Biol. Cybernetics 29, 29—36 (1978)

Biological Cybernetics

A Model for Sensorimotor Control and Learning*

M. H. Raibert

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA

Abstract. A model for motor learning, generalization, and adaptation is presented. It is shown that the equations of motion of a limb can be expressed in a parametric form that facilitates transformation of desired trajectories into plans. These parametric equations are used in conjunction with a quantized multi-dimensional memory organized by state variables. The memory is supplied with data derived from the analysis of *practice* movements. A small computer and mechanical arm are used to implement the model and study its properties. Results verify the ability to acquire new movements, adapt to mechanical loads, and generalize between similar movements.

Introduction

After two decades of intensive study, control theorists, interested in controlling more complicated non-linear devices (Bryson and Ho, 1969), and physiologists, guided by experimental findings (Hammond, 1956; Melvill Jones and Watt, 1971a), have begun to look beyond the servo feedback mechanism in order to examine the merits of *pre-planning* and the *central program* (Evarts et al., 1970; Melvill Jones and Watt, 1971b). For a limb comprised of interacting degrees of freedom, the transformation from a desired trajectory to such a motor plan, a set of actuator control signals, is a computationally expensive operation. Yet the nervous system's ability to use motor plans interchangebly with a number of effector systems argues that the problem has been efficiently solved (Raibert, 1976). Moreover, the biological solution allows the organism to learn through practice, to generalize training between similar movements, and to adapt to mechanical and sensory changes (Held and Hein, 1963; White, 1970; Miles and Fuller, 1974; Gonshor and Melvill Jones, 1976).

Experiments by Held (1961) and Hein and Held (1963), and a model proposed by Marr (1969) have combined to motivate a new model for motor control, motor learning, and sensorimotor integration. The idea that an internal signal, Helmholtz's *efference copy* (1867), distinguishes an organism's self-produced movements from externally induced movements lead to Held and Hein's now classical experiments. Their results, showing that active movement is essential to motor learning and sensorimotor adaptation, suggest that the nervous system assesses the response characteristics of a limb using an input-output analysis. The problem remains to formulate the extremely complicated equations of motion characterizing a limb's mechanical behavior in a way that permits such an input-output analysis. Marr supplies the clue in his cerebellar model by stating that the *context* in which an elemental movement is made influences the movement's execution. Extensions of this idea show that using state variables as parameters produces dramatic simplifications in the equations of motion, a result which lays the groundwork for the present model (Raibert, 1977b).

In this model two interacting processes plus auxiliary memory functions explain learning of new movements, transfer of training between similar movements, and adaptation to mechanical and sensory changes. *Parameterization*, the process of restating an equation with a subset of the independent variables held constant, recasts the equations of motion into a very simple form that allows learning based on practice. The parameterized equations, however, must be used in conjunction with a multi-dimensional memory in which *constants of mechanical description* are stored. *Learning*, the process which supplies data to this tabular memory, takes place when torque vectors

* This research was done while the author was a graduate student at the Massachusetts Institute of Technology in the Artificial Intelligence Laboratory and Department of Psychology. It was supported in part by training grant NGMS 5-T01-GM01064-15

0340-1200/78/0029/0029/$01.60

applied to the limb, T_m, are correlated with resulting acceleration vectors, $\ddot{\theta}$. Properties of the memory, its time-constant and accessing function, contribute to adaptation and generalization.

The proposed mechanism compensates for the kinematic and dynamic nature of a limb so that higher level motor processes can be free from such responsibilities. Since the movements specified by higher level processes need not be tailored to a particular limb, mechanically dissimilar limbs may be used interchangeably in the production of movement.

Model Description

When each of the terms contributing to the torque acting on the joints of a limb are included, Newton's equation for rotary motion may be expressed schematically as:

$$T_m - G(\theta) - B(\dot{\theta}) - C(\theta, \dot{\theta}) = J(\theta)\ddot{\theta}, \quad (1)$$

where T_m is the actuator torque vector; G is a vector-function for gravitational torque; B is a vector-function for frictional torque; C is a vector-function for Coriolis torque; J is a matrix-function for moment of inertia; θ, $\dot{\theta}$, and $\ddot{\theta}$ are the position, velocity, and acceleration vectors.

The full set of time-varying, non-linear equations with explicit expression of θ- and $\dot{\theta}$-dependence has been worked out by Kahn (1969). His equations involve about 1600 terms and 13000 multiplications for a general 3 degree of freedom limb.

