

ADAPTIVE NEURAL NETWORK CONTROL OF FES-INDUCED CYCLICAL LOWER LEG MOVEMENTS

S.H. Stroeve*, H.M. Franken*, P.H. Veltink* and W.T.C. van Luenen**

*Biomedical Engineering Division, Department of Electrical Engineering, University of Twente, The Netherlands

**Control, Systems and Computer Engineering Group, Department of Electrical Engineering, University of Twente, The Netherlands

Abstract. As a first step to the control of paraplegic gait by functional electrical stimulation (FES), the control of the swinging lower leg is being studied. This paper deals with a neural control system, that has been developed for this case. The control system has been tested for a model of the swinging lower leg using computer simulations. The neural controller was trained by supervised learning (SL) and by backpropagation through time (BTT). The performance of the controller with random initial weights was poor after training with BTT and fair after SL. BTT training of the neural controller with weights, which had been initialized by SL, resulted in good control. Training with BTT thus improved the performance of the controller that initially had been trained by SL. An adaptive neural control system based on BTT has been proposed and partially tested. The controller adapted relatively fast to the change of an important model parameter.

Keywords. Adaptive control; backpropagation through time; biocybernetics; functional electrical stimulation; neural nets; nonlinear control systems; self-adapting systems.

INTRODUCTION

Paralyzed limbs can be moved by electrically stimulating the appropriate muscles. Cyclical movements of the lower leg generated by stimulation of the knee extensor muscles (quadriceps) is being studied as a test case for control of cyclical leg movements induced by functional electrical stimulation (FES) (Veltink, 1991). Figure 1 shows the situation considered. The torque that can be generated by the quadriceps depends in a nonlinear fashion on the angular position and velocity of the lower leg. Together with the nonlinear passive elasticities and gravitational force, this makes the relation between the movements of the lower leg and the stimulus of the quadriceps highly nonlinear.

The quadriceps should be stimulated such, that the lower leg moves cyclically and reaches a maximum reference angle after each swing, during which disturbances may be present. This condition should be met, while minimizing muscle fatigue, so that the movement can be sustained as long as possible. The control system should adapt to changing system parameters, in particular to the maximum

achievable quadriceps torque, which decays due to fatigue.

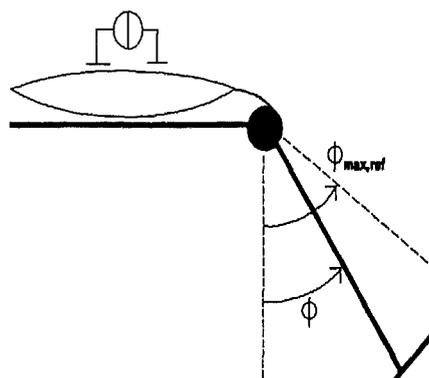


Fig. 1. Stimulation of quadriceps. The lower leg should reach $\phi_{max,ref}$.

METHODS

Model

The electrical stimulation parameters are comprised in one stimulus variable u lying in the interval $[0,1]$, '0' being no and '1' maximal stimulation of the quadriceps. A model that describes the relation between stimulus u and angular position and velocity of the lower leg has been based on literature (Davy and Audu, 1987; Winters and Stark, 1985, 1987; Yamaguchi, 1989). The model, which holds the nonlinear relations mentioned in the introduction, contains three state variables (activation of quadriceps N , angle ϕ and angular velocity ω), one input variable (stimulus u) and two output variables (angle ϕ and angular velocity ω). All model parameters will be assumed constant, except the maximum torque M_{\max} , which decays due to fatigue.

Cost Function

The quadriceps should be stimulated such, that the lower leg reaches a maximum reference angle after each swing, while minimizing muscle fatigue. This goal is comprised in the following cost function:

$$J = (\phi(T) - \phi_{\max,ref})^2 + \alpha \int_0^T u^2 dt \quad (1)$$

, where T is the end time of one cycle, $\phi_{\max,ref}$ is the maximum reference angle and α is a factor which controls the ratio between the two cost components. It is assumed that fatigue is related to the time integral of the quadratic stimulus. The value chosen for $\phi_{\max,ref}$ is 45° .

