Published in final edited form as:

IEEE Trans Neural Syst Rehabil Eng. 2016 December ; 24(12): 1405–1415. doi:10.1109/TNSRE.

2016.2535348

Semiparametric Identification of Human Arm Dynamics for Flexible Control of a Functional Electrical Stimulation Neuroprosthesis

Eric M. Schearer [Member, IEEE],

Department of Mechanical Engineering, Cleveland State University, Cleveland, OH, USA

Yu-Wei Liao [Student Member, IEEE],

Department of Mechanical Engineering, Northwestern University, Evanston, IL USA

Eric J. Perreault [Member, IEEE],

Department of Biomedical Engineering and Department of Physical Medicine and Rehabilitation, Northwestern University, Evanston, IL USA

Matthew C. Tresch [Member, IEEE],

Department of Biomedical Engineering and Department of Physical Medicine and Rehabilitation, Northwestern University, Evanston, IL USA

William D. Memberg,

Department of Biomedical Engineering, Case Western Reserve University, Cleveland, OH USA

Robert F. Kirsch [Member, IEEE], and

Department of Biomedical Engineering, Case Western Reserve University, Louis Stokes Veterans Affairs Medical Center, and the Department of Orthopedics, MetroHealth Medical Center Cleveland, OH USA

Kevin M. Lynch [Fellow, IEEE]

Department of Mechanical Engineering and the Northwestern Institute on Complex Systems, Northwestern University, Evanston, IL USA

Abstract

We present a method to identify the dynamics of a human arm controlled by an implanted functional electrical stimulation neuroprosthesis. The method uses Gaussian process regression to predict shoulder and elbow torques given the shoulder and elbow joint positions and velocities and the electrical stimulation inputs to muscles. We compare the accuracy of torque predictions of nonparametric, semiparametric, and parametric model types. The most accurate of the three model types is a semiparametric Gaussian process model that combines the flexibility of a black box function approximator with the generalization power of a parameterized model. The semiparametric model predicted torques during stimulation of multiple muscles with errors less than 20% of the total muscle torque and passive torque needed to drive the arm. The identified model allows us to define an arbitrary reaching trajectory and approximately determine the muscle stimulations required to drive the arm along that trajectory.

I. Introduction

Functional electrical stimulation (FES) is a promising technology for restoring movement control, including reaching, to people with paralysis. FES has had success in some applications [1–3], particularly those that can be accomplished with stimulation of relatively few muscles or larger groups of muscles working synergistically. Though there is a need for restoring more complex movements involving multiple muscles and joints, few general purpose FES controllers have been proposed for such tasks. A major impediment to developing such a controller is the lack of subject-specific models that describe the relationship between muscle stimulation and the corresponding actions of the neuromuscular system. The availability of such models permits the use of model-based strategies for feedforward and feedback control.

In this paper we present a method for building data-driven models of the arm controlled by FES for any specific subject. The models map muscle activations to joint torques and joint torques to arm motions. Use of the models will allow us to specify an arbitrary desired motion of the arm and use a model-based controller to stimulate muscles to achieve that motion.

Although there is a long history of musculoskeletal modeling, few complex models have been used for FES control in humans. A few notable works include those that have modified traditional parametric biomechanical models to represent the impairments present after spinal cord injury [4, 5]. Estimating and validating the parameters for these models is a daunting task, especially for the inverse models often needed for control. An alternative approach is to use artificial neural networks to identify black-box, multiple-input multiple-output mappings that replicate the essential behaviors of a paralyzed limb [6, 7]. These have been demonstrated to work well in simulation, but generally have been identified from modeled rather than actual limbs, and it is unclear how well the modeled properties of muscles extend to electrical stimulation [8].

To combine the strengths of both the parametric biomechanical models and the black-box approach, we have developed a semiparametric approach to modeling the input-output properties of an FES limb. Our approach uses semiparametric Gaussian process regression, which combines the ability of a nonlinear function approximator to capture the richness of the musculoskeletal dynamics with the generalization power of a parameterized model that includes prior knowledge of arm dynamics. Gaussian process regression has been useful in identifying the dynamics of robot arms [9], with better accuracy achieved by encoding prior knowledge of the robot dynamics in a semiparametric Gaussian process model [10].

This paper makes two primary contributions. First, we present an experimental method to identify the dynamics of a human arm by moving a person's hand with a robot while measuring the human arm's joint kinematics along with the interaction force at the interface of the hand and the robot's end effector. Second we evaluate the relative usefulness of the semiparametric identification method by comparing the generalization errors in predicting shoulder and elbow joint torques for parametric, semiparametric Gaussian process, and purely nonparametric Gaussian process model types.

II. Methods

This section presents a method to identify the dynamics of any particular human arm controlled by functional electrical stimulation and details the parametric, nonparametric, and semiparametric models used to represent the FES-controlled arm. We then describe experiments and analyses used to compare the generalization errors of the three model types in predicting arm dynamics and single-muscle torque production. Finally we describe experiments and analyses used to evaluate the utility of the semiparametric model type – which we found to be the most accurate model type – in predicting torques from multiple muscles used to control arm motion.

A. Experiment Setup and Data Gathering

We identified the dynamics of the arm of a single human subject with spinal cord injury and an implanted neuroprosthesis. The neuroprosthesis is capable of activating muscles in the shoulder and arm via FES. Our arm dynamics models map shoulder and elbow joint kinematics to shoulder and elbow joint torques in four degrees of freedom. The four degrees of freedom are the shoulder elevation plane, shoulder elevation, shoulder rotation, and elbow flexion as defined in [11]. Our model does not consider elbow pronation as it does not affect the position of the wrist, which is ultimately what we are trying to control.

The dynamics models have two components (Fig. 1). The first is the arm dynamics, which maps torques applied by the muscles and other external torques to the joint positions $\mathbf{q} \in \mathbb{R}^4$ and velocities $\dot{\mathbf{q}} \in \mathbb{R}^4$. The vector of joint positions and velocities is $\mathbf{x} = [\mathbf{q}^T \ \dot{\mathbf{q}}^T]^T$. Our control strategy requires determining the torques necessary to produce a given motion, so we identified the inverse arm dynamics directly rather than identifying and inverting the forward arm dynamics. The second dynamics model component is muscle torque production, which maps muscle activations $\mathbf{a} \in \mathbb{R}^9$ to the active torques the muscles apply about the shoulder and elbow. This mapping depends on joint positions and velocities.

