

Trajectory Control of Flexible Manipulators

Alessandro De Luca

Dipartimento di Informatica e Sistemistica, Università degli Studi di Roma
"La Sapienza", Italy

We present some feedback control techniques recently developed for the exact solution of trajectory tracking problems for manipulators with flexible elements. Two classes are considered: *i*) robots with rigid links but with elastic transmissions, in which flexibility is concentrated at the joints, and *ii*) robots with lightweight and/or long arms, where flexibility is distributed along the links. For robots with elastic joints, we introduce a generalized inversion algorithm for the synthesis of a dynamic feedback control law that gives input-output decoupling and full state linearization. For robots with flexible links, the end-effector trajectory tracking problem is solved based on the iterative computation of the link deformations associated with the desired output motion, combined with a state trajectory regulator. For both robot models, the control design is performed directly on the second-order dynamic equations.

1. Introduction

Modeling robot manipulators as rigid mechanical systems is an idealization that becomes unrealistic when higher performance is requested. Tasks involving fast motion and/or hard contact with the environment are expected to induce deflections in the robot components, eventually exciting an oscillatory behavior. There are two sources of vibration in robot manipulators: *joint flexibility*, due to the elasticity of motion transmission elements such as harmonic drives, belts, or long shafts [26], and *link flexibility*, introduced by a long reach and slender/lightweight construction of the arm [6, 17]. In order to be able to counteract the negative effects of flexibility, advanced robot control systems should be designed on the basis of a more complete dynamic model of the robot (see, e.g. [27] and [5])

In robotic systems with flexible elements, output trajectories are typically defined beyond the structural flexibility, i.e. in terms of link motion for robots with elastic joints or at the level of manipulator tip for robots with link flexibility. We address in this chapter the stable and accurate reproduction of such trajectories using model-based state feedback control. Standard tools for solving trajectory tracking problems in nonlinear systems, such as feedback linearization, input-output decoupling, or inversion control (see, e.g. [19]), are not sufficient in these cases, so that the application of more advanced control techniques should be investigated.

In particular, the complete dynamic model of robots with elastic joints fails to satisfy the necessary conditions for input-output decoupling and/or full linearization by static state feedback [15], as opposed to the case of rigid robots for which these methods are equivalent to the well-known computed torque technique. Use of the larger class of *dynamic state feedback* controllers is helpful, because elastic joint robots are in fact input-output invertible systems without zero dynamics. However, the linearizing dynamic compensator has been derived so far only for very simple manipulators, while a general synthesis method is still missing. In Sect. 2. we will provide a constructive answer to this problem. Furthermore, we will be able to characterize in a precise way a tight upper bound for the dimension of the needed dynamic compensator.

On the other hand, the mapping between the joint torque input and the end-effector position output in robots with flexible links is associated with an unstable zero dynamics [14], the nonlinear equivalent of non-minimum phase zeros in a linear setting. The straightforward application of inversion-based control leads in this case to an unbounded increase of the internal arm deformation and, eventually, to control explosion. Different approaches have been presented in order to overcome this problem while exactly tracking the desired tip motion of multi-link manipulators: inversion in the frequency domain, iterative learning control, nonlinear regulation, or a combination of these. In all cases, the key feature is the computation of the *bounded link deformations* (and of the joint motions) producing the desired trajectory of the manipulator tip. Based on this idea, Sect. 3. presents in a unified framework three different but rather equivalent solutions to the end-effector trajectory tracking problem. We report also some illustrative experimental results obtained on a prototype two-link planar manipulator with flexible forearm, available in our Robotics Laboratory [10].

The chapter is organized so that its two main parts, devoted respectively to elastic joint and flexible link manipulators, are self-contained and independent.

2. Robots with Elastic Joints

We study the dynamic feedback linearization problem for robot arms with elastic joints. In particular, we consider a specific class of dynamic models which is, however, general enough so as to include many interesting instances, like robots moving on a plane. All robots within this class cannot be linearized nor input-output decoupled using only static state feedback. The design of the dynamic control law is presented in a constructive way, without resorting to state-space equations. The obtained result enables to solve the trajectory tracking problem in a global sense and with a prescribed linear error dynamics.

2.1 Dynamic Modeling

Consider an open kinematic chain of $N + 1$ rigid bodies, interconnected by N joints undergoing elastic deformation. The robot is actuated by electrical drives which are assumed to be located at the joints. Let $q \in \mathbb{R}^N$ be the link positions, and $\theta \in \mathbb{R}^N$ be the motor (i.e. rotor) positions, as reflected through the gear ratios. With this choice, the difference $q_i - \theta_i$ is the i th joint deformation and the direct kinematics of the whole arm will be a function of the link variables q only. The following quite general assumptions are made:

Assumption 2.1. Joint deformations are small, so that elasticity in the joint is modeled as a linear spring.

Assumption 2.2. The rotors of the motors are modeled as uniform bodies having their center of mass on the rotation axis.

Assumption 2.2 implies that both the inertia matrix and the gravity term in the dynamic model will be independent from the position θ of the motors.

Following the Lagrangian approach, we compute the *kinetic energy* of the robot structure (including links and motors as rigid bodies) as

$$T = \frac{1}{2} \begin{bmatrix} \dot{q}^T & \dot{\theta}^T \end{bmatrix} \begin{bmatrix} B(q) & S(q) \\ S^T(q) & J \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{\theta} \end{bmatrix}, \quad (2.1)$$

where all blocks of the inertia matrix are $N \times N$ matrices: $B(q)$ contains the inertial properties of the rigid links, $S(q)$ accounts for the inertial couplings between motors and links, while $J = \text{diag}\{J_1, \dots, J_n\}$, $J_i > 0$, is the matrix of the effective rotor inertias of the motors.

Consider the standard case in which the i th motor is mounted on link $i - 1$ and moves link i . Since the kinetic energy of the i th motor does not depend on the motion of the i th link and of the subsequent ones, we have the following strong model property:

Property 2.1. Matrix $S(q)$ has the upper triangular structure

$$\begin{bmatrix} 0 & S_{12}(q_1) & S_{13}(q_1, q_2) & \cdots & S_{1N}(q_1, \dots, q_{N-1}) \\ 0 & 0 & S_{23}(q_2) & \cdots & S_{2N}(q_2, \dots, q_{N-1}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & S_{N-1,N}(q_{N-1}) \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}, \quad (2.2)$$

where the most general cascade dependence is shown for each single term.

For the ease of presentation, we will focus on a particular situation:

Assumption 2.3. Matrix S in Eq. (2.1) is constant.

For instance, Assumption 2.3 is valid for planar robots with any number of rotational joints or for a spatial 3R elbow manipulator. In the former case, it can be shown that the expression of the elements of S is $S_{ij} = J_j$, apart from those entries that are structurally zero.

The *potential energy* is given by the sum of the gravitational energy, for both motors and links, and of the elastic energy stored at the joints. By virtue of Assumptions 2.1 and 2.2, we have

$$U = U_g(q) + \frac{1}{2}(q - \theta)^T K(q - \theta), \quad (2.3)$$

in which $K = \text{diag}\{K_1, \dots, K_n\}$, $K_i > 0$ being the elastic constant of joint i .

