

Supplemental Information for: Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics

Timothy P. Lillicrap^{1,2*}, Stephen H. Scott^{2,3*}

¹Department of Pharmacology,
University of Oxford, Oxford, United Kingdom

²Centre for Neuroscience Studies, ³Department of Biomedical and Molecular Sciences,
Queen's University, Kingston, Ontario, Canada

*To whom correspondence should be addressed:

E-mail: timothy.lillicrap@pharm.ox.ac.uk (T.P.L.); steve.scott@queensu.ca (S.H.S.)

Supplemental Experimental Procedures:

Here we present the experimental, mathematical, and computational details which are necessary to allow replication of our results. We also briefly explain the statistics used to access bimodality in circular distributions.

Empirical results. Single neurons were recorded in primary motor cortex (M1) contralateral to the arm used to perform the task. In all four monkeys, neurons were recorded in the left hemisphere. In the fourth monkey, neurons were also recorded in the right hemisphere while the monkey performed the tasks with its left arm. For the reaching task, the average spike rate was averaged from 100 ms before movement onset (determined as 10% of peak hand velocity) through to peak hand velocity. The tuning of each neuron was assessed using planar regression, with the targets as the independent variable and the average spike rate of the neuron as the dependent variable. The orientation of the fitted plane defined the Preferred Movement Direction (PMD) of the neuron. The PMDs of the neurons recorded in the right hemisphere of the fourth monkey were mirror flipped about the vertical axis. In the posture task, the spike rate was averaged over a two second interval after the monkey had stabilized its hand over the target. Neural tuning was again assessed using planar regression, with the loads as the independent variables and the spike rate as the dependent variable. Muscle data was analyzed in a similar fashion and was taken from a previous study (Kurtzer et al. 2006).

Muscle activation. For muscle activation in our dynamic model, σ_u , we used a smooth version of a ramp function which is 0 for values less than or equal to zero and linear for values greater than zero. We used a smoothed version of the ramp function because gradient descent worked better if nonzero derivatives exist over the range of the function (i.e. tended to get caught in bad local minima less often). The particular function used was $\sigma_u(x) = \exp(2x - 5)$ for values less than $\frac{1}{10}$, and $\sigma_u(x) = 2x \exp(\frac{2}{10} - 5)$ for values greater than or equal to $\frac{1}{10}$. The particular form of this function was chosen to be simple and to hasten optimization. Note that another means of keeping muscle activity positive would be to use a linear function for $u(t)$ and then optimize

subject to the inequality constraint, $u(t) \geq 0$, see for example (Shah et al. 2004; Trainin et al. 2007). We expect that this approach would produce qualitatively similar results.

Sensory feedback filtering. In our dynamic network model, the units $z(t)$ received a filtered version of sensory feedback and goal/load information. The bank of sensory/goal/load filters we call, $v(t)$, and in practice we used 50 filters. An interpretation of these filters is beyond the scope of this paper. We think of them simply as providing our M1 units, $z(t)$, with a distributed and flexible form of the sensory/goal/load information required to produce good control. Note that we explored two other basic network architectures: one in which the bank of filters - or first layer of the network - was simply removed, and one in which the bank of filters was removed and the layer $z(t)$ was allowed recurrent connections amongst its units. Both of these produced qualitatively similar results but the former had difficulty learning good control solutions with the same number of neurons (due to a relative lack of flexibility), and the latter took longer to train due to the number of parameters added by recurrent connections.

Optimization and analysis. We tuned three scalar parameters by hand to ensure a good fit between real and simulated arm kinematics: α and β , which specify the relative importance of keeping the muscle activity and neural activity small, and a third scalar, γ , which specifies the relative importance of keeping network parameters small. In practice all three were set to 10^{-5} , though a wide range of values gave similar results as long as optimization produced realistic kinematics. With the exception of the smooth ramp muscle activation function, $\sigma_u(\cdot)$, which was chosen to keep muscle activity positive and be differentiable and give fast optimization, all other model parameters (i.e. those related to arm and muscle mechanics, see below) were either taken from literature or else optimized using gradient descent.

