LRA, Vol. 4, pp. 97–106

Robot Redundancy Resolution at the Acceleration Level

Alessandro De Luca[1], *Giuseppe Oriolo*[1],
and Bruno Siciliano[2]

We present different methods for solving robot kinematic redundancy at the acceleration level. Their features are discussed with respect to velocity solution schemes and potential benefits are highlighted. The following strategies are pursued: local optimization of objective functions that depend on both position and velocity, task augmentation with stable internal motion, and a second-order extended Jacobian approach. The resulting solutions are critically compared in the light of achieving enhanced task trajectory tracking performance with reduced computational complexity. The numerical results obtained for a case study with a planar arm validate the theoretical findings.

1. INTRODUCTION

Kinematic redundancy is purposely introduced in robot manipulators to obtain more versatile motions and enhanced interaction with the environment. The extra (redundant) degrees of freedom can be conveniently exploited to generate an internal joint motion that reconfigures the structure according to given task specifications. Most of the proposed techniques solve redundancy locally, that is, with information limited to the current point of the task trajectory, at the velocity level; recent surveys can be found in [1, 2]. Global methods have also been developed [3–6] to achieve optimal behavior along the whole task trajectory; in general, they outperform local methods but are computationally expensive—typically requiring the numerical solution of a two-point boundary value problem—and then impractical for real-time sensor-based robot control applications.

If one wishes to dynamically control a manipulator in the task (Cartesian) space, it is necessary to compute the explicit inverse kinematic transformations relating both joint velocity and acceleration to the given end-effector motion. On the other hand, the synthesis of joint accelerations in redundant robots usually requires a more involved analysis but allows to directly face dynamic issues, as opposed to velocity schemes which often result in poor dynamic performance. The performance of second-order versus first-order schemes have recently been discussed from an algorithmic point of view in [7].

The basic minimum joint acceleration norm solution was proposed in [8]. Inverse kinematics was also solved at the second-order level in the framework of augmented task space with task priority [9, 10]. In [11] it was first shown that solving redundancy at the acceleration level by local torque minimization may generate internal instability of joint velocities. The more general occurrence of this phenomenon was later investigated in [12], where addi-

---

[1] Dr. Alessandro De Luca is a Research Associate and Giuseppe Oriolo is a Doctoral Candidate, both at Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma "La Sapienza," Via Eudossiana 18, 00184 Roma, Italy.

[2] Dr. Bruno Siciliano is a Research Associate at Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy.

Manuscript received 2/15/92

0895-7533/$3.50 + .25

tional control of joint velocities in the null space of the end-effector Jacobian is provided as a remedy. A simpler scheme which, differently from [12], does not require the computation of the time derivative of the Jacobian pseudoinverse was introduced in [13]. A detailed analysis limited to self-motions, that is, joint movements with zero end-effector velocity, can be found in [14], while a stabilizing control law was proposed for this case in [15]. Nonetheless, acceleration solutions can be computed by symbolic differentiation of velocity solutions, for example [16]; these may avoid the above inconvenience of joint velocity instabilities, but do not allow full exploitation of the capabilities of second-order methods.

The goal of this paper is to present a number of feasible alternatives for redundancy resolution at the acceleration level. Connections with velocity schemes are pointed out, leading to the development of effective algorithms to design suitable joint acceleration contributions in the Jacobian null space. In particular, we present three methods: local optimization of objective functions that depend on both position and velocity, task augmentation with stable internal motion, and a second-order extended Jacobian approach. The advantages of the first two solutions are evidenced in terms of reduced computational burden and stability of arm internal motion. The satisfactory performance of these techniques is shown—also with respect to the classical minimum acceleration norm solution—on the basis of a simulated case study for a planar three-joint arm.

2. VELOCITY RESOLUTION SCHEMES

Consider a robot manipulator with n joints executing an m-dimensional task, with $n - m > 0$ being the number of redundant degrees of freedom. The associated direct kinematics is

$$\mathbf{p} = \mathbf{f}(\mathbf{q}), \tag{1}$$

with joint coordinates $\mathbf{q} \in IR^n$ and task variables $\mathbf{p} \in IR^m$. The first-order differential kinematics is

$$\dot{\mathbf{p}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \tag{2}$$

where the robot Jacobian $\mathbf{J} = \partial\mathbf{f}/\partial\mathbf{q}$ is an $m \times n$ matrix. In general, a task trajectory $\mathbf{p}(t)$ is given and the inverse kinematic problem consists in determining a joint trajectory $\mathbf{q}(t)$ satisfying (1) for each t. In view of the nonlinearity of (1), this problem may be solved at a differential level using (2). Due to arm redundancy, at each time instant there exists an infinity of joint velocity solutions of the form

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{p}} + (\mathbf{I} - \mathbf{J}^{\dagger}(\mathbf{q})\mathbf{J}(\mathbf{q}))\mathbf{v}, \tag{3}$$

where $\mathbf{J}^{\dagger}$ is the unique pseudoinverse of $\mathbf{J}$ [17] and $\mathbf{v}$ is an arbitrary vector in IR^n. The $n \times n$ matrix $\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}$ is the projection operator into the null space of the Jacobian. In the full row rank case, the expression of the pseudoinverse is $\mathbf{J}^{\dagger} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$.

Each choice of $\mathbf{v}(t)$ yields a particular motion of the arm in the joint space, always guaranteeing a correct task execution. This vector is usually determined through optimization of some performance criterion or by satisfying an augmented task specification [18].

The usual way for specifying $\mathbf{v}$ is through the iterative minimization of a configuration dependent objective $H(\mathbf{q})$ by the *projected gradient* method [19]

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{p}} - \alpha(\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\,\nabla_{\mathbf{q}}H, \tag{4}$$

where $\alpha > 0$ is a suitable scalar stepsize. It can be recognized [7] that solution (4) exactly minimizes the complete quadratic function

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \tfrac{1}{2}\dot{\mathbf{q}}^T\dot{\mathbf{q}} + \alpha(\nabla_{\mathbf{q}}H)^T\dot{\mathbf{q}} \tag{5}$$

at the current arm configuration $\mathbf{q}$. Thus, (4) is naturally a compromise between the minimization of the joint velocity norm and the optimization of the main criterion. Notice that when $\alpha = 0$ the basic minimum norm Jacobian pseudoinverse solution is recovered [20].

When the Jacobian has full rank, a convenient alternative to (4) is the *reduced gradient* method proposed in [21]. The joint vector is partitioned as $(\mathbf{q}_a, \mathbf{q}_b)$, with $\mathbf{q}_a \in IR^m$ and $\mathbf{q}_b \in IR^{n-m}$, in such a way that $\mathbf{J}_a = \partial\mathbf{f}/\partial\mathbf{q}_a$ is nonsingular. The first set of joint variables is used for satisfying (2), while optimization is properly reduced to the second set. The joint velocity is then computed as

$$\dot{q} = \begin{bmatrix} \dot{q}_a \\ \dot{q}_b \end{bmatrix} = \begin{bmatrix} J_a^{-1} \\ 0 \end{bmatrix}\dot{p} - \alpha\begin{bmatrix} \mathbf{J}_R J_R^T & -J_R \\ -J_R^T & I \end{bmatrix}\nabla_{\mathbf{q}} H, \tag{6}$$

where $\mathbf{J}_R = \mathbf{J}_a^{-1}\mathbf{J}_b$. The above two methods provide different updates, and in general the latter is more efficient in approaching the local optimum.

