Biol. Cybern. 64, 447–454 (1991)

Biological Cybernetics

A learning network model of the neural integrator of the oculomotor system

D. B. Arnold and D. A. Robinson

Departments of Biomedical Engineering and Ophthalmology, The Johns Hopkins University, School of Medicine, Baltimore, MD 21205, USA

Received August 14, 1990/Accepted in revised form December 18, 1990

Abstract. Certain premotor neurons of the oculomotor system fire at a rate proportional to desired eye velocity. Their output is integrated by a network of neurons to supply an eye positon command to the motoneurons of the extraocular muscles. This network, known as the neural integrator, is calibrated during infancy and then maintained through development and trauma with remarkable precision. We have modeled this system with a self-organizing neural network that learns to integrate vestibular velocity commands to generate appropriate eye movements. It learns by using current eye movement on any given trial to calculate the amount of retinal image slip and this is used as the error signal. The synaptic weights are then changed using a straightforward algorithm that is independent of the network configuration and does not necessitate backwards propagation of information. Minimization of the error in this fashion causes the network to develop multiple positive feedback loops that enable it to integrate a push-pull signal without integrating the background rate on which it rides. The network is also capable of recovering from various lesions and of generating more complicated signals to simulate induced postsaccadic drift and compensation for eye muscle mechanics.

Introduction

Eye-movement commands are encoded in certain premotor neurons projecting to the caudal pons as eye-velocity commands. The most obvious of these is the signal from a push-pull pair of semicircular canals that sense angular head velocity and use it as an eye-velocity command to execute the vestibulo-ocular reflex. The discharge rates of eye-muscle motoneurons, however, modulate with eye position as well as velocity. This prompted Robinson (1968) to postulate the existence of a network of neurons capable of temporal integration in the sense of Newtonian calculus. For horizontal deviations of the eyes this network was found to reside in the rostral medial vestibular nuclei and the nuclei prepositus hypoglossi (NPH). See Robinson (1989) for a review.

Several models have been put forward to explain how a group of neurons could perform temporal integration using positive feedback. This task is complicated because eye velocity signals ride on top of a constant background firing rate that is not integrated. In addition, the time constant of an individual neuron, roughly 5 ms, must be increased four orders of magnitude for the network to have the experimentally observed time constant (about 20 s). This could be accomplished, by using very precise amounts of positive feedback (Kamath and Keller 1976), however this made the models extremely sensitive to perturbations.

Cannon et al. (1983) and Cannon and Robinson (1985) developed a hard-wired, lateral inhibitory network, that could integrate a push-pull input signal without integrating the background rates and, by distributing feedback over hundreds of synapses, was relatively impervious to small perturbations. This network also had the appealing property that localized lesions produced a decrease in the time constant of the entire network. This is a phenomenon that is observed clinically and can only be produced with a model where integration occurs as a process distributed over many neurons.

The main limitation of this model, referred to subsequently as the Cannon model, is that its synaptic weights were explicitly specified; it was not designed for synaptic learning. There is ample physiological evidence that the neural integrator is capable of learning. Developmental studies by Weissman et al. (1989) showed that the integrator calibrates itself during the first months of life. It also participates in compensating for a hemilabyrinthectomy (Fetter and Zee 1988). In addition, the integrator can be altered through learning to produce post-saccadic ocular drift in humans (Kapoula et al. 1989). Another drawback to the hardwired model is that it does not lend itself easily to producing the great variety of signal strengths seen in individual neurons in physiological experiments. Learning networks, on the other hand, naturally produce the sort of vari-

ability one sees in single cell recordings. Finally, although Cannon's model was relatively impervious to changes in individual synaptic weights, it still required the network to hold global parameters very precisely, which seems unrealistic without some means of self-correction. Consequently, we modeled the neural integrator with a neural network capable of learning.

Methods

Configuration of the network

The basic network consists of a push-pull input from two canals, a variable number of interneurons, and two motoneurons (Fig. 1). One of the inputs to each interneuron, with membrane potential $y_i(t)$, is the sum of the activities of each of the interneurons, including itself, $f(y_j)$, weighted by the appropriate synaptic strength w_{ij} (Fig. 2). Each interneuron also receives both canal input signals, u_k, $k = 1, 2$, weighted by the synaptic strengths v_{ik}. Finally, each receives a tonic input equivalent to 100 spikes/s to correspond to the typical value seen experimentally.

