Advanced Neural Computers / R. Eckmiller (Editor)

Feedback-Error-Learning Neural Network for Supervised Motor Learning

Mitsuo Kawato
Cognitive Processes Department
ATR Auditory and Visual Perception Research Laboratories, Kyoto

Abstract

In supervised motor learning, where the desired movement pattern is given in task-oriented coordinates, one of the most essential and difficult problems is how to convert the error signal calculated in the task space into that of the motor command space. We proposed the feedback-error-learning approach where the feedback motor command is used as an error signal to train a neural network which then generates a feedforward motor command. We successfully applied this approach to learning control in joint space of PUMA, an industrial manipulator, trajectory control in visual coordinates even when PUMA was given an extra degree of freedom, and Softarm with muscle-like actuators (manufactured by Bridgestone Co.). I mathematically show that the feedback-error-learning approach can be considered a Newton-like method in functional space. Mathematical proof of the convergence of the feedback-error-learning control scheme is provided both for trajectory stability and learning stability. The feedback-error-learning neural network was originally proposed as a model for the lateral cerebellum and the parvocellular part of the red nucleus. It will be shown that functions of other parts of the cerebellum, such as the vermis or the flocculus, can also be understood from the feedback-error-learning scheme.

1 Introduction

One of the features of the central nervous system (CNS) in its control of movement is the capability of motor learning. For higher mammals, especially humans, supervised learning is probably the most important class of motor learning. In nearly every case, the teacher can not directly show the correct motor command to the student, but only can show the desired movement trajectory.

Consider a neural network which receives a desired motor pattern and outputs a motor command to realize the desired movement. The motor command is transmitted to the musculoskeletal system and some particular movement is realized. The realized trajectory is measured by various sensory systems, and compared to the desired movement pattern (teaching signal). If the teacher were to able to give the difference between the ideal motor command and the actual motor command, various supervised learning rules could be used to train the motor control network. However, since this is not possible, the problem of converting the error from task-oriented coordinates to the motor command space is an essential and difficult one. This problem was addressed by Jordan and Rumelhart [2] and termed, "supervised learning with a distal teacher". They proposed the forward and inverse modeling approach to resolve the problem.

We proposed the *feedback-error-learning* neural network as a model of the lateral cerebellum and the parvocellular part of the red nucleus [5,6]. This model constitutes one possible answer to the above error conversion problem. In the next section, three representative computational schemes to resolve the problem are reviewed and compared. We then mathematically formulate the feedback-error-learning approach as a Newton-like method in functional space. Mathematical proof of the trajectory and learning convergence of feedback-error-learning will be given. Finally, functional roles of various parts of the cerebellum are examined based on the feedback-error-learning.

2 Computational schemes to convert trajectory error into motor command error

A perfect feedforward control can be realized if the feedforward controller provides an inverse model of the controlled object. Let us formulate this statement. θ denotes an n-dimensional vector which represents the body coordinates, such as joint angles or muscle lengths, of a controlled object. τ represents an m-dimensional vector of motor commands such as joint torque or muscle tension. The state change of the controlled object is described by the following ordinary differential equations.

$$\begin{aligned} d\theta/dt &= \dot{\theta} \\ d\dot{\theta}/dt &= f(\theta, \dot{\theta}, \tau), \end{aligned} \qquad (1)$$

here f is an n-dimensional nonlinear vector function. x denotes a k-dimensional vector representing the task-oriented coordinates of the controlled object, for example, the retinal coordinates of the hand position. x is uniquely determined from θ according to the following nonlinear equation: $x = G(\theta)$, here G is a k-dimensional nonlinear vector function.

