

## ORIGINAL RESEARCH

## A physiologically based hypothesis for learning proprioception and in approximating inverse kinematics

Matt Simkins

MEMM Department, California State University, Chico, California

### Keywords

Inverse kinematics, position gradients, synergy, virtual points.

### Correspondence

Matt Simkins, MEMM Department, California State University, 1205 West 7th Street, Chico, CA 95929-0789.  
Tel: 530-898-6488  
Fax: 530-898-4070  
E-mails: msimkins@csuchico.edu; simkinsmatt@hotmail.com

### Funding Information

No funding information provided.

Received: 7 March 2016; Accepted: 15 March 2016

doi: 10.14814/phy2.12774

*Physiol Rep*, 4 (10), 2016, e12774,  
doi: 10.14814/phy2.12774

### Abstract

A long-standing problem in muscle control is the “curse of dimensionality”. In part, this problem relates to the fact that coordinated movement is only achieved through the simultaneous contraction and extension of multitude muscles to specific lengths. Couched in robotics terms, the problem includes the determination of forward and inverse kinematics. Of the many neurophysiological discoveries in cortex is the existence of position gradients. Geometrically, position gradients are described by planes in Euclidean space whereby neuronal activity increases as the hand approaches locations that lie in a plane. This work demonstrates that position gradients, when coupled with known physiology in the spinal cord, allows for a way to approximate proprioception (forward kinematics) and to specify muscle lengths for goal-directed postures (inverse kinematics). Moreover, position gradients provide a means to learn and adjust kinematics as animals learn to move and grow. This hypothesis is demonstrated using computer simulation of a human arm. Finally, experimental predictions are described that might confirm or falsify the hypothesis.

## Introduction

Proprioception is the sense of one’s kinematics. This includes a high-level, cortical understanding of posture, as well as a low-level representation of muscle lengths, joint angles, and their relation to Euclidean space. This low-level aspect of proprioception suggests the existence of a forward kinematic model. Forward kinematics determines position and orientation of the hand given some combination of joint angles, or equivalently, muscle lengths. Put another way, forward kinematics maps muscle lengths in “muscle space” (Kakei et al. 1999) to Cartesian coordinates of the hand in Euclidean space. Notwithstanding, the human arm has proprioceptive receptors in the joints and in the muscles (muscle spindles). These receptors are capable of sensing joint angles and muscle lengths, respectively. Muscle spindles play a predominant role in proprioception (Ferrell et al. 1987). Goal-directed postures or

inverse kinematics, considers kinematics in the opposite direction. Inverse kinematics relates hand locations in Euclidean space to muscle lengths in muscle space. Not only is the mapping from Euclidean coordinates to joint angles is nonlinear, the mapping from joint angles to muscle lengths is also nonlinear. In the case of manipulators that have kinematic redundancies, such as the human arm, hand locations maps to an infinite number of joint angles and muscle lengths.

Current motor theory posits that movements are coded dynamically. Given that activity in the motor cortex is known to encode for velocity (Ashe and Georgopoulos 1994; Moran and Schwartz 1999; Paninski et al. 2004), and force (Taira et al. 1996; Sergio and Kalaska 1998; Li et al. 2001), that assumption appears sound. Muscle synergies (Sherrington 1947; d’Avella and Bizzi 2005; Cheung et al. 2009; Tresch and Jarc 2009), or similarly motor primitives (Thoroughman and Shadmehr 2000; Flash and

Hochner 2005; Gizzi et al. 2010), provide an attractive solution for multijoint movements. According to this view, the central nervous system (CNS) sends a volley of time-varying activations to various muscle groups in order to achieve motion. Without doubt, such a scheme could generate trajectories, torques, and forces. However, from the perspective of accurate positioning of the hand or even shaping trajectories, muscle synergies, by themselves, are unworkable without a kinematic model of some kind. Lacking kinematics, a separate muscle synergy is needed to move the hand from every conceivable start location to every conceivable end location. Moreover, impedances or muscle fatigue would result in significant and cumulative position errors. One might argue that muscle synergies are workable provided that they incorporate proprioceptive afferents for start and end conditions (Bizzi and Abend 1985), but this is tantamount to saying that a kinematic model exists.

It is known that some cells in the motor cortex encode for a particular direction of motion. Activities of populations of motor cortical neurons encode a population vector that points in the direction of intended movement (Kettner et al. 1988). Population vectors were used to spectacular effect in controlling robotic arms in real time using neuronal activity in the motor cortex (Wessberg et al. 2000; Velliste et al. 2008; Vato et al. 2012). No doubt, such work has huge potential in terms of prosthetics and brain-controlled interfaces (Schwartz et al. 2006). However, even though population vectors allow a robot to accurately predict an animal's intended movement, this is only accomplished using an implicit inverse kinematic model for the robot's kinematics. The animal's ability to control its own skeletal muscles remains unresolved. It might be argued that the animal's CNS simply calculates kinematics in an analogous way to the robot. However, there exist no known neurological mechanisms that approximate the coordinate frame transformations used in robotics. Moreover, calculating inverse kinematics using conventional methods in robotics proves problematic when applied to complicated systems such as the musculature of the human arm (Full and Koditschek 1999). In this sense, the problem of calculating kinematics touches on aspects of the degrees of freedom problem (Bernstein 1967; Arimoto et al. 2005), or on the "curse of dimensionality" (Full and Koditschek 1999).

A complete theory of motor control must include kinematics. This statement derives from the inescapable truth that hand positions and arm postures are inextricably determined by muscle lengths, muscle attachment points, and skeletal dimensions. Kinematics captures the geometry of problems.

A physiologically based model for calculating kinematics is proposed. This hypothesis is not presented as an

overarching model that describes all aspects of muscle control, or even all aspects of kinematics. Problems such as kinematic redundancy resolution, orientation, trajectory shaping, force compensation, muscle contraction dynamics, and so on, are out of scope. Rather, the "virtual points" hypothesis is presented as a kinematic bridge between high-level cortical processing and low-level control of muscles. In order to validate the hypothesis, at least in principle, a model that describes virtual points was applied to the human arm and simulated on a computer.

## Methods

### Virtual points hypothesis – spatial representations

The concept of muscle synergy has different meanings in the literature, but it is often described in terms of muscle innervations or forces (Thoroughman and Shadmehr 2000; Cheung et al. 2005; Ting and Macpherson 2005; Danna-Dos-Santos et al. 2009). For these purposes, muscle synergies are described in the context of kinematics, which by definition, does not involve forces. Therefore, let us consider a different type of synergy called a muscle length synergy (MLS). An MLS is defined as a combination of muscle lengths for a given posture of the arm. The human arm has kinematic redundancies (not to be confused with redundant muscles). That is, for a given hand location, the arm might assume a range of different postures (Scholz and Schöner 1999). The movements being simulated will restrict the elbow and hand to move in a plane so that kinematic redundancies are not permitted.

In a sense, the musculoskeletal system constitutes a physical embodiment of a kinematic calculator. For example, one might imagine adjusting arm muscle lengths. Without having to calculate the resulting posture, the forward kinematic solution is essentially available by simply observing the physical hand location. Conversely, one might imagine moving the hand to some location and measuring the associated muscle lengths. Thus, the muscle lengths are also available without having to calculate inverse kinematics. The virtual points hypothesis relies on the fact that the muscle lengths associated with a given hand location, whatever those lengths happens to be, is a particular inverse kinematic solution for that location. Thus, if the hand were to move through space while being observed, the observer (or CNS) might record the coordinates in Euclidian space along with the corresponding set of muscle lengths. Indeed, learning kinematics does seem to depend heavily on visual feedback (Wolpert et al. 1995). In an overly simplistic model, a hand

location is accomplished by simply fetching the needed set of muscle lengths from a lookup table. The muscles would then change in length until the specified muscle lengths are achieved and a stable posture is obtained (Feldman 1966; McIntyre and Bizzi 1993). However, there is a problem. For the hand to span continuous Euclidian space, an infinite number of recorded coordinate points are required. Thus, for the CNS to employ such a strategy, an infinite amount of memory is required. Therefore, it is assumed that only a finite set of MLS's are retained. This set is denoted as a muscle length synergy matrix,  $\mathbf{S}$ .

Due to the finite information storage capacity in the CNS, it might seem that the hand would be restricted to move to specific locations in task space. However, this limitation is surmountable using interpolation. In terms of spaces, interpolation along a line requires two independent pieces of information. Interpolation along a surface requires three independent pieces of information. Interpolation along a volume requires four independent pieces of information. Given that the hand moves in 3D Euclidean space, it is assumed that interpolation will require four independent pieces of information. Another generalization relates to the dimensionality of coordinates. For a line, points describe coordinates. For a surface, coordinates relate to independent or nonparallel lines along the surface. The intersection of any two lines defines a coordinate point. For a volume, coordinates relate to independent planes. The intersection of any three nonparallel planes describes a coordinate point. Again, because the hand moves in 3D Euclidean space, it is assumed that position is described by planes. For these reasons, it is assumed that interpolation for each MLS involves the recruitment of four independent or nonparallel planes.

