RoMänSy 8

Proceedings of the Eight
CISM-IFToMM Symposium
on Theory and Practice
of Robots and Manipulators

Krakow, Poland, July 1990

Edited by
A Morecki, G Bianchi and K Jaworek

Kinematic Resolution of Redundancy via Joint-Space Decomposition

A. De Luca, G. Oriolo

Dipartimento di Informatica e Sistemistica, Universita degli Studi di Roma "La Sapienza", Via Eudossiana 18, 00184 Roma, Italy

ABSTRACT

A new approach for solving inverse kinematic problems in redundant manipulators is presented. Efficient computational methods are derived by reducing the problem in terms of the redundant degrees of freedom only. In particular, joint variables are decomposed into basic and independent ones, and the latter are obtained by optimizing a given criterion or imposing additional constraints. When redundancy is solved via optimization, the proposed approach yields the Reduced Gradient method, a very effective one as compared with the standard Projected Gradient. When a task-augmentation strategy is followed, the decomposition approach can be used to revisit the Task Priority concept, possibly avoiding the need for pseudoinverses. Simple examples illustrated analytically the advantages of the joint-space decomposition technique.

INTRODUCTION

Redundant robot arms offer enhanced motion dexterity and greater flexibility for dealing with complex task specifications, though requiring more sophisticated control algorithms. Redundancy is a relative concept for a manipulator, depending on the difference between the number n of dof's of the mechanical structure and the number m of variables needed to describe the assigned robot task.

When $n-m \geq 1$, the joint-space motion $\mathbf{q}(t)$ can be planned so as to accomplish a primary kinematic task, e.g. a specified end-effector path

$$\mathbf{p}(t)=\mathbf{f}(\mathbf{q}(t)), \qquad \mathbf{q}\in \mathbb{R}^n,\ \mathbf{p}\in \mathbb{R}^m, \quad m<n, \tag{1}$$

while optimizing a performance criterion and/or satisfying additional constraints. Singularity avoidance [1], collision-free motion in the presence of workspace obstacles [2,3], consideration of joint-range limits [4], and task-compatible arm posture selection [5,6,7] are common instances of redundancy utilization.

In most of the existing techniques, a major role is played by the pseudoinverse matrix $\mathbf{J}^{\dagger}(\mathbf{q})$ of the Jacobian $\mathbf{J}(\mathbf{q})=\partial\mathbf{f}/\partial\mathbf{q}$ of the robot direct kinematics. The general solution to $\dot{\mathbf{p}}=\mathbf{J}\dot{\mathbf{q}}$, the differential relation associated to (1), can be written as

$$\dot{\mathbf{q}}=\mathbf{J}^{\dagger}\dot{\mathbf{p}}+(\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J})\mathbf{v}, \tag{2}$$

where $\mathbf{v}\in\mathbb{R}^n$ is an arbitrary joint velocity, which is projected in the null space of $\mathbf{J}$ by means of the projection operator $\mathbf{P}=\mathbf{I}-\mathbf{J}^{\dagger}\mathbf{J}$ [8]. Vector $\mathbf{v}$ can be specified either within an optimization approach or by a suitable task-augmentation.

In order to obtain full benefit from a kinematically redundant design while still keeping limited the computational load, the efficiency issue should be addressed in any redundancy resolution scheme. In general, computation of the pseudoinverse requires the Singular Value Decomposition of $\mathbf{J}$, a numerically intensive and sensitive operation. When the Jacobian has full row rank, $\mathbf{J}^{\dagger}$ takes on the more explicit form $\mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$. Still, being the $n \times n$ projection matrix $\mathbf{P}$ of rank $n-m$, there is a considerable waste of information. As a matter of fact, this is inherent with the idea underlying (2): additional joint motions, which should not perturbate the primary task, are obtained by *projection* in the null space of $\mathbf{J}$.

As opposed to this approach, the concept of *decomposition* and *reduction* is exploited here. More specifically, the primary task constraint (1) is used to locally explicit m joint variables in terms of the remaining $n-m$, which represent the redundant degrees of freedom. The selection of values for these $n-m$ joint variables is the counterpart of the choice of vector $\mathbf{v}$ in (2), and again it can be made according to a given optimization criterion or by imposing additional constraints. In general, criteria or constraints depend on the whole set of joint variables but, as a result of the above joint-space decomposition, they can be reduced to functions of the independent variables only.

