A neural network model rapidly learning gains and gating of reflexes necessary to adapt to an arm’s dynamics

K. Th. Kalveram
Abteilung für Kybernetische Psychologie, Heinrich-Heine-Universität, Universitätsstrasse 1, W-4000 Düsseldorf, Federal Republic of Germany

Received February 10, 1992/Accepted in revised form June 30, 1992

Abstract. Effects of dynamic coupling, gravity, inertia and the mechanical impedances of the segments of a multi-jointed arm are shown to be neutralizable through a reflex-like operating three layer static feedforward network. The network requires the proprioceptively mediated actual state variables (here angular velocity and position) of each arm segment. Added neural integrators (and/or differentiators) can make the network exhibit dynamic properties. Then, actual feedback is not necessary and the network can operate in a pure feedforward fashion. Feedforward of an additional load can easily be implemented into the network using “descendent gating”, and a negative feedback control loop added to the feedforward control reduces errors due to external noise. A training, which combines a least squared error based simultaneous learning rule (LSQ-rule) with a “self-imitation algorithm” based on direct inverse modeling, enables the network to acquire the whole inverse dynamics, limb parameters included, during one short training movement. The considerations presented also hold for multi-jointed manipulators.

1 Introduction

Reflex-like processes can invert the dynamics of a two-jointed arm, that is, they can neutralize the effects of dynamic coupling, gravity, inertia and mechanical impedances of the arm segments on movement (Kalveram 1991a, b). The method proposed to control the arm’s dynamics is required to measure the state variables of the controlled system by means of proprioception (in this case angular velocity and position about the shoulder and elbow joint of the arm), and to use these measurements in a feedforward fashion, which can be called “proprioceptive feedforward”. The leading principle of this kind of dynamics control is to enable each limb segment to move independently from the others. In Kalveram (1991b), an analogue model of reflex-like processing applying this principle is outlined, however, no procedure has been given for identifying the parameters, except for a (time consuming) algorithm re-adjusting the model after fastening an additional load to the forearm. The purpose of the present paper is to replace the analogue model by a neural network model with feedforward topology, to specify a learning rule, which enables the network to adapt completely to the arm’s dynamics during a single short movement, and to generalize the considerations to arms with more than two degrees of freedom.

2 Theory

2.1 The task to be learned

The considerations will start using a (nonredundant) arm with two segments introduced in Kalveram (1991b). The movement of the arm is governed by two coupled differential equations:

\[ A\ddot{\theta}_1 + C\dot{\theta}_2 = -D\phi_1 + B\dot{\phi}_2, \]

\[ B\ddot{\theta}_2 + C\dot{\theta}_1 = -E + R_1(\phi_1 - \phi_{01}) = Q_1, \]

\[ -F + R_2(\phi_2 - \phi_{02}) = Q_2, \]

where \( \phi, \theta \) denote angular acceleration, velocity and position of arm segment \( i \) referred to segment \( i - 1 \) (\( i = 1, 2 \)), while \( \phi \) is referenced to body.

\[ A = M_1 + M_2 + m_1l_1^2 + l_1a_2m_2 \cos \varphi_2 \]

\[ B = M_2 \]

\[ C = M_2 + l_1a_2m_2 \cos \varphi_2 \]

\[ D = l_1a_2m_2 \sin \varphi_2 \]

\[ E = g(a_1m_1 + l_1m_2) \sin(\varphi_1 - \varphi_g) \]

\[ + ga_2m_2 \sin(\varphi_1 + \varphi_2 - \varphi_g) \]

\[ F = \]

\[ g = 9.81 \text{ m/s}^2, \varphi_g = \text{direction of gravity referenced to} \]


Fig. 1. Network, learning and controlling the forward dynamics of a two-jointed arm. In switch position 1 (outlined) the network is trained by "self-imitation". The teaching inputs (dashed thin arrows) are used only in this phase. After training, the switches are brought into position 2. Now, the network should be able to specify torques \( Q_1, Q_2 \), such that the actual angular accelerations match the given target accelerations. In both modes, the network relies on the current state (angular position and velocity) of each arm segment, angle \( \phi_e \) of reference to gravity included. See text for more information.

