

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,400

Open access books available

173,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Emg Signal Noise Removal Using Neural Networks

Vijay R. Mankar

*Head of Department, Govt. Polytechnic, Amravati
India*

1. Introduction

The bioelectric potentials associated with muscle activity constitute the Electromyogram, abbreviated as EMG. These potentials may be measured at the surface of the body near a muscle of interest or directly from the muscle by penetrating the skin with needle electrodes. Since most EMG measurements are intended to obtain an indication of the amount of activity of a given muscle, or group of muscles, rather than that of an individual muscle fiber, the pattern is usually a summation of the individual action potentials from the fibers constituting the muscle or muscles being measured. EMG electrodes pick up potentials from all muscles within the range of the electrodes, hence potentials from nearby large muscles may interfere with attempts to measure the EMG from smaller muscles, even though the electrodes are placed directly over the small muscles. Where this is a problem, needle electrodes inserted directly into the muscle are required. [Bronzino, J.D. (ed), 1995]

The action potential of a given muscle (or nerve fiber) has a fixed magnitude, regardless of the intensity of the stimulus that generates the response. Thus, in a muscle, the intensity with which the muscle acts does not increase the net height of the action potential pulse but does increase the rate with which each muscle fiber fires and the number of fibers that are activated at any given time. The amplitude of the measured EMG waveform is the instantaneous sum of all the action potentials generated at any given time. Because these action potentials occur in both positive and negative polarities at a given pair of electrodes, they sometimes add and sometimes cancel. Thus, the EMG waveform appears very much like a random-noise waveform, with the energy of the signal a function of the amount of muscle activity and electrode placement. Typical EMG waveforms are shown in Figure 1.

2. Emg measurements

Although action potentials from individual muscle fibers can be recorded under special conditions, it is the electrical activity of the entire muscle that is of primary interest. In this case, the signal is a summation of all the action potentials within the range of the electrodes, each weighted by its distance from the electrodes. Since the overall strength of muscular contraction depends on the number of fibers energized and the time of contraction, there is a correlation between the overall amount of EMG activity for the whole muscle and the strength of muscular contraction. In fact, under certain conditions of isometric contraction, the voltage-time integral of the EMG signal has a linear relationship to the isometric

voluntary tension in a muscle. There are also characteristic EMG patterns associated with special conditions, such as fatigue and tremor.

The EMG potentials from a muscle or group of muscles produce a noiselike waveform that varies in amplitude with the amount of muscular activity. Peak amplitudes vary from 25 μV to about 5 mV, depending on the location of the measuring electrodes with respect to the muscle and the activity of the muscle. A frequency response from about 5 Hz to well over 15000 Hz is required for faithful reproduction. [Childers, D.G., J.G. Webster, 1988]

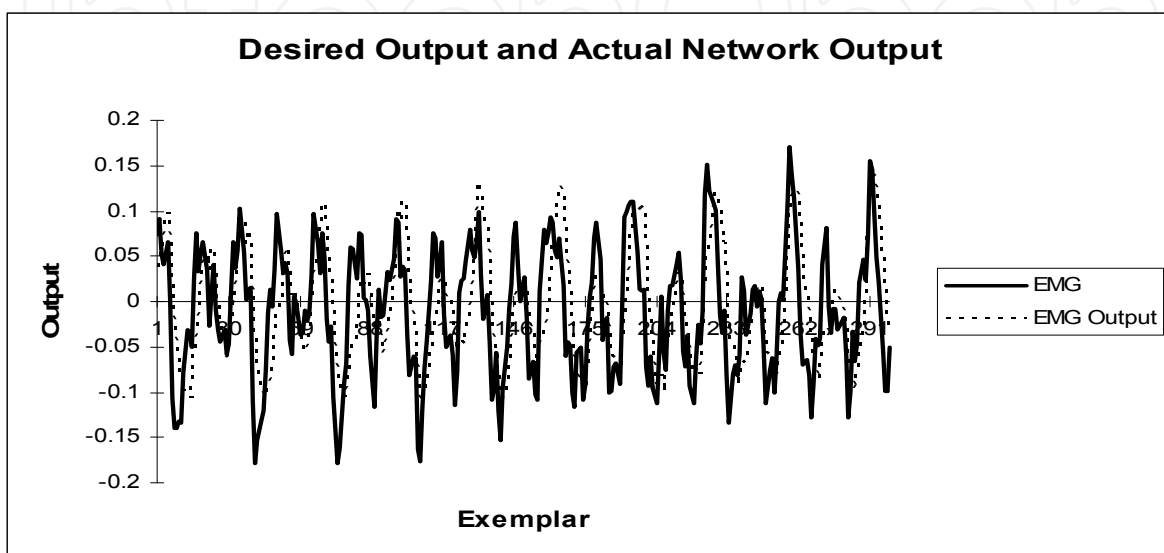


Fig. 1. Typical electromyogram waveform showing EMG signal and EMG output signal.

The amplifier for EMG measurements, like that for ECG and EEG, must have high gain, high input impedance and a differential input with good common-mode rejection. However, the EMG amplifier must accommodate the higher frequency band. In many commercial electromyographs, the upper-frequency response can be varied by use of switchable lowpass filters. [Cromwell, L., et al., 1980, John G. Webster, 2001] Unlike ECG or EEG equipment, the typical electromyograph has an oscilloscope readout instead of a graphic pen recorder. The reason is the higher frequency response required. Sometimes a storage cathode-ray tube is provided for retention of data, or an oscilloscope camera is used to obtain a permanent visual record of data from the oscilloscope screen.

The EMG signal can be quantified in several ways. The simplest method is measurement of the amplitude alone. In this case, the maximum amplitude achieved for a given type of muscle activity is recorded. Unfortunately the amplitude is only a rough indication of the amount of muscle activity and is dependent on the location of the measuring electrodes with respect to the muscle. Surface, needle, and fine-wire electrodes are all used for different types of EMG measurement. Surface electrodes are generally used where gross indication are suitable, but where localized measurement of specific muscles is required, needle or wire electrodes that penetrate the skin and contact the muscle to be measured are needed. As in neuronal firing measurements, both unipolar and bipolar measurements of EMG are used. [Brush, L.C., Cohen, B.J., 1995]

Another method of quantifying EMG is a count of the number of spikes or, in some cases, zero crossings, that occur over a given time interval. A modification of this method is a count of the number of times a given amplitude threshold is exceeded. Although these

counts vary with the amount of muscle activity, they do not provide an accurate means of quantification, for the measured waveform is a summation of a large number of action potentials that cannot be distinguished individually.

The most meaningful method of quantifying the EMG utilizes the time integral of the EMG waveform. With this technique, the integrated value of the EMG over a given time interval, such as 0.1 second, is measured and recorded or plotted. As indicated above, this time integral has a linear relationship to the tension of a muscle under certain conditions of isometric contraction, as well as a relationship to the activity of a muscle under isotonic contraction. As with the amplitude measurement, the integrated EMG is greatly affected by electrode placement, but with a given electrode location, these values provide a good indication of muscle activity. [Tompkins, W. J., 1999, Cromwell L. et al., 2004]

In another technique that is sometimes used in research, the EMG signal is rectified and filtered to produce a voltage that follows the envelope or contour of the EMG. This envelope, which is related to the activity of the muscle, has a much lower frequency content and can be recorded on a pen recorder, frequently in conjunction with some measurement of the movement of a limb or the force of the muscle activity.

3. Sources of errors

Errors can occur in a multitude of ways. These errors need to be considered, although may not be always present simultaneously:

- Errors due to tolerance of electronic components.
- Mechanical errors in meter movements.
- Component errors due to drift or temperature variation.
- Errors due to poor frequency response.
- In certain types of instruments, errors due to change in atmospheric pressure or temperature.
- Reading errors due to parallax, inadequate illumination, or excessively wide ink traces on a pen recording.

Two additional sources of error should not be overlooked. The first concerns correct instrument zeroing. Another source of error is the effect of the instrument on the parameter to be measured, and vice versa. This is especially true in measurements in living organisms. These errors lead to the noise in a system.

