MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

AIM 351

January 1976

A STATE SPACE MODEL FOR SENSORIMOTOR CONTROL AND LEARNING

by

Marc Raibert

ABSTRACT This is the first of a two-part presentation which deals with certain computer controlled manipulator problems. This first part discusses a model which is designed to address problems of motor control, motor learning, adaptation, and sensorimotor integration. In this section the problems are outlined and a solution is geven which makes use of a state space memory and a piece-wise linearization of the equations of motion. A forthcoming companion article will present the results of tests performed on an implementation of the model.

This report describes research done in the Department of Psychology and the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for this research is provided in part by grant NIH-5-T01-6MO-1064-14 supervised by H. L. Teuber, and by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75-C-0643.

Summary

A model is presented which deals with certain problems of motor control, motor learning, and sensorimotor integration. The use of efference copy, re-afference, and a state space memory are key factors in its operation. The following are functions and features of the proposed system:

1) Descriptions of desired movements are translated into motor commands which will produce the specified motions. The initial specification of the movement is free of information regarding the mechanics of the effector system.

2) The system demonstrates gradual improvement in performance as it gains experience from self-produced movements.

3) The performance of selected movements can be more rapidly improved through concentrated practice.

4) Practice of one movement may improve performance of another thereby showing a kind of transfer of training.

5) Adaptations to mechanical and certain sensory changes take place without an explicit error correction procedure.

6) Mechanical interactions between joints are automatically compensated in production of the motor command.

7) No constraints need be placed on the geometry of the limb or body-part under control.

8) The sensory information used by the model may be related to the joints of the limb, to visual space, or to any other coordinate system. It need satisfy no linearity constraints, but must be capable of uniquely describing each limb movement.

9) The computations performed by the model are quite simple, and especially suited to a parallel processing device.

In addition to a discussion of these properties in terms of the model's operation, an implementation using a computer and manipulator is introduced.

Introduction

The human motor system is characterized by properties which are not exhibited by traditional man-made machines. Most basic of these properties is the ability to learn. Initially the human infant exhibits discoordinated movements which have no apparent purpose and are skillessly executed. But as the child develops, his movements take on a different character. They become directed and effective, smooth and graceful. The improved dexterity is attributable in part to the experience the developing organism receives from his own attempts to move (2,3,10,12). The adult, moreover, is able to select particular movements and center his attention upon them through practice until a high level of performance has been reached. The human is not limited to making only those movements which have been the subject of previous practice -- it is often the case that a movement which has never before been attempted can be executed with a fair degree of precision.

Not only are we able to gain motor control of our bodies as we grow up but we are able to maintain control. In the normal situation this means that we make the adjustments needed to control our limbs even though the masses and sizes of the body parts undergo large changes throughout ontogeny. In the laboratory we are able to compensate for experimentally induced distortions made to our sensory inputs or to the environment (10,11,27).

Other properties of the human motor system are affected by the mechanical nature of the skeletal and muscular systems, and the laws of physics which they must obey (21). The forces and torques created by a muscle often influence a number of joints, even when the muscle is of the simple, single joint variety. Each joint is influenced by a number of muscles, not only because there are many muscles 'across' the joint, but because reaction torques are produced when muscles accelerate other joints of the body. Yet our nervous control system effectively compensates for these mechanical interactions when precise movements are called for.

Finally, our limbs are useful tools only if they will do our bidding, but our wishes are phrased in a language which motoneurons and muscle do not understand. If we start with the simple, perhaps schematic instruction, "Close your eyes and move your hand so that the tip of your finger travels a path which is a straight line." we are able to comply. We are able to comply even though this specification of the movement of the finger gives no explicit information about the requisite joint movements or muscle forces. This means that our motor system is able to convert a description of a movement given in one coordinate frame into a set of commands which are suited to act in an entirely different frame -- that of bone, joint, and muscle (1,8,18).

The purpose of this paper is to present a model of a motor sub-system which displays each of the properties mentioned above. The model presented here is not the only model which displays these properties of the biological motor system, nor is it supposed that other models cannot be proposed which do so (6,7,23,26,27). The present model does not presume to account for all aspects of biological motor control nor even completely account for those properties cited above. Nor can any strong claim be made regarding the basis for the similarities of behavior displayed by the mechanism under consideration here and the biological motor system. So if this model cannot be shown to be a necessary model, nor are its powers sufficient to describe the human motor system, nor can an effective demonstration of mutual causality with the biological motor system be made, why develop it and why use a computer and manipulator to study it?

The motor system is complicated, being comprised of numerous nervous and mechanical components, and perhaps numerous logical or functional components. When the psychologist or physiologist studies this system he has two choices. He can study the system in its entirety and be forced to deal with the myriad of variables and possibly changing strategies which characterize the behavior of most animals, or he can manipulate the physiological preparation in some way which, in addition to producing the desired effect, introduces unknown and unpredictable side effects and complications. The researcher working with the simulated model does not face this dilemma. Each sub-system or sub-sub-system is available for scrutiny and all interaction can be studied explicitly. He can be sure the strategy of operation will not change when parameters are changed or processes are removed. The physiologist is forced to work with variables of limited value and often prohibited from dealing with those variables of most conceptual importance. The response of the individual cell is recorded rather than that of the ensemble. The EMG is recorded rather than the force delivered by the muscle to the tendon. This is also not a problem for the modeller because every variable is available for measurement and modification.

Perhaps the most important strong point of the synthetic approach to motor research is the effect it has on the researcher's focus of attention. There is no reason to let the anatomist's subdivision of the nervous system be <u>the</u> guiding force in our research if the structures he delineates have little correspondence with the functional divisions of the motor system. Rather, we should organize our thoughts about functional units and conceptual issues, and this type of thinking results when the problems of design are faced. This approach is given further power when the formulations of a model are put in the form of a working mechanism, for only then can we be sure that our suppositions were correct and that all vagaries have been eliminated.

These statements are not meant to be critical of the purely empirical approach. The intention is to describe how in certain ways, the formulation and simulation of models is complementary to the approach normally taken by the neuroscientist, and how these formulations can be useful to him.

It goes almost without saying that the design, implementation, and study of such models can have direct application for those who wish to produce autonomous machines.

Constraints on the Problem

An often neglected first step in the study of the motor system should be the selection of a class of behavior upon which attention may be focussed. The human body is a versatile mechanism capable of a staggering variety of movements. The nervous system is also extremely versatile and employs a number of control strategies. Sometimes a limb is moved with great deliberation and precision in which there is simultaneous activation of agonist and antagonist muscles. Other times more free-flowing motions are made in which agonist muscles accelerate the masses of a limb to high velocity, the limb coasts until it is slowed and stopped by the force of antagonist muscles (16). Contrast, for example, the motion of a delicate paint stroke to that of a baseball pitch. Sometimes interaction with the environment is quite predictable and adjustments are virtually unnecessary while at other times the movement is nothing but a set of constant adjustments to external disturbances. When walking down a flight of stairs the position of each step and moment of foot contact is quite predictable. Conversely, standing still on a moving trolley car requires major adjustments at each lurch on the track. The aim of a movement may be to achieve a certain position, to move at a certain velocity, or to contact an object with a certain force. When one presses a button the position of the finger is important, while the velocity at which the violinist draws his bow influences the sounds which result. Imagine the effects of a masseur who cannot regulate the force of his ministrations.

For each of these types of movement the problems of control are different and one reasonably homogeneous solution would not be expected to apply to every case. In order to develop a model which can help us to better understand movement we should begin by acknowledging this diversity, and restrict our studies to a class of movements, the members of which are produced by one, unified control scheme or strategy. Of course, our readiness to make this choice indicates our belief in a certain discreteness of control function.

The model discussed here was designed to process ballistic movement. This means that once a movement is initiated no sensory information is used to alter that movement during its execution. Such movement can also be called open-loop, but it should be realized that, ultimately, the loop is closed. This is so because, though the sensory information obtained during a movement is not used to alter the progress of that movement, it may be used to alter the motor system in such a way that subsequent movements are influenced. A special case of the ballistic movement is that which is composed of a number of short, open-loop segments executed in sequence, where the sensory information produced during one segment influences the production of subsequent segments. This variation is not dealt with here, but is mentioned in order to suggest the ultimate usefulness of solutions to the open-loop control and learning problems. Results of research by Hammond (9) and Melvill-Jones (20) regarding the latency of effective compensation to disturbances in man further substantiate this view.

Since we are only studying open-loop movements, one might expect us to make the further restriction that all interactions with environmental disturbance be quite predictable. In fact, for simplicity, all interactions with the environment, other than gravity, have been eliminated from consideration. We are specifically interested in controlling limb position as a function of time in the absence of environmental influence.

The model under consideration here was not designed to account for all motor function. In addition to restricting the class of movements under study, we have limited the type of processing to be described. This means that other motor processors work along with the sub-system described here and the learning or execution of even a single movement relies on a number of processing elements. Figure 1 is an example of a familiar demonstration. Changing the motor apparatus used in the production of writing does not change the essential form of the characters produced, even though the muscular commands involved must be very different. Unless the subject learned to produce each form of output separately, (this was not the case in the example shown) we may draw two conclusions:

1) Motor programs exist in the nervous system which are expressed in a language which is independent of muscular and kinematic considerations. One such program can be used to produce movements in any of a number of limbs or body parts.

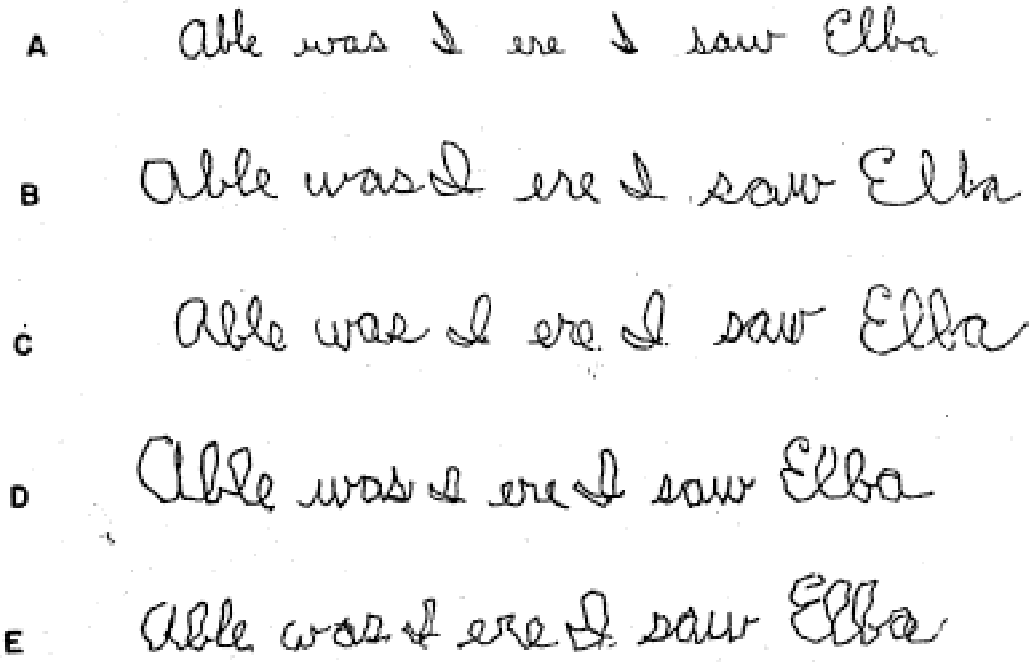

Fig. 1 Writing with the pen held by different parts of the body does not change the shapes of the letters even though different muscle and skeletal systems are used to produce each example. A) Pen held by right hand (dominant). B) Right arm was used to produce writing. C) Pen held in left hand. D) Pen held in mouth. E) Pen was taped to right foot. The subject had essentially no previous experience writing with any body part other than A.