The weighted sum of all these inputs is then passed through a first-order lag. Thus,

$$\dot{y}_i \times \tau + y_i = \sum_{j=1}^{j=N} w_{ij} \times f(y_j) + \sum_{k=1}^{k=2} v_{ik} \times u_k(t) + 100 \quad (1)$$

where N is the number of interneurons. The membrane time constant, τ, is 5 ms. The activity of each neuron, $f(y_i)$, is a linear function of the membrane potential, y_i, except that it is not allowed to go negative:

$$f(y_i) = y_i \quad \text{if} \quad y_i \geq 0 \quad (2)$$
$$f(y_i) = 0 \quad \text{if} \quad y_i < 0$$

A linear function was chosen because vestibulo-ocular interneurons are remarkable for their linear behavior.

The input to each motoneuron, m_i, $i = 1, 2$, is the weighted sum of the activity of each interneuron, $f(y_j)$, with strength z_{ij} and each has a tonic input of 100.

$$m_i = \sum_{j=1}^{j=N} z_{ij} \times f(y_j) + 100 \quad (3)$$

The activity of each motoneuron is equal to its input. Finally, in early versions the plant dynamics (muscles and orbital tissues) are ignored so that the network is asked only to integrate. Accordingly, eye position, $e_b(t)$, is taken to be the difference between the firing rates of the two motoneurons.

$$e_b(t) = m_1(t) - m_2(t) \quad (4)$$

This will be called the *basic* network.

In a later version, the oculomotor plant is included. Orbital mechanics are simulated by passing the difference between the outputs of the two motoneurons through a first-order lag with a time constant, τ_m, of 200 ms (Fig. 1).

$$\dot{e}_m(t) \times \tau_m + e_m(t) = m_1(t) - m_2(t) \quad (5)$$

This will be called the *modified* network. In both cases the resulting eye position, $e(t)$, is used to calculate the error. In our study we simplify the plant to one cyclopean eye driven by bilateral motor nuclei.

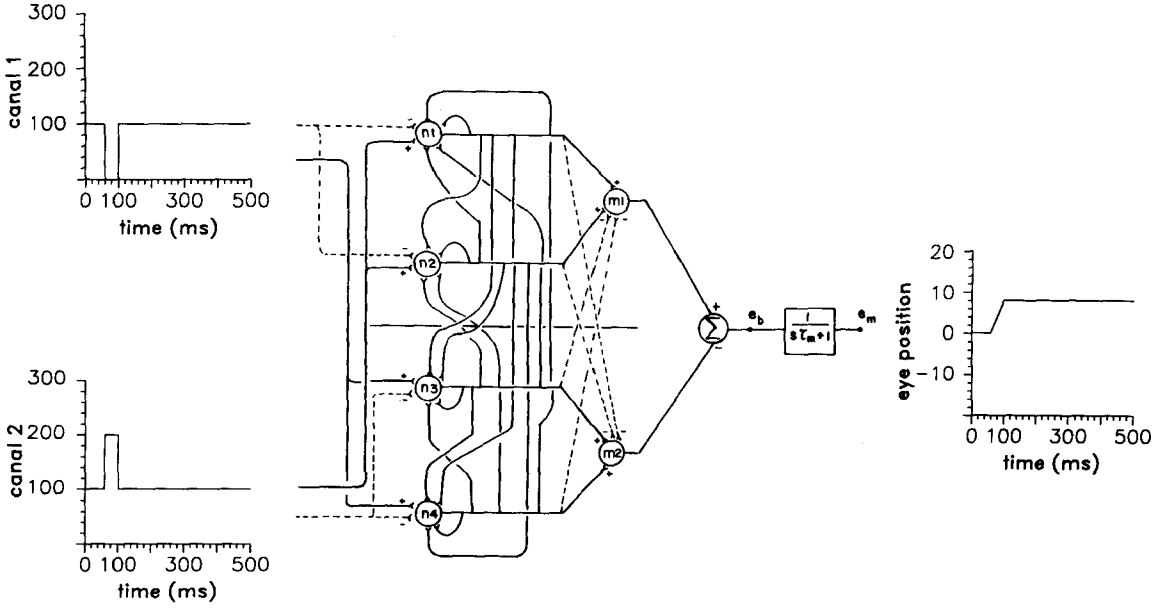

Fig. 1. Configuration of the basic and modified networks. Two canal inputs consisting of a rectangular pulse superimposed on a constant background rate project to each interneuron ($n1$–4). Each interneuron projects back to itself, to every other interneuron, and to both motoneurons. Eye position for the basic network, ($e_b(t)$), is the difference between the outputs of the two motoneurons. Eye position for the modified network ($e_m(t)$) is obtained by passing the difference between the outputs of the motoneurons through a first order lag with a time constant, τ_m, of 200 ms. Dashed input and output connections are inhibitory

