Conf-940550--10

A New Solution Method for the Inverse Kinematic Joint Velocity Calculations of Redundant Manipulators*

François G. Pin, Philippe F. R. Belmans, Jean-Christophe Culioli, Derek D. Carlson, and Faithlyn A. Tulloch
Autonomous Robotic Systems Group
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6025, MS-6364
Oak Ridge, TN 37831-6363, U.S.A.

Submitted to: 1994 IEEE International Conference on Robotics and Automation, May 8–13, 1994, San Diego, California

* Research sponsored by the Engineering and Geosciences Research Program, Office of Basic Energy Sciences, of the U.S. Department of Energy, under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

A New Solution Method for the Inverse Kinematic Joint Velocity Calculations of Redundant Manipulators

François G. Pin, Philippe F. R. Belmans, Jean-Christophe Culioli, Derek D. Carlson, and Faithlyn A. Tulloch
Autonomous Robotic Systems Group
Oak Ridge National Laboratory
P.O. Box 2008, Bldg. 6025, MS-6364
Oak Ridge, TN 37831-6363, U.S.A.

Abstract

A new analytical method to resolve underspecified systems of algebraic equations is presented. The method is referred to as the Full Space Parameterization (FSP) method and utilizes easily-calculated projected solution vectors to generate the entire space of solutions of the underspecified system. Analytic parameterizations for both the space of solutions and the null space of the system reduce the determination of a task-requirement-based single solution to a $m - n$ dimensional problem, where $m - n$ is the degree of underspecification, or degree of redundancy, of the system. An analytical solution is presented to directly calculate the least-norm solution from the parameterized space and the results are compared to solutions of the standard pseudo-inverse algorithm which embodies the (least-norm) Moore-Penrose generalized inverse. Application of the new solution method to a variety of systems and task requirements are discussed, and sample results using four-link planar manipulators with one or two degrees of redundancy and a seven degree-of-freedom manipulator with one or four degrees of redundancy are presented to illustrate the efficiency of the new FSP method and algorithm.

1. Introduction

This paper deals with the motion control of serial-link manipulators using solutions of the kinematic equations of the system:

$$\overline{X} = \overline{F}(\overline{q}) \quad (1)$$

or, since displacements are sought, solution of the velocity equations obtained by taking the time-derivatives of Eq. (1):

$$\dot{\overline{X}} = J(\overline{q})\dot{\overline{q}} \quad (2)$$

where $\overline{X}$ represents the $n \times 1$ vector of position and orientation of a given point of interest on the manipulator (generally the end-effector expressed in the n-dimensional Cartesian task space), $\overline{q}$ is the $m \times 1$ vector of joint coordinates for the m degrees of freedom (d.o.f.) manipulator, $\overline{F}$ represents the forward kinematic vector function for the system, an upper dot denotes a time-derivative, and $J(\overline{q})$ is the $n \times m$ Jacobian matrix of the system with component $J_{ij} = \partial F_i/\partial q_j$. Since Eq. (2) is typically highly non-linear, control of the system is generally performed using "loop-rate" cycles of calculation, with a linearized version of Eq. (2) providing first-order approximations for the displacement vectors $\Delta\overline{X}$ and $\Delta\overline{q}$ over the discretized time domain with time steps Δt:

$$\Delta\overline{X}/\Delta t \approx J_{\Delta t}\Delta\overline{q}/\Delta t \quad (3)$$

In solving Eq. (3) a high loop rate is generally desirable to minimize the errors generated by assuming that the linearized Jacobian matrix $J_{\Delta t}$ has constant coefficients during the time step Δt; and a feed-back loop in the calculational cycle needs to be utilized to minimize the error accumulation over time (tracking drift).

Solution of Eq. (3) at each time step essentially consists in solving a set of algebraic equations. When the manipulator is a redundant system, the Jacobian is a rectangular $n \times m$ matrix with $n < m$, and the system of equations represented by Eq. (3) is underspecified. If the Jacobian matrix is of rank n, there generally exists an infinity of solutions to the system of Eq. (3) and the matrix $J_{\Delta t}$ cannot be directly inverted.

In what follows, we will assume that the matrix $J_{\Delta t}$, thereafter simply denoted by J, is of rank n (the particular cases where the matrix is of rank $r < n$ will be treated in a following companion paper). Several methods exist to resolve underspecified systems of equations such as Eq. (3) with $n < m$, and Ref. [1] provides an excellent review of these methods for application to redundant manipulator control. Essentially, all existing methods (for instantaneous or real-time control) fall into two categories: those that use a generalized inverse to find a particular solution corresponding to a specific criterion (e.g., the pseudo inverse for the least norm of $\Delta\overline{q}$), on which is superimposed an homogenous solution corresponding to a secondary criterion or cost function (e.g., manipulability index [2], gradient projection method [3,4], obstacle or joint limit avoidance [5,6]) which produces a self-motion of the joints in the null space of the mapping J; and those that utilize a set of constraints or relationships on the task to generate an "augmented task space" (e.g., see [7], [8]) by adding some Cartesian space variables to the system, and produce an "extended Jacobian" (e.g., see [9]) which, if square, can then be inverted using direct inverse techniques.

