Experimental Evaluation of Nonlinear Adaptive Controllers

E. Burdet, A. Codourey, and L. Rey

Attractive methods for learning the dynamics and improving the control of robot manipulators during movements have been proposed for more than 10 years, but they still await applications. This article investigates practical issues for the implementation of these methods. Two nonlinear adaptive controllers, selected for their simplicity and efficiency, are tested on 2-DOF and 3-DOF manipulators. The experimental results show that the Adaptive FeedForward Controller (AFFC) is well suited for learning the parameters of the dynamic equation, even in the presence of friction and noise. The control performance along the learning trajectory and other test trajectories are also better than when measured parameters are used. However, when the task consists of driving a repeated trajectory, the adaptive look-up table MEMORY is simpler to implement. It also provides a robust and stable control, and results in even better performance.

Introduction

Motivation

Certain tasks, e.g., high-speed laser cutting or high-precision milling, require an accurate control during the whole movement. When these tasks are performed at high speed, it is necessary to compensate for the manipulator's dynamics in order to ensure a high precision [1, 2]. Many schemes have been proposed for learning the (nonlinear) dynamics of manipulators during movement in a non-invasive way, and compensating for it [3-14]. The mathematical properties of these adaptive controllers have also been studied extensively [15-17]. In contrast, only a few implementations have been realized so far, and to our knowledge there exists no industrial application yet. Considering the gap between the theory and the practice of nonlinear adaptive control, this article addresses practical issues for its implementation. It is tested whether nonlinear adaptive controllers fulfill the following typical industrial requirements:

- the resulting performance must be at least as good as with conventional methods.

Two tasks are considered: (i) improving control of arbitrary trajectories and (ii) improving control along a repeated trajectory. It is obvious that the first task is useful, but the second is also of great practical interest: most manipulators used in industry repeat a single movement.

Selecting Simple and Efficient Controllers

There are principally two kinds of adaptive controllers. Parametric controllers [15] use a model of the inverse dynamics and identify the model's parameters during the movements. In nonparametric controllers [18], the dynamics necessary to perform a movement are stored in a table-look-up or a neural network (note that controllers are not treated in this study). In this subsection, existing controllers of both types are reviewed, and rationales for choosing the algorithms most suitable for implementations are proposed. A more complete evaluation is contained in [19].

Parametric schemes. The most obvious model of the dynamics of robot manipulators is a dynamic equation, generally a rigid-body model equation. Good control performances require precise values of the manipulator parameters. These parameters are generally calculated from the CAD model; if higher precision is required, one normally disassembles the manipulator and measures the lengths, the masses and moments of inertia of the different mechanical parts, and the friction parameters. Alternatively, these parameters can be identified in a non-invasive way: the robot is controlled using linear joint feedback controllers, and the error during motion is used to determine the correct value of these parameters. The model error can be minimized with a (recursive) least-square algorithm [11, 6], or alternatively the tracking error can be minimized using a gradient algorithm [7, 13, 12]. The second method, which has been proved to provide stable control even during adaptation [15], requires fewer computations than the first one.

Whitcomb et al. [14] experimentally compared different controllers [7, 13, 12] based on the gradient descent of the tracking error. They found that nonlinear controllers clearly outperform linear controllers, and adaptive nonlinear controllers outperform their nonadaptive counterparts. They also found that the scheme with the simplest implementation, the Adaptive FeedForward Controller (AFFC) [12], provided slightly better control performance than the other controllers tested. (We renamed the Desired Compensation Adaptive Law from [12] to emphasize the fact that it is a natural adaptive extension of a feedforward controller.) We verified these results in preliminary experiments.

Nonparametric schemes. The parametric methods require a dynamic equation and must be applied on a trajectory exciting...
the entire dynamics [20, 7]. The corresponding effort stands in contrast with the simplicity of the task performed by most industrial manipulators: repeating a single movement. For this task, nonparametric controllers may be simpler to implement and more efficient than parametric controllers. In existing nonparametric controllers, the dynamics are gradually stored during the movement in an artificial neural network [3, 9, 10] or in a look-up table [4-7, 21, 8]. A particularly simple adaptive look-up table called MEMory has also been introduced [19].