2) Mechanisms exist in the nervous system which can translate general motor programs (as described in 1) into explicit instructions suitable for the muscles, mechanics, and sensors of a particular limb.

This type of architectural arrangement has been discussed by Arbib (1) and Waters (25) and the translation process has been mentioned by Marr (18) and Gelfand,et al (8). The power of such an arrangement is quite attractive. High level processors may formulate new movements or modify old ones, or make combinations without having to take the mechanical properties of the effectors into consideration. It is supposed that these processors may perform symbolic operations through which planning and strategy decisions may also be made. They specify to the translator what the output of the limb should be.

The translating mechanism, on the other hand, is not organized around motor programs, but around the myo-mechano-sensory system with which it communicates. It is free from the responsibilities of strategy and planning, and need not be capable of performing symbolic operations. Its only duty is to accept detailed descriptions of movements and translate them into appropriate muscular commands. But to perform this function kinematic, dynamic, and muscular information about the limb must be available in a usable form. This information may not be present in the infant, and certainly must change as the organism grows. It is therefore important to the effective operation of the translator that some mechanism maintains an up-to-date source of mechanical information about the limb.

The translating mechanism which converts descriptions of desired output into motor commands plus the support mechanism which acquires mechanical information and stores it in a usable form are the topics of interest in this paper and will be referred to collectively as the translator.

The Model

Let us begin the presentation of the model's operation by examining the nature of the computations required by the translation process. Descriptions of movements must be converted into motor commands. The acceleration of an object, taken with its initial conditions, gives a complete description of its movements and the force on an object is that which commands its every motion. For this simple, unconstrained system we can specify the desired acceleration and use Newton's equation, $F = Ma$, as a translator to find the necessary force. Of course, this also applies to rotary motion, $T = J\ddot{\emptyset}$, where T is the torque, J is the moment of inertia, and $\ddot{\emptyset}$ is the angular acceleration. If only a limb were so simple as that.

What is the acceleration of a limb? If we take the simple case where the coordinate system of interest is that of the limb's joints we can describe the acceleration by a vector, $\underline{\ddot{\emptyset}}$, whose number of elements equals the number of joints, N. ($\emptyset$ is the angular position of one joint and each prime marker indicates differentiation with respect to time. All vectors and matrices are delineated by an underscore.) The moment of inertia, however, must be expressed as a square matrix of rank N because, as mentioned earlier, torques applied to one joint will cause accelerations about every joint in the limb (See Fig. 2). The $\underline{J}$ matrix specifies the relationship between the torques and the resulting accelerations at each joint in the limb; $j_{ik} = \frac{\text{torque applied to joint k}}{\text{acceleration at joint i}}$. Unfortunately, the elements of this matrix are not constants for two reasons. Firstly, the amount of interaction between two joints is dependent on the angular position, $\underline{\emptyset}$, of the joints of the limb (also an N-element vector). For some configurations of the limb the amount of interaction is large while

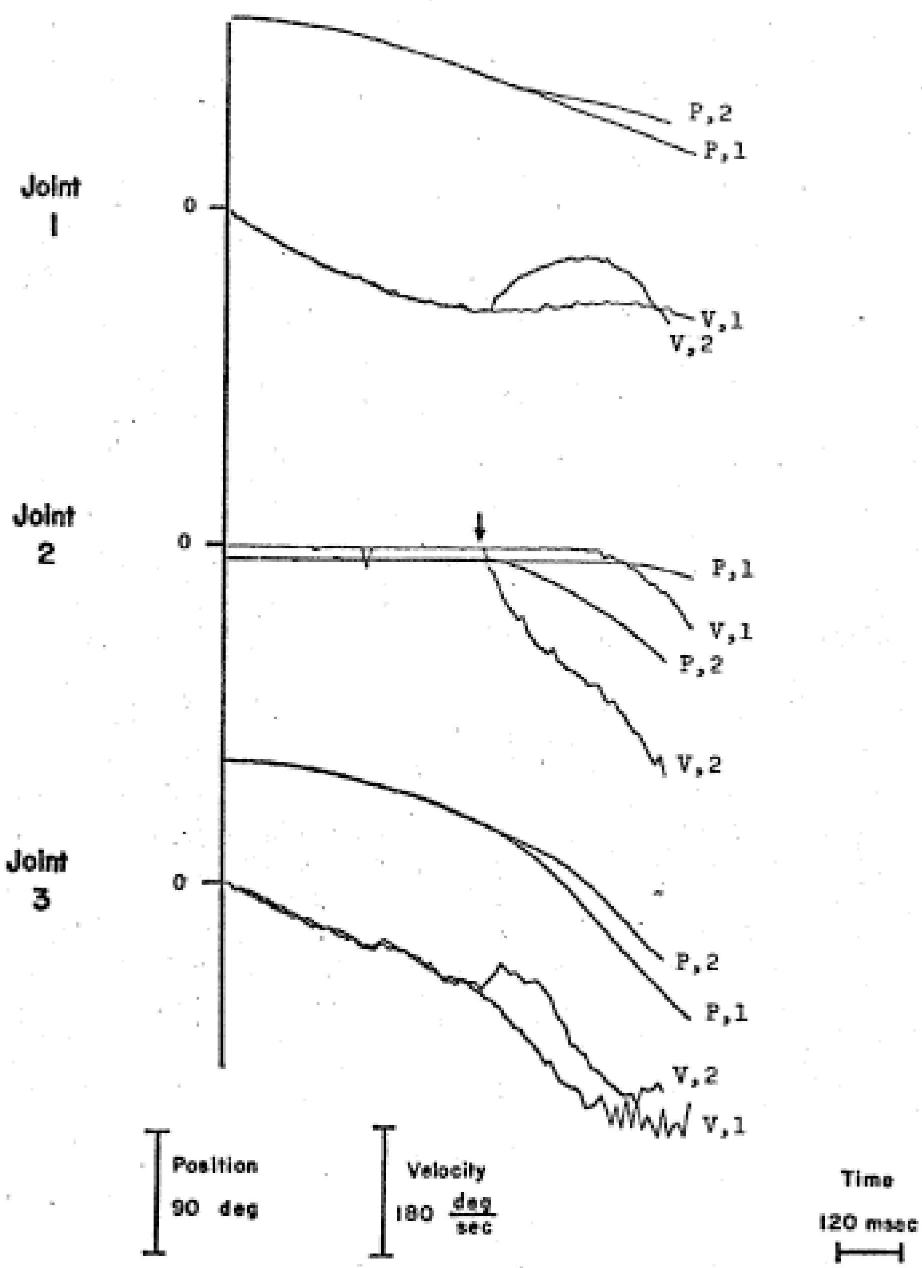

Fig. 2 The MIT-Vicarm manipulator (see Fig. 5) was used to demonstrate the potential for mechanical interactions among joints of a limb. These graphs show how torques applied to one joint of the manipulator can influence other joints. Each movement labelled as 1 was made by applying constant torque to each joint. In movement 2 the torque at joints 1 and 3 were unchanged, but a step of torque was applied to joint 2 after 500 msec. (at arrow). Note that the position and velocity trajectories of all three joints were affected. P-position; V-velocity.

for others it is small (See Fig. 3a and b). Secondly, the effective moment of inertia of a joint is determined, not only by the masses of the links which are moved, (a link is that part of a limb between two joints) but also by the distances between the masses and the center of rotation (See Fig. 3c and d).

For the limb having N joints, N links, N accelerations, and N^2 moments of inertia there are N torques. In addition to the torques applied by the muscle we must consider the acceleration of gravity and the damping forces due to friction. The acceleration of gravity must be represented by a vector of dimension N because each mass in the limb will be accelerated individually. These gravity factors are also not constant but depend on the relative positions of the masses and joints in the limb (See Fig. 3e and f). Frictional torques, also representable by an N-dimensional vector, are independent of $\underline{\emptyset}$, but depend on the velocity of the moving joint. Unlike the gravitational and moment of inertia terms, which, in principle, may be calcualted from a blueprint of a limb, the frictional terms often bear a complex relationship to the velocity of a joint.

A final factor relevant to the equations of motion which only introduces appreciable torques at high velocities, is the coriolis term. This torque is produced by accelerating an object about one axis while it is rotating about an orthogonal axis, and the direction of its action is about a third axis orthogonal to the plane of the other two.

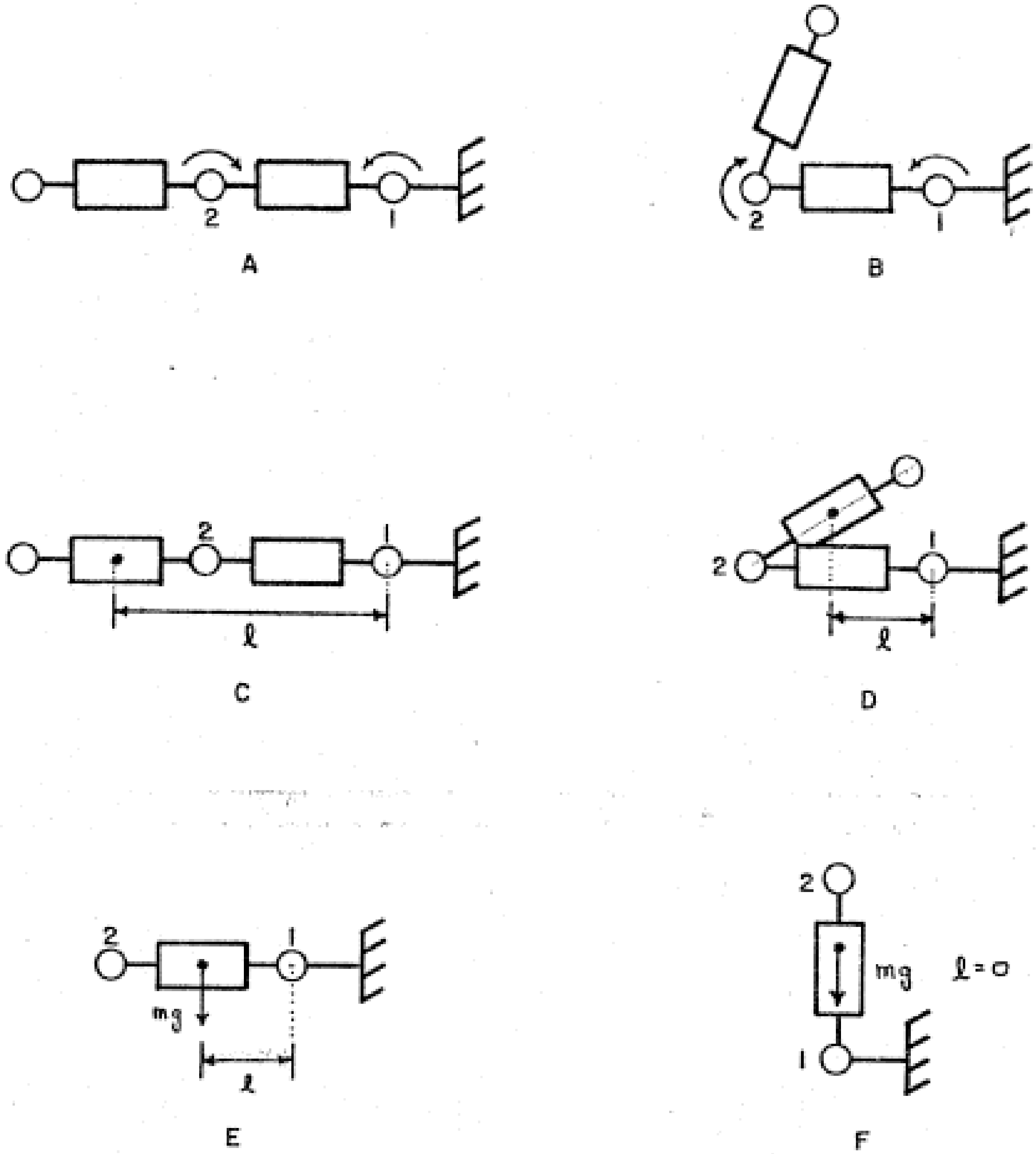

Fig. 3 A) Accelerations of joint 1 will cause large accelerations about joint 2 due to reaction torques. B) The reaction torque on joint 2 is smaller here than in A because of the position of joint 2 and its link. C) The moment of inertia of joint 1 is maximum because the center of mass of link 2 is far from the center of rotation. D) Here the moment of inertia is almost a minimum. E) The effects of the acceleration of gravity depend on the moment arms through which it acts. Here it is maximum for the link. F) The moment arm is zero and no torque is produced about joint 1 by gravity.