The actual angular accelerations \( \dot{\phi}_1, \dot{\phi}_2 \), velocities \( \phi_1, \phi_2 \), and positions \( \phi_1, \phi_2 \) produced by these torques are measured (proprioceptively) and fed back into the network's input. The blind teacher's signals \( Q'_1, Q'_2 \) are also measured by proprioception, and serve as target outputs to be associated with the inputs by the teaching inputs into the network, and indicated by thin dashed arrows. These teaching inputs are used only in the training phase. After training, the blind teacher is disabled. The outputs of the network are connected to the inputs of the system and the feedback lines conveying the actual angular accelerations are switched off and the target accelerations are switched on (switch position 2 in Fig. 1).

Provided the training is successfully completed the target accelerations \( \dot{\phi}_1, \dot{\phi}_2 \) then offered to the network should equal the actual accelerations attained by the physical system.

In order to perform in this manner, the network again requires the physical state of the system, that means, that the actual angular velocities \( \dot{\phi}_1, \dot{\phi}_2 \), and positions \( \phi_1, \phi_2 \), must continue to be returned to the network.

2.3 The network

Much evidence exists to confirm that the central nervous system, at least at the spinal level, is capable of performing multiplications. Notations like sensory gating or descendent modulation of reflexes (Gossard and Rossignol 1990) mean that the output of a (motor) neuron can be viewed essentially as a product of two or more variables at a presynaptic (premotor) level. In order to map this feature of the nervous system, the "power network" (Kalveram 1992) can be used, that is a three layer feedforward network with fixed synaptic weights in the hidden layer and plastic weights in the output layer. In contrast to ordinary backpropagation networks, the power network represents a set of \( k \times q \)-dimensional polynomials, where \( q \) means the number of input variables and \( k \) the number of output variables. Denoting the maximum order of the polynomials by \( p \geq 0 \), each hidden node computes an expression of the form

\[
x_{1}^r \cdot x_{2}^s \cdot \ldots \cdot x_{q}^t, \quad r, s, \ldots, t: \text{integers} \geq 0 \quad \text{with} \quad r + s + \ldots + t \leq p.
\]  

The output layer neurons of the power network have a linear activation function, and sum the appropriately
Fig. 2. Power network, representing the inverse dynamics of the arm. For clarity, trigonometrical computations have been assigned to an auxiliary "sin/cos-net" located at the input level. The hidden neurons perform multiplications of exponential inputs. Nodes 6 and 7 each use one input powered by 2, while all other exponents are \( 1 \) (however, the number \( 1 \) has been omitted). The neurons of the output layer sum the values put out by the hidden layer using the synaptic weights written at the related terminals. See (1) and text for more information.

\[
    z_j = \sum_{i=1}^{N} w_{ij} y_i, \quad j = 1, 2, \ldots, k, \tag{3}
\]

where \( z_j \) is output of the \( j \)th output neuron, \( y_i \) the output of the \( i \)th hidden neuron, \( N \) the number of hidden neurons respectively of expressions like (2), and \( w_{ij} \) the weight from the \( i \)th hidden to the \( j \)th output neuron. This type of network therefore can also be viewed as representing a set of \( q \)-dimensional power or even Taylor series.

Figure 2 describes a special configuration of such a power network, which is capable of inverting the arm's dynamics. The network has seven input and two output terminals. For simplicity, the network is subdivided into an auxiliary and a main power network. The computation of the trigonometrical functions from the inputs \( \phi_1, \phi_2, \phi_3 \) are assigned to the auxiliary "sin/cos-net". (In the auxiliary power network, \( \sin(x) \) and \( \cos(x) \) can be thought to be approximated – for instance – by four terms, related to the angle \( x \) (exponents 1, 3, 5, 7 resp. \( 0, 2, 4, 6, 8 \), yielding a total number of 17 hidden nodes in the sin/cos-net, the constant term included. The sin/cos-net may principally be incorporated into the main network. Then the number of hidden nodes in the main network would increase, because each hidden node receiving a sin or cos signal would split into four nodes). Using the sin/cos-net separately, the main power network can be seen to have 10 input nodes and 13 hidden nodes, as pointed out in Fig. 2. The numbers written at the synapses of the hidden layer nodes represent the exponents (see (2)). No number means an exponent \( = 1 \). Paying attention, for instance, to the hidden neurons 2 and 4, it can easily be seen, how the target accelerations \( \ddot{q}_1, \ddot{q}_2 \) put out, for instance, by CPGs "modulate" the proprioceptive afferent \( \phi_2 \). The terms written at the connections from the hidden to the output nodes correspond to (1). The accurate values of these terms are to be determined by learning.