Neural Network

The use of an artificial neural network (ANN) as a controller for this system has been evaluated. A two layer feedforward network with two inputs, 50 neurons in the hidden layer and one output has been used. No thresholds have been used. The transfer functions are sigmoid functions:

$$\eta(\xi) = \frac{0.3}{1 + \exp(-\xi)} \quad (2)$$

The angular position and velocity of the lower leg are input signals, the stimulus is output. The controller ideally generates the optimal stimulus (which results in minimal costs), that exists for each position-velocity pair. If the optimal mapping of a sufficient large part of state space has been captured by the controller, disturbances, which

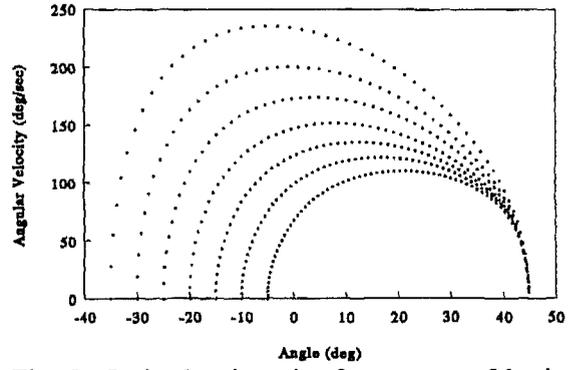


Fig. 2. Optimal trajectories for a range of begin angles. Subsequent dots are 0.01 sec apart from each other. With each position-velocity pair corresponds one stimulus u .

cause a deviation from the undisturbed optimal path, can be compensated for in an optimal way. It follows from the model and the minimal stimulus restriction of the cost function, that no stimulus should occur during the backward swing. Therefore the neural controller needs only to be trained for positive angular velocities. The controller has been trained for trajectories with begin angles in the range -35° to -5° , so that disturbances resulting in deviations of approximately $\pm 15^\circ$ from the undisturbed case are compensated for. Figure 2 shows the optimal trajectories for this range of begin angles. The controller should be able to generate close to optimal stimuli for position-velocity pairs in the dotted area of fig. 2. Two strategies have been used to train the network: backpropagation through time (BTT) and supervised learning (SL).

Backpropagation Through Time (BTT)

The BTT algorithm will be explained for a general discrete system and cost function:

$$\begin{aligned} \underline{x}(k+1) &= \underline{f}(\underline{x}(k), \underline{u}(k)) \\ \underline{x}(0) &= \underline{x}_0 \end{aligned} \quad (3)$$

$$J = l(\underline{x}(N)) + \sum_{k=0}^{N-1} g(\underline{x}(k), \underline{u}(k), k) \quad (4)$$

, where \underline{x} is a n -dimensional state vector, \underline{u} a m -dimensional input vector and the costs are calculated after N timesteps.

Ordered derivatives. The concept of the ordered

derivative as stated by Werbos (1988), will be used here. The conventional partial derivative $\partial J/\partial x_i(k)$ refers to the direct causal impact of $x_i(k)$ on J , while the ordered derivative $\partial^+ J/\partial x_i(k)$ (notice the plus sign) refers to the total causal impact, including both direct and indirect effects. Dynamic feedback is defined as the use of the chain rule for ordered derivatives, in order to calculate the derivatives of J with respect to the system variables. The following recursive relations hold for (3) and (4):

$$k=N: \frac{\partial^+ J}{\partial x_i(N)} = \frac{\partial l(\underline{x}(N))}{\partial x_i(N)} \quad i=1..n \quad (5)$$

$k < N$:

$$\begin{aligned} \frac{\partial^+ J}{\partial x_i(k)} &= \frac{\partial g(\underline{x}(k), \underline{u}(k), k)}{\partial x_i(k)} + \sum_{q=1}^n \left(\frac{\partial^+ J}{\partial x_q(k+1)} \cdot \frac{\partial x_q(k+1)}{\partial x_i(k)} \right) \\ \frac{\partial^+ J}{\partial u_j(k)} &= \frac{\partial g(\underline{x}(k), \underline{u}(k), k)}{\partial u_j(k)} + \sum_{q=1}^n \left(\frac{\partial^+ J}{\partial x_q(k+1)} \cdot \frac{\partial x_q(k+1)}{\partial u_j(k)} \right) \end{aligned} \quad (6)$$

