# Neural Network Modeling of Electrically Stimulated Muscle Under Non-Isometric Conditions

Abbas Erfanian[1] and Patrick E. Crago[2]

[1]Dept. of Biomed. Engr., Faculty of Electrical Engr., Iran University of Science and Technology, Tehran 16844, IRAN

[2]Dept. of Biomed. Engr., Case Western Reserve University, Cleveland, OH 44106, USA

E-mail: erfanian@sun.iust.ac.ir

**Abstract—***This paper is concerned with developing a force-generating model of electrically stimulated muscle under non-isometric condition based upon artificial neural networks. The stochastic stimulation patterns were applied to the sciatic nerve that innervates the soleus muscle of adult cats using a monopolar spiral cuff electrode. During stimulation, the muscle length is perturbed randomly about a nominal length. We employed the multilayer perceptron (MLP) with back-propagation learning algorithm and Radial Basis Function (RBF) network with stochastic gradient learning rule for muscle modeling, where the stimulation signal, muscle length, velocity of length perturbation, and past measured or predicted force constitute the input of the neural model and the predicted force is the output. The results show that time-varying neural networks would be able to track the muscle force with prediction error 4.1% after the convergence. In addition, when muscle force measurements are not available, the muscle force can be predicted using a time-invariant neural network with mean prediction error 12.7%, 15.2%, and 16.7% for muscle length perturbations of 6 mm, 3 mm, and 1.5 mm, respectively.*

**Keywords:** Muscle Modeling, Neural Network**,** Functional Neuromuscular Stimulation.

## 1. INTRODUCTION

Although a large number of mathematical models of electrically stimulated muscle have appeared (as reviewed in [1]), these efforts have not resulted in predictive models of muscle force that can be used for real-time control in neuroprostheses during non-isometric operations. *There is no available model that accurately predicts muscle force under non-isometric conditions.* In [2], a nonlinear non-isometric Hill-based model of stimulated muscle was presented. This muscle model was constructed as a combination of different independent blocks (i.e., activation dynamics, force-length and force-velocity relations, and series elastic element). The model assumes that the force-length and the force-velocity relations are uncoupled from the activation dynamics. However, some studies suggest that the shapes of the active force-length and the active force-velocity curves change with the level of the activation [3]. Moreover, the "active state" block of the Hill-type model has no physical interpretation. In [4], a Hill-base muscle model has been developed while the activation dynamics was coupled to the velocity. It is shown that introducing the activation-velocity coupling improved the generality over the uncoupled Hill-based model.

In this work, we develop a non-isometric systematic-based muscle model based upon neural networks. It is demonstrated that the neural network model of electrically stimulated muscle is capable of predicting the muscle force accurately under a wide range of non-isometric operations.

## 2. NEURAL NETWORK MODELS

Recently, universal approximation capabilities of multi-layered perceptron type networks with different types of activation functions have been reported by several authors [5]. The universal approximation capabilities of neural models suggest the possibility of using neural networks to identify nonlinear dynamical systems. Moreover, radial basis function (RBF) neural networks can offer approximation capabilities similar to those of the multi-layer perceptrons [6].

In this work, we employ a multilayer feedforward networks with back-propagation learning rule and radial basis function (RBF) neural network with gradient descent procedure for muscle modeling.

*A. Radial Basis Function (RBF) Neural Network*

The architecture of RBF networks is simple and consists of one hidden layer. The block diagram of a version of RBF network is shown in Fig. 1. The hidden layer is composed of a number of kernel nodes with kernel activation functions. The output of the network is simply a weighted linear summation of the kernel functions:

$$f(x) = \sum_{i=1}^{M} w_i \,.\, k(x) = \sum_{i=1}^{M} w_i \,.\, G\left[\frac{\|x - c_i\|}{\sigma_i}\right] \qquad (1)$$

where $x \in R^n$ is the input vector, $M$ is the number of kernel nodes in the hidden layer, $w_i$ $(1 \le i \le M)$ is the vector of weights from the *i*-th kernel node to the output nodes, $\|\bullet\|$ is Euclidean distance, and $\boldsymbol{k}$ is a radial symmetric kernel function. A Gaussian function is normally chosen as kernel function.

