Robotersysteme 7, 117–122 (1991)

Robotersysteme

The reduced gradient method for solving redundancy in robot arms *

A. De Luca and **G. Oriolo,** Roma

Abstract. An efficient computational scheme for solving inverse kinematic problems in redundant robot arms is presented. When following a given end-effector trajectory, successive internal arm configurations are in general selected by local optimization of a given performance criterion. Typically, joint displacements are derived using the Projected Gradient method, involving pseudoinversion of the robot Jacobian and projection in its null-space. However, this technique is computationally intensive. In this paper, an alternative approach is proposed based on the Reduced Gradient method, which allows to deal explicitly only with the redundant degrees of freedom. The superiority of this technique for solving redundancy is illustrated analytically in a simple case, and numerically by simulation of a four-link planar arm. Optimization of various criteria like manipulability, available joint range, and distance from obstacles is considered. Extensions of the method are also briefly discussed.

Eine reduzierte Gradientenmethode zur Lösung des inversen kinematischen Problems redundanter Roboterarme

Zusammenfassung. In diesem Beitrag wird ein effizientes Rechenschema zur Lösung des inversen kinematischen Problems vorgestellt. Für eine gegebene Endeffektor-Trajektorie werden sukzessive interne Armkonfigurationen durch lokale Optimierung eines gegebenen Gütekriteriums ausgewählt. Normalerweise werden Gelenkwinkelbewegungen durch Gradientenmethoden abgeleitet unter Verwendung der pseudoinversen Jacobi Matrix. Diese Technik ist allerdings sehr rechenintensiv. Ein alternativer Ansatz wird hierzu vorgeschlagen, der auf einer reduzierten Gradientenmethode berührt. Diese Methode erlaubt die Nutzung von ausschließlich redundanten Freiheitsgraden. Die Überlegenheit dieser Technik wird analytisch für einen einfachen Roboter und numerisch für einen planaren 4 Gelenkarm demonstriert. Optimalkriterien wie Handhabbarkeit, verbleibender Gelenkraum sowie Abstände zu Hindernissen werden berücksichtigt. Erweiterungen dieser Methode werden kurz erläutert.

1 Introduction

The complexity of tasks that advanced manipulators are required to execute in a real-life environment asks for robot arms with increased, human-like dexterity. Robots should be able to move safely in a crowded workspace, with objects to be manipulated and obstacles to be avoided. Typical industrial examples of such situations include spray painting of the inside of car bodies and assembly tasks with multiple exchange of forces/torques with the environment.

To acquire dextrous capabilities, the arm structure should possess a number of degrees of freedom larger than the one strictly needed for the general positioning, i.e. should be redundant. In this case, the inversion of the direct kinematic relation

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

between joint coordinates $q \in \mathbb{R}^N$ and task coordinates $p \in \mathbb{R}^M$, with $N > M$, yields an infinity of solutions. At a differential level, one has

$$dp = \mathbf{J}\,(q)\,dq\,, \tag{2}$$

with the Jacobian $\mathbf{J}$ being a non-square matrix.

In order to make the redundancy of the mechanical design really appealing, the robot controller should be programmed so to exploit as much as possible these extra degrees of freedom. The choice among the infinite number of joint trajectories realizing a given path of the end-effector can be done by optimizing a suitable criterion. Indeed, it is desirable to achieve a skilled robot performance with the minimum amount of additional complexity.

Kinematic redundancy is mainly used to enhance the functional workspace of the robot. In redundant arms, the crossing of kinematic singularities can be avoided through internal reconfiguration of the arm along the same given end-effector motion. This may be obtained by maximizing a criterion which measures the arm manipulability [1, 2]. Optimization schemes are useful also when obstacles are present in the robot workspace. While the end-effector trajectory is already planned so to avoid collision, redundancy can be used to guarantee a collision-free path for the whole arm, by means of various distance criteria [3–5]. There are also forbidden regions directly specified in the joint space: optimal avoidance of joint range limits has been considered by Liègeois (1977).

All the above techniques are based on the pseudoinverse of the Jacobian matrix, used to satisfy the end-effector dis-

* A version of this paper was presented at the 11th IFAC World Congress, Tallinn, USSR, August 1990

placement, combined with the specification of a null-space vector according to the chosen objective. Liègeois [6] recognized this scheme to be the transposition of the Projected Gradient (PG) method to the robot inverse kinematic problem. The major flaw of this optimization algorithm is that it requires a considerable amount of computations. This limit was pointed out by several authors who provided various improvements. In particular, Klein and Huang [7] suggested the use of a Gaussian elimination technique within this approach; Khalil and Chevallereau [8] proposed a different way for computing the pseudoinverse, while Dubey et al. [9] implemented a faster numerical scheme for a seven-dof robot with a spherical wrist.