$i=1..n; \quad j=1..m$

BTT algorithm. The BTT algorithm can now be summarized as follows:

++++

REPEAT

- a- Choose begin state.
- b- Calculate trajectory for times $k=0..N$.
- c- Calculate costs J of this trajectory.
- d- Calculate the ordered derivatives $\partial^+ J/\partial x_i(k)$ and $\partial^+ J/\partial u_j(k)$ for $k=N..0$ using (5) and (6).
- e- Calculate the ordered derivatives of the costs with respect to the weights of the neural network ($\partial^+ J/\partial w$) using backpropagation (= dynamic feedback) of $\partial^+ J/\partial u$.
- f- Adapt weights using gradient information $\partial^+ J/\partial w$.

UNTIL Convergence

++++

The weights were adapted using the average of the ordered derivatives $\partial^+ J/\partial w$ over N timesteps and the delta-bar-delta rule of Jacobs (1988).

Notice that when the costs only depend on the state \underline{x} and not on \underline{u} , and no thresholds are used, the gradient information $\partial^+ J/\partial w$ can also be calculated by forward propagating $\partial^+ J/\partial x_i(k)$ instead of backpropagating $\partial^+ J/\partial u_j(k)$. This will not be further examined here.

Supervised Learning (SL)

The training set consisted of paired values of state (input) and optimal stimulus (output). The optimal

stimuli were calculated with an algorithm that resembles the BTT algorithm. After a certain constant begin state has been chosen steps b, c and d of the BTT algorithm were performed, followed by an adaptation of $u_j(k)$ in the negative direction of $\partial^+ J/\partial u_j(k)$, until convergence was reached. After the optimal stimuli had been calculated for a range of states, the weights of the network were adapted by the following procedure: (1) back-propagation of $\partial J'/\partial u$, with $J' = \frac{1}{2}(u - u_{opt})^2$ (u : actual stimulus, u_{opt} : optimal stimulus), to calculate $\partial J'/\partial w$ for all weights of the network; (2) adaptation of the weights following the delta-bar-delta rule. Notice that the main difference between SL and BTT is, that for SL firstly the optimal stimuli are determined (steps a-d) and secondly the weights are adapted (steps e-f), while in the case of BTT training and optimization are performed simultaneously (steps a-f).

RESULTS

Training

The neural controller was trained by BTT and SL. In case of SL the initial weights were randomly chosen, for BTT two cases were tested: random initial weights and initial weights computed by SL.

SL random. Supervised learning of a neural controller with random initial weights resulted in fair control. The total training set consisted of the position-velocity pairs shown in fig. 2 and the corresponding optimal stimuli. Each dataset consisted of five input-output pairs from the total training set. After 250,000 datasets the average costs had been reduced by a factor 10 and remained about constant. Figure 3 shows the performance of the controller after SL. It shows the costs in u (i.e. $\alpha/u^2 dt$) and the reached end angle for several begin angles. These data are also given for the optimal control case. The end angle is within 3° of the optimal end angle, stimulus u is too high for low begin angles and too low for high begin angles. The average deviation in the end angle from the reference angle is 1.83° . The average of the costs in u is $2.93 \cdot 10^{-4}$, which is 3% lower than the average of the costs in u for optimal stimuli ($3.02 \cdot 10^{-4}$). The average of the total costs is $10.6 \cdot 10^{-4}$, which is 250% higher than the average of the minimal costs ($3.05 \cdot 10^{-4}$).

BTT random. Several simulations were performed in which neural networks with random initial weights were trained by BTT for a range of begin angles. In all these simulations the resulting networks performed badly. Stimulus u was high when it should be low and low when it should be

