Impedance Control in Brain-Machine Interfaces through a Physically-based Musculoskeletal Model of the Arm

Hyun K. Kim¹
Jose M. Carmena²
James Biggs¹
Timothy L. Hanson²
Miguel A. L. Nicolelis²,³
Mandayam A. Srinivasan¹

¹ The Touch Lab, Massachusetts Institute of Technology
² Department of Neurobiology, Center for Neuroengineering, Duke University
³ Department of Biomedical Engineering, Duke University

Hyun K. Kim
Graduate Research Assistant, The Touch Lab
Massachusetts Institute of Technology
Department of Mechanical Engineering
Research Laboratory of Electronics
50 Vassar St., Room 36-776A
1-(617) 253-2759
Email: hyunkim@mit.edu

S. James Biggs, Ph.D.
Research Scientist, The Touch Lab
77 Massachusetts Ave, 36-789
Cambridge, MA 02139
Phone: (617) 253-2598
Fax: (617) 258-7003
Email: jbiggs@mit.edu

Jose M. Carmena, Ph.D.
Research Scientist
Department of Neurobiology
Center for Neuroengineering
Duke University Medical Center
327 Bryan Research Bldg.
Durham, NC 2771
+1 919 668 1235 (office)
+1 919 684 0229 (lab)
+1 919 681 7055 (fax)
Email: carmena@neuro.duke.edu
Miguel A. L. Nicolelis, M.D., Ph.D.
Professor of Neurobiology and Biomedical Engineering.
Co-Director Center for Neuroengineering
Department of Neurobiology
Duke University Medical Center
101 Research Drive
Durham, NC 27710
+1 919 684 4580 (office)
+1 919 688 2248 (fax)
Email: nicoleli@neuro.duke.edu
webpage: http://www.nicolelislab.net

Mandayam Srinivasan
Director, The Touch Lab
77 Massachusetts Ave, 36-791
Cambridge, MA 02139
Phone: (617) 253-2512
Fax: (617) 258-7003
Email: srini@mit.edu
Abstract

Current demonstrations of Brain-Machine Interfaces have shown the possibility of controlling neuro-prostheses under pure motion control extremely successfully. For interaction with objects, pure motion control lacks the information required for versatile manipulation. This study investigates the idea of applying impedance control in BMI system. A musculoskeletal arm model based extraction algorithm was developed for this purpose. The new algorithm was tested on cortical recordings from a behaving monkey. The algorithm was able to predict motion parameters to an equivalent level of accuracy to that of a traditional extraction algorithm. Furthermore, it was able to predict interactions with external force fields that previous algorithms would not be able to predict.

1. Introduction

Research in Brain-Machine Interfaces (BMIs) has flourished with recent technological advances in measuring electrical activity of populations of single cortical neurons through chronic implants. There have been numerous demonstrations of non-human primates controlling robots or graphical cursors in real-time through signals collected from their cortical areas. These demonstrations can be divided largely into two categories; continuous control of position [1-5] and discrete control of more abstract information such as intended targets, intended actions and onset of movements[6-8].

In BMI systems that use the continuous control of position strategy, much of the work thus far has treated the primate as a pure motion source. This makes intuitive sense considering the numerous studies that suggest cortical activity are broadly tuned to higher-level features of hand movement such as position [9] and velocity [10, 11]. Using this approach, neural recordings has been successfully correlated to predict hand trajectory to a reasonable degree of accuracy [1-5]. Such a strategy proved to be an appropriate first pass at BMI, confirming the potential for decoding movement features from cortical areas to drive neuroprotheses for paralyzed patients.

In this paper, we discuss the next logical step. A motion source has infinite mechanical impedance. It is the complement of a force source, which has zero mechanical impedance. These ideal sources have behavior independent of the load they drive. However, the mammalian neuromuscular system is neither an ideal force source nor an ideal motion source, and its mechanical impedance is far from either of the two extremes. In fact, the neuromuscular system is highly adaptive and can modulate mechanical impedance. Numerous studies show that this modulation is essential for versatile interaction with the environment [12-16]. For example, when writing neatly on tax forms or soldering circuits, we increase the stiffness of our hand movements for precise position control. When catching an egg or handling delicate objects, we would decrease the stiffness. Such stiffness information, along with other dynamic properties such as
intended inertia and damping, is not present in the direct decoding of trajectories. By decoding mechanical impedance (intended inertia, damping and stiffness) from the neural activity, impedance control [12] can be implemented on the robot so that the users intended impedance is displayed to the environment. This would eventually allow the neuro-prosthesis user to perform a much wider range of everyday tasks that involve interaction with objects.

There can be two different ways of approaching the problem of decoding impedance. The choice of strategy would depend on a deep-rooted argument in neuroscience of how motion is encoded in the cortex. Does the cortex only specify high level movement features and leave the conversion of the relatively abstract representation to muscle commands to occur downstream or does the motor cortex, along with other cortical areas, calculate the inverse dynamics and kinematics to send muscle activity information directly? Georgopoulous et al. [9] originally observed that M1 neurons were broadly tuned to hand movement direction, giving each neuron a preferred direction for which it exhibits maximal firing rate. This was in direct contradiction with Evarts [17] original observation that M1 firings are correlated to forces. Yet, subsequent work done by various groups showed good correlation between hand space movement and neural activity [10, 11, 18]. Thus, many have accepted the idea of population vectors as a good working model of the motor cortex, suggesting the motor cortex controls the higher level features such hand movement, rather than lower level details related to joints and muscles. Yet there also exists some contrasting evidence. Kalaska et al. [19], and Sergio and Kalaska [20] observed that the same cells that show directional tuning for movement also encode for force on objects during movement in an isometric tasks. Also, several studies have found systematic differences in neural activity depending on joint configuration for identical movement directions [21-23]. These studies suggest that the M1 does not purely encode hand space motion as suggested. In recent publications, Todorov [24, 25] argued that the motor cortex activity is most highly correlated to muscle activity. His hypothesis was based on studies [22, 26-28] that show correlation between muscle activation (magnitude and onset time) and M1 firings. In support of his argument, he linearly related motor cortex activity to muscle activities, and used the activities to drive a simplified musculo-skeletal model that was able to reproduce a lot of the observations made by Georgopulous et al.[9] and Schwartz [10] among others.

Which model better describes the function of the motor cortex is still not certain. The most likely answer is that it is a mixture of both. In fact, studies done by Kakei et al. [23] and Li et al. [29] show that large percentages of cortical neurons correlate well to muscle activities and equally large percentages correlate well to direction. Returning to the question at hand, it is likely that impedance can also be decoded using either of the two models. It is plausible that neural activity in the cortex can correlate to higher-order impedance information such as the magnitude and direction of the end stiffness matrix much like it has correlated to higher-order movement features such as hand position. It is obvious that we can make conscious decisions about desired stiffness and so impedance information should be found in the cortical areas. In this case, impedance can be measured during a primate’s movement and be fitted directly to the neural activity as has been done with position and velocity in previous BMIs. The other approach would be to
assume that the cortex contains an inverse dynamics and kinematics model, and find the neurons that are highly correlated to muscle activity. The muscle activities would already contain the intended impedance information because the muscles act like damped springs driving the limb inertias. In this case, the endpoint impedance can be extracted by running the muscle activations through a computational musculoskeletal model.

Direct extraction of impedance parameters through linear filters, artificial neural networks and other extraction algorithms, may not be a practical option for use in BMI. The traditional direct method requires the measurement of the impedance parameters for the training of the algorithms. Impedance parameters could be estimated by applying small force perturbations under static conditions [30] and during movement [31]. However, impedance values vary with arm posture, the level of co-contraction and direction of perturbation. Therefore, the measurements need to be taken at all possible arm configurations, by perturbing in all direction over many trials and at various levels of co-contraction. This would be a very tedious process, and not an ideal procedure to perform before every BMI trial. Furthermore, training the monkey to apply different levels of co-contractions would be a difficult task in itself.

The much more sensible option is to predict impedances by establishing a musculoskeletal model. In this paradigm, a fitting algorithm would take the neural data as input, and output an intermediate variable. The intermediate variable would represent muscle activation that would be entered into the musculo-skeletal arm model, which would in turn output all parameters related to motion (both kinematic and dynamic). An added benefit would be that all the parameters would be properly inter-correlated and mutually consistent, whereas in previous BMIs all parameters were treated to be independent of each other. Given a training set that spans the entire “muscle activation space” of the test set, the algorithm could be trained on trajectory information alone, without the direct measurement of impedances.

The primary focus of this paper is in introducing a musculoskeletal model-based extraction algorithm for impedance control. To demonstrate a working algorithm, neural data from a behaving monkey is also collected and two tests are conducted. The first test checks if the extraction algorithm which includes a musculoskeletal arm model can predict movement parameters to a level of accuracy that is comparable to that shown in previous BMI studies. The second test investigates if the algorithm with the musculoskeletal model would be able to predict the outcome of dynamic interactions with environment, something that a traditional trajectory prediction algorithm could not do.

2. Methods

2.1 Musculoskeletal arm model
A realistic model of the monkey’s arm was essential for reliable extraction of motion parameters. The model was established by using biomechanical parameters and force generation models found in the literature.
2.2.1 Muscle geometry

The muscle geometry, namely the insertion points, origin points, and the line of actions of each muscle group, determines the two key transformations required for computing joint torques; the transformation from joint space to muscle length space and the transformation from joint space to moment arms. Muscle length and muscle velocity information is required for calculating the forces generated by the muscle, and the moment arms are required to calculate the resulting moments at each joint from the muscle forces.

For simplification purposes, the arm model was constrained to planar movement involving two degrees of freedom, one at the shoulder and the other at the elbow. The same six major muscle groups as used in Gribble et al.’s [32] planar arm model were used: single-joint flexors and extensor at the shoulder (pectoralis and deltoid) and elbow (biceps long head and triceps lateral head) and double-joint muscles spanning both joints (biceps short head and triceps long head). The origin and insertion points of each muscle group was taken from data measured by Wood et al. [33], where they used a 3-D digitizer to measure anatomical data of the human arm from a male cadaver. For the 2-D model described in this paper, Wood et al.’s 3-D data was projected on to the plane of the planar movement (Fig. 1). Subsequently, all dimensions were scaled linearly for the arm lengths of the monkey used in the experiments (upper arm = 4.33 inches, forearm = 7.87 inches (to center of the hand)).

***************
***Fig. 1
***************

The following abbreviations were used for the muscle groups in Fig. 1; pectoralis –PMJ, biceps short head – BSH, biceps long head – BLH, deltoids – DEL, triceps long head – TRIO and triceps lateral head – TRIA. Origin points for the pectoralis, deltoid, bicep short head and triceps long head were fixed to the main body and are given as x-y coordinates in the world coordinate frame. The origin point coordinates for the biceps long head and triceps lateral head, as well as the insertion points for the pectoralis and deltoid were fixed to the upper-arm and are given as radii and angles from upper limb at the shoulder joint. The insertion point coordinates for the biceps long head, biceps short head, triceps long head and the triceps lateral head are given as radii and angle from lower limb at the elbow joint. The dimensions are given in Table 1.

***************
***Table 1
***************

The three extensors (DEL, TRIA, TRIO) are wrapped around the bone at both the shoulder and the elbow. This is modeled as a pulley of radius 0.143 inches at the shoulder and 0.0711 inches at the elbow. The instantaneous moment arms of the extensors remain constant and correspond to the radius of the pulley. The moment arm of the flexors (PMJ,
BLH, BSH) could be calculated from muscle kinematics with the line of action of force approximated as straight lines from the origin to the insertion points. The length and its rate of change of the muscles could also be calculated with the same assumptions used for the moment arms. Graham and Scott [34] have measured moment arms for the same planar movement in monkeys (Macaca mulatta) and have shown polynomial coefficients for moment arm regressions to joint angles. This regression could have been used instead of our geometrical model for calculating moment arms. However, without the muscle geometry, the muscle lengths and their rate change could not be determined.

2.2.2 Muscle Force Model
A Hill-type model [35] was used for muscle force generation, with the muscle forces divided into passive and active components. The two components were given analytical expressions by approximating the shape of the dimensionless force-length curves shown by Zajac [35]. The active force was assumed to be quadratic in length and linear in activity, with the maximum force, $F_{max}$, occurring at the optimal fascicle length, $l_o$ of each muscle group. The muscle activation, $\alpha$, scales the quadratic function. This is given by equation 1 and shown in Figure 2a.

$$F_a = \alpha F_{max} \left[1 - 4 \left( \frac{l - l_o}{l_o} \right)^2 \right]^+$$

where,
$$[u]^+ = \max[u,0]$$

The passive force component was given by the equation,
$$F_p = \left[ \left( \frac{F_{max}}{\exp(K_{sh}(l - l_o)) - 1} \right) \exp\left( \frac{K_{sh}(l - l_o)}{0.5l_o} \right) - 1 \right]^+$$

for $l \geq l_o$ and $F_p = 0$ otherwise. $K_{sh} = 3$ was used for the exponential and determines the shape of the curve. The resulting passive force-length curve is shown in Figure 2b.

The optimal fascicle length, $l_o$, for the six muscle groups were taken from those measured by Cheng and Scott [36] from Macaca mulatta forelimbs (pectoralis – 7.9 cm, biceps short head – 6.6 cm, biceps long head – 5.4 cm, deltoids – 2.7 cm, triceps long head – 3.8 cm and triceps lateral head – 4.3 cm). $F_{max}$ was assumed to be proportional to the physiological cross sectional area (PCSA) of the muscles [32, 36]. The PCSA for the muscles were found by the linear regression of Macaca mulatta PCSA against total body weight, provided by Cheng and Scott [36]. The scale factor from PCSA to $F_{max}$ was estimated by comparing the passive muscle stiffness at $l_o$ given by differentiating equation 2, and the passive stiffness previously measured empirically for each muscle group. Gribble et al. [32] gives the following values for passive stiffness; pectoralis – 258.5 N/m, biceps short head – 36.5 N/m, biceps long head – 190.9 N/m, deltoids – 258.5 N/m, triceps long head – 116.3 N/m and triceps lateral head – 209.9 N/m. The optimal value for the PCSA to $F_{max}$ scale factor was found by minimizing the error of $F_{max}$ from PCSA and the estimated $F_{max}$ from the stiffness values. The resulting $F_{max}$ values were; pectoralis – 12.8 N, biceps short head – 4.40 N, biceps long head – 7.46 N, deltoids – 7.92 N, triceps long head – 16.2 N and triceps lateral head – 12.8 N.
The visco-elasticity of the tendons is not modeled separately. However, for determining the actual muscle fiber length, a percentage of the total origin-to-insertion length had to be assumed to be length of the tendon. For simplicity, the percentage of muscle fiber length, \( l \), per total origin-to-insertion length is assumed to be constant at 60% for the biceps, 90% triceps and 95% for the pectoralis and deltoids. Muscle fiber length ratio was estimated from anatomical drawings.

The resultant force of each muscle is given by simply adding the active and passive components as shown in figure 2c.

The velocity dependence of the total muscle force is given by the following equation,

\[
 f_{\text{total}} = \left[ a_1 + a_2\tan(a_3 + a_4) \right] (f_a + f_p), \tag{3}
\]

where \( \dot{l} \) is the rate of change of the muscle length and with \( a_1 = 0.80, a_2 = 0.50, a_3 = 0.43 \) and \( a_4 = 58 \, \text{s/m} \). This has the same form as the sigmoidal function used in Gribble et al. [32], and is consistent with Zajac’s [35] force-velocity curves. The same ratios for muscle fiber length to tendon length apply.

The muscle recruitment and force development dynamics could be modeled with a second [32] or first order [35] low pass filter. However, this is omitted from the model for simplicity, and is left to be processed by the artificial neural network through the delayed time bins, as will be discussed later.

2.2.3 Dynamic Model of the Arm
The dynamics of the arm during planar movement was modeled as a two-link manipulator with frictionless rotational joints as seen in figure 1. The equation of motion is given by,

\[
 I\ddot{\theta} + C(\dot{\theta}, \theta)\dot{\theta} + \tau_{\text{external}} = \tau_{\text{muscles}}, \tag{4}
\]

where \( I \dot{\theta} \) are the inertial toques, \( C(\dot{\theta}, \theta)\dot{\theta} \) are the coriolis and centripetal toques, \( \tau_{\text{muscles}} \) are the torques generated by the muscles and \( \tau_{\text{external}} \) are the external torque applied by the environment. The inertial parameters were obtained from the linear regressions produced from monkey measurements in Cheng and Scott [36].

2.2 Prediction Algorithm
A linear filter or an ANN could be used in between the musculoskeletal arm model and the neural data as a fitting algorithm for extracting the motor parameters. The inputs to the musculoskeletal model are muscle activations and should not have negative values. Therefore a nonlinear function is required to generate the muscle activations and thus an ANN was chosen instead of the linear model (Fig. 3).
An adjusted back propagation algorithm was developed so that the errors in the output parameters could be propagated back through the arm model and adjust the weights of the ANN accordingly for gradient descent on the cost function. The simple solution to this problem was to treat the final layer of the ANN as a hidden layer and the arm model as the output layer. The algorithm for updating a weight in a hidden layer is given by,

$$\Delta w_{ij} = \eta \delta_j \cdot x_{j}^{l-1}$$

$$\Delta b_i = \eta \delta_i$$  

$$\delta_j^l = \frac{\partial E}{\partial x_j} f'(u_j^l),$$  

where $\eta$ is the learning rate, $x_{j}^{l-1}$ is the activity of the $j$th neuron in the $l$-1th layer, $E$ is the cost function defined by half the sum of square errors of the output,

$$E = \frac{1}{2} (y_k - d_k)^2,$$  

$f(u)$ is the sigmoidal function, 

$$f(u) = \frac{1}{1 + e^{-u}}$$

and

$$u_i^l = \sum_{j=1}^{n} w_{ij}^l x_{j}^{l-1} + b_i^l.$$  

$\delta_i$ for the final layer in the ANN in our combined algorithm is given by,

$$\delta_i^l = \frac{\partial E}{\partial x_i} f'(u_i^l)$$

$$= \sum_k \left( \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial x_i} \right) f'(u_i^l)$$

$$= \sum_k \left( (y_k - d_k) \frac{\partial y_k}{\partial x_i} \right) f'(u_i^l),$$  

where, $y_k$ is one of the $n$ selected outputs from the musculoskeletal arm model and $d_k$ is the corresponding “desired” value from the training set. The gradient $\frac{\partial y_k}{\partial x_i}$ (the partial differential of the $k$th output variable with respect to $i$th muscle activation) can be found analytically or numerically. In this paper, it is found numerically by applying a small change in muscle activation and observing the resulting change in output [37].

$$\frac{\partial y_k}{\partial x_i} \approx \frac{\delta y_k}{\delta x_i}.$$  

The algorithm would eventually converge to minimize the cost function and given that the training set spans all the muscle activation space present in the test set, the algorithm should predict motion parameters in the test set to a reasonable degree of accuracy. The elegant feature of this algorithm is that the muscle activations in the hidden layer do not need to be measured. If the movement in the training set sufficiently spans the whole muscle activation space, the hidden layer should, in theory, converge to the actual muscle activations with accuracy depending on the accuracy of the arm model.

In this study, a training set of 10 minutes of neural signals and movement data was used to train the network to predict the subsequent 1 minute of movement. The neural activity could be represented by binning the neural spikes in reasonable size bins (100 ms bins were used for this study). The input into the ANN would be the current bin of neural activities from all recorded cells, plus a train of prior bins to compensate for transmission delays, muscle recruitment dynamics and lingering intention signals (nine prior 100 ms bins were used for this study).

Any motion variable, downstream from muscle activations, could be predicted using this musculoskeletal model based approach. A measurable variable needs to be selected as the output so that the ANN can be optimized with respect to error in that variable. For this study, the torques generated by the muscles at the two joints (shoulder and angle) were selected as the output variables for optimization. The muscle torques were selected because they were sufficiently encompassing variables (resulting accelerations, velocities and positions can all be determined by the torques). The desired torques were calculated by inputting the measured angle, angle velocities and angle accelerations into the dynamics equation (eq. [4]). Velocities and accelerations were computed from the joint encoder values using finite-difference approximations.

2.3 Electrophysiology and behavioral tasks

An adult female monkey (Macaca mulatta) was implanted with four arrays containing 32 microwires each (with 1mm separation between each wire) in the motor cortex (M1 – 64 electrodes), sensory motor cortex (S1 – 32 electrodes) and the premotor cortex (PMd – 32 electrodes) of the left hemisphere. The monkey was trained to do center-out tasks to 12 targets (7-15cm), while seated in a KINARM exoskeleton (BKin Technologies, Ontario, Canada). The KINARM device constrains the monkey’s arm in planar motion while its motors are able to apply external torques at the two joints. The cursor and target was displayed on a reflective surface, in front of the monkey and above the monkey’s hand. The cursor was superimposed on top of the monkey’s hand and updated to follow the hand’s motion. Two force conditions were applied during the center out tasks, either a null force condition or a viscous force condition. The viscous force was proportional to the velocity in Cartesian space.
Over a thirteen session period, the activity of a total of 201-391 cells was recorded simultaneously in each session. For the musculoskeletal model, only the neuronal activity of 136-207 cells in the motor cortex implants were used. It was assumed that the M1 neurons are most likely to be highly correlated to muscle activation [24, 25] and thus be ideal for the musculoskeletal model based predictions.

3. Results

Neural data from the monkey performing the behavior task was used to test the developed algorithm. As proposed in the introduction, two main tests were conducted.

3.1 Prediction of Motion parameters

The first check was to see if neuronal activities were related to motion at all. This was done with a simple linear filter, which has been proven to be a fast an accurate way of predicting kinematic parameters for online BMI [1, 2, 5, 38]. In the most recent of these studies [5], a linear Wiener filter was used to give a correlation coefficient for predicting position of $r = 0.55-0.85$. Fig. 4(a) shows the correlation coefficient for hand position prediction for the monkey in this study, using the same linear filter as in Carmena et al. [5]. The results are comparable. It is reasonable to assume that the implants in the monkey are recording neural data that is sufficiently correlated to arm movements, and thus proceed with fitting of the musculoskeletal arm model.

The first hypothesis as stated in the introduction was that the predictions from the model-based algorithm can give equivalent correlation coefficients to that given by a linear filter. Fig. 4(b) shows the prediction correlation coefficients for the monkey’s muscle tourques (shoulder and elbow) across 13 sessions, comparing the musculoskeletal model/ANN hybrid with the linear Wiener filters. Not only are the predictions through the musculoskeletal model comparable to that of the linear filter, they are actually significantly better for most sessions. The correlation coefficients are insensitive to scale and offsets. Therefore, the signal to error ratio (SER), defined as power of desired signal divided by the power of the error signal, is also plotted in Fig. 4(c) and (d). The SER is again comparable or better for all sessions for both elbow and shoulder torque. Fig. 4(e) and (f) shows a sample test set with 1 min of predicted trajectory from the musculoskeletal model plotted on the same plot as the observed trajectory for shoulder and elbow torque respectively. The plot shows that the model is able to predict the torque profile well, even the during the high frequency movements in the later part of the sample.
3.2. Prediction during force field tasks

It was claimed that the main advantage of the proposed algorithm is that it can extract impedances, and therefore predict how the monkey’s hand would interact with the environment. Viscous force fields were applied during two sessions to see the effects on the prediction of the linear filter and the musculoskeletal model based algorithm. For the model based algorithm, the training set had to span all of the muscle activation space present in the test set. Therefore the algorithms were trained on viscous force conditions to predict a null force test set, rather than the other way around. The torque amplitudes for the training set were roughly 5 times larger than that of the test set (Fig. 5a).

With the traditional linear filter, the predicted torque amplitudes from the neural activity were significantly larger than that of the actual observed torques (Fig. 5b shows a sample of one of the two sessions). The correlation coefficients for the predicted shoulder torques for the two sessions were 0.037 and 0.002. Given a training set with viscous force, the linear filter was not able to accurately predict torque under a different force field condition. It appears that the optimized linear filter was not able to extract the correct correspondence between the neural activities and the muscle activities that determine the dynamics of the movements.

Fig. 5c shows the prediction of shoulder torque using the musculoskeletal based algorithm for the same session as in Fig. 5b, plotted on the same scale. Fig. 5d shows the same plot on an enlarged scale. The correlation coefficients for predicting the null force field torques with the musculoskeletal model based algorithm was 0.572 and 0.549 for the two sessions. This is significantly higher than prediction correlation coefficients achieved using the linear filter and the difference in accuracy is apparent when comparing the plots in Fig. 5. The algorithm successfully predicted the reduced torque amplitudes of the null force field condition from the neuronal activity from a training set containing only much higher amplitude torques resulting from a viscous force field. This suggests that the musculoskeletal model based algorithm was able to extract information about the co-contraction of the muscles during the viscous force field task and train itself at the muscle activation level. The linear filter was only able to establish direct correlations between the neuronal activity and torque and thus co-contraction information is not present, whereas the model based approach was able to converge to a solution that is consistent at the muscle activation level.

Fig. 6(a)-(c) provides further evidence for this. Gold disc electrodes (Grass Instrument Co., West Warwick, RI) were used to measure surface EMGs at the biceps (BLH) and triceps (TRIA) of the monkey during a session with the viscous force field task. The EMG signals were amplified, high-pass filtered, rectified, smoothed (kernel convolution) and then normalized. The processed EMG signals are plotted on Fig. 6 along with the intermediate variable from the hidden layer of converged musculoskeletal model based
algorithm. It was previously postulated that given a full training set, the hidden layer would converge to muscle activation values through the back-propagation of error. The activation times and amplitude of the measured EMGs and the muscle activation variables resulting from the algorithm, shown in Fig. 6, are roughly in agreement. When comparing the two, one must consider the fact that surface EMGs are not direct measures of activation inputs to the muscle. The EMG signals are more likely a rough and noisy estimate of the force output of the muscles [39]. Given the uncertainty of the source of the EMG signals, the likeness of the two signals shown in the plots are rather remarkable. The plots provide strong evidence that the hidden layer has in fact converged to muscle activations, as desired. Also, comparing the triceps and biceps activations, it appears that the appropriate co-contractions of the muscles are extracted by the algorithm.

IV. Discussion
The comparison of the prediction accuracy across multiple sessions shows that the musculoskeletal model performs just as well as the linear model for free movement. In fact, in most sessions, the musculoskeletal based algorithm predicted the muscle torques at the joints significantly better than the linear model. This may be because joint torques are lower level variables that may not often be represented directly in the cortical areas. Higher order kinematic variables such as position and velocity have been predicted with great success using linear filters, because a large number of neurons correlate directly to these variables. However, to the author’s knowledge, there has been no evidence to show that cortical neurons directly encode muscle torques at the joint and that is possibly what is being observed in the slightly lower correlation coefficients. On the other hand, there is abundant evidence that cortical neurons are directly linked to muscle activities [22, 26-28]. The musculoskeletal model based algorithm extracts parameters starting at the muscle activation level. Any motor parameter that is down stream from muscle activities, such as muscle torques, can be extracted successfully. A linear filter may also extract muscle activation successfully if fitted to measurements of the activation. However, even when given muscle activation related neuronal activity, the linear filter would not able to model the nonlinearities of muscle force generation and transformations to moment needed to predict muscle torques accurately.

By selecting muscle torques as the output variable for comparison, one of the main strength of the musculoskeletal model based algorithm has been highlighted. A linear filter or any other non-model based extraction algorithm is most effective only when the predicted motor parameters are directly and strongly represented in the cortical areas. However, using the musculoskeletal model based algorithm, any motor parameter that is downstream from the muscle activities in the musculoskeletal system can be predicted. This includes torques, forces, impedances, positions and velocities among others. Furthermore, the parameters are extracted so that they are all mutually consistent, where as non-model based algorithms can only extract the parameters independently which results in an inconsistent set of predicted outputs.
The new algorithm has also demonstrated the ability to predict interactions with force fields. More notably, it was able to predict the resulting interaction in a new force field condition that was not present in the training set. Again this can be attributed to extraction at the muscle activation level which contains co-contraction information and also the modeling of the force-displacement and force-velocity properties of the muscles. The ability of this algorithm to generalize for different interactions will become important for BMI applications in the future when prosthetics would have to interact with objects. The BMI can not be trained for all possible interactions. No current extraction algorithm, be it a linear filter, an artificial neural network, population vectors, or a recurrent neural network, is able to generalize for all possible interactions given a limited training set. However, the only requirement on the training set for the musculoskeletal model based algorithms to be generalized, is that it covers the entire muscle activation space. This is a relatively lax requirement and is easily achievable. Training and prediction at the muscle activation level results in a comprehensive algorithm that is not restricted by the environmental forces in the training set and is able to predict a wide range of interactions during the test set.

The ability of the algorithm to generalize for all possible interactions can also be stated as its ability to predict impedances. The effective stiffness, damping and inertia information is always present in the model and could be displayed to the environment via the prosthesis to achieve the intended interaction. The algorithm is further strengthened by the fact that direct measurements of impedances, which are difficult to acquire, is not required. Although not included in this study, a behavioral task with divergent force fields [14] where the monkey would have had to control its arm stiffness to complete task, would have shown the advantages of predicting impedances more conspicuously.

It has to be pointed out that the musculoskeletal model is not entirely accurate. Vast simplifications have been made. Also, monkey data was not available for several parameters and human data was used instead. In fact, a large portion of the error in predictions may be attributed to inaccuracies in the model. Continued effort can be made to establish a more accurate model. However, it is also hoped that in an online system, the plasticity in the cortex would be able to adjust to the inaccuracies of the model. In other words, it is hoped that the user would be able to learn to control an arm with different dynamics. It is apparent in our everyday activities that the human brain is very adept at learning new dynamics, so it is not overly optimistic to predict that the user would be able to learn the dynamics of the musculoskeletal arm model.