The problem of feedforward control is to find the motor command $\tau_d(t)$ which realizes the desired movement pattern $x_d(t)$. First, the desired trajectory in body space is calculated from that in task space: $\theta_d(t) = G^{-1}(x_d(t))$. Second, the necessary motor command is calculated from the desired trajectory, velocity and acceleration $(\theta_d(t), \dot{\theta}_d(t), \ddot{\theta}_d(t))$ as a solution of the second equation of (1). Although this is an implicit equation with respect to τ, it is rewritten in explicit form as follows: $\tau_d(t) = h(\theta_d(t), \dot{\theta}_d(t), \ddot{\theta}_d(t))$. Consequently, in this case, the motor learning problem is equivalent to acquisition of the conjoined inverse kinematics model (IKM) and inverse dynamics model (IDM) $h \cdot G^{-1}$ in the feedforward controller.

In Fig. 1, I compare three computational approaches for learning the inverse model of a controlled object. The simplest approach is shown in Fig. 1a. The controlled object receives the torque input $\tau(t)$ and outputs the resulting trajectory $x(t)$. The inverse model is oriented in the input-output direction opposite to that of the controlled object, as shown by the arrow. That is, it receives the trajectory as an input and outputs the torque $\tau_i(t)$. The error signal $s(t)$ is given as the difference between the actual torque and the estimated torque: $s(t) = \tau(t) - \tau_i(t)$. This approach to acquiring an inverse model is referred to as direct inverse modeling [2]. Fig. 1b shows the method of combining a forward model and an inverse model, proposed by Jordan and Rumelhart [2]. First, the forward model of the controlled object is learned by monitoring both the input $\tau(t)$ and the output $x(t)$ of the controlled object. Next, the desired trajectory $x_d(t)$ is fed to the inverse model to calculate the feedforward motor command $\tau(t)$. The resulting error in the trajectory $x_d(t) - x(t)$ is back propagated through the forward model to calculate the error in the motor command, which is then used as the error signal for training the inverse model.

Fig. 1c shows the *feedback error learning* approach [5]. The total torque $\tau(t)$ fed to the controlled object is the sum of the feedback torque $\tau_{fb}(t)$ and the feedforward torque $\tau_{ff}(t)$,

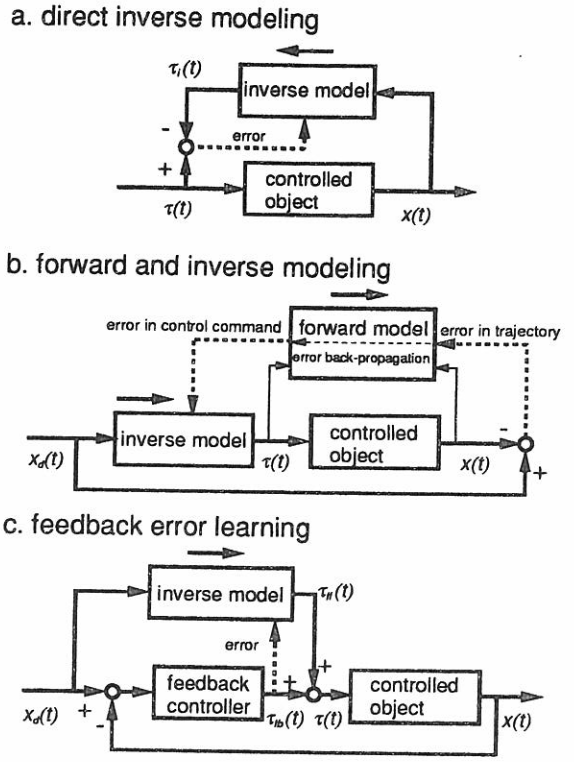

Figure 1: Three computational schemes for learning inverse model of a controlled object: a) Direct inverse modeling. b) Forward and inverse modeling. c) Feedback error learning scheme.

which is calculated by the inverse model. The inverse model receives the desired trajectory x_d and monitors the feedback torque $\tau_{fb}(t)$ for the error signal. It is expected that the feedback signal tends to zero as learning proceeds. We call this learning scheme *feedback error learning* to emphasize the importance of using the feedback torque (motor command) as the error signal of the heterosynaptic learning.

In summary, the direct inverse modeling approach avoids the error conversion problem by reversing the input and the output. The forward and inverse modeling approach converts trajectory error into motor command error by backpropagation through the forward model. In the feedback-error-learning approach the feedback controller converts trajectory error into motor command error.