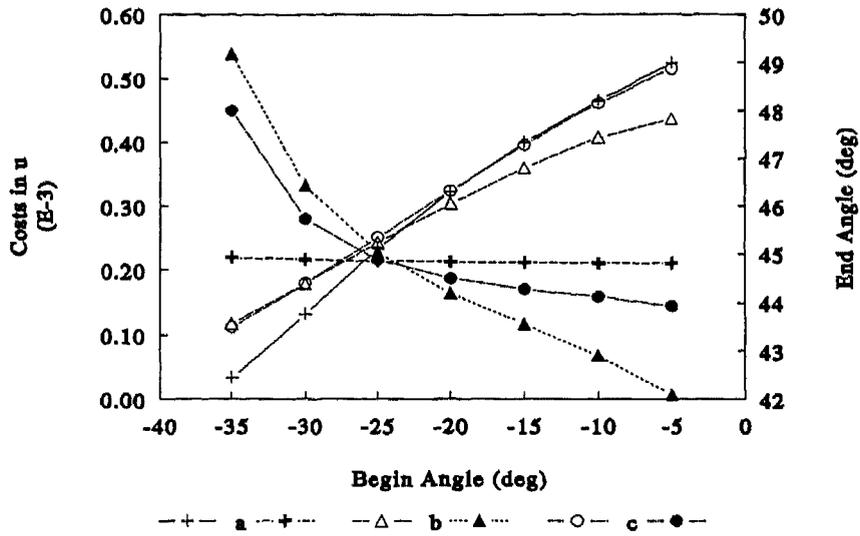


Fig. 3. Performance of controller. The costs in u (non bold) and end angles (bold) for: (a) optimal control, (b) ANN, with random weights, trained by SL random, (c) ANN, with weights initialized by SL, trained by BTT.

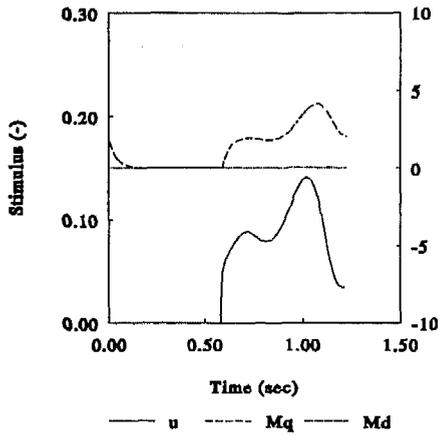


Fig. 4a. Cycle without disturbances (u : stimulus, Mq : quadriceps torque, Md : disturbance torque).

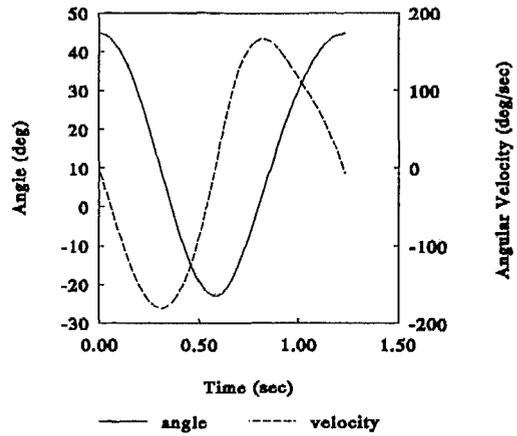


Fig. 4b. Cycle without disturbances (angle and velocity).

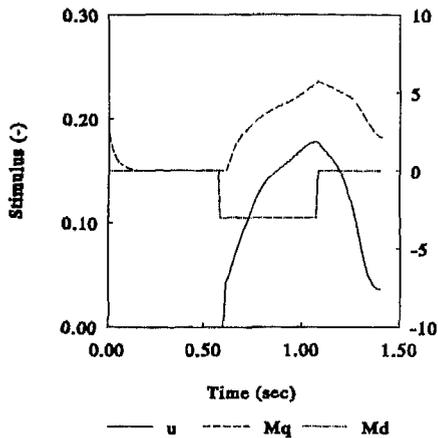


Fig. 5a. Cycle with torque disturbance and low begin angle (u : stimulus, Mq : quadriceps torque, Md : disturbance torque).

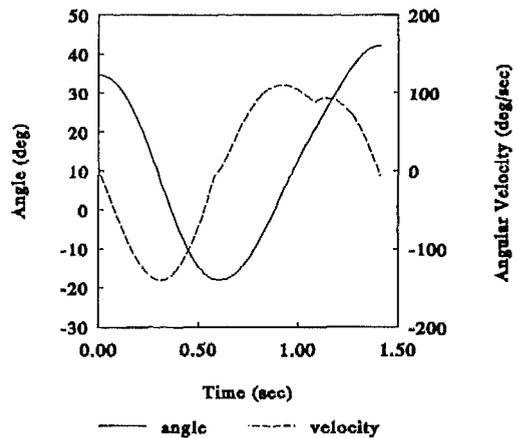


Fig. 5b. Cycle with torque disturbance and low begin angle (angle and velocity).

high. The costs were more than 2.5 times as high in comparison with a network which was trained by SL. Seemingly the network got 'stuck' in a local minimum of the cost criterion and was not able to find the global minimum.