In order to explicitly use the redundant degrees of freedom, the method of *task augmentation* [22, 23] allows the specification of $n - m$ additional task constraints as functions of the joint variables

$$\mathbf{p}_c = \mathbf{f}_c(\mathbf{q}), \tag{7}$$

whose differentiation gives

$$\dot{\mathbf{p}}_c = \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}} \tag{8}$$

with $\mathbf{J}_c = \partial\mathbf{f}_c/\partial\mathbf{q}$, an $(n - m) \times n$ matrix. The task-priority velocity solution to (2, 8) that avoids the (secondary) constraint task conflicting with the (primary) end-effector task is [9]

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{p}} + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})(\mathbf{J}_c(\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J}))^{\dagger}(\dot{\mathbf{p}}_c - \mathbf{J}_c\mathbf{J}^{\dagger}\dot{\mathbf{p}}), \qquad (9)$$

that, in view of the idempotency of the matrix $\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}$, can be simplified into [24]

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{p}} + (\mathbf{J}_c(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}))^{\dagger}(\dot{\mathbf{p}}_c - \mathbf{J}_c\mathbf{J}^{\dagger}\dot{\mathbf{p}}). \qquad (10)$$

An effective way to handle the occurrence of *algorithmic singularities*, due to rank deficiencies of the matrix $\mathbf{J}_c(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})$, is to adopt the *Jacobian transpose* solution [25]

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{p}} + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{J}_c^T\mathbf{K}_c\mathbf{e}_c, \qquad (11)$$

where $\mathbf{K}_c > \mathbf{O}$ weighs the feedback correction term $\mathbf{e}_c = \mathbf{p}_c - \mathbf{f}_c(\mathbf{q})$.

A method that somehow combines the features of optimization and task augmentation is the *extended Jacobian* technique proposed in [26]. Assume that the current configuration $\mathbf{q}$ is a constrained optimum, that is, necessarily $(-\mathbf{J}_R^T\,\mathbf{I})\nabla_{\mathbf{q}}H = \mathbf{0}$ [27]. Imposing the following (augmented) condition ensures propagation of optimality throughout arm motion:

$$\frac{d}{dt}\left((-\mathbf{J}_R^T\mathbf{I})\nabla_q\mathbf{H}\right) = \mathbf{G}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{O}. \qquad (12)$$

The velocity solution is then

$$\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J} \\ \mathbf{G} \end{bmatrix}^{-1} \begin{bmatrix} \dot{\mathbf{p}} \\ \mathbf{0} \end{bmatrix}, \qquad (13)$$

provided that no algorithmic singularities are encountered.

3. ACCELERATION RESOLUTION SCHEMES

The second-order differential kinematics associated to (1) is

$$\ddot{\mathbf{p}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}, \qquad (14)$$

and the general solution for joint acceleration is

$$\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\,\ddot{\mathbf{r}}(\mathbf{q}, \dot{\mathbf{q}}) + (\mathbf{I} - \mathbf{J}^{\dagger}(\mathbf{q})\mathbf{J}(\mathbf{q}))\mathbf{a}, \qquad (15)$$

where $\ddot{\mathbf{r}} = \ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}}$ and $\mathbf{a}$ is an arbitrary vector in $I\!R^n$.

The basic issue in selecting (15) versus (3) is the differential order at which redundancy resolution takes place. Indeed, first-order resolution typically provides only *path planning* at the joint level, whereas the second-order scheme gives a full *trajectory planning*, that is, with complete timing information. Correspondingly, equation (3) requires at the current time instant only the knowledge of the configuration $\mathbf{q}$, while both $\mathbf{q}$ and $\dot{\mathbf{q}}$ are needed to compute (15). In the latter case, the set $(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ can be directly fed into the robot inverse dynamics computation for task space control purposes.

In order to obtain joint accelerations from a first-order solution, one may differentiate symbolically (3). This corresponds to a specific choice of the null-space vector $\mathbf{a}$ in the second-order solution (15), as stated in the following [7]

Proposition. *Assume that* $\mathbf{p}(t) \in C^2$, *and that the initial joint positions* $\mathbf{q}(0)$ *and velocities* $\dot{\mathbf{q}}(0)$ *are such that* $\mathbf{f}(\mathbf{q}(0)) = \mathbf{p}(0)$, $\mathbf{J}(\mathbf{q}(0))\dot{\mathbf{q}}(0) = \dot{\mathbf{p}}(0)$. *Provided that* $\mathbf{J}(\mathbf{q}(t))$ *is always full rank, the joint trajectory* $\mathbf{q}(t)$ *generated by the first-order solution (3) for any* $\mathbf{v}(t) \in C^1$ *coincides with the one generated by the second-order solution (15) iff*

$$(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{a} = (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\,[\dot{\mathbf{J}}^{\dagger}(\dot{\mathbf{p}} - \mathbf{J}\mathbf{v}) + \dot{\mathbf{v}}\,], \qquad (16)$$

where $\dot{\mathbf{J}}^{\dagger} = d/dt\,(\mathbf{J}^{\dagger})$.

Proof. Differentiate (3) to obtain

$$\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}(\ddot{\mathbf{p}} - \dot{\mathbf{J}}\mathbf{v}) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\dot{\mathbf{v}}. \qquad (17)$$

Plugging (3) in (15) to eliminate $\dot{\mathbf{q}}$, and equating with (17) gives

$$(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{a} = (\dot{\mathbf{J}}^{\dagger} + \mathbf{J}^{\dagger}\dot{\mathbf{J}}\mathbf{J}^{\dagger})(\dot{\mathbf{p}} - \mathbf{J}\mathbf{v}) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\dot{\mathbf{v}}. \qquad (18)$$

In the full rank case, differentiating $\mathbf{J}\mathbf{J}^{\dagger} = \mathbf{I}$ with respect to time and premultiplying by $\mathbf{J}^{\dagger}$ leads to $\mathbf{J}^{\dagger}\dot{\mathbf{J}}\mathbf{J}^{\dagger} = -\mathbf{J}^{\dagger}\mathbf{J}\dot{\mathbf{J}}^{\dagger}$. Substituting this in (18), the thesis follows.

Equality (16) is limited to null-space projections of joint acceleration vectors. Thus, $\mathbf{a} = \dot{\mathbf{J}}^{\dagger}(\dot{\mathbf{p}} - \mathbf{J}\mathbf{v}) + \dot{\mathbf{v}}$ is a sufficient but not necessary condition for coincidence of first and second-order schemes.

