[20] B. Kiss, J. Lévine, and B. Lantos, "On motion planning for robotic manipulation with permanent rolling contacts," *Int. J. Robot. Res.*, vol. 21, no. 5/6, pp. 443–461, 2002.
[21] A. Cole, J. E. Hauser, and S. Sastry, "Kinematics and control of multifingered hands with rolling contact," *IEEE Trans. Autom. Control*, vol. 34, no. 4, pp. 398–404, Apr. 1989.
[22] N. Sarkar, X. Yun, and V. Kumar, "Control of contact interactions with acatastatic nonholonomic constraints," *Int. J. Robot. Res.*, vol. 16, no. 3, pp. 357–374, 1997.
[23] K. Harada and M. Kaneko, "Rolling based manipulation under neighborhood equilibrium," in *Proc. IEEE Int. Conf. Robot. Autom*, Seoul, Korea, 2001, pp. 2492–2498.
[24] R. Ozawa, S. Arimoto, S. Nakamura, and J. Bae, "Control of an object with parallel surfaces by a pair of finger robots without object sensing," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 965–976, Oct. 2005.
[25] C. C. Phipps and M. A. Minor, "Quasi-static rolling control of the rolling disk biped robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, CA, 2008, pp. 1239–1245.
[26] J. S. Dai and D. R. Kerr, "Analysis of force distribution in grasps using augmentation," *Proc. Inst. Mech. Eng. C, Mech. Eng. Sci.*, vol. 210, no. C1, pp. 15–22, 1996.
[27] M. P. do Carmo, *Differential Geometry of Curves and Surfaces*. Englewood Cliffs, NJ: Prentice-Hall, 1976.
[28] E. Cartan, *Riemannian Geometry in an Orthogonal Frame*. Singapore: World Scientific, 2001.
[29] H. Cartan, *Differential Forms*. New York: Dover, 1996, pp. 139–163.
[30] M. R. Rosenberg, *Analytical Dynamics of Discrete Systems*. New York: Plenum, 1977.
[31] F. Bullo and A. D. Lewis, *Geometric Control of Mechanical Systems: Modeling, Analysis, and Design for Simple Mechanical Control Systems*. New York: Springer-Verlag, 2005.

Iterative-Learning Hybrid Force/Velocity Control for Contour Tracking

Antonio Visioli, Giacomo Ziliani, and Giovanni Legnani

***Abstract*—In this paper, we propose a new method, which is based on an iterative-learning-control (ILC) algorithm, for the contour tracking of an object of unknown shape performed by an industrial robot manipulator. In particular, we consider (both implicit and explicit) hybrid force/velocity control whose performance is improved by repeating the task. Here, a time-based reference signal is not present, and therefore, a new approach has been developed, which is different from the typical applications of ILC. Experimental results show the effectiveness of the technique.**

***Index Terms*—Contour tracking, hybrid force/velocity control, industrial robot manipulators, iterative-learning control (ILC).**

I. Introduction

Despite the remarkable achievements in the design of control systems for complex robotic tasks, in industrial setting, there is still the need for methodologies that can make robot manipulators adapt themselves autonomously to semiunstructured tasks in order to cut reprogramming costs and to shorten the lead to production time. A typical example of an advanced task where a high degree of autonomy is necessary is the automatic tracking of objects of unknown shape. In contrast to standard industrial operations, where robots reproduce programmed paths with a little amount of feedback from the process under control, here, the robot end-effector has to contour the piece with a reference tangential velocity by exerting, at the same time, a predefined normal force. This is required, for example, in polishing and gluing applications, where the characteristics of the parts, which are in contact with the robot, do not change during the task. In any case, the contour-tracking task is required as a basic capability in a number of applications, such as grinding, deburring [1], [2], shape recovery [3], and kinematic calibration. For this purpose, hybrid force/velocity control [4] appears to be suitable to be adopted, because it controls the end-effector force in a selected direction and the end-effector velocity in the other complementary direction. Actually, two kinds of hybrid force/velocity control can be implemented [5]: 1) *explicit hybrid force/velocity control*, where the robot end-effector is controlled by directly imposing the joint torques based on the measured force and position/velocity errors, and 2) *implicit hybrid force/velocity control*, where the end-effector is controlled indirectly by suitably modifying the reference trajectories of the position/velocity inner control loop based on the measured force and position/velocity errors. From another point of view, iterative-learning control (ILC) is a well-known effective approach for the robot motion control when a repetitive task is required (see, for example, [6] and references therein). Despite the success of the technique in the robot free-motion control, actually, no investigations for the application of this concept for contour tracking have been proposed with the exception of the work of Naniwa and Arimoto [7], where, however, a reference trajectory is given (i.e., the shape of the piece to be tracked is perfectly known), and only simulation results are shown. It is worth stressing that an approach similar to ILC has been employed in [8], where the controller self-improves itself iteratively by estimating the parameters of the model of the robot and of the contour by means of a Kalman filter.