If we rewrite Newton's equation, but include each term introduced above we get:

$$\underline{T} - \underline{G}(\underline{\emptyset}) - B(\underline{\dot{\emptyset}}) - C(\underline{\emptyset},\underline{\dot{\emptyset}}) = J(\underline{\emptyset}) \cdot \underline{\ddot{\emptyset}} \qquad \text{(eq. 1)}$$

where: $\underline{T}$ is the muscle torque vector
$\underline{G}$ is the gravitational torque vector
$\underline{B}$ is the frictional torque vector
$\underline{C}$ is the coriolis torque vector
$\underline{J}$ is the moment of inertia matrix
$\underline{\emptyset}$, $\underline{\dot{\emptyset}}$, and $\underline{\ddot{\emptyset}}$ are the position, velocity, and acceleration vectors.

This equation may still look manageable, but the dependencies on $\underline{\emptyset}$ and $\underline{\dot{\emptyset}}$ are only implicit. Kahn (15) has worked out the explicit relationship using a computer program which performs algebraic manipulations and his results for a general limb having three links and three joints ($N = 3$) but no friction is reproduced in part in appendix A. These results are almost intractable, (the equations involve about 1600 terms and 13,000 multiplications) and virtually useless for a theory of motor function.

But before we totally discard these results let us examine a special set of circumstances under which simplifications can be made. If we look at the behavior of the limb during a very short interval of time we observe that eq. 1 still describes the behavior of the limb, as it must, but each term can be simplified. During a short interval, call it a time slice, or just a slice, we see that each joint has position and velocity, but during the slice their values change by only small amounts. We may neglect these small changes or reduce the duration of the slice to the point where the changes in velocity and position may be neglected. Once this is done each element of a vector or matrix in eq. 1 which had been dependent on the state of the system (the state of the limb is uniquely determined by the positions and velocities of all the joints) becomes a constant. We can represent our equations of motion as:

$$\underline{T} - \underline{G} - \underline{B} - \underline{C} = \underline{J} \cdot \ddot{\underline{\phi}} \qquad \text{(eq. 2)}$$

By grouping terms and making the equation explicit in the torque exerted by the muscle we can make one further simplification:

$$\underline{T} = \underline{J} \cdot \ddot{\underline{\phi}} + \underline{K} \qquad \text{(eq. 3)}$$

where: $\underline{K} = \underline{G} + \underline{B} + \underline{C}$

It must be remembered that this equation only applies to the motion of the limb during one time slice, and only the slice to which apply the values of the $N^2 + N$ constants which comprise $\underline{J}$ and $\underline{K}$. Nothing prevents us, however, from applying eq. 3 to other time slices provided we can find values of $\underline{J}$ and $\underline{K}$ appropriate to the state of the system prevailing during those slices. We can describe eq. 3 as the piece-wise linearized version of eq. 1, and the state for which the constants are chosen as the operating point. Although the development so far indicates that we calculate the torque needed at each joint of the limb, we will see shortly that the value calculated can be the net force exerted by the tendon, or a special version of the command to the muscle.

Supposing we have available the constants required, we could take the description of an entire movement, slice it up into enough time intervals so that the change in position and velocity for each joint is negligible, and determine the muscular torque needed to produce the desired acceleration for each interval. If the appropriate initial conditions were satisfied and each torque were applied for the duration of the interval for which it was computed, the resulting movement would closely resemble the originally specified movement. The reproduced movement could be made arbitrarily close to the desired movement by reducing the duration of the time slices, provided that the constants needed were available for each of these new, shorter slices. This scheme will only work if the accelerations present in the description of the desired movement are limited in magnitude to those produceable by the limb's muscles. Violation of this restriction will result in specification of a torque vector which is not achievable and the resulting motion, assuming that some attainable torque is used instead, will not conform to the desired response. It should be realized that this problem must be faced by any solution to the translating problem and is not unique to the solution given here.

The solution given so far is only a partial description of the computations performed by the translator since we have not yet indicated how the constants which describe the mechanical nature of the system are found, nor how they are affected by motor experience and changes in the mechanical system.

In 1950 von Holst and Mittlestadt used the concept of efference copy in a model which was designed to account for the ability of a fly to distinguish between internally and externally produced changes in sensory stimulation (22,24). 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. This internal signal is known as the efference copy. In Held's model of 1961 the Holstian view was augmented to allow attainment of perceptual accuracy even after changes were made to the meaning of sensory signals (10,11). In this model the efference copy was used to elicit the trace of previous re-afference and this trace was compared to the current afference. Young and Stark proposed an elaborate model in 1965 in order to account for the ability of a human performing a tracking task to change control strategies when there were changes in the dynamics of the controlled element (28). In that model the efference copy was used to drive an internal dynamic model of the controlled element, and the output of that model was compared with the afference from the control task.

In the present model the relationships between the efference copy and the re-afference are used to determine the constants which represent the mechanical properties of the limb and are used by the translating mechanism. As will be seen, 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, detailed information about the mechanics of the limb are found by examining the limb's input-output relations. Mechanical properties are derived directly from the results of the organism's attempts to move.

We return, once more, to the special equations of motion which govern the system's behavior during one time slice, with the understanding that what must be found are the $N^2 + N$ constants which comprise $\underline{J}$ and $\underline{K}$. If we consider the scalar equation ($N = 1$):

$$t = j \cdot \ddot{\phi} + k$$

it is known that we can find j and k by simultaneously solving two equations in two unknowns. We first make measurements of the torque and acceleration for two movements and calculate the desired data:

$$k = t_1 - \frac{t_1 - t_2}{\ddot{\phi}_1 - \ddot{\phi}_2} \cdot \ddot{\phi}_1$$

$$j = \frac{t_1 - t_2}{\ddot{\phi}_1 - \ddot{\phi}_2} \qquad \text{(eq. 4)}$$

For the case where $N \neq 1$, (for a limb having a number of joints) we must make $N^2 + N$ measurements of torque and acceleration and solve $N^2 + N$ equations. By analogy to eq. 4:

$$\underline{K} = \underline{T}_1 - [(\underline{P} - \underline{Q}) \cdot (\underline{R} - \underline{S})] \cdot \underline{\ddot{\phi}}_1$$

$$\underline{J} = (\underline{P} - \underline{Q}) \cdot (\underline{R} - \underline{S})^{-1} \qquad \text{(eq. 5)}$$

where:

$$\underline{P} = [\underline{T}_1 \vdots \underline{T}_2 \vdots \underline{T}_3 \vdots \ . \ . \ . \ \ \underline{T}_N]$$

$$\underline{Q} = [\underline{T}_{N+1} \vdots \underline{T}_{N+1} \vdots \underline{T}_{N+1} \vdots \ . \ . \ . \ \ \underline{T}_{N+1}]$$

$$\underline{R} = [\underline{\ddot{\phi}}_1 \vdots \underline{\ddot{\phi}}_2 \vdots \underline{\ddot{\phi}}_3 \vdots \ . \ . \ . \ \ \underline{\ddot{\phi}}_N]$$

$$\underline{S} = [\underline{\ddot{\phi}}_{N+1} \vdots \underline{\ddot{\phi}}_{N+1} \vdots \underline{\ddot{\phi}}_{N+1} \vdots \ . \ . \ . \ \ \underline{\ddot{\phi}}_{N+1}]$$

$\underline{T}_i$ and $\underline{\ddot{\phi}}_i$ are the i'th measurements of $\underline{T}$ and $\underline{\ddot{\phi}}$.

We see that these calculations can be performed if N + 1 sets of $\underline{\ddot{\phi}}_i$ and $\underline{T}_i$ are available where the acceleration vector is the response produced by issuing the torque vector as a command. These computations derive information about the mechanics of the system from the relationships between the efference copy, $\underline{T}$, and the re-afference, $\underline{\ddot{\phi}}$. Once again remember, we need to find the values of the $N^2 + N$ constants which are appropriate to the state which prevails during a particular time slice. If this is to be so, each measurement contributing to the calculation made by eq. 5 must have been made while the system was in or near the state of interest.

The procedure for finding the values of the mechanical constants $\underline{J}$ and $\underline{K}$ for one time slice are given above, but our goal is to process movements which are composed of many slices, each of which may correspond to different mechanical states of the limb. To ensure the achievement of this goal the operations of collecting data and calculating constants must be organized. The necessary organization arises from the consideration of a discrete state space and the use of two types of memory; the temporary buffer and the state space memory.

The system cannot have stored, nor can it calculate the constants needed for every attainable state of the limb, for the number of such constants is infinite. The best it can do is let each state be 'near' a state for which data are stored or can be stored. Let us divide the range of each dimension of the state space, (each joint's position and velocity) into M intervals. The 2N dimensional state space is then partitioned into $M^{(2N)}$ regions or hypercubes. If M is chosen to make the size of each hypercube reasonably small, and the values of $\underline{J}$ and $\underline{K}$ are available for one state in the hypercube, then all the states in that hypercube can be said to be near a state for which data are stored. If all the measurements contributing to the calculation of a set of constants, $\underline{J}$ and $\underline{K}$, were generated while the state of the limb was in one hypercube, the assumption can be made that the constants correspond to a state in that hypercube. This statement will surely be true for large M.

If one keeps in mind this notion of a discrete state space, the operation of the translator with respect to the acquisition of constants of mechanical description can be made clear. During self-produced movements data are generated which must subsequently be used to calculate the constants of mechanical description. The data for these computations are pairs of simultaneously generated acceleration and torque vectors. These pairs of vectors cannot always be used immediately because each application of eq. 5 requires N + 1 sets of vectors from the same region of state space. Since the state of the limb is constantly changing, only a limited amount of data from each movement is pertinent to a given region of the state space at a time, and the data that are available must be saved. Hence the temporary buffer. Although its use is quite different, the type of data stored in this buffer is similar to that of

Held's correlation store (10,11).

When N + 1 pairs of vectors from the same region of state space accumulate in the temporary buffer, $\underline{J}$ and $\underline{K}$ are calculated by the translator, and they must be saved. The state space memory is organized so that it can store $N^2 + N$ constants for each hypercube of the discrete state space -- $(N^2 + N) \cdot (M^{2N})$ constants in all. In certain cases values for $\underline{J}$ and $\underline{K}$ will be calculated for regions of the space for which previous results exist. In order to reduce noise and provide the ability to adapt to changes in the mechanical properties of the system, new and old values of $\underline{J}$ and $\underline{K}$ are averaged with some sort of weighting which favors recent data.