The robot dynamic model is obtained from the Euler-Lagrange equations for the Lagrangian $L = T - U$. Under the above assumptions, the $2N$ second-order differential equations have the form (see, e.g. [15] for a detailed derivation)

$$B(q)\ddot{q} + S\ddot{\theta} + c(q, \dot{q}) + g(q) + K(q - \theta) = 0 \quad (2.4)$$

$$S^T \ddot{q} + J\ddot{\theta} + K(\theta - q) = \tau, \quad (2.5)$$

where $c(q, \dot{q})$ are Coriolis and centrifugal terms, $g(q) = (\partial U_g / \partial q)^T$ are gravity terms, and $\tau \in \mathbb{R}^N$ are the torques supplied by the motors.

We note explicitly that in the case of a single link and for some other special kinematic structures with elastic joints (e.g. a 2R polar robot) it is found that $S = 0$, implying no inertial couplings between the link and the motor dynamics. The same situation is forced by a modeling assumption introduced in [25], namely by considering in the angular part of the kinetic energy of each rotor only the part due to its relative rotation. When $S = 0$, the model is always feedback linearizable by *static* state feedback. In the following, we will assume that at least one element in matrix S is different from zero. In this case, the control property of linearization by static feedback is always destroyed and we should look for a more general *dynamic* feedback controller in order to achieve full state linearization and input-output decoupling.

2.2 Generalized Inversion Algorithm

Let $q_d(t)$ be a desired smooth reference trajectory for the link variables q . A dynamic state feedback control for the input torques τ in Eq. (2.5) is a law of the form

$$\tau = \alpha(x, \xi) + \beta(x, \xi)v \quad (2.6)$$

$$\dot{\xi} = \gamma(x, \xi) + \delta(x, \xi)v \quad (2.7)$$

where $x = (q, \theta, \dot{q}, \dot{\theta}) \in \mathbb{R}^{4N}$ is the state of the robot, $\xi \in \mathbb{R}^M$ is the state of the dynamic compensator (of order M to be defined), and $v \in \mathbb{R}^N$ is the new

control input. Our objective is to design such a control law so that the closed-loop system made by Eqs. (2.4)–(2.5) and (2.6)–(2.7) is fully represented by decoupled chains of input-output integrators, i.e.

$$\frac{d^{r_i} q_i}{dt^{r_i}} = v_i, \quad i = 1, \dots, N, \quad (2.8)$$

with the additional requirement that

$$\sum_{i=1}^N r_i = 4N + M, \quad (2.9)$$

where r_i the closed-loop relative degree of the output variable q_i .

This problem formulation asks for both input-output decoupling and full state linearization (in the proper coordinates) of the closed-loop system. Sufficient conditions for the existence of a solution to this problem exist [20] and general algorithms for constructing the required control law can be found in [19].

It has been shown in [9] that the general model of robots with elastic joints (and thus, in particular, Eqs. (2.4)–(2.5)) can be linearized and input-output decoupled via dynamic state feedback. However, the actual construction of the dynamic controller has been a difficult task until now and was performed only on a case-by-case basis. An example of such a controller for a 2R planar robot can be found in [7]. One difficulty in deriving a systematic method for the synthesis of the controller (2.6)–(2.7) for robots with elastic joints is due to the fact that all available algorithms are defined in terms of a state-space representation of the system. The transformation of Eqs. (2.4)–(2.5) into first-order state equations, though simple, hides the physical role of the following algorithmic steps and makes them computationally more complex. In addition, it has been found [8] that the dimension M of the dynamic controller and the degrees r_i of the obtained linear input-output relations (2.8) depend on the number of joints as well as on the kinematic structure of the robot.

With the above limitations in mind, we propose a new general algorithm that proceeds in an incremental way by solving a series of partial linearization and input-output decoupling problems, directly defined on the robot dynamic model (2.4)–(2.5).

2.2.1 Step 1: Input-output decoupling with respect to θ . From the structure of Eq. (2.5), we define the following control law for τ

$$\tau = Ju + S^T \ddot{q} + K(\theta - q) \quad (2.10)$$

where $u \in \mathbb{R}^N$ is the new control input. This control law imposes the dynamics

$$B(q)\ddot{q} + Su + c(q, \dot{q}) + g(q) + K(q - \theta) = 0 \quad (2.11)$$

$$\ddot{\theta} = u. \quad (2.12)$$

The implementation of the control law (2.10) by state feedback requires the elimination of the link acceleration \ddot{q} . Solving for \ddot{q} in Eq. (2.11) and substituting in Eq. (2.10) gives

$$\tau = [J - S^T B^{-1}(q)S] u - S^T B^{-1}(q) [c(q, \dot{q}) + g(q) + K(q - \theta)] + K(\theta - q). \quad (2.13)$$

Equation (2.12) shows that a linear and decoupled relation has been obtained between each input component u_i and each output θ_i ($i = 1, \dots, N$), by using a static state feedback law $\tau = \tau(q, \theta, \dot{q}, u)$. In the closed-loop system, we have $2N$ states (namely, q and \dot{q}) that are unobservable from the output θ .

2.2.2 Step 2: Input-output decoupling with respect to f . By defining a new output f as

$$f = B(q)\ddot{q} + c(q, \dot{q}) + g(q) + Kq, \quad (2.14)$$

Equation (2.4) can be rewritten as

$$f(q, \dot{q}, \ddot{q}) + Su - K\theta = 0, \quad (2.15)$$

where Eq. (2.12) has been used. We note that output f has the dimension of a generalized force.

Differentiating twice Eq. (2.15), we obtain

$$\ddot{f}(q, \dot{q}, \ddot{q}) + S\ddot{u} - Ku = 0. \quad (2.16)$$

By defining the following control law for u

$$u = K^{-1} [S\ddot{u} + w'], \quad (2.17)$$

where $w' \in \mathbb{R}^N$ is the new control input, we would simply get

$$\ddot{f}(q, \dot{q}, \ddot{q}) = w', \quad (2.18)$$

i.e. a linear and decoupled relation between each input w'_i and each output f_i ($i = 1, \dots, N$).

Owing to the Property 2.1 of matrix S , the control law (2.17) inherits a hierarchical structure and is thus well defined, even if its implementation requires input differentiation. To avoid input differentiation, we proceed in a different way by adding on each input channel u_i a string of integrators. In particular, $2(i - 1)$ integrators, with states ϕ_{ij} , are put on the i th channel ($i = 2, \dots, N$; $j = 1, \dots, 2(i - 1)$):

$$\begin{aligned}
 u_1 &= \bar{w}_1 \\
 u_2 &= \phi_{21}, \quad \dot{\phi}_{21} = \phi_{22}, \quad \dot{\phi}_{22} = \bar{w}_2 \\
 &\vdots \\
 u_i &= \phi_{i1}, \quad \dot{\phi}_{i1} = \phi_{i2}, \quad \dots \quad \dot{\phi}_{i,2(i-1)} = \bar{w}_i \\
 &\vdots \\
 u_N &= \phi_{N1}, \quad \dot{\phi}_{N1} = \phi_{N2}, \quad \dots \quad \dots \quad \phi_{N,2(N-1)} = \bar{w}_N,
 \end{aligned} \tag{2.19}$$

where $\bar{w} \in \mathbb{R}^N$ is a temporary control input. The total number of added integrators is $N(N-1)$. Denote by ϕ the vector collecting the states of all these integrators.