Severe difficulties and/or shortcomings exist with each of these techniques and have been well documented and studied (e.g., see Refs. [1] through [9]). Among those drawbacks, to cite only a few, are the implicit task priority requirements, i.e., the fact that a solution found from the superposition of a particular solution obtained from a primary criterion and a homogenous solution obtained from a secondary criterion, typically does not satisfy both together (e.g., a solution obtained from a least-norm particular solution to which an obstacle-avoiding self-motion solution has been added, is not the least-norm solution of the obstacle-avoiding solutions); and the "artificial" algorithmic singularities that may be encountered with extended Jacobian and augmented task space approaches. With respect to singularities, it should be mentioned here that the Singular Value Decomposition (SVD) method (e.g., see [10]) is a particularly efficient method to alleviate the very significant problems of inverse kinematic manipulator control near singularities (e.g., see [11]). Its very high computational cost, however, may still represent an overkill for general (away from singularities) real-time operation of manipulators, while also suffering from some of the above-mentioned drawbacks. Overall, we contend that all of the existing methods present difficulties for application to *real-time* robotic control in *changing environments* because they each consit in finding a particular solution using an *a priori selected* criterion which at some time during the motion may, but typically does not over the entire trajectory, represent the actual requirements of the task motion. When the requirements do not correspond to the primary criterion anymore, one has to set a secondary criterion and essentially search in the extended space and/or null-space (without an explicit expression of it), to find a "better" solution satisfying the current constraints. Each task space extension and/or null space search requires its own analytic set up and specific algorithm. Consequently, when motions under real-time conditions would call for rapid changes in requirements (e.g. time optimality, least norm, obstacle avoidance, minimum torque, maximum global strength, optimum manipulability, etc.), these methods would require a full rescope of the solution algorithm in order to handle the particular secondary criteria and/or task space extension introduced *at each of the changes* in task requirement. In other words, real-time motion of autonomous manipulators under widely changing conditions would require the cumbersome storing on-board the robot of a correspondingly large library of complex algorithms (generalized inverse algorithms are complex and computationally costly), and the switching from one to the other as called by the real-time situation.

This paper proposes a different approach to the resolution of underspecified systems of equations, such as Eq. (3) for manipulators, in which the *entire space of solutions* is first determined (in the same fashion at each time step), and is very conveniently parameterized so that the subsequent calculation of a specific solution satisfying *all* the constraints and requirements of the particular time step becomes the matter of a few explicit programming statements. The following section describes how this new method, called the Full Space Parameterization (FSP) method, provides parameterizations for the entire space of solutions of Eq. (3), as well as for the corresponding null space (self-motion) of the mapping J, using easily-calculated projected vectors. Section 3 describes the analytic solution for the parameters that provide the least-norm solution within the space, and the results are compared to those of the pseudo-inverse, since this method has become "the standard" for generalized inverse techniques. Several sample problems involving 4 d.o.f. planar and 7 d.o.f. 3-D manipulators with respectively one and two, and one and four degrees of redundancy are presented to illustrate the results of the method in these comparisons of least-norm calculation. The last section briefly discusses other applications of the FSP that we have performed, and presents our conclusion.

2. Full Space Parameterization

One of the key ideas of the proposed methodology is to represent the set of solutions of the general underspecified system of Eq. (3) as an easily parameterized subspace of a much larger, but easily constructed vectorial space.

Consider the set of projected vectors $\overline{g}_k$ which are solutions of Eq. (3) and have $m-n$ fixed components. Without loss of generality these fixed components will be set to zero. There are C_m^{m-n} such m-dimensional vectors $\overline{g}_k$, each being associated with a n-dimensional vector $\overline{g}_k^*$ verifying the following property:

$$\Delta\overline{X} = J\overline{g}_k \iff \Delta\overline{X} = J_k\overline{g}_k^* \tag{4}$$

where J_k is the square submatrix of J obtained by removing the columns of J corresponding to the fixed zero components of $\overline{g}_k$. $\overline{g}_k^*$, therefore, is the n-dimensional vector constituted of the ordered n nonzero components of $\overline{g}_k$. Since the J_k, $k=1$, C_m^{m-n}, are square $n \times n$ matrices, each $\overline{g}_k^*$ corresponding to a non-singular matrix J_k can be very easily obtained from the right-hand side of Eq. (4). With J being of rank n, the existence of at least $m-n+1$ non-singular square matrices J_k, can be proven as follows.

