Biol. Cybern. 80, 369–382 (1999)

Biological Cybernetics
© Springer-Verlag 1999

A mathematical model of the adaptive control of human arm motions

Robert M. Sanner, Makiko Kosha

University of Maryland, Space Systems Laboratory, College Park, MD 20742, USA

Received: 20 September 1994 / Accepted in revised form: 18 November 1998

Abstract. This paper discusses similarities between models of adaptive motor control suggested by recent experiments with human and animal subjects, and the structure of a new control law derived mathematically from nonlinear stability theory. In both models, the control actions required to track a specified trajectory are adaptively assembled from a large collection of simple computational elements. By adaptively recombining these elements, the controllers develop complex internal models which are used to compensate for the effects of externally imposed forces or changes in the physical properties of the system. On a motor learning task involving planar, multi-joint arm motions, the simulated performance of the mathematical model is shown to be qualitatively similar to observed human performance, suggesting that the model captures some of the interesting features of the dynamics of low-level motor adaptation.

1 Introduction

Cybernetics, as envisioned by Norbert Wiener [33], is the unified study of the information and control mechanisms governing biological and technological systems. In the spirit of this unified vision, we explore below possible bridges between recent models of the adaptive control of multijoint arm motions developed separately in robotics and neuroscience, comparing both the underlying structure as well as the observable performance of the different models.

In neuroscience, new experiments in motor learning [28] present convincing evidence that humans develop internal models of the structure of any external forces which alter the normal dynamic characteristics of their arm motions. These adaptive models are then used to generate compensating torques which allow the arm to follow an invariant reference trajectory to a specified target. As a possible mechanism for this adaptation, it has been conjectured that the internal models, and the compensating torques they generate, are 'pieced together' from a collection of *motor computational elements*, representing abstractions of the actions of individual muscles and their neural control circuitry [18, 28]. Motor learning in this context can thus be viewed as a method for continuously adjusting the contribution of each computational element so as to offset the effects of new environmentally imposed forces.

On the other hand, working from first principles within the framework of nonlinear stability theory, a new class of robot control algorithms has been developed which closely mirrors this biological model [27]. Formalizing and extending previous biologically inspired manipulator control algorithms, e.g. [9, 14–16], these new controllers allow simultaneous learning and control of arbitrary multijoint motions with guaranteed stability and convergence properties.

This confluence of formal mathematics and observed neurobiology is quite interesting and merits further exploration. In this paper, we present an initial evaluation of the ability of the new robot control algorithm to also provide a model of the adaptation of human multijoint arm motions. Specifically, we compare the performance of the new algorithm on a simulation of one of the learning tasks used in [28] to the actual performance of human subjects on the same task. As shown below, the new model not only reproduces many of the measured end results of this motor learning task, but also captures a substantial component of the actual time evolution of the adaptation observed in human subjects. The qualitative correlation between the simulated and measured adaptive performance suggests that the proposed algorithm may provide a model for some of the interesting features of low-level motor adaptation.

2 Models of arm dynamics and control mechanisms

2.1 Arm dynamics and plausible controller structures

To a good approximation, human arm dynamics can be modeled as the motion of an open kinematic chain of

Correspondence to: R.M. Sanner
(e-mail: rmsanner@eng.umd.edu)

rigid links, attached together through revolute joints, with control torques applied about each joint [4, 30]. Within the limits of flexure of each joint, human arm motions can thus be modeled by the same equations used to model revolute robot manipulators, i.e.,

$$\mathbf{H}(q)\ddot{q} + \mathbf{F}(q,\dot{q}) + \mathbf{G}(q) + \mathbf{E}(q,\dot{q}) = \tau \tag{1}$$

where $q \in \mathscr{R}^n$ are the joint angle of the arm. The matrix $\mathbf{H} \in \mathscr{R}^{n\times n}$ is a symmetric, uniformly positive definite inertia matrix, the vector $\mathbf{F} \in \mathscr{R}^n$ contains the Coriolis and centripetal torques, the vector $\mathbf{G} \in \mathscr{R}^n$ contains the gravitational torques (and hence is identically zero for motions externally constrained to the horizontal plane), and finally the vector $\mathbf{E} \in \mathscr{R}^n$ represents any torques applied to the arm through interactions with its environment. The forcing input $\tau \in \mathscr{R}^n$ represents the control torques applied at each arm joint.

Given the recent progress in the development of adaptive, trajectory following, robotic control laws [23, 30, 31], it is natural to wonder whether in fact human arm control algorithms have a similar structure. Each adaptive robot control algorithm breaks down into two components. The first of these is a fixed one which, given perfect information about the dynamics of the robot and its environment, counteracts the natural dynamic tendency of the robot and ensures that the closed-loop arm motions are asymptotically attracted to a task-dependent *desired trajectory*. The second, adaptive component recursively develops an estimated model of the governing dynamics, which is then used in place of the (unknown) true model in the fixed component. The desired trajectory is present as a signal exogenous to the arm control loop, depending upon the nature of the task at hand; the controller uses this signal together with measurements of the position and velocity of each joint and the current estimated model to generate the necessary torques.

For such an algorithm to be plausible as a model of human arm motions, there must first be evidence in humans of a task-dependent desired trajectory which the arm attempts to follow. Experimental and theoretical evidence supporting this has been reported in [6, 11], which show that, in the absence of other constraints, for planar arm motions humans appear to make rest-to-rest, point-to-point motions in a manner which minimizes the derivative of hand acceleration. This is the so-called 'minimum jerk' trajectory through the arm's workspace. The desired arm trajectory is thus a function only of the end points of the required motion, and the *organization* of motion is decoupled from the *execution* of motion, as with an adaptive robot controller.

A second criterion for plausibility is evidence that humans adaptively form internal representations of their own and any environmental dynamics, and that these representations are used to ensure that arm motions follow the desired, i.e., minimum jerk, trajectory. Using a series of innovative experiments, the study [28] presented convincing evidence that both of these properties of human arm control are present, at least for two degrees of freedom, planar arm motions. By measuring how humans learned to compensate for an externally applied disturbance to their arm motions, this study concluded that the end-effect of human sensory-motor learning in a new dynamic environment is an internal representation, in joint coordinates, of the forces applied by the new environment. This representation is then used to generate compensating torques which allow the arm to follow the minimum jerk trajectory.

On the basis of these and prior experiments, [28] proposed a control law which may describe the structure of the motor control strategy employed in planar arm motions. The following section describes their control law mathematically and contrasts it with the structure of the control laws used for adaptive robot manipulators.

2.2 Arm control laws: mathematical descriptions

To explain the observed performance of their human subjects, the following trajectory tracking control law was proposed in [28]:

$$\begin{aligned}\tau(q,\dot{q},t) = {}& \widehat{\mathbf{H}}(q)\ddot{q}^m(t) + \widehat{\mathbf{F}}(q,\dot{q}) + \widehat{\mathbf{E}}(q,\dot{q}) \\ & - \mathbf{K}_\mathrm{D}\dot{\tilde{q}}(t) - \mathbf{K}_\mathrm{P}\tilde{q}(t)\end{aligned} \tag{2}$$

where $\mathbf{K}_\mathrm{D}$ and $\mathbf{K}_\mathrm{P}$ are constant, positive, definite matrices, $\tilde{q}(t) = q - q^m(t)$, and $q^m(t)$ is the model (minimum jerk) trajectory the joint angles q are required to follow. The terms $\widehat{\mathbf{H}}, \widehat{\mathbf{F}}$, and $\widehat{\mathbf{E}}$ represent learned estimates of the corresponding terms which appear in the equations of motion (1). This control law thus consists of constant linear feedback terms together with learned estimates of the nonlinear components of (1).

On the other hand, working directly from the mathematical structure of (1), an alternative, equally plausible controller structure can be identified which, as will be shown below, is directly amenable to stable, continuous, adaptive operation. To understand the structure of this algorithm, first define a new measure of the tracking error

$$s(t) = \left(\frac{\mathrm{d}}{\mathrm{d}t} + \Lambda\right)\tilde{q} = \dot{\tilde{q}}(t) + \Lambda\tilde{q}(t) \tag{3}$$

where Λ is a constant, positive, definite matrix. Note that this algebraic definition of the new error metric s also has a dynamic interpretation: The actual tracking errors $\tilde{q}$ are the output of an exponentially stable linear filter driven by s. Thus, a controller capable of maintaining the condition $s = 0$ will produce exponential convergence of $\tilde{q}(t)$ to zero, and hence exponential convergence of the actual joint trajectories to the desired trajectory $q^m(t)$.

