

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

Working Paper No. 332

December 1990

Correction of Force Errors for Flexible Manipulators in Quasi-Static Conditions

Antonio Bicchi
Claudio Melchiorri

Abstract

This paper deals with the problem of controlling the interactions of flexible manipulators with their environment. For executing a force control task, a manipulator with intrinsic (mechanical) compliance has some advantages over the rigid manipulators commonly employed in position control tasks. In particular, stability margins of the force control loop are increased, and robustness to uncertainties in the model of the environment is improved for compliant arms. On the other hand, the deformations of the arm under the applied load give rise to errors, that ultimately reflect in force control errors. This paper addresses the problem of evaluating these errors, and of compensating for them with suitable joint angle corrections. A solution to this problem is proposed in the simplifying assumptions that an accurate model of the arm flexibility is known, and that quasi-static corrections are of interest.

Copyright © Massachusetts Institute of Technology, 1990

A. I. Laboratory Working Papers are produced for internal circulation, and may contain information that is, for example, too preliminary or too detailed for formal publication. It is not intended that they should be considered papers to which reference can be made in the literature.

1 Introduction

When a manipulator interacts with the environment, there is a force exchange between the links of the robot and the object(s) being contacted. In many robotics applications, the assigned task consists in precisely applying a specified force and/or moment to the environment (force control problem); in addition, the control of motions of the robot end-effector along some directions can be specified (hybrid control problem). In order to accomplish such tasks, it is necessary to achieve a precise positioning of the part(s) of the manipulator in contact with the environment. On the other hand, force control of manipulators poses difficult problems, as abundantly illustrated in literature (a classical review is given by Whitney [1987]). Probably, the most important of these problems is the tendency of force control loops to become unstable in presence of small perturbations of the model parameters used to design the controller. One of the reasons of this difficulty can be described very loosely by noting that the control of the contact forces between two objects within a given accuracy is equivalent to controlling their relative position within an accuracy k times higher, where k here is the combined stiffness of the bodies, and is typically a very large number. Reducing the stiffness of the robot-environment system is therefore a viable way of alleviating force control problems: a more precise formulation of this concept can be found e.g. in [Roberts, Paul, and Hillberry, 1985], [Eppinger and Seering, 1987], and [Chiou and Shahinpoor, 1990]. This observation suggests an interesting application of flexible robots, that might encourage to design future robots with a built-in compliance. Besides this advantage, the design of lightweight, slender link robot arms meets other important requirements in applications such as space or underwater, and could reduce the cost of robot arms.

However, if the manipulator is composed of flexible links, and/or its joints are compliant, the positioning accuracy of the robot may be greatly reduced. Neglecting its flexibility, i.e. considering the manipulator as a chain of rigid bodies connected by perfect joints, the arm position in the world frame can be evaluated on the basis of the joint positions and the forward kinematic equations. This estimate of the arm posture is usually very inaccurate for flexible manipulators in contact with the environment, due to the deformation of its structure caused by contact forces.

In order to reduce these positioning errors, the force information provided by the force sensors used to close the force control loop can be exploited. However, these sensors are usually located as close as possible to the end-effector, so that the measured force is expressed in a reference frame whose position and orientation in the world frame depend in turn on the elastic deformations. A consequence is that the force information provided by the sensors cannot be directly used in the control loop. However, if a model of the arm compliance is available, it is possible, by means of a recursive algorithm, to compute that position and orientation. We will describe and use such an algorithm in section 4 to evaluate the real position and orientation of a flexible robot given the forces acting on it. A substantially equivalent method for predicting the deflection of serial manipulators with flexible links and

joints has been presented by Fresonke, Hernandez, and Tesar [1988].

Once the deflections of the arm under the given load are known, appropriate corrections of the robot inputs can be applied to minimize the force/position errors due to flexibility. Since the overall deflection of the robot is comprised of both joint and link flexibilities, the determination of this correction is not trivial.

In this paper, we will be mostly concerned with the problem of finding the optimal joint position in order to minimize the error in a force-control task. This problem is basically regarded as a planning phase preceding real-time implementation, so that a quasi-static assumption is made, and dynamic effects of flexibility are disregarded. In general, the modification that may be applied to the joint inputs will be able to compensate only in part for the deformation of the manipulator. On the other hand, the case when there are more joints than strictly necessary in the kinematic chain (redundant manipulators) is also considered. In order to homogeneously treat both such defective and redundant cases, the correction problem is cast in a nonlinear optimization framework, as described in section 3.

The paper is organized as follows. In section 2, the model considered for the links of the flexible manipulator is described, and the main assumptions used in this paper are given. In section 3 the problem of the force error compensation is formulated as a minimization problem, and an algorithm for its solution is presented. Section 4 describes some specific details of the algorithm in the context under consideration, while section 5 reports some examples. The final section 6 concludes with comments about the suggested technique and plans for future activity.

2 Model of the Flexible Arm

In this section, both a general description of the problem, considering a flexible robot arm in interaction with an environment that reacts to its movements, and the simplifying assumptions used in the rest of the paper are presented.

The general structure of a flexible arm dynamics can be described by a set of partial differential equations, of degree two for the torsional and axial modes, and four for the bending modes of the links. Consider for instance the simple manipulator link depicted in Figure 1, with constant cross-section and lying in a horizontal plane. In this case, only the axial and bending deflections of the link in the plane are of interest. The axial dynamics can be written as

$$EA \frac{\partial^2 x}{\partial \zeta^2} - \rho \frac{\partial^2 x}{\partial t^2} = 0 \quad (1)$$

where ζ is the coordinate along the undeformed beam axis, $x = x(\zeta, t)$ is the displacement at time t of the section initially in ζ , E is the elastic modulus of the link, and ρ its mass per unit length.