Obviously the network has a very simple structure due to the simple structure of (1): Maximum number of inputs to be combined at the hidden level is three (one case), the maximum power is two (two cases) and in eight of the input-output paths the related hidden nodes can, strictly speaking, be omitted.

If the log/antilog representation (Kalveram 1992) should be preferred, then the activation functions assigned to the input and hidden layer are \( -\ln x \) respectively \( e^{-x} \), and the numbers written at the synapses of
the hidden nodes then do not mean exponents, but weights. In this case, however, the inputs into the network require some normalization in order to ensure that the inputs into the input and hidden layer are restricted to finite positive numbers. Though the order of the polynomial approach remains constant, the number of hidden units increases somewhat because of the enhanced number of mixed terms.

2.4 The learning rule

In the training phase, the blind teacher is active and puts torques \( Q_1, Q_2 \) into the physical system. In the course of the generated movement a set of training vectors is then sampled in order to adjust the synaptic weights. The \( n \)th training vector, consisting of input and output values of the physical system taken at the same time, is defined as \( \{ (\phi_1^*, \phi_2^*, \psi_1, \psi_2, \phi_1, \phi_2), (Q_1, Q_2) \} \). In the present case, however, backpropagation and delta rule, commonly used to determine synaptic weights, are inefficient because they are "sequential" (local) learning rules, that is to say each training vector is immediately fed back into the network in order to modify the weights. Therefore, the LSQ-rule (Kalveram 1992), a simultaneous learning rule which demands a block of \( M \) training vectors to be taken into account "at once", will be applied: After sampling the \( n \)th training vector, the input part is offered to the network, yielding a vector \( \{ y_i \} \) of hidden node outputs, where \( i = 1, 2, \ldots, N \) and \( N \) denote the number of hidden nodes. This vector and the applied torques \( Q_1, Q_2 \), are retained in memory and the next training vector is sampled and treated in the same manner. At the end of the training, for each output \( Q_o \) of an output node \( M \) equations are to be solved. They can be written as

\[
\begin{align*}
(y_i) & \* (w_{ki}) = (Q_k), \quad i = 1, \ldots, N; \\
n = 1, \ldots, M; \quad N \leq M; \quad (k = 1, 2),
\end{align*}
\]

where \( (w_{ki}) \) denote the vector of the unknown weights to the output layer referring to output node \( k \). The superscript \( n \) indicates the running number of the training vector. \( \* \* \) means multiplication of the coefficient matrix \( (y_i) \) by the vector \( (w_{ki}) \). Therefore, (4) reflects a system of \( M \) linear equations with \( N \) columns at the left side, and two righthanded sides. For \( N = M \) the weights can be computed solving (4). The implicated learning rule is called "LEQ-rule" (Kalveram 1992). For \( N < M \) the weights are overdetermined. Then the weights must be computed applying a least squares approximation, called LSO-rule (Kalveram 1992). The rank \( R \) of (4) is assumed to be not lesser than the number \( N \) of hidden nodes. This can easily be achieved by selecting a somewhat randomly shaped training movement.

2.5 Adaptation to additional loads

In daily life, the most probable maladaptation arises from an additional load (mass \( m \)) attached to the tip of the arm, as may happen when gripping an unknown object by the hand, lifting it up, and putting it down at another location. In this case, the physical system changes its dynamic behavior, that is, all the parameters \( A \) to \( F \) in (1) are replaced by new values \( A_s \) to \( F_s \) describing the arm plus load. In order to preserve control performance, the inverse dynamics network therefore must be re-adjusted, applying, for instance, the training procedure outlined in Chaps. 2.2 and 2.4. This takes at least one additional movement. If however the load can be measured before the movement starts, the measured value can be fed forward, provided the network has been enlarged appropriately.

