

Reaching to Grasp With a Multi-Jointed Arm. I. Computational Model

ELIZABETH B. TORRES¹ AND DAVID ZIPSER²

¹*Division of Biology, California Institute of Technology, Pasadena 91125;* ²*Department of Cognitive Science, University of California San Diego, La Jolla, California 92093-0515*

Received 15 January 2002; accepted in final form 24 July 2002

Torres, Elizabeth B. and David Zipser. Reaching to grasp with a multi-jointed arm. I. Computational model. *J Neurophysiol* 88: 2355–2367, 2002; 10.1152/jn.00030.2002. The generation of goal-directed movements requires the solution of many difficult computational problems. Among these are transformations from extrinsic to intrinsic reference frames, specifying solution paths, removing under-specification due to excess degrees of freedom and path multiplicity, constraint satisfaction, and error correction. There are no current motor-control computational models that address these issues in the context of realistic arm movement with redundant degrees of freedom. In this paper, we conjecture there is a geometric stage between sensory input and physical execution. The geometric stage determines movement trajectories independently of forces. It uses a gradient technique that relies on the metric of the space of postures to resolve endpoint path selection, posture-change specification, error correction, and multiple constraint satisfaction on-line without preplanning. The model is instantiated in an arm with seven degrees of freedom that moves in three-dimensional space. Simulated orientation-matching movements are compared with actual human movement data to assess the validity of several of the model's behavioral predictions.

INTRODUCTION

Voluntary movements such as reaching and grasping are easy to do, but the computations and brain processes involved are hard to explain. Complicated problems of this kind can be simplified by decomposing them into sub-tasks. Most studies assume a division into sensory and motor stages, but further decomposition is needed to fully explain movement generation and to map it to specific brain areas. In this paper, we propose an additional stage that computes movement paths independent of forces or the speed of movement. We call it the “geometric stage.” The input to the geometric stage consists of a description of the target for movement and of the current posture. Its output represents posture movement paths.

Converting movement paths into actual limb movements requires additional computation. Here we refer to the systems that do these computations collectively as the “motor implementation stage.” The task of this stage is to implement the movement paths specified by the geometric stage. This implies that observed movement paths are independent of speed and are determined by the geometric stage without reference to motor implementation. We leave the issue of the temporal coupling between the geometric and motor stages to the discussion.

A major advantage of introducing a geometric stage is that

difficult problems such as excess degrees of freedom, path under-determination, error correction, and constraint satisfaction can be solved geometrically in a way that is biologically plausible and that simplifies the job of the motor implementation stage. We provide an explicit computational model of the geometric stage. The model's behavior is illustrated by simulating a seven-degree-of-freedom arm making reach to grasp movements in three dimensions. Comparison of this behavior with experimental results demonstrates the biological plausibility of the model.

Previous work

The motivation for introducing a separate geometric stage is the difficulty of solving the problem of under determinacy inherent in specifying the movements of many jointed limbs (Bernstein 1967). Two important sources of movement under determinacy are excess degrees of freedom and path redundancy. If there are more degrees of freedom in the limb than in the space in which it moves, there are infinitely many postures corresponding to each hand location and orientation. There are also infinitely many paths to the same goal whether or not there are excess degrees of freedom. It remains unknown how the brain generates the possible solution paths or how it chooses one path from all the possible ones.

These problems are studied in fields other than biological motor control such as robotics and computer animation, and techniques from these fields have proved useful. The excess degree-of-freedom (df) problem in robotics is often solved using a technique called resolved motion rate control (Whitney 1969). This method assumes that the desired hand trajectory is known. It then uses some optimization method, such as minimizing kinetic energy, to find a corresponding posture trajectory. This solves a subset of the reach to grasp problem, but the need for a hand trajectory and the complexity of kinematics computations make this approach unattractive from a biological point of view. Our model uses a computational strategy that comes from computer animation (Witkin et al. 1987) but modifies it to fit the task of reaching to grasp and to be biologically plausible. This is described in detail later. First we review some previous work.

The most common computational approach in biological motor control has been to consider the problem as analogous to one in classical mechanics. In this formulation, the trajectory that minimizes an energy-like quantity must be found. This

Address for reprint requests: E. B. Torres, Div. of Biology, California Institute of Technology, 333 Beckman Labs, M/C 216-76, Pasadena, CA 91125 (E-mail: etorres@vis.caltech.edu).

The costs of publication of this article were defrayed in part by the payment of page charges. The article must therefore be hereby marked “advertisement” in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

typically involves integrating a complicated function along all possible trajectories to find the one that is optimal according to some preset criterion. These integrals generally require that the final posture and the elapsed time be known prior to movement, making the computations daunting when the tasks are underdetermined. Because of this, almost all the work has been done with simplified systems having 2 df and moving in a plane, i.e., with no excess degrees of freedom. Under very simplified conditions, the minimizing integral equations can be solved, but in most cases they cannot and many numerical integrations are required to find the minimizing path. Models of this general type such as minimum jerk (Flash and Hogan 1985), minimum torque (Uno et al. 1989), minimum metabolic energy (Alexander 1997), and minimum variance (Harris and Wolpert 1998) have played a useful role in the study of planar arm movement.

Sometimes the problem can be simplified and realistic solutions found for multi-jointed arms in three dimension (3D). For example, Soechting et al. (1995) address the task of finding a realistic posture for a specified target hand location starting from different initial postures. They use an arm with 4 df, i.e., one extra degree of freedom, and make a series of reasonable assumptions about the geometry and physics of arm movements. The most realistic final postures were found to be the ones that minimize peak work. A search is required, using the simplex method, to find the posture that minimizes peak work. These postures are found to correspond quite closely to those found experimentally. This is a dynamic model since it deals with forces. It ignores path generation, and the computations involved do not seem to be biologically plausible.

The idea of separating geometry from dynamics in biological motor control was broached by Hinton (1984) but has not been widely used despite experimental data showing that there are brain areas with force-independent representations of movement (Kalaska et al. 1990).

Wang and Verriest (1998) do make clever use of the geometry of the arm to solve the same problem as Soechting et al. (1995). Realistic postures are achieved apparently because joint angular velocities are minimized. This model does take joint angle limits into account but requires tests that may not be biologically plausible. The algorithm, like most inverse kinematics algorithms, needs to be supplied with a path. A straight-line path is used as a default, but realistically curved arm paths also give good results.

A full review of the vast literature in this field is beyond the scope of this paper. However, we feel that explaining the experimental data on unconstrained arm motions may require a new strategy: one that is able to find complex paths, correct errors on-line, and satisfy multiple constraints in a biologically plausible way when the arm has redundant degrees of freedom. There is experimental evidence showing that reaching paths remain invariant under moderate changes in such physical variables as speed and loading (Atkeson and Hollerbach 1985; Nishikawa et al. 1999). This suggests that geometry might be a basic component in guiding the movement.

Geometric stage

The geometric stage is a computational module between sensory input and motor output. It transforms information about the goal of a movement, such as an object's location and

orientation, into a postural path that brings the hand to the target in the correct configuration to accomplish the goal. For example, grasping an object. The geometric stage requires information about the current posture as well as about the target.

The postural path output by the geometric stage is represented in terms of variables that unambiguously describe postures and can be controlled by the motor implementation stage. It is not yet known just what postural variables are controlled by the motor system. Joint angles and muscle lengths are possibilities. The computational strategy we propose here for the geometric stage will work with any set of postural variables. There are two important features of the geometric stage to keep in mind. First, it computes the geometrical aspects of movement but not the forces required to bring these changes about. And second, it specifies movement paths, independent of the speed of movement. If our conjecture about the geometric stage is correct, then movement paths should be independent of speed, and they should be accurately accounted for by a model of the geometric stage without reference to motor implementation.

The transformation from an extrinsic description of a goal for movement to a postural path is underdetermined. This means that an optimization technique must be used to compute it. The computations used should be biologically plausible as well as able to solve the excess df problem, select realistic postural paths, and respect joint angle limits. Because we don't know the actual variables controlled by the motor system, it should also be possible to parameterize the transformation, using experimental data, so that veridical paths are generated. We have found that a modified form of a gradient descent strategy, originally developed for computer graphics (Witkin et al. 1987), has all these properties so we use gradient descent to model the geometric stage.

Gradient descent requires a function that maps all the variables specifying both the current posture and the target for movement to a positive scalar that decreases as the posture approaches its target. Various disciplines use different names for this kind of function such as *cost*, *error*, *principle*, and *energy* function. We will use *cost* function. The distance remaining between the current hand configuration, and its target is a positive scalar that decreases to zero as the target is approached. We use this distance as the scalar for the cost function in the geometric stage. The gradient of the cost function is a vector whose components specify the relative amount to change each postural variable to get the hand closer to the target position and orientation. The gradient descent procedure for the reach-to-grasp task consists of repeatedly computing the gradient of the cost function and changing the posture a small amount each time until the hand is at the target in the correct orientation. In this way, it generates a postural path from the initial to the target posture.