In this paper, an ILC technique is applied to the normal-force control in the context of both explicit and implicit hybrid force/velocity control for the contour tracking of a piece of unknown shape. In this task, a time-based reference signal is not available (the duration of the different trials is not the same because of unavoidable errors in the tangential velocity), and therefore, a standard ILC approach, where the learning mechanism is related to the time, cannot be implemented. A different technique has therefore been devised to cope with this situation.

II. Hybrid Force/Velocity Control for Contour Tracking

We consider a contour-tracking task performed by a m-degree-of-freedom (DOF) robot manipulator. If $m > 3$, we assume that m_1 DOFs are actuated to achieve a predefined orientation of the end-effector with respect to the tracked object so that the kinematic configuration of the manipulator is completely defined by the other $m_2 := m - m_1$ joints coordinates $Q = [q_1, \ldots, q_{m_2}]^T$. The absolute reference frame xyz attached to the robot base is denoted as (0). The task frame (T) has its origin on the contact point between robot end-effector and tracked object, its n-axis is directed along the normal contact direction, the t-axis is directed along the tangential velocity direction, and the binormal axis b forms a right-handed frame. Let $F_{(0)} = [f_x, f_y, f_z]^T$ and $F_{(T)} = [f_t, f_n, f_b]^T$ be the vectors of the contact force in frame (0) and (T), respectively, while vector $V_{(T)} = [v_t, v_n, v_b]^T$ represents the Cartesian velocity in frame (T). The aim of the contour-tracking task is to control the normal force f_n and the tangential velocity v_t of the

Manuscript received June 30, 2009; revised October 28, 2009. First published February 22, 2010; current version published April 7, 2010. This paper was recommended for publication by Associate Editor L. Villani and Editor G. Oriolo upon evaluation of the reviewers' comments. This paper was presented in part at the IEEE International Conference on Robotics and Automation, Orlando, FL, May 2006.

A. Visioli is with the Dipartimento di Ingegneria dell'Informazione, University of Brescia, Brescia I-25123, Italy (e-mail: antonio.visioli@ing.unibs.it).

G. Ziliani and G. Legnani are with the Dipartimento di Ingegneria Meccanica e Industriale, University of Brescia, Brescia I-25123, Italy (e-mail: giacomo.ziliani@ing.unibs.it; giovanni.legnani@ing.unibs.it).

Digital Object Identifier 10.1109/TRO.2010.2041265

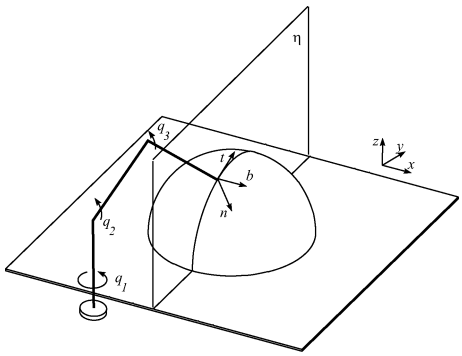

Fig. 1. Example of a 3-DOF manipulator in a 3-D contour-tracking task; the gripper must track an object with predefined tangential velocity and normal force while keeping on a predefined plane.

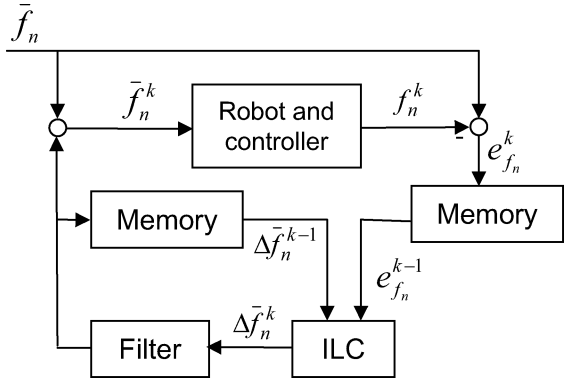

Fig. 2. ILC-based control scheme.

robot probe; the velocity component v_b should be null. In other words, the tool is moved along the contour by following the tangential direction of the piece (see Fig. 1). For this purpose, a hybrid force/velocity control can be effectively employed, both in its implicit and in its explicit form [5]. Because the shape of the workpiece is unknown, it is necessary to estimate the normal and tangential directions in order to determine at each instant set-point signals for the control system. Three-dimensional tasks require also the binormal. The estimation of the directions is based on the measure of the contact force between the tool and the piece. If friction is present in the contact, it is necessary to measure also the torque generated by the contact force with respect to the center of the tool [2]. Note that the standard hybrid force/velocity control law can be improved by employing an additional normal velocity control loop [9] and a friction-compensation strategy (see, for example, [10]). However, although a good performance is achieved with these control architectures, it appears that further significant improvements could be obtained. In particular, the following aspects are detrimental for the achievement of a high performance: 1) There is no rigorous methodology to tune the control design parameters (namely, the gains of the proportional–integral–derivative (PID) controllers), and therefore, although much care is adopted to select their values by a trial-and-error procedure, an optimal tuning is practically impossible to achieve; 2) the unavoidable backlash and elasticity in the joints, as well as the static friction, cause a higher error in the normal force when the joint torques changes its sign (for methods where these issues are addressed, see [3], [11], and [12]). For these reasons, the presence of automatic procedures that are capable of improving the contour-tracking performance, despite these issues, are highly desirable. In this paper, we propose the use of an ILC strategy for this purpose.

