Biol. Cybern. 75, 73–83 (1996)

Biological Cybernetics

Learning combined feedback and feedforward control of a musculoskeletal system

Sybert Stroeve

Measurement and Control Laboratory, Faculty of Mechanical Engineering and Marine Technology, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands

Received: 7 November 1995 / Accepted in revised form: 13 February 1996

Abstract. The goal of this paper is the learning of neuromuscular control, given the following necessary conditions: (1) time delays in the control loop, (2) non-linear muscle characteristics, (3) learning of feedforward and feedback control, (4) possibility of feedback gain modulation during a task. A control system and learning methodology that satisfy those conditions is given. The control system contains a neural network, comprising both feedforward and feedback control. The learning method is backpropagation through time with an explicit sensitivity model. Results will be given for a one degree of freedom arm with two muscles. Good control results are achieved which compare well with experimental data. Analysis of the controller shows that significant differences in controller characteristics are found if the loop delays are neglected. During a control task the system shows feedback gain modulation, similar to experimentally found reflex gain modulation during rapid voluntary contraction. If only limited feedback information is available to the controller the system learns to co-contract the antagonistic muscle pair. In this way joint stiffness increases and stable control is more easily maintained.

1 Introduction

The focus of this paper is on learning optimal control of musculoskeletal systems. The actuators in a musculoskeletal system are highly non-linear. Significant time delays are present in the control system. To deal with such a system the central nervous system employs feedback as well as feedforward neuromuscular control modes. Feedforward control is necessary in order to make fast movements in spite of the significant delays in the control loop. The feedforward controller contains the inverse dynamics of the musculoskeletal system and therefore it is a non-linear system. Feedback control is required for compensation of external disturbances and of errors in the internal model of the inverse dynamics. It follows from experiments (e.g. Kirsch et al. 1993) that the feedback gains are not constant but can be adapted by supraspinal pathways, depending on the task being executed. In summary, a neuromuscular control system is non-linear, combines feedback and feedforward control modes and contains varying feedback gains.

A framework for learning control of voluntary movements by the central nervous system is given by Kawato et al. (1987). The suggested control system consists of an internal model of the (forward) dynamics of the musculoskeletal system, an internal model of the inverse dynamics and a controller. The *controller* generates proper neural control signals based on the desired trajectory, the delayed and estimated states of the musculoskeletal system and inverse dynamics signals. The *internal model of the forward dynamics* predicts state values of the musculoskeletal system, which are used by the feedback controller. For training of the *inverse dynamics model* Kawato and coworkers introduced a learning strategy called feedback error learning (FEL). In this approach it is assumed that the neural input $\mathbf{u}$ of the musculoskeletal system is the sum of the control signals generated by separate feedback ($\mathbf{u}_f$) and feedforward ($\mathbf{u}_i$) controllers. The feedback system controls the musculoskeletal system in a basic, non-optimal way. The feedforward controller is trained using the feedback control signals. This learning method comes down to minimization of $E = \frac{1}{2T}\int_0^T \mathbf{u}_f'(t)\mathbf{u}_f(t)dt$ by a supervised learning method ($'$ denotes the transpose of the vector and T a training time-interval). Since $\mathbf{u}_f = \mathbf{u} - \mathbf{u}_i$ this means that the desired output of the inverse dynamics network in the supervised learning procedure is $\mathbf{u}$. If the feedback control signals tend to zero and persistently exciting desired trajectories have been applied, the inverse dynamics have been learned well.

Application of feedback error learning to robots (e.g. Kawato et al. 1987) as well as to neuromuscular control (Katayama and Kawato 1991, 1993; Gomi and Kawato 1992) shows that this methodology can lead to good results for systems with several degrees of freedom. It is stated how FEL could actually be applied in several specific neuromuscular systems, contributing to the biological plausibility of this learning concept. Limitations of the reported neuromuscular control simulations are: (1) the simple muscle models, which neglect many known non-linear characteristics, (2) the constant linear feedback controllers, which neglect the known feedback gain variability and (3) the lack of significant loop delays. Kawato and co-authors (1987) admit

the importance of loop delays in neuromuscular control and suggest an internal model of the forward dynamics of the musculoskeletal system to compensate for time delays. In the learning simulations, however, they left out the internal forward dynamics model, due to computer limitations, and omitted time delays. It can thus be concluded that they assumed a perfect internal model of the forward dynamics during learning of the inverse dynamics. However, the existence of a perfect internal model of the forward dynamics when the inverse dynamics are completely unknown is unlikely. The omission of feedback delays during feedback error learning is therefore not justified. In simulation of simultaneous adaptation of the vestibulo-ocular reflex and the optokinetic eye movement response using FEL, Gomi and Kawato (1992) included feedback delays of 10 and 20 [ms]. These delays are very small with respect to the applied movement frequency of 0.2 [Hz]. Stability of FEL for systems with significant delays can thus not be inferred from these simulations.

Gorinevsky (1993) included time delays in his neuromuscular control simulations concerning a two-link arm with two muscle pairs. He employed a linear feedback controller whose gains were varied according to a prespecified law. Furthermore the co-contraction level of the muscles was varied according to a predefined rule. A feedforward controller was trained using a paradigm called 'direct motor learning'. Limitations of this study concern (1) the omission of length-dependent elements in the muscle models and (2) the fact that feedback gains and co-contraction levels are predefined rather than learnt.

The goal of this paper is the learning of optimal control of a musculoskeletal system, given the necessary boundary conditions discussed above. These conditions are: (1) inclusion of time delays, (2) inclusion of non-linear muscle characteristics, (3) learning of feedforward and feedback control, (4) possibility of feedback gain modulation during a task.

In Sect. 2 a neuromuscular control system and a learning methodology which satisfy these conditions will be discussed. Learning and typical responses after learning will be discussed in Sect. 3.1. To stress the importance of the inclusion of time delays, the differences between controllers which were trained with and without delays will be analysed in Sect. 3.2. In Sect. 3.3 the training methodology is applied to study the effect of limited controller feedback. The simulation results obtained are compared with experimental data and further discussed in Sect. 4.

2 Methods

2.1 Musculoskeletal system

The musculoskeletal system to be controlled is a one degree of freedom link with an antagonistic muscle pair. The system equations are based on the work of Winters and Stark (1985). Each muscle contains first-order excitation dynamics, non-linear first-order activation dynamics, a Hill type force-velocity dependence, a Gaussian force-length dependence, a series elastic element and a parallel elastic element. Constant moment arms are assumed. The link dynamics are represented by two states, namely angle θ and angular velocity $\dot{\theta}$. Contraction of the agonist muscle results in a positive

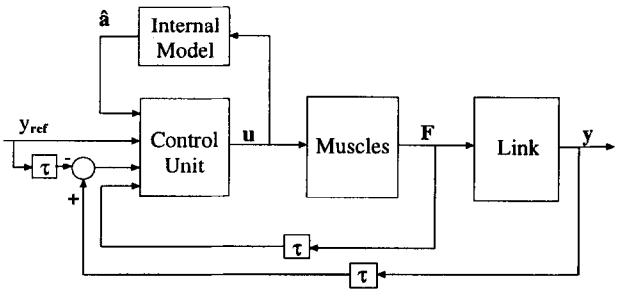

Fig. 1. Complete control system consisting of control unit, internal model, muscles and link. The internal model estimates the muscular activations: $\hat{\mathbf{a}} = [\hat{a}_1\ \hat{a}_2]^T$. The neural input vector $\mathbf{u} = [u_1\ u_2]^T$ controls the musculoskeletal system. Sensorial information is known after a delay τ and contains the muscular forces $\mathbf{F} = [F_1\ F_2]^T$ and joint angle and angular velocity $\mathbf{y} = [\theta\ \dot{\theta}]^T$. The reference trajectory is specified at a higher level: $\mathbf{y}_{ref} = [\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}]^T$

θ. The dynamics of each muscle can be described by three states, namely excitation e, activation a and contractile element length l_{ce}. The total musculoskeletal system thus is a non-linear eighth-order system. The neural input signal u of each muscle is normalized to the [0,1] interval. Appendix A gives all the system equations. The system is calculated using second-order Runge-Kutta integration with a constant time step $h = 0.5$ [ms].