In the following, we present a taxonomy of second-order redundancy resolution schemes which are transpositions of the above first-order schemes. In particular, we consider optimization of *mixed* configuration- and velocity-dependent objective functions, *stable* task augmentation, and a *second-order* extended Jacobian approach.

3.1 Optimization of Mixed Objective Functions

When moving to an acceleration level, it is natural to consider objective functions of the form $H(\mathbf{q}, \dot{\mathbf{q}})$, that is, that may depend on both configuration and velocity. However, there is a mathematical difficulty in optimizing a function of $2n$ differentially related variables using only the n-dimensional vector of joint accelerations $\mathbf{a}$ (see (15)). The appropriate framework to seek for a solution would be Calculus of Variations, which applies only to globally defined criteria [6]. A practical way to overcome this problem is to adopt a discrete-time approach, using an expansion that accounts for the effect of acceleration on the next position and velocity samples [7].

As with (4, 5), it is straightforward to recognize that the general acceleration solution (15) is the exact constrained minimizer of the complete quadratic function

$$L(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \tfrac{1}{2}\ddot{\mathbf{q}}^T\ddot{\mathbf{q}} - \mathbf{a}^T(\mathbf{q}, \dot{\mathbf{q}})\ddot{\mathbf{q}}. \tag{19}$$

At this point, a reasonable extension of the projected gradient method can be devised when considering $\mathbf{a}$ in (19) as generated from an objective function (to be minimized) that contains separate contributions of joint position and velocity terms. This leads us to choosing as mixed objective

$$H'(\mathbf{q}, \dot{\mathbf{q}}) = \tfrac{1}{2}\dot{\mathbf{q}}^T\mathbf{K}_v\dot{\mathbf{q}} + H(\mathbf{q}), \tag{20}$$

and 'blending' the negative gradients into $\mathbf{a}$ as

$$\begin{aligned}\mathbf{a} &= -\nabla_{\dot{\mathbf{q}}} H' - \lambda\nabla_{\mathbf{q}} H' \\ &= -\mathbf{K}_v\dot{\mathbf{q}} - \lambda\nabla_{\mathbf{q}} H.\end{aligned} \tag{21}$$

The structure of (20) is motivated by the desire of damping (with $\mathbf{K}_v > \mathbf{O}$) the joint velocity in the Jacobian null space, in order to avoid the kind of instability phenomena observed in [11, 12], but still trying to optimize the configuration dependent criterion $H(\mathbf{q})$. The weight λ establishes the relative importance between the two objectives. The resulting acceleration solution is then

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\ddot{\mathbf{r}} - (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})(\mathbf{K}_v\dot{\mathbf{q}} + \lambda\nabla_q H). \tag{22}$$

We remark that the above intuitive reasoning leads to a scheme which is similar to the one formally derived in [7].

3.2 Stable Task Augmentation

Following the guideline of the task augmentation velocity solution (10), the corresponding task-priority acceleration solution can be obtained as [9]

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\ddot{\mathbf{r}} + (\mathbf{J}_c(\mathbf{I} - \mathbf{J}^{+}\mathbf{J}))^{+}(\ddot{\mathbf{r}}_c - \mathbf{J}_c\mathbf{J}^{+}\ddot{\mathbf{r}}), \tag{23}$$

where $\ddot{\mathbf{r}}_c = \ddot{\mathbf{p}}_c - \dot{\mathbf{J}}_c\dot{\mathbf{q}}$ and, by differentiation of (8),

$$\dot{\mathbf{p}}_c = \mathbf{J}_c(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}. \tag{24}$$

Solution (23), besides achieving exact tracking of the end-effector acceleration task (14), provides also exact tracking of the constraint acceleration task (24) in the null space of the end-effector Jacobian.

In the same fashion, the second-order Jacobian transpose solution corresponding to (11) is [13]

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\ddot{\mathbf{r}} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\mathbf{J}_c^T(\mathbf{K}_{pc}\mathbf{e}_c + \mathbf{K}_{dc}\dot{\mathbf{e}}_c), \tag{25}$$

where $\dot{\mathbf{e}}_c = \dot{\mathbf{p}}_c - \mathbf{J}_c(\mathbf{q})\dot{\mathbf{q}}$, and $\mathbf{K}_{pc} > \mathbf{O}$, $\mathbf{K}_{dc} > \mathbf{O}$ are suitable matrices that shape the time behavior of the feedback correction term $\mathbf{e}_c$. Remarkably, this solution is computationally more advantageous than (23) since it avoids pseudoinversion of the matrix $\mathbf{J}_c(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})$—like for the velocity solution (11). However, the best performance is obtained in the case of constant constraint tasks, that is, $\dot{\mathbf{p}}_c = \mathbf{O}$. This feature is common to all inverse kinematics schemes based on the Jacobian transpose [28].

On the other hand, the above joint accelerations are not computed as time derivatives of first-order solutions but directly resolving second-order relationships, and may still suffer from instability problems. Following similar arguments which led to the introduction of a damping term in (22), a more well-behaved arm motion can be obtained with

$$\ddot{\mathbf{q}} = \mathbf{J}^{+}\ddot{\mathbf{r}} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})(\mathbf{J}_c^T(\mathbf{K}_{pc}\mathbf{e}_c + \mathbf{K}_{dc}\dot{\mathbf{e}}_c) - \mathbf{K}_v\dot{\mathbf{q}}_c). \tag{26}$$

When $\mathbf{e}_c = \mathbf{O}$, solution (26) guarantees exponential stability of joint velocities in the null space of the end-effector Jacobian [13, 15]. Moreover, note that in the case of $\dot{\mathbf{p}}_c = \mathbf{O}$, the added contribution in (26) serves as a regularizing term that ensures positive definiteness of the matrix $\mathbf{J}_c^T\mathbf{K}_{dc}\mathbf{J}_c + \mathbf{K}_v$ premultiplying $\dot{\mathbf{q}}$.

3.3 Second-Order Extended Jacobian Method

For the transposition of the extended Jacobian velocity solution (13) to the second-order level, two cases can be considered.

First, let the objective criterion still be of the form $H(\mathbf{q})$. Then, the acceleration solution which preserves optimality can be derived via direct differentiation of (13). In fact, differentiation of (12) with respect to time yields

$$\mathbf{G}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{G}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} = \mathbf{0} \tag{27}$$

which, arranged with (14), leads to

$$\ddot{\mathbf{q}} = \begin{bmatrix}\mathbf{J} \\ \mathbf{G}\end{bmatrix}^{-1}\begin{bmatrix}\ddot{\mathbf{r}} \\ -\dot{\mathbf{G}}\dot{\mathbf{q}}\end{bmatrix}. \tag{28}$$

It is not difficult to show that (28) is equivalent to the time derivative of (13). This equivalence is intrinsic to the exact nature of the extended Jacobian method.

