Biol. Cybern. 77, 173–183 (1997)

Biological Cybernetics

A model for learning human reaching movements

Amir Karniel, Gideon F. Inbar

Department of Electrical Engineering, Technion-IIT, Haifa 32000, Israel

Received: 21 May 1996 / Accepted in revised form: 10 June 1997

Abstract. Reaching movement is a fast movement towards a given target. The main characteristics of such a movement are straight path and a bell-shaped speed profile. In this work a mathematical model for the control of the human arm during ballistic reaching movements is presented. The model of the arm contains a 2 degrees of freedom planar manipulator, and a Hill-type, non-linear mechanical model of six muscles. The arm model is taken from the literature with minor changes. The nervous system is modeled as an adjustable pattern generator that creates the control signals to the muscles. The control signals in this model are rectangular pulses activated at various amplitudes and timings, that are determined according to the given target. These amplitudes and timings are the parameters that should be related to each target and initial conditions in the workspace. The model of the nervous system consists of an artificial neural net that maps any given target to the parameter space of the pattern generator. In order to train this net, the nervous system model includes a sensitivity model that transforms the error from the arm end-point coordinates to the parameter coordinates. The error is assessed only at the termination of the movement from knowledge of the results.

The role of the non-linearity in the muscle model and the performance of the learning scheme are analysed, illustrated in simulations and discussed. The results of the present study demonstrate the central nervous system's (CNS) ability to generate typical reaching movements with a simple feedforward controller that controls only the timing and amplitude of rectangular excitation pulses to the muscles and adjusts these parameters based on knowledge of the results. In this scheme, which is based on the adjustment of only a few parameters instead of the whole trajectory, the dimension of the control problem is reduced significantly. It is shown that the non-linear properties of the muscles are essential to achieve this simple control. This conclusion agrees with the general concept that motor control is the result of an interaction between the nervous system and the musculoskeletal dynamics.

Correspondence to: A. Karniel (Fax: +972 4 8323041, e-mail: karniel@tx.technion.ac.il)

1 Introduction

Reaching movement is a basic motor action. It is a simple action, yet it involves almost all aspects of motor control, from vision and proprioceptors, through many parts of the nervous system, to the muscles and the joints. Reaching movement is relatively easy to monitor by measuring speed, force and electromyography (EMG), yet it does not have an accepted comprehensive model. These features make reaching movement an attractive action to study. (See, for example, Jeannerod and Prablanc 1983; Gottlieb et al. 1989, 1995; Ghez et al. 1990; Kalaska and Crammond 1992; Jordan et al. 1994; Dominey et al 1995; Kalaska 1995; Berthier 1996; and many other studies in the references within these works.)

In this work the control of fast and short duration movements is modeled. The term 'reaching movement' is used here for a fast, ballistic, voluntary movement of the arm from a starting point to a given target. Fast movement implies a duration of a few hundred milliseconds. When such duration is considered, visual feedback cannot be operational during the movement, since visual interpretation lasts at least a hundred milliseconds. Even proprioception feedback cannot be effective in a simple proportional integral derivative (PID) control fashion, because the delays are on the order of 50 ms and because feedback gains are low in biological motor control systems (Inbar 1972; Hollerbach 1982). The proper control scheme for such a movement is therefore a feed-forward control. This is the reason such a movement is called a ballistic movement (see Desmedt 1983 for examples and definition of ballistic movements in various muscles).

Two salient features of reaching movements are that the hand paths are roughly straight, and the hand speed profiles are bell-shaped (Abend et al. 1982). Flash and Hogan (1985) proposed the minimum jerk model that explains these features as a result of optimization criteria to minimize acceleration change. This work supports the suggestion that these features are the result of the arm dynamics, and not of neural optimization (Massone and Myers 1996 and references therein).

The kinematics and dynamics of the anthropomorphic arm are hard to model and control because they are non-linear and time variant. Non-linearity already exists in the kinematics and dynamics of the 2 degrees of freedom (2DOF) manipulator and also in the muscle dynamics. In addition, the system exhibits time variance in the muscle parameters, especially due to fatigue. For fast movements it is reasonable to ignore fatigue during the movement, and assume a time-invariant model. But the non-linear properties of the muscles should not be ignored as they may be responsible for some of the observed features of movements and for a simple control scheme as demonstrated and analysed in the present work.

Many schemes of motor control include a model of the arm. Inverse model control can provide a good control scheme as has been suggested and analysed a long time ago (Inbar and Yafe 1976). However, since there is no unique solution for reaching a given target (i.e. the arm function is not invertible) it is difficult to directly train an inverse model (see Jordan 1996 for an example of this problem). Several other schemes have been proposed. One is to create a controller and a forward model and to propagate the error through the forward model in order to train the controller (Jordan and Rumelhart 1992). This scheme was implemented for an anthropomorphic arm model by Jordan et al. (1994). Their learning and control scheme achieved typical movements. But this control scheme is complicated as it needs to generate a complete model of the plant and to propagate the error for each movement through the forward model and throughout movement time. Another scheme is feedback error learning (Kawato and Gomi 1992; Kawato et al. 1992). It suggests using a combination of feedback control and inverse model. The feedback motor command is used as an error in order to update the inverse model. This solution was implemented for reaching movements (Hirayama et al. 1993) using a cascade neural network that learns the neural input throughout the trajectory duration. This work also achieved typical reaching movements, but again with large computational and hardware costs. These works used a linear model of the muscles (Jordan et al. 1994) or just the dynamics of a two-joint manipulator with constant viscosity coefficient (Hirayama et al. 1993). It will be shown here that for a more biologically oriented, non-linear model of the muscle, the same typical movement can be learned by a much simpler control and learning scheme, without the need to propagate the error through time and to train a net to produce the entire trajectory.

A six-muscle model was used recently to create a model of the 2DOF arm (Jordan et al. 1994; Massone and Myers 1996). The muscle model is a simplification of the mechanical model in Zangemeister et al. (1981). The non-linear properties of the muscle were ignored for simplicity. In the present work, however, the muscle model includes the non-linearity based on Hill's model of the muscle (Hill 1938). For more information about muscle models, mechanical models and dynamic limb movements, see also Hannaford and Winters (1990) and Seif-Naraghi and Winters (1990).

The neuronal excitation to the muscle can be described as rectangular pulses to the agonist and antagonist (Gottlieb et al. 1989; Gottlieb 1993). This is a useful simplification used by Gottlieb to describe motor programmes and motor strategies of controlling a single joint movement where the relations between the widths, heights and time delay of these pulses and the movement are outlined. This simplification is adopted here, and therefore rectangular pulses are used as excitations to the muscles.

In the model presented here, an algorithm to learn the proper parameters of activation for each movement is proposed. This algorithm is analysed analytically for a linear plant control and is further being simulated for single- and for two-joint movements. Since the reaching movement is a ballistic movement, only the final state of the arm is used for training and adaptation. This idea, to mask the sensory information during the movement and use it only at the end of the movement, has a physiological basis in the cerebellorubrospinal system, as described in a review by Keifer and Houk: 'There is evidence for an inhibitory gating phenomenon that serves to suppress sensory input at several stages in the circuit during the performance of a limb movement. Thus it is reasonable to conclude that the motor commands are not generated by continuous sensory feedback from the periphery. More likely, the cerebellorubral system operates generally in an open-loop feed-forward manner' (Keifer and Houk 1994).

