Proceedings of the 2000 IEEE
International Conference on Robotics & Automation
San Francisco, CA • April 2000

Feedforward/Feedback Laws for the Control of Flexible Robots*

Alessandro De Luca

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy
deluca@labrob.ing.uniroma1.it

Abstract

We present a survey of the nominal motion generation schemes and of the associated simple control solutions for robots displaying flexibility effects. Two model classes are considered: robots with elastic joints but rigid links and robots with flexible links. Model-based feedforward laws are derived for the two basic motion tasks of state-to-state transfer in given time and exact trajectory execution. In particular, we present a new solution to the finite-time reconfiguration problem for a one-link flexible arm. Finally, we use the developed commands into a simple feedback scheme that requires only standard sensors on the motors.

1 Introduction

The rigidity assumption in the dynamic modeling of robot manipulators 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, exciting an oscillatory behavior.

There are two sources of vibration in robot manipulators: *i)* concentrated *joint elasticity*, caused by transmission elements such as harmonic drives, belts, or long shafts —typical of industrial robots [1], and *ii)* distributed *link flexibility*, introduced by a long reach and slender/lightweight construction of the arm [2]. In both cases, the robotic systems contain additional generalized coordinates that exceed in number the available command inputs, making the motion planning and control problems more difficult.

In order to be able to counteract the negative effects of flexibility, control laws should be designed on the basis of more complete dynamic models [3, 4, 5]. Controllers aimed at precise positioning (regulation tasks) or accurate execution of trajectories (tracking tasks) usually consists of the combination of a nominal feedforward action and a robustifying linear/nonlinear feedback part. In flexible robots, one main role of feedback control is the active damping of vibrations. Quite often, it is relevant to complete the nominal motion task in a prescribed finite time. In these cases, structural vibrations should be fully compensated within the feedforward term.

A number of advanced feedback solutions already exist in the literature (see the surveys in [6, 7]). Most of them require feedback from the whole state of the robot, implying the presence of additional sensors for the deformation variables (strain gauges, accelerometers, visual, and so on), beside the encoder/tachometer pairs mounted on the joint motors. To avoid these extra sensors, a state observer can be included in the control law at the expense of a more complex design.

In this paper, after reviewing briefly the dynamic models of robots with elastic joints or with flexible links, we consider first the problem of generating suitable model-based feedforward laws for the basic motion tasks of state-to-state transfer in given time and exact trajectory execution. The results, partly available in the referenced literature, are organized here in a systematic way. In particular, we present a new solution to the finite-time reconfiguration problem for a one-link flexible arm. The computed feedforward commands can be incorporated into a simple feedback controller that needs only position and velocity measurements at the level of the robot motors. The resulting scheme follows the so-called nonlinear regulation paradigm [8] for nonlinear systems. It can be considered as the counterpart for flexible robots of the widespread pre-computed torque plus joint PD feedback control for rigid manipulators.

*Work supported by MURST and CNR funds.

2 Dynamic models of flexible robots

2.1 Robots with elastic joints (EJ)

Consider a robot actuated by electrical drives with N joints undergoing elastic deformation. Let $q \in I\!R^N$ be the link positions and $\theta \in I\!R^N$ be the motor positions, as reflected through the gear ratios. In view of small joint deformations, elasticity is modeled as a linear spring. The rotors of the motors are balanced uniform bodies, so that the inertia matrix and the gravity term in the dynamic model will be independent of θ.

Following the Lagrangian approach, the robot dynamic model consists of $2N$ second-order differential equations (see, e.g., [6] for a detailed derivation)

$$M(q)\ddot{q} + S\ddot{\theta} + c(q,\dot{q}) + g(q) + K(q-\theta) = 0 \quad (1)$$
$$S^T\ddot{q} + J\ddot{\theta} + K(\theta - q) = \tau, \quad (2)$$

where the inertia matrix $M(q)$, the Coriolis and centrifugal terms $c(q,\dot{q})$, and the gravity terms $g(q)$ are all related to the rigid links, the diagonal matrix $J > 0$ contains the effective motor inertias, S accounts for the inertial couplings between motors and links (here assumed to be constant), and $K > 0$ is the diagonal matrix of the joint stiffness constants. Usually, the ith motor is mounted on link $i-1$ and moves link i. As a result, matrix S is always strictly upper triangular [4]. In eq. (2), $\tau \in I\!R^N$ are the torques supplied by the motors. Joint damping can be included by adding a term $D(\dot{q} - \dot{\theta})$ in eq. (1) and its opposite in eq. (2).

For a one-link robot with an elastic joint as well as for other multi-link special kinematic structures with elastic joints (e.g., a $2R$ polar robot) it is found that $S = 0$. The same situation is forced in general by the modeling assumption introduced in [3], namely by considering in the angular part of the kinetic energy of each rotor only the term due to its relative rotation. In those cases, the dynamic model (1–2) simplifies to

$$M(q)\ddot{q} + c(q,\dot{q}) + g(q) + K(q-\theta) = 0 \quad (3)$$
$$J\ddot{\theta} + K(\theta - q) = \tau. \quad (4)$$

The following control properties are known for the above two dynamic models.