The training sets for the network consist of random batches of 100 reaches in the accessible workspace of the limb. The network is trained in a minibatch mode where updates for 100 random reaches are calculated and applied for a set number of iterations of conjugate gradient descent, and then another 100 random reaches are drawn and the process is repeated. We also checked after each optimization that learning produced behaviour which looked similar to that executed by the monkeys (i.e. the network performed reaches with bell shaped velocity profiles which were roughly straight in handspace, and qualitatively resemble those produced by monkeys, and maintained posture within 0.25cm of the target in the loaded-posture task).

Then, we examined the distribution of preferences of the network using the Rayleigh test for bimodality (Batschelet 1981; see **Bimodal statistics below**). We repeated this procedure, both optimization and analysis, for 10 networks. Each network had 1000 units in $z(t)$, and the synaptic weights of each network, W^{out} , were initialized randomly (from a Normal distribution with a mean of 0 and variance 0.001) prior to the optimization.

Variations on simulation setup. In the **Experimental Procedures** we have detailed the basic architecture and details of the network model we employed. We tried varying: [1] the size

and structure of the network (e.g. from 100 to 1000 units in $z(t)$), [2] the unit activation function (e.g. using smooth ramp and $\tanh(\cdot)$ functions instead of the standard sigmoid function), [3] the muscle activation function $\sigma_u(\cdot)$ (e.g. using the standard sigmoid instead of the ramp function employed), [4] the regularization scalars α and β which weighted the importance of keeping neural and muscle activity small (e.g. combinations of values between 10^{-4} and 10^{-7}), [5] the distribution from which elements of W^{out} were drawn (e.g. the Uniform distribution centred on 0 and the Normal distribution with different levels of variance), [6] the movement duration (between 300-450ms), [7] integration timestep (between 10-40ms), [8] the form of the peripheral feedback (e.g. giving feedback in hand-based coordinates instead of joint-based coordinates), and [9] the requirement of having a terminal-time cost by instead using an instantaneous cost which penalized both neural/muscle activity and distance to target at each time step, thus converting the cost function to that of an infinite-horizon type problem; in this case, the network was required to find a tradeoff between arriving at the target quickly and keeping neural/muscle activity small.

In all of these cases, as long as the optimization was successful, we observed results qualitatively similar to those reported in the main text. There do of course exist extreme parametrizations of the cost function, and network structures that produce results qualitatively different from those reported. Trivially, if the regularization constants are set too high (e.g. $\alpha > 0.1$) then optimization will fail to find kinematic behaviour resembling empirical data – in these cases, the optimal solution is to simply produce little or no movement, and the usual pattern of unit preferences is correspondingly disrupted.

As well, we found that if the network structure is such that there is not enough flexibility (in terms of the muscle activation patterns achievable), then the unit preferences reported in our results may also be disrupted. We insured flexibility of the unit layer activations, $z(t)$, via either an input filter layer, or else via recurrent connections among the units in $z(t)$. It is likely that simple inclusion of additional units in the $z(t)$ vector would also suffice.

Static model analysis. In the static model, nearly all units (>90%) in both the centre-out and loaded-posture were significantly tuned ($p < 0.05$) to target velocity and target load. Thus, each preference distribution consisted of approximately 10,000 unit preferences. As well, the optima found for each of the 10 repeats for a given condition tended to be qualitatively consistent. For example, if the mean PMD or PTD distribution was found to be significantly bimodal for a condition then all 10 repeats exhibited roughly the same bimodal distribution: within a given condition the standard deviation for distribution orientation, θ , was never greater than 5° and for distribution skews, r , the standard deviation was never greater than 0.01. Finally, it was also found that for every repeat and condition, if the distribution was found to be bimodal, then a bootstrap determined it to be highly significant, with $p < 10^{-3}$ in every case. Thus, we here report only the mean orientation and skew for each condition.

For all the conditions examined using the static model, the dynamic model made qualitatively similar predictions. Thus, at least for the conditions examined here (e.g. where reaches were

over a distance short enough that our local approximations roughly hold) our static model appears to be a good approximation to the full dynamic model.