Recall that J, of dimension $(n \times m)$, is constituted of m columns c_i, and is of rank n; therefore there exist at least one square matrix J_1 which is invertible. For ease of indexing and without loss of generality, we assume this matrix is constituted by the first n columns of matrix J, therefore:

$$J_1 = (c_1 \ldots c_n) \quad . \tag{5}$$

Let us construct the matrix J_2 by substituting one column c_l, $l \in \{1, \ldots, n\}$ of J_1 by column c_{n+1} of J.

$$J_2 = (c_1 \quad \ldots \quad c_{l-1} \quad c_{n+1} \quad c_{l+1} \quad \ldots \quad c_n) \quad . \tag{6}$$

Two cases may happen:

- $\det(J_2) \neq 0$ and so the matrix J_2 is invertible, or
- $\det(J_2) = 0$ and c_{n+1} is either a column constituted of 0's, or a linear combination of columns $c_i, i \in \{1, \ldots, n\} - \{l\}$. Let c_h denote one of these columns c_i, $i \in \{1, \ldots, n\} - \{l\}$. The first case means that the component Δq_{n+1} of the joint velocity vector $\Delta\overline{q}$ has no influence on the end-effector motion and any value for Δq_{n+1} is acceptable. We can set $\Delta q_{n+1} = 0$ and solve the problem as an $m-1$ dimensional problem in joint space. In that case, we have to find $(m-n)$ sub-matrices J_k instead of $(m-n+1)$. We thus can neglect J_2 and continue the search process with J_3, etc.

In the latter case, the previous substitution is cancelled, and instead we substitute c_h, $h \in \{1, \ldots, n\} - \{l\}$ by c_{n+1}. J_2 therefore becomes

$$J_2 = (c_1 \ \ldots \ c_{h-1} \ c_{n+1} \ c_{h+1} \ \ldots \ c_l \ \ldots \ c_n) \ . \quad (7)$$

Since c_{n+1} is a linear combination of columns c_i, $i \in \{1, \ldots, n\} - \{l\}$, which includes c_h, we have:

$$c_{n+1} = \sum_i \alpha_i c_i \quad (8)$$

and

$$\begin{aligned} \det(J_2) &= \det(c_1 \ \ldots \ c_{h-1} \ c_{n+1} \ c_{h+1} \ \ldots \ c_n) \\ &= \prod_i \alpha_i \det(c_1 \ \ldots \ c_{h-1} \ c_h \ c_{h+1} \ \ldots \ c_n) \\ &= \prod_i \alpha_i \det(J_1) \\ &\neq 0 \end{aligned} \quad (9)$$

Therefore the matrix J_2 is invertible.

This process can be applied to the $(m-n)$ columns c_j, $j \in \{n+1, \ldots, m\}$, and consequently there exist at least $(m-n+1)$ invertible square submatrices J_k.

Notice here that in the general case, many more than $m-n+1$ submatrices of J will be invertible and will lead to a projected vector solution since, even in the latter case above where a column c_h of the linear combination of c_{n+1} needs to be substituted, c_h will not be the only column in the linear decomposition of c_{n+1} (unless the manipulator is "folded" on itself in a singular configuration), and therefore, other matrices J_2' could be shown to be non singular and to lead to a projected vector. In similar fashion, other J_k' could be shown to be non singular and to lead to projected vectors during the successive identification of the invertible $J_k, k = 1, m-n+1$.

Let us now consider a maximal set of p independent vectors $\overline{g}_k, k = 1, \ldots, p$ among those found through the submatrices inversion, and introduce the space spanned by this family: Span $\{\overline{g}_1, \overline{g}_2, \ldots \overline{g}_p\}$.

We can construct the affine space E (a subspace of Span $\{\overline{g}_1, \ldots \overline{g}_p\}$), of dimension $\dim(E) = p-1$, defined by:

$$E = \left\{ \Delta\overline{q} \in \mathbb{R}^m, \Delta\overline{q}(t_1, \ldots, t_p) = \sum_{i=1}^{p} t_i \overline{g}_i ; \quad \sum_{k=1}^{p} t_k = 1 \right\} \quad (10)$$

Proposition: $E \subset \mathcal{E}$, where $\mathcal{E}$ is the space of solutions of Eq. (3).