BTT initialized by SL. Training with BTT of the controller, that had initially been trained by SL, improved its performance. The average costs decreased about a factor two after 76,000 iterations of BTT. Figure 3 shows that the end angle is within 1° deviation from the optimal end angle for begin angles from -30° to -5°. Only for a begin angle of -35° the deviation is more: 3°. For this begin angle the stimulus should be very small. Because of the averaging property of the neural network, the stimulus u is too large for this begin angle. The costs of the stimulus J_u are about equal to $J_{u,opt}$ for begin angles larger than -25° and absolutely a little larger for lower begin angles. The average deviation in the end angle from the reference angle is 1.00°. The average of the costs in u is $3.20 \cdot 10^{-4}$, which is 6% higher than the average of the costs in u for optimal stimuli ($3.02 \cdot 10^{-4}$). The average of the total costs is $5.87 \cdot 10^{-4}$, which is 90% higher than the average of the minimal costs ($3.05 \cdot 10^{-4}$). The performance of the controller is satisfactory, since there is only a small deviation of the end angles from the reference angle and of the costs in u from the minimal costs.

Performance of Neural Controller

Figures 4 and 5 show two examples of the behaviour of the closed loop system, containing the controller and the model. The neural controller is deactivated (i.e. $u=0$) for negative angular velocities. The system response for one normal cycle without disturbances is shown in fig. 4. At the end of the cycle the starting point has again been reached. Figure 5 shows that the control system is well capable of compensating for an opposing torque during 0.5 second of the forward swing and a -10° deviation from the normal starting point. The maximum angle is only 3° smaller than the reference angle.

Network Adaptation to Changing Model Parameter

BTT after SL resulted in a better controller, at the cost of a longer training phase. Another advantage of the BTT algorithm is its ability to compensate for changing model parameters directly. This is possible, since a model is used each iteration for the calculation of the gradients $\partial^+ J / \partial w$. In the case of SL a whole new training set would have to be

derived in order to account for changing parameters.

In our application one parameter is known to vary significantly with fatigue: the maximum torque M_{max} . The network adaptation with BTT after a change in M_{max} was therefore investigated.

Step decrease of M_{max} . M_{max} was decreased 10% (from 50 to 45 Nm), which caused the average end angle to decline from 45.0° to 43.5°. After (only) 200 iterations the average costs had decreased 30% and declined almost no more. The average end angle had increased from 43.5° to 44.8°.

Quasi ramp decrease of M_{max} . M_{max} was decreased with 1 Nm per 200 iterations from 50 to 25 Nm. Without adaptation of the controller the average end angle decreased from 45° to 36°. With adaptation the average end angle decreased from 45.0° to 43.5°.

Adaptive Neural Control System

Figure 6 shows a proposal for a configuration of an adaptive neural control system, that is based upon training with BTT. The system consists of two parts: (1) the control and parameter estimation part, (2) the weight adaptation part. The actual controller (neural network 1) is connected in closed loop with the lower leg. Angle information x is fed back to the input of the neural network. The stimulus u is input for both the lower leg and its model. The parameters of model 1 are adapted in response to the difference between the output of the model and the physical system. In subsystem 2 neural network 2 is continuously being trained by BTT for a range of begin states, so as to update the weights. Actual model parameters are frequently communicated from subsystem 1 to 2 and updated weights from subsystem 2 to 1. The frequency may depend on the rate of change of the model parameters.

CONCLUSION

The computer simulations have shown, that the presented adaptive neural control system configuration, based on BTT, might be a very useful one. The neural controller was well capable of controlling the model of the lower leg for a wide variety of torque disturbances, after it had been trained sequentially by SL and BTT. The controller adapted relatively fast to a changing model parameter. Still for our purposes a faster adaptation is desired. Therefore implementation of the weight adaptation process on transputers is being evaluated. The controller has not as yet been tested experimentally on a human leg.