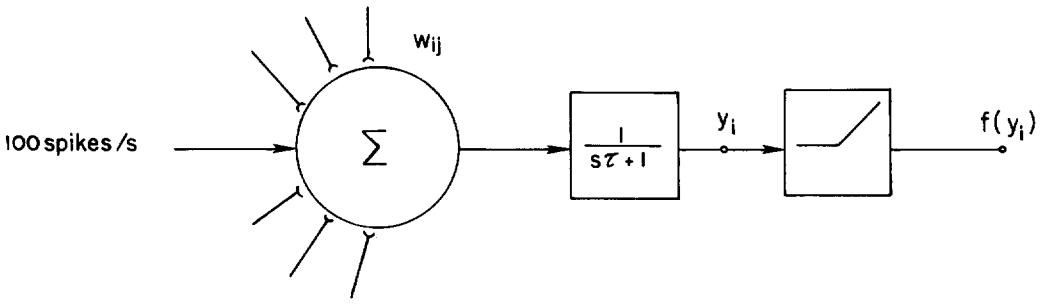

Fig. 2. The transfer function of a neuron. Each neuron receives the weighted sum of the outputs of every neuron projecting to it, as well as a tonic input of 100 spikes/s. This sum is passed through a first order lag with a time constant of 5 ms to give the membrane potential y_i, and then through a rectifying function to give the activity of the neuron, $f(y_i)$

Learing algorithm

Clearly, a network capable of integrating a time-varying signal must have recurrent connections. This precludes the use of the traditional back-propagation algorithm of Rumelhart et al. (1986). Although, recurrent algorithms based on back-propagation do exist (e.g. Pearlmutter 1989), we chose to use a simpler algorithm that will work with any network configuration and does not require information to travel across synapses in two directions.

Our input signals were time-varying canal afferent responses to head movements (Fig. 1). We chose rectangular velocity pulses superimposed on a background rate of 100 spikes/s. For the basic network the rectangular pulses were 100 spikes/s in magnitude, 40 ms in duration, and began 60 ms into the trial period of 500 ms. To accommodate the 200 ms time constant of the plant, the periods for the modified network were increased to 700 ms in length and had rectangular pulses of 50 spikes/s which began at 500 ms and lasted 100 ms. These inputs were presented to the network which produced a time-varying eye position trace, $e(t)$ from (4) or (5). The velocity of the eye with respect to the world, or gaze velocity $\dot{g}(t)$, was then calculated by adding the eye velocity to the head velocity. Since the world is taken to be stationary, $\dot{g}(t)$ equals the rate at which images slip across the retina. This is the physiological error signal believed to set up and recalibrate brainstem oculomotor circuits. It is seen in the retina of subprimates as direction-selective neurons as well as in the accessory optic system, inferior olive, and flocculus of most vertebrates.

The error, E, was calculated as the root mean square of the gaze velocity over the period of interest.

$$E = \left(\int_{t=t_0}^{t=t_1} (\dot{g}(t))^2 \mathrm{dt} \right)^{1/2} \tag{6}$$

basic network: $t_0 = 50$ ms, $t_1 = 500$ ms

modified network: $t_0 = 400$ ms, $t_1 = 700$ ms

The period $0-t_0$ allowed the system to settle into steady state from its initial conditions.

The network started with small random weights (between -0.01 and 0.01 for the basic network and between -0.1 and 0.1 for the modified network, although the exact range is not critical). The initial error, E, was calculated with the random weights. One of the weights, w_{ij} was then changed by a small amount, Δw_{ij}, and the corresponding error, $E(w_{ij} + \Delta w_{ij})$, was calculated. The partial derivative of the error with respect to that weight was approximated by:

$$\frac{\partial E}{\partial w_{ij}} \approx \frac{E(w_{ij} + \Delta w_{ij}) - E}{\Delta w_{ij}} \tag{7}$$

This evaluation was repeated for each weight in the network (including the v and z weights). The weights were then simultaneously updated using the following delta rule:

$$w'_{ij} = w_{ij} - k(E) \times \frac{\dfrac{\partial E}{\partial w_{ij}}}{\left| \dfrac{\partial E}{\partial w_{ij}} \right|_{\max}} \tag{8}$$

where $\left| \dfrac{\partial E}{\partial w_{ij}} \right|_{\max}$ is the maximum magnitude of all the partial derivatives found in this iteration and w'_{ij} are the updated weights.

Normalization by the denominator allows $k(E)$ to determine the maximum value by which any weight can be changed, which prevents unanticipated large changes that can lead to instability. Initially $k(E)$ was set to 0.015 for the basic network and 0.050 for the modified network and was updated after each iteration as follows:

$$k'(E) = \frac{E_2}{E_1} \times k(E) \quad \text{if} \quad E_2 < E_1 \tag{9}$$

$$k'(E) = k(E) \quad \text{if} \quad E_2 \geq E_1$$

where $k(E)$ is the value of k before the weights are updated when the error is E_1, and $k'(E)$ is the value of k after the weights have been updated when the error is E_2. Throughout, Δw_{ij} was assigned half the value of $k(E)$.

