数理解析研究所講究録
第 678 巻 1989 年 26-42

Neural Network Models for Formation and Control of Multi-joint Arm Trajectory

川人 光男
Mitsuo Kawato

ATR Auditory and Visual Perception Research Laboratories,
Twin 21 Bldg. MID Tower, Shiromi 2-1-61,
Higashi-ku, Osaka 540 Japan

Running Headline: Formation and Control of Trajectory

A computational model for voluntary movement is proposed (Fig. 1) which accounts for Marr's [15] first level for understanding complex information-processing systems: i.e., computational theory.

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 the task-oriented coordinates must be selected from out of an infinite number of possible trajectories, which lead to the glass whose spatial coordinates are provided by the visual system (trajectory determination in Fig. 1). Second, the spatial coordinates of the desired trajectory must be reinterpreted in terms of a corresponding set of body coordinate, such as joint angles or muscle lengths (coordinates transformation in Fig. 1). Finally, motor commands, that is muscle torque, must be generated to coordinate the activity of many muscles so that the desired trajectory is realized (generation of motor command in Fig. 1). Several lines of experimental evidence suggest that the the three informations in Fig. 1: desired trajectory in visual coordinates, the desired trajectory in body coordinates and the active torque are internally represented in the brain [13].

However, it must be noted that we do not adhere to the hypothesis of the step-by-step information processing shown by the bottom line of Fig. 1. Rather, our model indicates that there are other information processings which can realize the desired trajectory. In the middle line of Fig. 1, the motor command is obtained directly from the desired trajectory represented in the task-oriented coordinates: that is, the two problems (coordinates transformation and generation of motor command) are simultaneously solved. We [10] proposed that some parts of sensory association cortex (areas 2, 5 and 7) are the locus of this computation by an iterative learning algorithm. That is, the motor command is not determined at once, but in a step-wise, trial and error fashion in the course of a set of repetitions. In this motor learning, short term memory of time history of trajectory and torque are required.

Further, in the uppermost line of Fig. 1, the motor command is calculated directly from the goal of movement: that is, the three problems (trajectory determination, coordinates transformation and generation of motor command) are simultaneously solved.

First, the problem of the determination of the trajectory will be investigated. Second, the problem of the generation of motor command will be examined.

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 of motor control problems are ill-posed in the sense that the solution is not unique.

We listed three ill-posed control problems in Fig. 2. First, consider the trajectory determination problem for a human planar two-joint arm movement when the starting, via and end points and the movement time are specified (Fig. 2 top). There are infinite number of possible trajectories satisfying these conditions. So, 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 middle). Even when the hand trajectory is strictly determined, the time course of the thee joint angles can not uniquely be determined. We note that human hands 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 bottom). Even when the time course of the joint angle is specified, there are infinite number of tension waveforms of the two muscles which realize the same movement trajectory.

To resolve ill-posedness of these problems, we need to introduce some performance index other than the above conditions. We will propose such objective function in the next section. It is worthwhile to evaluate computational schemes or neural network models for sensory-motor control on the standard whether they can cope with the ill-posedness inherent in these problems.

Formation of trajectory: minimum torque-change model

Flash and Hogan [3] provide a mathematical model and experimental data which suggest that the desirable trajectory is first planned using task-oriented (visual) coordinates. They proposed that the trajectory followed by the subject 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.

$$C_J = \int_0^{t_f} \{(\frac{d^3x}{dt^3})^2 + (\frac{d^3y}{dt^3})^2\}dt$$

The minimum jerk model reproduces both the qualitative features and the quantitative details observed experimentally [3]. 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 extracorporal space.

Based on the idea that the objective function must be related to the dynamics, Uno, Kawato and Suzuki [18] proposed the following alternative quadratic measure of performance:

$$C_T = \int_0^{t_f} \sum_{i=1}^{n} (\frac{dT_i}{dt})^2 dt,$$

here T_i is the torque fed to the i-th actuator out of n actuators. The objective function is the sum of the square of the rate of change of torque, integrated over the entire movement. One can easily see that the two objective functions C_J and C_T are closely related. However, it must be emphasized that the objective function C_T critically depends on the dynamics of the musculoskeletal system. Due to this fact, it is much more difficult to determine the unique trajectory which minimizes C_T. Uno et al. [18] overcame this difficulty by developing an iterative scheme, so the unique trajectory and the associated motor command (torque) can be determined simultaneously. That is, the three problems of trajectory formation, coordinates transformation and generation of motor command are solved simultaneously by this algorithm. Mathematically, the iterative learning scheme can be regarded as a Newton-like method in function space.

