TR－A－0086

Optimization and Learning in Neural Networks for Formation and Control of Coordinated Movement

Mitsuo KAWATO

1990. 7.23

ATR 視聴覚機構研究所

〒619-02 京都府相楽郡精華町乾谷　☎07749-5-1411

ATR Auditory and Visual Perception Research Laboratories

Inuidani, Seika-cho, Soraku-gun, Kyoto 619-02 Japan

Telephone: +81-7749-5-1411
Facsimile: +81-7749-5-1408
Telex: 5452-516 ATR J

Optimization and Learning in Neural Networks for Formation and Control of Coordinated Movement[1]

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

[1]to appear in: D. Meyer (Ed.), *Attention and Performance, XIV*. Hillsdale, New Jersey: Lawrence Erlbaum. (1991)

Abstract

We proposed that the trajectory followed by human subject arms tended to minimize the time integral of the square of the rate of change of torque. Based on this computational model, the cascade neural network model, which utilizes a forward model of a controlled object, reproduced Fitts's law of speed-accuracy tradeoff as well as various invariant features of path and velocity profiles of multi-joint arm movement. For supervised motor learning, conversion of the error signal calculated in the task space into that of the motor command space is most essential and difficult. We proposed a feedback-error-learning approach in which the feedback motor command is used as an error signal to train an inverse model of the controlled object, which then generates a feedforward motor command. Here, we propose a unified neural network model which integrates the two previous models. In this model, for very skilled movements relaxation computation is conducted using both the forward and inverse models of the controlled object, while only the inverse model acquired by the feedback-error-learning is utilized for relatively difficult or less skilled movements.

Contents

1. Introduction

Studies of computational neuroscience and neural computing have increased exponentially in recent years. Marr (1982), a pioneer in this field, pointed out that an information processing device (brain) must be understood at the following different levels before one can be said to have understood it completely. (i) Computational theory. (ii) Representation and algorithm. (iii) Hardware implementation. Based on detailed knowledge of the neural circuits in the cerebellum, Marr (1969) and Albus (1971) proposed neural network models of the cerebellum. In these "perceptron" models, the efficacy of a parallel fiber–Purkinje cell synapses was assumed to change when conjunction of the parallel fiber input and the climbing fiber input occurs. Ito et al. (Ito, 1984) demonstrated the presence of the putative heterosynaptic plasticity of Purkinje cells in the flocculus of the cerebellum, which plays an essential role in the adaptive control of the vestibulo-ocular reflex. Fujita (1982) expanded the Marr-Albus model to an adaptive filter model for the vestibulo-ocular reflex from a dynamical system viewpoint. Consequently, a splendidly comprehensive understanding of the adaptive control of the vestibulo-ocular reflex was provided by these works, which accounts for all the above three levels.

Investigation of voluntary movement is much more difficult than that of the vestibulo-ocular reflex for the following reasons. First, control objects of voluntary movements (e.g. a hand or leg) have highly nonlinear dynamics with multiple degrees of freedom. Second, many neural networks and pathways are hierarchically involved in execution of voluntary movement (Allen & Tsukahara, 1974). Third, volition participates in the highest level of control. Fourth, an understanding at the computational level is not trivial.

Based on the pioneering work by Saltzman (1979) and Hollerbach (1982), we proposed a computational model of voluntary movement, as shown in Fig. 1..1, which accounts for Marr's first level (Kawato, Furukawa, & Suzuki, 1987). Consider a thirsty person reaching for a glass of water on a table. The goal of the movement is moving the arm toward the glass to reduce thirst. First, one desirable trajectory in task-oriented coordinates must be selected from the infinite number of possible trajectories which lead to the glass, whose spatial coordinates are provided by the visual system (determination of trajectory). Second, the spatial coordinates of the desired trajectory must be reinterpreted in terms of a corresponding set of body coordinates, such as joint angles or muscle lengths (transformation of coordinates). Finally, motor commands (e.g. torque) must be generated to coordinate the activity of many muscles so that the desired trajectory is realized (generation of motor command).

We do not adhere to the hypothesis of the step-by-step information processing shown by the three straight arrows in Fig. 1..1. Rather, Uno, Kawato and Suzuki (1989) proposed a learning algorithm which calculates the motor command directly from the goal of the movement represented by some performance index (thin curved arrow in Fig. 1..1). Further, as shown by the thick curved arrow in Fig. 1..1, motor command can be obtained directly from the desired trajectory represented in task-oriented coordinates (Kawato, Isobe, Maeda & Suzuki, 1988). In this respect, our model differs from the three-level hierarchical movement plan proposed by Hollerbach (1982). We will discuss later how the jump-over, direct computations shown by the curved arrows and the serial, step-by-step computations shown by the straight arrows cooperate and take partial charge of the computational work for voluntary movement.

Several lines of experimental evidence suggest that the information in Fig. 1..1 is internally represented in the brain. First, Bizzi, Accornero, Chapple and Hogan (1984) reported experiment results which indicate that the desired trajectory is explicitly planned in the brain. When the forearm of a deafferented monkey was quickly forced to the target position early in the movement, the arm returned to some intermediate point between the initial and final target positions, then gradually approached the final position again. A trajectory which connects the above intermediate points for various times of perturbation can be regarded as the desired, planned trajectory. This experimental fact is not consistent with the "final position control" hypothesis proposed earlier by the same authors. Furthermore, Flash and Hogan (1985) provided strong evidence to indicate that movement is planned at task-oriented coordinates (visual coordinates) rather than at the joint or muscle level. Second, the presence of the transcortical loop (Evarts, 1981), which is the negative feedback loop via the cerebral cortex, indicates that the desired trajectory must be represented also in body coordinates, since signals from proprioceptors are expressed in body coordinates. Finally, Cheney and Fetz (1980) showed that discharge frequencies of primate corticomotoneuronal cells in the motor cortex were fairly proportional to active forces (torque). Consequently, the CNS must adopt, at least partly, the step-by-step computation strategy for the control of voluntary movement.

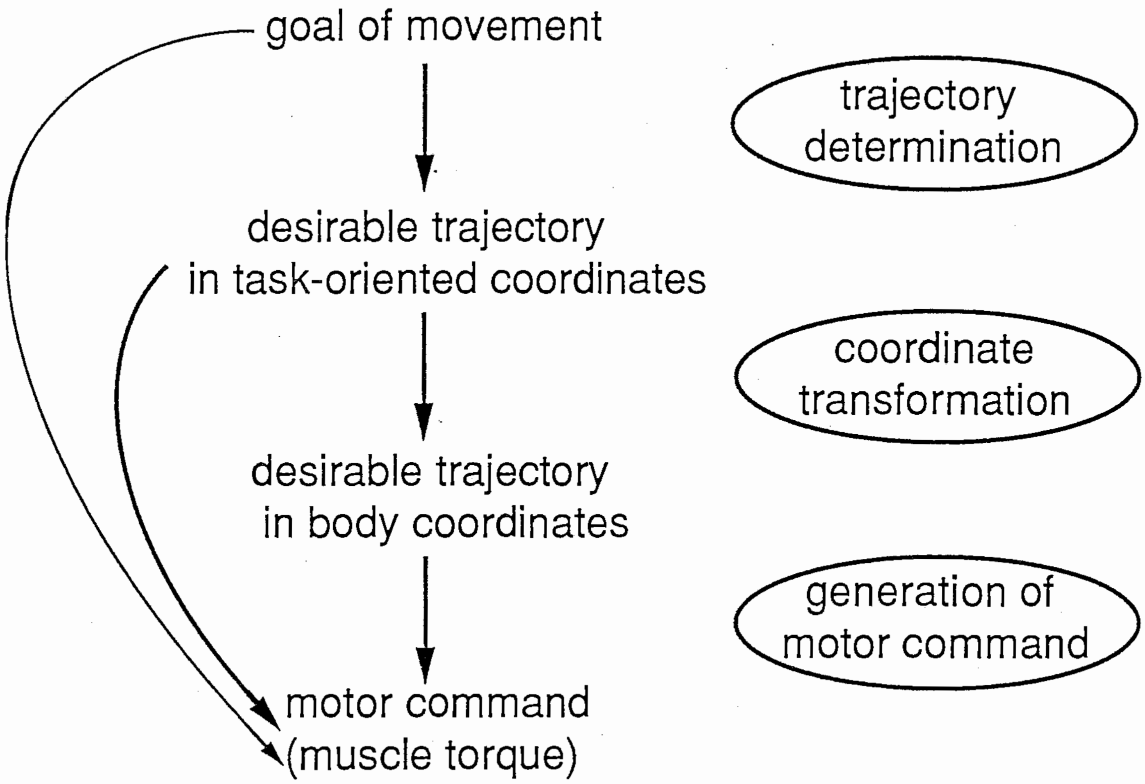

Figure 1..1: Computational models for information processing and internal information representation in the brain for sensory-motor control of voluntary movement.

2. Ill-posed Motor Control Problems

A problem is well-posed when its solution exists, is unique and depends continuously on the initial data. Ill-posed problems fail to satisfy one or more of these criteria. Most motor control problems are ill-posed in the sense that the solution is not unique.

We list three ill-posed control problems in Fig. 2..1. First, consider the trajectory determination problem for a planar, two-joint arm movement within a plane, when the starting point, the via-point and the end point, as well as the movement time, are specified (Fig. 2..1, top). There are an infinite number of possible trajectories satisfying these conditions. Thus, the solution is not unique and the problem is ill-posed.

The second ill-posed problem is the inverse kinematics problem in a redundant manipulator with excess degrees of freedom. For example, consider a three-degrees-of-freedom manipulator in a plane (Fig. 2..1, middle). The inverse kinematics problem is to determine the three joint angles (three degrees of freedom) when the hand position in the Cartesian coordinates (two degrees of freedom) is given. Because of the redundancy, even when the time course of the hand position is strictly determined, the time course of the three joint angles can not uniquely be determined. We note that human arms have excess degrees of freedom.

The third ill-posed motor control problem is the inverse dynamics problem in a manipulator with agonist and antagonist muscles (actuators). Consider a single joint manipulator with a pair of muscles (Fig. 2..1, bottom). The inverse dynamics problem is to determine the time courses of agonist and antagonist muscle tensions when the joint angle time course is determined. Even when the time course of the joint angle is specified, there are an infinite number of tension waveforms of the two muscles which realize the same joint angle time course, as indicated by the thick and thin curves at the bottom of Fig. 2..1.

There are two different approaches which resolve these ill-posed problems. The first approach is to utilize a feedback controller. The feedback controller selects one specific motor command in the inverse dynamics and inverse kinematics problems even for redundant manipulators. The second approach is to introduce a smoothness performance index.