The Translation Equations

By treating the state variables θ and $\dot{\theta}$ as parameters (i.e. letting them assume a number of fixed values), a simplified parametric form of Kahn's equations can be found:

$$T_m - G|_{(\theta=\alpha)} - B|_{(\dot{\theta}=\beta)} - C|_{(\theta=\alpha, \dot{\theta}=\beta)} = J|_{(\theta=\alpha)}\ddot{\theta}, \quad (2a)$$

where α is a parametric position vector; β is a parametric velocity vector.

Or, more compactly:

$$T_m - G_\alpha - B_\beta - C_{\alpha\beta} = J_\alpha \ddot{\theta}. \quad (2b)$$

Here each of the vector-function relationships $G(\theta)$, $B(\dot{\theta})$, $C(\theta, \dot{\theta})$, and $J(\theta)$ has become a parameterized set of constants. By grouping terms and making the equation explicit in muscle torque one further simplification can be made:

$$T_m = J_\alpha \cdot \ddot{\theta} + K_{\alpha\beta}, \quad (3)$$

where $K_{\alpha\beta} = G_\alpha + B_\beta + C_{\alpha\beta}$.

Equation (3) is the *translation equation*. It is linear in θ and without hidden dependencies on θ or $\dot{\theta}$.

We now define a state space having $2N$ dimensions, and associate one state variable $\{\theta_1, \theta_2 \ldots \theta_N, \dot{\theta}_1, \dot{\theta}_2 \ldots \dot{\theta}_N\}$ with each dimension. For any point in this space, $(\theta, \dot{\theta})$, there exists a set of values for J_α and $K_{\alpha\beta}$, such that (3) describes behavior of a limb passing through that point. Furthermore, since the values of the components of J and K vary smoothly throughout the space, i.e.:

$$\nabla j_{ij,\alpha} < \infty \quad (i=1,2 \ldots N; j=1,2 \ldots N)$$

$$\nabla k_{i,\alpha\beta} < \infty \quad (i=1,2 \ldots N)$$

the space can be divided into a large number of hyper-regions throughout each of which the behavior of the limb, and the corresponding values of J and K, are reasonably uniform. This approach becomes useful when values of J and K corresponding to particular $(\theta, \dot{\theta})$ are available from a tabular memory that is also organized by state variables: Desired trajectories, $\ddot{\theta}_D(t)$, are processed by (3) after division into intervals of duration Δt, where values for J and K for each interval are obtained from the *state space memory*.

Since data will not always be available for every hyper-region (assuming the system begins tabula rasa), performance will be more robust if a memory accessing function is used that takes into account the gradual variations of mechanical behavior through state space—if data from a particular hyper-region are not available, data from neighboring regions may be used instead. In addition to robustness, transfer of training between similar practiced and unpracticed movements is an expected consequence of such an accessing function.

The Inversion Equations

Von Holst and Mittelstaedt used the efference copy, an internal copy of the motor command, to account for the fly's ability to distinguish between internally and externally produced changes in sensory stimulation (von Holst, 1954; Mittelstaedt, 1958). Their notion was that the relationship between an externally generated signal describing changes in sensory stimulation and an internally generated signal describing impending changes in the position of the sensory surface would always give unambiguous information about movement in the external world. In Held's model, (1961; Hein and Held, 1962) the Holstian view was augmented to allow attainment of perceptual accuracy even after changes were made to the meaning of the sensory signals. The efference copy was used to elicit the trace of previous reafference, which in turn was compared to the current afference. In 1965 Young and Stark model-

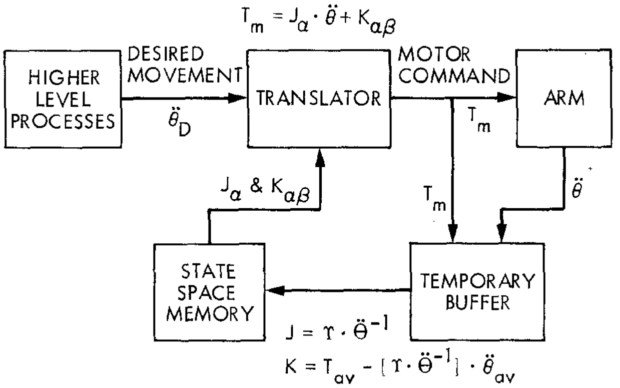

Fig. 1. Major components of the model. The translator converts descriptions of desired trajectories into motor commands suited to the kinematic and dynamic properties of a particular limb. The operation employs the tabular equations of motion in conjunction with the state space memory. Each movement of the limb generates data which, when processed by the inversion equations, contribute to the state space memory, and consequently, to future translations

led the ability of humans performing a tracking task to change control strategies when there were changes in the dynamics of the controlled element (Young and Stark, 1965). In that model the efference copy was used to drive an *internal dynamic model* of the controlled element, the output of which was compared with afference from the control task.