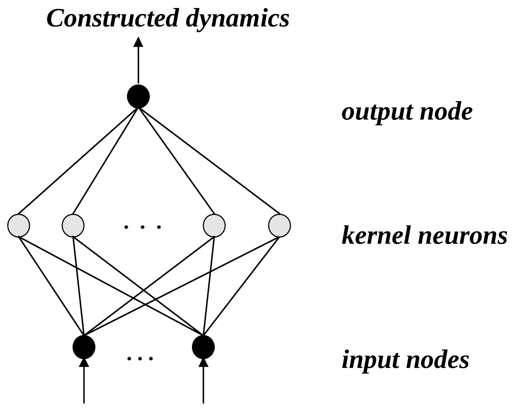


Fig. 1. Architecture of a radial basis function network

The vectors $\boldsymbol{C_i}$ represent the locations of the kernel functions in $\boldsymbol{R^n}$ . Finally, $\boldsymbol{\sigma_i}$ is the smoothing factor or kernel bandwidth of the *i*-th kernel node. The degree of localization can be adjusted by kernel bandwidth. Now, the key question is how to determine the parameters of RBF network appropriately for the purpose of functional approximation. One approach is that all the free parameters of the network undergo the gradient descent procedure, i.e. least-mean-square (LMS). The summary of algorithm is as follows [7]:

$$y(n) = \sum_{i=1}^{M} w_i(n) \cdot k\left(x(n); c_i(n); \sigma_i^2(n)\right)$$

$$e(n) = d(n) - y(n)$$

$$w_i(n+1) = w_i(n) + \mu \cdot e(n) \cdot k\left(x(n); c_i(n); \sigma_i^2(n)\right)$$

$$c_i(n+1) = c_i(n) + 2\mu \times e(n) \times w_i(n) \times k\left(x(n); c_i(n); \sigma_i^2(n)\right) \times \frac{x(n) - c_i(n)}{\sigma_i^2(n)}$$

$$\sigma_i^2(n+1) = \sigma_i^2(n) + \mu \times e(n) \times w_i(n) \times k\left(x(n); c_i(n); \sigma_i^2(n)\right) \times \frac{\|x(n) - c_i(n)\|^2}{\sigma_i^2(n)}$$

$$k\left(x(n); c_i(n); \sigma_i^2(n)\right) = \exp\left(-\frac{1}{\sigma_i^2(n)} \|x(n) - c_i(n)\|^2\right)$$

where $d$ is the desired values and $\mu$ is a *learning-rate* parameter that lies in the range $0 < \mu < 1$.

*B. Time-Varying and Time-Invariant Neural Network*

Two modes of network operation are of interest: time-varying and time-invariant neural network. In time-varying mode, the leaning process never stops and continuously adapts the free parameters of the network to variations in the incoming signals. In contrast, in time-invariant mode, once the training process has completed, the synaptic weights of the network freeze their values thereafter.

## 3. Application of the Neural Networks to the Muscle Modeling

The input-output behavior of electrically stimulated muscle is highly nonlinear and time-varying. A popular mathematical modeling formalism for discrete-time nonlinear dynamical system is the following nonlinear difference equation:

$$y(t) = f\left[y(t-1),\ y(t-1), \ldots, y(t-m),\ u(t-1),\ u(t-2), \ldots, u(t-n)\right]$$

where $[u(t), y(t)]$ represents the input-output pair of the system at time *t*. The output at time *t* depends both on its past *m* values as well as the past *n* values of the input.