We illustrate the gradient technique using an arm with 7 df that does reach-to-grasp movements in three dimensions. For simplicity, we first use a posture configuration space consisting of a particular set of joint angles. There is no a priori reason to think that the paths generated by this choice of posture configuration space will give realistic movement paths, and, in fact, they do not. Later we show how to generalize the gradient method to respect joint limit constraints and use any set of postural variables. In particular we use parameters obtained

from experimental observation to generate realistic paths with the model.

Model: computing the gradient

The model deals with two phases of grasping: *transport*, moving the hand to the object, and *orientation*, rotating the hand to match the object axis. Therefore the cost function has to map joint angles to a single value that combines distance to the target with the difference in orientation between hand and target. We first describe using gradient descent for transport.

Consider an arm consisting of constant length segments linked end to end by n flexible joints. One end of the arm is attached to a fixed point in a 3D reference frame and the other end, i.e., the hand, remains unconnected, Fig. 1. Let X , the workspace, be the set of points that can be reached by the hand in 3D space. Let Q , the configuration space, or specifically here, the joint angle space, be an open subset of R^7 . The components of Q are any set of variables, i.e., joint angles, which can specify all arm *postures*. For our example we use the Euler angle sequence given in Fig. 1 to describe a 7 df arm.

There is a vector valued function, f that maps Q onto X , i.e., every hand location $x \in X$ can be written as $f(q)$ for at least one $q \in Q$, and every q maps to some $x \in X$

where

$$x = (x_1, x_2, x_3)$$

$$q = (q_1, \dots, q_7)$$

$$f(q) = (f_{x_1}(q), f_{x_2}(q), f_{x_3}(q))$$

Because the arm segment lengths are considered constant, we ignore them in the notation used here. We are interested in cases where the joint angles are limited to their physiological ranges and $n > 3$, i.e., there are redundant degrees of freedom. The function f for the 7-jointed arm in Fig. 1 can be found in the literature (Benati et al. 1980).

A path in Q is a directed line that connects two postures q^{initial} and q^{final} . The path in X traced by the hand corresponding to a path in Q is called its image in X . A solution to the transport problem is obtained by moving along the path given by the negative of the gradient of the cost function

$$dq = -\nabla r(q^{\text{initial}}, x^{\text{target}}) dt \tag{1}$$

Where $r(q^{\text{initial}}, x^{\text{target}})$ is the cost function representing the distance in X between $f(q)$ and x^{target} and dt is a scalar parameter. The cost, $r(q^{\text{initial}}, x^{\text{target}})$, is a positive function with the unique global minimum *zero*, which occurs at the target point. The cost, $r(q^{\text{initial}}, x^{\text{target}})$, can be written explicitly using f

$$r(q^{\text{initial}}, x^{\text{target}}) = \sqrt{\sum_{i=1}^3 (x_i^{\text{target}} - f_i(q))^2} \tag{2}$$

The gradient of $r(q^{\text{initial}}, x^{\text{target}})$ is given by the chain rule

$$\nabla r(q^{\text{initial}}, x^{\text{target}}) = \nabla r(x, x^{\text{target}}) J(f(q)), (J = \text{Jacobian}) \tag{3}$$

Equation 1 is in the form of a dynamical system that specifies both the direction and relative speed of movement along the solution path. However, we are only interested in the solution path itself. This path is invariant with respect to speed. The speed given by Eq. 1 is not relevant to the speed of limb movement, so when computing movement paths, we generally use a normalized version of the gradient that gives a constant speed.

Next we consider using gradient descent to reduce the difference in orientation between the hand and the target object. We need to find a single value that decreases with this difference in orientation and is a function of q and the orientation of the target. Our strategy is to first find the matrix that represents a rotation that will align the target with the hand. This rotation can then be represented by a single angle ϕ about the principal axis \vec{n} (Altman 1986). This angle can be used in the cost function. The general scheme is given in the following text.

Solving the orientation-matching problem

Consider three reference frames: one fixed at the shoulder S , one on the hand H , and one on the object O , Fig. 2. The relative orientation of these frames can be represented with orthogonal rotation matrices: H for the rotation from shoulder to hand, O from shoulder to object, and $R = O^{-1}H$, from object to shoulder to hand, where $O^{-1} = O^T$ is the rotation from object to shoulder. Note that H depends on all seven joint angles of the posture, q , while O is given as input to the geometric stage. The matrix R represents the orientation difference we want to

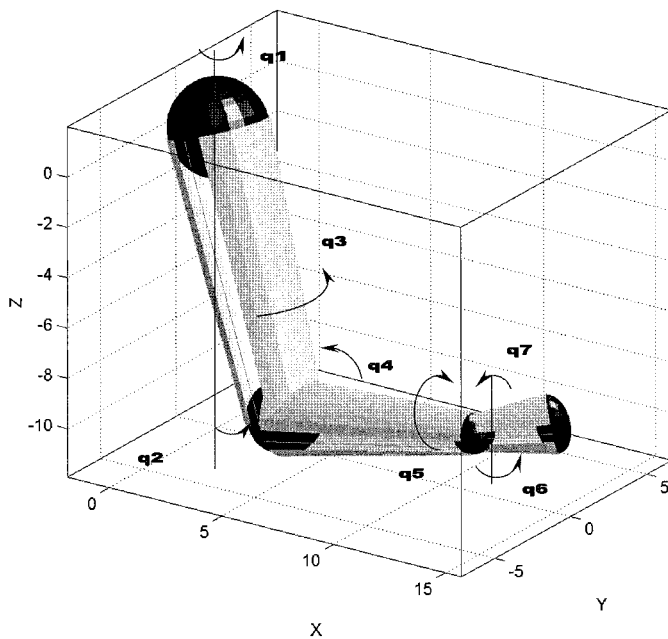


FIG. 1. The arm is modeled as a system of 3 rigid-body segments. The Euler-Angle sequence XYZ is used to model a 7 df arm. The shoulder has a ball-and-socket joint with 3 df, abduction-adduction (q_1), flexion-extension (q_2), and pronation-supination (q_3). The elbow joint has 2 df, flexion-extension (q_4) and pronation-supination (q_5). The wrist joint also has 2 df, abduction-adduction (q_6) and flexion-extension (q_7). There is a fixed frame at the shoulder and 3 moving frames, 1 at each joint. The position and orientation of each frame are described relative to the previous frame by $(R^s, [0,0,UL]^s)$, $(R^e, [0,0,FL]^e)$, and $(R^w, [0,0,HL]^w)$. The kinematics chain is used to describe position and orientation of each joint relative to the fixed shoulder frame by: (R^s, S) , $(R^s \cdot R^e, E)$, $(R^s \cdot R^e \cdot R^w, W)$ where endpoint positions at shoulder, elbow, wrist, and hand are given relative to the shoulder fixed frame by: $S = [0,0,0]^s$, $E = S + R^s(q_1, q_2, q_3) \cdot [0,0,UL]^s$, $W = E + R^s(q_1, q_2, q_3) \cdot R^e(q_4, q_5) \cdot [0,0,FL]^e$, $H = W + R^s(q_1, q_2, q_3) \cdot R^e(q_4, q_5) \cdot R^w(q_6, q_7) \cdot [0,0,HL]^w$.

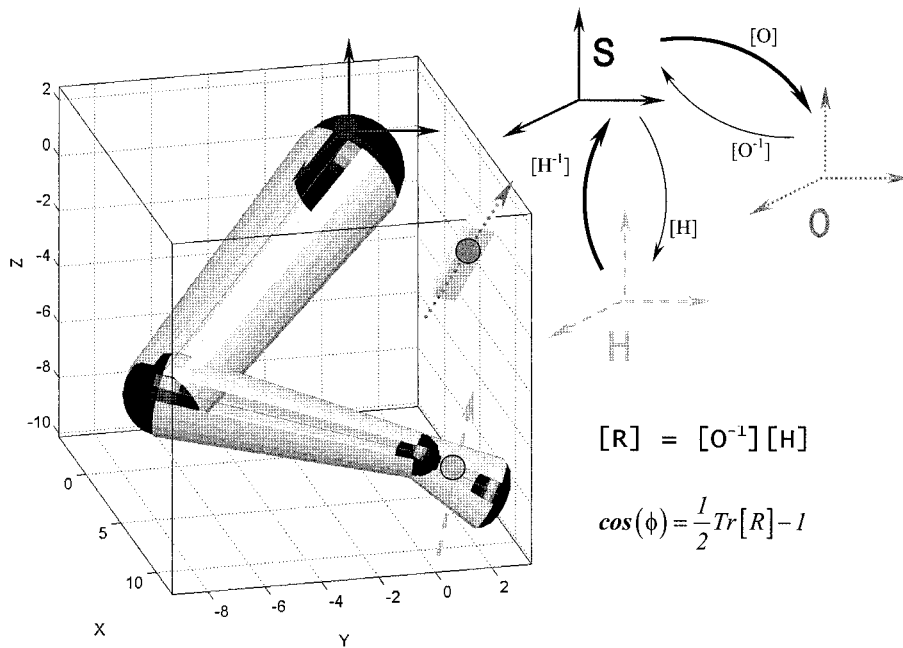


FIG. 2. Matching the orientation of the hand to that of the target for a successful grasp: the matrix that represents a rotation that will align the target with the hand is given by $R = O^{-1} \cdot H$, where O represents the given target orientation, and the hand's orientation is represented by $H(q) = R^x(q_1, q_2, q_3) \cdot R^y(q_4, q_5) \cdot R^z(q_6, q_7)$. The rotation encoded in R can be represented by a single angle ϕ about the principal axis. This angle ϕ is used in the cost function. When $\phi = 0$, the hand matches the orientation of the object.

reduce and depends on both the current posture and the object's orientation.