The resulting method is particularly efficient since it directly deals with the extra degrees of freedom only, thus avoiding the use of pseudoinversion. Moreover, through the selection of a specific decomposition of joint variables, an explicit command on the particular arm configuration is provided.

In the following, the joint-space decomposition approach will be illustrated within both the optimization and the task-augmentation frameworks for solving redundancy. In the former case, this will lead to a new effective resolution algorithm, the Reduced Gradient method, which is alternative to the well-known Projected Gradient method [4]. In the latter case, a computational improvement will be gained over existing schemes, such as Task-Priority [6], together with a higher flexibility in the constraints handling. Simple analytic examples are used to illustrate fully the above features. A discussion of possible drawbacks of the proposed method together with indications for their solution concludes the paper.

REDUNDANCY RESOLUTION VIA OPTIMIZATION

When redundancy is resolved via local maximization of a performance criterion $H(\mathbf{q})$, the standard approach consists in taking $\mathbf{v}$ in (2) in the direction of the gradient $\nabla_{\mathbf{q}} H(\mathbf{q}) \triangleq (\partial H/\partial \mathbf{q})^T$. The resulting scheme

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

in which α is an adjustable stepsize, was recognized to be the transposition of the *Projected Gradient* (PG) method [4]. Joint velocities (3) tend to increase in time the value of the performance criterion H along the specified end-effector path.

The joint-space decomposition approach offers an alternative way to deal with the same optimization problem. Assuming full row rank for the Jacobian $\mathbf{J}$, it is always possible to rewrite the differential kinematics as

$$\dot{\mathbf{p}} = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_a + \mathbf{J}_b(\mathbf{q})\dot{\mathbf{q}}_b, \tag{4}$$

where $\mathbf{J}_a$ is a square nonsingular matrix. This implies a partition of the joint coordinates $\mathbf{q}$ into two subsets $\mathbf{q}_a \in \mathbb{R}^m$ and $\mathbf{q}_b \in \mathbb{R}^{n-m}$, respectively the basic and the independent variables. The latter locally parametrize the arm redundancy, so that a *formal* inverse function $\mathbf{q}_a = \Phi(\mathbf{p}, \mathbf{q}_b)$ for (1) can be obtained [9]. This explicitation descends directly from the Implicit Function Theorem applied to the direct kinematics. Therefore, when the independent velocity $\dot{\mathbf{q}}_b$ is specified, choosing the basic velocity as

$$\dot{\mathbf{q}}_a = \frac{\partial \Phi}{\partial \mathbf{p}}\dot{\mathbf{p}} + \frac{\partial \Phi}{\partial \mathbf{q}_b}\dot{\mathbf{q}}_b = \mathbf{J}_a^{-1}\dot{\mathbf{p}} - (\mathbf{J}_a^{-1}\mathbf{J}_b)\dot{\mathbf{q}}_b, \tag{5}$$

will guarantee satisfaction of the primary task in any case.

The selection of $\dot{\mathbf{q}}_b$ can be made on the basis of the criterion $H(\mathbf{q}_a, \mathbf{q}_b)$, once this is reduced to $H'(\mathbf{q}_b) \triangleq H(\Phi(\mathbf{p}, \mathbf{q}_b), \mathbf{q}_b)$. In particular, the velocity vector $\dot{\mathbf{q}}_b$ will be chosen to provide the maximum increase for H'. Since the time derivative of H' is (using (5))

$$\dot{H}' = \frac{\partial H'}{\partial \mathbf{q}_b}\dot{\mathbf{q}}_b = \left(\frac{\partial H}{\partial \mathbf{q}_a}\frac{\partial \Phi}{\partial \mathbf{q}_b} + \frac{\partial H}{\partial \mathbf{q}_b}\right)\dot{\mathbf{q}}_b = \begin{bmatrix}\frac{\partial H}{\partial \mathbf{q}_a} & \frac{\partial H}{\partial \mathbf{q}_b}\end{bmatrix}\begin{bmatrix}-\mathbf{J}_a^{-1}\mathbf{J}_b \\ \mathbf{I}\end{bmatrix}\dot{\mathbf{q}}_b, \tag{6}$$

then the independent joint velocity will be given by

$$\dot{\mathbf{q}}_b = \alpha\left(\frac{\partial H'}{\partial \mathbf{q}_b}\right)^T = \alpha\begin{bmatrix}-(\mathbf{J}_a^{-1}\mathbf{J}_b)^T & \mathbf{I}\end{bmatrix}\nabla_{\mathbf{q}}H. \tag{7}$$