Remark 1. Note that in a three-dimensional (3-D) task (see Fig. 1), a position control loop must maintain the gripper on the plane η.

III. Iterative-Learning-Control Strategy

We assume that the same task (i.e., the contour of the same unknown workpiece) is required to be executed repetitively. In this context, the use of an ILC is appealing, as already demonstrated in a large number of works. However, the standard application of an ILC strategy requires a prespecified time-based reference signal in order to synchronize the same conditions in the different repetitions, i.e., that the same robot reference position is predefined at the same time instants in each repetition. However, in the considered contour-tracking task, there is no such time-based reference signal. Thus, an alternative approach has been developed.

A. General Idea

The general idea is to adopt an ILC strategy to modify the normal-force set-point $\bar{f}_n$ according to the normal-force error that has been measured at the previous repetition of the task. This concept is depicted in Fig. 2, where the learning control law, i.e., the modification of the normal-force set-point at each time instant of the hth trial, is expressed as

$$\bar{f}_n^h = \bar{f}_n + \Delta\bar{f}_n^h \tag{1}$$

where

$$\Delta\bar{f}_n^h = \Delta\bar{f}_n^{h-1} + g \cdot e_{f_n}^{h-1} \tag{2}$$

e_{f_n} is the measured normal-force error, and g is a user-chosen parameter (its selection will be discussed in Section IV-B). Note that $\Delta\bar{f}_n^0 = 0$, and $e_{f_n}^0 = 0$, i.e., the learning control law is not applied at the first repetition.

B. New Indexing

As already mentioned, the normal-force error is not related directly to a given time instant at each repetition. Thus, there is the need to relate the measured force error and the normal-force set-point modification with other variables in order to find the value of $e_{f_n}^{h-1}$ and $\Delta\bar{f}_n^{h-1}$ at each time instant. In this context, it is worth considering that the contour-tracking performance is obviously influenced by the kinematic and kinetostatic characteristics of the robot, which can be represented effectively by the force manipulability ellipsoids. The same reasoning also applies to the inertial and stiffness ellipsoids that have similar shapes. Another relevant parameter is the direction of the force exchanged with the environment. The size and the orientation of the ellipsoids depend on the Jacobian matrix J and, therefore, on the joint coordinates Q, while the force direction can be represented by two angles. Thus, the kinetostatic and dynamic behavior of the manipulator can be described in general by $m_2 + 2$ parameters. However, the absolute orientation of the ellipsoid is not important, but only its relative one is important with respect to the direction of the force. Thus, the number of variables can be reduced to m_2 by grouping the configurations with equivalent properties. With this aim, it is possible to define the state-variable vector

$$\tilde{Q} = [\tilde{q}_1, \ldots, \tilde{q}_{m\,2}]^T = J(Q)^T F_u, \qquad F_u = \frac{F_{(0)}}{|F_{(0)}|} \tag{3}$$

where F_u is the unit vector defining the direction of the force. By remembering that the joint torques/forces can be evaluated as $J^T(Q)F_{(0)}$, it is possible to show that for revolute joints, $\tilde{q}_i$ represents the arm distance of the gripper force with respect to the ith joint, while for prismatic joints, it is the cosine of the angle between the force F_u

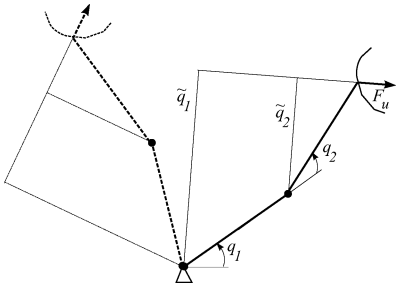

Fig. 3. Meaning of the state vector $\tilde{Q}$ for a 2-DOF manipulator; the figure shows two configurations with the same value of $\tilde{Q}$, which depends on the relative position of the robot with respect to the contact between the manipulator and the workpiece.

and the axis a_i of the prismatic joint (namely, $\tilde{q}_i = a_i \cdot F_u$). As shown in Fig. 3, the state variables $\tilde{Q}$ assume the same value in all the situations in which the kinetostatic configuration of the manipulator with respect to the piece are identical (same relative position and relative force direction). Note that the ith element of the state-variable vector can assume values in a range between $-\tilde{q}_{i,\max}$ and $\tilde{q}_{i,\max}$, where $\tilde{q}_{i,\max}$ can be evaluated easily by knowing the manipulator geometry. For the sake of clarity and without loss of generality, hereafter, we assume $m_2 = 3$, which is the most typical situation in the industrial context.