An alternative and more efficient approach is presented in this paper, based on the Reduced Gradient (RG) method for nonlinear constrained optimization. The resulting scheme solves redundancy in a natural way. In fact, only the extra degrees of freedom are used for optimization, while the remaining joint variables are in charge of the task satisfaction. This allows for large computational savings. In the following, the redundancy resolution problem will be formally revisited as an optimization problem. Comparison between the RG and PG methods will be carried out both analytically and numerically, showing the inherent benefits of the proposed approach.

2 Optimal redundancy resolution

For a redundant robot arm the optimal choice of joint configurations can be defined by the following nonlinear optimization problem:

$$\max_{\boldsymbol{q}} H(\boldsymbol{q}), \qquad \text{s.t.} \qquad \boldsymbol{f}(\boldsymbol{q}) - \boldsymbol{p} = \mathbf{0}. \tag{3}$$

$H(\boldsymbol{q})$ is a generic objective function, while the vector constraint in (3) follows from the direct kinematics of the arm. Typically, this optimization problem cannot be solved in a closed-form and an iterative process has to be devised. Starting from an initial point $\boldsymbol{q}^0$ in the joint space, an update is defined as:

$$\boldsymbol{q}^{k+1} = \boldsymbol{q}^k + \boldsymbol{dq}^k. \tag{4}$$

If $\boldsymbol{q}^0$ is the actual arm configuration at some initial instant, the iterates (4) generated by the chosen optimization algorithm can be taken as reference values to be tracked by a closed-loop robot controller.

When $\boldsymbol{p}$ is constant, the solution (3) is the optimal posture $\boldsymbol{q}^*$ for the specified end-effector location. The optimization process will produce a sequence of intermediate joint configurations providing a *self-motion* of the arm. This is true as long as the constraint is never violated.

If a time-varying $\boldsymbol{p}$ is specified, i.e. a trajectory for the end-effector, (3) is transformed into a succession of subproblems obtained by discretization of the end-effector path into a sequence $\{\boldsymbol{p}^k\}$. Letting $\boldsymbol{dp}^k = \boldsymbol{p}^{k+1} - \boldsymbol{p}^k$, the k-th subproblem becomes of the form:

$$\max_{\boldsymbol{dq}} H(\boldsymbol{q}^k + \boldsymbol{dq}), \qquad \text{s.t.} \qquad \boldsymbol{f}(\boldsymbol{q}^k + \boldsymbol{dq}) - (\boldsymbol{p}^k + \boldsymbol{dp}^k) = \mathbf{0}. \tag{5}$$

Any optimization algorithm can be used for solving (5). In general, some iterations will be needed to find an *exact* solution to each subproblem, so that use of this scheme in real-time is possible only if the algorithm is extremely fast. Therefore, $\boldsymbol{dq}^k$ in (4) is usually chosen as the first iterate computed toward the optimum of (5).

One way of determining $\boldsymbol{dq}^k$ is given by the Projected Gradient method [10], namely by projecting the gradient of $H(\boldsymbol{q})$ onto the tangent space of the constraint. Since $\mathbb{R}^N = \mathcal{N}(\mathbf{J}) \oplus \mathcal{R}(\mathbf{J}^{\mathrm{T}})$ holds, the displacement $\boldsymbol{dq}^k$ is obtained as the sum of two vectors belonging to these complementary subspaces. The projection matrix in the null-space of the Jacobian is $\mathbf{P} \triangleq \mathbf{I} - \mathbf{J}^\dagger \mathbf{J}$, where $\mathbf{J}^\dagger$ is the unique pseudoinverse of $\mathbf{J}$ (Boullion and Odell, 1971). Hence, the PG update results in

$$\boldsymbol{dq}^k = \mathbf{J}^\dagger \boldsymbol{dp}^k + [\mathbf{I} - \mathbf{J}^\dagger \mathbf{J}]\, \nabla_q H(\boldsymbol{q}^k), \tag{6}$$

where $\nabla_q H \triangleq (\partial H / \partial \boldsymbol{q})^{\mathrm{T}}$ and all matrices are evaluated at $\boldsymbol{q}^k$.

The computations involved in (6) are quite cumbersome for redundant robots with several degrees of freedom. In fact, the pseudoinverse of $\mathbf{J}$ is obtained in general via a singular value decomposition technique. When the Jacobian has full row rank, $\mathbf{J}^\dagger$ can be evaluated more directly as $\mathbf{J}^\dagger = \mathbf{J}^{\mathrm{T}}(\mathbf{J}\mathbf{J}^{\mathrm{T}})^{-1}$. However, this still requires a product of matrices and a $M \times M$ matrix inversion. Moreover, the $N \times N$ projection matrix $\mathbf{P}$ has rank equal to $(N-M)$ – the rank deficiency being a considerable waste of information.