All semiconductor junctions generate noise, which limits the detection of small signals. Op Amps have transistor input junctions, which generate both noise-voltage sources and noise-current sources as indicated in Figure 2. For low source impedance, only the noise voltage v_n is important; it is large compared with the $i_n R$ drop caused by the current noise i_n . The noise is random, but the amplitude varies with frequency. For example, at low frequencies the noise power density varies as $1/f$ (flicker noise), so a large amount of noise is present at low frequencies. At the midfrequencies, the noise is lower and can be specified in rms units of $V \cdot \text{Hz}^{-1/2}$. In addition, some silicon planar-diffused bipolar integrated-circuit op amps exhibit bursts of noise. [Geddes, L.A., L.E. Baker, 1989] The noise currents flow through the external equivalent resistances so that total rms noise voltage is

$$v_t = \{[v_n^2 + (i_n R_1)^2 + (i_n R_2)^2 + 4kTR_1 + 4kTR_2]BW\}^{1/2}$$

where R_1 and R_2 = equivalent source resistances

v_n = mean value of the rms noise voltage, in $V \cdot Hz^{-1/2}$, across the frequency range of interest,
 BW = noise bandwidth, Hz.

i_n = mean value of the rms noise current, in $A \cdot Hz^{-1/2}$, across the frequency range of interest,
 k = Boltzmann's constant, T = temperature, K,

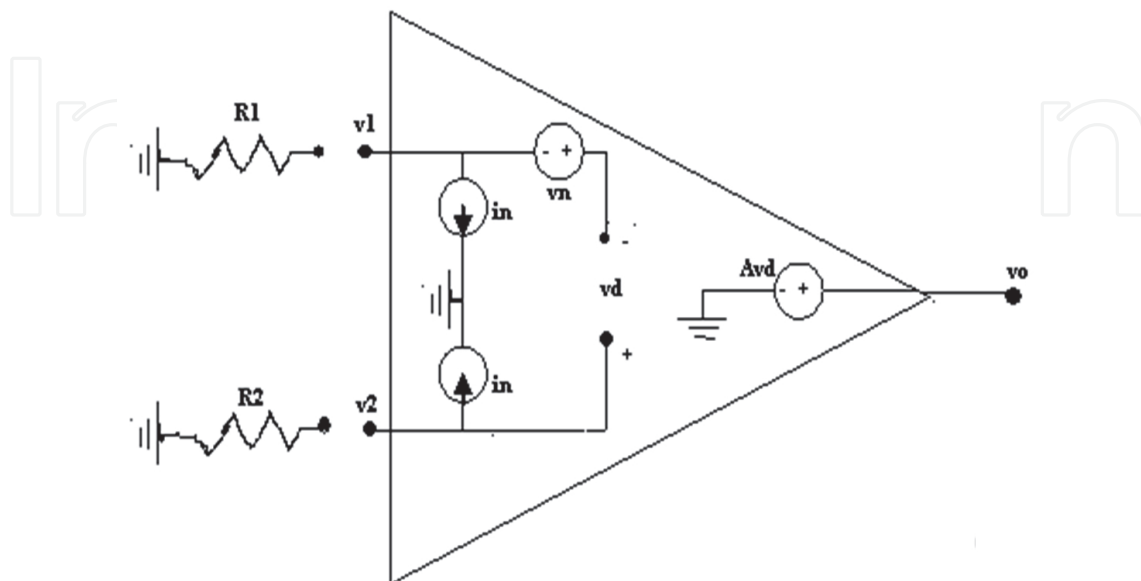


Fig. 2. Noise sources in an Op Amp

Signal enhancement in noisy environment is a challenge problem since decades. Noise is added to a signal under measurement almost in an uncontrolled manner. Signal-processing systems pick-up "unwanted" noise signal alongwith desired signal. These noise signals result in performance degradation of those systems. Noise classification can be used to reduce the effect of environmental noises on signal processing tasks. NN's are proposed as alternative optimization techniques to handle problems in signal processing. Prior a neural network maps each input feature vector into output vector, it must have first learnt the classes of feature vectors through a process that partitions a set of feature vectors. This is called discrimination or classification, which involve machines learning.

4. Importance of neural networks

A Neural Network is a massively parallel distributed processor made of simple processing element having natural propensity for storing experimental knowledge and making it available for use. It has the ability to acquire the knowledge from its environment through a learning process and to store acquired knowledge through inter-neuron connection strengths (synaptic weights). The procedure used to perform the learning process is called a learning algorithm, the function of which is to modify the synaptic weights of the network in an orderly fashion to attain a desirable design objective.

The use of neural network offers the useful properties and capabilities like nonlinearity, input/output mapping, adaptivity, evidential response, contextual information, fault tolerance, VLSI implementability, uniformity of analysis and design and neurobiological analogy. Neural networks, because of their massively parallel nature, can perform computation at a very high rate. Neural networks can adapt to a change in the data and

learn the characteristics of input signals due to their adaptive nature. Neural networks can also perform functional approximation and signal filtering operation because of their nonlinear nature. Hence, neural networks are widely used for problem solving in engineering that are difficult for conventional computers or human beings [Haykin, S, 1986].

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements. Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown below.

The network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in this supervised learning, to train a network as shown in figure 3.

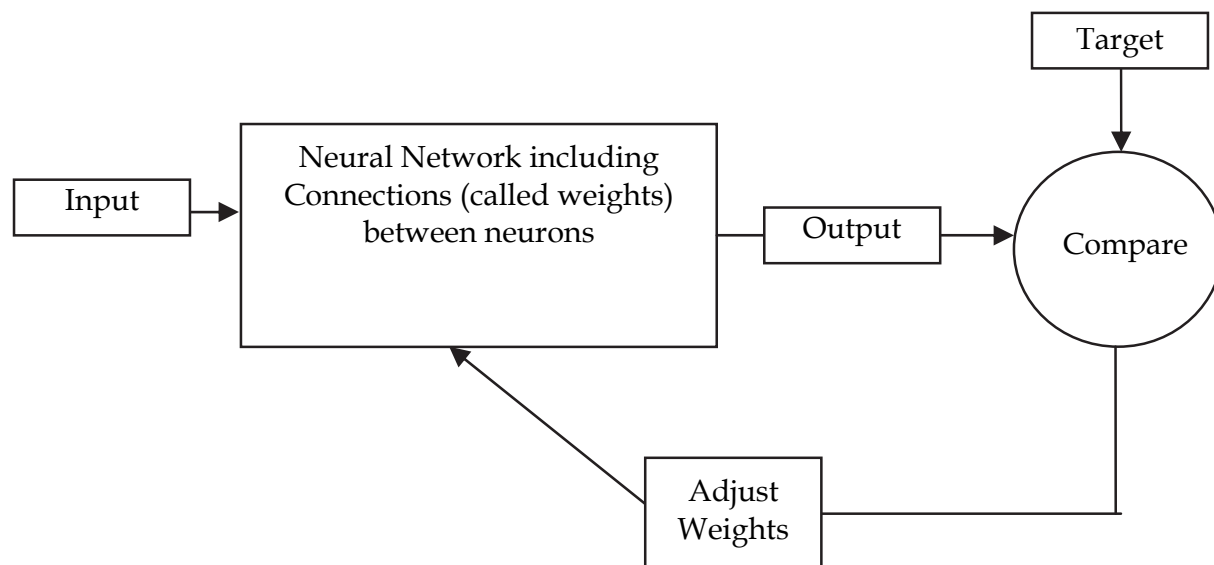


Fig. 3. Neural Network in Supervised Learning Mode.

Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as “on line” or “adaptive” training. Neural networks have been trained to perform complex functions in various fields of application including pattern recognition, identification, classification, speech, vision, control systems and signal processing.

The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make. The field of neural networks has a history of some five decades but has found solid applications only in the past fifteen years, and the field is still developing rapidly. Neural network will

be a useful tool for industry, education and research, a tool that will help users find what works and what doesn't, and a tool that will help develop and extend the field of neural networks.

5. Neural network approach

There are numerous real life situations where the exactness of the measurements is required. In Biomedical applications, due to complicated situations, the measurements are noisy. Neural Networks can be used to obtain reasonably good accuracy in removal of noise or elegantly filtering out the desired signals. At a high level, the filtering problem is a special class of function approximation problem in which the function values are represented using time series. A time series is a sequence of values measured over time in the discrete or continuous time units. Literature survey revealed that the Neural Networks can also be effectively used for solving the nonlinear multivariable regression problem. [Xue, Q.Z., et. al., 1992, Richard D. de Veaux, et. al., 1998] Also, there is a wide scope for an exact neural network with the performance indices approaching to their ideal values, i.e. $MSE = 0$, and correlation coefficient $r = 1$ [J.C. Principe, et. al. , 2000].

Signal filtering from present observations is a basic signal processing operation by use of filters. Conventional parametric approaches to this problem involve mathematical modeling of the signal characteristics, which is then used to accomplish the filtering. In a general case, this is relatively a complex task containing many steps for instance model hypothesis, identification and estimation of model parameters and their verification. However, using a Neural Network, the modeling phase can be bypassed and nonlinear and nonparametric signal filtering can be performed. As the thresholds of all neurons are set to zeros, unknown variables for one step ahead filtering are only the connection weights between the output neurons and the j^{th} neuron in the second layer, which can be trained by available sample set [Widrow, B, et al., 1975].