Computational details. All of the simulations were run in Matlab. Gradient descent was accomplished via a Matlab optimization package called `minFunc` (Mark Schmidt 2005; <http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html>), similar to the popular `minimize.m` (Carl Rasmussen 2006; http://www.mit.edu/~rsalakhu/code_DBM/minimize.m), but found to be faster and able to find deeper minima for our application. From `minFunc`, we used a preconditioned non-linear conjugate gradient (PCG) descent algorithm. Other second order gradient descent algorithms were tried (e.g. quasi-Newton with limited-memory BFGS updating and a preconditioned Hessian-free Newton rule) and gave similar results but were slower than PCG. Standard stochastic gradient descent was also tried and provided similar results but was found to be prohibitively slow to be used for the bulk of our simulations.

All of the gradient descent algorithms tried required user supplied partial derivatives to be compute backward in time via backpropogation through-time modified for our model. This was the most computationally expensive facet of our model and was accomplished via a handcrafted Mex function written in C. Numeric and automatic computation of gradients were tried but found to be extremely slow for a network of even moderate size (i.e. > 10 units). The static model was optimized in the same way as the dynamic model, i.e. using PCG, except that no gradients through time were required.

Musculoskeletal model specification. In our dynamic model, the mapping from muscle state and neural command to joint torques is given by: $\tau(t) = M \cdot \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t))$, where $\mathbf{h}(\cdot, \cdot)$ is the function which computes the force generated by each muscle (taking into account the force-length-velocity function detailed below). Recall that we have described above how the muscle activities, $\mathbf{u}(t)$, are computed from neural activities. We used a moment arm matrix, M , which is a rough average of empirical values (Graham & Scott 2003; Trainin et al. 2007):

$$\begin{pmatrix} 2 & -2 & 0 & 0 & 1.5 & -2 \\ 0 & 0 & 2 & -2 & 2 & -1.5 \end{pmatrix} \quad (1)$$

We included the dependency of force production on muscle length and velocity. Throughout this section, the state of the arm, i.e. the shoulder and elbow angles and velocities, are written for convenience in a couple different ways, $\mathbf{x}(t) = [\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}] = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]$; as well, we use $\ddot{\boldsymbol{\theta}}$ for the joint accelerations. We assume a linear and 1-to-1 mapping between arm state and muscle state. The length and velocity of each muscle, i.e. l_i and \dot{l}_i , are determined by:

$$l_i = 1 + \frac{M_{1,i} \cdot (\theta_{1,i}^0 - \theta_1)}{L_i^0} + \frac{M_{2,i} \cdot (\theta_{2,i}^0 - \theta_2)}{L_i^0} \quad (2)$$

$$\dot{l}_i = \frac{M_{1,i} \cdot \dot{\theta}_1}{L_i^0} + \frac{M_{2,i} \cdot \dot{\theta}_2}{L_i^0} \quad (3)$$

where the subscripts index into the relevant matrix or vector and L^0 and θ^0 are the optimal length matrix and joint angle vector given by:

$$\theta^0 = 2\pi \begin{pmatrix} 15.0 & 4.88 & \text{n/a} & \text{n/a} & 4.5 & 2.12 \\ \text{n/a} & \text{n/a} & 80.86 & 109.32 & 92.96 & 91.52 \end{pmatrix} / 360 \quad (4)$$

and,

$$L^0 = (7.32 \quad 3.26 \quad 6.4 \quad 4.26 \quad 5.95 \quad 4.04) \quad (5)$$

In all of these matrices, the column indicates the muscle group and the row (if there is more than one) indicates the joint. Note that in some cases the optimal angle is given as n/a - indicating that there is no optimal angle for that muscle about that joint. In these equations for length and velocity, n/a is simply treated as 0. We used averages, weighted by the normalized physiological cross sectional area (PCSA) of each muscle, for the optimal muscle length, and optimal joint angles (given as, θ^0 , and L^0) (Cheng & Scott 2000). The normalized PCSA of each muscle was calculated as the PCSA of the muscle divided by the total PCSA of the relevant muscle group (Graham & Scott 2003; **monoarticular shoulder flexors**: Pectoralis major (clavicular and sternocostal head), Deltoid anterior, **monoarticular shoulder extensors**: Deltoid posterior, Deltoid middle, **monoarticular elbow flexors**: Brachialis, Brachioradialis, Extensor carpi radialis longus, **monoarticular elbow extensors**: Triceps lateral, Triceps long, **biarticular flexors**: Biceps long, Biceps short, **biarticular extensors**: Dorsopitrochlearis, Triceps long).