To gather kinematic and torque data for identifying the models, a HapticMaster (Moog FCS) robot moved the subject's arm along smooth reaching trajectories by applying force, which the robot measured, at the subject's distal forearm (Fig. 2). The robot has a three-axis force sensor at its end effector. The subject's hand and forearm were placed in a soft cast, and the cast was attached by a magnet to a ball-and-socket joint on a tie-rod end. The tie-rod end was bolted to the robot's end effector as seen in Fig. 2(c). The magnet acted as a safety mechanism, releasing the subject's hand if the robot applied more force than the magnet could hold. A mobile arm support (Fig. 2) applied force at the subject's forearm to compensate for gravity.

We computed the shoulder and elbow joint torques $\boldsymbol{\tau}_r$ equivalent to the forces the robot applied. The subscript on $\boldsymbol{\tau}_r$ stands for “robot”, as these are torques from the robot. If the robot were not available to move the arm, these torques must be supplied by stimulation of the subject's muscles or some other means in order to produce the same arm motion. The force $\mathbf{f}_m \in \mathbb{R}^3$ was measured at the robot's end effector and was transmitted through the ball-and-socket joint. The ball and socket joint by nature does not transmit moments. The force at the ball-and-socket joint was transformed to a wrench $\mathbf{f}_r \in \mathbb{R}^6$ at the subject's wrist. The

moment is the cross-product of the vector $\mathbf{r}_{wb} \in \mathbb{R}^3$ from the wrist to the ball-and-socket joint and the force at the ball-and-socket joint $\mathbf{f}_m$, so $\mathbf{f}_r = [\mathbf{f}_m \mathbf{r}_{wb} \times \mathbf{f}_m]^\top$. The corresponding torques about the shoulder and elbow joints are $\tau_r = J_r^\top(\mathbf{q})\mathbf{f}_r \in \mathbb{R}^4$ where $J_r \in \mathbb{R}^{6 \times 4}$ is the kinematic Jacobian of the subject's arm at the wrist. We computed the Jacobian analytically given a kinematic model of the arm.

The joint positions and velocities were computed using data collected by an Optotrak Certus Motion Capture System (Northern Digital, Inc.) and the robot. The motion capture system recorded the global position and orientation of rigid bodies with reflective markers attached to the subject's chair, upper arm, and forearm (Fig. 2). The robot reported the position of its end effector based on motor encoder data. From this sensor data we estimated the arm joint positions and velocities offline using an extended Rauch-Tung-Striebel (RTS) smoother, which is essentially an extended Kalman filter run forward and backward. We implemented the RTS smoother by adding smoothing to the extended Kalman filter method and MATLAB® code from [12].

We identified the inverse arm dynamics by moving the arm with zero muscle stimulation (Fig. 1(a)) so that $\mathbf{a}$ is a vector of zeros, and

$$\tau_r = \mathbf{p}(\mathbf{x}), \quad (1)$$

where $\mathbf{p}(\mathbf{x}) \in \mathbb{R}^4$ is the torque required to drive the arm including the mobile arm support along a trajectory.

To identify muscle torque production the robot moved the arm while stimulating a single muscle group with maximum pulse width (Fig. 1(b)) so that $\mathbf{a}$ was a vector with all zeros except for one muscle group with an activation of one. When stimulating muscles the robot torque is equal to the difference of the robot torques when stimulating no muscles and the torque contribution of the muscles

$$\tau_r = \mathbf{p}(\mathbf{x}) - R(\mathbf{x})\mathbf{a} \quad (2)$$

where $R(\mathbf{x}) \in \mathbb{R}^{4 \times 9}$ is the linear mapping from the vector of muscle activations to the resulting shoulder and elbow torques. Each column of $R(\mathbf{x})$ represents the amount of torque produced by each of the muscle groups at 100% activation. The individual muscle torques add linearly if there is no current spillover to adjacent muscles and there are no interactions of connective tissue between muscles. This linearity assumption is approximately true for the human subject who participated in these experiments [13]. We directly identified $\mathbf{p}(\mathbf{x}) - R(\mathbf{x})\mathbf{a}$ and then subtracted this total torque from the previously identified inverse arm dynamics torques to compute the torque production for each muscle. We did this experiment for each muscle group that can be stimulated for our subject with SCI.

B. Arm Dynamics Models

Using the data gathered as described in Subsection II-A we identified models for the inverse arm dynamics and for muscle torque production. The inputs to the models are the joint positions and velocities $\mathbf{x} = [\mathbf{q}^\top, \dot{\mathbf{q}}^\top]^\top$. The model outputs are the shoulder and elbow joint torques $\mathbf{t}(\mathbf{x})$ equivalent to the forces that the robot applies to move the arm. These torques are either those when no muscles are stimulated $\mathbf{p}(\mathbf{x})$ or those when a single muscle is stimulated $\mathbf{p}(\mathbf{x}) - R(\mathbf{x})\mathbf{a}$. Activation is not an explicit model input as it was held constant during identification trials, and we assume torque scales linearly with activation. In this subsection we describe parametric, nonparametric, and semiparametric model types to represent the mapping from joint kinematics to torques.

1) Parametric Model—A model that is linear in its parameters is preferred because we can use linear regression to identify the model and avoid solving a nonlinear optimization problem. We used a simple parametric model for the torque about a single degree of freedom $t_p(\mathbf{x})$, where the subscript p stands for parametric, is dominated by joint stiffness and damping as in

$$t_p(\mathbf{x}) = \mathbf{h}(\mathbf{x})^\top \boldsymbol{\beta} = k_s(\theta - \theta_0) + k_d\dot{\theta}, \quad (3)$$

where θ and $\dot{\theta}$ are the position and velocity of the degree of freedom, θ_0 is the equilibrium position of the degree of freedom, and k_s and k_d are the stiffness and damping. Here $\boldsymbol{\beta} = [-k_s\theta_0 \ k_s \ k_d]^\top$ is a column vector of parameters to be identified, and $\mathbf{h}(\mathbf{x}) = [1 \ \theta \ \dot{\theta}]^\top$ is a column vector of measured variables. The equilibrium position is the position where the degree of freedom rests when no external forces are applied, defined by the muscle group being activated, the properties of passive tissues, and the mobile arm support.