In the last decade, NN, have given rise to high expectations for model free statistical estimation from a finite number of sample. The goal of predictive learning is to estimate or learn an unknown functional mapping between the input variables and the output variables, from the training set of known input output samples. The mapping is typically implemented as a computational procedure in software. Once the mapping is obtained from the training data, it can be used for predicting the output value, given only the values of the input variables [Khandpur R.S., 2001]. In the research work referred, the several techniques for noise removal from biomedical signals like EMG, [Abdelhafid Zeghib, et. al.,2007, Umut Gundogdu, et. al., 2006] EEG, [Mercedes Cabrerizo, et. al.,2007, David Coufal, 2005], and ECG [Mahesh S. Chavan, et. al.,2006, Hafizah Husain, Lai Len Fatt, 2007] using signal processing techniques [Chunshien Li, 2006, M.Areziki, et. al.,2007,] and neural networks have been presented.

5.1 Recurrent networks

Recurrent networks are the proper neural network to be selected when identifying a nonlinear dynamical process. Such networks are attractive with their capabilities to perform highly nonlinear dynamic mapping and their ability to store information for later use. Moreover, they can deal with time-varying input or output through their own natural

temporal operation. There are two types of recurrent neural networks: fully recurrent neural networks and partially recurrent neural networks. Many learning algorithms have been developed. Partially recurrent networks are back-propagation networks with proper feedback links. It allows the network to remember cues from the recent past. In these architectures, the nodes receiving feedback signals are context units. According to the kind of feedback links, two major models of partially recurrent networks are encountered as described below. [Ezin C. Eugène, 2008]

Fully Recurrent Networks feed back the hidden layer to itself. Partially recurrent networks start with a fully recurrent net and add a feedforward connection that bypasses the recurrency, effectively treating the recurrent part as a state memory. These recurrent networks can have an infinite memory depth and thus find relationships through time as well as through the instantaneous input space. Most real-world data contains information in its time structure. Recurrent networks are the state of the art in nonlinear time series prediction, system identification, and temporal pattern classification.

5.2 Jordan network

This network model is realized in adding recurrent links from the network's output to a set of context units C_i , of a context layer and from the context units to themselves. Context units copy the activations of output node from the previous time step through the feedback links with unit weights. Their activations are governed by the differential equation

$$C'_i(t) = -\alpha C_i(t) + y_i(t)$$

where the y_i 's are the activations of the output nodes and α is the strength of the self-connections.

Despite the use of the Jordan sequential network to recognize and distinguish different input sequences with sequences of increasing length, this model of network encounters difficulties in discriminating on the basis of the first cues presented.

5.3 Multilayer perceptron approach

Multilayer Perceptrons (MLPs) are layered feedforward networks typically trained with static backpropagation. These networks have found their ways into countless applications requiring static pattern classification. Their main advantage is that they are easy to use, and that they can approximate any input/output map. The key disadvantages are that they train slowly, and require lots of training data (typically three times more training samples than network weights). The multilayer perceptron (MLP) is one of the most widely implemented neural network topologies. For static pattern classification, the MLP with two hidden layers is a universal pattern classifier. In other words, the discriminant functions can take any shape, as required by the input data clusters. When the weights are properly normalized and the output classes are normalized to 0/1, the MLP achieves the performance, which is optimal from a classification point of view. In terms of mapping abilities, the MLP is believed to be capable of approximating arbitrary functions, which is important in the study of nonlinear dynamics, and other function mapping problems.

MLPs are normally trained with the backpropagation algorithm. The LMS learning algorithm proposed by Widrow can not be extended to hidden PEs, since the desired signal is not known. The backpropagation rule propagates the errors through the network and

allows adaptation of the hidden PEs. Two important characteristics of the multilayer perceptron are its nonlinear processing elements (PEs) which have a nonlinearity that must be smooth and their massive interconnectivity i.e. any element of a given layer feeds all the elements of the next layer. The multilayer perceptron is trained with error correction learning, which means that the desired response for the system must be known.

Jordan (1986) described the first MLP architecture with recurrent connections for sequence generation. The input layer has two parts: plan units representing external input and the identity of the sequence and state units that receive one-to-one projections from the output layer, forming decay trace STM. After a sequence is stored into the network by back propagation training, it can be generated by an external input representing the identity of the sequence. This input activates the first component of the sequence in the output layer. This component feeds back to the input layer and, together with the external input, activates the second component, and so on. A particular component of a sequence is generated by the part of the sequence prior to the component, earlier components having lesser roles due to exponential decay. Elman (1990) later modified Jordan's architecture by having the hidden layer connect to a part of the input layer, called the context layer. The context layer simply duplicates the activation of the hidden layer in the previous time step. Elman used this architecture to learn a set of individual sequences satisfying a syntactic description, and found that the network exhibits a kind of syntax recognition. This result suggests a way of learning high-level structures, such as natural language grammar.

5.4 Elman neural network

Elman Neural Network (ENN) is a type of partial recurrent neural network, which consists of two-layer back propagation networks with an additional feedback connection from the output of the hidden layer to its input layer. The advantage of this feedback path is that it allows ENN to recognize and generate temporal patterns and spatial patterns. This means that after training, interrelations between the current input and internal states are processed to produce the output and to represent the relevant past information in the internal states. As a result, the ENN has been widely used in various fields from a temporal version of the Exclusive-OR function to the discovery of syntactic or semantic categories in natural language data. However, since ENN often uses back propagation (BP) to deal with the various signals, it has proved to be suffering from a sub-optimal solution problem. At the same time, for the ENN, it is less able to find the most appropriate weights for hidden neurons and often get into the sub-optimal areas because the error gradient is approximated. [Yves St-Amant, et al., 1998]

In the Elman neural network, after the hidden units are calculated, their values are used to compute the output of the network and are also all stored as "extra inputs" (called context unit) to be used when the next time the network is operated. Thus, the recurrent contexts provide a weighted sum of the previous values of the hidden units as input to the hidden units. The activations are copied from hidden layer to context layer on a one for one basis, with fixed weight of 1.0 ($w=1.0$). The forward connection weight is trained between hidden units and context units as well as other weights. [Edward A. Clancy, 1995] Both the Jordan and Elman networks have fixed feedback parameters and there is no recurrence in the input-output path. These networks can be trained approximately with straight Back propagation. Elman's context layer receives input from the hidden layer, while Jordan's context layer receives input from the output as shown in figure 4.

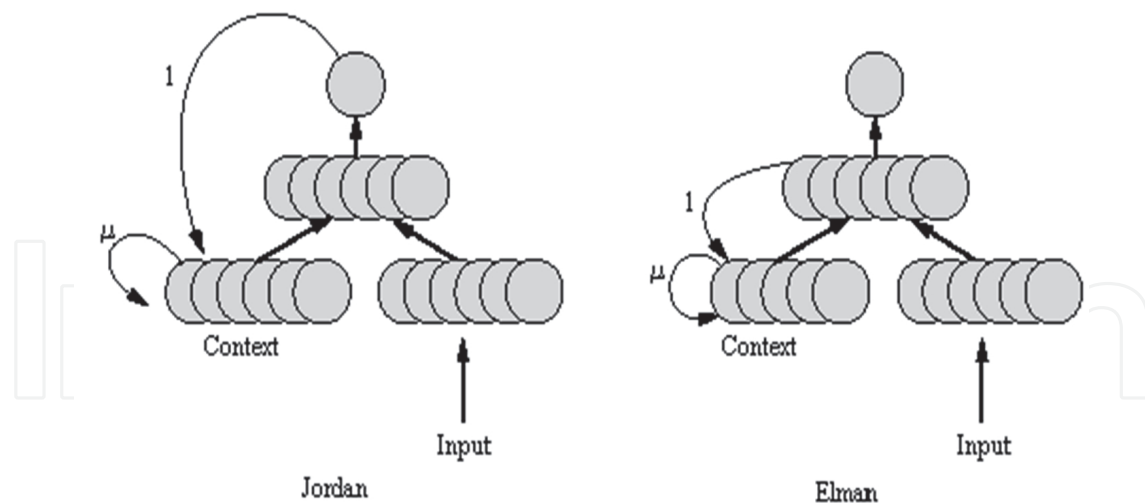


Fig. 4. Jordan and Elman network.

5.5 Generalized feedforward networks

Generalized feedforward networks are a generalization of the MLPs such that connections can jump over one or more layers. In theory, a MLP can solve any problem that a generalized feedforward network can solve. In practice, however, generalized feedforward networks often solve the problem much more efficiently. A classic example of this is the two spiral problem. Without describing the problem, it suffices to say that a standard MLP requires hundreds of times more training epochs than the generalized feedforward network containing the same number of processing elements.

5.6 Modular feedforward Networks

Modular Feedforward Networks are a special class of MLP. These networks process their input using several parallel MLPs, and then recombine the results. This tends to create some structure within the topology, which will foster specialization of function in each sub-module. In contrast to the MLP, modular networks do not have full interconnectivity between their layers. Therefore, a smaller number of weights are required for the same size network (i.e. the same number of PEs). This tends to speed up training times and reduce the number of required training exemplars. There are many ways to segment a MLP into modules. It is unclear how to best design the modular topology based on the data. There are no guarantees that each module is specializing its training on a unique portion of the data.