2.2 Control system

The following conditions with respect to the control system were formulated: (1) it must be non-linear, (2) it must comprise both feedforward and feedback control modes, (3) both parts must be adaptive, (4) feedback gains can be modulated during a control task (after learning) and (5) delays must be accounted for. Conditions 1 to 4 are satisfied by a controller which is one neural network with feedforward as well as feedback input signals. Such a system is non-linear (condition 1) and contains both control modes (condition 2) which are both adaptive (condition 3), since the whole network is adaptive. Furthermore the feedback gains can be modulated by the feedforward signals during a control task (condition 4). The fifth condition is satisfied by applying delayed feedback signals.

The control system is schematically depicted in Fig. 1. The total loop delay, which consists of neural processing time and efferent and afferent transport times, is estimated to be 50 [ms] (as in Gerdes and Happee 1994). The total loop delay is modelled by a feedback delay τ between the receptors and the controller.

The control unit is a multilayer perceptron neural network. The network contains one hidden layer with 15 units employing sigmoidal activation functions. Also the output layer contains sigmoidal activation functions, limiting the neural control signals to the [0, 1]-interval, as required. The neural network is represented by:

$$\mathbf{u}(t) = \Gamma_2(\mathbf{W}_2\Gamma_1(\mathbf{W}_1\mathbf{s}(t) + \mathbf{b}_1) + \mathbf{b}_2) \quad (1)$$

$$\sigma(x) = 1/(\exp(-x) + 1) \quad (2)$$

with $\mathbf{s}(t)$ network input, $\mathbf{u}(t)$ network output, $\mathbf{W}_1$ and $\mathbf{W}_2$ weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ bias vectors, and Γ_1 and Γ_2 arrays of sigmoidal functions σ.

The two-dimensional network output vector $\mathbf{u}(t)$ specifies the neural input of the muscles. The network input vector $\mathbf{s}(t)$ consists of feedforward and feedback signals. An aspect of this study is the effect of limited feedback information on the control strategies. The controller configuration specified here has a complete input set, but subsets will also be used. It is assumed that the central nervous system (CNS) plans a movement trajectory in joint coordinates. The desired link angle $\theta_{ref}(t)$, angular velocity $\dot{\theta}_{ref}(t)$ and angular acceleration $\ddot{\theta}_{ref}(t)$ are specified to the controller. Since a static system is applied for the control of a system with complex muscular dynamics and the muscular activations are important state variables which can not be measured, it is assumed that an accurate internal model of the excitation-activation dynamics exists. This internal model which represents only a small part of the total musculoskeletal system, predicts the actual activation of each muscle accurately. The controller employs the estimated activations $\hat{\mathbf{a}}(t)$.

The musculoskeletal system provides the CNS with the following proprioceptive information: length and velocity of each muscle, measured by muscle spindles; muscle force, measured by Golgi tendon organs; and angular position and velocity of a joint, measured by receptors in synovia and ligaments. Since the reference trajectory is expressed in joint variables, it is assumed that the proprioceptive information is transformed to the joint domain. This transformation involves a (straightforward) static mapping to link angle and velocity. The (delayed) feedback input signals of the controller are the muscle force of both muscles $\mathbf{F}_m(t-\tau)$, angle error $e(t-\tau)$ and angular velocity error $\dot{e}(t-\tau)$:

$$e(t-\tau) = \theta(t-\tau) - \theta_{ref}(t-\tau) \tag{3}$$

$$\dot{e}(t-\tau) = \dot{\theta}(t-\tau) - \dot{\theta}_{ref}(t-\tau) \tag{4}$$

2.3 Learning optimal neuromuscular control

To achieve optimal control of the system discussed in the previous section, backpropagation through time (BTT) is applied as the learning algorithm. This learning algorithm is usually employed with backpropagation through an identification model of the plant (e.g. Nguyen and Widrow 1991). Applying backpropagation through an identification model calculates the cost gradient with respect to the model input $\nabla_{\mathbf{u}} J$ from the cost gradient with respect to the model output $\nabla_{\mathbf{y}} J$. The cost function specifies the desired control behaviour. Now implicitly the Jacobian or sensitivity matrix of the identification model has been utilized. This matrix contains the partial derivatives $\partial y_i / \partial u_j$ for all inputs and outputs. In this paper a sensitivity model which calculates the exact Jacobian for an arbitrary working-point is applied. This means that the learned control is not hampered by an imperfect identification model, as is desired in an optimization study.

In the following a general description of the BTT algorithm for learning control simulations will be given. Given a non-linear, time-invariant dynamical system with input vector $\mathbf{u}(t)$, state vector $\mathbf{x}(t)$ and output vector $\mathbf{y}(t)$:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \tag{5}$$

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) \tag{6}$$

A general static (neural network) controller $\mathbf{q}$ with input vector $\mathbf{s}(t)$ and output $\mathbf{u}(t)$ is specified by:

$$\mathbf{u}(t) = \mathbf{q}(\mathbf{s}(t)) \tag{7}$$

The input vector $\mathbf{s}(t)$ may contain all kinds of (possibly delayed) signals which seem relevant for the control task. Logical candidates are reference signals $\mathbf{r}(t)$ and system output signals $\mathbf{y}(t)$. The desired control behaviour over the interval $[0, T]$ is specified by a cost function J:

$$J = \int_0^T l(\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{r}(t), t, T) dt \tag{8}$$

Application of BTT and computer simulation requires discretization of the system dynamics (by e.g. a Runge-Kutta method) and of the cost function:

$$\mathbf{x}(k+1) = \mathbf{f}_d(\mathbf{x}(k), \mathbf{u}(k)) \tag{9}$$

$$J = \sum_{k=0}^{N} l(\mathbf{x}(k), \mathbf{y}(k), \mathbf{u}(k), \mathbf{r}(k), k, N) \tag{10}$$

Discrete and continuous time are now associated by:

$$t = hk \tag{11}$$

with h the discretization time-step.