We proposed the feedback-error-learning neural network model in which an inverse model of the controlled object is learned by using a feedback motor command as an error signal for training. Because the feedback motor command is determined uniquely even for redundant controlled objects, the

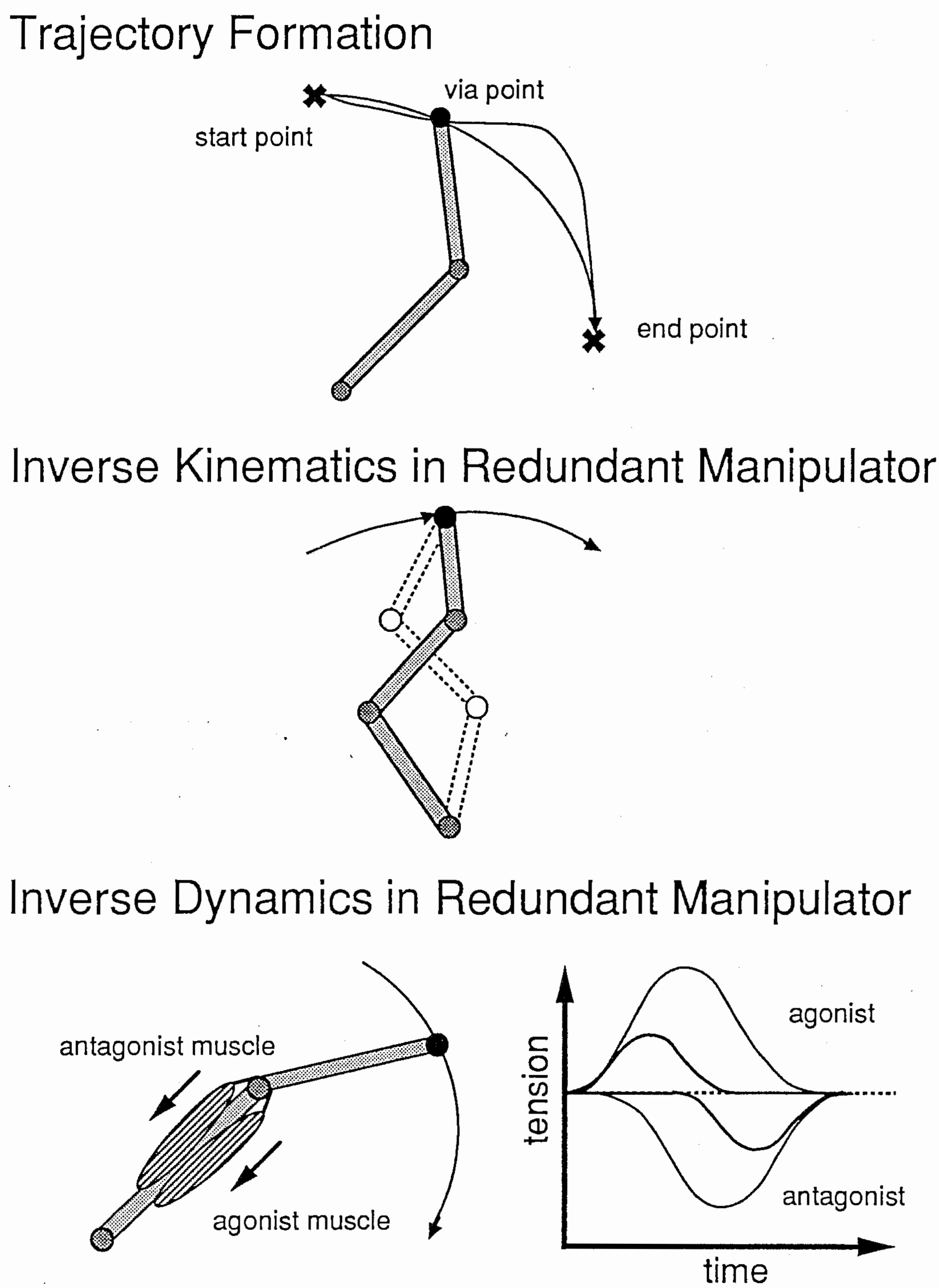

Figure 2..1: Three ill-posed problems in sensory-motor control.

inverse model can be acquired even for ill-posed inverse kinematics or dynamics problems. Applications of the feedback-error-learning neural network models to learning trajectory control of redundant arms were successfully conducted. This corresponds to the first approach stated above.

As the second approach, we proposed the minimum torque-change-model as a computational theory and the cascade neural network model as its hardware implementation. This approach utilizes a forward model of the controlled object in relaxation calculation, and can resolve all the three ill-posed problems shown in Fig. 2..1 simultaneously.

We will show that if a feedback controller with Moore-Penrose pseudo-inverse matrix is used, the feedback-error-learning approach can realize an approximation to the minimum motor-command-change trajectory which is a natural extension of the minimum torque-change model.

In this paper, these two approaches and its relationship will be explained. Finally, we propose a unified neural network model which integrates the two previous models. In this model, for very skilled movements relaxation computation is conducted using both the forward and inverse models of the controlled object, while only the inverse model acquired by the feedback-error-learning is utilized for relatively difficult or less skilled movements.

3. Trajectory Formation Based on Optimization Principle

3.1 Minimum Torque-change Model

In this section, we explain two experimentally confirmed objective functions for voluntary movements, that is the minimum jerk model and the minimum motor-command-change model.

The most marked and beautiful experiment features of human multi-joint arm movements are that hand paths between two points are roughly straight and hand tangential speed profiles are bell-shaped (Kelso, Southard, & Goodman, 1979; Morasso, 1981; Abend, Bizzi, & Morasso, 1982; Atkeson & Hollerbach, 1985; Flash & Hogan, 1985; Uno, Kawato, & Suzuki, 1989). In order to account for such kinematic features of human multi-joint arm movements, Flash and Hogan (1985) proposed a mathematical model, the "*minimum jerk model*". They proposed that the trajectory followed by the subject's arms tended to minimize the following quadratic measure of performance: the integral of the square of the jerk (rate of change of acceleration) of the hand position (X, Y), integrated over the entire movement.

$$C_J = 1/2 \int_0^{t_f} \{(\frac{d^3X}{dt^3})^2 + (\frac{d^3Y}{dt^3})^2\} dt. \quad (3.1)$$

Here, (X, Y) are Cartesian coordinates of the hand, and t_f is the movement duration. Flash and Hogan showed that the unique trajectory which yielded the best performance was in good agreement with the experiment data on movement within the region just in front of the body. Their analysis was based solely on the kinematics of movement, independent of the dynamics of the musculoskeletal system, and was successful only when formulated in terms of the motion of the hand in extracorporeal space.

Based on the idea that the objective function must be related to the dynamics, Uno, Kawato and Suzuki (1989) proposed the following alternative quadratic measure of performance:

$$C_T = 1/2 \int_0^{t_f} \sum_{i=1}^{m} (\frac{d\tau^i}{dt})^2 dt, \quad (3.2)$$

here, τ^i is the torque fed to the ith of m actuators. The objective function is the sum of the square of the rate of change of the torque, integrated over the entire movement. One can easily see that the two objective functions, C_J and C_T, are closely related since the rate of change of torque is locally proportional to the jerk. However, it must be emphasized that the objective function C_T critically depends on the dynamics of the musculoskeletal system.

For the movements between pairs of targets just in front of the body, predictions of both the models were in good agreement with the experimental data. However, the trajectories predicted by the minimum torque-change model were quite different from the minimum jerk model in four behavioral situations. In one situation, past experiment data support the minimum torque-change model (see below). The other three situations were not examined in past experiments. Uno et al. (1989) in recent experiments examined human planar arm movement in three different situations and found that the minimum torque-change model predicted the real data better than the minimum jerk model.

First, when the starting point is an arm outstretched to the side and the end point is in front of the body, the path was curved in the minimum torque-change model, but always straight in the minimum jerk model. The hand paths of 16 human subjects were all curved.

Second, consider movements between two points while resisting a spring, one end of which is attached to the hand while the other is fixed. The minimum jerk model always predicts a straight path regardless of the external forces. On the other hand, the minimum torque-change model predicted a curved trajectory and an asymmetrical speed profile for the movement with the spring. These predictions are in close agreement with experiment data.

Third, we examined vertical movements which are affected by gravity. The minimum jerk model always predicts a straight path between two points. On the other hand, the minimum torque-change model predicted curved paths for large up and down movements, roughly straight paths for small fore and aft movements. The speed profiles were bell-shaped for both movements. These predictions are in close agreement with experiment data of Atkeson and Hollerbach (1985).

Finally, the most compelling evidence is about a pair of via-point movements. Consider two subcases, with identical start and end points, but with dictated mirror-image via-points. If one notices invariance of the objective function C_J of the minimum jerk model under translation, rotation and roll, it is easy to see that the minimum jerk model predicts identical paths with respect to roll as well as identical speed profiles for the two subcases. On the other hand, the minimum torque-change model predicts two different trajectories. For the concave path, the speed profile has two peaks. However, for the convex path, the speed profile has only one peak. These predictions are in close agreement with the human data (Uno et al., 1989).

Summarizing these comparisons, the trajectory derived from the minimum jerk model is determined only by the geometric relationship of the initial, final and intermediate points, whereas the trajectory derived from the minimum torque-change model depends not only on the relationship of these three points but also on the arm posture (in other words, the relative location of the shoulder for the three points), and external forces.

The minimum jerk model postulates the smoothest possible trajectory in task-oriented Cartesian coordinates, while the minimum torque-change model postulates the smoothest possible trajectory in the motor command space. This essential difference induces a difference in the capability of resolving ill-posed motor control problems shown in Fig. 2..1. The minimum jerk model formulated in task-oriented coordinates can only determine the desired trajectory in task-oriented coordinates, and hence can not resolve the ill-posed inverse kinematics or inverse dynamics problem for redundant manipulators. However, the minimum torque-change model can resolve all three ill-posed problems shown in Fig. 2..1 at the same time when the locations of the desired end point, desired via-points and obstacles are given in task-oriented coordinates.

If we adopt the minimum torque-change model as a computational scheme, it leads to two important conceptual assumptions. First, because the smoothness criterion is expressed in the motor command space which is more central than the task-oriented coordinates where movement conditions are expressed, in order to connect these two spaces the brain needs to acquire, by training, an internal model of the controlled object and continuously utilize it for trajectory formation. Second, the brain must solve all the three ill-posed problems shown in Fig. 2..1 simultaneously by a direct, jump-over computation shown in Fig. 1..1.