During non-isometric contraction, the muscle can be described as a three-input single output system. Stimulation signals as well as muscle length and contraction velocity constitute the input of the muscle. Therefore, the nonlinear difference equation for electrically stimulated muscle can be written as:

$$\begin{aligned} y(t) = f[&y(t-1), y(t-1), \ldots, y(t-m), \\ &s(t-1), s(t-1), \ldots, s(t-n), \\ &l(t-1), l(t-2), \ldots, l(t-p), \\ &v(t-1), v(t-2), \ldots, v(t-q)]. \end{aligned}$$

where $y$ denotes the muscle force, $s$ is the stimulation signals, $l$ is the muscle length, and $v$ represents the velocity of contraction. From the universal approximation capabilities of the neural networks, it follows that a neural network can be constructed to approximate the $\boldsymbol{f}$.

In this work, we apply both time-varying and time-invariant neural networks to the muscle force prediction. In time-varying neural network, muscle force prediction is based on the past measured muscle force while the learning process is going on. In time-invariant neural network, muscle force prediction is based on the past predicted muscle force.

## 4. Experiments

The experiments were performed on adult cats. The stochastic stimulation patterns were applied to the sciatic nerve that innervates the soleus muscle using a monopolar spiral cuff electrode. During stimulation, the muscle length is perturbed randomly about a nominal length (8 mm short of maximum). Three different levels of length perturbation were used (6, 3, and 1.5 mm). Fig.

1 shows a typical stimulation sequence, length perturbation, and corresponding muscle output force.

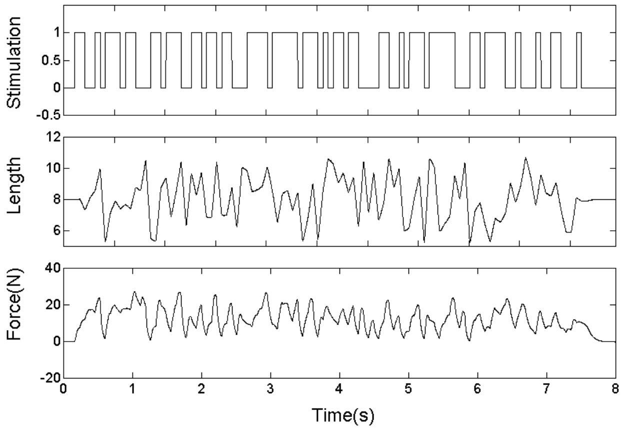


Fig. 2. Stochastic stimulation pattern (top), muscle length perturbation (middle), measured muscle force (bottom).

## 5. RESULTS

### *A. Back-Propagation Neural Network*

A multi-layer perceptron (MLP) network with two hidden layers each containing 20 sigmoid units and one linear output is considered. **Table 1** summarizes the average force prediction errors obtained for time-varying neural network model after 5000 training epochs for different values of ($m$, $n$, $p$, $q$). The networks were trained with data obtained during maximum length perturbation (6 mm) and were evaluated with data obtained during all three different levels of length perturbation. Model order ($m$, $n$, $p$, $q$) determination was accomplished by comparing the average predication-error criterion obtained, for different model orders. We found that the prediction error was minimized by model order $(m=1,\ n=1,\ p=3,\ q=3)$.

The results show that time-varying neural networks would be able to track the muscle force with prediction error 4.1% after the convergence. The interesting result is that if the level of muscle length perturbation is reduced to 3 mm or 1.5 mm, the prediction error decreases to 2.8 or 2.1% respectively. **Fig. 3** shows the muscle force prediction, obtained using time-varying network. It is observed that the output of the network overlapped with the muscle force.

When muscle force measurements are not available, the muscle force can be predicted using a time-invariant neural network. **Table 2** summarizes the results of muscle force prediction. It is observed that the structure with no output feedback ($m$=0) provides more accurate muscle force prediction than that with output feedback. **Fig. 4** shows the muscle force prediction during different muscle length perturbations, using time-invariant neural network.