Trajectories derived from the minimum torque-change model are quite different from those of the minimum jerk model under the following behavioral situations. (i) Big horizontal free movement between two targets. (ii) Constrained and horizontal movement between two targets. (iii) Vertical arm movement between two targets (see experimental data of [2]). (iv) Free and horizontal movement via a point. Uno et al. [18] recently examined human arm trajectories under these situations and found that the minimum torque-change model reproduced these experimental data better.

Since the dynamics of the human arm or the robotic manipulator is nonlinear, the problem to find the unique trajectory which minimizes C_T is a nonlinear optimization problem. The central nervous system does not seem to adopt the iterative algorithm which we proposed in [18]. It was reported that some neural-network models can solve difficult optimization problems such as the traveling salesman problem or early visions by minimizing "energy" through the network dynamics. We [11] proposed a neural-network model, which automatically generates the torque which minimizes C_T without explicit handling of the cost function. This network can be regarded as one example of autonomous motor pattern generators such as a neural oscillator for rhythmic movements.

We recently developed the model to a repetitive network for learning of the vector field of the ordinary differential equation which describes forward dynamics of the controlled object (Fig. 3). The model consists of many identical three layer unit networks which are connected in a cascade with some bypath and electrical connections. The unit network consists of three layers of neurons. The first layer represents the time course of the torque and the trajectory. The third layer represents the change of the trajectory within a unit time, that is, the vector field times the unit time. The output line at the right side represents the time course of the trajectory. Operations of this network are divided into the learning phase and the pattern generating phase. In the learning phase, this network acquires internal model of vector field of forward dynamics of the controlled object between the first and the third layers using synaptic plasticity while monitoring the realized trajectory as a teaching signal. In the pattern generating phase, electrical coupling between neighboring neurons in the first layer is activated. Then the network changes its state autonomously by feedforward and feedback synaptic connections within it. The stable equilibrium state of the network corresponds to minimum energy state and hence the network outputs the torque which realizes the minimum torque-change trajectory. This model has several conceptual similarities with the sequential network conjoined with a forward model network which was proposed by M. Jordan [7]. We emphasize that the proposed repetitive network model can not only resolve the trajectory determination problem but also resolve the inverse kinematics and inverse dynamics problems for redundant manipulators (Fig. 2).

Hierarchical neural network for control and learning

Ito [5] proposed that the cerebrocerebellar communication loop is used as a reference model for the open-loop control of voluntary movement. Allen and Tsukahara [1] proposed a comprehensive model, which accounts for the functional roles of several brain regions in the control of voluntary movement. Tsukahara and Kawato [17] proposed a theoretical model of the cerebro-cerebello-rubral learning system based on recent experimental findings of the synaptic plasticity. Expanding on these previous models and adaptive filter model of the cerebellum [4], we proposed a neural network model for the control of and learning of voluntary movement [9].

In our model, the association cortex sends the desired movement pattern expressed in the body coordinates, to the motor cortex, where the motor command, that is torque to be generated by muscles, is then somehow computed. The actual motor pattern is measured by proprioceptors and sent back to the motor cortex via the transcortical loop. Then, feedback control can be performed utilizing error in the movement trajectory. However, feedback delays and small gains both limit controllable speeds of motions.

The cerebrocerebellum–parvocellular part of the red nucleus system receives synaptic inputs from wide areas of the cerebral cortex and does not receive peripheral sensory input. That is, it monitors both the desired trajectory and the motor command but it does not receive information about the actual movement. Within the cerebrocerebellum–parvocellular red nucleus system, an internal neural model of the inverse-dynamics of the musculoskeletal system is acquired. The inverse-dynamics of the musculoskeletal system is defined as the nonlinear system whose input and output are inverted (trajectory is the input and motor command is the output). Once the inverse-dynamics model is acquired by motor learning, it can compute a good motor command directly from the desired trajectory.

Learning of inverse-dynamics model by feedback motor command as an error signal

The simplest learning approach for acquiring the inverse dynamics model of a controlled object is shown in Fig. 4a. In Fig. 4 the controlled object is called as a manipulator. As shown in Fig. 4a, the manipulator receives the torque input $T(t)$ and outputs the resulting trajectory $\theta(t)$. The inverse dynamics model is set in the opposite input-output direction to that of the manipulator, as shown by the arrow. That is, it receives the trajectory as an input and outputs the torque $T_i(t)$. The error signal $s(t)$ is given as the difference between the real torque and the estimated torque: $s(t) = T(t) - T_i(t)$. This approach to acquire an inverse dynamics model is called direct inverse modeling by M. Jordan [6].