Setting $\mathbf{J}_R \triangleq \mathbf{J}_a^{-1}\mathbf{J}_b$, the overall joint-space solution is

$$\begin{bmatrix}\dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_b\end{bmatrix} = \begin{bmatrix}\mathbf{J}_a^{-1} \\ \mathbf{0}\end{bmatrix}\dot{\mathbf{p}} + \alpha\begin{bmatrix}\mathbf{J}_R\mathbf{J}_R^T & -\mathbf{J}_R \\ -\mathbf{J}_R^T & \mathbf{I}\end{bmatrix}\nabla_{\mathbf{q}}H. \tag{8}$$

A discretized version of (8) was introduced in [10] as the *Reduced Gradient* (RG) method for solving the inverse kinematic problem of redundant manipulators. Though the projection and the reduction approaches follow a similar philosophy, the PG and the RG methods are different, in that *distinct* joint motions are generated along the same end-effector path. However, it can be shown that the same joint velocity direction is generated when a self-motion ($\dot{\mathbf{p}} = 0$) is performed in the particular case $n - m = 1$, although the RG naturally produces a larger stepsize [10].

The superiority of the RG method is twofold. First, the computational complexity in (8) is less than in (3). The key point is that the RG method needs only to invert an $m \times m$ minor of $\mathbf{J}$ — which is directly available — as opposed to the PG method that requires the inversion of the matrix product $\mathbf{J}\mathbf{J}^T$, which may result in much more involved expressions. Second, convergence of the RG method is known to be faster on the average [11], as numerical evidence has shown. These considerations are particularly relevant when on-line implementation of redundancy resolution schemes is of concern.

REDUNDANCY RESOLUTION VIA TASK-AUGMENTATION

Another major framework for solving redundancy is set up through the consideration of an extended task, obtained augmenting the primary relation (1) with additional constraints. More in general, the performance requirements imposed on the manipulator motion can be organized in r possibly time-varying subtasks with a decreasing order of priority. The ordered tasks can be expressed as

$$\mathbf{p}_i(t) = \mathbf{f}_i(\mathbf{q}(t)), \quad i = 1, \ldots, r, \quad \mathbf{p}_i \in I\!R^{m_i}, \tag{9}$$

where $\mathbf{p}_i(t)$ is the desired time evolution for the $i-$th subtask. Defining $\mathbf{J}_i(\mathbf{q}) = \partial \mathbf{f}_i / \partial \mathbf{q}_i$ and considering for simplicity the case of two tasks, the *Task-Priority* (TP) method [6] gives a joint-velocity

$$\dot{\mathbf{q}} = \mathbf{J}_1^\dagger \dot{\mathbf{p}}_1 + \tilde{\mathbf{J}}_2^\dagger (\dot{\mathbf{p}}_2 - \mathbf{J}_2 \mathbf{J}_1^\dagger \dot{\mathbf{p}}_1) + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)(\mathbf{I} - \tilde{\mathbf{J}}_2^\dagger \tilde{\mathbf{J}}_2)\mathbf{w}, \tag{10}$$

with $\tilde{\mathbf{J}}_2 \triangleq \mathbf{J}_2(\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)$, and $\mathbf{w}$ an arbitrary $n-$vector. Assuming full row rank for $\mathbf{J}_1$, this solution generates joint motions which realize the high-priority task and satisfy the low-priority one in the least-squares sense. In particular, if $m_1 + m_2 \leq n$, both tasks are executed exactly except when *algorithmic singularities* [12] are encountered. In that case $\mathcal{N}(\mathbf{J}_1(\mathbf{q})) \cup \mathcal{N}(\mathbf{J}_2(\mathbf{q})) \neq I\!R^n$, and while the second task is relaxed, the first one is still preserved. Moreover, if $m_1 + m_2 < n$ holds strictly, vector $\mathbf{w}$ provides space for further redundancy utilization, while in any other case the third term in (10) vanishes. As a result, the TP method complies automatically with degeneracies and lack of sufficient degrees of freedom.