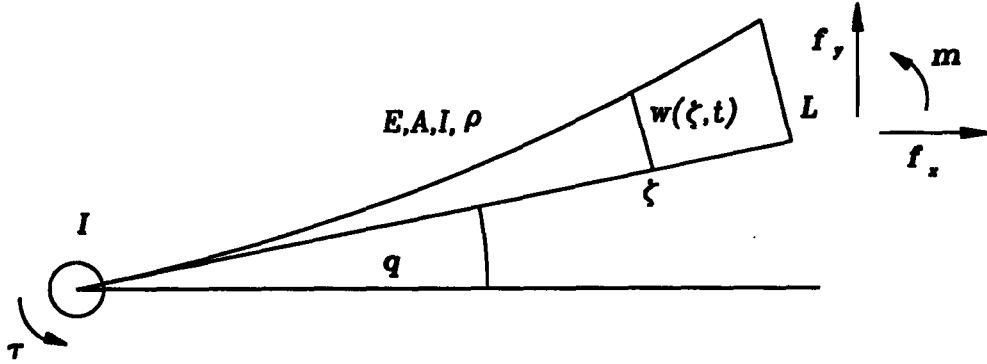


Figure 1: A simple manipulator link and its elastic model

The bending dynamics of the flexible link can be written as

$$EJ \frac{\partial^4 y}{\partial \zeta^4} + \rho \frac{\partial^2 y}{\partial t^2} = 0 \quad (2)$$

where J is the cross-sectional moment of inertia of the beam, and y is defined as (see e.g. [Korolov and Chen, 1988])

$$y(\zeta, t) = w(\zeta, t) + \zeta q.$$

Here, w is the elastic displacement measured from the undeformed axis (see Figure 1), and q is the nominal (hub) position of the joint. Note that lumped joint elasticity is not considered in this example.

The overall manipulator dynamic relationship can be built (for serial link arms) by concatenating differential equations of this type by their boundary conditions, that express the equilibrium of control torques, inertial moments, and constraints at the ends of the links. In the example of Figure 1, boundary conditions for eq. (1) can be written as

$$EA \left. \frac{\partial x}{\partial \zeta} \right|_{\zeta=L} = f_x, \quad (3)$$

$$x(0, t) = 0, \quad (4)$$

where A is the cross-section area, L is the total length of the link, and f_x is the axial component of the force exerted on the link's end by the hub of the next link (or by the environment if the distal link of the arm is considered). The four boundary conditions for eq. (2) are:

$$EJ \left. \frac{\partial^2 y}{\partial \zeta^2} \right]_{\zeta=0} + \tau - J_h \ddot{q} = 0; \quad (5)$$

$$y(0, t) = 0; \quad (6)$$

$$EJ \left. \frac{\partial^2 y}{\partial \zeta^2} \right]_{\zeta=L} = m; \quad (7)$$

$$EJ \left. \frac{\partial^3 y}{\partial \zeta^3} \right]_{\zeta=L} = f_y, \quad (8)$$

where τ is the control torque on the joint preceding the link under consideration, J_h the lumped moment of inertia at the hub, f_y and m the bending component of the force and the torque applied at the link's extremity.

The force \mathbf{f} and torque \mathbf{m} at the link end balance next link's forces and control torque, except for the distal link, for which they represent the environment reaction on the robot end-effector. These interaction forces, in general both time and space-varying, act as a forcing input on the dynamic chain.

When only quasi-static conditions are considered, dynamic terms in the relationships above can be neglected. Therefore, eq. (1) and eq. (2) reduce to the following ordinary differential equations of elastic beams:

$$\frac{\partial^2 x}{\partial \zeta^2} = 0, \quad (9)$$

$$\frac{\partial^4 y}{\partial \zeta^4} = 0. \quad (10)$$

Deflections of links under the loads applied at their extremities can be determined by integrating such relationships. For instance, integrating eq. (9) with the boundary conditions (3), (4), we obtain:

$$\Delta x = \frac{L \mathbf{f}_x}{EA},$$

while integration of eq. (10) with the boundary conditions (6)-(8) leads to:

$$\begin{aligned} \Delta y &= \frac{L^3 \mathbf{f}_y}{3EJ} + \frac{L^2 \mathbf{m}}{2EJ}, \\ \Delta \theta &= \frac{L^2 \mathbf{f}_y}{2EJ} + \frac{L \mathbf{m}}{EJ}, \end{aligned}$$

where Δx , Δy are the displacements of the extremity of the link along the link axis and normal to it, respectively, and $\Delta\theta$ is the rotation of the link's end.

In the three-dimensional case, similar relationships hold, and a general linear equation relating the six elastic displacements of the link's end with the applied load can be written in matrix form as:

$$\begin{bmatrix} \Delta d \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} \frac{1}{EA} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{L^3}{3EJ} & 0 & 0 & 0 & \frac{L^2}{2EJ} \\ 0 & 0 & \frac{L^3}{3EJ} & 0 & -\frac{L^2}{2EJ} & 0 \\ 0 & 0 & 0 & \frac{L}{GJ_0} & 0 & 0 \\ 0 & 0 & -\frac{L^2}{2EJ} & 0 & \frac{L}{EJ} & 0 \\ 0 & \frac{L^2}{2EJ} & 0 & 0 & 0 & \frac{L}{EJ} \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} \quad (11)$$

where $\Delta\mathbf{d}$ and $\Delta\theta$ are the displacements and rotation vectors of the link's extremity in 3D space, G is the shear modulus of the link's material, and J_0 is the polar momentum of the beam cross section.

The forces and torques on the distal ends of the links can be evaluated by means of static equilibrium considerations, using the recursive algorithm presented in section 4. The recursive calculation is started at the last link, where the load coincides with the robot - environment interaction forces, which are assumed to be known (by means e.g. of a wrist mounted force/torque sensor). However, this knowledge is relative to a reference frame E fixed to the end-effector. By solving the elastic displacements of every link, the geometric relationship of the end-effector frame E with the fixed (base) frame B can be found. Therefore, also the relationship between the interaction forces in E and in B is established as

$$\begin{bmatrix} {}^B\mathbf{f} \\ {}^B\mathbf{m} \end{bmatrix} = \begin{bmatrix} {}^B\mathbf{R}_E & \mathbf{0} \\ {}^B\mathbf{P}_\otimes {}^B\mathbf{R}_E & {}^B\mathbf{R}_E \end{bmatrix} \begin{bmatrix} {}^E\mathbf{f} \\ {}^E\mathbf{m} \end{bmatrix} \quad (12)$$

where ${}^B\mathbf{P}_\otimes$ is a skew-symmetric matrix equivalent to the cross product (${}^B\mathbf{p} \times$), being ${}^B\mathbf{p}$ the vector of the linear displacements of the end-effector caused by both the flexibility and the kinematics of the mechanism, and ${}^B\mathbf{R}_E$ is the corresponding rotation matrix.