Proof: from Eq. (4), we have

$$\Delta\overline{X} = J\overline{g}_i, \quad \forall_i \in \{1, p\}, \quad (11)$$

and therefore

$$t_i \Delta\overline{X} = J t_i \overline{g}_i \quad \forall t_i \in \mathbb{R}, \quad \forall_i \in \{1, p\} \quad . \quad (12)$$

Consequently,

$$\sum_{i=1}^{p} (t_i \Delta\overline{X}) = \sum_{i=1}^{p} (J t_i \overline{g}_i) \quad (13)$$

and

$$\left(\sum_{i=1}^{p} t_i \right) \Delta\overline{X} = J \left(\sum_{i=1}^{p} t_i \overline{g}_i \right) \quad . \quad (14)$$

Thus, any vector $\Delta\overline{q}$ of E, $\Delta\overline{q} = \sum_{i=1}^{p} t_i \overline{g}_i$ with $\sum_{i=1}^{p} t_i = 1$, is a solution of Eq. (3). Therefore, since $E \subset \mathcal{E}$, and $\dim(E) = p-1$, then $E = \mathcal{E}$ for $p = \dim(\mathcal{E}) + 1$.

Consequently, since $\dim(\mathcal{E}) \leq m-n$, at most $m-n+1$ linearly independent vectors $\overline{g}_k$ suffice to parameterize the full space of solution of Eq. (3) using Eq. (10). Moreover, each of these $m-n+1$ vectors is obtained from inversion of an $(n \times n)$ square submatrix, which, even in the worst case of $m-n=1$, is computationally less expensive than the inversion of $(m \times m)$ matrices required in generalized inverse or augmented task space methods.

From Eq. (14) and the dimensionality relations above, it is straightforward to show that the null space $\mathcal{N}$ of the mapping J can also be directly derived and parameterized as:

$$\mathcal{N} = \left\{ \Delta\overline{q} \in \mathbb{R}^m, \Delta\overline{q}(t_1, \ldots, t_{m-n+1}) = \sum_{i=1}^{m-n+1} t_i \overline{g}_i ; \quad \sum_{k=1}^{m-n+1} t_k = 0 \right\} \quad (15)$$

An important feature of the method proposed here is that the underspecified problem in the m-dimensional articular space has been reduced to an $m-n$ dimensional problem in the parameter space of $\{t_1, \ldots t_{m-n+1}\}$ with the constraint $\sum_{k=1}^{m-n+1} t_k = 1$ or $\sum_{k=1}^{m-n+1} t_k = 0$ depending on the solution being sought in the space E of solutions of Eq. (3) or in the null space $\mathcal{N}$ of the mapping J. This is quite a reduction in dimensionality, since, for example, a 7 d.o.f. manipulator operated in $n = 6$ dimensional task space (3-D position and orientation) requires only 2 linearly independent vectors $\overline{g}_1$ and $\overline{g}_2$ to construct E or $\mathcal{N}$, and only one parameter t_1 (since $t_2 = 1 - t_1$) to be found for any desired particular solution (e.g. least norm, etc.).

3. Analytical Determination of Particular Solutions

Only a few authors have studied solution techniques involving linear combinations of vector sets similar to that discussed here for manipulator redundancy resolution, and they have done so only for very particular control situations (e.g., see the "joint blocking" technique for acceleration-level control in [12]) or for unconstrained conditions (e.g., see [13]). Here, we show how the general Lagrangian-type constrained optimization formalism can be used to derive the solution parameter set $\{t_1, \ldots t_p\}$, in either Eq. (10) or Eq. (15), for a general task requirement seeking to optimize a criterion $Q(\Delta\overline{q}(t_i))$ under a set of r constraints $C^j(\Delta\overline{q}(t_i)) = 0$; $j = 1, r$; $i = 1, m-n+1$.

Since derivation of the wide variety of analytical solutions that correspond to the various *types* of criteria and/or constraints typically involved in manipulator control is clearly beyond the scope of this paper (many of these solutions for both discrete and continuous criteria, and for various types of constraints such as joint limit avoidance, obstacle avoidance, etc., have been derived in [14] and will appear in companion papers), we only present here an example of such an analytical solution for a "general" least-norm criterion with a set of constraints expressed in a form most encountered in manipulator kinematics [14]. Since the focus of this initial paper is to present comparisons of the FSP method with the "standard" least norm (of the joint velocities) algorithm, we then reduce the general solution to this particular case, and apply it to several illustrative manipulator systems.

For a general criterion $Q(\Delta\overline{q}(t_i))$, $i = 1, m-n+1$ to be optimized in the space defined by Eq. (10) with a set of r general constraints $C^j(\Delta\overline{q}(t_i)) = 0$, $j = 1$, r; the Lagrangian is:

$$\mathcal{L}(t_i, \mu, \nu_j) = Q(t_i) + \mu\left(\sum_{i=1}^{m-n+1} t_i - 1\right) + \sum_{j=1}^{r} \nu_j C^j(t_i) \tag{16}$$

and the optimality conditions are:

$$\frac{\partial \mathcal{L}}{\partial t_i} = 0, i = 1, m-n+1; \frac{\partial \mathcal{L}}{\partial \mu} = 0; \frac{\partial \mathcal{L}}{\partial \nu_j} = 0, j = 1, r \tag{17}$$