To get a single scalar value for the orientation difference, we use the fact that for any rotation matrix, such as R , there is a principal axis \vec{n} about which a single rotation ϕ has the same effect as the rotation matrix. A cost function for reaching-to-grasp can be obtained by combining ϕ with the distance term for transport derived in the preceding text

$$r = \sqrt{\sum_{i=1}^3 (x_i^{\text{target}} - f_i(q))^2 + \alpha(\phi(q, O))^2} \quad (4)$$

Where α is a parameter that can be adjusted so that arm movement and hand rotation overlap realistically in time as the arm moves toward the target, and $\cos(\phi) = \frac{1}{2}\text{Trace}(R) - 1$. In our simulations, we set the parameter α to a constant chosen so that the transport and orientation errors decrease proportionally. However, whether these errors change independently or not remains an open question that needs to be addressed experimentally.

It is important to note that each term in Eq. 4 depends on all seven joint angles. This means that the wrist and upper arm are *not* treated as separate modals. Posture changes are made in a way that simultaneously changes hand position and orientation. This is important because rotation of any joint angle can potentially change either the location or the orientation of the hand (Jeannerod 1988).

The gradient descent algorithm starts at q^{initial} , takes a small step in the direction of the negative gradient to a new point q^{current} . This procedure is repeated using successive values of q^{current} until the value of r goes to zero, at which point x^{current} equals x^{target} , and the hand orientation matches the object's. In practice, computation is stopped when r is less than some small tolerance. Ideally the steps should be infinitesimal, but the gradient is approximated quite well using moderately large steps. An important feature of the gradient paradigm is that the trajectories obtained in Q provide a complete description of all

arm configurations as movement proceeds from initial to final posture.

An important property of Eq. 4 is that r goes to zero if and only if the hand is at the target in the correct orientation. This enables the system to avoid the problem of local optima that can plague gradient descent. If the gradient of r goes to zero and r is not zero, then a perturbation can be made in the trajectory and gradient descent continued until r is zero. This is not a trivial feature of the exact form of the cost function (Eq. 4). Rather it comes from the fact that distance and orientation difference are geometrical properties of the system that go to zero only when the reach-to-grasp task is complete.

Generalization of the model

Generalizing the model enables it to produce realistic postural paths and obey constraints such as joint angle limits. Generalization is based on the fact that the paths traced out by the gradient depend on how posture configuration space is represented. Thus by transforming configuration space, we can get veridical gradient paths.

Our initial description of the gradient method used joint angles as axis of a Cartesian system to represent posture configuration space. This representation was chosen for illustrative simplicity. We now explore how to transform this Cartesian system to generate arbitrary gradient paths. We can do this either by transforming configuration space and then computing a new gradient or by computing the original gradient and then operating directly on the gradient vector.

These techniques are illustrated by using the gradient method to generate paths for the tip of an arm with only two joint angles moving in a plane. In this system, there is no excess df. However, the paths are underdetermined. We first find the path using a posture configuration space with each joint angle as an axis of a Cartesian coordinate system. This gives curved paths. Then we transform posture configuration space so that the paths are straight lines.

The arm is modeled with two unit-length segments and two joints, q_1 at the shoulder and q_2 at the elbow. The forward function $x = F(q)$ in Cartesian configuration space is given by

$$f_{x_1}(q) = \cos(q_1) + \cos(q_1 + q_2)$$

$$f_{x_2}(q) = \sin(q_1) + \sin(q_1 + q_2)$$

Where $[f_{x_1}(q), f_{x_2}(q)]$ represent the coordinates of the arm's endpoint.

The translation cost function in this case is

$$r(x^{\text{target}}, q) = \sqrt{(x_1^{\text{target}} - f_{x_1}(q))^2 + (x_2^{\text{target}} - f_{x_2}(q))^2}$$

A typical curved arm tip path produced by the gradient of r projected onto X is shown in Fig. 3A. The gradient path on the cost surface is shown in Fig. 3B.

If we want the tip of the arm to always follow straight lines we can, find a transformation, $\tilde{q} = \tilde{G} \cdot q$, from the original to new coordinates such that the gradient in the new system gives straight paths in X . The gradient with respect to the new coordinates is

$$\tilde{\nabla} r(x^{\text{target}}, q) = \tilde{G}^{-1} \cdot \nabla r(x^{\text{target}}, q)$$

Note that the new gradient is obtained by premultiplying the original gradient by the inverse of the transformation matrix. This matrix must be positive definite, but is otherwise not restricted. If \tilde{G}^{-1} is symmetric, the transformation is called "metric" and has nice mathematical properties. However, we have not been able to account for all of our data using symmetric positive-definite metric transformations. In most cases, it has proved more intuitive to construct \tilde{G}^{-1} directly using knowledge about the desired constraints or experimental data. In the present example, however, \tilde{G} can be found analytically (Gray 1998) and is given by

$$\tilde{G}(q) = J(f(q))^T \cdot G(f(q)) \cdot J(f(q))$$

$$= \begin{bmatrix} 2(1 + \cos(q_2)) & 1 + \cos(q_2) \\ 1 + \cos(q_2) & 1 \end{bmatrix}$$

$$G(f(q)) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

An example of straight-line movement in X and the corresponding gradient path on the new cost surface are given in Fig. 3, C and D.

In the following sections, we use the techniques for generalizing the gradient to develop a more realistic model of the geometric stage.

Constraint satisfaction in goal-directed motions

There are many constraints imposed on arm movements. Some limit joint rotation others save effort by biasing the upper arm to remain closer to its resting posture. We implement these constraints using the properties of the gradient operator described in the preceding text.

The general idea is to premultiply the gradient by a positive-definite matrix, denoted by C instead of \tilde{G}^{-1} , to represent the constraints. This yields a new gradient vector, dq^C , guaranteed to decrease the original cost because positive-definite implies that $dq \cdot C \cdot dq^t > 0$ so the new gradient vector dq^C is within 90° of the old dq . In addition, $dq^C = C \cdot dq = 0$ if and only if $dq = 0$, so all minima of the cost function are still minima, and no new minima are introduced.

Constraints are implemented using diagonal C 's. Each non-zero component of C acts on a single joint angle because it affects only one of the components of the gradient vector. The constraints use sums of products of logistic functions that

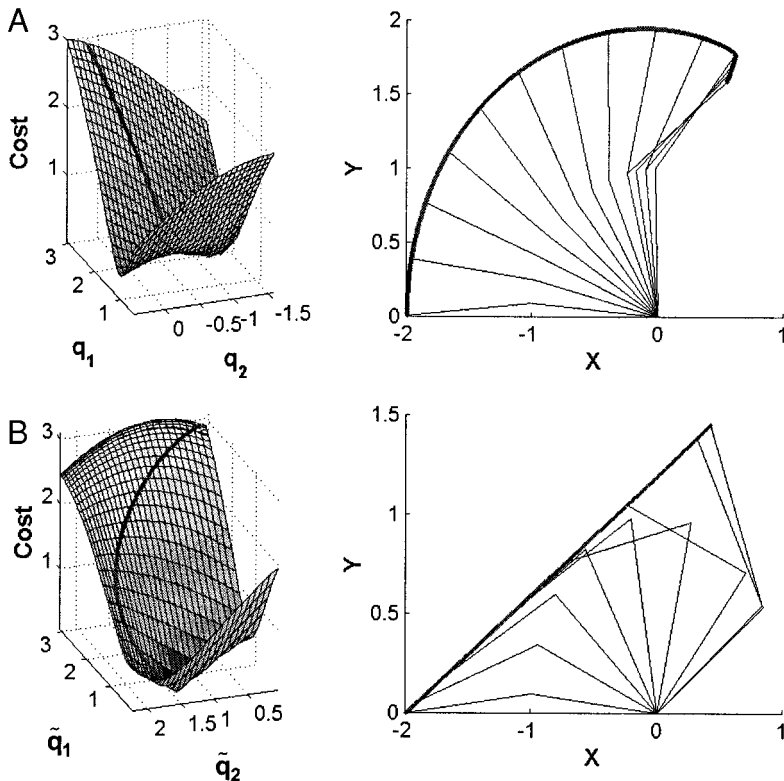


FIG. 3. Change of coordinates. A: extrinsic hand of 2 segment arm resulting from using the gradient in Cartesian joint angle space. B: cost surface corresponding to A with the gradient path shown on the surface. C: extrinsic hand of arm resulting from using the gradient in a transformed space that gives straight hand paths. D: cost surface corresponding to C with the gradient path shown on the surface