3 The reduced gradient method

An alternative method for solving (5) is based on the observation that the actual number of free variables in the problem is only $N-M$. Therefore, the search for an optimal displacement $\boldsymbol{dq}^k$ can be more efficiently performed within a reduced space of joint variables. This leads to the Reduced Gradient method, for which a preliminary assumption is needed.

Nondegeneracy Assumption. At every specified $\boldsymbol{p}$, there exists a $\boldsymbol{q} \in \mathbb{R}^N$ such that $\boldsymbol{f}(\boldsymbol{q}) = \boldsymbol{p}$, for which a partition can be found

$$\boldsymbol{q} = (\boldsymbol{q}_a, \boldsymbol{q}_b), \qquad \boldsymbol{q}_a \in \mathbb{R}^M, \qquad \boldsymbol{q}_b \in \mathbb{R}^{N-M},$$

yielding a nonsingular matrix

$$\nabla_{q_a} \boldsymbol{f}(\boldsymbol{q}) = \mathbf{J}_a^{\mathrm{T}}(\boldsymbol{q}).$$

This assumption restricts the admissible end-effector poses, discarding those for which the Jacobian *necessarily* looses full row rank. According to the partition of $\boldsymbol{q}$, the Jacobian matrix is decomposed into two blocks $(\mathbf{J}_a, \mathbf{J}_b)$, so that the differential relation (2) can be rewritten as

$$\boldsymbol{dp} = \mathbf{J}_a(\boldsymbol{q})\, \boldsymbol{dq}_a + \mathbf{J}_b(\boldsymbol{q})\, \boldsymbol{dq}_b. \tag{7}$$

The Implicit Function Theorem is invoked to locally explicit the constraint $f(q_a, q_b) - p = 0$. This provides an expression for the *basic* joint variables q_a in terms of the *independent* variables q_b and of the known end-effector pose p:

$$q_a = g(q_b, p). \tag{8}$$

From the same Theorem, an expression is found for the derivative of (8) with respect to q_b:

$$\frac{\partial g}{\partial q_b} = -\mathbf{J}_a^{-1}\mathbf{J}_b. \tag{9}$$

The reduced gradient of the objective function $H(q)$ with respect to q_b is defined as the gradient of the function $H'(q_b) \triangleq H(g(q_b, p), q_b)$, considered for a fixed p. From the chain rule

$$\nabla_{q_b} H'(q_b) = [-(\mathbf{J}_a^{-1}\mathbf{J}_b)^{\mathrm{T}} \quad \mathbf{I}]\, \nabla_q H. \tag{10}$$

At iteration k, for the independent joint variables q_b a step is taken in the reduced gradient direction:

$$dq_b^k = \nabla_{q_b} H'(q_b^k). \tag{11}$$

The displacement for the basic joint variables q_a is then chosen so to satisfy (7):

$$dq_a^k = \mathbf{J}_a^{-1}\, dp^k - (\mathbf{J}_a^{-1}\mathbf{J}_b)\, dq_b^k. \tag{12}$$

The overall update of the RG method can be written in a compact form as:

$$\begin{bmatrix} dq_a^k \\ dq_b^k \end{bmatrix} = \begin{bmatrix} \mathbf{J}_a^{-1} \\ \mathbf{0} \end{bmatrix} dp^k + \begin{bmatrix} \mathbf{J}_R \mathbf{J}_R^{\mathrm{T}} & -\mathbf{J}_R \\ -\mathbf{J}_R^{\mathrm{T}} & \mathbf{I} \end{bmatrix} \nabla_q H(q^k), \tag{13}$$

where $\mathbf{J}_R \triangleq \mathbf{J}_a^{-1}\mathbf{J}_b$. This should be compared for complexity with the update (6) of the PG method. It is worth noting that neither the pseudoinverse $\mathbf{J}^\dagger$ nor the projection matrix $\mathbf{P}$ are needed in (13). As a matter of fact, only one key matrix, i.e. $\mathbf{J}_R$, has to be determined. The computational burden is essentially limited to the inversion of the $M \times M$ matrix $\mathbf{J}_a$, which is directly available as part of the robot Jacobian.

It should be stressed that the PG and the RG methods do *not* provide in general the same updates in the overall space of joint configurations. The asymptotic rate of convergence is similar for the two methods, although for most problems the RG method will converge faster [10]. Moreover, each iteration of the RG is less time-consuming, so that its use in on-line implementations is more convenient.