In this work the Hill-type non-linear muscle model, rectangular excitation pulses, feedforward control and learning from knowledge of results with an artificial neural network are incorporated to produce a simple model for learning human reaching movements. It is important to clarify that when the global term central nervous system (CNS) is used in this work, it refers to the cerebellorubrospinal system. The physiological counterpart of each component of the model is referred to in general terms only. Lack of certainty of the basic functional principles of nervous system operation prohibits at present any accurate comparison between the model and the CNS.

The rest of this paper is organized as follows. In section 2, the model of the arm, the muscles and the CNS are described. In section 3, two parts of the model are analysed: the muscle non-linearity and the learning scheme. In section 4, simulations of reaching movement are presented. And finally, in section 5 the results are discussed and conclusions are drawn.

2 The Model

In this section the model of the joints, the muscles and the CNS is described. The model of the arm is a 2DOF manipulator with six muscles. This part of the model is taken from the literature with minor changes (Zangemeister et al. 1981; Asada and Slotin 1986; Winters and Stark 1987; Arnon 1990; Massone and Myers 1996). The CNS model contains a pattern generator (PG) that creates rectangular excitations to the muscles, an artificial neural network (ANN) that produces the proper parameters of excitation pulses to the PG given the target, and a learning algorithm which includes a sensitivity model. All the parts of this model are described in detail

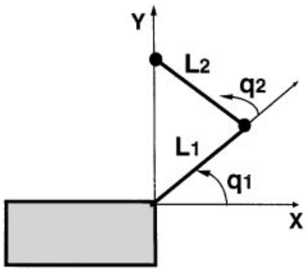

Fig. 1. A 2 degrees of freedom (2DOF) human arm model: $q1$ and $q2$ are the joint angles; $L1$, $L2$ are the lengths of the upper arm and the forearm; X, Y are the end-point coordinates and their base is the first joint , i.e. the shoulder

in order to make this model easy to understand and reconstruct for further research.

2.1 A two DOF manipulator

The arm is modeled as a 2DOF mechanical manipulator as shown in Fig. 1.

To calculate the position and velocity of the end point in (x, y) given the joint angles and angular velocities $(q1, q2)$ the direct kinematics equations are needed: (1) for position and (2) for velocity.

$$x = L_1 \cdot \cos(q_1) + L_2 \cdot \cos(q_1 + q_2)$$
$$y = L_1 \cdot \sin(q_1) + L_2 \cdot \sin(q_1 + q_2) \tag{1}$$

$$V = J \cdot \dot{q};$$
$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} -L_1 \cdot S_1 - L_2 \cdot S_{12} & -L_2 \cdot S_{12} \\ L_1 \cdot C_1 + L_2 \cdot C_{12} & L_2 \cdot C_{12} \end{bmatrix} \cdot \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \tag{2}$$

$$C_i \equiv \cos(q_i) \quad S_i \equiv \sin(q_i)$$
$$C_{12} \equiv \cos(q_1 + q_2) \quad S_{12} \equiv \sin(q_1 + q_2)$$

Since the muscles generate forces in the joints, one needs to calculate the position and velocities of the joints given the moments in the joints. This problem is known as the direct dynamics problem. It is solved numerically using the inverse dynamic equation:

$$F = H \cdot (q) \cdot \ddot{q} + C(q,\dot{q}) \cdot \dot{q} + G \tag{3}$$

For the 2DOF planar manipulator there are no forces due to gravity. Therefore $G = 0$, and the matrices in (3) are:

$$H = \begin{bmatrix} I_1 + I_2 + M_1 \cdot r_1^2 + M_2 \cdot (L_1^2 + r_2^2 + 2 \cdot L_1 \cdot r_2 \cdot C_2) & I_2 + M_2 \cdot (r_2^2 + L_1 \cdot r_2 \cdot C_2) \\ I_2 + M_2 \cdot (r_2^2 + L_1 \cdot r_2 \cdot C_2) & I_2 + M_2 \cdot r_2^2 \end{bmatrix}$$
$$C = \begin{bmatrix} -M_2 \cdot L_1 \cdot r_2 \cdot S_2 \cdot \dot{q}_2 & -M_2 \cdot L_1 \cdot r_2 \cdot S_2 \cdot (\dot{q}_1 + \dot{q}_2) \\ M_2 \cdot L_1 \cdot r_2 \cdot S_2 \cdot \dot{q}_1 & 0 \end{bmatrix} \tag{4}$$

Assuming cylindric units, one can use the following inertia and centre of mass:

$$I_1 = \frac{M_1 \cdot L_1^2}{12}, \quad I_2 = \frac{M_2 \cdot L_2^2}{12}, \quad r_1 = \frac{L_1}{2}, \quad r_2 = \frac{L_2}{2} \tag{5}$$

and for a typical arm, the following dimensions are used (taken from Arnon 1990):

$$M_1 = 2.52 \text{ kg} \quad M_2 = 1.3 \text{ kg}$$
$$L_1 = 0.33 \text{ m} \quad L_2 = 0.32 \text{ m} \tag{6}$$

For more information about the development of dynamic and kinematic equations see, for example, Asada and Slotin (1986).

This 2DOF model gives us the end-point position and velocity from the joint forces. The muscle model equations that generate these forces are described next.

2.2 The muscles

The human arm has many muscles that act on a single joint. Some act across two joints simultaneously. The present model is a simplified six-muscle model that represents all the muscles of the 2DOF arm. In the model, there are two muscles that act on each individual joint and two muscles that act on both joints simultaneously. The torque at each joint is a weighted sum of the forces of the muscles that act on that joint. Following Massone and Myers (1996) one can write:

$$\tau_1 = F_{1-\text{flex}} - F_{1-\text{ext}} + a \cdot F_{3-\text{flex}} - a \cdot F_{3-\text{ext}}$$
$$\tau_2 = F_{2-\text{flex}} - F_{2-\text{ext}} + b \cdot F_{3-\text{flex}} - b \cdot F_{3-\text{ext}} \tag{7}$$

where τ_i is the torque at joint i and F_j is the torque created by a muscle of type j. Index 1 is for the shoulder, index 2 is for the elbow, and index 3 is for the muscles that act across both joints. The subscript ext stands for an extensor muscle and flex for a flexor. Coefficients a and b are chosen to be $a = 0.6$ and $b = 0.4$ as in Massone and Myers (1996).

For each muscle a commonly selected Hill-type nonlinear mechanical model is used (Zangemeister et al 1981; Winters and Stark 1987). The model is shown schematically in Fig. 2.