3.2 Cascade Neural Network Model

It was reported that some neural network models can solve computationally difficult nonlinear optimization problems such as the traveling salesman problem (Hopfield & Tank, 1985) or early visions (Poggio, Torre, & Koch, 1985). Since the dynamics of the human arm or a robotic manipulator is nonlinear, finding the unique trajectory which minimizes C_T is a nonlinear optimization problem. This is a rather difficult optimization problem since the smoothness criterion is represented in the motor command space while movement conditions such as locations of target points, via-points and obstacles are represented in task-oriented coordinates. To accommodate the two requirements simultaneously, some model of the controlled object must be used to convert constraints from one space to the other space.

The cascade neural network model shown in Fig. 3..1 was proposed to coherently resolve all three ill-posed problems shown in Fig. 2..1 based on the minimum torque-change criterion (Kawato, Maeda, Uno, & Suzuki, 1990). The model calculates the minimum torque-change trajectory and the corresponding necessary torque based on information about locations of the target point, via-points and obstacles, which is given by the higher motor center. This model is called the cascade neural network model since many network units are arranged in the cascade formation shown in Fig. 3..1. This cascade structure corresponds to the dynamical property of the controlled object, and it provides a forward model of the controlled object as a whole. The minimum torque-change criterion is embedded as hardware (electrical synapses) in the model. It first acquires a forward model of a controlled object by training and then calculates the motor command by relaxation computation utilizing the learned forward model. In a sense, the network first learns the energy which should be minimized, and then minimizes the learned energy. The minimum torque-change model is (i) a computational model for the trajectory formation problem. The cascade model provides understanding on the (ii) representation and algorithm level, and (iii) hardware level, for the same problem.

The controlled object in Fig. 3..1 stands for an arm, a body, legs, a speech articulator and so on. A state of the controlled object such as joint angles of the arm is denoted by θ. The time derivative of the state such as joint angular velocities is denoted by $\dot{\theta}$. The motor command is denoted by τ. The joint torque is a special case of the motor command. Generally, the controlled object is described by the following dynamical equations:

$$d\theta/dt = \dot{\theta} \tag{3.3}$$
$$d\dot{\theta}/dt = f(\theta, \dot{\theta}, \tau). \tag{3.4}$$

The cascade structure of the neural network model shown in Fig. 3..1 exactly corresponds to the temporal structure of the above dynamical equation. However, the above dynamical equation is represented in continuous time while the network adopts the discrete time representation. That is, the model consists of many identical four-layer network units, and the jth network unit corresponds to time $j\Delta t$. If there are N network units, the model can generate a movement up to duration of $N\Delta t$.

The network unit consists of four layers of neurons. The first layer represents the time course of the trajectory and the torque. The third layer calculates the change of the trajectory within a unit of time, that is $\Delta t \cdot f(\theta, \dot{\theta}, \tau)$. The fourth layer and the output line on the right side represent the estimated time course of the trajectory. Neurons in the fourth layer calculate the next state by summing the previous state with its change during the unit of time. In the above dynamical equation, the rate of change of the state depends only on the current state, the current time derivative of the state and the current value of the motor command. Correspondingly, each network unit only receives information about the current state, the current time derivative and the current motor command.

Operations of this network are divided into a learning phase and a pattern-generating phase. In the learning phase, the common input torque is fed to both the controlled object and the neural network model. The realized trajectory from the controlled object is used as a teaching signal to acquire the forward model between the first and the third layers of the network unit.

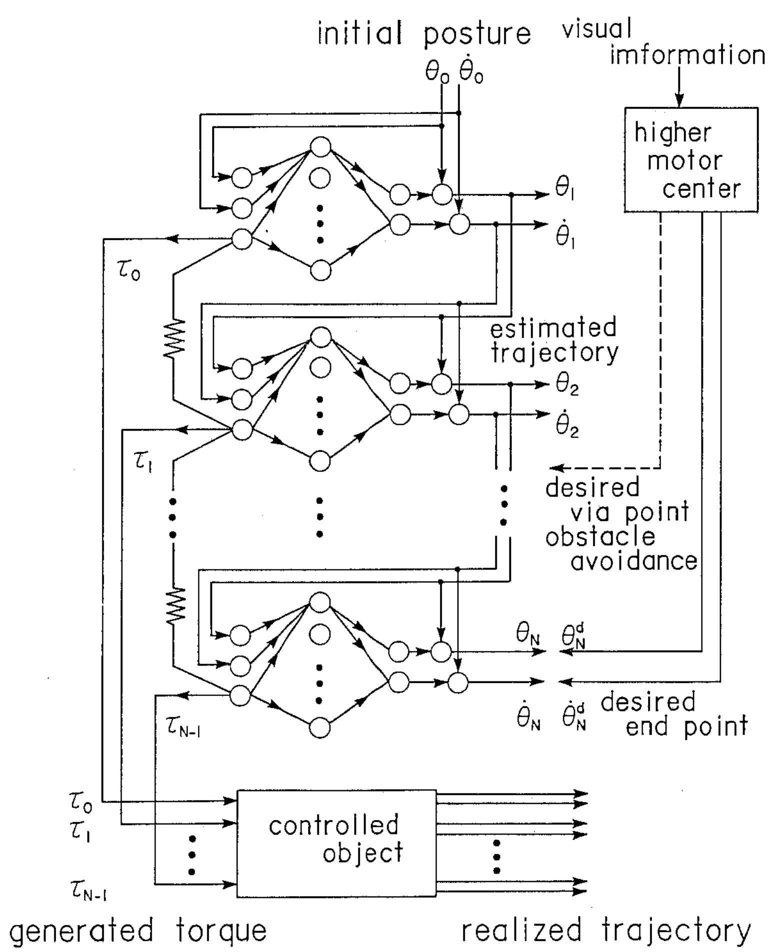

Figure 3..1: A repetitively structured cascade neural network model for trajectory formation based on the minimum torque-change criterion.

The back-propagation learning algorithm (Rumelhart, Hinton, & Williams, 1986) can be applied during this phase. In this feedforward calculation mode, the function of each neuron is assumed rather simple to be the linear weighted summation of synaptic inputs and the sigmoid nonlinear transformation, as widely assumed (Rumelhart et al., 1986).

Once this learning is completed, the cascade network provides a forward model of the controlled object. In the pattern-generating phase (Fig. 3..1), the cascade model computes the torque which realizes the minimum torque-change trajectory while satisfying various movement conditions by relaxation of its state variable τ according to the following dynamics. To guarantee the smoothness of the torque, electrical couplings between neurons representing torques at neighboring times in the 1st layer (see electrical resistance in Fig. 3..1) are activated. The electric current flow through the gap junction tends to equalize torque values at neighboring times and decreases the criterion C_T. The higher motor center gives information about locations of the desired target point, the desired via-point and the locations of obstacles to be avoided, to the 4th layer of the network units. Satisfaction of these movement conditions needs information conversion since these conditions are represented in task-oriented coordinates while the state variable τ of relaxation is in the motor command space.

One of the most attractive features of the multi-layer feedforward neural network model is that the network can calculate the partial derivative of its output with respect to its input in parallel using learned synaptic weights based on the error backpropagation algorithm once it acquires the mapping from its input to output. The cascade network utilizes this characteristic as follows. First, errors in task-oriented coordinates are calculated at the fourth layer of the network units as the difference between estimated hand positions and desired target and so on. That is, the desired target position plays the role of the teaching signal in conventional learning procedure. Then backpropagation of these errors all through the cascade structure is done using the algorithm of Rumelhart et al. (1986). This procedure converts the errors in the task-oriented coordinates into errors in the motor command space. In the forward calculation through the cascade structure, information flows from the past to the future, from top to bottom, in Fig. 3..1. On the other hand, in backpropagation calculation, information flows from the future to the past, from bottom to top, in Fig. 3..1. Thus, the backpropagation procedure in the cascade model is "backpropagation through time" (Werbos, 1988). The state of torque neuron changes in proportion to the calculated error. This guarantees that the error in the movement conditions is decreased.

Although the state of the cascade neural network model changes continuously in time and in parallel during the trajectory generation phase, for ease of understanding, network operations are listed serially as follows: (i) Assume initial torque values at N different times. (ii) The forward model of the controlled object acquired in the cascade structure estimates the trajectory at N different times from the given torque time course and the initial state of the controlled object. (iii) At the fourth layer of the network units, the estimated position of the hand is compared with the desired target, desired via-points, etc., and errors in the task-oriented coordinates are calculated. (iv) These errors are backpropagated all through the cascade structure to calculate errors for the torque neurons. They are errors in the torque space which are responsible for the estimated error in the task-oriented coordinates. (v) Electric current flows from the neuron with the larger torque value to that with the smaller torque value. (vi) Torque values of neurons in the first layer of the network units change in proportion to the two forces calculated in procedures in (iv) and (v). (vii) Return to procedure (ii) unless torque values reach equilibrium.

It can be mathematically shown that the cascade network settles down to a stable equilibrium point where the summation of the smoothness criterion multiplied by an electrical conductance of the gap junction and the error in movement conditions is minimum. Consequently, the required torque time course to generate the minimum torque-change trajectory can be calculated by relaxation. An appropriate delay line should be inserted between the first layer of the cascade network and the controlled object in Fig. 3..1 so that the controlled object is moved by this calculated motor command.

3.3 Fitts's Law Reproduced by Cascade Network

The cascade network executes the steepest descent motion with respect to the weighted sum of the smoothness criterion C_T of (3.2) and the hard constraint regarding movement conditions. The value of the electrical conductance is the weight of the smoothness term. The electrical conductance must be slowly decreased to zero so that the hard constraint is strictly satisfied. This is well known as the "penalty method" in optimal control theory. Furthermore, for the cascade model to calculate the exact minimum torque-change trajectory, the number of relaxation iterations needs to be sufficiently large. On the other hand, when the electrical conductance is fixed and the number of iterations is rather small, the cascade model can not calculate the exact torque, and the hand does not reach the desired target using the feedforward control alone. Thus, one observes an error between the final position and the desired target location. Fortunately, we found that this is not the weak point of our cascade model but rather its virtue. First, the cascade model reproduced the planning-time accuracy trade-off. That is, for a fixed electrical conductance, the final position error of the hand controlled by the cascade network alone decreases as the iteration number increases. Second, speed-accuracy trade-off of the arm movement, well known as Fitts's law (Fitts, 1954), was reproduced by the cascade model (Hirayama, Kawato, & Jordan, 1990).