Learning of the controller $\mathbf{q}$ requires knowledge of the effect of the control signal $\mathbf{u}(k)$ on the costs J. The total causal impact of $u_j(k)$ on J is represented by the ordered derivative (Werbos 1988):

$$\frac{\partial^+ J}{\partial u_j(k)} \tag{12}$$

This ordered derivative $\partial^+ J / \partial u_j(k)$ measures how much a small change in u_j at time k affects the costs J over the whole trajectory. The ordered derivative accounts for the direct and indirect effects of a change in $u_j(k)$ on J, as opposed to the conventional partial derivative $\partial J / \partial u_j(k)$ which refers only to the direct effect. The required ordered derivatives are calculated recursively by:

$$\frac{\partial^+ J}{\partial x_i(N)} = \frac{\partial l(N)}{\partial x_i(N)} + \sum_{q=1}^{p} \frac{\partial l(N)}{\partial y_q(N)} \frac{\partial y_q(N)}{\partial x_i(N)} \tag{13}$$

$$\frac{\partial^+ J}{\partial x_i(k)} = \frac{\partial l(k)}{\partial x_i(k)} + \sum_{q=1}^{p} \frac{\partial l(k)}{\partial y_q(k)} \frac{\partial y_q(k)}{\partial x_i(k)} + \sum_{q=1}^{n} \frac{\partial^+ J}{\partial x_q(k+1)} \frac{\partial x_q(k+1)}{\partial x_i(k)} \tag{14}$$

$$k = 1 \ldots N-1, \quad i = 1 \ldots n$$

$$\frac{\partial^+ J}{\partial u_j(k)} = \frac{\partial l(k)}{\partial u_j(k)} + \sum_{q=1}^{n} \frac{\partial^+ J}{\partial x_q(k+1)} \frac{\partial x_q(k+1)}{\partial u_j(k)} \tag{15}$$

$$k = 0 \ldots N-1, \quad j = 1 \ldots m$$

where n, m and p are the sizes of the state, input and output vectors, respectively. The partial derivatives $\partial l(k) / \partial x_i(k)$, $\partial l(k) / \partial u_j(k)$ and $\partial l(k) / \partial y_q(k)$ can be calculated directly given the function $l(k)$. The partial derivatives $\partial y_q(k) / \partial x_i(k)$

are known by computation of the system Jacobian $\nabla_x \mathbf{g}$. The partial derivatives $\partial x_q(k+1)/\partial x_i(k)$ and $\partial x_q(k+1)/\partial u_j(k)$ are calculated in an indirect way depending on the integration method. Computation of these derivatives is most straightforward for first-order Euler's integration method, given by:

$$\mathbf{x}(k+1) = \mathbf{x}(k) + h\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k)) \tag{16}$$

It can easily be seen that the required partial derivatives can now be computed, if the Jacobian matrices $\nabla_x \mathbf{f}$ and $\nabla_u \mathbf{f}$ are known:

$$\frac{\partial x_i(k+1)}{\partial x_j(k)} = \begin{cases} 1 + h\frac{\partial f_i}{\partial x_j}(\mathbf{x}(k), \mathbf{u}(k)), & i = j \\ h\frac{\partial f_i}{\partial x_j}(\mathbf{x}(k), \mathbf{u}(k)), & i \neq j \end{cases} \tag{17}$$

$$\frac{\partial x_i(k+1)}{\partial u_j(k)} = h\frac{\partial f_i}{\partial u_j}(\mathbf{x}(k), \mathbf{u}(k)) \tag{18}$$

Also more complex integration methods can be used for the dynamic feedback calculation. Appendix B gives the equations for second-order Runge-Kutta, as has been applied in the simulations.

Given $\partial^+ J/\partial u_j(k)$ the ordered derivatives of the costs with respect to the weights of the neural network $\partial^+ J(k)/\partial w$ are calculated for each time-step k using backpropagation. By analogy with standard supervised learning this derivative information can be applied in a pattern or batch-like way for updating the weights. A batch update means adaptation of the weights in the negative direction of the derivative, averaged over the whole trajectory:

$$\Delta w = -\frac{\eta}{N}\sum_{k=0}^{N-1}\frac{\partial^+ J(k)}{\partial w} \tag{19}$$

A pattern update signifies weight adaptation for each time-step:

$$\Delta w(k) = -\eta\frac{\partial^+ J(k)}{\partial w}, \quad k = 0 \ldots N-1 \tag{20}$$

where the learning constant η for batch updating (19) is of course larger than that for pattern updating (20). Since the pattern update was found to be more effective than the batch variant, this former method was used in all simulations presented in this paper.

Figure 2 schematically shows the BTT learning procedure for a control system without feedback delays. In this example the control signal $\mathbf{u}(k)$ is based on a reference signal $\mathbf{r}(k)$ and a system output $\mathbf{y}(k)$. The system has been unfolded over N time-steps. After time-step N the costs over the whole trajectory are calculated. Going backward in time the ordered gradient vector $\nabla^+_{\mathbf{x}(k)} J$ is calculated using (13) and (14), and $\nabla^+_{\mathbf{u}(k)} J$ is computed according to (15).

2.4 Learning parameters

In the previous section a general formulation of the learning algorithm was given. Some specific procedures and parameters as applied in the performed simulations are now described. The applied cost function J is:

$$J = \frac{1}{T}\int_0^T (\theta(t) - \theta_{ref}(t))^2 + \gamma a_1^2(t) + \gamma a_2^2(t)\, dt \tag{21}$$

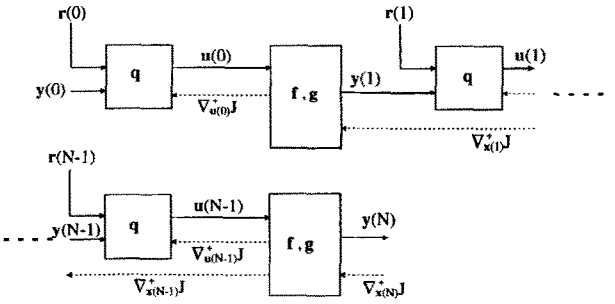

Fig. 2. Diagram of the backpropagation through time (BTT) learning procedure. The control system, which consists of a controller **q** and a dynamical system (**f**, **g**), has been unfolded over N time-steps. Going forward in time (*continuous arrows*) the control signals $\mathbf{u}(k)$ and the system outputs $\mathbf{y}(k)$ are calculated, employing the reference signals $\mathbf{r}(k)$. Going backward in time (*dashed arrows*) the ordered gradient vectors of the costs J with respect to the states $\mathbf{x}(k)$ and the control inputs $\mathbf{u}(k)$ are computed

with $\gamma = 0.05$ and $T = 0.3$ [s]. This cost function minimizes angle deviations as well as muscular activations. The activation signals are minimized in order to attain low energy consumption. The parameters were chosen such that satisfactory simulation results were attained.

As specified in Sect. 2.1, the integration time-step for the musculoskeletal system is 0.5 [ms]. This short time-step is necessary because of the inclusion of a stiff series elastic element in the musculoskeletal model. The output dynamics of the musculoskeletal system do not necessitate such short update times for the controller. This update time was chosen to be 2.5 [ms]: five times as large as the integration time-step. Unfolding the system in time, as is shown in Fig. 2, now implies that each controller block is followed by five system dynamics blocks. Unfolding the system over time T means a concatenation of 120 such combined blocks in the BTT algorithm.

An iteration during learning lasts $T_i = 5$ [s]. In each iteration a reference signal is specified. The applied reference signals consist of the sum of six sinusoids with frequencies and amplitudes randomly chosen from uniform distributions: $[\omega_{min}, \omega_{max}]$ and $[A_{min}, A_{max}]$, respectively. The amplitude of the reference signal is normalized such that it does not exceed A_{max}. The distribution parameters are: $\omega_{min} = 0.1$ [rad/s], $\omega_{max} = 8.0$ [rad/s], $A_{min} = 0.1$ [rad], $A_{max} = 1.25$ [rad].

The learning parameter η was constant during all simulations. A value $\eta = 20$ gave good and mostly stable results.