5.7 Principal Component Analysis Networks (PCAs)

Principal Component Analysis Networks (PCAs) combine unsupervised and supervised learning in the same topology. Principal component analysis is an unsupervised linear procedure that finds a set of uncorrelated features, principal components, from the input. A MLP is supervised to perform the nonlinear classification from these components. The fundamental problem in pattern recognition is to define data features that are important for the classification (feature extraction). One wishes to transform the input samples into a new space (the feature space) where the information about the samples is retained, but the dimensionality is reduced. This will make the classification job much easier. Principal component analysis (PCA) is such a technique. PCA finds an orthogonal set of directions in

the input space and provides a way of finding the projections into these directions in an ordered fashion. The first principal component is the one that has the largest projection. The orthogonal directions are called the eigenvectors of the correlation matrix of the input vector, and the projections the corresponding eigenvalues. Since PCA orders the projections, the dimensionality can be reduced by truncating the projections to a given order. The reconstruction error is equal to the sum of the projections (eigenvalues) left out. The features in the projection space become the eigenvalues. This projection space is linear. PCA is normally done by analytically solving an eigenvalue problem of the input correlation function. Also, PCA can be accomplished by a single layer linear neural network trained with a modified Hebbian learning rule.

5.8 Radial Basis Function (RBF) Networks

Radial Basis Function (RBF) Networks are nonlinear hybrid networks typically containing a single hidden layer of processing elements (PEs). This layer uses gaussian transfer functions, rather than the standard sigmoidal functions employed by MLPs. The centers and widths of the gaussians are set by unsupervised learning rules, and supervised learning is applied to the output layer. These networks tend to learn much faster than MLPs. If a generalized regression (GRNN) / probabilistic (PNN) net is chosen, all the weights of the network can be calculated analytically. In this case, the number of cluster centers is by definition equal to the number of exemplars, and they are all set to the same variance. This type of RBF is used only when the number of exemplars is so small (<100) or so dispersed that clustering is ill-defined. Radial basis functions networks have a very strong mathematical foundation rooted in regularization theory for solving ill-conditioned problems. The RBF networks can be constructed as shown in figure 5. Every input component (ρ) is brought to a layer of hidden nodes. Each node in the hidden layer is a ρ multivariate Gaussian function,

$$G(x; x_i) = \exp \left[\frac{-1}{2\sigma_i^2} \sum_{k=1}^p (x_k - x_{ik})^2 \right]$$

of mean x_i (each data point) and variance σ_i . These functions are called radial basis functions. Finally, linearly weight the output of the hidden nodes to obtain

$$F(x) = \sum_{i=1} w_i (G(x; x_i))$$

But, it may lead to a very large hidden layer (number of samples of training set).

This solution can be approximated by reducing the number of PEs in the hidden layer, but cleverly position them over the input space regions, i.e. where more input samples are available. This requires the estimation of the positions of each radial basis function and its variance (width), as well as computes the linear weights.

Self-Organizing Feature Maps (SOFMs) transform the input of arbitrary dimension into a one or two dimensional discrete map subject to a topological (neighborhood preserving) constraint. The feature maps are computed using Kohonen unsupervised learning. The output of the SOFM can be used as input to a supervised classification neural network such as the MLP. This network's key advantage is the clustering produced by the SOFM which reduces the input space into representative features using a self-organizing process. Hence, the underlying structure of the input space is kept, while the dimensionality of the space is

reduced. These nets are one layer nets with linear PEs but use a competitive learning rule. In such nets there is one and only one winning PE for every input pattern (i.e. the PE whose weights are closest to the input pattern). In competitive nets, only the weights of the winning node get updated. Kohonen proposed a slight modification of this principle with tremendous implications. Instead of updating only the winning PE, in SOFM nets the neighboring PE weights are also updated with a smaller step size. This means that in the learning process (topological) neighborhood relationships are created in which the spatial locations correspond to features of the input data. These data points that are similar in input space can be mapped to small neighborhoods in Kohonen's SOFM layer.

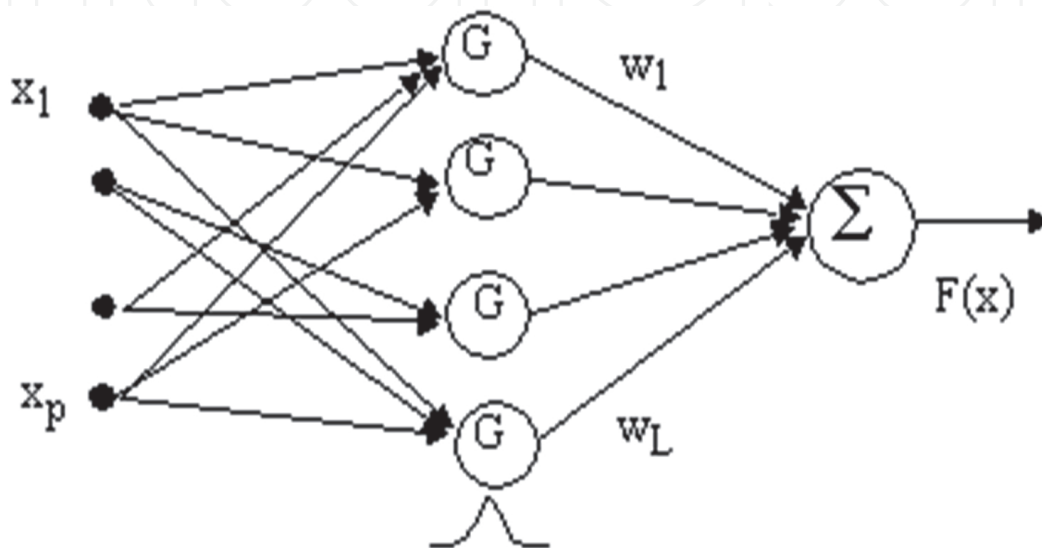


Fig. 5. Radial Basis Function (RBF) Network

6. Performance measures

6.1 MSE (Mean Square Error)

The formula for the mean square error is

$$MSE = \sum_{j=0}^p \sum_{i=0}^n (d_{ij} - y_{ij})^2 / NP$$

where P = number of output processing elements, N = number of exemplars in the data set, y_{ij} = network output for exemplar i at processing element j , d_{ij} = desired output for exemplar i at processing element j .

Learning of a neural network is a stochastic process that depends not only on the learning parameters, but also on the initial conditions. Thus, if it is required to compare network convergence time or final value of the MSE after a number of iterations, it is necessary to run each network several times with random initial conditions and pick the best.

6.2 r (Correlation coefficient)

The size of the mean square error (MSE) can be used to determine how well the network output fits the desired output, but it doesn't necessarily reflect whether the two sets of data move in the same direction. For instance, by simply scaling the network output, the MSE

can be changed without changing the directionality of the data. The correlation coefficient (r) solves this problem. By definition, the correlation coefficient between a network output x and a desired output d is:

$$r = \frac{\sum_i (x_1 - \bar{x})(d_1 - \bar{d})}{\sqrt{\frac{\sum_i (d_1 - \bar{d})^2}{N}} \sqrt{\frac{\sum_i (x_1 - \bar{x})^2}{N}}}$$

The Numerator is the covariance of the two variables and the denominator is the product of the corresponding standard deviation. The correlation coefficient is confined to the range $[-1,1]$. When $r = 1$, there is a perfect positive linear correlation between x and d , that is, they co-vary, which means that they vary by the same amount. When $r = -1$, there is a perfectly linear negative correlation between x and d , that is, they vary in opposite ways. When $r = 0$, there is no correlation between x and d , that is, the variables are uncorrelated. Intermediate values reveal partial correlations, e.g. $r = 0.9$, which states that the fit of the linear model to the data is reasonably good.

Correlation Coefficient, r tells how much of the variance of d is captured by a linear regression on the independent variable x , and hence r is a very effective quantifier of the modeling result. It has the greatest advantage with respect to the MSE as it is automatically normalized, while the MSE is not. But, r is blind to the differences in means as it is a ratio of variances, i.e. as long as the desired data and input co-vary, r will be small, in spite of the fact that they may be far apart in actual value. Hence, both parameters (r and MSE) are required when testing the results of regression.