We examined the speed-accuracy trade-off by the cascade network model with a fixed electrical conductance of 0.001 and a fixed iteration number of 2,500 (Hirayama et al., 1990). Five different point-to-point movements shown in Fig. 3..2a were examined. The start and target points (T1–T6) are the same as those in the human behavioral experiments by Uno et al. (1989). The origin of Fig. 3..2a is the location of the shoulder. Fig. 3..2a shows the hand paths and Fig. 3..2b shows the corresponding hand tangential velocities for five different movements with a 0.7 second duration. Hand paths are roughly straight and hand velocities are bell shaped. Thus, major qualitative features of human multi-joint movement were reproduced, even though the conductance and the number of iteration were fixed. Then, the cascade network generated the 5 trajectories for 6 different movement durations (0.5, 0.6, 0.7, 0.8, 0.9, 1.0 second). This can be done by a single network while changing the number of the network unit to which the target position is given. Fig. 3..2c plotted the movement time MT as a function of the right hand side of the following Fitts's law equation for these 30 movements.

$$MT = a + b \log_2(2A/W), \tag{3.5}$$

where A is the movement amplitude, W is the target width, a and b are constants. Fig. 3..2c shows that the cascade network reproduced Fitts's law quite well. In this plot, the final position error of the hand is identified with the target size W.

The classical explanation of Fitts's law invokes feedback corrections at long intervals (see for example Keele, 1986). We think this explanation breaks down if one considers a relatively long feedback delay. The loop time, which consists of the sensory processing including photoreceptors in the retina, planning and motor command generation and activation of muscles, may exceed 100 milliseconds. If one has experience with conventional feedback control, it is evident that control of say 700-millisecond movement with 100-millisecond feedback

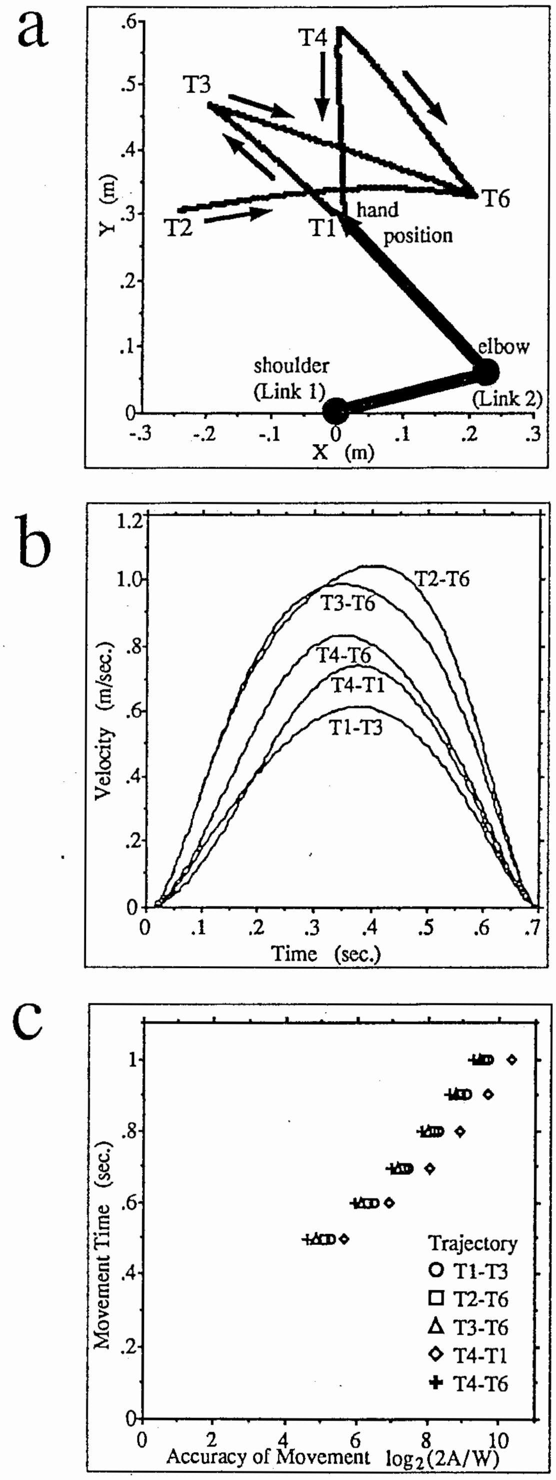

Figure 3..2: a. 5 hand paths produced by the cascade neural network model with a fixed electrical conductance of 0.001 and a fixed iteration number 2500 for 0.7 second movement duration. b. Hand tangential velocity profiles for the hand paths shown in a. c. Movement time-accuracy trade-off predicted by the cascade neural network model.

delay is very difficult.

Even a very elegant and comprehensive theory "stochastic optimized-submovement models" proposed by Meyer, Smith, Kornblum, Abrams & Wright (1990) relies on the feedback signal to start a secondary corrective submovement in order to hit the target. Because the typical oscillation of movement velocity or acceleration around the end of the first ballistic movement is continuous and does not contain any jerky component which is a sign of delay of feedback signal, the submovement models do not seem to be compatible with existence of the feedback delay which is at least 50 milliseconds even for somatosensory information. This is because according to the submovement hypotheses there must exist at least 50 milliseconds dead time before the motor command for the second submovement is effected after the final position error of the first ballistic movement is detected. We think that the submovements of Meyer et al. should be considered as results of physical oscillation due to visco-elastic properties of the musculoskeletal system in conjunction with a controller which maintains postures. We will later explain one candidate of this posture controller: the inverse statics model (Katayama & Kawato, 1990), based on spring-like properties of agonist and antagonist muscles.

On the contrary, we totally agree the viewpoint that variability in motor-output processes is responsible for errors in rapid movements, which is the basic assumption of the above model. This concept was originally proposed as the impulse variability model, and validated by behavioral experiments of controlling ballistic force pulse (Schmidt, Sherwood, Zelaznik, & Leikind, 1985). We think our study provides one possible neural mechanism which explains the stochastic variability of the time course of the feedforward motor command. From simulations of Hirayama et al. (1990) and Uno & Suzuki (1990), we can infer that the calculated feedforward torque contains stochastic variability which is induced by variability in iteration numbers of relaxation computation, variability in electrical resistance values (Hirayama et al., 1990), or variability in learned forward model (Uno & Suzuki, 1990). Furthermore, the cascade model explains this feedforward-torque variability for multiple degree of freedom controlled object with a realistic dynamics. This dynamics contains centripetal and Coriolis forces which represent interactions between different freedoms and frictional forces. These realistic forces reject basic assumptions of Meyer et al. (1990) such as force time rescalability or symmetry.

Results shown in Fig. 3..2c tempted us to invoke a totally feedforward mechanism which gives Fitts's law. However, we do not intend to totally deny the role of the feedback loop in movement. We are concerned about the concavity of the Fitts's law functions in Fig. 3..2c. If some visual feedback is utilized for the long movements, this might be enough to pull the concavity down. We hypothesize that the initial ballistic part of the movement is controlled by an entirely feedforward mechanism like the cascade network, and the later oscillatory part is explained by spring-like properties of the musculoskeletal system combined with a posture controller which specifies levels of stationary motor commands for groups of muscles based on the sensory feedback information about the target location. However, we emphasize that the use of feedback information is stationary such as by the inverse statics model.

In summary, speed-accuracy trade-off could be explained by difficulty in feedforward neural calculation of ballistic motor command for a multiple degree of freedom controlled object. Although our present study is very primitive compared with those in cognitive science (for example very elegant and comprehensive model by Meyer et al., 1990), for the first time, it tries to explain the motor command variability based on a specific, neural model of motor command generation mechanism. Furthermore the current model expands a theory from a single degree of freedom with oversimplified dynamics to a coordinated movement with realistic dynamics. Probably, the strongest point of the cascade model is that it can reproduce both quantitative features of multi-joint movements and Fitts's law.

3.4 Minimum Motor-Command-Change Trajectory

Although the minimum torque-change criterion was found and ascertained for arm movement, it does not depend on any special feature of the arm as a controlled object. We believe that a computational principle such as the minimum torque-change model must be independent of the controlled object, but inherent in the central nervous system itself. The structure of the cascade network (gap junction) also suggests that the computational principle has its origin in the central nervous system rather than in the periphery. It is known that primate corticomotoneuronal cells in the motor cortex represent muscle forces (Cheney & Fetz, 1980). In controlling the musculo-skeletal system with agonist and antagonist muscle groups for each joint, joint torque is an inappropriate control variable and muscle tensions seem more suitable. Thus, it is natural to assume that a variant of the minimum torque change model, such as the minimum muscle-tension-change model is more appropriate for musculoskeletal systems or other controlled objects such as a speech articulator.

Uno, Suzuki and Kawato (1989) examined the minimum muscle-tension-change model for a two-link manipulator with 6 muscles as a model of the human arm. We found that the minimum muscle-tension-change model is better than the minimum torque-change model in the sense that it can reproduce human data for wider range of physical parameters of the arm. Furthermore, we even postulate that the final answer to the trajectory formation problem is the minimum motor-command-change model since the motor command is the variable directly represented in the central, motor-control neural network.

It is well known that an articulator shows a very smooth movement whereas the phoneme sequence sounds discrete and distinctive to us. This kind of data seems to support that a smoothness constraint such as the minimum motor-command-change model exists also for the articulatory movement, that is speech.

In speech synthesis, a long series of phonemes must be uttered continuously. From the trajectory formation standpoint, this implies that many via-points are specified for one continuous trajectory. Thus, trajectory formation of via-point movement is very important for articulatory movement. The cascade neural network model can generate a trajectory which passes through a specified via-point. The time when the via-point must be passed may or may not be specified. Even when the time is not specified, the cascade network can find the best time to pass the via-point on the basis of the minimum torque-change criterion (Kawato et al., 1990). This may be called the intrinsic timing plan. This, intrinsic timing control capability, is one of the most attractive features of our cascade network.

4. Computational Schemes for Supervised Motor Learning

4.1 Three Learning Schemes

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. For example, parents teach correct pronunciation of words to children in the sound space. However, they can not directly show firing patterns of nerve fibers innervating articulator muscles. Other examples are: small children mimic various movement patterns of older children, beginners in some sports learn by watching expert athletes.

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. It is generally possible to compare this realized trajectory with the desired movement pattern (teaching signal) and to calculate the error between the two patterns. 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 (Jordan, 1990; Jordan, & Rumelhart, 1990) and termed, "supervised learning with a distal teacher". They proposed the forward and inverse modeling approach to resolve the problem. Their approach will be discussed later in connection with our approach. Barto (1990) reviewed this problem and compared several different approaches.

We proposed the *feedback-error-learning* neural network as a model of the lateral cerebellum and the parvocellular part of the red nucleus (Tsukahara, & Kawato, 1982; Kawato et al., 1987). This model constitutes one possible answer to the above error conversion problem.