However, even under the most convenient hypotheses, the limitation of the above scheme is still in the computational load, namely in the extensive use of pseudoinverses. Conversely, the joint-space decomposition approach is helpful in reducing the complexity and in organizing efficiently the single algorithmic steps.

Consider as above the case of two tasks with dimension m_1 and m_2. For simplicity, suppose also that each task can be always executed by itself, and that the robot arm is redundant for the high-priority task, i.e. rank $\big(\mathbf{J}_1(\mathbf{q})\big) = m_1 < n$ and rank $\big(\mathbf{J}_2(\mathbf{q})\big) = m_2 \leq n$. In analogy with (5), satisfaction of the first task is guaranteed by

$$\dot{\mathbf{q}}_a = \mathbf{J}_{1a}^{-1} \dot{\mathbf{p}} - (\mathbf{J}_{1a}^{-1} \mathbf{J}_{1b}) \dot{\mathbf{q}}_b, \tag{11}$$

in which $\mathbf{J}_{1a}$ has full rank m_1, and $\mathbf{J}_1 = [\mathbf{J}_{1a}\, \mathbf{J}_{1b}]$ after proper reordering of variables. Substituting (11) into the low-priority task yields

$$\dot{\mathbf{p}}_2 = \mathbf{J}_{2a} \dot{\mathbf{q}}_a + \mathbf{J}_{2b} \dot{\mathbf{q}}_b = \mathbf{J}_{2a} \mathbf{J}_{1a}^{-1} \dot{\mathbf{p}}_1 + (\mathbf{J}_{2b} - \mathbf{J}_{2a} \mathbf{J}_{1a}^{-1} \mathbf{J}_{1b}) \dot{\mathbf{q}}_b, \tag{12}$$

where $\mathbf{J}_2 = [\mathbf{J}_{2a}\, \mathbf{J}_{2b}]$ is the partition inherited from $\mathbf{J}_1$ ($\mathbf{J}_{2a}$ has dimension $m_2 \times m_1$). Equation (12) can be rewritten in a compact form as

$$\dot{\hat{\mathbf{p}}}_2 \triangleq \dot{\mathbf{p}}_2 - \mathbf{J}_{2a} \mathbf{J}_{1a}^{-1} \dot{\mathbf{p}}_1 = (\mathbf{J}_{2b} - \mathbf{J}_{2a} \mathbf{J}_{1a}^{-1} \mathbf{J}_{1b}) \dot{\mathbf{q}}_b \triangleq \hat{\mathbf{J}}_2 \dot{\mathbf{q}}_b. \tag{13}$$

Three cases may be considered at this point.

(*i*) $m_2 \le n - m_1$ and $\mathcal{N}(\mathbf{J}_1) \cup \mathcal{N}(\mathbf{J}_2) = \mathbb{R}^n$. Vector $\mathbf{q}_b$ is in turn decomposed into two blocks of dimensions m_2 and $n-(m_1+m_2)$ respectively, which are denoted $(\mathbf{q}_{ba}, \mathbf{q}_{bb})$, with a slight abuse of notation. Correspondingly, $\hat{\mathbf{J}}_2$ is partitioned as $[\hat{\mathbf{J}}_{2a}\hat{\mathbf{J}}_{2b}]$, so that

$$\dot{\mathbf{q}}_{ba} = \hat{\mathbf{J}}_{2a}^{-1}\dot{\hat{\mathbf{p}}}_2 - \hat{\mathbf{J}}_{2a}^{-1}\hat{\mathbf{J}}_{2b}\dot{\mathbf{q}}_{bb}, \tag{14}$$

in which $\hat{\mathbf{J}}_{2a}$ has full rank m_2. Such a choice is always possible in this case, although it may require a careful selection of the basic variables in (11). The second task is exactly satisfied by (14). When m_1+m_2 is strictly less than n, $\dot{\mathbf{q}}_{bb}$ is still available for further use (e.g. for optimization by a *doubly* reduced gradient method), while otherwise $\hat{\mathbf{J}}_{2b}$ vanishes.