Use of this metric allows the development of control laws for (1) which directly exploit the natural passivity (conservation of mechanical energy) property of these systems [31]. Consider the following control law

$$\tau(q,\dot{q},t) = \widehat{\mathbf{H}}(q)\ddot{q}^r(t) + \widehat{\mathbf{C}}(q,\dot{q})\dot{q}^r(t) + \widehat{\mathbf{E}}(q,\dot{q}) - \mathbf{K}_\mathrm{D}(t)\dot{\tilde{q}}(t) - \mathbf{K}_\mathrm{D}(t)\Lambda\tilde{q}(t) \tag{4}$$

where $\dot{q}^r(t) = \dot{q}^m(t) - \Lambda\tilde{q}(t)$, and both $\mathbf{K}_\mathrm{D}$ and Λ are positive definite matrices, with $\mathbf{K}_D$ possibly time varying. For fixed feedback gains, this is very similar to (2), but substituting $\ddot{q}^r(t)$ for $\ddot{q}^m(t)$, and utilizing the known (but nonunique) factorization

$$\mathbf{F}(q,\dot{q}) = \mathbf{C}(q,\dot{q})\dot{q}$$

Denoting by $\tau^o(q,\dot{q},t)$ the control law obtained using the actual matrices H, C and the actual external forces $\mathbf{E}$, the resulting closed loop equations of motion can be written, after some manipulation, as

$$\mathbf{H}\dot{s} = -\mathbf{K}_\mathrm{D}s - \mathbf{C}s + \tilde{\tau}(q,\dot{q},t) \tag{5}$$

where $\tilde{\tau} = \tau - \tau^o$. With these dynamics, the uniformly positive definite energy function $V(s,t) = s^\mathrm{T}\mathbf{H}(q(t))s/2$ has a time derivative

$$\dot{V}(s,t) = -s^\mathrm{T}\mathbf{K}_\mathrm{D}(t)s + s^\mathrm{T}\tilde{\tau} + s^\mathrm{T}(\dot{\mathbf{H}} - 2\mathbf{C})s/2$$

Conservation of energy for the mechanical system (1) identifies a specific $\mathbf{C}$ for which the matrix $\dot{\mathbf{H}} - 2\mathbf{C}$ is skew symmetric, thus rendering the last term above identically zero. The energy function thus satisfies the *dissipation inequality* [31]

$$\dot{V}(s,t) \leq -k_\mathrm{D} \parallel s \parallel^2 + s^T\tilde{\tau}$$

where k_D is a uniform lower bound on the eigenvalues of $\mathbf{K}_\mathrm{D}(t)$. The closed-loop dynamics hence describe a passive input-output relation between s and $\tilde{\tau}$, a fact which is instrumental in the development of stable, online adaptation mechanisms, such as those examined below. Moreover, if $\tilde{\tau} \equiv 0$, that is if 'perfect' knowledge is incorporated into the controller, then the energy function is actually a Lyapunov-function for the system, showing that $s(t)$, and hence $\tilde{q}(t)$, decay exponentially to zero from any initial conditions using $\tau = \tau^o$ [30].

3 Adaptive arm control and 'neural' networks

How might the control law (4) be implemented biologically and, more importantly, how do the components of the controller evolve in response to changing environmental force patterns or changes in the physical properties of the arm? In the following section, this question is considered from both a mathematical and a biological viewpoint, and the two vantages are shown to suggest quite similar solutions.

3.1 Adaptive robot controllers and possible biological analogs

Adaptive robot applications exploit a factorization of the nonlinear components

$$\tau^\mathrm{nl}(q,\dot{q},t) = \mathbf{H}(q)\ddot{q}^r(t) + \mathbf{C}(q,\dot{q})\dot{q}^r(t) + \mathbf{E}(q,\dot{q}) = \mathbf{Y}(q,\dot{q},t)a \tag{6}$$

where prior knowledge of the exact structure of the equations of motion is used to separate the (assumed known) nonlinear functions comprising the elements of $\mathbf{H}, \mathbf{C}$, and $\mathbf{E}$ from the (unknown but constant) physical parameters in the vector a. Indeed, with this factorization, the control law

$$\tau(q,\dot{q},t) = -\mathbf{K}_\mathrm{D}(t)s(t) + \mathbf{Y}(q,\dot{q},t)\widehat{a}(t) \tag{7}$$

which uses estimates of the physical parameters in place of the true values, coupled with the continuous adaptation law

$$\dot{\widehat{a}}(t) = -\mathbf{\Gamma}\mathbf{Y}^\mathrm{T}(q,\dot{q},t)s(t) \tag{8}$$

where Γ is a symmetric, positive, definite matrix controlling the rate of adaptation, results in globally stable operation and asymptotically perfect tracking of any sufficiently smooth desired trajectory [30].

Although this solution is mathematically elegant, it seems unlikely that the human nervous system is specifically hardwired with a particular set of nonlinear functions to be used in motion control laws. Moreover, since generally the forces, $\mathbf{E}$, imposed by the environment will be quite complex and variable in structure, it is not apparent how such a simple parameterization could adequately capture the entire possible range of environments which might be encountered.

From a biological perspective, the study [28] suggests, similar to [3, 18], that internal models of the nonlinear terms, and the compensating torques they generate, are 'pieced together' from elementary structures collectively called *motor computational elements*. These structures represent abstractions of the low-level biological components of motor control, whose contributions at any time depend upon the instantaneous configuration $q(t)$ and instantaneous velocity $\dot{q}(t)$ of the arm. Moreover, experimental evidence suggests that these computational elements are *additive*, so that simultaneous stimulation of two motor control circuits results in a (time-varying) output torque which is the sum of the torques which would result from separate stimulation of each circuit [8, 20].

These observations suggest that the structure of the adaptive nonlinear component of human control of arm motions can be represented using a superposition of the form

$$\widehat{\tau}_i^\mathrm{nl}(q,\dot{q},t) = \sum_{k=1}^{N} \widehat{\alpha}_{i,k}(t)\varphi_k(q,\dot{q},t) \tag{9}$$

where $\widehat{\tau}_i^\mathrm{nl}(q,\dot{q},t)$ is an (adaptive) estimate of the nonlinear torque required about the ith joint given the current state of the limb and the desired motion. Each φ_k represents a (configuration- and velocity-dependent) torque produced by a motor computational element, and $\widehat{\alpha}_{i,k}(t)$ are weights which represent the relative strength of each elementary torque at time t. Motor learning in this context can thus be viewed as a method for re-weighting the elementary torques so as to offset the effects of new environmentally imposed forces or changes in arm physical properties.

A possible mechanism for learning these relative weightings is suggested by the Hebbian model of neuroplasticity [10], in which a synaptic strength is modified according to the temporal correlation of the firing rates of the neurons it joins. Since the end product of the motor learning tasks considered herein is achieved when the arm follows the desired trajectory, a natural first approximation to the dynamics of the learning process which incorporates the above ideas is:

$$\dot{\widehat{\alpha}}_{i,k}(t) = -\gamma\varphi_k(q(t), \dot{q}(t), t)s_i(t)$$

where γ is a constant which controls the rate of learning. In this scheme, each weight $\widehat{\alpha}_{i,k}$ evolves in time according to the correlation of the elementary torque output, φ_k, and the tracking error measure s_i.

Significantly, the two previous equations precisely express the structure of the adaptive component of a class of recently developed robot control algorithms [27], which join the stable 'neural' control algorithms in [26] with the adaptive robot algorithm in [30]. The following two sections make this connection formally, exploring in detail the structure of this algorithm and its relation to the motor computational element conjecture.

3.2 Modeling the motor computational elements

To develop a firm mathematical basis for the ideas developed in the preceding section, consider the following alternative representation of the nonlinear component of the required control input:

$$\begin{aligned}\tau^{\mathrm{nl}}(q, \dot{q}, t) &= \mathbf{H}(q)\ddot{q}^r(t) + \mathbf{C}(q, \dot{q})\dot{q}^r(t) + \mathbf{E}(q, \dot{q}) \\ &= \mathbf{M}(q, \dot{q})v(q, \dot{q}, t)\end{aligned}$$

or, in component form,

$$\tau_i^{\mathrm{nl}}(q, \dot{q}, t) = \sum_{j=1}^{2n+1} M_{i,j}(q, \dot{q})v_j(q, \dot{q}, t)$$

where $v_l = \ddot{q}_l^r$, $v_{l+n} = \dot{q}_l^r$, for $l = 1 \dots n$, and $v_{2n+1} = 1$. Unlike expansion (6), which decomposes τ^{nl} into a matrix of known functions, $\mathbf{Y}$, multiplying a vector of unknown constants a, this expansion decomposes τ^{nl} into a matrix of (potentially) unknown functions $\mathbf{M}$, multiplying a vector of known signals v. Without the prior information assumed above, an adaptive controller capable of producing the required control input must learn each of the unknown component *functions*, $M_{i,j}(q, \dot{q})$, as opposed to the conventional model which must learn only the unknown *constants*, a.