Property 1 *The dynamic model (1–2) can be transformed into a fully linear one via dynamic state feedback [9], with a compensator of dimension* $2N(N-1)$. *The relation from the new input to the output q is given by N independent chains of* $2(N+1)$ *integrators.*

Property 2 *The dynamic model (3–4) can be transformed into a fully linear one via static state feedback [3]. The relation from the new input to the output q is given by N independent chains of* 4 *integrators.*

These properties imply that EJ robots have no zero dynamics [10] associated to the output q.

2.2 Robots with flexible links (FL)

Consider a robot with N flexible links, interconnected by (rigid) rotational joints. Link deformations are small and thus only linear elastic effects are present. We assume that each link can only bend in one lateral direction (i.e., in the plane normal to the preceeding joint axis), being stiff with respect to axial forces and torsion. The bending deformation $w_i(x_i, t)$ at a generic point $x_i \in [0, \ell_i]$ along the ith link (of length ℓ_i) is then modeled, using separation in space and time, as

$$w_i(x_i, t) = \sum_{j=1}^{N_{ei}} \phi_{ij}(x_i)\delta_{ij}(t), \qquad i = 1, \dots, N, \quad (5)$$

where the N_{ei} spatial components $\phi_{ij}(x_i)$ are the assumed modes of deformation satisfying geometric and/or dynamic boundary conditions, while $\delta_{ij}(t)$ are the associated deformation coordinates. The use of approximated finite-dimensional expansions in the form (5) is preferred for handling general multi-link flexible manipulators.

Let $\theta \in I\!R^N$ be the (joint) angular positions associated to the rigid motion and $\delta \in I\!R^{N_e}$ be the link deformation variables, where $N_e = \sum_{i=1}^N N_{ei}$. To simplify derivations, the total kinetic energy of the system is evaluated in the undeformed configuration $\delta = 0$. The robot dynamic model for control design consists of $N + N_e$ second-order differential equations (see, e.g., [5] or [7] for details)

$$M_{\theta\theta}(\theta)\ddot{\theta} + M_{\theta\delta}(\theta)\ddot{\delta} + c_\theta(\theta,\dot{\theta},\dot{\delta}) + g_\theta(\theta,\delta) = B_\theta\tau \quad (6)$$
$$M_{\theta\delta}^T(\theta)\ddot{\theta} + M_{\delta\delta}\ddot{\delta} + c_\delta(\theta,\dot{\theta}) + g_\delta(\theta) + K\delta = B_\delta\tau. \quad (7)$$

The dependence in the Coriolis and centrifugal terms c_θ and c_δ stems from the structure of the blocks in the overall inertia matrix $M(\theta)$. Matrix $K > 0$ represents the arm modal stiffness. Modal damping can be introduced by adding a term $D\dot{\delta}$ (with $D \geq 0$) in eq. (7). The input matrices B_θ and B_δ in eqs. (6–7) take on special forms depending on the reference frames used for describing the link deformations with eq. (5). If these frames are clamped at each link base (*clamped frames*), we have $B_\theta = I_{N\times N}$ and $B_\delta = 0$ so that the torques $\tau \in I\!R^N$ appear only in eq. (6).

The tip position of the ith link may be characterized by the pointing angle from its base

$$y_{Ei} = \theta_i + \sum_{j=1}^{n_{ei}} \frac{\phi_{ij}(\ell_i)}{\ell_i}\delta_{ij}, \qquad i = 1, \dots, N. \quad (8)$$

These angles can be organized into a vector $y_E = \theta + \Phi_E \delta \in I\!R^N$ that is related to the end-effector location.

For a single flexible link moving in the horizontal plane with n_e assumed modes of deflection, the dynamic model collapses to

$$m_{\theta\theta}\ddot{\theta} + m_{\theta\delta}\ddot{\delta} = b_\theta \tau \quad (9)$$
$$m_{\theta\delta}^T \ddot{\theta} + M_{\delta\delta}\ddot{\delta} + K\delta = b_\delta \tau, \quad (10)$$

namely that of a linear system. Using a reference frame linked to the instantaneous center of mass of the link (*pinned frame*), the scalar θ will be the angle between the absolute x-axis and the axis pointing at the center of mass. After suitable orthonormalization of the assumed modes, the inertia and stiffness matrices become diagonal and eqs. (9–10) reduce to [11]

$$J_0 \ddot{\theta} = \tau, \quad (11)$$
$$\ddot{\delta}_i + \omega_i^2 \delta_i = \phi_i'(0)\tau, \qquad 1, \ldots, n_e, \quad (12)$$

where J_0 is the total arm inertia, ω_i are the angular eigenfrequencies of the link, and the prime denotes differentiation w.r.t. space.

The following control properties are known for the above dynamic models.