Three representative computational schemes to resolve the problem are reviewed and compared. The objective of these learning schemes is to acquire a feedforward controller for an unknown controlled object. A perfect feedforward control can be realized if the feedforward controller provides an inverse model of the controlled object. An inverse dynamics model and an inverse kinematics model are formulated in the following. θ denotes an n-dimensional vector which represents body coordinates, such as joint angles or muscle lengths, of a controlled object. $\dot{\theta}$ represents the corresponding velocity vector. 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 same ordinary differential equations shown in the previous section.

$$\begin{aligned} d\theta/dt &= \dot{\theta}, \\ d\dot{\theta}/dt &= f(\theta, \dot{\theta}, \tau), \end{aligned} \tag{4.1}$$

here f is an n-dimensional nonlinear vector function. The forward dynamics problem is to find the body space trajectory $(\theta(t), \dot{\theta}(t))$ when the motor command $\tau(t)$ is given. Conversely, the inverse problem is to find the motor command $\tau(t)$ which realizes a given trajectory $(\theta(t), \dot{\theta}(t))$.

Next, the forward kinematics and inverse kinematics problems are formulated. x denotes a k-dimensional vector representing 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:

$$x = G(\theta), \tag{4.2}$$

here G is a k-dimensional nonlinear vector function. The forward kinematics is to determine x from θ based on the above equation. The inverse kinematics is to compute θ from x.

The problem of feedforward control is to find the motor command $\tau_d(t)$ which realizes the desired movement pattern $x_d(t)$. This could be done if one can solve first the inverse kinematics problem, and then the inverse dynamics problem. 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 (4.1). Although this is an implicit equation with respect to τ, it is rewritten in an 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. 4..1, three computational approaches for learning the inverse model of a controlled object are compared. They are somewhat independent of types of neural network models which actually constitute the inverse model.

The simplest approach is shown in Fig. 4..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 by Jordan & Rosenbaum (1988). Direct inverse modeling was proposed and used by Albus (1975), Miller, Glanz, & Kraft (1987), Kuperstein (1988) and Atkeson & Reinkensmeyer (1988).

Fig. 4..1b shows the method of combining a forward model and an inverse model, proposed by Jordan and Rumelhart (Jordan, 1990; Jordan & Rumelhart, 1990). 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.

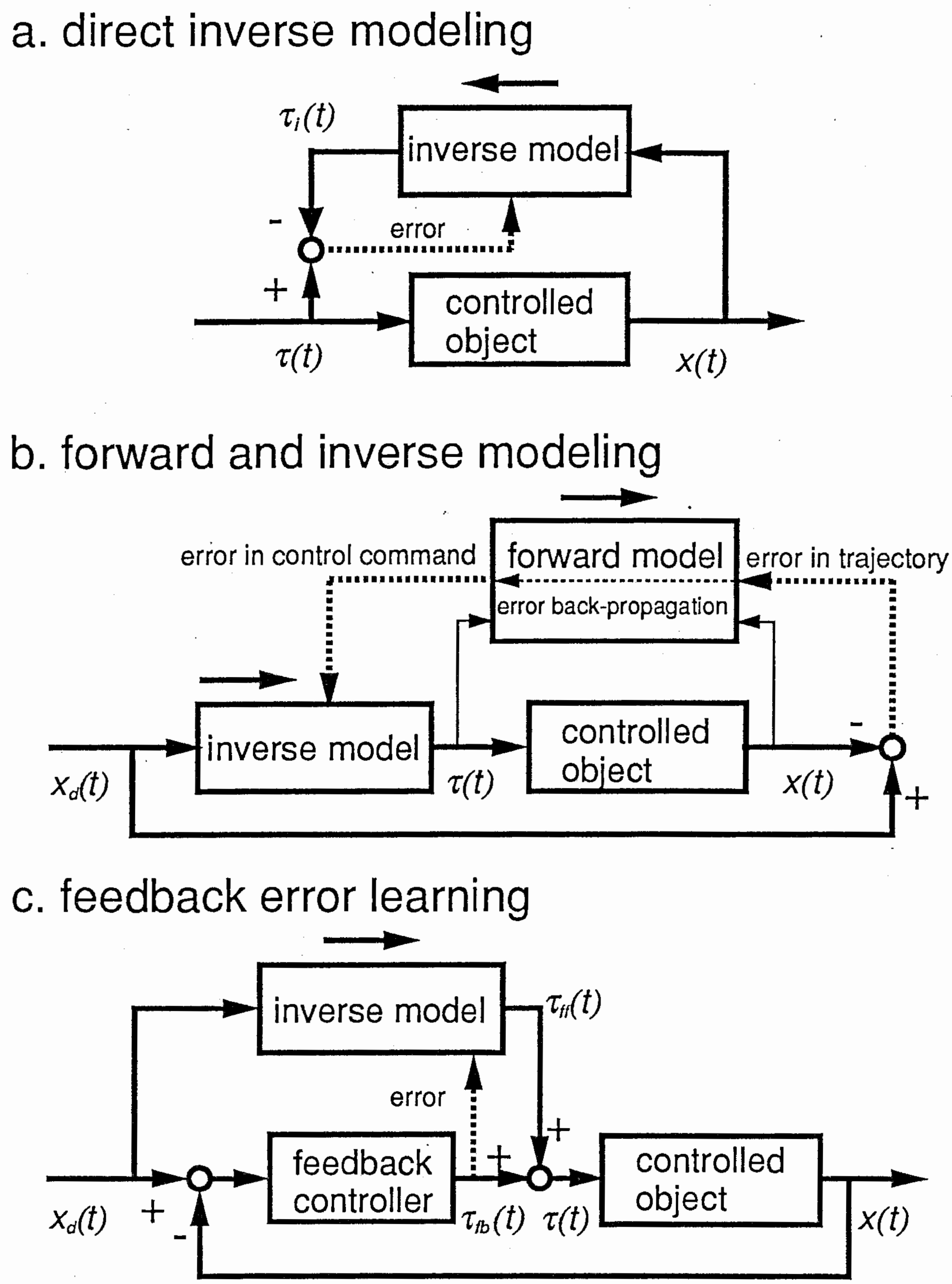

Figure 4..1: Three computational schemes for learning inverse model of a controlled object: a) Direct inverse modeling. b) Forward and inverse modeling. c) Feedback error learning scheme.

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. 4..1c shows the alternative computational approach which we proposed and termed *feedback error learning* (Kawato et al., 1987). The total torque $\tau(t)$ fed to the controlled object is the sum of the feedback torque $\tau_{fb}(t)$ and the feedforward torque $\tau_{ff}(t)$, 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.

4.2 Stability of Feedback-Error-Learning

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" (Jordan & Rosenbaum, 1988).

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.

Stability of the feedback-error-learning scheme was mathematically proved based on the notion that it can be regarded as a Newton-like method in a functional space (Kawato, 1990). First, a functional F from the motor command τ

to the error in trajectory is defined as follows: $F(\tau) = \theta_d - \theta(\tau)$. Because τ and θ are both functions of time t, F is a functional. The inverse dynamics problem is to calculate τ_d from θ_d, and hence is equivalent to obtaining the zero of the functional F. If the well known Newton method is used for this problem, the inverse derivative of F must be calculated. It can be calculated by using the variational equation of the dynamics equation (4.1) of the controlled object. The change in the motor command $\delta\tau$ for each iteration step is calculated by the Newton method from the error in trajectory $F(\tau) = \delta\theta = \theta_d - \theta$ as follows.

$$\begin{aligned} \delta\tau &= F'^{-1}(\tau)\delta\theta \\ &= (\partial f(\theta,\dot{\theta},\tau)/\partial\tau)^{-1}\{-\partial f(\theta,\dot{\theta},\tau)/\partial\theta \cdot \delta\theta - \partial f(\theta,\dot{\theta},\tau)/\partial\dot{\theta} \cdot \delta\dot{\theta} + \delta\ddot{\theta}\} \end{aligned} \quad (4.3)$$

Although this equation looks quite complicated, all calculation can be done by temporal differentiation of the trajectory error $\delta\theta$ and matrix manipulation. 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:

$$\delta\tau = M(\delta\theta) = K_P(\theta_d - \theta) + K_D(\dot{\theta}_d - \dot{\theta}) + K_A(\ddot{\theta}_d - \ddot{\theta}). \quad (4.4)$$

This approximation is validated since the product factor $(\partial f(\theta,\dot{\theta},\tau)/\partial\tau)^{-1}$ in the third term of (4.3) 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 (4.3) 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 (4.3) 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 functional F essentially gives the forward dynamics $\theta(\tau)$. F'^{-1} is the inverse of its derivative, and hence provides a linear inverse dynamics model. Thus, it becomes clear that the feedback controller in the feedback-error-learning scheme plays the role of a linear approximation of the inverse dynamics model. In this sense, the forward and inverse modeling of Jordan & Rumelhart (1990) and the feedback-error-learning scheme are in sharp contrast with each other. In the former scheme, *backpropagation* through the *forward model* of the controlled object converts trajectory error into motor command error. In the latter scheme, *a linear approximation of the inverse model* (i.e. feedback controller) converts trajectory error into motor command error.

Stability of the feedback-error-learning scheme can be proved for a learning control system shown in Fig. 4..1c. The dynamics of the controlled object is described by (4.1). The neural network IDM calculates τ_{ff} from the desired trajectory θ_d and the synaptic weights w:

$$\tau_{ff} = \psi(w, \theta_d, \dot{\theta}_d, \ddot{\theta}_d). \tag{4.5}$$

The shape of the function ψ depends on the kind of neural network model used for the IDM. The synaptic modification rule of the feedback-error-learning scheme is represented in a general manner as follows:

$$dw/dt = (\partial \tau_{ff}/\partial w)^T \tau_{fb}. \tag{4.6}$$

The learning stability and trajectory stability of the feedback-error-learning scheme were proved based on two reasonable assumptions. The learning stability implies that the synaptic weight w asymptotically converges to the optimal value $\tilde{w}$ which realizes the perfect IDM as follows: $\psi(\tilde{w}, \theta_d, \dot{\theta}_d, \ddot{\theta}_d) = h(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)$. The trajectory stability implies that the realized trajectory θ asymptotically converges to the desired trajectory θ_d. For mathematical proof, we utilized a Liapunov function for an average equation of the total system which contains the controlled object, the feedback controller, the feedforward neural network controller and its learning rule (Kawato, 1990).

4.3 Feedback-Error-Learning for Ill-posed Problems

For clarity, I explained the feedback-error-learning scheme in the simple case where the inverse kinematics and the inverse dynamics are well-posed problems, that is G^{-1} and h both exist. For a kinematically redundant controlled object with $n > k$, G^{-1} is one to many and hence can not generally be defined. For a dynamically redundant controlled object with $m > n$, $h(\theta, \dot{\theta}, \ddot{\theta})$ is one to many and hence can not generally be defined. The feedback-error-learning scheme can still resolve these ill-posed inverse kinematics and inverse dynamics problems, even for redundant controlled objects such as human arms. Furthermore, the obtained solution becomes an approximation to the minimum motor-command-change trajectory.