6.3 The N/P ratio

It describes the complexity of a neural network and is given by
 $N/P = \text{Total Training Samples} / \text{Number of connection weights}.$

6.4 Time elapsed per epoch per exemplar (t)

It helps to calculate the speed of a network and is given by
 $t = \text{Time elapsed for } n \text{ samples} / (n \text{ samples} \times \text{total training samples})$

7. Database descriptions

The EMG signal appears like a random-noise waveform, with the energy of a signal, a function of amount of the muscle activity and electrode placement. The waveform was obtained at a sweep speed of 10 milliseconds per cm; amplitude of 1 mV per cm. The data related to EMG noise signal was obtained from standard data sources available. (Veterans Administration Hospital, Portland) The EMG signal under consideration had three sample patterns, 408, 890 and 1500 samples. Out of total samples, 50% samples constitute the training set, 20% samples are used for cross-validation and 30% samples are chosen for testing set. The training set is used to train the neural network. During the learning, the weights and biases are updated dynamically using the back propagation algorithm. The validation set is used to determine the performance of the neural network on patterns that are not trained during learning. Its major goal is to avoid the over training during the

learning phase. The testing set is used to check the overall performance of the network. The input PE and output PE were chosen to be one, as it is a single input (i.e. noisy EMG input) and a single output (i.e. desired or filtered EMG output), SISO system. The neural network defined has the other parameters like context unit (time) = 0.8, transfer function = Tanhaxon, learning rule = momentum. Termination criterion for experimentation is minimum MSE in both training and cross-validation stages with maximum epochs = 1000 and learning rate is fixed to 0.01. MSE criterion is limited to 1%.

The noisy EMG and desired EMG signals are inputted to Neural Networks and desired signal is expected with mean square error limited to 1%. The MLP, General Feed Forward, Modular Neural Network, Jordan/ Elman Network, RBF Neural Network, and Recurrent Network neural networks have been tried for optimal performance and it is found that the Neural Networks are optimally performing. The performance measures like MSE and Correlation Coefficient (r) are specified in the table 8. [Strum R. D., 2000, Andrzej Izvorski, et al., 2004]

8. Simulation

The results are obtained on Neuro Solutions platform and accordingly, simulations are carried out on noisy EMG input and desired EMG signal. The noisy EMG input was inputted to different neural networks with number of hidden layers varying from 2 to 4. The neural networks with input, hidden and output layer with varying parameters like processing elements, transfer function, learning rule, step size and momentum were tested with maximum epoch value, 1000.

After training the Neural Networks on a noisy input and desired output data values with different sample patterns and under different (training, cross-validation and testing samples swapped) conditions, the expected results were obtained with minimum MSE and maximum correlation coefficient around the estimated values as shown below. The other parameters like processing element per hidden layer, transfer function, learning rule were also varied. The results for optimum parameters for SISO system under consideration are given in following tables.

a) GEN. FF. NN, (05, 05, 07), 60% TR, 15% CV, 25% Test SP

Figure 6 depicts the variation of average of minimum MSE for 5 runs vs. number of PEs in the second hidden layer. It is observed that for five processing elements in the second hidden layer, the MSE on CV attained its minimum value. When the PE's are increased beyond 5, the MSE on CV was seen to increase. Therefore, 5 PE's are chosen for second hidden layer.

Figure 7 depicts the variation of average Training MSE vs. number of Epochs. Five different runs with new random initialization of connection weights of NN's are shown below. It is observed that for each run (training cycle), average MSE decreases as number of epochs increases. It is worthwhile to notice that this trend of decrease in MSE is consistent for 5 runs.

Figure 8 shows the variation of desired output and actual NN output vs. number of exemplars. The covariance between the desired output and the actual NN output is indicated by the correlation coefficient, $r = 0.636240018$.

The neural network was trained five times and the best performance with respect to MSE of training was observed during the 1st run at the end of 1000 epochs. Similarly, the best Cross-Validation performance was noticed during the 4th run at the end of 12 epochs, as depicted in the following table 1.

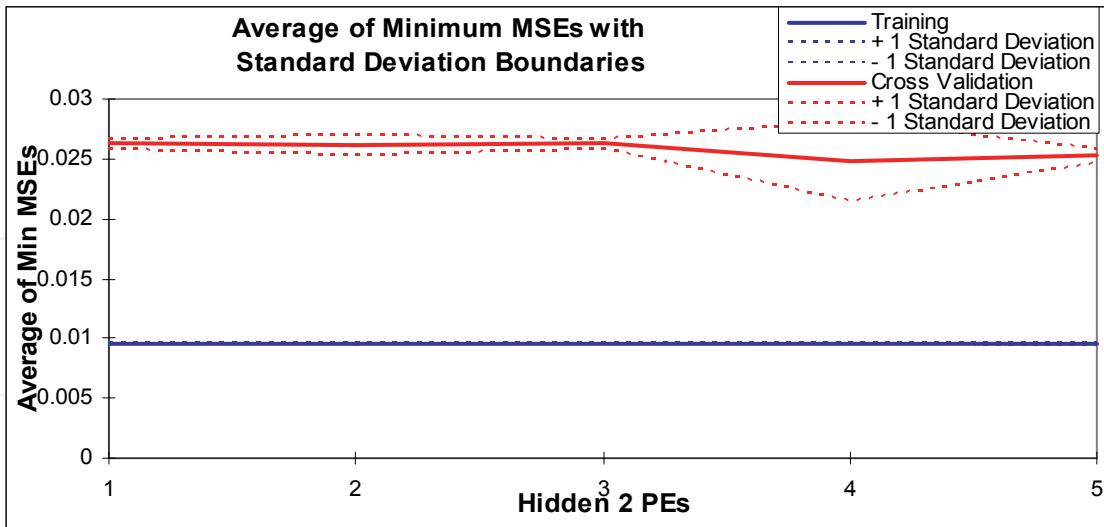


Fig. 6.

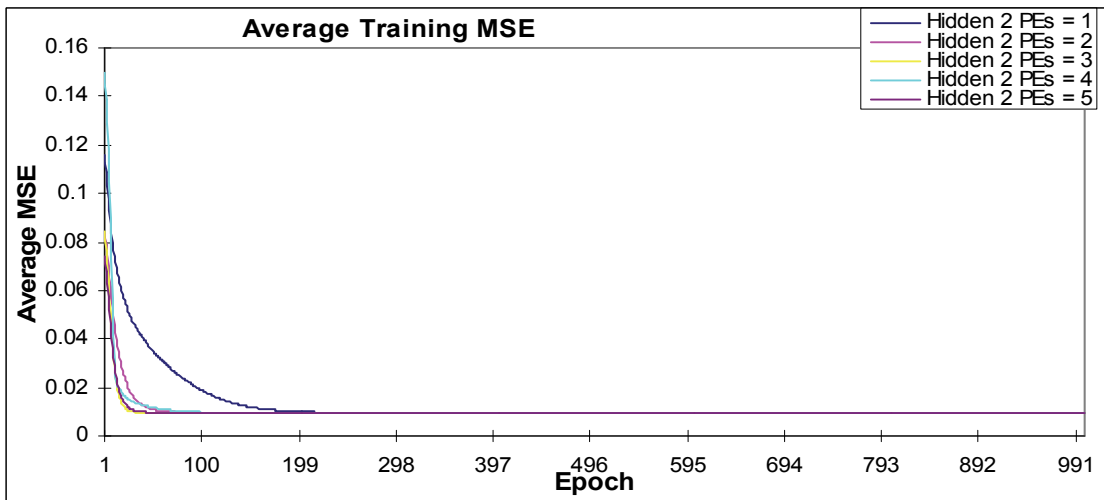


Fig. 7.

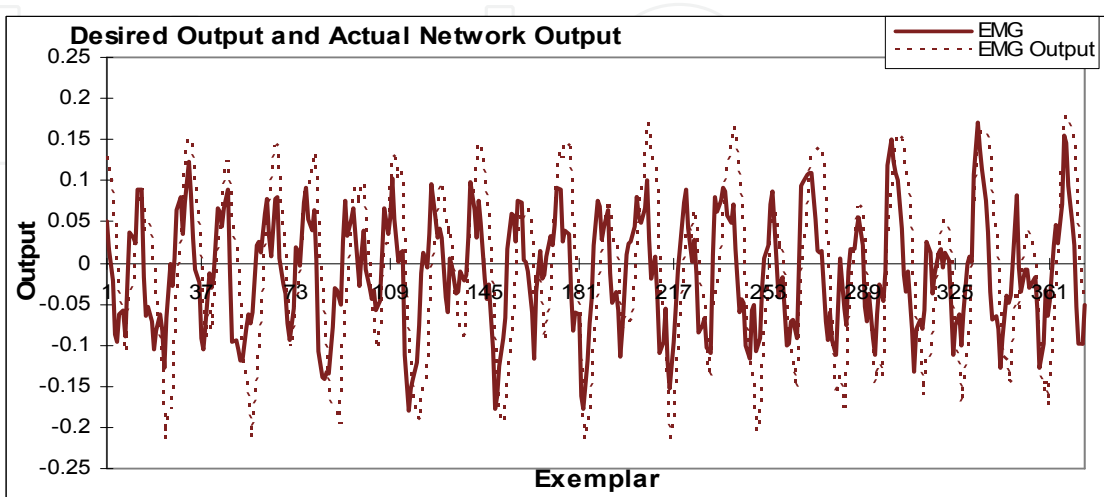


Fig. 8.

Best Networks	Training	Cross Validation
Hidden 2 PEs	5	4
Run #	1	4
Epoch #	1000	12
Minimum MSE	0.009501057	0.018979003
Final MSE	0.009501057	0.018979003

Table 1.