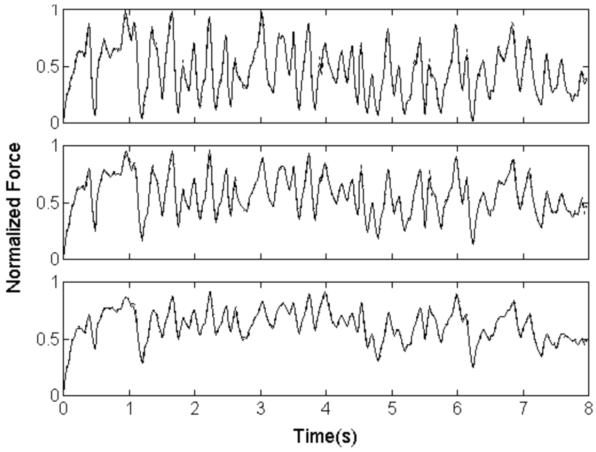


Fig. 3. The measured (solid line) and predicted (dotted line) muscle force obtained using time-varying MLP network during muscle length perturbation 6 mm (top), 3 mm (middle), and 1.5 mm (bottom).

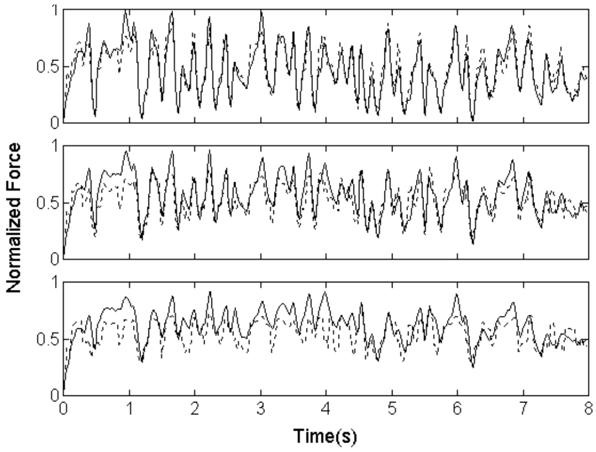


Fig. 4. The measured (solid line) and predicted (dotted line) muscle force obtained using time-invariant MLP network during muscle length perturbation 6 mm (top), 3 mm (middle), and 1.5 mm (bottom).

### *B. Radial Basis Function (RBF) Neural Network*

Table 3 shows the muscle force prediction error obtained using time-varying RBF network with different number of hidden units. Form the results of Table 1, the model order is chosen to have the value $m$=1, $n$=1, $p$=3, and $q$=3. It is found that increasing the complexity of the network does not necessary improve the performance of the model. Table 4 summarizes the results for time-invariant RBF network.

## 6. CONCLUSION

The results of this work show that neural network modeling of electrically stimulated muscle provides more accurate muscle force prediction than Hill-base model. Moreover, the performance of MLP network in the muscle force prediction is almost the same as that of RBF network. However, it should be noted that the RBF network does not involve error back-propagation.

## REFERENCES

[1] Zahalak, G.I. (1992). An overview of muscle modeling, in R.B. Stein, P.H. Peckham, & D.P. Popovic (Eds.), *Neural Prostheses: Replacing Motor Function After Disease or Disability* (pp. 17-57). New York, Oxford University Press.

[2] Durfee, W.K., & Palmer, K.L. (1994). Estimation of force-activation, force-length, and force-velocity properties in isolated, electrically stimulated muscle. *IEEE Trans. Biomed. Eng.*, **41**, 205-216.

[3] Joyce, G.C., Rack, P.M.H., & Westbury, D.R. (1969). The mechanical properties of cat soleus muscle during controlled lengthening and shortening movements. *J. Physiol.* **204**, 461-474.

[4] Shue, G., Crago, P. E., & Chizeck, H. J., (1995), Muscle-joint models incorporating activation dynamics, moment-angle, and moment-velocity properties, *IEEE Trans. Biomed. Eng.*, **42**, 212-223.

[5] Scarsell, F. & Tsoi, A.C. (1998). Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, **11**, pp. 15-37.

