measure for robot manipulators," *Proc. 1990 IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, pp. 656-661, 1990.
 - [7] D.N. Nenchev, "Redundancy resolution through local optimization: A review," *J. of Robotic Systems*, vol. 6, pp. 769-798, 1989.
 - [8] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," *J. of Intelligent and Robotic Systems*, vol. 3, pp. 202-212, 1990.
 - [9] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," *Int. J. of Robotics Research*, vol. 6, no. 1, pp. 32-42, 1987.
 - [10] K.C. Suh and J.M. Hollerbach, "Local versus global torque optimization of redundant manipulators," *Proc. 1987 IEEE Int. Conf. on Robotics and Automation*, Raleigh, NC, pp. 619-624, 1987.
 - [11] K. Kazerooni and Z. Wang, "Global versus local optimization in redundancy resolution of robotic manipulators," *Int. J. of Robotics Research*, vol. 7, no. 5, pp. 3-12, 1988.
 - [12] D.P. Martin, J. Baillieul, and J.M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," *IEEE Trans. on Robotics and Automation*, vol. 5, pp. 529-533, 1989.
 - [13] R.V. Dubey, J.A. Euler, and S.M. Babcock, "An efficient gradient projection optimization scheme for a seven-degree-of-freedom redundant robot with spherical wrist," *Proc. 1988 IEEE Int. Conf. on Robotics and Automation*, Philadelphia, PA, pp. 28-36, 1988.
 - [14] A. De Luca and G. Oriolo, "The reduced gradient technique for solving redundancy in robot arms," *Robotersysteme*, vol. 7, pp. 117-122, 1991.
 - [15] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE J. of Robotics and Automation*, vol. 3, pp. 471-475, 1987.
 - [16] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," *IEEE J. of Robotics and Automation*, vol. 4, pp. 403-410, 1988.
 - [17] H. Seraji, "Configuration control of redundant manipulators: Theory and implementation," *IEEE Trans. on Robotics and Automation*, vol. 5, pp. 472-490, 1989.
 - [18] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," *Proc. 1985 IEEE Int. Conf. on Robotics and Automation*, St. Louis, MO, pp. 722-728, 1985.
 - [19] D.R. Baker and C.W. Wampler, "On the inverse kinematics of robot manipulators with redundancy," *Int. J. of Robotics Research*, vol. 7, no. 3, pp. 3-21, 1988.
 - [20] C.A. Klein and C.-H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 13, pp. 245-250, 1983.
 - [21] D.E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. on Man-Machine Systems*, vol. 10, pp. 47-53, 1969.
 - [22] J. Baillieul and D.P. Martin, "Resolution of kinematic redundancy," *Proc. of Symposia in Applied Mathematics*, vol. 41, pp. 49-89, 1990.
 - [23] T. Shamir and Y. Yomdin, "Repeatability of redundant manipulators: Mathematical solution of the problem," *IEEE Trans. on Automatic Control*, vol. 33, pp. 1004-1009, 1988.
 - [24] Y. Nakamura, H. Hanafusa, T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," *Int. J. of Robotics Research*, vol. 6, no. 2, pp. 3-15, 1987.
 - [25] P. Chiacchio and B. Siciliano, "A closed-loop Jacobian transpose scheme for solving the inverse kinematics of nonredundant and redundant wrists," *J. of Robotic Systems*, vol. 6, pp. 601-630, 1989.
 - [26] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," *Int. J. of Robotics Research*, vol. 10, pp. 410-425, 1991.
 - [27] L. Sciavicco and B. Siciliano, "Coordinate transformation: A solution algorithm for one class of robots," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 16, pp. 550-559, 1986.
 - [28] A. Balestrino, G. De Maria, L. Sciavicco, and B. Siciliano, "An algorithmic approach to coordinate transformation for robotic manipulators," *Advanced Robotics*, vol. 2, pp. 315-404, 1988.
 - [29] Z. Novaković and B. Siciliano, "A new second-order inverse kinematics solution for redundant manipulators," in *Advances in Robot Kinematics*, S. Stifter

- and J. Lenarčič (Eds.), Springer-Verlag, Wien, A, pp. 408-415, 1991.
- [30] J.M. Hollerbach and K.C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE J. of Robotics and Automation*, vol. 3, pp. 308-316, 1987.
- [31] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," *J. of Robotic Systems*, vol. 6, pp. 133-148, 1989.
- [32] A. Hemami, "On a human-arm-like mechanical manipulator," *Robotica*, vol. 5, pp. 23-28, 1987.
- [33] S. Chiaverini, B. Siciliano, and O. Egeland, "Redundancy resolution for the human-arm-like manipulator," *Robotics and Autonomous Systems*, vol. 8, pp. 239-250, 1991.
- [34] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 108, pp. 163-171, 1986.
- [35] C.W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares method," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 16, pp. 93-101, 1986.
- [36] O. Egeland, J.R. Sagli, I. Spangelo, and S. Chiaverini, "A damped least-squares solution to redundancy resolution," *Proc. 1991 IEEE Int. Conf. on Robotics and Automation*, Sacramento, CA, pp. 945-950, 1991.
- [37] L. Sciavicco and B. Siciliano, "The augmented task space approach for redundant manipulator control," *Proc. 2nd IFAC Symp. on Robot Control*, Karlsruhe, D, pp. 125-129, 1988.
- [38] A.A. Maciejewski, "Kinetic limitations on the use of redundancy in robotic manipulators," *IEEE Trans. on Robotics and Automation*, vol. 7, pp. 205-210, 1991.
- [39] A. De Luca, "Zero dynamics in robotic systems," in *Nonlinear Synthesis*, C.I. Byrnes and A. Kurzhanski (Eds.), Progress in Systems and Control Series, Birkhäuser, Boston, MA, 1991.
- [40] B. Siciliano and J.-J.E. Slotine, "A general framework for managing multiple tasks in highly redundant robotic systems," *Proc. 5th Int. Conf. on Advanced Robotics*, Pisa, I, pp. 1211-1216, 1991.
- [41] A. De Luca and G. Oriolo, "Issues in acceleration resolution of robot redundancy," *Prepr. 3rd IFAC Symp. on Robot Control*, Vienna, A, pp. 665-670, 1991.
- [42] T. Yoshikawa, "Manipulability of robotic mechanisms," *Int. J. of Robotics Research*, vol. 4, no. 2, pp. 3-9, 1985.