Figure 4 is a diagram of the system under discussion. Operation of the model can be summarized as follows. High level processors produce descriptions of desired movements which are presented to the translator. These descriptions explicitly state the time course of the movement so that position, velocity, and acceleration information are available for each dimension of the coordinate space in use. The desired movement is sectioned into time intervals or slices, each of duration Δt. For each time slice eq. 3 is used in conjunction with the mechanical information in the state space memory, to find the forces which will be used in an attempt to produce a movement having the desired accelerations. The command forces calculated are issued to the limb and, while the movement is in progress a copy of the command, the efference copy, and a copy of the sensory signals which indicate the progress of the movement, the re-afference, are stored in the temporary buffer with labels which indicate the region of the state space to which they apply. Subsequently, the contents of the temporary buffer and eq. 5 are used to

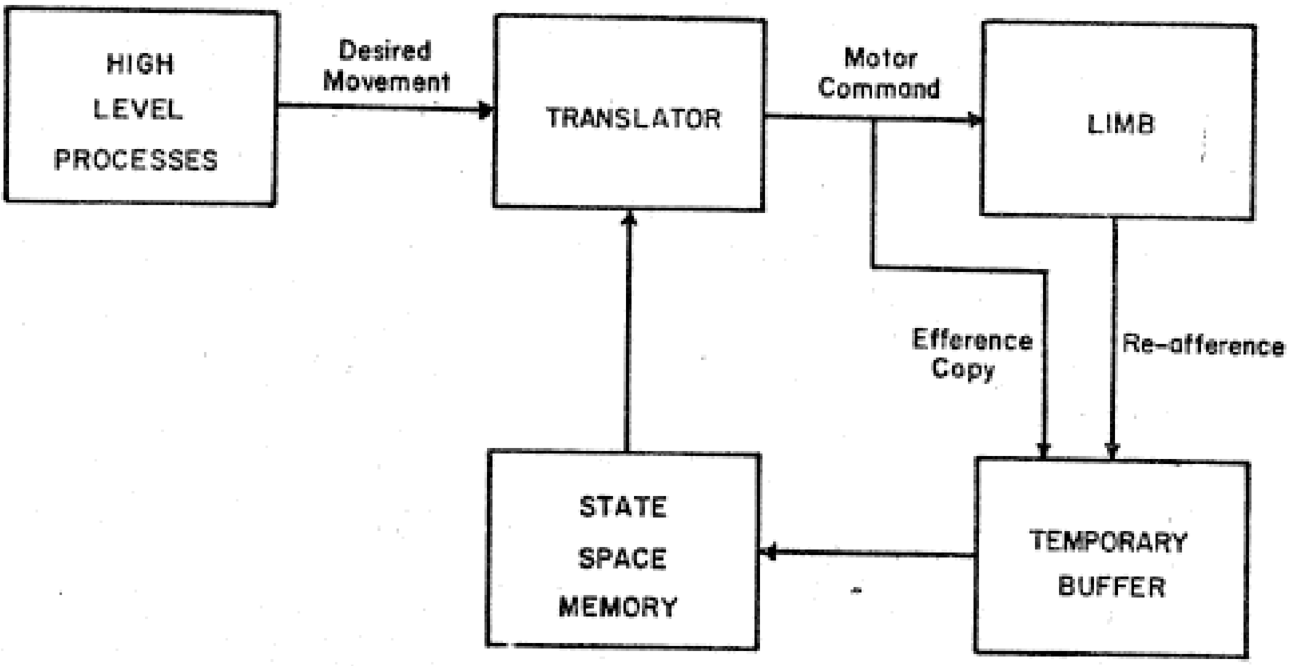

Fig. 4 The major components of the translator model are shown. A description of its operation is given in the text.

find values of $\underline{J}$ and $\underline{K}$, and these results are stored in the state space memory in combination with data that might have been stored there previously.

It is important to note that the geometry and linearity of the sensors and the geometry of the limb place no constraints on the translator presented here. This is a direct result of the piece-wise linearization process to which we have submitted the mechanical system. From a practical point of view, this means that re-afference can take the form of visual feedback just as readily as joint oriented proprioceptive feedback. It means that the joints can be revolute or sliding. It also means that the forces applied by the muscles can undergo non-linear transformations due to the joint-tendon geometry, without consequence.

Unfortunately, one constraint still exists on the relationship between the motor command and the force exerted by a muscle on its tendon. This relationship must be linear in the following sense:

$$T = a(\underline{\phi},\underline{\dot{\phi}}) \cdot E + b(\underline{\phi},\underline{\dot{\phi}}) \qquad \text{(eq. 6)}$$

where: T is the force produced at the tendon
E is the motor command
$a(\underline{\phi},\underline{\dot{\phi}})$ and $b(\underline{\phi},\underline{\dot{\phi}})$ are state dependent constants.

This restriction says that for any given state of the limb, incremental changes in the motor command must produce proportional changes in the tendon force of the muscle. This is a very weak form of linearity. It does not say that the force in the tendon needs to be proportional to the motor command, nor that the force delivered by the muscle must be constant if the command does not change. Indeed, if a human arm is moving and the command to the muscle does not change, the force at the tendon will increase if the muscle is stretched and decrease if unloaded. The only requirement is that, given the same mechanical conditions (i.e. the state does not change) all increases in command produce changes in force which are related by a constant multiplier.

Actually, the fact that this relationship must be true for this model to work, even if it requires the presence of some processing to ensure it, can be viewed as a feature from the translator's point of view. If this relationship were not guaranteed, the adaptive nature of the translator would respond to changes in the mechanical system caused by muscular fatigue, and that process would reduce the system's immunity to noise.

Behavior of the Model

Initially, the performance of the translating mechanism will be quite poor. Every attempt to use information about the mechanical character of the limb will be frustrated because the state space memory will be empty. Data about the mechanics of the limb are only available after movements have been processed. If no data from the state space memory are available two things can happen. The translator can use some preset or genetically encoded constants and proceed to generate a set of commands even though the resulting movement may be quite different from the one desired. Alternately, some other control system could take over when the translator finds that it has no usable information. Under this circumstance the translator would not take part in the production of the movement. In either case it is important that the remainder of the translator's functions (i.e. the analysis of the efference copy and the re-afference) be performed when the movement is executed, even though the resulting movement may bear little resemblance to that specified by the high level processor. If this were not the case the translator would never have the opportunity to build up its memory and improve. (This would be something like the fellow who cannot get a job because he has no experience, and cannot get any experience without a job.)

As more and more movements of the limb are made, more and more data describing the mechanics of its operation are available to the translator. During this period of data acquisition the quality of movement produced by the translator will gradually improve. It should be stressed that the reason for this improvement is not that errors in previous movements are explicitly corrected, nor that errors in the constants which specify the mechanical properties are explicitly corrected. Movement errors can only be detected if a comparison between desired and produced movements is made and this is never done by the system presented here. Motor performance is gradually improved with experience for two reasons:

1) Each movement submitted for translation requires data from a number of regions in the state space memory. More of these data are availablewhen the system is more experienced, because these data are generated directly from the movements which comprise experience.

2) If there is any noise in the system (there always is noise in physical systems) the data available from the state space memory become more accurately specified when they are calculated a number of times because noise is reduced through averaging.

Though the general level of motor performance is improved with experience, the performance of a specific movement can be improved through concentrated practice. In this case improvement is accelerated because a higher percentage of the incoming data are relevant to the regions of the state space memory which will be used to attempt the movement of interest. It is also true that more movement data of any kind are available during practice. While heavy practice of one or a group of movements should improve the ability to make the practiced movements, in some cases other movements will also be facilitated. This will be true when the other movements are similar to those practiced. By similar we mean that the same regions of the state space memory are used to generate the movement. This type of transfer might correspond to Thorndyke's "identical elements" theory, although he probably had a higher level process in mind (13,19). The type of transfer described here, from a highly practiced movement to a similar, but less practiced one also contributes to the appearance of a general improvement of motor performance. In fact, the characteristic which prompts us to call the improvement general is that entirely new movements are performed with only modest amounts of error, though never before explicitly practiced. It must be understood that the effectiveness of concentrated practice upon the practiced and similar movements is influenced to a large degree by the details of the practice strategy -- details which are not considered here.

When the constants for a region of the state space memory are calculated a number of times, we can expect the average of those calculations to converge upon the true value of the mechanical properties they represent. This will happen when the mechanical properties of the limb are constant but there is noise in the system. In the event the mechanical properties are not constant -- a situation which can occur when the organism grows, the muscles get stronger, or the sensory elements change -- repeated calcualtions of the mechanical constants will reflect the changing properties and ultimately converge some time after the changing limb stabilizes. The exact nature of this adaptation process will depend on the rules of combination which apply to the storage of new data in the state space memory. The only statement on this score to be made here is that a weighted average which favors recent data will perform in an adaptive way. Improved noise rejection will be demonstrated, however, if the time-constant of the memory is as long as possible, while still being short with respect to the time-constant of changes in the mechanical properties of the limb.

At this point we must reiterate that all the properties of the motor system are not being attributed to the translating mechanism. Just because the translator learns and adapts does not mean that other processors do not also learn and adapt. It is assumed that they do.

Testing the Model

The model presented here will be most useful after it has been tested. This is so for the neurophysiologist who wants to understand man and for the engineer who wants to design more advanced autonomous machines.

To many neurophysiologists, testing a model of motor function means showing that the biological motor system conforms to predictions drawn from the model. There is another kind of test -- one which verifies that the properties and powers alleged to come with a model are actually products of the operations and computations it performs. This second type of test should be required of a model before its predictions are related to the biological organism, for only after such a test does confirmation of predictions about the biological system result in our increased understanding. Without verification accurate predictions only mean that the modeller predicted correctly, not that the model describes the behavior under study. While issues regarding the biological system are of little consequence to the engineer, he must also have a verified model if he is to proceed to problems of technology, cost, practicality, and usefulness.

We have begun to conduct tests of an implementation of the model presented. The embodiment of the motor system used in those tests consists of a small computer (PDP11/45), some simple analog circuitry, and the MIT-Vicarm manipulator (See Fig. 5). Preliminary results show a rudimentary ability to learn, but the presentation of conclusive data awaits the elimination of some methodological difficulties and the completion of a thorough set of experiments. These experiments are designed to show that:

1) The system can use the results of practice to improve its performance.

2) Attempts to practice one movement can favorably influence performance of other movements.

3) The system can adapt to mechanical disturbances caused by the application of inertial, constant force, and variable force loads.

4) The system can adapt to disturbances caused by distortions of the sensory signals used in learning.

5) The form of the reafference signal is not constrained to any one coordinate system.

The results of these experiments will be presented in a forthcomming companion to the current article.

Fig. 5 (next page) The layout of the first three joints of the MIT-Vicarm manipulator are shown. $\emptyset_1$ acts about the vertical axis. The manipulator is about the size of a human arm; l_0=.273m, l_1=l_3=.059m, l_2=l_4=.203m. Each joint is provided with a DC torque motor, a potentiometer, a tachometer, and a clutch-type brake. The diagram is from reference 14 with modifications.