The dependence of force on the length and velocity (this function is often referred to as the force-length and force-velocity curves) of the muscle (in normalized units of L^0 and L^0 / second respectively) is here given by a modified version of the equation proposed by Brown et al. 1999:

$$f_l(l) = \exp\left(\text{abs}\left(\frac{l^\beta - 1}{\omega}\right)\right) \quad (6)$$

$$f_{fv}(l, \dot{l}) = \begin{cases} \frac{V_{max} - \dot{l}}{V_{max} + (c_{V0} + c_{V1})\dot{l}}, & \dot{l} \leq 0 \\ \frac{b_V - (a_{V0} + a_{V1}\dot{l} + a_{V2}\dot{l}^2)\dot{l}}{b_V + \dot{l}} & \dot{l} > 0 \end{cases} \quad (7)$$

These functions are poorly behaved, both at zero velocity (e.g. the force-velocity function has a jump discontinuities in its first derivative), and outside of the physiological range over which they were fit (e.g. they grow very large very quickly). In order to make sure that learning with conjugate gradient descent (which depends on the cost function and its derivatives being smooth) worked properly, we consider the combined force-length and force-velocity function, which we will call the force-length-velocity curve. This function, given as $f_{flv} = f_l(l) \cdot f_v(l, \dot{l})$, scales muscle force multiplicatively.

This function has all of the undesired qualities mentioned above: jump discontinuities in its derivatives, and exponential growth outside of the physiological ranges for muscle length and velocity. Thus, we replaced this function in our model with a simple, feedforward sigmoidal neural network with 5 hidden nodes. We will designate this network function by, $\hat{f}_{flv}(\cdot, \cdot)$. The network was trained to approximate the $f_{flv}(\cdot, \cdot)$ function. The network closely approximates the force-length-velocity curve over the range of interest (an error of <0.0001 for all points in the range of physiological interest), has smooth derivatives everywhere, and is nicely bounded (between 0 and 2). The parameters for the force-length and force-velocity curves are given by (Brown et al. 1999): $\beta = 1.55$, $\omega = 0.81$, $\rho = 2.12$, $V_{max} = -7.39$, $c_{v0} = -3.21$, $c_{v1} = 4.17$, $b_v = 0.62$, $a_{v0} = -3.12$, $a_{v1} = 4.21$, $a_{v2} = -2.67$. Thus, since the force-length-velocity curve scales force multiplicatively, the joint torques are given by: $\tau(t) = M \cdot h(x(t), u(t)) = M \cdot [u(t) \bullet \hat{f}_{flv}(l, \dot{l})]$, if $\hat{f}_{flv}(l, \dot{l})$ is a vector version of the force-length-velocity network function and \bullet is the element-wise product.

Next, in our model the mapping from arm state and joint torques to arm state change given by: $\Delta x(t+1) = f(x(t), \tau(t))$. In particular, we used the forward dynamics of the two joint limb given by (Todorov 2004):

$$\ddot{\theta} = \mathcal{M}(\theta)^{-1} (\tau - C(\theta, \dot{\theta}) - \mathcal{B}\dot{\theta}) \quad (8)$$

$$\mathcal{M}(\theta) = \begin{pmatrix} a_1 + 2a_2 \cos(\theta_2) & I_2 + \cos(\theta_2) \\ I_2 + \cos(\theta_2) & I_2 \end{pmatrix} \quad (9)$$

$$C(\theta, \dot{\theta}) = a_2 \cos(\theta_2) \begin{pmatrix} -\dot{\theta} (2\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{\theta}_2^2 \end{pmatrix} \quad (10)$$

$$\mathcal{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} \quad (11)$$

where, $a_1 = I_1 + I_2 + m_2 l_1^2$ and $a_2 = m_2 l_1 c_1$. The length, mass, moment of inertia and distance from proximal end to the centre of mass of the two segments are: $l_1 = 0.145$ and $l_2 = 0.284$

metres; $m_1 = 0.2108$ and $m_2 = 0.1938$ kilograms; $I_1 = 0.025$ and $I_2 = 0.045$ kilograms · metres²; $c_1 = 0.0749$ and $c_2 = 0.0757$ metres, respectively (Cheng & Scott 2000).