The preceding description of planes has a physiological basis in the arm area of the motor cortex and somatosensory cortex. It was found that the discharge rate in primates increased as the hand moves toward planar surfaces in the Euclidean space. These planes were termed "position gradients" (Georgopoulos et al. 1984; Kettner et al. 1988). The discharge rate or tuning curve, of position gradients are modeled in this work as a normal distribution that is centered on a plane. In other words, the normal distance to a plane is modeled as the random variable. Note, a normal distribution was selected with the caveat that another bell-shaped function might model neurophysiology of position gradients with greater accuracy.

The planes that describe position gradients are used to define a coordinate system. This coordinate system is partitioned using the mutual intersections of four sets of planes. The planes within each set of planes are all parallel and equally spaced from neighboring planes within that set. Four unit vectors are used to describe plane orientation and are given as follows,

$$\hat{\mathbf{u}}_1 = \frac{1}{2}\hat{\mathbf{i}} + \frac{1}{2}\hat{\mathbf{j}} + \frac{1}{\sqrt{2}}\hat{\mathbf{k}} \quad (1)$$

$$\hat{\mathbf{u}}_2 = -\frac{1}{2}\hat{\mathbf{i}} + \frac{1}{2}\hat{\mathbf{j}} + \frac{1}{\sqrt{2}}\hat{\mathbf{k}} \quad (2)$$

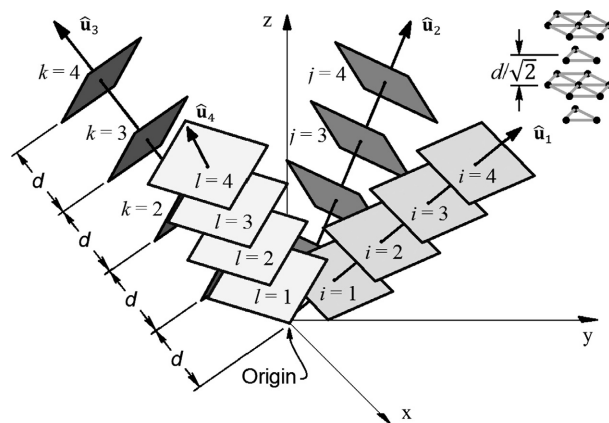
$$\hat{\mathbf{u}}_3 = \frac{1}{2}\hat{\mathbf{i}} - \frac{1}{2}\hat{\mathbf{j}} + \frac{1}{\sqrt{2}}\hat{\mathbf{k}} \quad (3)$$

$$\hat{\mathbf{u}}_4 = -\frac{1}{2}\hat{\mathbf{i}} - \frac{1}{2}\hat{\mathbf{j}} + \frac{1}{\sqrt{2}}\hat{\mathbf{k}} \quad (4)$$

where  $\hat{\mathbf{i}}$ ,  $\hat{\mathbf{j}}$ , and  $\hat{\mathbf{k}}$  are unit vectors that point in the  $x$ ,  $y$ , and  $z$  directions, respectively. The location of a given plane within each set of planes is given by an index. The distance between planes of consecutive indices is  $d$ , and the indices are given by  $i$ ,  $j$ ,  $k$ , and  $l$  for planes parallel to  $\hat{\mathbf{u}}_1$ ,  $\hat{\mathbf{u}}_2$ ,  $\hat{\mathbf{u}}_3$ , and  $\hat{\mathbf{u}}_4$  respectively. A graphical depiction is provided in Figure 1.

The entire ensemble of plane intersections forms a grid of regularly spaced points in the Euclidean space. Note, the unit vectors given in equations (1–4) form a  $45^\circ$  angle with the  $x$ - $y$  plane. However, the position gradient planes will continue to have mutual points of intersection for angles other than  $45^\circ$  and  $45^\circ$  was assumed ad hoc. The geometry of this grid of intersecting planes is revisited further on. Before continuing, a cautionary warning is in order. Specifying coordinates and indices using the planes described by equations 1–4 is cumbersome. This makes the mathematical description of position gradients appear complicated and unintuitive. However, outside of indexing, the underlying calculations are actually quite simple.

Ultimately, the position gradient arrangement in Figure 1 is used as a means to learn proprioception and to estimate inverse kinematics. Before embarking on those



**Figure 1.** Plane geometry. All 4 sets of planes have the same plane spacing. The geometry of plane intersection points is depicted in the upper right.

topics, the underlying model is described in equations 5–11. The discharge rate or tuning curve, at distance  $\chi$  from each plane is modeled by the normal distribution function given as follows,

$$N(\text{mean} = 0, \text{var} = c^2 d^2) = \frac{1}{cd\sqrt{2\pi}} \exp\left(-\frac{\chi^2}{2(cd)^2}\right) \quad (5)$$

where the “mean” value is assumed to equal zero and the “standard deviation” is assumed to equal  $cd$ . A plane spacing of  $d = 40$  mm and  $c = 0.389$ , was used for all simulations. As is described latter on, the selection of  $c$  and  $d$  are related. A justification for these selections is provided in the Results section.

Let  $\mathbf{p}$  denote the *observed* location of the hand. In terms of cortical processing, it is uncertain which specific anatomical feature of the arm or hand, plays a predominant role for positioning. For the purposes of simulation, hand positions are assumed to relate to the center of the wrist joint. Being able to identify which planes have the smallest normal distance from  $\mathbf{p}$  is needed for future calculations. Along those lines, assume a  $1 \times 3$  vector of Cartesian coordinates, given by  $\mathbf{p}$ , denotes hand position. The plane indices that most closely neighbor  $\mathbf{p}$  are given by the integers along the diagonal of the following,

$$\mathbf{I}(\mathbf{p}) = \begin{bmatrix} i = \|\mathbf{p} \cdot \hat{\mathbf{u}}_1/d\| & 0 & 0 & 0 \\ 0 & j = \|\mathbf{p} \cdot \hat{\mathbf{u}}_2/d\| & 0 & 0 \\ 0 & 0 & k = \|\mathbf{p} \cdot \hat{\mathbf{u}}_3/d\| & 0 \\ 0 & 0 & 0 & j = \|\mathbf{p} \cdot \hat{\mathbf{u}}_4/d\| \end{bmatrix} \quad (6)$$

where the ‘ $\|\cdot\|$ ’ operator rounds the dot products to the nearest integer. The projections of  $\mathbf{p}$  onto the planes are given by the following equation.

$$\mathbf{D}(\mathbf{p}) = \begin{bmatrix} \mathbf{p} \cdot \hat{\mathbf{u}}_1 & 0 & 0 & 0 \\ 0 & \mathbf{p} \cdot \hat{\mathbf{u}}_2 & 0 & 0 \\ 0 & 0 & \mathbf{p} \cdot \hat{\mathbf{u}}_3 & 0 \\ 0 & 0 & 0 & \mathbf{p} \cdot \hat{\mathbf{u}}_4 \end{bmatrix} \quad (7)$$

The distance from  $\mathbf{p}$  to the nearest plane is determined by the following.

$$\Delta(\mathbf{p}) = \mathbf{D}(\mathbf{p}) - d\mathbf{I}(\mathbf{p}) \quad (8)$$

Substituting  $\Delta(\mathbf{p})$  for  $\chi$  in Eq. 5 calculates the discharge rate at location  $\mathbf{p}$  from each of the planes of the nearest virtual point. These discharge rates are given along the diagonal of the following matrix.

$$\Omega(\mathbf{p}) = \frac{1}{cd\sqrt{2\pi}} \exp\left(-\frac{\Delta(\mathbf{p})^2}{2(cd)^2}\right) \quad (9)$$

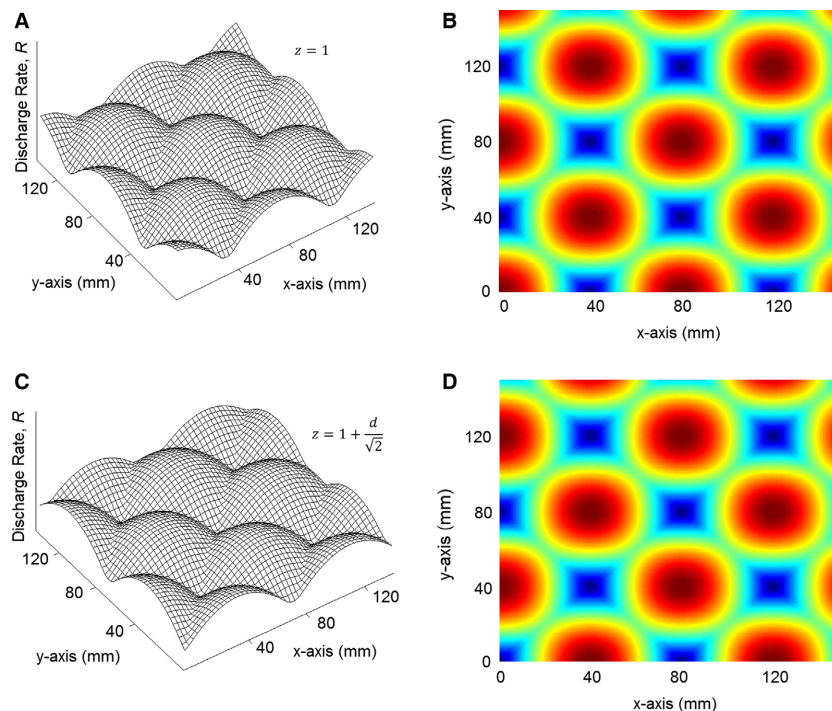
Note, matrix  $\Delta(\mathbf{p})$  is being squared, and raised to an exponential. These operations act individually on the elements for diagonal matrices such as this. Finally, the combined discharge rate, at location  $\mathbf{p}$ , from all four planes is given as follows.