Direct inverse modeling does not seem to be used by the central nervous system. The main reason is that after the inverse model is acquired, before it can be input from the desired trajectory instead of the actual trajectory, large-scale connection changes must be carried out while preserving minute one-to-one correspondence. In engineering applications, one drawback of the direct inverse modeling approach seems to be that it does not necessarily achieve a particular target trajectory $x_d(t)$, even when the training period is sufficiently long. In this sense, the learning is not "goal-directed" [2]. The forward and inverse modeling approach of Jordan & Rumelhart is goal-directed because the error for learning is defined as the square of the difference between the desired trajectory and the realized trajectory. I will provide a mathematical proof that the feedback error learning approach is also goal-directed.

The direct inverse modeling method can not cope with the learning control of a redundant controlled object. Jordan [2] clearly explained the reason for this in the one-to-many inverse kinematics problem. The forward and inverse modeling approach can resolve the illposedness of the problem by learning performance index as synaptic weights in the inverse model. Any feedback controller selects one specific motor command even for redundant controlled objects. However, the desired trajectory can not be exactly realized by the feedback control alone. Because of the above-mentioned desirable characteristics inherent in the feedback controller, the feedback error learning approach can realize the learning control of controlled objects which are redundant either at the kinematics or dynamics level [3,4].

Kano et al. succeeded in learning trajectory control within the stereo camera coordinates even when PUMA was given an extra degree of freedom [3]. In this study, the feedback controller calculated the feedback motor command by multiplying the error in the visual coordinates with the Moore-Penrose pseudoinverse of the coordinate transformation. Because the pseudoinverse matrix finds the solution with the smallest norm, this method is closely related to our minimum motor-command change criterion.

Katayama et al. solved the ill-posed inverse dynamics problem for an arm-like manipulator (Bridgestone SoftArm) with rubbertuators which are air driven, muscle-like actuators. Because each joint contains agonist and antagonist rubbertuators, there is no unique solution to determining tensions to realize a particular joint angle movement. In this experiment, we obtained a roughly minimal muscle-tension change trajectory with the feedback-error-leaning scheme [4].

3 Feedback-error-learning as a Newton-like method in functional space

The inverse dynamics problem is to calculate the required motor command $\tau_d(t)$ so that the desired trajectory $\theta_d(t)$ in the body space is realized. The dynamics of the controlled object is described by (1). The Newton method is first described. It is more convenient to restrict the inverse dynamics problem to that in a finite time interval $[0,T]$. A functional $F(\tau) = \theta_d - \theta(\tau)$ is defined as a mapping from a function space $\Xi = C[0,T]$ of the motor command to a function space $\Theta = C^2[0,T]$ of the body space trajectory. The first derivative of F can be computed through the solution of the following variational equation:

$$\begin{aligned} F'(\tau)\delta\tau &\equiv J\delta\tau = \delta\theta. && (2) \\ d(\delta\theta)/dt &= \delta\dot{\theta}, \quad \delta\theta(0) = 0, \\ d(\delta\dot{\theta})/dt &= \partial f(\theta,\dot{\theta},\tau)/\partial\theta\cdot\delta\theta + \partial f(\theta,\dot{\theta},\tau)/\partial\dot{\theta}\cdot\delta\dot{\theta} \\ &\quad +\partial f(\theta,\dot{\theta},\tau)/\partial\tau\cdot\delta\tau, \quad \delta\dot{\theta}(0) = 0. && (3) \end{aligned}$$

According to the Newton method, we can calculate the motor command error from (3) as follows:

$$\begin{aligned} \delta\tau &= F'^{-1}(\tau)\delta\theta \\ &= (\partial f(\theta,\dot{\theta},\tau)/\partial\tau)^{-1}\{-\partial f(\theta,\dot{\theta},\tau)/\partial\theta\cdot\delta\theta - \partial f(\theta,\dot{\theta},\tau)/\partial\dot{\theta}\cdot\delta\dot{\theta} + \delta\ddot{\theta}\}. && (4) \end{aligned}$$