The direct inverse modeling does not seem to be used in the central nervous system because of the following reasons. First, after the inverse-dynamics model is acquired, large scale connection change must be done for its input from the actual trajectory to the desired trajectory, while preserving the minute one-to-one correspondence, so that it can be used in feedforward control. Second, we need other supervising neural network which determines when the connection change should be done. Third, this method which separates the learning and control modes can not cope with dynamics change of a controlled object. Fourth, this learning scheme is not goal directed. Finally, it can not cope with the second and the third ill-posed problems in Fig. 2. M. Jordan explained this reason in the many to one inverse kinematics problem associated with motor control of redundant manipulators with excess degrees of freedom [6,7].

Fig. 4b shows the alternative computational approach which we proposed and called as *feedback error learning*. This block diagram includes the motor cortex (feedback gain K and summation of feedback and feedforward commands), the transcortical loop (negative feedback loop) and the cerebrocerebellum–parvocellular red nucleus system (inverse dynamics model).

The total torque $T(t)$ fed to an actuator of the manipulator is a sum of the feedback torque $T_f(t)$ and the feedforward torque $T_i(t)$, which is calculated by the inverse-dynamics model. The inverse-dynamics model receives the desired trajectory θ_d represented in the body coordinates such as joint angles or muscle lengths, and monitors the feedback torque $T_f(t)$ as the error signal.

The feedback error learning scheme has several advantages over other motor learning schemes including direct inverse modeling. First, the teaching signal or the desired output for the neural network controller is not required. Instead, the feedback torque is used as the error signal. Second, the control and learning are done simultaneously. Third, backpropagation of the error signal through the controlled object or through a forward model of the controlled object [6] is not necessary. Fourth, the learning is goal directed. Finally, it can resolve the ill-posedness in the second and the third problems in Fig. 2 because of good characteristics inherent in the feedback controller.

It is expected that the feedback signal tends to zero as learning proceeds. We call this learning scheme as *feedback error learning* emphasizing the importance of using the feedback torque (motor command) as the error signal of the heterosynaptic learning.

There are two possibilities about how the central nervous system computes nonlinear transformations required for making an inverse dynamics model of a nonlinear controlled object. One is that they are computed by nonlinear information processing within the dendrites of neurons [8,9,16]. The other is that they are realized by neural circuits, and are acquired by motor learning [12].

Examining the first possibility, we [16] have successfully applied the feedback error learning neural network to trajectory control of an industrial robotic manipulator (Kawasaki-Unimate PUMA 260) with prepared nonlinear transformations which were derived from a dynamics equation of a manipulator idealized mechanical model. A simple training movement pattern lasting for 6s was 300 times given. Both the error of trajectory and the feedback torque decreased dramatically during 30min learning. Moreover, the effect of learning for faster and quite different movement pattern from the training pattern was marked, that is the network has great capability of learning generalization.

Regarding the second possibility, we [12] succeeded in learning control of the robotic manipulator by an inverse-dynamics model made of a three-layer neural network (Fig. 5).

In this network, nonlinear transformation was made only of cascade of linear weighted summation and sigmoid nonlinearity. That is, we did not use any a priori knowledge about the dynamical structure of the controlled object. The learning went well and the network has some extent of generalization capability. In the learning, we still used the feedback torque command as the error signal.

Summary

In order to control voluntary movements, the central nervous system must solve the following three computational problems at different levels: (1) determination of a desired trajectory in the visual coordinates, (2) transformation of trajectory from visual coordinates to body coordinates and (3) generation of motor command. Based on physiological information and previous models, computational theories are proposed for the first two problems, and a hierarchical neural network model is introduced to deal with motor command. Combination of the second and the third approach was found to be very efficient for learning trajectory control of an industrial robotic manipulator [14].

References

[1] Allen, G.I. and Tsukahara, N.(1974). *Physiol. Rev.* 54, 957–1006.

[2] Atkeson, C.G. and Hollerbach, J.M.(1985). *J Neurosci.* 5,2318–2330.

[3] Flash, T. and Hogan, N.(1985). *J. Neurosci.* 5, 1688–1703.

[4] Fujita, M.(1982). *Biol. Cybern.* 45, 195–206.