Introducing the symbol ${}^B\mathbf{K}_E$ for the force transformation matrix, and the vector $\mathbf{w} = [\mathbf{f}^T \ \mathbf{m}^T]^T$ (*wrench*), we rewrite for convenience eq. (12) as

$${}^B\mathbf{w} = {}^B\mathbf{K}_E {}^E\mathbf{w} \quad (13)$$

3 Compensation of Force Errors

In general, the wrench computed in eq. (13) is not equal to the desired one. It is therefore required to take some corrective actions, i.e. to apply proper set-points to the position/force controllers of the manipulator joints in order to compensate for the error.

The correction of interaction force errors may be cast in a nonlinear optimization problem form. The kernel of the adopted algorithm may be related to the steepest descent method [Press et al., 1986], and it has been utilized in other fields of robotics, in particular in the solution of the inverse kinematic problem for redundant and non-redundant manipulators, see for instance [Balestrino, DeMaria and Sciavicco, 1984], [Wolovich and Elliott, 1984], [Sciavicco and Siciliano, 1986, 1988], [Das, Slotine and Sheridan, 1988].

The basic idea is the following. Define a wrench error as

$$\mathbf{e} = \begin{bmatrix} {}^B\mathbf{f}_d \\ {}^B\mathbf{m}_d \end{bmatrix} - \begin{bmatrix} {}^B\mathbf{f} \\ {}^B\mathbf{m} \end{bmatrix} = {}^B\mathbf{w}_d - {}^B\mathbf{w}$$

and a quadratic positive definite function as

$$V(\mathbf{e}) = \frac{\mathbf{e}^T \mathbf{P} \mathbf{e}}{2} \quad (14)$$

where \mathbf{P} is a symmetric positive definite matrix. Because of equations (12)-(13), the wrench ${}^B\mathbf{w}$ is a function of \mathbf{q} , so that $V(\mathbf{e})$ depends on the joint position vector. The optimal joint positions \mathbf{q} are those minimizing $V(\mathbf{q})$, and therefore we seek a control law for the arm joint positions such that the robot is driven towards the optimal configuration. Since $V(\mathbf{q})$ may be regarded as a Lyapunov function, the convergence to its minimum is guaranteed if the position control law is such that the value of V is kept decreasing along the trajectories of the system.

In the continuous time domain, the convergence to the minimum is achieved if the following Lyapunov condition is satisfied:

$$\dot{V} = \mathbf{e}^T \mathbf{P} \dot{\mathbf{e}} < 0. \quad (15)$$

Since

$$\mathbf{e} = {}^B\mathbf{w}_d - {}^B\mathbf{w} = {}^B\mathbf{w}_d - {}^B\mathbf{K}_E {}^E\mathbf{w}$$

it follows that

$$\dot{\mathbf{e}} = {}^B\dot{\mathbf{w}}_d - \frac{\partial({}^B\mathbf{K}_E {}^E\mathbf{w})}{\partial t}$$

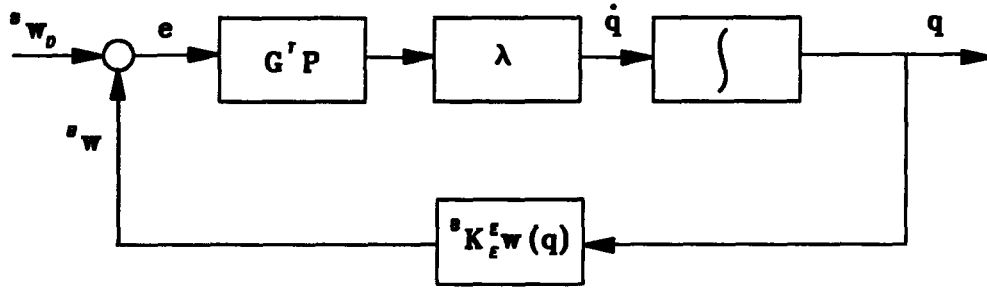


Figure 2: Block diagram of the proposed algorithm.

or

$$\dot{\mathbf{e}} = {}^B \dot{\mathbf{w}}_d - \mathbf{G} \dot{\mathbf{q}} \quad (16)$$

being $\mathbf{G} = \frac{\partial ({}^B \mathbf{K}_{\xi}^T \mathbf{w})}{\partial \mathbf{q}}$ the Jacobian matrix of ${}^B \mathbf{w}$. If the update law for the joint position is chosen as

$$\dot{\mathbf{q}} = \lambda + \left[\frac{(\mathbf{e}^T \mathbf{P} {}^B \dot{\mathbf{w}}_d)}{(\mathbf{e}^T \mathbf{P} \mathbf{G} \mathbf{G}^T \mathbf{P} \mathbf{e})} \right] \mathbf{G}^T \mathbf{P} \mathbf{e}, \quad \lambda > 0, \quad (17)$$

it can be easily shown that the condition (15) is verified.

Sciavicco and Siciliano [1988], using the same algorithm for solving the inverse kinematic problem, pointed out how eq. (17) may be, for computational convenience, simplified to

$$\dot{\mathbf{q}} = \lambda \mathbf{G}^T \mathbf{P} \mathbf{e},$$

allowing in this case the function \dot{V} to be negative-definite only outside a region of the error space containing the stability point $\mathbf{e} = 0$. In this way, the maximum tracking error is directly related to $\dot{\mathbf{x}}_d$ and inversely to λ , while in steady state, since $\dot{\mathbf{x}}_d = 0$, the tracking error will be in any case zero. In this situation, an increase in the gain λ results into a reduction of the tracking error, which may therefore be arbitrarily reduced. The resulting algorithm is shown in the form of a block diagram in Figure 2.