Although this equation looks quite complicated, it must be emphasized that all calculation can be done by differentiation of the body space error $\delta\theta = \theta_d - \theta$ and matrix calculation. However, because we do not know the dynamics of the controlled object (i.e. f), we can not use this Newton method. A quite general solution to this difficulty is the Newton-like method which approximates F'^{-1} by some simpler operator M. One apparent candidate is the following:

$$\delta\tau = M(\delta\theta) = K_P(\theta_d - \theta) + K_D(\dot{\theta}_d - \dot{\theta}) + K_A(\ddot{\theta}_d - \ddot{\theta}). \quad (5)$$

This approximation is validated since the product factor $(\partial f(\theta,\dot{\theta},\tau)/\partial\tau)^{-1}$ in the third term of (4) corresponds to the inertia matrix which is symmetrical, and positive definite. Furthermore, in the simplest case, the product factor $-(\partial f(\theta,\dot{\theta},\tau)/\partial\tau)^{-1}\partial f(\theta,\dot{\theta},\tau)/\partial\dot{\theta}$ of $\delta\dot{\theta}$ in the second term of (4) corresponds to a viscosity coefficient. Similarly, the product factor $-(\partial f(\theta,\dot{\theta},\tau)/\partial\tau)^{-1}\partial f(\theta,\dot{\theta},\tau)/\partial\theta$ of $\delta\theta$ in the first term of (4) corresponds to the stiffness of a virtual spring. Thus, all three factors can be approximated by positive diagonal matrices K_P, K_D and K_A, which can be regarded as gains of proportional, differentiation and acceleration feedbacks, respectively. Thus, the PDA feedback controller can calculate $\delta\tau$ as a Newton-like method. The feedback error learning which uses the feedback motor command as the error signal to modify the feedforward motor command was originally proposed by us [5,6].

In the forward and inverse modeling, the forward model provides a dynamical model of the controlled object. Thus, the backpropagation through the forward model is also a dynamical process which is described by differential equations. On the other hand, the feedback-error-learning scheme utilizes a linearized approximation of the inverse dynamics model (i.e. feedback controller) which can convert trajectory error into motor command error by relatively simple computation.

4 Stability of feedback-error-learning scheme

In this section, stability of the feedback-error-learning scheme is proved for a learning control system shown in Fig. 1c. The dynamics of the controlled object is described by (1). In this section, we consider only the inverse dynamics problem without kinematics. Further, we treat a simple case where f is invertible for clarity. Hence, the dynamics of the controlled object can be described as follows: $\tau = h(\theta,\dot{\theta},\ddot{\theta})$. The total torque fed to the controlled object is the sum of the feedback torque τ_{fb} calculated by a PDA feedback controller and the feedforward torque τ_{ff} calculated by the neural network IDM.

$$h(\theta,\dot{\theta},\ddot{\theta}) = \tau = \tau_{fb} + \tau_{ff} = K_P\xi + K_D\dot{\xi} + K_A\ddot{\xi} + \tau_{ff}, \tag{6}$$

here $\xi = \theta_d - \theta$. The neural network IDM calculates τ_{ff} from the desired trajectory θ_d and the synaptic weights w:

$$\tau_{ff} = \psi(w,\theta_d,\dot{\theta}_d,\ddot{\theta}_d). \tag{7}$$

The shape of the function ψ depends on the kind of neural network model used for the IDM. Here, I assume that an appropriate neural network model is chosen and that it can realize the exact inverse dynamics for an optimal set of synaptic weights $\tilde{w}$:

$$h(\theta,\dot{\theta},\ddot{\theta}) = \psi(\tilde{w},\theta,\dot{\theta},\ddot{\theta}). \tag{8}$$

The synaptic modification rule of the feedback-error-learning scheme is represented in a general manner as follows:

$$dw/dt = (\partial\tau_{ff}/\partial w)^T\tau_{fb}. \tag{9}$$

Because this expression is quite general, it includes several learning rules for a prepared-nonlinearity neural network and MLP which we used in previous studies [3,4,5,6].