Though designed to explain perception, MacKay's idea of *evaluation* rather than *elimination* (MacKay, 1972) is similar to the mechanism described here. In the present model the relationship between efference copy, T_m and reafference, $\ddot{\theta}$, is used to compute descriptions of the mechanical properties of the limb, represented in (3) by J and K. Reafference is obtained when the limb's sensors, (or other sensors which can monitor the limb's activity), are used to measure the acceleration vector during a movement. Efference copy is a record of the actuator torque vector, internally available from the source of motor commands or possibly from actuator sensors. The use made of the efference copy in this model is somewhat unique in that *there is no comparator, no error signal is calculated,* and *no error correction procedure is used.* Rather, the limb's properties are found by examining the relationship between input and output, command and response. As a result the local minima problems associated with search procedures are avoided (Tsypkin, 1971).

Since the simplified equations of motion are linear, values of J and K can be found in a straightforward manner, provided that sets of measurements $\{(T_{m,1}, \ddot{\theta}_1), (T_{m,2}, \ddot{\theta}_2)\ldots\}$ are available:

$$J = \tau \cdot \ddot{\Theta}^{-1}, \tag{4a}$$

$$K = T_{av} - [\tau \cdot \ddot{\Theta}^{-1}] \cdot \ddot{\theta}_{av}, \tag{4b}$$

where

$$\tau = [T_1 \vdots T_2 \vdots \ldots \vdots T_N] - [T_{N+1} \vdots T_{N+1} \vdots \ldots \vdots T_{N+1}];$$

$$\ddot{\Theta} = [\ddot{\theta}_1 \vdots \ddot{\theta}_2 \vdots \ldots \vdots \ddot{\theta}_N] - [\ddot{\theta}_{N+1} \vdots \ddot{\theta}_{N+1} \vdots \ldots \vdots \ddot{\theta}_{N+1}];$$

T_i and $\ddot{\theta}_i$ are the i'th measurements of T and $\ddot{\theta}$; X_{av} denotes the average: $(X_1 + X_2 + \ldots + X_{N+1})/(N+1)$. (Note: all torques are motor torques—the m subscript has been dropped.)

Equation (4) is the *inversion equation.* These calculations can be performed if $N+1$ input-output pairs, $(T_{m,i}, \ddot{\theta}_i)$, also called measurement vectors, are available. All measurements contributing to such a calculation must have been made while the limb was near a single hyper-region of interest. A temporary buffer is postulated to store such measurements until appropriate sets are available for inversion. The resulting values of J and K are then stored in the state space memory in combination with previously stored data. The dynamic updating of the state space memory, adding data as they are available and combining them with old data, allows the system to adapt to changes in the limb's kinematic and dynamic properties, in addition to improving immunity to the effects of inverting noisy measurements, typically a problem when inverting physical data.

The block diagram shown in Figure 1 summarizes the model's operation: High level processes produce descriptions of desired movements, $\ddot{\theta}_D(t)$, which are presented to the translator. The desired movement is sectioned into time intervals, each of duration Δt. For each time slice (3) the translation equations, used in conjunction with the constants of mechanical description, J and K values from the state space memory, generate a motor plan that will replicate the desired trajectory. The calculated force commands are issued to the limb and, during the movement, a copy of the command, T_m, and a copy of the sensory signals that indicate progress of the movement, $\ddot{\theta}$, are stored in a temporary buffer. Subsequently, the contents of the buffer and the inversion equations, (4), are used to calculate values of J and K, which are stored in the state space memory in combination with data that might have been stored there previously.