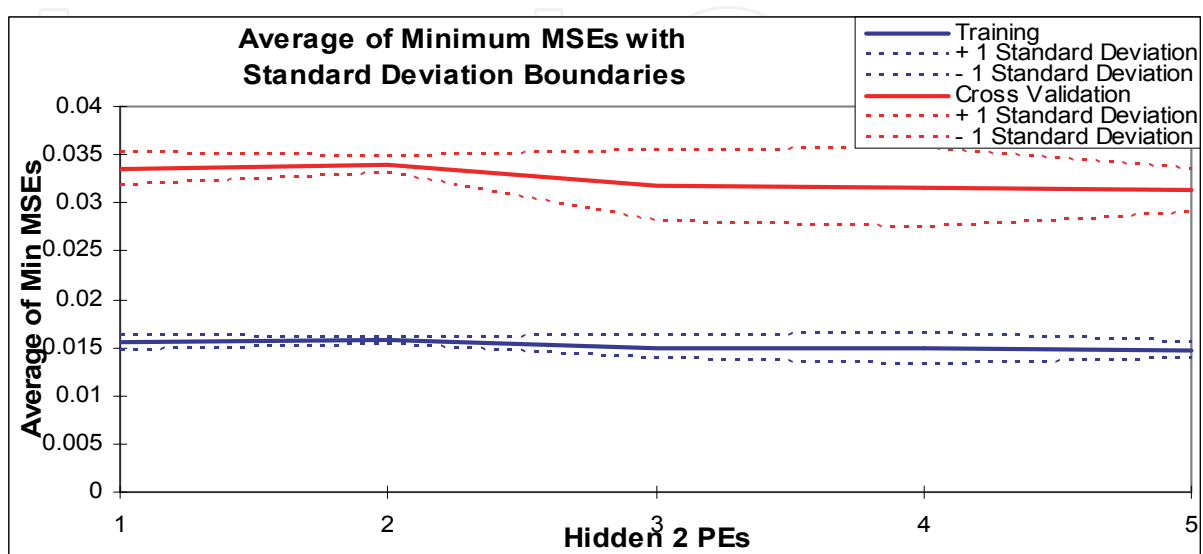
Different performance measures are listed in the following table 2.

Performance	EMG
MSE	0.004679485
NMSE	1.318501294
MAE	0.068195457
Min Abs Error	5.04679E-05
Max Abs Error	0.176143173
r	0.636240018

Table 2.

b) JORDAN / ELMAN NN, (08, 04, 02), 50% TR, 20% CV, 30% Test SP

Figure 9 depicts the variation of average of minimum MSE for 5 runs vs. number of PEs in the second hidden layer. It is observed that for four processing elements in the second hidden layer, the MSE on CV attained its minimum value. When the PE's are increased beyond 4, the MSE on CV was seen to increase. Therefore, 4 PE's are chosen for second hidden layer.



Different performance measures are listed in the following table 3.

Performance	EMG
MSE	0.015119372
NMSE	0.812867701
MAE	0.103271719
Min Abs Error	9.59247E-05
Max Abs Error	0.42812628
r	0.805945071

Figure 10 depicts the variation of average Training MSE vs. number of Epochs. Five different runs with new random initialization of connection weights of NN's are shown below. It is observed that for each run (training cycle), average MSE decreases as number of epochs increases. It is worthwhile to notice that this trend of decrease in MSE is consistent for 5 runs.

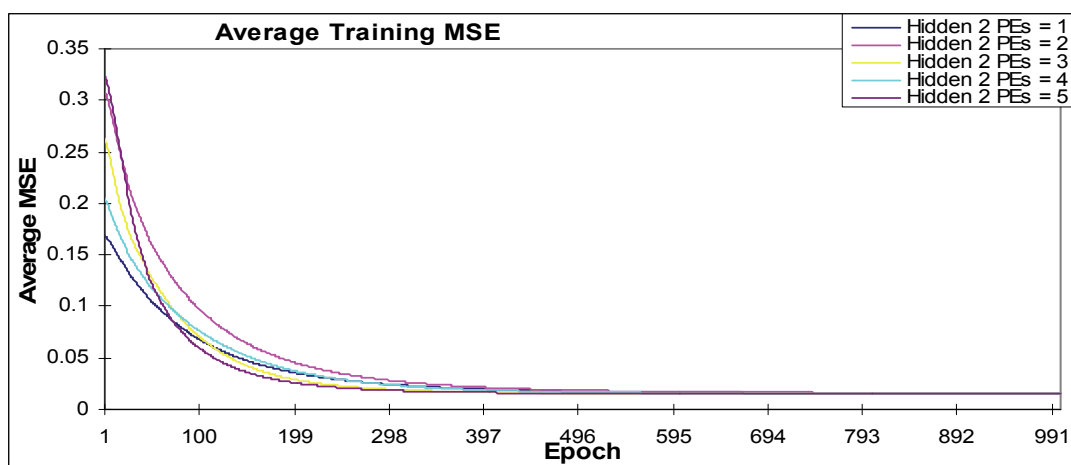
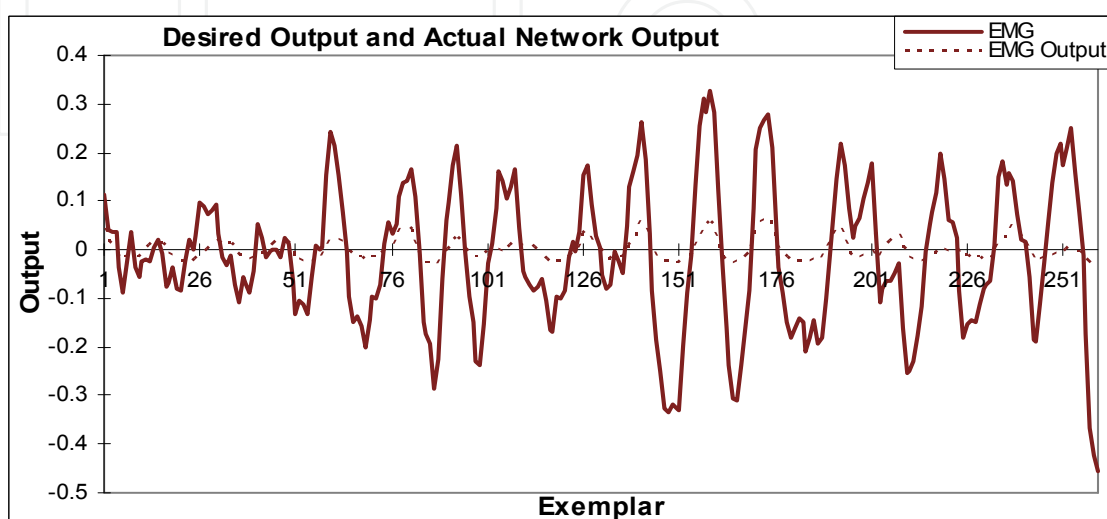


Figure 11 shows the variation of desired output and actual NN output vs. number of exemplars. The covariance between the desired output and the actual NN output is indicated by the correlation coefficient, $r = 0.805945071$



The neural network was trained five times and the best performance with respect to MSE of training was observed during the 2nd run at the end of 1000 epochs. Similarly, the best Cross-Validation performance was noticed during the 2nd run at the end of 1000 epochs, as depicted in the following table 4.

Best Networks	Training	Cross Validation
Hidden 2 PEs	4	4
Run #	2	2
Epoch #	1000	1000
Minimum MSE	0.009983631	0.021040779
Final MSE	0.009983631	0.021040779

c) GEN. FF NN, (04, 02), 50% TR, 20% CV, 30% Test SP

Figure 12 depicts the variation of average of minimum MSE for 5 runs vs. number of PEs in the first hidden layer. It is observed that for four processing elements in the first hidden layer, the MSE on CV attained its minimum value. When PEs are increased beyond 4, the MSE on CV was seen to increase. Therefore, 4 PEs are chosen for first hidden layer.

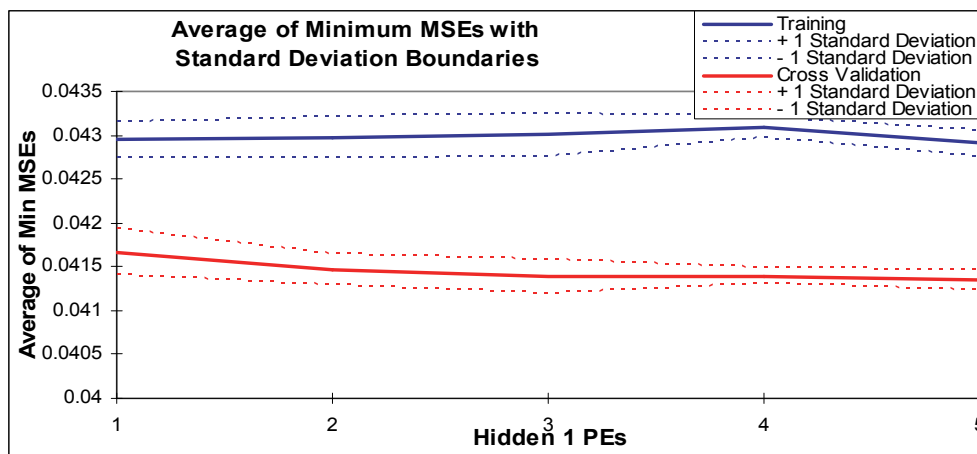


Figure 13 depicts the variation of average Training MSE vs. number of Epochs. Five different runs with new random initialization of connection weights of NNs are shown below. It is observed that for each run (training cycle), average MSE decreases as number of epochs increases. It is worthwhile to notice that this trend of decrease in MSE is consistent for all the five runs.