In discrete time, the stability proof of the algorithm is more complex. A detailed discussion of such proof has been presented in [Das, Slotine, and Sheridan, 1988] for the inverse kinematic solution of redundant manipulators. One of the major modifications in the discrete time version of the algorithm is that, in order both to obtain the maximum convergence rate for the scheme and to avoid instability problems, the gain λ has to be updated at each

sampling period T . In fact, given the discrete time version of the Lyapunov function, eq. (14), at $t = nT$

$$V_n = \frac{\mathbf{e}_n^T \mathbf{P} \mathbf{e}_n}{2}$$

the goal is now to make negative the difference $V_{n+1} - V_n$. If the joint velocities are computed at the instant $t = nT$ as

$$\dot{\mathbf{q}}_n = \lambda_n \mathbf{G}_n^T \mathbf{P} \mathbf{e}_n,$$

the convergence of the algorithm is guaranteed with the choice

$$\lambda_n = \frac{1}{T} \frac{\mathbf{e}_n^T \mathbf{P}^T \mathbf{G}_n \mathbf{S}_n \mathbf{G}_n^T \mathbf{P} \mathbf{e}_n}{\mathbf{e}_n^T \mathbf{P}^T \mathbf{G}_n \mathbf{S}_n \mathbf{G}_n^T \mathbf{P} \mathbf{G}_n \mathbf{S}_n \mathbf{G}_n^T \mathbf{P} \mathbf{e}_n}$$

where \mathbf{S}_n is a diagonal matrix whose elements are properly computed to limit the maximum values of $\mathbf{G}_n^T \mathbf{P} \mathbf{e}_n$, [Das, Slotine, and Sheridan, 1988].

A final observation relative to this scheme concerns the possibility of getting “stuck”, i.e. to generate a null joint velocity vector, $\dot{\mathbf{q}} = 0$, also if $\mathbf{e} \neq 0$. This happens when $\mathbf{P} \mathbf{e} \in \text{Null}(\mathbf{G}^T)$. However, this does not represent a serious limitation to the applicability of the algorithm. As a matter of fact, besides being an easily detectable condition ($\dot{\mathbf{q}} = 0$; $\mathbf{e} \neq 0$), it can be argued that the term $\mathbf{P} \mathbf{e}$ can be easily modified in such a way that $\mathbf{P} \mathbf{e} \notin \text{Null}(\mathbf{G}^T)$. This is performed by adding suitable constraints on the joint space (as done by Das, Slotine and Sheridan [1988]), or in the task space (as suggested by Sciavicco and Siciliano [1986]). Obviously, in this latter case if the trajectory has some components which are constantly in $\text{Null}(\mathbf{G}^T)$, no algorithm will be able to compensate for tracking errors.

4 Computation of the Matrices ${}^B \mathbf{K}_E$ and \mathbf{G}

In order to apply the algorithm, it is necessary to compute the matrices ${}^B \mathbf{K}_E$ and $\mathbf{G} = \frac{\partial ({}^B \mathbf{K}_E {}^E \mathbf{w})}{\partial \mathbf{q}}$, eq. (13), eq. (16). In this section, these two matrices are calculated for a manipulator with a generic kinematic structure with n degree-of-freedom.

4.1 Computation of the force/torque transform matrix ${}^B \mathbf{K}_E$

In the previous sections, the wrench error has been defined as

$$\mathbf{e} = {}^B \mathbf{w}_d - {}^B \mathbf{K}_E {}^E \mathbf{w}$$

where the wrench transformation matrix, see eq. (12), is defined as

$${}^B\mathbf{K}_E = \begin{bmatrix} {}^B\mathbf{R}_E & \mathbf{0} \\ {}^B\mathbf{P}_\otimes & {}^B\mathbf{R}_E \end{bmatrix} \quad (18)$$

The sub-matrices ${}^B\mathbf{R}_E$ and ${}^B\mathbf{P}_\otimes$ are in general complex functions of the compliance of the joints, of the link flexibility and of the kinematics of the manipulator. Since these quantities depend on the interaction forces and the joint position, in general it is not trivial to compute the matrix ${}^B\mathbf{K}_E$. A possible and efficient way to determine this matrix is to use a recursive method, consisting in computing the effects of the force/torque applied to the i -th link on the $(i-1)$ -th link, starting with the distal link. These effects are expressed by a transformation matrix \mathbf{K}_i , with the same structure of the transformation matrix in eq. (18), and such that

$$\mathbf{w}_{i-1} = \mathbf{K}_i \mathbf{w}_i \quad (19)$$

Therefore, the matrix ${}^B\mathbf{K}_E$ can be expressed as the product of the \mathbf{K}_i as

$${}^B\mathbf{K}_E = \mathbf{K}_1 \mathbf{K}_2 \dots \mathbf{K}_n = \prod_{i=1}^n \mathbf{K}_i \quad (20)$$

where the generic term \mathbf{K}_i , according to the hypothesis of small deformations, can be computed as

$$\mathbf{K}_i = \mathbf{K}_{i,kin}(q_i) \mathbf{K}_{i,joint}(\tau_i) \mathbf{K}_{i,flex}(\mathbf{w}_i) \quad (21)$$

where $\mathbf{K}_{i,kin}$ depends on the geometry of the link, $\mathbf{K}_{i,joint}$ on the joint compliance, and $\mathbf{K}_{i,flex}$ on the link flexibility. In the following, the expressions of these three transformation matrices are given. For the sake of simplicity, in the following discussion only rotational joints are taken into account in the kinematic structure of the manipulator: however, equivalent considerations may be done also for the case of prismatic joints.

The elements of the matrices $\mathbf{R}_{i,kin}$ and $\mathbf{P}_{\otimes,i,kin}$ are only functions of the geometric and kinematic parameters of the i -th link. Using the Denavit-Hartenberger notation, q_{i-1} is the nominal joint angle, α_i and d_i are the twist and the offset of the i -th link, and a_i represents the length of the link. The expressions of the matrices $\mathbf{R}_{i,kin}$ and $\mathbf{P}_{\otimes,i,kin}$ are

$$\mathbf{R}_{i,kin} = \begin{bmatrix} c_{q_{i-1}} & -s_{q_{i-1}} & 0 \\ s_{q_{i-1}} c_{\alpha_i} & c_{q_{i-1}} c_{\alpha_i} & -s_{\alpha_i} \\ s_{q_{i-1}} s_{\alpha_i} & c_{q_{i-1}} s_{\alpha_i} & c_{\alpha_i} \end{bmatrix} \quad (22)$$

and

$$\mathbf{P}_{\otimes,i,kin} = \begin{bmatrix} 0 & -c_{\alpha}d_i & a_i s_{q_{i-1}} - c_{q_{i-1}} s_{\alpha} d_i \\ c_{\alpha}d_i & 0 & -a_i c_{q_{i-1}} - s_{q_{i-1}} s_{\alpha} d_i \\ -a_i s_{q_{i-1}} + c_{q_{i-1}} s_{\alpha} d_i & a_i c_{q_{i-1}} + s_{q_{i-1}} s_{\alpha} d_i & 0 \end{bmatrix} \quad (23)$$

where $s_x = \sin(x)$ and $c_x = \cos(x)$, as usual in transform matrices.