[5] Ito, M.(1970). *Intern. J. Neurol.* 7, 162–176.

[6] Jordan, M.I. and Rosenbaum, D.A.(1988). *COINS Technical Report 88-26*, 1–68.

[7] Jordan, M.I.(1988). *COINS Technical Report 88-27*, 1–41.

[8] Kawato, M., Hamaguchi, T., Murakami, F. and Tsukahara, N.(1984). *Biol. Cybern.* 50, 447–454.

[9] Kawato, M., Furukawa, K. and Suzuki, R.(1987). *Biol. Cybern.* 57, 169–185.

[10] Kawato, M., Isobe, M., Maeda, Y. and Suzuki, R.(1988). *Biol. Cybern.* 59, 161–177.

[11] Kawato, M., Uno, Y., Isobe, M. and Suzuki, R.(1988). *IEEE Control Systems Magazine.* 8, 8–16.

[12] Kawato, M., Setoyama, T. and Suzuki, R.(1988). *Proceedings of the International Neural Networks Society First Annual Meeting.* 342.

[13] Kawato, M.(1988). *Advanced Robotics.* 3, No. 3.

[14] Kawato, M., Isobe, M. and Suzuki, R.(1988). *In Dynamic Interaction in Neural Networks: Models and Data,* ed. Arbib, M.A. and Amari, S., Berlin, Heidelberg, New York: Springer-Verlag.

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

[16] Miyamoto, H., Kawato, M., Setoyama, T. and Suzuki, R. (1988). *Neural Networks.* 1, 251–265.

[17] Tsukahara, N. and Kawato, M.(1982) *In Lecture Notes in Biomathematics,* vol. 45, *Competition and Cooperation in Neural Nets.* ed. Amari, S. and Arbib, M.A., pp.430–441. Berlin, Heidelberg, New York:Springer-Verlag.

[18] Uno, Y., Kawato, M. and Suzuki, R.(1988). *Biol. Cybern.* submitted.

Figure Legends

Fig. 1 Computational theories and internal information representation in the brain for sensory-motor control of voluntary movement. Computational problems are shown in rectangles.

Informations internally represented in the brain are shown in ovals. Possible algorithms are shown in parentheses.

Fig. 2 Three ill-posed problems in sensory-motor control.

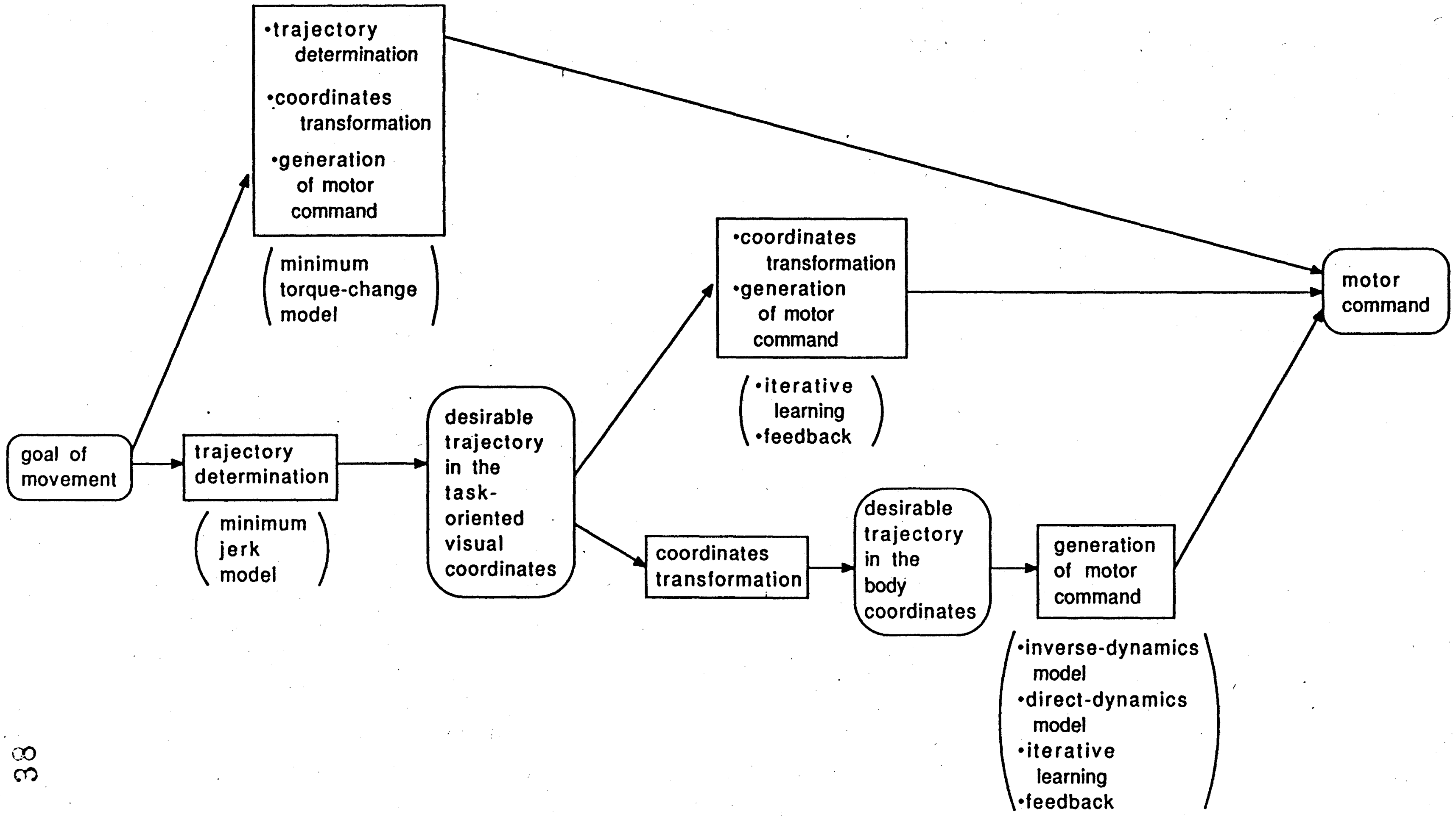
Fig. 3 A repetitive neural network model learns and minimizes energy for generation of torque waveforms which realize minimum torque-change arm trajectory.