In principle, when a trial of the contour-tracking task is performed, at each time instant, the values of both $e^h_{f_n}$ and $\Delta\bar{f}^h_n$ are associated with the value of (the measured) $\tilde{Q}$. From a practical point of view, the whole range of $\tilde{q}_i$, namely, $[-\tilde{q}_{i,\max}, \tilde{q}_{i,\max}]$, $i = 1, 2, 3$, can be divided in N_i equally spaced intervals so that two 3-D arrays of cells are created to store the values of the measured force errors and of the normal-force set-point modifications. The values associated to the cells of the two 3-D arrays are initialized to zero. It has to be taken into account that, because of the nature of the task and the nonideal repeatability of the manipulator, the trajectory followed by the manipulator is not exactly the same at each iteration, and therefore, it might happen that in a certain trial, the trajectory of the robot is related to elements $(\tilde{q}_1, \tilde{q}_2, \tilde{q}_3)$ of the 3-D array that has not been indexed in previous trials. To improve robustness, the ILC algorithm, which modifies the normal-force set-point, must be able to retrieve the required information even if at each iteration, the manipulator does not follow exactly the same path. Further, the set-point should vary smoothly with respect to the time in order to avoid vibrations as much as possible. For these reasons, the data that are necessary to modify the set-point value are disseminated (with a suitable weight) in a neighborhood of the path (see Fig. 4).

Thus, at each time instant, a two-element $\times$ two-element $\times$ two-element portion of the 3-D array is considered, i.e., the element that includes the actual manipulator configuration and the seven ones (two for each dimension) that are closest to it. In this context, the distances between the centers of the elements of the 3-D array and the actual configuration of the manipulator are employed as a suitable measure. Then, a weight $w_{i,j,k}$ is calculated for each considered element (i, j, k), depending on the distance $d_{i,j,k}$ of its center from the actual position $(\tilde{q}_1, \tilde{q}_2, \tilde{q}_3)$ of the robot (see Fig. 5). In particular, the following criteria have been adopted for this purpose.

1) The sum of the weights has to be equal to one in order to avoid the introduction of any gain in the data.
2) The weight has to decrease when the distance of the element from the position of the manipulator increases.
3) The weight has to be zero if the distance is greater than the dimension of an element (so that cells that are outside the correct $2 \times 2 \times 2$ portion are not affected).

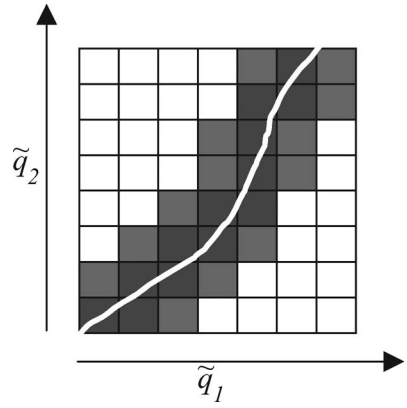

Fig. 4. (Dark and pale gray) Cells that are involved in the ILC during the execution of a trajectory (a 2-D array is shown for the sake of clarity).

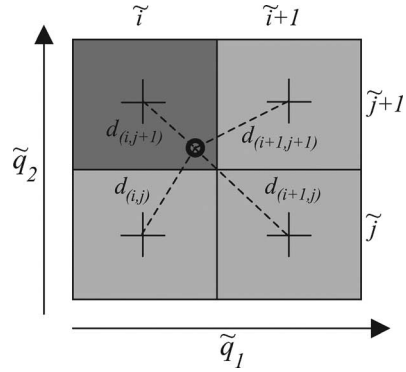

Fig. 5. Cells affected by the ILC strategy at each time instant; **O**: manipulator position; $+$: cell center. A 2-D array is shown for the sake of clarity.

The following function, which satisfies the previous requirements, has therefore been selected:

$$w_{i,j,k} = \frac{p^+(1 - d_{i,j,k})}{\sum_{i,j,k} p^+(1 - d_{i,j,k})} \tag{4}$$

where $p^+(\delta)$ denotes the positive part function, i.e.,

$$p^+(\delta) := \frac{\delta + |\delta|}{2}. \tag{5}$$

The values of the measured normal-force error and the force set-point modification are stored in each cell of the $2 \times 2 \times 2$ portion of the 3-D array, with the determined associated weight.

Actually, another aspect has to be considered. In fact, due to the discretization of $\tilde{q}_1$, $\tilde{q}_2$, and $\tilde{q}_3$, and due to the small sampling time, there might be different values of e_{f_n} and $\Delta\bar{f}_n$ to be considered (at different time instants) within the same cell (i.e., for the same element of the 3-D array). However, it is obvious that only one value for each variable has to be associated with a cell. For this reason, it is necessary to store the sum of the values of each variable over the time of permanence of the manipulator inside a cell (denoted as $\Delta\bar{F}_n$ and E_{f_n}, respectively) and the sum of the corresponding weight as well (denoted as $W_{i,j,k}$). Then, the normal-force error and the set-point modification to be actually adopted in the learning control law (2) while passing through a cell are determined by dividing the sums of the variables with the sum of the weights that have been stored in the cell itself.

It is worth stressing at this point that the tracking performance also depends on, for instance, the dynamics of interaction and the local

geometry of the workpiece, such as the curvature. An extended formulation of the procedure could consider other parameters, like curvature, velocity, etc, but the number of the state variable would increase dramatically, thus making the overall method difficult to be employed in industrial settings.

