To appear in: M. Gazzaniga, (Ed.). *The Cognitive Neurosciences.* Cambridge, MA: MIT Press.

Computational motor control

Michael I. Jordan

Department of Brain and Cognitive Sciences

Massachusetts Institute of Technology

Daniel M. Wolpert

Sobell Department of Neurophysiology

Institute of Neurology

University College London

Running Title: Computational motor control

Abstract

We discuss some of the computational approaches that have been developed in the area of motor control. We focus on problems relating to motor planning, internal models, state estimation, motor learning and modularity. The aim of the chapter is to demonstrate, both at a conceptual level and through consideration of specific models, how computational approaches shed light on problems in the control of movement.

Introduction

The study of motor control is fundamentally the study of sensorimotor transformations. For the motor control system to move its effectors to apply forces on objects in the world or to position its sensors with respect to objects in the world, it must coordinate a variety of forms of sensory and motor data. These data are generally in different formats and may refer to the same entities but in different coordinate systems. Transformations between these coordinate systems allow motor and sensory data to be related, closing the sensorimotor loop. Equally fundamental is the fact that the motor control system operates with dynamical systems, whose behavior depends on the way energy is stored and transformed. The study of motor control is therefore also the study of dynamics. These two interrelated issues—sensorimotor transformations and dynamics—underly much of the research in the area of motor control.

From a computational perspective the motor system can be considered as a system whose *inputs* are the motor commands emanating from the controller within the central nervous system (CNS) (Figure 1 center). In order to determine the behavior of the system in response to this input an additional set of variables, called *state* variables, must also be known. For example, in a robotic model of the arm the motor command would represent the torques generated around the joints and the state variables would be the joint angles and angular velocities. Taken together, the inputs and the state variables are sufficient to determine the future behavior of the system. It is unrealistic, however, to assume that the controller in the CNS has direct access to the state of the system that it is controlling; rather, we generally assume that the controller has access to a sensory feedback signal that is a function of the state. This signal is treated as the *output* of the abstract computational system.

Insert Figure 1 about here

In this chapter we will consider five issues which arise in considering the general computational schema in Figure 1. The first issue is that of *motor planning*, which we consider to be the computational process by which the desired outputs of the system are specified given an extrinsic task goal. Second, we explore the notion of *internal models*, which are systems which mimic—within the CNS—the behavior of the controlled system. Such models have a variety of roles in motor control that we will elucidate. Third, we consider the problem of *state estimation*, which is the process by which the unknown state of the motor system can be estimated by monitoring both its inputs and outputs. This process requires the CNS to integrate its internal state estimates, obtained via an internal model, with the sensory feedback. We then consider how internal models are refined through *motor learning*. Finally we discuss how *multiple* internal models can be used in motor control.

While many of the concepts discussed are applicable to all areas of motor control, including eye movements, speech production and posture, we will focus on arm movements as an illustrative system.

Motor Planning

The computational problem of motor planning arises from a fundamental property of the motor system; the reduction in the degrees of freedom from neural commands through muscle activation to movement kinematics (Bernstein, 1967) (Figure 2). Even for the simplest of tasks, such as moving the hand to a target location, there are an infinite number of possible paths that the hand could move along and for each of these paths there are an infinite number of velocity profiles (trajectories) the hand could follow. Having specified the hand path and velocity, each location of the hand along the path can be achieved by multiple combinations of joint angles and, due to the overlapping actions of muscles and the ability to co-contract, each arm configuration can be achieved by many different muscle activations. Motor planning can be considered as the computational process of selecting a single solution or pattern of behavior at the levels in the motor hierarchy (Figure 2), from the many alternatives which are consistent with the task.

Insert Figure 2 about here

Given the redundancy in the motor system it is illuminating that experimental observations of unconstrained point-to-point reaching movements have demonstrated that several aspects of movements tend to remain invariant, despite variations in movement direction, movement speed, and movement location (Morasso, 1981; Flash and Hogan, 1985). First, as shown in Figure 3a, the motion of the hand tends to follow roughly a straight line in space. This observation is not uniformly true; significant curvature is observed for certain movements, particularly horizontal movements and movements near the boundaries of the workspace (Atkeson and Hollerbach, 1985; Soechting and Lacquaniti, 1981; Uno et al., 1989). The tendency to make straight line movements, however, characterizes a reasonably large class of movements and is somewhat surprising given that the muscles act in rotating coordinate systems attached to the joints. Second, the movement of the hand is smooth. Higher derivatives of the hand motion, such as the velocity and acceleration, tend to vary smoothly in time. Consider the plot of tangential speed shown in Figure 3b. Scrutiny of the curve in the early phase of the motion reveals that the slope of the plot of speed against time is initially zero and increases smoothly. This is striking given that the positional error is maximal at the beginning of the movement. Third, the shape of the plot of hand speed is unimodal and roughly symmetric (bell–shaped). There are exceptions to this observation as well, particularly for movements in which the accuracy requirement lead to feedback playing an important role (Beggs and Howarth, 1972; MacKenzie et al., 1987; Milner and Ijaz, 1990), but again this observation characterizes a reasonably large class of movements.

Insert Figure 3 about here

Optimal control approaches

Optimization theory provides a computational framework which is natural for a selection process such as motor planning (Bryson and Ho, 1975). Rather than describing the kinematics of a movement directly, in terms of the time-varying values of positions or angles, the movement is described more abstractly, in terms of a global measure such as total efficiency, smoothness, accuracy or duration. This global measure encodes the cost of the movement, and the optimal movement is the movement that minimizes the cost. In this framework the cost function is a mathematical means for specifying the plan. The variables that determine the cost, and that are therefore planned, determine the patterns of behavior observed. For example, we might postulate that the trajectory followed by the hand is the one that minimizes the total energy expended during the movement. The theoretical goal is to formulate a single such postulate, or a small set of postulates, that account for a wide variety of the data on reaching.

Let us begin by considering the mathematical machinery that is needed to describe "optimal" movements. Letting T denote the duration of a movement, and letting $x(t)$ denote the value of degree of freedom x at time t, a movement is a function $x(t)$, $t \in [0, T]$. There are an infinite number of such functions. An optimization approach proposes to choose between these functions by comparing them on the basis of a measure of cost. The cost function is a *functional*, a function that maps functions into real numbers. To every movement corresponds a number, which provides a basis of comparison between movements. The cost function in dynamic optimization is generally taken to be of the following form:

$$J = \int_0^T g(x(t), t)dt, \tag{1}$$

where $g(x(t), t)$ is a function that measures the instantaneous cost of the movement. The instantaneous cost g typically quantifies aspects of the movement that are considered undesirable, such as jerkiness, or error, or high energy expenditure. Integrating the instantaneous cost provides the total measure of cost J.

The mathematical techniques that have been developed for optimizing expressions such as Equation 1 fall into two broad classes: those based on *calculus of variations* and those based on *dynamic programming* (Bryson and Ho, 1975). Both classes of techniques provide mathematical conditions that characterize optimal trajectories. For certain classes of problems, these conditions provide equations that can be solved once and for all, yielding closed form expressions for optimal trajectories. For other classes of problems, the equations must be solved numerically.

Note that a description of movement in terms of an optimization principle does not necessarily imply that there is a computational process of optimization underlying the actual control of movement. Optimization theory simply stipulates that the system operates at the minimum of the "cost function" (the measure that is optimized), but does not commit itself to any particular computational process that puts the system at the minimum. While many possible cost functions have been examined (Nelson, 1983) there are two main classes of model proposed for point-to-point movements—kinematic and dynamic based models.

Kinematic costs

The cost function in kinematic based models contain only geometrical and time-based properties of motion and the variables of interest are the positions (e.g. joint angles or hand Cartesian coordinates) and their corresponding velocities, accelerations and higher derivatives. Based on the observation that point-to-point movements of the hand are smooth when viewed in a Cartesian framework, it was proposed that the squared first derivative of Cartesian hand acceleration or 'jerk' is minimized over the movement (Hogan, 1984; Flash and Hogan, 1985). Letting $x(t)$ denote the position at time

t, the minimum jerk model is based on the following cost function:

$$J = \int_0^T \left[\frac{d^3 x}{dt^3} \right]^2 dt, \tag{2}$$

where T is the duration of the movement. Using the calculus of variations, Hogan showed that the trajectory that minimizes this cost function is of the following form:

$$x(t) = x_0 + (x_f - x_0)[10(t/T)^3 - 15(t/T)^4 + 6(t/T)^5], \tag{3}$$

where x_0 and x_f are the initial and final positions, respectively.

The minimum jerk model takes the smoothness of motion as the basic primitive and predicts straight-line Cartesian hand paths with bell-shaped velocity profiles which are consistent with empirical data for rapid movements made without accuracy constraints. (Flash and Hogan, 1985) point out that the natural generalization of Equation 2 to three spatial dimensions involves taking the sum of the squares of the jerk along each dimension, which is equivalent to three independent minimizations. Thus, the resulting trajectories are all of the form of Equation 3, differing only in the values of the initial and final positions. Since these differences simply scale the time-varying part of Equation 3, the result is always a straight-line motion in space. While penalizing higher derivatives of kinematic variables such as snap, crackle or pop (the next three derivatives after jerk) produce very similar trajectories, penalizing lower derivatives of kinematic variables, such as acceleration or velocity lead to non-zero accelerations at the initial time which is inconsistent with the behavioral data.

One aspect of the minimum jerk model that is unsatisfying is the need to pre-specify the duration T. (Hoff, 1992) has extended the minimum jerk model by allowing the duration to be a free parameter. Because longer movements can always be made smoother than short movements, Hoff created a tradeoff between duration and smoothness by penalizing duration. This is accomplished via the following cost function:

$$J = \int_0^T \left\{ \gamma \left[\frac{d^3 x}{dt^3} \right]^2 + 1 \right\} dt, \tag{4}$$

where T is free. The term of unity in the integrand increases the cost of movements as a function of their duration and trades off against the cost due to smoothness. The parameter γ quantifies the relative magnitudes of these two costs. Hoff has shown that this model can reproduce the results from experiments in which the locations of targets are switched just before the onset of movement (Pelisson et al., 1986; Georgopoulos et al., 1981). The model successfully predicts both movement trajectories and movement durations.