Simulation results [22] suggest that the look-up table approach is simpler to implement and more robust to noise than the neural network approach.

Specific Points Addressed in this Article

According to the above analysis, the AFFC is the most promising parametric controller and MEMory is the best nonparametric controller for improving the control along a repeated trajectory. We decided to further study these two controllers. Experiments with the AFFC were directed at complementing the analysis of Whitcomb et al. [14], by:

- Testing the generalization and the control performance after learning;
- Examining whether the AFFC is able to identify the friction (friction varies with age and temperature, and it is difficult to identify);
- Investigating the robustness to noise experimentally (most real plants experience noise, yet it is difficult to simulate noise realistically).

MEMory was also analyzed beyond [19]. A mathematical analysis of the convergence of the learning was done, and experiments were performed for examining:

- The ability of MEMory to compensate for the dynamics of plants that are difficult to model by a differential equation, e.g., plants experiencing friction and noise;
- How its parameters must be chosen;
- The relative control performances of MEMory and AFFC.

This article reports these investigations. The manipulators on which the experiments were performed are described in the next section. The tests performed on the AFFC and the analysis of MEMory are presented in the third and fourth sections, respectively.

Experimental Setup

The Planar Parallel Manipulator

The first manipulator used for our experiments is a small planar manipulator (Fig. 1). It is composed of six limbs (of length \( l = 12 \) cm, mass \( m \), and moment of inertia \( J \)), joined by two metallic plates (of masses \( m_{A} \) and \( m_{C} \)). Its joints have a large friction. This manipulator is driven by two direct-drive DC-motors and can move with high speed, inducing highly nonlinear effects and dynamic coupling. It is controlled by a VME-68040 board enabling a sampling frequency of 333.3 Hz.

If the velocities are obtained by taking the derivative of the encoder position, then the noise level is high (Fig. 6). In most of our experiments, the joint velocities were instead determined with velocity sensors using potentiometers and an analog differentiator. In order to test experimentally the robustness to noise of the controller, some trials used the velocities resulting from the derivative of the encoder position (without filtering).

The rigid body equation, derived using the Lagrange method [19], was written as a linear function

$$
\tau = \Psi(q, \dot{q}, \ddot{q})p
$$

of the parameters vector \( p \),

\[
p = \begin{pmatrix}
\ell^2 m_c, \ell^2 m, 3 J + J_{\text{rot}}, \ell^2 m_s, b_1, b_2, b_3
\end{pmatrix}^T,
\]

where

\[
\Psi^* = \begin{bmatrix}
\frac{s_1 q_1^2 - q_2 q_2^2 + \ddot{q}_1}{3} & \frac{s_2 q_2^2 - q_3 q_3^2 + \ddot{q}_2}{3} & \frac{\ddot{q}_1}{4}
\end{bmatrix},
\]

\[
\begin{bmatrix}
\ddot{q}_1 \\
\ddot{q}_2 \\
\text{sign}(\ddot{q}_3)
\end{bmatrix}
\]

and

\[
\begin{bmatrix}
\dot{q}_1 \\
\dot{q}_2 \\
\dot{q}_3
\end{bmatrix}
\]
where \( s_{12} = \sin(q_1 + q_2) \) and \( c_{12} = \cos(q_1 + q_2) \), with \( q_1 \) and \( q_2 \) defined in Fig. 1.

By measuring the motor torques for different velocities (Fig. 2), it was verified that the shape of real friction was well modeled by \( b_i q_i + b_i \text{sign}(q_i) \), \( (i = 1, 2) \), corresponding to the last four rows of the \( \Psi' \) matrix. The friction parameters \( b_i \) varied greatly, depending on the moment the measurement was performed, i.e., whether the motor was cold or not.

The Micro-Delta Robot

MEMory was implemented on the planar parallel manipulator described above and on the micro-Delta robot [1, 23]. The micro-Delta is a 3-DOF parallel robot made of three kinematic chains linked at the traveling plate (Fig. 3). Each of the three chains is driven by a motor fixed to the robot base. Portescap 23 W DC motors with a planetary gear box having a ratio of 24:1 are used. The parallel rods (forearms) assure that the traveling plate always remains parallel to the robot base. The hardware used for control is based on a network of four transputers, enabling a 500 Hz sampling frequency.