Differentiating $2(i-1)$ times the i th scalar equation in (2.16) (or, equivalently, $2i$ times the i th equation in (2.15)), and keeping into account the dynamic extension (2.19), we obtain

$$\begin{aligned}
 \frac{d^{2i} f_i}{dt^{2i}} &= - \sum_{j=i+1}^N S_{ij} \frac{d^{2i} u_j}{dt^{2i}} + K_i \frac{d^{2(i-1)} u_i}{dt^{2(i-1)}} \\
 &= -S_{i,i+1} \bar{w}_{i+1} - \sum_{j=i+2}^N S_{ij} \phi_{j,2i+1} + K_i \bar{w}_i,
 \end{aligned} \tag{2.20}$$

for $i = 1, \dots, N$. By defining recursively the following control law for \bar{w}

$$\begin{aligned}
 K_N \bar{w}_N &= w_N \\
 K_{N-1} \bar{w}_{N-1} &= S_{N-1,N} \bar{w}_N + w_{N-1} \\
 K_i \bar{w}_i &= S_{i,i+1} \bar{w}_{i+1} + \sum_{j=i+2}^N S_{ij} \phi_{j,2i+1} + w_i \\
 &\quad (i = N-2, N-3, \dots, 1),
 \end{aligned} \tag{2.21}$$

we obtain

$$\frac{d^{2i} f_i}{dt^{2i}} = w_i, \quad i = 1, \dots, N. \tag{2.22}$$

Equation (2.22) shows again a linear and decoupled relation between each input w_i and each output f_i ($i = 1, \dots, N$), resulting now from the application of the linear dynamic compensator $u = u(\phi, w)$ obtained through Eqs. (2.19) and (2.21). Indeed, when combining this compensator with Eq. (2.13), a nonlinear dynamic state feedback $\tau = \tau(q, \theta, \dot{q}, \phi, w)$ is defined for the original robot torque input. Note also that the total number of states of the robot and of the compensator is $4N + N(N-1) = N(N+3)$ whereas, from Eqs. (2.22), the number of states on the input-output channels is $N(N+1)$. Therefore, in the closed-loop system we have still $2N$ states that are unobservable from the output f .

2.2.3 Step 3: Input-output decoupling with respect to q . As the last algorithmic step, we tackle the input-output decoupling and linearization problem for the original output q . The mapping from f to q , represented by Eq. (2.14), contains the main nonlinearities of the robot link dynamics. In order to cancel them in a well-defined way, we need to dynamically balance the input-output relations in Eqs. (2.22). In fact, differentiating $2(N - i)$ times the i th equation in (2.22) we get

$$\frac{d^{2(N-i)}}{dt^{2(N-i)}} \frac{d^{2i} f_i}{dt^{2i}} = \frac{d^{2N}}{dt^{2N}} \left(b_i^T(q) \ddot{q} + c_i(q, \dot{q}) + g_i(q) + K_i q_i \right) = \frac{d^{2(N-i)} w_i}{dt^{2(N-i)}}, \quad (2.23)$$

for $i = 1, \dots, N$, where $b_i(q)$ the i th column of the link inertia matrix $B(q)$. To avoid differentiation of the input w , we add $2(N - i)$ integrators, with states ψ_{ij} , on the i th channel ($i = 1, \dots, N - 1; j = 1, \dots, 2(N - i)$):

$$\begin{aligned} w_N &= \bar{v}_N \\ w_{N-1} &= \psi_{N-1,1}, & \dot{\psi}_{N-1,1} &= \psi_{N-1,2}, & \dot{\psi}_{N-1,2} &= \bar{v}_{N-1} \\ &\vdots & &\vdots & &\ddots \\ w_i &= \psi_{i1}, & \dot{\psi}_{i1} &= \psi_{i2}, & \dots & \dot{\psi}_{i,2(i-1)} = \bar{v}_i \\ &\vdots & &\vdots & &\ddots \\ w_1 &= \psi_{11}, & \dot{\psi}_{11} &= \psi_{12}, & \dots & \psi_{1,2(N-1)} = \bar{v}_1 \end{aligned} \quad (2.24)$$

where $\bar{v} \in \mathbb{R}^N$ is a temporary control input. The total number of integrators is again $N(N - 1)$. Denote by ψ the vector collecting the states of all these integrators.

Resume the vector notation and rewrite Eqs. (2.23), using Eqs. (2.24), as

$$\frac{d^{2N}}{dt^{2N}} \left(B(q) \ddot{q} + c(q, \dot{q}) + g(q) + Kq \right) = \bar{v}. \quad (2.25)$$

Performing differentiation term by term gives

$$B(q) q^{\{2(N+1)\}} + n(q, \dot{q}, \dots, q^{\{2N+1\}}) = \bar{v}, \quad (2.26)$$

where

$$n = \sum_{k=1}^{2N} \binom{2N}{k} B^{\{k\}}(q) q^{\{2(N+1)-k\}} + c^{\{2N\}}(q, \dot{q}) + g^{\{2N\}}(q) + Kq^{\{2N\}}, \quad (2.27)$$

and we have used the compact notation $x^{\{i\}} = d^i x / dt^i$. Therefore, by defining the linearizing control law

$$\bar{v} = B(q)v + n(q, \dot{q}, \dots, q^{\{2N+1\}}), \quad (2.28)$$

we finally obtain

$$\frac{d^{2(N+1)}q_i}{dt^{2(N+1)}} = v_i, \quad i = 1, \dots, N. \quad (2.29)$$

Note that Eq. (2.28) can be seen a generalization of the computed torque method for rigid robots and is globally defined thanks to the positive definiteness of the link inertia matrix $B(q)$.

Input-output decoupling and linearization has been achieved by means of the nonlinear dynamic feedback $w = w(q, \dot{q}, \dots, q^{\{2N+1\}}, \psi, v)$, obtained from Eqs. (2.24) and (2.28). Note that the dependence of this control law on \ddot{q} and on higher derivatives can be eliminated recursively, in terms of the robot states $(q, \theta, \dot{q}, \dot{\theta})$ and of the compensator states (ϕ, ψ) .

Define the total state of the dynamic compensator as $\xi = (\phi, \psi)$, which is of dimension $M = 2N(N - 1)$. By combining Eqs. (2.13), (2.19), (2.21), (2.24), and (2.28) we obtain a nonlinear dynamic state feedback control law $\tau = \tau(x, \xi, v)$ with the structure (2.6)–(2.7). Furthermore, Eqs. (2.29) are in the form (2.8) with uniform relative degrees $r_i = 2(N+1)$, for all $i = 1, \dots, N$. Condition (2.9) on the sum of the relative degrees, which guarantees full state linearization beside input-output decoupling, is fulfilled. In fact, the number of states on the input-output channels $(2N(N + 1))$ equals the sum of the number of states of the robot $(4N)$ and of the compensator $(2N(N - 1))$. Thus, we have no more unobservable states left in the closed-loop system, which is in turn completely described by the linear dynamics (2.29).

A number of final remarks are in order.