The minimum-jerk model has recently been used in an attempt to find a unified framework within which to understand two properties of trajectory formation—local isochrony and the two-thirds power law. Whereas "global" isochrony refers to the observation that the average velocity of movements increase with the movement distance thereby maintaining movement duration nearly constant, "local" isochrony refers to the subunits of movement. For example, if subjects trace out a figure eight in which the two loops are of unequal size, the time to traverse each loop is approximately equal. By approximating the solution of minimum jerk when the path is constrained (only the velocity along the path could be varied) local isochrony becomes an emergent property of the optimization of jerk (Viviani and Flash, 1995). The two-thirds power law, $A \propto C^{\beta}(\beta \approx 2/3)$, is based on the observation of the relationship between path curvature (C) and hand angular velocity (A) during drawing or scribbling (Lacquaniti et al., 1983) (for a more general formulation of the law see (Viviani and Schneider, 1991)). It has been shown that the minimum-jerk solution for movement along a constrained path approximates the solution given by the two-thirds power law (Viviani and Flash, 1995). One area of debate is the extent to which the two-thirds power law is a manifestation of a plan rather than a control constraint. Gribble and Ostry, utilizing a simple dynamical model have shown that the two-thirds power law could be an emergent property of the visco-elastic properties of the muscles (Gribble and Ostry, 1996). One feature which has not yet been explained by such emergent property models, however, is the fact that the exponent of the power law, β, changes systematically through development, from a value of 0.77 at age 6 to an adult value of 0.66 (2/3) at around 12 years (Viviani and Schneider, 1991).

Dynamic costs

The cost function in dynamic based models depends on the dynamics of the arm, and the variables of interest include joint torques, forces acting on the hand and muscle commands. Several models have been proposed in which the cost function depends on dynamic variables such as torque change, muscle-tension or motor-command (Kawato, 1992; Uno et al., 1989). One critical difference between the kinematic and dynamic based models is the separability of planning and execution. The specification of the movement in kinematic models, such as minimum jerk, involves the positions and velocities of the arm as a function of time. Therefore, a separate process is required to achieve these specifications and this model is a hierarchical, serial plan-and-execute model. In contrast, the solution to dynamic models, such as minimum torque change, are the motor commands required to achieve the movement and therefore planning and execution are no longer separate processes.

(Uno et al., 1989) have presented data that are problematic for the minimum jerk model. First, they studied trajectories when an external force (a spring) acted on the hand. They found that subjects made curvilinear movements in this case, which is not predicted by the minimum jerk model.[1] Second, they studied movements with via points and observed that symmetrically placed via points did not necessarily lead to symmetric paths of the hand in space. Finally, they studied large range movements and observed significant curvilinearity in the paths of motion. These observations led Uno, et al. to suggest an alternative optimization principle in which forces play a role. They proposed penalizing the rate of change of torque, a quantity which is locally proportional to jerk (under static conditions). This principle is captured by the following cost function:

$$J = \int_0^T \sum_{i=1}^{n} \left[\frac{d\tau_i}{dt} \right]^2 dt, \tag{5}$$

where $d\tau_i/dt$ is the rate of change of torque at the i^{th} joint. Uno et al. showed that this *minimum torque change* cost function predicts trajectories that correspond to the trajectories they observed empirically.

Minimum variance planning

Although both minimum jerk and minimum torque-change are able to capture many aspects of observed trajectories they have several features which make them unsatisfying as models of movement. First, there has been no principled explanation why the CNS should choose to optimize such quantities as jerk or torque-change, other than these models predict smooth trajectories. These models do not propose any advantage for smoothness of movement but simply assume that smoothness is optimized. Furthermore, it is still unknown whether the CNS could estimate such complex quantities as jerk or torque-change, and integrate them over the duration of a trajectory. Lastly the models provide no principled way of selecting the movement duration which is a free parameters in both models.

In an attempt to resolve these problems Harris and Wolpert have recently proposed the minimum-variance theory of motor planning for both eye and arm movements (Harris and Wolpert, 1998). They suggest that biological noise is the underlying determinant of both eye and arm movement planning. In the model they assume that the neural control signal is corrupted by noise, thereby causing trajectories to deviate from the desired path. These deviations will be accumulated over the duration of a movement leading to variability in the final position. If the noise were independent

[1] The minimum jerk model is entirely kinematic (forces do not enter into the optimization procedure); thus, it predicts straight line motion regardless of external forces. This argument assumes that subjects are able in principle to compensate for the external force.

of the control signal, then the accumulated error could be minimized by making the movement as rapidly as possible. However, the key assumption of the hypothesis is that the noise in the neural control signal increases with the mean level of the signal. In the presence of such signal-dependent noise, moving as rapidly as possible may actually increase the variability in the final position. This is because, for low-pass systems such as the eye or arm, moving faster requires larger control signals, which would carry more noise. Since the resulting inaccuracy of the movement may require corrective behavior, such as further movements, moving very fast may become counter-productive (Meyer et al., 1988; Harris, 1995). Accuracy could be improved by having small amplitude control signals, but the movement will consequently be slow. Thus, signal-dependent noise inherently imposes a trade-off between movement duration and terminal accuracy. This is in agreement the well-known observation that faster arm movements, for a given amplitude, are more inaccurate (Fitts, 1954). The key point is that for a given amplitude and duration of movement, the final positional variance will depend critically on the actual neural commands and subsequent velocity profile.

Harris and Wolpert suggest that the feedforward sequence of neural commands are selected so as to minimize the final positional variance while keeping the duration to the minimum compatible with the accuracy constraints of a particular task. This minimum-variance theory accurately predicts the velocity profiles of both saccades and arm movements (Figure 4). Moreover, the profiles are relatively insensitive to large changes in the parameters of the arm, so that even when the inertia and viscosity of the arm or the time-constants of the muscle were individually halved or doubled, the optimal profile remained essentially unchanged (Figure 4c). This is consistent with the observation that when the arm is subject to elastic, viscous or inertial loads, the bell-shaped velocity profile is regained after a period of adaptation (Flash and Gurevich, 1991; Shadmehr and Mussa-Ivaldi, 1994; Lackner and DiZio, 1994; Brashers-Krug et al., 1996; Sainburg and Ghez, 1995; Goodbody and Wolpert, 1998).

Insert Figure 4 about here

The minimum variance approach has several important ramifications. Primarily, it provides a biologically plausible theoretical underpinning for both eye and arm movements. In contrast, it is difficult to explain the biological relevance of such factors as jerk or torque-change. Moreover, there is no need for the CNS to construct highly derived signals to estimate the cost of the movement, which is now simply variance of the final position or the consequences of this inaccuracy, such as the time spent in making corrective movements (Meyer et al., 1988; Harris, 1995). Such costs are directly available to the nervous system and the optimal trajectory could be learned from the experience of repeated movements. This model is therefore a combined kinematic and dynamic model. The variance is determined by the dynamics of the system but the consequences of this variance may be assessed in extrinsic coordinates, such as a visual error. Finally, the model explains why theories based on smoothness have been so successful. To change the velocity of the eye or arm rapidly requires large changes in the driving signal. As such large signals would generate noise it pays to avoid abrupt trajectory changes.

Internal models

The basic task of a control system is to manage the relationships between sensory variables and motor variables. There are two basic kinds of transformations that can be considered: sensory-to-motor transformations and motor-to-sensory transformations. The transformation from motor variables to sensory variables is accomplished by the environment and by the musculoskeletal system; these physical systems transform efferent motor actions into reafferent sensory feedback. It is also possible, however, to consider internal transformations, implemented by neural circuitry, that mimic the external motor-to-sensory transformation. Such internal transformations are known as *internal forward models*. Forward *dynamic* models predict the next state (e.g. position and velocity) given the current state and the motor command whereas forward *output* models predict the sensory feedback. This in contrast to inverse models which invert the system by providing the motor command which will cause a desired change in state. As inverse models produce the motor command required to achieve some desired result they have a natural use as a controller (see the section on Motor Learning).

Motor prediction: Forward models

In this section we discuss possible roles for forward models in motor control and motor learning (cf. (Miall and Wolpert, 1996)). One such role is as an ingredient in a system that uses a copy of the motor command—an "efference copy"—to anticipate and cancel the sensory effects of movement—the "reafference". This role for forward models has been extensively studied in the field of eye movement control (see (Jeannerod, 1997) for a review). In the case of limb control, a forward model may subserve the additional function of canceling the effects on sensation induced by self-motion and distinguishing self-produced motion from the sensory feedback caused by contact with objects in the environment.

Another role for forward models is to provide a fast internal loop that helps stabilize feedback control systems. Feedback control in biological systems is subject to potential difficulties with stability, because the sensory feedback through the periphery is delayed by a significant amount (Miall et al., 1993). Such delays can result in instability when trying to make rapid movements under feedback control. Two strategies can maintain stability during movement with such delays—intermittency and prediction. Intermittency, in which movement is interspersed with rest, is seen in manual tracking and saccadic eye movement. The intermittency of movement allows time for veridical sensory feedback to be obtained (a strategy often used in adjusting the temperature of a shower where the time delays are large). Such intermittency can arise either from a psychological refractory period after each movement (Smith, 1967) or an error deadzone (Wolpert et al., 1992) in which the perceived error must exceed a threshold before a new movement is initiated. Alternatively, in predictive control a forward model is used to provide internal feedback of the predicted outcome of an action which can be used before sensory feedback is available thereby preventing instability (Miall et al., 1993). In effect the control system controls the forward model rather than the actual system. Because the loop through the forward model is not subject to peripheral delays, the difficulties with stability are lessened. The control signals obtained within this inner loop are sent to the periphery and the physical system moves along in tandem. Of course, there will be inevitable disturbances acting on the physical system that are not modeled by the internal model, thus the feedback from the actual system cannot be neglected entirely. However, the *predictable* feedback can be canceled by delaying the output from the forward model. Only the unpredictable components of the feedback, which are likely to be small, are used in correcting errors within the feedback loop through the periphery. This kind of feedback control system, which uses a forward model both for mimicking the plant and for canceling predictable feedback, is known in the engineering literature as a "Smith predictor." Miall, et al. have proposed that the cerebellum acts as a Smith predictor.