Regarding the physical system, it holds \( m_s = m_2 + m, M_s = M_2 + m l^2, \) and \( a_s m_s = a_2 m_2 + l^2 m \). Therefore the parameters in (1) change to

\[
\begin{align*}
A_s &= A + m(l_1^2 + l_2^2) \quad + m l_2 \cos \phi_2 \\
B_s &= B + ml_2 \sin \phi_2 \\
C_s &= C + ml_2^2 \quad + m l_1 \cos \phi_2 \\
D_s &= D \quad + m l_1 \sin \phi_2 \\
E_s &= E + ml_2 \sin(\phi - \phi_2) \quad + gm l_2 \sin(\phi_2 + \phi_2 - \phi_2) \\
F_s &= F \quad + gm l_2 \sin(\phi_1 + \phi_2 - \phi_2).
\end{align*}
\]

Figure 3 outlines the principle to expand the network: The physically induced nine additional terms are compensated for by the same number of hidden nodes (numbered 14 to 22) added to the network, which correspond to these terms. Furthermore, one additional input called "\( m \)" into the hidden layer is provided. With Fig. 2 and Fig. 3 taken together, automatic load compensation is accomplished by feeding the value of \( m \) into the new input. Regarding an unknown object, its mass \( m \) must be assessed, e.g. supplying a ramp shaped variable starting from \( m = 0 \). This will enhance force, until the arm lifts off. Just at this moment, the ramp generating unit must be stopped. Its output now represents the true additional mass and gates the newly provided pathways, thus enabling immediate feedforward compensation of the load. Now the pattern generator can start to execute the planned movement.

In order to train the expanded network, the training vectors must provide the additional input variable \( m \), and the sample of training vectors must refer to a training movement with \( m = 0 \) and \( m > 0 \) (see Chap. 3).

2.6 Performance test

After training, the network's performance can be tested by bringing the switches (see Fig. 1) into position 2 and offering a target trajectory to the input, represented by a sequence or pairs of angular accelerations \( (\dot{\phi}_1, \dot{\phi}_2) \). The training can be regarded as successful if the corresponding actual accelerations match the targets, otherwise the procedure must be repeated using, for instance, another training movement and/or more training vectors.

3 A simulation experiment

In a simulation experiment, the computer model of an arm (see Kalveram 1991b) was used in the training
environment of Fig. 1. Length and mass of both the arm segments were set to 0.5 m resp. 1 kg. Mass was assumed to be equally distributed over the segment length, yielding moments of inertia of $I_1 = I_2 = 0.0833 \, \text{kg} \cdot \text{m}^2$ and distances to centers of masses of $a_1 = a_2 = 0.25 \, \text{m}$. Referring to the two joints $(i = 1, 2)$, the selected values of damping and stiffness were $R_i = 5 \, \text{Nms}$ and $D_i = 5 \, \text{Nm}$. The angular mechanical equilibrium positions and reference to gravity were fixed to zero for simplicity. Using the pre-knowledge of the mathematical structure of the physical system, a power network of 25 hidden nodes was established, 13 of which had input connections and exponents as shown in Fig. 2, nine of those as shown in Fig. 3, while the remaining 3 nodes represented terms combining input variables like $\dot{\varphi}_1, \dot{\varphi}_2, \ddot{\varphi}_1, \ddot{\varphi}_2, \dddot{\varphi}_1, \dddot{\varphi}_2$, or $\dot{\varphi}_1, \dot{\varphi}_2 \cos \varphi_2$. Though abundant, these additional terms serve to demonstrate that all unused terms will drop out when applying the LSQ-rule (see also discussion). The auxiliary sin/cos-net was assumed to be given. At the beginning, all weights of the output layer were left open. The training movement initiated by the blind teacher was similar to a random walk (see Fig. 4.1) and had a duration of about 0.5 s. Equally spaced in time, 30 training vectors were sampled, the first half of which with $m = 0$, the second with $m = 0.05 \, \text{kg}$ attached at the forearm tip. Then the LSQ-rule was applied. The resulting weights of the two output nodes are listed in Table 1. The weights indeed accurately reflect the structure and the parameters implemented in the "physical" arm model.

In order to demonstrate the control performance of the trained network, two performance tests were employed. The first test procedure has already been described in Kalveram (1991b, pp. 69–70): A sequence of eight CPG-controlled aiming movements (period duration $T_0 = 96 \, \text{ms}$) was executed, each movement followed by final damping ($R_f = 50 \, \text{Nms}$). The only modifications are that now the coordinates $x, y$ of the tip of the arm both remained positive during the whole test sequence, and final damping was combined with final negative feedback angular position control (gain $D_f = 500 \, \text{Nm}$). The arm performs the test sequence loaded by the mass $m = 0.1 \, \text{kg}$ fixed at the tip of the