Properties of the Model

Initial performance will be quite poor since every attempt to use information about the mechanical character of the limb will be frustrated—the state space memory will be tabula rasa—empty. As movements of the limb are made, data describing the mechanics of its operation become available. During this period of data acquisition the quality of translations will gradually improve. Practice will facilitate mastery of a practiced

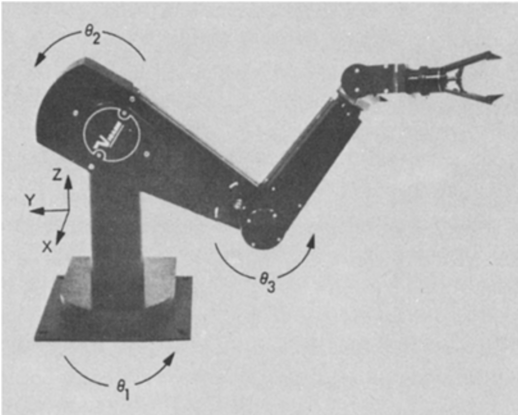

Fig. 2. Layout of the first three joints of the MIT-Vicarm manipulator. The manipulator is about the size of a human arm; base-to-shoulder = 0.273 m, shoulder-to-elbow = elbow-to-wrist = 0.203 m. Each joint is provided with a DC torque motor, a potentiometer, a tachometer, and a clutch-type brake

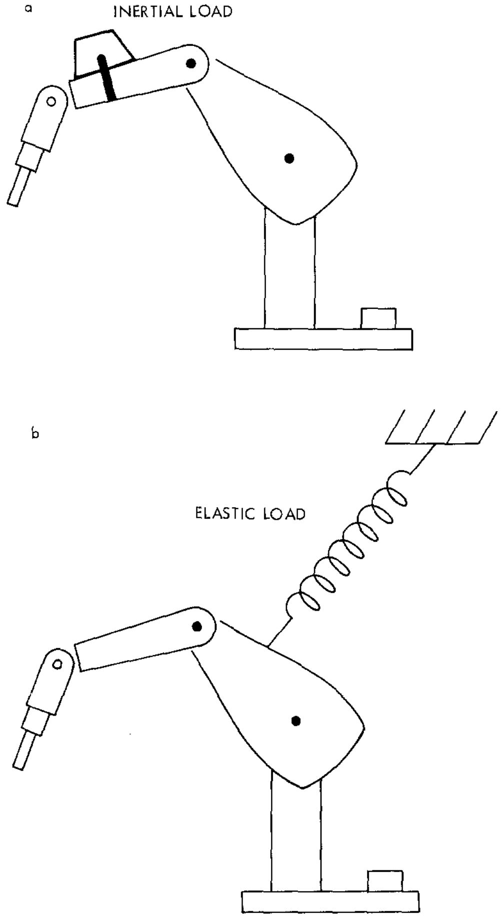

Fig. 3a and b. Two methods of applying loads in order to disturb the manipulator's behavior are shown. **a** A 0.19 kg weight is attached to the third link of the manipulator. **b** A 1.85 kg/m spring is attached from the second link to "ground". When movements start the spring is stretched 0.83 m and runs from coordinates (0.17 m, 0.0 m, 0.25 m) to (0.02 m, 0.70 m, 1.20 m); see Figure 2

movement, while similar (but not identical), movements will be improved more slowly. If the mechanical properties of the limb or sensors should change then the model adapts, since new constants of mechanical description are continuously being computed and stored.

The State Space Model can control limbs having a wide variety of dynamic and kinematic properties. A single translator can learn to control almost any limb or body part. This is a direct result of the tabular nature of the equations which describe the mechanical system. Though the development given above deals with torques applied to the joint, the actuator terms given in (3) and (4) can be force applied to a tendon. In fact, actuators and sensors need not be affiliated with any one joint or subset of joints. Reafference can take the form of visual feedback just as readily as joint oriented proprioceptive feedback, provided the choice is made before learning commences and desired trajectories are described in the chosen coordinate system.

In order to evaluate and verify the power of the model, a set of computer programs embodying the various elements are used to control a mechanical arm. Tests of this implementation reveal the model's weakness and illustrate its strengths.

Methods

A PDP-11/45 computer is used to perform all computations, to issue commands to the manipulator, and to make measurements. The three joints of the MIT-Vicarm manipulator that allow the wrist to be positioned arbitrarily within the arm's work space, are used ($N=3$) (Horn and Inoue, 1974). See Figure 2. Each joint is powered by a DC torque motor and provided with a potentiometer and tachometer. When a movement is made the computer specifies the torque delivered by each motor and measures angular position and velocity every 10 ms. In addition, velocity samples taken every 500 µs allow the limb's average accelerations to be estimated over 60 ms intervals using least-mean-square error techniques ($\Delta t = 60$ ms).