The basic scheme outlined above can be further improved. In fact, for a given set of basic variables q_a, the inversion of the matrix $\mathbf{J}_a$ in (12) may become ill-conditioned, thus affecting the convergence rate of the algorithm. To overcome this problem, the whole set of joint variables can be explored at each iteration, and the basic variables q_a are chosen so to give the best conditioned $\mathbf{J}_a$ matrix, e.g. using the absolute value of the determinant as a measure.

Another limiting factor, which is common to all methods based on the local linearization of the constraints, is that the generated update is feasible only to the first order. Uncompensated numerical errors build up, resulting in a deviation from the desired end-effector path. The inclusion of a correction procedure considerably improves constraint satisfaction. Denote by $\tilde{q}^{k+1}$ the temporary update provided by (13) in (4). An additional move $\widetilde{dq}^k = (\widetilde{dq}_a^k, \mathbf{0})$ is performed in the direction orthogonal to the tangent plane at q^k

$$\widetilde{dq}_a^k = -\mathbf{J}_a^{-1}(q^k)\, \tilde{e}^{k+1}, \tag{14}$$

using the 'expected' error $\tilde{e}^{k+1} = f(\tilde{q}^{k+1}) - p^{k+1}$. In principle, this step should be repeated several times to restore feasibility, especially when the constraint curvature is large. However, very good results have been obtained in numerical tests using a single correction step. At iteration k, the complete update for q_a becomes

$$dq_a^k = \mathbf{J}_a^{-1}(dp^k - \tilde{e}^{k+1}) - (\mathbf{J}_a^{-1}\mathbf{J}_b)\, \mathbf{d}\mathbf{q}_b^k, \tag{15}$$

and $q^{k+1} = \tilde{q}^{k+1} + \widetilde{dq}^k$. This correction procedure makes the method inherently more robust, in much the same way as feeding back the cartesian error does in a closed-loop control scheme.

Finally, note that in (11) a unit stepsize was assumed along the reduced gradient direction. To improve convergence properties, a stepsize $\alpha \neq 1$ could be taken, either constant or generated by a line search.

4 Analytic results

In this section, a simple redundant robot will be used to illustrate how the RG method for solving redundancy leads to an efficient computational scheme. Moreover, it will be shown that the RG and PG methods generate the same search directions when used for optimizing a self-motion in robots with only one degree of redundancy.

Example 1 (3-R Robot). Consider a planar robot arm with three rotational joints, for which the degree of redundancy of the end-effector positioning task is one. For simplicity, all links are assumed of equal length l. Using as joint coordinates the absolute angles q_i referred to the x-axis, the direct kinematics becomes

$$p_x = l(c_1 + c_2 + c_3), \qquad p_y = l(s_1 + s_2 + s_3), \tag{16}$$

where the shorthand notation $s_i := \sin q_i$, $c_i := \cos q_i$ is used. The Jacobian matrix is

$$\mathbf{J}(q) = l \begin{bmatrix} -s_1 & -s_2 & -s_3 \\ c_1 & c_2 & c_3 \end{bmatrix}. \tag{17}$$

Note that a minor of this matrix, composed of columns i and j, is nonsingular if $s_{i-j} := \sin(q_i - q_j)$ is nonzero.

Assuming full rank for the Jacobian, the pseudoinverse $\mathbf{J}^\dagger$ and the null-space projection matrix $\mathbf{P}$ are computed as:

$$\mathbf{J}^\dagger = \frac{1}{l\Delta} \begin{bmatrix} c_2 s_{2-1} + c_3 s_{3-1} & s_2 s_{2-1} + s_3 s_{3-1} \\ c_1 s_{1-2} + c_3 s_{3-2} & s_1 s_{1-2} + s_3 s_{3-2} \\ c_1 s_{1-3} + c_2 s_{2-3} & s_1 s_{1-3} + s_2 s_{2-3} \end{bmatrix}$$

$$\mathbf{P} = \frac{1}{\Delta} \begin{bmatrix} s_{3-2}^2 & s_{3-1}\, s_{2-3} & s_{2-1}\, s_{3-2} \\ s_{3-1}\, s_{2-3} & s_{3-1}^2 & s_{2-1}\, s_{1-3} \\ s_{2-1}\, s_{3-2} & s_{2-1}\, s_{1-3} & s_{2-1}^2 \end{bmatrix}$$

where $\Delta = s_{2-1}^2 + s_{3-1}^2 + s_{3-2}^2$. These two matrices, used in (6), fully define the PG method.