C. Iterative-Learning-Control Algorithm

Based on the considerations made in the previous sections, the proposed ILC algorithm is formulated as follows.

At the hth iteration, for each time instant the following hold.

1) The position of the manipulator in the new coordinates is determined as follows:

$$\tilde{q}_{lN} = \frac{\tilde{q}_l + q_{l,\max}}{2q_{l,\max}} N_l, \qquad l = 1,2,3 \tag{6}$$

where N_l, for $l = 1,2,3$, is the number of discretization intervals.

2) The elements of the 3-D array related to the manipulator position are determined as follows:

$$\tilde{i} = \mathrm{trunc}(\tilde{q}_{1N}), \qquad \tilde{j} = \mathrm{trunc}(\tilde{q}_{2N}), \qquad \tilde{k} = \mathrm{trunc}(\tilde{q}_{3N}) \tag{7}$$

where the truncation function is defined as

$$\mathrm{trunc}(\delta) := \max\{n \in \mathbb{N} : n \le \delta\}, \qquad \delta \in \mathbb{R}. \tag{8}$$

The elements of the 3-D array that are involved in the learning procedure are therefore given by

$$(\tilde{i},\tilde{j},\tilde{k}), \quad (\tilde{i}+1,\tilde{j},\tilde{k}), \quad (\tilde{i},\tilde{j}+1,\tilde{k}), \quad (\tilde{i}+1,\tilde{j}+1,\tilde{k})$$
$$(\tilde{i},\tilde{j},\tilde{k}+1), \quad (\tilde{i}+1,\tilde{j},\tilde{k}+1), \quad (\tilde{i},\tilde{j}+1,\tilde{k}+1)$$
$$(\tilde{i}+1,\tilde{j}+1,\tilde{k}+1).$$

3) The distances of the manipulator position from the centers of the involved cells are determined as follows:

$$d_{i,j,k} = \sqrt{(i-\tilde{q}_1)^2 + (j-\tilde{q}_2)^2 + (k-\tilde{q}_3)^2}$$
$$i = \tilde{i}, \tilde{i}+1, \qquad j = \tilde{j}, \tilde{j}+1, \qquad k = \tilde{k}, \tilde{k}+1. \tag{9}$$

4) The weights are calculated as follows:

$$w^h_{i,j,k} = \frac{p^+(1-d_{i,j,k})}{\sum_{i,j,k} p^+(1-d_{i,j,k})}$$
$$i = \tilde{i}, \tilde{i}+1, \qquad j = \tilde{j}, \tilde{j}+1, \qquad k = \tilde{k}, \tilde{k}+1. \tag{10}$$

5) For each of the considered eight cells, the previously stored values of the sums of the normal-force error and of the set-point correction are retrieved and weighted ($i = \tilde{i}, \tilde{i}+1, j = \tilde{j}, \tilde{j}+1, k = \tilde{k}, \tilde{k}+1$)

$$\Delta\bar{f}^{h-1}_{n(i,j,k)} = \frac{\Delta\bar{F}^{h-1}_{n(i,j,k)}}{W^{h-1}_{i,j,k}}, \qquad e^{h-1}_{f_n(i,j,k)} = \frac{E^{h-1}_{f_n(i,j,k)}}{W^{h-1}_{i,j,k}}. \tag{11}$$

6) The normal-force set-point change to be actually applied is calculated ($i = \tilde{i}, \tilde{i}+1, j = \tilde{j}, \tilde{j}+1, k = \tilde{k}, \tilde{k}+1$) as follows:

$$\Delta\bar{f}^h_n = \sum_{i,j,k} (w^h_{i,j,k} \cdot (\Delta\bar{f}^{h-1}_{n(i,j,k)} + g \cdot e^{h-1}_{f_n(i,j,k)})). \tag{12}$$

7) The exerted normal force is measured and the normal-force error is calculated as follows:

$$e^h_{f_n} = \bar{f}_n - f_n. \tag{13}$$

8) The 3-D array elements, after being initialized at the beginning of the repetition as $\Delta\bar{F}^h_{n(i,j,k)} = \Delta\bar{F}^{h-1}_{n(i,j,k)}$, $E^h_{f_n(i,j,k)} = E^{h-1}_{f_n(i,j,k)}$, and $W^h_{i,j,k} = W^{h-1}_{i,j,k}$ are updated for the next iteration as follows:

$$\Delta\bar{F}^h_{n(i,j,k)} = \Delta\bar{F}^h_{n(i,j,k)} + w^h_{i,j,k} \cdot \Delta\bar{f}^h_{n(i,j,k)}$$
$$E^h_{f_n(i,j,k)} = E^h_{f_n(i,j,k)} + w^h_{i,j,k} \cdot e^h_{f_n}$$
$$W^h_{i,j,k} = W^h_{i,j,k} + w^h_{i,j,k}. \tag{14}$$

D. Filtering

With the devised algorithm, abrupt modifications of the normal-force set-point might occur, in general, and therefore, the object might not be correctly tracked if the bandwidth of the control system is not sufficiently high. Indeed, a delayed corrective action might introduce oscillations and eventually cause a loss of contact. A simple strategy to solve this problem is to suitably low-pass filter the output of the learning controller (see Fig. 2) [13], [14]. In this paper, we choose to use a second-order Butterworth low-pass filter with a cutoff frequency ω_0. The selection of the value of this design parameter is discussed in Section IV-B.