2.5 Controller analysis

During learning the controller is evaluated by calculating the costs J, root mean squared angle deviations and average neural input signals for three sinusoidal reference signals with amplitude A_{max} and frequencies at 20%, 50% and 80% of the reference frequency interval. Apart from evaluation by system responses the controller is analysed by linearization of the controller along the performed trajectory. Input-output linearization provides us with information on the relation of feedforward and feedback control components and on the variation of feedback gain factors.

Given the backpropagation algorithm, linearization of the controller **q** is very straightforward. In the applied backpropagation algorithm the cost gradient with respect to the network output vector $\nabla_{\mathbf{s}} J$ is employed to calculate the cost gradient with respect to the network weights $\nabla_{\mathbf{w}} J$ and also with respect to the network input vector $\nabla_{\mathbf{u}} J$. In order to calculate $\partial u_j / \partial s_i$ for the working point **s** the derivative of the costs with respect to one of the outputs is defined as 'one' and all others as 'zero':

$$\frac{\partial J}{\partial u_l} \equiv \begin{cases} 1 & l = j \\ 0 & l \neq j, \quad l = 1 \ldots m \end{cases} \tag{22}$$

with m the size of the vector **u**. Backpropagation now renders the sensitivity vector of this particular output for variations of the input signals:

$$\frac{\partial u_j}{\partial s_i} = \frac{\partial J}{\partial s_i} \tag{23}$$

since

$$\frac{\partial J}{\partial s_i} = \sum_{j=1}^{m} \frac{\partial J}{\partial u_j} \frac{\partial u_j}{\partial s_i} \tag{24}$$

3 Results

In this section the results of learning the musculoskeletal system with BTT will be discussed. In Sect. 3.1 typical learning curves and responses after training will be reviewed. To demonstrate the importance of learning with sensorial delays, it will be shown that different control strategies have been acquired after learning with and without the presence of those delays in Sect. 3.2. Finally a self-emerging control strategy, found after learning with limited sensorial feedback, will be discussed in Sect. 3.3.

3.1 Learning curves and typical responses

Learning processes with the same learning parameters result in somewhat different learning curves, since there are some stochastic factors in the learning process. Those factors are the randomly chosen initial weights and parameters determining the reference signals. Figure 3 shows the average learning curves plus standard deviations of five trials for a typical learning parameter set. Each of the learning processes takes a computation time of $5\frac{1}{2}$ h on a Silicon Graphics Indigo 2 workstation. Observe the fast decay of the costs in the first 250 [s] and the slow decay subsequently. The small standard deviation indicates that the influence of the stochastic factors is not large. Examination of the cost components which are reflected in the root mean squared error (RMSE) and average neural input, shows a slower initial decay of the neural control signals than that of the RMSE. The higher neural input signals indicate an increased coactivation level during the initial learning phase. Because of slight fluctuations in the individual cost curves, the average curve is not monotonically decaying. Therefore the average costs, RMSE and neural input over the final 500 iterations, denoted as $\bar{J}_f$, $\bar{E}_f$ and $\bar{u}_f$, will be taken as a measure of the results achieved.

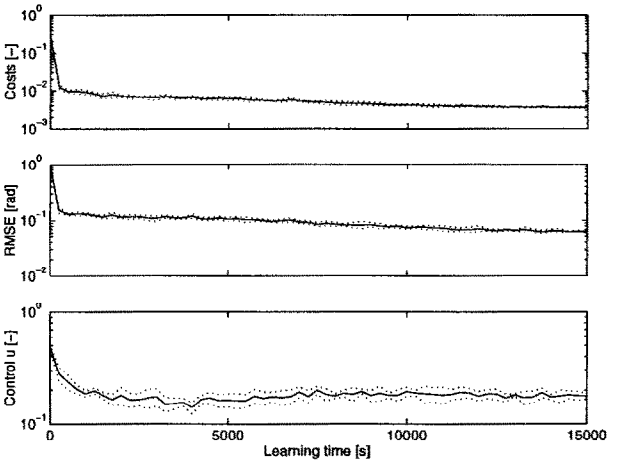

Fig. 3. Average learning curves (n = 5) plus standard deviations showing the course of costs (*top*), RMSE (*middle*) and average neural input (*bottom*)

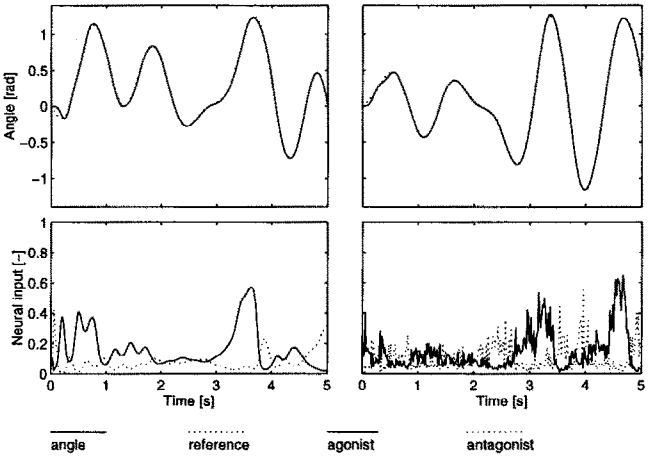

Fig. 4. System responses for typical multi-sinusoidal reference signals after learning has been completed. The *left-hand figure* shows achieved and reference angles plus agonistic and antagonistic control signals for a system without measurement noise. The *right-hand figure* shows those signals for a system where measurement noise is included

After learning, good combined feedforward and feedback control of the musculoskeletal system with sensorial delays has been attained. Figure 4 shows the system responses of a system with and without measurement noise for typical reference signals, as applied during learning. Only small errors occur and the amplitude of the neural input signals is low, indicating a low energy consumption. Furthermore, it follows that the system is not noise sensitive. White measurement noise clearly affects the neural input signals, since it is not attenuated by the (static) controller. The trajectory achieved is, however, barely influenced, given the low-frequency pass characteristics of the musculoskeletal system.

The system that was trained with multi-sinusoidal reference signals can control all kinds of movements that lie in the training frequency spectrum. A good example of such a movement, for which there exist abundant experimental data, is a fast goal-directed movement. Typical experimental data of fast arm movements (e.g. Happee 1992) show triphasic electromyographic (EMG) patterns: activation of agonistic muscles to accelerate the limb, activation of an-

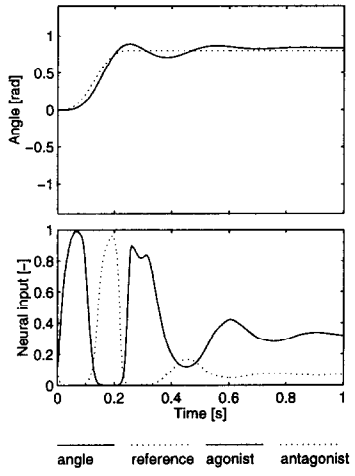

Fig. 5. System response for a step-like reference signal showing triphasic neural input signal

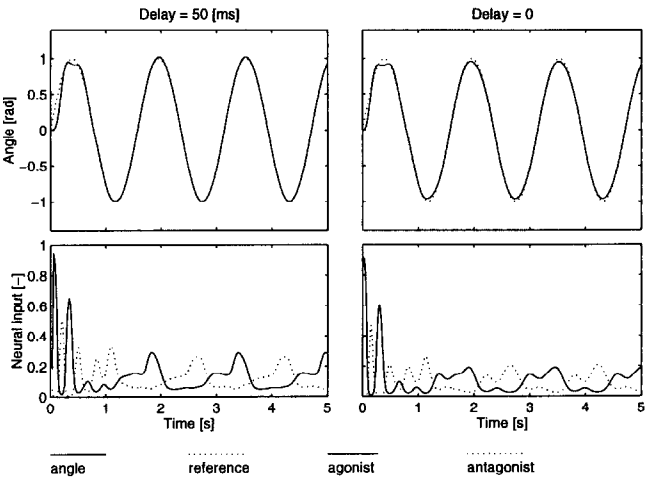

Fig. 6. Responses for a sinusoidal reference signal (4 [rad/s]) of a system with (*left*) and without (*right*) delays

tagonistic muscles to decelerate the limb and activation of agonistic muscles to compensate for too much deceleration. In this way a time-optimal movement is possible. Figure 5 shows the response of the trained neural control system for a fast step-like reference signal. A triphasic neural input signal, similar to EMG measurements, can be clearly observed. After the desired position has been reached, a low agonistic neural input is applied to resist the force applied by the parallel elasticity.