For each muscle, n_i represents the neural input. The value of n_i is normalized to the range [0, 1]. This neural input is smoothed with a first-order filter that represents the activation contraction coupling in the muscle. Its output is the normalized hypothetical force, F_0, which is converted to the hypothetical force, T_0, after scaling by the maximum force, F_{max}, of each muscle [see (9)]. The viscosity element, B, represents the relation between force and velocity from Hill's model:

$$B = \begin{cases} (a \cdot T_0)/(b + v) & v \geqslant 0 \\ a' \cdot T_0 & v < 0 \end{cases} \tag{8}$$

$$a = 1.25 \quad a' = 3 \quad b = 1 \quad v = -\dot{x}_0/m$$

The viscosity, B, was assumed, for simplicity, to be a constant by Jordan et al. (1994) and Massone and Myers (1996) in order to obtain a linear muscle model. In this

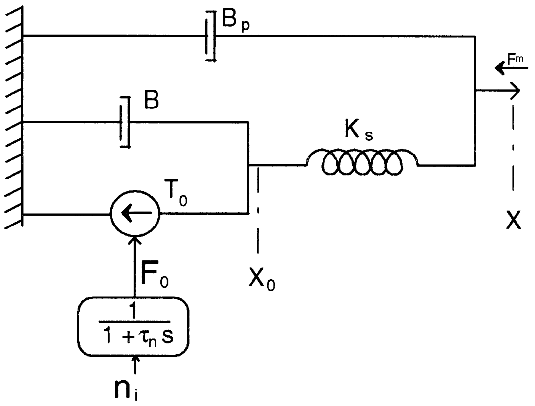

Fig. 2. The mechanical model of the muscle: n_i is the neural input, the first-order filter represents the activation contraction coupling, T_o is the hypothetical force in the muscle, B represents the relation between force and velocity from Hill's model, while the other elements represent the mechanical properties of the tendon and other connective tissues around the joint

work the full non-linear model is used. This property has turned out to provide the desired smooth performance with a simple control signal, as is described in the model analysis section. The other mechanical elements represent the muscles and connective tissue properties around the joint. The equations that describe this model are:

$$\dot{F}_0 = \frac{1}{\tau_n}\cdot(n_i - F_0)$$
$$T_0 = F_0 \cdot F_{\max}\cdot m$$
$$\dot{X}_0 = \frac{(K_s\cdot(X - X_0) - T_0)}{B} \tag{9}$$
$$F_m = B_p\cdot\dot{X} + K_s\cdot(X - X_0)$$

where m is the mean arm moment. The values of the parameters were as follows:

$B_p = 0.2$[N*s/rad], $K_s = 30$[N/rad], $\tau_n = 0.04$[s],

m = 0.03[m], $F_{\max}$(shoulder) = 800[N],

$F_{\max}$(elbow) = 700[N], $F_{\max}$(double joint) = 1000[N].

The values above were taken from Massone and Myers (1996) with minor changes: only one filter was used as the second one has a negligible time constant, and the parallel elastic element was eliminated because its value is small and has no important effect on fast movements. Its main effect is on the steady state, where it causes a drift toward the resting point of the muscle. In the present work only fast reaching movements are studied, and not maintenance of a fixed position. The major departure of the present model from Massone and Myers (1996) is in the non-linear muscle model which finds expression in the viscosity, B, according to Hill's model (8).

Another minor change is in the relationship between each muscle length and the joint angle. In the present model the arm is in the horizontal plane, and the initial point is when the shoulder is at 45 deg and the elbow is at 90 deg, as seen in Fig. 1. The relationships between each muscle length X and the joint angle q are:

$$\begin{aligned}
X_{\text{Shoulder_Flexor}} &= (5\cdot\pi/4 - q_1)\cdot m\\
X_{\text{Shoulder_Extensor}} &= (3\cdot\pi/4 + q_1)\cdot m\\
X_{\text{Elbow_Flexor}} &= (\pi - q_2)\cdot m\\
X_{\text{Elbow_Extensor}} &= q_2\cdot m\\
X_{\text{Double-Joint_Flexor}} &= (9\cdot\pi/4 - q_1 - q_2)\cdot m\\
X_{\text{Double-Joint_Extensor}} &= (3\cdot\pi/4 + q_1 + q_2)\cdot m
\end{aligned} \tag{10}$$

With the manipulator kinematics and dynamics given, the nervous system that controls the arm and produces the neural input to each muscle can be modeled.

2.3 The nervous system model

The motor neurons that activate the muscles obtain their inputs from the CNS and from the sensors in the arm (muscle spindles, Golgi organs, skin and joint sensors, etc.). Because of the delays and time constants of the muscle activation processes, and since the modeled reaching movements are very fast, there cannot be an efficient real-time feedback control. Consequently, in the proposed model, a pure forward control scheme will be considered. The sensory information will be used, at the end of the movement, for learning and adaptation only.

The model is based on the following simplifications and assumptions:

- The target location is given. This assumption is based on experiments that suggest that the information about the target location in the workspace is mapped in the cortex before the execution of the movement. See, for example, Georgopoulos et al. (1993).
- Feed-forward control scheme. The information about the performance is used only after the execution of the movement. This assumption is based on the short duration of the movement in relation to the delays which prohibit an efficient feedback control (Inbar 1972; Hollerbach 1982), and on the work of Keifer and Houk (1994) which suggests further evidence for the masking of sensory information during the movement.
- Typical rectangular excitations to the muscles. In the CNS there are mechanisms that can generate characteristic stimuli to the muscles, for example, the adjustable pattern generator (APG) in the work of Houk and Wise (1992). The typical shape of the muscle excitations are modeled to be rectangular, and their timing and amplitude are changed according to the desired movement (see Gottlieb 1993).

A schematic description of the CNS control model for reaching movement is given in Fig. 3. The target is chosen in the cortex and is used as the input. According to the target, a set of parameters which define the excitation to the muscles is selected. These parameters are amplitudes and timings. The APG generates the appropriate pulses to each muscle, and these commands go through the spinal cord to the muscles and generate the movement. At the end of the movement, and only then,

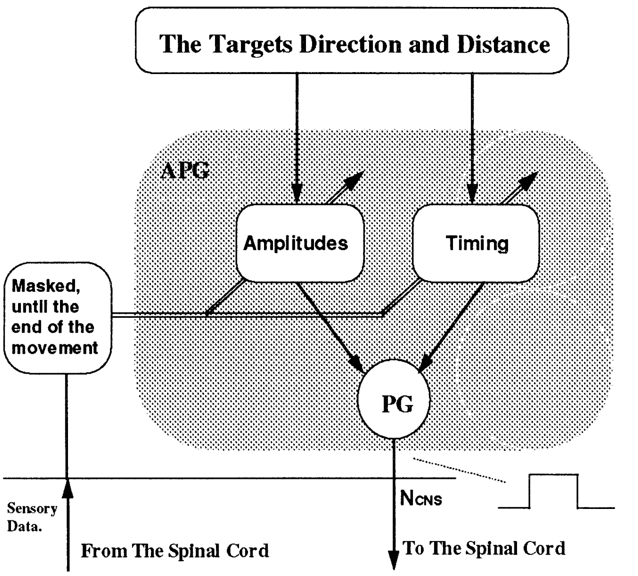

Fig. 3. The CNS model: N_{CNS} is the activation command to the muscles. There is a pattern generator (*PG*) for each muscle, that creates a rectangular excitation according to the timing and amplitude parameters which are the output of the artificial neural network (ANN). The input to the ANN is the desired target. The sensory information is available only at the end of the movement, and only then is the ANN updated