<table>
<thead>
<tr>
<th>Hidden node</th>
<th>$Q_1$</th>
<th>$Q_2$</th>
<th>Hidden node</th>
<th>$Q_1$</th>
<th>$Q_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4167</td>
<td>0.0833</td>
<td>14</td>
<td>0.5000</td>
<td>0.2500</td>
</tr>
<tr>
<td>2</td>
<td>0.1250</td>
<td>0.1250</td>
<td>15</td>
<td>0.2500</td>
<td>0.2500</td>
</tr>
<tr>
<td>3</td>
<td>0.0833</td>
<td>0.0833</td>
<td>16</td>
<td>0.2500</td>
<td>0.2500</td>
</tr>
<tr>
<td>4</td>
<td>0.1250</td>
<td>0.0000</td>
<td>17</td>
<td>0.2500</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>-0.2500</td>
<td>0.0000</td>
<td>18</td>
<td>-0.5000</td>
<td>0.0000</td>
</tr>
<tr>
<td>6</td>
<td>0.0000</td>
<td>0.1250</td>
<td>19</td>
<td>0.0000</td>
<td>0.2500</td>
</tr>
<tr>
<td>7</td>
<td>-0.1250</td>
<td>0.0000</td>
<td>20</td>
<td>-0.2500</td>
<td>0.0000</td>
</tr>
<tr>
<td>8</td>
<td>5.0000</td>
<td>0.0000</td>
<td>21</td>
<td>-4.9050</td>
<td>0.0000</td>
</tr>
<tr>
<td>9</td>
<td>0.0000</td>
<td>5.0000</td>
<td>22</td>
<td>-4.9050</td>
<td>0.0000</td>
</tr>
<tr>
<td>10</td>
<td>5.0000</td>
<td>0.0000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.0000</td>
<td>5.0000</td>
<td>23</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>12</td>
<td>-2.2575</td>
<td>0.0000</td>
<td>24</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>13</td>
<td>-2.4525</td>
<td>-2.4525</td>
<td>25</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

Fig. 4. Performance test of the inverse dynamics network after training using self-imitation and LSQ-rule. 1 Trace of the training movement ($g$: gravitational force). 2 Additional load ($m = 0.1 \, \text{kg}$) descendently compensated for. Both traces are equal to the intended target trajectories. 3 Additional load not compensated for. 4 Without additional load, but under delayed (16 ms) state feedback. a Traces of the sequence of 8 aiming movements. The endpoint of the CPG-controlled part of each movement trace is marked by the related movement number. Thin lined traces refer to subsequent negative feedback control. b Character "ä", drawn by the tip of the arm. See text for more information.
forearm, the load being compensated for “descending- 
ently”. Referred to the related maximum values \( \Delta \phi_{\text{max}} \) 
and \( \phi_{\text{max}} \), the error is defined by the mean relative 
angular position error (PE) and velocity error (VE) 
remaining just when the CPGs have finished their peri-
durations. The yielded trace is outlined in Fig. 4.2a, the errors 
are PE = VE = 0.

In contrast, Fig. 4.3a shows the trace coming out 
the additional load not being compensated for. Now, 
the errors are PE = 0.13 and VE = 0.060, with overall 
movement duration \( T_m \) being 4.2 times greater than in 
the compensated case. The trace in Fig. 4.4a results 
from an unloaded arm, but with delayed (16 ms) feedback 
of angular velocity and position. In this case, the 
errors are PE = 0.4 and VE = 0.012, with \( T^* = 5.4 \) To.

In the second performance test, the target accelerations 
were chosen to draw the character “b” by the tip of the 
idealized arm, that is the arm defined only by its 
kinematic properties, regardless of the forces that 
generate the motion. Movement length was 1.2 s. Under 
the same conditions as in Fig. 4.2a too, no differences 
between the movements of the “real” (Fig. 4.2b) and 
the idealized arm are observable. Figure 4.3b contrasts 
this trace to the movement which results when the 
additional load is not taken into account by the net-
work. The effect of delayed status feedback on the 
unloaded arm is shown in Fig. 4.4b. Obviously, already 
moderately delayed feedback mostly deteriorates per-
formance in all cases.

4 The extension of the theory

4.1 Time delays and proprioceptive state measurements

In biological systems, inevitable time delays in the 
feedback loops must be taken into account, which can 
seriously deteriorate control performance. Furthermore, 
the proprioceptively recorded signals are in general 
functions of the variables \( \dot{\phi} \), \( \phi \), \( \varphi \), and represent 
them in a mixed fashion, not purely. Figure 5 comprises 
the extensions of the model in order to meet these 
requirements. Now, each symbol \( \dot{\phi} \), \( \phi \), \( \varphi \) refers to a 
vector, the number of components of which being equal 
to the joint count of the arm under consideration.