$$R(\mathbf{p}) = \text{trace}(\Omega(\mathbf{p})) \quad (10)$$

Because the planes in equation 10 describe vector fields of increasing discharge rate as the hand approaches a given plane, the discharge rate,  $R$ , is especially high when the hand approaches locations where all four planes intersect. These points are referred to as “virtual points”. Though difficult to graph, the discharge rate around these points is analogous to the three-dimensional probability density function or cloud that is associated with an electron orbiting a hydrogen atom. A grid of virtual points is depicted graphically in Figure 2 for two heights along the  $z$  axis. As the hand sweeps across the  $x$  and  $y$  axes, the discharge rate increases as the hand approaches the various virtual points. The discharge rate is depicted along the vertical axes in Figure 2A and C. It is depicted as a color map in (B) and (D).

Notice in Fig. 2 that levels of peak activity alternate for different distances along  $z$ . In particular, for the plane orientations given by equations 1–4, the virtual point locations toggle as  $z$  increment by  $d/\sqrt{2}$ . An executable script in MATLAB is provided in Appendix I. That script generates, and graphs this grid. For added perspective on the three-dimensional structure, this virtual point grid geometry is very similar to the Bravais lattices found in hexagonal close packed crystals.

The contention that the motor cortex partitions space into families of equally spaced position gradients is a theoretical prediction and has not been reported experimentally. Notwithstanding, there is precedent for such an organization in the entorhinal cortex (Moser et al. 2008). It was found that “grid cells” exhibit patterns of activity that resemble Figure 2 for navigation tasks. Admittedly, grid cells were only reported for two-dimensional navigation. However, it stands to reason that grid cells might



**Figure 2.** Discharge rate activity for movement along the  $x$ - $y$  plane.

also be at work for three-dimensional navigation tasks such as tree climbing or flight, and that grid cell firing for planar movement might extend to three dimensions, as is the case here. A reasonable objection to the activity depicted in Figure 2 is that grid cell-like activity has not been observed in the arm area of the cortex. Indeed, Figure 2 is not intended to depict position gradients, but rather, subcortical activity which is generated by a summation of position gradients. Equation 10 is revisited when proprioceptive learning and adaption are considered. The predicted grid cell-like activity in subcortical regions is considered further in the Discussion section.

The relationship between MLS's and position gradients now becomes easier to describe. Virtual points are cortical representations of points in external Euclidean space. When the hand is located at a virtual point, the set of associated muscle lengths constitutes an MLS given by

$$\mathbf{s}_{ijk} = (L_1, L_2, \dots, L_9) \quad (11)$$

where  $\mathbf{s}_{ijk} \in \mathbf{S}$ . For this work, six muscles in the elbow and two muscles in the shoulder are modeled. An anatomically fictitious ninth muscle is also included for reasons that are discussed later on. In this way, MLSs map points in Euclidean external space to muscle lengths in "muscle space". Though, nine muscles are being simulated in this work, the hypothesis places no restrictions

on the number of muscles being included in each MLS. The relationship between external space and muscle space is a few-to-many mapping; in that, more than one muscle is required to move a joint. Subscripts  $i$ ,  $j$ , and  $k$  denote the index of the MLS within  $\mathbf{S}$ . Indices  $i$ ,  $j$ , and  $k$  have two interpretations. In equation 6, they relate locations in Euclidean space using plane indices. In equation 11, they index to sets of muscle lengths in  $\mathbf{S}$ .

Consider what would happen if an arm MLS were somehow activated. The muscles of the arm should change in length such that the arm assumes a posture that moves the hand to the location of a virtual point. Of course, the musculature of the arm is viscoelastic and the arm would be unable to maintain the posture with perfect rigidity (Feldman and Latash 2004). Thus, displacing the hand from the virtual point would result in a restoring force. This restoring force should increase with increasing displacement. Indeed, such a phenomenon was observed using microstimulation of spinal cord gray matter in frogs (Bizzi et al. 1991; Giszter et al. 1993), rats (Tresch and Bizzi 1999), and cats (Lemay and Grill 2004). When spinal cord tissue was stimulated in these animals, their extremity would move to a repeatable location in space. When the extremity was displaced from one of these locations, the extremity exhibited a restoring force toward its point of attraction. This phenomenon was termed "convergent force fields" (Bizzi et al. 1991). Therefore,

interneurons in the C3–C7 region of the spinal cord are possible candidates for storing arm MLSs. Locating MLSs closer to muscles (i.e., in the spinal cord as opposed to the cortex) also reduces latencies associated with muscle length error correction. This follows the same reasoning that explains why the stretch reflex or the nociceptive withdrawal reflex, are spinal phenomena (Andersen 2007).

### Virtual points hypothesis – inverse kinematic estimation

In principle, equation 10 could specify a posture by increasing the neuronal activity for a given set of four planes. Peak discharge rate for the corresponding virtual point would then project onto an MLS in the spinal cord. However, the motor cortex must also perform interpolation between virtual points. In the following section, the mathematics of interpolation is described. Before embarking on a formal mathematical description, interpolation is described presently in words. Any desired posture will position the hand at a location that is neighbored by a collection of virtual points. Let  $\mathbf{p}_D$  denote the *desired* location of the hand in Euclidean space. Note,  $\mathbf{p}_D$  differs from  $\mathbf{p}$ ; in that,  $\mathbf{p}$  relates the observed or actual hand location, while  $\mathbf{p}_D$  relates to a target location. The position gradients of closer virtual points will exhibit larger discharge rates than more distant virtual points. Those discharge rates are used to weight the MLSs that correspond to the given virtual points. This makes sense because virtual points near  $\mathbf{p}_D$  will map to an MLS that is more similar to the muscle lengths needed to achieve an arm posture that places the hand at  $\mathbf{p}_D$ . As such, they should receive a higher weight. Distant virtual points will map the MLSs that are less similar to the muscle lengths that achieve  $\mathbf{p}_D$ . As such, they should receive a lower weight.

The volley of descending weights or efferent activity, could be construed as the transmission of a “forward model” (Miall and Wolpert 1996). The scaled MLSs are then neurologically summed (Carandini and Heeger 1994) at lower levels to find a set of muscle lengths that achieve the interpolated posture. In other words, a collection of MLSs are activated simultaneously using various weights. Consistent with this theory, experimental evidence shows that simultaneous stimulation of several spinal cord sites results in a summative effect of convergent force fields (Tresch and Bizzi 1999).

The preceding remarks assume that efferent activity projects onto interneurons within the spinal cord. Descending weights are then used to excite neurons that encode for muscle lengths. Thereafter, motor neurons projecting from the ventral root are used to

modulate skeletal muscle length. Presumably, spinal cord processes, such as the stretch reflex, in combination with muscle spindle afferent feedback, would act to eliminate the error between the specified muscle lengths and the actual muscle lengths. This error correcting process would persist until the muscles achieve the desired lengths in accordance with the equilibrium point hypothesis (Feldman 1986). Modeling of neuronal activity in spinal cord grey matter is outside the scope of this work.

Returning to the model, assume proprioception is learned. In other words, assume that the elements of  $\mathbf{S}$  are populated with MLSs that accurately correspond to the muscle lengths of the arm when the hand is located at each virtual point. The vector between two virtual points is given by

$$\mathbf{p}_R(r, p, q) = [rd \quad pd \quad qd/\sqrt{2}] \quad (12)$$

where  $p$ ,  $q$ , and  $r$ , are integers that describe relative indexing between virtual points. A derivation that describes the spacing between virtual points is given in Appendix II. The location of a virtual point relative to  $\mathbf{p}_D$  is given by,

$$\mathbf{p}_{VP}(\mathbf{p}_D, r, p, q) = \mathbf{p}_D + \mathbf{p}_R(r, p, q) \quad (13)$$

where  $\mathbf{p}_{VP}$  is a function of desired hand position and the relative indexing. Substituting equation 13 into equation 8 provides the distance from  $\mathbf{p}_D$  to the planes of  $\mathbf{p}_{VP}$  and is given by,

$$\Delta(\mathbf{p}_D, r, p, q) = \mathbf{D}(\mathbf{p}_D) - d\mathbf{I}(\mathbf{p}_{VP}(\mathbf{p}_D, r, p, q)) \quad (14)$$

where  $\Delta$  is a  $4 \times 4$  diagonal matrix. Equation 10 summed the discharge rates from the four closest position gradients or equivalently the closest virtual point. However, equation 14 is being used for interpolation and more distant virtual points should contribute lower levels of activity than closer points. Recall that the position gradient discharge rate for each of the four planes of a given virtual point is given by the diagonal elements in equation 9. As such, the minimum activity from  $\mathbf{p}_D$  to the virtual point is found by expressing equation 9 as a function of equation 14 and in finding the minimum value among the diagonal elements. This is accomplished as follows.