State Space Memory

Though only $N+1$ measurement vectors are theoretically required for each inversion (here $N+1=4$), improved noise immunity is obtained by using the generalized inverse (Rust et al., 1966; Albert, 1972) to invert sets of 8 vectors. The resulting data are stored in a hash coded disk memory in weighted combination with data previously stored for the same hyper-region. Each new entry receives a weight of $1/\tau$, and previous data a weight of $(\tau-1)/\tau$, where τ is the memory's time constant.

The memory is 6 dimensional, (one dimension for each state variable), and quantized. Each dimension is partitioned into 10 ranges producing 10^6 possible hyper-regions. A single hyper-region measures $(15\,\mathrm{deg})^3$ by $\left(13\,\frac{\mathrm{deg}}{\mathrm{s}}\right)^3$. These regions are quite small and the mechanical properties of the arm are fairly constant throughout. Each access of the memory yields a weighted average of data from the desired hyper-region and all closest neighbors. Data from the

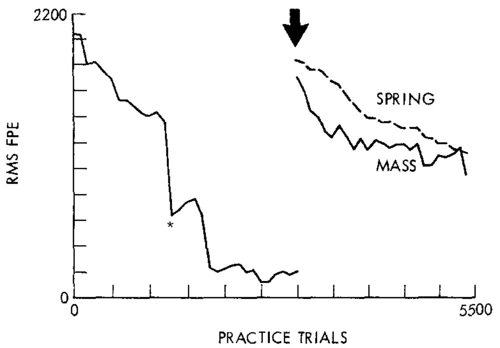

Fig. 4. Left: Acquisition of prototype PR-12 is shown as 3000 practice trials are executed and analyzed. Arrow: One of the two loads shown in Figure 3 are applied. Right: The time course of adaptation to the two types of load is recorded ($\tau = 10$)

desired hyper-region are given a weight equal to the number of times data were stored in that region. Neighbors are given a weight of 1 if any data are present, otherwise zero

Practice and Test

Input-output data are generated by exercising the arm under control of a practice program. This program generates a sequence of approximations to a pre-specified desired movement, called a *prototype.* Periods of practice, analysis of practice data (4), and execution of test movements, generated by (3) to test performance, are alternated during a learning session. Each test movement is evaluated by finding the root-mean-square position-error (RMS PE) or final-position-error (RMS FPE) for the three joints.

Adaptation is measured by manipulating the mechanical state of the arm during a training session. These manipulations are accomplished by applying inertial and elastic loads to the arm so that the static and dynamic properties are affected (see Fig. 3). Generalization of training is measured by testing performance of a set of prototypes, after practice of only one. The members of this set vary systematically in similarity to the practiced prototype. A learning index, LI, facilitates presentation of the generalization data:

$$\mathrm{LI} = \frac{\Sigma(e_0 - e_i)}{\Sigma e_0},$$

where e_i is the RMS FPE for the i'th test movement; e_0 is the pre-training performance value; Σ is the sum from $i=1$ to $n-1$; n is the number of test movements.

Further details of the implementation are available in Raibert (1977a).

Results

The left half of Figure 4 is a learning curve for 3000 practice trials. As predicted, performance improves as more practice data are generated and analyzed. Rapid *jumps* in performance arise when new hyper-regions of the memory are first provided with data, (asterisk in Fig. 4). When a load is applied to the arm (arrow in Fig. 4), adaptation slowly takes place during the next 2500 practice trials under the new mechanical regime, as shown on the right side of Figure 4. Modification of τ, the memory's time constant, results in improved rates of adaptation, though very small values of τ also introduce some instability (see Fig. 5).

