6-16-1994

Analysis of Myoelectrical Signals for Building a Dextrous Hand

Christopher T. Creel

Kishan Mehrotra
Syracuse University, mehrotra@syr.edu

Chilukuri K. Mohan
Syracuse University, ckmohan@syr.edu

Sanjay Ranka
Syracuse University

SU-CIS-94-3

Analysis of Myoelectrical Signals for Building a Dextrous Hand

Christopher T. Creel, Kishan Mehrotra,
Chilukuri Mohan, & Sanjay Ranka

June 16, 1994

School of Computer and Information Science
Syracuse University
Suite 4-116, Center for Science and Technology
Syracuse, NY 13244-4100

Analysis of Myoelectrical Signals for Building a Dextrous Hand

Christopher T. Creel
Kishan Mehrotra, Chilukuri Mohan, Sanjay Ranka

School of Computer and Information Science
4-116 Center for Science and Technology
Syracuse University
Syracuse, NY 13244-4100

email: ctcreel/kishan/mohan/ranka@top.cis.syr.edu
tel: (315) 443-2368

June 16, 1994

Abstract

We analyze techniques for myoelectrical signals classification for the purpose of designing a multifunctional prosthetic device for human amputees. The main advantage of our system over existing models is that it is more robust, easier to work with, more general, and efficient enough to run in real time. We achieve this with the help of "Supervised Growing Cell Structures." an artificial neural network model designed by Fritzke [10]. The current paper focuses on the flexion of the index, middle and ring fingers, as these are the most difficult movements to tackle.

Keywords: Dexterous Hand, Phantom Limbs, Statistical Analysis, Neural Networks, Classification

1 Introduction

The current state of the art in artificial prosthesis has only two degrees of freedom; the opening and closing of a "gripper." The most sophisticated of this prostheses allows the wearer to control the speed and strength of this opening and closing [1]. In addition to this limited range of movement, the amputee must endure rigorous training in order to learn how to use the prosthesis. This training often involves unnatural movements of their phantom limb in order to obtain the desired results in the prosthesis [2]. The goal of this project is to design an interface to a prosthesis that would be more natural for the amputee to use and provide greater functionality. Although our study is confined to the development of an artificial arm, the basic ideas extend much beyond this limited scope. Our system could be used to classify myoelectric signals obtained from any muscle which controls a phantom limb and use these signals for a variety of other functions, i.e., wheel chair control or a crude language interface. The interface we have researched reads myoelectrical signals (MES) from specific muscles in the arm between the elbow and the point of amputation and attempts to classify those signals with mathematical transformations, statistical techniques, and artificial neural networks. In the next section we provide a brief review of the previous attempts to classify myoelectrical signals. We then discuss the pertinent anatomy and touch on the issue of phantom limbs. In section 3 we describe our method of MES data collection. Sections 4, 5, and 6 contain the statistical and neural networks classification procedures, respectively. Section 7 contains a comparison of neural networks classifications.

2 Current Technology

The artificial prosthesis currently available allows an amputee to open and close a gripper. These two degrees of freedom provide only crude functionality. For more sophisticated functions, more degrees of freedom are needed. Of course, a more sophisticated device than a gripper is also needed. Instead of using a simple gripper, our goal is to use an artificial hand similar to the one designed by the University of Utah and the Massachusetts Institute of Technology shown in Figure 1 [12].

Using this hand, our goal is to achieve the movements listed in Table 1.

Flexing the index finger	Pronating the hand
Flexing the middle finger	Abducting the hand
Flexing the ring finger	Abduction of the thumb
Reversal of movements 1 to 4	Adduction of the thumb
Flexing the wrist	Flexing the thumb
Extending the wrist	Extending the thumb

Table 1: A list of desired freedom of movements sought in a dexterous hand

The interpretation of the myoelectrical signals produced by the muscles in what remains of the amputee's arm is what restricts the functionality of current prosthetic devices. The