Fig. 4 Two schemes for learning inverse dynamics model of a controlled object. a. direct inverse modeling. b. feedback error learning scheme.

Fig. 5 A feedback error learning neural network model. The inverse dynamics model is acquired in the three layer neural network.

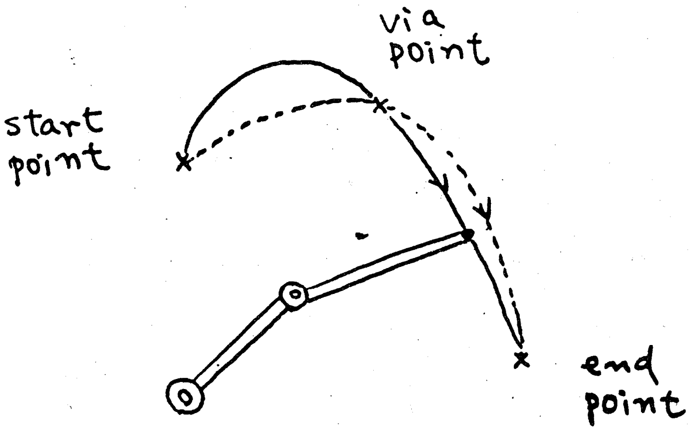
Fig. 1

Computational Theory and Information Representation
(with Algorithms)

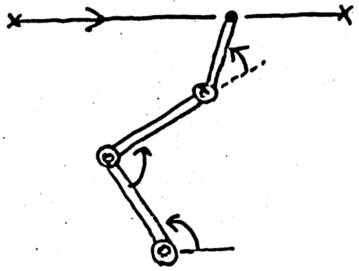

goal of movement
trajectory determination
minimum jerk model
desirable trajectory in the task-oriented visual coordinates
•trajectory determination
•coordinates transformation
•generation of motor command
minimum torque-change model
•coordinates transformation
•generation of motor command
•iterative learning
•feedback
coordinates transformation
desirable trajectory in the body coordinates
generation of motor command
•inverse-dynamics model
•direct-dynamics model
•iterative learning
•feedback
motor command