This cycle of presentation of the input and calculation of the new weights was repeated until the error was below 0.02, because such a network generally had a time constant greater than 20 s. The time constant was calculated by measuring the eye position just after the velocity pulse, e_p, and then letting the simulation run for 20 s at which time a second measurement of eye

position, e_{20}, was taken. These two measurements gave the time constant via,

$$T = \frac{20}{\ln\left(\frac{e_p}{e_{20}}\right)} \tag{10}$$

Results

Trials with four interneurons

The first experiments were done using only four interneurons. In early trials each weight was allowed to be excitatory or inhibitory and it was found that the resulting configurations had random distributions of excitatory and inhibitory weights. This is contradicted by experimental results of Baker and Berthoz (1975), who found that stimulation of the vestibular nerve produces inhibition of the ipsilateral NPH and excitation of the contralateral NPH. Thus, in later trials, we separated the interneurons into two equal bilateral groups and required ipsilateral input weights to be inhibitory, contralateral input weights to be excitatory and vice-versa for output weights. This was done by setting a weight to zero if it had the wrong polarity after updating.

This network generally converged to an acceptable solution in about 500 iterations. The initial and final configurations of a typical network are shown in Table 1. The final networks integrate by using positive feedback, because that is the only mechanism available to them. There are so many pathways through which a unit can excite itself it is difficult to attach significance to each individual weight; however, one general pattern can be discerned: the connections between interneurons, which could be either excitatory or inhibitory, ended up in a "push-pull" configuration. That is, in 20 trained networks, each ipsilateral w weight ended up excitatory and each contralateral w weight ended up inhibitory. Interestingly, this lateral inihibitory configuration is reminiscent of the Cannon network.

A typical input waveform and the corresponding output waveform of a trained network are shown in Fig. 3A and B. Figure 3C-F shows the responses of the interneurons, each of which reflects the output of the network, the integrated signal, with only a difference in background rate and gain (change in unit output/change in eye position). Neurophysiological experiments have revealed similar behaviour in the region of the neural integrator. Cells either carry the integrated (eye position) signal or they do not; no cells carry partially integrated signals. It is interesting to note that it is only the output of the motoneurons that is used to calculate the error. There is, consequently, no obvious reason for each interneuron to carry nothing but the fully integrated signal and yet in every trial this was the case.

Although the network was trained on only one input waveform, because its defining equations (1 to 5) are linear (as long as the membrane potential is positive) it will integrate any time-varying signal. For instance, we showed that the network will produce a cosine wave in response to a sine wave.

Recovery from lesions

The purpose of this study was to create a model that was capable of self-organization. Evidence that the real

Table 1. Initial and final weights of a basic network. Input weights correspond to connections between the two canals ($c1$ and $c2$) and the four interneurons ($n1$–$n4$). Each ipsilateral weight is inhibitory, while each contralateral weight is excitatory. Recurrent weights correspond to connections between interneurons ($n1$–$n4$). Although initially each weight has an equal chance of being excitatory or inhibitory, the final ipsilateral weights are all excitatory, while the final contralateral weights are all inhibitory. Output weights correspond to connections between interneurons and the two motoneurons ($m1$ and $m2$). Each ipsilateral output weight is excitatory, while each contralateral output weight is inhibitory

Input weights

to		$n1$		$n2$		$n3$		$n4$	
		initial	final	initial	final	initial	final	initial	final
from	$c1$	−0.0099	−0.026	−0.0099	−0.029	0.0026	0.0048	0.0062	0.0039
	$c2$	0.0028	0.0055	0.0098	0.0062	−0.0071	−0.023	−0.0028	−0.017

Recurrent weights

to		$n1$		$n2$		$n3$		$n4$	
		initial	final	initial	final	initial	final	initial	final
from	$n1$	−0.0066	0.16	0.0057	0.25	0.0051	−0.20	0.0039	−0.19
	$n2$	−0.0022	0.24	0.0016	0.35	0.0051	−0.20	−0.0054	−0.27
	$n3$	0.0029	−0.24	−0.0007	−0.35	−0.0050	0.22	−0.0020	0.20
	$n4$	−0.0037	−0.22	−0.0018	−0.32	−0.001	0.20	0.0093	0.19

Output weights

from		$n1$		$n2$		$n3$		$n4$	
		initial	final	initial	final	initial	final	initial	final
to	$m1$	0.0029	0.018	0.0098	0.045	−0.0074	−0.050	−0.0044	−0.048
	$m2$	−0.0008	−0.039	−0.053	−0.066	0.0035	0.024	−0.0084	0.029