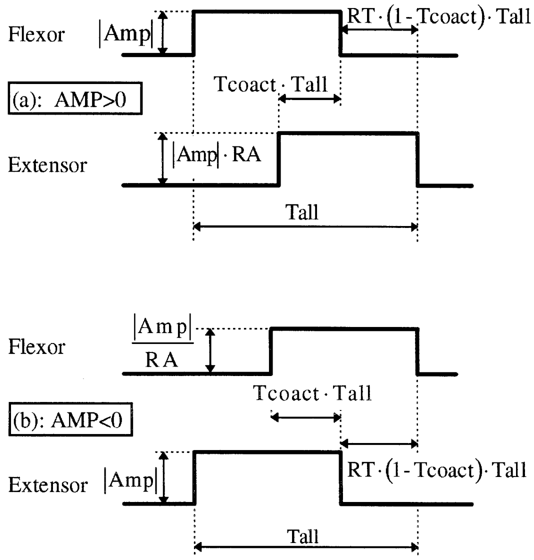

Fig. 4. The shape of the pulses and the parameters that define it. When AMP > 0 the flexor is activated before the extensor (**a**); when AMP < 0, the flexor follows the extensor (**b**)

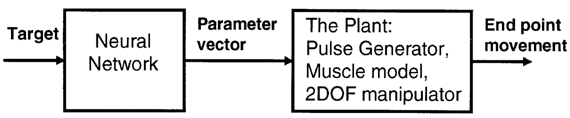

Fig. 5. The problem of finding the parameter vector, given the target. The problem is how to train the ANN in a way that it will produce the proper excitation parameters in order to bring the arm to the desired end-point state

the sensory information is analysed, and the error found is used to change the weights in the neural network that generates the parameters. These changes would affect the next neural control pulses and therefore only the subsequent movements.

In this framework the first step is to define the shape of the excitation and how it relates to the timing and amplitudes (which are called here 'the parameters'). In the next section a learning scheme is described that determines the relation between the desired target and the parameters.

2.3.1 The neuromuscular excitation and its parameters. The muscle excitation is controlled by the CNS. It has a typical shape, but its timing and amplitude change according to the desired movement. The choice of parameters is described in Fig. 4. Each pair of muscles (flexor and extensor) receives two pulses that come one after the other. A vector of five parameters defines these two pulses. The parameters are [Amp, RA, Tall, Tcoact, RT]. Amp is the amplitude of the activation in the range $[-1, 1]$. If Amp < 0, then the order of activation is reversed (see Fig. 4b). RA is the ratio of amplitudes between the flexor and the extensor. Tall is the total time of the excitation. Tcoact is the duration of the coactivation of both muscles. RT is the ratio of duration between muscle activations. The parameters were selected for their physiological interpretation.

2.4 The learning algorithm

Let us put the model of the arm, muscles and PG's described above in one box, and a neural network controller in another box, as shown in Fig. 5. The question is how to train the neural network to produce the proper parameters given the desired target.

To train the neural network, in most methods one needs the error in the coordinates of the neural network output. The problem is that the error is given in the coordinates of the plant output. The transformation between the plant output and the parameters is very complicated and not necessary to compute. What is needed is the relationship between the errors of the output and the errors of the parameters. These errors are small in the trained network. The simplest scheme, which is a linear model, was implemented, i.e. a linear model for the relation between changes in the plant output and changes in its input. This linear model is called here the sensitivity model. The linear relation is a simplifying assumption that holds only for small regions of the workspace. This drawback can be overcome as demonstrated and discussed in the simulation and discussion sections. It should be stressed that this system is discrete in its nature since the inputs are targets transformed to a parameter vector by the neural network. Only the final state is the output of the plant. This state is used to train the network for better performance in the next movement which is the next discrete event. A schematic description of the system is given in Fig. 6 for the general case. In this model, IN is a vector describing the desired state of the arm at the end

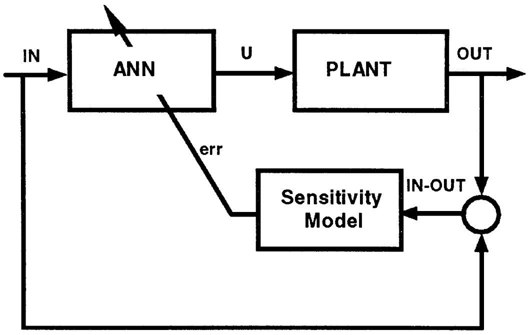

Fig. 6. Schematic description of the learning algorithm using the sensitivity model, which is a linear transformation from the output coordinates to the parameters coordinates

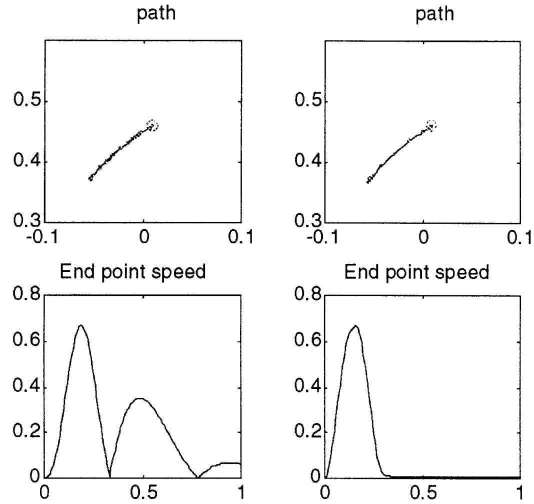

Fig. 7. A comparison between the speed profile of the end-point of the arm with a linear muscle model (*left*) and with a non-linear muscle model (*right*) in response to the typical rectangular pulse activation of the muscles. In the *upper figures* the initial end-point positions and the paths are shown. In the *lower figures* the end-point tangential speed profiles are shown. Only the non-linear muscle model yields a bell-shaped speed profile with a smooth stop. (The units are meters and seconds)

of the movement, U is the excitation parameters vector, and OUT is a vector describing the actual state of the arm end-point at the end of the movement.

The algorithm:

1. Initiate the ANN and the sensitivity model.
2. For a desirable target, IN, calculate U and OUT.
3. Make a small perturbation on U, calculate OUT and update the sensitivity model. (This stage can be done using previously performed close movements.) This stage can be skipped if the sensitivity model is satisfactory.
4. Transform the output error to the parameter error using the sensitivity model and calculate the desired network output, Ud:

$$\mathrm{Ud} = \mathrm{U} + \mathrm{C} \cdot \mathrm{err} = \mathrm{U} + \mathrm{C} \cdot \mathrm{Sen} \cdot (\mathrm{IN} - \mathrm{OUT}) \quad (11)$$

The constant, C, is in the range [0, 1], and it determines the learning rate.
5. Update the ANN weights to get U closer to Ud. This can be done using any learning algorithm such as back-propagation or Levenbarg-Markart.
6. Go to stage 2.

This algorithm is analysed and simulated in the next sections. Here and in the next section the specific properties of the ANN are not given since any ANN and learning algorithm that minimize some output error function will do. In the simulation section the exact specifications of the ANN used are given.

3 Model analysis

In this section two parts of the model are examined. The first part is the muscle model and its non-linearity and the role it may have in the performance of movements. The second part is the learning algorithm and its properties for some very simple cases.