The motor computational element conjecture suggests that approximations to the necessary functions are 'pieced together' from the simpler functions φ_k. Significantly, abstract models of biological computation strategies have been shown to have precisely this function approximation property [5, 7, 12, 21], provided each $M_{i,j}(q, \dot{q})$ is continuous in q and $\dot{q}$. Indeed, if this is the case, for many different computational models there exists an expansion which satisfies, for any $(q, \dot{q})$ contained in a prespecified compact set $A \subset \mathscr{R}^{2n}$,

$$\left| M_{i,j}(q, \dot{q}) - \sum_{k=1}^{N} c_{i,j,k}g_k(q, \dot{q}, \xi_k) \right| \leq \epsilon_{i,j}$$

for any chosen accuracy $\epsilon_{i,j}$. This expansion approximates each component of the matrix $\mathbf{M}$ using a single hidden layer 'neural' network design with $(q, \dot{q})$ as the network inputs; here g_k is the model of the signal processing performed by a single 'neural' element or node, ξ_k are the 'input weights' associated with node k, and $c_{i,j,k}$ is the output weight associated with that node.

This theoretical result has been further strengthened with the development of constructive algorithms allowing a precise specification of N and ξ_k based upon estimates of the *smoothness* of the functions being approximated [26]. For example, in radial basis function models, i.e., models for which $g_k(x, \xi_k) = g(\|hx - \xi_k\|)$ for some positive scaling parameter h, the parameters ξ_k can be chosen to encode a uniform mesh over the set A whose spacing is determined by bounds on the significant frequency content of the Fourier transform of the functions being approximated. This analysis thus leaves only the specific output weights, $c_{i,j,k}$, to be learned in order to accurately approximate the unknown functions $M_{i,j}$.

Since the size of the required networks rises rapidly with the number of independent variables in the functions to be learned, a practical implementation will maximally exploit prior information to reduce the network size. To this end, in [27] it is noted that the term $\mathbf{C}(q, \dot{q})\dot{q}^r$ may be further decomposed as

$$\mathbf{C}(q, \dot{q})\dot{q}^r = \mathbf{C}_1(q)[\dot{q}\dot{q}^r]$$

where $\mathbf{C}_1(q) \in \mathscr{R}^{n \times n^2}$, and $[\dot{q}\dot{q}^r] \in \mathscr{R}^{n^2}$ contains all possible combinations $\dot{q}_i\dot{q}_j^r$, for $i, j = 1, \dots, n$. If a similar decomposition of $\mathbf{E}$ is assumed, for example $\mathbf{E}(q, \dot{q}) = \mathbf{E}_1(q)p(\dot{q})$ where now $\mathbf{E}_1(q) \in \mathscr{R}^{n \times n}$ and $p(\dot{q}) \in \mathscr{R}^n$ represents an assumed known $\dot{q}$ dependence, the number of independent variables in each unknown function is reduced by a factor of 2. Indeed, the nonlinear terms can under these conditions be decomposed as

$$\tau^{\mathrm{nl}}(q, \dot{q}, t) = \mathbf{N}(q)w(q, \dot{q}, t)$$

where $w \in \mathscr{R}^{n(n+2)}$ now contains the elements of $\ddot{q}^r$, $[\dot{q}\dot{q}^r]$, and $p(\dot{q})$.

Thus, assuming the functions required for each component of τ^{nl} are sufficiently smooth, a network approximation of the form

$$\tau_i^{\mathscr{N}}(q, \dot{q}, t) = \sum_{j=1}^{n(n+2)} \sum_{k=1}^{N} c_{i,j,k}g_k(q, \xi_k)w_j(q, \dot{q}, t) \tag{10}$$

can accurately approximate the required nonlinear control input for appropriate values of the network parameters N, ξ_k, and $c_{i,j,k}$. Indeed, defining $d = \tau^{\mathrm{nl}} - \tau^{\mathscr{N}}$ one has

$$|d_i(q, \dot{q}, t)| \leq \sum_{j=1}^{n(n+2)} \epsilon_{i,j} |w_j(q, \dot{q}, t)|$$

for any inputs $(q, \dot{q}) \in A$, where each $\epsilon_{i,j}$ is now the worst case network approximation error to the components of N. Since A is compact, and since the smooth, minimum jerk cartesian paths produce correspondingly smooth desired joint trajectories, $\|d\|$ is uniformly bounded provided $(q, \dot{q})$ can be guaranteed to remain in A. Furthermore, this bound can be made arbitrarily small by increasing the size of the approximating network [27].

There remains to specify the set A, defining the region of the arm's state space on which the networks must have good approximating abilities. For planar arm motions, note that the joint variables can be mathematically constrained to lie in the compact set $(-\pi, \pi]^n$, and most are physically constrained to lie in a strict subset of this. Moreover, since there are physical limits on the torques which can be exerted by the muscles, and physical limits on the possible range of joint angles, there are corresponding limits on the range of joint velocities which can be produced. The mechanical properties of the arm thus naturally determine the 'nominal operating range', A, in which the motor controller would need to develop an accurate model of the arm's dynamics.

The expansion (10) utilizes "neural" network theory, the motor computational element conjecture of [28], and established theories of adaptive robot control to develop a representation of each weighted computational element as

$$\alpha_{i,k}\varphi_k(q, \dot{q}, t) = g_k(q, \xi_k) \sum_{j=1}^{n(n+2)} c_{i,j,k} w_j(q, \dot{q}, t)$$

However, representation (10) allows a more finely grained approach to the control problem than expansion (9), and one more closely tied to the natural dynamics of the system, by allowing independent adjustment of each of the $c_{i,j,k}$ to determine the required input. Moreover, the above construction is by no means unique: instead of (10), which uses a single network to approximate the components of $\tau^{\mathcal{N}}$, one could easily imagine different networks approximating each of these terms. This strategy would produce a family of different g_k defining each motor control element, corresponding to the different nodes used in each approximating network.

Note that, while the $\mathbf{N}w$ decomposition used above allows a significant reduction in the number of network nodes and weights required to achieve a specified tracking accuracy [27], there is no biological significance claimed for this simplification. It is done purely to minimize the calculations required in the simulations below, and a more general implementation could of course use the original $\mathbf{M}v$ parameterization and basis functions g_k with both joint angles and joint velocities as inputs.

Additionally, the analysis in this section shows merely that it is possible to construct the necessary control inputs from a collection of very simple elements; it says nothing about the actual structure of the elements used in human motor control. Further experimentation is needed to determine a precise description of these elements in humans, and to thus permit development of a truly accurate model of human motion control. Significantly, however, the algorithm in [27] requires only that the collection $\{\varphi_k\}$ be mathematically 'dense' in the class of possible functions needed in the control law. Provided the elements employed can satisfy the above approximation conditions, the resulting controller, coupled with the adaptation mechanism developed in the following section, will asymptotically track any smooth desired trajectory. This relative insensitivity to the specific structure of the approximating elements suggests that high quality simulation models can be developed without precise knowledge of human physiology, an idea which will be explored more fully in Sec. 4.

3.3 Adapting the motor computational elements

To understand how the representations developed above might be adaptively used, a comparison of the 'neural' expansion (10) with the adaptive robot algorithm (7)–(8) suggests the control law

$$\tau(q, \dot{q}, t) = -\mathbf{K}_{\mathrm{D}}(t)s(t) + \hat{\tau}^{\mathcal{N}}(q, \dot{q}, t) \tag{11}$$

where

$$\hat{\tau}_i^{\mathcal{N}}(q, \dot{q}, t) = \sum_{j=1}^{n(n+2)} \sum_{k=1}^{N} \hat{c}_{i,j,k}(t) g_k(q, \xi_k) w_j(q, \dot{q}, t) \tag{12}$$

which again uses estimates of the required parameters in place of the (assumed unknown) actual values. Use of (11) with dynamics (1) produces the closed-loop dynamics

$$\mathbf{H}\dot{s} = -\mathbf{K}_{\mathrm{D}}s - \mathbf{C}s + \mathbf{Y}^{\mathcal{N}}\tilde{c} + d$$

where the elements of the vector $\tilde{c}(t)$ are of the form $\hat{c}_{i,j,k}(t) - c_{i,j,k}$, and the elements of $\mathbf{Y}^{\mathcal{N}} \in \mathscr{R}^{n \times N(n^2+2n)}$ are the corresponding combinations $g_k(q, \xi_k) w_j(q, \dot{q}, t)$.

These are precisely the closed-loop dynamics of the adaptive system obtained using (7), but for the presence of the disturbance term, d, describing the discrepancy between the required nonlinear torques and the best possible network approximation to them. As shown in [26, 27], proper treatment of this disturbance is fundamental to the development of a successful adaptation algorithm. The control and adaptation laws (7)–(8) depend upon the *globally exact* linear parameterization $\tau^{\mathrm{nl}} = \mathbf{Y}a$. The representation (10), however, provides only the *locally approximate* parameterization $\tau^{\mathrm{nl}} = \mathbf{Y}^{\mathcal{N}} c + d$, requiring special care in the design of an adaptation mechanism.