We used linear regression to estimate the vector of parameters $\boldsymbol{\beta}$ and predict the torque given a new test input $\mathbf{x}_*$. With data from smooth reaching trajectories we constructed the matrix $H \in \mathbb{R}^{n \times 3}$ where each row is $\mathbf{h}(\mathbf{x})^\top$ evaluated at each training input, and $\mathbf{y}$ is a column vector of n training outputs. The torque prediction $t_p(\mathbf{x}_*)$ of the parametric model and its variance $v_p(\mathbf{x}_*)$ given a test input $\mathbf{x}_*$ are

$$t_p(\mathbf{x}_*) = \mathbf{h}(\mathbf{x}_*)^\top \hat{\boldsymbol{\beta}} = \mathbf{h}(\mathbf{x}_*)^\top (H^\top H)^{-1} H^\top \mathbf{y} \quad (4)$$

$$v_p(\mathbf{x}_*) = \sigma_n^2 (1 + \mathbf{h}(\mathbf{x}_*)^\top (H^\top H)^{-1} \mathbf{h}(\mathbf{x}_*)), \quad (5)$$

where σ_n^2 is the variance in the torque output, and $\hat{\boldsymbol{\beta}}$ is the estimate of the parameters.

Alternatively one might include a typical parameterization of inertial terms, Coriolis and centrifugal terms, and gravity along with polynomial functions of the joint positions and velocities for passive muscle forces as in [14]. However, predictions of torque were

inaccurate when using this full parametric model. The transition from the full parametric model with coupled degrees of freedom to the simpler single-joint model in (3) was based on a number of observations. First, because the reaching motions for this study were generally slow, the inertial, Coriolis, and centrifugal terms were small compared to the gravity terms along with the passive stiffness and damping of the muscles and arm support. Second, the mobile arm support essentially cancels out gravity and still provides stiffness. This leaves only stiffness and damping as captured in (3). Due to inertial terms not playing a significant role in torque predictions we did not include accelerations in the input vector for our models $\mathbf{x} = [\mathbf{q}^T, \dot{\mathbf{q}}^T]^T$.

2) Nonparametric Model—The second of the three models is a purely nonparametric Gaussian process model [9]. A Gaussian process can be viewed as a distribution of random functions defined by a mean function and covariance function. Gaussian process regression (GPR) amounts to assuming a prior distribution of random functions, collecting training samples, and then choosing those functions from the prior distribution that best agree with the training samples. The remaining functions make up the posterior distribution which is also a Gaussian process.

GPR suffers much less from overfitting than do parametric physics-based models and completely avoids the need to collect rich enough data to guarantee identifiability of parameters; this is because GPR determines model complexity systematically based on the available data by maximizing the marginal likelihood or evidence of the training data [9]. This balances training error and model complexity.

Our nonparametric model for predicting arm joint torques has a Gaussian process prior distribution with zero mean and a squared-exponential covariance function $k(\mathbf{x}, \mathbf{x}')$,

$$t_n(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}')). \quad (6)$$

Given a test input $\mathbf{x}_*$, which is the joint positions and velocities, the joint torque prediction $t_n(\mathbf{x}_*)$ of the nonparametric model and its variance $v_n(\mathbf{x}_*)$ are the mean and variance of the posterior Gaussian process evaluated at the test input

$$t_n(\mathbf{x}_*) = \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{y} \quad (7)$$

$$v_n(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (K + \sigma_n^2 I)^{-1} \mathbf{k}_* \quad (8)$$

where $\mathbf{y}$ is a column vector of n training outputs, $\mathbf{k}_* \in \mathbb{R}^n$ is a vector where the i^{th} element is the covariance function evaluated at the test input and the i^{th} training input, and $K \in \mathbb{R}^{n \times n}$ is a matrix of the covariance function evaluated at all combinations of training inputs.

3) Semiparametric Model—The last of the three models is a semiparametric Gaussian process model which incorporates knowledge of the system encoded in a parametric model. The semiparametric model uses its locally relevant Gaussian process component to predict complex dynamics near training data and its globally relevant parametric part to generalize to places in the input space that are far from the training data.

The semiparametric model is a Gaussian process where the mean function is the parametric model in (3) and the covariance function is the sum of a squared-exponential covariance function and a term that takes into account the uncertainty in the parameters of the parametric model,

$$t_s(\mathbf{x}) \sim \mathcal{GP}(\mathbf{h}(\mathbf{x})^\top \mathbf{b}, k(\mathbf{x}, \mathbf{x}') + \mathbf{h}(\mathbf{x})^\top B \mathbf{h}(\mathbf{x})). \quad (9)$$

This is the semiparametric Gaussian process form proposed in [15]. It makes the prior assumption that the parameters $\boldsymbol{\beta}$ of the linear model are distributed normally with mean $\mathbf{b} \in \mathbb{R}^k$ and covariance $B \in \mathbb{R}^{k \times k}$. We used the mean and covariance of the parameters found from fitting the parametric model to a data set to define the prior parameter distribution for the semiparametric model.

The vertical scale hyperparameter of the square-exponential covariance function $k(\mathbf{x}, \mathbf{x}')$ was found by maximizing the marginal likelihood or evidence of the training data, while the length scale parameters were chosen a priori to be small. This allowed the nonparametric part of the model to dominate in making predictions for test inputs close to the training inputs and the parametric part of the model to dominate for test inputs that are not close to the training inputs.