3.1 The non-linear muscle model

The non-linear properties of muscle have been known for a long time (Hill 1938), but they were considered a problem for the modeller (at least the engineer modeller). In the present work the non-linear property was necessary in order to create a fast movement with a smooth stop using the simple rectangular excitation. The control of rectangular pulses is much simpler than the creation of a complex profile of excitations during movement. That is the reason why it is suggested here that the non-linearity in the muscle may play a functional role by allowing the use of such simple control signals. The engineering idea which separates the controller from the plant is not applicable to a biological system. The biological system evolved as a whole to create the best performance. So the muscle must be seen as part of the controller and not only as a force generator [for more examples of the importance of mechanical systems in motor control, see Full (1994)].

The performance of a linear model and a non-linear model in response to the rectangular control pulses is seen in Fig. 7. For the linear muscle model the viscosity is constant, $B = 4.5$[N*s/rad], instead of the Hill-type relation in (8).

It can be seen that the arm with the linear muscle model, does not stop in response to pulses, but has an overshoot and oscillatory behaviour at the end of the movement. Under the same conditions, the non-linear muscle can evoke a fast movement with a smooth stop. This is only a demonstration, but this example is representative of the improved arm performance achieved with a non-linear muscle model under the assumed conditions. The reason for this phenomenon can be explained by observing the behavior of a simple second-order system in its standard form. The transfer function in the Laplace transform domain is:

$$a \cdot \frac{w_n^2}{s^2 + 2 \cdot \xi \cdot w_n \cdot s + w_n^2} \quad (12)$$

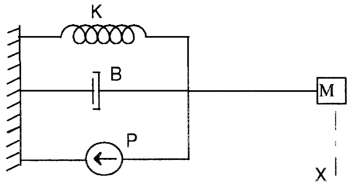

Fig. 8. Second-order mechanical model of the muscles and the joint. In this model type K is positively related to the activation P

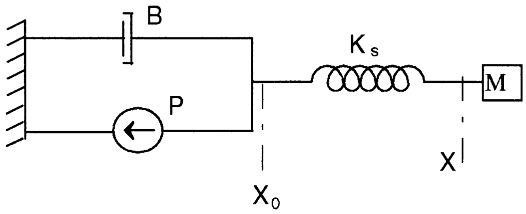

Fig. 9. Third-order mechanical model of the muscles and the joint. In this model type B is positively related to the activation P.

All the characteristics are known for such a system. w_n is the natural frequency, ξ is the damping coefficient, and a is the gain. Let us look at the overshoot (OS) and the time to reach the maximum (t_{max}):

$$t_{max} = \pi/(w_n \cdot \sqrt{1-\xi^2})$$

$$\mathrm{OS} = \mathrm{EXP}(-\pi \cdot \xi/\sqrt{1-\xi^2}) \quad (13)$$

For the human arm, the system is underdamped, i.e. $\xi < 1$ (see, for example, Inbar 1996). It can be seen from (13) that as ξ gets smaller, the movement speeds up but the overshoot increases. So in a linear system a trade-off exists between small overshoot and fast movement. In a non-linear system the parameter can change during the movement to achieve a fast movement without any overshoot, as seen in Fig. 7.

In order to examine the effect of the non-linearity, two commonly used types of muscle model were analysed. The conclusion, as shown below, is that the non-linearity, which is different in each type of model, has the same effect on the normalized second-order model parameters.

Let us look, first, at the mechanical second-order model in Fig. 8.

The transfer function of this model is (14) and the parameters are (15)

$$\frac{X}{P} = \frac{-1}{M \cdot s^2 + B \cdot s + k} \quad (14)$$

$$a = -\frac{1}{k} \qquad w_n = \sqrt{\frac{k}{M}} \qquad \xi = \frac{B}{\sqrt{4 \cdot M \cdot k}} \quad (15)$$

In trying to find the parameters of such a model, it was shown (Inbar 1996) that K changes during the movement and it has a positive relation to the activation of the muscle.

Now let us look at the following model (Fig. 9), which is the type used in the present paper:
Its transfer function is (16).

$$\frac{X}{P} = \frac{-k_s}{B \cdot M \cdot s^2 + k_s \cdot M \cdot s + B \cdot k_s} \cdot \frac{1}{s} \quad (16)$$

In our model the input, P, is a pulse, which brings X to a new location. This is equivalent to applying a step and multiplying the transfer function by S to get a second-order transfer function. The parameters of this function are (17).

$$a = -\frac{1}{B} \quad w_n = \sqrt{\frac{k_s}{M}} \quad \xi = \frac{\sqrt{M \cdot k_s}}{2 \cdot B} \quad (17)$$

For this model, B is changing according to Hill's model, and it has a positive relation to the activation of the muscle.

If one looks at the parameters of the second-order systems (15, 17) one can see that B, in the second model plays a similar role to K in the first model.

The important aspect for this work is that, for both kinds of modeling, at the end of the movement, when the excitations terminate, ξ goes up, and the arm can stop smoothly without overshoot.

3.2 The learning algorithm in some simple cases

For the simple case of the linear invertible plant, one can prove that the algorithm converges. The following theorem and its proof relate to the algorithm described in Sect. 2.4.

Theorem: Let the plant transformation matrix be P, the sensitivity transformation matrix *Sen*, and the desired ANN output U_d. If (i) the plant is linear and invertible, (ii) $Sen = P^{-1}$ and $C = 1$, and (iii) the ANN learning algorithm is such that at each step $|U - U_d|$ is getting smaller, then the whole algorithm converges asymptotically.

Proof: First let us define a target function to minimize:

$$Z \equiv \|\mathrm{IN} - \mathrm{OUT}\| \quad (18)$$

Looking at Fig. 6, one can write for each step:

$$U_d(i) = U(i) + C \cdot Sen \cdot (\mathrm{IN} - \mathrm{OUT}(i)) \quad (19)$$

and using the theorem assumption:

$$\begin{aligned} U_d(i) &= U(i) + 1 \cdot P^{-1} \cdot (\mathrm{IN} - \mathrm{OUT}(i)) \\ &= U(i) + P^{-1} \cdot \mathrm{IN} - P^{-1} \cdot \mathrm{OUT}(i) \\ &= U(i) + P^{-1} \cdot \mathrm{IN} - U(i) = P^{-1} \cdot \mathrm{IN} \end{aligned} \quad (20)$$

which means that U_d is constant for this case.

From (19) it can be written:

$$|U(i) - U_d| = |P^{-1} \cdot (\mathrm{IN} - \mathrm{OUT}(i))| \quad (21)$$

And since $|U - U_d|$ is assumed to be always getting smaller:

$$|U(i+1) - U_d| < |U(i) - U_d| \Rightarrow |P^{-1} \cdot (\mathrm{IN} - \mathrm{OUT}(i+1))| < |P^{-1} \cdot (\mathrm{IN} - \mathrm{OUT}(i))| \Rightarrow Z(i+1) < Z(i) \quad (22)$$

The target function Z is a monotonically descending function. From the global convergence theorem (see Luenberger 1984) it can be concluded that the algorithm converges asymptotically.

This was only to show that the above scheme is a reasonable procedure with the given assumptions. However, the human arm is a non-linear system that may not be invertible, and therefore the sensitivity model must be acquired by learning. The overall system model performance can be investigated only through simulation, which is carried out in the next section.