Trajectory Formation

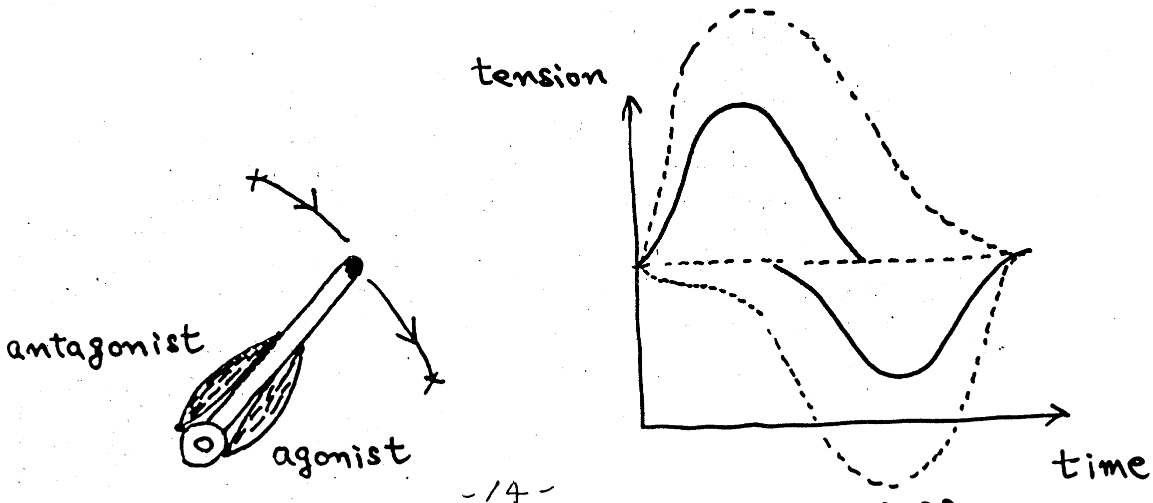
Fig. 2



Inverse Kinematics in Redundant Manipulator



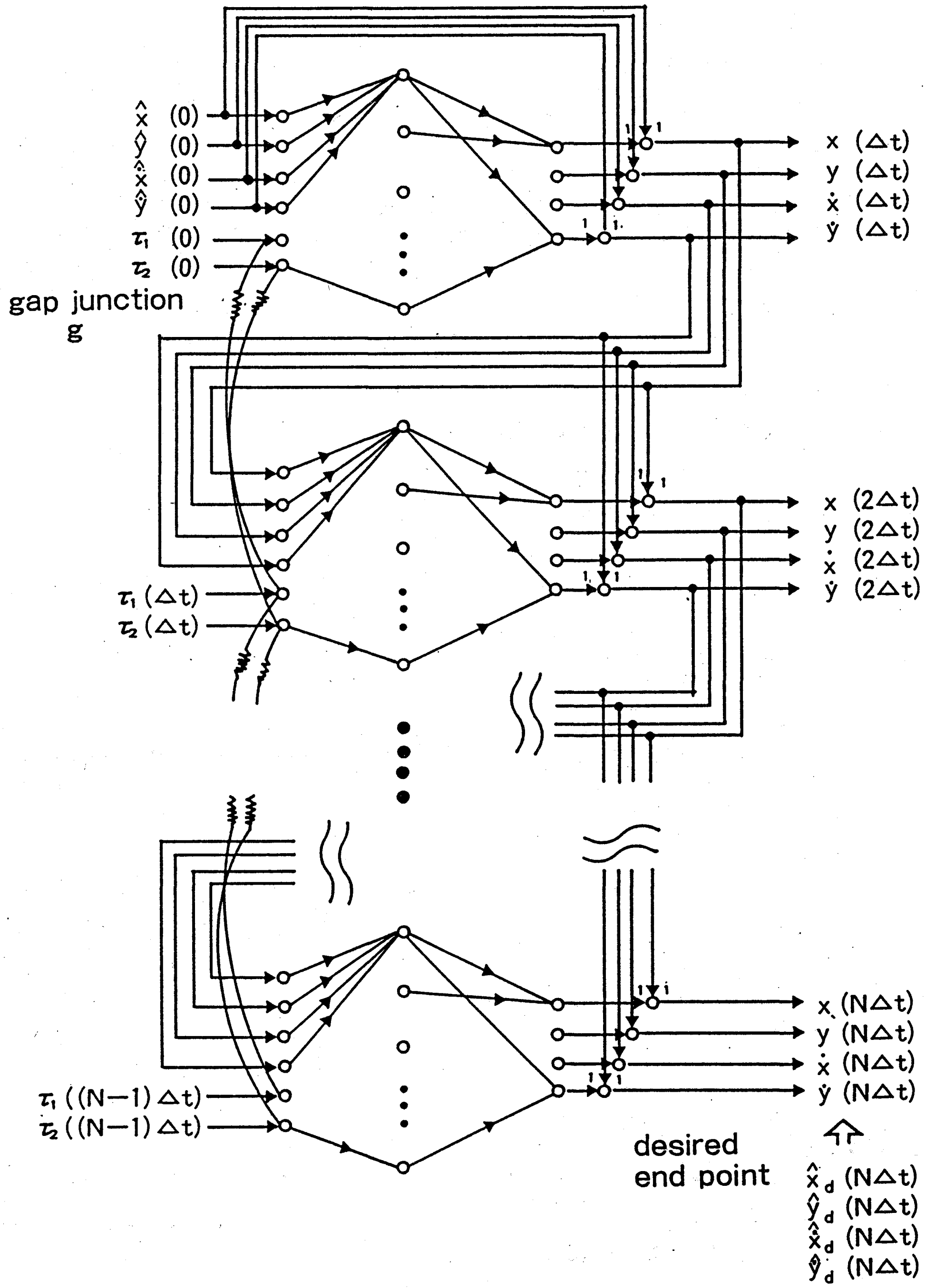
Inverse Dynamics in Redundant Manipulator