Let now the objective be also a function of the joint velocity, $H(\mathbf{q}, \dot{\mathbf{q}})$. As previously emphasized, in general the current state $(\mathbf{q}, \dot{\mathbf{q}})$ will not extremize H. Assume instead that the current velocity $\dot{\mathbf{q}}$ extremizes H for the given configuration $\mathbf{q}$, as expressed by the $n - m$ conditions $(-\mathbf{J}_h^T\ \mathbf{I})\ \nabla_{\mathbf{q}}H = \mathbf{0}$. Local optimality of velocity will be preserved if the acceleration is such that

$$\frac{d}{dt}\left(\left(-\mathbf{J}_R^T\ \mathbf{I}\right)\nabla_{\dot{\mathbf{q}}}H\right) = \mathbf{K}(\mathbf{q},\dot{\mathbf{q}})\ddot{\mathbf{q}} - \mathbf{s}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{0}. \tag{29}$$

Using this constraint to 'square' the system (14), the accelerations are uniquely determined as

$$\ddot{\mathbf{q}} = \begin{bmatrix}\mathbf{J}\\ \mathbf{K}\end{bmatrix}^{-1}\begin{bmatrix}\ddot{\mathbf{r}}\\ \mathbf{s}\end{bmatrix}. \tag{30}$$

It is interesting to point out the equivalence of this second-order extended Jacobian method with the derivative of the exact first-order solution in the case of objective functions that are quadratic in $\dot{\mathbf{q}}$. For example, consider $H(\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T\dot{\mathbf{q}}$, which is exactly optimized at velocity level by (3) with $\mathbf{v} = \mathbf{0}$, or $\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\dot{\mathbf{p}}$. The necessary condition (29) can be equivalently expressed now as

$$\frac{d}{dt}\left(\left(\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}\right)\dot{\mathbf{q}}\right) = \mathbf{0}, \tag{31}$$

resulting in

$$(\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\ddot{\mathbf{q}} = (\mathbf{J}^{\dagger}\dot{\mathbf{J}} + \dot{\mathbf{J}}^{\dagger}\mathbf{J})\dot{\mathbf{q}}. \tag{32}$$

Using idempotency of the matrix $\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}$, and noting that for any $\ddot{\mathbf{q}}$ there exists a vector $\mathbf{a}$ such that $(\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\ddot{\mathbf{q}} = (\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\mathbf{a}$, the equivalence is shown by applying (16) with $\mathbf{v} = \mathbf{0}$. Thus, solution (30) can be rewritten in this case as $\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}\ddot{\mathbf{p}} + \dot{\mathbf{J}}^{\dagger}\mathbf{J}\dot{\mathbf{q}}$.

3.4 Further Considerations

The following considerations can be drawn from a transversal analysis of the presented acceleration resolution schemes.

- The functional dependence of the objective H for the first and third schemes is a crucial point for discriminating the performance of one solution with respect to another. It clearly emerged that, although approaching the problem at the acceleration level gives more flexibility in the definition of objective functions, no local optimization technique allows to reach optimality for both position and velocity by acting only on the instantaneous acceleration.
- The first two schemes are easier to implement for practical trajectory tracking tasks since they do not require the knowledge of a particular constrained initial state. Although gain selection is critical in (22) and (26), a higher design flexibility is introduced anyway. Instead, the third scheme loses its nice features if not initialized in an optimal state which will be propagated along the trajectory; in addition, it is more prone to algorithmic singularity problems.
- As a matter of fact, the actual algorithms implemented are discrete-time versions of the continuous-time solutions illustrated above. Therefore, in order to improve the robot tracking performance in the task space, it is advisable to modify the desired acceleration quantities with robustifying feedback terms. So, the end-effector acceleration $\ddot{\mathbf{p}}$ can be replaced as

  $$\ddot{\mathbf{p}} \rightarrow \ddot{\mathbf{p}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}, \tag{33}$$

  where $\mathbf{e} = \mathbf{p} - \mathbf{f}(\mathbf{q})$ and $\dot{\mathbf{e}} = \dot{\mathbf{p}} - \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$. A similar modification can be introduced for $\ddot{\mathbf{p}}_c$. Use of feedback correction was indeed mandatory for the solutions (25, 26), since the Jacobian transpose does not provide a true inverse mapping.

Last but not least, it should not be dismissed that any acceleration scheme inherently offers the possibility of taking into account robot dynamic effects in the resolution of redundancy. Let

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{n}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{u} \tag{34}$$

be the well-known manipulator dynamic model with inertia matrix $\mathbf{B}$, nonlinear terms $\mathbf{n}$, and input joint torques $\mathbf{u}$. When the squared norm of the joint torque $\frac{1}{2}\mathbf{u}^T\mathbf{u}$ is considered as a criterion, the objective function remains quadratic and positive definite in terms of the joint acceleration $\ddot{\mathbf{q}}$, being $\mathbf{B} > \mathbf{O}$, and the optimal acceleration solution can be immediately obtained in closed-form. More in general, this is true whenever a quadratic energy-related objective function is chosen.

In particular, since local minimization of pure joint torque may lead to unstable "whipping" phenomena in the joint space [11], closed-form acceleration schemes have been proposed that counteract this effect. Among them we mention the squared inverse inertia weighted method in [29] and the addition of a stabilizing velocity-acceleration term in [30] (see also (19)).

4. CASE STUDY

The proposed methods for resolving redundancy at the acceleration level have been simulated on a planar arm with three rotating joints and equal unitary link lengths.

In the following, absolute joint coordinates q_i (with respect to the x-axis) are used so that the robot Jacobian is simply

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -\sin q_1 & -\sin q_2 & -\sin q_3 \\ \cos q_1 & \cos q_2 & \cos q_3 \end{bmatrix}. \qquad (35)$$

The end-effector trajectory to be tracked is the circle described by

$$x(t) = 1 + \sin \pi t, \qquad (36)$$
$$y(t) = 1 + \cos \pi t.$$

In all trials, the initial state has been chosen as

$$\mathbf{q}(0) = (115° \ 63° \ 11°), \qquad (37a)$$
$$\dot{\mathbf{q}}(0) = (0 \quad 0 \quad 0), \qquad (37b)$$

with the end-effector on the path but with an initial velocity error, corrected though in all schemes by the addition of the end-effector feedback action (33). Simulations were run for 4 sec, corresponding to two complete cycles, using a second order Runge-Kutta integration method with 5 msec sampling time.