During training, the switches are held in position 1. The 
time delay of the proprioceptively fed back state signals 
is counteracted automatically since the measurements 
of the torques put out by the teacher are also accom-
plished by means of proprioception (in Fig. 5 indicated 
by the delay unit \( \Delta t \)). During training therefore, serious 
problems due to time delays should not occur. The mixed 
signal problem can be solved using a neural 
network which decodes the needed variables online from 
the mixed inputs \( \Phi_1 \), \( \Phi_2 \), \( \Phi_3 \) (an instant representing 
a possible “mixture” is given in Fig. 5). Purporting 
such a “state decoding network” (acceleration included), 
and presuming the time delay of the teacher’s 
output has been compensated for, the inverse dynamics 
network can be trained as described in the Chaps. 2.2, 
2.4 and 3.

However, though optimally trainable, the network 
cannot exert optimal control after training since the 
state variables are still necessary for proper operation, 
own used in feedforward direction. In this case, how-
ever, their delay cannot be compensated for. In order to 
overcome this problem, the pattern generator which 
provides the angular target acceleration vector \( \Phi \), is 
assumed also to provide the related angular velocities 
and positions. That is to say, the pattern generator 
must predict the state variables of the arm from the 
target variables. Since the network is supposed to be 
accurately adjusted, this can be achieved by implement-
ing neural integrators integrating the target acceler-
ations put out by the acceleration pattern generator 
(APG) once or twice. The switches in Fig. 5 being 
brought into position 2, the network then uses the 
predicted state variables instead of the actual state 
feedback. This can be called “control using predicted 
state feedback”, whereby the time delay is ruled out 
and control will be exerted in a pure feedforward 
fashion. Regarding this kind of control, it can no longer 
be decided which of the predictive variables put out by 
the pattern generator is the “master”, and which are the 
“slaves”. In contrast to Fig. 1, in Fig. 5 therefore the 
predictive input variables are treated equally.

Regarding the switches in Figs. 1 and 5, experimental 
evidence is given supporting the existence of such a 
mechanism, namely presynaptic inhibition of afferents 
coming from muscle spindles and tendon organs 
(Rudomin 1990). Even in the frog, reticulospinal path-
ways seem to be able to control the release of sensory 
transmitters in the spinal cord by acting upon interneu-
rns responsible for presynaptic inhibition, thereby en-
able or disabling sensory information to pass (Gonzales et al. 1991). If actual sensory information is disabled, then (more) centrally produced signals conveying, for instance, predicted sensory information, or control signals produced by pattern generators, could be gated to the motoneurons. Experimental evidence for this part of the switching process may also be found in the framework of descendent sensory gating and/or modulation.

4.2 The blind teacher

In the most simple case, the blind teacher may start a genetically determined fixed action pattern. However, he can be blind with respect to the controller’s task, but he must not. In Fig. 5, he is assumed to act as a “clumsy” negative feedback controller with respect to an angular target position, thus preserving the controllability of the arm, though to a lesser degree, also in the training condition. Furthermore: it is possible to replace the simple negative feedback controller by a second control system similar to the first. In this case, each controller can play the role of the blind teacher with respect to the other controller being trained at that moment.