As long as the joint angles and velocities remain within their nominal operating range A, the impact of the disturbance term can be accommodated by using *robust* adaptation methods, for example, by adding a weight decay term to the adaptive algorithm. The adaptive law (8) becomes in this case

$$\dot{\hat{c}}_{i,j,k}(t) = -\gamma(\omega(t)\hat{c}_{i,j,k}(t) + s_i(t)w_j(q,\dot{q},t)g_k(q(t),\xi_k)) \tag{13}$$

where $\omega(t) = 0$ if $\|\hat{c}(t)\| < c_0$, and $\omega(t) = \omega_0 > 0$ otherwise, and here c_0 is an upper bound on the total magnitude of the parameters required to accurately approximate τ^{nl}. The parameter γ is a positive constant which controls the rate of learning, and can be different for each weight. An equally effective robust modification to the adaptive law, and one which may be more biological, is to allow each parameter value to *saturate* using a projection algorithm of the form

$$\dot{\hat{c}}_{i,j,k}(t) = \mathscr{P}(-\gamma s_i(t)w_j(q,\dot{q},t)g_k(q(t),\xi_k), \hat{c}_{i,j,k}(t), c_{\max}) \tag{14}$$

where $\mathscr{P}(x,y,z) = x$ if $-z < y < z$, or if $y \le -z$ and $x > 0$, or if $y \ge z$ and $x < 0$; $\mathscr{P}(x,y,z) = 0$ otherwise.

Above it has been argued that the architecture of the arm ensures that the joint state variables remain within an easily computable nominal range, A. In the more general nonlinear control case considered in [26, 27], where such physical arguments may not be immediately obvious, a modification to the control law (11) is required if the state variables ever leave their predefined nominal range. This is accomplished by adding a 'supervisory' or robust component to (11) itself, whose action is mathematically determined to force the state back into the nominal range. Indeed, the mechanical constraints in a human arm, through the forces they exert as the arm approaches the feasible configuration boundaries, can be viewed as a realization of this 'supervisory' control action. Similarly, it can be argued that painful stimulus from hyperextension of a joint would provoke an analogous 'over-ride' of the nominal motor control strategy, forcing the limb to return to a more relaxed configuration. The formal details of the required modifications are provided in [27].

This combination of 'neural' approximation, robust online adaptation, and robust 'supervisory' action (if required) can be proven to result in a globally stable closed-loop system. In addition, the actual joint trajectories can be shown to asymptotically converge, in the mean, to a small neighborhood of the desired trajectories [26, 27]. Explicitly, the convergence can be expressed as

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T \|\tilde{q}(t)\|^2 \mathrm{d}t \le \frac{\epsilon}{k_{\mathrm{D}}^2\lambda^2} \tag{15}$$

where

$$\epsilon \overset{\Delta}{=} \sup_t \sup_{x\in A} \sum_{i=1}^{n} |d_i(q,\dot{q},t)|^2$$

and λ is the smallest eigenvalue of Λ. Larger linear feedback gains and/or networks with better approximation capabilities will thus reduce the asymptotic tracking errors.

4 Simulating human motor learning

While the gross features of the controller (11)–(14) seem to agree with experimental observations about the structure of human motor control mechanisms, it is not known to what extent this algorithm accurately models the actual biological mechanisms of adaptation. Instead, the algorithm constitutes a testable hypothesis about adaptive motor control, where mathematics and nonlinear control theory have been used to bridge the gaps in available neurobiological data. This section thus presents a preliminary evaluation of the properties of this model, qualitatively comparing its performance during a specific motor learning task with that of human subjects.

The selected task is a simulation of the experiment used to train the human subjects in [28]. This experiment consists of making short reaching motions constrained to the horizontal plane, while the arm is perturbed by an unknown, but deterministic, pattern of externally applied forces. When no perturbing forces are applied, the observed human arm motions are virtually straight lines from the starting point to the desired target, with a bell-shaped velocity profile in agreement with a minimum jerk trajectory. Under the influence of the perturbations, the motions are initially sharply deflected from the nominal straight line motion. With practice, however, these deflections are almost entirely eliminated, reflecting the adaptation of the motor control strategies utilized by the subjects.

As will be shown below, not only is this behavior reflected in the simulation using the proposed adaptive control law, but the actual time evolution of the controller performance closely resembles that recorded from the human subjects.

4.1 Simulation construction

A simulation model of two degree of freedom arm motions (see Fig. 1) was created using the dynamics (1), with $n = 2$, and the representative human arm mass and length parameters reported in [28]. The two degrees of freedom here correspond to elbow and shoulder rotations in the horizontal plane; for the purposes of these experiments, the hand can be considered rigidly attached to the end of the arm, contributing no additional degrees of freedom.

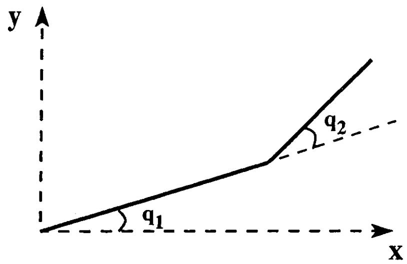


Fig. 1. Model of two degrees of freedom, planar arm motions

The desired trajectories driving the arm motions were computed from the experimental tasks used to train the human subjects in [28]. Each of these tasks consisted of a 10 cm reaching motion, possibly in the presence of an (initially) unknown pattern of environmental forces. The desired endpoint for each reaching motion moved in a pseudorandom fashion throughout a 15 by 15 cm workspace centered at (0.26 m, 0.42 m) relative to the subject's shoulder (the location of the q_1 joint in the model pictured in Fig. 1).

To generate the sequence of desired motions, the hand was initially placed in the center of the workspace. A direction was chosen randomly from the set $\{0°, 45°, \ldots, 315°\}$ measured clockwise with 0° corresponding to motion in the $+y$ direction. The desired endpoint for the reaching motion was then 10 cm along this direction. After the hand reached this target, a new target was chosen at a distance of 10 cm from the old target and along a new randomly selected direction. The selection process was modified to keep the targets within the 15 by 15 cm workspace.

To generate a desired trajectory corresponding to each reaching motion, a minimum jerk hand path of duration 0.65 s was assumed, as in [28]. In the simulation, the hand was allowed a total of 1.3 s to reach the desired target before a new target was selected. Thus, the desired trajectory for each reaching motion consisted of a 0.65 s minimum jerk path to the target, followed by a 0.65 s hold at the target.

The controller was initialized with perfect 'self-knowledge', i.e., at $t = 0$, $\hat{\mathbf{H}} = \mathbf{H}$ and $\hat{\mathbf{C}} = \mathbf{C}$, but no knowledge of any external forces, i.e., $\hat{\boldsymbol{E}} = 0$ at $t = 0$. This was accomplished by modifying the control law (11) slightly, so that

$$\tau(q, \dot{q}, t) = -\mathbf{K}_{\mathrm{D}} s + \mathbf{H}(q)\ddot{q}^r(t) + \mathbf{C}_1(q)[\dot{q}\dot{q}^r(t)] + \widehat{\tau}^{\mathscr{N}}(q, \dot{q}, t) \tag{16}$$

with $\widehat{\tau}^{\mathscr{N}}$ still given by (12) and all $\widehat{c}_{i,j,k}(0) = 0$.. The adaptive network contributions in this case thus learn only the departures from the nominal model the controller has developed from an assumed prior 'lifetime' of practice. This initialization is by no means necessary, and is done only to facilitate comparison with the experimental and simulation results reported in [28]. The robotic examples considered in [27] demonstrate the ability of the algorithm to track any desired trajectory quickly with no such prior information.

The torques applied about each joint were determined using the control laws (14), (12), and (16), together with the gain matrices

$$\mathbf{K}_{\mathrm{D}} = \begin{bmatrix} 2.3 & 0.9 \\ 0.9 & 2.4 \end{bmatrix} \quad \Lambda = \begin{bmatrix} 6.5 & 0.064 \\ 0.064 & 6.67 \end{bmatrix}$$

which were computed using the representative joint stiffness and viscosity coefficients reported in [19, 28]. Note that measurements of human arm motion suggest that the effective stiffness component becomes smaller as $\tilde{q}$ increases [29]. While the proposed control law (11) has the flexibility to accomodate these time-varying feedback gains, in the absence of an analytic model for these variations in humans and to facilitate comparisons with [28], this feature has not been exploited here. Similarly, since the arm motions required to perform the experiment are well within the workspace of the simulated arm, a simulation of the constraint forces imposed by the joint limits was not included.