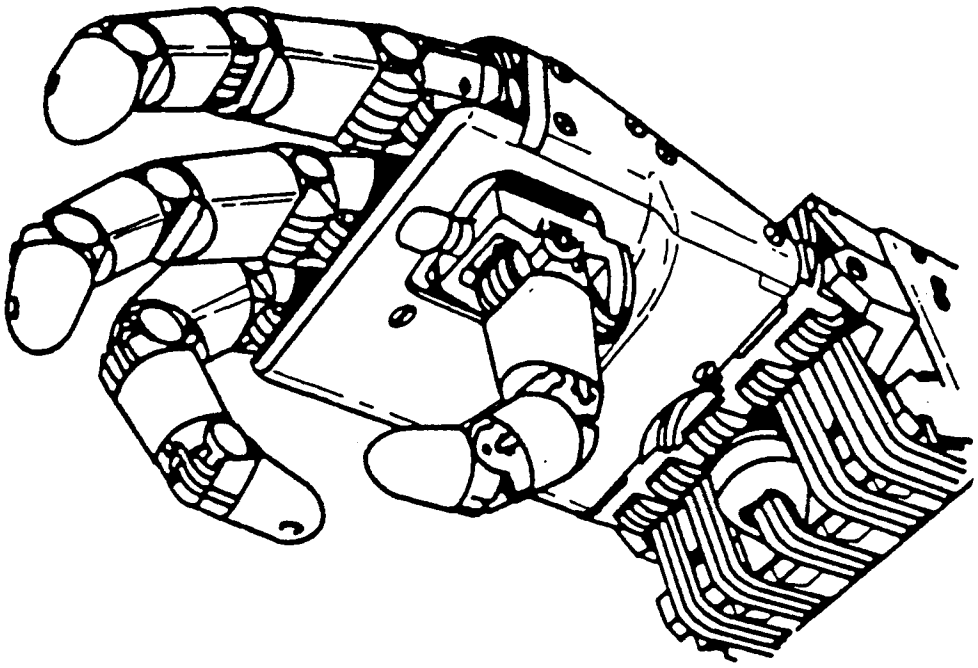

Figure 1: Utah/M.I.T. Dexterous Hand

introduction of noise into the system by other muscles, magnetic fields created by the equipment used in signal detection, or salts on the surface of the skin can corrupt the signals. Even the amount of fat in the arm, which can muffle the signals from the muscles, plays a role in the shape of the signals. In addition to this noise, the muscles of each individual produce different signals to accomplish the same movement . The main tool used to obtain useful information from MES is a collection of statistical techniques developed by Graupe [4] between 1975 and 1985. These techniques rely primarily on the amplitude of the MES signal. The primary drawbacks to these techniques are their lack of robustness and inability to differentiate between multiple myoelectrical signals. In 1988, Hudgins and his research team [2] discussed the idea of using artificial neural networks to classify MES. In their 1993 technical report [8] they perform classification tasks with feed-forward back propagation(FFBP)artificial neural networks [9] for readings taken from both the biceps and the triceps muscles. Their FFBP artificial neural network contained 30 inputs, 8 hidden units, and 4 output units to classify elbow extension, flexion and rotation. They were able to get up to 91% accuracy for normal individuals and up to 85% accuracy for amputees. One of the problems with the feed- forward back propagation network, and one that is stated in Hudgins technical report, is that the architecture of the network is subject specific, i.e., a specific network is needed for each individual. In this report we apply statistical procedures, feed-forward back propagation network, Kohonen network, and the 'Growing-Cell-Structure' network, an alternative classification method developed by Bernd Fritzke.

The Growing-Cell-Structures neural network is adaptive. i.e., it doesn't require a prespecified architecture for each individual. For this reason we expect that the performance of the Growing-Cell-Structure network would be superior in many aspects than other network models.

3 Pertinent Anatomy

There is a complicated division of labor in the forearm. One muscle can play a role in several seemingly unrelated movements. We must identify those muscles whose primary function is to perform the movements we outlined above and then deal with any special cases. Our present interest is in capturing the flexion of the index finger, middle finger and ring finger. The three agonists of the fingers are the flexor digitorum superficialis, the flexor digitorum profundus (see Figure 2) and the extensor digitorum. The primary function of each of these muscles is to flex the fingers (in the case of the flexor digitorum superficialis and the flexor digitorum profundus) and to extend the fingers (in the case of the extensor digitorum). Their secondary function is to flex and extend the wrist respectively, see [5]. This secondary function is an asset to us; we can simply incorporate this special case into the training of our networks for the purpose of classifications. The only problem that arises with these three muscles is that only two of them are accessible with surface electrodes. The flexor digitorum profundus "rises deep," meaning that it is below other muscles and not directly underneath the skin. To reach these muscles, one would have to use needle electrodes, a situation we wanted to avoid. Thus, the muscles in the forearm that can give us information about the movement of the fingers are the flexor digitorum superficialis and the extensor digiti. As we are only interested in the flexion of the fingers for now, only one muscle, the flexor digitorum superficialis is used to collect MES.

Gray's Anatomy gives the following definition of the flexor digitorum superficialis:

> The flexor digitorum superficialis flexes first the middle, [then the last joints], furthest from the palm, of the finger. It is also a flexor of the wrist. It is particularly involved in rapid, forceful flexion of the digits in grasping electromyographically silent in unresisted flexion.