This robot experiences friction at the joints. In order to compensate for the rotational backlash, an ellicoidal spring is placed between the arm of the robot and its base. This spring gives a preload to the gear box and has the effect that, at rest, the arms are always pulled up. The amount of preload can be adjusted easily such that the force under normal conditions is higher than the gravity and dynamic forces. The dynamics of the micro-Delta are highly nonlinear and difficult to model accurately by differential equations, so a parametric controller can hardly be used.

Testing the Adaptive FeedForward Controller (AFFC) Algorithm

The AFFC [12] is characterized by the control law

\[
\tau = \Psi(q, \dot{q}, \ddot{q}) \sigma + Ds + \alpha |s|^2 s, \quad s = \dot{e} + \Lambda e,
\]

where \( e = q_d - q \) is the error between the desired and the actual joint position, \( D \) and \( \Lambda \) are two symmetric positive definite matrices, \( \alpha \geq 0 \), and the adaptation law

\[
p_{\text{new}} = p_{\text{old}} + \Delta p, \quad \Delta p = \Gamma \Psi' s,
\]

with \( \Gamma \) a positive definite matrix composed of learning factors.

How should \( \alpha \) be chosen? The \( \alpha \)-term in the control law (2) corrects for the use of the desired state instead of the actual state and thus contributes to control stability [12]. We tested different values of \( \alpha \). We found that the \( \alpha \)-term influenced the results only weakly, and slightly less tracking error was obtained with \( \alpha = 0 \). These surprising results may be due to the fact that the \( \alpha \)-term amplifies the noise of the actual state in a nonlinear way.

All the results presented in this article were obtained with \( \alpha = 0 \). For simplicity, diagonal matrices with positive coefficients were used for \( D \) and \( \Lambda \).

Learning Strategy

A periodical learning trajectory formed of six smoothly joined fifth-order polynomials was used [19], which has a high excitation level and is thus likely to be favorable for the identification of the dynamics [20, 24]. The maximal velocity is about 4.5 rad/s, and the maximal acceleration about 45 rad/s².

As matrix of learning factors we used:

\[
\Gamma = \epsilon \cdot \text{diag}(\gamma_1, ..., \gamma_k).
\]

The \( \gamma_i > 0 \) were chosen so that all the parameters \( \psi_{ij} \) converged approximately at the same time. The use of a diagonal matrix of learning factors reduces the operations necessary for computing the adaptation law relative to a nondiagonal matrix, for example a matrix based on the minimization of the square tracking error [25]. Fast learning resulted from the choice

\[
e = e_1 + e_2,
\]

with \( e_1 \) decreasing exponentially and \( e_2 \) a small constant.
Experiments and Results

The AFFC was tested on the small planar manipulator. The first experiment consisted of learning the parameters starting from \( p = 0 \), i.e., without prior knowledge. The control performances without and with noise were then tested along the learning trajectory and other (test) trajectories. In a second experiment, only the friction parameters were adapted. A third experiment consisted of adding a mass on the end-effector and adapting only the corresponding parameter. For these second and third experiments, the values of the other parameters were frozen by setting the corresponding learning factors to 0. The results of the three experiments were:

- The position and velocity errors decreased to the minimal value in about six seconds (Fig. 4). The eight weights converged to their final value in about 1 minute. The learned parameters were of the same order as the measured one, but had different values [19].
- With the measured and learned dynamics, most of the feedback error was corrected by the feedforward prediction (Fig. 5). The errors in position and velocity were slightly smaller with the learned parameters than with the measured ones (see Table 1). This held true for the learning trajectories and for arbitrary test trajectories. This implies in particular that the friction had been identified well.
- The noise had no adverse effect on learning. The position and velocity errors decreased as quickly as without noise. With noise, the identification of the parameters needed a slightly longer time. The resulting control was good (Fig. 6).
- When only the friction was identified, the convergence time was about 40 s., i.e., two-thirds of the time needed to learn the whole dynamics.
- Reacting to a large change in the mass of the end effector needed about 0.7 s, i.e., about 100 times less than for learning the entire dynamics (Fig. 7).