Remark 2.1. The stable tracking of output reference trajectories $q_{di}(t)$ ($i = 1, \dots, N$) is realized by any standard control technique for linear single input-single output systems. Using, e.g. pole assignment, we design

$$v_i = q_{di}^{\{2(N+1)\}} + \sum_{j=0}^{2N+1} a_{ij} \left(q_{di}^{\{i\}} - q_i^{\{i\}} \right), \quad i = 1, \dots, N, \quad (2.30)$$

where the a_{ij} 's are coefficients of Hurwitz polynomials

$$s^{2(N+1)} + a_{i,2N+1}s^{2N+1} + \dots + a_{i2}s^2 + a_{i1}s + a_{i0}, \quad i = 1, \dots, N, \quad (2.31)$$

having prescribed roots in the complex right-half plane. From Eqs. (2.30) it also follows that perfect tracking requires $2(N + 1)$ -times differentiable trajectories (degree of smoothness).

Remark 2.2. In the implementation of the above tracking controller based on linearization and input-output decoupling via dynamic feedback, the main computational effort is concentrated in the evaluation of the term (2.27), which in turn requires the explicit expressions of the linearizing coordinates $q^{\{i\}}$ ($i = 2, \dots, 2N + 1$) (see also Eqs. (2.30)). These computations are easily customized for a specific robot arm since all components of the control law are defined in terms of the available dynamic model elements. Moreover, we require to invert the link inertia matrix $B(q)$ only once. This inverse

can be stored and then used repeatedly in computing the expressions of the linearizing coordinates.

Remark 2.3. We have implicitly assumed that all the strictly upper triangular elements of matrix S in Eq. (2.4) are different from zero. If some of these elements vanish, the dimension of the required dynamic compensator will decrease together with the lengths of the input-output integrators chains (2.8). These output relative degrees may also not be equal to each other. Therefore, the value $M = 2N(N - 1)$ is in general only an upper bound to the dimension of the linearizing dynamic controller, in agreement with the results obtained in [8]. For the planar $2R$ robot with elastic joints considered in [7], we have $N = 2$ and a constant non-zero value ($S_{12} = J_2$) for the single nontrivial element in matrix S . The upper bound is then attained in this case: a dynamic compensator of order 4 leads to two chains of 6 input-output integrators.

Remark 2.4. It is a simple exercise to verify that, when $S = 0$, the three steps of the above algorithm build up the static feedback linearizing controller of [25]. In particular, the dynamic extensions in Eqs. (2.19) and (2.24) vanish.

3. Robots with Flexible Links

We consider the inverse dynamics problem for robot arms with flexible links, i.e. the computation of the input torque that allows exact tracking of a trajectory defined for the manipulator end-effector. We restrict ourselves to finite-dimensional dynamic models of flexible manipulators. A stable inversion controller is derived numerically, based on the computation of bounded link deformations and, from these, of the required feedforward torque associated with the desired tip motion. Three different algorithms are presented for this computation, all defined on the second-order robot dynamic equations. Stable trajectory tracking is then obtained by adding a (partial) state feedback, within a nonlinear regulation approach.

3.1 Dynamic Modeling

Consider an open kinematic chain structure, with a fixed base and N moving flexible links, interconnected by N (rigid) rotational joints. Each link deformation is distributed in nature and would be best described by an infinite-dimensional model, typically that of an Euler beam with proper boundary conditions at the two ends [23, 2]. However, for all but the most simple structures, it is impossible to solve for the exact time evolution of the arm deflection. Therefore, the use of approximated finite-dimensional models is preferred for multi-link flexible manipulators.

In the following, some simplifying assumptions are made:

Assumption 3.1. Link deformations are small, so that only linear elastic effects are present.

Assumption 3.2. For each link, flexibility is limited to the plane of nominal rigid motion, i.e. the plane normal to the preceding joint axis.

Assumption 3.2 implies that each link can only bend in one lateral direction, being stiff with respect to axial forces and to torsion. In view of this, the bending deformation $w_i(x_i, t)$ at a generic point $x_i \in [0, \ell_i]$ along the i th link of length ℓ_i is modeled, using separation in time and space, as

$$w_i(x_i, t) = \sum_{j=1}^{N_{ei}} \phi_{ij}(x_i) \delta_{ij}(t), \quad i = 1, \dots, N, \quad (3.1)$$

where the N_{ei} spatial components $\phi_{ij}(x_i)$ are *assumed modes* of deformation satisfying geometric and/or dynamic boundary conditions, while $\delta_{ij}(t)$ are the associated generalized coordinates.

Let $\theta \in \mathbb{R}^N$ be the vector of joint angular positions, and $\delta \in \mathbb{R}^{N_e}$ the vector of link deformations, where $N_e = \sum_{i=1}^N N_{ei}$. The arm kinematics and its kinetic and potential energy can be described in terms of θ , δ , and their first derivatives. The Euler-Lagrange equations provide the dynamic model of an N -link flexible manipulator in the form of $N + N_e$ second-order differential equations (see, e.g. [5] for details):

$$\begin{bmatrix} B_{\theta\theta}(\theta, \delta) & B_{\theta\delta}(\theta, \delta) \\ B_{\theta\delta}^T(\theta, \delta) & B_{\delta\delta}(\theta, \delta) \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} c_{\theta}(\theta, \delta, \dot{\theta}, \dot{\delta}) \\ c_{\delta}(\theta, \delta, \dot{\theta}, \dot{\delta}) \end{bmatrix} + \begin{bmatrix} g_{\theta}(\theta, \delta) \\ g_{\delta}(\theta, \delta) \end{bmatrix} + \begin{bmatrix} 0 \\ D\dot{\delta} + K\delta \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}. \quad (3.2)$$

The positive-definite symmetric inertia matrix B is partitioned in blocks according to the rigid and flexible components, c is the vector of Coriolis and centrifugal forces, g is the vector of gravitational forces, $K > 0$ and $D \geq 0$ are diagonal matrices, of dimensions $N_e \times N_e$, representing the arm modal stiffness and damping, while τ is the torque at the joints. Note that no input torque appears in the right-hand side of the last N_e equations (3.2), because link deformations in Eqs. (3.1) are described in the reference frames clamped at each link base.

Remark 3.1. The model structure (3.2) holds for any finite-dimensional approximation of distributed flexibility. However, specific choices for the assumed modes ϕ_{ij} may imply convenient simplifications in the block $B_{\delta\delta}$ of the inertia matrix. In particular, orthonormality of the modes of each link induces a decoupled structure for the diagonal inertia subblocks of $B_{\delta\delta}$, which in turn may collapse into a constant diagonal matrix.

Remark 3.2. A rather common approximation is to evaluate the total kinetic energy of the system in the undeformed configuration $\delta = 0$. This implies that the inertia matrix B , and thus also the Coriolis and centrifugal terms

c , are independent of δ , and that the velocity terms c_δ lose the quadratic dependence on $\dot{\delta}$. If, in addition, $B_{\delta\delta}$ is constant, also c_θ loses the quadratic dependence on $\dot{\delta}$, while each component of c_δ becomes a quadratic function of $\dot{\theta}$ only.