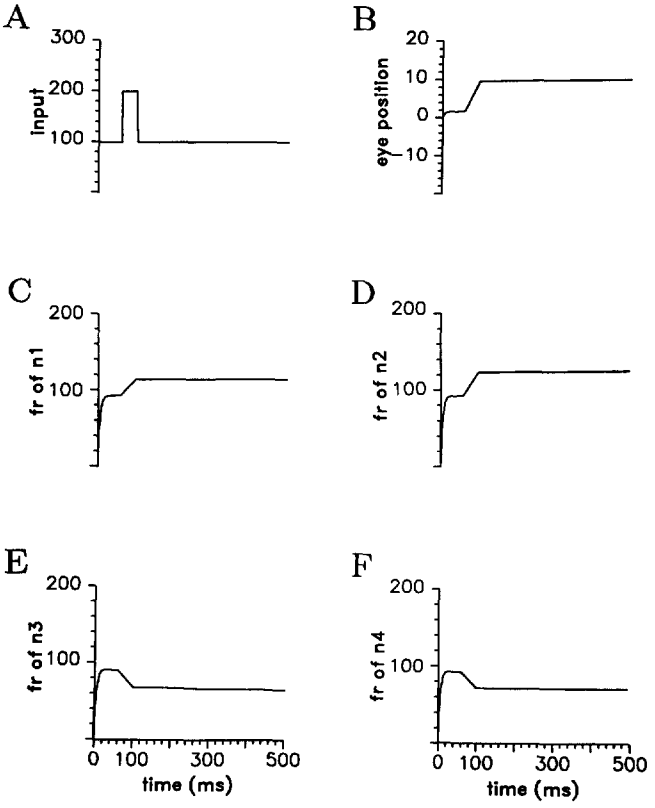

Fig. 3 A–F. Response of a four interneuron basic network to a rectangular pulse of head velocity. **A** One of the push-pull canal afferent input signals. **B** Eye position generated in response to the input signals. **C-F** The responses of the interneurons to the input signals, where *fr* is firing rate in spikes/s are *n*1–*n*4 are the interneurons 1–4

neural integrator has this property is partly based on the fact that it is able to recover from certain lesions (Fetter and Zee 1988). To demonstrate that our network also had this ability several lesions were simulated from which the network recovered. The first such simulation involved "killing" one neuron by zeroeing all its output weights. A network with 16 interneurons was used for this, because eliminating one neuron in a four-neuron model would obviously have a devastating effect. In its original random configuration the network had a time constant of 15 ms which grew to 20 s after 500 iterations (see Fig. 4). Killing a neuron reduced the time constant to 150 ms. The network then recovered after 500 more iterations.

The decrease in time constant due to the lesion is surprisingly large and is at odds with the results of a similar lesion in the model of Cannon et al. (1983). That model was hard-wired and achieved positive feedback by lateral inhibition similar to the present model. Removal of one cell out of 32 caused only a minor deterioration in integrator performance. The reason is that in their model lateral inhibition was largely confined only to neighboring cells. Thus, loss of a cell caused less integrator action in only a small section of the model leaving the rest of it to behave normally. In our model, all cells are allowed to, and do, project

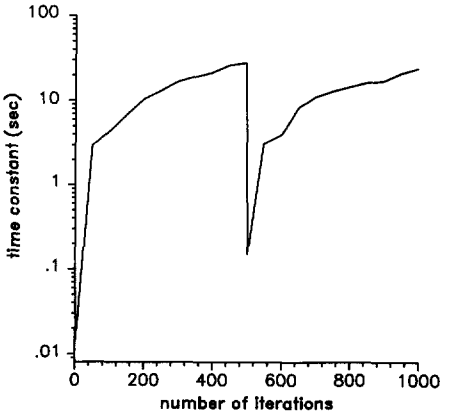

Fig. 4. Time constant of a 16 interneuron network during learning starting from a random configuration and after a lesion to one of the interneurons at 500 iterations

significantly to all other cells in the model so that loss of one cell interrupts many of the loops by which all cells in the model feed back upon themselves. Although our model can repair itself and so achieve robustness, its sensitivity to lesions suggests that the network is too distributed and possibly could be improved by requiring cells to give stronger synaptic inputs to, in some sense, nearby cells and less to more remote cells.

