Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

Learning Control for Redundant Manipulators

Alessandro De Luca Francesco Mataloni

Dipartimento di Informatica e Sistemistica
Università degli Studi di Roma "La Sapienza"
Via Eudossiana 18, 00184 Roma, Italy

Abstract

An iterative scheme is proposed for learning the input torques that produce a specified repetitive end-effector trajectory for a redundant manipulator, without explicit knowledge of the robot dynamic model. The approach does not rely on a specific inverse kinematic solution. A main aspect of the learning problem for redundant robots is that the number of driving error signals (in the task space) is strictly less than the number of inputs to be determined (in the joint space). During the iterative process, control effort is transferred from a linear feedback law designed for the end-effector error to the learned feedforward term. Stabilization of the closed-loop system is obtained by joint velocity damping. The basic learning algorithm is designed in the frequency domain, while a digital implementation of the necessary signal filtering improves learning performance, both in speed and uniformity of convergence. Simulations are reported for a three-link planar manipulator. The inclusion of a kinematic singularity avoidance scheme is also illustrated.

1. Introduction

Accurate trajectory tracking can be achieved in robot arms using nonlinear state-feedback or feedforward model-based control laws. These are commonly denoted as computed torque or inverse dynamics techniques [1]. In presence of uncertainties, the control design should include a robustifying strategy, ranging from variable structure to adaptive control solutions [2]. When parametric uncertainty is large or unmodeled dynamics become relevant, these techniques may produce unsatisfactory results, especially during transients.

If the required task is repetitive in nature, as in many robotic industrial operations, an appealing alternative for dealing with largely unknown plants is provided by learning control [3–5]. In this approach, the input torque required to move the robot arm along a specified trajectory is built iteratively from successive experiments. Starting from the closed-loop error obtained at the current trial, a feedforward signal is computed off line and stored for successive application. For conventional rigid robots, convergence conditions have been obtained and the learning technique has been successfully implemented for reproducing both joint and cartesian trajectories. It can be shown that the learning process asymptotically inverts the plant dynamics along the given trajectory [6]. Thus, learning can be seen as a method for obtaining the nominal torque, without explicitly computing through the (uncomplete or uncertain) dynamic model.

Recently, increasing interest has been devoted to the design and control of redundant manipulators, i.e. having a number of degrees of freedom larger than the one strictly needed for performing a given task. The extra motion capabilities of such robotic structures offer advantages in executing dexterous tasks in complex industrial and non-conventional environments, and have been used for avoiding obstacles in the configuration space [7], for maximizing arm manipulability [8], or for uniformly distributing torque demand among the joints [9].

In general, no closed-form solution exists to the inverse kinematic problem for redundant robots. Therefore, several numerical algorithms have been proposed for computing joint paths from cartesian ones, based on differential motion. Redundancy is then exploited for optimizing various criteria, using projected [10] or reduced gradient [11] methods. However, when joint trajectories are generated using only the kinematic model of the robot and local information on the end-effector path, dynamic performance is generally unpredictable [12]. In order to obtain overall satisfactory joint torques, exact knowledge of the dynamic model of the robot is needed. Moreover, only global resolution schemes are able to guarantee stable and bounded joint solutions [13]. These requirements and techniques are somewhat restrictive since robot dynamic parameters are known only with a large degree of approximation and global schemes are computationally intensive, asking for the solution of two-point boundary value problems.

As an alternate control approach, it is proposed here to extend learning to redundant manipulators, so to achieve a desired end-effector motion without restraining the arm to a particular feasible joint trajectory.

This strategy combines the inherent simplicity of learning algorithms with the possibility of avoiding an explicit inverse kinematic solution. Moreover, since learning is intrinsically defined on the whole trajectory, the resulting torques will typically display global stability properties.

In the presence of redundancy, the learning problem has the following peculiarities:

- the number of driving signals, i.e. the error components in the task space, is *strictly less* than the number of input joint torques to be determined, implying a sort of 'loose' excitation for learning purposes;
- a feedback controller should be available that stabilizes the robot motion at the *joint space* level, without explicit knowledge of the arm mechanical characteristics.

As a first consequence, the control action has to be iteratively acquired as a generalized force in the task space. Transformation of the obtained cartesian forces into applied joint torques will then be made using the Jacobian transpose of the direct kinematics. On the other hand, the stabilization feature is needed in order to prove convergence of standard learning algorithms [5]. It is known that, in the absence of gravity, a proportional-derivative feedback on the position error (i.e. a model-independent law) guarantees asymptotic stability for conventional robots [14]. Under the same assumption, a simple modification is given here for redundant manipulators. In particular, asymptotic stability of arm configurations will be obtained using PD cartesian error feedback plus joint velocity damping. The gravity case can be handled as usual, adding integral control action.

The learning algorithm that will be used is a refined version of a frequency-based one proposed in [6], which generates the feedforward term to be applied at the next trial by taking advantage both of the control effort at the current trial and of the learning memory. For linear systems, it has been shown that convergence conditions can be derived which are weaker than the standard ones. The basic algorithm is designed as a filter in the frequency domain. Use of digital techniques in the actual implementation of the learning filters leads in general to a better performance [15]. A Weighted Centered Arithmetic Mean (WCAM) processing of sampled data is presented, which avoids oscillations in the learning process and increases the speed of convergence.

A three-link planar arm with rotational joints is considered in this paper as a case study. Simulation results are reported for straight and curved cartesian paths. The inclusion of a singularity avoidance scheme within the learning process is also evaluated.