4.3 Error treatment and mixed control

One kind of error arises from a maladjusted controller, another is due to environmental noise acting upon the physical state of the arm. In order to reduce, prevent or to compensate for these errors, the controller must be tied to “reality”. Referring to the outlined switch positions, Fig. 5 gives an example of a mixed control design meeting these requirements. The control design takes advantage of the fact that not all state variables need to be predicted in every case. Referring to the feedforward mechanism, actual measurement of angular position instead of the corresponding predicted value is taken as input into the network. This allows precise feedforward compensation of gravitational forces, also during longer resting periods (\( \dot{\theta} = \theta = 0 \)), where errors may occur due to drifting integrators providing the predicted position values. During an active movement, the feedforward connection of actual positions does not need to be switched over to predicted positions, as long as the delay time remains small enough compared to the highest frequency occurring in the position signal. Now, look at the closed negative position feedback loop laid upon the feedforward control mechanism. Processing of the position error \( d\varphi \) now enables the arm to maintain the attained posture against external transient perturbations. Negative velocity feedback may be added in order to preserve stability (not shown in Fig. 5). Since some types of proprioceptors, e.g. muscle spindles or joint capsules, deliver a composite position/velocity signal, such an afferent may also directly be used, yielding a shorter reaction time. While moving actively, the negative position feedback loop is left uncommitted if no external perturbations occur, since the equilibrium position defined by this loop is shifted according to the position just attained. If perturbated, the system exhibits an automatic error reduction. However, if the individual decides that an error is caused by maladjustment, he should engage the blind teacher and start a re-adjustment procedure. Another possibility is indicated in Fig. 5 by the dashed arrow branching from \( \ddot{\varphi} \). In this case, position error – or more appropriately: acceleration error – can be used to change the synaptic weights immediately (e.g. by the delta rule). This method, however, is applicable only if the fed back signal is not delayed. At last, pay attention to the descendent pathways labelled \( \varphi_1 \) and \( m \). They convey the reference angle to gravity, and the mass \( m \) additionally attached to the forearm, both signalled by appropriate sensory or perceptual mechanisms (see Figs. 2, 3 and Chap. 2.5). Thus, possible disturbing environmental changes can be fed forward in order to prevent errors even before they become manifest.

Obviously, the considerations in the present Chap. 4.3 can easily be associated with ideas and mechanisms already known as “follow-up servo hypothesis” (Merton 1951), “servo assistance” (Matthews 1964), “alpha-gamma co-activation” (Phillips 1969), “adaptive control” (for instance Arbib 1981), “descendent gating of reflexes” (Nashner 1976), “feedback-error-learning” (Miyamoto et al. 1988), “task dynamics” (Saltzman and Kelso 1987) and so on. The purpose of these considerations is to demonstrate, that all the mechanisms mentioned can be integrated into a simple fast adapting neural network model of low level dynamics control.

4.4 Extended degrees of freedom and the redundancy problem

Finally, it should be mentioned that the model proposed for the control of the dynamics of a two-jointed arm can easily be extended also to an arm with three degrees of freedom, represented for instance by the human arm allowed to execute non-restricted shoulder movements, or by the PUMA robot, gripper excluded (the PUMA is an industrial robotic device, established with a human-like arm with two segments movable in a vertical plane. The “shoulder” is mounted on the top of a standard turnable round a vertical axis). Also, a multi-jointed arm, the movements of which being restricted to a subspace (e.g. a plane) with lesser dimensions than joint counts, is included. This is due to the fact that in all cases, the forward dynamics of the system to be controlled can be expressed by three or more coupled differential equations from the same type as (1):

\[
Q_i = \sum_{k=1}^{n} M_{ik}(\varphi)\ddot{\varphi}_k + \sum_{k,m=1}^{n} \sum_{i} M_{ikm}(\varphi)\dot{\varphi}_i \dot{\varphi}_k + M_i(\varphi),
\]

\[
\varphi = (\varphi_1, \ldots, \varphi_n, \varphi_\theta),
\]  

where \( (M_{ik}) \), \( (M_{ikm}) \), \( (M_i) \) are generalized mass matrices, besides of \( \varphi \) only dependent from the arm’s geometrical structure and the distribution of mass along the arm segments (see for instance Shahinpoor 1987 p. 289 and 295). For the PUMA \((n = 3)\), that is gripper excluded, a typical component of such a matrix is
$M_{12} = c \sin(\varphi_2 + \varphi_3)\cos(\varphi_2 + \varphi_3)$, where $c$ is a mass and structure dependent constant. Compared to (1), only the number of terms is considerably greater, which demands provision of more input, hidden, and output nodes in the network, especially more hidden nodes representing the non-linearities. The problem of controlling the singularities of an arm with redundant degrees of freedom does not occur in this approach: This problem can be treated separately from the dynamics control, because it has been shifted to the kinematics controller, which in turn controls the pattern generator providing the angular target signals to be fed into the inverse dynamics network.