Given a test input $\mathbf{x}_*$, which is the joint positions and velocities, the joint torque prediction $t_s(\mathbf{x}_*)$ of the semiparametric model and its variance $v_s(\mathbf{x}_*)$ are the mean and variance of the posterior Gaussian process evaluated at the test input

$$t_s(\mathbf{x}_*) = \mathbf{h}(\mathbf{x}_*)^\top \bar{\boldsymbol{\beta}} + \mathbf{k}_*^\top K_y^{-1} (\mathbf{y} - H^\top \bar{\boldsymbol{\beta}}) \quad (10)$$

$$v_s(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^\top K_y^{-1} \mathbf{k}_* + (\mathbf{h}(\mathbf{x}_*) - H K_y^{-1} \mathbf{k}_*)^\top (B^{-1} + H K_y^{-1} H^\top)^{-1} (\mathbf{h}(\mathbf{x}_*) - H K_y^{-1} \mathbf{k}_*) \quad (11)$$

where $K_y = K + \sigma_n^2 I$. The posterior estimate of the parameters of the linear model is $\bar{\boldsymbol{\beta}} = (B^{-1} + H K_y^{-1} H^\top)^{-1} (H K_y^{-1} \mathbf{y} + B^{-1} \mathbf{b})$. The first term on the right-hand side of (10) is the predicted output of the parameterized linear model. The second term is the prediction of the residuals of the parameterized linear model, and its magnitude depends on how close the

test input is to the training inputs. Note that the terms K_y and $\mathbf{k}_*$ for the semiparametric model will be different in general than those terms for the nonparametric model.

All Gaussian process computations were made using the GPML toolbox for MATLAB® which is based on [9].

C. Experiments to Identify and Evaluate the Models

This subsection describes specifically how we identified the models on a human subject, how we evaluated the models' single-muscle torque predictions on data used in the identification process, and how we evaluated the utility of the semiparametric model type in predicting torques from multiple muscles used to control arm motion.

1) Subject—A 57-year-old woman who sustained a hemisection of the spinal cord at the C1–C2 level from a gunshot wound in 1994 participated in this study. She cannot move her right arm, but she has some sensation and pain hypersensitivity. She experiences hypertonia in some of her arm muscles. Due to hypertonia the range of motion of her right arm is limited. More details on this subject are included in [16] (Subject 1). Protocols were approved by the institutional review boards at Northwestern University (IRB NO. STU00018382) and MetroHealth Medical Center (IRB NO. 04-00014).

The subject has an implantable stimulator-telemeter [17–19] located in her abdomen for stimulating muscles in her right arm and shoulder complex. The device has leads that carry current to stimulation electrodes. Some of the leads are attached to intramuscular electrodes [20], which are surgically implanted in muscles. Other leads are attached to nerve cuff electrodes [21], each of which is wrapped around a nerve that activates one or more muscles. We refer to each muscle or group of muscles stimulated by a single electrode as a muscle group. In this experiment we used the nine muscle groups shown in Table I.

Power and control signals are sent to the implanted unit through the skin via an inductive radio-frequency link [17–19]. Stimulation to each muscle group uses bi-phasic, charge-balanced pulses delivered at 13 Hz. This frequency was selected to produce reasonably fused contractions while minimizing muscle fatigue. The stimulation amplitude was fixed at a different level for each muscle group (see Table I), while pulse duration could be varied between 0–200 μ s, providing a means to control the force generated by each muscle group. The maximum pulse width for each muscle group was selected to prevent the subject from feeling pain, to prevent spillover to activate other muscles, and when no further force could be achieved with a larger pulse width.

The vector of stimulation pulse widths for the muscle groups is the control signal. We refer to the pulse width as a stimulation input. We run real-time control code on a computer that sends stimulation input commands to the implanted unit.

2) Model Identification Protocols—During two identification sessions, which occurred 19 days apart, we gathered data to identify the models and compare their accuracy in predicting torques for stimulation of single muscle groups. During each session several trials were run where the robot moved the subject's hand in a straight line from a target position to

a home position over three seconds, paused for one second, and then moved the subject's hand back to the target position over the next three seconds. The position along the straight line was defined using a fifth-order polynomial in time. The polynomial's six parameters allow for defining six constraints: initial and final positions as well as zero initial and final velocities and accelerations. The polynomial is a smooth trajectory of positions, velocities, and accelerations.

The two home positions (Fig. 2) were in front of the subject's mouth and in front of and just below the subject's shoulder. The four target positions (Fig. 2) were the furthest the subject's hand could be comfortably moved high and to the right, low and to the right, high and to the left, and low and to the left. These four targets did not touch the absolute limit of the subject's workspace for which the elbow is fully extended. The distance from each target position to each home position varied from 18 cm to 38 cm, and the time to reach from target to home was a constant three seconds, so the maximum speed of the hand varied from 11 cm/s to 24 cm/s.

During each trial we either stimulated no muscles in order to identify the inverse arm dynamics or stimulated one muscle group to identify the muscle torque production for that muscle group. To eliminate the effect of activation dynamics, we began stimulation one second before the hand motion began and continued until one second after motion ended. This ensured a constant maximum activation state during the reaching trial. We executed three sets of 80 trials during the first session and two sets during the second session. During each set trials were executed for every combination of the four targets, two home positions, and nine muscle groups plus one with no muscles stimulated ($4 \times 2 \times 10 = 80$ trials). The order of targets, muscles, and home positions was randomized, but executed in blocks of 20 trials for each target.

For each reaching trial we computed the difference between the models's torque predictions and the torques derived from the robot's force measurements. When a model makes predictions for a trajectory which was not used to train the model this difference is called the generalization error. We compared the distribution of errors for all trials to the distribution predicted by the models. We also computed the RMS error for each individual reaching trial.

To compare the ability of the parametric, nonparametric, and semiparametric models to predict torques for new inputs, we computed learning curves for each model type. The learning curve is the generalization error as a function of the amount of data used to train the model. There were eight trajectories during the identification sessions. For each model type models were identified using every combination of one, two, three, four, five, six, and seven of the trajectories. This amounts to leaving out seven, six, five, four, three, two, or one trajectory for cross-validation.