Property 3 *The dynamic model (6–7) can be input-output linearized and decoupled with respect to the output θ via static state feedback [12]. The associated zero dynamics is stable (asympotically stable with a modal damping $D > 0$). The zero dynamics associated to the output y_E is in general unstable [13].*

Property 4 *The linear dynamic model (9–10) is controllable. The transfer function from τ to θ (joint output) is minimum phase*[1] *(strictly minimum phase if $D > 0$). For uniform mass distribution of the link, the transfer function from τ to y_E (end-effector output) is always non-minimum phase.*

3 Motion tasks

We classify the required motion tasks for flexible (EJ or FL) robots as follows:

State-to-state transfer. The robot should be reconfigured from one (typically, equilibrium) state to another. A finite completion time is usually specified. Intermediate states can be arbitrary, provided that elastic deformations are kept limited.

Trajectory execution. A given path is assigned with an associated timing law. The trajectory can be specified in terms of different output functions, e.g., at the motor, joint, or cartesian level, namely before or beyond the structural flexibility. Only bounded deformation state solutions are feasible.

[1] Its zeros are in the left hand side of the complex plane.

A basic remark is in order. For rigid manipulators, state-to-state transfers are particular cases of trajectory execution tasks. Since the number N of actuators equals the number of generalized coordinates, one can always fit a (twice-differentiable) trajectory interpolating the initial and final state so as to perform the transfer in a given completion time. Moreover, in the non-redundant case the level of trajectory definition is not really relevant, provided that kinematic singularities are not encountered. On the other hand, structural flexibility implies the presence of extra coordinates in the robotic system, namely N additional variables for EJ robots and N_e for FL robots. Depending on the level of definition, the output trajectory may either induce a unique trajectory for the whole robot state or multiple possible evolutions. The relevance of this will become clear in Sec. 4 and 5.

The above motion tasks can be performed with two standard classes of control laws, typically used in combination:

Feedforward control. The whole motion task should be known in advance so that the required input torque can be computed off line, based on the available dynamic model of the robot. This solution works satisfactorily when no perturbations act on the system and the initial state is correctly guessed.

Feedback control. Partial or full state measurements are used within a feedback law for asymptotic stabilization of an equilibrium state (regulation) or around a reference trajectory (trajectory tracking). Local or global stabilization results may be pursued.

4 Feedforward laws for EJ robots

4.1 Trajectory execution

Let the motion task be specified by a desired trajectory $q = q_d(t)$, $t \in [0, T]$, for the robot links. This may come from the standard (rigid) kinematic inversion of a cartesian trajectory.

The nominal input torque for this task, and the associated robot state trajectory, are easily computed for the model (3–4). From eq. (3), we have

$$\theta_d = q_d + K^{-1}\left[M(q_d)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d)\right], \quad (13)$$

which is the desired motor trajectory. Differentiating twice yields

$$\ddot{\theta}_d = \ddot{q}_d + K^{-1}\Big[M(q_d)q_d^{[4]} + 2\dot{M}(q_d)q_d^{[3]} + \ddot{M}(q_d)\ddot{q}_d + \ddot{c}(q_d, \dot{q}_d) + \ddot{g}(q_d)\Big],$$

where we have used the notation $x^{[i]} = d^i x/dt^i$. By substitution in eq. (4), we obtain

$$\tau_d = J\ddot{\theta}_d + M(q_d)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d). \qquad (14)$$

In order to be exactly reproduced by means of the input torque (14), the desired output trajectory $q_d(t)$ should be at least 4 times differentiable.

The same computation is slightly more complex for the complete model (1–2) and is mainly based on the strictly upper triangular structure of matrix $S \neq 0$. By setting $f(q, \dot{q}, \ddot{q}) = M(q)\ddot{q} + c(q, \dot{q}) + g(q) + Kq$, we rewrite eq. (1) as

$$f(q, \dot{q}, \ddot{q}) + S\ddot{\theta} - K\theta = 0. \qquad (15)$$

The desired motor trajectory $\theta_d(t)$ is obtained starting from the last scalar equation in (15), which reads

$$f_N(q, \dot{q}, \ddot{q}) - k_N \theta_N = 0.$$

From this

$$\theta_{N,d} = \frac{1}{k_N} f_N(q_d, \dot{q}_d, \ddot{q}_d).$$

The before last equation in (15)

$$f_{N-1}(q, \dot{q}, \ddot{q}) - s_{N-1,N}\ddot{\theta}_N - k_{N-1}\theta_{N-1} = 0$$

gives then

$$\theta_{N-1,d} = \frac{1}{k_{N-1}} \left[f_{N-1}(q_d, \dot{q}_d, \ddot{q}_d) - \frac{s_{N-1,N}}{k_N} \ddot{f}_N(q_d, \dot{q}_d, \ddot{q}_d) \right],$$

where $\theta_{N,d}$ has been differentiated twice. Proceeding upwards in a similar way, one arrives to the first equation in (15)

$$f_1(q, \dot{q}, \ddot{q}) - \sum_{j=2}^{N} s_{1,j}\ddot{\theta}_j - k_1\theta_1 = 0$$

that gives

$$\theta_{1,d} = \frac{1}{k_1} \left[f_1(q_d, \dot{q}_d, \ddot{q}_d) - \sum_{j=2}^{N} s_{1,j}\ddot{\theta}_{j,d}(q_d, \dot{q}_d, \ldots, q_d^{[2N]}) \right].$$