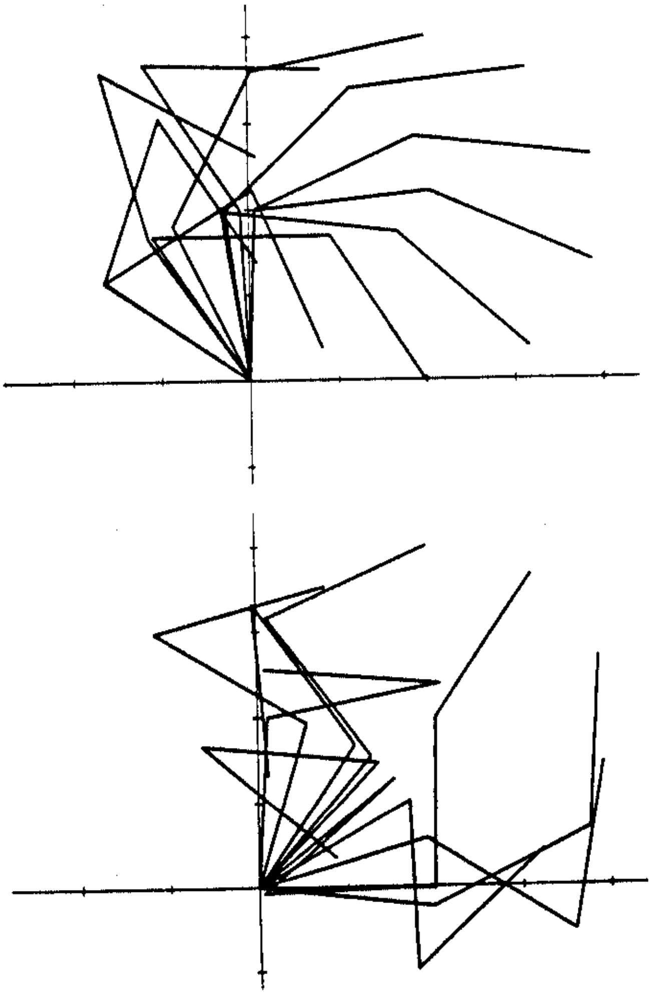

Figure 1. Stroboscopic motion of the arm with the minimum norm acceleration solution (38): first cycle (top); second cycle (bottom).

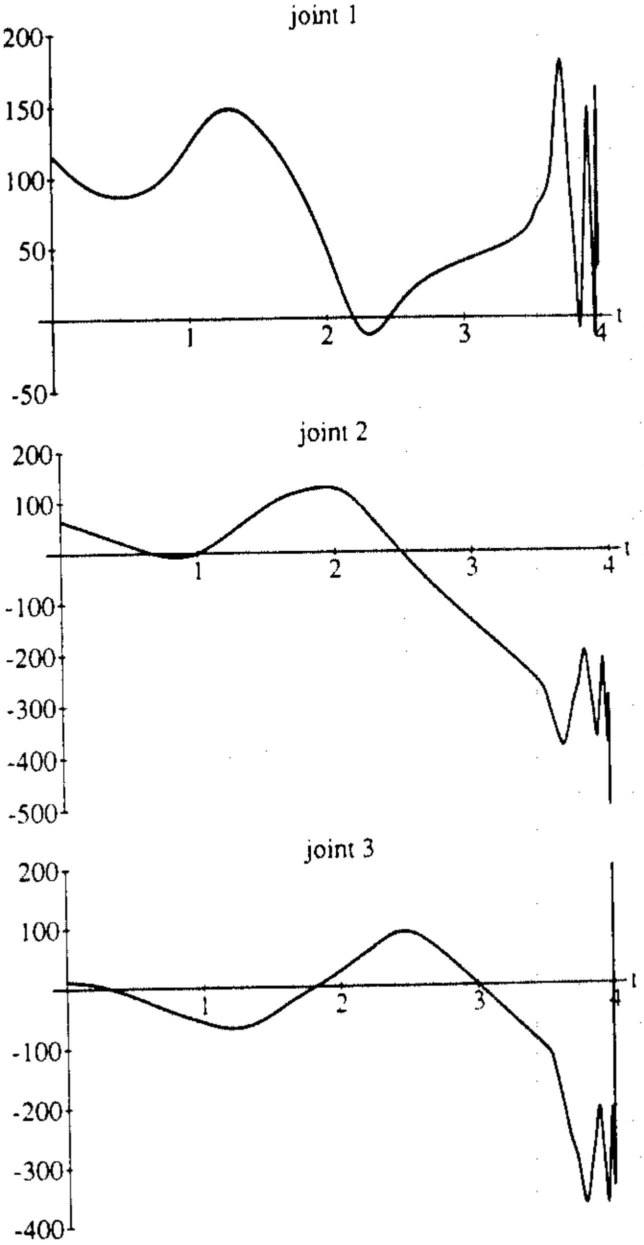

Figure 2. Joint trajectories with the minimum norm acceleration solution (38) (in degs).

In order to show the limitations of a simple acceleration scheme, the minimum acceleration norm solution [8]

$$\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}(\ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) \qquad (38)$$

is tested first, using $\mathbf{K}_p$=diag (100, 100) and $\mathbf{K}_d$ = diag (20, 20). The resulting stroboscopic motion of the arm is sketched in Figure 1.

The joint trajectories (Fig. 2) reveal the build-up of unstable velocities near the completion of the second cycle. These joint velocities are theoretically unobservable at the end-effector level, but when velocity increases numerical saturation occurs and the Cartesian performance is badly degraded.

Next, the second-order optimization method (22) is used in conjunction with the maximization of the following convenient manipulability measure [31]

$$H(\mathbf{q}) = \sin^2(q_2 - q_1) + \sin^2(q_3 - q_2). \qquad (39)$$

Note that the initial configuration (37a) is a maximum for (39), when the end-effector is constrained to be located at $\mathbf{p} = (1 \quad 2)^T$. The addition of the Cartesian error feedback leads to the solution

$$\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}(\ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) - (\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\,(\mathbf{K}_v\,\dot{\mathbf{q}} - \lambda\nabla_{\mathbf{q}}H), \qquad (40)$$

where $\mathbf{K}_p = \text{diag}\,\{100, 100\}$, $\mathbf{K}_d = \text{diag}\,\{20, 20\}$, $\mathbf{K}_v = \text{diag}\,\{50, 50, 50\}$, and $\lambda = 1000$ are used. The overall motion of the arm is now smoother, and repeats itself after the first cycle completion (Figure 3); the joint trajectories (Figure 4) demonstrate the expected stability and a satisfactory general behavior.

Finally, the stable task augmentation scheme (26) is applied using as single additional constraint variable $f_c(\mathbf{q}) = H(\mathbf{q})$ as in (39), and setting the desired constant value $p_c = 2$. Then, the resulting joint acceleration is

$$\ddot{\mathbf{q}} = \mathbf{J}^{\dagger}(\ddot{\mathbf{p}} - \dot{\mathbf{J}}\dot{\mathbf{q}} + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + (\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\,(\mathbf{j}_c^T(k_{pc}e_c - k_{dc}\dot{\mathbf{j}}_c\dot{\mathbf{q}}) - \mathbf{K}_v\dot{\mathbf{q}}), \qquad (41)$$

in which $\mathbf{K}_p = \text{diag}\,\{100, 100\}$, $\mathbf{K}_d = \text{diag}\,\{20, 20\}$, $\mathbf{K}_v = \text{diag}\,\{40, 40, 40\}$, $k_{pc} = 1000$, and $k_{dc} = 5$ are used, and where $\mathbf{j}_c$ is now just a row vector. From the global motion of the arm (Figure 5) and the joint trajectories (Figure 6) similar stability properties as in the previous case are exhibited.

