Proceedings of the 1998 IEEE
International Conference on Robotics & Automation
Leuven, Belgium • May 1998

Stable Inversion Control for Flexible Link Manipulators*

Alessandro De Luca

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

Stefano Panzieri Giovanni Ulivi

Dipartimento di Informatica e Automazione
Università di Roma Tre
Via della Vasca Navale 79, 00146 Roma, Italy
{panzieri,ulivi}@uniroma3.it

Abstract

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. 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. For a general class of multi-link flexible manipulators, three alternative computational algorithms are presented, all defined on the second-order robot dynamic equations. Trajectory tracking is obtained by adding a (partial) state feedback, within a nonlinear regulation approach. Experimental results are reported for the FLEXARM robot.

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, exciting an oscillatory behavior.

A source of vibration in robot manipulators is *link flexibility*, introduced by a long reach and slender/lightweight construction of the arm [1]. 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 (see, e.g., [2]).

In robotic systems with multiple flexible links, output trajectories are typically defined at the level of manipulator tip, i.e., beyond the structural flexibility. Standard tools for solving trajectory tracking problems in nonlinear systems, such as feedback linearization, input-output decoupling, or inversion control (see, e.g., [3]), are not sufficient in this case. In fact, 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 [4], the nonlinear equivalent of non-minimum phase zeros in a linear setting. Therefore, a straightforward application of inversion-based control would lead to a theoretically unbounded increase of the internal arm deformation and, in practice, to control saturation.

In this paper, we address the problem of reproduction of end-effector trajectories, with internal stability, using model-based state feedback control. Different approaches have already been presented in order to keep the internal arm deformation limited, while tracking the desired tip motion of multi-link manipulators: inversion in the frequency domain [5, 6], iterative learning control [7, 8], nonlinear regulation [9, 10], or a combination of these. We recognize here that the key common feature is the computation of *bounded link deformations* (with associated joint trajectories) allowing the desired motion of the manipulator tip.

After reviewing the basic modeling assumptions in sect. 2, we present in sect. 3 a unified framework for three alternative numerical solutions to the end-effector trajectory tracking problem. In sect. 4 we report experimental results obtained with the FLEXARM robot, a prototype two-link planar manipulator with a flexible forearm [11].

2 Dynamic Modeling

Consider an open kinematic chain structure, with a fixed base and N moving flexible links, interconnected by N rotational joints. As usual, the use of approximate finite-dimensional models is preferred for multi-link flexible manipulators. In the following, some as-

*This work is supported by *MURST 40%* and *CNR 95.00106.CT07* funds.

sumptions are made:

Assumption 1 *Link deformations are small, so that only linear elastic effects are present.*

Assumption 2 *For each link, flexibility is limited to the plane of nominal rigid motion, i.e., to the plane normal to the preceeding joint axis.*

While Assumption 1 is standard, Assumption 2 is introduced for simplifying the control task. It 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 ith link of length ℓ_i is modeled, using separation in time and space, as

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

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 $\theta \in I\!R^N$ be the vector of joint angular positions, and $\delta \in I\!R^{N_e}$ 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.

For the ease of presentation, we neglect gravity effects and introduce a further assumption:

Assumption 3 *The total kinetic energy of the system is evaluated in the undeformed configuration $\delta = 0$.*

The following dynamic model for control design is then obtained (see, e.g., [2] for details):

$$\begin{bmatrix} B_{\theta\theta}(\theta) & B_{\theta\delta}(\theta) \\ B_{\theta\delta}^T(\theta) & B_{\delta\delta} \end{bmatrix} \begin{bmatrix} \ddot{\theta} \\ \ddot{\delta} \end{bmatrix} + \begin{bmatrix} c_\theta(\theta,\dot{\theta},\dot{\delta}) \\ c_\delta(\theta,\dot{\theta}) \end{bmatrix} + \begin{bmatrix} 0 \\ D\dot{\delta} + K\delta \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \end{bmatrix}. \quad (2)$$

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, $K > 0$ 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 (2), because link deformations in eq. (1) are described in the reference frames clamped at each link base. By Assumption 3, the inertia matrix B, and thus also the Coriolis and centrifugal terms c, are independent of δ, while the velocity terms c_δ lose the quadratic dependence on $\dot{\delta}$. In addition, being $B_{\delta\delta}$ a constant, c_θ loses also the quadratic dependence on $\dot{\delta}$.