2. Preliminaries

The dynamics of a rigid robot arm with N joints is

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{S}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{u}, \qquad (1)$$

where $\mathbf{q} \in I\!R^N$ are the generalized joint coordinates, $\mathbf{B}$ is the positive definite inertia matrix, $\mathbf{S}\dot{\mathbf{q}}$ contains the centripetal and Coriolis terms, and $\mathbf{g}$ is the gravitational force vector. Matrix $\mathbf{S}$ can be chosen so that $\dot{\mathbf{B}} - 2\mathbf{S}$ is anti-symmetric. Let the task $\mathbf{p}_d(t)$ be specified in terms of end-effector coordinates $\mathbf{p} \in I\!R^M$, related to the joint coordinates by the direct kinematics

$$\mathbf{p} = \mathbf{k}(\mathbf{q}), \qquad \mathbf{k}: I\!R^N \rightarrow I\!R^M. \qquad (2)$$

Since $M < N$, the manipulator Jacobian $\mathbf{J}$ in

$$\dot{\mathbf{p}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \qquad (3)$$

is a non-square $M \times N$ matrix. The cartesian error $\mathbf{e}$ is defined as

$$\mathbf{e} = \mathbf{p}_d - \mathbf{p}. \qquad (4)$$

Assume now that the robot lives in a constant gravitational field. All configurations $(\mathbf{q},\dot{\mathbf{q}})$ with $\dot{\mathbf{q}} = \mathbf{0}$ are then equilibrium states of the undriven system (1). The following theorem provides a feedback law that globally stabilizes the redundant manipulator at a fixed cartesian point $\mathbf{p}_d$, with zero joint velocity, independently from the particular model parameters.

Theorem. *Consider the dynamics (1), with* $\mathbf{g}(\mathbf{q}) = \mathbf{0}$, *and assume that* rank $\mathbf{J} = M$ *in (3). Then, the control law*

$$\mathbf{u} = \mathbf{J}^{\mathrm{T}}(\mathbf{q})\left[\mathbf{K}_P\mathbf{e} + \mathbf{K}_V\dot{\mathbf{e}}\right] - \mathbf{K}_D\dot{\mathbf{q}}, \qquad (5)$$

with symmetric $\mathbf{K}_P > \mathbf{O}$, $\mathbf{K}_V \geq \mathbf{O}$, *and* $\mathbf{K}_D > \mathbf{O}$, *is such that* $\dot{\mathbf{q}} \rightarrow \mathbf{0}$ *and* $\mathbf{e} \rightarrow \mathbf{0}$.

Proof. Define the Lyapunov candidate

$$V = \frac{1}{2}\mathbf{e}^{\mathrm{T}}\mathbf{K}_P\mathbf{e} + \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{B}(\mathbf{q})\dot{\mathbf{q}}. \qquad (6)$$

Since $\dot{\mathbf{e}} = -\dot{\mathbf{p}} = -\mathbf{J}\dot{\mathbf{q}}$, substituting (5) into (1), and using the antisymmetry of $\dot{\mathbf{B}} - 2\mathbf{S}$, the time derivative $\dot{V}$ becomes

$$\begin{aligned}\dot{V} &= \mathbf{e}^{\mathrm{T}}\mathbf{K}_P\dot{\mathbf{e}} + \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}\dot{\mathbf{B}}\dot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathrm{T}}\mathbf{B}\ddot{\mathbf{q}} \\ &= -\mathbf{e}^{\mathrm{T}}\mathbf{K}_P\mathbf{J}\dot{\mathbf{q}} + \frac{1}{2}\dot{\mathbf{q}}^{\mathrm{T}}(\dot{\mathbf{B}} - 2\mathbf{S})\dot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathrm{T}}\mathbf{u} \\ &= -\dot{\mathbf{q}}^{\mathrm{T}}(\mathbf{J}^{\mathrm{T}}\mathbf{K}_V\mathbf{J} + \mathbf{K}_D)\dot{\mathbf{q}} < 0, \qquad \text{for } \dot{\mathbf{q}} \neq \mathbf{0}.\end{aligned} \qquad (7)$$

When $\dot{\mathbf{q}} = \mathbf{0}$, the closed-loop system is described by $\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{J}^{\mathrm{T}}(\mathbf{q})\mathbf{K}_P\mathbf{e}$, so that $\ddot{\mathbf{q}} = \mathbf{0}$ iff $\mathbf{K}_P\mathbf{e} \in \mathcal{N}(\mathbf{J}^{\mathrm{T}})$. By the rank assumption on $\mathbf{J}$, the arm cannot get stucked with $\mathbf{e} \neq \mathbf{0}$, eventually coming to a rest configuration with zero cartesian error. Q.E.D.

It should be pointed out that the final configuration $\mathbf{q}^*$ reached by the robot under the action of (5) is not a specific one, apart from being such that $\mathbf{k}(\mathbf{q}^*) = \mathbf{p}_d$. Thus, the above result shows asymptotic attraction of closed-loop trajectories toward the invariant set of zero-velocity configurations associated to the desired end-effector location. This is a desirable feature rather than a drawback, because no a priori configuration is imposed to the arm, apart from the one that will be produced by the learning process itself. Note also that the $N \times N$ matrix $\mathbf{J}^T\mathbf{K}_V\mathbf{J}$ in (7) is only positive semi-definite, having at most rank $M < N$. Therefore, the damping $\mathbf{K}_D$ on the joint velocity is strictly needed to prove asymptotic stability. Viceversa, one may also set $\mathbf{K}_V = 0$ in (5).

3. Learning Algorithm

3.1 The basic scheme

Following [6], the general input-output structure of the proposed learning control is given in Fig. 1. For ease of description, consider a linear scalar plant with transfer function $p(s)$. Let $y_d(t)$ be the desired repetitive time behavior for the system output $y(t)$, and $e(t) = y_d(t) - y(t)$ the error. An error feedback controller with transfer function $c(s)$ (e.g. a PD law) is closed around the system, with the main purpose of stabilization. The resulting closed-loop transfer function is denoted by $w(s)$.