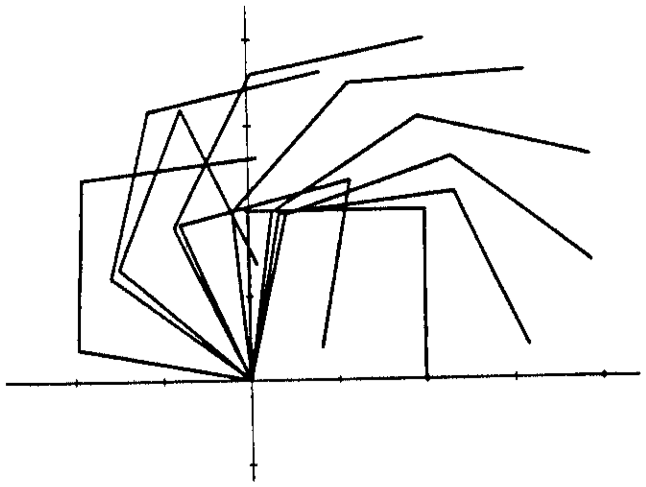

Figure 3. Stroboscopic motion of the arm with the mixed objective function optimization solution (40).

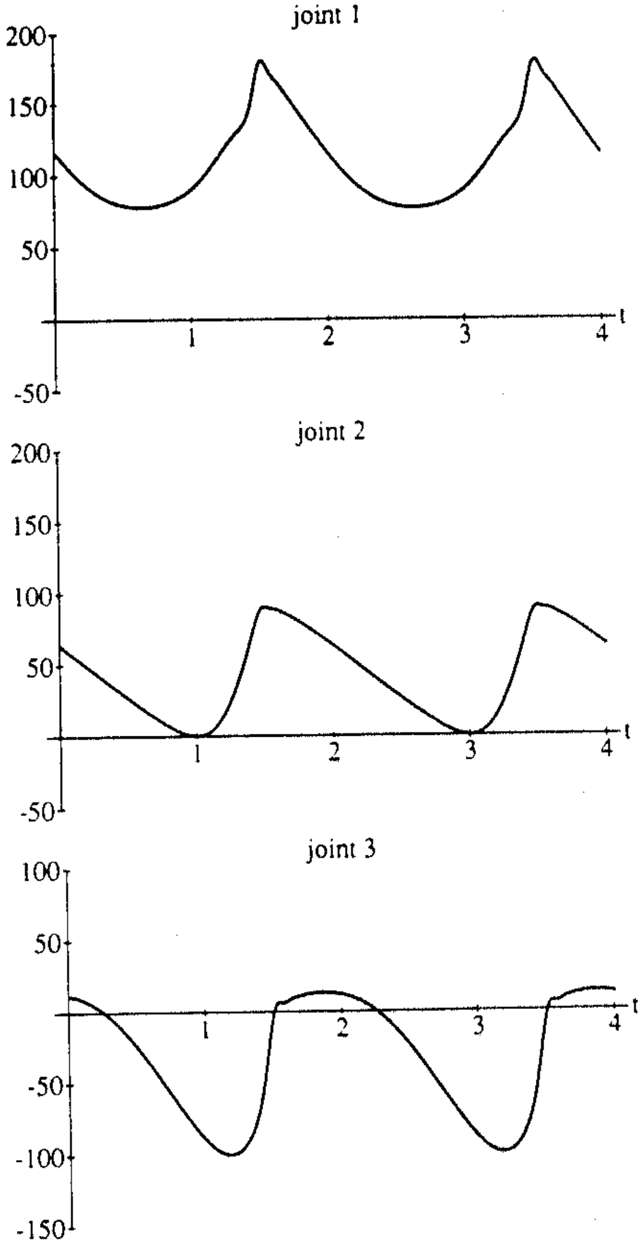

Figure 4. Joint trajectories with the mixed objective function optimization solution (40) (in degs).

Simulation results with the second-order extended Jacobian method are not reported. In fact, if the manipulability measure (39) were used as objective function, a quite involved expressions would be obtained for the solution (28). On the other hand, if the joint velocity norm is chosen, this is equivalent to using a pseudoinverse solution, as shown in Section 3.3. In the present case, the applicability of such a method is discarded by the occurrence of a kinematic singularity at 3/4 of the first cycle.

For the purpose of comparing the performance of the three simulated schemes, Figure 7 displays the evolutions of the joint velocity norm while the

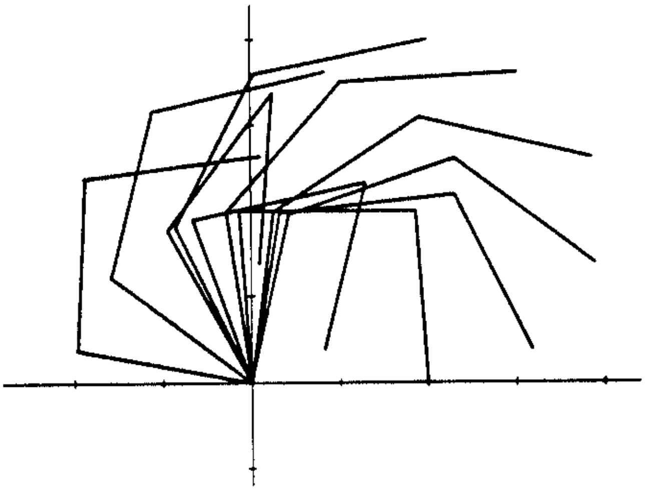

Figure 5. Stroboscopic motion of the arm with the stable task augmentation solution (41).

manipulability measures are reported in Figure 8.

It can be seen that the minimum acceleration solution progressively "accumulates" joint velocity when manipulability tends to decrease, eventually running into trouble after passing close to a singular arm configuration. The two other schemes present high velocity values in correspondence to sudden manipulability losses; however, their built-in dissipation capability allows to recover from these large velocities, without losing track of the end-effector trajectory.

Finally, it should be stressed that the chosen Cartesian path is particularly demanding since it requires the arm to undergo a large reconfiguration in the vicinity of the xy-plane origin. High velocity peaks are essentially due to the absence of preview in local resolution schemes; indeed, a global synthesis of joint accelerations would overcome this problem.

5. CONCLUSIONS

Resolving redundancy at the acceleration level may allow improved robot performance, also in view of a dynamic analysis of the motion control problem. However, some caution should be taken in trying to directly extend first-order schemes to the second-order level. A general result has been given for stating the equivalence between a velocity and an acceleration resolution method. Still, the simple differentiation of a given velocity solution fails to capture the need of internal arm motion stabilization, while yielding computationally inefficient schemes.