Study of MEMory

Nonparametric controllers start as feedback controllers. When a movement is repeated, they gradually learn the force necessary to perform this movement and use it as feedforward in subsequent movements [18]. The MEMory controller is based on the idea that the feedback torque corresponds approximately to the dynamics which are not yet compensated for. After each run, the feedback torque is simply added to the feedforward, which is stored in a table. In order to prevent memorization of non-reproducible effects, it was proposed in [6] to use a causal filter and in [21, 26, 27] a forgetting factor. In MEMory a non-causal smoothing is applied on the feedback torque before memorizing it. In this way, the non-reproducible part of the dynamics is filtered, and only its reproducible part should be stored in the table.

Description of the Algorithm

We assume that the manipulator dynamics are composed of a reproducible part \( \tau_r \) and an irreproducible part \( \tau_{irr} \):

\[
\tau_k = \tau_r + \tau_{irr}, \quad k = 1 \ldots K,
\]

where \( k \) is the time index. We further assume that \( \tau_r \) is a random noise term with

\[
\sum_{m=1}^{M} \tau_{irr,m} = 0 \quad \forall k.
\]

Sensor noise and small oscillations fulfill this condition. The control law for the \( j \)-th run is

\[
\tau_{f}(j) = \tau_{ff,j}(j) + \tau_{fb,j}(j) \quad \forall k\text{,}
\]

where \( \tau_{fb,j} \) is the feedback torque vector at time \( k \) (Fig. 8). The learning rule is defined by:

\[
\tau_{fb,j}(0) = 0 \text{,}
\]
Fig. 7. AFFC: Reaction to a modification of the mass of the end-effector. This figure shows the evolution of the parameter corresponding to this mass when only this parameter is adapted.

\[
\tau_{ff}(j+1) = \tau_{ff}(j) + \frac{\lambda}{2M+1} \sum_{n=-M}^{M} \tau_{ff,n}(j),
\]

where \(\lambda\), \(0 \leq \lambda \leq 2\), is the learning factor. The sum in equation (9) constitutes a non-causal filter. We did not use a (common) causal filter as proposed by some authors [18], because this would induce a delay, which sometimes causes instability, as will be seen below.

**Mathematical Properties**

Let \(\tau\) denote the joint torques during the whole movement. In discrete time, \(\tau = (\tau^e)\) is a \(K \times D\)-matrix, with \(K\) the highest time index, and \(D\) the number of motors of the manipulator. Each column \(\tau^d\) of this matrix corresponds to the torque of motor \(d\) during the whole movement, while each row \(\tau_e\) corresponds to the vector of motor torques at time \(k\Delta t\).

We consider that the reproducible torque \(\tau_p\) belongs to the vector space \(V\) of functions of the time which are \(C^2\) (i.e., twice continuous and differentiable) except in a finite set \(S\). For example, according to the rigid-body model, the torque \(\tau_p\) of the small manipulator is smooth along the learning trajectory except in three places (Fig. 5). We introduce an \(l_1\)-norm on \(V\). For every \(v\) from \(V\)

\[
|v'| = \sum_{k=0}^{K} |v_k|. \tag{10}
\]

where \(||\|\) denotes the usual vector norm, i.e.,

\[
|v_k| = \left( \sum_{d=1}^{D} (v'_d)^2 \right)^{1/2}.
\]

We further assume that \(\tau, \tau_{ff},\) and \(\tau_{fb}\) belong to \(l_1(V)\).

With these notations, the smoothing (9) is a convolution

\[
\tau_{ff}(j+1) = \tau_{ff}(j) + \lambda g^* \tau_{fb}(j) \tag{11}
\]

of the feedback torque with a stepwise constant function \(g\). (As usual, the convolution is performed for every dimension independently.) In discrete time, \(g\) is defined by

\[
g(k) = \frac{1}{2M+1} \quad k \leq M
\]

\[
= 0 \quad k > M \tag{12}
\]

The smoothing \(g^*\) is a linear operator from \(l_1(V)\) onto itself. We note that any symmetric function \(f\) with \(f(k) \geq 0 \forall k\) and \(\sum_{k} f(k) = 1\) could be used instead of \(g\). The mathematical developments that will follow remain valid in this case. The case of continuous time is also similar.