For the sake of simplicity, we will consider in the next section the following dynamic model for control design

$$\begin{bmatrix} B_{\theta\theta}(\theta) & B_{\theta\delta}(\theta) \\ B_{\theta\delta}^T(\theta) & B_{\delta\delta} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} c_\theta(\theta, \dot{\theta}, \dot{\delta}) \\ c_\delta(\theta, \dot{\theta}) \end{bmatrix} + \begin{bmatrix} 0 \\ D\dot{\delta} + K\delta \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}, \quad (3.3)$$

where gravity effects are not included and the previous remarks have been taken into account. In particular, the dependence of c_θ on $\dot{\delta}$ is linear.

Since our objective is the tracking of an end-effector trajectory, we conveniently define as system output

$$y = \theta + \Phi_e \delta, \quad (3.4)$$

where the constant $N \times N_e$ matrix Φ_e is defined as

$$\Phi_e = \text{block diag} \{ \phi_{i1}(\ell_i)/\ell_i, \dots, \phi_{i,N_{ei}}(\ell_i)/\ell_i \}. \quad (3.5)$$

The output y_i is a linear approximation of the angle pointing from the i th link base to its end. According to Assumption 3.2, the direct kinematics of the flexible manipulator, i.e. the position and orientation of the arm tip, can be written only in terms of the components of y .

We finally point out that, in the presence of uniform mass distribution for each link, any dynamic model of the form (3.2) (or (3.3)) retains the same relevant control feature: the zero dynamics associated with output (3.4) is always unstable (see, however, [12]).

3.2 Stable Inversion Control

Inversion control is effective for tracking joint trajectories of a flexible link manipulator (i.e. with $y = \theta$ in place of Eq. (3.4)), because the zero dynamics is stable in this case [13]. The direct extension of an inversion control law to the tip output (3.4) leads to closed-loop instabilities, due to non-admissible feedback cancellation effects.

Frequency domain inversion has been proposed in [3, 4] as one of the first solutions to this instability problem. By working in the Fourier domain, this method defines the required open-loop control torque in one step (for linear models of one-link flexible arms) or in few iterations (in multi-link manipulators). *Learning control* has been applied in [24, 22] for iteratively building the input torque over repeated trials on the same desired output trajectory. In both approaches, the generated torque is noncausal: a non-zero input is applied in time *before* the actual start of the output trajectory. This preloading effect brings the flexible manipulator in the proper initial

state that enables reproduction of the desired trajectory, while preserving an overall bounded link deformation. *Nonlinear regulation* has been used in [11, 21]; asymptotic output tracking is obtained by closing a stabilizing state feedback around the reference state trajectory. Finally, separation of stable and unstable zero dynamics and noncausal operation are the main features of the *stable inversion* approach proposed in [28, 16].

All the above methods share a common idea: in order to exactly reproduce an end-effector trajectory, the links of a flexible manipulator should experience a specific output-related bounded deformation history. Any attempt to control the arm deformation in a different way, e.g. trying to reduce as much as possible link deformation like in vibration damping control [14], destroys exact tracking and/or induces closed-loop instability.

Let $y_d(t)$ be a desired smooth reference trajectory for the tip, defined in a closed finite interval $[0, T]$. From Eq. (3.4), we can eliminate θ and $\dot{\theta}$ in the last N_e equations of (3.3), obtaining

$$B_{\delta\delta}\ddot{\delta} + D\dot{\delta} + K\delta + B_{\theta\delta}^T(y - \Phi_e\delta) \left(\ddot{y} - \ddot{\Phi}_e\delta \right) + c_\delta(y - \Phi_e\delta, \dot{y} - \dot{\Phi}_e\delta) = 0, \quad (3.6)$$

which is a dynamic constraint to be always satisfied by tip motion and link deformations. Plugging the desired evolution $y_d(t)$ in Eq. (3.6) gives a set of nonlinear differential equations for the only unknown function $\delta(t)$. Suppose that a bounded solution $\delta_d(t)$ can be found (together with its first and second time derivatives). We can use then the first N equations in (3.3) for defining the nominal input torque

$$\tau_d = B_{\theta\theta}(\theta_d)\ddot{\theta}_d + B_{\theta\delta}(\theta_d)\ddot{\delta}_d + c_\theta(\theta_d, \dot{\theta}_d, \dot{\delta}_d), \quad (3.7)$$

where

$$\theta_d(t) = y_d(t) - \Phi_e\delta_d(t) \quad (3.8)$$

is the required joint motion. As a result, the main bottleneck is the computation of a bounded solution $\delta_d(t)$ to Eq. (3.6) evaluated at $y = y_d(t)$. In the following, we present three alternative numerical methods.

3.2.1 Method 1: Approximate nonlinear regulation. In nonlinear output regulation [19], the control law is formed by two contributions: a feedforward term driving the system output along its desired evolution, and a state feedback term necessary to stabilize the closed-loop dynamics around the reference state trajectory. For a flexible link manipulator, the feedforward term is given by Eq. (3.7) while the desired link deformation $\delta_d(t)$ is part of the reference state trajectory to be computed.

The output reference trajectory should be generated by an exosystem with state denoted by Y_d . Each component of the reference state trajectory will be specified as a nonlinear function of the exosystem state Y_d . For a flexible manipulator it is then sufficient to determine $\delta_d = \pi(Y_d)$ and $\dot{\delta}_d = (\partial\pi/\partial Y_d)\dot{Y}_d$.

In particular, the vector function $\pi(Y_d)$ should satisfy Eq. (3.6), evaluated along the reference output evolution:

$$B_{\delta\delta}\ddot{\pi}(Y_d) + D\dot{\pi}(Y_d) + K\pi(Y_d) + B_{\theta\delta}^T(y_d - \Phi_e\pi(Y_d))\left(\ddot{y}_d - \Phi_e\ddot{\pi}(Y_d)\right) + c_\delta(y_d - \Phi_e\pi(Y_d), \dot{y}_d - \Phi_e\dot{\pi}(Y_d)) = 0. \quad (3.9)$$

An approximate solution $\hat{\pi}(Y_d)$ to Eq. (3.9) can be obtained by a numerical approach, as in many practical nonlinear regulation problems [18]. If in this approximation we use a class of basis elements that are bounded functions of their arguments, $\hat{\pi}(Y_d)$ will necessarily be a bounded function over time as long as the trajectory $y_d(t)$ and its derivatives are bounded.

In [11], a polynomial function of Y_d was used as $\hat{\pi}(Y_d)$. The nonlinear terms in Eq. (3.9) are expanded in Taylor series and the constant coefficients in $\hat{\pi}(Y_d)$ are determined through the polynomial identity principle. Computational savings are obtained via a recursive procedure, solving Eq. (3.9) for increasing expansion orders until the final desired precision is obtained.

As noted in [16], this approach does not allow the use of noncausal inputs. Therefore, although the computed link deformation $\delta_d(t) = \hat{\pi}(Y_d(t))$ may be different from zero at time $t = 0$, there is no way to preload the manipulator to such a value.

3.2.2 Method 2: Iterative inversion in the frequency domain. For linear non-minimum phase mechanical systems, a stable inversion algorithm in the frequency domain has been introduced in [3] by regarding both the input $\tau(t)$ and the output $y_d(t)$ as periodic functions. Provided that the involved signals are Fourier-transformable, all quantities will automatically be bounded over time.

An extended interval of definition is considered for the output reference trajectory, namely with $t \in [-\Delta, T + \Delta]$, where Δ gives enough time to preload and discharge the internal deformation in the flexible manipulator, without motion of its end-effector. Indeed, we have: $y_d(t) = y_d(0)$, for $t \in [-\Delta, 0]$, and $y_d(t) = y_d(T)$, for $t \in [T, T + \Delta]$.