Substituting (7) and (8) into (6), we obtain the following equation.

$$\begin{aligned}
\tau_{imag} &\equiv \tilde{\tau}_{ff} - \tau_{ff} \\
&= \psi(\tilde{w},\theta_d,\dot{\theta}_d,\ddot{\theta}_d) - \psi(w,\theta_d,\dot{\theta}_d,\ddot{\theta}_d) \\
&= \psi(\tilde{w},\theta_d,\dot{\theta}_d,\ddot{\theta}_d) - h(\theta,\dot{\theta},\ddot{\theta}) + \tau_{fb} \\
&= h(\theta_d,\dot{\theta}_d,\ddot{\theta}_d) - h(\theta,\dot{\theta},\ddot{\theta}) + \tau_{fb} \\
&\equiv \Delta h(\theta_d,\xi,\dot{\xi},\ddot{\xi}) + \tau_{fb},
\end{aligned} \tag{10}$$

here τ_{imag} is an imaginary torque which is fed to an imaginary dynamical system described by the last expression of (10). The imaginary torque τ_{imag} is generated by an estimation error of $\tilde{w}$ by w. If $w = \tilde{w}$ then $\tau_{imag} = 0$ for any θ_d. The last expression defines an imaginary dynamical system which is a combination of the variational system of the controlled object around the desired trajectory θ_d and the feedback controller with the input τ_{imag}:

$$\Delta h(\theta_d, \xi, \dot{\xi}, \ddot{\xi}) + K_P\xi + K_D\dot{\xi} + K_A\ddot{\xi} = \tau_{imag}. \tag{11}$$

This equation (11) defines an imaginary nonautonomous dynamical system with respect to ξ with the input τ_{imag}. Here, I make the first assumption about the stability of this imaginary dynamical system.

Assumption 1 *If $\tau_{imag} = 0$, then the solution ξ of (11) will tend to 0 as t goes to infinity.*

Thus, global asymptotic stability of the origin of (11) for zero input is assumed here. We make the second assumption about the feedback motor command τ_{fb}.

Assumption 2 *τ_{fb} is approximately proportional to $\lambda\tau_{imag}$ with $0 < \lambda < 1$.*

This assumption can be justified in two different ways. Let us consider an expansion of the imaginary dynamical system (11).

$$\begin{aligned}(K_A + \partial h(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)/\partial\ddot{\theta})\ddot{\xi} \;&+\; (K_D + \partial h(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)/\partial\dot{\theta})\dot{\xi} + (K_P + \partial h(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)/\partial\theta)\xi \\ +o(\xi, \dot{\xi}, \ddot{\xi}) \;&=\; \tau_{imag},\end{aligned} \tag{12}$$

here $o(\xi, \dot{\xi}, \ddot{\xi})$ is a higher order term in ξ, and hence a small order compared with the linear terms for small ξ. If the feedback gains are sufficiently large compared with the partial derivatives of h, then Assumption 2 holds with $\lambda \cong 1$. If these feedback gains are chosen as in (4) so that the feedback-error-learning scheme becomes a good approximation of the Newton method as described in the previous section, then Assumption 2 holds with $\lambda \cong 1/2$. The following function V is examined as a candidate of the Liapunov function.

$$V = 1/2 \cdot \tau_{imag}^T\tau_{imag} \geq 0. \tag{13}$$

The equality of (13) holds only when $w = \tilde{w}$ for nontrivial θ_d. It is quite natural to assume that θ_d is not always zero. I calculate the time derivative of V.