With these conditions, analytical solutions for the $m-n+1$ dimensional vector $\overline{t}$ with components $(t_1, \ldots t_{m-n+1})$ can be found such that the resulting $\Delta\overline{q} = \sum_{i=1}^{m-n+1} t_i \overline{g}_i$ optimizes Q while satisfying all the constraints. As an example of such an analytical derivation, consider a general criterion

$$Q = \|\Delta\overline{Z}(\overline{q}, \Delta\overline{q}) - \Delta\overline{Z}_r\|^2 \tag{18}$$

where $\Delta\overline{Z}_r$ represents a given reference operational vector characterizing the state to be acheived by system, and $\Delta\overline{Z}$ is an operational vector function of the joint positions and displacements. Let $B(\overline{q})$ be a matrix such that

$$\Delta\overline{Z} = B(\overline{q})\Delta\overline{q} \tag{19}$$

and define the vector $\overline{H}$ and matrix G as:

$$\overline{H}, H_k = \Delta\overline{Z}^T B\overline{g}_k; \quad k = 1, \ m-n+1 \tag{20}$$

$$G, G_{ij} = \overline{g}_i^T B^T B\overline{g}_j; i = 1, m-n+1; j = 1, m-n+1 \tag{21}$$

where the upper T sign denotes a transpose.

Assume the r constraints $C^j(\Delta\overline{q}(\overline{t}) = 0)$ are expressed as

$$\overline{\beta^j}^T \overline{t} - 1 = 0 \; ; \; j = 1, \ r \tag{22}$$

a form to which many kinematic constraints (e.g., joint limits, obstacle avoidance, etc.) can be reduced [14]. Then the optimality conditions [Eq. (17)] become:

$$\begin{cases} G\overline{t} + \overline{H} + \mu\overline{e} + \sum_{i=1}^{r} \nu_i \overline{\beta}^i = \overline{o} \\ \overline{e}^T\overline{t} = 1 \\ \overline{\beta^j}^T \overline{t} = 1; \quad j = 1, \ r \end{cases} \tag{23}$$

where $\overline{e}$ and $\overline{o}$ are the $m-n+1$ dimensional vectors $\overline{e}^T = (1, 1, 1, \ldots 1)$ and $\overline{o}^T = (0, 0, \ldots 0)$, respectively. Setting $\overline{\nu}^T = (\nu_1, \ldots \nu_r)$ and $a = \overline{e}^T G^{-1}\overline{e}$; and defining the vector $\overline{b}$, $\overline{c}$, and $\overline{d}$, and the matrix A by: $b_i = \overline{e}^T G^{-1}\overline{\beta}^i$, $c_i = \overline{\beta^i}^T G^{-1}\overline{e}$, $d_i = 1 + \overline{\beta^i}^T G^{-1}\overline{H}$, $A_{ij} = c_i b_j - a\overline{\beta^i}^T G^{-1}\overline{\beta}^j$, $i = 1, r, j = 1, r$; the solution of Eq. (23) for the Lagrange multipliers and parameter set can be written as:

$$\overline{\nu} = A^{-1}(a\overline{d} - \overline{c}(1 + \overline{e}^T g^{-1}\overline{H})) \tag{24}$$

$$\mu = -(1 + \overline{\nu}^T\overline{b} + \overline{e}^T g^{-1}\overline{H})/a \tag{25}$$

$$\overline{t} = -G^{-1}(\mu\overline{e} + \sum_{i=1}^{r} \nu_i\overline{\beta}^i + \overline{H}) \tag{26}$$

For comparison with the pseudo-inverse algorithm which optimizes the least norm of the joint displacements in unconstrained conditions, we set $\Delta\overline{Z}_r = \overline{o}, \Delta\overline{Z} = \Delta\overline{q}$, and consequently, B in Eq. (19) is the identity matrix. Thus G becomes the Grammian matrix of the vectors $\overline{g}_i, i = 1, m-n+1$, and $\overline{H} = \overline{o}$. Since in the unconstrained condition $\overline{\beta}^i = \overline{o}, i = 1, r$, Eq. (26) reduces to

$$\overline{t}^* = G^{-1}\overline{e}/(\overline{e}^T G^{-1}\overline{e}) \tag{27}$$

and the least norm solution is obtained as:

$$\Delta\overline{q}_{l.n.} = \sum_{i=1}^{m-n+1} t_i^*\overline{g}_i \tag{28}$$

4. Experimental Results

The IMSL standard pseudo-inverse algorithm and the FSP method described above with the least-norm solution of Eqs. (27) and (28) were both implemented on several redundant manipulator test-beds and tested on a variety of trajectories. In all cases, a feedback loop from the output of the redundancy resolution was utilized to add to the desired $\Delta\overline{X}^{n+1}$ vector at time step $n+1$ the error between the desired configuration $\overline{X}^n$ at time step n and the configuration actually resulting from the joint displacement solution of time step n. The actual configurations, calculated using the forward kinematic Eq. (1), are those plotted in the following figures.