Hemilabyrinthectomy was simulated with four interneurons by removing the inputs from one of the canals. As can be seen in Fig. 5A, the outputs of the interneurons became grossly imbalanced. If the network were able to make saccades to reset the eye position, the

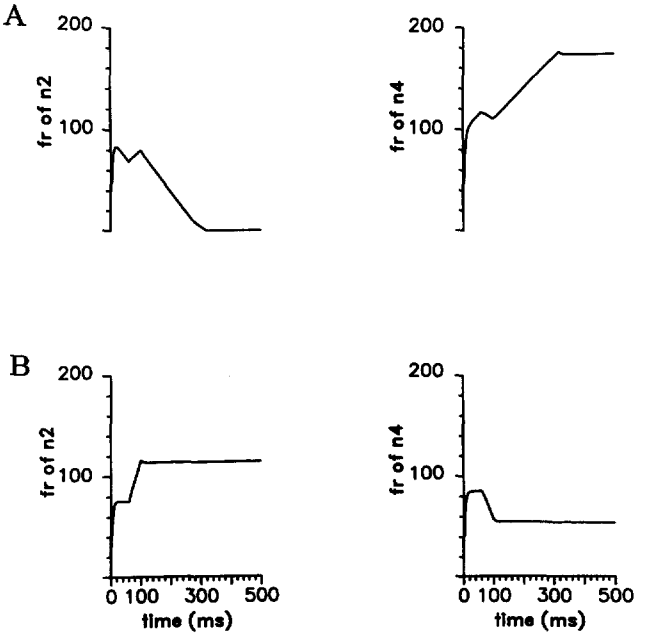

Fig. 5 A, B. Firing rates, *fr*, of interneurons *n*2 and *n*4 before and after hemilabyrinthectomy. **A** Hemilabyrinthectomy was simulated by zeroing the input from one of the canals. Half of the interneurons responded by firing at an unnaturally high rate, while the other half went into cut-off. **B** After 750 iterations the network learned to integrate with only a single canal input

neurons would have produced a nystagmus similar to that seen clinically. The network was then asked to learn to integrate with only a single canal input. Despite this gross insult, the system recovered in 750 iterations and had a final normal gain and time constant (Fig. 5B). One should recall that interneurons receive a tonic input that remains during re-learning. We feel that recovery would not have occurred without this input, but this conclusion is clouded by the fact that the network could not have learned initially without it.

Induction of post-saccadic drift

Recent experiments (Kapoula et al. 1989) have shown that it is possible to induce post-saccadic drift in humans. This was done by having a subject watch a random dot pattern while making spontaneous saccades. After each saccade the entire pattern slid briefly in the direction opposite to the saccade. After several hours the subjects' eyes would spontaneously drift backwards after making a saccade in the dark. This illustrates the ability of the neural integrator to perform signal processing more complex than just integration in order to reduce retinal slip. To simulate an analogous situation with our model, we started with a four-interneuron network trained to integrate. Then we asked that the desired eye velocity, after an eye movement, be negative at 5 deg/s, rather than zero. The network responded by learning, in only 200 iterations, to produce the appropriate drift in the opposite direction after making an eye movement. While our model receives vestibular, rather than saccadic inputs, we only wanted to show that it really should be thought of as a function generator that can produce whatever output signal is necessary (within reason) to reduce retinal slip.

Compensation for eye muscle plant

Using the modified network configuration shown in Fig. 1 we wished to show that the model could produce an output waveform that would compensate for plant mechanics (muscles plus passive orbital tissues) as well as integrate. The first network we tried had no constraints on the polarity of the weights. The learning algorithm was identical to that of the basic network, except for the following changes. The error was not evaluated during a 15 ms interval immediately after the onset of the velocity pulse and the 15 ms after its return to the background level. These intervals were excluded, because the network, if successful, must produce the sum of the integrated signal (eye position) and the original signal (eye velocity) as shown in (5). Since individual neurons had time constants of 5 ms, it was impossible for the network to respond to instantaneous changes in the input and this would create large, spurious errors. In addition, the error for a given set of weights (6) was evaluated for two input sequences with velocity pulses in opposite directions. The sum of these two errors was then used to calculate the partial derivatives and update the weights (7–9). This was done because the background rates of half the neurons