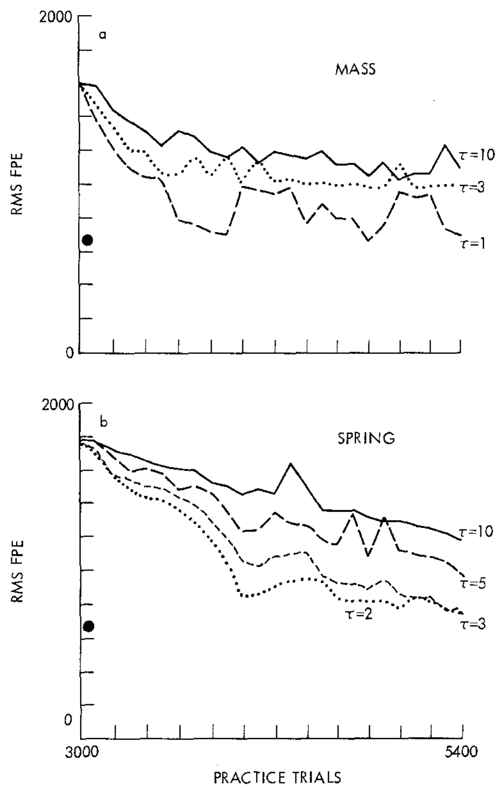

Fig. 5a and b. The memory's time-constant is systematically varied. Smaller values of τ yield more rapid, but noisier adaptations. **a** inertial load; **b** spring load (prototype PR-11). Closed circles indicate pre-adaptation levels

Verification of the model's ability to generalize data derived from the practice of one movement to other similar movements is illustrated in Figures 6 and 7. Throughout a 2400 trial learning session performance of the practiced prototype, PR-20, improves the most (Fig. 6a). Each of the other prototypes exhibit various degrees of improvement depending on their similarity to PR-20 (see caption to Fig. 6). These generalization data are summarized quantitatively in Figure 7 (diamonds), where the learning index, LI, is plotted for each prototype. To control for the possibility of gradients due to the particular choice of prototypes, a different member of the prototype set, PR-23, was practiced. The results, shown in Figures 6b and 7 (triangles), reveal a similar pattern: the practice prototype shows the most improvement, with other movements improving according to their similarity to the practice prototype.

Figures 4 through 7 verify the model's basic attributes:

1) Motor commands are generated suited to the kinematic and dynamic properties of the effector mechanism.

2) The quality of the motor commands improves with practice, though no error correction is used.

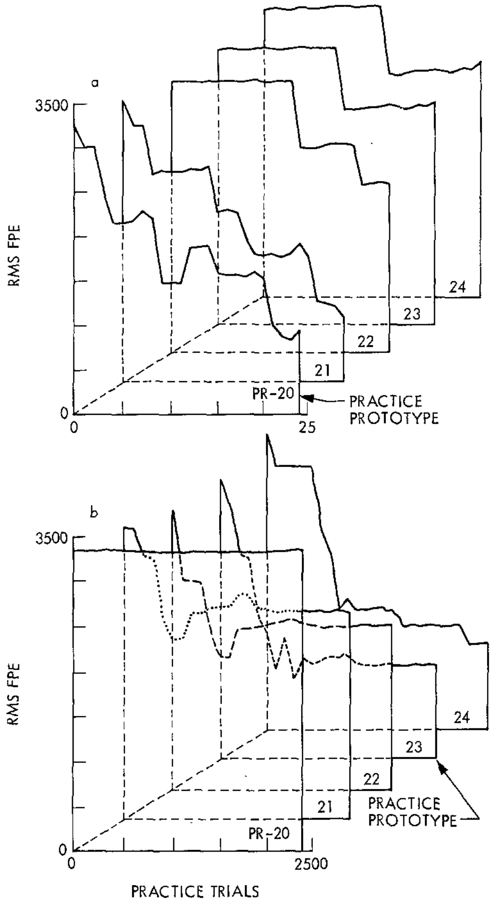

Fig. 6a and b. Five prototypes were used to generate these learning curves so that generalization could be shown. The prototypes share a common ending position and duration, but vary systematically in starting position; [0.285 m, −0.145 m, 0.12 m), (0.265, −0.145, 0.1), (0.245, −0.145, 0.3), (0.245, −0.165, 0.6), (0.245, −0.185, 0.4), respectively]. **a** Prototype PR-20 was practiced and prototypes PR-20, PR-21, PR-22, PR-23, and PR-24 were tested. **b** Prototype PR-23 was practiced and the entire set was tested

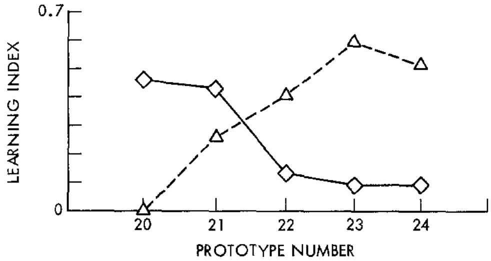

Fig. 7. Generalization curves summarizing the data of Figure 5 are shown. (Diamonds) Prototype PR-20 was practiced. (Triangles) Prototype PR-23 was practiced

3) Practice of one movement improves performance of others, provided they are similar to the practice prototype.