Another interesting example of a forward model arises in the literature on speech production. (Lincoln, 1979) studied an experimental task in which subjects produced vowel sounds while their jaw was held open by a bite block. Lindblom et al. observed that the vowels produced by the subjects had formant frequencies in the normal range, despite the fact that unusual articulatory postures were required to produce these sounds. Moreover, the formant frequencies were in the normal range during the first pitch period, before any possible influence of acoustic feedback. Lindblom et al. proposed a model of the control system for speech production that involved placing a forward model of the vocal tract in an internal feedback pathway.

Finally, forward models can also play a role in both state estimation and motor learning, as we discuss in the following section.

Inverse models

We can also consider internal models that perform a transformation in the opposite direction, from sensory variables to motor variables. Such transformations are known as *internal inverse models*, and they allow the motor control system to transform desired sensory consequences into the motor actions that yield these consequences. Internal inverse models are the basic module in open-loop control systems.

Internal inverse models also play an important role in motor control. A particularly clear example of an inverse model arises in the vestibulo-ocular reflex (VOR). The VOR couples the movement of the eyes to the motion of the head, thereby allowing an organism to keep its gaze fixed in space. This is achieved by causing the motion of the eyes to be equal and opposite to the motion of the head (Robinson, 1981). In effect, the VOR control system must compute the motor command that is predicted to yield a particular eye velocity. This computation is an internal inverse model of the physical relationship between muscle contraction and eye motion.

There are many other such examples of inverse models. Indeed, inverse models are a fundamental module in open-loop control systems: they allow the control system to compute an appropriate control signal without relying on error-correcting feedback.[2]

An important issue to stress in our discussion of internal models is that internal models are not required to be detailed or accurate models of the external world. Often an internal model need only provide a rough approximation of some external transformation in order to play a useful role. For example, an inaccurate inverse model can provide an initial open-loop "push" that is corrected by a feedback controller. Similarly, an inaccurate forward model can be used inside an internal feedback loop, because the feedback loop corrects the errors. This issue will arise again in the following section when we discuss motor learning, where we will see that an inaccurate forward model can be used to learn an accurate controller.

State estimation

Although the state of a system is not directly available to the controller, it is possible to estimate the state indirectly. Such a state estimator, known as an "observer" (Goodwin and Sin, 1984), produces its estimate of the current state by monitoring the stream of inputs (motor commands) and outputs (sensory feedback) of the system (Figure 1). By using both sources of information the observer can

[2]Note, however, that inverse models are not the *only* way to implement open-loop control schemes. As we have seen, an open-loop controller can also be implemented by placing a forward model in an internal feedback loop. See (Jordan, 1996) for further discussion.

reduce its uncertainty in the state estimate and is robust to sensor failure. In addition, as there are delays in sensory feedback, the observer can use the motor command to produce more timely state estimates than would be possible using sensory feedback alone.

Although many studies have examined integration among purely sensory stimuli (for a psychophysical review see (Welch and Warren, 1986)) little is known of how sensory and motor information is integrated during movement. When we move our arm in the absence of visual feedback, there are three basic methods that the CNS can use to obtain an estimate of the current state (e.g., the position and velocity) of the hand. The system can make use of sensory inflow (the information available from proprioception), it can make use of motor outflow (the motor commands sent to the arm), or it can combine these two sources of information. While sensory signals can directly cue the location of the hand, motor outflow generally does not. For example, given a sequence of torques applied to the arm (the motor outflow) an internal model of the arm's dynamics is needed to estimate the arm's final configuration.

To explore whether an internal model of the arm is used in sensorimotor integration, Wolpert, et al. studied a task in which subjects—after initially viewing their arm in the light—made arm movements in the dark (Wolpert et al., 1995). The subjects' internal estimate of their hand location was assessed by asking them to visually localize the position of their hand (which was hidden from view) at the end of the movement. Wolpert et al. developed a model of this sensorimotor integration process (Figure 5). This Kalman filter model of the sensorimotor integration process is based on a formal engineering model from the optimal state estimation field (Goodwin and Sin, 1984). The Kalman filter is a linear dynamical system that produces an estimate of the location of the hand by using both the motor outflow and sensory feedback in conjunction with a model of the motor system, thereby reducing the overall uncertainty in its estimate (Kalman and Bucy, 1961). This model assumes that the localization errors arise from two sources of uncertainty, the first from the variability in the response of the arm to the motor command and the second in the sensory feedback given the arm's configuration. The Kalman filter model can be considered as the combination of two processes which together contribute to the state estimate. The first, feedforward process (upper part) uses the efferent outflow along with the current state estimate to predict the next state by simulating the movement dynamics with a forward model. The second, feedback process (lower part) compares the sensory inflow with a prediction of the sensory inflow based on the current state. The sensory error, the difference between actual and predicted sensory feedback, is used to correct the state estimate resulting from the forward model. The relative contributions of the internal simulation and sensory correction processes to the final estimate are modulated by the time varying Kalman gain so as to provide optimal state estimates. Unlike simpler models which do not integrate both sensory and motor information, this model could account for the empirical data (Wolpert et al., 1995). This suggests that a forward model is used by the CNS to maintain an estimate of the state of the motor system.

Insert Figure 5 about here

(Duhamel et al., 1992) have presented neurophysiological data that imply a role for an internal forward model in the saccadic system. Duhamel, et al. have proposed that one of the roles of area LIP of parietal cortex is to maintain a retinal representation of potential saccadic targets. Such a representation simplifies the task of saccade generation, because the transformation from a retinal representation to a motor representation is relatively simple. The retinal representation must be updated, however, whenever the eyes move in the head. This updating process requires an internal forward model that embodies knowledge of the retinal effects of eye movements. In particular, for a given eye movement (a motor action), the brain must predict the motion of objects on the retina (the sensory consequences). This predicted motion is added to the current retinal representation to yield an updated retinal representation.

Motor Learning

In the previous section we have seen several ways in which internal models can be used in a control system. Inverse models are the basic building block of open-loop control. Forward models can also be used in open-loop control, and have additional roles in state estimation and compensation for delays. It is important to emphasize that an internal model is a form of knowledge about the environment (cf. (Ghez et al., 1990; Lacquaniti et al., 1992; Shadmehr and Mussa-Ivaldi, 1994)). Many motor control problems involve interacting with objects in the external world, and these objects generally have unknown mechanical properties. There are also changes in the musculoskeletal system due to growth or injury. These considerations suggest an important role for adaptive processes. Through adaptation the motor control system is able to maintain and update its internal knowledge of external dynamics.

Recent work on motor learning has focused on the representation of the inverse dynamic model. When subjects make point-to-point movements in which the dynamics of their arm is altered, for example by using a robot to generate a force field acting on the hand, they initially show trajectories which deviate from their normal paths and velocity profiles (Shadmehr and Mussa-Ivaldi, 1994;

Lackner and DiZio, 1994). However, over time subjects adapt and move naturally in the presence of the force field. This can be interpreted as adaptation of the inverse model or the incorporation of an auxiliary control system to counteract the novel forces experienced during movement. Several theoretical questions have been addressed using this motor learning paradigm. The first explored the representation of the controller and in particular whether it was best represented in joint or Cartesian space (Shadmehr and Mussa-Ivaldi, 1994). This was investigated by examining the generalization of motor learning at locations in the workspace remote from where subjects had adapted to the force field. By assessing in which coordinate system the transfer occurred evidence was provided for joint-based control. Another important advance was made in a study designed to answer whether the order in which states (position and velocities) were visited was important for learning or whether having learned a force field for a set of states subjects would be able to make natural movements when visiting the states in a novel order (Conditt et al., 1997). The findings showed that the order was unimportant and argue strongly against a rote learning of individual trajectories. The learning of novel dynamic has been shown to undergo a period of consolidation after the perturbation has been removed (Brashers-Krug et al., 1996). Subjects' ability to perform in a previously experienced field was disrupted if a different field was presented immediately after this initial experience. Consolidation of this motor learning appears to be a gradual process with a second field after four hours having no effect on subsequent performance in the first field. This suggests that motor learning undergoes a period of consolidation during which time the motor learning or memory is fragile to being disrupted by different motor learning.

In this section we discuss the computational algorithms by which such motor learning could take place, focusing on five different approaches: direct inverse modeling, feedback error learning, distal supervised learning, reinforcement learning and unsupervised bootstrap learning. All of these approaches provide mechanisms for learning general sensorimotor transformations. They differ principally in the kinds of data that they require, and the kinds of auxiliary supporting structure that they require.

The first three schemes that we discuss are instances of a general approach to learning known as *supervised learning*. A generic supervised learning system is shown in Figure 6. A supervised learner requires a *target output* corresponding to each input. The error between the target output and the actual output is computed and is used to drive the changes to the parameters inside the learning system. This process is generally formulated as an optimization problem in which the cost function is one-half the squared error:

$$J = \frac{1}{2}\|\mathbf{y}^* - \mathbf{y}\|^2. \tag{6}$$

The learning algorithm adjusts the parameters of the system so as to minimize this cost function. For details on particular supervised learning algorithms, see (Hertz et al., 1991).

Insert Figure 6 about here

In the following, we assume that the controlled system or *plant* (the musculoskeletal system and any relevant external dynamical systems) is described by a set of state variables $\mathbf{x}[n]$, an input $\mathbf{u}[n]$ and an output $\mathbf{y}[n]$. These variables are related by the following dynamical equation:

$$\mathbf{x}[n+1] = f(\mathbf{x}[n], \mathbf{u}[n]), \tag{7}$$

where n is the time step, and f is the *next-state equation*. We also require an *output equation* that specifies how the output $\mathbf{y}[n]$ is obtained from the current state:

$$\mathbf{y}[n] = g(\mathbf{x}[n]). \tag{8}$$

We use the notation $\mathbf{y}^*[n]$ to refer to a desired value (a target value) for the output variable, and the notation $\hat{\mathbf{x}}[n]$ to refer to an internal estimate of the state of the controlled system.