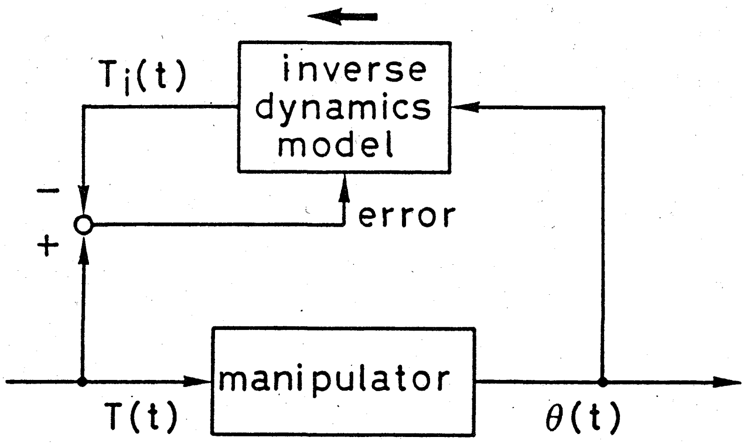
"impedance control"

- Fig. 3

Trajectory Formation
(Energy Minimization)



a: direct inverse modeling



b. feedback error learning

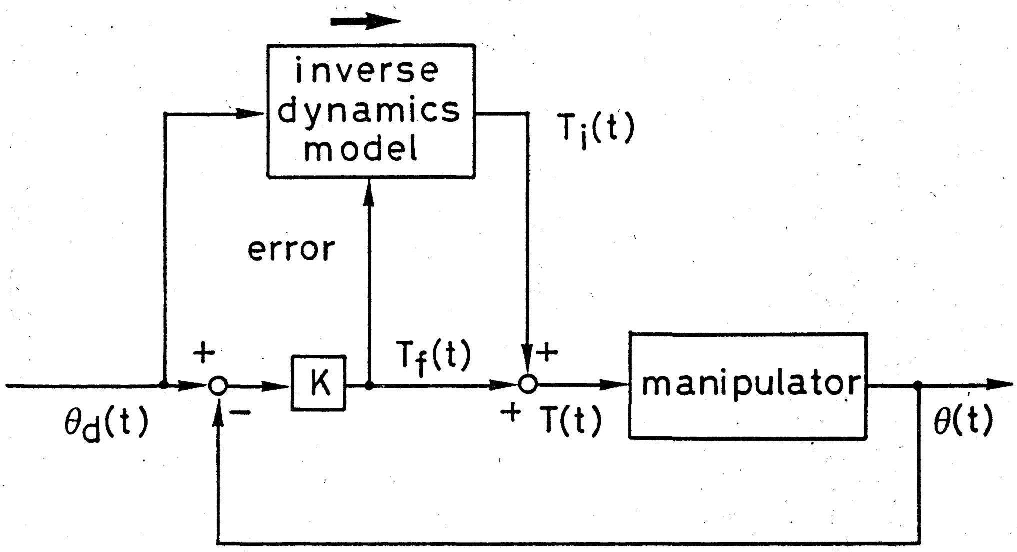
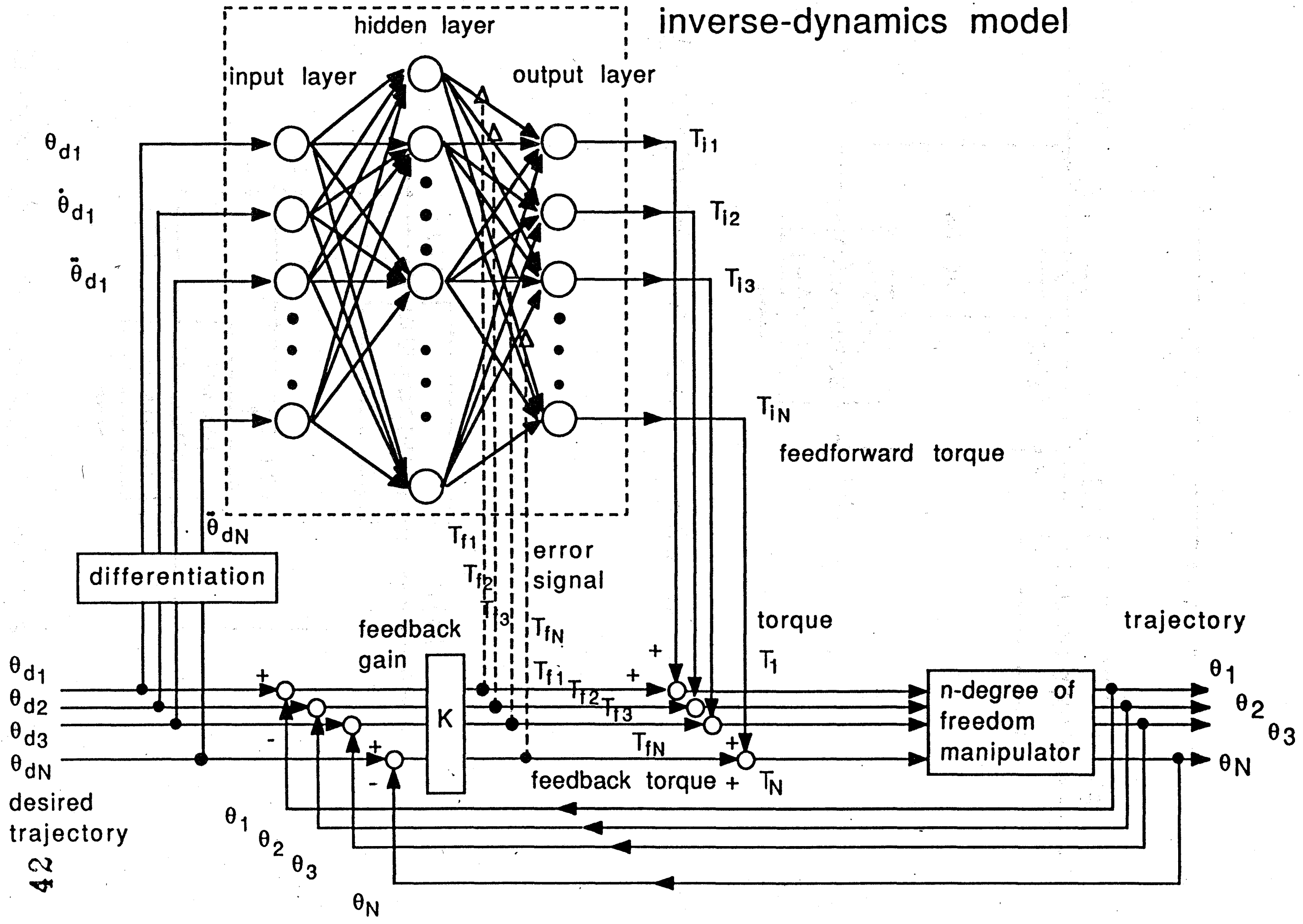


Fig. 4

Fig. 5


inverse-dynamics model
hidden layer
input layer
output layer
θ_{d1}
$\dot{\theta}_{d1}$
$\ddot{\theta}_{d1}$
$\ddot{\theta}_{dN}$
T_{i1}
T_{i2}
T_{i3}
T_{iN}
feedforward torque
differentiation
T_{f1}
T_{f2}
T_{f3}
T_{fN}
error
signal
feedback
gain
K
torque
T_1
trajectory
θ_{d1}
θ_{d2}
θ_{d3}
θ_{dN}
desired
trajectory
42
+
-
T_{f1}
T_{f2}
T_{f3}
T_{fN}
feedback torque +
T_N
n-degree of
freedom
manipulator
θ_1
θ_2
θ_3
θ_N
θ_1
θ_2
θ_3
θ_N