$$\omega(\mathbf{p}_D, r, p, q) = \min(\Omega(\mathbf{p}_D, r, p, q)_{11}, \dots, \Omega(\mathbf{p}_D, r, p, q)_{44}) \quad (15)$$

Note, equation 15 is a scalar, not a matrix as was the case for equation 9. Recall in an earlier discussion that

interpolation within a volume requires four independent pieces of information. For this reason, four planes with different orientations were included in this model. The need for four nonparallel planes is reflected in equation 15. For perspective, interpolation was first tried using only three planes. Interpolation must consider the nearest planes that surround  $\mathbf{p}_D$ . If this is done using only three plane orientations, then interpolation must include two planes with the same orientations and consecutive indices. In that case, the volume that surrounds  $\mathbf{p}_D$  is a sliver-shaped polyhedron with a distant virtual point as one of its vertices. Recall that virtual points are being used to perform interpolation from the Euclidian space to muscle space. For nonlinear interpolations such as this, the reference points or virtual points, should be as close  $\mathbf{p}_D$  as possible. For a sliver-shaped polyhedron, the distant virtual point is equally likely to be included in the interpolation as the closer points. This has the effect of increasing the error to such an extent that interpolation is no more accurate than simply selecting the nearest virtual point. It is for this reason that four different plane orientations were required.

The discharge rate given by equation 15 is used as part of a weighed arithmetic mean that includes a collection of virtual points that neighbor  $\mathbf{p}_D$ . More specifically, interpolation will make use of the  $3^3 = 27$  neighboring virtual points in the vicinity of  $\mathbf{p}_D$ . The scaling factor for each virtual point is calculated as follows.

$$\hat{\omega}(\mathbf{p}_D, r, p, q) = \frac{\omega(\mathbf{p}_D, r, p, q)}{\sum_{r=-1}^1 \sum_{p=-1}^1 \sum_{q=-1}^1 \omega(\mathbf{p}_D, r, p, q)} \quad (16)$$

The indices of the MLSs in  $\mathbf{S}$  that are scaled by equation 16 are calculated using diagonal elements within the following  $4 \times 4$  matrix.

$$\boldsymbol{\eta}(\mathbf{p}_D, r, p, q) = \mathbf{I}(\mathbf{p}_{VP}(\mathbf{p}_D, r, p, q)) \quad (17)$$

Finally, the muscle lengths in  $\mathbf{S}$  are interpolated as a function of  $\mathbf{p}_D$  using equation 17 for the indices of  $\mathbf{s}$  as follows,

$$\mathbf{L}(\mathbf{p}_D) = \sum_{r=-1}^1 \sum_{p=-1}^1 \sum_{q=-1}^1 \hat{\omega}(\mathbf{p}_D, r, p, q) \mathbf{s}_{\eta_{11}\eta_{22}\eta_{33}} \quad (18)$$

where  $\mathbf{L}$  is a  $1 \times 9$  vector of interpolated muscle lengths. Note, indexing to a coordinate only requires 3 planes. Therefore, the subscript  $\eta_{44}$  is omitted from equation 18. Beyond the complexities associated with indexing, equation 18 is essentially a simple equation that sums muscle lengths by a weighted arithmetic mean of nearby position gradients.

## Virtual points hypothesis – proprioceptive learning/adaption

Animals must continually adapt to changing kinematics as they mature and grow. Having described virtual points and their relationship with MLSs, a learning/adaption process is describable. As the hand moves through space, the hand will inevitably pass close to virtual points. Let  $\mathbf{s}(\mathbf{p})$  denote the muscle lengths for the hand's current position. The location of the hand is discernable from visual or tactile observation, and this will result in some level of discharge rate given by equation 10. If the hand moves near a virtual point, the associated muscle lengths constitute a fairly accurate MLS solution for that location. If  $\mathbf{s}(\mathbf{p})$  happens to map the hand closer to the virtual point than the formerly learned MLS, then the MLS should be updated with the better solution. In this way, the CNS is constantly checking and updating  $\mathbf{S}$ . This process is now stated mathematically. Recall that the first three diagonal elements in equation 6 provide the indices of the nearest virtual point to  $\mathbf{p}$ . Learning is thus accomplished using the following condition statement.

$$\mathbf{s}_{\mathbf{I}(\mathbf{p})_1, \mathbf{I}(\mathbf{p})_2, \mathbf{I}(\mathbf{p})_3} = \begin{cases} \mathbf{s}(\mathbf{p}), & \text{if } R(\mathbf{p}) > R(\mathbf{p}_{\mathbf{I}(\mathbf{p})_1}, \mathbf{I}(\mathbf{p})_2, \mathbf{I}(\mathbf{p})_3) \\ \mathbf{s}_{\mathbf{I}(\mathbf{p})_1, \mathbf{I}(\mathbf{p})_2, \mathbf{I}(\mathbf{p})_3}, & \text{otherwise.} \end{cases} \quad (19)$$

The update process in equation 19 suggests that MLS's are retained and updated. If MLS's were retained in the spinal cord, this in turn suggests a spinal learning mechanism that is mediated by higher cortical levels (Kargo and Giszter 2000; Cheung et al. 2009; Roh et al. 2011). A number of experiments have demonstrated spinal motor leaning mechanisms, that is, neuroplasticity, (Chen and Wolpaw 2002; Wolpaw and Chen 2006; Edgerton et al. 2005) and subcortical motor learning has been proposed as a mechanism in stroke recovery (Simkins et al. 2014).

The inequality in equation 19 evaluates if the discharge rate of the current hand position is greater than the discharge rate that would occur if the hand were positioned using the previously stored MLS. In doing this comparison, equation 19 suggests that the CNS calculates position gradients from MLSs, presumably in the reverse process of equation 18. While this assumption was used for the purposes of simulation, there are alternative, and more plausible means for updating  $\mathbf{S}$ . This update process essentially reduces to a problem of retaining information based on an elevated discharge rate. In this case, the retained information are muscle lengths. The elevated discharge rate result from cortical activity associated with equation 10. From an algorithmic perspective, learning processes such as this are used widely in artificial neural networks and there is a substantial body of relevant work

in reinforcement learning (Kaelbling et al. 1996). A physiologically relevant model would leverage work done using long-term potentiation. Indeed, long-term potentiation models have a long history and numerous models have been proposed since long-term potentiation was confirmed experimentally in the hippocampus (Bliss and Collingridge 1993). Thus, inclusion of a learning model in the spinal cord would largely duplicate past work. For this reason, equation 19 was deemed a sufficient simplification for these purposes.

This model includes two distinct mechanisms. Notice that equation 19 is a function of the actual hand position,  $\mathbf{p}$ , not the desired position,  $\mathbf{p}_D$ . This distinction relates to the fact that actual hand positions, or even trajectories, may differ from what is desired (Bizzi et al. 1983; Won and Hogan 1995). For example, if the desired position was behind a fixed obstacle, the hand might simply push against the obstacle (Bizzi et al. 1992). If a desired trajectory was interrupted by an unexpected force field, the field might push the hand along a different trajectory (Shadmehr and Mussa-Ivaldi 1994; Mussa-Ivaldi and Patton 2000). One might say that equation 18 relates to where the hand should go while equation 19 relates to where the hand is. Estimating where the hand should go relates to movement planning and efferent activity. Evaluating where the hand is relates to learning and reafferent activity. Thus, even though equations 18 and 19 utilize similar mechanisms, the two calculations are essentially different. This distinction is revisited in the Discussion section.

## Computer simulation

The simulation is divided into a “learning” phase and an “adaption” phase. During learning, the CNS learns to position an adult sized arm from an empty  $\mathbf{S}$  matrix, that is, from a “blank slate”. During the adaption phase, the CNS adjusts a previously populated  $\mathbf{S}$  matrix to adapt to changing kinematics that is associated with limb growth.

In the learning phase, the  $\mathbf{S}$  matrix is initially populated with zeros. Learning is accomplished by populating  $\mathbf{S}$  with MLSs according to equation 19 using randomly generated movements, similar to a Monte Carlo method. One billion randomly generated joint angles, or “iterations” were performed. An analogy of this learning phase is of an infant flailing its arms at random. Let  $\text{FK}(\mathbf{s}_{ijk})$  denote the forward kinematic solution, or position, that the hand achieves when the muscles are stretched to the lengths given by  $\mathbf{s}_{ijk}$ . Recall that equation 12 described the relative location of a virtual point,  $\mathbf{p}_R$ , using indices  $p$ ,  $q$ , and  $r$ . However, expressing equation 12 as a function of  $i$ ,

$j$ , and  $k$  describes the Cartesian coordinate of a virtual point relative to the origin, see Figure 1. The average distance between  $\text{FK}(\mathbf{s}_{ijk})$  and virtual point locations,  $\mathbf{p}_R(i, j, k)$ , are given by

$$\hat{d}_{\text{learn}} = 1/\mu \sum_{m=1}^{\mu} \|\text{FK}(\mathbf{s}_{ijk}) - \mathbf{p}_R(i, j, k)\| \quad (20)$$

where  $\mu$  is the number of elements in  $\mathbf{S}$  with nonzero MLSs. As the hand moves early in the learning process, it is unlikely that the hand will have traveled close to most virtual points and  $\hat{d}_{\text{learn}}$  will be large. As the hand continues to move randomly, it will have more opportunities to pass nearer to virtual points and  $\hat{d}_{\text{learn}}$  will diminish in size. Thus, equation 20 is a measure of the completeness of proprioceptive learning.