When the system is nonlinear the inversion algorithm is applied repeatedly, using successive linear approximations of the whole flexible manipulator equations around the nominal trajectory [4].

The same idea can be used in a simpler fashion, namely iterating the linearization process on the flexible dynamics only. For, rewrite Eq. (3.6) as

$$B_{\delta\delta}\ddot{\delta} + D\dot{\delta} + K\delta + f(y, \dot{y}, \ddot{y}, \delta, \dot{\delta}, \ddot{\delta}) = 0, \quad (3.10)$$

with

$$f = B_{\theta\delta}^T(y - \Phi_e\delta)\left(\ddot{y} - \Phi_e\ddot{\delta}\right) + c_\delta(y - \Phi_e\delta, \dot{y} - \Phi_e\dot{\delta}). \quad (3.11)$$

We compute a bounded link deformation $\delta_d(t)$ associated with the end-effector motion $y_d(t)$ by the following algorithm:

1. Choose an initial $\delta^{(0)}(t)$, with first and second time derivatives, over the time interval $[-\Delta, T + \Delta]$. Typically, $\delta^{(0)}(t) \equiv 0$. Set $k = 0$.
2. Using Eq. (3.11), define the forcing term

$$f^{(k)}(t) = f(y_d(t), \dot{y}_d(t), \ddot{y}_d(t), \delta^{(k)}(t), \dot{\delta}^{(k)}(t), \ddot{\delta}^{(k)}(t)), \quad (3.12)$$

and solve

$$B_{\delta\delta}\ddot{\delta} + D\dot{\delta} + K\delta + f^{(k)}(t) = 0 \quad (3.13)$$

using the FFT method as in [3]. Denote the solution as $\delta^{(k+1)}(t)$, defined for $t \in [-\Delta, T + \Delta]$.

3. If $\|\delta^{(k+1)}(t) - \delta^{(k)}(t)\| \leq \epsilon_\delta$ for all $t \in [-\Delta, T + \Delta]$, set $\delta_d(t) = \delta^{(k+1)}(t)$ and stop. Else, set $k = k + 1$ and go to step 2.

3.2.3 Method 3: Iterative learning in the time domain. Robot learning control allows to acquire from experiments (or from simulations on an accurate dynamic model) the input torque needed for reproducing a desired output trajectory [1]. The trajectory is repeated several times and, at the end of each trial, the tracking error is used for updating the command to be applied at the next iteration. This method is well established for rigid robots, with simple PD-like updates of the input command.

In the presence of link flexibility, additional filtering of high-frequency signal components is needed to guarantee convergence. Since the tracking error processing is performed off-line, noncausal filtering is allowed (i.e. we can update the command at a given instant using also error samples at later instants of the previous trial), as well as anticipated shifting of signals in time. In this way, we can learn the input torque to be applied for $t \in [-\Delta, T + \Delta]$, even outside the interval of actual definition of the output trajectory [24].

A similar approach is proposed here for the numerical solution of Eq. (3.6). Again, we limit the learning process to the flexible dynamics. Instead of using the tracking error, define a *deformation torque error* as

$$e = B_{\delta\delta}\ddot{\delta} + D\dot{\delta} + K\delta + B_{\theta\delta}^T(y_d - \Phi_e\delta) \left(\ddot{y}_d - \Phi_e\ddot{\delta} \right) + c_\delta(y_d - \Phi_e\delta, \dot{y}_d - \Phi_e\dot{\delta}), \quad (3.14)$$

namely the left-hand side of Eq. (3.6), evaluated on the desired output trajectory y_d . Indeed, an admissible link deformation history $\delta(t)$ satisfies

$$e(\delta, \dot{\delta}, \ddot{\delta}, t) = 0, \quad \forall t \in [-\Delta, T + \Delta]. \quad (3.15)$$

According to the iterative learning paradigm, we compute the link deformation $\delta_d(t)$ associated with the end-effector motion $y_d(t)$ by the following algorithm:

1. Choose an initial $\delta^{(0)}(t)$, with first and second time derivatives, over the time interval $[-\Delta, T + \Delta]$. Typically, $\delta^{(0)}(t) \equiv 0$. Set $k = 0$.

2. Using Eq. (3.14), define

$$e^{(k)}(t) = e(\delta^{(k)}(t), \dot{\delta}^{(k)}(t), \ddot{\delta}^{(k)}(t), t). \quad (3.16)$$

If

$$\|e^{(k)}(t)\| \leq \epsilon_\epsilon, \quad \forall t \in [-\Delta, T + \Delta], \quad (3.17)$$

set $\delta_d(t) = \delta^{(k)}(t)$ and stop. Else, process the error $e^{(k)}(t)$ by finite-impulse response (FIR) filters as in [24], obtaining a filtered version $e_f^{(k)}(t)$ and its derivative $\dot{e}_f^{(k)}(t)$.

3. Update by the following PD-like learning rule

$$\delta^{(k+1)}(t) = \delta^{(k)}(t) - K_{LP} e_f^{(k)}(t) - K_{LD} \dot{e}_f^{(k)}(t), \quad (3.18)$$

with sufficiently small learning gains $K_{LP} > 0$ and $K_{LD} > 0$. Set $k = k+1$ and go to step 2.

A number of final remarks are in order.

Remark 3.3. Although a complete convergence analysis is lacking for Methods 2 and 3, their success is supposed to depend on the strong underlying linear structure of equation (3.13) and definition (3.16), respectively. In particular, the dependence on $\delta(t)$ in the forcing term $f^{(k)}$ of Eq. (3.13), in view of Assumption 3.1, is a small perturbation affecting the reference output trajectory $y_d(t)$. Similarly, the nonlinear time-varying part in $e^{(k)}$ is mainly due to $y_d(t)$ and thus, being repetitive in nature, is well handled by the learning process. Moreover, the update (3.18) can be seen as a step of the gradient method for solving Eq. (3.15).

Remark 3.4. In all methods, the evaluation of $\delta_d(t)$ and $\dot{\delta}_d(t)$ at time $t = 0$, together with the use of Eq. (3.8) for the robot joint angles, provides the correct initial state producing a bounded evolution for the link deformation. If the flexible manipulator starts in this deformed state, use of Eq. (3.7) yields *exact* tracking of the end-effector trajectory.

Remark 3.5. If the initial state is not on its computed reference trajectory, a stabilizing term should be added in order to drive the state towards this solution, and only *asymptotic* output tracking can be guaranteed. This can be accomplished—at least locally—using a linear state feedback regulator, characterized by a matrix F ,

$$\tau = \tau_d + F \begin{bmatrix} \theta_d - \theta \\ \dot{\theta}_d - \dot{\theta} \\ \delta_d - \delta \\ \dot{\delta}_d - \dot{\delta} \end{bmatrix}, \quad (3.19)$$

with τ_d given by Eq. (3.7). One can also use a simpler stabilizing matrix F in Eq. (3.19), as in the *partial state* feedback controller

$$\tau = \tau_d + F_P(\theta_d - \theta) + F_D(\dot{\theta}_d - \dot{\theta}), \quad (3.20)$$

with positive definite (diagonal) matrices F_P and F_D [14]. Note that control (3.19) (as well as (3.20)) can be also applied over the entire interval $[-\Delta, T + \Delta]$, yielding a more robust version of a noncausal trajectory regulator.