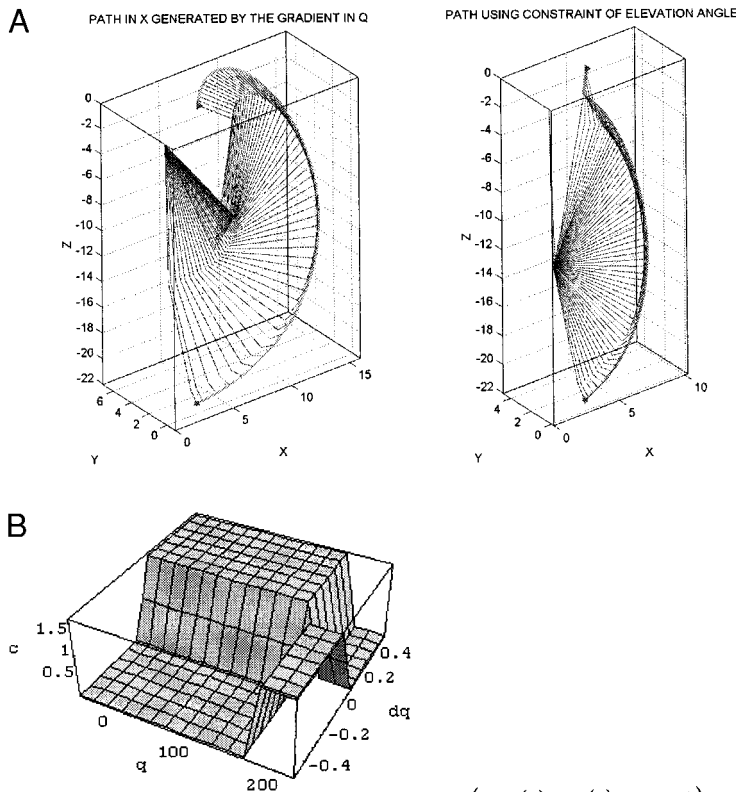


FIG. 4. Constraining the elevation angle to move efficiently (in the minimum effort sense). A: compares the paths in 3-dimensional (3D) space produced by the “plain gradient” (left) vs. the corrected gradient (right). B: the scalar function $c[dq_i(t), q_i(t), x^{target}]$ and the surface it spans for $q_i \in [0, 180^\circ]$, $k = 25$ $dq_i \in [-0.5, 0.5]$. The function essentially “speeds up” movements toward rest and “slows down” movements away from rest at the shoulder joint. In this parameterization, $q_{rest} = 180^\circ$, and the elevation angle has index $i = 2$. For any given posture q , where q_2 is between 0 and 180° , the 2nd term in the exponential will be negative. If the corresponding dq is negative, i.e., it “wants” to move the arm away from rest (toward 0°), the term within brackets in the exponential is positive and the negative exponential goes to 0, so the gradient component dq_2 is multiplied by a small positive constant $a^- = 0.05$. This slows down the rate of change of that joint. If the current increment dq_2 is positive, thus bringing the arm toward rest, the expression within the brackets in the exponential is negative. Therefore the exponential is a large positive value in the denominator that makes the second term in the sum go to 0. The dq_2 is multiplied by a large positive constant $a^+ = 1.75$, which increases the rate of change of that joint. The overall effect is a more human-like motion of the arm.

$$c_{ii}(dq_i(t), q_i(t), x^{target}) = a^+ + \frac{a^- - a^+}{1 + e^{-[(kdq_i)(q_i - q_{rest})]}}$$

approximate step functions. This allows the logical requirements of the constraint to be implemented using continuous functions.

Simulated constraints

SAVING EFFORT. The upper arm requires more effort to elevate than the more distal arm segments because it is heavier. Effort can be saved by discriminating against elevating the upper arm away from its resting position and enhancing movement back toward rest. The matrix C in this case has all diagonal components equal 1.0 except for the one associated with the elevation angle of the upper arm. This component is set to

$$a^+ + \frac{a^- - a^+}{1 + e^{-[(kdq_i)(q_i - q_{rest})]}}$$

Where a^+ and a^- are the enhancement and discrimination factor, respectively, $a^+ > a^- > 0$, and k is a constant that determines the sharpness of the transition between the a^- and a^+ states. All the other diagonal components of C are set to 1. Figure 4 depicts the

trajectories resulting from simulating motions using the gradient method under this constraint. The values of a^- , a^+ , and k were chosen to illustrate the effect clearly. With large values of k , the preceding expression is approximately equal to a^+ for movement down toward rest and to a^- for movements up away from rest.

MOVING COMFORTABLY. The limitations on joint angle flexion can also be expressed using a diagonal matrix, but now each diagonal component is a function that goes to zero when the associated change in joint angle moves it outside its allowed range and stays at one otherwise. To get such a function, we combine a function having partial derivatives of 1.0 in range and 0.0 out of range multiplied by a direction discriminating function of the kind used in the preceding text. An example is

$$c_{ii}^2(dq(t), q(t)) = \left(\frac{1}{(1 + e^{[kdq_i]})(1 + e^{[k(q_i^{max} - q_i)])})} + \frac{1}{(1 + e^{-[kdq_i]})(1 + e^{[k(q_i - q_i^{min})]})} \right) + \frac{1}{(1 + e^{-[k(q_i^{max} - q_i)])})(1 + e^{-[k(q_i - q_i^{min})]})}$$

TABLE 1. Average mean absolute deviations from the difference between model and data (6 subjects)

	Shoulder			Elbow		Wrist	
	Abduction	Flexion	Pronation	Flexion	Pronation	Abduction	Flexion
S1	0.0016	0.0039	0.0035	0.016	0.0044	0.0197	0.0189
S2	0.0015	0.0032	0.00032	0.0097	0.0035	0.0231	0.0270
S3	0.0019	0.0021	0.0030	0.0139	0.0034	0.0218	0.0219
S4	0.0010	0.0021	0.0018	0.0101	0.005	0.0283	0.0212
S5	0.0011	0.0014	0.0016	0.0083	0.0021	0.0112	0.0099
S6	0.0016	0.0016	0.0031	0.0137	0.0046	0.0201	0.0142