Direct inverse modeling

How might a system acquire an inverse model of the plant? One approach is to present various test inputs to the plant, observe the outputs, and provide these input-output pairs as training data to a supervised learning algorithm by reversing the role of the inputs and the outputs. That is, the plant output is provided as an input to the learning controller, and the controller is required to produce as output the corresponding plant input. This approach, shown diagrammatically in Figure 7 is known as direct inverse modeling (Widrow and Stearns, 1985; Atkeson and Reinkensmeyer, 1988; Kuperstein, 1988; Miller, 1987). Note that we treat the plant output as being observed at time n. The input to the learning controller is the current plant output $\mathbf{y}[n]$ and the delayed state estimate $\hat{\mathbf{x}}[n-1]$. The controller is required to produce the plant input that gave rise to the current output, in the context of the delayed estimated state. This is a supervised learning problem, in which the plant input $\mathbf{u}[n]$ serves as the target in the following cost function:

$$J = \frac{1}{2}\|\mathbf{u}[n] - \hat{\mathbf{u}}[n]\|^2, \tag{9}$$

where $\hat{\mathbf{u}}[n]$ denotes the controller output at time n.

Insert Figure 7 about here

An example of a problem for which direct inverse modelling approach is applicable is the *inverse kinematics* problem. The problem is to learn a sensorimotor transformation between the desired position of the hand in spatial coordinates and a corresponding set of joint angles for the arm that achieve that position. To learn such a transformation, the system tries a random joint angle configuration (a vector $\mathbf{u}[n]$), and observes the resulting hand position (the vector $\mathbf{y}[n]$). The system gathers a number of such pairs and uses a supervised learning algorithm to learn a mapping from $\mathbf{y}[n]$ to $\mathbf{u}[n]$.

Nonlinear systems and the nonconvexity problem

The direct inverse modeling approach is well-behaved for linear systems and indeed can be shown to converge to correct parameter estimates for such systems under certain conditions (Goodwin and Sin, 1984). For nonlinear systems, however, a difficulty arises that is related to the general "degrees-of-freedom problem" in motor control (Bernstein, 1967). The problem is due to a particular form of redundancy in nonlinear systems (Jordan, 1992). In such systems, the "optimal" parameter estimates (i.e., those that minimize the cost function in Equation 9) in fact may yield an incorrect controller.

Consider the following simple example. Figure 8 shows a one degree-of-freedom "archery" problem: A controller chooses an angle u and an arrow is projected at that angle. Figure 8b shows the parabolic relationship between distance traveled and angle. Note that for each distance, there are two angles that yield that distance. This implies that a learning system using direct inverse modeling sees two different targets paired with any given input. If a least-squares cost function is used (cf. Equation 9), then the system produces an output which is the average of the two targets, which by the symmetry of the problem is 45 degrees. Thus the system converges to an incorrect controller that maps each target distance to the same 45 degree control signal.

Insert Figure 8 about here

Feedback error learning

(Kawato et al., 1987) have developed a direct approach to motor learning known as *feedback error learning*. Feedback error learning makes use of a feedback controller to guide the learning of the feedforward controller.[3] Consider a composite feedback-feedforward control system, in which the total control signal is the sum of the feedforward component and the feedback component:

$$\mathbf{u}[n] = \mathbf{u}_{ff}[n] + \mathbf{u}_{fb}[n].$$

In the context of a direct approach to motor learning, the signal $\mathbf{u}[n]$ is the target for learning the feedforward controller (cf. Figure 7). The error between the target and the feedforward control signal is $(\mathbf{u}[n] - \mathbf{u}_{ff}[n])$, which in the current case is simply $\mathbf{u}_{fb}[n]$. Thus an error for learning the feedforward controller can be provided by the feedback control signal (see Figure 9).

Insert Figure 9 about here

An important difference between feedback error learning and direct inverse modeling regards the signal used as the controller input. In direct inverse modeling the controller is trained "off-line;" that is, the input to the controller for the purposes of training is the actual plant output, not the desired plant output. For the controller to actually participate in the control process, it must receive the desired plant output as its input. The direct inverse modeling approach therefore requires a switching process—the desired plant output must be switched in for the purposes of control and the actual plant output must be switched in for the purposes of training. The feedback error learning approach provides a more elegant solution to this problem. In feedback error learning, the desired plant output is used for both control and training. The feedforward controller is trained "on-line;" that is, it is used as a controller while it is being trained. Although the training data that it receives—pairs of actual plant inputs and desired plant outputs—are not samples of the inverse dynamics of the plant, the system nonetheless converges to an inverse model of the plant because of the error-correcting properties of the feedback controller.

By utilizing a feedback controller, the feedback error learning approach also solves another problem associated with direct inverse modeling. Direct inverse modeling is not *goal directed*; that is, it is not sensitive to particular output goals (Jordan and Rosenbaum, 1989). This is seen by simply observing that the goal signal $(\mathbf{y}^*[n])$ does not appear in Figure 7. The learning process samples

[3] A *feedforward controller* is an internal inverse model of the plant that is used in an open-loop control configuration. See (Jordan, 1996) for further discussion of feedforward and feedback control.

randomly in the control space, which may or may not yield a plant output near any particular goal. Even if a particular goal is specified before the learning begins, the direct inverse modeling procedure must search throughout the control space until an acceptable solution is found. In the feedback error learning approach, however, the feedback controller serves to guide the system to the correct region of the control space. By using a feedback controller, the system makes essential use of the error between the desired plant output and the actual plant output to guide the learning. This fact links the feedback error learning approach to the indirect approach to motor learning that we discuss in the following section. In the indirect approach, the learning algorithm is based directly on the output error.

Distal supervised learning

In this section we describe an indirect approach to motor learning known as distal supervised learning. Distal supervised learning avoids the nonconvexity problem and also avoids certain other problems associated with direct approaches to motor learning (Jordan, 1990; Jordan and Rumelhart, 1992). In distal supervised learning, the controller is learned indirectly, through the intermediary of a forward model of the plant. The forward model must itself be learned from observations of the inputs and outputs of the plant. The distal supervised learning approach is therefore composed of two interacting processes, one process in which the forward model is learned and another process in which the forward model is used in the learning of the controller.

Insert Figure 10 about here

The distal supervised learning approach is illustrated in Figure 10. There are two interwoven processes depicted in the figure. One process involves the acquisition of an internal forward model of the plant. The forward model is a mapping from states and inputs to predicted plant outputs and it is trained using the *prediction error* $(\mathbf{y}[n] - \hat{\mathbf{y}}[n])$, where $\hat{\mathbf{y}}[n])$ is the output of the forward model. The second process involves training the controller. This is accomplished in the following manner. The controller and the forward model are joined together and are treated as a single *composite learning system*. If the controller is to be an inverse model, then the composite learning system should be an identity transformation (i.e., a transformation whose output is the same as its input). This suggests that the controller can be trained indirectly by training the composite learning system to be an identity transformation. This is a supervised learning problem in which the entire composite learning system (the system inside the dashed box in the figure) corresponds to the box labeled "Learner" in Figure 6. During this training process, the parameters in the forward model are held fixed. Thus the composite learning system is trained to be an identity transformation by a constrained learning process in which some of the parameters inside the system are held fixed. By allowing only the controller parameters to be altered, this process trains the controller indirectly.

Training a system to be an identity transformation means that its supervised error signal is the difference between the input and the output. This error signal is just the *performance error* $(\mathbf{y}^*[n] - \mathbf{y}[n])$ (cf. Figure 10). This is a sensible error term—it is the observed error in motor performance. That is, the learning algorithm trains the controller by correcting the error between the desired plant output and the actual plant output.

Insert Figure 11 about here

Let us return to the "archery" problem. Let us assume that the system has already acquired a perfect forward model of the function relating u to y, as shown in Figure 11. The system can now utilize the forward model to recover a solution u for a given target y^*. This can be achieved in different ways depending on the particular supervised learning technique that is adopted. One approach involves using the local slope of the forward model to provide a correction to the current best guess for the control signal. (This corresponds to using gradient descent as the algorithm for training the composite learning system.) As seen in the figure, the slope provides information about the direction to adjust the control signal in order to reduce the performance error $(y^* - y)$. The adjustments to the control signal are converted into adjustments to the parameters of the controller using the chain rule.

An advantage of working with a forward model is that the nonconvexity of the problem does not prevent the system from converging to a unique solution. The system simply heads downhill toward one solution or the other (see Figure 11). Moreover, if particular kinds of solutions are preferred (e.g., the left branch versus the right branch of the parabola), then additional constraints can be added to the cost function to force the system to search in one branch or the other (Jordan, 1990).

Suppose finally that the forward model is imperfect. In this case, the error between the desired output and the predicted output is the quantity $(\mathbf{y}^*[n] - \hat{\mathbf{y}}[n])$, the *predicted performance error*. Using this error, the best the system can do is to acquire a controller that is an inverse of the forward model. Because the forward model is inaccurate, the controller is inaccurate. However, the predicted performance error is not the only error available for training the composite learning system. Because the actual plant output $(\mathbf{y}[n])$ can still be measured after a learning trial, the true performance error ($\mathbf{y}^*[n] - \mathbf{y}[n]$) is still available for training the controller. This implies that the output of the forward model can be discarded; the forward model is needed only for the structure that it provides as part of the composite learning system (e.g., the slope that it provides in Figure 11). Moreover, for this purpose an exact forward model is not required. Roughly speaking, the forward model need only provide coarse information about how to improve the control signal based on the current performance error, not precise information about how to make the optimal correction. If the performance error is decreased to zero, then an accurate controller has been found, regardless of the path taken to find that controller.