To apply the RG method, any partition of the joint vector $\boldsymbol{q}$ yielding a nonsingular matrix $\mathbf{J}_a$ can be chosen. For instance, assuming $s_{2-1} \neq 0$, let $\boldsymbol{q}_a = (q_1, q_2)$ and $\boldsymbol{q}_b = q_3$, which results in the partition of $\mathbf{J}$

$$\mathbf{J}_a = l \begin{bmatrix} -s_1 & -s_2 \\ c_1 & c_2 \end{bmatrix}, \quad \mathbf{J}_b = l \begin{bmatrix} -s_3 \\ c_3 \end{bmatrix}. \tag{18}$$

Therefore, the matrix characterizing the RG method is

$$\mathbf{J}_R^{\mathrm{T}} = \frac{1}{s_{2-1}} [s_{2-3} \quad s_{3-1}]\,. \tag{19}$$

The reader may appreciate the simplicity of this derivation, which confirms the improvement obtained with the RG method. In this case, a step is computed as

$$\boldsymbol{dq}^k = \begin{bmatrix} \mathbf{J}_a^{-1} \\ 0 \end{bmatrix} \boldsymbol{dp}^k + \frac{1}{s_{2-1}^2} \begin{bmatrix} -s_{2-3} \\ -s_{3-1} \\ s_{2-1} \end{bmatrix} \cdot [-s_{2-3} \quad -s_{3-1} \quad s_{2-1}]\, \nabla_{\boldsymbol{q}} H(\boldsymbol{q}^k)\,.$$

Indeed, similar expressions are obtained (modulo an index permutation) when a different set of basic variables is chosen.

Example 2 (Self-motions of robots with $N - M = 1$). For the class of robot arms with one degree of redundancy, under the nondegeneracy assumption, the Jacobian can always be partitioned as $\mathbf{J} = (\mathbf{J}_a, \boldsymbol{j}_b)$, where $\mathbf{J}_a$ is an $(N-1) \times (N-1)$ nonsingular matrix and $\boldsymbol{j}_b$ is just a column. Suppose that a self-motion is performed to extremize an assigned criterion $H(\boldsymbol{q})$, at a given end-effector location $\bar{\boldsymbol{p}}$. In this case $\boldsymbol{dp} = \mathbf{0}$, and the only relevant matrix in the PG method is the matrix $\mathbf{P} = \mathbf{I} - \mathbf{J}^{\dagger} \mathbf{J}$. The following expression holds for the pseudoinverse of a matrix partitioned as $\mathbf{J}$ (Boullion and Odell, 1971)

$$\mathbf{J}^{\dagger} = \begin{bmatrix} \mathbf{J}_a^{-1} (\mathbf{I} - \beta\, \boldsymbol{j}_b \boldsymbol{j}_b^{\mathrm{T}} (\mathbf{J}_a \mathbf{J}_a^{\mathrm{T}})^{-1}) \\ \beta\, \boldsymbol{j}_b^{\mathrm{T}} (\mathbf{J}_a \mathbf{J}_a^{\mathrm{T}})^{-1} \end{bmatrix}, \tag{20}$$

where $\beta \triangleq 1/(1 + \boldsymbol{j}_b^{\mathrm{T}} (\mathbf{J}_a \mathbf{J}_a^{\mathrm{T}})^{-1} \boldsymbol{j}_b)$. After some manipulation, it can be shown that the projection matrix takes in this case the form

$$\mathbf{P} = \beta \begin{bmatrix} (\mathbf{J}_a^{-1} \boldsymbol{j}_b)(\mathbf{J}_a^{-1} \boldsymbol{j}_b)^{\mathrm{T}} & -\mathbf{J}_a^{-1} \boldsymbol{j}_b \\ -(\mathbf{J}_a^{-1} \boldsymbol{j}_b)^{\mathrm{T}} & 1 \end{bmatrix}. \tag{21}$$

Comparing (21) with the expression (13) of the RG method, as applied to self-motions, it is easy to see that

$$\boldsymbol{dq}_{\mathrm{PG}} = \beta \cdot \boldsymbol{dq}_{\mathrm{RG}}\,. \tag{22}$$

Therefore, the two methods provide the same direction in the joint space, up to the iteration-dependent scaling factor β. This was not unexpected, since for these robots the null-space of the Jacobian is one-dimensional. It is possible to show that this scaling factor is eqivalently rewritten as

$$\beta = \frac{\det^2 \mathbf{J}_a}{\det(\mathbf{J}\,\mathbf{J}^{\mathrm{T}})} \leq 1\,. \tag{23}$$

This general result applies indeed to the previous example.