The matrix $\mathbf{K}_{i,joint}$ is composed by the sub-matrices $\mathbf{R}_{i,joint}$ and $\mathbf{P}_{\otimes,i,joint}$

$$\mathbf{R}_{i,joint} = Rot(z_i, k_{c,i}\tau_i) = \begin{bmatrix} c_c & -s_c & 0 \\ s_c & c_c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$\mathbf{P}_{\otimes,i,joint} = \begin{bmatrix} 0 & 0 & a_i \sin(k_{c,i}\tau_i) \\ 0 & 0 & a_i [1 - \cos(k_{c,i}\tau_i)] \\ -a_i \sin(k_{c,i}\tau_i) & -a_i [1 - \cos(k_{c,i}\tau_i)] & 0 \end{bmatrix}$$

where $k_{c,i}$ and τ_i are the joint compliance and the torque acting on the joint, respectively.

Finally, the i -th matrix $\mathbf{K}_{i,flex}$ is composed by the submatrices $\mathbf{R}_{i,flex}$ and $\mathbf{P}_{\otimes,i,flex}$, whose elements are functions of the elastic displacements of the link. The evaluation of such displacements is not an easy task in general. Some sophisticated techniques, such as finite elements methods, can be applied profitably to this problem; in other cases, a direct calibration of displacements under known loads can be a viable solution to obtain a flexibility model for the arm. For our purposes here, however, a rather simple flexibility model for the links, considered as slender beams with constant section, can suffice. In the hypothesis of small elastic deformations, the linear and rotational displacements of the link, expressed as a 6-dimensional vector, are as shown in section 2, eq. (11):

$$\begin{bmatrix} \Delta \mathbf{d}_{i,flex} \\ \Delta \theta_{i,flex} \end{bmatrix} = \begin{bmatrix} \frac{1}{EA} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{a_i^3}{3EJ} & 0 & 0 & 0 & \frac{a_i^2}{2EJ} \\ 0 & 0 & \frac{a_i^3}{3EJ} & 0 & -\frac{a_i^2}{2EJ} & 0 \\ 0 & 0 & 0 & \frac{a_i}{GJ_o} & 0 & 0 \\ 0 & 0 & -\frac{a_i^2}{2EJ} & 0 & \frac{a_i}{EJ} & 0 \\ 0 & \frac{a_i^2}{2EJ} & 0 & 0 & 0 & \frac{a_i}{EJ} \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \mathbf{m}_i \end{bmatrix}$$

The matrix $\mathbf{R}_{i,flex}$ is computed as a rotation matrix about an axis parallel to $\delta\theta_{i,flex}$ of an angle given by the module of $\delta\theta_{i,flex}$. As pointed out in [Fresonke, Hernandez and Tesar, 1988], this approximation is valid as long as small deformations are assumed.

The recursive computation of eq. (19), starting from the distal joint for which the force/torque sensor provides ${}^E\mathbf{w}$, yields the deformations generated by the flexibility and the joint compliance, and therefore all the matrices in eq. (20), allowing the transformation of the force vector from the frame E located in the end-effector to the base frame B .

4.2 Computation of the Jacobian Matrix $\mathbf{G} = \frac{\partial {}^B\mathbf{w}}{\partial \mathbf{q}}$

The application of the algorithm presented in section 3 requires the computation of the matrix $\mathbf{G} = \frac{\partial {}^B\mathbf{w}}{\partial \mathbf{q}} = \frac{\partial ({}^B\mathbf{K}_E {}^E\mathbf{w})}{\partial \mathbf{q}}$, in which, in general, both ${}^B\mathbf{K}_E$ and ${}^E\mathbf{w}$ are functions of the joint position vector \mathbf{q} . Since the wrench transformation matrix ${}^B\mathbf{K}_E$ is a function of the joint positions, see equations (12), (13), the Jacobian matrix \mathbf{G} may be computed as

$$\begin{aligned} \mathbf{G} &= \frac{\partial ({}^B\mathbf{K}_E {}^E\mathbf{w})}{\partial \mathbf{q}} = \frac{\partial (\prod_{i=1}^n \mathbf{K}_i {}^E\mathbf{w})}{\partial \mathbf{q}} \\ &= \frac{\partial (\prod_{i=1}^n \mathbf{K}_i)}{\partial \mathbf{q}} {}^E\mathbf{w} + \left(\prod_{i=1}^n \mathbf{K}_i \right) \frac{\partial {}^E\mathbf{w}}{\partial \mathbf{q}} \\ &= \frac{\partial (\prod_{i=1}^n \mathbf{K}_i)}{\partial \mathbf{q}} {}^E\mathbf{w} + \left(\prod_{i=1}^n \mathbf{K}_i \right) \frac{\partial {}^E\mathbf{w}}{\partial \mathbf{q}}. \end{aligned}$$