Reinforcement learning

Reinforcement learning algorithms differ from supervised learning algorithms by requiring significantly less information to be available to the learner. Rather than requiring a vector performance error or a vector target in the control space, reinforcement learning algorithms require only a scalar evaluation of performance. A scalar evaluation signal simply tells a system if it is performing well or not; it does not provide information about how to correct an error. A variety of reinforcement learning algorithms have been developed and ties have been made to optimal control theory (Sutton and Barto, 1998). A strength of reinforcement learning algorithms is their ability to learn in the face of delayed evaluation.

In the simplest reinforcement learning paradigm, there is a set of possible responses at each point in the state, goal space. Associated with the i^{th} response is a probability p_i of selecting that response as an output on a given trial. Once a response is selected and transmitted to the environment, a scalar evaluation or reinforcement signal is computed as a function of the response and the state of environment. The reinforcement signal is then used in changing the selection probabilities for future trials: If reinforcement is high, the probability of selecting a response is increased, otherwise, the probability is decreased. Typically, the probabilities associated with the remaining (unselected) responses are also adjusted in some manner, so that the total probability sums to one.

Reinforcement learning algorithms are able to learn in situations in which very little instructional information is available from the environment. In particular, such algorithms need make no comparison between the input goal (the vector $\mathbf{y}^*$) and the result obtained (the vector $\mathbf{y}$) to find a control signal that achieves the goal. When such a comparison can be made, however, reinforcement learning is still applicable but may be slower than other algorithms that make use of such information. Although in many cases in motor control it would appear that a comparison with a goal vector is feasible, the question is empirical and as yet unresolved (Adams, 1984): Does feedback during learning serve to strengthen or weaken the action just emitted or to provide structural information about how to change the action just emitted into a more suitable action?

Bootstrap learning

The techniques that we have discussed until now all require that either an error signal or an evaluation signal is available for adjusting the controller. It is also possible to consider a form of motor learning that needs no such corrective information. This form of learning improves the controller by building on earlier learning. Such an algorithm is referred to in the adaptive signal processing literature as "bootstrap learning" (Widrow and Stearns, 1985).

How might a system learn without being corrected or evaluated? Let us work within the framework discussed above for reinforcement learning, in which one of a set of responses is chosen with probability p_i. Suppose that due to prior learning the system performs a task correctly a certain fraction of the time, but still makes errors. The learning algorithm is as follows: The system selects actions according to the probabilities that it has already learned, and rewards its actions indiscriminately. That is, it rewards both correct and incorrect actions. We argue that the system can still converge to a control law in which the correct action is chosen with probability one. The reason for this is that the responses of nonselected actions are effectively weakened, because the sum of the p_i must be one. That is, if p_i is strengthened, p_j must be decreased, for all j not equal to i. Given that the system starts with the correct action having a larger probability of being selected than the other actions, that action has a larger probability of being strengthened, and thus an even larger probability of being selected. Thus if the initial balance in favor of the correct action is strong enough, the system can improve.

We know of no evidence that such a learning process is utilized by the motor control system, but the possibility appears to have never been investigated directly. The algorithm has intuitive appeal because there are situations in which it seems that mere repetition of a task can lead to improvement. The simplicity of the algorithm is certainly appealing from the point of view of neural implementation.

Modularity

While the discussion in earlier sections have focused on how a single internal model could be used in motor control, recent models have begun to investigate the computational advantages of using a set of internal models. A general computational strategy for designing modular learning systems is to treat the problem as one of combining multiple models, each of which is defined over a local region of the input space. Such a strategy has been introduced in the "mixture of experts" architecture for supervised learning (Jacobs et al., 1991b; Jordan and Jacobs, 1994). The architecture involves a set of function approximators known as "expert networks" (usually neural networks) that are combined by a classifier known as a "gating network" (Figure 12). These networks are trained simultaneously so as to split the input space into regions where particular experts can specialize. The gating network uses a soft split of the input data thereby allowing data to be processed by multiple experts, the contribution of each is modulated by the gating module's estimate of the probability that each expert is the appropriate one to use. This model has been proposed both as a model of high-level vision (Jacobs et al., 1991a) and of the role of the basal ganglia during sensorimotor learning (Graybiel et al., 1994).

Insert Figure 12 about here

Ghahramani and Wolpert (1997) have proposed this model to account for experimental data on visuomotor learning. Using a virtual reality system a single visual target location was remapped to two different hand positions depending on the starting location of the movement. Such a perturbation creates a conflict in the visuomotor map which captures the (normally one-to-one) relation between visually perceived and actual hand locations. One way to resolve this conflict is to develop two separate visuomotor maps (i.e. the expert modules), each appropriate for one of the two starting locations. A separate mechanism (i.e. the gating module) then combines, based on the starting location of the movement, the outputs of the two visuomotor maps. The internal structure of the system was probed by investigating the generalization properties in response to novel inputs, which in this case are the starting locations on which it has not been trained. As predicted by the mixture of experts model subjects were able to learn both conflicting mappings, and to smoothly interpolate from one visuomotor map to the other as the starting location was varied.

Multiple paired forward-inverse model

There are three potential benefits, with regard to motor control, in employing the modularity inherent in the mixture-of-experts models over a non-modular system when learning inverse models. First, the world is essentially modular, in that we interact with multiple qualitatively different objects and environments. By using multiple inverse models, each of which might capture the motor commands necessary when acting with a particular object or within a particular environment, we can achieve an efficient coding of the world. In other words the large set of environmental conditions in which we are required to generate movement requires multiple behaviors or sets of motor commands, each embodied within an inverse model. Second, the use of a modular system allows individual modules to participate in motor learning without affecting the motor behaviors already learned by other modules. Such modularity can therefore reduce interference between what is already learned and what is to be learned, thereby both speeding up motor learning while retaining previously learned behaviors. Third, many situations which we encounter are derived from a combinations of previously experienced contexts such as novel conjoints of manipulated objects and environments. By modulating the contribution to the final motor command of the outputs of the inverse modules, an enormous repertoire of behaviors can be generated. With as few as 32 inverse models, in which the outputs of each model either contributes or does not contribute to the final motor command, we have 2^{32} or 10^{10} behaviors—sufficient for a new behavior for every second of one's life. Therefore, multiple internal models can be regarded conceptually as motor primitives, which are the building blocks used to construct intricate motor behaviors with an enormous vocabulary.

Given that we wish to use multiple internal models, several questions naturally arise—how are they acquired during motor learning so as both to divide up the repertoire of behaviors for different contexts, while preventing new learning from corrupting previously learned behaviors? As at any given time we are usually faced with a single environment and a single object to be manipulated, the brain must solve the selection problem of issuing a single set of motor commands from its large vocabulary. That is how can the outputs of the inverse models be switched on and off appropriately in response to different behavioral contexts to generate a coordinated final motor command?

Insert Figure 13 about here

These issues have been addressed by Wolpert and Kawato (1998) in the *multiple paired forward-inverse model*. Based on the benefits of a modular approach and the experimental evidence for modularity Wolpert & Kawato (Wolpert and Kawato, 1998; Kawato and Wolpert, 1998) have proposed that the problem of motor learning and control is best solved using multiple controllers, that is inverse models. At any given time, one or a subset of these inverse models will contribute to the final motor command. The basic idea of the model is that multiple inverse models exist to control the system and each is augmented with a forward model. The system therefore contains multiple pairs of corresponding forward and inverse models (see Figure 13). Within each pair the inverse and forward internal models are tightly coupled both during their acquisition, through motor learning, and use, through gating of the inverse models' outputs dependent on the behavioural context. Key to this model is the responsibility signals which reflects, at any given time, the degree to which each pair of forward and inverse models should be responsible for controlling the current behaviour. This responsibility signal is derived from the comparison of the errors in prediction of the forward models. The smaller this prediction error, the higher the module's responsibility. The responsibilities are then used to control the learning within the forward models, with those models with high responsibilities receiving proportionally more of their error signals than modules with low responsibility. Over time the forward models learn to divide up the system dynamics experienced and the responsibilities will reflect the extent to which each forward model captures the current behaviour of the system. By using the same responsibility signal to gate the learning of the inverse models, which are updated by a training system such as feedback-error-learning, these models learn the appropriate controls for the context captured by their paired forward model. A final component of the model is a responsibility predictor which tries to predict the responsibility of the module from sensory information alone. The responsibility predictor, whose inputs are sensory signals, estimates the responsibility before movement onset whereas the forward models generate the responsibility after the consequences of movement are known. These two signals are merged to determine the final responsibility estimate which determines the contribution of each inverse model's output to the final motor command. This combined model is capable of learning to produce appropriate motor commands under a variety of contexts and is able to switch between controllers as the context changes.

Acknowledgements

Preparation of this paper was supported in part by grants from the Human Frontier Science Program, the McDonnell-Pew Foundation, the Wellcome Trust, the Medical Research Council, the Royal Society and by grant N00014-90-J-1942 awarded by the Office of Naval Research. Michael Jordan is an NSF Presidential Young Investigator.

References

Adams, J. A. (1984). Learning of movement sequences. *Psychological Bulletin*, 96:3–28.

Atkeson, C. G. and Hollerbach, J. M. (1985). Kinematic features of unrestrained vertical arm movements. *J. Neurosci.*, 5:2318–2330.

Atkeson, C. G. and Reinkensmeyer, D. J. (1988). Using associative content-addressable memories to control robots. *IEEE Conference on Decision and Control*.

Beggs, W. D. A. and Howarth, C. I. (1972). The movement of the hand towards a target. *Quarterly Journal of Experimental Psychology*, 24:448–453.

Bernstein, N. (1967). *The Coordination and Regulation of Movements*. Pergamon, London.

Brashers-Krug, T., Shadmehr, R., and Bizzi, E. (1996). Consolidation in human motor memory. *Nature*, 382:252–255.

Bryson, A. E. and Ho, Y. C. (1975). *Applied Optimal Control*. Wiley, New York.