4 Simulation of reaching movements

The algorithm of learning and performing a reaching movement was simulated both for a single- and for a two-joint system. In each simulation, ten targets were presented to the model. The reaching movements were simulated sequentially. Each cycle of ten reaching movements and ten perturbed movements is considered an epoch. After each epoch the sensitivity model and the ANN are updated. The sensitivity model is updated to the optimal linear model, with respect to the mean square error (MSE), using the last 200 movements. The ANN is a one hidden layer neural network with 16 elements in the hidden layer. Each hidden element has a hyperbolic tangent sigmoid activation function, and each output unit is a linear function of the hidden units output. The ANN is updated using two epochs of the Levenberg-Markart algorithm. The Matlab implementation of those algorithms was used (see Demuth and Beale 94). For a comprehensive foundation of ANN, see Haykin (1994).

4.1 Single joint

In single-joint simulations the shoulder was fixed and the lower arm moved about the elbow joint. Two muscles of the elbow were activated, and two parameters, Amp and RA, were the control parameters. The rest of the parameters were fixed to the following values: Tall = 20, Tcoact = 0.4, RA = 0.5. These values were chosen heuristically, after some trial and error, to create a fast movement (i.e. total movement time of less than half a second) and to allow movements that cover the entire workspace. The errors were two: (i) signed distance from the target and (ii) speed at the end of the movement. The results of such a simulation are given in Fig. 10, where one can see that the arm reaches the target (in the circle) and the speed profile has a smooth bell shape. The error during the learning is shown in Fig. 11, as a function of the number of epochs. It is not expected that the error will converge to zero since the sensitivity model is linear, while the relation between the errors may not be.

The reason for using signed distance is that the relation between the distance error and the excitation amplitude is positive in one direction and negative in the other. In such a situation no linear sensitivity model can be adequate. The solution is to use a signed distance error (i.e. positive in one direction and negative in the other). This solution will be referred to also as a proper selection

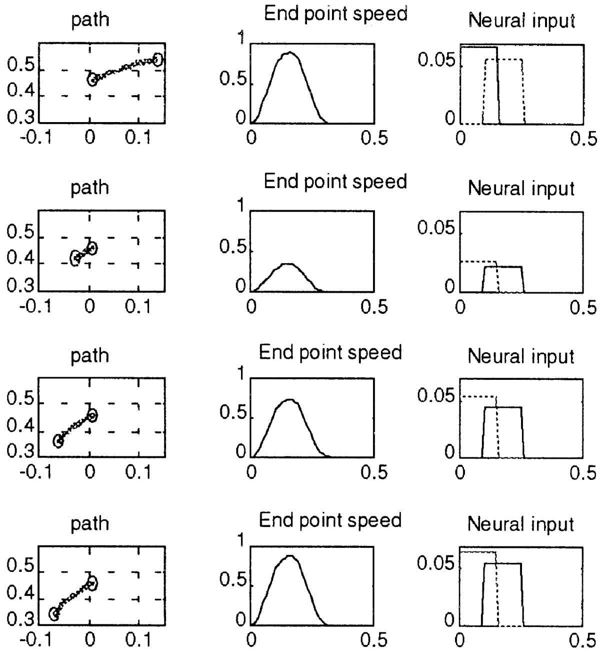

Fig. 10. Single-joint learning. Four examples were taken from the learning set. The paths of the end-points and the initial and final positions are at the *left*. The speed profiles are in the *middle*. The muscle excitation signals are at the *right*. (The units are meters and seconds)

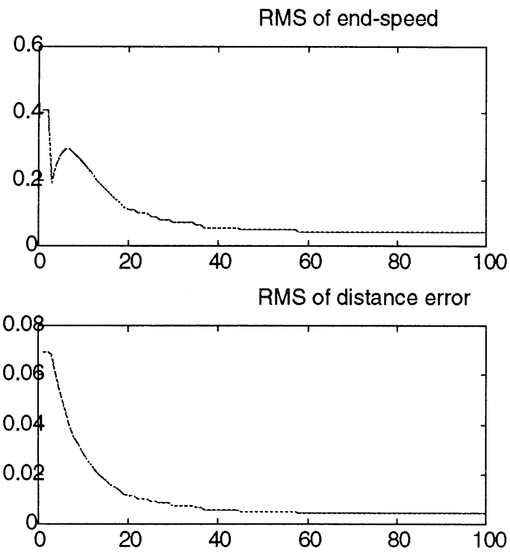

Fig. 11. Learning error of single-joint training. The errors are end speed (m/s) and signed distance error (m). The errors are not necessarily monotonically decreasing because the sensitivity model is not always correct

of parameters and error. Another possible solution is to use two different sensitivity models for each area. These solutions are considered again for the two-joint case.

4.2 Two joints

In two-joint simulations, four muscles were activated and four parameters, Amp and RA for each joint, were

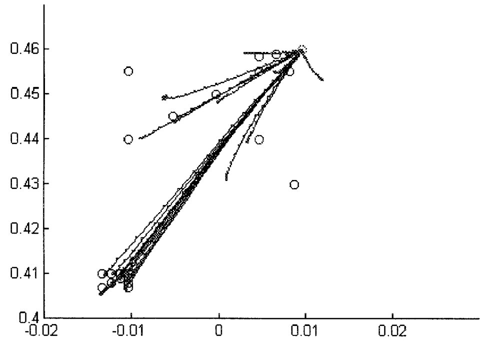

Fig. 12. Comparison of close targets vs spread targets. The close targets are easier to learn. In the *bottom left corner* are 10 close targets of the first simulation. The other 10 targets are spread and are related to the second simulation. (Units are meters)

the control parameters. The rest of the parameters were fixed to the following values: Tall = 20, Tcoact = 0.4, RA = 0.5. The errors were three: (i) distance from the target, (ii) direction error (difference between the desired and the actual direction), and (iii) the speed at the end of the movement.

Ten targets were spread over the workspace, and the algorithm was activated in order to learn to reach them. One can see in Figs. 12, 14 and 15 that the speed profile is smooth and approximately bell-shaped, and the path is close to a direct line.

In Fig. 12 the results of two simulations and two sets of targets are shown. The first simulation involved a set of 10 targets in a small area (about $x = -0.01$, $y = 0.41$). The other set of ten targets was spread over the workspace. One can see that for the close targets, the error is much smaller than for the spread targets.

There are two solutions for the spread targets. The first is to map the working area with separate sensitivity models in order to obtain a piecewise linear sensitivity model. The second solution is to define the parameters and error in a way that enlarges the area where the linear sensitivity model is appropriate. The second solution can be regarded as pre-processing or post-processing, before or after the sensitivity model. This approach was implemented by converting the direction error to two individual joint angle errors. This conversion is a kinematic conversion, and it requires knowledge of the geometrical dimensions of the arm. This transformation is justified since the nervous system has this knowledge. The geometrical dimensions are fixed under normal operating conditions. They may change very slowly during growth, and so they can be learned through another long-term mechanism. The results after the last modification are much better, and an example is shown in Figs. 13, 14, and 15. It can be seen that the movements are approximately straight lines and the speed profile is bell-shaped.