Figure 1 shows two example trajectories for a simulated 4 d.o.f. planar manipulator, with equal length in all four links. Both the position (x, y) and orientation (ψ) of the end-effector are controlled, leading to one degree of redundancy (d.o.r.) with $n = 3$ and $m = 4$. In this case, there are $C_4^3 = 4$ possible (3×3) submatrices and, if invertible, 4 projected vectors $\overline{g}_i$. Since only $m - n + 1 = 2$ independent vectors are necessary, there are $C_4^2 = 6$ sets of vectors that could potentially be used to construct the entire space E from Eq. (10). In an actual implementation, all these combinations would of course not need to be calculated (although computations in parallel are feasible and would not significantly slow down the algorithm). For the purpose of our verification studies, however, each feasible combination was calculated at every time step of each test trajectory (typically all 4 submatrices were invertible and all 6 combinations were found suitable), each combination leading to a corresponding (and of course different) solution for $\overline{t}$ from Eq. (27). The resulting solutions $\Delta\overline{q}$, from Eq. (28) were compared with each other and with the pseudo-inverse solution, systematically showing better than five digit accuracy in every case.

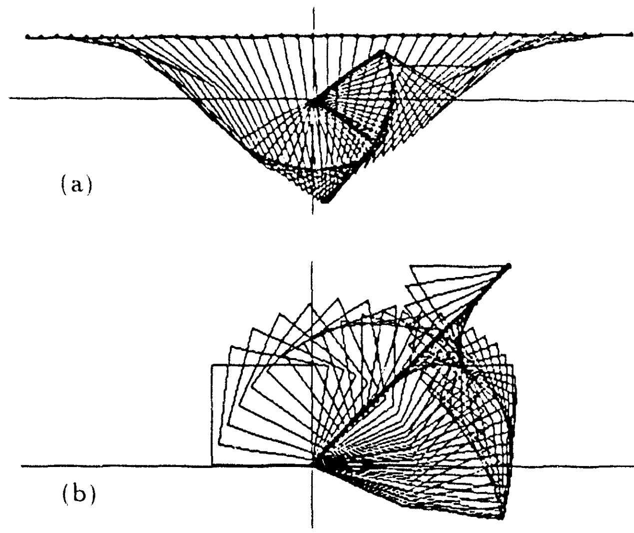

Fig. 1. Least-norm solution of the FSP method on sample trajectories of a 4 d.o.f. planar manipulator with 1 d.o.r. (position and orientation control) with a) 180°, b) 360° total orientation change over the trajectory.

Figure 2 shows two example trajectories for the simulated 4 d.o.f. planar manipulator controlled in position (x, y) only, therefore with 2 d.o.r. In this case, there are 6 (2×2) square submatrices and $C_6^3 = 20$ possible combinations of vectors $\overline{g}_i$. The four combinations that involved three vectors $\overline{g}_i$ exhibiting a zero at the same component were rejected, and all feasible others were investigated. Typically all 16 were found suitable and leading to a parameter solution $\overline{t}$. Comparisons of the resulting solutions $\Delta\overline{q}$ with the pseudo-inverse solution also showed better than five digit accuracy in every case.

The FSP method was also implemented on several hardware systems, including the 7 d.o.f. CESARm manipulator [4],[15],[16]. On this system, FSP-based control schemes were implemented for control in 3-D of the position of the wrist, the position of the end-effector, or the position and orientation of the end-effector, leading respectively to redundancy resolution with 1 d.o.r. on a 4 d.o.f. 3-D system, and 4 d.o.r. and 1 d.o.r. on a 7 d.o.f. 3-D system. Although closed loop at the redundancy resolution level (as explained previously), the control schemes drove the manipulator in open loop fashion (i.e., with no joint-encoder data feedback). Figure 3 shows an example of the FSP results using a sequence of 3-D graphics displaying the motion of the 7 d.o.f. CESARm (located on the HERMIES-III mobile robot) over a sample position and orientation control trajectory with least-norm minimization. For everyone of these tests in 3-D, the FSP solutions were compared with the results of the pseudo-inverse algorithm and showed better than five digit accuracy.