4) Control of the arm is maintained or reattained, despite changes in its mechanical properties.

Discussion

Examination of (4) reveals that $\ddot{\theta}_D$ is absent from the computation of J and K, the constants of mechanical description. Without knowledge of the desired response an error signal cannot be computed. The present system is able to learn without error information and is, therefore, somewhat unique among models for control. Systems that do use error correction rely on the signed magnitude and sometimes the derivatives of error in selecting the next, and hopefully, *better* command (Fu, 1971; Tsypkin, 1971). Unfortunately, local error data are not always useful in finding global maxima that correspond to *best* commands, and *hill-climbing* problems may result. The parametric equations are so simple, however, that a search procedure is not required for solution. Application of (4) only requires that $N+1$ independent measurements be available for the same hyper-region.

The power and simplicity of the model derive from the combined use of parameterization *and* learning. Without learning, the constants that make the parameterized equations usable can only be found by evaluating extremely complicated differential equations. Learning without parameterization, on the other hand, requires inversion of non-linear trigonometric differential equations comprised of thousands of terms. Parameterization makes learning possible, and learning makes parameterization usable.

A mechanism has been described that precomputes a set of motor commands that are executed in the absence of feedback. Few practical applications (biological applications included), can tolerate the imprecision of such open-loop operation, yet the problems of motor planning can probably be best developed in this type of isolation. Ultimately it will be necessary to find a compromise between pre-planning and servo control, and the compromise will yield dividends: The same data that are so useful in planning will facilitate on-line error correction, both processes benefitting from experience. For the sake of clarity of results and presentation, however, consideration of *plan+servo* models has been postponed for future study.

Young and Stark (1965) and others have proposed the use of an *internal dynamic model* to allow learning and adaptation. Their idea is that information describing the response of the plant to commands can be used to adjust an internal dynamic model that will be used in future selection of commands. Although this idea can be made to work, another concept which represents a different point of view is stressed here—the *internal inverse dynamic model* (Paul, 1972; Waters, 1974). The idea of the inverse is that a motor learning system should have a transfer function which converts responses into commands—the inverse of the operation performed by the mechanical device. When the inverse *and* the device are operated in cascade the transfer function is the identity matrix—the desired result. The internal dynamic model allows simulation of the inverse function with an approach similar to *analysis by synthesis* (Eden, 1962). Because it uses sets of extremely simple equations to describe the plant's behavior, however, the present model calculates the required inverse functions directly, for each region of space. We remind the reader that this is an argument of "point of view" rather than "computational approach". Iterative techniques for computing an inverse, Young and Stark's approach, are quite common and legitimate.

Summary

A model is proposed that translates descriptions of desired trajectories into motor plans. The processes which provide input to this translator need not deal with the mechanical properties of the manipulator, and the specified trajectories may be expressed in a coordinate system appropriate to the available sensors. The translator's outputs are motor commands suited to the kinematic and dynamic properties of a particular manipulator and its actuators.

The model employs parameterized equations of motion in conjunction with a multi-dimensional memory organized by state variables. The memory is supplied with data derived from the analysis of practice movements. The analysis performed is quite simple and does not employ error correction or search techniques, as do many learning schemes currently in use. Since iterative methods are avoided, problems involving local minima are not encountered.

A small computer and three joints of the MIT-Vicarm manipulator were used to implement the model and assess its properties. Tests have verified the ability to:

1) Acquire usable mechanical descriptions of the manipulator, and to use those descriptions to pre-plan effective trajectories.

2) Adapt to mechanical disturbances caused by inertial and elastic loads, and acquire control after the gravity vector is modified.

3) Generalize information derived from the practice of one movement to the execution of other similar movements.

4) Use a Cartesian coordinate system for specification of desired trajectories when measurement data are provided in that system, even though motor commands are expressed in joint coordinates.

Acknowledgements. I would like to thank Fred Drenckhahn for maintaining the manipulator, and John Hollerbach, Berthold Horn, Joe Malpeli, Matt Mason, and Whitman Richards for their criticisms and contributions.