(*ii*) $m_2 \le n - m_1$ but $\mathcal{N}(\mathbf{J}_1) \cup \mathcal{N}(\mathbf{J}_2) \ne \mathbb{R}^n$. The inversion in (14) is no more allowed, revealing that the second task cannot be fully achieved. A first possible strategy is to use a minimum-norm error solution of (13), yielding $\dot{\mathbf{q}}_b = \hat{\mathbf{J}}_2^{\dagger}\dot{\hat{\mathbf{p}}}_2$. Comparing this and (11) with the TP method (10) still shows some numerical advantage, since pseudoinversion is used only once, and on a $m_2 \times (n-m_1)$ matrix. A second alternative naturally follows from the decomposition approach. The low-priority task is further subdivided, extracting one *largest realizable part*, of dimension $\overline{m}_2 < m_2$. This is characterized by the property $\mathcal{N}(\mathbf{J}_1) \cup \mathcal{N}(\overline{\mathbf{J}}_2) = \mathbb{R}^n$, where $\overline{\mathbf{J}}_2$ is a (non unique) selection of $\overline{m}_2$ rows from the Jacobian $\mathbf{J}_2$. The redundancy resolution is then completed using a relation similar to (14).

(*iii*) $m_2 > n - m_1$. Beside directly pseudoinverting the overspecified relation (13), this case can be reduced to one of the previous. In fact, it may be reasonable to consider, within the low-priority assignments, a set of subtasks of higher value (at most $n-m_1$) to be executed. Case (*i*) or (*ii*) is recovered, depending on the presence of algorithmic singularities. In particular, suppose $n-m_1$ rows of $\mathbf{J}_2$ are extracted and no degeneracy occurs. Then the velocity vector $\dot{\mathbf{q}}_b$ is computed as

$$\dot{\mathbf{q}}_b = \hat{\mathbf{J}}_{21}^{-1}\dot{\hat{\mathbf{p}}}_{21}, \qquad \text{with} \quad \hat{\mathbf{p}}_2 = \begin{bmatrix} \hat{\mathbf{p}}_{21} \\ \hat{\mathbf{p}}_{22} \end{bmatrix}, \; \hat{\mathbf{J}}_2 = \begin{bmatrix} \hat{\mathbf{J}}_{21} \\ \hat{\mathbf{J}}_{22} \end{bmatrix}. \tag{15}$$

with obvious notation. This choice produces one left-inverse solution [8] for the original low-priority task.

It should be pointed out that the decomposition approach works in parallel both at the task and at the joint level, so to generate more convenient solutions. When compared with the pseudoinversion-based TP method (10), the presented method (e.g. (11) and (14), or (11) and (15)) yields different joint motions, except for the *square singularity-free* case; in this situation, the joint-decomposition approach obviously gives the same solution, being just a different taxonomy of computational steps.

ANALYTIC EXAMPLES

Example 1. Consider the PPR planar robot arm of Fig.1. When only the position of the end-effector is of interest, $m=2$ and the direct kinematics is $p_x = q_1 + lc_3$, $p_y = q_2 + ls_3$, with the shorthand notation $s_i = \sin q_i, c_i = \cos q_i$, l being the length of the third link. The Jacobian is always of full rank

$$\mathrm{J}(\mathrm{q}) = \begin{bmatrix} 1 & 0 & -ls_3 \\ 0 & 1 & lc_3 \end{bmatrix}, \tag{16}$$

and therefore the manipulability index [1] of this robot is constant. However, simple calculations show that the *dynamic* manipulability [13] (i.e. weighted with the inertia matrix $\mathbf{B}(\mathbf{q})$) takes on the form $H(\mathbf{q}) = \det\big(\mathbf{J}(\mathbf{q})\mathbf{B}(\mathbf{q})\mathbf{J}^T(\mathbf{q})\big) = a + b\cos^2 q_3$, with a and b positive constants; therefore, it can be used as a performance criterion to be maximized. When using the PG method (3), the following expressions are obtained for the pseudoinverse of $\mathbf{J}$ and for the null-space projection matrix $\mathbf{P}$