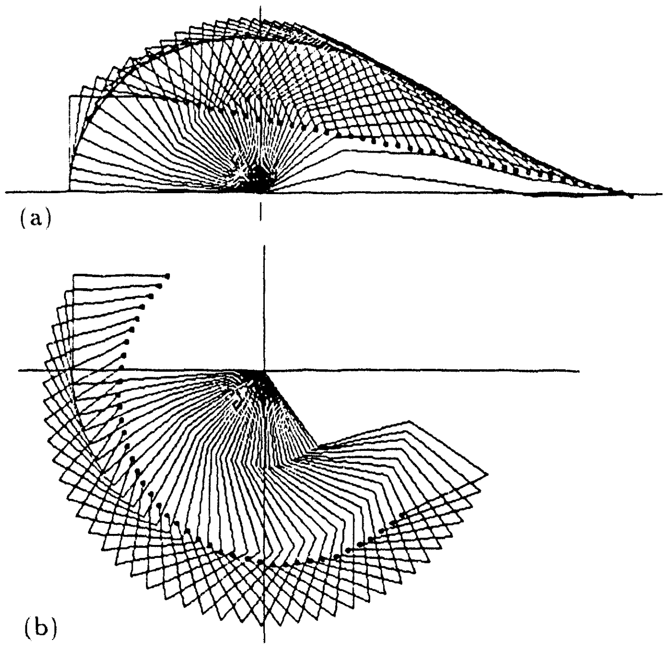

Fig. 2. Least-norm solution of the FSP method on a) straight line, b) circular, sample trajectories of a 4 d.o.f. planar manipulator with 2 d.o.r. (position control only).

5. Conclusion

A new approach for controlling redundant manipulators under varying criteria and constraints has been presented. The approach uses the Full Space Parameterization (FSP) method to generate the entire solution space as well as the null space for the underspecified system. Within these spaces, the optimization problem to find a specific solution reduces to an $m-n$ dimensional problem in parameter space. An example of analytical solution has been derived for the generic case of a criterion involving the least norm of a general operational vector funtion with a set of kinematic and/or environment constraints imposed on the system. The solution has then been reduced to the case of unconstrained least norm of the joint displacements sc that comparisons could be performed between the new FSP approach and the "standard" pseudo-inverse algorithm. In these comparisons, the two methods provided identical results to within numerical round-off errors (at the sixth significant digit). As part of these comparisons, sample results obtained with the FSP were presented for a 4 d.o.f. planar manipulator and a 7 d.o.f. 3-D manipulator.

Acknowledgments

This research was supported by the Engineering and Geosciences Research Program of the Office of Basic Energy Sciences of the U.S. Department of