$$\begin{aligned}dV/dt &= \tau_{imag}^T\partial\tau_{imag}/\partial w \cdot dw/dt + dV/d\hat{t} \\ &= \tau_{imag}^T\{-\partial\tau_{ff}/\partial w\}(\partial\tau_{ff}/\partial w)^T\tau_{fb} + dV/d\hat{t} \\ &= \tau_{imag}^T\{-\partial\tau_{ff}/\partial w\}(\partial\tau_{ff}/\partial w)^T\lambda\tau_{imag} + dV/d\hat{t} \\ &= -\lambda\{(\partial\tau_{ff}/\partial w)^T\tau_{imag}\}^T\{(\partial\tau_{ff}/\partial w)^T\tau_{imag}\} + dV/d\hat{t},\end{aligned} \tag{14}$$

here $dV/d\hat{t}$ is the partial derivative of V with respect to the time t while fixing w constant. Thus, it can be expressed as follows:

$$dV/d\hat{t} = \partial V/\partial\theta_d \cdot \dot{\theta}_d + \partial V/\partial\dot{\theta}_d \cdot \ddot{\theta}_d + \partial V/\partial\ddot{\theta}_d \cdot d\ddot{\theta}_d/dt \tag{15}$$

Now, I average the rate of change of V for a time interval which is shorter than the time constant of change of w but much longer than the mixing rate of the desired trajectory θ_d. Thus, the rate of change of w is assumed much slower than that of θ_d. Then, if θ_d is a strongly stationary stochastic process, the time average of the second term of (14) vanishes since V is regarded as a Baire function of the strongly stationary process.

$$\overline{dV/d\hat{t}} = d\overline{V}/d\hat{t} = 0. \tag{16}$$

This is because, when w is constant, V is a function of θ_d alone. Then, from the definition of the strongly stationary stochastic process, the average of V is constant. Consequently we obtain:

$$\overline{dV/dt} = -\lambda\overline{\{(\partial\tau_{ff}/\partial w)^T\tau_{imag}\}^T\{(\partial\tau_{ff}/\partial w)^T\tau_{imag}\}} \leq 0 \tag{17}$$

The equalities of (17) and (13) hold only when $w = \tilde{w}$ for nontrivial θ_d. Consequently, I can conclude that w asymptotically converges to the optimal set of synaptic weights $\tilde{w}$. Then, because $\tau_{imag} = 0$ for $w = \tilde{w}$, θ asymptotically converges to θ_d based on Assumption 1.

5 Feedback-error-learning neural network as models of different parts of cerebellum

The cerebellum is divided into separate sagittal regions with distinctive anatomical connections although the cellular organization of the cerebellar cortex is simple, regular and uniform. These divisions form three functionally distinct parts of the cerebellum: the vestibulocerebellum, the spinocerebellum, and the cerebrocerebellum [1].

The cerebrocerebellum is the lateral zone of the cerebellum. Its inputs originate exclusively in pontine nuclei that relay information from the cerebral cortex, and its output is conveyed by the dentate nucleus to the thalamus and then to the motor cortex. The feedback-error-learning neural network was originally proposed as a model for the cerebrocerebellum and the parvocellular part of the red nucleus [5,6]. Fig. 2c shows this model of the lateral part of the cerebellar hemisphere. In this figure, the feedback controller and the summation of the feedforward and feedback command reside in the motor cortex of the cerebrum. The feedback loop is the transcortical loop. The desired trajectory is sent to the cerebellum and the motor cortex from the association cortex. The output of the cerebellum is sent back to the motor cortex via the thalamus.

The spinocerebellum includes the vermis at the midline and the intermediate zone of the hemispheres. These two regions are the areas of the cerebellum that receive sensory information from the periphery. The vermis is related to control of posture. Gomi and I are now proposing a closed loop control system based on the feedback error learning as shown in Fig. 2b. It can be considered as a model of the vermis.

The vestibulocerebellum occupies the flocculonodular lobe. Flocculus is known to play a role in adaptive modification of the vestibulo-ocular reflex [1]. Its circuit diagram is shown in Fig. 2a. Although this circuit does not contain any feedback controller, the visual system measures the combined eye and head velocity and plays a role in converting trajectory error into the motor command error. Consequently, the function of the flocculus can also be understood from the feedback-error-learning concept.