Some of the neurons maintained high correlation coefficients for predicting position with a linear filter under different force conditions. These neurons are most likely the “kinematic neurons” that encode higher order parameters of position and velocity. In fact, when investigating their directional tuning, these were the cells that did not vary in tuning properties under a different force field. There were also neurons that varied is preferred direction significantly. These corresponded to the neurons that could not predict motion under different force field conditions through a linear filter. These were most likely the “dynamic neurons”. By locating and recording from kinematic neurons alone, a BMI could be operated under pure motion control for all force field conditions. However,
these neurons would not contain impedance information and thus versatile interactions with the environment would not be possible.

A detailed analysis of the neuronal activity, including investigating the change in tuning properties of the neurons during force field tasks and discriminating kinematic and dynamic neurons has been left for another study. Also, an online demonstration of BMI with the musculoskeletal model is left for the subsequent study. The main goal of this paper is in introducing a new approach to extracting continuous motion parameters for BMI. The musculoskeletal model based algorithm was built from empirical measurements and muscle models found in the literature and tested with behavioral tasks on a monkey with cortical implants. It has passed the two main tests that the authors felt were necessary for the algorithm to be considered significant. The tests showed that the algorithm can predict motor parameters to the same level of accuracy as a linear filter and that it can further predict interactions with the environment, which previous extraction algorithms could not do. Furthermore, the algorithm could extract a complete set of motor variables that were consistently inter-correlated within the dynamics of the system.

References


Figures
- Muscle origins mounted to main body (Fixed)
- Muscle origins/insertions mounted to upper-arm (Moving)
- Muscle insertions mounted to lower-arm (Moving)
Fig. 2
Fig. 3
Fig. 4
(a) 

(b) 

(c)
<table>
<thead>
<tr>
<th>Muscle Group</th>
<th>Origin</th>
<th>Insertion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$a_i$</td>
<td>$b_i$</td>
</tr>
<tr>
<td>1PMJ</td>
<td>-1.877 in.</td>
<td>1.093 in.</td>
</tr>
<tr>
<td>2BSH</td>
<td>-0.3156 in.</td>
<td>0.4972 in.</td>
</tr>
<tr>
<td>3BLH</td>
<td>0.3519 in.</td>
<td>126.1°</td>
</tr>
<tr>
<td>4DEL</td>
<td>-0.2008 in.</td>
<td>-0.2374 in.</td>
</tr>
<tr>
<td>5TRIO</td>
<td>0.2296 in.</td>
<td>-0.3645 in.</td>
</tr>
<tr>
<td>6TRIA</td>
<td>1.953 in.</td>
<td>8.69°</td>
</tr>
</tbody>
</table>

Table 1
**Figure Legends**

Fig. 1. Muscle geometry for the musculoskeletal arm model. The following abbreviations are used for the muscle group; pectoralis – PMJ, biceps short head – BSH, biceps long head – BLH, deltoids – DEL, triceps long head – TRIO and triceps lateral head – TRIA. (a)-(d) mark the coordinates for the origin and insertion points for each muscle group. The coordinate values are shown in Table 1. A is an enlarged view of the shoulder joint and B is an enlarged view of the elbow joint.

Fig. 2. Block diagram for muscle force generation model. The total force is the sum of the passive and active muscle forces.

Fig. 3. Block diagram of prediction algorithm. The algorithm takes neuronal activity input and the ANN outputs muscle activation as the intermediate variable. The muscle activations drive the computational musculoskeletal model.

Fig. 4. (a) Correlation coefficient for Cartesian position prediction across 12 sessions from a linear Wiener filter. (b) Correlation coefficient for predicting muscle torques at the elbow and shoulder. The plot compares the linear Wiener filter with the musculoskeletal model based algorithm. (c) SER (signal to error ratio) plotted for muscle torques at the shoulder. (d) SER (signal to error ratio) plotted for muscle torques at the elbow. (e) 1 min of sample prediction trajectory for muscle torques at the shoulder. The predictions are from the musculoskeletal model based algorithm. (f) 1 min of sample prediction trajectory for muscle torques at the elbow. The predictions are from the musculoskeletal model based algorithm.

Fig. 5. (a) Observed muscle torques at the shoulder for a training (viscous force field) and test set (null force field). (b) Predicted trajectory from a linear filter for torques at the shoulder. (c) Predicted trajectory from the musculoskeletal model based algorithm for torques at the shoulder. (d) same as (c) on an enlarged scale.

Fig. 6. (a)-(c) 3 × 100 second samples of normalized surface EMG measurements plotted with the intermediate variable from the musculoskeletal model based algorithm. Each figure contains two plots, one from the biceps and the other from the triceps.

Table 1. Dimensions for the musculoskeletal arm model muscle geometry shown in Fig. 1.