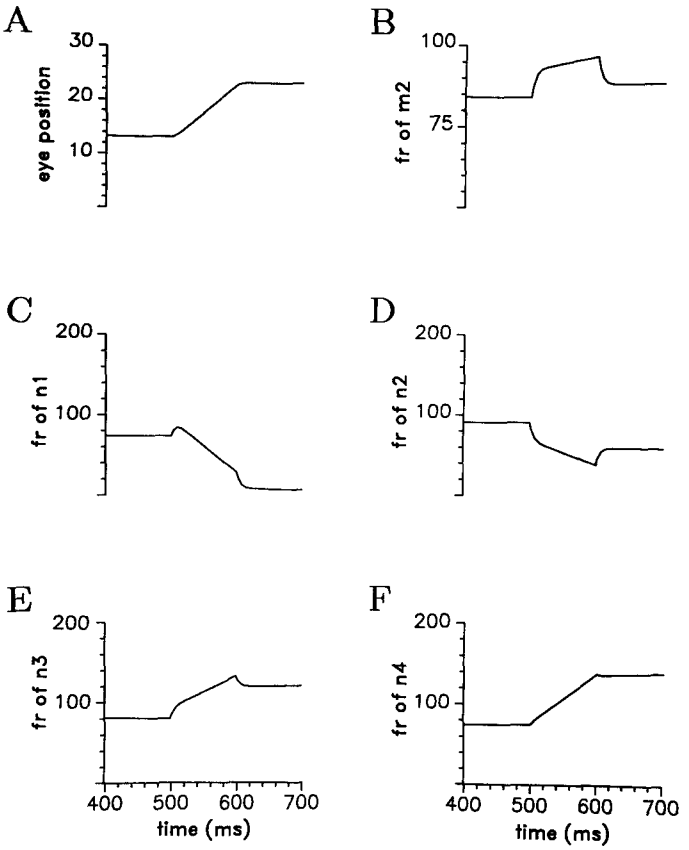

Fig. 6 A–F. Eye position, firing rate, *fr*, of a motoneuron, and activity of interneurons of a modified network in response to a rectangular pulse of head velocity. **A** Eye position **B** Firing rate of a motoneuron (*m*2) that encodes eye velocity and eye position **C** Firing rate of an interneuron (*n*1) with eye velocity and eye position signals in opposite directions. **D** and **E** Firing rate of interneurons (*n*2 and *n*3) with eye velocity and eye position signals in the same direction. **F** Firing rate of an interneuron (*n*4) which encodes only eye position

tended to go negative while the other half got very large during learning if only a single input was used.

After approximately 1000 iterations the error was reduced to the tolerance level. Eye position, motoneuron firing rate, and firing rate of the interneurons after learning for a typical network are shown in Fig. 6. The motoneuron carries the sum of the eye position and eye velocity signals in accordance with (5). Three of the four interneurons have an output proportional to eye velocity as well as eye position. The fourth (*n*4) responds primarily to eye position. In *n*1, the eye position and eye velocity signals are in opposite directions. Studies of the NPH in cats (Lopez-Barneo et al. 1982) have also noted that neurons tend to be either sensitive to eye velocity as well as eye position (burst-tonic) or to eye position alone (tonic). One characteristic of certain neurons in this model, that has been noted only rarely in monkeys (S.G. Lisberger, personal communication), is the presence of position and velocity signals in opposite direction. This phenomenon occurred in about 25% of the neurons in the ten simulations that we made.

As with the basic network the inputs and outputs of the network were not in a "push-pull" configuration

after the network had learned (i.e. each weight had an equal probability of being excitatory or inhibitory). Consequently, to make the network conform to experimental observations, we required the ipsilateral input weights to be inhibitory and the contralateral input weights to be excitatory. This resulted in the network developing an inhibitory commisural system. It also had the effect of eliminating the spurious neurons, leaving interneurons with either position and velocity signals in the same direction or with only position signals.

Discussion

The intent of this project is to mimic a developing oculomotor system where synapses are forming and undergoing modification. Through the minimization of retinal slip the model learns to process signals in a manner similar to the vestibulo-ocular reflex. The resulting behaviour and underlying structure can then be compared with observations from electrophysiological experiments. The oculomotor system, and particularly the neural integrator is extraordinarily well suited for neural network modeling. Unlike practically any other system in the brain, the signals encoded by brainstem neurons subserving eye movements have been very well characterized. The inputs and outputs of the neural integrator have been well-defined, and retinal slip is known to act as the external teacher. The only element of the system which remains completely unknown is the actual learning algorithm that is used by the brain to eliminate retinal slip.

It is assumed that the brain has some way of knowing how to modify individual synapses in order to reduce retinal slip. Our model accomplished this by changing each synapse a test amount in order to assess its contribution to the production of retinal slip, and then modifying that synapse accordingly. This was incorporated into a gradient descent algorithm that converged in most situations. The formation of partial derivatives and the use of gradient descent is somewhat artificial; simpler learning rules would have worked as well, but would have taken too long to converge. Our algorithm was approximately as efficient as that of other gradient descent algorithms, for example that of Williams and Zipser (1989), in that each iteration requires the order of N^4 operations, where N is the number of neurons. Nonetheless, when the number of neurons becomes large the network taken an unreasonable amount of time to converge. This problem might be solved by using a Hebbian type algorithm that could simultaneously update each of the synaptic weights based on a directional retinal slip signal.