5 Numerical results

The proposed redundancy resolution method has been used for a 4-R planar robot arm with all links of unit length (Fig. 1), executing both internal self-motions and end-effector trajectories. For positioning tasks, this robot has two degrees of redundancy and thus it is suitable to fully illustrate the different behavior of the RG and the PG methods. In terms of absolute joint coordinates, the Jacobian of this arm is

$$\mathbf{J}(\boldsymbol{q}) = \begin{bmatrix} -s_1 & -s_2 & -s_3 & -s_4 \\ c_1 & c_2 & c_3 & c_4 \end{bmatrix}. \tag{24}$$

Different objective functions $H(\boldsymbol{q})$ were tested for optimization. In order to maximize the joint range availability, the criterion proposed by Liègeois [6] was used. Since relative joint angles are $\theta_i = q_i - q_{i-1}$, for the 4-R arm this will be written as

$$H_1(\boldsymbol{q}) = -\frac{1}{2} \sum_{i=1}^{4} \left(\frac{(q_i - q_{i-1}) - \bar{\theta}_i}{\theta_{m,i} - \theta_{M,i}} \right)^2, \tag{25}$$

where $[\theta_{m,i}, \theta_{M,i}]$ is the admissible range for joint i and $\bar{\theta}_i$ is its center. In the first term of the summation, $q_0 = 0$. For obstacle avoidance, the case of a small disc centered at $(\bar{x}, 0)$ has been considered. In particular, $\bar{x}$ is such that only interference with the fourth link can occur. Properly modifying the criterion proposed by Baillieul (1986), it was chosen to maximize

$$H_2(\boldsymbol{q}) = \left| \bar{x}\, s_4 - \sum_{i=1}^{3} s_{4-i} \right|. \tag{26}$$

Finally, a test with a manipulability measure was carried out. For the 4-R arm an index which is equivalent to the

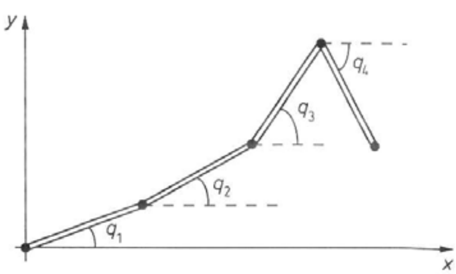

Fig. 1. A four-link planar robot arm

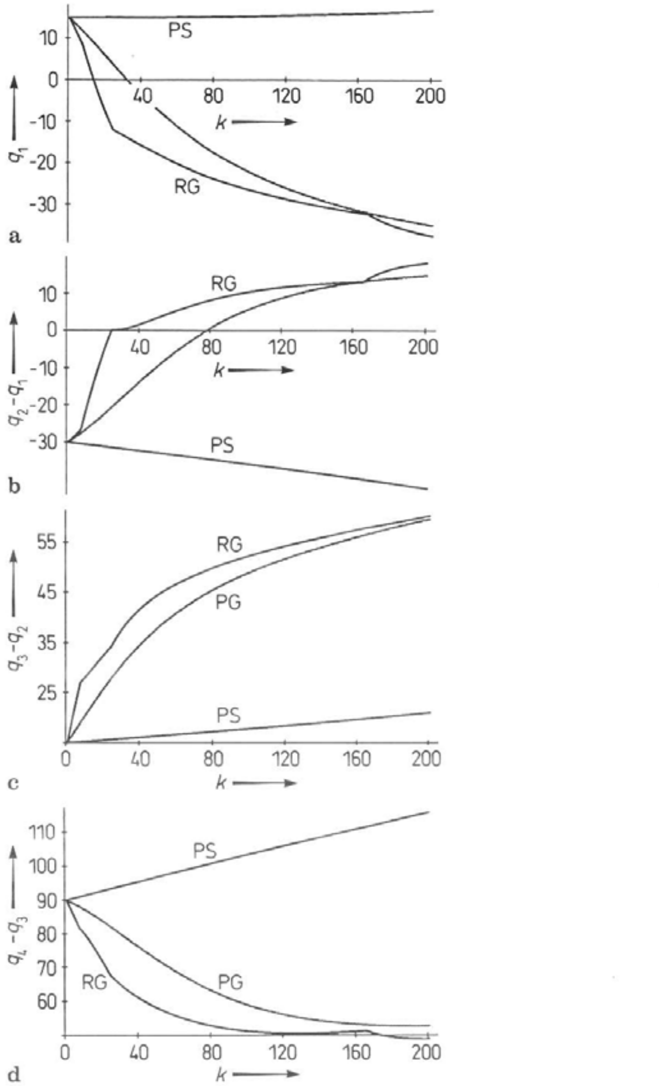

Fig. 2. **a** Avoiding joint limits. Joint 1, **b** avoiding joint limits. Joint 2, **c** avoiding joint limits. Joint 3, **d** avoiding joint limits. Joint 4