4.2 Network design

A radial basis function network was employed, with nodes $g_k(q) = g(hq - k)$ for a fixed scale parameter $h > 0$ and a fixed range of translations $k \in \mathscr{K} \subset \mathscr{Z}^2$. Such a network is known to be capable of approximating continuous functions with an accuracy proportional to h^{-r}, uniformly on the interior of a domain spanned by the translates k/h, $k \in \mathscr{K}$ [22, 26]. The constant $r > 0$, quantifying the rate of convergence of the approximation, depends upon the specific basis function g as well as the smoothness of the functions being approximated by the network.

For this study, a Gaussian basis function was chosen, with $g(q) \exp(-\|q\|/2)$, and the domain on which good approximation is required is the subset of $[-\pi, \pi]^2$ containing the range of joint angles required during the experiment, which here is $A = [-.5, 2] \times [.5, 2.5]$. The scale factor, h, in the network was chosen as $h = 2$, ensuring that the 'width' of the Gaussians was broad enough to allow the possibility of generalization of network learning across the set A, and the translation range was correspondingly chosen as $\mathscr{K} = [-5, \ldots, 8] \times [-3, \ldots, 9]$, producing a total of 182 nodes. Larger values of h would allow a theoretically better approximation (since ϵ in (15) is proportional to h^{-r}), at the expense of a larger network size (since the range of translates $k/h, k \epsilon \mathscr{K}$ must still cover the same set A) and decreased generalization capability (since each Gaussian will be more 'narrow', and thus contribute little to the approximation at points in A remote from its center k/h). The analyses in [25, 26] provide a more precise mathematical discussion of these tradeoffs. The specific values chosen above attempt to balance the conflicting goals of high accuracy, small network size, and good generalization potential, but for this preliminary study no attempt was made to optimize this tradeoff.

The resulting network can be expressed as

$$\hat{\tau}_i^{\mathscr{N}}(q, \dot{q}, t) = \sum_{j=1}^{8} \sum_{k \in \mathscr{K}} \hat{c}_{i,j,k}(t) g(hq - k) w_j(q, \dot{q}, t)$$

where

$$w^T = [\ddot{q}_1^r, \ddot{q}_2^r, \dot{q}_1\dot{q}_1^r, \dot{q}_1\dot{q}_2^r, \dot{q}_2\dot{q}_1^r, \dot{q}_2\dot{q}_2^r, \dot{q}_1, \dot{q}_2]$$

and there are thus a total of 2912 adjustable output weights which must be learned. Each output weight was updated during the experiment using (14) with the conservative upper bound $c_{\max} = 75$. The learning rates were chosen to vary with j; the specific values $\gamma_j = 0.01$

for $j = 1, \ldots, 6$ and $\gamma_j = 0.04$ for $j = 7, 8$ were used in the simulation.

4.3 Adaptive controller performance

To display the evolving performance of the controller, a set of 8 reaching motions originating at the center of the workspace and extending 10 cm along each of the directions in the above set was used. The resulting 'star pattern', corresponding to the minimum jerk trajectories to these targets, is shown in Fig. 2. With no external forces acting on the system and no additional loads placed upon the arm, the control law as initialized should be able to perfectly track these trajectories, and indeed, Fig. 2 is exactly reproduced using the baseline controller.

In the presence of new environmental forces, however, substantial deviations from these trajectories are expected until the controller builds up a sufficient internal model of the new forces. Figure 3 shows the initial performance of the controller on the star pattern when the arm is subject to the field

$$\mathbf{E}(q, \dot{q}) = J^T(q)BJ(q)\dot{q} \tag{17}$$

where

$$B = \begin{bmatrix} -10.1 & -11.2 \\ -11.2 & 11.1 \end{bmatrix}$$

and J is the Jacobian of the mapping from joint to Cartesian coordinates. This is the field used with one group of experimental subjects in [28]; Fig. 3 is in fact quite similar to the measured human behavior on initial exposure to this force field, as well as to the output of Shadmehr and Mussa-Ivaldi's own simulation model. (Recall from Sect. 2 that, even without adaptation, the proposed control law differs in several ways from (2) proposed in [28].)

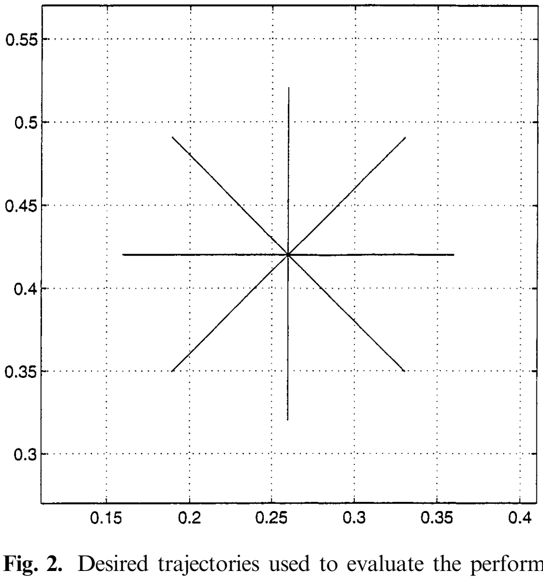


Fig. 2. Desired trajectories used to evaluate the performance of the proposed control law

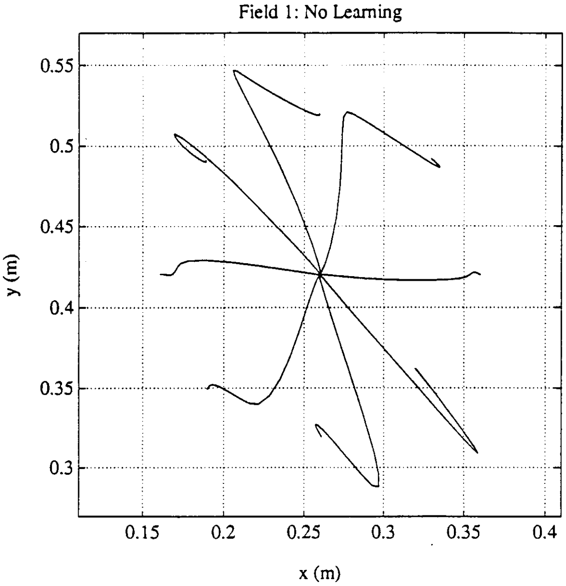


Fig. 3. Tracking of the desired trajectory upon initial exposure to the environmental force pattern given by (17)

Having established the baseline controller performance, both without external forces and in the field described by (17), a series of 250 reaching motions were simulated in the presence of the field (17), using the psuedorandom procedure described above. The controller's performance on the star pattern was then evaluated, followed by another set of 250 reaching motions, and so on. In this fashion, a total of 1000 reaching motions were simulated, giving the controller an opportunity to build up a model of the forces, **E**, and four equally spaced 'snapshots' were generated of the evolving controller performance on the canonical star pattern.

The results of this simulation are summarized in Fig. 4, which bears a noticeable resemblance to the comparable plots of human performance reported in [28] and shown for comparison in Fig. 5. In particular, the orientation, length, and rate of diminution of the 'hooks' in the deviations from the desired trajectory agree well with the human data. It is evident from this figure that the controller is gradually learning to counteract the influence of the applied force field so as to regain the baseline tracking of the desired trajectories shown in Fig. 2.

4.4 'Aftereffects' of adaptation

As pointed out in, [28] it is possible that the baseline performance is being recovered, not by developing an internal model of the new forces, but rather by making the linear parts of the controller more robust, for example 'stiffening' the joints by increasing K_D and Λ. Indeed, the bound (15) above displays this possibility explicitly. The network contribution to the control law