3.3 Experimental Results

We report here some experimental results obtained for the end-effector trajectory tracking of FLEXARM, a two-link planar manipulator with a flexible forearm and direct-drive DC motors available at the Robotics Laboratory of DIS. The dynamic model of the arm can be found in [10], where two modes are used for describing the forearm bending in the horizontal plane of motion. The essential data are as follows: the length of the first rigid link and of the flexible forearm are, respectively, $\ell_1 = 0.3$ m and $\ell_2 = 0.7$ m; the forearm weight is 1.8 kg and its first two eigenfrequencies are at 4.7 and 14.4 Hz.

Since the first link is rigid, the output is defined as (see Eq. (3.4))

$$y = \begin{bmatrix} \theta_1 \\ \theta_2 + \phi_{21}(\ell_2)\delta_{21}/\ell_2 + \phi_{22}(\ell_2)\delta_{22}/\ell_2 \end{bmatrix}. \quad (3.21)$$

The reference trajectory is a 7th-order polynomial with zero initial and final velocity, acceleration, and jerk for both scalar outputs. The first output (joint 1) moves 45° , while the second output (tip of flexible forearm) moves 90° in 2 s.

The control law is given by Eq. (3.20) with $F_{P1} = 100$, $F_{P2} = 130$, $F_{D1} = 6$, and $F_{D2} = 8$. Method 3 was used for the computation of link deformation, extending learning over 3 s ($\Delta = 0.5$ s). Convergence of the deformation torque error with tolerance $\epsilon_e = 10^{-6}$ Nm in Eq. (3.17) is reached within 30 iterations on the nominal model.

Figures 3.1 and 3.2 show the desired and actual trajectory for the two outputs, with maximum errors of about 0.8° and 1° respectively. The applied joint torques are given in Fig. 3.3 (off-set values are due to noise and to some residual gravity effects caused by imperfect balancing of the structure).

In Fig. 3.4, the computed velocity profiles of the two joints ($\dot{\theta}_{d1}$ and $\dot{\theta}_{d2}$ from Eq. (3.8)) and of the angular deformation at the tip ($\dot{w}_2(\ell_2)/\ell_2 = (\phi_{21}(\ell_2)\dot{\delta}_{d,21} + \phi_{22}(\ell_2)\dot{\delta}_{d,22})/\ell_2$) are compared with the actual ones. The differences come from the inaccuracy of the model used for control computations. Before $t = 0.5$ s and after $t = 2.5$ s, it is possible to appreciate the preloading and discharging effects.

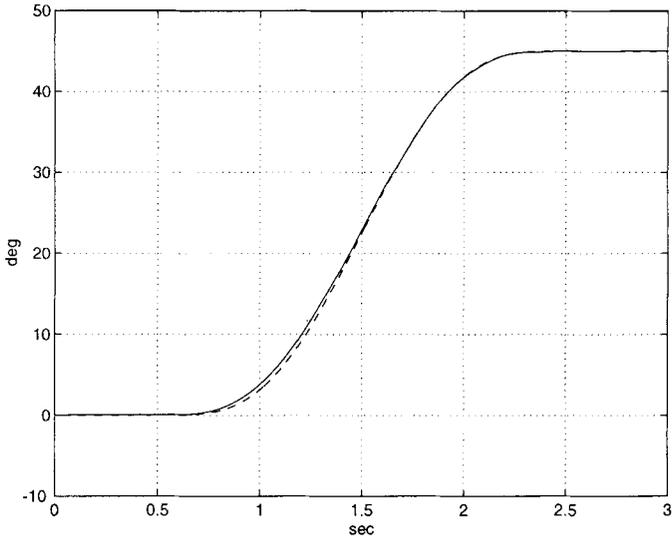


Fig. 3.1. Desired and actual trajectory for joint 1

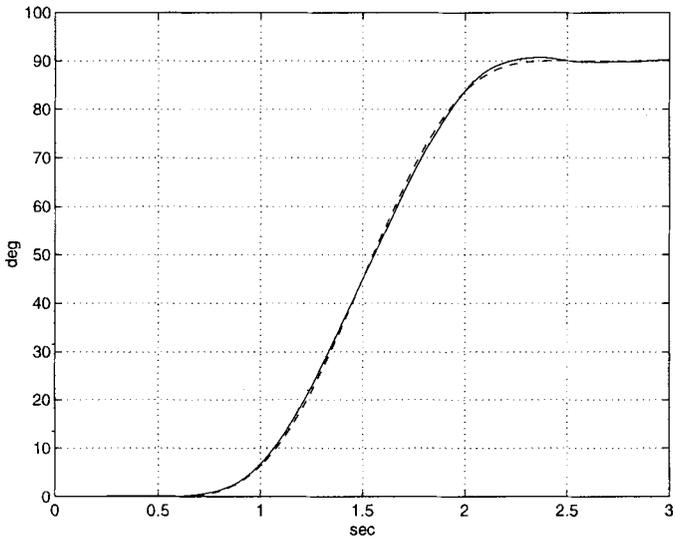


Fig. 3.2. Desired and actual trajectory for tip of link 2

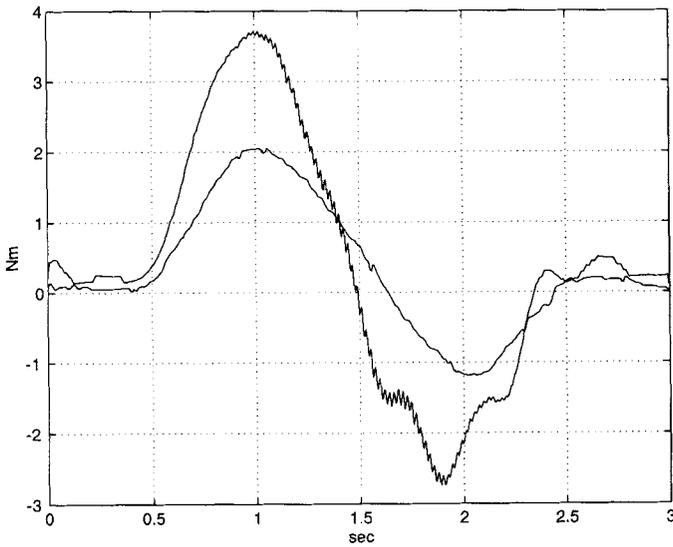


Fig. 3.3. Joint torques

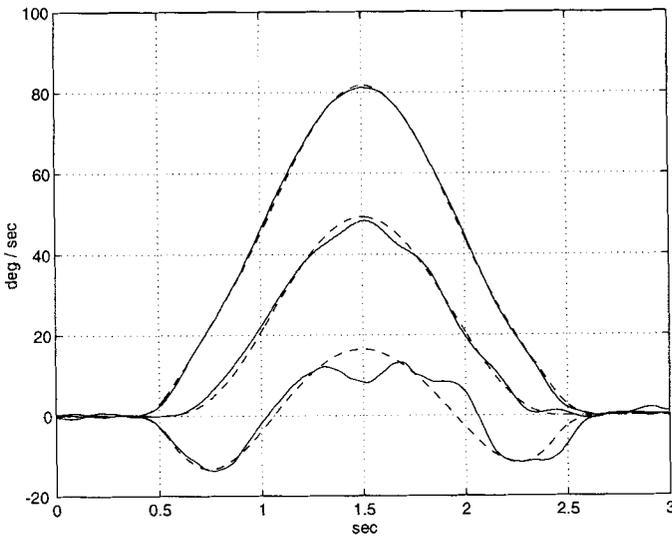


Fig. 3.4. Computed and actual velocity of joint 2 (top), joint 1 (center), and tip deformation (bottom)

4. Conclusions

The consideration of joint and link flexibility in robot manipulators gives rise to interesting theoretical issues when designing controllers for accurately tracking end-effector trajectories.