Conditt, M. A., Gandolfo, F., and Mussa-Ivaldi, F. A. (1997). The motor system does not learn dynamics of the arm by rote memorization of past experience. *J. Neurophysiol.*, 78:1:554–560.

Duhamel, J. R., Colby, C. L., and Goldberg, M. E. (1992). The updating of the representation of visual space in parietal cortex by intended eye movements. *Science*, 255:90–92.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movements. *J. Exp. Psychol.*, 47:381–391.

Flash, T. and Gurevich, I. (1991). Human motor adaptation to external loads. *IEEE Eng. in Med. & Biol. Soc. Conference*, 13:885–886.

Flash, T. and Hogan, N. (1985). The co-ordination of arm movements: An experimentally confirmed mathematical model. *J. Neurosci.*, 5:1688–1703.

Georgopoulos, A. P., Kalaska, J. F., and Massey, J. T. (1981). Spatial trajectories and reaction times of aimed movements: Effects of practice, uncertainty and change in target location. *J. Neurophysiol.*, 46:725–743.

Ghahramani, Z. and Wolpert, D. M. (1997). Modular decomposition in visuomotor learning. *Nature*, 386:392–395.

Ghez, C., Gordon, J., Ghilardi, M. F., Christakos, C. N., and Cooper, S. E. (1990). Roles of proprioceptive input in the programming of arm trajectories. *Cold Spring Harbor Symp. Quant. Biol*, 55:837–847.

Goodbody, S. J. and Wolpert, D. M. (1998). Temporal and amplitude generalization in motor learning. *J. Neurophysiol.*, 79:1825–1838.

Goodwin, G. C. and Sin, K. S. (1984). *Adaptive Filtering Prediction and Control*. Prentice-Hall, Englewood Cliffs, NJ.

Graybiel, A. M., Aosaki, T., Flaherty, A. W., and Kimura, M. (1994). The basal ganglia and adaptive motor control. *Science*, 265(5180):1826–1831.

Gribble, P. L. and Ostry, D. J. (1996). Origins of the power-law relation between movement velocity and curvature - modeling the effects of muscle mechanics and limb dynamics. *J. Neurophysiol.*, 76:5:2853–2860.

Harris, C. M. (1995). Does saccadic under-shoot minimize saccadic flight-time? A Monte-carlo study. *Vision Res.*, 35:691–701.

Harris, C. M. and Wolpert, D. M. (1998). Signal-dependent noise determines motor planning. *Nature (In Press)*.

Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the theory of nbeural computation*. Addison Wesley, Redwood City, CA.

Hoff, B. R. (1992). *A computational description of the organization of human reaching and prehension*. PhD Thesis, University of Southern California.

Hogan, N. (1984). An organizing principle for a class of voluntary movements. *J. Neurosci.*, 4:2745–2754.

Jacobs, R. A., Jordan, M. I., and Barto, A. G. (1991a). Task decomposition through competition in a modular connectionist architecture: The what and where vision tasks. *Cognitive Science*, 15(2):219–250.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991b). Adaptive mixture of local experts. *Neural Computation*, 3:79–87.

Jeannerod, M. (1997). *The Cognitive Neuroscience of Action*. Blackwell, Oxford.

Jordan, M. I. (1990). Motor learning and the degrees of freedom problem. In Jeannerod, M., editor, *Attention and Performance, XIII*. Erlbaum, Hillsdale, NJ.

Jordan, M. I. (1992). Constrained supervised learning. *J. of Mathematical Psychology*, 36:396–425.

Jordan, M. I. (1996). Computational aspects of motor control and motor learning. In Heuer, H. and Keele, S., editors, *Handbook of Perception and Action: Motor Skills*. Academic Press, New York.

Jordan, M. I. and Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214.

Jordan, M. I. and Rosenbaum, D. A. (1989). Action. In Posner, M. I., editor, *Foundations of Cognitive Science*. MIT Press, Cambridge, MA.

Jordan, M. I. and Rumelhart, D. E. (1992). Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16:307–354.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction. *J. of Basic Engineering (ASME)*, 83D:95–108.

Kawato, M. (1992). Optimization and learning in neural networks for formation and control of coordinated movement. In Meyer, D. and Kornblum, S., editors, *Attention and Performance, XIV: Synergies in Experimental Psychology, Artificial Intelligence, and Cognitive Neuroscience - A Silver Jubilee.*, pages 821–849. MIT press, Cambridge, Massachusetts.

Kawato, M., Furawaka, K., and Suzuki, R. (1987). A hierarchical neural network model for the control and learning of voluntary movements. *Biol. Cybern.*, 56:1–17.

Kawato, M. and Wolpert, D. M. (1998). Internal models for motor control. In Glickstein, M. and Bock, R., editors, *Sensory Guidance of Movement*. Novartis (In press).

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

Lackner, J. R. and DiZio, P. (1994). Rapid adaptation to Coriolis force perturbations of arm trajectory. *J. Neurophysiol.*, 72(1):299–313.

Lacquaniti, F., Borghese, N. A., and Carrozzo, M. (1992). Internal models of limb geometry in the control of hand compliance. *Journal of Neuroscience*, 12:1750–1762.

Lacquaniti, F., Terzuolo, C. A., and Viviani, P. (1983). The law relating kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54:115–130.

Lincoln, G. A. (1979). Testosterone. *Br. Med. Bull.*, 35:167–167.

MacKenzie, C. L., Marteniuk, R. G., Dugas, C., Liske, D., and Eickmeier, B. (1987). Three dimensional movement trajectories in fitts' task: Implications for control. *Quarterly Journal of Experimental Psychology*, 39:629–647.

Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., and Smith, J. E. K. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychol. Rev.*, 98:340–370.

Miall, R. C., Weir, D. J., Wolpert, D. M., and Stein, J. F. (1993). Is the cerebellum a Smith Predictor? *J. Motor Behav.*, 25(3):203–216.

Miall, R. C. and Wolpert, D. M. (1996). Forward models for physiological motor control. *Neural Networks*, 9:8:1265–1279.

Miller, W. T. (1987). Sensor-based control of robotic manipulators using a general learning algorithm. *IEEE J. of Robotics and Automation*, 3:157–165.

Milner, T. E. and Ijaz, M. M. (1990). The effect of accuracy contraints on three-dimensional movement kinematics. *Neurosci.*, 35:365–374.

Morasso, P. (1981). Spatial control of arm movements. *Exp. Brain Res.*, 42:223–227.

Nelson, W. L. (1983). Physical principles for economies of skilled movements. *Biol. Cybern.*, 46:135–147.

Pelisson, D., Prablanc, C., Goodale, M. A., and Jeannerod, M. (1986). Visual control of reaching movements without vision of the limb. II. evidence of fast unconscious processes correcting the trajectory of the hand to the final position of a double-step stimulus. *Exp. Brain Res.*, 62:303–311.

Robinson, D. A. (1981). Control of eye movements. In *Handbook of Physiology: The Nervous System II*, pages 1275–1320. -, -.

Sainburg, R. L. and Ghez, C. (1995). Limitations in the learning and generalization of multijoint dynamics. *Society For Neurosci. Abstracts*, 21(1):686.

Shadmehr, R. and Mussa-Ivaldi, F. (1994). Adaptive representation of dynamics during learning of a motor task. *J. Neurosci.*, 14:5:3208–3224.

Smith, M. C. (1967). Theories of the psychological refractory period. *Psych. Bull.*, 67:202–213.

Soechting, J. F. and Lacquaniti, F. (1981). Invariant characteristics of a pointing movement in man. *J. Neurosci.*, 1:710–720.

Sutton, R. and Barto, A. G. (1998). *Reinforcement learning.* MIT Press, Cambridge, MA.

Uno, Y., Kawato, M., and Suzuki, R. (1989). Formation and control of optimal trajectories in human multijoint arm movements: Minimum torque-change model. *Biol. Cybern.*, 61:89–101.

van der Helm, F. C. T. and Rozendaal, L. A. (1998). Musculoskeletal systems with intrinsic and proprioceptive feedback. In Winters, J. M. and Crago, P. E., editors, *Biomechanics and Neural Control of Movement.* Springer-Verlag: In Press, New York.

Viviani, P. and Flash, T. (1995). Minimum-jerk model, two-thirds power law and isochrony: converging approaches to the study of movement planning. *J. Exp. Psychol. HPP*, 21:32–53.

Viviani, P. and Schneider, R. (1991). A developmental study of the relationship between geometry and kinematics in drawing movements. *J. Exp. Psychol. HPP*, 17:198–218.

Welch, R. B. and Warren, D. H. (1986). Intersensory Interactions. In Boff, K. R., Kaufman, L., and Thomas, J. P., editors, *Handbook of Perception and Human Performance. Volume I: Sensory Processes and Perception.* Wiley.

Widrow, B. and Stearns, S. D. (1985). *Adaptive signal processing.* Prentice-Hall, Englewood Cliffs, NJ.

Wolpert, D. M. (1997). Computational approaches to motor control. *Trends Cogn. Sci.*, 1:6:209–216.

Wolpert, D. M., Ghahramani, Z., and Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, 269:1880–1882.

Wolpert, D. M. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks (In press).*

Wolpert, D. M., Miall, R. C., Winter, J. L., and Stein, J. F. (1992). Evidence for an error deadzone in compensatory tracking. *J. Motor Behav.*, 24:4:299–308.

Figure Legends

Figure 1. The motor system is shown schematically along with the five themes of motor control we will review (see text for details). The motor system (center) has inputs—the motor commands—which causes it to change its states and produce an output—the sensory feedback. For clarity not all lines are shown.

Figure 2. The levels in the motor hierarchy are shown with the triangles between the levels indicating the reduction in the degrees of freedom between the higher and lower levels. Specifying a pattern of behavior at any level completely specifies the patterns at the level below (many-to-one: many patterns at the higher level correspond to one pattern at the lower) but is consistent with many patxdterns at the level above (one-to-many). Planning can be considered as the process by which particular patterns, consistent with the extrinsic task goals, are selected at each level. Reprinted with permission from (Wolpert, 1997).