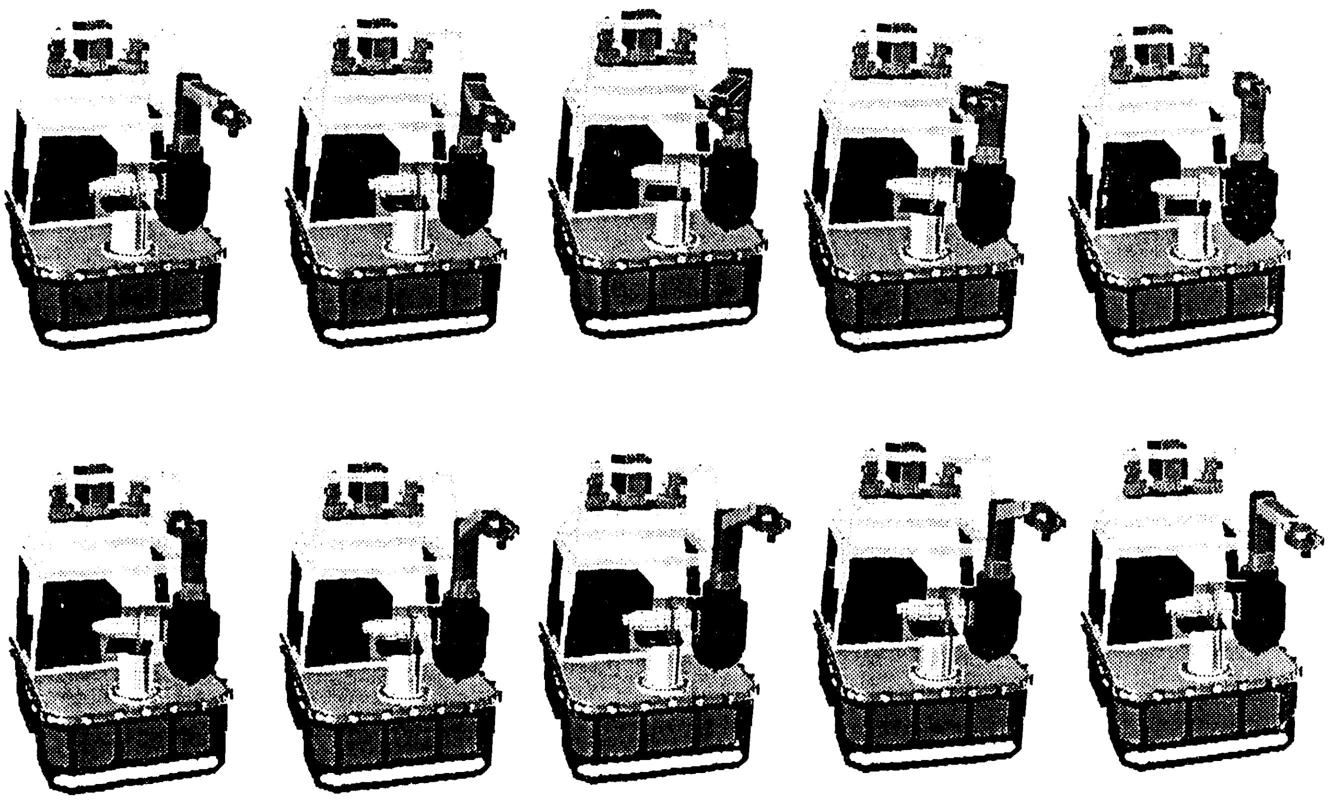

Fig. 3. Example of least-norm solution of the FSP method for 3-D position and orientation control of the 7 d.o.f. CESARm manipulator's end-effector over a square trajectory while maintaining constant orientation.

Energy under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. Part of the work was performed while the second and third authors, and the fourth and fifth authors were respectively in DOE-sponsored Postdoctoral Research Fellowship programs and Undergraduate Science and Engineering Reseach programs, administered by the Oak Ridge Institute for Science and Engineering.

References

[1] B. Siciliano, "Kinematic Control of Redundant Robot Manipulators: A Tutorial," *Journal of Intelligent and Robotic Systems* **3**: 201–212, 1990.

[2] T. Yoshikawa, "Manipulability of Robotic Mechanisms," *International Journal of Robotics Research* **4**(2): 3–9, 1985.

[3] C. Chevallereau and W. Khalil, "A New Method for the Solution of the Inverse Kinematics of Redundant Robots," *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, Washington, 37–42, 1988.

[4] R. V. Dubey, J. A. Euler, and S. M. Babcock, "An Efficient Gradient Projection Optimization Scheme for a Seven-Degree-of-Freedom Redundant Robot with Spherical Wrist, *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, Washington, 28–36, 1988.

[5] A. Liégeois, "Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms, *IEEE Transactions on Systems, Man, and Cybernetics* **7**, 868–871, 1977.

[6] A. A. Maciejewski and C. A. Klein, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," *International Journal of Robotics Research* **4**(3), 109–117, 1985.

[7] L. Sciavicco and B. Siciliano, "Solving the Inverse Kinematic Problem for Robotic Manipulators," *Proceedings of the 6th CISM-IFToMM Symposium on Theory and Practice of Robots and Manipulators*, (eds. A. Morecki, G. Bianchi, and K. Kedzior), MIT Press, Cambridge, 107–114, 1987.

[8] O. England, "Task-Space Tracking with Redundant Manipulators," *IEEE Journal of Robotics and Automation* **3**, 471–475, 1987.

[9] J. Baillieul, "Kinematic Programming Alternatives for Redundant Manipulators," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, Silver Spring, 722–728, 1985.

[10] G. H. Golub and C. F. Van Loan, "Matrix Computations" Johns Hopkins University Press, Baltimore, 1983.

[11] A. A. Maciejewski and C. A. Klein, "The Singular Value Decomposition: Computation and Applications to Robotics," *International Journal of Robotics Research* **8**(6), 63–79, 1989.

[12] S. Hirose and S. Ma, "Redundancy Decomposition Control for Multi-Joint Manipulator," *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, IEEE Computer Society Press, Washington, 119-124, 1989.

[13] P. F. R. Belmans and J.-C. Culioli, "A New Approach to Solve the Kinematics Resolution of a Redundant Robot," Oak Ridge National Laboratory Technical Report No. ORNL/TM-11435, 1990.

[14] F. G. Pin, "The Full Space Parameterization (FSP) Method for Solution of Underspecified Systems of Algebraic Equations: Theory and Application to Redundant Manipulator Control with Changing Criteria and Constraints," Oak Ridge National Laboratory Technical Report No. ORNL/TM-12510, 1993.

[15] J. N. Anderson and W. D. Combs, "Link Mass Properties of the CESAR Research Manipulator," Martin Marietta Energy System, Inc. Technical Report No. MCTR-0987-14, September 1987.

[16] R. V. Dubey, J. A. Euler, and S. M. Babcock, "Real-Time Implementation of an Optimization Scheme for Seven-Degree-of-Freedom Redundant Manipulators," *IEEE Trans. on Robotics and Automation* **7**(5), 579–588, 1991.

END

DATE FILMED

4/8/94