For a class of robots with elastic joints in which static state feedback fails to achieve exact linearization and input-output decoupling, we have introduced a new general algorithm for the synthesis of a dynamic feedback law reaching the same control goal.

In the case of robots with flexible links, an example of nonlinear systems with unstable zero dynamics, we have shown that some recently developed control techniques address essentially the same problem, namely the characterization of the bounded link deformation associated with a specified end-effector trajectory. For this critical computation three algorithms have been proposed.

Although we have assumed throughout the chapter ideal conditions, with perfect knowledge of the robot dynamic models and availability of full state measures, the presented results may be a starting point for the definition of adaptive and robust controllers, possibly using only output feedback.

For both types of flexible manipulators, only the second-order robot equations have been used in the analysis and in the control design. More physical insight is gained by working directly with the dynamic model terms, while control computations can be quite reduced, especially in the case of elastic joints. We regard this as a step toward the algorithmic design of advanced nonlinear controllers for general mechanical systems, without resorting to the state-space format.

Acknowledgement. The algorithm in Sect. 2.2 is the result of a joint work with Pasquale Lucibello. The experimental results on the two-link flexible arm in Sect. 3.3 were obtained by Stefano Panzieri. This work is supported by *MURST 40%* and *CNR 95.00106.CT07* funds.

References

- [1] Arimoto S, Kawamura S, Miyazaki F 1984 Bettering operation of robots by learning. *J Robot Syst.* 1:123–140
- [2] Barbieri E, Özgüner Ü 1988 Unconstrained and constrained mode expansions for a flexible slewing link. *ASME J Dyn Syst Meas Contr.* 110:416–421
- [3] Bayo E 1987 A finite-element approach to control the end-point motion of a single-link flexible robot. *J Robot Syst.* 4:63–75
- [4] Bayo E, Serna M A, Papadopoulos P, Stubbe J 1989 Inverse dynamics and kinematics of multi-link elastic robots: An iterative frequency domain approach. *Int J Robot Res.* 8(6):49–62
- [5] Book W J 1984 Recursive Lagrangian dynamics of flexible manipulator arms. *Int J Robot Res.* 3(3):87–101

- [6] Book W J 1990 Modeling, design, and control of flexible manipulator arms: A tutorial review. In: *Proc 29th IEEE Conf Decision Contr.* Honolulu, HI, pp 500–506
- [7] De Luca A 1988 Dynamic control of robots with joint elasticity. In: *Proc 1988 IEEE Int Conf Robot Automat.* Philadelphia, PA, pp 152–158
- [8] De Luca A 1988 Control properties of robot arms with joint elasticity. In: Byrnes C I, Martin C F, Saeks R E (eds) *Analysis and Control of Nonlinear Systems.* North-Holland, Amsterdam, The Netherlands, pp 61–70
- [9] De Luca A, Lanari L 1995 Robots with elastic joints are linearizable via dynamic feedback. In: *Proc 34th IEEE Conf Decision Contr.* New Orleans, LA, pp 3895–3897
- [10] De Luca A, Lanari L, Lucibello P, Panzieri S, Ulivi G 1990 Control experiments on a two-link robot with a flexible forearm. In: *Proc 29th IEEE Conf Decision Contr.* Honolulu, HI, pp 520–527
- [11] De Luca A, Lanari L, Ulivi G 1991 End-effector trajectory tracking in flexible arms: Comparison of approaches based on regulation theory. In: Canudas de Wit C (ed) *Advanced Robot Control.* Springer-Verlag, Berlin, Germany, pp 190–206
- [12] De Luca A, Lucibello P, Ulivi G 1989 Inversion techniques for trajectory control of flexible robot arms. *J Robot Syst.* 6:325–344
- [13] De Luca A, Siciliano B 1993 Inversion-based nonlinear control of robot arms with flexible links. *AIAA J Guid Contr Dyn.* 16:1169–1176
- [14] De Luca A, Siciliano B 1996 Flexible Links. In: Canudas de Wit C, Siciliano B, Bastin G (eds) *Theory of Robot Control.* Springer-Verlag, London, UK, pp 219–261
- [15] De Luca A, Tomei P 1996 Elastic Joints. In: Canudas de Wit C, Siciliano B, Bastin G (eds) *Theory of Robot Control.* Springer-Verlag, London, UK, pp 179–217
- [16] Devasia S, Chen D, Paden B 1996 Nonlinear inversion-based output tracking. *IEEE Trans Automat Contr.* 41:930–942
- [17] Fraser A R, Daniel R W 1991 *Perturbation Techniques for Flexible Manipulators.* Kluwer, Boston, MA
- [18] Huang J, Rugh W J 1992 An approximation method for the nonlinear servomechanism problem. *IEEE Trans Automat Contr.* 37:1395–1398
- [19] Isidori A 1995 *Nonlinear Control Systems.* 3rd Edition, Springer-Verlag, London, UK
- [20] Isidori A, Moog C H, De Luca A 1986 A sufficient condition for full linearization via dynamic state feedback. In: *Proc 25th IEEE Conf Decision Contr.* Athens, Greece, pp 203–208
- [21] Lucibello P, Di Benedetto M D 1993 Output tracking for a nonlinear flexible arm. *ASME J Dyn Syst Meas Contr.* 115:78–85
- [22] Lucibello P, Panzieri S 1996 End point trajectory control with internal stability of a flexible link by learning. In: *Proc 1996 IEEE Int Conf Robot Automat.* Minneapolis, MN, pp 2117–2123
- [23] Meirovitch L 1967 *Analytical Methods in Vibrations.* Macmillan, New York
- [24] Panzieri S, Ulivi G 1995 Disturbance rejection of iterative learning control applied to trajectory tracking for a flexible manipulator. In: *Proc 3rd Euro Contr Conf.* Roma, Italy, pp 2374–2379
- [25] Spong M W 1987 Modeling and control of elastic joint robots. *ASME J Dyn Syst Meas Contr.* 109:310–319
- [26] Sweet L M, Good M C 1985 Redefinition of the robot motion control problem. *IEEE Contr Syst Mag.* 5(3):18–24

- [27] Tomei P 1991 A simple PD controller for robots with elastic joints. *IEEE Trans Automat Contr.* 36:1208–1213
- [28] Zhao H, Chen D 1993 Exact and stable tip trajectory tracking for multi-link flexible manipulator. In: *Proc 32nd IEEE Conf Decision Contr.* San Antonio, TX, pp 1371–1376