Figure 3. a) observed hand paths for a set of point-to-point movements from (Uno et al., 1989) (with permission). The coordinate system is centred on the shoulder with x and y in the transverse and sagittal directions respectively. b) observed velocity profiles for movements from T1 to T3 in a). Reprinted with permission from (Uno et al., 1989)

Figure 4. (a) Theoretical optimal trajectory for minimizing end-point variance with signal-dependent noise for a second order skeletal model of a one dimensional arm with inertia 0.25 Kg m^2 and viscosity 0.2 Nm/rad driven by a second order linear muscle with time constants of 30 and 40 ms (parameters taken from (van der Helm and Rozendaal, 1998)). (b) Eight velocity profiles for the model in (a) in which the inertia, viscosity and time constants are individually doubled or halved. The trajectory is essentially invariant to these large changes in the dynamics of the arm. Reprinted with permission from (Harris and Wolpert, 1998)

Figure 5. Sensorimotor integration model. Reprinted with permission from (Wolpert et al., 1995)

Figure 6. A generic supervised learning system. The vector $\mathbf{y}$ is the actual output and the vector $\mathbf{y}^*$ is the target output. The error between the target and the actual output is used to adjust the parameters of the learner.

Figure 7. The direct inverse modeling approach to learning a controller. The state estimate $\hat{\mathbf{x}}[n]$ is assumed to be provided by an observer (not shown).

Figure 8. (a) An "archery" problem. (b) The parabolic relationship between distance traveled (y) and angle (u) for a projectile. For each value of y there are two corresponding values of u, symmetrically placed around 45 degrees.

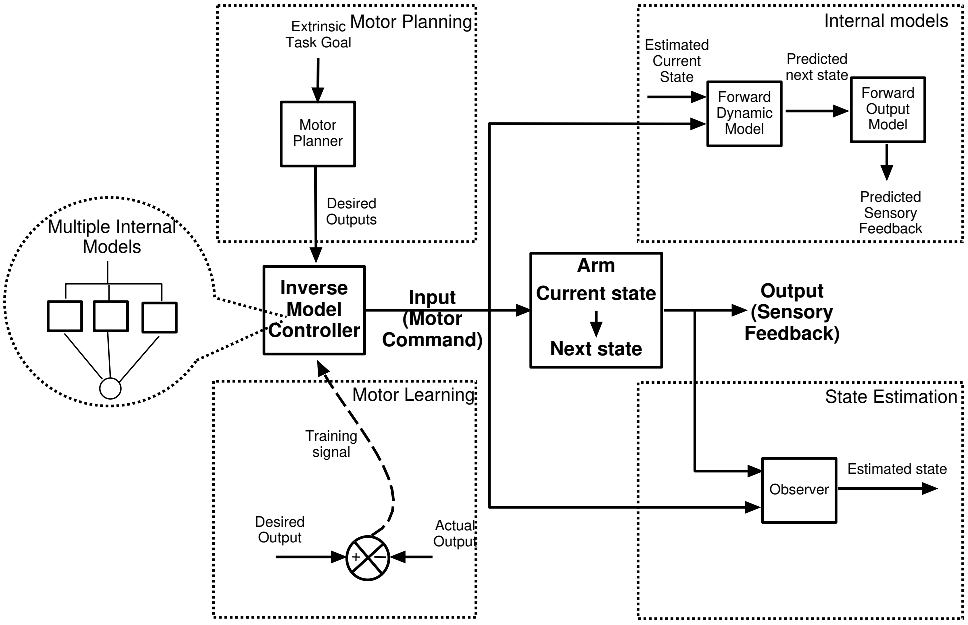
Figure 9. The feedback error learning approach to learning a feedforward controller. The feedback control signal is the error term for learning the feedforward controller.

Figure 10. The distal supervised learning approach. The forward model is trained using the *prediction error* $(\mathbf{y}[n] - \hat{\mathbf{y}}[n])$. The subsystems in the dashed box constitute the *composite learning system.* This system is trained by using the *performance error* $(\mathbf{y}^*[n] - \mathbf{y}[n])$ and holding the forward model fixed. The state estimate $\hat{\mathbf{x}}[n]$ is assumed to be provided by an observer (not shown).

Figure 11. A forward model for the archery problem. Given a value of u, the forward model allows the output y to be predicted, the error $y^* - y$ to be estimated and the slope to be estimated at u. The product of the latter two quantities provides information about how to adjust u in order to make the error smaller.

Figure 12. The mixture of experts architecture. Based on the state vector $\mathbf{x}$ the gating network assigns credit to the expert networks by calculating normalized mixing coefficients g_i. The expert networks implement a mapping from inputs $\mathbf{x}$ to outputs μ_i for the region of the state space in which they are appropriate.

Figure 13. The multiple paired forward-inverse model. Each inverse model is paired with a corresponding forward model which is used to convert performance errors into motor errors for training the inverse model. The ensemble of models is controlled by a "responsibility estimator" whose job it is to partition the state space into regions corresponding to the different local model pairs. Reprinted with permission from (Wolpert and Kawato, 1998)



Motor Planning
Extrinsic Task Goal
Motor Planner
Desired Outputs
Internal models
Estimated Current State
Forward Dynamic Model
Predicted next state
Forward Output Model
Predicted Sensory Feedback
Multiple Internal Models
Inverse Model Controller
Input (Motor Command)
Arm
Current state
Next state
Output (Sensory Feedback)
Motor Learning
Training signal
Desired Output
Actual Output
State Estimation
Observer
Estimated state

Figure 1

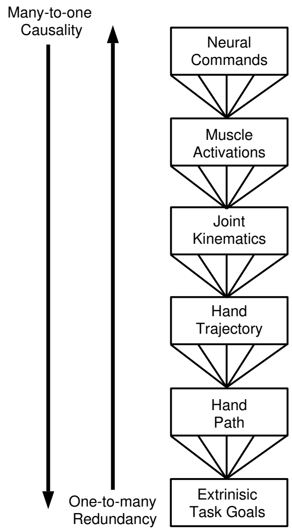

Many-to-one
Causality
Neural
Commands
Muscle
Activations
Joint
Kinematics
Hand
Trajectory
Hand
Path
One-to-many
Redundancy
Extrinisic
Task Goals

Figure 2

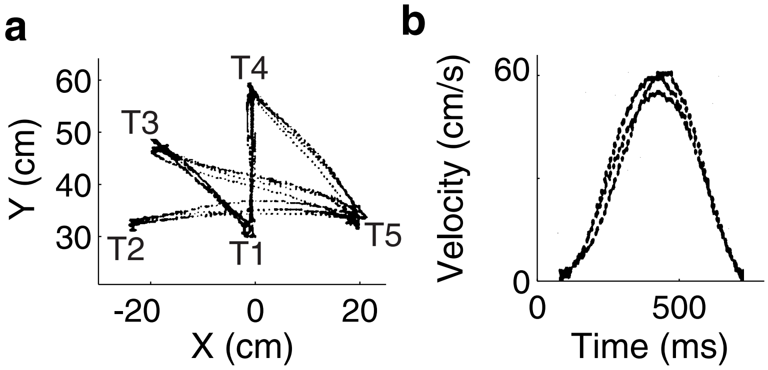

a
b
T4
T3
T2
T1
T5
Y (cm)
60
50
40
30
-20
0
20
X (cm)
Velocity (cm/s)
60
0
0
500
Time (ms)

Figure 3

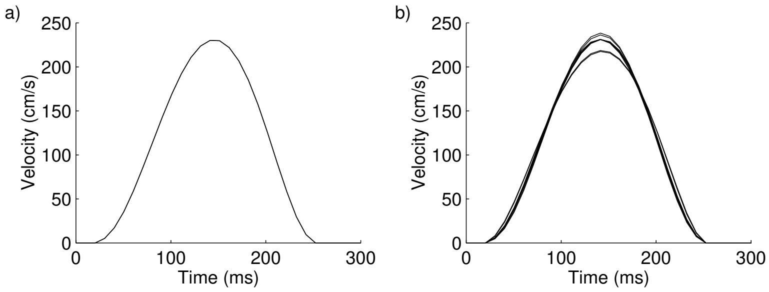

a)
250
200
150
100
50
0
Velocity (cm/s)
0
100
200
300
Time (ms)
b)
250
200
150
100
50
0
Velocity (cm/s)
0
100
200
300
Time (ms)

Figure 4

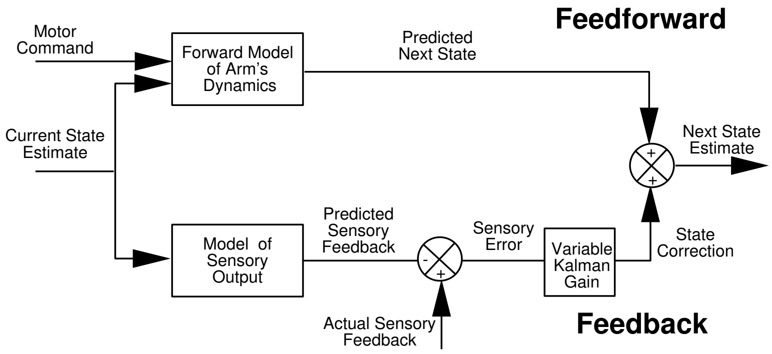

Feedforward
Motor Command
Forward Model of Arm's Dynamics
Predicted Next State
Current State Estimate
Next State Estimate
+
+
Model of Sensory Output
Predicted Sensory Feedback
-
+
Sensory Error
Variable Kalman Gain
State Correction
Actual Sensory Feedback
Feedback