3) Control Utility Protocols—We evaluated the utility of the semiparametric model type in predicting torques from multiple muscles used to control arm motion during a single session which occurred two days after the second identification session. To compare errors in torque predictions during these multi-muscle trials to those of the single-muscle identification trials, the robot moved the subject's hand along the same eight reaching

trajectories as in the identification sessions. During each movement multiple muscle groups were stimulated according to a controller based on the semiparametric model trained on all data from the second identification session. The semiparametric model was used because it made the best predictions in identification trials (see III-A). The controller (Fig. 3) solves the following optimization problem:

$$\begin{aligned} & \underset{\alpha, \mathbf{f}}{\text{minimize:}} \quad \|\alpha\|_2^2 + c\|\mathbf{f}\|_2^2 \\ & \text{subject to:} \quad R(\mathbf{x}_*)\alpha + J_r(\mathbf{q})^\top \mathbf{f} = \mathbf{p}(\mathbf{x}_*) \quad \alpha_i \in [0, 1] \quad \forall i \in \{1, 2, \dots, 9\}. \end{aligned} \quad (12)$$

This optimization problem finds the minimum sum of squared muscle activations that can produce the joint torques required to move the arm along the trajectory. Given the activations, muscle group recruitment curves previously identified via the deconvolved ramp method [22] were inverted to determine the stimulation inputs to the muscle groups. When the muscles could not produce sufficient torque to achieve a desired position and velocity, the controller computed the additional compensating force $\mathbf{f} \in \mathbb{R}^3$ the robot must apply, where c is a large constant that penalizes using robot force if the muscle torques are sufficient. The robot force was then discarded, and instead the robot's feedback controller corrected any errors in trajectory following insuring that the subject's hand followed the desired trajectory. We solved for the muscle stimulations offline, but the control problem can be solved in real-time if feedback control is required.

Each of the eight reaching trajectories was run six times with two exceptions. We ran the T4 to B1 trajectory twice and T4 to B2 trajectory once (see Fig. 2) because the subject was uncomfortable with the stimulation pattern used for these trajectories. For each point along the eight trajectories we computed the error in the model's prediction of torques resulting from muscle stimulation according to the controller compared to the joint torques required to produce the same trajectory as the measured robot forces.

III. RESULTS

A. Single-Muscle Model Identification and Validation

The models we identified predict the joint torques and their confidence intervals. An example of a prediction of our models is shown in Fig. 4. A semiparametric model was trained on seven of the eight reaching trajectories, and the model predicted the torques for the remaining trajectory, which we call the test trajectory. The model was given the joint angles and joint velocities of the test trajectory to predict the torques. Fig. 4 shows the same nominal hand trajectory when not stimulating any muscles (a) and when the radial nerve, which targets the triceps muscles, was stimulated (b). Muscle stimulation resulted in slightly different trajectories for (a) and (b) in joint space. Stimulating the radial nerve, which activates triceps, increased the elbow flexion torque required to drive the arm.

The observed generalization errors of the semiparametric models match well the models' predicted distributions (Fig. 5). Generalization error is the error in predicting torques for trajectories not used to train the models. The joint torques from 91% of all reaching trials fell within the 95% confidence intervals of the semiparametric models' predictions. The observed error distribution is slightly non-Gaussian, with a higher frequency of small errors and slightly longer distribution tails than the Gaussian distribution predicted by the models.

The semiparametric models were generally more accurate and required less training data than the parametric and nonparametric models. When comparing the same muscle group, session, and number of training trajectories the semiparametric models had smaller generalization errors than the nonparametric model in 99% of cases with the difference significant ($p < 0.05$) in 48% of cases. The semiparametric models had smaller errors than the parametric models in 99% of cases with the difference significant ($p < 0.05$) in 32% of cases. The semiparametric model's error was less than the error of a nonparametric model trained with one additional trajectory in 75% of cases, with 13% of cases significant. The semiparametric model's error was less than the error of a nonparametric model trained with two additional trajectories in 31% of cases, with 1% of cases significant. The learning curves in Fig. 6 illustrate the more accurate torque predictions and the reduced data requirements of the semiparametric models for the case of stimulating the radial nerve.

The advantage of the semiparametric type over the nonparametric type was most apparent for small training sets. This can be seen for all muscle groups by comparing the green and red bars in Fig. 7 (a) where just one trajectory was used to train models. When the radial nerve was stimulated for example (Fig. 6), with training sets of one, two, three, or four trajectories the semiparametric models had significantly smaller mean errors than did the nonparametric models. With a small amount of training data the nonparametric models could not generalize well while the semiparametric models with their fixed basis functions made better predictions.

The advantage of the semiparametric type over the simple parametric model was most significant with larger training sets. This can be seen for all muscles by comparing the red and blue bars in Fig. 7 (b) where seven of eight trajectories were used to train models. For example, when the radial nerve was stimulated (Fig. 6), with training sets of four, five, or six trajectories the semiparametric models had significantly smaller mean errors than did the parametric models. With ample training data the semiparametric models had the flexibility of the Gaussian process to capture more complex dynamics while the parametric models were limited by their fixed basis functions.

B. Multiple-Muscle Control Utility

The semiparametric models' accuracy in torque predictions when stimulating multiple muscles used to control arm motion was comparable to accuracy when stimulating individual muscle groups. This is seen by comparing the multiple-muscle error on the far right in Fig. 7 (b) to the single-muscle errors in the same figure; errors were roughly 20% of the maximum torque magnitude for both single-muscle predictions and multiple-muscle predictions. Note that the errors were normalized by the total muscle torque, so the absolute errors were larger when stimulating multiple muscles. These results are from the multi-

muscle trials during the control utility session, which occurred two days after the models were identified. The robot drove the subject's hand along each of the eight reaching trajectories while stimulating muscle groups chosen according to the optimization problem in (12) given the semiparametric models identified during the second identification session.

The semiparametric models were capable of accurately predicting torques while stimulating multiple muscles given the predicted confidence intervals for some specific trajectories, but not for others. For 73% of all torque measurements over all the multi-muscle reaching trials, the actual joint torques fell within the 95% confidence intervals of the semiparametric models' predictions. This obviously suggests that there were cases with large errors outside of those confidence intervals. Large errors occurred when muscle stimulation could not produce sufficient torque to fully execute a movement, so the robot made a significant contribution to torque. In these large error cases, many muscles combined to produce a small net torque.