<table>
<thead>
<tr>
<th>Table 1. AFFC: Test Results</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Error</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>(10^2 \int q^2)</td>
</tr>
<tr>
<td>(10^2 \int \dot{q}^2)</td>
</tr>
<tr>
<td>(10^{-1} \int q^2)</td>
</tr>
<tr>
<td>(10^{-1} \int \dot{q}^2)</td>
</tr>
</tbody>
</table>

April 1998
From Equations (6) and (8), it follows that
\[ g^* \tau_p(j) = g^* \tau_p + g^* \tau_n(j) - g^* \tau_p(j). \]
(13)

By its definition (7), noise \( \tau_n \) is attenuated by the smoothing (9):
\[ g^* \tau_n(j) = 0 \quad \forall j. \]
(14)

From (11), (13), and (14) follows the iteration
\[ \tau_p(j + 1) = (1 - \lambda g^*) \tau_p(j) + \lambda g^* \tau_p, \]
(15)

where \( I \) is the identity. By induction on \( j \), one finally gets
\[ \tau_p = (I - \lambda g^*)^j \tau_p. \]
(16)

If we can show that
\[ \lambda g^* \rightarrow 0 \quad (j \rightarrow \infty), \]
then it will follow:
\[ \tau_p(j) \rightarrow \tau_p \quad (j \rightarrow \infty), \]
(18)
i.e., the memorized torque converges to the reproducible part of the dynamics.

Proof of Convergence

We will now study when the convergence (18) occurs. When \( \tau_p(t) \) is \( C^2 \) onto \([ (k - M) \Delta t, (k + M) \Delta t ] \), it can be approximated in this interval by its second-order Taylor expansion around \( k \Delta t \):
\[ \frac{d^2 \tau_p}{dt^2} \bigg|_{k \Delta t} (t - (k \Delta t))^2 + o((t - k \Delta t)^2). \]
(19)

It then follows that for every \( k \)
\[ (g^* \tau_p)_k = \tau_p(k \Delta t) + \frac{1}{2M + 1} \sum_{m = -M}^{m = M} (m - k) + o((M \Delta t)^2) \]
(20)

and
\[ (1 - \lambda g^*) \tau_p = (1 - \lambda) \tau_p + \lambda o((M \Delta t)^2). \]
(21)

Using this relation, an upper bound can be derived for \((1 - \lambda g^*) \tau_p\):
\[ [(1 - \lambda g^*) \tau_p] \leq (1 - \lambda) \tau_p + 2Ko((M \Delta t)^2) \]
(22)
or
\[ \frac{(1 - \lambda g^*) \tau_p}{\tau_p} \leq 1 - \lambda + 2Ko((M \Delta t)^2) \]
(23)

From this relation, we see that when \( M \Delta t \) is small, then there exists a \( \lambda (M), 0 < \lambda (M) < 2 \), such that \( \beta < 1 \).

As for any function \( \tau \) from \( l_1(\mathbb{V}) \), \( \max_{V \in \mathbb{V}} |g^* \tau| \leq \max_{V \in \mathbb{V}} |V| \).

Inequality (23) is also valid for the higher powers \( \tau_p, j \geq 1 \), with the same \( \beta \). An iteration on \( j \) gives then
\[ (1 - \lambda g^*) \tau_p \leq \beta^j \tau_p. \]
(24)

When \( \beta < 1 \), this shows the convergence (18) in the \( l_1 \) sense.

Considering the set \( S \) of points where \( \tau_p \) is not \( \mathbb{C}^2 \) leads to add following term to the right side of Inequality (23):
\[ \max_{V \in \mathbb{V}} |(2M + 1) \Delta t| S \ll 1. \]
(25)
This term does not change the above conclusions, because it generally is very small.

It may be surprising that in the definition of the learning law (9) the learning parameter \( \lambda \) can be larger than 1. When \( 0 < \lambda < 1 \), \((1 - \lambda g^*)\tau^r\) is monotonously decreasing and thus \( \tau^r_{pf}(j) \) is monotonously increasing to \( \tau^r \). When \( 1 < \lambda < 2 \), \((1 - \lambda g^*)\tau^r\) is an alternate series, and \( \tau^r_{pf}(j) \) is alternatively larger and smaller than \( \tau^r \). In both cases, however, \( \tau^r_{pf}(j) \) converges with increasing \( j \) to \( \tau^r \).