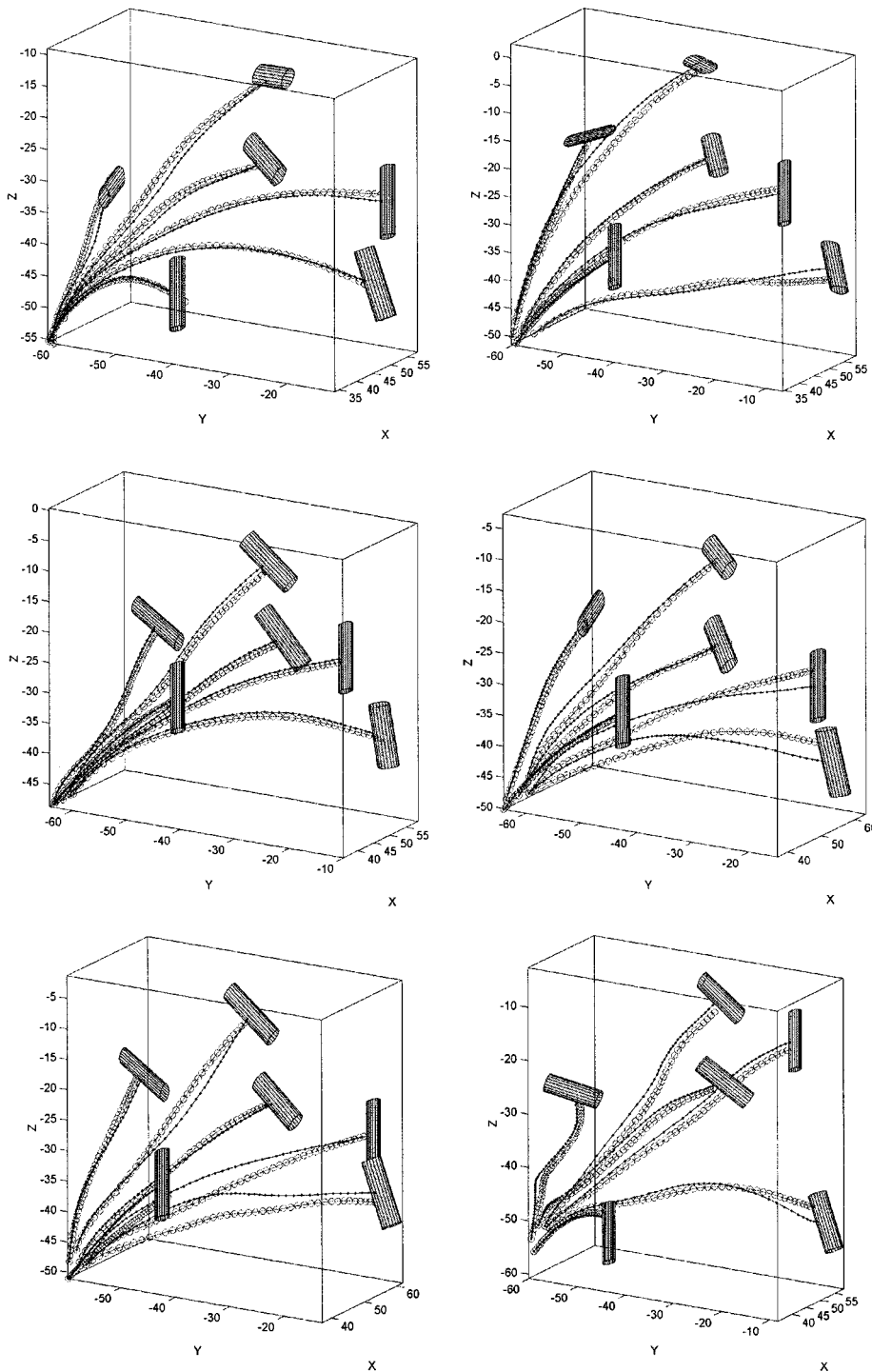


FIG. 5. Experimental example of metric correction: Unconstraint arm movement paths were recorded from 6 subjects who performed orientation-matching motions to 6 targets located throughout the reachable workspace and oriented differently. Circled paths are the subjects' averaged across 6 repetitions. Superimposed are the model's paths obtained after the change of metric under the original coordinates, or equivalently, after a coordinate transformation in Euclidean space. Table 1 shows the mean absolute deviations per joint for each subject averaged across all 6 targets. They were obtained in the following way: for each subject, per target, we compute the difference between the average data path (across 6 repetitions) and the corresponding model path, obtain the mean, subtract each value from it, take absolute values, and then find the mean of the result.

Where q_i^{\min} and q_i^{\max} are the limits of joint angle i flexion. Here a^+ is set to 1.0 and a^- to 0.0. The arm elevation and joint angle range constraint can be applied simultaneously by multiplying their matrices.

Fitting the model to experimental data

Realistic gradient paths can be obtained by fitting the matrix \tilde{G}^{-1} to experimental data. The task used to get data to demonstrate this was a reach-to-grasp movement with a hand-held

cylinder to match target position and orientation. The resulting seven-joint angle postural path data were recorded using the Polhemus Fastrak motion-tracking system (Torres 2001). Constant components for \tilde{G}^{-1} were found by optimization (APPENDIX A). In some cases, the paths could be fit with symmetrical \tilde{G}^{-1} ; but in general, this was not the case.

The matrix \tilde{G}^{-1} is computed across the entire data set for each subject. Table 1 lists the mean absolute deviations (in radians) of the difference between the data and the model for each subject. The results of simulating paths with the model

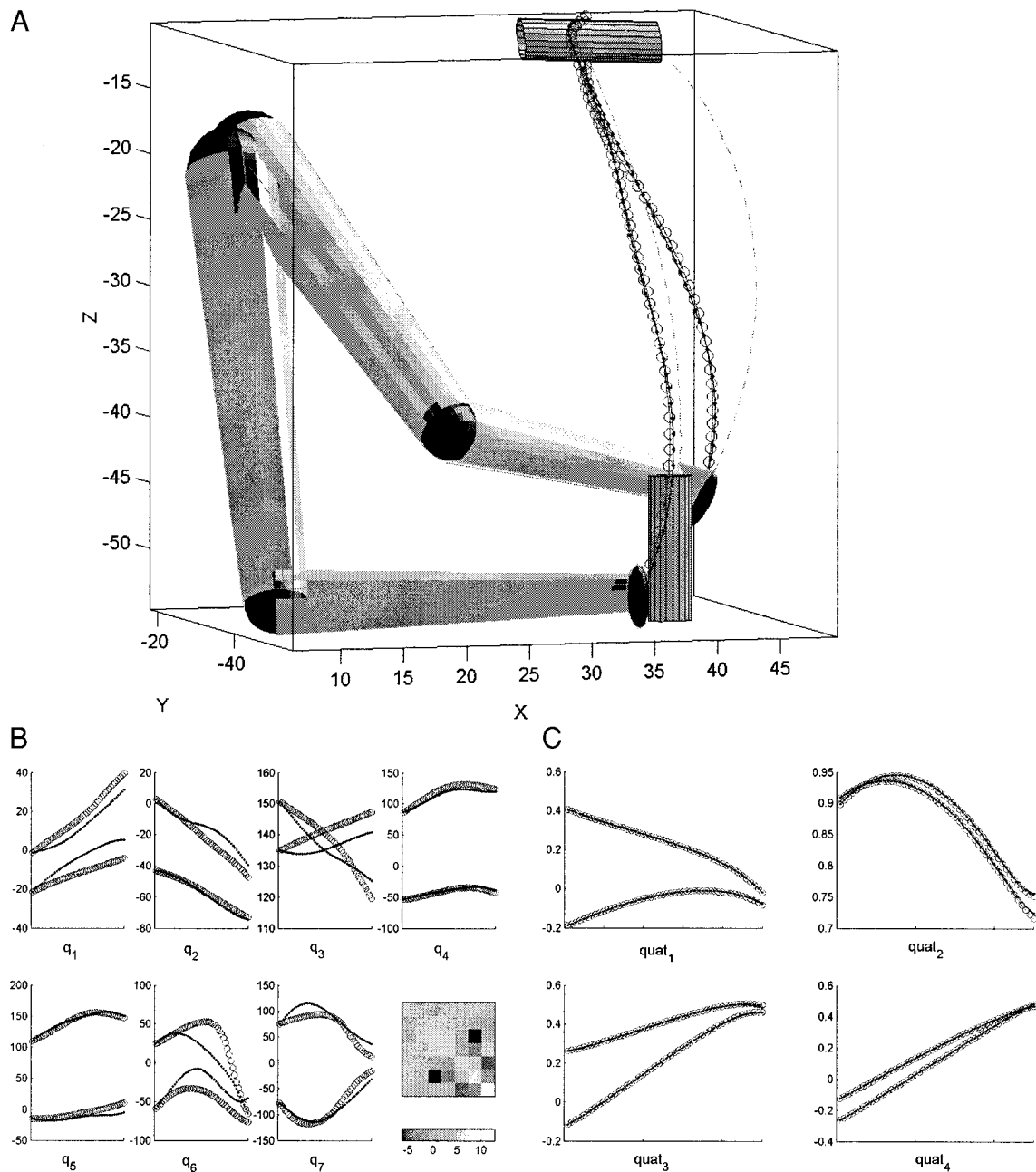


FIG. 6. Effect of starting posture on hand paths (A), posture paths (B), and hand orientation paths (C). Two different starting postures are used to make orientation-matching motions to the same target position and orientation. The model predicts a difference in postural paths and consequently on the hand position and orientation paths. We have confirmed this prediction experimentally (Torres 2001). In the figure, the data paths from 1 subject to 1 target (averaged over 6 repetitions) are plotted against the model-generated paths. The broken-line paths correspond to the gradient-generated path assuming Euclidean metric. The circled paths are from the hand position and orientation sensor data (the latter in quaternion form). The superimposed dotted-line paths are those obtained after the change of metric using $dq = -G^{-1} \cdot \nabla r(q^{\text{initial}}, x^{\text{target}}, o^{\text{target}}) \cdot dt$. In A and C, we relaxed the symmetric-positive-definite constraint in the computation of G^{-1} and computed the best fit in the least square sense. B and Table 2 show there is a larger error in the corrected paths when we constrained the matrix to be symmetric positive-definite. Note that in this case, the matrix has to account for 2 different postural paths in configuration space.

using the experimentally derived \tilde{G}^{-1} are superimposed on the actual data paths from the hand sensor are plotted in Fig. 5 The close correspondence between model and observed paths shows that the model can be parameterized to fit the data with a constant linear transformation of the gradient vector given by an arbitrary joint angle representation.