Demonstrating the adaption phase is accomplished by increasing the length of  $L_u$  and  $L_f$  by five discrete lengths. For each growth size, the arm is then moved at random as if the arm performed numerous, unforeseen movements associated with activities of daily living. In this way, equation 19 adjusts  $\mathbf{S}$  to new kinematics. Growth is simulated as if the upper arm and forearm were lengthened instantaneously by 1, 2, 3, 4, or 5 mm. Thus, the overall arm length (upper arm plus forearm) increases by 2, 4, 6, 8, and 10 mm, respectively. A test trajectory is then used to evaluate the position errors caused by growth. Of course, the notion that the overall arm length might grow instantaneously, or “overnight,” by 10 mm, or even 2 mm, is implausible. The purpose of modeling proprioceptive adaption is to demonstrate the ability of equation 19 to adapt to changing kinematics. In reality, this process would occur imperceptively during maturation as the limbs grow slowly over time.

The test trajectory consists of 140 desired points, ( $\mathbf{p}_D$ 's), or “via points”. Thus, a series of hand positions was used to simulate hand trajectories, or “movement from posture” (Merton 1953). The via points forms a 200 mm by 80 mm rectangle. The average error for all 140 points along this test trajectory was calculated by following equation.

$$\overline{\text{error}} = \frac{1}{140} \sum_{m=1}^{140} \|\mathbf{p}_m - \mathbf{p}_{Dm}\| \quad (21)$$

where  $m$  is the  $m$ th via point along the test trajectory.

A two-joint simulation of the arm was implemented in MATLAB (Mathworks Inc., Natick, MA, USA). Hand positioning was simulated using the following forward kinematic model,

$$x(\theta_1, \theta_2) = L_u \cos(\theta_1) + L_f \cos(\theta_1 + \theta_2) \quad (22)$$



$$y(\theta_1, \theta_2) = L_u \sin(\theta_1) + L_f \sin(\theta_1 + \theta_2) - 600 \quad (23)$$

$$z(\theta_3) = \theta_3 \quad (24)$$

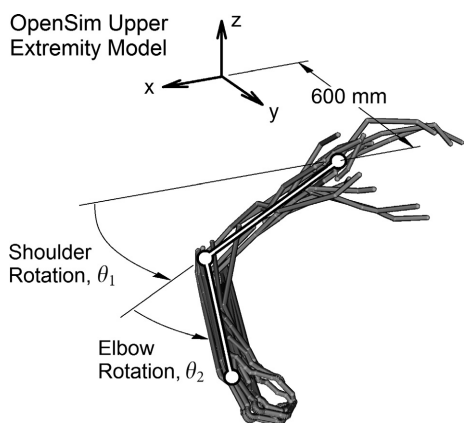
where  $\theta_1$  is the shoulder angle,  $\theta_2$  is the elbow angle,  $L_u$  is the upper arm length of 361 mm, and  $L_f$  is the forearm length of 278 mm. These lengths match the average length for an adult man (Winter 2005). The purpose of  $\theta_3$  is discussed later on.

There is an important point worth mentioning for equations 22 and 23. These are forward kinematic solutions that map joint angles to Cartesian coordinates. They are used for simulation purposes. Inverse kinematics calculates joint angles, or by extension, muscle lengths, as a function of Euclidian coordinates. The inverse problem is much more difficult to calculate than the forward problem. For a first hand perspective, consider solving for

$\theta_1(x, y)$  and  $\theta_2(x, y)$  using equations 22 and 23. Despite their simple appearance, the inverse solution is not easily obtainable. Solving for inverse kinematics often requires geometric reasoning, algebraic substitutions, and trigonometric identities. In many cases, there is no inverse solution and iterative techniques, such as Newton's method, are required. An inverse kinematic solution for equations 22 and 23 is never used in this work. Instead, inverse kinematics is estimated using position gradients and MLSs exclusively. Demonstrating a way for the CNS to perform such complicated calculations by leveraging known physiology is core to this work.

Muscles are being simulated, not joint angles. Therefore, joint angles in equations 22 and 23 need to be converted into muscle lengths. Muscle lengths were estimated as a function of joint angles using OpenSim 3.0 (Stanford University, Palo Alto, CA; Delp et al. 2007), in conjunction with an upper limb muscle model (Holzbaur et al. 2005). Joint angles ranged from  $23^\circ$  to  $-90^\circ$  for  $\theta_1$ , and  $3^\circ$  to  $130^\circ$  for  $\theta_2$ . These angles were selected such that the elbow and wrist were constrained to move in a sectional plane that intersects the glenohumeral joint, see Figure 3 (Mussa-Ivaldi 1997). Curve fits were then applied to the simulated muscle lengths using a second-order polynomial. The muscle names, abbreviations, curve fits, and goodness-of-fit ( $R^2$ ) are given in Table 1. This nine muscle model includes three agonist and three antagonist muscles pairs for the elbow, as well as an agonist and antagonist muscle for the shoulder.

The actual functional relationship between muscle length and joint angles are certainly more complicated than is suggested by second-order polynomials. However, the purpose of simulating the nonlinear relationships in Table 1 is to show that the model can learn and adapt to nonlinear muscle kinematics. As is evident from the  $R^2$  values in Table 1, the lowest  $R^2$  is 98%. Therefore, the



**Figure 3.** The OpenSim muscle model. The origin of the torso coordinate frame is located behind the shoulder.

**Table 1.** Summary of the muscle model.

Muscle	Abbreviation	2nd Curve fit	$R^2$
Shoulder			
Anterior Deltoid	DEL1	$L_1 = -7e-4\theta_1^2 - 0.6\theta_1 + 102.6$	0.99
Posterior Deltoid	DEL3	$L_2 = -1e-3\theta_1^2 + 0.7\theta_1 + 121$	0.98
Elbow			
Triceps Long Head	TRllong	$L_3 = -1e-3\theta_2^2 + 1.1\theta_2 + 100$	>0.99
Triceps Medial Head	TRlmed	$L_4 = 3e-4\theta_2^2 + 0.3\theta_2 + 68.2$	0.99
Triceps Lateral Head	TRllat	$L_5 = 2e-3\theta_2^2 + 0.5\theta_2 + 71.4$	0.99
Biceps Brachii (long head)	BIClong	$L_6 = -5e-3\theta_2^2 - 6e-2\theta_2 + 166$	0.99
Biceps Brachii (short head)	BICshort	$L_7 = -5e-3\theta_2^2 + 0.1\theta_2 + 197.4$	0.97
Brachial	BRD	$L_8 = -8e-3\theta_2^2 - 0.2\theta_2 + 194.4$	0.99
Fictitious	Z	$L_9 = \theta_3$	1.0

actual kinematic equations would differ only slightly from these second-order approximations. Another approximation in Table 1 relates to independence. Specifically, BIClong, BICshort, and TRlong actually change in length for both elbow rotation and shoulder rotation. However, they are modeled as only changing in length in response to elbow flexion and extension.

An anatomically fictitious muscle is included in equation 24, and in Table 1 as muscle Z. Z varies 1:1 with  $\theta_3$ . Not only is prismatic translation anatomically incorrect but also the Z muscle has no antagonist. Inclusion of Z relates to the fact that the four plane orientations given by equations 1–4 establish a coordinate system that is intrinsically three dimensional. For example, consider the test trajectory. The test trajectory is constrained to a plane that is parallel to the  $x$ - $y$  plane (not to be confused with position gradient planes). However, the virtual points used for interpolation reside above and below the trajectory plane. Those virtual points must have nonzero elements in  $\mathbf{S}$ . In order to populate those MLSs in  $\mathbf{S}$ , the hand must be allowed to traverse space in the  $z$  direction during random movements. Therefore,  $\mathbf{S}$  was populated by allowing Z to translate through a distance of  $6d/\sqrt{2} \approx 170$  mm.