Having computed componentwise the desired motor trajectory, we differentiate it twice obtaining formally

$$\ddot{\theta}_d = \ddot{\theta}_d \left(q_d, \dot{q}_d, \ldots, q_d^{[2(N+1)]} \right).$$

The nominal input torque is computed by summing eqs. (1) and (2) as

$$\tau_d = (J + S)\ddot{\theta}_d + (M(q_d) + S^T)\ddot{q}_d + c(q_d, \dot{q}_d) + g(q_d). \qquad (16)$$

The desired output trajectory $q_d(t)$ should now be at least $2(N+1)$ times differentiable in order to be exactly reproduced by means of the input torque (16).

On the other hand, if the motion task is specified in terms of a desired trajectory for the robot motor position, $\theta = \theta_d(t)$, $t \in [0, T]$, in order to compute the nominal torque we have to choose initial values for $q(0)$ and $\dot{q}(0)$ (position and velocity of the links). From the model (1–2), forward integration of

$$\ddot{q} = -M^{-1}(q) \left[S\ddot{\theta}_d + c(q, \dot{q}) + g(q) + K(q - \theta_d) \right]$$

yields the desired evolution $q_d(t)$ (setting $S = 0$ works for the model (3–4)). Use of eq. (2) (or eq. (4)), evaluated along $\theta = \theta_d(t)$, gives the desired nominal torque τ_d. Indeed, this torque and the resulting link motion will depend on the choice of $q(0)$ and $\dot{q}(0)$.

4.2 State-to-state transfer

The previous trajectory generation schemes can be used also for the state-to-state transfer task in a given time T. A desired trajectory $q_d(t)$, $t \in [0, T]$, is specified by interpolating suitable boundary conditions at $t = 0$ and $t = T$. The evolution of $\theta_d(t)$ is then uniquely determined as a consequence of the fact that robots with elastic joints have no zero dynamics associated to the output q (i.e., Properties 1 and 2). Eq. (16) provides the required nominal torque. Note that if the robot moves from one equilibrium state to another, the configuration (θ, q) and the input torque τ at the initial and final time should satisfy

$$g(q_e) = K(\theta_e - q_e) = \tau_e.$$

5 Feedforward laws for FL robots

5.1 State-to-state transfer

Consider a generic state-to-state transfer motion task. In particular, we have a rest-to-rest motion if the flexible link arm is reconfigured from one equilibrium state to another. In the absence of gravity, the arm is undeformed.

For the one-link case in the horizontal plane, we can solve this problem using an idea similar to the case of elastic joint robots. In fact, in view of Property 4, we can always design an output function y such that the associated transfer function has no zeros (i.e., the system has no zero dynamics). This output has maximum relative degree (equal to the state space dimension $2(n_e + 1)$ of the flexible robot) and can be used, together with its derivatives up to the order $2n_e + 1$, as a new state representation of the system. The state-to-state transfer is then solved by defining an interpolating trajectory $y_d(t)$, with appropriate boundary conditions at time $t = 0$ and $t = T$. For this, a polynomial of degree $4n_e + 3$ will be sufficient.

We show this using the model (11–12). Let the design output be, similarly to eq. (8),

$$y = \theta + \sum_{i=1}^{n_e} c_i\delta_i = \theta + c^T\delta, \qquad (17)$$

with the coefficients c_i ($i = 1, \ldots, n_e$) to be determined. These are computed by imposing the independence from input τ of the first $2n_e$ derivatives of the output (17). Due to the second order structure of system (11–12), the torque may appear only in the first n_e even derivatives. We have

$$\ddot{y} = \Big(\frac{1}{J_0} + \sum_{i=1}^{n_e} c_i\phi_i'(0)\Big)\tau - \sum_{i=1}^{n_e} c_i\omega_i^2\delta_i$$

from which we set $\sum c_i\phi_i'(0) = -1/J_0$. Next,

$$y^{[4]} = -\sum_{i=1}^{n_e} c_i\omega_i^2\phi_i'(0)\,\tau + \sum_{i=1}^{n_e} c_i\omega_i^4\delta_i$$

from which we set $\sum c_i\omega_i^2\phi_i'(0) = 0$. Proceeding further in the same way, we obtain finally the following linear system

$$V \cdot \mathrm{diag}\{\phi_1'(0), \ldots, \phi_{n_e}'(0)\} \cdot c = b, \qquad (18)$$

with $b^T = [\,-1/J_0 \;\; 0 \;\; \ldots \;\; 0\,]$ and the Vandermonde matrix

$$V = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ \omega_1^2 & \omega_2^2 & \ldots & \omega_{n_e}^2 \\ \omega_1^4 & \omega_2^4 & \ldots & \omega_{n_e}^4 \\ & \ldots & & \\ \omega_1^{2(n_e-1)} & \omega_2^{2(n_e-1)} & \ldots & \omega_{n_e}^{2(n_e-1)} \end{bmatrix},$$