The direct inverse modeling method can not cope with the learning control of a redundant controlled object. Jordan (1990) clearly explained the reason for this in the one-to-many inverse kinematics problem. The forward and inverse modeling approach can resolve the ill-posedness 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 this desirable characteristic 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 (Kano, Kawato, Uno, & Suzuki, 1990; Katayama & Kawato, 1990).

Kano et al. (1990) succeeded in learning trajectory control within the stereo camera coordinates even when an industrial manipulator PUMA was given an extra degree of freedom. 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 torque-change model.

Katayama & Kawato (1990) 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.

We explain one example of a feedback controller to clarify the difference between the minimum muscle-tension-change model and the minimum motor-command-change model. Consider a single joint with flexor and extensor muscles. Tensions of these muscle can be represented by the following quite simplified model.

$$\begin{aligned} T_f &= M_f - kM_f\theta - bM_f\dot{\theta}, \\ T_e &= M_e + kM_e\theta + bM_e\dot{\theta}. \end{aligned} \quad (4.7)$$

Here, M_f and M_e are motor commands for the flexor and extensor, θ is the joint angle, and k and b are elasticity and viscosity coefficients respectively. This model reflects experimental data that both stiffness and viscosity of muscles increase with their tensions. Muscle tensions are not directly proportional to motor commands. Small changes in the two motor commands are decomposed into a coactivation component C and a bias component B between agonist and antagonist muscles.

$$\begin{aligned} C &= (\Delta M_f + \Delta M_e)/2, \\ B &= (\Delta M_f - \Delta M_e)/2. \end{aligned} \tag{4.8}$$

It is noted that the cocontraction component C mainly controls the mechanical impedance of the arm while the difference component B mainly controls the virtual equilibrium trajectory (see Hogan, 1984).

$$\Delta\tau = a(1+r, -1+r)\begin{pmatrix} \Delta M_f \\ \Delta M_e \end{pmatrix} = J_D \begin{pmatrix} \Delta M_f \\ \Delta M_e \end{pmatrix}. \tag{4.9}$$

Here, a is a moment arm of muscles and $r = -(k\theta + b\dot{\theta})$. r eventually turns out to be the ratio of the coactivation component to the bias component of pair of muscles, C/B. J_D is the Jacobian matrix from small changes in the motor commands to the small change in the joint torque. We use the Moore-Penrose pseudoinverse matrix $J_D^{\#}$ of the Jacobian for design of the feedback controller so that the change in the motor command is minimal.

$$\begin{pmatrix} \Delta M_f \\ \Delta M_e \end{pmatrix} = J_D^{\#}\Delta\tau = \frac{1}{2a(1+r^2)}\begin{pmatrix} 1+r \\ -1+r \end{pmatrix}\{K_P(\theta_d - \theta) + K_D(\dot{\theta}_d - \dot{\theta}) + K_F(F_d - F)\}. \tag{4.10}$$

Here θ_d and F_d are desired joint angle and force, and K_P, K_D and K_F are feedback gains. This equation indicates that the coactivation C prevails more for the larger θ and the larger $\dot{\theta}$. Thus, this model may explain the coactivation EMG pattern seen in the final phase of a fast movement, and coactivation for exerting considerable force at a posture far from the rest posture. These feedback motor commands are accumulated in the feedforward neural network controller by the feedback-error-learning. Thus, the trajectory realized by the feedback-error-learning scheme with the Moore-Penrose pseudoinverse approximates the minimum motor-command-change model, which minimizes the following criterion:

$$C_M = 1/2\int_0^{t_f}\{(\frac{dM_f}{dt})^2 + (\frac{dM_e}{dt})^2\}dt. \tag{4.11}$$

Another characteristics of our work on SoftArm was that two inverse models were automatically trained for trajectory control (Katayama & Kawato, 1990). The one is the usual inverse dynamics model (IDM) which compensates dynamic forces due to movements of links. IDM is trained, that is fed of the feedback command as the error signal when the arm is moving. The other inverse model is the inverse statics model (ISM) which solves the static equilibrium problem between agonist and antagonist muscle groups. ISM is trained when the arm is in a static posture. When a desirable static posture is given, ISM calculates motor commands for muscle groups to attain the posture while taking into account of nonlinear spring-like properties of muscles and statics equilibrium between opposing muscles. In some sense, IDM takes care of an open-link architecture of the skeletal system while ISM takes care of a closed-loop architecture of the muscle system.

4.4 Simultaneous Learning of Feedback and Feedforward Controller

As shown in the previous section, design of the feedback controller which calculates the feedback command with the smallest norm is not a trivial task for kinematically and dynamically redundant controlled objects. Thus the feedback controller itself should be learned by monitoring both the input and the output from the controlled object. The feedback controller calculates the small change in the motor command ΔM from the small change in the task-oriented coordinates Δx by multiplying it with the Moore-Penrose pseudoinverse matrix $J^{\#}(\theta)$ which nonlinearly depends on the body coordinates θ.

$$\Delta M = J^{\#}(w, \theta)\Delta x. \tag{4.12}$$

Here, w is the synaptic weight for calculation of $J^{\#}$ from θ based on a specific structure of a neural network for $J^{\#}$. For acquiring the smallest norm solution, we add a linear decay term in a synaptic modification rule as follows.

$$dw/dt = \sum_{i,j}(\partial J^{\#}_{ij}/\partial w)^T[\Delta x\{\Delta M - J^{\#}(w, \theta)\Delta x\}^T - \lambda J^{\#}(w, \theta)]_{ij}. \tag{4.13}$$

The learning should occur on line while monitoring the previous motor command change ΔM with the present state change Δx. Simultaneous learning of the feedback and the feedforward controller, with the shorter and the longer time constants respectively, during real time control is expected to find the unique solution which is an approximation of the minimum motor-command-change trajectory for a redundant controlled objects.

4.5 Feedback-Error-Learning Neural Network Models for 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 (Ito, 1984). Given this histological uniformity of the cerebellar cortex and different functional modules, Ito (1970) stated: "What is the role of the cerebellum should thus be asked in the following two ways; i) common throughout the cerebellum, how a given portion of the cerebellum processes the incoming and outgoing information?; ii) specific to each part of the cerebellum, how a given portion is involved in regulation of a certain particular motor activity?". In this section, I try to answer these questions coherently based on the feedback-error-learning scheme.

The cerebrocerebellum is the lateral zone of the cerebellum. Its inputs originate exclusively in pontine nuclei which 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 (Tsukahara & Kawato, 1982; Kawato et al., 1987) based on the pioneering work of Ito (1970). Fig. 4..2d 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 which receive sensory information from the periphery. The vermis is related to control of posture. The intermediate part plays an adaptive role in control of locomotion. We proposed a closed loop control system based on the feedback-error-learning scheme as shown in Fig. 4..2b and c (Gomi & Kawato, 1990). They can be considered models of the vermis and the intermediate part, respectively. In these closed-loop control systems, the feedback controller plays a three-fold role. First, it converts the trajectory error into the motor-command error as a linear approximation of the inverse model of the controlled object.

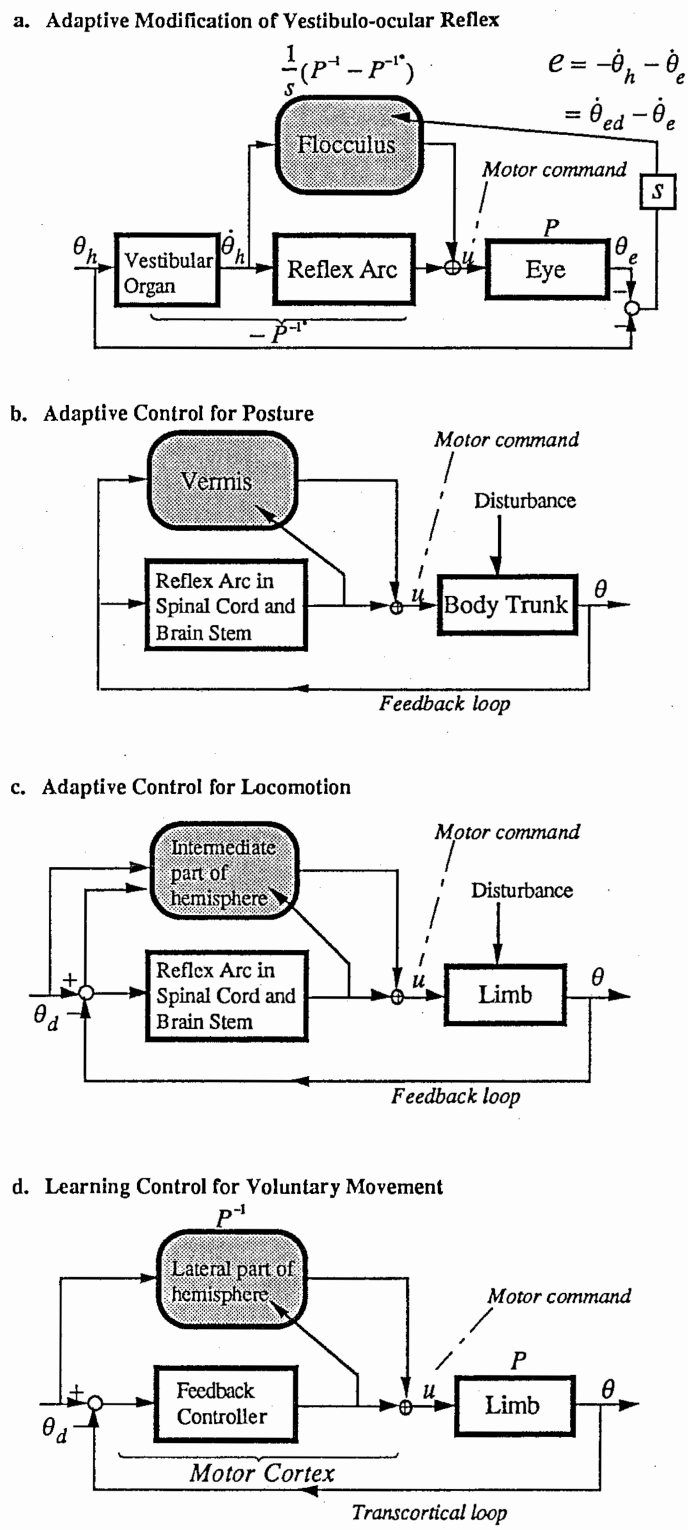

Figure 4..2: Functional roles of different parts of cerebellum interpreted based on the feedback error learning scheme: a) Flocculus for adaptive modification of the vestibulo-ocular reflex. b) Vermis for adaptive control of posture. c) Intermediate part for adaptive control of locomotion. d) Lateral hemisphere for learning of voluntary motor control.