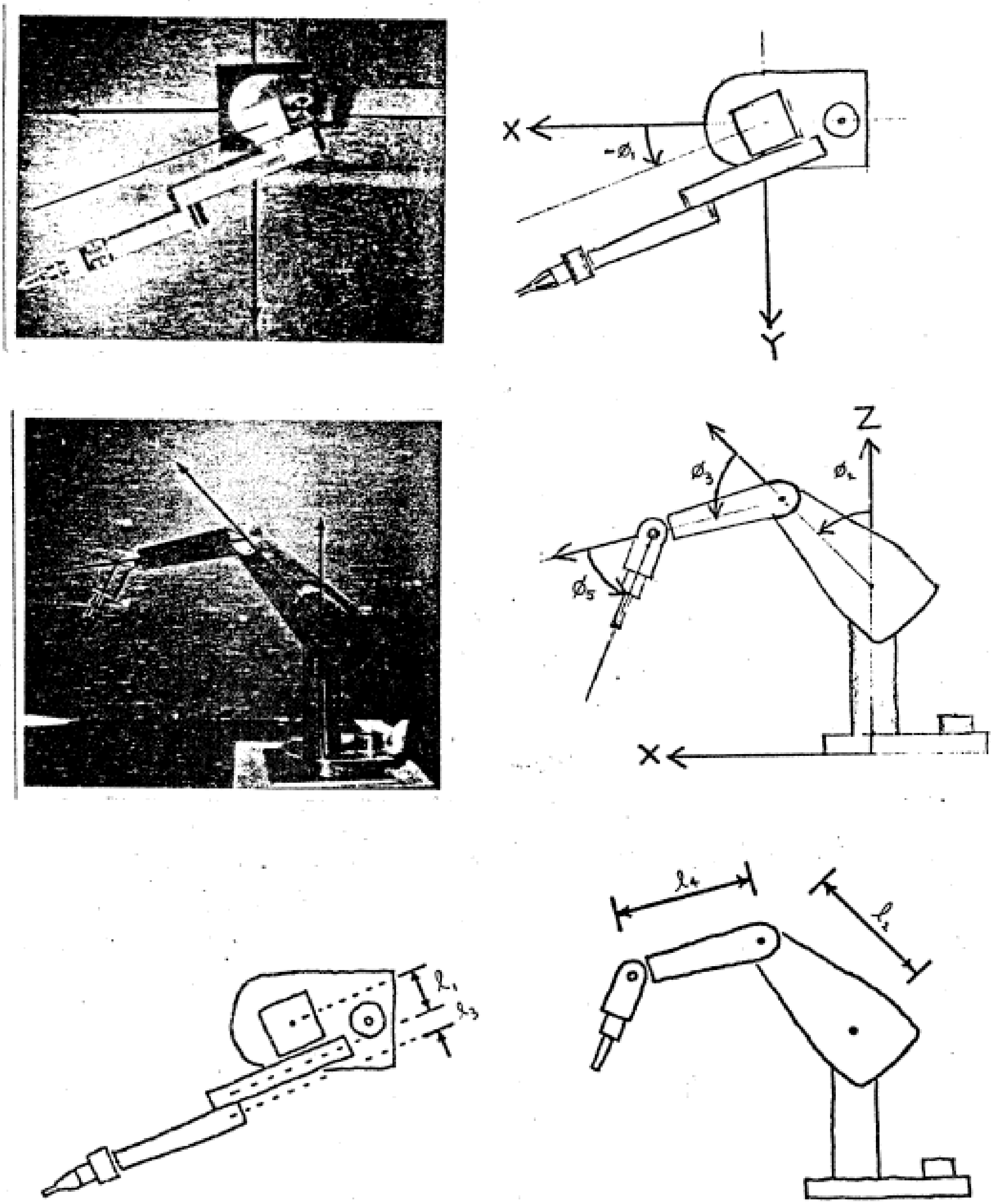

Fig. 5

Appendix A

The following is taken from appendix A found in Kahn (15). Of the 15 pages needed to expand equations A.1, A.2, and A.3 only 3 are shown here to give the reader the flavor of his findings.

EQUATIONS OF MOTION FOR A GENERAL KINEMATIC CHAIN CONTAINING THRFE REVOLUTE JOINTS

In this appendix, equations of motion are given for a general kinematic chain containing three revolute joints. These equations were obtained by expanding the matrix products in Eq. (2.18) on a digital computer using an algebraic manipulation program called REDUCE [46]. In terms of the REDUCE notation, the equations of motion are

$$\begin{aligned} &M11\ \ddot{\theta}_1 + M12\ \ddot{\theta}_2 + M13\ \ddot{\theta}_3 + D111\ \dot{\theta}_1^2 + D122\ \dot{\theta}_2^2 + D133\ \dot{\theta}_3^2 \\ &+ D112\ \dot{\theta}_1\dot{\theta}_2 + D113\ \dot{\theta}_1\dot{\theta}_3 + D123\ \dot{\theta}_2\dot{\theta}_3 + GR1 = T_1 \end{aligned} \tag{A.1}$$

$$\begin{aligned} &M12\ \ddot{\theta}_1 + M22\ \ddot{\theta}_2 + M23\ \ddot{\theta}_3 + D211\ \dot{\theta}_1^2 + D222\ \dot{\theta}_2^2 + D233\ \dot{\theta}_3^2 \\ &+ D212\ \dot{\theta}_1\dot{\theta}_2 + D213\ \dot{\theta}_1\dot{\theta}_3 + D223\ \dot{\theta}_2\dot{\theta}_3 + GR2 = T_2 \end{aligned} \tag{A.2}$$

$$\begin{aligned} &M13\ \ddot{\theta}_1 + M23\ \ddot{\theta}_2 + M33\ \ddot{\theta}_3 + D311\ \dot{\theta}_1^2 + D322\ \dot{\theta}_1^2 + D333\ \dot{\theta}_3^2 \\ &+ D312\ \dot{\theta}_1\dot{\theta}_2 + D313\ \dot{\theta}_1\dot{\theta}_3 + D323\ \dot{\theta}_2\dot{\theta}_3 + GR3 = T_3 \end{aligned} \tag{A.3}$$