IV. Experimental Results

A. Experimental Setup

The robot employed in the experiments is a 2-DOF planar industrial selective compliance assembly robot arm (SCARA) robot. Thus, a 2-DOF (planar) task is actually addressed. A description of its dynamic model can be found in [15]. Both links have the same length $l_1 = l_2 = 0.33$ m. The two joints are actuated by dc motors (which are driven by conventional pulsewidth-modulated (PWM) amplifiers) through harmonic drive speed reducers whose reduction ratio is 1/100. Motor rotations are measured by means of two incremental encoders with 2000 pulses/revolution resolution. Velocity is estimated through numerical differentiation whose output is then processed by a low-pass second-order Butterworth filter with a 100-Hz cutoff frequency and 1.0 damping ratio (note that, alternatively, Kalman filter techniques can be used [16], [17]). An ATI 65/5 force/torque sensor capable of measuring forces in the range of ± 65 N and with a resolution of 0.05 N is mounted at the manipulator wrist. The corresponding signals are processed at 7.8 kHz frequency by an industry standard architecture (ISA) digital signal processor (DSP)-based board and collected by the robot controller at 1 kHz. The contact is achieved by means of a proper probe endowed with a ball bearing with an 8-mm diameter whose aim is to reduce tangential friction forces that may arise from the contact with the piece.

B. Tuning of the Design Parameters

Based on the robot geometry, we have $\tilde{q}_{1,\max} = l_1 + l_2$ and $\tilde{q}_{2,\max} = l_2$. Then, the values $N_1 = 700$ and $N_2 = 350$ have been chosen by taking into account the tradeoff between resolution and computational efficiency. The proposed ILC strategy therefore involves the tuning of two design parameters, namely, the ILC gain g and the filter cutoff frequency ω_0. This can be done quite easily with a trial-and-error procedure by taking into account the physical meaning of the two parameters. Actually, a higher value of g causes a faster convergence, but a too high value may render the system unstable. In any case, the tuning of the parameters is not a critical issue. The values of $g = 1$ and $\omega_0 = 20$ Hz have been conveniently selected for the implicit controller, while for the explicit controller, the value of $g = 0.5$ has been selected,

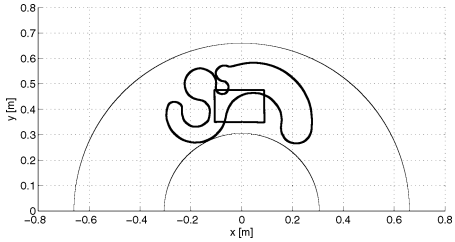

Fig. 6. Position of the two workpieces in the robot workspace.

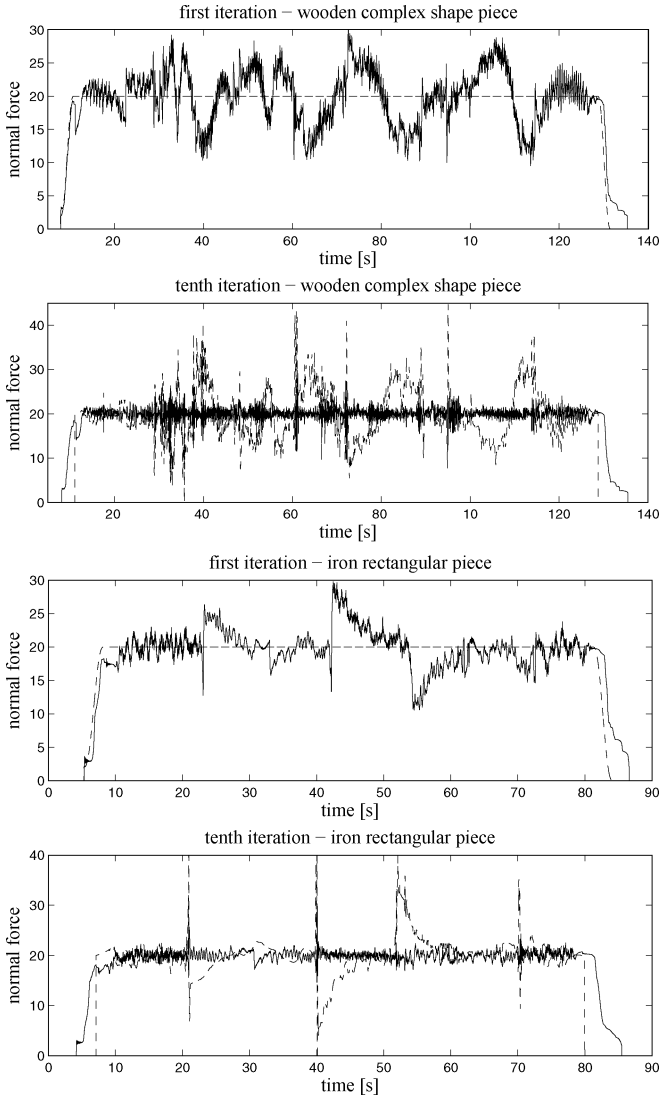

Fig. 7. Resulting (solid line) normal forces and (dashed line) set-point at first and tenth iteration. (Top) Complex shape piece. (Bottom) Rectangular piece.

and the filter was not necessary (this is due to the high bandwidth of the explicit controller with respect to the implicit one).