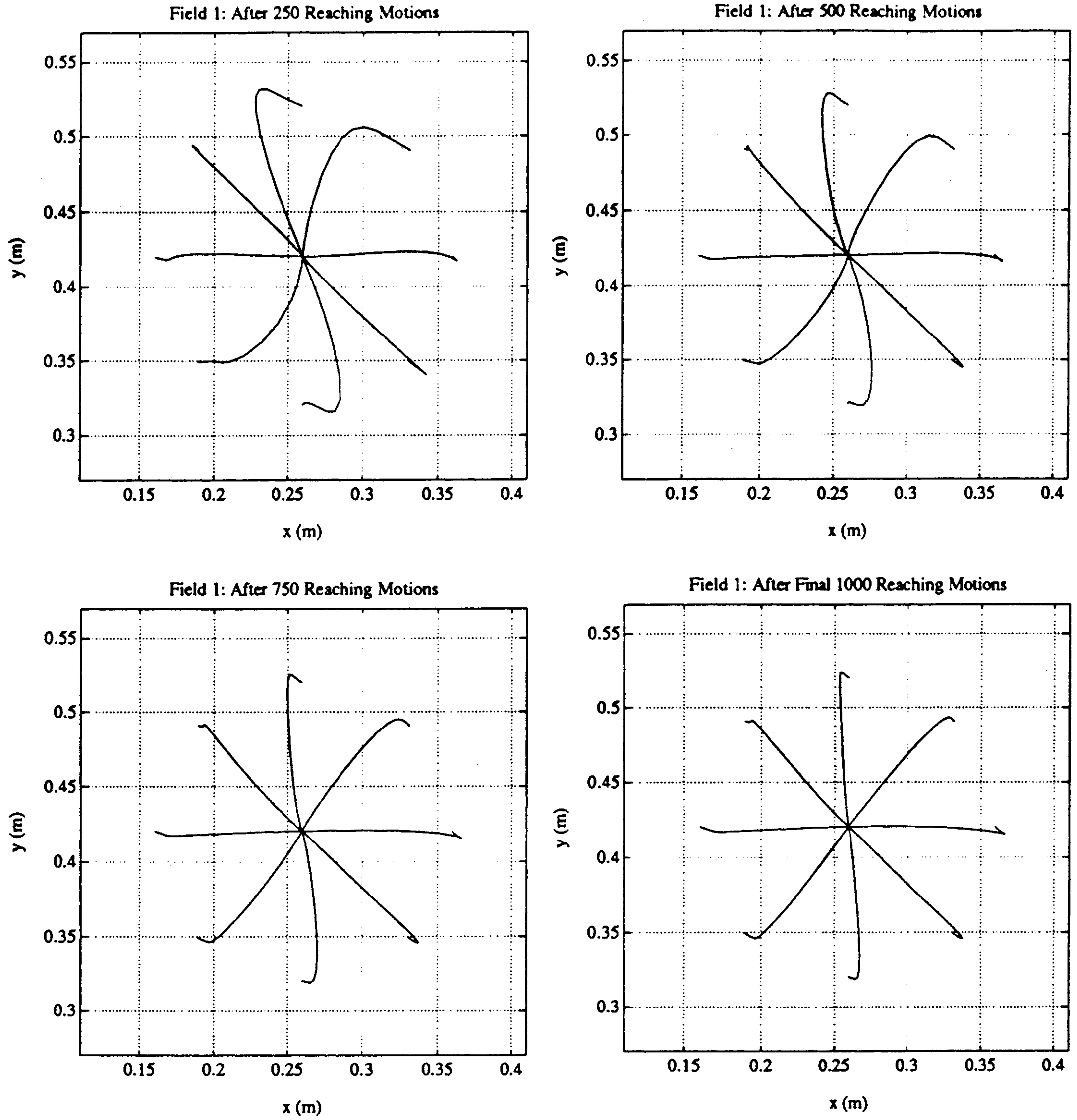


Fig. 4. Evolution of the performance of the adaptive control algorithm as a function of training time. After attempting to track 1000 pseudorandom motions throughout the workspace, perfect tracking of the desired trajectories of Fig. 2 is nearly completely recovered. Compare with the measured human performance on the same task in shown Fig. 5

might thus simply augment the linear feedback terms. To resolve this issue, at the same time each of the above 'snapshots' was taken, a second snapshot was generated, again evaluating the performance of the controller on the star pattern, but here with no environmental forces applied, i.e., with $\mathbf{E} = 0$.

If the controller is simply learning to increase the linear feedback, its performance with the field 'off' should exactly resemble the baseline performance of Fig. 2, since the magnitudes of the feedback gains will not affect the perfect tracking observed in this situation. If, on the other hand, the controller is developing an internal model of **E**, and using this model to modify the torques it commands, then when the environmental forces suddenly vanish, there should be substantial deviations from the baseline performance, since the controller will be generating torques to counteract a field which is no longer present. Indeed, these deviations should increase as a function of learning time, eventually resembling 'mirror images' of the deviations seen in Fig. 4. These deviations away from the baseline performance under the nominal (no field) operating conditions have been termed the 'aftereffects' of adaptation by [28].

Figure 6 shows that, indeed, the controller exhibits significant aftereffects, and that the magnitude of these deviations increases steadily, eventually resembling a 'mirror image' of the trajectory deviations seen in Fig. 4. The adaptive networks are thus indeed being used to model and offset the force field. There is generally good qualitative agreement with the comparable plots of the aftereffects recorded in human subjects, although on some of the legs, notably those at 90°, 135° and 315°, the deviations are less severe than observed in the human data. Otherwise, the orientation, magnitude, and rate of growth of the trajectory deviations agree with those observed in humans.

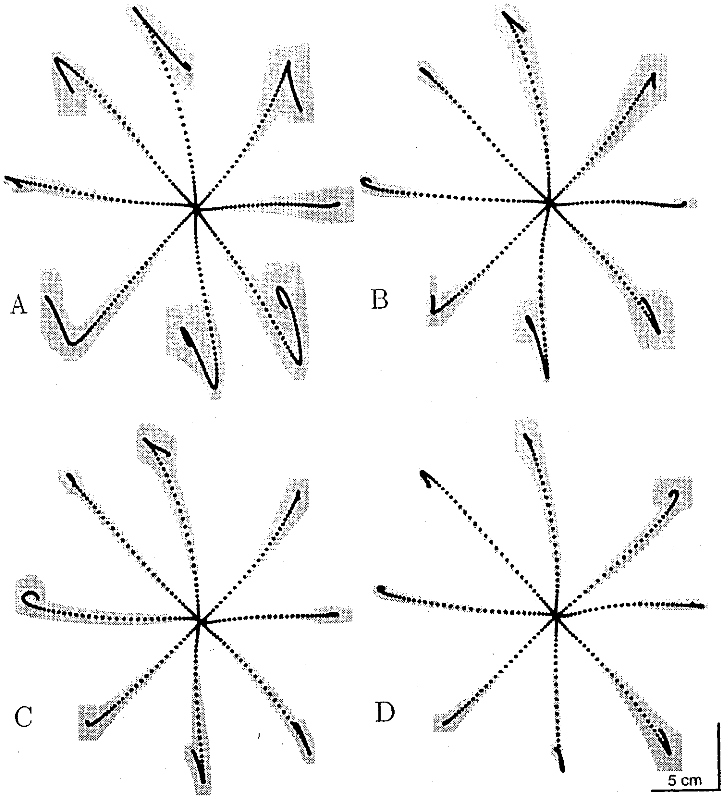


Fig. 5. Human performance on the same simulated task, reprinted from [28]

4.5 Generalization and persistency of excitation

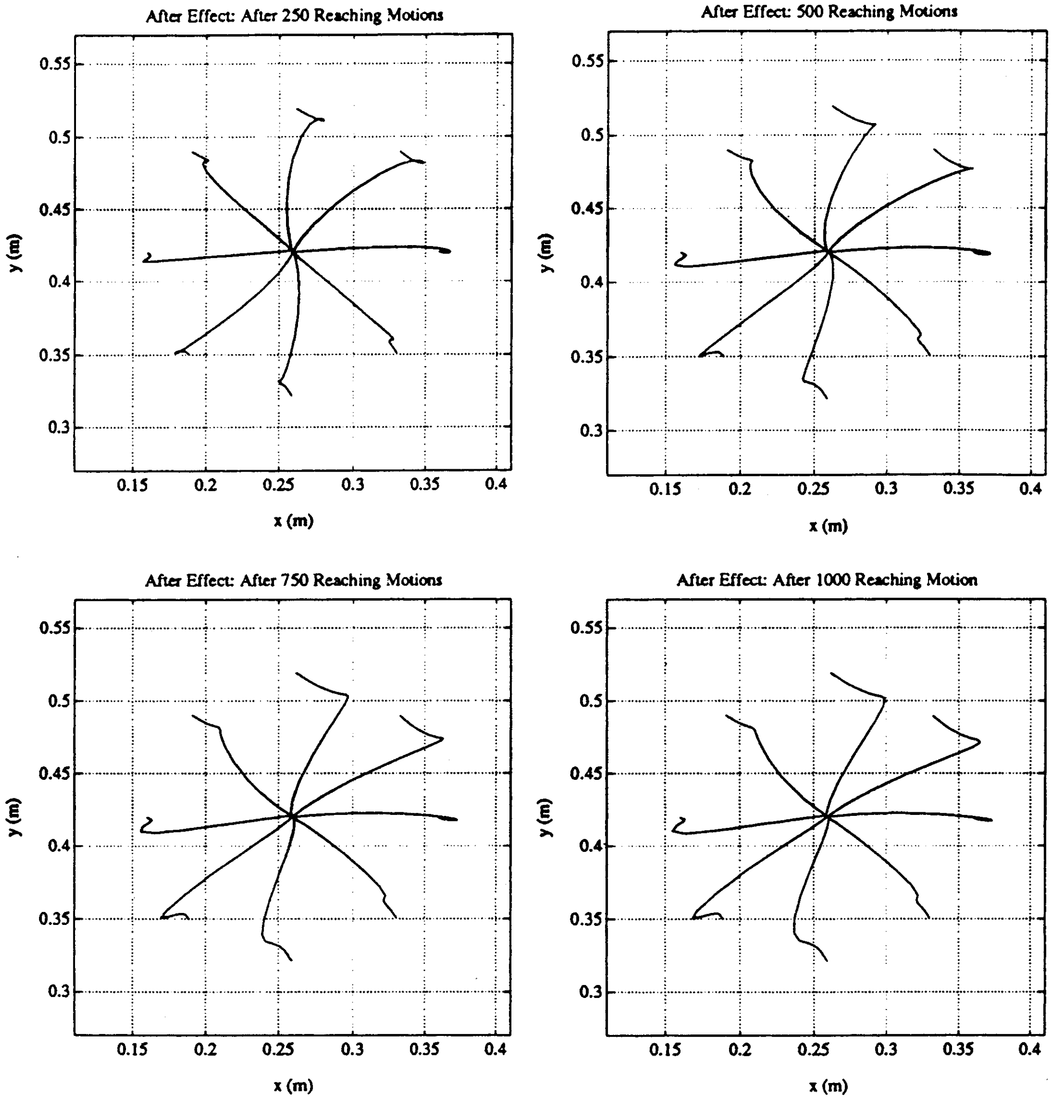
It is important also to evaluate the extent to which the model learned by the network can *generalize* to novel regions of the state space. In the human experiments reported in [28], the test subjects clearly showed that learning in the original 15 by 15 cm workspace influenced the performance of identical reaching tasks conducted in a different workspace. This observation led Shadmehr and Mussa-Ivaldi to conclude that the motor computational elements were 'broadly tuned' across the state space: That is, the observed adaptation was not an extremely localized, 'look-up table' phenomenon, but rather utilized elements φ_k which contribute significantly over a large range of joint angles and velocities. This observation was incorporated into the construction of the simulation, by appropriately selecting the variance of the Gaussians used in the adaptive networks. The relatively small Gaussian scaling parameter used in the network ensures that changes to the weights $\hat{c}_{i,j,k}$ will produce effects in the control law far outside the original workspace. Figure 7 illustrates this generalization by showing that aftereffects are observed on motions performed far outside the workspace used for training.