Second, it guarantees global trajectory stability as a usual feedback controller. Third, it defines an inverse reference model in the model reference adaptive control. For example, if we prepare the PDA feedback controller in Cartesian space, it defines the mechanical impedance of the hand tip in the task-oriented coordinates. In this case, K_A determines the virtual inertia, K_D viscosity, and K_P stiffness. Consequently, the feedback-error-learning scheme in the closed loop system can perform Hogan's (1985) impedance control by learning.

The vestibulocerebellum occupies the flocculonodular lobe. The flocculus is known to play a role in adaptive modification of the vestibulo-ocular reflex (Ito, 1984). Its circuit diagram is shown in Fig. 4..2a. Since this system does not contain any feedback loop, it first appears that its function can not be understood in the feedback-error-learning formulation. However, the visual system plays the role of feedback controller in learning. The visual system which is an origin of the climbing fiber input calculates the negative of the summation of the head and eyeball velocities from the retinal slip: $-\dot{\theta}_h - \dot{\theta}_e$. Because the head velocity $\dot{\theta}_h$ is the negative of the desired velocity of eyeball for a perfect vestibulo-ocular reflex, the summation of the two velocities is just equal to the differential negative feedback term: $e = -\dot{\theta}_h - \dot{\theta}_e = \dot{\theta}_{ed} - \dot{\theta}_e$. Consequently, the function of the flocculus can also be understood from the feedback-error-learning concept, or more generally, as a Newton-like method in a functional space.

5. Integrated Model for Trajectory Formation and Learning

As mentioned earlier, there are two different approaches which resolve ill-posed motor control problems. One approach is to introduce a performance index. Another approach is to utilize a feedback controller. The feedback controller selects one specific motor command in the inverse dynamics and inverse kinematics problems even for redundant manipulators. The minimum jerk model formulated in task-oriented coordinates can resolve ill-posed trajectory formation problems, but can not resolve ill-posed inverse kinematics and inverse dynamics problems for redundant manipulators. The feedback control approach can not resolve the ill-posed trajectory formation problem in spite of the early hypothesis of the *end point control* (see Bizzi et al., 1984). Thus, a combination of these two approaches can resolve all three ill-posed problems. This is the step-by-step computational approach in Fig. 1..1. This has been studied by many researchers (Hogan, 1984; Flash, 1987; Mussa-Ivaldi, Morasso, &

Zaccaria, 1988; Massone & Bizzi, 1989; Kano, Kawato, Uno, & Suzuki, 1990) including us.

We hypothesize that the cascade network (direct, jump-over computation in Fig. 1..1) is used for very skilled movements, while step-by-step computation is utilized for relatively difficult or less skilled movements. That is, we suppose that the computational scheme adopted by the brain changes with motor learning. We have some experimental data which seem to support this idea. First, in the human arm movement with the external spring force (Fig. 6 of Uno et al., 1989), subjects first tended to generate trajectories of various shapes at the beginning of the experiment when they were still not accustomed to the spring. After tens of repetitions, subjects began to consistently generate a curved hand path, which is the minimum torque-change trajectory (Uno et al., 1989). Second, Uno et al. (unpublished observation) introduced nonlinear coordinate transformation between the hand position on a 2-dimensional position digitizer and the CRT coordinates where the end point, the start point and the hand position were displayed. Because of the nonlinear transformation, a straight line on the CRT corresponds to a curve on the digitizer, and vice versa. Subjects first generated roughly straight hand paths on the CRT. This is close to the minimum jerk trajectory in the visual task space (CRT coordinate). After several periods of training, they tended to generate roughly straight hand paths on the digitizer (i.e. curved paths on the CRT), which are the minimum torque-change trajectories (Uno et al., unpublished observation). These experiment data could be explained if the step-by-step computation is taken over by direct computation with motor learning. In the first case, the forward dynamics model of an arm in combination with the spring must be relearned. In the second case, the forward kinematics model of an arm in combination with the imposed nonlinear transformation between the digitizer and the CRT must be relearned. Thus, the step-by-step computation seems to be temporarily utilized until the forward model is relearned. Schneider and Zernicke (1989) reported decrease of jerk cost during practice. This might be explained as improved control performance caused by an intensive learning of forward dynamics and kinematics of the arm for a special task. Some motor control schemes, such as the equilibrium trajectory approach (Hogan, 1984; Flash, 1987), do not support efficient movement refinement during practice.

Although the cascade neural network model reproduced both the quantitative features of multi-joint trajectories and Fitts's law, it has at least two weak points as a model of the brain. The first is the large iteration number of relaxation calculation. When the initial torque was set zero, typical iteration number were from several hundreds to thousands, although we showed that this can be reduced as small as 50 by moving a virtual target point in the course of relaxation (Kitano, Kawato, Uno & Suzuki, 1990). The second is the necessity of backpropagation during the relaxation calculation. Backpropagation is biologically implausible. Backpropagation during the learning phase can be substituted by other learning algorithm such as the associative reward penalty learning (Barto & Anandan, 1985). However, backpropagation in relaxation calculation seems to be indispensable in the present form of the cascade model.

Fig. 5..1 shows a unified neural network model which integrates the feedback-error-learning scheme and the cascade neural network model, and hence can explain the above mentioned change of motor control strategy with learning. Furthermore, it is a natural extension of the cascade network model by resolution of the above two weak points. IDM stands for the inverse dynamics model and is learned by the feedback-error-learning. FDM stands for the forward dynamics model and is just the cascade neural network model in the feedforward calculation mode of the estimated trajectory from the motor command time course. Here, Ψ and Ψ^{-1} are forward and inverse dynamics, respectively.

For a relatively difficult movement, in which FDM is not accurate and movement planning time is not sufficient, one-shot calculation with only IDM from the minimum jerk trajectory θ_J is used. This is the step-by-step control strategy. However, if FDM is quite accurate and the planning time is sufficient, the total network relaxes its states so that the minimum motor-command-change trajectory is generated. In this relaxation, both the motor command τ and the trajectory θ change somewhat independently. This is the first point of essential differences between this unified model and the cascade neural network model, in which only τ is the relaxation state. The smoothness criterion is imposed on dynamical change on τ, and the boundary condition like target point is imposed on dynamical change on θ. Thus there is no need to transform one of the two constraints from the motor command space to trajectory space or vice versa. Consequently, no backpropagation is necessary in relaxation of this model. This is the second essential difference between the cascade model and this unified model. Compatibility between τ and θ is doubly guaranteed by the calculation of IDM and FDM as follows. That is, τ-dynamics contains a force $\Psi^{-1}(\theta) - \tau$ which brings τ closer to the estimated $\tau = \Psi^{-1}(\theta)$ by IDM from θ. Similarly, θ-dynamics contains a term $\Psi(\tau) - \theta$. Because the initial condition of τ is not zero and is equal to what realizes the minimum jerk trajectory θ_J, the relaxation iteration number is much smaller than the

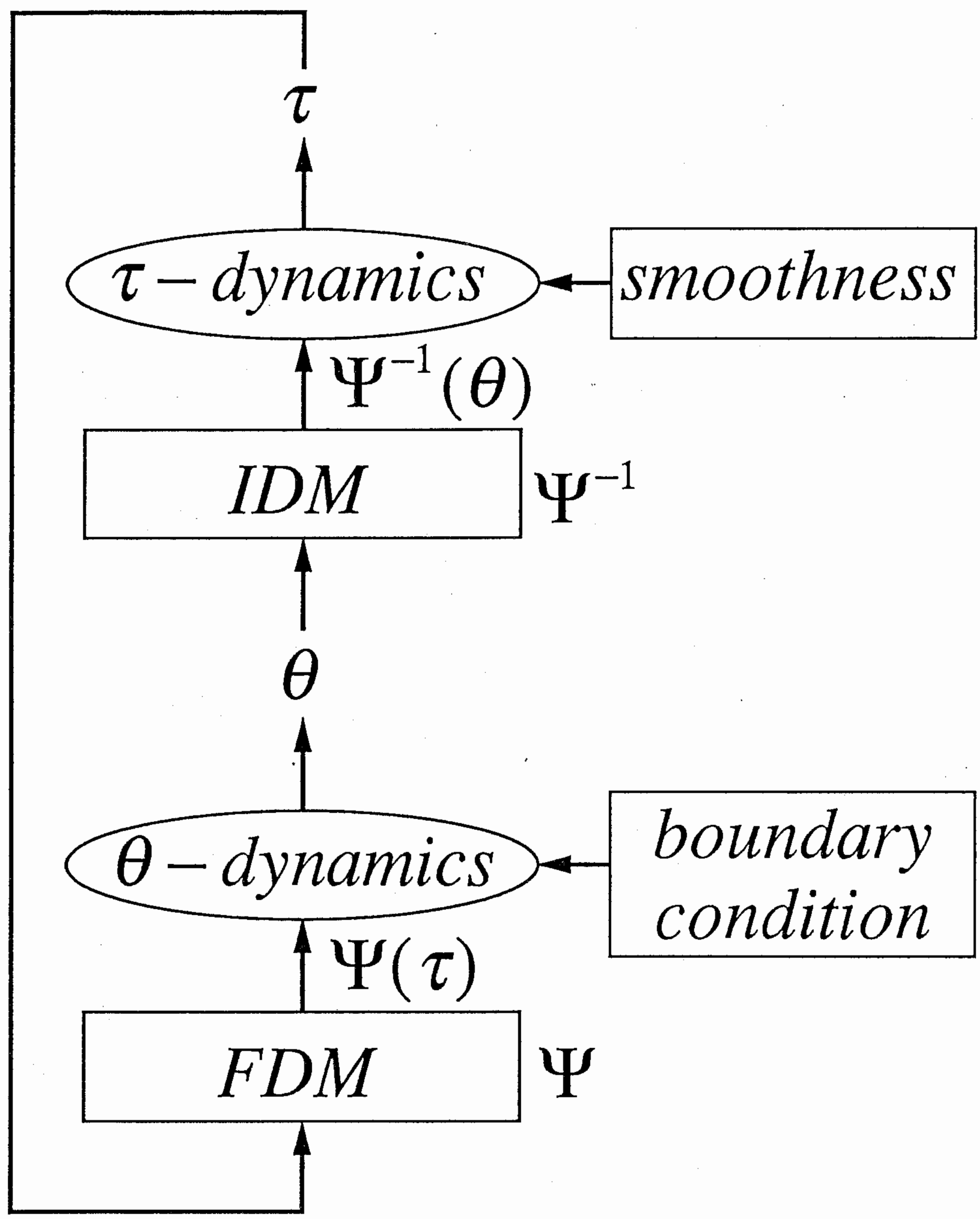

Figure 5..1: A unified neural network model for trajectory planning and trajectory learning. IDM is the inverse dynamics model which is acquired by the feedback-error-learning scheme. FDM is the forward dynamics model and is equivalent to the cascade network used for forward calculation.

cascade model.