C. Results

The ILC strategy has been applied for the contour tracking of two planar objects of different (unknown) shape, which are placed in different positions of the manipulator workspace, as shown in Fig. 6. In particular, a wooden complex shape and an iron rectangular piece with

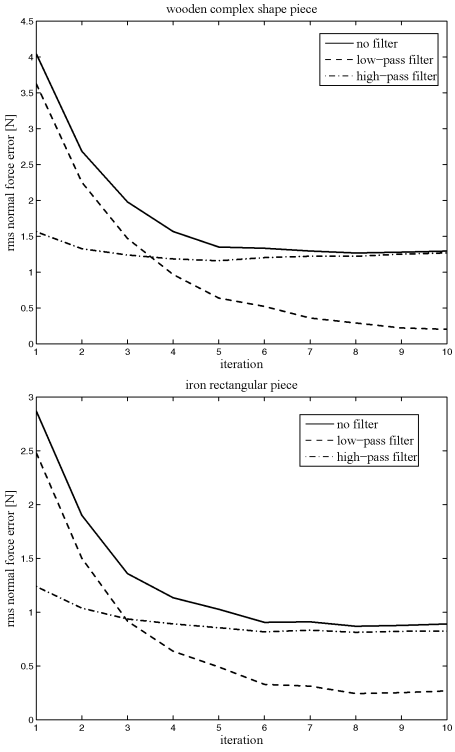

Fig. 8. Resulting rms normal-force errors along the different repetitions. (Top) Complex shape piece. (Bottom) Rectangular piece.

sharp edges have been considered. In both cases, the normal-force set-point is 20 N, while the reference tangential velocity is 20 mm/s for the wooden piece and 30 mm/s for the iron piece. Both the implicit and explicit hybrid force/velocity controller has been tested. Results obtained are very similar, and only those related to the explicit control are presented for the sake of brevity. Fig. 7 shows the results of the tracking of the two objects. Two plots are shown for each object. The upper one presents the normal-force set-point and the actual force at the first repetition, while the lower one represents the actual force at the tenth repetition and (dashed line) the corresponding set-point, which has been modified by the ILC strategy. Note that the ILC strategy is applied after the guarded move phase, when the tangential velocity is different from zero. The rms-error improvements along the different repetitions are plotted in Fig. 8. In particular, to evaluate the role of the ILC technique better, the rms errors have been reported after filtering the force signals with a low-pass and with a high-pass filter (the cutoff frequency in both cases is 0.25 Hz). Note that the resulting tangential velocity has not been reported for the sake of brevity. In any case, the ILC strategy does not significantly affect its performance. From the presented results, it can be easily deduced that the proposed new ILC strategy is very effective in significantly reducing the normal-force error after just a few trials, especially that with a relatively slow dynamics (compare the different lines in Fig. 8).

V. Conclusion

In this paper, we have shown that a newly devised ILC strategy can be effectively adopted for the contour tracking of objects of unknown shape. Both implicit and explicit hybrid force/velocity control laws have been considered. The role of the design parameters have been discussed, and experimental results have been presented in order to clearly evaluate the performance that can be achieved. It appears that just a

few iterations are sufficient to significantly decrease the normal-force tracking error. Further, the proposed algorithm can cope with situations in which it is necessary to contour objects that are not perfectly identical to each other or that are not positioned exactly in the same location.

REFERENCES

[1] G. Ferretti, G. Magnani, and P. Rocco, "Triangular force/position control with application to robotic deburring," *Mach. Intell. Robot. Control*, vol. 2, pp. 83–91, 2000.

[2] G. Ziliani, G. Legnani, and A. Visioli, "A mechatronic approach for robotic deburring," *Mechatronics*, vol. 17, pp. 431–441, 2007.

[3] S. Ahmad and C. N. Lee, "Shape recovery from robot contour-tracking with force feedback," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1990, pp. 447–452.

[4] M. H. Raibert and J. J. Craig, "Hybrid position/force control of manipulators," *ASME J. Dyn. Syst., Meas., Control*, vol. 102, pp. 126–133, 1981.

[5] J. Roy and L. L. Whitcomb, "Adaptive force control of position/velocity controlled robots: Theory and experiments," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 121–137, Apr. 2002.

[6] K. L. Moore, "Iterative learning control An expository overview," *Appl. Comput. Controls, Signal Process., Circuits*, vol. 1, pp. 151–214, 1999.

[7] T. Naniwa and S. Arimoto, "Learning control for robot tasks under geometric endpoint constraints," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 432–441, Jun. 1995.