[6] Park, J., & Sandberg, I.W. (1993). Approximation and radial-basis-function networks. *Neural Computation*, **5**, 305-316.

[7] Haykin, S. (1999). *Neural Networks* (pp. 256-317). Englewood Cliffs NJ: Prentice Hall.


**TABLE 1**

AVERAGE MUSCLE FORCE PREDICTION ERROR USING TIME-VARYING MLP NETWORK

| ***(m,n,p,q)*** | **(1,1,1,1)** | **(1,1,2,2)** | **(1,1,3,3)** | **(1,1,4,4)** | **(1,2,3,3)** |
|---|---|---|---|---|---|
| ***Length Perturbation*: 6** | 5.4% | 4.2% | 4.1% | 4.0% | 4.2% |
| ***Length Perturbation*: 3** | 3.5% | 3.0% | 2.8% | 2.9% | 3.0% |
| ***Length Perturbation*: 1.5** | 2.4% | 2.2% | 2.1% | 2.2% | 2.4% |
| ***mean*** | 3.7% | 3.1% | 3.0% | 3.0% | 3.2% |

**TABLE 2**

AVERAGE MUSCLE FORCE PREDICTION ERROR USING TIME-INVARIANT MLP NETWORK

| ***(m,n,p,q)*** | **(1,1,1,1)** | **(1,1,2,2)** | **(1,1,3,3)** | **(1,1,4,4)** | **(0,1,2,2)** | **(0,1,3,3)** | **(0,1,4,4)** | **(0,2,3,3)** |
|---|---|---|---|---|---|---|---|---|
| ***Length Perturbation*: 6** | 21.0% | 15.0% | 27.0% | 14.6% | 16.5% | 13.6% | 13.2% | 12.7% |
| ***Length Perturbation*: 3** | 20.1% | 19.1% | 31.5% | 19.2% | 15.4% | 15.9% | 17.9% | 15.2% |
| ***Length Perturbation*: 1.5** | 21.4% | 18.6% | 30.9% | 21.8% | 16.0% | 17.4% | 19.5% | 16.7% |
| ***mean*** | 20.8% | 17.6% | 29.8% | 18.5% | 16.9% | 15.6% | 16.9% | 14.9% |

**TABLE 3**

AVERAGE MUSCLE FORCE PREDICTION ERROR USING TIME-VARYING RBF NETWORK

THE MODEL ORDER HAS THE VALUE $(m=1,\ n=1,\ p=3,\ q=3)$.

| ***Hidden Neurons*** | **20** | **40** | **60** | **80** | **100** |
|---|---|---|---|---|---|
| ***Length Perturbation*: 6** | 4.2% | 4.1% | 4.2% | 4.3% | 4.8% |
| ***Length Perturbation*: 3** | 2.8% | 2.6% | 2.8% | 2.7% | 2.7% |
| ***Length Perturbation*: 1.5** | 2.0% | 1.9% | 2.0% | 2.0% | 1.9% |
| ***mean*** | 3.0% | 2.9% | 3.0% | 3.0% | 3.1% |

**TABLE 4**

AVERAGE MUSCLE FORCE PREDICTION ERROR USING TIME-INVARIANT RBF NETWORK

THE MODEL ORDER HAS THE VALUE $(m=0,\ n=2,\ p=3,\ q=3)$.

| ***Hidden Neurons*** | **20** | **40** | **60** | **80** | **100** |
|---|---|---|---|---|---|
| ***Length Perturbation*: 6** | 16.4% | 16.4% | 16.0% | 18.1% | 17.1% |
| ***Length Perturbation*: 3** | 15.4% | 15.9% | 14.8% | 13.9% | 15.3% |
| ***Length Perturbation*: 1.5** | 18.4% | 19.9% | 17.1% | 14.3% | 15.8% |
| ***mean*** | 16.7% | 17.4% | 16.0% | 15.4% | 16.1% |