classical $\sqrt{\det \mathbf{J}\,\mathbf{J}^{\mathrm{T}}}$ of Yoshikawa [1] is

$$H_3(\boldsymbol{q}) = \sum_{i=1}^{3} s^2_{(i+1)-i}\,. \tag{27}$$

In the first numerical example the end-effector path is given as a straight line connecting point (2.93, 1) to (2.5, 0.7). The cartesian path is discretized with 200 samples and should be executed without violating the joint limits, which are set at $\pm 90^\circ$ of relative angle between successive links. At the starting point, the arm configuration is $\boldsymbol{q} = (15^\circ, -15^\circ, 0^\circ, 90^\circ)$, with the fourth joint at its upper limit. Figure 2a–d show the evolution of the four relative joint angles when using the PG method, the RG method or simple pseudoinversion (PS). For the first two, maximization of H_1 is performed. The PS solution, plotted for comparison, violates the fourth joint limit since it naturally produces the minimum norm displacement of the joints. Instead, both PG and RG methods remain feasible along the whole path, although the latter drives sooner the fourth joint away from its limit. Note that, in correspondence to changes of the set of basic variables, a sudden slope variation occurs in the RG joint path.

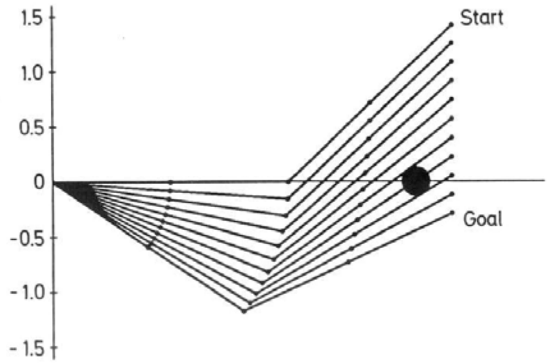

Fig. 3. Obstacle avoidance. PS solution

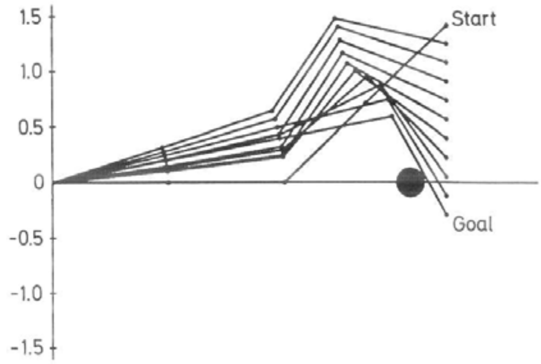

Fig. 4. Obstacle avoidance. RG solution

In the second simulation, the robot task is to follow a vertical line from (3.41, 1.41) to (3.41, -0.3), in the presence of a disc obstacle of radius $r = 0.1$, located at $\bar{x} = 3.1$. The arm starts with $\boldsymbol{q} = (0^\circ, 0^\circ, 45^\circ, 45^\circ)$. The natural motion generated without any optimization, i.e. using the PS method, collides as expected with the obstacle (Fig. 3). On the contrary, Fig. 4 shows the safe arm movement generated by the RG method maximizing the criterion H_2. The amount of computation needed for this obstacle avoidance scheme can be estimated in approx. 12 trigonometric evaluations, 11 products, and 10 additions for each iteration.

In the last numerical example, the arm performs a self-motion to recover manipulability, while keeping the end-effector fixed at the origin. The almost singular initial configuration $\boldsymbol{q} = (-3^\circ, 3^\circ, 177^\circ, -177^\circ)$ is shown in Fig. 5, together with the configuration $\boldsymbol{q} = (-45^\circ, 45^\circ, 135^\circ, -135^\circ)$, where the index H_3 assumes its maximum value. Both RG and PG optimization techniques yield this same final solution. However, as indicated in Fig. 6, the RG method is faster, achieving convergence within half of the PG iterations. In this case, the set of basic variables is (q_1, q_4) and does not change during the whole motion.