which is always nonsingular being $\omega_i \neq \omega_j$, for $i \neq j$. Solving eq. (18) gives the vector of coefficients c. With the following invertible state transformation

$$\begin{bmatrix} y \\ \ddot{y} \\ \vdots \\ y^{[2n_e]} \end{bmatrix} = T \begin{bmatrix} \theta \\ \delta_1 \\ \vdots \\ \delta_{n_e} \end{bmatrix}, \quad \begin{bmatrix} \dot{y} \\ y^{[3]} \\ \vdots \\ y^{[2n_e+1]} \end{bmatrix} = T \begin{bmatrix} \dot{\theta} \\ \dot{\delta}_1 \\ \vdots \\ \dot{\delta}_{n_e} \end{bmatrix}, \qquad (19)$$

where

$$T = \begin{bmatrix} 1 & c_1 & \ldots & c_{n_e} \\ 0 & -c_1\omega_1^2 & \ldots & -c_{n_e}\omega_{n_e}^2 \\ & & \ldots & \\ 0 & (-1)^{n_e}c_1\omega_1^{2n_e} & \ldots & (-1)^{n_e}c_{n_e}\omega_{n_e}^{2n_e} \end{bmatrix},$$

any initial and desired final states $(\theta, \delta, \dot{\theta}, \dot{\delta})$ are mapped into boundary conditions for the interpolating polynomial $y_d(t)$ and its derivatives. The nominal input torque for state-to-state trasfer is then

$$\tau_d = \frac{y_d^{[2(n_e+1)]} - (-1)^{n_e+1}\sum_{i=1}^{n_e} c_i\omega_i^{2(n_e+1)}\delta_i}{(-1)^{n_e}\sum_{i=1}^{n_e} c_i\omega_i^{2n_e}\phi_i'(0)}. \qquad (20)$$

The above technique is a generalization of the rest-to-rest method proposed in [14].

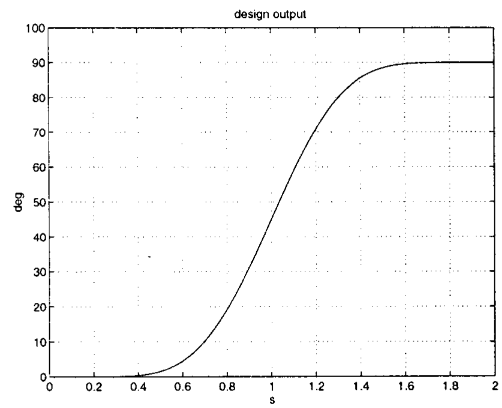

Figure 1: Interpolating profile for the design output

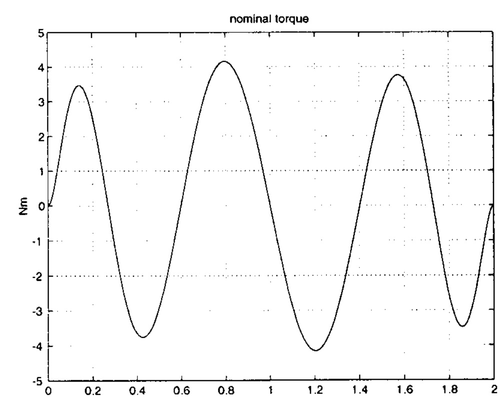

Figure 2: Torque for rest-to-rest motion of a one-link flexible arm

In Figs. 1–4 we show the results of a rest-to-rest slew motion of $90°$ in $T = 2$ s for a one-link flexible arm with $n_e = 3$ modes, uniform mass $m = 2.0825$ kg, and

$$J_0 = 0.3038 \text{ kgm}^2,$$
$$\omega^T = [\,4.7175 \quad 14.3949 \quad 26.9193\,] \text{ Hz}.$$

The design trajectory is given by a 15th-degree polynomial. Note that the clamped joint angle (the one that a motor encoder would measure) and the tip angle given by eq. (8) oscillate around the smooth design trajectory, but this effect vanishes exactly at the final time, together with the deformation variables δ. Also, the input torque is continuous and is zero outside the time interval $[0, T]$.