Comparison of the model with behavioral observations

The arm trajectories computed by the geometric stage will be reflected in observed behavior if, as we conjecture, the motor stage implements these trajectories. This means that various features of the trajectories computed using the gradient

TABLE 2. Average mean absolute deviations from the difference between model and data

Fitting Matrix	Shoulder			Elbow		Wrist	
	Abduction	Flexion	Pronation	Flexion	Pronation	Abduction	Flexion
Least square	0.0004	0.0007	0.0003	0.0012	0.0026	0.0105	0.0106
Symmetric positive definite	0.0023	0.0040	0.0041	0.0040	0.0053	0.0242	0.0200

One subject in the two different initial-postures experiment.

paradigm should be comparable to actual arm movements. For example, simulated orientation-matching motions using this method reflect the dependence of final posture on the starting posture and show the co-articulation of hand orientation with arm movement observed in reach-to-grasp movements (Desmurget et al. 1998; Soechting et al. 1995). Differences in posture paths and hand position and orientation paths due to two initial postures are shown in Fig. 6, A–C, respectively (see also Table 2). Simulated paths are superimposed on the actual paths from one of our subjects.

Paths resulting from movements from a single starting posture to targets located in the same position but oriented differently are shown in Fig. 7 (see also Table 3). The effect of target orientation on the position hand paths, the orientation hand paths, and the postural paths are shown in Fig. 7, A–C, respectively. They are also superimposed on the motion paths of one subject. A detailed analysis of these experiments will be presented elsewhere.

Error correction

There are many sources of error that can cause deviations from the desired movement path. An important task for any real sensory-motor system is to compensate for these errors. An advantage of the gradient method is that some of these errors are automatically corrected at the geometric stage. We simulate two sources of error due to perturbations. Figure 8A illustrates the hand position paths resulting from perturbations applied in configuration space. At each time step, two different levels of random noise are used to affect the incremental directional signal output by the gradient. Because the geometric stage continually receives input regarding the current posture and visual goal, the hand moves toward the target irrespective of where the perturbation has put it. The paths are deformed but still reach the target. The real paths observed from this kind of noise would be smoother due to inertia in the physical arm. In Fig. 8B, we simulate a sudden change in target position during the movement. The model also handles these kinds of on-line perturbations and achieves the target because it receives visual input related to the current goal position and orientation at each time step (Prablanc and Martin 1992).

Arbitrary trajectories

Real hand movements are often complex with many rapid changes in direction. Examples are paths that avoid obstacles and tasks such as drawing and writing. The gradient method makes it possible to generate these arbitrary trajectories without preplanning. This is because both the x^{target} and the o^{target} input to the geometric stage can be updated continually during movement. The path followed by the hand will closely approximate the changing target position and orientation in the input

stream. One source of this stream of targets could be the visual system. Generating a stream of x^{target} and o^{target} values from vision is not trivial, but it is a clearly defined problem for which the visual system is well endowed. The key point is that we have separated the task of generating a hand path in an extrinsic 3D reference frame from the job of finding intrinsic values e.g., joint angle changes, that can be used by the motor stage to implement movement. Figure 9 gives an example of a complex hand path generated using this method.

DISCUSSION

A major advantage of treating geometry independently of forces is that differential optimization techniques that are relatively simple to implement in a biologically plausible way can be used. This allows incremental changes to be computed as movement proceeds and eliminates the need to compute the whole trajectory ahead of time. This facilitates on-line error correction, consistent with experimental observations (Castiello et al. 1998; Desmurget and Prablanc 1997; Prablanc and Martin 1992), and simplifies complex tasks that require rapidly changing trajectories such as obstacle avoidance and writing. Models that use a differential approach are not new to motor control (Bullock and Grossberg 1988; Hoff and Arbib 1993). However, we specifically address the problem of redundancy and extend our model to handle the issue of coordinate representations in relation to task constraints.

The gradient method used to implement the geometric stage uses only continuous functions to compute paths, correct errors and enforce constraints. This is biologically significant because simple neural networks can compute these functions. We have trained feed forward networks to do the gradient computation and found that the model neurons in these network have response properties very similar to those of real neurons in brain areas involved in arm movements (Torres 2001). This shows that the gradient method is consistent with current experimental data.

A candidate for the location of the geometric state in the brain is posterior parietal cortex. This area serves as an interface between the sensory and motor systems, and it is known for playing a central role in transforming information from sensory to motor reference frames. In particular, reaching experiments using loads find populations of neurons in this region with force-free movement representations, in contrast to M1 populations that are generally modulated by force (Kalaska et al. 1990).

The temporal coupling between the geometric stage and motor implementation stage is an important issue for the biological plausibility of the model. The geometric stage can supply the motor stage with a continuous stream of joint angle changes. This could account for rapid responses to unanticipated perturbations during movement. Compensation for feed-

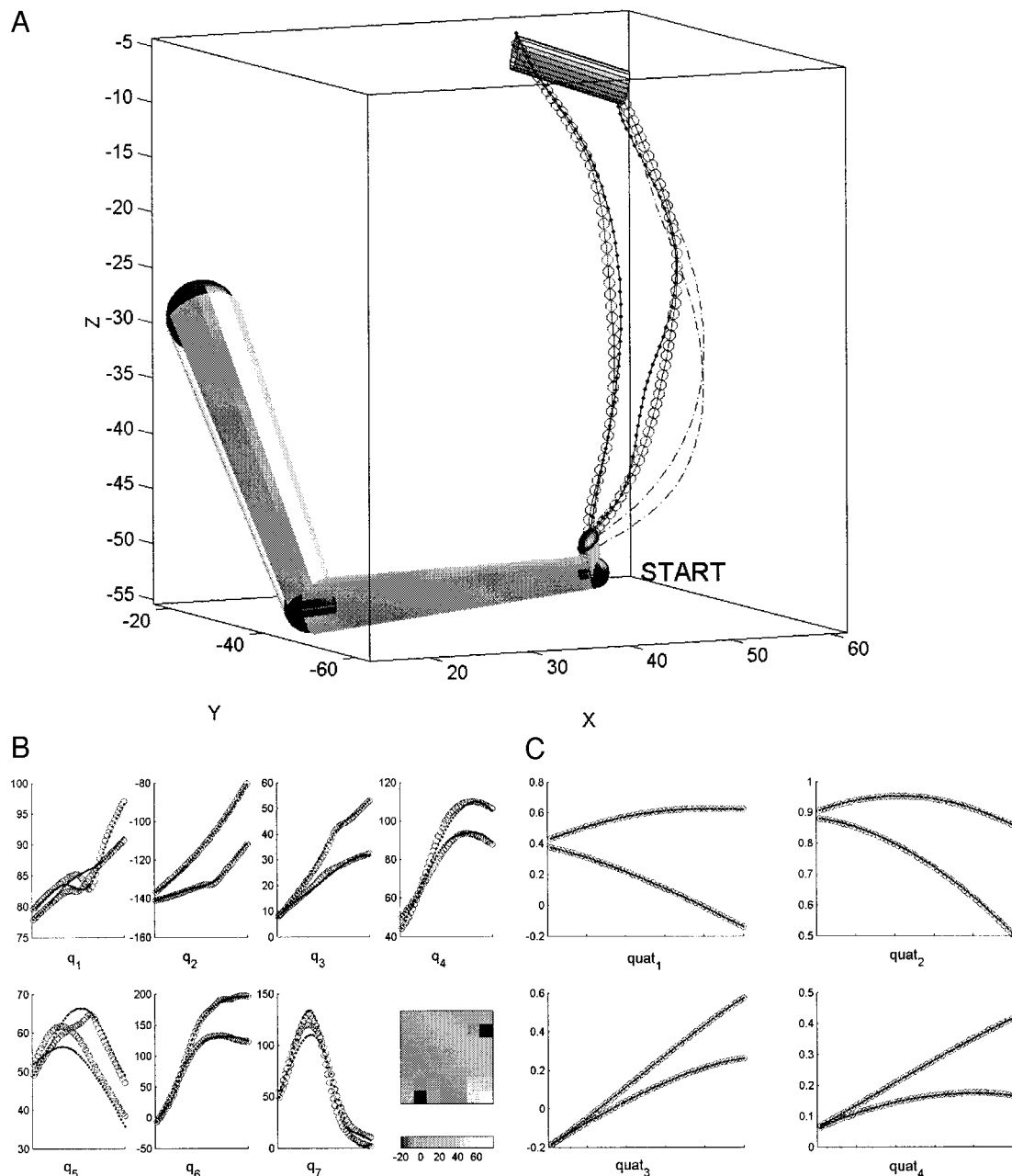


FIG. 7. Effect of target orientation on hand paths (A), posture paths (B), and orientation hand paths (C). The model predicts that movements starting from the same initial posture toward a target located at the same position but oriented differently cause differences in the arm postural paths, and in the corresponding hand position and orientation paths. Position hand paths produced by the gradient method in a Euclidean space are shown as broken lines. We also show experimental data corresponding to one subject's sensor hand path (averaged across 6 repetitions) (circled lines) superimposed on the paths generated by the model after the metric correction. The task was to match 2 different orientations, as if grasping the cylinder from above (palm faces down) vs. doing so from below (palm faces up). Subjects were holding a cylinder to match the target orientation with it and avoid the obstacle avoidance problem that arises from the interactions between the fingers and the object to be grasped at the end of the movement. *B*: the joint angle paths recovered from the position and orientation sensor data superimposed on the model-generated paths. The symmetric-definite-positive matrix used to correct the gradient is also shown here. *C*: the sensor orientation hand path (circled lines) compared with those obtained from the model (dotted lines).