In other words, M12 is the coefficient of $\ddot{\theta}_2$ in the first equation, D213 is the coefficient of $\dot{\theta}_1\dot{\theta}_3$ in the second equation, and GR3 is the gravity term in the third equation; other terms are similarly defined.

```
      M1 *
       (A1**2   +   2 * A1 * X1   +   SA1**2 * K122   -   SA1**2 * K133   -   2 * SA1 * CA1 * K123   +   K133)
   +   M2 * ST2**2 *
       ( -   A2**2 * SA1**2   -   2 * A2 * X2 * SA1**2   +   SA1**2 * SA2**2 * K222   -   SA1**2 * SA2**2 * K
233   -   2 * SA1**2 * SA2 * CA2 * K223   -   SA1**2 * K222   +   SA1**2 * K211)
   +   M2 * ST2 * CT2 *
       ( -   2 * A2 * Y2 * SA1**2 * CA2   +   2 * A2 * Z2 * SA1**2 * SA2   +   2 * SA1**2 * SA2 * K213   -
2 * SA1**2 * CA2 * K212)
   +   M2 * ST2 *
       ( -   2 * A1 * Y2 * CA2   +   2 * A1 * Z2 * SA2   -   2 * A2 * S2 * SA1 * CA1   -   2 * A2 * Y2 * SA1
* CA1 * SA2   -   2 * A2 * Z2 * SA1 * CA1 * CA2   -   2 * S2 * X2 * SA1 * CA1   -   2 * SA1 * CA1 * SA2 * K212
 -   2 * SA1 * CA1 * CA2 * K213)
   +   M2 * CT2 *
       (2 * A1 * A2   +   2 * A1 * X2   -   2 * S2 * Y2 * SA1 * CA1 * CA2   +   2 * S2 * Z2 * SA1 * CA1 * SA2
 +   4 * SA1 * CA1 * SA2**2 * K223   +   2 * SA1 * CA1 * SA2 * CA2 * K222   -   2 * SA1 * CA1 * SA2 * CA2 * K233
 -   2 * SA1 * CA1 * K223)
   +   M2 *
       (A1**2   +   A2**2   +   2 * A2 * X2   +   S2**2 * SA1**2   +   2 * S2 * Y2 * SA1**2 * SA2   +   2 * S2
* Z2 * SA1**2 * CA2   -   2 * SA1**2 * SA2**2 * K222   +   2 * SA1**2 * SA2**2 * K233   +   4 * SA1**2 * SA2 * C
A2 * K223   +   SA1**2 * K222   -   SA1**2 * K233   +   SA2**2 * K222   -   SA2**2 * K233   -   2 * SA2 * CA2 *
K223   +   K233)
   +   M3 * ST2**2 * ST3**2 *
       ( -   A3**2 * SA1**2 * SA2**2   +   2 * A3**2 * SA1**2   -   2 * A3 * X3 * SA1**2 * SA2**2   +   4 * A
3 * X3 * SA1**2   +   K311 * SA1**2 * SA2**2   -   2 * K311 * SA1**2   +   K322 * SA1**2 * SA2**2 * SA3**2   -
 K322 * SA1**2 * SA2**2   -   2 * K322 * SA1**2 * SA3**2   +   2 * K322 * SA1**2   -   K333 * SA1**2 * SA2**2 *
SA3**2   +   2 * K333 * SA1**2 * SA3**2   -   2 * K323 * SA1**2 * SA2**2 * SA3 * CA3   +   4 * K323 * SA1**2 * S
A3 * CA3)
   +   M3 * ST2**2 * ST3 * CT3 *
       ( -   2 * A3 * Y3 * SA1**2 * SA2**2 * CA3   +   4 * A3 * Y3 * SA1**2 * CA3   +   2 * A3 * Z3 * SA1**2
* SA2**2 * SA3   -   4 * A3 * Z3 * SA1**2 * SA3   -   2 * K312 * SA1**2 * SA2**2 * CA3   +   4 * K312 * SA1**2 *
 CA3   +   2 * K313 * SA1**2 * SA2**2 * SA3   -   4 * K313 * SA1**2 * SA3)
   +   M3 * ST2**2 * ST3 *
       (2 * A2 * Y3 * SA1**2 * CA3   -   2 * A2 * Z3 * SA1**2 * SA3   -   2 * A3 * S3 * SA1**2 * SA2 * CA2   -
 2 * A3 * Y3 * SA1**2 * SA2 * CA2 * SA3   -   2 * A3 * Z3 * SA1**2 * SA2 * CA2 * CA3   -   2 * S3 * X3 * SA1**2
* SA2 * CA2   -   2 * K312 * SA1**2 * SA2 * CA2 * SA3   -   2 * K313 * SA1**2 * SA2 * CA2 * CA3)
   +   M3 * ST2**2 * CT3 *
       ( -   2 * A2 * A3 * SA1**2   -   2 * A2 * X3 * SA1**2   -   2 * S3 * Y3 * SA1**2 * SA2 * CA2 * CA3   +
  2 * S3 * Z3 * SA1**2 * SA2 * CA2 * SA3   +   2 * K322 * SA1**2 * SA2 * CA2 * SA3 * CA3   -   2 * K333 * SA1**
2 * SA2 * CA2 * SA3 * CA3   +   4 * K323 * SA1**2 * SA2 * CA2 * SA3**2   -   2 * K323 * SA1**2 * SA2 * CA2)
   +   M3 * ST2**2 *
       ( -   A2**2 * SA1**2   -   A3**2 * SA1**2   -   2 * A3 * X3 * SA1**2   +   S3**2 * SA1**2 * SA2**2   +

   2 * S3 * Y3 * SA1**2 * SA2**2 * SA3   +   2 * S3 * Z3 * SA1**2 * SA2**2 * CA3   +   K311 * SA1**2   -   2 * K
322 * SA1**2 * SA2**2 * SA3**2   +   K322 * SA1**2 * SA2**2   +   K322 * SA1**2 * SA3**2   -   K322 * SA1**2   +
  2 * K333 * SA1**2 * SA2**2 * SA3**2   -   K333 * SA1**2 * SA2**2   -   K333 * SA1**2 * SA3**2   +   4 * K323
* SA1**2 * SA2**2 * SA3 * CA3   -   2 * K323 * SA1**2 * SA3 * CA3)
   +   M3 * ST2 * CT2 * ST3**2 *
       (4 * A3 * Y3 * SA1**2 * CA2 * CA3   -   4 * A3 * Z3 * SA1**2 * CA2 * SA3   +   4 * K312 * SA1**2 * CA2 *
 CA3   -   4 * K313 * SA1**2 * CA2 * SA3)
   +   M3 * ST2 * CT2 * ST3 * CT3 *
       ( -   2 * A3**2 * SA1**2 * CA2   -   4 * A3 * X3 * SA1**2 * CA2   +   2 * K311 * SA1**2 * CA2   +   2
* K322 * SA1**2 * CA2 * SA3**2   -   2 * K322 * SA1**2 * CA2   -   2 * K333 * SA1**2 * CA2 * SA3**2   -   4 * K3
23 * SA1**2 * CA2 * SA3 * CA3)
   +   M3 * ST2 * CT2 * ST3 *
       ( -   2 * A2 * A3 * SA1**2 * CA2   -   2 * A2 * X3 * SA1**2 * CA2   -   2 * S3 * Y3 * SA1**2 * SA2 * C
A3   +   2 * S3 * Z3 * SA1**2 * SA2 * SA3   +   2 * K322 * SA1**2 * SA2 * SA3 * CA3   -   2 * K333 * SA1**2 * SA
2 * SA3 * CA3   +   4 * K323 * SA1**2 * SA2 * SA3**2   -   2 * K323 * SA1**2 * SA2)
   +   M3 * ST2 * CT2 * CT3 *
       ( -   2 * A2 * Y3 * SA1**2 * CA2 * CA3   +   2 * A2 * Z3 * SA1**2 * CA2 * SA3   +   2 * A3 * S3 * SA1*
*2 * SA2   +   2 * A3 * Y3 * SA1**2 * SA2 * SA3   +   2 * A3 * Z3 * SA1**2 * SA2 * CA3   +   2 * S3 * X3 * SA1**
2 * SA2   +   2 * K312 * SA1**2 * SA2 * SA3   +   2 * K313 * SA1**2 * SA2 * CA3)
   +   M3 * ST2 * CT2 *
       (2 * A2 * S3 * SA1**2 * SA2   +   2 * A2 * Y3 * SA1**2 * SA2 * SA3   +   2 * A2 * Z3 * SA1**2 * SA2 * CA
3   -   2 * A3 * Y3 * SA1**2 * CA2 * CA3   +   2 * A3 * Z3 * SA1**2 * CA2 * SA3   -   2 * K312 * SA1**2 * CA2 *
CA3   +   2 * K313 * SA1**2 * CA2 * SA3)
   +   M3 * ST2 * ST3**2 *
       (4 * A3 * Y3 * SA1 * CA1 * SA2 * CA3   -   4 * A3 * Z3 * SA1 * CA1 * SA2 * SA3   +   4 * K312 * SA1 * CA
1 * SA2 * CA3   -   4 * K313 * SA1 * CA1 * SA2 * SA3)
   +   M3 * ST2 * ST3 * CT3 *
       ( -   2 * A3**2 * SA1 * CA1 * SA2   -   4 * A3 * X3 * SA1 * CA1 * SA2   +   2 * K311 * SA1 * CA1 * SA2
 +   2 * K322 * SA1 * CA1 * SA2 * SA3**2   -   2 * K322 * SA1 * CA1 * SA2   -   2 * K333 * SA1 * CA1 * SA2 * S
A3**2   -   4 * K323 * SA1 * CA1 * SA2 * SA3 * CA3)
   +   M3 * ST2 * ST3 *
       ( -   2 * A1 * A3 * CA2   -   2 * A1 * X3 * CA2   -   2 * A2 * A3 * SA1 * CA1 * SA2   -   2 * A2 * X3
* SA1 * CA1 * SA2   -   2 * S2 * Y3 * SA1 * CA1 * CA3   -   2 * S2 * Z3 * SA1 * CA1 * SA3   +   2 * S3 * Y3 * SA
1 * CA1 * CA2 * CA3   -   2 * S3 * Z3 * SA1 * CA1 * CA2 * SA3   -   2 * K322 * SA1 * CA1 * CA2 * SA3 * CA3   +
 2 * K333 * SA1 * CA1 * CA2 * SA3 * CA3   -   4 * K323 * SA1 * CA1 * CA2 * SA3**2   +   2 * K323 * SA1 * CA1 * C
A2)
   +   M3 * ST2 * CT3 *
       ( -   2 * A1 * Y3 * CA2 * CA3   +   2 * A1 * Z3 * CA2 * SA3   -   2 * A2 * Y3 * SA1 * CA1 * SA2 * CA3
 +   2 * A2 * Z3 * SA1 * CA1 * SA2 * SA3   -   2 * A3 * S2 * SA1 * CA1   -   2 * A3 * S3 * SA1 * CA1 * CA2   -
 2 * A3 * Y3 * SA1 * CA1 * CA2 * SA3   -   2 * A3 * Z3 * SA1 * CA1 * CA2 * CA3   -   2 * S2 * X3 * SA1 * CA1
-   2 * S3 * X3 * SA1 * CA1 * CA2   -   2 * K312 * SA1 * CA1 * CA2 * SA3   -   2 * K313 * SA1 * CA1 * CA2 * CA3)
   +   M3 * ST2 *
       (2 * A1 * S3 * SA2   +   2 * A1 * Y3 * SA2 * SA3   +   2 * A1 * Z3 * SA2 * CA3   -   2 * A2 * S2 * SA1 *
 CA1   -   2 * A2 * S3 * SA1 * CA1 * CA2   -   2 * A2 * Y3 * SA1 * CA1 * CA2 * SA3   -   2 * A2 * Z3 * SA1 * CA1
```

```
 * CA2 * CA3   -   2 * A3 * Y3 * SA1 * CA1 * SA2 * CA3   +   2 * A3 * Z3 * SA1 * CA1 * SA2 * SA3   -   2 * K312
* SA1 * CA1 * SA2 * CA3   +   2 * K313 * SA1 * CA1 * SA2 * SA3)

  +   H3 * CT2 * ST3**2 *
      (  -   2 * A3**2 * SA1 * CA1 * SA2 * CA2   -   4 * A3 * X3 * SA1 * CA1 * SA2 * CA2   +   2 * K311 * SA1
 * CA1 * SA2 * CA2   -   2 * K322 * SA1 * CA1 * SA2 * CA2 * SA3**2   -   2 * K322 * SA1 * CA1 * SA2 * CA2   -
2 * K333 * SA1 * CA1 * SA2 * CA2 * SA3**2   -   4 * K323 * SA1 * CA1 * SA2 * CA2 * SA3 * CA3)

  +   H3 * CT2 * ST3 * CT3 *
      (  -   4 * A3 * Y3 * SA1 * CA1 * SA2 * CA2 * CA3   +   4 * A3 * Z3 * SA1 * CA1 * SA2 * CA2 * SA3   -
4 * K312 * SA1 * CA1 * SA2 * CA2 * CA3   +   4 * K313 * SA1 * CA1 * SA2 * CA2 * SA3)

  +   H3 * CT2 * ST3 *
      (  -   2 * A1 * Y3 * CA3   -   2 * A1 * Z3 * SA3   -   2 * A3 * S2 * SA1 * CA1 * CA2   +   4 * A3 * S3
* SA1 * CA1 * SA2**2   -   2 * A3 * S3 * SA1 * CA1   +   4 * A3 * Y3 * SA1 * CA1 * SA2**2 * SA3   -   2 * A3 * Y
3 * SA1 * CA1 * SA3   +   4 * A3 * Z3 * SA1 * CA1 * SA2**2 * CA3   -   2 * A3 * Z3 * SA1 * CA1 * CA3   -   2 * S
2 * X3 * SA1 * CA1 * CA2   +   4 * S3 * X3 * SA1 * CA1 * SA2**2   -   2 * S3 * X3 * SA1 * CA1   +   4 * K312 * S
A1 * CA1 * SA2**2 * SA3   -   2 * K312 * SA1 * CA1 * SA3   +   4 * K313 * SA1 * CA1 * SA2**2 * CA3   -   2 * K31
3 * SA1 * CA1 * CA3)

  +   H3 * CT2 * CT3 *
      (2 * A1 * A3   +   2 * A1 * X3   -   2 * S2 * Y3 * SA1 * CA1 * CA2 * CA3   +   2 * S2 * Z3 * SA1 * CA1 *
CA2 * SA3   +   4 * S3 * Y3 * SA1 * CA1 * SA2**2 * CA3   -   2 * S3 * Y3 * SA1 * CA1 * CA3   -   4 * S3 * Z3 *
SA1 * CA1 * SA2**2 * SA3   +   2 * S3 * Z3 * SA1 * CA1 * SA3   -   4 * K322 * SA1 * CA1 * SA2**2 * SA3 * CA3   +
  2 * K322 * SA1 * CA1 * SA3 * CA3   +   4 * K333 * SA1 * CA1 * SA2**2 * SA3 * CA3   -   2 * K333 * SA1 * CA1 *
SA3 * CA3   -   8 * K323 * SA1 * CA1 * SA2**2 * SA3**2   +   4 * K323 * SA1 * CA1 * SA2**2   +   4 * K323 * SA1
 * CA1 * SA3**2   -   2 * K323 * SA1 * CA1)

  +   H3 * CT2 *
      (2 * A1 * A2   +   2 * S2 * S3 * SA1 * CA1 * SA2   +   2 * S2 * Y3 * SA1 * CA1 * SA2 * SA3   +   2 * S2
* Z3 * SA1 * CA1 * SA2 * CA3   +   2 * S3**2 * SA1 * CA1 * SA2 * CA2   +   4 * S3 * Y3 * SA1 * CA1 * SA2 * CA2 *
SA3   +   4 * S3 * Z3 * SA1 * CA1 * SA2 * CA2 * CA3   -   4 * K322 * SA1 * CA1 * SA2 * CA2 * SA3**2   +   2 * K
322 * SA1 * CA1 * SA2 * CA2   +   4 * K333 * SA1 * CA1 * SA2 * CA2 * SA3**2   -   2 * K333 * SA1 * CA1 * SA2 * C
A2   +   8 * K323 * SA1 * CA1 * SA2 * CA2 * SA3 * CA3)

  +   H3 * ST3**2 *
      (2 * A3**2 * SA1**2 * SA2**2   -   A3**2 * SA1**2   -   A3**2 * SA2**2   +   4 * A3 * X3 * SA1**2 * SA2*
*2   -   2 * A3 * X3 * SA1**2   -   2 * A3 * X3 * SA2**2   -   2 * K311 * SA1**2 * SA2**2   +   K311 * SA1**2
+   K311 * SA2**2   -   2 * K322 * SA1**2 * SA2**2 * SA3**2   +   2 * K322 * SA1**2 * SA2**2   +   K322 * SA1**2
 * SA3**2   -   K322 * SA1**2   +   K322 * SA2**2 * SA3**2   -   K322 * SA2**2   +   2 * K333 * SA1**2 * SA2**2
* SA3**2   -   K333 * SA1**2 * SA3**2   -   K333 * SA2**2 * SA3**2   +   4 * K323 * SA1**2 * SA2**2 * SA3 * CA3
  -   2 * K323 * SA1**2 * SA3 * CA3   -   2 * K323 * SA2**2 * SA3 * CA3)

  +   H3 * ST3 * CT3 *
      (4 * A3 * Y3 * SA1**2 * SA2**2 * CA3   -   2 * A3 * Y3 * SA1**2 * CA3   -   2 * A3 * Y3 * SA2**2 * CA3
  -   4 * A3 * Z3 * SA1**2 * SA2**2 * SA3   +   2 * A3 * Z3 * SA1**2 * SA3   +   2 * A3 * Z3 * SA2**2 * SA3   +
 4 * K312 * SA1**2 * SA2**2 * CA3   -   2 * K312 * SA1**2 * CA3   -   2 * K312 * SA2**2 * CA3   -   4 * K313 * S
A1**2 * SA2**2 * SA3   +   2 * K313 * SA1**2 * SA3   +   2 * K313 * SA2**2 * SA3)