6. Conclusions

In this chapter we developed computational theories and neural network models for trajectory formation and trajectory control. The minimum torque-change model reproduced various quantitative features of multi-joint arm trajectories. The cascade neural network model utilizes the forward model of the controlled object and backpropagation through it for relaxation calculation of the motor command, which realizes the minimum torque-change trajectory. The cascade model reproduced Fitts' law.

We then explained the feedback-error-learning approach to acquire the feedforward controller as an inverse model of the controlled object. A specific method to use pseudoinverse feedback controller is shown to be related with the minimum motor-command-change trajectory, which is a natural extension of the minimum torque-change trajectory.

Finally, we proposed an integrative computational and neural network model which utilizes both the forward and inverse models of the controlled object for relaxation calculation of the motor command. It does not use backpropagation in computation of the motor command and the trajectory. The iteration number is small. It naturally explains change of motor control strategy from the step-by-step approach, which plans the trajectory based on the minimum jerk criterion and executes it by the inverse model acquired by the feedback-error-learning, to the direct approach by relaxation calculation of the motor command based on the minimum motor-command-change criterion.

References

[1] Abend, W., Bizzi, E., & Morasso, P. (1982). Human arm trajectory formation. *Brain,* **105**, 331–348.

[2] Albus, J. S. (1971). A theory of cerebellar functions. *Mathematical Bioscience,* **10**, 25–61.

[3] Albus, J. S. (1975). A new approach to manipulator control: The cerebellar model articulation controller (CMAC). *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and, Control,* **97**, 220–227.

[4] Allen, G. I., & Tsukahara, N. (1974). Cerebrocerebellar communication systems. *Physiological Review,* **54**, 957–1006.

[5] Atkeson, C. G., & Hollerbach, J. M. (1985). Kinematic features of unrestrained vertical arm movements. *Journal of Neuroscience,* **5**, 2318–2330.

[6] Atkeson, C. G., & Reinkensmeyer, D. J. (1988). Using associative content-addressable memories to control robots. *Proceedings of IEEE Conference on Decision and Control,* 792–797, December 7-9, 1988, Austin, Texas.

[7] Barto, A. G. (1990). Connectionist learning for control: An overview. In T. Miller, R. Sutton, & P. Werbos (Eds.), *Neural networks for robotics and control.* Cambridge, MA: MIT Press.

[8] Barto, A.G., & Anandan, P.(1985). Pattern-recognizing stochastic learning automata. *IEEE Transactions on Systems, Man, and Cybernetics,* **SMC-15**, 36—375.

[9] Bizzi, E., Accornero, N., Chapple, W., & Hogan, N. (1984). Posture control and trajectory formation during arm movement. *Journal of Neuroscience,* **4**, 2738–2744.

[10] Cheney, P. D., & Fetz, E. E. (1980). Functional classes of primate corticomotoneuronal cells and their relation to active force. *Journal of Neurophysiology,* **44**, 773–791.

[11] Evartz, E. V. (1981). Role of motor cortex in voluntary movements in primates. In V. B. Brooks (Ed.), *Handbook of physiology* section 1: Volume II, part 2. (pp. 1083–1120). Bethesda: American Physiological Society.

[12] Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of moment. *Journal of Experimental Psychology,* **47**, 381–391.

[13] Flash, T. (1987). The control of hand equilibrium trajectories in multi-joint arm movements. *Biological Cybernetics,* **57**, 257–274.

[14] Flash, T., & Hogan, N. (1985). The coordination of arm movements: An experimentally confirmed mathematical model. *Journal of Neuroscience,* **5**, 1688–1703.

[15] Fujita, M. (1982). Adaptive filter model of the cerebellum. *Biological Cybernetics,* **45**, 195–206.

[16] Gomi, H., & Kawato, M. (1990). Learning control for a closed loop system using feedback-error-learning. *Proceedings of IEEE Conference on Decision and Control,* 1990, December, Hawaii.

[17] Hirayama, M., Kawato, M., & Jordan, M. I. (1990). Speed-accuracy trade-off of arm movement predicted by the cascade neural network model. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics,* 1990, November, Los Angeles.

[18] Hogan, N. (1984). An organizing principle for a class of voluntary movements. *Journal of Neuroscience,* **4**, 2745–2754.

[19] Hogan, N. (1985). Impedance control: an approach to manipulation: part I - theory. *Journal of Dynamics Systems, Measurement, and Control,* **107**, 1–7.

[20] Hollerbach, J. M. (1982). Computers, brains and the control of movement. *Trends in Neuroscience,* **5**, 189–192.

[21] Hopfield, J. J., & Tank, T. W. (1985). "Neural" computation of decisions in optimization problems. *Biological Cybernetics,* **52**, 141–152.

[22] Ito, M. (1970). Neurophysiological aspects of the cerebellar motor control system. *International Journal of Neurology,* **7**, 162–176.

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

[24] Jordan, M. I. (1990) Motor learning and the degrees of freedom problem. In M. Jeannerod (Ed.), *Attention and performance XIII.* (pp.796-836). Hillsdale: Lawrence Erlbaum Associates.

[25] Jordan, M. I., & Rosenbaum, D. A. (1988). Action. In M. I. Posner (Ed.), *Handbook of cognitive science.* Massachusetts: MIT Press.

[26] Jordan, M. I., & Rumelhart, D. E. (1990). Supervised learning with a distal teacher. submitted to: *Cognitive Science.*

[27] Kano, M., Kawato, M., Uno, Y., & Suzuki, R. (1990). Learning trajectory control of a redundant arm by feedback-error-learning. *Japan IEICE Technical Report,* **NC89-61**, 1–6. (in Japanese)

[28] Katayama, M., & Kawato, M. (1990). Learning trajectory and force control of an artificial muscle arm by parallel-hierarchical nerual network model. Proceedings of IEEE Neural Information Processing Systems 1990.

[29] Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movement. *Biological Cybernetics,* **57**, 169–185.

[30] Kawato, M., Isobe, M., Maeda, Y., & Suzuki, R. (1988). Coordinates transformation and learning control for visually-guided voluntary movement with iteration: A Newton-like method in a function space. *Biological Cybernetics,* **59**, 161–177.

[31] Kawato, M. (1990). Feedback-error-learning neural network for supervised motor learning. In R. Eckmiller (Ed.), *Neural networks for sensory and motor systems,* Amsterdam: Elsevier.

[32] Kawato, M., Maeda, M., Uno, Y., & Suzuki, R. (1990). Trajectory formation of arm movement by cascade neural network model based on minimum torque-change criterion. *Biological Cybernetics,* **62**, 275–288.

[33] Keele, S. W. (1986). Motor control. In J. K. Boff, L. Kaufman, & J. P. Thomas (Eds.), *Handbook of perception and human performance,* Vol. II (pp. 30-1–30-60). New York: Wiley.

[34] Kelso, J. A. S., Southard, D. L., & Goodman, D. (1979). On the nature of human interlimb coordination. *Science,* **203**, 1029–1031.

[35] Kitano, M., Kawato, M., Uno, Y., & Suzuki, R. (1990). Optimal trajectory control by the cascade neural network model for industrial manipulator. *Japan IEICE Technical Report,* **NC89-64**, 19–24. (in Japanese)

[36] Kuperstein, M. (1988). Neural model of adaptive hand-eye coordination for single postures. *Science,* **239**, 1308–1311.

[37] Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology,* **202**, 437–470.

[38] Marr, D. (1982). *Vision.* New York: Freeman.

[39] Massone, L., & Bizzi, E. (1989). A neural network model for limb trajectory formation. *Biological Cybernetics,* **61**, 417–425.

[40] Meyer, D. E., Smith, J. E. K., Kornblum, S., Abrams, R. A., & Wright, C. E. (1990). Speed-accuracy tradeoffs in aimed movement: Toward a theory of rapid voluntary action. In M. Jeannerod (Ed.), *Attention and performance XIII.* (pp.173-226). Hillsdale: Lawrence Erlbaum Associates.

[41] Miller, W. T., Glanz, F. H., & Kraft, L. G. (1987). Application of a general learning algorithm to the control of robotic manipulators. *International Journal of Robotics Research,* **6**, 84–98.

[42] Morasso, P. (1981). Spatial control of arm movements. *Experimental Brain Research,* **42**, 223–227.

[43] Mussa-Ivaldi, F. A., Morasso, P. & Zaccaria, R. (1988). Kinematic networks - A distributed model for representing and regularizing motor redundancy, *Biological Cybernetics,* **60**, 1–16.

[44] Poggio, T., Torre, V., & Koch, C. (1985). Computational vision and regularization theory. *Nature,* **317**, 314–319.

[45] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature,* **323**, 533–536.

[46] Saltzman, E. L. (1979). Levels of sensorimotor representation. *Journal of Mathematical Psychology,* **20**, 91–163.

[47] Schneider, K., & Zernicke, R. F. (1989). Jerk-cost modulations during the practice of rapid arm movements. *Biological Cybernetics,* **60**, 221–230.

[48] Schmidt, R. A., Sherwood, D. E., Zelaznik, H. N., & Leikind, B. J. (1985). Speed-accuracy trade-offs in motor behavior: Theories of impulse variability. In H. Heuer, U. Kleinbeck & K. H. Schmidt (Eds.), *Motor behavior: Programming, control, and acquisition* (pp.79-123). Berlin: Springer-Verlag.

[49] 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 S. Amari & M. A. Arbib (Eds.), *Competition and cooperation in neural nets.* (pp.430–441). New York: Springer-Verlag.

[50] Uno, Y., Kawato, M., & Suzuki, R. (1989). Formation and control of optimal trajectory in human multijoint arm movement – minimum torque-change model –. *Biological Cybernetics,* **61**, 89–101.

[51] Uno, Y., Suzuki, R., & Kawato, M. (1989). Minimum muscle-tension-change model which reproduces human arm movement. *Proceedings of the 4th Symposium on Biological and Physiological Engineering,* 299–302. in Japanese.

[52] Uno, Y., & Suzuki, R. (1990). Quantitative relationship between internal model of motor system and movement accuracy, movement distance and movement time. *Proceedings of 1990 Annual Conference of Japan Neural Network Society,* in Japanese.

[53] Werbos, P. J. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural Networks*, **1**, 339–356.