[8] F. Lange and G. Hirzinger, "Iterative self-improvement of force feedback control in contour tracking," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992, pp. 1399–1404.

[9] F. Jatta, G. Legnani, A. Visioli, and G. Ziliani, "On the use of velocity feedback in hybrid force/velocity control of industrial manipulators," *Control Eng. Pract.*, vol. 14, no. 9, pp. 1045–1055, 2006.

[10] F. Jatta, G. Legnani, and A. Visioli, "Friction compensation in hybrid force/velocity control of industrial manipulators," *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 604–613, Apr. 2006.

[11] G. Ferretti, G. Magnani, and P. Rocco, "Toward the implementation of hybrid position/force control in industrial robots," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 838–845, Dec. 1997.

[12] F. Aghili and M. Namvar, "A self-tuning torque feedback for control of manipulators," presented at the IFAC Symp. Robot Control, Bologna, Italy, 2006.

[13] R. W. Longman, "Iterative learning control and repetitive control for engineering practice," *Int. J. Control*, vol. 73, no. 10, pp. 930–954, 2000.

[14] M. Norrlof, "On analysis and implementation of iterative learning control," Ph.D. dissertation, Linköping Univ., Linköping, Sweden, 1998.

[15] A. Visioli and G. Legnani, "On the trajectory tracking control of industrial SCARA robot manipulators," *IEEE Trans. Ind. Electon.*, vol. 49, no. 1, pp. 224–232, Feb. 2002.

[16] P. R. Belanger, P. Dobrolovny, A. Helmy, and X. Zhang, "Estimation of angular velocity and acceleration from shaft-encoder measurements," *Int. J. Robot. Res.*, vol. 17, no. 11, pp. 1225–1233, 1998.

[17] S. J. Ovaska and S. Valiviita, "Angular acceleration measurement: A review," *IEEE Trans. Instrum. Meas.*, vol. 47, no. 5, pp. 1211–1217, Oct. 1998.

EMG-Based Control of a Robot Arm Using Low-Dimensional Embeddings

Panagiotis K. Artemiadis and Kostas J. Kyriakopoulos

***Abstract*—As robots come closer to humans, an efficient human–robot-control interface is an utmost necessity. In this paper, electromyographic (EMG) signals from muscles of the human upper limb are used as the control interface between the user and a robot arm. A mathematical model is trained to decode upper limb motion from EMG recordings, using a dimensionality-reduction technique that represents muscle synergies and motion primitives. It is shown that a 2-D embedding of muscle activations can be decoded to a continuous profile of arm motion representation in the 3-D Cartesian space, embedded in a 2-D space. The system is used for the continuous control of a robot arm, using only EMG signals from the upper limb. The accuracy of the method is assessed through real-time experiments, including random arm motions.**

***Index Terms*—Dimensionality reduction, electromyographic (EMG)-based control, EMG signals, neurorobotics.**

I. INTRODUCTION

Despite the fact that robots came about approximately 50 years ago, the way humans control them is still an important issue. In particular, since robots are being used more frequently in everyday life tasks (e.g., service robots, robots for clinical applications), the human–robot interface plays a role of utmost significance. In this paper, a new mean of control interface is proposed, according to which, the user performs natural motions with his/her upper limb, while superficially recorded electromyographic (EMG) activity of the muscles of the upper limb is transformed to kinematic variables that are used to control a robot arm.

EMG signals have often been used as control interfaces for robotic devices. However, in most cases, only discrete control has been realized, focusing only, for example, at the directional control of robotic wrists [1]. This can cause many problems regarding smoothness of motion, while from a teleoperation point of view, a small number of finite commands or postures can critically limit the areas of application. Since most previous studies focused on EMG signal discrimination, a variety of algorithms have been proposed for this scope. A statistical log-linearized-Gaussian-mixture-neural network has been proposed in [1] to discriminate EMG patterns for wrist motions. A small number of researchers have tried to build continuous models to decode arm motion from EMG signals. The Hill-based muscle model [2], [3] is more frequently used in [4]. However, only a few degrees of freedom (DOFs) were analyzed (i.e., 1 or 2), since the nonlinearity of the model and the large numbers of unknown parameters for each muscle make the analysis rather difficult. A neural-network model was used to extract continuous arm motion in the past using EMG signals [5]; however, the movements analyzed were restricted to single-joint, isometric motions.

Manuscript received October 28, 2008; revised December 11, 2009. First published January 29, 2010; current version published April 7, 2010. This paper was recommended for publication by Associate Editor F. Thomas and K. Yamane and Editor W. K. Chung upon evaluation of the reviewers' comments. This work was supported by the European Commission under the contract NEUROBOTICS (FP6-IST-001917) project.

P. K. Artemiadis was with the Control Systems Lab, School of Mechanical Engineering, National Technical University of Athens, Athens, Greece. He is now with the Department of Mechanical Engineering, Massuchusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: partem@mit.edu).

K. J. Kyriakopoulos is with the Control Systems Laboratory, School of Mechanical Engineering, National Technical University of Athens, Athens 15780, Greece (e-mail: kkyria@mail.ntua.gr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2009.2039378