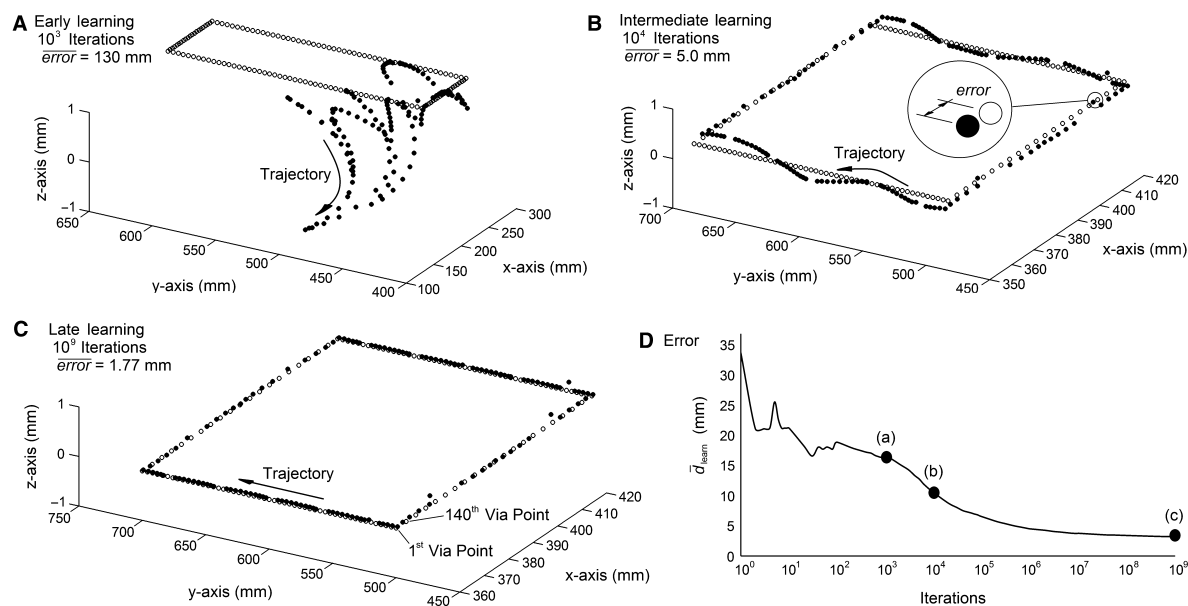
## Results

Depicted in Figure 4A–C are three test trajectories at various stages in the learning process. Hollow circles are

target points,  $\mathbf{p}_D$ , and solid point indicate the position of the hand,  $\mathbf{p}$ , using the interpolated muscle lengths. After only  $10^3$  iterations, set  $\mathbf{S}$  is insufficiently learned and equation 18 is unable to accurately interpolate muscle lengths. Accordingly,  $\overline{\text{error}} = 130$  mm, see (A). By  $10^9$  iterations,  $\mathbf{p}_D$  and  $\mathbf{p}$  have more overlap and  $\overline{\text{error}}$  drops to only 1.77 mm, see (C). Thus, Figure 4A–C demonstrates the ability of equation 18 to interpolate muscle lengths. It also demonstrates the ability of equation 19 to learn kinematics starting with a matrix of zeros.

Depicted in Figure 4D is a plot of  $\overline{d}_{\text{learn}}$  versus the number of iterations. As the number of iterations continues beyond  $10^9$ ,  $\overline{d}_{\text{learn}}$  is expected to continue approaching zero. Accordingly, the approximations given by equation 18 would continue to improve, albeit slightly. The relationship between  $\overline{d}_{\text{learn}}$  and  $\overline{\text{error}}$  is apparent from the (A), (B), and (C) reference points included in (D).

By  $10^9$  iterations,  $\overline{d}_{\text{learn}}$  is 3.21 mm, see Figure 4D. Though this number of iterations might appear large, it is not unreasonable. This computer simulation is inherently discrete. Each iteration simulates the hand moving instantaneously from one point to another. In reality, the hand traverses space in a continuous manner. To justify this level of iteration, let us consider the likelihood of achieving a  $\overline{d}_{\text{learn}} = 3.21$  mm using a continuous trajectory. The geometry of virtual points described by equation 12 is somewhat complicated. As a rough approximation, assume that the hand randomly sweeps across space and that the distance between virtual points is  $d$ . As such, the



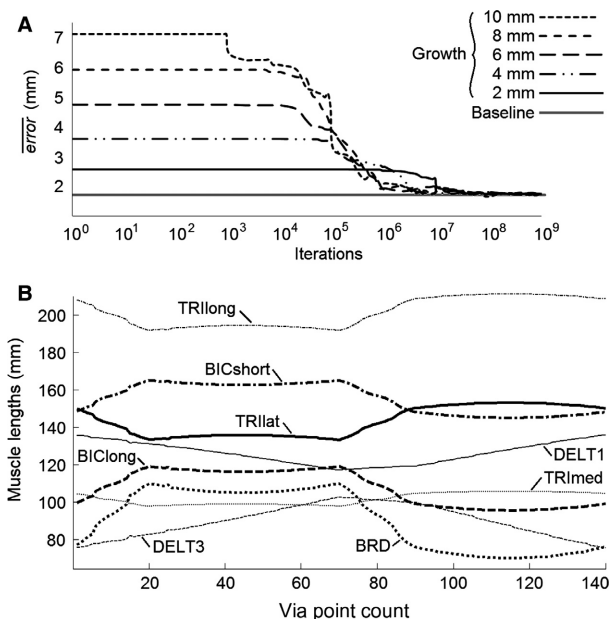
**Figure 4.** Parts A–C depict the rectangular test trajectory that is generated during various stages of learning. Note, (A) is rescaled but the trajectory is identical to (B) and (C). Depicted in (D) is the average distance between learned solutions and virtual points.

hand-to-virtual point proximity will range from 0 (hand is incident on the virtual point) to  $d/2$  (hand is in between two virtual points). The probability that the hand would pass a virtual point with a proximity of between 0 and 3.21 mm is given by  $3.21/(0.5d) = 0.16$ . If it is assumed that the distance between virtual points is approximately  $d$ , then for every meter that the hand travels, it sweeps past  $1/d = 25$  virtual points. For every meter that the hand travels, it will near roughly  $0.16 \times 25 = 4$  virtual points with a proximity  $\leq 3.21$  mm. Therefore,  $\bar{d}_{\text{learn}}$  seems reasonable although  $10^9$  iterations were required using simulation.

Recall that equation 5 included  $c = 0.389$  and  $d = 40$ . A justification for these constants is now provided. Recall that the tuning curve was modeled with a normal distribution. Increasing  $c$  effectively spreads the position gradient activity across a larger volume. In other words,  $cd$  in equation 9 substitutes for the “standard deviation,” or  $\sigma$ . Constant  $c$  was obtained iteratively by minimizing  $\overline{\text{error}}$ . Constant  $c$  equal to 0.389 optimally reduces  $\overline{\text{error}}$  independently of  $d$ . From the perspective of a sensitivity analysis,  $\overline{\text{error}}$  doubles when  $c$  increases by 50% or decreases by 15%. Therefore, interpolation in equation 18 depends critically on  $c$ . This also suggests that interpolation predominantly involves virtual points in the immediate vicinity of  $\mathbf{p}_D$ . Note, equation 19 was comparatively insensitive to  $c$  in that variations in  $c$  had no noticeable effect on proprioceptive learning or adaption.

These considerations beg the question as to why a plane spacing of  $d = 40$  mm was selected. Using a plane spacing of 40 mm with minimal error translates to an  $\sigma = 15.56$  mm. Being derived from a normal distribution, this means that the 95% of the position gradient discharge rate activity would occur within  $4\sigma$ , or approximately 62 mm. While there is no known model that describes position gradients in exactly the same way that they are described in this work, graphical depictions of position gradient data was provided in (Kettner et al. 1988). In that work, position gradient activity was presented within a 152 mm cubic workspace. The 62 mm wide swath of activity that is assumed in this work approximately matches the swath of activity depicted in (Kettner et al. 1988).

Proprioceptive adaption is depicted in Figure 5. The numbers of iterations are depicted along a logarithmic scale. Again, the plots in Figure 5A depict proprioceptive adaption after the upper arm and forearm are each lengthened by 1, 2, 3, 4, or 5 mm. For the first  $10^3$ – $10^5$  iterations, the hand moves using an  $\mathbf{S}$  matrix that was learned with the pregrowth kinematics. As such, the new kinematics is not yet learned and the growth causes an error that is proportional to the growth. By  $10^9$  iterations, equation 19 has updated  $\mathbf{S}$  to account for the growth and



**Figure 5.** Proprioceptive adaption is summarized in (A). Depicted in (B) are the interpolated muscle lengths that guide the hand through the test trajectory.

the  $\overline{\text{error}}$  is comparable to the pregrowth baseline  $\overline{\text{error}}$  of 1.77 mm. Again, even though these levels of growth are exaggerated, they demonstrate the ability of equation 19 to adapt to changing kinematics through maturation.

The interpolated muscle lengths after  $10^9$  iterations are depicted in Figure 5B. These muscle lengths guide the hand to via points plotted in Figure 4C. Though the muscle lengths in Figure 5B are plotted as continuous lines, they are actually discrete sets of muscle lengths for each via point. Ordinarily, such calculations would require inverse kinematic solutions. The results depicted in Figure 5B are of particular importance because they graphically demonstrate the ability of equation 18 to calculate muscle lengths. In other words, Figure 5B demonstrates the ability of equation 18 to estimate inverse kinematics in muscle space given desired points in Euclidian space.