The stability issue has been taken into account explicitly both by introducing an additional velocity term in a local objective function to be minimized and by augmenting the task with a joint damping constraint. Moreover, an acceleration solution which shares part of the exact minimization property of the extended Jacobian technique has been derived.

A case study has been presented that gives evidence of the superiority of the proposed redundancy resolution second-order schemes over the classical one with minimization of acceleration norm.

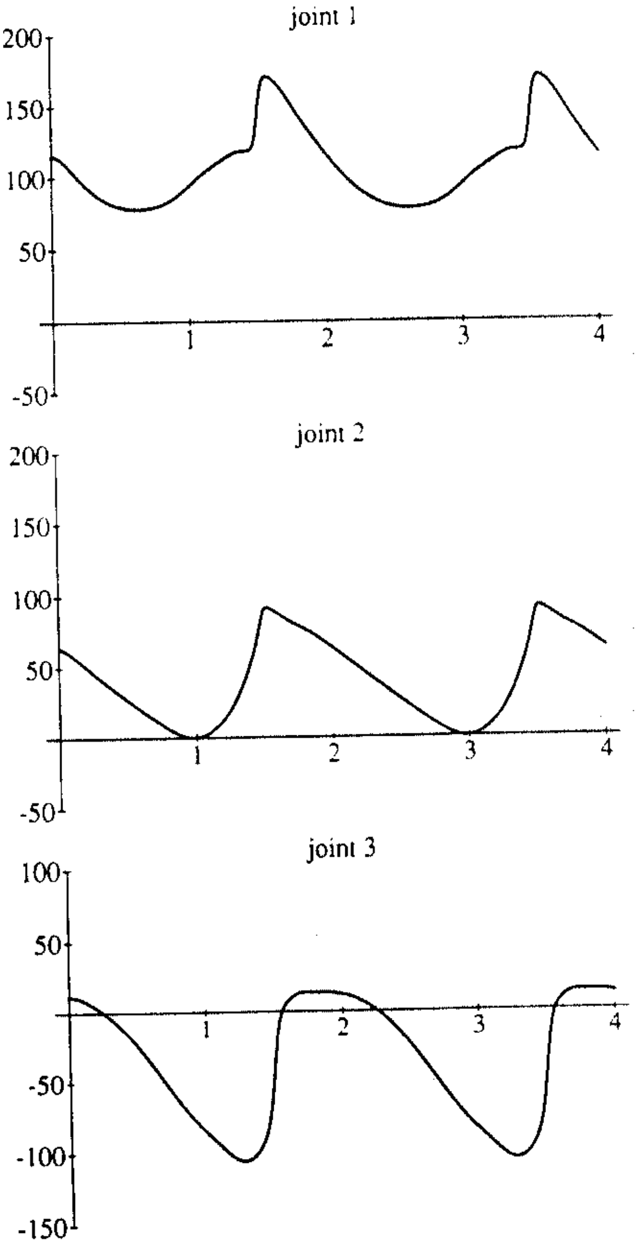

Figure 6. Joint trajectories with the stable task augmentation solution (41) (in degs).

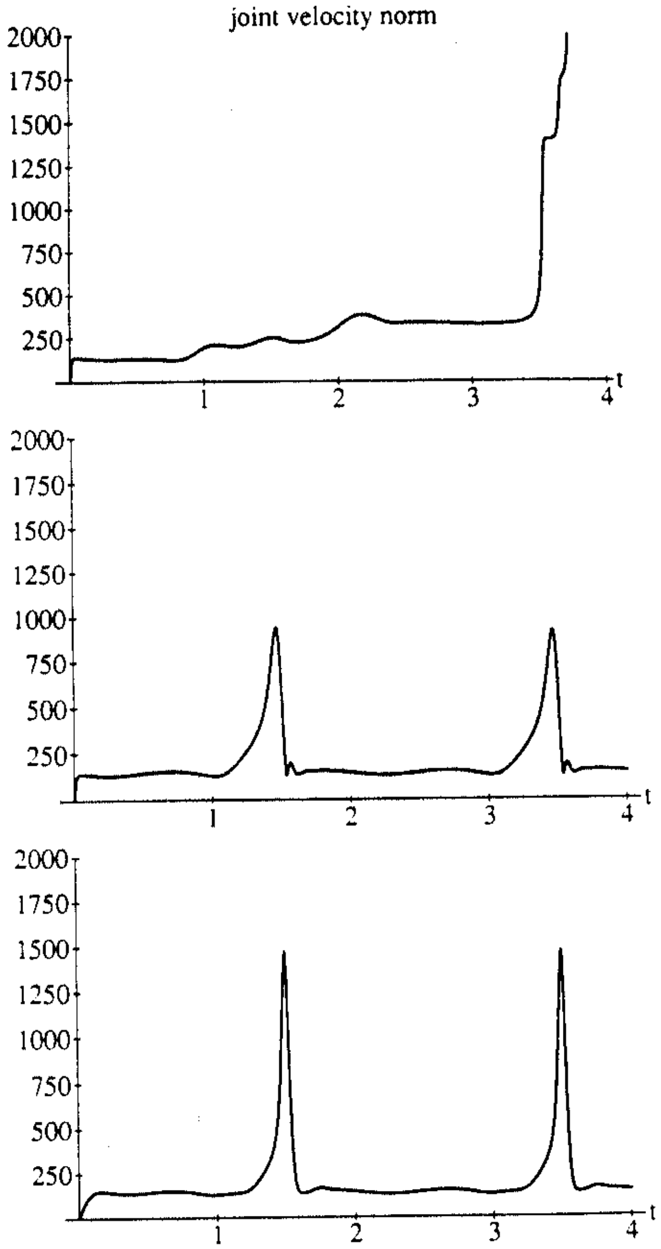

Figure 7. Joint velocity norm: solution (38) (top); solution (40) (center); solution (41) (bottom).