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

$$y = \theta + \Phi_e \delta, \quad (3)$$

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

$$\Phi_e = \text{block diag}\,\{\phi_{i1}(\ell_i)/\ell_i, \ldots, \phi_{i,N_{ei}}(\ell_i)/\ell_i\}\,. \quad (4)$$

The output y_i is a linear approximation of the angle pointing from the ith link base to its end. According to Assumption 2, the direct kinematics of the flexible manipulator, i.e., the position of the arm tip, can be written in terms of the components of y.

We finally point out that the model structure (2) holds for any finite-dimensional approximation of distributed flexibility. Moreover, in the presence of uniform mass distribution for each link, any dynamic model of the form (2) retains the same relevant control feature, namely the zero dynamics associated with output (3) is always unstable (see, however, [12]).

3 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)), because the zero dynamics is stable in this case [13]. The direct extension of an inversion control law to the tip output (3) leads to closed-loop instabilities, due to inadmissible feedback cancellation effects.

Frequency domain inversion has been proposed in [5, 6] 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 a linear model of one-link flexible arms) or in few iterations (in multi-link manipulators). *Learning control* has been applied in [7, 8] for iteratively building the input torque over repeated trials on the same desired output trajectory. Both approaches generate a non-causal input command: a non-zero torque 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 [9, 10]:

asymptotic output tracking is obtained by closing a stabilizing state feedback around a computed reference state trajectory. Finally, separation of stable and unstable zero dynamics and non-causal operation are the main features of the *stable inversion* approach proposed in [14, 15].

All the above methods share a common idea: in order to exactly reproduce an end-effector trajectory, the links of a flexible manipulator should undergo 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 [4], destroys exact tracking and may induce closed-loop instability.

Let $y_d(t)$ be a desired smooth reference trajectory for the tip, defined in a finite interval $[0, T]$. Using eq. (3), we can eliminate θ and $\dot{\theta}$ from the last N_e equations in (2), obtaining

$$B_{\delta\delta}\ddot{\delta} + D\dot{\delta} + K\delta + B_{\theta\delta}^T(y - \Phi_e\delta)\left(\ddot{y} - \Phi_e\ddot{\delta}\right) + c_\delta(y - \Phi_e\delta, \dot{y} - \Phi_e\dot{\delta}) = 0, \quad (5)$$

which is a dynamic constraint to be always satisfied by tip motion and link deformations. Plugging the desired evolution $y_d(t)$ in eq. (5) 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 (2) for defining the nominal input torque

$$\tau_d = B_{\theta\theta}(\theta_d)\ddot{\theta}_d + B_{\theta\delta}(\theta_d)\ddot{\delta}_d + c_\theta(\theta_d, \dot{\theta}_d, \dot{\delta}_d), \quad (6)$$

where

$$\theta_d(t) = y_d(t) - \Phi_e\delta_d(t) \quad (7)$$

is the required joint motion. As a result, the main bottleneck is the computation of a bounded solution $\delta_d(t)$ to eq. (5) (evaluated at $y = y_d(t)$). In the following, we present three alternative numerical methods.

3.1 Method 1: Approximate nonlinear regulation

In nonlinear output regulation [3], 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. (6), 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. (5), evaluated along the reference output evolution:

$$\begin{aligned} B_{\delta\delta}\ddot{\pi}(Y_d) &+ D\dot{\pi}(Y_d) + K\pi(Y_d) \\ &+ B_{\theta\delta}^T(y_d - \Phi_e\pi(Y_d))\left(\ddot{y}_d - \Phi_e\ddot{\pi}(Y_d)\right) \\ &+ c_\delta\left(y_d - \Phi_e\pi(Y_d), \dot{y}_d - \Phi_e\dot{\pi}(Y_d)\right) = 0. \end{aligned} \quad (8)$$

An approximate solution $\widehat{\pi}(Y_d)$ to eq. (8) can be obtained by a numerical approach, as in many practical nonlinear regulation problems [16]. If we use a class of basis approximating functions that are bounded in their arguments, $\widehat{\pi}(Y_d)$ will necessarily be a bounded function over time as long as the trajectory $y_d(t)$ and its derivatives are bounded.

In [9], a polynomial function of Y_d has been used as $\widehat{\pi}(Y_d)$. The nonlinear terms in eq. (8) are expanded in Taylor series and the constant coefficients in $\widehat{\pi}(Y_d)$ are determined through the polynomial identity principle. Computational savings are obtained via a recursive procedure, solving eq. (8) for increasing expansion orders until the final desired precision is obtained.

As noted in [15], this approach does not allow the use of noncausal inputs. Therefore, although the computed link deformation $\delta_d(t) = \widehat{\pi}(Y_d(t))$ may be different from zero at time $t = 0$, there is no way to *preload* the manipulator to such a value.

3.2 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 [5], 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 [6, 17].

The same idea can be used in a simpler fashion, namely by iterating the linearization process only on the flexible dynamics. For, expand the inertial term $B^T_{\theta\delta}(\cdot)$ around a joint configuration $\bar{\theta}$, e.g., the final one, as

$$B^T_{\theta\delta}(y - \Phi_e\delta) = B^T_{\theta\delta}(\bar{\theta}) + \tilde{B}^T_{\theta\delta}(y - \Phi_e\delta, \bar{\theta}) \quad (9)$$

and rewrite eq. (5) as

$$\bar{B}\ddot{\delta} + D\dot{\delta} + K\delta + f(y, \dot{y}, \ddot{y}, \delta, \dot{\delta}, \ddot{\delta}, \bar{\theta}) = 0, \quad (10)$$

with $\bar{B} = B_{\delta\delta} - B^T_{\theta\delta}(\bar{\theta})\Phi_e$, and

$$f = B^T_{\theta\delta}(y - \Phi_e\delta)\ddot{y} - \tilde{B}^T_{\theta\delta}(y - \Phi_e\delta, \bar{\theta})\Phi_e\ddot{\delta} + c_\delta(y - \Phi_e\delta, \dot{y} - \Phi_e\dot{\delta}). \quad (11)$$

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. (11), define the forcing term

   $$f^{(k)}(t) = f(y_d(t), \dot{y}_d(t), \ddot{y}_d(t), \delta^{(k)}(t), \dot{\delta}^{(k)}(t), \ddot{\delta}^{(k)}(t), \bar{\theta}), \quad (12)$$

   and find a bounded solution $\delta(t)$ of

   $$\bar{B}\ddot{\delta} + D\dot{\delta} + K\delta + f^{(k)}(t) = 0 \quad (13)$$

   using the FFT method as in [5]. Denote the solution as $\delta^{(k+1)}(t)$, defined for $t \in [-\Delta, T + \Delta]$.
3. If $\|\delta^{(k+1)}(t) - \delta^{(k)}(t)\| \leq \varepsilon_\delta$ 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.

3.3 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 [18]. 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 [7].

A similar approach is proposed here for the numerical solution of eq. (5). Again, we limit the learning process to the flexible dynamics. Instead of using the tracking error, define a *deformation torque error* as

$$e = B_{\delta\delta}\ddot{\delta} + D\dot{\delta} + K\delta + B^T_{\theta\delta}(y_d - \Phi_e\delta)\left(\ddot{y}_d - \Phi_e\ddot{\delta}\right) + c_\delta(y_d - \Phi_e\delta, \dot{y}_d - \Phi_e\dot{\delta}), \quad (14)$$

namely the left-hand side of eq. (5), evaluated on the desired output trajectory y_d. Indeed, an admissible link deformation history $\delta(t)$ satisfies

$$e(\delta, \dot{\delta}, \ddot{\delta}, t) = 0, \qquad \forall t \in [-\Delta, T + \Delta]. \quad (15)$$

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. (14), define

   $$e^{(k)}(t) = e(\delta^{(k)}(t), \dot{\delta}^{(k)}(t), \ddot{\delta}^{(k)}(t), t). \quad (16)$$

   If

   $$\|e^{(k)}(t)\| \leq \varepsilon_e, \qquad \forall t \in [-\Delta, T + \Delta], \quad (17)$$

   set $\delta_d(t) = \delta^{(k)}(t)$ and stop. Else, process the error $e^{(k)}(t)$ by finite-impulse response (FIR) filters as in [7], obtaining a filtered version $e^{(k)}_f(t)$ and its derivative $\dot{e}^{(k)}_f(t)$.
3. Update by the following PD-like learning rule

   $$\delta^{(k+1)}(t) = \delta^{(k)}(t) - K_{LP}\, e^{(k)}_f(t) - K_{LD}\, \dot{e}^{(k)}_f(t), \quad (18)$$

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

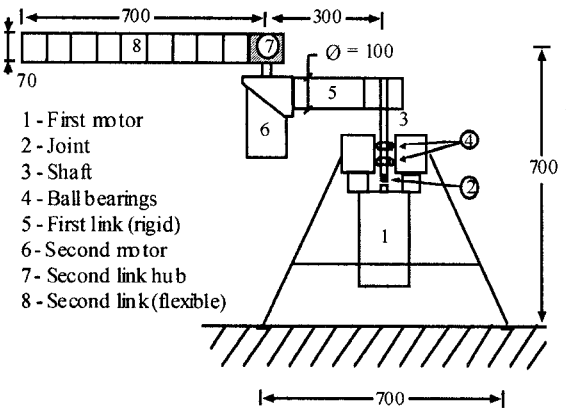

Figure 1: The FLEXARM

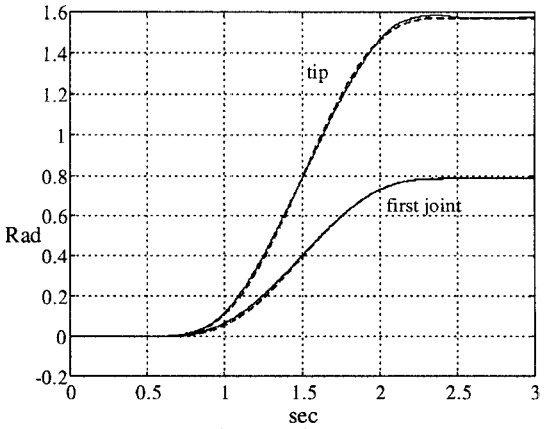

Figure 2: Desired (dashed) and actual trajectories

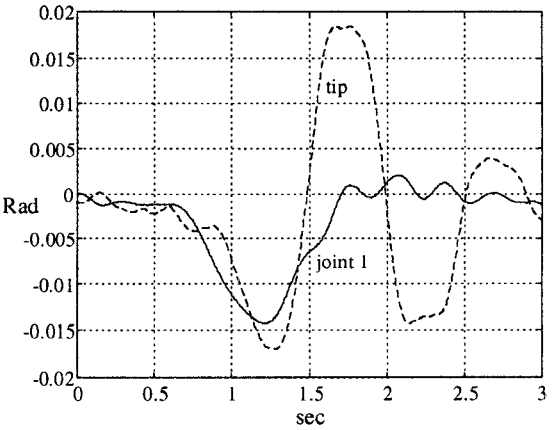

Figure 3: Output tracking errors

3.4 Final remarks

- 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 (13) and definition (14), respectively. In particular, the dependence on $\delta(t)$ in the forcing term $f^{(k)}$ of eq. (13), in view of Assumption 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 (18) can be seen as a step of the gradient method for solving eq. (15).

- In all methods, the evaluation of $\delta_d(t)$ and $\dot{\delta}_d(t)$ at time $t = 0$, together with the use of eq. (7) 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. (6) yields *exact* tracking of the end-effector trajectory.

- 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,

$$\tau = \tau_d + F \begin{bmatrix} \theta_d - \theta \\ \dot{\theta}_d - \dot{\theta} \\ \delta_d - \delta \\ \dot{\delta}_d - \dot{\delta} \end{bmatrix}, \qquad (19)$$

with τ_d given by eq. (6). One can also use a simpler stabilizing matrix F in eq. (19), as in the *partial state* feedback controller

$$\tau = \tau_d + F_P(\theta_d - \theta) + F_D(\dot{\theta}_d - \dot{\theta}), \qquad (20)$$

with positive definite (diagonal) matrices F_P and F_D [4]. Note that control (19) (or (20)) can be applied over the whole interval $[-\Delta, T+\Delta]$, yielding a more robust version of a non-causal trajectory regulator.

4 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 (see Fig. 1). The dynamic model of the arm can be found in [11], 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 vector is defined as (see eq. (3))

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

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

The control law is given by eq. (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 $\varepsilon_e = 10^{-6}$ Nmin eq. (17) is reached within 30 iterations on the nominal model.

Figure 2 shows the desired and actual trajectory for the two outputs, with maximum errors of about 0.8° and 1° respectively (see Fig. 3). The applied joint torques are given in Fig. 4. The small offset values are due to noise and to some residual gravity effects caused by imperfect balancing of the structure.

In Fig. 5, the computed velocity profiles of the two joints, $\dot{\theta}_{d1}$ and $\dot{\theta}_{d2}$ from eq. (7), 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. Similarly, the computed and the measured forearm deformation at the tip are reported in Fig. 6. The differences come from the inaccuracy of the model used for control computations. Before $t = 0.5$ s and after $t = 2.5$ s, the preloading and discharging effects can be appreciated.

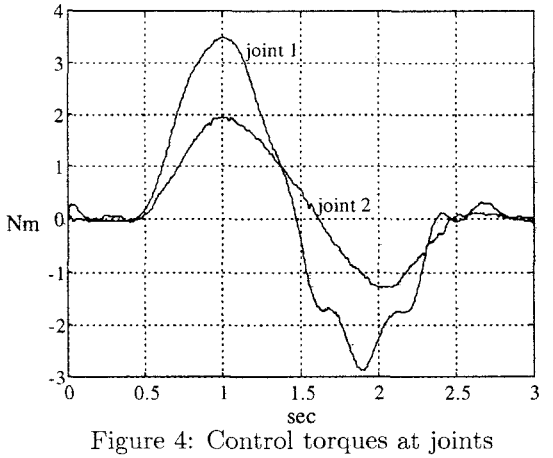

Figure 4: Control torques at joints

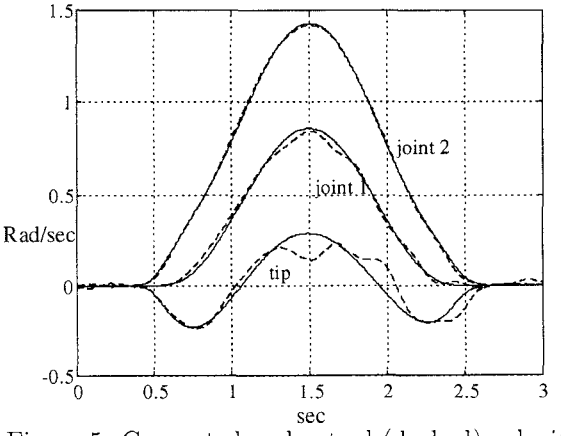

Figure 5: Computed and actual (dashed) velocities

5 Conclusions

The problem of designing controllers that make use of stable model inversion has been considered. Three numerical methods based on approximate nonlinear regulation, frequency domain learning, and time domain learning can be effectively used been to compute the bounded deformation history of a flexible arm manipulator whose tip trajectory has been assigned. The last two methods are able to generate the non-causal nominal input torque required to preload the flexible link to the proper initial state that enables reproduction of the desired trajectory. The experimental results show the good level of tracking accuracy that can be obtained in practice.

References

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

[2] W.J. Book, "Recursive Lagrangian dynamics of flexible manipulator arms," *Int. J. of Robotics Research*, vol. 3(3), pp. 87–101, 1984.

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

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

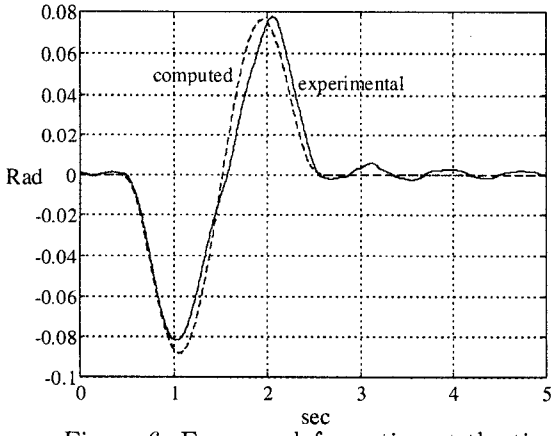

Figure 6: Forearm deformation at the tip

[5] E. Bayo, "A finite-element approach to control the end-point motion of a single-link flexible robot," *J. of Robotic Systems*, vol. 4, pp. 63–75, 1987.

[6] 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(6), pp. 49–62, 1989.

[7] S. Panzieri and G. Ulivi, "Disturbance rejection of iterative learning control applied to trajectory tracking for a flexible manipulator," *3rd European Control Conf.*, Roma, I, pp. 2374–2379, 1995.

[8] P. Lucibello and S. Panzieri, "End point trajectory control with internal stability of a flexible link by learning," *1996 IEEE Int. Conf. on Robotics and Automation*, Minneapolis, MN, pp. 2117–2123, 1996.

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

[10] P. Lucibello and M.D. Di Benedetto, "Output tracking for a nonlinear flexible arm," *ASME J. of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 78–85, 1993.

[11] 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*, Honolulu, HI, pp. 520–527, 1990.

[12] A. De Luca, P. Lucibello, and G. Ulivi, "Inversion techniques for trajectory control of flexible robot arms," *J. of Robotic Systems*, vol. 6, pp. 325–344, 1989.

[13] 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, pp. 1169–1176, 1993.

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

[15] S. Devasia, D. Chen, and B. Paden, "Nonlinear inversion-based output tracking," *IEEE Trans. on Automatic Control*, vol. 41, pp. 930–942, 1996.

[16] J. Huang and W.J. Rugh, "An approximation method for the nonlinear servomechanism problem," *IEEE Trans. on Automatic Control*, vol. 37, pp. 1395–1398, 1992.

[17] A. De Luca and R. De Simone, "Controllo di traiettoria per robot con bracci flessibili basato sull'inversione in frequenza della dinamica," *41st ANIPLA National Conf.*, Torino, I, pp. 227–236, 1997.

[18] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robots by learning," *J. of Robotic Systems*, vol. 1, pp. 123–140, 1984.