Acknowledgment

The author wishes to express his thanks to Drs. E. Yodogawa and K. Nakane of ATR Auditory and Visual Perception Research Laboratories for their continuing encouragement.

References

[1] Ito M (1984) The cerebellum and neural control. Raven Press, New York

[2] Jordan MI (1988). Supervised learning and systems with excess degrees of freedom. *COINS Technical Report 88-27*, 1–41.

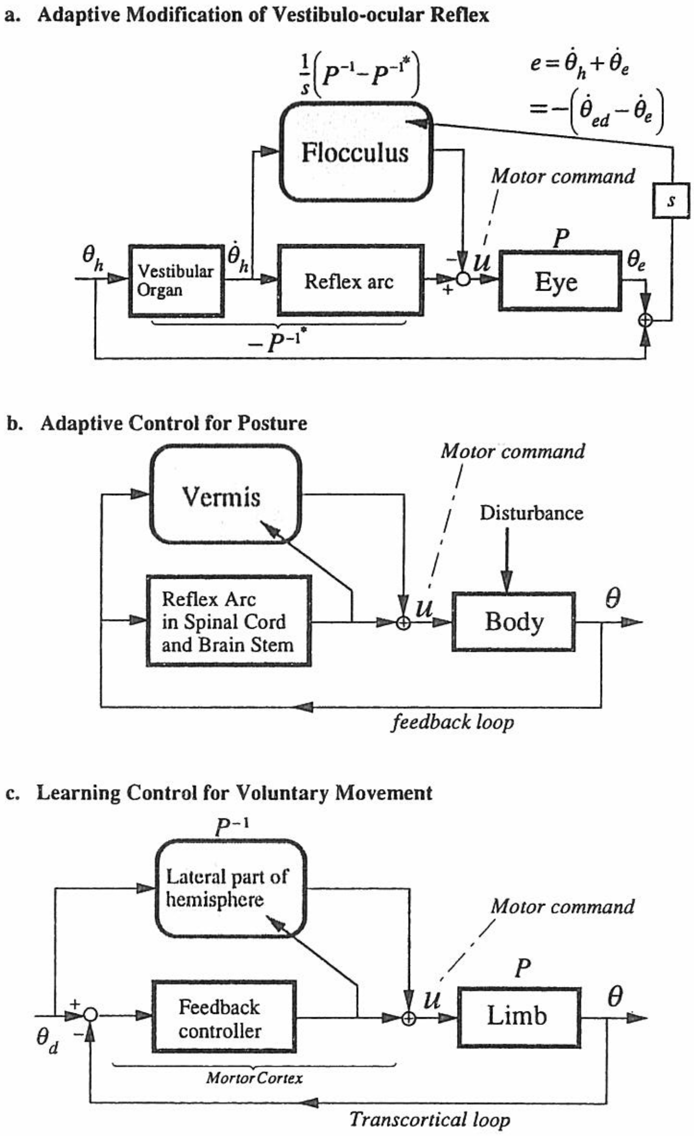

Figure 2: Functional roles of different parts of cerebellum interpreted based on the feedback error learning scheme: a) Flocculus for adaptive modification of the vestibulo-ocular reflex. b) Vermis for adaptive control of posture. c) Lateral hemisphere for learning of voluntary motor control.

[3] Kano M, Kawato M, Suzuki R (1989) Acquisition of inverse-kinematics and inverse-dynamics model of a redundant arm by feedback error learning. *Jpn IEICE Tech Rep* **MBE88-171**, pp 91–96 (in Japanese)

[4] Katayama M, Kawato M (1989) Learning control of a redundant controlled-object by feedback error learning. *Jpn IEICE Tech Rep* **MBE89-76**, pp 83–90 (in Japanese)

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

[6] Tsukahara N, Kawato M (1982) Dynamic and plastic properties of the brain stem neuronal networks as the possible neuronal basis of learning and memory. In: Amari S, Arbib MA (eds) Lecture notes in biomathematics, **45** *Competition and cooperation in neural nets.* Springer-Verlag, Berlin Heidelberg New York, pp 430-441