Figure 5

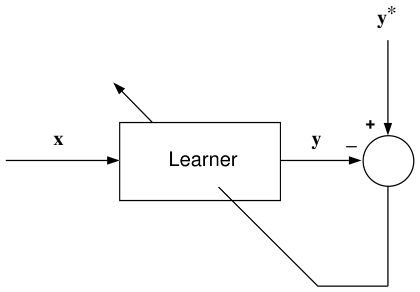


Figure 6

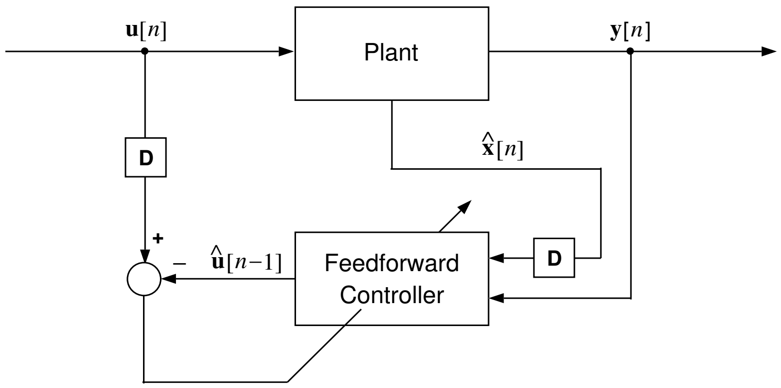


Figure 7

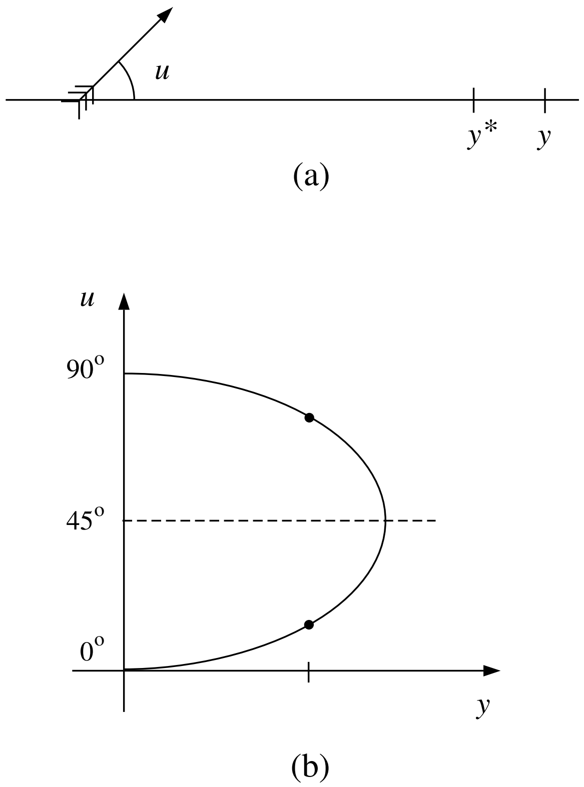


Figure 8

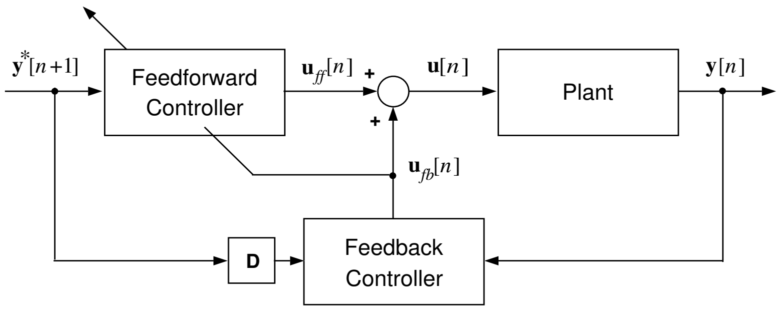

y*[n+1]
Feedforward Controller
u_ff[n]
+
u[n]
Plant
y[n]
+
u_fb[n]
D
Feedback Controller

Figure 9

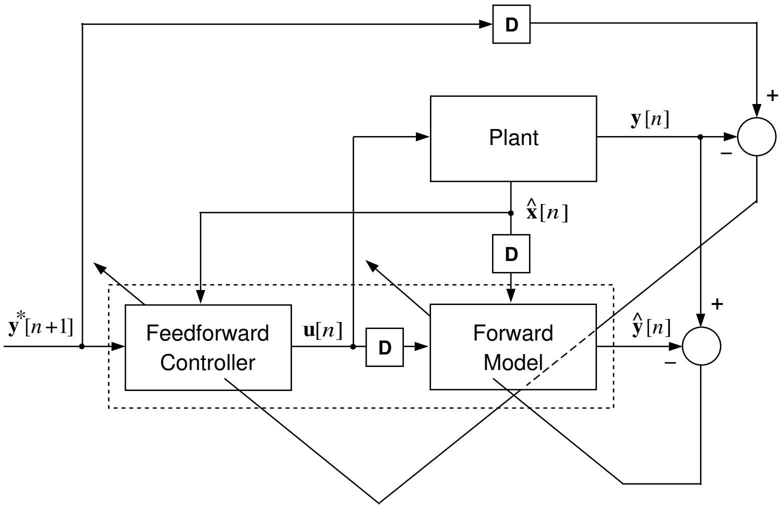

D
Plant
y[n]
+
−
x̂[n]
D
y*[n+1]
Feedforward Controller
u[n]
D
Forward Model
ŷ[n]
+
−

Figure 10

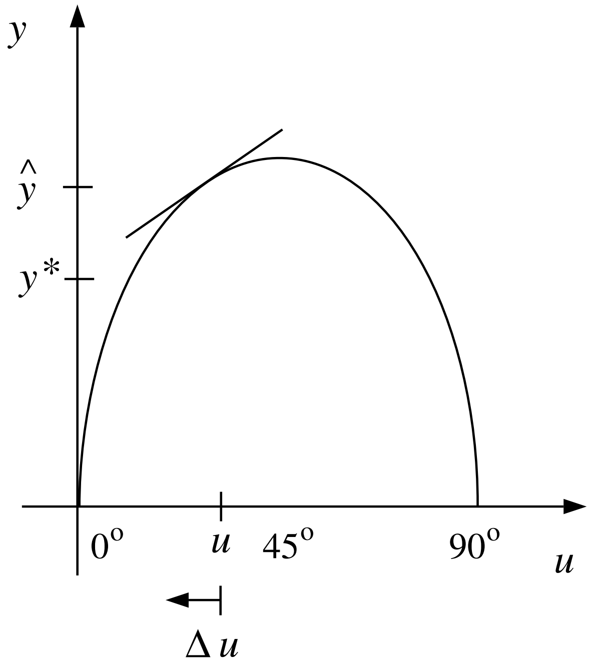


Figure 11

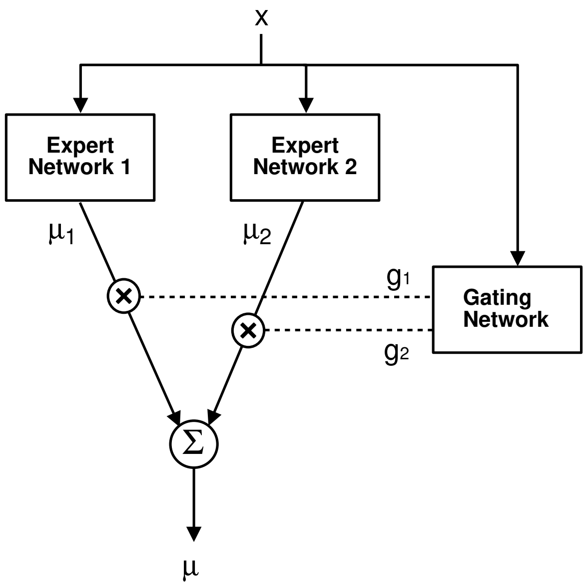

x
Expert Network 1
Expert Network 2
μ1
μ2
g1
g2
Gating Network
Σ
μ

Figure 12

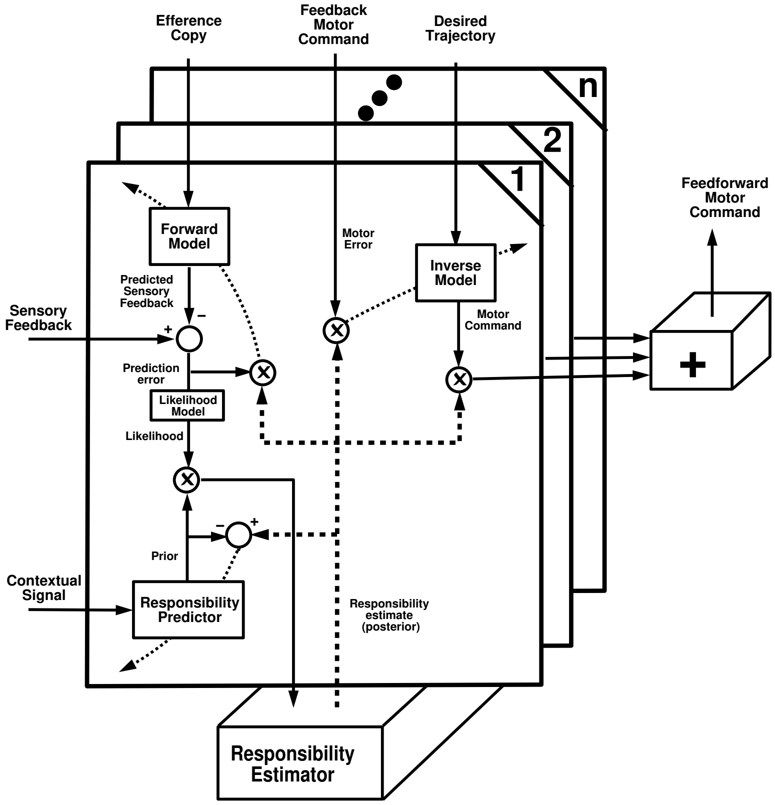

Efference Copy
Feedback Motor Command
Desired Trajectory
n
2
1
Forward Model
Motor Error
Inverse Model
Feedforward Motor Command
Predicted Sensory Feedback
Sensory Feedback
Motor Command
Prediction error
Likelihood Model
Likelihood
Prior
Contextual Signal
Responsibility Predictor
Responsibility estimate (posterior)
Responsibility Estimator

Figure 13