## Discussion

Position gradients and convergent force fields were exploited in this work. Through computer simulation, it was shown that these physiological mechanisms can be structured in a way to learn kinematics, select muscle lengths in order to position the hand as desired, and to adapt to changing kinematics through maturation. Fundamentally, the hypothesis is simple. Muscle lengths are calculated using a weighted sum of MLSs. Learning was accomplished using a simplified condition statement

involving those weights. This construction, if true, helps to bridge the gap between low-level muscle enervations in muscle space with high-level cortical representations of positioning in the Euclidian space. Moreover, position gradients were modeled as tuning curves. In turn, tuning curves were used as a means of spatial interpolation for the hand. The resulting patterns of activity resemble patterns of activity observed in grid cells. An analogous mechanism might be at work for spatial navigation tasks involving grid cells. Given that tuning curves are also observed in connection to auditory and visual processing, tuning curves may provide a similar function for auditory and visual interpolation.

Even though this model was constructed from experimentally determined findings from the literature, several key assumptions were made. The position gradients that were modeled as planes used orientations and plane spacing that were selected ad hoc. Despite the fact position gradients were found in animals, the various orientations that they assume are unknown. With respect to plane spacing, the simulated distances were 40 mm. A smaller plane spacing would result in more virtual points, thereby increasing the information storage demands on the CNS. However, this would provide the benefit of reducing position error. Experimentally determined proprioceptive position errors are significantly larger than the 1.77 mm average error achieved in this work (Van Beers et al. 1998). Therefore, plane spacing much less than 40 mm would seem unlikely.

This model makes several specific predictions about position gradients. First, for a given limb, there should be a minimum of four different position gradient orientations. Second, position gradients of a given orientation should repeat with equal spacing. Third, there should exist locations where four planes, of different orientation, converge at a common point. These are theoretically derived predictions. Given that the location and orientation of position gradients are measurable, these predictions are not only specific, they are falsifiable.

As was alluded to in the Methods section, position gradients are used in two distinct ways. Equation 18 uses position gradients to weight MLSs as a means of interpolation. That sort of calculation lends itself to motor planning and the generation of efferent commands. Equation 19 uses position gradients to update MLSs as a means of achieving proprioceptive learning. That sort of calculation lends itself to reafferent processing. Accordingly, position gradients were observed in both the motor cortex, and area 5 for somatosensory association (Ashe and Georgopoulos 1994). Given that the motor cortex is associated with dispatching efferent information, and area 5 is associated with processing reafferent information (Georgopoulos et al. 1984), it stands to reason that equation 18

would model activity in the motor cortex while equation 19 models activity in area 5. Assuming that MLSs reside in subcortical structures, possibly in interneurons of the spinal cord, the hypothesis predicts that area 5 should have projections along the spinal cord whereby MLSs are continually evaluated and updated. Recall that Figure 2 depicted levels of activity that were allegedly similar to grid cell behavior. A fourth, and final prediction is that descending commands to MLSs should exhibit grid cell-like activity. More specifically, for each nerve fiber that projects onto an MLS, equation 18 should generate increasing levels of activity as the hand nears the associated virtual point. Assuming that MLSs are retained within spinal grey matter, this activity should be present along descending pathways within the spinal cord.

## Acknowledgment

This work was partially funded by a 2015 faculty development grant, “Robotic Biomimicry of Muscle Control” from the Office of Research and Sponsored Programs at California State University, Chico.

## Conflict of Interest

None declared.

## Appendix I

This MATLAB script generates the plot in Figure 2B and D using equation 10. The volume under the plot depicts a convolution of discharge rate along the  $x$ - $y$  plane.

```
z = 1;
c = 0.389;
d = 40; %Distance between planes
U = [1/2 1/2 1/sqrt(2); -1/2 1/2 1/sqrt(2); 1/2 -1/2
1/sqrt(2); ...
-1/2 -1/2 1/sqrt(2)]; %Plane directions as unit vector
inc = 0.3; %Set to 1.0 for accurate axis scaling.
for i = 1:500
for j = 1:500
impulses = 0; %Represents action potential rate
P = [i*inc,j*inc,z];
for k = 1:4;
D = [dot(P,U(1,:)) dot(P,U(2,:)) dot(P,U(3,:)) ...
dot(P,U(4,:))];
I = round(D/d); %Plane index
for m = 1:4
W = 1/(d*c*sqrt(2*pi()))*exp(-(D(m)...
-I(m)*d)^2/(2*(c*d)^2));
Impulses = impulses + W;
end
```

```

end
IMPs(i,j) = impulses;
end
end
axis('off'), surface(IMPs, 'EdgeColor', 'none')

```

## Appendix II

Consider the plane orientation given by equations 1–4. For each index where  $i = j = k = l$ , all four planes have a common intercepts at the  $x$ ,  $y$ , and  $z$  axes. Given that each unit vector forms a  $45^\circ$  angle with the  $x$ - $y$ , plane it follows that the triangle formed by the unit vector, the axes, and the plane forms a  $45^\circ$  right triangle. Given that the hypotenuse of these triangle are length  $d$  times the index, it follows that the intercepts are at the product of the index, the plane distance, and the  $\sqrt{2}$ . Given these intercepts, equation 1 is rewritten as equations of a plane as follows,

$$\frac{1}{2}(x_1 = 0) + \frac{1}{2}(y_1 = 0) + \frac{1}{\sqrt{2}}(z_1 = \sqrt{2}di) + C = 0 \quad (\text{A1})$$

where subscript 1 indicates that this is the plane equation for (1). Solving for C in equation A1 we find that  $C = -di$ . By similar reasoning, the plane equations for 1–3 are

$$\frac{1}{2}x_1 + \frac{1}{2}y_1 + \frac{1}{\sqrt{2}}z_1 - di = 0 \quad (\text{A2})$$

$$-\frac{1}{2}x_2 + \frac{1}{2}y_2 + \frac{1}{\sqrt{2}}z_2 - dj = 0 \quad (\text{A3})$$

$$\frac{1}{2}x_3 - \frac{1}{2}y_3 + \frac{1}{\sqrt{2}}z_3 - dk = 0 \quad (\text{A4})$$

where subscripts 2 and 3 indicate that these are the plane equations for equations 2 and 3, respectively. For these plane equations to mutually intersect at virtual points, the planes must be independent. In other words, the four family of planes must have different orientations. Accordingly, the location of virtual points occurs when  $x_1 = x_2 = x_3$ ,  $y_1 = y_2 = y_3$ , and  $z_1 = z_2 = z_3$ . This system of plane equations is expressed as follows,

$$\begin{bmatrix} di \\ dj \\ dk \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{\sqrt{2}} \\ \frac{1}{2} & -\frac{1}{2} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} x_{ijk} \\ y_{ijk} \\ z_{ijk} \end{bmatrix} \quad (\text{A5})$$

Given that all three planes are nonparallel, (A5) is full rank. Solving simultaneously obtains the following,

$$x_{ijk} = d(i - j) \quad (\text{A6})$$

$$y_{ijk} = d(i - k) \quad (\text{A7})$$

$$z_{ijk} = \frac{d}{\sqrt{2}}(j + k) \quad (\text{A8})$$

## References

- Andersen, O. K. 2007. Studies of the organization of the human nociceptive withdrawal reflex. Focus on sensory convergence and stimulation site dependency. *Acta Physiol.* 189:1–35.
- Arimoto, S., M. Sekimoto, H. Hashiguchi, and R. Ozawa. 2005. Natural resolution of ill-posedness of inverse kinematics for redundant robots: a challenge to Bernstein's degrees-of-freedom problem. *Adv. Robot.* 19:401–434.
- Ashe, J., and A. Georgopoulos. 1994. Movement parameters and neural activity in motor cortex and area 5. *Cereb. Cortex* 4:590–600.
- d'Avella, A., and E. Bizzi. 2005. Shared and specific muscle synergies in natural motor behaviors. *Proc. Natl Acad. Sci. USA*, 102:3076–3081.
- Bernstein, N. 1967. *The coordination and regulation of movements.* Pergamon, London.
- Bizzi, E., and W. K. Abend. 1985. Control of multijoint movement. *Comparative neurobiology, modes of communication in the nervous system.* Wiley, New York, pp 255–277.
- Bizzi, E., N. Accornero, W. Chapple, and N. Hogan. 1983. Posture control and trajectory formation during arm movement. *J. Neurosci.* 4:2738–2744.
- Bizzi, E., F. A. Mussa-Ivaldi, and S. Giszter. 1991. Computation underlying the execution of movement: a biological perspective. *Science* 25:2877–291.
- Bizzi, E., N. Hogan, F. A. Mussa-Ivaldi, and S. Giszter. 1992. Does the nervous system use equilibrium-point control to guide single and multiple joint movements? *Behav. Brain Sci.* 15:603–613.
- Bliss, T. V. P., and G. L. Collingridge. 1993. A synaptic model of memory: long-term potentiation in the hippocampus. *Nature* 361:31–39.
- Carandini, M., and D. J. Heeger. 1994. Summation and division by neurons in primate visual cortex. *Science* 264:1333–1336.
- Chen, X. Y., and J. R. Wolpaw. 2002. Probable corticospinal tract control of spinal cord plasticity in the rat. *J. Neurophysiol.* 87:645–652.