References

Albert, A.: Regression and the Moore-Penrose pseudoinverse. New York: Academic Press 1972

Bryson, A.E., Ho, Y.C.: Applied optimal control. Waltham: Ginn 1969

Eden, M.: Handwriting and pattern recognition. IRE trans. Inf. Theory, IT-8, 160—166 (1962)

Evarts, E.V., Bizzi, E., Burke, R.E., Delong, M., Thach, W.T., Jr.: Central control of movement. Neurosc. Res. Progr. Bull. **9** (1970)

Fu, K.S.: Learning control systems and intelligent control systems: an intersection of artificial intelligence and automatic control. IEEE trans. Auto. Contr., 70—72 (1971)

Gonshor, A., Melvill Jones, G.: Extreme vestibulo-ocular adaptation induced by prolonged optical reversal of vision. J. Physiol. **256**, 361—379 (1976)

Hammond, P.H.: The influence of prior instruction to the subject on an apparently involuntary neuro-muscular response. J. Physiol. **127**, 17—18 (1956)

Hein, A., Held, R.: A neural model for labile sesorimotor coordination. In: Biological prototypes and synthetic systems, Vol. 1, pp. 71—74. Bernard, E., Kare, M., eds. New York: Plenum Press 1962

Held, R.: Exposure-history as a factor in maintaining stability of perception and coordination. J. Nerv. Ment. Dis. **132**, 26—32 (1961)

Held, R., Hein, A.: Movement-produced stimulation in the development of visually guided behavior. J. Comp. Physiol. Psych. **56**, 872—876 (1963)

Helmholtz, H.V.: Handbook of physiological optics, Vol. 3. Leipzig: Voss 1867

Holst, E.V.: Relations between the central nervous system and the peripheral organs. Brit. J. Anim. Behav. **2**, 89—94 (1954)

Horn, B.K.P., Inoue, H.: Kinematics of the MIT-AI-Vicarm manipulator. MIT Working Paper 69, May 1974

Kahn, M.E.: The near-minimum time control of open-loop articulated kinematic chains. Stanford Artificial Intelligence Memo No. 106, December, 1969

Marr, D.: A theory of cerebellar cortex. J. Physiol. **202**, 437—470 (1969)

Melvill Jones, G., Watt, D.O.D.: Observations on the control of stepping and hopping movements in man. J. Physiol. **219**, 709—727 (1971a)

Melvill Jones, G., Watt, D.O.D.: Muscular control of landing from unexpected falls in man. J. Physiol. **219**, 729—737 (1971b)

Miles, F., Fuller, J.H.: Adaptive plasticity in the vestibulo-ocular responses of the rhesus monkey. Brain Res. **80**, 512—516 (1974)

Mittelstaedt, H.: The analysis of behavior in terms of control systems. In: Group processes, transactions of the fifth conference, pp. 45—84. Schaffner, B., ed. New Jersey: Princeton 1958

Paul, R.: Modelling trajectory calculation and servoing of a computer controlled arm. Stanford Artificial Intelligence Memo No. 177, November, 1972

Raibert, M. H.: A state space model for sensorimotor control and learning. MIT Artificial Intelligence Memo No. 351, January, 1976

Raibert, M. H.: Control and learning by the State Space Model: experimental findings. MIT Artificial Intelligence Memo No. 412, March, 1977

Raibert, M. H.: Analytical equations vs. table look-up for manipulation: a unifying concept. To be presented at IEEE Conference on Decision and Control, 1977

Rust, B., Burrus, W. R., Schneeberger, C.: A simple algorithm for computing the generalized inverse of a matrix. Commun. ACM **9**, 381—387 (1966)

Tsypkin, Y. Z.: Adaptation and learning in automatic systems. (Translated by Nikoloic, Z. J.) New York: Academic Press 1971

Waters, R. C.: A mechanical arm control system. MIT Artificial Intelligence Memo No. 301, January, 1974

White, B. L.: Experience and the development of motor mechanisms in infancy. In: Mechanisms of motor skill development, pp. 95—132. Conolly, K., ed. London: Academic Press 1970

Young, L. R., Stark, L.: Biological control systems—a critical review and evaluation: developments in manual control. NASA, Cr-190, 1965

Received: November 29, 1977

Dr. Marc H. Raibert
Jet Propulsion Laboratory
Calif. Inst. of Technology
4800 Oak Grove Drive
Pasadena, CA 91103, USA