An interesting and well-known feature of direct adaptive control systems is that such devices will build only as complete an internal model as is sufficient to accurately track the commanded desired trajectories. In the experiments simulated above, there is thus no guarantee that the internal model which permits recovery of the baseline performance in a given workspace will coincide with the actual structure of the environmental forces. In fact, note that the actual field (17) does not deviate substantially during the experiment (and simulation) from the linearization

$$\mathbf{E}_L(q,\dot{q}) = J^T(q_0)BJ(q_0)\dot{q} \tag{18}$$

where q_0 are the joint angles which place the hand in the center of the original workspace. This would suggest that even though the computational elements are broadly tuned, the controller will not correctly generalize its learning to a new workspace, but rather will use an internal model more similar to the linearized field seen in the original workspace.

Indeed, after performing the 1000 reaching motions described above, Fig. 8 shows the performance of the controller, operating in the same force field (17), but tracking a star pattern centered instead at (−0.114 m, 0.48 m) relative to the shoulder. Compare this with Fig. 9, which shows the performance of the same controller, again tracking the relocated star pattern, but operating instead in the linearized field (18). These plots reveal that, while the controller has clearly generalized its learning, it has not developed a complete model of the force field (17). While neither plot displays excellent tracking of the star pattern in the new workspace, the tracking in the linearized field, shown in Fig. 9, is qualitatively closer to the tracking ultimately achieved in the original workspace (the last 'snapshot' in Fig. 5). This suggests that during its training, the controller developed only a locally accurate model of the actual field, more similar to $\mathbf{E}_L$ than to $\mathbf{E}$. When the controller



After Effect: After 250 Reaching Motions

After Effect: 500 Reaching Motions

After Effect: After 750 Reaching Motions

After Effect: After 1000 Reaching Motion

Fig. 6. Evolution of the "aftereffects" of adaptation as a function of training time. After attempting to track 1000 pseudorandom motions throughout the workspace, the trajectory perturbations produced by the aftereffects resemble a mirror image of the perturbations seen in Fig. 4. Compare with the measured human performance on the same task reported in [28]

attempts to use this model in the new workspace, where the true field is quite different from $\mathbf{E}_L$, the poor tracking performance observed in Fig. 8 results. Figures 7 through 9 are again similar to the corresponding plots of human performance reported in [28], although legs, notably those at 0°, 45°, and 224°, the deviations observed in the human data are significantly worse than those in the simulation data.

Continued practice in the new workspace would cause convergence to the new desired trajectories, recovering again the baseline behavior. In general, however, if all the required motions can be accurately followed without a completely accurate model, there is no pressure for the system to improve its controller. By instead choosing appropriately 'exciting' desired motions, so that perfect tracking would require a perfect model, the internal model of the environmental forces can be made to asymptotically converge to the actual force structure. The *persistency of excitation* conditions, which mathematically define the required trajectories, are reviewed in [30] in a robotic content, while in [32] the structure of these conditions for applications employing Gaussian networks is discussed.

4.6 Additional observations

The results above were obtained with no special consideration given to the specific network elements used, save to ensure that they were broadly tuned, and that the collection had a good approximating power for a large class of functions. Since it is very unlikely that actual motor computational elements have a precisely Gaussian structure, the qualitative agreement of the simulation results with the observed human performance illustrates the relative insensitivity of the proposed model to the exact structure of the elementary functions employed. The specific values for the

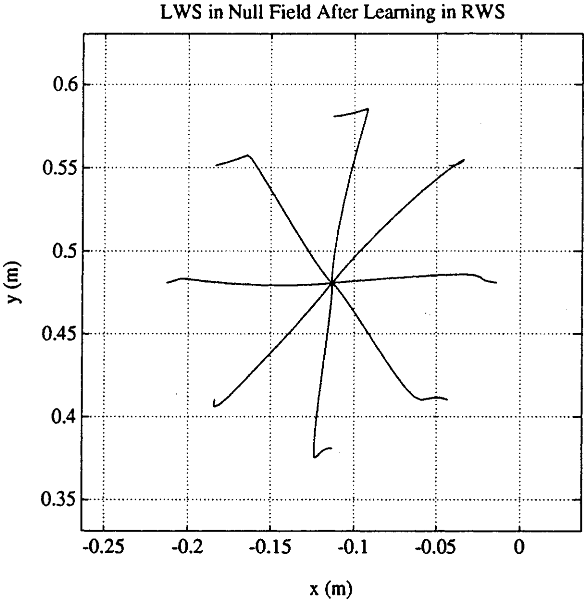


Fig. 7. Tracking of a star pattern using a null field in a workspace centered at (−0.114 m, 0.48 m) relative to the shoulder, after training in the field (17) in the original workspace. The presence of aftereffects in the new workspace indicates that the algorithm has generalized its learning

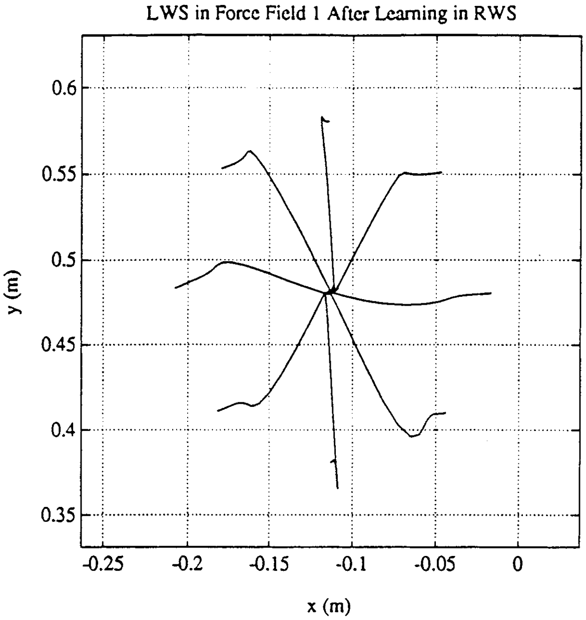


Fig. 8. Tracking of a star pattern centered in the new workspace after training in the original workspace. The motion is still perturbed by the environmental force field (17)

adaptation gains, however, were chosen by trial and error to yield incremental performance improvement qualitatively similar to the reported human performance. Recall that the learning rates are free parameters in the stable adaptation mechanisms (13), (14); any positive values will result in a stable, convergent algorithm. The specific choices which would match the simulated learning rate to that observed in humans, however, could not be predicted *a priori*.