- Cheung, V., A. d'Alvella, M. Tresch, and E. Bizzi. 2005. Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors. *J. Neurosci.* 25:6419–6434.
- Cheung, V., L. Piron, M. Agostini, S. Silvoni, A. Turolla, and E. Bizzi. 2009. Stability of muscle synergies for voluntary actions after cortical stroke in humans. *PNAS* 106:19563–19568.
- Danna-Dos-Santos, A., E. Shapkova, A. Shapkova, A. Degani, and M. Latash. 2009. Postural control during upper body locomotor-like movements: similar synergies based on dissimilar muscle modes. *Exp. Brain Res.* 193:565–579.
- Delp, S. L., F. C. Anderson, A. S. Arnold, P. Loan, A. Habib, C. T. John, et al. 2007. Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE Trans. Biomed. Eng.* 54:1940–1950.
- Edgerton, R., N. Tillakaratne, A. Bigbee, R. de Leon, and R. Roy. 2005. Plasticity of the spinal neural circuitry after injury. *Annu. Rev. Neurosci.* 27:145–167.
- Feldman, A. G. 1966. Central and reflex mechanisms of motor control. *Biofizika* 11:667.
- Feldman, A. G. 1986. Once more on the equilibrium-point hypothesis (lambda model) for motor control. *J. Mot. Behav.* 18:17–54.
- Feldman, A. G., and M. L. Latash. 2004. Testing hypotheses and the advancement of science: recent attempts to falsify the equilibrium point hypothesis. *Exp. Brain Res.* 161:91–103.
- Ferrell, W. R., S. C. Gandevia, and D. I. McCloskey. 1987. The role of joint receptors in human kinaesthesia when intramuscular receptors cannot contribute. *J. Physiol.* 386:63–71.
- Flash, T., and B. Hochner. 2005. Motor primitives in vertebrates and invertebrates. *Curr. Opin. Neurobiol.* 15:1–7.
- Full, R. J., and D. E. Koditschek. 1999. Templates and anchors: neuromechanical hypotheses of legged locomotion on land. *J. Exp. Biol.* 202.23:3325–3332.
- Georgopoulos, A. P., R. Caminiti, and J. Kalaska. 1984. Static spatial effects in motor cortex and area 5: quantitative relations in a two-dimensional space. *Exp. Brain Res.* 54:446–454.
- Giszter, S., F. A. Mussa-Ivaldi, and E. Bizzi. 1993. Convergent force fields organized in the frog's spinal cord. *J. Neurosci.* 13:467–492.
- Gizzi, L., J. Nielsen, F. Felici, Y. Ivanenko, and D. Farina. 2010. Impulse of activation but not motor modules are preserved in locomotion of subacute stroke patients. *J. Neurophysiol.* 106:202–210.
- Holzbaum, K. R. S., W. M. Murray, and S. Delp. 2005. A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control. *Ann. Biomed. Eng.* 33:829–840.
- Kaelbling, L. P., M. L. Littman, and A. W. Moore. 1996. Reinforcement learning: a survey. *J. Artif. Intell. Res.* 4:237–285.
- Takei, S., D. Hoffman, and P. Strick. 1999. Muscle and movement representation in the primary motor cortex. *Science* 285:2136–2139.
- Kargo, W., and S. Giszter. 2000. Afferent roles in hindlimb wipe-reflex trajectories: free-limb kinematics and motor patterns. *J. Neurophysiol.* 83:1480–1501.
- Kettner, R. E., A. B. Schwartz, and A. P. Georgopoulos. 1988. Primate motor cortex and free arm movements to visual targets in three-dimensional space. III. Positional gradients and population coding of movement direction from various movement origins. *J. Neurosci.* 8:2938–2947.
- Lemay, M. A., and W. M. Grill. 2004. Modularity of motor output evoked by intraspinal microstimulation in cats. *J. Neurophysiol.* 91:502–514.
- Li, C.-S., C. Padoa-Schioppa, and E. Bizzi. 2001. Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field. *Neuron* 30:593–607.
- McIntyre, J., and E. Bizzi. 1993. Servo Hypothesis for the biological control of movement. *J. Mot. Behav.* 25:193–202.
- Merton, P. A. 1953. *Speculations on the Servo-Control of Movement*. Ciba Foundation Symposium-The Spinal Cord. John Wiley, New York, pp. 247–260.
- Miall, R. C., and D. M. Wolpert. 1996. Forward models for physiological motor control. *Neural Netw.* 9:1265–1276.
- Moran, D. W., and A. B. Schwartz. 1999. Motor cortical representation of speed and direction during reaching. *J. Neurophysiol.* 82:2676–2692.
- Moser, E. I., E. Kropff, and M. B. Moser. 2008. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.* 31:69–89.
- Mussa-Ivaldi, F. A. 1997. Nonlinear force fields: a distributed system of control primitives for representing and learning movements. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 84–90.
- Mussa-Ivaldi, F. A., and J. Patton. 2000. Robots can teach people how to move their arm. *Proceedings for the 2000 IEEE International Conference on Robotics & Automation*, San Francisco, CA, pp. 300–305.
- Paninski, L., M. R. Fellows, N. G. Hatsopoulos, and J. P. Donoghue. 2004. Spatiotemporal tuning of motor cortical neurons for hand position and velocity. *J. Neurophysiol.* 91:515–532.
- Roh, J., V. Cheung, and E. Bizzi. 2011. Modules in the brain stem and spinal cord underlying motor behaviors. *J. Neurophysiol.* 106:1363–1378.
- Scholz, J. P., and G. Schöner. 1999. The uncontrolled manifold concept: identifying control variables for a functional task. *Exp. Brain Res.* 126:289–306.

- Schwartz, A. B., X. T. Cui, D. J. Weber, and D. W. Moran. 2006. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron* 52:205–220.
- Sergio, L. E., and J. F. Kalaska. 1998. Changes in the temporal pattern of primary motor cortex activity in a directional isometric force versus limb movement task. *J. Neurophysiol.* 80:1577–1583.
- Shadmehr, R., and F. A. Mussa-Ivaldi. 1994. Adaptive representation of dynamics during learning of a motor task. *J. Neurosci.* 14:3208–3224.
- Sherrington, C. S. 1947. *The integrative action of the nervous system.* Yale University Press, New Haven. pp. 301–304.
- Simkins, M., A. Burleigh-Jacobs, and J. Rosen. 2014. Stroke-induced synergistic phase shifting and its possible implications for recovery mechanisms. *Exp. Brain Res.* 232:3489–3499.
- Taira, M., J. Bolino, N. Smyrnis, A. Georgopoulos, and J. Ashe. 1996. On the relations between single cell activity in the motor cortex and the direction and magnitude of three-dimensional static isometric force. *Exp. Brain Res.* 109:367–376.
- Thoroughman, K. A., and R. Shadmehr. 2000. Learning of action through adaptive combination of motor primitives. *Nature* 407:742–747.
- Ting, L., and J. Macpherson. 2005. A limited set of muscle synergies for force control during a postural task. *J. Neurophysiol.* 93:609–613.
- Tresch, M. C., and E. Bizzi. 1999. Responses to spinal microstimulation in the chronically spinalized rat and their relationship to spinal systems activated by low threshold cutaneous stimulation. *Exp. Brain Res.* 129:401–416.
- Tresch, M. C., and A. Jarc. 2009. The case for and against muscle synergies. *Curr. Opin. Neurobiol.* 19:601–607.
- Van Beers, R. J., A. C. Sitting, and J. J. Denier Van Der Gon. 1998. The precision of proprioceptive position sense. *Exp. Brain Res.* 122:367–377.
- Vato, A., M. Semprini, E. Maggioni, F. D. Szymanski, L. Fadiga, S. Pnzeri, et al. 2012. Shaping the dynamics of a bidirectional neural interface. *PLoS Comput. Biol.* 8:e1002578.
- Velliste, M., S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz. 2008. Cortical control of a prosthetic arm for self-feeding. *Nature* 453:1098–1101.
- Wessberg, J., C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, et al. 2000. Real-time prediction of hand trajectory by ensembles of cortical neurons in primate. *Nature* 408:361–365.
- Winter, D. A. 2005. *Biomchanics and motor control of human movement*, 3rd ed.. John Wiley & Sons, New York.
- Wolpaw, J. R., and X. Y. Chen. 2006. The cerebellum in maintenance of a motor skill: a hierarchy of brain and spinal cord plasticity underlies H-reflex conditioning. *Learn Mem.* 13:208–215.
- Wolpert, D. M., Z. Ghahramani, and M. I. Jordan. 1995. Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study *Exp. Brain Res.* 103:460–470.
- Won, J., and N. Hogan. 1995. Stability properties of human reaching movements. *Exp. Brain Res.* 107:125–136.