  +   H3 * ST3 *
      (  -   2 * A2 * Y3 * CA3   +   2 * A2 * Z3 * SA3   +   2 * A3 * S2 * SA1**2 * SA2   -   4 * A3 * S3 * S
A1**2 * SA2 * CA2   -   2 * A3 * S3 * SA2 * CA2   +   4 * A3 * Y3 * SA1**2 * SA2 * CA2 * SA3   -   2 * A3 * Y3 *
 SA2 * CA2 * SA3   +   4 * A3 * Z3 * SA1**2 * SA2 * CA2 * CA3   -   2 * A3 * Z3 * SA2 * CA2 * CA3   -   2 * S2 *
 X3 * SA1**2 * SA2   -   4 * S3 * X3 * SA1**2 * SA2 * CA2   -   2 * S3 * X3 * SA2 * CA2   +   4 * K312 * SA1**2


* SA2 * CA2 * SA3   -   2 * K312 * SA2 * CA2 * SA3   +   4 * K313 * SA1**2 * SA2 * CA2 * CA3   -   2 * K313 * SA
2 * CA2 * CA3)

  +   H3 * CT3 *
      (2 * A2 * A3   +   2 * A2 * X3   +   2 * S2 * Y3 * SA1**2 * SA2 * CA3   -   2 * S2 * Z3 * SA1**2 * SA2 *
SA3   +   4 * S3 * Y3 * SA1**2 * SA2 * CA2 * CA3   -   2 * S3 * Y3 * SA2 * CA2 * CA3   -   4 * S3 * Z3 * SA1**2
* SA2 * CA2 * SA3   +   2 * S3 * Z3 * SA2 * CA2 * SA3   -   4 * K322 * SA1**2 * SA2 * CA2 * SA3 * CA3   +   2 *
K322 * SA2 * CA2 * SA3 * CA3   +   4 * K333 * SA1**2 * SA2 * CA2 * SA3 * CA3   -   2 * K333 * SA2 * CA2 * SA3 *
CA3   -   8 * K323 * SA1**2 * SA2 * CA2 * SA3**2   +   4 * K323 * SA1**2 * SA2 * CA2   +   4 * K323 * SA2 * CA2
* SA3**2   -   2 * K323 * SA2 * CA2)

  +   H3 *
      (A1**2   +   A2**2   +   A3**2   +   2 * A3 * X3   +   S2**2 * SA1**2   +   2 * S2 * S3 * SA1**2 * CA2
 +   2 * S2 * Y3 * SA1**2 * CA2 * SA3   +   2 * S2 * Z3 * SA1**2 * CA2 * CA3   -   2 * S3**2 * SA1**2 * SA2**2
 +   S3**2 * SA1**2   +   S3**2 * SA2**2   -   4 * S3 * Y3 * SA1**2 * SA2**2 * SA3   +   2 * S3 * Y3 * SA1**2 *
SA3   +   2 * S3 * Y3 * SA2**2 * SA3   -   4 * S3 * Z3 * SA1**2 * SA2**2 * CA3   +   2 * S3 * Z3 * SA1**2 * CA3
 +   2 * S3 * Z3 * SA2**2 * CA3   +   4 * K322 * SA1**2 * SA2**2 * SA3**2   -   2 * K322 * SA1**2 * SA2**2   -
 2 * K322 * SA1**2 * SA3**2   +   K322 * SA1**2   -   2 * K322 * SA2**2 * SA3**2   +   K322 * SA2**2   +   K322
 * SA3**2   -   4 * K333 * SA1**2 * SA2**2 * SA3**2   +   2 * K333 * SA1**2 * SA2**2   +   2 * K333 * SA1**2 * S
A3**2   -   K333 * SA1**2   +   2 * K333 * SA2**2 * SA3**2   -   K333 * SA2**2   -   K333 * SA3**2   +   K333
-   8 * K323 * SA1**2 * SA2**2 * SA3 * CA3   +   4 * K323 * SA1**2 * SA3 * CA3   +   4 * K323 * SA2**2 * SA3 * C
A3   -   2 * K323 * SA3 * CA3)

##########################################################################################################

H12   =


      H2 * ST2 *
      (  -   A1 * Y2 * CA1 * CA2   +   A1 * Z2 * CA1 * SA2   -   A2 * S2 * SA1   -   A2 * Y2 * SA1 * SA2   -
  A2 * Z2 * SA1 * CA2   -   S2 * X2 * SA1   -   SA1 * SA2 * K212   -   SA1 * CA2 * K213)

  +   H2 * CT2 *
      (A1 * A2 * CA1   +   A1 * X2 * CA1   -   S2 * Y2 * SA1 * CA2   +   S2 * Z2 * SA1 * SA2   +   2 * SA1 * S
A2**2 * K223   +   SA1 * SA2 * CA2 * K222   -   SA1 * SA2 * CA2 * K233   -   SA1 * K223)

  +   H2 *
      (A2**2 * CA1   +   2 * A2 * X2 * CA1   +   CA1 * SA2**2 * K222   -   CA1 * SA2**2 * K233   -   2 * CA1 *
 SA2 * CA2 * K223   +   CA1 * K233)

  +   H3 * ST2 * ST3**2 *
      (2 * A3 * Y3 * SA1 * SA2 * CA3   -   2 * A3 * Z3 * SA1 * SA2 * SA3   +   2 * K312 * SA1 * SA2 * CA3   -
  2 * K313 * SA1 * SA2 * SA3)

  +   H3 * ST2 * ST3 * CT3 *
      (  -   A3**2 * SA1 * SA2   -   2 * A3 * X3 * SA1 * SA2   +   K311 * SA1 * SA2   +   K322 * SA1 * SA2 *
SA1**2   -   K322 * SA1 * SA2   -   K333 * SA1 * SA2 * SA3**2   -   2 * K323 * SA1 * SA2 * SA3 * CA3)

  +   H3 * ST2 * ST3 *
      (  -   A1 * A3 * CA1 * CA2   -   A1 * X3 * CA1 * CA2   -   A2 * A3 * SA1 * SA2   -   A2 * X3 * SA1 * SA
2   +   S2 * Y3 * SA1 * CA3   -   S2 * Z3 * SA1 * SA3   +   S3 * Y3 * SA1 * CA2 * CA3   -   S3 * Z3 * SA1 * CA2
* SA3   -   K322 * SA1 * CA2 * SA3 * CA3   +   K333 * SA1 * CA2 * SA3 * CA3   -   2 * K323 * SA1 * CA2 * SA3**2
 +   K323 * SA1 * CA2)
```

```
   +   M3 * ST2 * CT3 *
        (   -   A1 * Y3 * CA1 * CA2 * CA3   +   A1 * Z3 * CA1 * CA2 * SA3   -   A2 * Y3 * SA1 * SA2 * CA3   +
A2 * Z3 * SA1 * SA2 * SA3   -   A3 * S2 * SA1   -   A3 * S3 * SA1 * CA2   -   A3 * Y3 * SA1 * CA2 * SA3   -   A3
 * Z3 * SA1 * CA2 * CA3   -   S2 * X3 * SA1   -   S3 * X3 * SA1 * CA2   -   K312 * SA1 * CA2 * SA3   -   K313 *
SA1 * CA2 * CA3)

   +   M3 * ST2 *
        (A1 * S3 * CA1 * SA2   +   A1 * Y3 * CA1 * SA2 * SA3   +   A1 * Z3 * CA1 * SA2 * CA3   -   A2 * S2 * SA1
   -   A2 * S3 * SA1 * CA2   -   A2 * Y3 * SA1 * CA2 * SA3   -   A2 * Z3 * SA1 * CA2 * CA3   -   A3 * Y3 * SA1 *
 SA2 * CA3   +   A3 * Z3 * SA1 * SA2 * SA3   -   K312 * SA1 * SA2 * CA3   +   K313 * SA1 * SA2 * SA3)

   +   M3 * CT2 * ST3**2 *
        (   -   A3**2 * SA1 * SA2 * CA2   -   2 * A3 * X3 * SA1 * SA2 * CA2   +   K311 * SA1 * SA2 * CA2   -   K
322 * SA1 * SA2 * CA2 * SA3**2   -   K322 * SA1 * SA2 * CA2   -   K333 * SA1 * SA2 * CA2 * SA3**2   -   2 * K323
 * SA1 * SA2 * CA2 * SA3 * CA3)

   +   M3 * CT2 * ST3 * CT3 *
        (   -   2 * A3 * Y3 * SA1 * SA2 * CA2 * CA3   +   2 * A3 * Z3 * SA1 * SA2 * CA2 * SA3   -   2 * K312 * S
A1 * SA2 * CA2 * CA3   +   2 * K313 * SA1 * SA2 * CA2 * SA3)

   +   M3 * CT2 * ST3 *
        (   -   A1 * Y3 * CA1 * CA3   +   A1 * Z3 * CA1 * SA3   -   A3 * S2 * SA1 * CA2   +   2 * A3 * S3 * SA1
 * SA2**2   -   A3 * S3 * SA1   +   2 * A3 * Y3 * SA1 * SA2**2 * SA3   -   A3 * Y3 * SA1 * SA3   +   2 * A3 * Z3
 * SA1 * SA2**2 * CA3   -   A3 * Z3 * SA1 * CA3   -   S2 * X3 * SA1 * CA2   +   2 * S3 * X3 * SA1 * SA2**2   -
S3 * X3 * SA1   +   2 * K312 * SA1 * SA2**2 * SA3   -   K312 * SA1 * SA3   +   2 * K313 * SA1 * SA2**2 * CA3   -
   K313 * SA1 * CA3)

   +   M3 * CT2 * CT3 *
        (A1 * A3 * CA1   +   A1 * X3 * CA1   -   S2 * Y3 * SA1 * CA2 * CA3   +   S2 * Z3 * SA1 * CA2 * SA3   +
 2 * S3 * Y3 * SA1 * SA2**2 * CA3   -   S3 * Y3 * SA1 * CA3   -   2 * S3 * Z3 * SA1 * SA2**2 * SA3   +   S3 * Z3
 * SA1 * SA3   -   2 * K322 * SA1 * SA2**2 * SA3 * CA3   +   K322 * SA1 * SA3 * CA3   +   2 * K333 * SA1 * SA2**
2 * SA3 * CA3   -   K333 * SA1 * SA3 * CA3   -   4 * K323 * SA1 * SA2**2 * SA3**2   +   2 * K323 * SA1 * SA2**2
  + . 2 * K323 * SA1 * SA3**2   -   K323 * SA1)

   +   M3 * CT2 *
        (A1 * A2 * CA1   +   S2 * S3 * SA1 * SA2   +   S2 * Y3 * SA1 * SA2 * SA3   +   S2 * Z3 * SA1 * SA2 * CA3
   +   S3**2 * SA1 * SA2 * CA2   +   2 * S3 * Y3 * SA1 * SA2 * CA2 * SA3   +   2 * S3 * Z3 * SA1 * SA2 * CA2 * C
A3   -   2 * K322 * SA1 * SA2 * CA2 * SA3**2   +   K322 * SA1 * SA2 * CA2   +   2 * K333 * SA1 * SA2 * CA2 * SA3
**2   -   K333 * SA1 * SA2 * CA2   +   4 * K323 * SA1 * SA2 * CA2 * SA3 * CA3)

   +   M3 * ST3**2 *
        (   -   A3**2 * CA1 * SA2**2   -   2 * A3 * X3 * CA1 * SA2**2   +   K311 * CA1 * SA2**2   -   K322 * CA1
 * SA2**2 * SA3**2   -   K322 * CA1 * SA2**2   -   K333 * CA1 * SA2**2 * SA3**2   -   2 * K323 * CA1 * SA2**2 *
SA3 * CA3)

   +   M3 * ST3 * CT3 *
        (   -   2 * A3 * Y3 * CA1 * SA2**2 * CA3   +   2 * A3 * Z3 * CA1 * SA2**2 * SA3   -   2 * K312 * CA1 * S
A2**2 * CA3   +   2 * K313 * CA1 * SA2**2 * SA3)

   +   M3 * ST3 *
        (   -   2 * A2 * Y3 * CA1 * CA3   +   2 * A2 * Z3 * CA1 * SA3   -   2 * A3 * S3 * CA1 * SA2 * CA2   -
2 * A3 * Y3 * CA1 * SA2 * CA2 * SA3   -   2 * A3 * Z3 * CA1 * SA2 * CA2 * CA3   -   2 * S3 * X3 * CA1 * SA2 * CA
2   -   2 * K312 * CA1 * SA2 * CA2 * SA3   -   2 * K313 * CA1 * SA2 * CA2 * CA3)


   +   M3 * CT3 *
        (2 * A2 * A3 * CA1   +   2 * A2 * X3 * CA1   -   2 * S3 * Y3 * CA1 * SA2 * CA2 * CA3   +   2 * S3 * Z3 *
 CA1 * SA2 * CA2 * SA3   +   2 * K322 * CA1 * SA2 * CA2 * SA3 * CA3   -   2 * K333 * CA1 * SA2 * CA2 * SA3 * CA3
   +   4 * K323 * CA1 * SA2 * CA2 * SA3**2   -   2 * K323 * CA1 * SA2 * CA2)

   +   M3 *
        (A2**2 * CA1   +   A3**2 * CA1   +   2 * A3 * X3 * CA1   +   S3**2 * CA1 * SA2**2   +   2 * S3 * Y3 * CA
1 * SA2**2 * SA3   +   2 * S3 * Z3 * CA1 * SA2**2 * CA3   -   2 * K322 * CA1 * SA2**2 * SA3**2   +   K322 * CA1
 * SA2**2   +   K322 * CA1 * SA3**2   +   2 * K333 * CA1 * SA2**2 * SA3**2   -   K333 * CA1 * SA2**2   -   K333 *
 CA1 * SA3**2   +   K333 * CA1   +   4 * K323 * CA1 * SA2**2 * SA3 * CA3   -   2 * K323 * CA1 * SA3 * CA3)
##################################################################################################

M13   =


        M3 * ST2 * ST3 *
        (   -   A1 * A3 * CA1   -   A1 * X3 * CA1   +   S2 * Y3 * SA1 * CA2 * CA3   -   S2 * Z3 * SA1 * CA2 * SA
3   +   S3 * Y3 * SA1 * CA3   -   S3 * Z3 * SA1 * SA3   +   K322 * SA1 * SA3 * CA3   -   K333 * SA1 * SA3 * CA3
   -   2 * K323 * SA1 * SA3**2   +   K323 * SA1)

   +   M3 * ST2 * CT3 *
        (   -   A1 * Y3 * CA1 * CA3   +   A1 * Z3 * CA1 * SA3   -   A3 * S2 * SA1 * CA2   -   A3 * S3 * SA1   -
  A3 * Y3 * SA1 * SA3   -   A3 * Z3 * SA1 * CA3   -   S2 * X3 * SA1 * CA2   -   S3 * X3 * SA1   -   K312 * SA1 *
 SA3   -   K313 * SA1 * CA3)

   +   M3 * CT2 * ST3 *
        (   -.  A1 * Y3 * CA1 * CA2 * CA3   +   A1 * Z3 * CA1 * CA2 * SA3   +   A2 * Y3 * SA1 * SA2 * CA3   -
A2 * Z3 * SA1 * SA2 * SA3   -   A3 * S2 * SA1   -   A3 * S3 * SA1 * CA2   -   A3 * Y3 * SA1 * CA2 * SA3   -   A3
 * Z3 * SA1 * CA2 * CA3   -   S2 * X3 * SA1   -   S3 * X3 * SA1 * CA2   -   K312 * SA1 * CA2 * SA3   -   K313 *
SA1 * CA2 * CA3)

   +   M3 * CT2 * CT3 *
        (A1 * A3 * CA1 * CA2   +   A1 * X3 * CA1 * CA2   -   A2 * A3 * SA1 * SA2   -   A2 * X3 * SA1 * SA2   -
 S2 * Y3 * SA1 * CA3   +   S2 * Z3 * SA1 * SA3   -   S3 * Y3 * SA1 * CA2 * CA3   +   S3 * Z3 * SA1 * CA2 * SA3
 +   K322 * SA1 * CA2 * SA3 * CA3   -   K333 * SA1 * CA2 * SA3 * CA3   +   2 * K323 * SA1 * CA2 * SA3**2   -   K
323 * SA1 * CA2)

   +   M3 * CT2 *
        (   -   A3**2 * SA1 * SA2   -   2 * A3 * X3 * SA1 * SA2   -   K322 * SA1 * SA2 * SA3**2   +   K333 * SA1
 * SA2 * SA3**2   -   K333 * SA1 * SA2   +   2 * K323 * SA1 * SA2 * SA3 * CA3)

   +   M3 * ST3 *
        (A1 * Y3 * SA1 * SA2 * CA3   -   A1 * Z3 * SA1 * SA2 * SA3   -   A2 * Y3 * CA1 * CA2 * CA3   +   A2 * Z3
 * CA1 * CA2 * SA3   -   A3 * S3 * CA1 * SA2   -   A3 * Y3 * CA1 * SA2 * SA3   -   A3 * Z3 * CA1 * SA2 * CA3   -
   S3 * X3 * CA1 * SA2   -   K312 * CA1 * SA2 * SA3   -   K313 * CA1 * SA2 * CA3)

   +   M3 * CT3 *
        (   -   A1 * A3 * SA1 * SA2   -   A1 * X3 * SA1 * SA2   +   A2 * A3 * CA1 * CA2   +   A2 * X3 * CA1 * CA
2   -   S3 * Y3 * CA1 * SA2 * CA3   +   S3 * Z3 * CA1 * SA2 * SA3   +   K322 * CA1 * SA2 * SA3 * CA3   -   K333
 * CA1 * SA2 * SA3 * CA3   +   2 * K323 * CA1 * SA2 * SA3**2   -   K323 * CA1 * SA2)

   +   M3 *
        (A3**2 * CA1 * CA2   +   2 * A3 * X3 * CA1 * CA2   +   K322 * CA1 * CA2 * SA3**2   -   K333 * CA1 * CA2
```