$$\mathbf{J}^{\dagger} = \frac{1}{1+l^2}\begin{bmatrix} 1+l^2{c_3}^2 & l^2 s_3 c_3 \\ l^2 s_3 c_3 & 1+l^2{s_3}^2 \\ -l s_3 & l c_3 \end{bmatrix}, \mathbf{P} = \frac{1}{1+l^2}\begin{bmatrix} l^2{s_3}^2 & -l^2 s_3 c_3 & l s_3 \\ -l^2 s_3 c_3 & l^2{c_3}^2 & -l c_3 \\ l s_3 & -l c_3 & 1 \end{bmatrix}. \quad (17)$$

In order to apply the RG method, a decomposition of the joint vector $\mathbf{q} = (\mathbf{q}_a, \mathbf{q}_b)$ is needed, which in this case is conveniently chosen as $\mathbf{q}_a = (q_1, q_2)$, $\mathbf{q}_b = q_3$, leading to the following partition of $\mathbf{J} = (\mathbf{J}_a, \mathbf{J}_b)$

$$\mathbf{J}_a = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{J}_b = \begin{bmatrix} -l s_3 \\ l c_3 \end{bmatrix}. \quad (18)$$

Therefore, the matrix appearing in the RG solution (8) is simply $\mathbf{J}_R^T = (\mathbf{J}_a^{-1}\mathbf{J}_b)^T = [-l s_3 \quad l c_3]$. It is immediate to see that the amount of computation needed by the RG method is very limited. The joint velocity is obtained as

$$\dot{\mathbf{q}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}\dot{\mathbf{p}} + \begin{bmatrix} l s_3 \\ -l c_3 \\ 1 \end{bmatrix}[l s_3 \quad -l c_3 \quad 1]\,\nabla_{\mathbf{q}} H. \quad (19)$$

Note that, except for the scaling term $1/(1+l^2)$, the outer product in the rhs of (19) gives the rank-one projection matrix $\mathbf{P}$, directly in the most convenient factorized form. Nonetheless, when performing end-effector motion the RG and the PG methods produce different joint trajectories. In fact, after some manipulation

$$\dot{\mathbf{q}}_{RG} = \begin{bmatrix} \dot{p}_x - 2bl s_3^2 c_3 \\ \dot{p}_y + 2bl c_3^2 s_3 \\ -2b s_3 c_3 \end{bmatrix} \neq \frac{1}{1+l^2}\begin{bmatrix} (1+l^2c_3^2)\dot{p}_x + l^2 s_3 c_3 \dot{p}_y - 2bl s_3^2 c_3 \\ l^2 s_3 c_3 \dot{p}_x + (1+l^2 s_3^2)\dot{p}_y + 2bl c_3^2 s_3 \\ -l s_3 \dot{p}_x + l c_3 \dot{p}_y - 2b s_3 c_3 \end{bmatrix} = \dot{\mathbf{q}}_{PG}, \quad (20)$$

which also shows the more natural solution provided by the RG method.

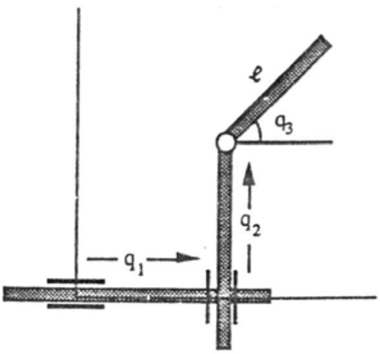

Fig. 1 – PPR robot arm

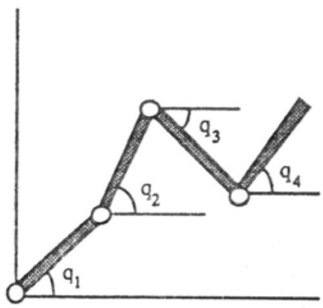

Fig. 2 – 4R robot arm

Example 2. Consider the 4R planar robot arm of Fig.2, having all links of unit length. Assume that the task is ordered in two levels of priority. The high priority subtask is specified as a trajectory for the end-effector, so that