5 Discussion

(1) The presented considerations exhibit that a power network, in conjunction with the simultaneous LSQ (LEQ) training rule, is capable of acquiring the accurate inverse dynamics of an arm. This calls for only one short training movement, if the network is embedded into a self-imitation training scheme. The network can then optimally apply the knowledge thus obtained, if predicted (proprioceptive) feedback of angular position and velocity of each arm segment is available during an active movement. In this case, control is exerted in a pure feedforward fashion. The principle of control can be viewed as an implicit solution of the coupled differential equations of the arm, performed in time with the ongoing movement at a low level of control, whereby the power network provides the accurate parameters and the "patch board", and the pattern generator the integrators. The learnt ability encloses fast aiming movements as well as slow trajectory tracking movements or maintenance of posture. The considerations also include systems with several, even redundant, degrees of freedom, because redundancy control is shifted to the kinematics controller.

(2) In the simulation experiment, the preknowledge of the mathematical structure of the physical system leads directly to the restricted structure of the controlling network. In a theoretical sense, however, this preknowledge is not a prerequisite to the learning performance of the model. The power network is an all purpose function approximator, due to the property of the power respectively Taylor series the network represents. Therefore, if the mathematical structure is unknown, also the higher order and mixed order terms in the series (with respect to the preselected highest exponent values) must be included at the beginning. This means that the number of hidden nodes representing these higher and mixed order terms may become exorbitant, exceeding the capacity of our technical equipment currently available. However, in natural networks, limitations in the computational capacity need not to be taken into account. Therefore, if available, a LSQ-like learning-rule will discard all connections in the network which are related to unused, or almost unused, terms in the physical system, like the expressions 23, 24 and 25 in the simulated network (see Chap. 3), thereby only retaining the most important synapses. This is in line with observations showing that networks in newborns often are established with a great number of synapses, which however are rapidly reduced, probably due to early learning.

(3) Training of the inverse dynamics network using an "error-feedback-learning" scheme (Miyamoto et al. 1988) cannot be as effective as using self-imitation. This may be derived from general system's theory, the output of a system thereafter depends on its input and its actual state. Conversely, in order to learn the rule by which the output is associated with the input, correct examples of input, output and state combinations must be offered, at least on average. This applies also to a neural network, because nobody can re-create lost information. Using error-feedback-learning, full information cannot be provided at the beginning of the training, neither by the network yet to be trained, nor by the "clumsy" error feedback controller, nor by both taken together. The only source of the complete information is the system to be controlled itself. Nevertheless, in an error feedback control scheme the error can converge against zero (Kawato 1989), especially if, besides the target angles, also the first and second derivatives of the targets are offered to the network. In this case, the control signal together with its derivatives will predict the system's actual state the better, the longer learning proceeds. However, it can be expected that it will take a lot of time and hundreds of training movements, or thousands of training vectors to overcome, if at all, the lack of information, in this case imposed by the method. Regarding biological systems, another handicap of error feedback learning is that time delays in the reflex loops cannot be compensated for since the network operates actively while being trained. Therefore, under delayed sensory feedback, the feedback error is biased, which may mislead the learning progress from the beginning.

(4) Concerning biological relevance, some inevitable conditions influencing control performance, such as delays in the reflex loops, or the mixed state representation in the proprioceptive signals, or additional unknown loads, have been taken into account by extending the model. Another problem arises from fluctuations of the proprioceptive signals, which, however, can be overcome by sampling a number of training vectors considerably greater than absolutely necessary, before applying the LSQ-rule. This method can be expected to operate like a low pass filter ruling out (high frequency) erroneous fluctuations.

A serious question is, how the LSQ-rule could be implemented in the nervous system. Providing appropriate time constants for changes of the synaptic weights of the output layer (Miyamoto et al. 1988) possibly may contribute to solve this problem, however, the question must be left open in the present paper. Nevertheless, two points are worth mentioning at the end of the paper: (a) The backpropagation rule, frequently used in theoretical biology, also suffers from lack of physiological evidence. (b) A simultaneous learning rule can be four orders of magnitude more
efficient than sequential rules like the delta rule or backpropagation (Kalveram 1992). Therefore, evolution should strongly prefer those individuals capable of applying simultaneous learning, even if the number of training vectors operated at one time is small.

Referring to this background, the present paper describes a principle, how the obviously difficult problem, namely to implement the inverse dynamics of an individual's own limbs into the nervous system, may accurately be solved, using known properties of the nervous system, combined with relatively simple models of neural networks.

References


Kalveram K Th (1990) A neural network acquiring the inverse of a tool or limb transformation by "self-imitation learning". Poster at the ICNC-10th Cybernetic Congress of the DGK, Duesseldorf. (Reprint available on request from the author)