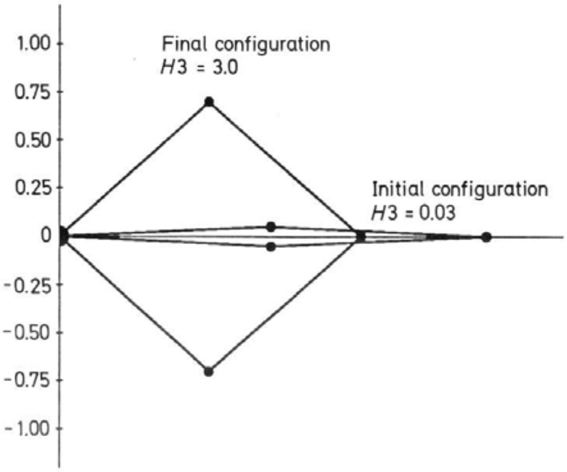

Fig. 5. Self-motion. Initial and final configurations

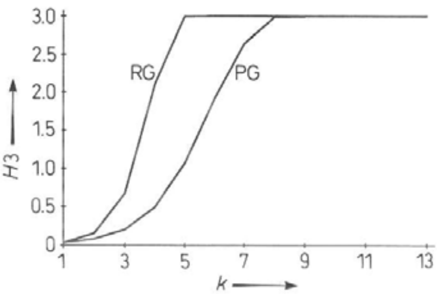

Fig. 6. Self-motion. Manipulability index vs. iterations

All simulations were performed with a constant stepsize $\alpha = 0.1$. Numerical evidence has shown that, when the correction term (14) is included, both RG and PG methods are less sensitive to the actual stepsize value.

6 Conclusions

A new computational scheme for solving redundancy in robot arms has been presented, based on the Reduced Gradient optimization method. The application of this technique to robotics relies on the physical intuition of the problem. The free selection of basic variables provides an explicit command on the choice of a particular inverse solution. With this respect, the approach can be recasted in the framework of inverse kinematic functions, as defined by Wampler [11]. The RG method has been applied here for local optimization of position-dependent criteria at a kinematic level. However, the method may be used also in global resolution schemes, as well as for revisiting other kinematic approaches like the task-priority method of Nakamura et al. [5]. In fact, the joint-space decomposition idea does not need the use of pseudoinversion and projection even when redundancy is resolved through task-augmentation [12]. Due to its generality, the presented scheme can be extended also to acceleration/torque resolution of redundancy [13], taking into account the full robot dynamic model. Finally, it is worth to mention that the explicit inversion of matrix $\mathbf{J}_R$ – the main requirement of the method – can be avoided by the proper application of the closed-loop scheme for coordinate transformations introduced in [14].

Acknowledgements

This work was partially supported by the Consiglio Nazionale delle Ricerche, grant no. 89.00521.67 (Progetto Finalizzato Robotica).

References

1. Yoshikawa, T.: Manipulability of robotic mechanisms. Int. J. Robotics Res. 4 (1985) 3–9
2. Klein, C. A.; Blaho, B. E.: Dexterity measures for the design and control of kinematically redundant manipulators. Int. J. Robotics Res. 6 (1987) 72–83
3. Maciejewski, A. A.; Klein, C. A.: Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments. Int. J. Robotics Res. 4 (1985) 109–117
4. Baillieul, J.: Avoiding obstacles and resolving kinematic redundancy. 1986 IEEE Conf. on Robotics and Automation, San Francisco, CA (1986) 1698–1704
5. Nakamura, Y.; Hanafusa, H.; Yoshikawa, T.: Task-priority based redundancy control of robot manipulators. Int. J. Robotics Res. 6 (1987) 3–15
6. Liègeois, A.: Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans. Systems, Man, and Cybern., SMC-7, 12 (1977) 868–871
7. Klein, C. A.; Huang, C. H.: Review of pseudoinverse control for use with kinematically redundant manipulators. IEEE Trans. Systems, Man, and Cybern., SMC-13 (1983) 245–250
8. Chevallereau, C.; Khalil, W.: A new method for the solution of the inverse kinematics of redundant robots. 1988 IEEE Conf. on Robotics and Automation, Philadelphia, PA (1988) 37–42
9. Dubey, R. V.; Euler, J. A.; Babcock, S. M.: An efficient gradient projection optimization scheme for a seven-degree of freedom redundant robot with spherical wrist. 1988 IEEE Conf. on Robotics and Automation, Philadelphia, PA (1988) 28–36
10. Luenberger, D. G.: Linear and nonlinear programming. Addison-Wesley, Reading 1984
11. Wampler, C. W.: Inverse kinematic functions for redundant manipulators. 1987 IEEE Conf. on Robotics and Automation, Raleigh, NC (1987) 610–617
12. De Luca, A.; Oriolo, G.: Kinematic resolution of redundancy via joint-space decomposition. 8th IFToMM Symp. on Theory and Practice of Robots and Manipulators (RoManSy '90), Krakow, PL (1990)
13. De Luca, A.; Oriolo, G.: Efficient dynamic resolution of robot redundancy. 1990 American Control Conf., San Diego, CA (1990) 221–227
14. Balestrino, A.; De Maria, G.; Sciavicco, L.; Siciliano, B.: An algorithmic approach to coordinate transformation for robotic manipulators. Advanced Robotics, 2 (1988) 327–344

Dr. A. De Luca
G. Oriolo
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Eudossiana 18
184 Rome
Italy

Received December 6, 1990