back delays, when there is tight temporal coupling between stages, requires a feed forward dynamical model. Such a model undoubtedly exists, perhaps in the cerebellum.

It is not necessary for the geometric stage to operate in a tightly coupled mode. It can also generate postural paths without movement taking place. This is done by continuously adding the changes in posture given by the gradient to the

current posture, which is what is done in our simulations and is just numerical integration of Eq. 1. In this way, partial or complete paths can be computed before any movement starts. This would allow the motor implementation stage to carry out the predictive preparations known to be made before movements start (Gribble 1999). The geometric stage can also be invoked at any point during a preplanned movement to adjust

TABLE 3. Average mean absolute deviations from the difference between model and data

Fitting Matrix	Shoulder			Elbow		Wrist	
	Abduction	Flexion	Pronation	Flexion	Pronation	Abduction	Flexion
Least square	0.0024	0.0016	0.002	0.0054	0.0035	0.0115	0.0206
Symmetric positive definite	0.0029	0.0048	0.0033	0.0110	0.0053	0.0126	0.0267

One subject in the two different target-orientation experiment.

for unexpected perturbations. If the geometric stage is located in a posterior parietal area, this could account for the observation that this part of the brain is involved in correcting movement when target positions change (Desmurget et al. 1999).

Considerations about coupling between stages suggest that there is a need for some kind of buffer memory for path information. Using such a buffer the geometric stage could first provide a path and then monitor ongoing movement to correct for deviations from the path or for changes in target position. Such a buffer would be needed by most other motor control

mechanisms and probably should be looked for by physiologists.

The gradient implementation makes the geometric stage a type of error correction mechanism, so it can handle errors on-line. This occurs as long as the geometric stage receives feedback on the current status of the parameters of interest. There is experimental evidence suggesting the sensory-motor pathway is capable of on-line error correction (Prablanc and Martin 1992).

Our model does not disambiguate redundancies in the way

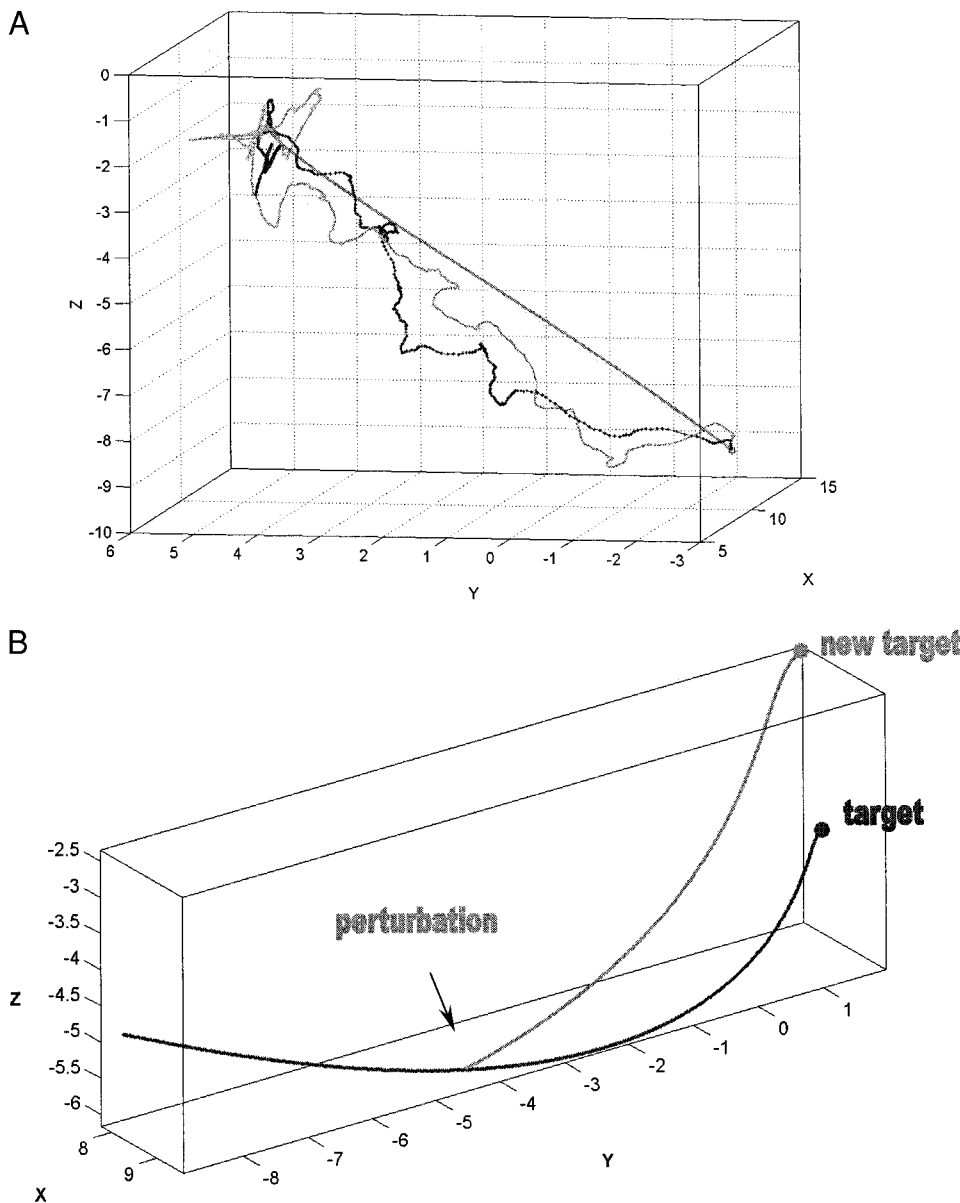


FIG. 8. The gradient descent approach permits on-line error correction because it receives continuous updated feedback about the actual configuration of the arm. A: 2 different levels of random noise have been applied to the gradient vector components at each time step (light and dark dotted paths). These paths are perturbed, but the hand gets to the target. The smooth path is generated by the gradient without noise. B: the gradient descent approach permits to correct on-line perturbations of target position because it receives continuous feedback from the current visual goal in the form of target position and orientation. In the next time step, it will compensate for the change in target position and output a signal that correctly re-directs the hand to the new target.

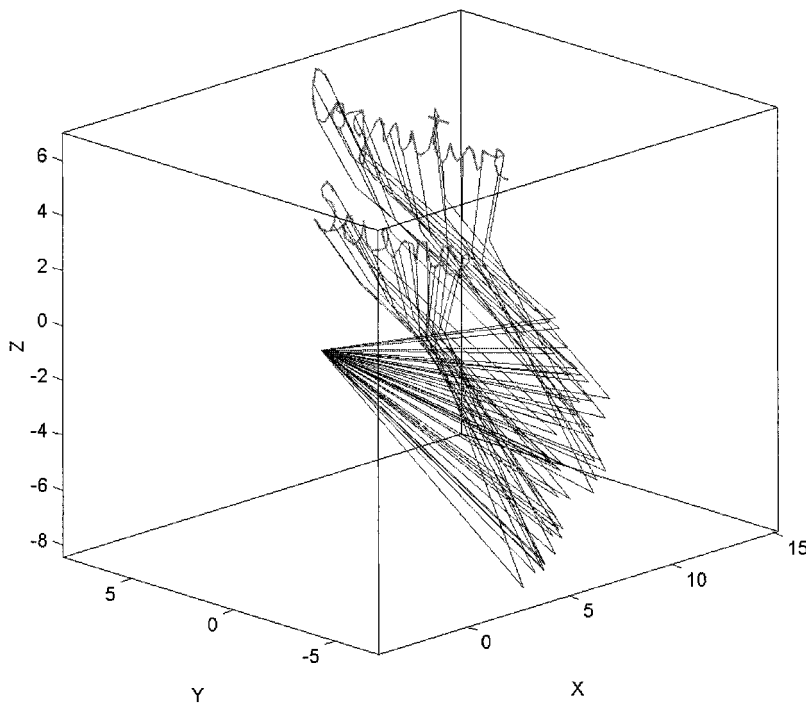


FIG. 9. The gradient method can generate arbitrary trajectories. This figure shows a hand-written pattern, represented by a stream of input targets, traced out by the hand. Several frames are selected to show some of the postures in the sequence. Notice the smooth transition from posture to posture. This is achieved by using the posture for each point in the pattern as the starting posture for moving to the next given target position.

targets can be grasped. Instead a particular grasp orientation must be given as the orientation input. However, the model is able to deal with all grasp orientations. This was demonstrated by simulation in the paper where we show how postural paths are affected by target orientation. How and where this disambiguation is carried out in the brain is unknown.