A particular case of multi-muscle torque predictions is shown in Fig. 8. Each of the torque predictions follows each of the actual torques, and each of the actual torques falls almost entirely within the confidence intervals of the model's predictions (Fig. 8(b)). There was bias error in the shoulder torque predictions. We attribute this to changes in the subject's shoulder stiffness in the two days between the model identification and the control utility sessions. In this example the robot moved the subject's hand from target T2, which was high and to the right, to target B1, which was in front of the subject's mouth, and back to target T2. There was little pectoral stimulation (Fig. 8(a)) because almost zero positive elevation plane torque was required. The radial nerve, which targets triceps, was stimulated at the start and end of the trial to provide negative elbow flexion torque to hold the arm in the extended position. The musculocutaneous nerve, which targets biceps and brachialis, was stimulated in the middle of the movement to provide positive elbow flexion torque (Fig. 8 (a) and (b) far right).

IV. Discussion

This paper presents a method to identify models of the dynamics of any specific human arm controlled by functional electrical stimulation. The inverse arm dynamics model specifies the joint torques required to drive an arm on a desired trajectory. The model of muscle torque production specifies the shoulder and elbow torques that the muscles provide. We showed that the models accurately predicted torques within confidence intervals, and we showed that the semiparametric model type generally made more accurate torque predictions than parametric and nonparametric model types.

A. Single-Muscle Model Identification and Validation

The errors in predictions of torques were on average less than 20% of the maximum torques achievable by the subject (Fig. 7(b)). The most relevant comparison is the work of Freeman and colleagues [14] in parametric modeling of the arm dynamics with FES in subjects with stroke. In [14] normalized errors are generally a bit smaller (7.7%–24% in Table 3). The comparatively larger errors in our study are expected. Our current study examined nine muscle groups' effects about multiple degrees of freedom for motion in three dimensions,

while [14] reports errors for the elbow flexion torque for planar movements of the arm when stimulating only triceps. To our knowledge there are no other published studies on FES motion control of multiple degrees of freedom of the arm or leg for arbitrary goals.

One source of error in our torque predictions is the uncertainty in measurement of the inputs to our model. Motion capture systems give imperfect estimates of joint positions and velocities due to kinematic modeling assumptions and markers moving as muscles contract. We can estimate uncertainties in predictions of joint positions and velocities with the RTS smoother and in the future incorporate these uncertainties into the Gaussian process framework to make better torque predictions.

Another source of error is the sparseness of the training sets used to identify the models. We chose specific reaching movements because they are relevant to tasks such as pushing buttons or reaching for food. For this set of reaching movements, the choice of training trajectories from the set did not have a significant impact on the models' prediction errors. Collecting larger training sets could be useful, but expensive in terms of time to train the models. We could potentially collect more informative training data that includes more varied combinations of joint positions and velocities as is done with persistently exciting trajectories in robotic identification [23]. This is especially challenging as we only have control over the position of the subject's hand, but we do not have direct control over joint movements without a model that already makes accurate predictions.

Another source of error lies in the variability in the response of the arm to stimulation. An example is the difference between the three elbow flexion torques for three nominally identical trials in Fig. 4(b). This natural variability can be due to spasticity, fatigue, and the loosening of muscles over the duration of a session among other factors.

Despite these errors our model estimates confidence intervals that capture uncertainty in the predictions. A major strength of Gaussian process regression and statistical models in general is the ease of estimation of confidence intervals when compared to nonlinear parameter estimation or artificial neural networks. These confidence intervals can be used in control as there are multiple sets of stimulations that might have the same deterministic result, but one set for which we have the greatest confidence in achieving the desired hand motion giving the uncertainty in muscle torque predictions.

The semiparametric model is especially well-suited for use in human subjects with SCI. The semiparametric model type is superior to the nonparametric type for smaller training sets and superior to the parametric type for larger training sets (Fig. 6). When working with people with spinal cord injuries, gathering large amounts of system identification data is not feasible in a single session, hence the need for the semiparametric type vs. the nonparametric type, but larger data sets can be gathered and refined over time, hence the need for the semiparametric type vs. the parametric type.

B. Multiple-Muscle Control Utility

When stimulating multiple muscles the model had similar magnitude normalized errors as when stimulating individual muscle groups (Fig. 7(b)). This result is especially encouraging

for a number of reasons. First, the activation states of muscles were uncertain during the multiple-muscle trials as they were estimated via recruitment curves; activation was known exactly – 100% activation – in the single-muscle trials. Second, the normalized errors were similar for multiple-muscle and single-muscle trials despite the multiple-muscle trials occurring two days after the models were identified. Finally, approximating the total torque due to stimulation of multiple muscle groups as a linear sum of individual muscle-group torques did not increase error over single-muscle trials; this reinforces our earlier results in force control [13].

C. Implications for Control

The model developed in this work could become an integral part of a model-based controller for arm motions. Although the current work focuses on evaluating the model's torque predictions, we have already demonstrated that the approach used in the control utility experiments can be used for feedforward control of arm motion [24]. Extending this method to feedback control will be essential to improve tracking performance and disturbance rejection, but there remain many challenges.

One is planning for limitations in controllability of the arm. For any given set of joint positions and velocities of the subject's arm, the muscles controlled by FES may not be able to produce accelerations in all directions in joint space due to specific patterns of denervation and weakness associated with each subject. This results in a system that is not locally controllable at each state. None of the eight hand trajectories described in this paper consisted entirely of states for which the arm was locally controllable. Some trajectories had a small percentage of uncontrollable states, while those that required significant shoulder elevation torque had many uncontrollable states. In this work we used an external robot to resolve the controllability issue [12]. An alternative solution would be to explicitly consider controllability in the trajectory planning process, which has been recently explored in robotics [25].

Reducing the feedforward prediction errors would also improve performance of a feedback controller. This is challenging due to the ever changing response of a person's muscles to electrical stimulation. This change occurred within a single session in our study, as can be seen in the variation in elbow flexion torque in Fig. 4(b). It also occurred from session to session, as can be seen in the bias errors in shoulder torques in Fig. 8(b). Although our models made torque predictions when stimulating multiple muscles with similar accuracy to another study with fewer muscles stimulated [14], it is unclear how well our identification methods will generalize over longer periods of time or for other subjects.

In general, the variation of response to stimulation across subjects and over time is a challenge in FES control. Our approach has been to identify subject-specific models to capture differences across subjects. Our approach can be extended to address the challenge of time-varying response by making adaptive updates to the models. The computational challenges associated with this approach could be addressed by updating only the parametric part of our model. Another option is using locally-weighted regression [26], which is less computationally expensive than Gaussian process regression, but places formal requirements on the distribution of training data. Alternatively a promising combination of the benefits of

Gaussian process regression and locally weighted regression is local Gaussian process regression [27].

D. Conclusion

The method presented in this paper is critical in the design of flexible controllers for functional electrical stimulation. It allows us to use a model-based controller where we predict the joint torques required to produce a desired arm motion. It also allows us to determine the stimulation inputs to the muscles to supply the required torques. Now we can specify a hand trajectory, compute the corresponding trajectory in joint space, and then determine the joint torques required to execute the trajectory.

Previous work has paved the way by focusing on controlling single joints with antagonist pairs of muscles [28, 29], controlling multiple joints with fixed stimulation patterns [17, 30, 31], or by demonstrating flexible control in computer simulation [6]. Our method unlocks the potential to flexibly control arbitrary movements of multiple degrees of freedom with multiple muscles for any human subject with an FES neuroprosthesis.

Acknowledgments

This work was supported by NSF grant 0932263, NIH NINDS grant N01-NS-5-2365, NSF Graduate Fellowship DGE-0824162, and the Cleveland State University Graduate Faculty Travel Award Program.

References

1. Ragnarsson KT. Functional electrical stimulation after spinal cord injury: Current use, therapeutic effects and future directions. *Spinal Cord*. 2008; 46:255–274. [PubMed: 17846639]
2. Peckham PH, Knutson JS. Functional electrical stimulation for neuromuscular applications. *Annual Reviews in Biomedical Engineering*. 2005; 7:327–360.
3. Sheffler LR, Chae J. Neuromuscular electrical stimulation in neurorehabilitation. *Muscle Nerve*. 2007; 35(5):562–590. [PubMed: 17299744]
4. Davoodi, R., Loeb, GE. A computer model of the human arm to study the control of FES-assisted reaching; 24th Annual International Conference of the IEEE Engineering in Medicine and Biology Society; 2002.
5. Kirsch, RF., Acosta, AM. Model-based development of neuroprostheses for restoring proximal arm function; 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society; 2001.
6. Blana D, Kirsch RF, Chadwick EK. Combined feedforward and feedback control of a redundant, nonlinear, dynamic musculoskeletal system. *Medical & Biological Engineering & Computing*. 2009; 47:533–542. [PubMed: 19343388]
7. Park H, Durand DM. Motion control of musculoskeletal systems with redundancy. *Biological Cybernetics*. 2008; 99:503–516. [PubMed: 18985380]
8. Perreault EJ, Heckman CJ, Sandercock TG. Hill muscle model errors during movement are greatest within the physiologically relevant range of motor unit firing rates. *Journal of Biomechanics*. 2003; 36(2):211–218. [PubMed: 12547358]
9. Rasmussen, CE., Williams, CKI. *Gaussian Processes for Machine Learning*. Cambridge, MA: The MIT Press; 2006.
10. Nguyen-Tuong, D., Peters, J. Using model knowledge for learning inverse dynamics; International Conference on Robotics and Automation; 2010.
11. Wu G, van der Helm FC, Veeger HE, Makhsous M, Roy PV, Anglin C, Nagels J, Karduna AR, McQuade K, Wang X, Werner FW, Buchholz B. ISB recommendation on definitions of joint

coordinate systems of various joints for the reporting of human joint motion – part II: Shoulder, elbow, wrist and hand. *Journal of Biomechanics*. 2005; 38(5):981–992. [PubMed: 15844264]

12. Todorov E. Probabilistic inference of multijoint movements, skeletal parameters and marker attachments from diverse motion capture data. *IEEE Transactions on Biomedical Engineering*. 2007; 54(11):1927–1939. [PubMed: 18018688]
13. Schearer EM, Liao Y, Perreault EJ, Tresch MC, Memberg WD, Kirsch RF, Lynch KM. Multi-muscle FES force control of the human arm for arbitrary goals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2014; 22(3):654–663. [PubMed: 24122573]
14. Freeman CT, Hughes AM, Burridge JH, Chappell PH, Lewin PL, Rogers E. A model of the upper extremity using FES for stroke rehabilitation. *Journal of Biomechanical Engineering*. 2009; 131(3):1–12.
15. O'Hagan A. Curve fitting and optimal design for prediction. *Journal of the Royal Statistical Society*. 1978; 40:1–42.
16. Polasek KH, Hoyen HA, Keith MW, Kirsch RF, Tyler DJ. Stimulation stability and selectivity of chronically implanted multicontact nerve cuff electrodes in the human upper extremity. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2009; 17(5):428–437. [PubMed: 19775987]
17. Smith B, Peckham PH, Keith MW, Roscoe DD. An externally powered, multichannel, implantable stimulator for versatile control of paralyzed muscle. *IEEE Transactions on Biomedical Engineering*. 1987; 34(7):499–508. [PubMed: 3610200]
18. Smith B, Tang Z, Johnson MW, Pourmehdi S, Gazdik MM, Buckett JR, Peckham PH. An externally powered, multichannel, implantable stimulator-telemeter for control of paralyzed muscle. *IEEE Transactions on Biomedical Engineering*. 1998; 45(4):463–475. [PubMed: 9556963]
19. Hart RL, Bhadra N, Montague FW, Kilgore KL, Peckham PH. Design and testing of an advanced implantable neuroprosthesis with myoelectric control. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2011; 19(1):45–53. [PubMed: 20876029]
20. Memberg WD, Peckham PH, Keith MW. A surgically-implanted intramuscular electrode for and implantable neuromuscular stimulation system. *IEEE Transactions on Rehabilitation Engineering*. 1994; 2(2):80–91.
21. Naples GG, Mortimer JT. A spiral nerve cuff electrode for peripheral nerve stimulation. *IEEE Transactions on Biomedical Engineering*. 1988; 35(11):905–916. [PubMed: 3198136]
22. Durfee WK, MacLean KE. Methods for estimating isometric recruitment curves of electrically stimulated muscle. *IEEE Transactions on Biomedical Engineering*. 1989; 36(7):654–666. [PubMed: 2744790]
23. Ortega R, Spong MW. Adaptive motion control of rigid robots: A tutorial. *Automatica*. 1989; 25(6):877–888.
24. Schearer, EM., Liao, Y., Perreault, EJ., Tresch, MC., Memberg, WD., Kirsch, RF., Lynch, KM. Evaluation of a semiparametric model for high-dimensional FES control; 7th International IEEE EMBS Conference on Neural Engineering; 2015.
25. Shkolnik, A., Walter, M., Tedrake, R. Reachability-guided sampling for planning under differential constraints; International Conference on Robotics and Automation; 2009.
26. Atkeson CG, Moore AW, Schaal S. Locally weighted learning. *Artificial Intelligence Review*. 1997; 11:11–73.
27. Nguyen-Tuong D, Seeger M, Peters J. Model learning with local gaussian process regression. *Advanced Robotics*. 2009; 23(15):2015–2034.
28. Sharma N, Stegath K, Gregory CM, Dixon WE. Nonlinear neuromuscular electrical stimulation tracking control of a human limb. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2009; 17(6):576–584. [PubMed: 19497828]
29. Jezernik S, Wassink RGV, Keller T. Sliding mode closed-loop control of fes: Controlling the shank movement. *IEEE Transactions on Biomedical Engineering*. 2004; 51(2):263–272. [PubMed: 14765699]
30. Zhang D, Zhu K. Modeling biological motor control for human locomotion with functional electrical stimulation. *Biological Cybernetics*. 2007; 96:70–97.

31. Li, P., Hou, Z., Zhang, F., Tan, M., Wang, H., Hong, Y., Zhang, J. An FES cycling control system based on CPG; 31st Annual International Conference of the IEEE EMBS; 2009. p. 1569-1572.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Fig. 1.

The torques from the robot are the shoulder and elbow joint torques that would produce an identical arm motion as the forces applied by the robot to the subject's distal forearm during a reaching trial. When no muscle groups are stimulated (a), the torques from the robot are those torques $p(x)$ required to drive the arm on a trajectory. When one muscle group is stimulated (b) the torques from the robot are equal to the difference between $p(x)$ and the active muscle torques. In each trial we choose the muscle group activation; we compute the robot torques, joint positions, and joint velocities based on sensor measurements; using our identification techniques we infer the inverse of the arm dynamics block and the forward muscle torque production block.

Fig. 2. Instrumentation for experiments. Pictured is the subject's right arm. Shown in (a) and (b) are the placement of the rigid bodies with reflective markers for optical tracking, a mobile arm support with rubber bands providing supporting force, and the robot that drives the subject's hand from green targets (T1, T2, T3, and T4) in front of the subject's body to blue targets (B1 and B2) close to the subject's body. The subject's hand and distal forearm (c) are in a soft cast that is attached via a magnet to a ball-and-socket joint at the end effector of the HapticMaster robot.

Fig. 3.

Controller block diagram for stimulation of multiple muscles during control utility experiments.

Fig. 4.

Filtered joint angles and joint torques for the arm trajectory from the target T1 to target B1 and back to T1 (see Fig. 2) during Session 1. Results for trials with no stimulation (a) and for trials with stimulation of the radial nerve (b) are shown. The trajectory was run three times. The same nominal joint trajectory was followed for each repetition, so only one of three joint angle trajectories is pictured. Torque predictions were made using a semiparametric model that was trained with the seven trajectories that were different from the trajectory pictured. The shaded regions bounded by thin dashed lines represent the 95% confidence intervals of the predictions of the semiparametric model.

Fig. 5.

Distribution of model errors: The blue histogram represents the generalization errors for all semiparametric models trained with seven of eight trajectories and tested with the one trajectory not included in the training set. The errors are normalized by the standard deviation of the models' predictions. The orange distribution is the distribution of model errors predicted by semiparametric Gaussian process regression.

Fig. 6.

Learning curve for the normalized generalization error magnitude for reaching trajectories when the radial nerve was stimulated during the second identification session. The error magnitude is the torque error over all four degrees of freedom normalized by the maximum torque magnitude for all trials with this muscle group during the second identification session. Error bars represent the mean plus or minus the standard error of the mean.

Fig. 7.

Mean normalized generalization error magnitude for models trained with one trajectory (a) and seven trajectories (b) during the second identification session. Error bars represent the mean plus or minus the standard error of the mean. Labels for muscle groups are those used in Table I with the addition of “P” representing passive trials where no muscle groups were stimulated. Results from the multi-muscle torque predictions of semiparametric models are labeled “MM”. The “MM” models were trained using eight trajectories from the second identification session and tested during the control utility session. Each bar represents the magnitude of the torque error over all four degrees of freedom. The errors were normalized by the maximum torque magnitude for all trials with each muscle group during the second identification session. Errors for multi-muscle stimulation trials during the control utility session were normalized by the total predicted muscle torque.

Fig. 8.

Muscle group stimulations (a) and total joint torques when multiple muscle groups are stimulated (b) for the arm trajectory from the target T2 to target B1 and back to T2 (see Fig. 2) during the control utility session. The shaded regions bounded by thin dashed lines represent the 95% confidence intervals of the semiparametric model's torque predictions.

TABLE I

Stimulation electrodes used

Electrode Placement	Muscles Targeted	Approximate Function	Type	Current Amplitude (mA)	Max Pulse Width (μ s)
radial nerve	triceps	elbow extension	nerve cuff	0.8	47
axillary nerve	deltoids	arm abduction	nerve cuff	2.1	50
thoracodorsal nerve	latissimus dorsi	arm adduction	nerve cuff	0.8	25
long thoracic nerve	serratus anterior	scapular abduction	nerve cuff	1.4	12
musculocutaneous nerve	biceps, brachialis	elbow flexion	nerve cuff	0.8	45
suprascapular nerve	supraspinatus, infraspinatus	shoulder stability, humeral rotation	nerve cuff	1.4	15
rhomboids	rhomboids	scapular adduction	intramuscular	18.0	100
lower pectoralis	lower pectoralis	shoulder horizontal flexion	intramuscular	18.0	115
upper pectoralis	upper pectoralis	shoulder horizontal flexion	intramuscular	20.0	70