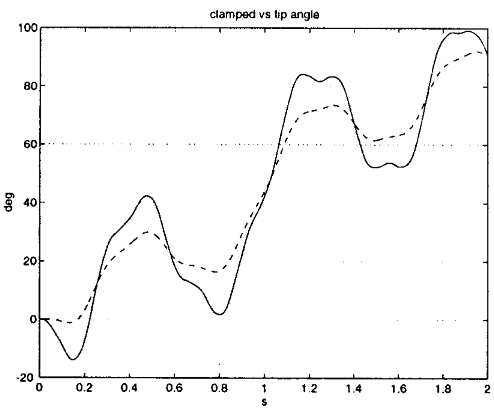

Figure 3: Evolution of the clamped joint angle (—) and of the tip angle (- -)

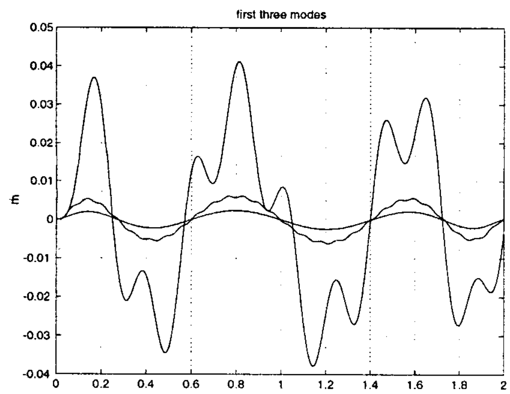

Figure 4: Evolution of first three flexible modes

5.2 Trajectory execution

Consider now trajectory execution tasks for the general model (6–7), in which clamped reference frames will be used. Let $\theta = \theta_d(t)$, $t \in [0, T]$, be the desired trajectory assigned to the robot joints. From eq. (7), we obtain

$$\begin{aligned}\ddot{\delta} &= -M_{\delta\delta}^{-1}\left[M_{\theta\delta}^T(\theta_d)\ddot{\theta}_d + c_\delta(\theta_d, \dot{\theta}_d) + g_\delta(\theta_d) - K\delta\right] \\ &= \Gamma(\theta_d, \dot{\theta}_d, \ddot{\theta}_d) - M_{\delta\delta}^{-1}K\delta, \qquad (21)\end{aligned}$$

which is a linear and marginally stable system with a forcing term Γ. Forward integration of eq. (21) from a generic $(\delta(0), \dot{\delta}(0))$ provides the nominal deformation history $\delta_d(t)$ (and its first two derivatives). Substituting these into eq. (6) yields the nominal torque

$$\tau_d = M_{\theta\theta}(\theta_d)\ddot{\theta}_d + M_{\theta\delta}(\theta_d)\ddot{\delta}_d + c_\theta(\theta_d, \dot{\theta}_d, \dot{\delta}_d) + g_\theta(\theta_d, \delta_d). \qquad (22)$$

If instead the desired trajectory is specified in terms of the end-effector output as $y_E = y_{Ed}(t)$, $t \in [0, T]$ (see eq. (8)), a computation similar to the one in eq. (21) is bound to fail, unless the initial state of deformation of the flexible arm is properly chosen. In fact, rewrite eq. (7) by substituting $y_{Ed} - \Phi_E\delta$ for θ:

$$M_{\theta\delta}^T(y_{Ed} - \Phi_E\delta)\left[\ddot{y}_{Ed} - \Phi_E\ddot{\delta}\right] + M_{\delta\delta}\ddot{\delta} +$$
$$c_\delta(y_{Ed} - \Phi_E\delta, \dot{y}_{Ed} - \Phi_E\dot{\delta}) + g_\delta(y_{Ed} - \Phi_E\delta) + K\delta = 0.$$

By approximating the nonlinear terms at $\delta = \dot{\delta} = 0$ (i.e., linearizing around the nominal output trajectory $y_{Ed}(t)$), we obtain

$$\left[M_{\delta\delta} - M_{\theta\delta}^T(y_{Ed})\Phi_E\right]\ddot{\delta} + K\delta = \Gamma_E(y_{Ed}, \dot{y}_{Ed}, \ddot{y}_{Ed}),$$

with $\Gamma_E = -\left[M_{\theta\delta}^T(y_{Ed})\ddot{y}_{Ed} + c_\delta(y_{Ed}, \dot{y}_{Ed}) + g_\delta(y_{Ed})\right]$ as a forcing term. This is a linear time-varying differential equation in δ with the matrix weighting the highest-order derivative being *not* positive definite. This suggests that attempting forward integration for generic initial conditions may lead to an unbounded solution $\delta_d(t)$ over time. As a matter of fact, for each desired end-effector trajectory $y_{Ed}(t)$, there exists a *unique* initial deformation state $(\delta(0), \dot{\delta}(0))$ leading to a bounded solution, any other choice being unfeasible.

In order to compute such initial deformation state, several alternative ways have been followed. In [15] a frequency domain inversion was proposed for a one-link flexible arm modeled as in eqs. (9–10). The idea is to look at the motion task as one window of a periodic behavior in which all quantities are implicitly assumed to be bounded. Working with Fourier transforms of the desired end-effector output trajectory and of the linear dynamic model terms, the nominal torque is computed by inversion in the frequency domain and then transformed back in the time domain. The obtained torque profile extends also before and after the output motion time interval $[0, T]$, being in principle defined over $(-\infty, +\infty)$. The problem of state initialization is thus completely bypassed and the correct values of $(\delta(0), \dot{\delta}(0))$ are obtained through the application of the non-causal part (i.e., for $t \in (-\infty, 0)$) of the nominal torque $\tau_d(t)$. The same results were obtained in [16] for a linear model of one-link flexible arms, but working only in the time domain. Stable and antistable eigenvalues of the inverse system matrix are treated separately and the associated differential equations are integrated forward and, respectively, backward in time so that only bounded solutions are found.