In general, the computation of this matrix requires the knowledge of the manipulator-environment interaction, eq. (11), for the calculation of $\frac{\partial {}^E\mathbf{w}}{\partial \mathbf{q}}$. In following, only the case in which ${}^E\mathbf{w}$ may be considered constant in the range of motion caused by elastic displacements is taken into account. The matrix \mathbf{G} may be computed as

$$\mathbf{G} \approx \frac{\partial (\prod_{i=1}^n \mathbf{K}_i)}{\partial \mathbf{q}} {}^E\mathbf{w} = \mathbf{T} {}^E\mathbf{w}$$

where \mathbf{T} is a three-dimensional matrix composed by n \mathbf{T}_i (6x6) matrices given by

$$\mathbf{T}_i = \left(\prod_{j=1}^{i-1} \mathbf{K}_{j,kin} \mathbf{K}_{j,joint} \mathbf{K}_{j,flex} \right) \frac{\partial \mathbf{K}_{j,kin}}{\partial q_i} \left(\prod_{j=i}^n \mathbf{K}_{j,joint} \mathbf{K}_{j,flex} \mathbf{K}_{j+1,kin} \right)$$

where conventionally we assume $\mathbf{K}_{n+1,kin} = \mathbf{I}_6$. Therefore, the generic i -th column of the matrix \mathbf{G} is calculated as

$$\mathbf{G}_i = \mathbf{T}_i {}^E\mathbf{w}$$

5 Case Studies

In this section, two examples of the application of the algorithm to the force control of flexible manipulators in interaction with the environment are illustrated. One of the characteristics of the presented technique is that the algorithm allows consideration of robots with arbitrary number of degrees of freedom (both redundant and defective), as well as with arbitrary link shape. Note that in eq. (22), (23) the complete set of Denavit-Hartenberg parameters are taken into account.

The case studies address the simulation of two planar manipulators with rotational joints. The manipulators are a planar 3 degree-of-freedom and a redundant 5 degree-of-freedom planar arm. All the links of both manipulators are assumed to have the same geometrical and mechanical structure, with a circular cross-section with radius 10 mm, and a length of 1000 mm. The material elastic constants are those of common steel, while for the joints a compliance of $k_{ci} = 10^3$ (Nm/rad) has been assumed. The matrix \mathbf{P} is, in both the examples, an identity matrix with proper dimensional units. Finally, in both the cases, the presented algorithm is iterated until a value $V(\mathbf{e}) < 10^{-6}$ for the error function is reached.

5.1 A 3 Degree-of-Freedom Planar Manipulator

The manipulator considered in the first example consists of three links and three rotational joints with parallel axes. A wrench ${}^B\mathbf{w} = [0 \ 10 \ 0 \ 0 \ 0 \ 0]^T$ (N)-(Nm) is supposed to be applied at the tip of the arm. The initial - undeformed - pose of the manipulator is shown in Figure 3.a. The effects of the applied wrench on the compliant structure are reported in Figure 3.b, where the deformed configuration of the manipulator is shown. It has to be pointed out that, since the joint encoders are placed before the elastic elements, the arm appears to the control system to be in the undeformed configuration reported in Figure 3.a. In this situation, if the force set-point ${}^B\mathbf{w}_d = [0 \ 10 \ 0 \ 0 \ 0 \ 30]^T$ (N)-(Nm) is specified, a force error ${}^B\mathbf{e} = [1.1561 \ 0.067 \ 0 \ 0 \ 0 \ 0.02506]^T$ (N)-(Nm) results. In Figure 3.c the computed corrective actions on the joint positions are shown, i.e. the nominal configuration to which the arm joints have to be positioned to minimize the force error \mathbf{e} . In Figure 3.d the final (compensated) configuration of the arm is shown. In Table 1.a,b numerical results relative to this case are presented. In particular, Table 1.a reports the initial position/orientation, the measurements of the joint encoders, the initial applied wrench expressed in the base frame, the wrench error and the initial - virtual - joint rotation equivalent to the rotation/flexion of the structure. In Table 1.b the relative data after the correction are reported. The final value of the error function is $V(\mathbf{e}) < 10^{-6}$, and this result is achieved in 335 steps.

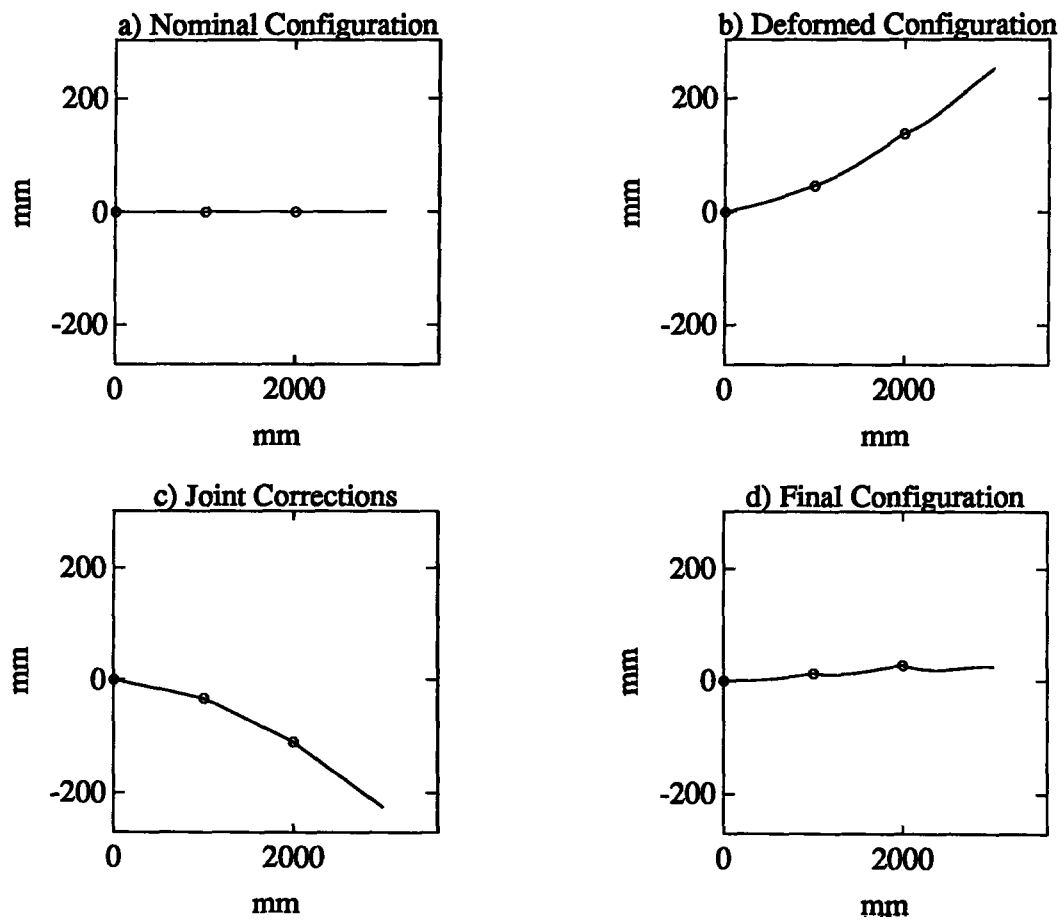


Figure 3: The four configurations of the three-link arm discussed in section 5.1.