References

1. Arbib, M. A., The Metaphorical Brain (Wiley & Sons, Inc., N.Y., 1972). pp. 148-151.

2. Bilodeau, E. A., (ed.), Acquisition of Skill (Academic Press, N. Y., 1966).

3. Conolly, K., (ed.) Mechanisms of Motor Skill Development (Academic Press, London, 1970).

4. Cratty. B. J., Movement Behavior and Motor Learning (Lea & Febiger, Philadelphia, 1964).

5. Evarts, E. V., Bizzi, E., Burke, R. E., Delong, M., and Thach, W. T., Jr., "Central Control of Movement", Neuroscience Research Program Bulletin, Vol. 9, No. 1, 1970.

6. Freedy, A., Hull, F. C., Lucaccini, L. F., and Lyman, J., "A Computer Based Learning System for Remote Manipulator Control", IEEE trans. on Systems, Man, and Cybernetics, Vol. SMC-1, No. 4, 1971, OO. 356-363.

7. Fu, K. S., "Learning Control Systems and Intelligent Control Systems: An Intersection of Artificial Intelligence and Automatic Control", IEEE trans. on Automatic Control, Feb., 1971, pp. 70-72.

8. Gelfand, I. M., Gurfinkel, V. S., Tsetlin, M. L., and Shik, M. L. "Some Problems in the Analysis of Movement." In: Models of the Structural-Functional Organization of Certain Biological Systems. Gelfand, I. M., (ed.) (M.I.T. Press, Cambridge, Mass., 1971) p. 335.

9. Hammond, P. H. "The Influence of Prior Instruction to the Subject on an Apparently Involuntary Neuro-muscular Response." J. Physiol., 127, 1956, pp. 17-18.

10. Hein, A., and Held, R., "A Neural Model for Labile Sensorimotor Coordinations." In: Biological Prototypes and Synthetic Systems. Bernard, E., and Kare, M. (eds.) Vol. 1, (Plenum Press, New York, 1962) pp. 71-74.

11. Held, R., "Exposure-History as a Factor in Maintaining Stability of Perception and Coordination.", J. Nervous & Mental Diseases, 1961, 132, pp. 26-32.

12. Held, R., and Bauer, J. A., Jr., "Development of Sensorially-Guided Reaching in Infant Monkeys.", Brain Research, 1974, 71, pp. 265-271.

13. Hilgard, E. R. and Bower, G. H. Theories of Learning (Prentice-Hall, Inc., New Jersey, 1975), pp. 37-38.

14. Horn, B. K. P. and Inoue, H., "Kinematics of the M.I.T.-AI-Vicarm Manipulator", M.I.T. Working Paper 69, May 1974.

15. Kahn, M. E., "The Near-Minimum Time Control of Open-Loop Articulated Kinematic Chains" Ph. D. Dissertation, Stanford University, Computer Science Dept., Dec., 1969, pp. 145-161.

16. Kelley, D., Manual and Automatic Control (J. Wiley & Sons, Inc., New York, 1968).

17. Levine, W. S., of University of Maryland, Department of Electrical Engineering, Personal Communication of 9/30/75.

18. Marr, D., "A Theory of Cerebellar Cortex", J. Physiol., 1969, 202, p. 468.

19. McGeoch, J. A., The Psychology of Human Learning, 2nd Ed. Revised by Irion, A. L. (David McKay Co., Inc., New York, 1952). pp. 299-354.

20. Melvill-Jones, G. and Watt, D. O. D., "Observations on the Control of Stepping and Hopping Movements in Man." J. Physiol., 1971, 219, pp. 709-727.

21. Meriam, F. L., Dynamics (John Wiley & Sons, Inc., New York, 1966).

22. Mittelstaedt, H. "The Analysis of Behavior in Terms of Control Systems". In: Group Processes, Transactions of the Fifth Conference, Princeton, New Jersey, Schaffner, B., (ed.) pp. 45-84, 1958.

23. Raibert, M. H., Doctoral Thesis Proposal, In preparation.

24. von Holst, E., "Relations between the Central Nervous System and the Peripheral Organs", Brit. J. Animal Behavior, 1954, 2, pp. 89-94.

25. Waters, R. C., "A Mechanical Arm Control System", Artificial Intelligence Memo No. 301, Jan., 1974, M.I.T.

26. Whitney, D. E., "State Space Models of Remote Manipulation Tasks" Ph. D. Thesis, M.I.T. Department of Mechanical Engineering, Jan., 1968.

27. Young, L. R., "On Adaptive Manual Control", Ergonomics, Vol. 12, 1969, No. 4, p. 661.

28. Young, L. R. and Stark, L., "Biological Control Systems -- A Critical Review and Evaluation: Developments in Manual Control", NASA, CR-190, 1965.