Extensions of these approaches to the multi-link flexible arm case involve either an iterative inversion in the frequency domain [17, 18], based on successive linearization of the residual dynamics, or the numerical time integration of suitable nonlinear operators [19, 20]. Along the same line, the iterative learning technique in [21] can be seen as a gradient-type method for computing the unique bounded deformation $\delta_d(t)$ associated to $y_{Ed}(t)$.

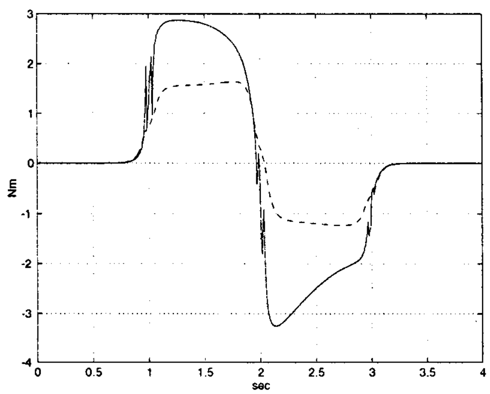

Figure 5: Torques for a bang-bang acceleration trajectory of the FLEXARM (τ_1 = ––, τ_2 = - -)

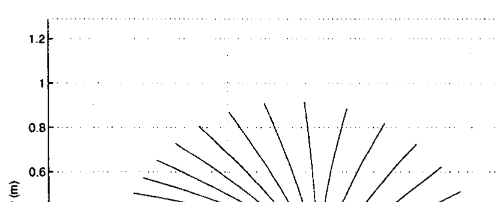

Figure 6: Stroboscopic motion of the FLEXARM

In Figs. 5–6 we report the results obtained after 8 iterations of the algorithm in [18] on the FLEXARM, a two-link planar manipulator with a very flexible forearm developed in our Lab. The dynamic model and the parameter values for the FLEXARM can be found in [22]. The desired output trajectory y_{Ed} has a bang-bang acceleration profile for a 90° slew motion of both links in $T = 2$ s. The interval of output motion is centered in a time window of 4 s, over which the computation of the input torques is performed. In Fig. 5, the precharging torques (before $t = 1$ s), needed to bring the flexible arm in the proper initial deformation state, and the discharging torques (after $t = 3$ s) are quite evident. The stroboscopic motion of the flexible arm in Fig. 6 shows that these initial and final internal link deflections do not perturb the end-effector position.

6 Feedback control

All the feedforward computations of the previous sections can be incorporated into a simple feedback control strategy aimed at robustyfying the behavior in the presence of inaccurate information on the actual initial state, small disturbances, or model uncertainties.

If we assume that only the position and velocity of the motors of an EJ or FL robot can be measured, and are thus available for feedback, a simple linear PD scheme with feedforward compensation can be designed as

$$\tau = \tau_d + K_P(\theta_d - \theta) + K_D(\dot{\theta}_d - \dot{\theta}) \qquad (23)$$

where $K_P > 0$ (at least) and $K_D > 0$. Maybe not surprisingly, this same structure can be used for any flexible robot and for state-to-state transfer as well as for executing trajectories defined at different output levels. Therefore, the controller (23) works both for regulation and tracking purposes. The key point is to use the specific feedforward torque $\tau_d(t)$ and (partial) state references $\theta_d(t)$ and $\dot{\theta}_d(t)$ corresponding to the case at hand. For example, for tracking a desired link trajectory $q_d(t)$ of an EJ robot, we use eq. (14) for τ_d and eq. (13) for θ_d, which are based on the dynamic model (3–4). In order to perform in feedback mode a rest-to-rest maneuver with a one-link flexible arm, τ_d is computed from eq. (20) while θ_d and $\dot{\theta}_d$ are obtained by inverting the state transformation (19) with $y = y_d(t)$ (in particular, the position reference of the joint clamped angle is displayed in Fig. 3). Other instances can be easily recovered by the reader.

On a formal basis, the stability analysis of the control law (23) for regulation tasks in the presence of gravity is detailed in [4] for a constant q_d of EJ robots and in [23] for a constant θ_d of FL robots. The constant torque τ_d is equal to $g(q_d)$ (see eq. (14)) for EJ robots and, respectively, to $g_\theta(\theta_d, \delta_d)$ (see eq. (22)) for FL robots.

7 Conclusions

We have surveyed and classified different alternatives available for generating the torque commands needed to perform typical motion tasks in flexible robots. In doing so, we have proposed a new scheme for rest-to-rest motion (or, more in general, for state-to-state transfer) of a one-link flexible arm. The generalization of such a scheme to multi-link flexible arms is a current research problem. In general, the absence or presence (and in the latter case, also the stable/unstable character) of zero dynamics associated to the chosen output of the flexible robot plays a dominant role in assessing the feasibility of the various computational schemes. A PD-type control law based on the nominal feedforward computations is a viable solution for a cheap but effective implementation of a feedback controller for the various motion tasks. A global stability analysis for the trajectory tracking capabilities of this feedback law is missing.