Finally, the mapping from joint coordinates and velocities to the hand position and velocity in cartesian space, $\mathbf{y}(t)$, is accomplished via the mapping:

$$\mathbf{y}(t) = \mathbf{g}(\mathbf{x}(t)) = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \\ -\dot{\theta}_1 (l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)) + \dot{\theta}_2 (-l_2 \sin(\theta_1 + \theta_2)) \\ \dot{\theta}_1 (l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2)) + \dot{\theta}_2 (-l_2 \cos(\theta_1 + \theta_2)) \end{bmatrix} \quad (12)$$

Note that in this context l_1 and l_2 refer to segment lengths.

Static model plant approximation. For the static model we approximated the dynamic mappings, $\mathbf{f}(\cdot, \cdot)$, $\mathbf{g}(\cdot)$, $\mathbf{h}(\cdot, \cdot)$, and M by linearizing their time averaged effect across the first half of the movement in the case of the reaching task (in the posture task the time average and momentary values for these mappings are the same near the optima and so no time averaging was required). These time averaged linearization of the dynamic mappings are respectively called, F , G , H , and M .

To examine the effect that various abstractions of the musculoskeletal model had on preference distributions, these four matrices were appropriately modified. Muscle F-L/V properties were removed from the model by setting all the elements of the scaling matrix, H , to 1. To remove biarticular muscles from the model the corresponding columns of the moment arm matrix M were removed so that the network could only control the limb via the 4 remaining monoarticular muscles. To remove “muscles” from the model entirely, so that the network effectively controlled the limb via a torque actuator at each joint, we removed the non-linear muscle activation function and set the moment arm matrix M to be the 2×2 identity matrix; i.e. $\mathbf{u} = W^{\text{out}} \cdot \mathbf{z}$. To remove intersegmental terms from the arm dynamics so that the network controlled each joint independently we set the off diagonal terms of F to be zero. Finally, we dropped the geometry of the limb by setting G to be the 2×2 identity matrix, simplifying the limb to a point-mass controlled by forces.

Bimodal statistics. We characterized the distributions of preferred movement and preferred torque directions (given as angles ϕ_i for $i \in \{1, \dots, N\}$ where N is the number of neuron preferred directions) using the Rayleigh r statistic (Batschelet 1981). In the unimodal case the statistic captures the skew of the distribution and is given by:

$$r = \frac{1}{N} \left(\left(\sum_{i=1}^N \cos(\phi_i) \right)^2 + \left(\sum_{i=1}^N \sin(\phi_i) \right)^2 \right) \quad (13)$$

which implies that $r \in [0, 1]$. An r of 0 implies that the distribution of angles is perfectly uniformly distributed around the circle. An r of 1 implies that the angles are all oriented in the same direction. In the paper we have focused primarily on whether a given simulation produced a bimodal distribution in preferred directions since the empirical distributions of interest were bimodally distributed. In order to assess the bimodality of a set of angles we use a version of the Rayleigh designed for this purpose where the angles in the distribution are simply doubled:

$$r = \frac{1}{N} \left(\left(\sum_{i=1}^N \cos(2\phi_i) \right)^2 + \left(\sum_{i=1}^N \sin(2\phi_i) \right)^2 \right) \quad (14)$$

Note that this statistic is designed to look for bimodal distributions where the two modes are oriented at 180 degrees from each other. It is not designed to deal with bimodal distributions where the modes are otherwise oriented.

To assess the significance of these scores we used a simple bootstrap technique wherein we drew N samples 100,000 times from a uniform distribution on the interval $[0, 2\pi]$. For each draw of N samples, we computed the unimodal and bimodal Rayleigh r scores. We then examined the likelihood that a particular r observed in an empirical or simulated distribution with N samples happened by chance by computing where in the bootstrapped distribution it fell.

If a distribution of angles is determined to be significantly skewed, we find the dominant axis of the skew, θ , using the following formula (Batschelet 1981):

$$\theta = \arctan \left(\frac{\sum_{i=1}^N \sin(\phi_i)}{\sum_{i=1}^N \cos(\phi_i)} \right) \quad (15)$$

In the bimodal case we use the modified version:

$$\theta = \text{mod} \left(\frac{1}{2} \arctan \left(\frac{\sum_{i=1}^N \sin(\phi_i)}{\sum_{i=1}^N \cos(\phi_i)} \right) \right) \quad (16)$$

which gives an angle in the range $[0, \pi]$.

Biased experience. For these simulations, the plant dynamics were simplified to be a 2D pointmass. To simulate learning with biased spatial experience in artificial networks, we had them learn to make reaches where the distribution over movement directions was skewed. Target directions for training reaches were drawn randomly from a bimodal von Mises distribution with the two modes oriented straight forward and backwards. The standard von Mises

distribution is roughly equivalent to a Gaussian distribution over the range $[-\pi, \pi]$ and is given by:

$$f(x|\mu, \kappa) = \frac{\exp(\kappa \cos(x - \mu))}{2\pi I_0(\kappa)} \quad (17)$$

and the bimodal version of this distribution is given by:

$$f(x|\mu, \kappa) = \frac{\exp(\kappa \cos(2(x - \mu)))}{2\pi I_0(\kappa)} \quad (18)$$

where, $I_0(\cdot)$, is the modified Bessel function of order 0. In the simulations we parameterized the distribution using $\mu = \pi/2$, $\kappa = 1/2$, for a skewed distribution with most reaches oriented away from the body or towards it, and very few reaches to the side. Because this distribution is highly skewed, we included the standard centre out reaches in the training set in order to ensure that the network would eventually be able to execute these reaches.

For the 2D point-mass the network outputs and plant dynamics were given as follows: the output from the network is converted to the two forces applied to the mass via $\mathbf{u}(t)$, where $\sigma_{\mathbf{u}}(\cdot)$ was the standard network sigmoid function, so that the elements of $\mathbf{u}(t)$ lie in the interval $[-10, 10]$. The point-mass state evolves according to,

$$\Delta \mathbf{x}(t) = \mathbf{A}\mathbf{x}(t-1) + \mathbf{B}\mathbf{u}(t) \quad (19)$$

where,

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (20)$$

and,

$$\mathbf{B} = \begin{pmatrix} \frac{(\Delta t)^2}{2m} & 0 \\ 0 & \frac{(\Delta t)^2}{2m} \\ \frac{\Delta t}{m} & 0 \\ 0 & \frac{\Delta t}{m} \end{pmatrix} \quad (21)$$

where, m , is the mass of the point (taken to be 1 in simulations) and Δt is the time step for Euler integration (which, as mentioned above, was taken to be 20 ms). In the case of the

point-mass we compute the observed state, $y(t)$, operated on by the cost function via the identity, $y(t) = g(x(t)) = I \cdot x(t)$, where I is the 4×4 identity matrix.

Supplemental References:

Batschelet, E. (1981). *Circular Statistics in Biology*. New York: Academic Press.

Brown, I.E., Cheng, E.J., Loeb, G.E. (1999). Measured and modeled properties of mammalian skeletal muscle. II. The effects of stimulus frequency on force-length and force-velocity relationships. *J. Muscle Res. Cell Motil.* 20, 627-643.

Cheng, E.J., Scott, S.H. (2000). Morphometry of *Macaca Mulatta* Forelimb. I. Shoulder and elbow muscles and segment inertial parameters. *J. Morphol.* 245, 206-224.

Graham, K.M., Scott, S.H. (2003). Morphometry of *Macaca Mulatta* Forelimb. III. Moment arm of Shoulder and Elbow Muscles. *J. Morphol.* 255, 301-314.

Kurtzer, I., Herter, T.M., Scott, S.H. (2006) Nonuniform distribution of reach-related and torque-related activity in upper arm muscles and neurons of primary motor cortex. *J. Neurophysiol.* 96, 3220-3230.

Li, W., Todorov, E. (2004). Iterative linear-quadratic regulator design for nonlinear biological movement systems. In proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics, vol 1, 222-229.

Shah, A., Fagg, A.H., Barto, A.G. (2004). Cortical involvement in the recruitment of wrist muscles. *J. Neurophysiol.* 91, 2445-2456.

Trainin, E., Meir, R., Karniel, A. (2007). Explaining patterns of neural activity in the primary motor cortex using spinal cord and limb biomechanics models. *J. Neurophysiol.* 97, 3736-3750.