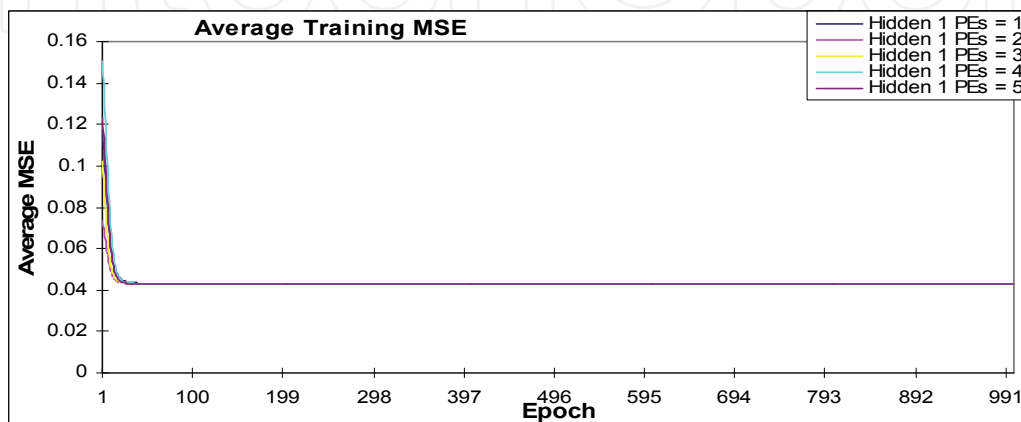
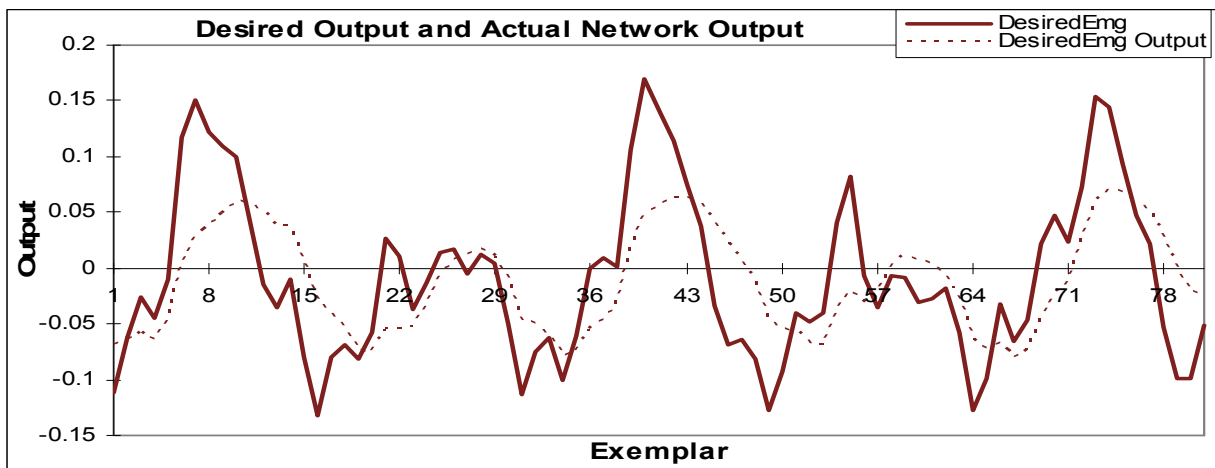


Figure 14 shows the variation of desired output and actual NN output vs. number of exemplars. The covariance between the desired output and the actual NN output is indicated by the correlation coefficient, $r = 0.662274425$.



Different performance measures are listed in the following table 5.

Performance	DesiredEmg
MSE	0.003100258
NMSE	0.566522078
MAE	0.04580669
Min Abs Error	0.001058364
Max Abs Error	0.123692824
r	0.662274425

The neural network was trained five times and the best performance with respect to MSE of training was observed during the 1st run at the end of 1000 epochs. Similarly, the best Cross-Validation performance was noticed during the 2nd run at the end of 23 epochs, as depicted in the following table 6.

Best Networks	Training	Cross Validation
Hidden 1 PEs	3	5
Run #	1	2
Epoch #	1000	23
Minimum MSE	0.04264781	0.041166622
Final MSE	0.04264781	0.041874009

The optimum results experiment-wise are presented in table 7 in consolidated form.

Expt. No.	Type of ANN	Hidden Layer Variation H1H2H3	Correlation Coefficient r	MSE			N/P	t (μ sec)
				Training	Cross validation	Testing		
01	Gen.NN	05,05,07	0.636240018	0.009501057	0.018979003	0.004679485	12.5	6.2
02	Jor/Elmn	08,04,02	0.805945071	0.009983631	0.021040779	0.015119372	9.782	6.5
03	Gen.NN	04,02	0.662274425	0.04264781	0.041874009	0.003100258	14.5	57.7

9. Conclusion

EMG signal is a very important biomedical signal associated with muscle activity, giving useful information about nerve system in order to detect abnormal muscle electrical activities that occur in many diseases and conditions like muscular dystrophy, inflammation of muscles, pinched nerves, peripheral nerve damages, amyotrophic lateral sclerosis, disc herniation, myasthenia gravis, and others. The detection and measurement of low frequency and lower magnitude EMG signals is noise-prone. Removal of noise from an EMG signal using various Neural Networks has been studied. It is demonstrated that Jordan/Elman Neural Network and Generalized Feed Forward Neural Network elegantly reduce the noise from the EMG signal.

One of the simplest methods is to observe the better performing network, how the MSE, which is the square difference between the network's output and the desired response, changes over training iterations. The goal of the stop criterion is to maximize the network's generalization. If the training is successful and the network's topology is correct, the network will apply its 'past experience' to the unseen data and will produce a good solution. If this is the case, then the network will be able to generalize based on the training set. It is observed that the networks presented in Table 7 are exhibiting better MSE values in the training, cross-validation and testing phase, which are distinctly designed. The MSE values are found to be in the desired range i. e. nearly equal to 0.01 in all the phases. The networks are not memorizing the training patterns, nor rattling in the local minima. These networks can be considered to provide good generalization.

The difference between the noisy EMG signal and the desired EMG signal is computed as a performance measure (MSE) and is found to be in the expected range approaching to 0.01. The minimum MSE criterion is found satisfactory (0.0099-0.01) in trained Jordan/Elman Neural Network and found to perform better during testing phase (0.01) as it is evident from all similar graphs as in Figure 9.

The correlation coefficient ' r ' is a very effective quantifier of the modeling results, which describes the covariance between the desired output and the actual neural network's output. As can be seen from Table 7, the generalized neural network is found to have maximum correlation coefficient ' r ' value, under several varying test conditions. The generalized neural network is found to effectively utilize the knowledge embedded in the input data and the desired response as compared to MLP, Modular Neural Network, Jordan/ Elman Neural Network, RBF Neural Network, Time Lag Recurrent Network and Recurrent Neural Network. The ' r ' is nearly close to 1 in most of the experiments, indicating a better linear correlation between the desired output and the actual neural network's output. Only the result of three Neural Networks have been presented here.

Also, the correlation coefficient (r) is found to be in the desired range (0.805945071 at sr. no. 2 Table 7), so that the network output and the desired output co-varies, i.e. varying by the same amount as depicted in figure 11.

As can be seen from the Table 7, the number of hidden layers used in the topology is either 2 or 3. This indicates that a simple neural network configurations can be very conveniently used to better generalize the input/ output mapping, in accordance with the theory of generalization, which correlates the number of PEs in the hidden layer, number of hidden layers with the mapping ability of the neural networks. N/P ratio describes the complexity of the neural network. The obtained values (9.782 to 14.5) of the ratio N/P shows that the neural network so designed is simpler to design and is capable of good generalization, with a better ability to learn from training exemplars. The neural networks presented in Table 7. are found to have smaller number of degrees of freedom achieving the desired performance and hence the networks can be said to be optimal.

Also, moderately smaller values of N/P shows that the Jordan/ Elman Neural Network so designed is simpler to design and is capable of generalization.