3.2 Effect of sensorial delay

An important, though often neglected feature in neuromuscular control systems is the presence of time delays. To gain a better understanding of the effect of delays on the learned control strategy, learning with and without delays will be compared. For this analysis a sinusoidal reference signal with a frequency of 4 [rad/s] (approximately the mean of the reference spectrum) and an amplitude of 1 [rad] is used. Figure 6 shows the responses of the system with and without delay for this reference signal. After a start-up effect regular patterns and small errors remain. Although the RMSE and average neural input of the system with delay are somewhat higher, the responses look rather similar. Analysis of the control systems for the whole reference frequency spectrum shows that the system without delay attains lower costs for all frequencies. Since the omission of delays simplifies the control task, this is as expected. Those lower costs are in particular due to lower neural input signals.

In spite of the similarity of the neural input signals the controllers are different. This follows from the simple experiment in which the controller that was trained without delay is asked to control the system with delay: unstable control results. This can also be concluded from a comparison of the contribution of feedback and feedforward control parts for both systems. Figure 7 shows the sensitivity of the agonistic neural input for controller input variations, as defined in Sect. 2.5. Notice that the sensitivity for signals specifying the reference trajectory is higher for the system with delays than for the one without delays. This means that feedforward

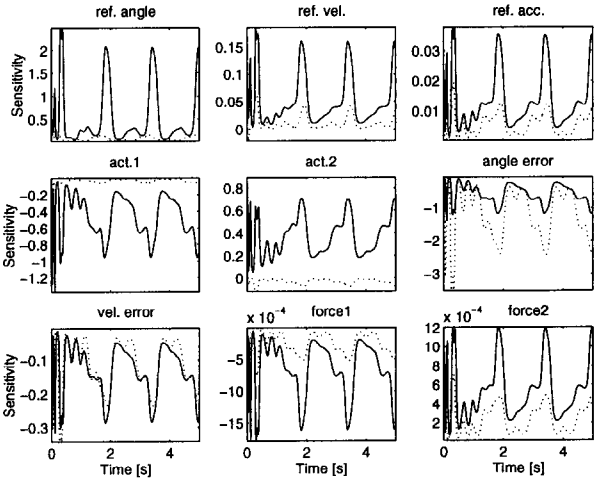

Fig. 7. Sensitivity of agonistic control signal for all controller input signals: with (*continuous line*) and without (*dotted line*) delays. The controller inputs are: reference angle, velocity and acceleration, angle and velocity error, activations and forces of both the muscles

control is more prominent in the system with delay. Also, the sensitivity for the estimated activations, which represent a part of the feedforward control too, is higher for the system with delay. On the other hand the system without delay shows higher sensitivity for the angle error, whereas the sensitivity for the velocity error is similar for both the systems. Feedback control is thus more prominent in the system without delay. The sensitivity for the two other feedback signals, namely the muscle forces, is higher for the system with delay. Apparently, information regarding the arm acceleration, which is highly correlated with muscle forces, is barely needed in the system without delay. The system with delay, however, employs this knowledge to acquire a phase lead.

Another striking phenomenon of Fig. 7 is that the sensitivities fluctuate. Comparison of the agonistic neural input in Fig. 6 with the sensitivities for the (realistic) system with delay makes it clear that the sensitivities fluctuate in correspondence with the neural input signal. The sensitivity for

Table 1. Learning with limited feedback: average results plus standard deviations (n = 5) for final costs $\bar{J}_f$, RMSE $\bar{E}_f$ and neural input $\bar{u}_f$

Conf. no.	Controller input signals	$\bar{J}_f$ (10^{-3})[−]	$\bar{E}_f$ (10^{-2}) [rad]	$\bar{u}_f$ (10^{-1})[−]
1	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}\ \hat{\mathbf{a}}\ e\ \dot{e}\ \mathbf{F}$	3.64 (±0.091)	6.55 (±0.24)	1.78 (±0.095)
2	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}\ e\ \dot{e}\ \mathbf{F}$	3.68 (±0.086)	6.94 (±0.17)	1.71 (±0.044)
3	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}\ \hat{\mathbf{a}}\ e\ \dot{e}$	4.00 (±0.082)	6.57 (±0.26)	2.08 (±0.047)
4	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}\ e\ \dot{e}$	5.66 (±0.25)	10.1 (±0.11)	2.34 (±0.041)
5	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}\ e$	12.5 (±0.78)	15.0 (±1.2)	4.38 (±0.031)
6	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}\ \hat{\mathbf{a}}$	6.87 (±0.35)	14.2 (±1.0)	2.70 (±0.20)
7	$\theta_{ref}\ \dot{\theta}_{ref}\ \ddot{\theta}_{ref}$	14.4 (±1.7)	15.5 (±1.6)	4.70 (±0.097)

Conf., configuration

those signals is high when the muscle is activated and low when it is not. Since the sensitivities of the angle error, velocity error and muscle forces are equivalent to the feedback gains for those signals, it follows that those feedback gains vary in correspondence with the neural input signals.

3.3 Limited sensorial feedback

The controller in the former sections has sufficient information concerning the state of the musculoskeletal system: (delayed) joint angle and angular velocity, activations and (delayed) muscle forces. To gain more insight into the significance of those controller input signals, the results of learning with limited feedback will be discussed in this section. Sensorial delays were always included in these simulations. Table 1 shows the results. It specifies the average final costs $\bar{J}_f$, RMSE $\bar{E}_f$ and neural input $\bar{u}_f$ for several controller configurations. Notice that the reference trajectory (θ_{ref}, $\dot{\theta}_{ref}$, $\ddot{\theta}_{ref}$) and estimated activation signals ($\hat{\mathbf{a}}$) are part of the feedforward control, whereas the error (e, $\dot{e}$) and force ($\mathbf{F}$) signals specify the delayed feedback information from the musculoskeletal system. Controller configurations 1 to 5 are thus combined feedback/feedforward controllers, while configurations 6 and 7 are feedforward controllers.

Table 1 shows that configuration (Conf.) 1, which contains the most ample set of input signals, achieves the lowest costs. Omitting only the estimated activation signals (Conf. 2) does not, however, lead to significant performance degradation. Omitting only the force signals (Conf. 3) does not result in increased deviations from the reference trajectory, but leads to a slightly increased co-contraction level. If both force feedback and activation estimation are left out (Conf. 4) a significant cost increase results, caused by both an increase in deviations from the reference trajectory and a rise in the neural input signals. If in addition no velocity feedback is present and control is thus based only on the reference trajectory and position feedback (Conf. 5), even higher deviations and input signals result. Omitting the position feedback in Conf. 5 gives a pure feedforward controller (Conf. 7). This system achieves RMSEs similar to Conf. 5 and somewhat higher neural input signals. Addition of the activation dynamics (Conf. 6), making the feedforward control dynamic, improves the results significantly. In particular the neural input is now much lower.