Test of Convergence

The convergence of the algorithm was tested on many different trajectories performed with the small planar manipulator and the micro-Delta, Fig. 9 shows the evolution of the error during a trajectory repeated by the micro-Delta. One sees that the tracking error decreases in only a few runs and remains small. Fig. 10 shows the corresponding motor torque before and after learning. One sees that the feedback torque is reduced almost to 0, thus the inverse dynamics are well compensated. The error after learning is only a few (\( \approx 2 \)) increments, i.e., the dynamics have been learned at the physical limit of the system.

Robustness, Stability, and Filtering

A fundamental issue is whether control will remain stable and whether learning will succeed with a high level of noise and other non-reproducible perturbations. The robustness of learning control schemes was addressed analytically in [28, 21, 26]. These investigations rely on the rigid-body model equation, but the dynamics of many manipulators, e.g., the micro-Delta, differ significantly from this model. It is therefore necessary to test the robustness of non-parametric controllers experimentally.

The first noise robustness test for MEMory was realized on the small planar manipulator and consisted of learning while using the derivation of the encoder signal as the velocity signal. As for the AFFC, the learning was not disturbed by this large noise [19]. We do not present the corresponding figures, as they are similar to Figs. 5 and 6 obtained with the AFFC.

A more difficult test consisted of implementing MEMory on a large version of the planar manipulator of Fig. 1. Due to the extended limb length (80 cm), this manipulator is flexible, which makes it difficult to control. It is driven by two DC motors and can move with a maximal velocity of about 2 m/s. Using AFFC on this large planar manipulator (with the dynamic equation (1)) resulted in unstable control. Fig. 11 shows the very noisy torque due to the flexible limbs, the filtering of this torque using a causal filter and the noncausal smoothing (9). Several different first- and second-order filters (and in particular the filter proposed in [6]) were tested. All these filters resulted in unstable control after some repetitions, probably due to the delay induced by these filters. In contrast, using the noncausal smoothing, the control remained stable, and learning provided a significant and steady decrease of the tracking error.

How to Choose \( M \) and \( \lambda \)?

When \( M \) is too small, MEMory stores a signal with high frequencies which deteriorates the compensation. How must \( M \) be chosen to attenuate these high frequencies? The magnitude of the Fourier transform of the feedback torque is computed. As the frequencies corresponding to \( \tau_f \) can be evaluated from a plot of the feedback torque before learning (Fig. 11), the cut-off frequency \( \omega_0 \) of the filter can be chosen slightly above the corresponding peaks. In this way, \( \omega_0 \) will be below the peaks corresponding to the highest frequencies, i.e., to \( \tau^r \). \( \omega_0 \) can also be related analytically to \( M \). From (11)

\[
\Delta \tau_{pf}(j + 1) = \tau_{fs}(j + 1) - \tau_{fs}(j) = \lambda g^* \tau_{fs}(j).
\]

Applying a Fourier-transform on this relation gives

![Figure 11: MEMory: Torque measured during a movement of the large parallel planar manipulator (dotted line), and two filtered versions. The solid line corresponds to a second-order Butterworth filter and the dashed line to the noncausal filter of MEM. One time step equals 3 ms, and the torque is in Nm.](image)

![Figure 12: MEMory: Evolution of the position error of joint 2 for different learning factors \( \lambda \) and mean parameters \( M \) (implementation on the micro-Delta manipulator). Dotted line: \( M = 0 \) and \( \lambda = 0.2 \); dashed line: \( M = 2 \) and \( \lambda = 0.05 \); solid line: \( M = 4 \) and \( \lambda = 0.2 \); dashed-dotted line: \( M = 6 \) and \( \lambda = 0.2 \).](image)
In the small manipulator, $M = 0$ resulted in unstable learning and $M \geq 2$ in stable learning (Fig. 12).