One problem with retinal slip alone as an error signal is that it is the output eye velocity that is being optimized and so the background rates of the neurons cannot be controlled by learning. Consequently, we set the neuronal background rates at 100 spikes/s originally and in most cases they remained nearby during learning. In the case of the modified network this stability was only achieved when each error was evaluated for two different sequences with head movements in opposite direction. The question of how neuronal background rates in the brainstem arise and are controlled remains an intriguing one that was beyond the scope of this model.

It should also be noted that in the real oculomotor system, neurons in the nucleus prepositus hypoglossi and medial vestibular nucleus that comprise the neural integrator are actually third-order cells that receive inputs from second-order cells in the vestibular nucleus. In our model integrator, cells received inputs directly from the canal afferents. However, it is not hard to see that inclusion of relay neurons corresponding to second-order vestibular cells would have resulted in an equivalent circuit.

Recent three dimensional models of the oculomotor system (Tweed and Vilis 1987) have shown that correct calculation of eye position during an eye movement requires the integrator to consider initial eye position as well as the angular velocity. While this is a small effect unless the eye is considerably deviated from the center, it underlines our contention that the integrator is capable of signal processing that is more complicated than simple integration. Although it is beyond the scope of this study, we feel that given the appropriate learning sequences our network would be able to develop a vertical integrator and couple it to the horizontal integrator in such a way as to calculate eye position in three dimensions.

Our network was able to simulate the behaviour of the neural integrator in many physiological and pathological situations. It was able to learn to integrate, to recover from lesions, and to perform the more complicated signal processing tasks of the real neural integrator. The network, however, with no constraints on synaptic sign, did not come up with the push-pull configuration that the real integrator has been shown to have. This implies that constraints, most likely in the form of genetic hard-wiring, are imposed on the real neural integrator that restrict its final configuration. When the inputs and outputs of our network were forced to obey push-pull it invariably developed an inhibitory commissural system between the interneurons. Integration took place as the result of positive feedback that was distributed throughout the network. Nonetheless, each interneuron in our network carried a position signal (sometimes in combination with a velocity signal), and no partially integrated signals were seen.

In summary, we have shown that error-driven learning based on retinal slip can produce a network that has many characteristics similar to the neural integrator of the oculomotor system. It remains to develop a learning rule which can produce a network with similar qualities that is more efficient and biologically plausible.

Acknowledgements. D.B.A. was supported by a Howard Hughes Medical Institute Pre-doctoral Fellowship, and D.A.R was supported by grant EY00598 from the National Eye Institute of the National Institutes of Health. We thank C.V. Bridges for help with the illustrations.

References

Baker R, Berthoz A (1975) Is the prepositus hypoglossi nucleus the source of another vestibulo-ocular pathway? Brain Res 86:121–127

Cannon SC, Robinson DA (1985) An improved neural network model for the neural integrator of the oculomotor system: more realistic neuronal behavior. Biol Cybern 53:93–108

Cannon SC, Robinson DA, Shamma S (1983) A proposed neural network for the integrator of the oculomotor system. Biol Cybern 49:127–136

Fetter M, Zee DS (1988) Recovery from unilateral hemilabyrinthectomy in rhesus monkey. J Neurophysiol 59:370–393

Kamath BY, Keller EL (1976) A neurological integrator for the oculomotor system. Math Biosci 30:341–352

Kapoula Z, Optican LM, Robinson DA (1989) Visually induced plasticity of post-saccadic ocular drift in normal humans. J Neurophysiol 61:879–891

Lopez-Barneo J, Darlot C, Berthoz A, Baker R (1982) Neuronal activity in prepositus nucleus correlated with eye movement in the alert cat. J Neurophysiol 47:329–352

Pearlmutter BA (1989) Learning state space trajectories in recurrent neural networks. Neural Comput 1:263–269

Robinson DA (1968) Eye movement control in primates. Science 161:1219–1224

Robinson DA (1989) Integrating with neurons. Ann Rev Neurosci 12:33–45

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumelhart DE, McClelland JL, PDP Research Group (eds) Parallel distributed processing: explorations in the microstructure of cognition, vol 1: Foundations. MIT Press, Cambridge, pp 318–362

Tweed D, Vilis T (1987) Implications of rotational kinematics for the oculomotor system in three dimensions. J Neurophysiol 58:832–849

Weissman BM, DiScenna AO, Leigh RJ (1989) Maturation of the vestibulo-ocular reflex in normal infants during the first two months of life. Neurology 39:534–538

Williams RJ, Zipser D (1989) A learning algorithm for continually running fully recurrent neural networks. Neural Comput 1:270–280

Donald B. Arnold
Room 355 Woods Research Building
The Wilmer Institute
The Johns Hopkins Hospital
600 N. Wolfe St.
Baltimore, MD 21205
USA