There is room for improvement in the performance of the proposed controller scheme since only the amplitudes were controlled and the other parameters were held fixed. Further investigation of the learning scheme is also possible. The size of the ANN and the number of hidden units were not optimized, learning algorithms were not compared, and generalization capabilities were not checked. The reason for not doing the above lay in the long computational time to simulate the model. It is important to mention here that most of the computational time was used in calculating the arm dynamics and

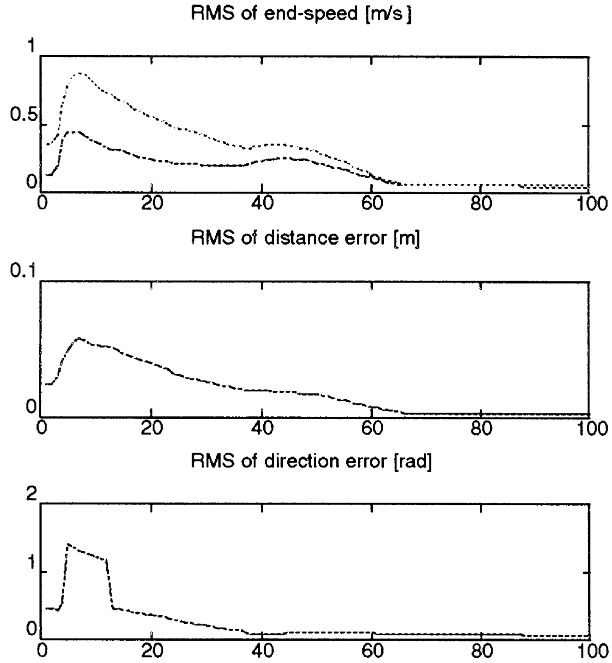

Fig. 13. Learning error of two-joint training (with kinematic pre-processing). The shown errors are: end speed error (for each joint), distance error and direction error

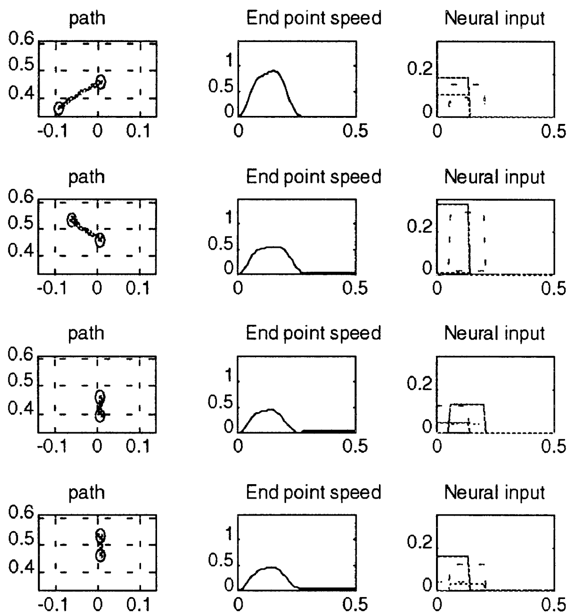

Fig. 14. Two-joint learning. Four examples from the learning set of the simulation with kinematic pre-processing. The paths of the end-points and the initial and final positions are at the *left*. The speed profiles are in the *middle*. The muscle excitations are at the *right*. (Units are meters and seconds)

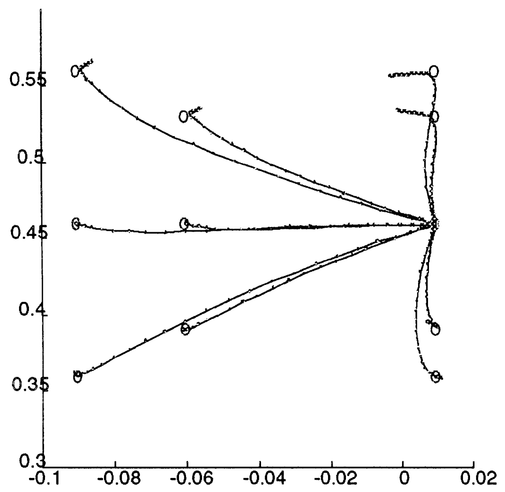

Fig. 15. Reaching movement for all 10 targets in the learning set. This is the result of the simulation with kinematic pre-processing. (Units are meters)

the non-linear muscle dynamics and not in the ANN control. These long calculations are needed only in simulation. In the biological system they are a by-product of the muscles and the arm structure. However, despite the shortcoming of the simplified controller, the presented results are sufficient to demonstrate that the proposed learning algorithm for the parameters of rectangular excitation pulses is adequate for controlling and performing typical reaching movements.

5 Discussion

5.1 The muscle model

The role of the muscle properties in creating a smooth bell-shaped speed profile was demonstrated. There are some weaknesses to the muscle model that should be noted. The mechanical model, which is a Hill-type model, is based on small signal isometric or isotonic conditions. The behaviour of the muscle parameters during movement is not clear yet, and thus the model should be regarded as an approximation only. A second drawback of the model is its being a time-invariant model. The muscle parameters are changing significantly under changing conditions (angular position, velocity, loads, etc.), and fatigue must be considered in cases of a long period of training.

The investigation of the model by studying the parameters of a second-order system (Fig. 9) as an approximation to the entire model is supported by a recent robustness study of the same arm model with a linear muscle model (Myers and Massone 1997). The relationships achieved in that work between the viscosity, B, and the time-to-peak and maximum velocity can be deduced from the parameters of the simplified second-order model (17). Another work that was published recently by Krylow and Rymer (1997) concerns the role of the muscle properties in producing smooth movements. The present work complements their efforts. They presented experimental data, and here the model was simulated within a simple control strategy and the importance of the non-linear properties was stressed and analysed.

The importance of the mechanical part of the motor control system cannot be over stressed. In this work, another example was added to this idea, which was well described in Full (1994). A comprehensive simple muscle model is important if one is to achieve a proper understanding of the motor control mechanisms in the human nervous system. Muscles provide the only natural observable output for nervous system activity. Thus, they are the window through which one can look at the brain and the mind. The motor control research can be regarded as the act of polishing this window. In this respect, it is the first act in understanding the nervous system.

5.2 The control

Control of the muscles can be achieved with simple rectangular pulses. There is no need for the nervous system to calculate the trajectory in order to achieve a typical reaching movement. This can simplify the controller significantly. And it has physiological support in the form of hierarchic control through PG's which create the rectangular pulses and let the upper system specify only a few parameters. The rectangular shape is simple for simulation and analysis. In the biological system, because of the temporal and spatial filtering, there is a large set of shapes that would give the same qualitative results. The main conclusion is that only a few parameters of the excitation need to be controlled by the CNS due to the dynamic properties of the neuromuscular and the musculoskeletal systems.

After this study was carried out, a work with a similar parametric control scheme by Bock et al. (1993) came to our attention. In their work a 2DOF robotic arm is controlled with prototypical control torque signals that are defined by a set of parameters. Their control signals are not rectangular, and their robot naturally does not have any muscles. Their work serves as an example of a potential use of a biologically based approach to robot control.

5.3 The learning scheme

The control scheme with the linear sensitivity model was chosen for its simplicity. It may have an additional advantage since one can extract the relationship between the excitation parameters and the end position state from the sensitivity model.