Also, $M$ should not be too high. From Equation (23), we namely know that a large $M$ will restrict the range of possible $\lambda(M)$. This observation was confirmed by our experiments. For the micro-Delta, convergence was obtained for $0 < \lambda < 2$, and $\lambda = 0.9$ provided the fastest learning. As a large $M$ must be used for the large planar manipulator (Fig. 11), the learning was convergent only when $\lambda \leq 0.1$. When $\lambda > 0.1$, the feedback error soon diverged.

Inequality (23) could be used for determining the range of possible $\lambda(M)$. However, this relation is coarse, and an experimental approach is likely to provide better results. The practical strategy we propose for choosing $M$ and $\lambda$ is to start with a very small value of $\lambda$ and determine experimentally the best $M$ (larger than the value determined by examining the feedback torque and using Equation (30)). Then, the value of $\lambda$ can be increased until satisfactory learning speed is obtained. For the micro-Delta, $M = 5$ and $\lambda = 0.9$ were determined in this way and provided the results shown in Figs. 9 and 10.

Comparison of MEMory with Parametric Controllers

The control performances obtained with MEMory were compared with those obtained with AFFC, i.e., with (according to the results given earlier) the best method for identifying the parameters of the rigid-body model. These experiments were performed on the small planar manipulator.

Fig. 13 compares the open loop performance after learning obtained with the parametric controller (with AFFC) and with MEM. From this figure, it is clear that MEM compensates better for the plant dynamics than the dynamic equation. This means that MEM has been able to correct systematic errors of the rigid-body model.

MEMory was finally compared with a hybrid controller using a feedforward term from a MEMory in addition to the feedforward term from a rigid-body model dynamic equation (from AFFC). Other hybrid controllers were proposed in [7] and [6]. We found similar results with our hybrid controller and with MEMory alone.

We conclude, in contrast to [6], that it is possible to compensate well for the arm dynamics without parametric model.

Conclusions

This article examined practical aspects of nonlinear adaptive control.

It was shown experimentally that AFFC [12] is robust to noise, enables identification of the friction, and results in better control performance than when the parameters of the dynamic equation are measured. In addition, this controller requires few computations in comparison to other controllers based on the rigid-body model equation and is stable and non-invasive. For all these reasons, we think that using AFFC is currently the best way of identifying the parameters of the robot dynamic equation.

One practical problem with AFFC (and generally with the identification of the parameters of the dynamic equation) is to find a trajectory along which the parameters can be well identified [20, 29, 30]. In this context, it was proposed in [31, 21] to perform the dynamic calibration in several steps, by using several simple trajectories that excite the dynamics corresponding to certain parameters only. We have recently designed an algorithm for switching off the adaptation of parameters varying over time (i.e., friction and the transported mass) when the trajectory does not excite sufficiently the corresponding dynamics [32]. Using this algorithm, it is possible, after an initial calibration, to update the value of time-varying parameters continuously without risking to learn wrong values.
If the task consists of repeating a single movement, it is not necessary to derive the rigid-body dynamic model and identify its parameters. It is simpler and more efficient to use the adaptive look-up table MEMory [19] for learning and compensating the arbitrary dynamics along the repeated trajectory. In this article a domain of the parameters space in which convergent learning is insured was found analytically. The experiments performed confirmed the theoretical analysis and showed that the MEM controller is stable, robust to noise, and compensates for the dynamics at the physical limit of the system. Altogether, these results suggest that nonlinear adaptive controllers can increase the precision of real robots significantly.

Acknowledgments

The experiments reported in this article have been performed at ETH-Zuerich and EPFL-Lausanne, Switzerland. The authors thank Professors G. Schweitzer and R. Clavel for providing the facilities, B. Gruehl, J. Luthiger, M. Nuttin, A. Schmid, and B. Sprenger for their valuable contributions to the experiments, and the reviewers for their constructive comments.

References


Etienne Burdet obtained masters degrees in mathematics and physics and a Ph.D. in robotics from ETH-Zuerich, Switzerland. He is currently a postdoctoral fellow at the School of Kinesiology, Simon Fraser University, Canada. He is doing research in robotics and in neuroscience. He addressed problems related to the visual control of a robot/human arm, the application of nonlinear adaptive control, and human motor adaptation.