Figure 8 shows the response of a typical system with only position feedback (Conf. 5) for a sinusoidal reference signal.

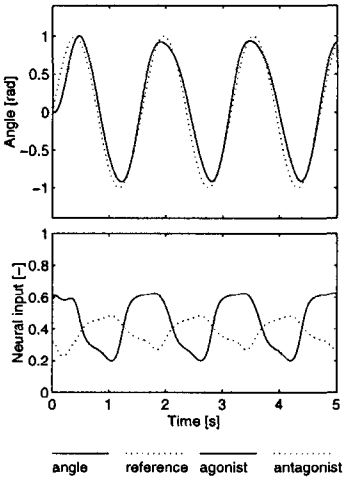

Fig. 8. Control with limited feedback: only joint angle and reference variables are available to the controller

Compare this response with that of a complete feedback system in Fig. 6. In particular the increased co-contraction level stands out. In spite of the limited (and delayed) feedback signal the controller is able to steer the system fairly well. Due to the co-contraction the stiffness of both the muscles is increased and this results in an increase in joint stiffness. The average instantaneous joint stiffness ($\partial T/\partial\theta$) is 1.3×10^2 [Nm/rad] for the complete feedback case of Fig. 6, whereas it is 3.3×10^2 [Nm/rad] for the limited feedback case of Fig. 8.

The performance of the control system with limited feedback is strongly dependent on the frequency spectrum of the reference signal. Figure 9 shows the RMSE and average neural input as a function of the frequency of a sinusoidal reference signal with an amplitude of 1.0 [rad] for control systems with complete feedback and with limited feedback. The systems are trained either with the normal (broad range) reference frequency spectrum ([0,8] rad/s) or with a low-range spectrum of [0,4] rad/s. It can be seen that large deviations from the reference trajectory occur for high frequencies (> 6 [rad/s]) in the case of limited feedback. The musculoskeletal system thus cannot be controlled properly for high frequencies if only position feedback is available to the controller. Furthermore, the performance of the system with limited feedback depends strongly on the frequency spectrum of the reference signal as applied *during* learning. Figure 9 shows that the average neural input of the system with limited feedback is significantly reduced

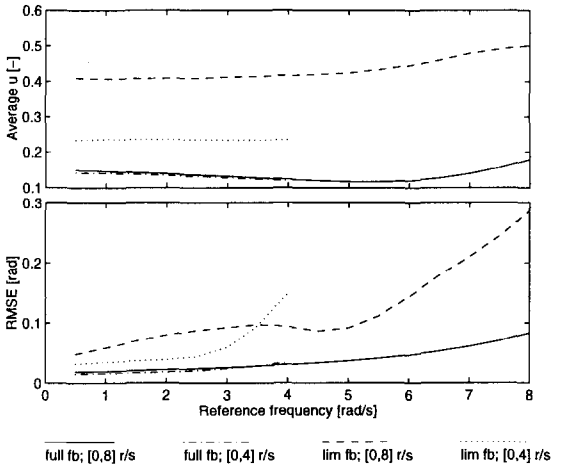

Fig. 9. RMSE and average neural input as a function of reference frequency for the system trained with wide range ([0,8] rad/s) and low range ([0,4] rad/s) reference frequencies: complete feedback, wide range (*continuous line*); complete feedback, low range (*dash-dotted line*); position feedback only, wide range (*dashed line*); position feedback only, low range (*dotted line*)

after learning with low frequencies. Control now employs lower co-contraction levels. Besides, for all but the highest frequencies smaller deviations from the reference trajectory are attained.

4 Discussion

The simulation results show that good control of a one degree of freedom musculoskeletal system, embracing many known non-linear characteristics, is attained. The main component of the control system is a static multilayer perceptron neural network, trained by the BTT algorithm. What do these results tell us about real life? The validity of the simulation results is linked with the validity of the musculoskeletal model, the control system and the learning method.

The musculoskeletal model used (Appendix A) is based on the work of Winters and Stark (1985). It contains many essential elements which have often been neglected in neuromuscular control simulations, including excitation/activation dynamics, length and velocity dependent contractile force, and parallel and series elasticities. It is a suitable model for the neuromuscular control simulations performed here.

The control system consists of three parts: (1) a reference trajectory generator, (2) the neural network controller and (3) the activation estimator. The main component is the controller, implemented by a static multilayer perceptron network. This neural network structure was chosen because it is known to have powerful representational capabilities (Leshno et al. 1993) and does not need a priori specifications, except for the dimensions of the network. Since the processes in a biological neural network are much more complex than those in its artificial namesake, the artificial neurons are not intended as precise models of real neurons. Furthermore the signals in the neuromuscular system are scalars in the simulations, whereas in reality they are represented in a distributed way. The information flow in the real nervous tissue can, however, at a higher system level be represented by an artificial neural network. The distributed information processing in the two systems is a nice similarity. The main omission in this work seems to be the assumption of a static neural control unit, while dynamic processes are known to be present in nervous tissue. The significance of such dynamic processes in sensory-motor control is, however, not known and the results achieved indicate that good control can be reached by a static mapping. Furthermore it should be observed that the control system is dynamic by inclusion of the activation estimation and the trajectory generation process. It can be concluded that the applied system is a suitable model of the biological neural control system at the input-output level.

In the BTT learning algorithm, a cost function containing deviations from the desired trajectory and activation signals is evaluated over a certain time and a gradient measure of the controller parameters is found using sensitivity models of the controller and the musculoskeletal system. The controller parameters are adapted along the negative gradient. Although some elements of this learning procedure might be present in biological learning, the applied learning process is mainly considered as an optimization procedure. Following the reasoning of Churchland and Sejnowski (1992) that real-life learning is also an optimization procedure, it follows that valid predictions can be made by applying computer learning, given realistic assumptions about the control system (controller + musculoskeletal system).

It was shown in Sect. 3.1 that plausible neural input signals were found using the described neuromuscular control system and optimization method. In particular, triphasic neural input signals, comparable with experimental data, were achieved for a step-like response.

Sensitivity analysis of the trained controller in Sect. 3.2 showed significant gain modulations for both feedforward and feedback signals during voluntary movement. Moreover those sensitivities were found to fluctuate in correspondence with the neural input signals. In the experiments of Kirsch et al. (1993) the time variations of stretch reflex dynamics throughout rapid voluntary changes in the isometric contraction level of the human triceps surae muscles were examined. They found that the stretch reflex gain was significantly modulated during changes in voluntary contraction level, increasing as the subject contracted the muscles and decreasing as the subject relaxed. In particular the time course of the gain changes closely followed the level of the EMG. Since in both the experimental case of Kirsch and co-workers and the simulations in this paper voluntary muscular activation is considered, the gain changes can be compared qualitatively, although in the simulations no isometric restraints were applied. The analogous gain changes thus support the validity of the neuromuscular control simulations. Notice that the often made assumption of constant feedback gains is not valid.

Sensitivity analysis further showed that neglecting delays in the neuromuscular control system leads to significantly different control strategies. Such neglect is therefore wrong if one is eager to make general valid neuromuscular control simulations.