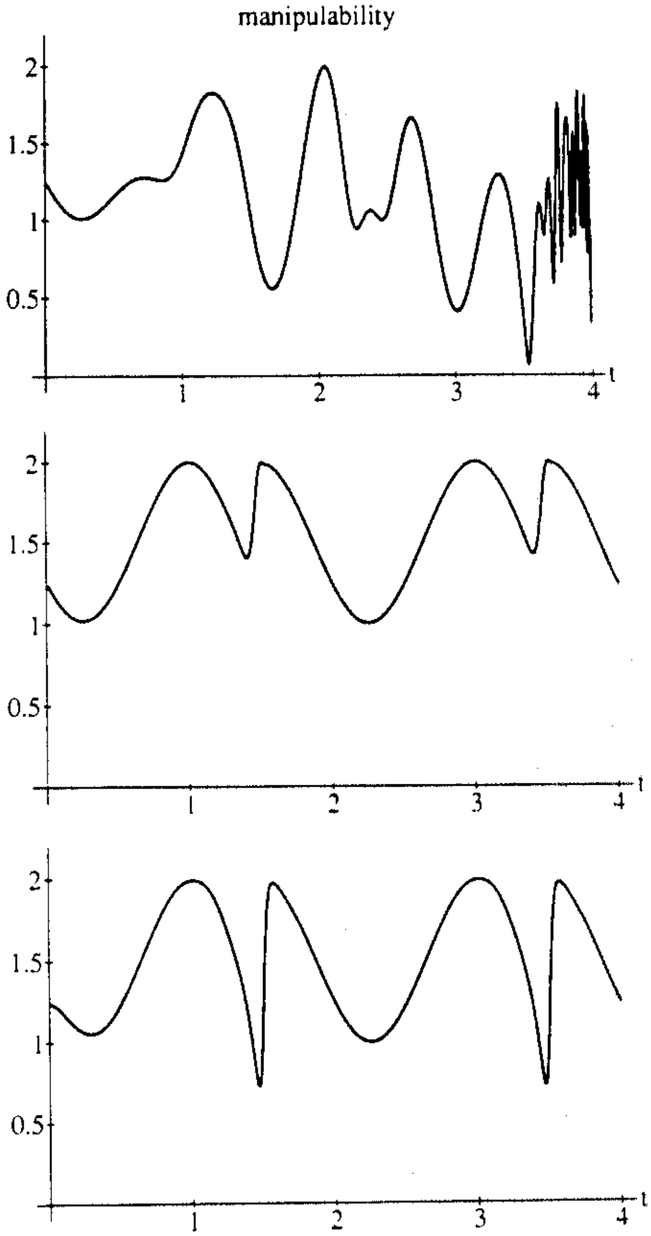

Figure 8. Manipulability measure: solution (38) (top); solution (40) (center); solution (41) (bottom).

ACKNOWLEDGMENTS

This paper is based on work supported by *Consiglio Nazionale delle Ricerche*, contracts no. 91.01946.PF67 and no. 91.01963.PF67 (*Progetto Finalizzato Robotica*).

REFERENCES

[1] D. N. Nenchev, "Redundancy resolution through local optimization: A review, " *Journal of Robotic Systems, 6*, 769–798 (1989).

[2] B. Siciliano, "Kinematic control of redundant robot manipulators: A tutorial," J. of Intelligent and Robotic Systems, *3*, 201–212 (1990).

[3] Y. Nakamura and H. Hanafusa, "Optimal redundancy control of robot manipulators," Int. J. of Robotics Research, *6*, (1), 32–42 (1987).

[4] J. M. Hollerbach and K. C. Suh, "Local versus global torque optimization of redundant manipulators," Proc. 1987 IEEE Int. Conf. on Robotics and Automation, Raleigh, NC, 619–624 (1987).

[5] K. Kazerounian and Z. Wang, "Global versus local optimization in redundancy resolution of robotic

manipulators," Int. J. of Robotics Research, *7*, (5), 3–12 (1988).

[6] D. P. Martin, J. Baillieul, and J. M. Hollerbach, "Resolution of kinematic redundancy using optimization techniques," IEEE Trans. on Robotics and Automation, *5*, 529–533 (1989).

[7] A. De Luca and G. Oriolo, "Issues in acceleration resolution of robot redundancy," Prep. 3rd IFAC Symp. on Robot Control, Vienna, Austria, 665–670 (1991).

[8] Oh Khatib, "Dynamic control of manipulators in operational space," Proc. 6th IFToMM Congr. on Theory of Machines and Mechanisms, New Dehli, India, 1123–1131 (1983).

[9] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators," Int. J. of Robotics Research, *6*, (2), 3–15 (1987).

[10] L. Sciavicco and B. Siciliano, "The augmented task space approach for redundant manipulator control," Proc. 2nd IFAC Symp. on Robot Control, Karlsruhe, Germany, 125–129 (1988).

[11] J. M. Hollerbach and K. C. Suh, "Redundancy resolution of manipulators through torque optimization," IEEE J. of Robotics and Automation, *3*, 308–316 (1987).

[12] P. Hsu, J. Hauser, and S. Sastry, "Dynamic control of redundant manipulators," J. of Robotic Systems, *6*, 133–148 (1989).

[13] 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, Vienna, Austria, 408–415 (1991).

[14] A. A. Maciejewski, "Kinetic limitations on the use of redundancy in robotic manipulators," IEEE Trans. on Robotics and Automation, *7*, 205–210 (1991).

[15] 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, Boson, MA (1991).

[16] 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, Italy, 1211–1216 (1991).

[17] T. L. Boullion and P. L. Odell, Generalized Inverse Matrices, John Wiley, New York, NY (1971).

[18] Y. Nakamura, Advanced Robotics: Redundancy and Optimization, Addison-Wesley, Reading, MA (1991).

[19] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," IEEE Trans. on Systems, Man, and Cybernetics, *7*, 868–871 (1977).

[20] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," IEEE Trans. on Man-Machine Systems, *10*, 47–53 (1972).

[21] A. De Luca and G. Oriolo, "The reduced gradient technique for solving redundancy in robot arms," Robotersysteme, *7*, 117–122 (1991).

[22] O. Egeland, "Task-space tracking with redundant manipulators," IEEE J. of Robotics and Automation, *3*, 471–475 (1987).

[23] L. Sciavicco and B. Siciliano, "A solution algorithm to the inverse kinematic problem for redundant manipulators," IEEE J. of Robotics and Automation, *4*, 403–410 (1988).

[24] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," Int. J. of Robotics Research, *4*, (3), 109–117 (1985).

[25] P. Chiacchio, S. Chiaverini, L. Sciavicco, and B. Siciliano, "Closed-loop inverse kinematics schemes for constrained redundant manipulators with task space augmentation and task priority strategy," Int. J. of Robotics Research, *10*, 410–425 (1991).

[26] J. Baillieul, "Kinematic programming alternatives for redundant manipulators," Proc. 1985 IEEE Int. Conf. on Robotics and Automation, St. Louis, MO, 722–728 (1985).

[27] P. H. Chang, "A closed-form solution for inverse kinematics of robot manipulator with redundancy," IEEE J. of Robotics and Automation, *3*, (5), 393–403 (1987).

[28] B. Siciliano, "A closed-loop inverse kinematic solution for on-line joint-based robot control," Robotica, *8*, 231–243 (1990).

[29] A. Nedungadi and K. Kazerounian, "A local solution with global characteristics for the joint torque optimization of a redundant manipulator," J. of Robotic Systems, *6*, 631–654 (1989).

[30] A. De Luca and G. Oriolo, "Efficient dynamic resolution of robot redundancy," Proc. 1990 American Control Conf., San Diego, CA, 221–227 (1990).

[31] T. Yoshikawa, "Manipulability of robotics mechanisms," Int. J. of Robotics Research, *4*, (2), 3–9 (1985).