The time elapsed per epoch per exemplar (t) helps to calculate the speed of a network. Time t describes the training time elapsed per epoch per exemplar. Smaller values of t make it evident that the designed neural network requires less training time and hence, is faster.

Thus, from above, it can be concluded that Neural Networks can be designed to perform better as far as the overall performance is concerned. The designed neural networks have right combination of PEs and hidden layers to solve the given problem with acceptable training times and performance. Other neural networks are also performing optimally and are situation dependent. Accordingly, an optimal network needs to be selected for a particular application.

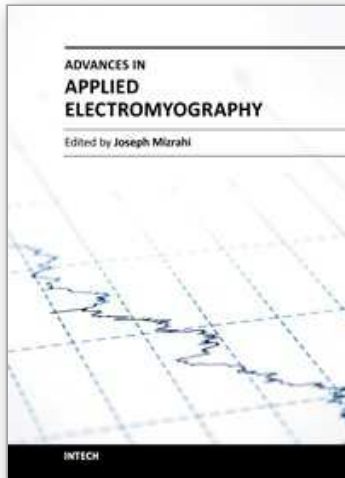
10. References

- Bronzino, J.D. (ed), "The Biomedical Engineering Handbook", Boca Raton, FL: CRC Press, 1995
- Brush, L.C. et al., "The guide to Biomedical Standards ", 20th ed. Bera, CA: Quest publishing Co., 1995.
- Cohen, B.J., "Medical Terminology: An Illustrated Guide", 2nd ed. Philadelphia: Lippincott, 1995.
- Cromwell, L., F. J. Webell, and E.A. Pfeiffer. "Biomedical Instrumentation and Measurements", 2nd ed. Englewood Cliffs. NJ: Prentice-Hall, 1980.
- John G. Webster (Ed.), "Medical Instrumentation : Application and Design", 3rd ed., John Wiley and Sons, Inc. 2001
- Tompkins, W. J. (ed.) "Biomedical Digital Signal Processing" PHI, ND, 1999.
- Cromwell L., Weibell F. J., Pfeiffer E.A. "Biomedical Instrumentation and Measurements", PHI, N.D., 2004
- Ezin C. Eugène, " Discriminating Noisy Sentences with Partially Recurrent Neural Networks", [www.cari-info.org/actes 2006/ 59.pdf](http://www.cari-info.org/actes%202006/59.pdf), 30.06.2008.
- Xue, Q.Z., Hu, Y. H. and Tompkins, W.J. 1992. " Neural-Network-based Adaptive Matched Filtering for QRS Detection", IEEE Trans. Biomed Engg., BMN-39(4): 317-29.
- Richard D. de Veaux, Jennifer Schumi, Jason Schweinsberg. Lyle H. Ungar, "Prediction Intervals for Neural Networks via Nonlinear Regression", Technometrics, Vol. 40, No. 4, pp. 273-282, Nov. 1998.

- J.C. Principe, N.R. Euliano and W.C. Lefebvre, "Neural and Adaptive Systems", New York, USA: John Wiley and Sons, Inc. 2000.
- Haykin, S "Adaptive Filter theory", Englewood Cliffs, NJ: Prentice Hall, 1986.
- Widrow, B, Glover, J.R., John M., et al "Adaptive Noise Canceling Principles and Applications", Proc. IEEE, 63(12): 1692-1716, 1975.
- Khandpur R.S. "Handbook of Biomedical Instrumentation", Tata McGraw-Hill, N.D., 2001.
- Abdelhafid Zeghib, et. al., "Sampling Frequency and Pass-Band Frequency Effects on Neuromuscular Signals (EMG) Recognition", Proc. of 6th WSEAS Intl. Conf. on Signal Processing, Robotics and Automation, Corfu Island, Greece, Feb.16-19, 2007, pp 107-114.
- Umut Gundogdu, et. al., "Investigation of Muscle Fatigue Using Temporal and Spectral Moments", Proc. of 5th WSEAS Intl. Conf. on Signal Processing, Istanbul, Turkey, May 27-29, 2006, pp10-14.
- Mercedes Cabrerizo, et. al., "EEG Analysis using Neural Networks for Seizure Detection", Proc. of 6th WSEAS Intl. Conf. on Signal Processing, Robotics and Automation, Corfu Island, Greece, Feb.16-19, 2007, pp 121-126.
- David Coufal, "Classification of EEG Signals by Radial Neuro-Fuzzy System", Proc. of 4th WSEAS Intl. Conf. on Computational Intelligence, Man-Machine Systems and Cybernetics, Miami, Florida, USA, Nov. 17-19,2005, pp 224-232.
- Mahesh S. Chavan, et. al., "Use of Kaiser Window For ECG Processing", Proc. of 5th WSEAS Intl. Conf. on Signal Processing, Robotics and Automation, Madrid, Spain, Feb.15-17, 2006, pp285-289.
- Hafizah Husain, Lai Len Fatt, "Efficient ECG Signal Classification Using Sparsely Connected Radial Basis Function Neural Network", 6th WSEAS Intl. Conf. on Circuits, Systems, Electronics, Control and Signal Processing, Cairo, Egypt, Dec. 29-31, 2007, pp 412-416.
- M.Areziki, et. al., "Analysis of Fast Recursive Least Squares Algorithms for Adaptive Filtering", Proc. of 11th WSEAS Intl. Conf. on Systems, Agios Nikolaos, Crete Island, Greece, Jul.23-25, 2007, pp 474-481.
- Chunshien Li, "Adaptive Noise Cancellation with Computational-Intelligence-based Approach", Proc. of 5th WSEAS Intl. Conf. on Signal Processing, Istanbul, Turkey, May 27-29, 2006, pp93-98.
- Edward A. Clancy, Neville Hogan, "Multiple Site Electromyograph Amplitude Estimation", IEEE Trans. on Biomedical Engineering, Vol. 42, No. 2, Feb.1995, pp 203-211.
- Yves St-Amant, Denis Rancourt, Edward A. Clancy, "Influence of Smoothing Window Length on Electromyogram Amplitude Estimates", IEEE Trans. on Biomedical Engineering, Vol. 45, No. 6, June 1998, pp 795-800.
- Lin Wang, Thomas S. Buchanan, "Prediction of Joint Moments using a Neural Network Model of Muscle Activations from EMG Signals", IEEE Trans. on Neural Systems and Rehabilitation Engineering, Vol. 10, No. 1, Mar. 2002, pp 30-37.
- Andrew Hamilton-Wright, Daniel W. Stashuk, "Physiologically Based Simulation of Clinical EMG Signals", IEEE Trans. on Biomedical Engineering, Vol. 52, No. 2, Feb. 2005, pp 171-183.
- ZhiQiang Zhang, Zheng Tang and Catherine Vairappan, "A Novel Learning Method for Elman Neural Network Using Local Search", Neural Information Processing - Letters and Reviews Vol. 11, No. 8, pp 181- 188, August 2007.

- Geddes, L.A., and L.E. Baker, "Principles of Applied Biomedical Instrumentation", 3rd ed. New York: Wiley, 1989.
- Strum R. D., Kirk D.E., "Contemporary Linear Systems Using Matlab", Thomson Learning, 2000.
- Childers, D.G., J.G. Webster (ed.), "Evoked potentials", Encyclopedia of Medical Devices and Instrumentation. New York: Wiley, 1988, pp. 1245-1254.
- Andrzej Izworski, Piotr Bamia, "Interpretation of the Biomedical Signals using RBF- type Neural Networks", WSEAS trans on Circuits and systems, Issue 9, Vol.3, Nov 2004, pp 1799-1803.

IntechOpen



Advances in Applied Electromyography

Edited by Prof. Joseph Mizrahi

ISBN 978-953-307-382-8

Hard cover, 212 pages

Publisher InTech

Published online 29, August, 2011

Published in print edition August, 2011

The electrical activity of the muscles, as measured by means of electromyography (EMG), is a major expression of muscle contraction. This book aims at providing an updated overview of the recent developments in electromyography from diverse aspects and various applications in clinical and experimental research. It consists of ten chapters arranged in four sections. The first section deals with EMG signals from skeletal muscles and their significance in assessing biomechanical and physiologic function and in applications in neuro-musculo-skeletal rehabilitation. The second section addresses methodologies for the treatment of the signal itself: noise removal and pattern recognition for the activation of artificial limbs. The third section deals with utilizing the EMG signals for inferring on the mechanical action of the muscle, such as force, e.g., pinching force in humans or sucking pressure in the cibarial pump during feeding of the hematophagous hemiptera insect. The fourth and last section deals with the clinical role of electromyograms in studying the pelvic floor muscle function.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Vijay R. Mankar (2011). EMG Signal Noise Removal Using Neural Networks, *Advances in Applied Electromyography*, Prof. Joseph Mizrahi (Ed.), ISBN: 978-953-307-382-8, InTech, Available from: <http://www.intechopen.com/books/advances-in-applied-electromyography/emg-signal-noise-removal-using-neural-networks>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen