Feedback Control of Functional Electrical Stimulation for Arbitrary Upper Extremity Movements

Reza Sharif Razavian, Borna Ghannadi, John McPhee
Systems Design Engineering
University of Waterloo
Waterloo, Ontario, Canada
Email: rsharifr@uwaterloo.ca, bghannad@uwaterloo.ca, mcphee@uwaterloo.ca

***Abstract*—Functional electrical stimulation (FES) is a type of neuroprosthesis in which muscles are stimulated by electrical pulses in order to compensate for the loss of voluntary movement control. Modulating the stimulation intensities to reliably generate movements is a challenging control problem. For the first time, this paper presents a feedback controller for FES to control arm movements in a 2D (table-top) task space. This feedback controller is based on a recent human motor control model, which uses muscle synergies to simplify the calculations and improve control performance. The experimental results show that this control scheme can produce arbitrary movements in the 2D task space, with less than 2 cm hand position error from the specified targets.**

I. Introduction

Functional electrical stimulation (FES) is the process of applying electrical pulses to skeletal muscles in order to produce force and perform an action. Artificially stimulating the muscles to produce movements in individuals with motor impairment (e.g. spinal cord injury or stroke patients) has always been an intriguing idea; however, there are basic challenges that prevent FES to be widely used.

The biggest challenge is associated with the complexity of the human musculoskeletal system; it is a redundant, non-linear and noisy system with significant uncertainties. The actuators in the human body are the muscles, many of which are co-activated by the nervous systems to produce a certain motion. We do not know how the nervous system calculates the required muscle activities. Thus, we have yet to mimic the same process to produce naturally-looking motions with FES.

There have been attempts to develop feedback motion controllers for FES, but the results are far from perfect. Proof of concept studies for more advanced feedback controllers exist that report successful control of simple motions (e.g. knee extension by [1]–[4], elbow extension by [5], and two-degree-of-freedom ankle motion by [6]). However, their generalizability to multi-joint and multi-muscle systems has not been reported.

Controlling the upper extremity movements with FES is a more challenging control problem, mostly because of the larger number of degrees of freedom and the involved muscles. Initial steps toward solving this problem include a single-muscle controller for a one-dimensional reaching task [7], control of isometric hand force in a multi-muscle FES system [8], estimation of FES-produced joint torques during a prescribed 3D movement [9], and feed-forward control of the reaching tasks [10]. Controllers for elbow extension using the measured shoulder joint velocities have also been developed [11], [12]. However, no feedback controller exists for the upper extremity reaching movements that allows reaching to arbitrarily specified targets.

The developed motion controllers in [7]–[10] have all taken a joint space approach, in which the effect of muscle activities on joint torques are calculated, and then inverse kinematics/dynamics are used to convert the information from task space (hand position, which is the controlled variable) to joint space (for which the models are developed). This added complexity can be removed by developing the formulation in the task space, and considering the effect of muscles on the task space forces, rather than the joint torques.

In this paper, we employ a biologically plausible human motor control model [13], [14] as the feedback controller for FES. The advantage of this model is in its task space representation, and utilization of muscle synergies to simplify calculations. The goal of the controller is to drive the arm to follow arbitrary trajectories in a two-dimensional workspace. To the best of our knowledge, this is the first feedback FES controller for the multi-joint and multi-muscle arm movements.

II. Methods

A. The experimental setup

Our experimental setup is shown in Fig. 1. In this setup, RehaStim V.1 with Science Mode (Hasomed GmbH, Germany) is used as the stimulation device. Five muscles (anterior and posterior deltoid, biceps and triceps brachii, and pectoralis major) are stimulated via electrode pads placed on the skin and over the muscle motor points [15]. The stimulation frequency and current are fixed at 30 Hz and 20 mA,

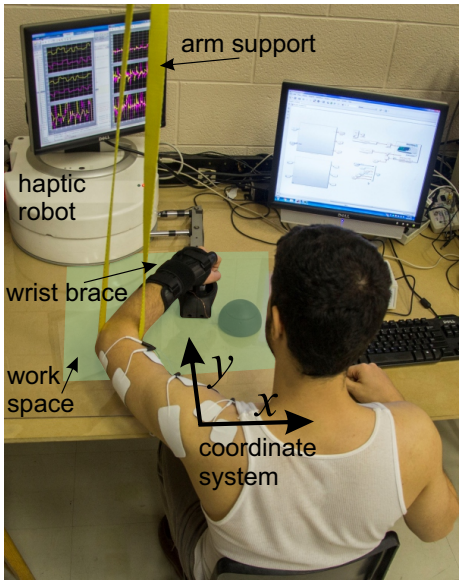

Fig. 1. The experimental setup

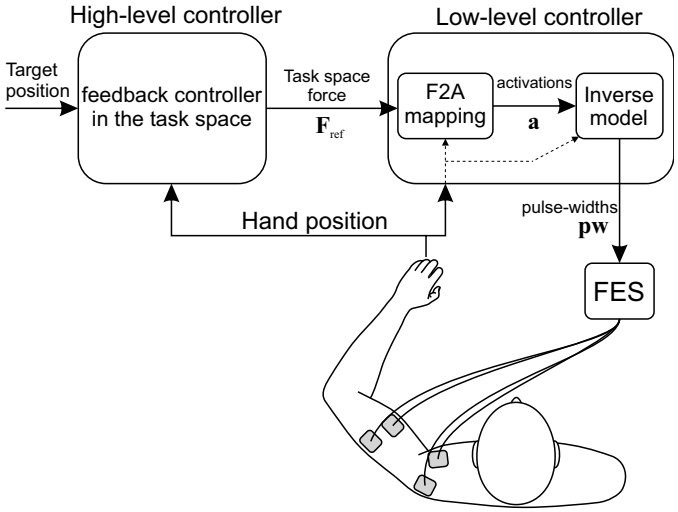

Fig. 2. The overview of the motor control model as the FES feedback controller

respectively, and stimulation pulse width is modulated using a computer running Matlab/Simulink.

The goal in this setup is to use FES to control the position of the hand, and drive it to any arbitrary target in a 2D (tabletop) space. This FES system was tested on the left arm of a healthy subject (28 years old male). To make sure that the subject's voluntary actions are minimized, he was asked to relax, and no information about the target positions were visible to him.

To control the hand position in the 2D tabletop space, its position and the applied force are measured using a 2D haptic robot (Quanser Inc. Canada). The robot can be programmed to be locked in place (for isometric trials), or move with/without applying resistance (during motion trials). The subject sits upright in front of the robot, and is instructed to relax during stimulations to minimize the voluntary force production. A wrist brace is used to restrict wrist motion, and also firmly connect the hand to the robot's end-effector. To reduce the arm-table friction and keep the arm in a horizontal plane, the arm is suspended using a sling. Since the arm always remains elevated by the rope at a certain height, the effective degrees of freedom of the arm is reduced to two, and (x, y) position of the hand is sufficient to describe the posture (given that the torso movement is restricted).

B. FES controller architecture

Our FES controller is the real-time implementation of a synergy-based motor control framework [13], [14]. The advantage of this motor control model is in its task space representation, which simplifies the calculations and allows for efficient control of the movements in the task space. Task space in this context refers to the collection of the kinematic variables that are actively controlled. Therefore, the task space in this paper is defined as a 2D Cartesian space, specified by the (x, y) position of the hand. Alternatively, for example, if the objective was to control the elbow flexion angle only, the task space would be a 1-dimensional space defined by elbow angle (the hand position would be irrelevant in this case).

The schematic of our FES feedback controller is shown in Fig. 2. The high-level controller in our motor control implementation is a PID controller, whose input is the position error in the task space. This task space controller defines a 2D reference force vector, $\mathbf{F}_{ref}$, that is required to reach to the specified target. This force command should be fulfilled by activating the muscles. The role of the low-level controller in this motor control framework is to translate the task space force command to muscle activations, and consequently to stimulation pulse-width (pw) which is fed to the FES device. The details of this force-to-activation (F2A) mapping are described fully in [14] and are briefly explained below.

The F2A mapping takes advantage of the known actions of a number of *muscle synergies* in the task space. A muscle synergy in this work is defined as the co-activation of a number of muscles with known relative ratios. The difference between our definition of muscle synergies from the traditional definitions is the dependence of activation ratios on the posture (posture-dependent synergies). In a given posture, defined by the hand's (x, y) position in the task space, the activation of a single synergy produces a certain isometric hand force. The collection of the force vectors produced by individual synergies can be considered as a *basis set* for the task space. In other words, an arbitrary hand force vector can be constructed by a linear combination of these synergy-produced basis vectors.

The known synergies and basis vectors (pre-computed off-line, see section II-C2) can be used to solve for the muscle activations in real-time, instead of the time-consuming op-

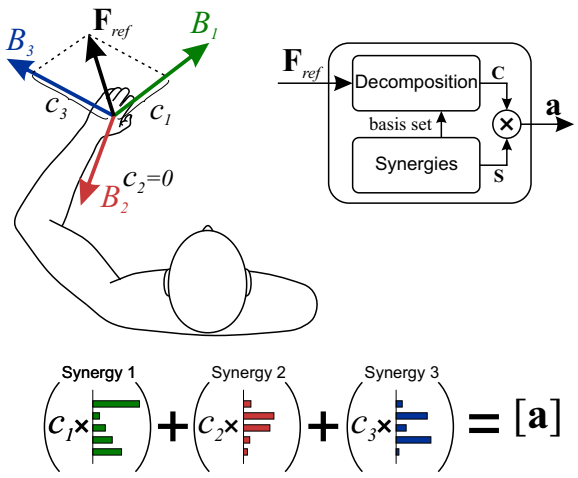

Fig. 3. The schematic of the force-to-activation mapping. Each synergy produces a certain force in the task space that can be considered as a basis vector. An arbitrary reference force, $\mathbf{F}_{ref}$, can be decomposed onto this basis set, to calculate the coefficient of each synergy, c_i. Combing the synergies with the corresponding coefficients gives the vector of muscle activations that produces the reference force.

timization process (e.g. in [8], [10]). The force command defined by the task space controller can be decomposed onto the basis set to calculate the coefficient of each basis vector (Fig. 3). Then, the synergies are combined using the calculated coefficients to obtain the muscle activations that produce the reference force. The schematic and block diagram of this process is shown in Fig. 3.

C. Implementation of the FES controller

To implement this motor control framework as the FES controller, *models* of stimulated muscles are needed. Because of the task-dependent architecture of the motor control framework, it is only required to identify a mapping that transforms the muscle stimulation intensity to the task space force vector—no explicit information about kinematics of the body (e.g. segment lengths and joint angles) and joint moments is necessary.

1) Identifying pulse-width-to-force mappings: The muscle models used in this work are static input-output functions that estimate the 2D task space force from the stimulation intensity, pw, and the hand position:

$$\begin{aligned} \mathbf{F}_{est}: \quad & \mathbb{R}^3 \rightarrow \mathbb{R}^2 \\ & (pw, x, y) \mapsto (F_x, F_y) \end{aligned} \tag{1}$$

To obtain mappings of this form, individual muscles are isometrically stimulated (robot is locked in place) one by one with the patterns shown in Fig. 4, and the hand force/position is measured by the robot. The stimulations are repeated at nine different positions in the task space (a 3-by-3 grid of 30 cm length).

The mapping in (1) is constructed by training two separate functions: one for the direction of the force, and one for its amplitude. The reason for this separation is to ensure that the direction of the estimated force remains the same for all intensity levels (ideally this direction should only depend on the geometry of the musculoskeletal system). This assumption may not be entirely true, as higher stimulation intensities may excite the neighbouring muscle fibres and alter the direction of the hand force. Nonetheless, the motor control model relies on this assumption; thus, the force direction mapping is trained despite the changing stimulation intensity, and is only a function of posture.

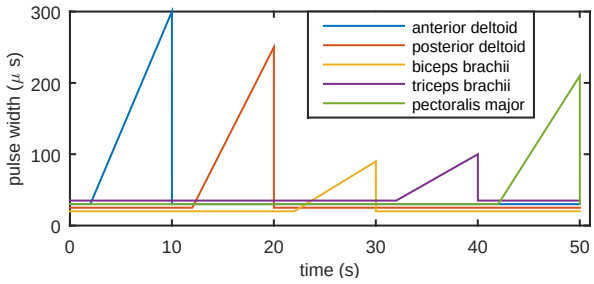

Fig. 4. The stimulation profiles used to train the muscle mappings. For each muscle, the maximum pulse width is the threshold of the stimulations becoming uncomfortable for the subject, and the base value is the onset of feeling the stimulations on the skin. These profiles are constant for all postures.

The direction mapping is trained using the best line of action of the measured forces at various postures. This line of action is obtained by fitting a line (zero y-intercept) to the measured force (as seen in the enlarged plot in Fig. 5). Next, the unit vector, $\hat{\mathbf{u}}_F$, representing this direction is found, and second-order polynomial surfaces ($\mathcal{P}(x, y)$) are used to fit its u_x and u_y components as functions of the hand position.

$$\hat{\mathbf{u}}_F = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \begin{bmatrix} \mathcal{P}_1(x, y) \\ \mathcal{P}_2(x, y) \end{bmatrix} \tag{2}$$

Fig. 5 shows the estimated force directions resulting from pectoralis major muscle stimulation.

The amplitude mapping is obtained by training an artificial neural network (ANN). The inputs to the training are the measured (pw, x, y) triplets, and the output is the amplitude of the measured force. As an example, Fig. 6 shows the trained mapping (with one hidden layer neuron) for the pectoralis major muscle.

The two functions are then combined to obtain the *posture-dependent pulse-width-to-force mappings* for individual muscles:

$$\mathbf{F}_{est}(pw, x, y) = \hat{\mathbf{u}}_F(x, y).F_{est}(pw, x, y) \tag{3}$$

where $\hat{\mathbf{u}}_F$ is the function that estimates the force unit vector (the polynomial surfaces), and F_{est} is the ANN mapping that estimates its amplitude.

Since the pulse width non-linearly affects the hand force, it cannot be used in the synergy framework mentioned above to solve for muscle activity levels—muscle synergy theory relies on superposition principle, with works only in linear systems. Therefore, it is useful to define the *muscle activation* as the linear scaling of maximum muscle force as:

$$a(pw, x, y) = \frac{F_{est}(pw, x, y)}{F_{est}(pw_{max}, x, y)} \tag{4}$$

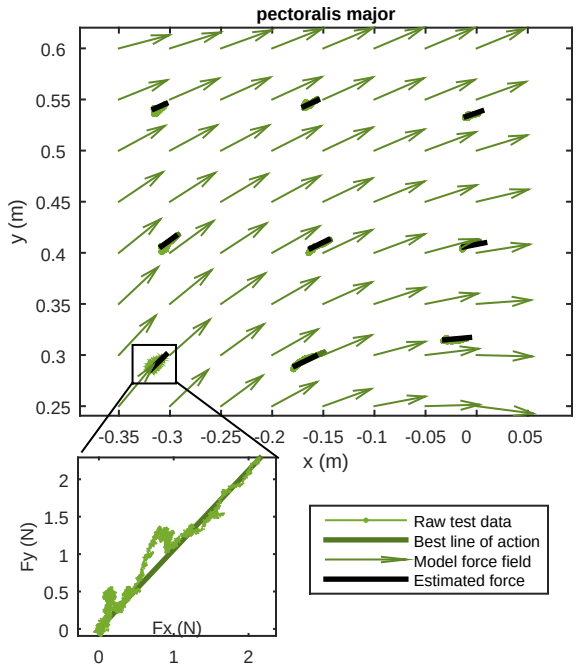

Fig. 5. An example of the trained mapping for the direction of the force in the entire work space (pectoralis major muscle). A line of action found by fitting a line to the measured force data is shown in the zoomed plot. The arrows show the direction of the force estimated by the mapping. The lines represent the measured data (thin lines), the best line of action (thick lines), and the estimated force (direction and magnitude) by the pulse-width-to-fore mapping. The force measurements are shown with 1N = 0.005m scale in this plot. The hand position is given in the coordinate system located at the left shoulder, $+x$ is to the right, and $+y$ is forward.

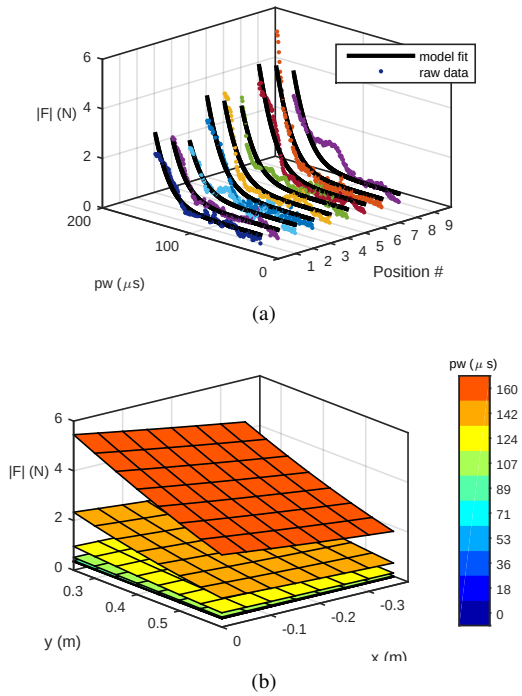

Fig. 6. (a) The ANN mapping fitted to the measured data (pectoralis major muscle as an example). The data collected at 9 different position in the work space. (b) The mapping estimations for the amplitude of the force as a function of stimulation pulse width in the entire work space. The results shown belong to pectoralis major muscle.

which can be used in F2A mapping without problem, instead of the pulse width. Once the required muscle activations are calculated through the processes in F2A mapping, *inverse muscle models* can be used to calculate the needed stimulation pulse widths.

The trained mapping in (3) is a *forward* mapping (transforming pw to hand force). In order to calculate the pulse-width required to generate a certain amount of muscle activation, an inverse mapping is also trained using another ANN model (with 8 hidden layer neurons). To ensure the integrity of inverse and forward mappings, the inverse mapping is trained using the simulated data obtained from the forward mapping, instead of the measured data.

2) Obtaining synergies: For the muscle synergies to be useful in our motor control framework, they need to have an important feature: their basis vectors must span the task space in such a way that any force vector could be a linear combination of the basis vectors with *positive coefficients* (muscles are pull-only elements). This requirement implies that: 1) at least three synergies are required to control the motion in the 2D task space, and 2) their basis vectors cannot all be in one half-plane. Furthermore, the structure of the synergies has great effect on the efficiency of the control loop [14]. As a result, proper identification of the synergies is a critical step in this motor control model.

To obtain the synergies, a large number of optimization problems are solved. In a certain posture, *the best* set of muscle activations that result in a target hand force are found. The target hand forces, $\mathbf{F}_{trgt}$, in our optimizations are 15 vectors equally spaced in the horizontal plane, and the optimization algorithm tries to minimize the objective function J:

$$\mathbf{a}^* = \underset{0<a<1}{\arg\min}\{J\} \tag{5}$$

where

$$J = \omega_1 \left| \left(\sum_{i=1}^{5} \mathbf{F}_{est}^i \right) - \mathbf{F}_{trgt} \right|^2 + \omega_2 \sum_{i=1}^{5} pw_i + \omega_3 \sum_{i=1}^{5} {F_{est}^i}^2 \tag{6}$$

where the index i represents the i^{th} muscle, and ω's are weighting factors to balance and non-dimensionalize the terms. In this cost function, the first term is the constraint of total force being close to the target force, the second term penalizes high stimulations, and the last one has the effect of sharing the load among similar-function muscles. The weightings ($w_1 = 1\times10^3\ N^{-2}$, $w_2 = 1\times10^{-6}\ \mu s^{-1}$, $w_3 = 1\ Ns^{-2}$ are chosen so as to result in the co-activation of the muscles that is qualitatively similar to the natural human behavior.

The optimal activations that produce the 15 target forces can be gathered in the matrix:

$$\mathbf{A}_{5\times15} = [\mathbf{a}_1^*, \mathbf{a}_2^*, \ldots, \mathbf{a}_{15}^*] \tag{7}$$

The usual practice to obtain muscle synergies is to use non-negative matrix factorization (NNMF, [16]) on such a data

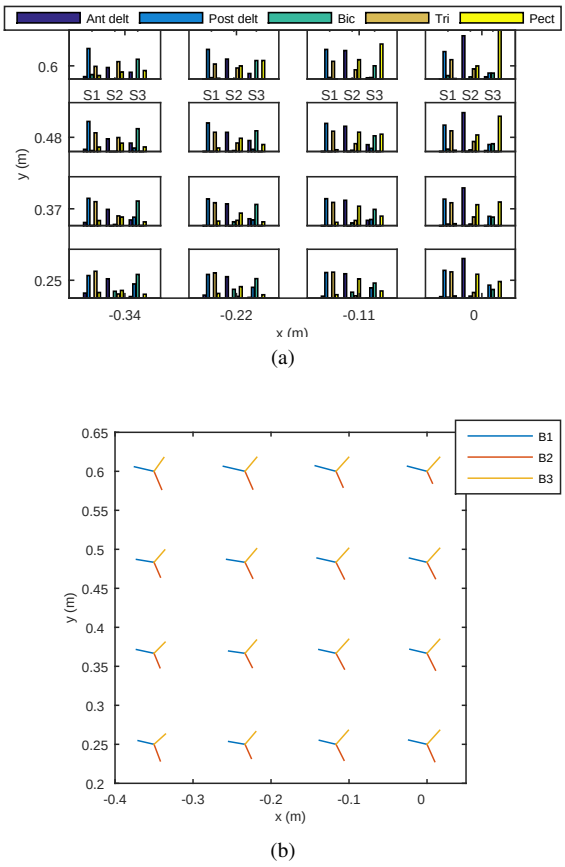

Fig. 7. (a) The posture-dependent synergies, and (b) posture-dependent basis set shown at multiple points in the work space. The color-coded basis vectors are shown with 1N = 0.005m scale in this plot.

matrix to break it into a synergy matrix $\mathbf{S}$ and a coefficient matrix $\mathbf{C}$ as:

$$\mathbf{A}_{5\times 15} \simeq \mathbf{S}_{5\times n}\mathbf{C}_{n\times 15} \tag{8}$$

where n is the number of synergies (represented by the columns of $\mathbf{S}$).

This process can be repeated at a multitude of postures to obtain the *posture-dependent synergy matrices* (i.e. $\mathbf{S} = \mathbf{S}(x, y)$, Fig. 7a). Next, individual synergies can be fed to the muscle mappings to obtain the estimate of the total hand force that each synergy produces. The collection of these forces form the posture-dependent basis set for the task space (see Fig. 7b).

During real-time control of the movement, the hand position is measured by the robot, which is used to interpolate between the previously calculated synergy matrices and basis sets, to obtain the synergies/bases at the current posture. This information is then used in the F2A mapping to get the muscle activations, which are finally converted by the inverse mappings to pulse-width commands to be fed to the FES device.

III. Results

The closed-loop FES control performance is shown in Fig. 8. Two control methods are compared here. One is

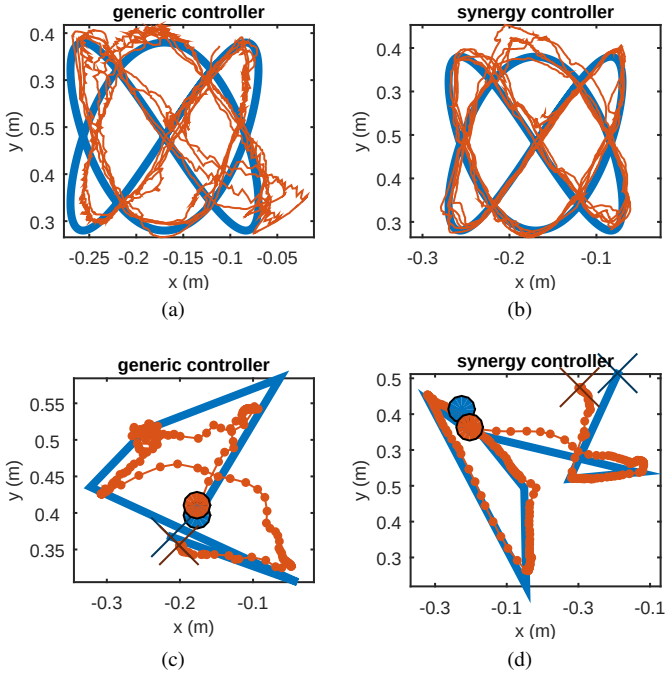

Fig. 8. The feedback control performance resulting from the two control methods. The plots show the reference (thick blue line) and actual movements (thin red lines) of the hand in the task space. Two scenarios are shown: a deterministic pattern in (a) and (b), and randomly generated paths in (c), and (d). In random motion trials, the starting positions are shown by the coloured circles.

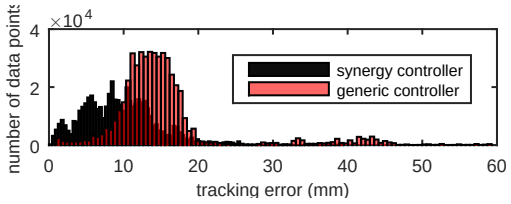

Fig. 9. The histogram of the tracking error resulting from the two controllers.

the synergy-based motor control model described above. The second controller is a model-free *generic* controller that has no knowledge of the action of the muscles. This controller co-activates anterior deltoid and pectoralis major muscles (with equal intensities) when the high-level controller outputs a command in $+x$ direction. Similarly, it activates the posterior deltoid when the command is in $-x$ direction, triceps for commands in $+y$, and biceps in $-y$ directions.

Because of the information available to the synergy-based controller about the muscles' action, it results in smoother trajectories, and smaller error. Fig. 9 shows the comparison of the tracking error resulting from the two control methods during the deterministic motion trials. The tracking error of the synergy-based motor control is 39% lower (mean$\pm$std of the tracking error is 9.51$\pm$5.0 mm for the synergy controller, and 15.5$\pm$8.0 mm for the generic one).

IV. Discussion

The motor controller presented in this paper takes advantage of the known muscle actions in the task space,

in order to reduce the computational time and improve control performance. The generic controller, however, lacks such information, and as a result, suffers from poor control performance.

One can argue that the generic controller is similar to the synergy controller, by assuming that it has four orthogonal basis vectors ($\pm x$ and $\pm y$ directions), and decomposes the high-level controller command onto these directions. In this case, the controller suffers from the inaccurate representation of muscle actions. This lack of knowledge about the actual muscle actions is the major cause of occasional chattering during generic movement control.

To improve the proposed control performance, it is possible to include the arm dynamics in the calculations. All the information stored in the controller (in the form of the synergies and the basis set) belongs to isometric (stationary) conditions. As a result, the controller works best when the velocities are small. To account for the velocity dependent accelerations and inertia effects, the arm's equations of motion, as well as its model parameters are needed. Unfortunately, obtaining such information is time consuming, which will increase the proposed method's preparation time (about 10-15 minutes in our tests). In a clinical setting where therapy time is limited, one should weight the benefit of the improved control performance over the cost of the time spent on parameter identification.

The controller was tested on one healthy subject. However, there is no restriction in the controller architecture that prevents the method from being applied to impaired individuals. As long as the muscles are strong enough and respond to the stimulations, the controller can be applied. There are potential challenges with the clinical population that may impact the controller performance. For example, stroke patients may suffer from spastic muscles and day-to-day change in neuromuscular responses, which may adversely affect the FES controller.

A prominent problem with all the percutaneous FES systems is the inconsistent response of the muscles to the stimulations between subjects, and in a subject between different sessions. The presented method is no exception, and needs to be calibrated for each subject and therapy session. The strength of the method, on the other hand, is its inherent subject-specificity.

V. Conclusion

In this paper we presented the application of a motor control model as the feedback controller of functional electrical stimulation devices. As an advantage, this control method only required information in a task space, which disentangles the control task from many of the complexities of the musculoskeletal system, allowing for fast feedback control of the motions. As a result, our FES controller is the first of its kind that can be used to generate and control arbitrary movements in a task space. Our experimental results showed that the synergy-based motor control model can be used to effectively produce movement in the 2D task space.