$$\dot{\mathbf{p}}_1 = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} -s_1 & -s_2 & -s_3 & -s_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix} \dot{\mathbf{q}} = \mathbf{J}_1(\mathbf{q})\dot{\mathbf{q}}, \qquad (21)$$

while the low-priority task assigns a desired time-varying arm posture $\mathbf{p}_2(t) = \mathbf{q}(t)$. Thus, $m_1 = 2$ and $m_2 = 4$. Using the TP method, since $\mathbf{J}_2 = \mathbf{I}$, (10) collapses to:

$$\dot{\mathbf{q}}_{TP} = \mathbf{J}_1^{\dagger}\dot{\mathbf{p}}_1 + (\mathbf{I} - \mathbf{J}_1^{\dagger}\mathbf{J}_1)\dot{\mathbf{p}}_2. \qquad (22)$$

Under the full rank hypothesis for $\mathbf{J}_1$, the required pseudoinverse is

$$\mathbf{J}_1^{\dagger} = \frac{1}{\Delta} \begin{bmatrix} c_2 s_{2-1} + c_3 s_{3-1} + c_4 s_{4-1} & s_2 s_{2-1} + s_3 s_{3-1} + s_4 s_{4-1} \\ c_1 s_{1-2} + c_3 s_{3-2} + c_4 s_{4-2} & s_1 s_{1-2} + s_3 s_{3-2} + s_4 s_{4-2} \\ c_1 s_{1-3} + c_2 s_{2-3} + c_4 s_{4-3} & s_1 s_{1-3} + s_2 s_{2-3} + s_4 s_{4-3} \\ c_1 s_{1-4} + c_2 s_{2-4} + c_3 s_{3-4} & s_1 s_{1-4} + s_2 s_{2-4} + s_3 s_{3-4} \end{bmatrix}, \qquad (23)$$

where $\Delta = \det(\mathbf{J}_1\mathbf{J}_1^T) = \sum_{i=2}^{4}\sum_{j=1}^{i-1} s_{i-j}^2$, and $s_{i-j} = \sin(q_i - q_j)$. With this method, the second task will only be accomplished with a minimum-norm error distributed on all joints.

For the decomposition approach, case (*iii*) applies and the joint variables can be partitioned in two sets of dimension $m_1 = 2$ and $n - m_1 = 2$. There are cases, like for a macro-micro manipulator structure [14], when an inverse solution which has a tighter command on the last dof's is of particular interest. This is naturally accomplished with a joint-space decomposition. Suppose that $\mathbf{q}_a = (q_1, q_2)$ and $\mathbf{q}_b = (q_3, q_4)$ yield a nonsingular $\mathbf{J}_a$ block — the first two columns of $\mathbf{J}_1$ in (21). The velocity for the basic joints is then

$$\dot{\mathbf{q}}_a = \mathbf{J}_a^{-1}(\dot{\mathbf{p}}_1 - \mathbf{J}_b\dot{\mathbf{q}}_b) = \frac{1}{s_{2-1}} \begin{bmatrix} c_2 & s_2 \\ -c_1 & -s_1 \end{bmatrix} \left(\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix} - \begin{bmatrix} -s_3 & -s_4 \\ c_3 & c_4 \end{bmatrix} \dot{\mathbf{q}}_b \right). \qquad (24)$$

The choice of the independent joint velocity $\dot{\mathbf{q}}_b$ can be made in terms of the secondary task specification, as reduced to these variables. It can be shown that, deriving $\hat{\mathbf{J}}_2$ as in (13), a 2×2 identity minor appears which can be chosen as $\hat{\mathbf{J}}_{21}$. Use of (15) specifies $\dot{\mathbf{q}}_b(t) = \dot{\hat{\mathbf{p}}}_{21}(t) = \dot{\mathbf{p}}_{21}(t)$, i.e. directly the desired time evolution of the $\mathbf{q}_b = (q_3, q_4)$ variables. On the other hand, the evolution $\dot{\mathbf{q}}_a(t)$ of the basic variables given by (24) will not match the desired one $\dot{\mathbf{p}}_{22}(t)$, as specified by the originary second task.