In Sect. 3.3 it was shown that control under the condition of limited or no feedback results in significant co-contraction of the muscle pair. The co-contraction causes

increased joint stiffness. Because of the high joint stiffness the musculoskeletal system is less vulnerable to non-optimal control decisions based on limited feedback only. This can be seen as follows. Consider the neural control signal of a muscle u as composed of a perfect control component u_p and a noise component u_n:

$$u = u_p + u_n \tag{25}$$

The perfect control component u_p consists of a constant co-contraction component and a varying task-dependent component. The noise component u_n depends on the available system information. If the controller has only limited system information, noise u_n is high. The noise component u_n causes a muscle force F_n which produces deviation of the achieved from the desired trajectory. If the joint stiffness $(\partial T/\partial\theta)$ is high, due to a high co-contraction level, the effect of F_n is only small. Although co-contraction requires more energy consumption, it is thus advantageous if u_n is high.

The level of u_n is not only dependent on the input signal set of the controller, but also on the task. It was shown in Sect. 3.3 that learning with a lower reference frequency spectrum results in lower co-contraction levels. This lower frequency spectrum imposes an easier task on the system and therefore smaller co-contraction levels are necessary.

If better feedforward control is made possible by inclusion of internal estimation of the muscular activations, co-contraction levels are reduced. This indicates that addition of more dynamics to the neural control system enables it to form a better inverse dynamics model and so reduce its co-contraction level.

Experimental data of Sainburg et al. (1995) indicate that patients who are functionally deafferented indeed employ co-contraction for an arm movement task which barely requires co-contraction in normal subjects. Nevertheless, the trajectories attained by deafferented patients deviate significantly from the desired one. Experimental results from Bizzi et al. (1978) show that monkeys employ increased co-contraction levels for head movements after deafferentation. The movement trajectories are not strongly affected.

In conclusion, a viable method for learning neuromuscular control under several necessary conditions has been discussed. The results compare well with experimental data. Extension of the applied methods to dynamical neural networks and musculoskeletal systems with more degrees of freedom is theoretically possible and the practical implications are currently being investigated.

Acknowledgements. The author would like to acknowledge Frans van der Helm, Ton van Lunteren, Henk Stassen and the anonymous reviewer for their valuable comments on the manuscript.

Appendices
A Musculoskeletal system

This appendix specifies the one degree of freedom musculoskeletal system as applied in the learning simulations.

A.1 Muscle model

The muscle model is based on the work of Winters and Stark (1985). It contains first-order excitation dynamics, non-linear first-order activation dynamics, a contractile element (CE) with a Hill type force-velocity dependence and a Gaussian force-length dependence, a series elasticity (SE) and a parallel elasticity (PE). The third-order model contains the following states: excitation e, activation a and contractile element length l_{ce}. Its inputs are: neural input u, muscle length l_m, muscle velocity $\dot{l}_m$. The output of the system is muscle force F_m.

$$\mathbf{x} = [e\ a\ l_{ce}] \tag{A.1}$$
$$\mathbf{u} = [u\ l_m\ \dot{l}_m] \tag{A.2}$$
$$y = F_m \tag{A.3}$$
$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{A.4}$$
$$y = g(\mathbf{x}, \mathbf{u}) \tag{A.5}$$

The systems $\mathbf{f}(\mathbf{x}, \mathbf{u})$ and $g(\mathbf{x}, \mathbf{u})$ are defined by the following set of equations:

$$\dot{e} = (u - e)/\tau_{ne} \tag{A.6}$$
$$\dot{a} = (e - a)/\tau \quad \tau = \begin{cases} \tau_{ac} & e \geq a \\ \tau_{da} & e < a \end{cases} \tag{A.7}$$
$$\dot{l}_{ce} = \begin{cases} \dot{l}_m & a \leq \delta \\ F_{vce}^{-1}(a, l_{ce}, l_m) & a > \delta \end{cases} \tag{A.8}$$
$$F_m = F_{pe}(l_m) + F_{se}(l_m, l_{ce}) \tag{A.9}$$

$$F_{vce}^{-1}(a, l_{ce}, l_m) = \begin{cases} \dfrac{V_{sh} v_{max}(a, l_{ce})(F_{vce}(a, l_{ce}, l_m) - 1)}{F_{vce}(a, l_{ce}, l_m) + V_{sh}} & 0 \leq F_{vce} \leq 1 \\ \dfrac{-V_{shl} V_{sh} v_{max}(a, l_{ce})(F_{vce}(a, l_{ce}, l_m) - 1)}{F_{vce}(a, l_{ce}, l_m) + k_{ce1}} & 1 < F_{vce} \leq V_{ml} \end{cases} \tag{A.10}$$

$$F_{vce}(a, l_{ce}, l_m) = \frac{F_{se}(a, l_{ce}, l_m)}{a F_{max} F_{lce}(l_{ce})} \tag{A.11}$$

$$F_{se}(a, l_{ce}, l_m) = \begin{cases} 0 & l_{se} \leq 0 \\ max(k_{se1}(e^{k_{se2} l_{se}} - 1), a\, k_{se3} F_{lce}(l_{ce})) & l_{se} > 0 \end{cases} \tag{A.12}$$

$$l_{se} = l_m - l_{ce} - l_t \tag{A.13}$$
$$v_{max}(a, l_{ce}) = V_{vm}(1 - V_{er} + V_{er} a F_{lce}(l_{ce})) \tag{A.14}$$
$$F_{lce}(l_{ce}) = \exp\left[-\left(\frac{l_{ce} - l_{ce0}}{l_{cesh}}\right)^2\right] \tag{A.15}$$
$$F_{pe}(l_m) = \begin{cases} 0 & l_m \leq l_{pe0} \\ k_{pe1}(e^{k_{pe2}(l_m - l_{pe0})} - 1) & l_m > l_{pe0} \end{cases} \tag{A.16}$$

where $F_{vce}^{-1}(a, l_{ce}, l_m)$ is the inverse force-velocity relation of the CE, F_{pe} is the force exerted by the PE, F_{se} is the force exerted by the SE, F_{vce} is the relative force of the CE due to the force-velocity relation, l_{se} is the length of the SE, v_{max} is the maximum velocity of the CE and F_{lce} is the

Table A1. Parameters of the muscle model

Parameter	Value	Unit
τ_{ne}	0.04	[s]
τ_{ac}	0.01	[s]
τ_{da}	0.05	[s]
δ	10^{-6}	[-]
F_{max}	1000	[N]
l_0	0.2	[m]
l_t	$0.25l_0$	[m]
l_{ce0}	$0.75l_0$	[m]
l_{cesh}	$0.25l_0$	[m]
SE_{xm}	$0.05l_0$	[m]
SE_{sh}	3	[-]
V_{vm}	3	[m/s]
V_{sh}	0.25	[-]
V_{shl}	0.5	[-]
V_{er}	0.5	[-]
V_{ml}	1.3	[-]
l_{pe0}	l_0	[m]
PE_{xm}	$0.4l_0$	[m]
PE_{sh}	3	[-]