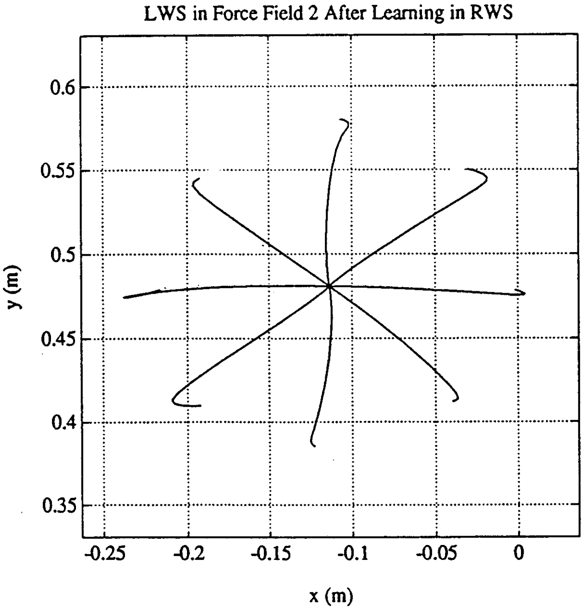


Fig. 9. Tracking of a star pattern centered in a different workspace than that used for training. The perturbing field is now given by (18), even though (17) was used in the training

Similarly, while the asymptotic recovery of the desired trajectory depends mathematically only on the approximation power of the aggregate collection $\{g_k\}$, the manner in which the network generalizes its learning to a new workspace is much more sensitive to the specific choice of basis function. A very large choice of h in the Gaussian network above, for example, might produce comparable results on the learning task, but exhibit virtually no generalization in the new workspace, since such a network exhibits quite local learning. Different choices of the 'shape' of g (for example, sigmoidal as opposed to Gaussian) would similarly produce different generalization properties. No attempt was made in this study to tune the choice of basis functions to better match the generalization observed in humans; this will be a topic of future investigation.

Finally, note that while the simulated experiments described above required only learning the unknown **E**, the control and adaptation laws used are capable of accommodating much more complex changes in the dynamics. For example, if the arm were to suddenly grab a massive, oddly shaped object, such as a tennis racket or bowling ball, the matrices H and C would suddenly change, requiring comparable modifications to the nonlinear components of the control law to ensure continued tracking accuracy. Similar changes occur to these components of the dynamics over longer time periods, as skeletal structure and musculature change. The proposed algorithm naturally has the flexibility to adapt to these more general changes in the dynamics.

5 Concluding remarks

In this paper, we have attempted to illustrate the strong similarity between models of adaptive motor control suggested by recent experiments with human and animal subjects, and the structure of new robotic control laws derived mathematically. In both models, the nonlinear component of the torques required to track a specified reference trajectory is assembled from a collection of very simple, elementary functions. By adaptively recombining these functions, the controllers can develop internal models of their own dynamics and of any externally applied forces, and use these adaptive models to compute the required torques. Biologically, the elementary functions represent abstractions of the actions of individual muscles and their neural control circuitry. Mathematically, however, the elementary functions can be any collection of basis elements which permit accurate reconstruction of continuous functions, such as those comprising current 'neural' network models.

Instead of iterative training methods, we have proposed a continuously adaptive model which has a strong Hebbian flavor. By using the adaptive elements in a method which fully exploits the underlying passive mechanical properties of arm motions, the resulting strategy of simultaneous learning and control can be guaranteed to produce stable, convergent operation. This continuous model has enabled not only reproduction of many of the end-results of the particular motor learning task examined, but also captures a significant component of the actual time evolution of the adaptation observed in human subjects.

The insensitivity of the proposed algorithm to the specific choice of basis functions is quite encouraging. The actual structure of the computational elements underlying human motor control may not resemble any of the biological computation models currently under investigation, including those employed herein. Since the performance of the model does not depend upon a specific choice of computational element, only upon the properties of the aggregate, the adaptive control model described above may capture some of the interesting features of actual low-level motor adaptation. Indeed, the underlying idea – continuously patching together complex control strategies from a collection of simple elements – is not only biologically plausible, it represents a sound engineering solution to the problem of learning in unstructured environments.

References

1. Arimoto S, Kawamura S, Miyazaki F (1984) Bettering operation of robots by learning. J Rob Sys **1**:123–140
2. Atkeson CG, Reinkensmeyer DJ (1990) Using associative content- addressable memories to control robots in: Miller TW, Sutton RS, Werbos PJ (eds) Neural networks for control MIT Press, Cambridge, Mass.
3. Bizzi E, Mussa-Ivaldi F, Giszter S (1991) Computations underlying the execution of movement: a novel biological perspective. Science **253**:287–291
4. Craig JJ (1986) Introduction to robotics: mechanics and control Addison-Wesely, Reading, Mass.
5. Cybenko G (1989) Approximations by superposition of a sigmoidal function. Math Cont Sig Sys **2**:303–314
6. Flash T, Hogan N (1985) The coordination of arm movements: an experimentally confirmed mathematical model. J Neurosci **5**:1688–1703
7. Girosi F, Poggio T (1990) Networks and the best approximation property. Biol Cybern **63**:169–176
8. Giszter S, Mussa-Ivaldi F, Bizzi E (1993) Convergent force fields organized in the frog's spinal cord. J Neurosci **13**:467–491
9. Gomi H, Kawato M (1993) Neural-network control for a closed-loop system using feedback-error-learning. Neural Networks 6:933–946
10. Hebb DO (1948) The organization of behavior. Wiley, New York
11. Hogan N (1984) An organizing principle for a class of voluntary movements. J Neurosci **4**:2745–2754
12. Hornik K, Stinchcombe M, White H (1989) Multilayer feedforward networks are universal approximators. Neural Networks **2**:359–366
13. Jordan MI (1980) Learning inverse mappings using forward models. Proc. 6th Yale Workshop on Adaptive and Learning Systems pp 146–151
14. Jordan MI, Rumelhart DE (1992) Forward models: supervised learning with a distal teacher. Cogn Sci **16**:307–354
15. Kawato M (1989) Adaptation and learning in control of voluntary movement by the central nervous system. Adv Rob **3**:229–249
16. Kawato M, Furukawa K, Suzuki F (1987) A hierarchical neural-network model for control and learning of voluntary movement. Biol Cybern **57**:169–185
17. Miller WT, Glanz FH, Kraft LG (1987) Application of a general learning algorithm to the control of robotic manipulators. Int J Rob Res **6**:84–98
18. Mussa-Ivaldi F, Giszter S (1992) Vector field approximation: a computational paradigm for motor control and learning. Biol Cybern **67**:491–500
19. Mussa-Ivaldi F, Hogan N, Bizzi E (1985) Neural , mechanical, and geometric factors subserving arm posture in humans. J Neurosci **5**:2732–2743
20. Mussa-Ivaldi F, Giszter S, Bizzi E (1994) Linear combinations of primitives in vertebrate motor control. Proc Nat Acad Sci **91**:7534–7538
21. Poggio T, Girosi F (1990) Networks for approximation and learning. Proc IEEE **78**:1481–1497
22. Powell MJD (1992) The theory of radial basis function approximation in 1990. In: Light WA (ed) Advances in numerical analysis, Vol II. Wavelets, subdivision algorithms, and radial basis functions. Oxford University Press, Oxford, pp 105–210
23. Sadegh N, Horowitz R (1990) Stability and robustness analysis of a class of adaptive controllers for robotic manipulators. Int J Rob Res **9**:74–92
24. Sanger T (1994) Neural network learning control of robot manipulators using gradually increasing task difficulty. IEEE Trans Rob Aut 10:323–333
25. Sanner RM (1993) Stable adaptive control and recursive identification of nonlinear systems using radial Gaussian networks. PhD Thesis, MIT Department of Aeronautics and Astronautics
26. Sanner RM, Slotine J-JE (1997) Gaussian networks for direct adaptive control. IEEE Trans Neural Networks **3**:837–863
27. Sanner RM, Slotine J-JE (1995) Stable adaptive control of robot manipulators using "neural" networks. Neural Comput 7:753–788
28. Shadmehr R, Mussa-Ivaldi F (1994) Adaptive representation of dynamics during learning of a motor task. J Neurosci **14**:3208–3224
29. Shadmehr R, Mussa-Ivaldi FA, Bizzi E (1993) Postural force fields of the human arm and their role in generating multi-joint movements. J Neurosci **13**:43–62
30. Slotine J-JE, Li W (1987) On the adaptive control of robotic manipulators. Int J Rob Res **6**:3

31. Slotine J-JE, Li W (1991) Applied nonlinear control. Prentice-Hall, Englewood Cliffs, NJ

32. Slotine J-JE, Sanner RM (1993) Neural networks for adaptive control and recursive identification: a theoretical framework. In: Trentelman HL, Willems JC (eds) Essays on control: perspectives in the theory and its applications. Birkhauser, Boston

33. Wiener N (1961) Cybernetics: or control and communication in the animal and the machine, 2nd edn. MIT Press, Cambridge, Mass.