CONCLUSIONS

The above analytic examples illustrate clearly the advantages and the simplicity of the joint-space decomposition approach. Numerical simulations reported in [10] give also evidence of the faster convergence of the Reduced Gradient method for maximizing available joint range, distance from obstacles or manipulability. The improvement over existing techniques is particularly significant for highly redundant systems.

Some remarks are in order concerning the choice of the basic set of joint variables. At each step the method needs to extract from a full row rank Jacobian a square nonsingular minor to be inverted (like $\mathbf{J}_a$ in (5), or $\hat{\mathbf{J}}_{2a}$ in (14)). Indeed, to this aim the associated partition of $\mathbf{q}$ may be forced to change over time (*). This event represents

(*) A change of basis may also be desirable, even if not necessary, so as to yield the best conditioned minor.

a possible drawback, since discontinuous joint velocities will be produced in general. However, there are various approaches to this problem, e.g. by smoothing joint-space "corners" or by rescaling the time-profile along the end-effector path so to stop at such points. A more rigorous solution requires reformulating the whole kinematic problem at the acceleration level, where discontinuities are feasible. In this spirit, joint-space decomposition has been effectively extended to the second-orded level in [15], where torque optimization is considered for resolving redundancy.

REFERENCES

[1] T. Yoshikawa, "Manipulability of robotics mechanisms", *Int. J. Robotics Res.*, vol. 4, no. 2, pp. 3–9, 1985.

[2] A.A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments", *Int. J. Robotics Res.*, vol. 4, no. 3, pp. 109–117, 1985.

[3] J. Baillieul, "Avoiding obstacles and resolving kinematic redundancy", in *Proc. IEEE Int. Conf. on Robotics and Automation* (San Francisco, CA, Apr. 7–10, 1986), pp. 1698–1704.

[4] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms", *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-7, no. 12, pp. 868–871, 1977.

[5] S.L. Chiu, "Task compatibility of manipulators postures", *Int. J. Robotics Res.*, vol. 7, no. 5, pp. 13–21, 1988.

[6] Y. Nakamura, H. Hanafusa, and T. Yoshikawa, "Task-priority based redundancy control of robot manipulators", *Int. J. Robotics Res.*, vol. 6, no. 2, pp. 3–15, 1987.

[7] P. Chiacchio and B. Siciliano, "A closed-loop Jacobian transpose scheme for solving the inverse kinematics of nonredundant and redundant wrists", *J. Robotic Syst.*, vol. 6, no. 5, pp. 601–630, 1989.

[8] T.L. Boullion and P.L. Odell, *Generalized Inverse Matrices*. New York, NY: Wiley Interscience, 1971.

[9] C.W. Wampler, "Inverse kinematic functions for redundant manipulators", in *Proc. IEEE Int. Conf. on Robotics and Automation*, (Raleigh, NC, Mar. 31– Apr. 3, 1987) pp. 610–617.

[10] A. De Luca and G. Oriolo, "The reduced gradient method for solving redundancy in robot arms", *11th IFAC World Congress* (Tallinn, USSR, Aug. 13–17, 1990).

[11] D.G. Luenberger, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1984.

[12] J. Baillieul, "Kinematic programming alternatives for redundant manipulators", in *Proc. IEEE Int. Conf. on Robotics and Automation*, (St. Louis, MO, Mar. 25–28, 1985), pp. 722–728.

[13] T. Yoshikawa, "Dynamic manipulability of robot manipulators", *J. Robotic Syst.*, vol. 2, no. 1, pp. 113–124, 1985.

[14] O. Egeland, "Task-space tracking with redundant manipulators", *IEEE J. Robotics and Automation*, vol. RA-3, no. 5, pp. 471–475, 1987.

[15] A. De Luca and G. Oriolo, "Efficient dynamic resolution of robot redundancy", *American Control Conference*, (San Diego, CA, May 23–25, 1990).

This work was partially supported by CNR *Progetto Finalizzato Robotica*, Linea 4.3.

ISBN 83-900501-2-9

WARSAW UNIVERSITY OF TECHNOLOGY PUBLICATIONS
00-665 WARSAW, NOWOWIEJSKA 24