As demonstrated, the suggested control scheme works well when limited to restricted areas of the workspace. But with proper pre-processing and/or choice of parameters, the entire workspace can be covered. The proposed algorithm is not proven to converge for all starting conditions. It can get into local minima, or even diverge during the learning phase to unreasonable movements outside the workspace. For robotics implementation, a more robust algorithm is needed. But this algorithm was sufficient for the purposes of modeling reaching movement and demonstrating the concept of plant controller interaction. Final position errors were of the order of millimeters, which is reasonable biologically speaking. But in simulation without noise, a better accuracy can be achieved. The way to achieve it is, as mentioned, a piecewise linear model or a better choice of parameters and errors. In this work only 2 parameters of 5 were used, and only 4 muscles from 6 were activated. Certainly, the proper incorporation of these control parameters would improve the performance. On the other hand, more free parameters will introduce redundancy, which is a major problem for any learning scheme. Finally, for the sake of completeness, external forces, noises, robustness, and energy considerations can be added to the model.

The entire model presented in this paper was a simplified one for learning and performing fast reaching movements. In other slower or more complicated movements, feedback must be incorporated during the movement or between short pre-programmed movements.

References

Abend W, Bizzi E, Morasso P (1982) Human arm trajectory formation. Brain 105:331–348

Arnon Y (1990) Arm trajectory planning by a recurrent neural network. MSc Thesis, Weizmann Institute of Science, Israel

Asada H, Slotin JE (1986) Robot analysis and control. Wiley Interscience, New York

Berthier NE (1996) Learning to reach: a mathematical model. Dev Psychol 32:811–823

Bock O, D'Eleuterio GMT, Lipitkas J, Grodski JJ (1993) Parametric motion control of robotic arms: a biologically based approach using neural networks. Telematics Inform 10:179–185

Demuth H, Beale M (1994) Neural network toolbox: for use with MATLAB. The MathWorks, Inc., Natick, Massachusetts

Desmedt JE (1983) Size principle of motoneuron recruitment and the calibration of muscle force and speed in man. In: Desmedt JE (ed) Motor control mechanisms in health and disease. Raven Press, New York

Dominey PF, Vindras P, Prablanc C, Pelisson D (1995) Eye-hand coordination in reaching movements. In: Arbib MA (ed) The handbook of brain theory and neural networks. MIT Press, Cambridge, Mass.

Flash T, Hogan N (1985) The coordination of arm movements: an experimentally confirmed mathematical model. J Neurosci 5:1688–1703

Full JR (1994) The importance of mechanical systems in understanding arthropod neural control of locomotion. Proceedings of the Eighth Yale Workshop on Adaptive and Learning Systems, pp 21–26

Georgopoulos AP, Taira M, Lukashin A (1993) Cognitive neurophysiology of the motor cortex. Science 260:47–52

Ghez C, Gordon J, Ghilardi MF, Christakos CN, Cooper SE (1990) Roles of proprioceptive input in the programming of arm trajectories. Cold Spring Harbor Symp Quant Biol LV:837–847

Gottlieb GL (1993) A computational model of the simplest motor program. J Motor Behav 25:153–161

Gottlieb GL, Corcos DM, Agarwal GC (1989) Strategies for the control of voluntary movements with one mechanical degree of freedom. Behav Brain Sci 12:189–250

Gottlieb GL, Chen C, Corcos DM (1995) 'Adequate control theory' for human single-joint elbow flexion on two tasks. Ann Biomed Eng 23:388–398

Hannaford B, Winters J (1990) Actuator properties and movement control: biological and technological models. In: Winters JM, Woo SL (eds) Multiple muscle systems. Springer, Berlin Heidelberg, New York

Haykin SS (1994) Neural networks: a comprehensive foundation. Macmillan Collage Publishing, New York

Hill AV (1938) The heat of shortening and dynamic constants of muscle. Proc R Soc Lond [B] 126:136–195

Hirayama M, Kawato M, Jordan MI (1993) The cascade neural network model and a speed-accuracy trade-off of arm movement J Motor Behav 25:162–174

Hollerbach JM (1982) Computers, brains and the control of movement. Trends in Neuro Sci 5:189–192

Houk JC, Wise SP (1992) Outline for a theory of motor behavior: involving cooperative actions of cerebellum, basal ganglia and cerebral cortex. Center for the study of Neuronal Populations and Behavior, Technical Report 5, Institute for Neuroscience Northwestern University

Inbar GF (1972) Muscles spindles in muscle control. II. Analysis of muscle servo model. Kybernetik 11:123–129

Inbar GF (1996) Estimation of human elbow joint mechanical transfer function during steady state and during cyclical movements. In: Gath I, Inbar GF (eds) Advances in processing and pattern analysis of biological signals. Plenum Press, New York

Inbar GF, Yafe A (1976) Parameter and signal adaptation in the stretch reflex loop. In: Homma S (ed) Progress in brain research, Vol 44, pp 317–337

Jeannerod M, Prablanc C (1983) Visual control of reaching movements in man. In: Desmedt JE (ed) Motor control mechanisms in health and disease. Raven Press, New York

Jordan MI (1996) Computational aspects of motor control and motor learning. In Heuer H, Keele SW (eds) Handbook of perception and action, Vol 2. Motor skills, Academic Press, London

Jordan MI, Rumelhart DE (1992) Forward models: supervised learning with distal teacher. Cogn Sci 16:307–354

Jordan MI, Flash T, Arnon Y (1994) A model of the learning of arm trajectories from spatial targets. J Cogn Neurosci 6: 359–376

Kalaska JF (1995) Reaching movements: implications of connectionist models. In: Arbib MA (ed) The handbook of brain theory and neural networks, MIT Press, Cambridge, Massachusetts

Kalaska JF, Crammond DJ (1992) Cerebral cortical mechanisms of reaching movements. Science 255:1517–1523

Kawato M, Gomi H (1992) A computational model of four regions of the cerebellum based on feedback-error learning. Biol Cybern 68:95–103

Kawato M, Gomi H, Katayama M, Koike Y (1992) Supervised learning for coordinative motor control. Proceeding of the third NEC research symposium, pp 126–161

Keifer J, Houk JC (1994) Motor function of the cerebellorubrospinal system. Physiol Rev 74:509–542

Krylow AM, Rymer WZ (1997) Role of intrinsic muscle properties in producing smooth movements. IEEE Trans Biomed Eng 44: 165–176

Luenberger DG (1984) Linear and nonlinear programming Addison-Wesley, New York

Massone LLE, Myers JD (1996) The role of plant properties in arm trajectory formation: a neural network study. IEEE Trans Syst Man Cybern [B] 26:719–732

Myers JD, Massone LLE (1997) The role of the plant properties in point-to-point arm movements: a robustness study. Biol Cybern 76:173–180

Seif-Naraghi AH, Winters JM (1990) Optimized strategies of scaling goal-directed dynamic limb movements. In:Winters JM, Woo SL (eds) Multiple muscle systems. Springer, Berlin Heidelberg, New York

Winters JM, Stark L (1987) Muscle models: what is gained and what is lost by varying model complexity. Biol Cybern 55:403–420

Zangemeister WH, Lehman S, Stark L (1981) Simulation of head movement trajectories: model and fit to main sequence. Biol Cybern 41:19–32