References

[1] F. Previdi, "Identification of black-box nonlinear models for lower limb movement control using functional electrical stimulation," *Control Engineering Practice*, vol. 10, no. 1, pp. 91–99, 2002.

[2] A. Ajoudani and A. Erfanian, "A Neuro-Sliding Mode Control With Adaptive Modeling of Uncertainty for Control of Movement in Paralyzed Limbs Using Functional Electrical Stimulation.," *IEEE transactions on bio-medical engineering*, vol. 56, no. 7, pp. 1771–1780, 2009.

[3] N. Kirsch, N. Alibeji, and N. Sharma, "Nonlinear model predictive control of functional electrical stimulation," *Control Engineering Practice*, vol. 58, pp. 319–331, jan 2017.

[4] N. Kirsch, N. Alibeji, and N. Sharma, "Nonlinear Model Predictive Control of Functional Electrical Stimulation," in *Volume 2: Diagnostics and Detection; Drilling; Dynamics and Control of Wind Energy Systems; Energy Harvesting; Estimation and Identification; Flexible and Smart Structure Control; Fuels Cells/Energy Storage; Human Robot Interaction; HVAC Building Energy M*, p. V002T27A005, ASME, oct 2015.

[5] P. E. Crago, W. D. Memberg, M. K. Usey, M. W. Keith, R. F. Kirsch, G. J. Chapman, M. A. Katorgi, and E. J. Perreault, "An elbow extension neuroprosthesis for individuals with tetraplegia," *IEEE Transactions on Rehabilitation Engineering*, vol. 6, no. 1, pp. 1–6, 1998.

[6] H. Park and D. M. Durand, "Motion control of musculoskeletal systems with redundancy.," *Biological cybernetics*, vol. 99, pp. 503–16, dec 2008.

[7] C. T. Freeman, A.-M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, and E. Rogers, "A model of the upper extremity using FES for stroke rehabilitation.," *Journal of biomechanical engineering*, vol. 131, p. 031011, mar 2009.

[8] E. M. Schearer, Y.-W. Liao, E. J. Perreault, M. C. Tresch, W. D. Memberg, R. F. Kirsch, and K. M. Lynch, "Multi-Muscle FES Force Control of the Human Arm for Arbitrary Goals," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, pp. 654–663, may 2014.

[9] E. M. Schearer, Y.-W. Liao, E. J. Perreault, M. Tresch, W. D. Memberg, R. F. Kirsch, and K. M. Lynch, "Semiparametric Identification of Human Arm Dynamics for Flexible Control of a Functional Electrical Stimulation Neuroprosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. April, pp. 1–1, 2016.

[10] E. M. Schearer, Y.-W. Liao, E. J. Perreault, M. C. Tresch, W. D. Memberg, R. F. Kirsch, and K. M. Lynch, "Evaluation of a Semi-Parametric Model for High-Dimensional FES Control," *2015 7Th International Ieee/Embs Conference on Neural Engineering (Ner)*, pp. 304–307, 2015.

[11] D. Popovic and M. Popovic, "Tuning of a nonanalytical hierarchical control system for reaching with FES," *IEEE Transactions on Biomedical Engineering*, vol. 45, no. 2, pp. 203–212, 1998.

[12] M. Popovic and D. Popovic, "Cloning biological synergies improves control of elbow neuroprostheses: Simulating natural control of the extremities using inductive learning," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 1, pp. 74–81, 2001.

[13] R. Sharif Razavian, N. Mehrabi, and J. McPhee, "A model-based approach to predict muscle synergies using optimization: application to feedback control," *Frontiers in Computational Neuroscience*, vol. 9, no. October, pp. 1–13, 2015.

[14] R. Sharif Razavian and J. McPhee, "A motor control framework for the fast control of a 3D musculoskeletal arm motion using muscle synergy," in *The 4th Joint International Conference on Multibody System Dynamics*, 2016.

[15] M. Behringer, A. Franz, M. McCourt, and J. Mester, "Motor point map of upper body muscles," *European Journal of Applied Physiology*, vol. 114, no. 8, pp. 1605–1617, 2014.

[16] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in Neural Information Processing Systems*, vol. 13, no. 1, pp. 556–562, 2000.