Table A2. Parameters of the musculoskeletal model

Parameter	Value	Unit
r_1	0.04	[m]
r_2	0.04	[m]
B_l	0.2	[Nms/rad]
I_l	0.25	[kgm^2]

relative force of the CE due to the force-length relation. All other symbols denote (constant) muscle parameters which are specified in Table A1 and in the following equations:

$$k_{ce1} = -1 - (1 + V_{sh}V_{shl})(V_{ml} - 1) \tag{A.17}$$

$$k_{se1} = \frac{F_{max}}{e^{SE_{sh}} - 1} \tag{A.18}$$

$$k_{se2} = SE_{sh}/SE_{xm} \tag{A.19}$$

$$k_{se3} = F_{max}V_{ml} \tag{A.20}$$

$$k_{pe1} = \frac{F_{max}}{e^{PE_{sh}} - 1} \tag{A.21}$$

$$k_{pe2} = PE_{sh}/PE_{xm} \tag{A.22}$$

A.2 Musculoskeletal model

The muscle model discussed is used as the basis of a one degree of freedom antagonistic musculoskeletal system. The system is defined by the following set of equations:

$$\dot{\mathbf{x}} = \mathbf{f}_s(\mathbf{x}, \mathbf{u}) \tag{A.23}$$

$$\mathbf{y} = \mathbf{g}_s(\mathbf{x}) \tag{A.24}$$

$$\mathbf{x} = [\theta\ \dot{\theta}\ e_1\ a_1\ l_{ce1}\ e_2\ a_2\ l_{ce2}]^T \tag{A.25}$$

$$\mathbf{u} = [u_1\ u_2]^T \tag{A.26}$$

$$\mathbf{y} = [F_1\ F_2\ l_1\ l_2] \tag{A.27}$$

with θ link angle, $\dot{\theta}$ angular velocity, e_i excitation, a_i activation, l_{cei} contractile element length, u_i neural input, F_i force and l_i length of both muscles ($i = 1, 2$). The systems $\mathbf{f}_s(\mathbf{x}, \mathbf{u})$ and $\mathbf{g}_s(\mathbf{x})$ are described by:

$$l_1 = l_0 - r_1x_1 \tag{A.28}$$

$$\dot{l}_1 = -r_1x_2 \tag{A.29}$$

$$\mathbf{z}_1 = [x_3\ x_4\ x_5]^T \tag{A.30}$$

$$l_2 = l_0 + r_2x_1 \tag{A.31}$$

$$\dot{l}_2 = r_2x_2 \tag{A.32}$$

$$\mathbf{z}_2 = [x_6\ x_7\ x_8]^T \tag{A.33}$$

$$\dot{\mathbf{z}}_i = \mathbf{f}_i(\mathbf{z}_i, l_i, \dot{l}_i, u_i) \tag{A.34}$$

$$F_i = g_i(\mathbf{z}_i, l_i) \tag{A.35}$$

$$\dot{x}_1 = x_2 \tag{A.36}$$

$$\dot{x}_2 = (F_1r_1 - F_2r_2 - B_lx_2)/I_l \tag{A.37}$$

with l_0 rest length, $\dot{l}_i$ velocity, r_i moment arm, $\mathbf{z}_i$ states, f_i and g_i system equations of both muscles ($i = 1, 2$), B_l damping and I_l inertia of the link. Table A2 contains the parameter values of the musculoskeletal system.

B Dynamic feedback for second-order Runge-Kutta

A second-order Runge-Kutta integration method is given by:

$$\mathbf{r}_0 = h.\mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{B.1}$$

$$\mathbf{s} = \mathbf{x} + \tfrac{2}{3}\mathbf{r}_0 \tag{B.2}$$

$$\mathbf{r}_1 = h.\mathbf{f}(\mathbf{s}, \mathbf{u}) \tag{B.3}$$

$$\mathbf{x}(k+1) = \mathbf{x}(k) + (\mathbf{r}_0 + 3\mathbf{r}_1)/4 \tag{B.4}$$

The index k has often been omitted for convenience. If the elements of the Jacobian matrices $\nabla_x\mathbf{f}$ and $\nabla_v\mathbf{f}$ are denoted by $\lambda_{ij}(\mathbf{x}, \mathbf{u})$ and $\mu_{ij}(\mathbf{x}, \mathbf{u})$ then the partial derivatives $\partial x_i(k+1)/\partial x_j(k)$ and $\partial x_i(k+1)/\partial u_j(k)$ can be computed by:

$$\frac{\partial s_i}{\partial x_j} = \begin{cases} 1 + \frac{2}{3}h\lambda_{ij}(\mathbf{x}, \mathbf{u}), & i = j \\ \frac{2}{3}h\lambda_{ij}(\mathbf{x}, \mathbf{u}), & i \neq j \end{cases} \tag{B.5}$$

$$\frac{\partial s_i}{\partial u_j} = \tfrac{2}{3}h\mu_{ij}(\mathbf{x}, \mathbf{u}) \tag{B.6}$$

$$\frac{\partial x_i(k+1)}{\partial x_j(k)} = \begin{cases} 1 + h/4(\lambda_{ij}(\mathbf{x}, \mathbf{u}) + 3\sum_{q=1}^{n}(\lambda_{iq}(\mathbf{s}, \mathbf{u})\frac{\partial s_q}{\partial x_j})), & i = j \\ h/4(\lambda_{ij}(\mathbf{x}, \mathbf{u}) + 3\sum_{q=1}^{n}(\lambda_{iq}(\mathbf{s}, \mathbf{u})\frac{\partial s_q}{\partial x_j})), & i \neq j \end{cases} \tag{B.7}$$

$$\frac{\partial x_i(k+1)}{\partial u_j(k)} = h/4(\mu_{ij}(\mathbf{x}, \mathbf{u}) + 3\sum_{q=1}^{n}(\lambda_{iq}(\mathbf{s}, \mathbf{u})\frac{\partial s_q}{\partial u_j}) + 3\mu_{ij}(\mathbf{s}, \mathbf{u})) \tag{B.8}$$

References

Bizzi E, Dev P, Morasso P, Polit A (1978) Effect of load disturbances during centrally initiated movements. J Neurophysiol 41:542–556

Churchland PS, Sejnowski TJ (1992) The computational brain. MIT Press, Cambridge, Mass

Gerdes VGJ, Happee R (1994) The use of an internal representation in fast goal directed movements: a modelling approach. Biol Cybern 70:513–524

Gomi H, Kawato M (1992) Adaptive feedback control models of vestibulocerebellum and spinocerebellum. Biol Cybern 68:105–114

Gorinevsky DM (1993) Modelling of direct motor learning in fast human arm movements. Biol Cybern 69:219–228

Happee R (1992) Goal-directed arm movements. I: Analysis of EMG records in shoulder and elbow muscles. J Electromyogr Kinesiol 2:165–178

Katayama M, Kawato M (1991) A parallel-hierarchical neural network model for motor control of a musculo-skeletal system. Syst Comput Japan 22(6):95–105

Katayama M, Kawato M (1993) Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. Biol Cybern 69:353–362

Kawato M, Furukawa K, Suzuki R (1987) A hierarchical neural-network model for control and learning of voluntary movement. Biol Cybern 57:169–185

Kirsch FK, Kearney RE, MacNeil JB (1993) Identification of time-varying dynamics of the human triceps surae stretch reflex. I. Rapid isometric contraction. Exp Brain Res 97:115–127

Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Networks 6:861–867

Nguyen DH, Widrow B (1991) The truck backer-upper: an example of learning in neural networks. In: Miller WT et al. Neural networks for control. MIT Press, Cambridge, Mass, pp 287–300

Sainburg RL, Ghilardi MF, Poiznar H, Ghez C (1995) Control of limb dynamics in normal subjects and patients without proprioception. J Neurophysiol 73:820–835

Werbos P (1988) Generalization of backpropagation with application to a recurrent gas market model. Neural Networks 1:339–356

Winters JM, Stark L (1985) Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. IEEE Trans Biomed Eng 32:826–839