	<i>Initial data</i>
tip position	$\mathbf{T}_{init} = \begin{bmatrix} 0.9933 & -0.1156 & 0.0000 & 2.9885 \\ 0.1156 & 0.9933 & 0.0000 & 0.2513 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$
joint encoders (deg)	$\theta_{set-point} = [0.0000 \quad 0.0000 \quad 0.0000]$
wrench	${}^B \mathbf{w}_{init} = [-1.1561 \quad 9.9330 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 29.9749]^T$
wrench error	${}^B \mathbf{e}_{init} = [1.1561 \quad 0.0670 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.0251]^T$
equivalent joint rot. (deg)	$\theta_{init} = [3.4962 \quad 2.2134 \quad 0.9288]$

Table 1.a: The initial configuration data for example 1.

	<i>Final data</i>
tip position	$\mathbf{T}_{final} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 3.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0259 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$
joint encoders (deg)	$\theta_{set-point} = \begin{bmatrix} -1.9043 & -2.4893 & -2.2469 \end{bmatrix}$
wrench	${}^B\mathbf{w}_{final} = \begin{bmatrix} 5.3 \cdot 10^{-7} & 10.0000 & 0.0000 & 0.0000 & 0.0000 & 30 \end{bmatrix}^T$
wrench error	${}^B\mathbf{e}_{final} = \begin{bmatrix} -5.3 \cdot 10^{-7} & 1.4 \cdot 10^{-14} & 0.0000 & 0.0000 & 0.0000 & 5.3 \cdot 10^{-7} \end{bmatrix}^T$
equivalent joint rot. (deg)	$\theta_{final} = \begin{bmatrix} 3.4962 & 2.2134 & 0.9288 \end{bmatrix}$

Table 1.b: The final configuration data for example 1.

5.2 A 5 Degree-of-Freedom Planar Redundant Manipulator

The second manipulator taken into consideration is a redundant planar arm. The robot, shown in Figure 4.a, is a 5 degree-of-freedom robot. A wrench ${}^E\mathbf{w} = \begin{bmatrix} 0 & 10 & 0 & 0 & 0 & 0 \end{bmatrix}^T$ (N)-(Nm) is supposed applied to the tip of the arm, as in the previous case. In this case, the force error, with a set point ${}^B\mathbf{w}_d = \begin{bmatrix} 0 & 10 & 0 & 0 & 0 & 50 \end{bmatrix}^T$ (N)-(Nm), is ${}^B\mathbf{e} = \begin{bmatrix} 2.29975 & 0.4598 & 0 & 0 & 0 & 0.3656 \end{bmatrix}^T$ (N)-(Nm). In Figure 4.a-d the initial undeformed and deformed configurations, as well as the corrective actions and the final position of the manipulator are shown. The numerical results relative to this example are reported in Table 2.a,b. In this case, because of the larger number of joints and the greater value of the initial error, a value of the error function $V(\mathbf{e}) = 9.110^{-6}$ is reached after 1427 iterations of the algorithm.

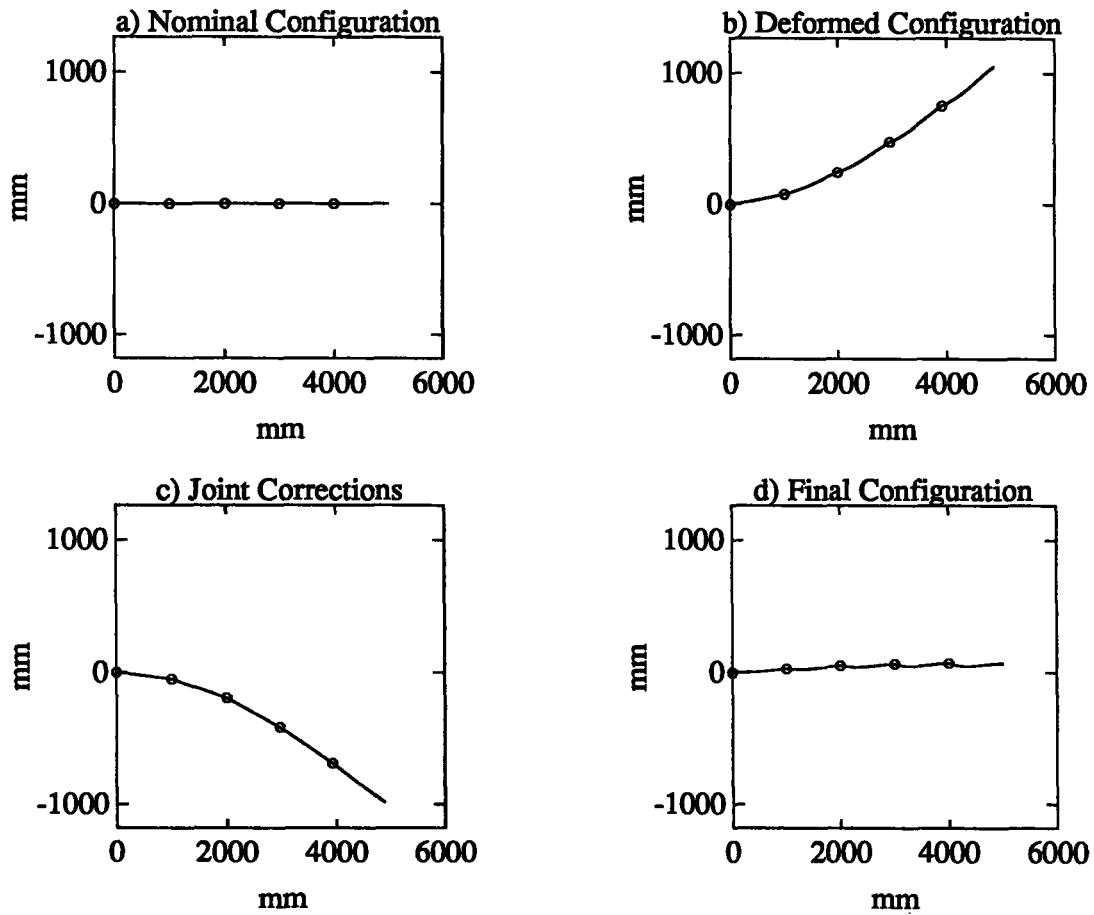


Figure 4: The four configurations of the three-link arm discussed in section 5.2.