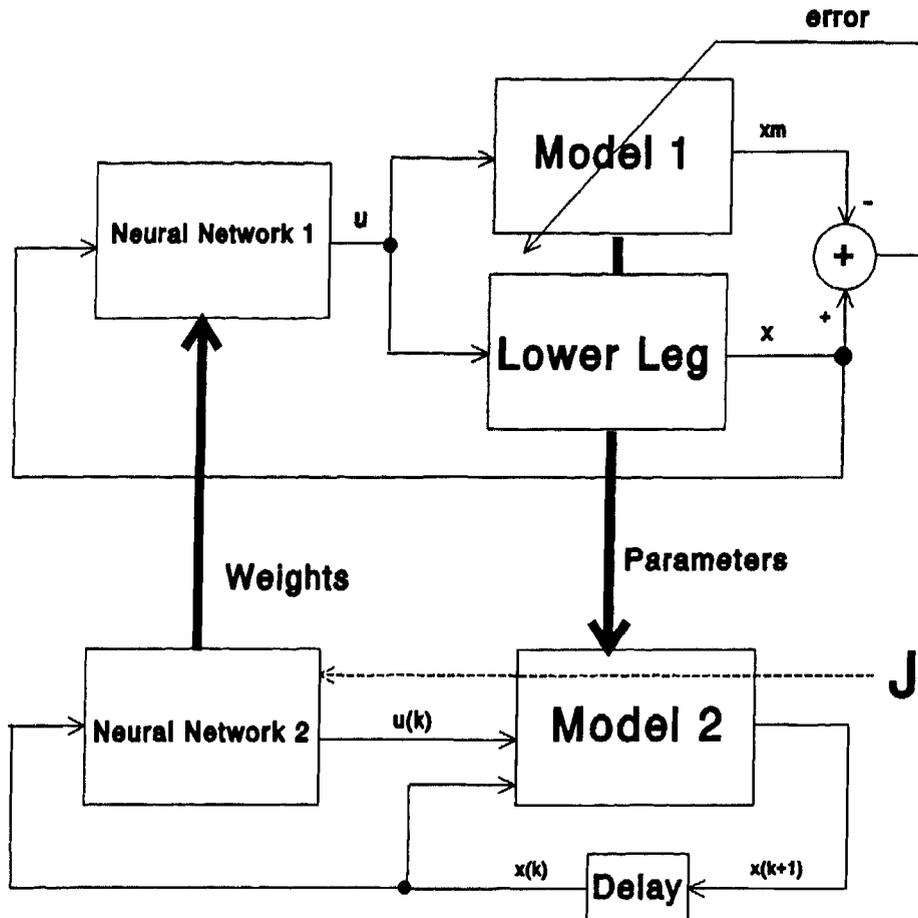


Fig. 6. Adaptive neural control system. See text for description.

REFERENCES

- Audu, M.L. and D.T. Davy (1985). The influence of muscle model complexity in musculoskeletal motion modelling. *J. of Biom. Eng.*, **107**, 147-157.
- Jacobs, G. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks*, **1**, 295-307.
- Nguyen, D. and B. Widrow (1991). The truck backer-upper: an example of self-learning in neural networks. In W.T. Miller, R.S. Sutton and P.J. Werbos (Ed.), *Neural networks for control*, 1st ed. MIT Press, Cambridge, Mass. pp.287-300.
- Veltink, P.H. (1991). Control of FES-induced cyclical movements of the lower leg. *Med. & Biol. Eng. & Comput.*, **29**, NS8-NS12.
- Werbos, P.J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, **1**, 339-356.
- Winters, J.M. and L. Stark (1985). Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. *IEEE Trans. Biom. Eng.*, **32**, 826-838.

Winters, J.M. and L. Stark (1987). Muscle models: What is gained and what is lost by varying muscle complexity. *Biol. Cybern.*, **55**, 403-420.

Yamaguchi, G.T. (1989). A skeletal model for the normal walking gait. In Yamaguchi, G.T., *Feasibility and conceptual design of functional neuromuscular stimulation systems for the restoration of natural gait to paraplegics based on dynamic musculoskeletal models*, PhD thesis, Stanford University, Palo Alto. Chap. 4, pp. 43-69.