References

[1] L.M. Sweet and M.C. Good, "Redefinition of the robot motion control problem," *IEEE Control Systems Mag.*, vol. 5, no. 3, pp. 18–24, 1985.

[2] W.J. Book, "Modeling, design, and control of flexible manipulator arms: A tutorial review," *29th IEEE Conf. on Decision and Control*, pp. 500–506, 1990.

[3] M.W. Spong, "Modeling and control of elastic joint robots," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 109, no. 4, pp. 310–319, 1987.

[4] P. Tomei, "A simple PD controller for robots with elastic joints," *IEEE Trans. on Automatic Control*, vol. 36, no. 10, pp. 1208–1213, 1991.

[5] W.J. Book, "Recursive Lagrangian dynamics of flexible manipulator arms," *Int. J. of Robotics Research*, vol. 3, no. 3, pp. 87–101, 1984.

[6] A. De Luca and P. Tomei, "Elastic joints," in *Theory of Robot Control*, C. Canudas de Wit, B. Siciliano, G. Bastin (Eds.), pp. 179–217, Springer-Verlag, Berlin, 1996.

[7] A. De Luca and B. Siciliano, "Flexible links," in *Theory of Robot Control*, C. Canudas de Wit, B. Siciliano, G. Bastin (Eds.), pp. 219–261, Springer-Verlag, Berlin, 1996.

[8] A. De Luca, "Nonlinear regulation of robot motion," *1st European Control Conf.*, pp. 1045–1050, 1991.

[9] A. De Luca and P. Lucibello, "A general algorithm for dynamic feedback linearization of robots with elastic joints," *1998 IEEE Int. Conf. on Robotics and Automation*, pp. 504–510, 1998.

[10] A. Isidori, *Nonlinear Control Systems*, 3rd Edition, Springer-Verlag, 1995.

[11] F. Bellezza, L. Lanari, and G. Ulivi, "Exact modeling of the slewing flexible link," *1990 IEEE Int. Conf. on Robotics and Automation*, pp. 734–739, 1990.

[12] A. De Luca and B. Siciliano, "Inversion-based nonlinear control of robot arms with flexible links," *AIAA J. of Guidance, Control, and Dynamics*, vol. 16, no. 6, pp. 1169–1176, 1993.

[13] A. De Luca, L. Lanari, and G. Ulivi, "End-effector trajectory tracking in flexible arms: Comparison of approaches based on regulation theory," in *Advanced Robot Control*, C. Canudas de Wit (Ed.) Lecture Notes in Control and Information Sciences, vol. 162, pp. 190–206, Springer-Verlag, Berlin, 1991.

[14] H. Yang, H. Krishnan, and M.H. Ang Jr., "A simple rest-to-rest control command fro a flexible link robot," *1997 IEEE Int. Conf. on Robotics and Automation*, pp. 3312–3317, 1997.

[15] E. Bayo, "A finite-element approach to control the end-point motion of a single-link flexible robot," *J. of Robotic Systems*, vol. 4, no. 1, pp. 63–75, 1987

[16] D.-S. Kwon and W.J. Book, "An inverse dynamic method yielding flexible manipulator state trajectories," *1990 American Control Conf.*, pp. 186–193, 1990.

[17] E. Bayo, M.A. Serna, P. Papadopoulus, and J. Stubbe, "Inverse dynamics and kinematics of multi-link elastic robots: An iterative frequency domain approach," *Int. J. of Robotics Research* vol. 8, no. 6, pp. 49–62, 1989.

[18] A. De Luca and R. De Simone, "Controllo di traiettoria per robot con bracci flessibili basato sull'inversione in frequenza della dinamica," *41° Conv. ANIPLA (Automation 1997)*, pp. 227–236, 1997.

[19] H. Zhao and D. Chen, "Exact and stable tip trajectory tracking for multi-link flexible manipulator," *32nd IEEE Conf. on Decision and Control*, pp. 1371–1376, 1993.

[20] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Trans. on Automatic Control*, vol. 41, no. 7, pp. 930–942, 1996.

[21] A. De Luca, S. Panzieri, and G. Ulivi, "Stable inversion control for flexible link manipulators," *1998 IEEE Int. Conf. on Robotics and Automation*, pp. 799–805, 1998.

[22] A. De Luca, L. Lanari, P. Lucibello, S. Panzieri, and G. Ulivi, "Control experiments on a two-link robot with a flexible forearm," *29th IEEE Conf. on Decision and Control*, pp. 520–527, 1990.

[23] A. De Luca and B. Siciliano, "Regulation of flexible arms under gravity," *IEEE Trans. on Robotics and Automation*, vol. 9, no. 4, pp. 463–467, 1993.