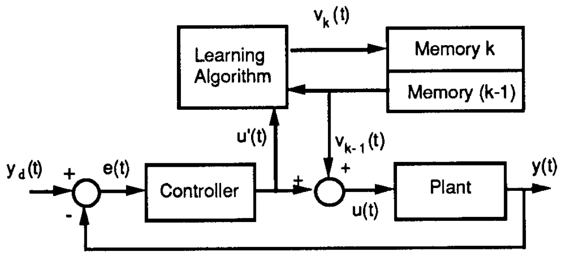

Fig. 1 – General block diagram of learning control

At the kth trajectory trial, a previously stored memory contribution $v_{k-1}(t)$ is added to the feedback controller output, so to generate the plant input $u(t) = u'(t) + v_{k-1}(t)$. Note that the control effort $u'(t)$ of the closed-loop controller is present only for a non-zero output error $e(t)$. The system output can be rewritten in the Laplace domain as

$$y(s) = w(s)y_d(s) + \frac{p(s)}{1 + p(s)c(s)}v_{k-1}(s). \tag{8}$$

The update v_k of the learning process (that will be the feedforward at trial $k+1$) is chosen as

$$v_k(s) = \alpha(s)u'(s) + \beta(s)v_{k-1}(s), \tag{9}$$

with properly selected filters $\alpha(s)$ and $\beta(s)$. Due to the repetitive nature of the desired output, it can be shown that the iterative process $\{v_k\}$ will converge for all values of $s = j\omega$ such that

$$|\beta(s) - \alpha(s)w(s)| < 1. \tag{10}$$

The above contraction condition generalizes the one found in [3], where $\beta(s) \equiv 1$. It guarantees the existence of limit values ($k \to \infty$), both for the error and the memory signals:

$$\begin{aligned} e_\infty(s) &= \frac{1 - \beta(s)}{1 - \beta(s) + \alpha(s)w(s)} \frac{y_d(s)}{1 + p(s)c(s)}, \\ v_\infty(s) &= \frac{\alpha(s)w(s)}{1 - \beta(s) + \alpha(s)w(s)} \frac{y_d(s)}{p(s)}. \end{aligned} \tag{11}$$

When $\beta(s) \equiv 1$, these collapse to

$$e_\infty(s) = 0, \qquad v_\infty(s) = \frac{y_d(s)}{p(s)}, \tag{12}$$

showing that an exact learning is obtained ($e_\infty(t) = 0$) and that learning asymptotically inverts the plant along the given reference trajectory. If $\beta(s)$ is restricted to be one, the convergence condition (10) may not hold preventing to obtain even part of the nice properties expressed by (12). Introduction of a non unitary and frequency-dependent $\beta(j\omega)$ enforces convergence of the learning process, in the face of a tracking performance degradatation. Nevertheless, frequency components of the reference y_d lying in the range where $|\beta(j\omega)| = 1$ will still be exactly reproduced.

Condition (10) is a guide in the choice of α and β. With this respect, a very helpful tool is the Nyquist diagram of $\beta(j\omega) - \alpha(j\omega)w(j\omega)$, drawn from experimental data or using a nominal $w(s)$ (see [6] for examples). Note that the dynamic behavior $w(s)$ of the closed-loop system may be dominated by the chosen feedback controller $c(s)$, resulting in an easier and more successfull design of $\alpha(s)$ and $\beta(s)$.

3.2 Improvements

The above reasoning can be applied to a linearized and decoupled version of a multi-input multi-output nonlinear plant, viz. the robot arm. Being the design based on a linear and decoupled approximation of the plant, even when condition (10) is satisfied nominally, there still could be stability problems in the real learning process. In the robotic application, convergence proofs in [3,5] show that this is not the case, at least in principle. These proofs extend directly to the generalized learning algorithm (9). In practice, the presence of transmission elements with a high reduction ratio generally masks the nonlinear and interacting part of arm dynamics. Thus, the design of a learning control based on the above linear approximation generally results satisfactory. However, space is still left for speeding up the whole process.

Some modifications of the basic learning scheme are proposed next, that were found valuable in improving convergence speed and in considerably reducing the oscillatory behavior during learning. The main idea in choosing $\alpha(s)$ and $\beta(s)$ is to shape the phase response of the system, compensating also for the lag typically introduced by the closed-loop controller but without modifying the overall gain. Based on this, the design proceeds as follows:

- $\alpha(s)$ is designed in the frequency-domain, based on the best available linearized model;
- $\beta(s)$ is realized in discrete-time as a WCAM filter, weighting the input samples so to attenuate high-frequency content of the output signal;
- a backward shifting of Δ future samples of the output of filter $\alpha(s)$ is included, so to further improve learning speed.

Note that non-causal operations are feasible in the above signal processing, since the learning algorithm is performed off line at the end of each trajectory trial.

A CAM filter of order n uses $2n+1$ input samples located symmetrically with respect to the current time instant, producing an averaged output with mean value centered at the current sample. A WCAM filter uses symmetric weights a_i on the input data. If the sampling time is T_c, this filter can be characterized by the following transfer function:

$$W^{(n)}_{WCAM}(s) = \frac{a_0}{2n+1} + \sum_{i=1}^{n} \frac{a_i e^{-siT_c} + a_i e^{siT_c}}{2n+1}. \quad (13)$$

Setting $\omega_c = 2\pi/T_c$, since

$$W^{(n)}_{WCAM}(j\omega) = \frac{a_0}{2n+1} + \sum_{i=1}^{n} \frac{2a_i \cos(\frac{2\pi i\omega}{\omega_c})}{2n+1} \quad (14)$$

is a real function, $W^{(n)}_{WCAM}(j\omega)$ will introduce no phase modification to the input signal — apart from a possible sign inversion. The set of weights in (14) will be determined by imposing proper conditions on the magnitude, in particular

$$W^{(n)}_{WCAM}(0) = 1, \; W^{(n)}_{WCAM}(j\omega_k) = 0, \; k = 1, \ldots, n, \quad (15)$$

for n selected angular frequencies ω_k. In alternative, some of the free coefficients could be used to impose zero values also to derivatives of (16). Figure 2 shows, for $n = 2$ and as a function of ω/ω_c, a comparison between the magnitude of an unweighted (dashed line). and of two types weighted filters. The first weighted filter (continuous line) imposes zero values at $\omega_1 = \omega_c/3$ and $\omega_2 = \omega_c/2$, while the second one (dotted line) is obtained by setting

$$W^{(2)}_{WCAM}(j\frac{\omega_c}{2}) = 0, \; \frac{d^2 W^{(2)}_{WCAM}(j\omega)}{d\omega^2}\Big|_{\omega=\omega_c/2} = 0, \quad (16)$$

resulting in

$$a_0 = 1.875, \quad a_1 = 1.25, \quad a_2 = 0.3125. \quad (17)$$

With these coefficients, the transfer function $W^{(2)}_{WCAM}$ is always positive and no sign inversion occurs.

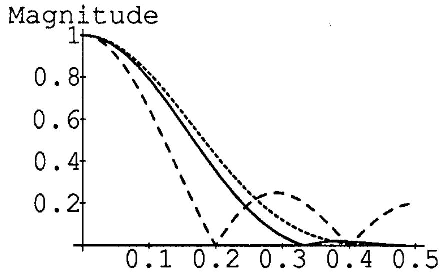

Fig. 2 – Comparison between CAM and WCAM

As for the shifting step Δ used to modify $\alpha(s)$, a simple optimization procedure was implemented for determining its optimal value with respect to the maximum allowable bandwidth.

3.3 Overall control law

Combining the stabilizing feedback controller (5) with the learning algorithm (9), designed , and introducing the above modifications yields the proposed learning control scheme for redundant manipulators. For cartesian learning, the on-line control law is

$$\mathbf{u} = \mathbf{J}^{\mathrm{T}}(\mathbf{q})[\mathbf{K}_P\mathbf{e} + \mathbf{K}_V\dot{\mathbf{e}} + \mathbf{v}_{k-1}] - \mathbf{K}_D\dot{\mathbf{q}}, \quad (18)$$

while the off-line learning update is for each scalar cartesian component $(j = 1, \ldots, M)$

$$\begin{aligned} v_{j,k}(s) &= W^{(n)}_{j,WCAM}(s)[\widehat{\alpha}_j(s)u'_j(s) + \widehat{\beta}_j(s)v_{j,k-1}(s)], \\ \widehat{\alpha}_j(s) &= e^{s\Delta_j}\alpha_j(s), \quad (19) \\ \widehat{\beta}_j(s) &= W^{(n)}_{j,WCAM}(s). \end{aligned}$$

The closed-loop control effort is $\mathbf{u}' = \mathbf{u} - \mathbf{J}^T(\mathbf{q})\mathbf{v}_{k-1}$. In the first trajectory trial ($k = 1$) a feedforward $\mathbf{v}_0(t)$ is used, containing a priori information on the system. Note that an additional WCAM filter is included in (19) for generating the next learning feedforward.

This scheme can be further modified by introducing a rotation matrix $\mathbf{R}(\mathbf{p}_d)$, relating cartesian position errors and commands to their counterparts in task coordinates, i.e. in terms of local tangential, normal and binormal components with respect to the desired path. A cleaner tuning of the end-effector behavior as well as independence from path orientation are gained in this way. The final block diagram of the learning controller is reported in Fig. 3.

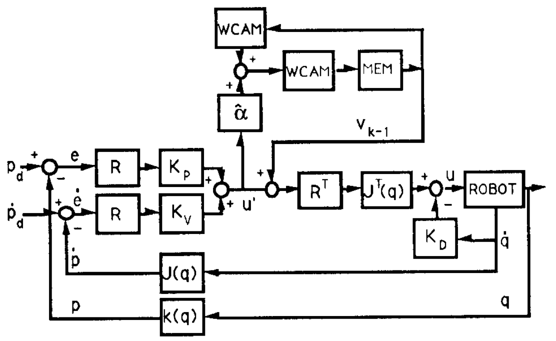

Fig. 3 – Learning control for redundant robots

4. Case Study

Consider a $3R$ planar robot arm, moving on a horizontal plane ($\mathbf{g}(\mathbf{q}) = \mathbf{0}$) and executing end-effector trajectories. The dynamic model results in more compact equations when given in terms of absolute joint angles q_i, in place of the standard Denavit-Hartenberg angles θ_i (see Fig. 4). Indeed,

$$\mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \theta = \mathbf{T}\theta, \qquad (20)$$

so that torques $\mathbf{u}_\theta$ produced by the motors give rise to generalized forces $\mathbf{u} = \mathbf{T}^{-T}\mathbf{u}_\theta$ to be used in (1). The elements of the symmetric inertia matrix $\mathbf{B}(\mathbf{q})$ are

$$\begin{aligned} b_{11} &= I_1 + m_1 d_1^2 + (m_2 + m_3)\ell_1^2 \\ b_{12} &= (m_2 d_2 + m_3 \ell_2)\ell_1 \cos(q_2 - q_1) \\ b_{13} &= m_3 d_3 \ell_1 \cos(q_3 - q_1) \\ b_{22} &= I_2 + m_2 d_2^2 + m_3 \ell_2^2 \\ b_{23} &= m_3 d_3 \ell_2 \cos(q_3 - q_2) \\ b_{33} &= I_3 + m_3 d_3^2 \end{aligned} \qquad (21)$$

while the velocity terms $\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ are

$$\begin{aligned} h_1 &= -(m_2 d_2 + m_3 \ell_2)\ell_1 \sin(q_2 - q_1)\dot{q}_2^2 \\ &\quad - m_3 d_3 \ell_1 \sin(q_3 - q_1)\dot{q}_3^2 \\ h_2 &= (m_2 d_2 + m_3 \ell_2)\ell_1 \sin(q_2 - q_1)\dot{q}_1^2 \\ &\quad - m_3 d_3 \ell_2 \sin(q_3 - q_2)\dot{q}_3^2 \\ h_2 &= m_3 d_3 \ell_1 \sin(q_3 - q_1)\dot{q}_1^2 + m_3 d_3 \ell_2 \sin(q_3 - q_2)\dot{q}_2^2 \end{aligned} \qquad (22)$$

where m_i and ℓ_i are mass and length of link i, I_i is its moment of inertia w.r.t. the center of mass, and d_i is the distance of the center of mass of link i from the axis of joint i. Apart from the link length, the rest of these quantities is unknown.

The Jacobian of the arm kinematics in absolute coordinates is

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} -\ell_1 \sin q_1 & -\ell_2 \sin q_2 & -\ell_3 \sin q_3 \\ \ell_1 \cos q_1 & \ell_2 \cos q_2 & \ell_3 \cos q_3 \end{bmatrix}. \qquad (23)$$

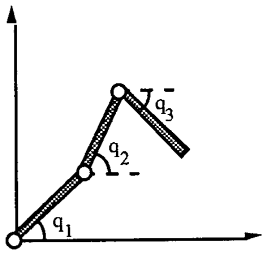

Fig. 4 – 3R planar robot arm

The first design step is the selection of the gain matrices in the stabilizing controller (18). Any choice $\mathbf{K}_P > \mathbf{0}$, $\mathbf{K}_V \geq \mathbf{0}$, $\mathbf{K}_D > \mathbf{0}$ is feasible. When using rotation matrices $\mathbf{R}$ as in Fig. 3, diagonal $\mathbf{K}_P$ and $\mathbf{K}_V$ will affect separately the dynamic behavior along the tangent and along the normal of the cartesian trajectory. On the other hand, a diagonal damping matrix $\mathbf{K}_D$ in joint coordinates θ results in

$$\mathbf{K}_D = \begin{bmatrix} D_1 + D_2 & -D_2 & 0 \\ -D_2 & D_2 + D_3 & -D_3 \\ 0 & -D_3 & D_3 \end{bmatrix}, \qquad (24)$$

when absolute coordinates $\mathbf{q}$ are used.

The actual synthesis of learning filters will be made using the linearized model of the robot arm around a fixed point $(\mathbf{q}, \dot{\mathbf{q}}) = (\bar{\mathbf{q}}, \mathbf{0})$, $\mathbf{u} = \mathbf{0}$. Define $\bar{\mathbf{J}} = \mathbf{J}(\bar{\mathbf{q}})$, $\bar{\mathbf{B}} = \mathbf{B}(\bar{\mathbf{q}})$, and let the linear feedback controller (including task rotation $\bar{\mathbf{R}} = \mathbf{R}(\mathbf{k}(\bar{\mathbf{q}}))$) be

$$\Delta\mathbf{u} = \bar{\mathbf{J}}^T\bar{\mathbf{R}}^T[\mathbf{K}_P\bar{\mathbf{R}}\Delta\mathbf{e} + \mathbf{K}_V\bar{\mathbf{R}}\Delta\dot{\mathbf{e}}] - \mathbf{K}_D\Delta\dot{\mathbf{q}}, \quad (25)$$

with an obvious meaning of the incremental vectors. The closed-loop transfer matrix from $\Delta \mathbf{p}_d$ to $\Delta \mathbf{p}$ is the full 2×2 matrix

$$\mathbf{W}(s) = \bar{\mathbf{J}}\left[s^2\bar{\mathbf{B}} + s(\bar{\mathbf{J}}^{\mathrm{T}}\widehat{\mathbf{K}}_V\bar{\mathbf{J}} + \mathbf{K}_D) + \bar{\mathbf{J}}^{\mathrm{T}}\widehat{\mathbf{K}}_P\bar{\mathbf{J}}\right]^{-1}\bar{\mathbf{J}}^{\mathrm{T}} \cdot \left(s\widehat{\mathbf{K}}_V + \widehat{\mathbf{K}}_P\right), \tag{26}$$

with $\widehat{\mathbf{K}}_P = \bar{\mathbf{R}}^{\mathrm{T}}\mathbf{K}_P\bar{\mathbf{R}}$, $\widehat{\mathbf{K}}_V = \bar{\mathbf{R}}^{\mathrm{T}}\mathbf{K}_V\bar{\mathbf{R}}$. For $j = 1, 2$, the scalar filters $\widehat{\alpha}_j(s)$ and $\widehat{\beta}_j(s)$, will be designed considering only the diagonal elements of $\mathbf{W}(s)$, evaluated using nominal data. The following simple parametric forms will be assumed:

$$\begin{aligned}\widehat{\alpha}_j(s) &= e^{s\Delta_j}k_{\alpha,j}\frac{1+s\tau_{z,j}}{1+s\tau_{p,j}},\\ \widehat{\beta}_j(s) &= W^{(1)}_{j,WCAM}(s) = \frac{1}{2} + \frac{e^{-sT_c}+e^{sT_c}}{4}.\end{aligned} \tag{27}$$

It should be stressed that the above system linearization acts only as a convenient support in the synthesis of the learning filters, but it is not strictly necessary.

5. Simulation Results

The following robot data have been used in simulation:

$$\begin{aligned}\ell_i &= 0.5\,\mathrm{m}, & d_i &= 0.25\,\mathrm{m},\\ m_i &= 10\,\mathrm{kg}, & I_i &= 0.2083\,\mathrm{kgm}^2,\end{aligned} \quad i = 1, \ldots, 3. \tag{28}$$

The feedback controller (5) was designed with:

$$\begin{aligned}&k_{P,j} = 200\,\mathrm{N/m}, \quad k_{V,j} = 40\,\mathrm{Nsec/m}, \quad j = 1, 2;\\ &D_1 = D_2 = 30\,\mathrm{Nmsec}, \quad D_3 = 20\,\mathrm{Nmsec}.\end{aligned} \tag{29}$$

Finally, the learning algorithm is implemented using:

$$\begin{aligned}k_{\alpha,j} &= 0.5, & \tau_{p,j} &= 0.0083\,\mathrm{sec},\\ \tau_{z,j} &= 0.1\,\mathrm{sec}, & \Delta_j &= 0.025\,\mathrm{sec},\end{aligned} \quad j = 1, 2. \tag{30}$$

The sampling time is $T_c = 0.025$ sec.

The learning control was tested on two types of cartesian trajectories with different timings. The first one is a straight line joining (0.5, 0) to (0.1, 0.4) in 1 sec, with a bang-coast-bang acceleration law and zero initial and final velocity:

$$\begin{aligned}&p_x(s) = 0.5 - \frac{s}{\sqrt{2}},\ p_y(s) = \frac{s}{\sqrt{2}}, \quad s \in [0, 0.4\sqrt{2}]\\ &\ddot{s}(t) = \begin{cases} 2.5\sqrt{2}, & t \in [0, 0.2]\\ 0, & t \in [0.2, 0.8]\\ -2.5\sqrt{2}, & t \in [0.8, 1]\end{cases}\end{aligned} \tag{31}$$

The robot arm starts at rest from the configuration $\mathbf{q} = (-\pi/2, 0, \pi/2)$. The sequence of executed end-effector paths during successive trials is shown in Fig. 5 for a 1 sec time interval, while RMS values of the normal and tangential errors versus trials are presented in Fig. 6, showing an improvement of two orders of magnitude. The proposed control algorithm learns the required input torque in about 10 iterations, starting with $\mathbf{v}_0 = \mathbf{0}$. The motion of the whole robot arm at the first and tenth trials are displayed in Figs. 7 and 8, where the obtained smooth behavior is a result of the inclusion of joint velocity damping. The transfer of control effort from feedback control to feedforward learning is quite evident when comparing Fig. 9 with 11, and Fig. 10 with 12, representing respectively the tangential and the normal force components in the task space. Final non-zero values in Figs. 9–10 correspond to the presence of a residual end-effector position error at time $t = 1$ sec. On the other hand, the learned signals build up the missing feedforward contribution, even at the initial and final instants. Smoothing effects of the learning algorithm can be clearly appreciated in the obtained signals. Figure 13 shows the actual torques $\mathbf{u}_\theta = \mathbf{T}^{\mathrm{T}}\mathbf{u}$ that are applied at each joint in the final trial, with $\mathbf{u}$ given by (18). Note that, once the end-effector has reached the terminal point, the arm may still have some internal self-motion but this will soon be damped out by the term $-\mathbf{K}_D\dot{\mathbf{q}}$.

The second test trajectory is a parabolic path joining (0, 0.5) to (0.5, 1) in 1 sec, with sinusoidal velocity:

$$\begin{aligned}&p_x(s) = 0.5s,\ p_y(s) = 0.5 + 0.5s^4, \quad s \in [0, 1]\\ &\dot{s}(t) = 0.5\pi \sin \pi t, \quad t \in [0, 1]\end{aligned} \tag{32}$$

The arm starts at rest from $\mathbf{q} = (0, \pi/2, \pi)$. Ten trials are enough to reduce the tracking error to zero, even in this more demanding case (Figs. 14–16). Cartesian forces computed at the tenth iteration behave as expected, with an initial push in the x-direction, followed by a much steeper increase in the y-direction. In the final part, with the lowering of the desired speed on the path, also the amount of force needed for trajectory tracking is reduced.

As a final example, the problem of singularity avoidance has been considered. The inclusion of a local strategy based only on kinematic variables is quite simple. The joint torques (18) can be modified as

$$\mathbf{u}_S = \mathbf{u} + \mathbf{K}_S(\mathbf{q})\nabla_{\mathbf{q}}H(\mathbf{q}), \quad \mathbf{K}_S(\mathbf{q}) \geq \mathbf{0}, \tag{33}$$

where $H(\mathbf{q})$ is any convenient manipulability measure to be maximized. A third trajectory is chosen, joining the two points (−0.25, 0.567) and (0.15, 0.567) with a straight line and bang-coast-bang acceleration profile. Starting from $\mathbf{q} = (-2\pi/3, \pi/2, \pi/2)$, the manipulator is forced to approach a folded singular configuration and a disastrous motion occurs without a prevention scheme (Fig. 17). Use of (33) with

$$H(\mathbf{q}) = \sqrt{\det \mathbf{J}(\mathbf{q})\mathbf{J}^{\mathrm{T}}(\mathbf{q})}, \quad \mathbf{K}_S(\mathbf{q}) = \frac{0.025}{H(\mathbf{q})}\mathbf{I} \quad (34)$$

recovers the nice behavior of Fig. 18, where the large motion of the first joint avoids the lining up of links.

6. Conclusions

A new controller for redundant manipulators has been proposed, that iteratively learns the input torques required to execute repetitive end-effector trajectories. The learning algorithm generates a feedforward term for the next trial, combining both the feedback control signal and the feedforward used at the current trial. Peculiar to the redundant case, the feedback stabilization part, which is model-independent, should definitely include a joint velocity damping action.

Digital processing of signals, intended to obtain low-pass filtering without introducing phase lags, prevents oscillatory behavior during the learning process and enforces convergence. The convergence speed and the final tracking accuracy are both improved, in face of the simplicity of the control law. The same learning structure allows the straightforward inclusion of a singularity avoidance scheme.

Finally, it was found that torques obtained with the present learning method are smaller in norm than those computed using the nominal dynamic model along joint trajectories derived through a kinematic resolution scheme which minimizes joint displacements. Such a comparison deserves further investigation, in order to understand intrinsic properties of torques acquired through the learning process.

Acknowledgements

This paper is based on work supported by the *Consiglio Nazionale delle Ricerche*, grant no. 90.00381.PF67 (*Progetto Finalizzato Robotica*). We would like to thank Prof. Fernando Nicolò for his promoting support and Dr. Giovanni Ulivi for stimulating discussions.

References

[1] T.J. Tarn, A.K. Bejczy, A. Isidori, Y. Chen, "Nonlinear feedback in robot arm control," *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, NV, pp. 736–751, Dec. 1984.

[2] J.-J.E. Slotine, W. Li, "On the adaptive control of robot manipulators," *Int. J. of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[3] S. Arimoto, S. Kawamura, F. Miyazaki, "Bettering operation of robots by learning," *J. of Robotic Systems*, vol. 1, no. 2, pp. 123–140, 1984.

[4] T. Omata, S. Hara, M. Nakano, "Nonlinear repetitive control with application to trajectory control of manipulators," *J. of Robotic Systems*, vol. 4, no. 5, pp. 631–652, 1987.

[5] S. Arimoto, F. Miyazaki, S. Kawamura, "Motion control of robotic manipulator based on motor program learning," *Proc. 2nd IFAC Symp. on Robot Control (SYROCO'88)*, Karlsruhe, FRG, Oct. 1988.

[6] A. De Luca, G. Paesano, G. Ulivi, "A frequency-domain approach to learning control of robots," *Proc. 4th IEEE Int. Symp. on Intelligent Control*, Albany, NY, pp. 66–70, Sep. 1989.

[7] A.A. Maciejewski, C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," *Int. J. of Robotics Research*, vol. 4, no. 3, pp. 109–117, 1985.

[8] T. Yoshikawa, "Manipulability of robotics mechanisms," *Int. J. of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[9] J.M. Hollerbach, K.C. Suh, "Redundancy resolution of manipulators through torque optimization," *IEEE J. of Robotics and Automation*, vol. RA-3, no. 4, pp. 308–316, 1987.

[10] A. Liégeois, "Automatic supervisory control of the configuration and behavior of multibody mechanisms," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. SMC-7, no. 12, pp. 868–871, 1977.

[11] A. De Luca, G. Oriolo, "The reduced gradient method for solving redundancy in robot arms," *Proc. 11th IFAC World Congress*, Tallinn, USSR, vol. 9, pp. 143–148, Aug. 1990.

[12] C. A. Klein, A.I. Chirco, "Dynamic simulation of a kinematically redundant manipulator system," *J. of Robotic Systems*, vol. 4, no. 1, pp. 5–23, 1987.

[13] Y. Nakamura, H. Hanafusa, "Optimal redundancy resolution control of robot manipulators," *Int. J. of Robotics Research*, vol. 6, no. 1, pp. 32–42, 1987.

[14] S. Arimoto, F. Miyazaki, "Stability and robustness of PID feedback control for robot manipulators of sensory capability," *Proc. 1st Int. Symp. on Robotics Research*, M. Brady and R. P. Paul Eds., pp. 783–799, MIT Press, Cambridge, 1984.

[15] C. Cosner, G. Anwar, M. Tomizuka, "Plug in repetitive control for industrial robotic manipulators," *Proc. 1990 IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, pp. 1970–1975, May 1990.

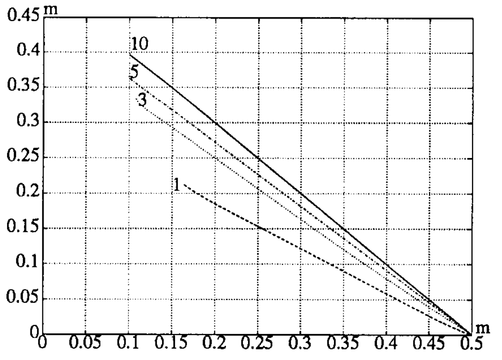

Fig. 5 -- Motion trajectories in the 1st 3rd 5th 10th trial (Ex #1)

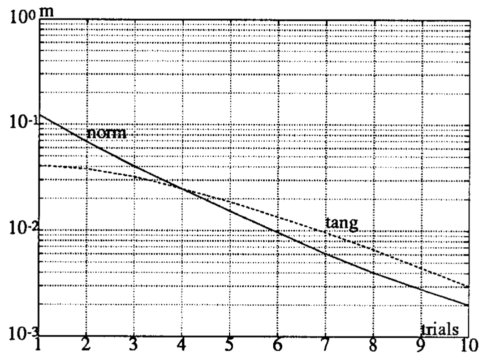

Fig. 6 -- RMS error (normal and tangential) vs. trial number (Ex #1)

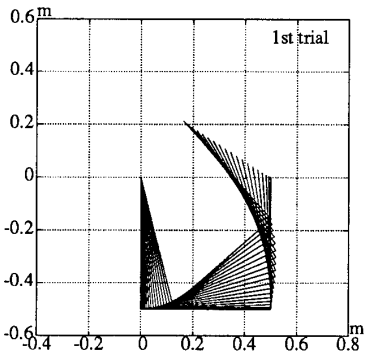

Fig. 7 -- Initial robot motion (Ex #1)

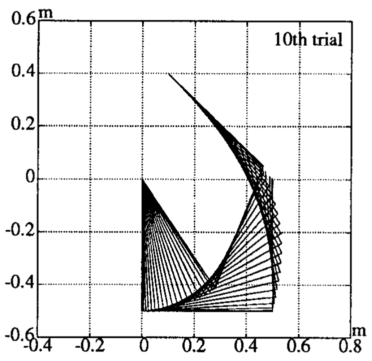

Fig. 8 -- Final robot motion (Ex #1)

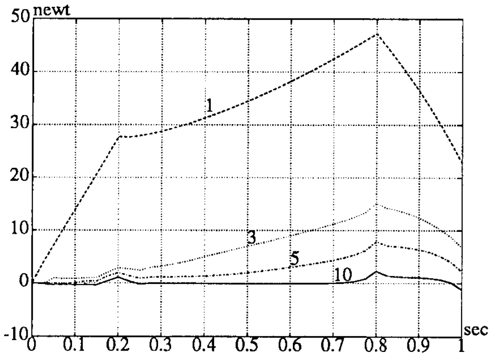

Fig. 9 -- Tangential force in the 1st 3rd 5th 10th trial (Ex #1)

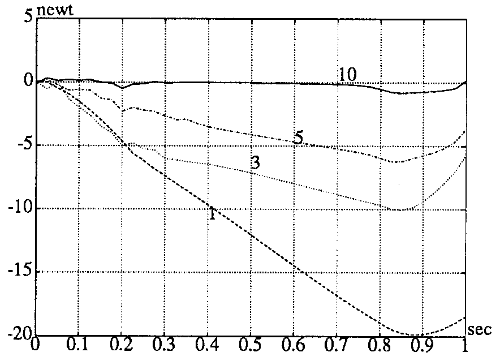

Fig. 10 -- Normal force in the 1st 3rd 5th 10th trial (Ex #1)

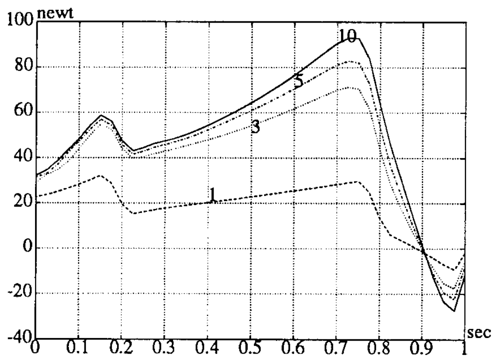

Fig. 11 -- Tangential learning in the 1st 3rd 5th 10th trial (Ex #1)

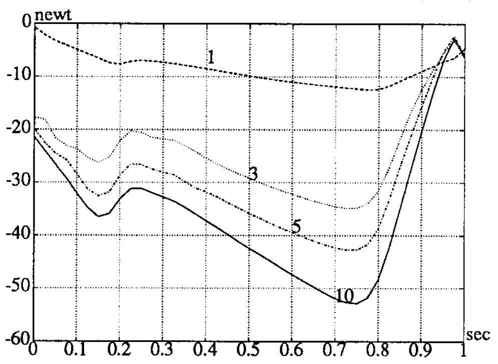

Fig. 12 -- Normal learning in the 1st 3rd 5th 10th trial (Ex #1)

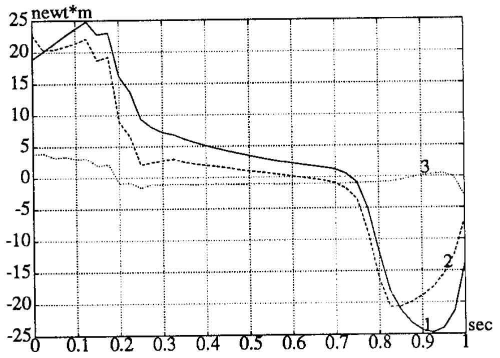

Fig. 13 -- Joint torques (Ex #1)

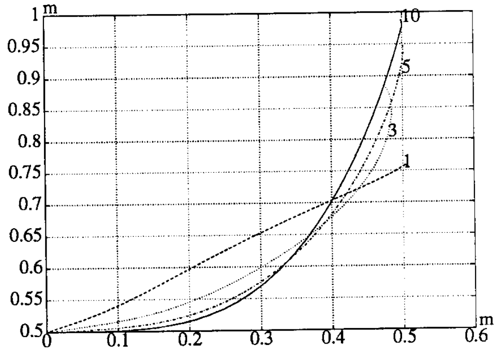

Fig. 14 -- Motion trajectories in the 1st 3rd 5th 10th trial (Ex #2)

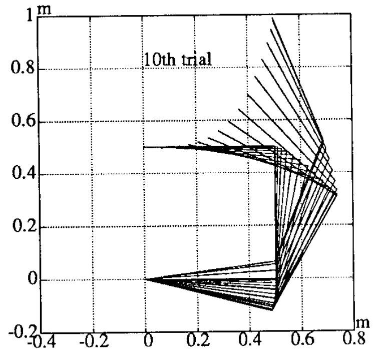

Fig. 15 -- Final robot motion (Ex #2)

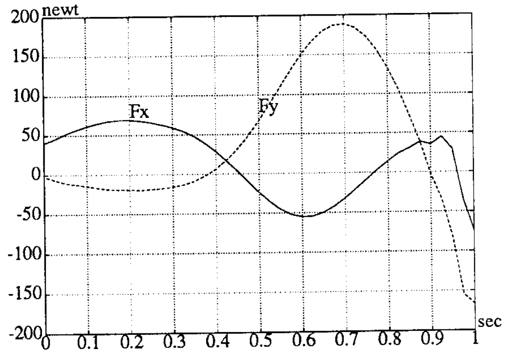

Fig. 16 -- Cartesian forces in the 10th trial (Ex #2)

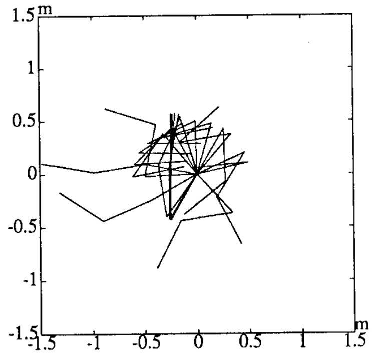

Fig. 17 -- Robot motion (Ex #3: without prevention)

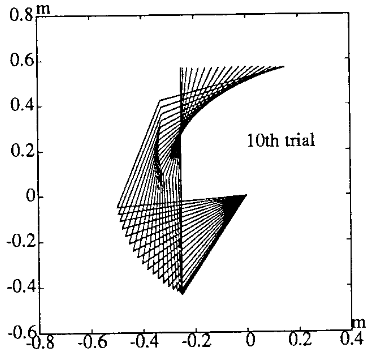

Fig. 18 -- Robot motion (Ex #3: with prevention)