	<i>Initial data</i>
tip position	$\mathbf{T}_{init} = \begin{bmatrix} 0.9540 & -0.2998 & 0.0000 & 4.8719 \\ 0.2998 & 0.9540 & 0.0000 & 1.0526 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$
joint encoders (deg)	$\theta_{set-point} = [0.0000 \quad 0.0000 \quad 0.0000]$
wrench	${}^B \mathbf{w}_{init} = [-2.9975 \quad 9.5402 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 49.6342]^T$
wrench error	${}^B \mathbf{e}_{init} = [2.9975 \quad 0.4598 \quad 0.0000 \quad 0.0000 \quad 0.0000 \quad 0.3657]^T$
equivalent joint rot. (deg)	$\theta_{init} = [6.0317 \quad 4.7724 \quad 3.4963 \quad 2.2134 \quad 0.9288]$

Table 2.a: The initial configuration data for example 2.

	<i>Final data</i>
tip position	$\mathbf{T}_{final} = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 5.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0720 \\ 0.0000 & 0.0000 & 1.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 1.0000 \end{bmatrix}$
joint encoders (deg)	$\theta_{set-point} = \begin{bmatrix} -2.7892 & -5.4324 & -4.5501 & -2.9964 & -1.7230 \end{bmatrix}$
wrench	$B\mathbf{w}_{final} = \begin{bmatrix} -8.8 \cdot 10^{-7} & 10.0000 & 0.0000 & 0.0000 & 0.0000 & 50 \end{bmatrix}^T$
wrench error	$B\mathbf{e}_{final} = \begin{bmatrix} 8.8 \cdot 10^{-7} & 4.1 \cdot 10^{-14} & 0.0000 & 0.0000 & 0.0000 & -2.0 \cdot 10^{-7} \end{bmatrix}^T$
equivalent joint rot. (deg)	$\theta_{final} = \begin{bmatrix} 6.0674 & 4.7829 & 3.4983 & 2.2136 & 0.9288 \end{bmatrix}$

Table 2.a: The final configuration data for example 2.

6 Conclusions

In this paper, an algorithm for the compensation of interaction force errors for flexible manipulators has been presented. Because of joint compliance and link flexibility, the problem is non-linear; the proposed solution employs the framework of non-linear optimization techniques. In fact, the presented algorithm is related to the steepest descent method, a well-known technique in this area. The method is also similar to algorithms proposed for the kinematic solution of redundant manipulators. The basic idea is to control joint positions so as to minimize a quadratic function force errors. A Lyapunov stability analysis demonstrates the stability and convergence of the method. Examples of the proposed algorithm are reported, showing the effectiveness of the technique also for the case of redundant arms. The proposed solution is used in this paper as an off-line reference generator.

Further developments of this research will investigate extensions of the method to more general problems. In particular, activity is currently in progress in the following areas:

- Modification of the method to take into account interaction forces that vary (in the end-effector reference frame) during the execution of the task, such as e.g. gravity and inertial loading;
- Modification of the iterative algorithm in order to achieve better convergence speed and trajectory smoothness, so as to allow the application of the method as an on-line (real-time) high-level force controller for flexible manipulators.

Acknowledgements

The authors wish to thank Dr. J. Kenneth Salisbury for the stimulating discussions. Support for this research is provided by the University Research Initiative Program under Office Naval Research Contract N00014-86-K-0685. Support for both the authors as Visiting Scientists at the A.I. Lab. has been provided by a NATO-CNR joint fellowship program, grants No. 215.22/07 and No. 215.23/15.

References

- Balestrino, A., De Maria, G., Sciavicco, L., "Robust Control of Robotic Manipulators", Proc. 9th. IFAC World Congress, Budapest, Hungary, 1984.
- Chiou, B.C., and Shahinpoor, M., "Stability Considerations for a two Link Force Controlled Flexible Manipulator", Proc. IEEE Int. Conf. on Robotics and Automation, May 1990.
- Das, H., Slotine, J-J.E., Sheridan, T.B., "Inverse Kinematic Algorithms for Redundant Systems", Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, CA, 1988.
- Eppinger, S.D., and Seering, W.P., "Understanding Bandwidth Limitations in Robot Force Control", Proc. IEEE Int. Conf. on Robotics and Automation, March 1987.
- Fresonke, D.A., Hernandez, E., Tesar, D., "Deflection Prediction for Serial Manipulator", Proc. IEEE Int. Conf. on Robotics and Automation, 1988.
- Korolov, V.V., and Chen, Y.H.: "Robust Control of a Flexible Manipulator Arm", Proc. IEEE Conf. Robotics and Automation, Philadelphia, PA, 1988.
- Press, W.H., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T., "Numerical Recipes: The Art of Scientific Computing", Cambridge University Press, 1986.
- Roberts, R.K., Paul, R.P., and Hillberry, B.M., "The Effect of Wrist Sensor Stiffness on the Control of Robot Manipulators", Proc IEEE Int. Conf. on Robotics and Automation, March 1985.
- Sciavicco, L., Siciliano, B., "Coordinate Transformation: A Solution Algorithm for One Class of Robots", IEEE Trans. Syst., Man, Cybern., Vol. SMC-16, No. 4, July/Aug. 1986.
- Sciavicco, L., Siciliano, B., "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulator", IEEE Jour. of Robotics and Automation, Vol. 4, No. 4, Aug. 1988.
- Whitney, D.E., "Historical Perspective and State of the Art in Robot Force Control", Int. Jour. of Robotic Research, Vol. 6, No. 1, Spring 1987.
- Wolovich, W.A., Elliott, H., "A Computational Technique for Inverse Kinematics", Proc. 23rd. Conf. on Decision and Control, Las Vegas, NV, Dec. 1984.