The model presented here is oversimplified to provide clarity. However, the general principles are not limited to the simplified form of the model but can be generalized to more realistic situations. The gradient method, in particular, is very flexible. It is not limited to joint angle space but will work with any posture configuration space. So long as the postures expressed in one space can be related to the parameters in another space, it is possible to compute the corresponding transformation rule in the gradient computation. This has biological significance because we do not yet know what space(s) the brain uses to encode postures but can be reasonably confident that the model can be adapted to the representations used in the brain.

The control of speed independently of trajectory is advantageous because it allows either to be changed rapidly during the course of movement. The conjecture that trajectories and speed are specified independently has some experimental support (Atkeson and Hollerbach 1985; Nishikawa et al. 1999). It also raises some interesting questions such as *how* and *where* speed is specified. One possibility is that speed information is fed to the geometric stage where it is used to determine the step size, or length, of the gradient vector passed to the motor stage. Changing the step size would not significantly affect hand trajectories but could be used to modulate speed.

Reaching for moving and rotating targets fits nicely into this paradigm. Delays can be compensated for by providing target locations that lead actual target positions. There is evidence that the visual system does advance the locations of moving objects because we perceive them ahead of their actual locations (Kowler 1989).

APPENDIX A

Let $Y = M \cdot X$, where $Y = dq^{\text{Data}}$ and $X = dq^{\text{Model}}$.

We define the error $E = \sum_{i=1}^n (y_i - M \cdot x_i)^2$ and want to solve for M for all n , the number of points in the data set (all points in all paths). We need to minimize E over M by solving $(\partial E / \partial M) = 0$. Taking the derivative with respect to M and using the chain rule amounts to solving

$$\frac{\partial E}{\partial M} = 2 \sum_{i=1}^n (y_i - Mx_i)(-x_i^T) = 2 \sum_{i=1}^n (-y_i x_i^T + Mx_i x_i^T) = 0, \quad \text{where}$$

$$M = \left(\sum_i y_i x_i^T \right) \left(\sum_i x_i x_i^T \right)^{-1} \quad \text{and we set } G^{-1} = M$$

When we want the matrix M to be symmetric positive definite $M = B \cdot B^T$, we solve the same least-square problem subject to this constraint

$$\begin{aligned} E &= \text{Trace}\{(Y - BB^T X)(Y - BB^T X)^T\} \\ &= \text{Trace}\{(Y - BB^T X)(Y^T - X^T BB^T)\} \\ &= \text{Trace}\{-YX^T BB^T - BB^T XY^T + BB^T XX^T BB^T\} \\ &= \text{Trace}\{-2YX^T BB^T + XX^T BB^T BB^T\} \end{aligned}$$

set $(\partial E / \partial B) = 0$ and minimize E with respect to B to solve for B and obtain $G^{-1} = B \cdot B^T$.

We thank E. Todorov, J. Mitchell, and H. C. Gromoll for very useful suggestions and comments.

This work was supported by National Institute of Neurological Disorders and Stroke Grants 3 F31 NS-10109-05S1 (to E. Torres) and ICS9820 (systems developments to D. Zipser).

REFERENCES

- ALEXANDER RM. A minimum energy cost hypothesis for human arm trajectories. *Biol Cybern* 76: 97–105, 1997.
 ALTMAN SL. *Rotations. Quaternions and Double Groups*. Oxford: University Press, 1986, p. 65–79.

- ATKESON CG AND HOLLERBACH JM. Kinematics features of unrestrained vertical arm movements. *J Neurosci* 5: 2318–2330, 1985.
- BENATI M, GAGLIO S, MORASSO P, TAGLIASCO V, AND ZACCARIA R. Anthropomorphic robotics. I. Representing mechanical complexity. *Biol Cybern* 38: 125–140, 1980.
- BERNSTEIN NA. *The Coordination and Regulation of Movements*. London, UK: Pergamon, 1967.
- BULLOCK D AND GROSSBERG S. Neural dynamics of planned arm movements: emergent invariants and speed-accuracy properties during trajectory formation. *Psychol Rev* 95: 49–90, 1988.
- CASTIELLO U, BENNETT K, AND CHAMBERS H. Reach-to-grasp: the response to a simultaneous perturbation of object position and size. *Exp Brain Res* 120: 31–40, 1998.
- DESMURGET M, EPSTEIN CM, TURNER RS, PRABLANC GE, ALEXANDER GE, AND GRAFTON ST. Role of the posterior parietal cortex in updating reaching movements to a visual target. *Nat Neurosci* 2: 563–567, 1999.
- DESMURGET M, GREY H, AND PRABLANC C. Final posture of the upper limb depends on the initial position of the hand during prehension movements. *Exp Brain Res* 119: 511–516, 1998.
- DESMURGET M AND PRABLANC C. Postural control of three-dimensional prehension movements. *J Neurophysiol* 77: 452–464, 1997.
- FLASH T AND HOGAN N. The coordination of arm movements: an experimentally confirmed mathematical model. *J Neurosci* 5: 1688–1703, 1985.
- GRAY A. *Modern Differential Geometry of Curves and Surfaces with Mathematica* (2nd ed.). Boca Raton, FL: CRC, 1998.
- GRIBBLE P. Compensation for interaction torques during single- and multijoint limb movement. *J Neurophysiol* 82: 2310–2326, 1999.
- HARRIS CM AND WOLPERT DM. Signal-dependent noise determines motor planning. *Nature* 394: 780–784, 1998.
- HINTON G. Parallel computations for controlling the arm. *J Mot Behav* 16: 171–194, 1984.
- HOFF B AND ARBIB MA. Models of trajectory formation and temporal interaction of reach and grasp. *J Mot Behav* 25: 175–192, 1993.
- JEANNEROD M. *The Neural and Behavioural Organization of Goal-Directed Movements*. Oxford, UK: Clarendon, 1988, p. 55.
- KALASKA JF, COHEN DAD, HYDE ML, AND PRUD'HOMME M. Parietal area 5 neuronal activity encodes movement kinematics not movement dynamics. *Exp Brain Res* 80: 351–364, 1990.
- KOWLER E. Cognitive Expectations, not habits, control anticipatory smooth oculomotor pursuit. *Vision Res* 29: 1049–1057, 1989.
- NISHIKAWA KC, MURRAY ST, AND FLANDERS M. Do arm postures vary with the speed of reaching? *J Neurophysiol* 81: 2582–2586, 1999.
- PRABLANC C AND MARTIN O. Automatic control during hand reaching at undetected two-dimensional target displacements. *J Neurophysiol* 67: 455–469, 1992.
- SOECHTING JF, BUNEO CA, HERRMANN U, AND FLANDERS M. Moving effortlessly in three dimensions: does Donders' Law apply to arm movement? *J Neurosci* 15: 6271–6280, 1995.
- TORRES EB. *Theoretical Framework for the Study of Sensori-motor Integration* (PhD thesis). La Jolla, CA: University of California, 2001.
- UNO Y, KAWATO M, AND SUZUKI R. Formation and control of optimal trajectory in human multijoint arm movement. Minimum torque-change model. *Biol Cybern* 61: 89–101, 1989.
- WANG X AND VERRIEST JP. A geometric algorithm to predict the arm reach posture for computer-aided ergonomic evaluation. *J Visual Comput Anim* 9: 33–47, 1998.
- WHITNEY DE. Resolved motion rate control of manipulators and human prostheses. *IEEE Trans Man-Machine Syst* 10: 47–53, 1969.
- WITKIN A, FLEISCHER K, AND BARR A. Energy constraints on parameterized models. *Comput Graph* 21: 225–232, 1987.