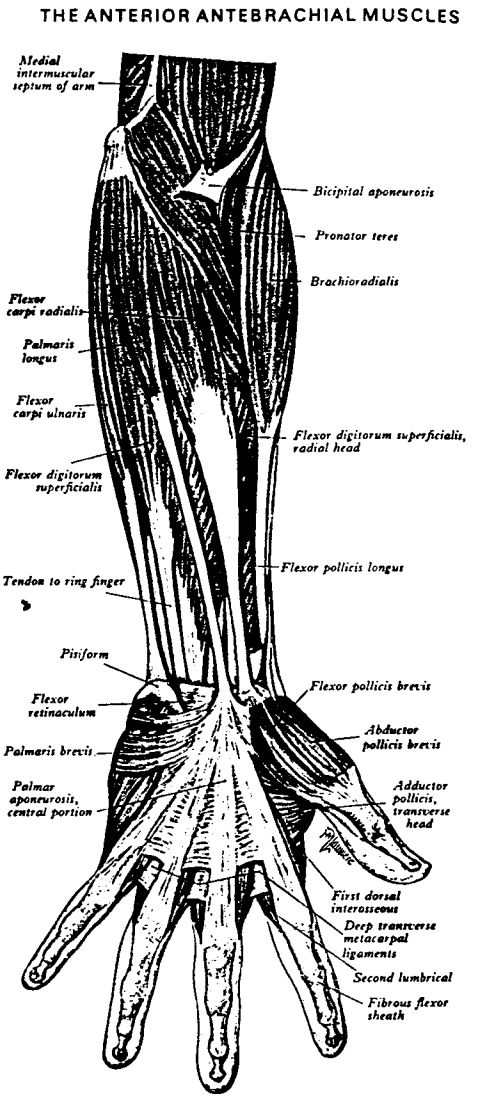

Figure 2: The superficial flexor muscles of the left arm.

During the fine delicate movements of the fingers (i.e., when tying a knot or painting a picture), the flexor digitorum superficialis is electromyographically silent, only the muscles that are physically in the hand and fingers are active. The primary function of the flexor digitorum superficialis is gross movements of the fingers. The problem with collecting MES from only the flexor digitorum superficialis now becomes apparent; it is not active during the fine delicate movements of the fingers. Therefore, we are guaranteed a certain level of failure. As long as no other means of detecting such signals are employed or are available, we will not be able to classify the finer movements of the fingers. In this preliminary phase, we discard this approach and concentrate on the primary function of the muscles that are available to us. This preliminary phase will enable us to decide if using neural networks is a viable approach towards building a dexterous hand.

3.1 Phantom Limbs

There needs to be an assurance that an amputee can control his/her flexor digitorum superficialis just as a normal individual. A common misconception is that once a person looses a limb it diminishes from the mind as well. There is growing evidence against this misconception. As stated in [6] all amputees initially experience a "phantom limb," and in most cases this sensation will persists throughout the rest of their lives. This phantom limb is strikingly real to the amputee; so much so that "below elbow" amputees often talk with their phantom hand(s), or "below knee" amputees accidentally walk on phantom feet. In the amputee, the nerve impulses still travel down the median, ulnar and radial nerves (the nerves that control the muscles in the forearm). These impulses are simply cut short at the point of amputation. As long as the nerve connections from the brain to the muscles in the remaining stump are faithful, and the person can operate their phantom limb as normal limb, the muscles should react just as if there was still a limb connected. There is very little evidence supporting this claim though. We are currently planning a study to confirm that amputees can control the muscles that will be important to our study just as a normal individual.

4 Our Techniques

This section describes the methodology we have used in the collection of the data and includes a brief description of the hardware we use to collect the signals.

4.1 Hardware for MES Collection

There are a wide variety of surface electrodes on the market today. The most popular type is the miniature silver-silver chloride surface electrode [3]. This type of surface electrode has been reliably used for collecting MES's for many years. The signal read by the electrode is then sent to a recording device. But, depending on the resistance of the wire leading from the electrode, the signal will begin to decay very quickly unless it is amplified as soon as possible. The obvious solution is to put an amplifier as close to the electrodes as possible.

This can become very cumbersome depending on the size of the amplifying equipment. We use a preamplifier built by Motion Control to overcome this difficulty. A preamplifier is a 50mm × 15mm × 7mm device which houses three Ag-AgCl electrodes and an amplifier. Thus the signal is immediately amplified and allows for maximum clarity from the surface electrodes with minimal bulk. These preamplifiers require a very small current, 6 to 12 volts, provided through hardware that, in turn, is under software control. Figure 3 shows the system of hardware in detail that we are using for data collection. National Instruments manufactures all of these hardware components, thus the names for most of the hardware. For the sake of brevity, we skip a detailed discussion of the electronic components in figure 3 and suffice it to say that these devices deliver the signal from the preamplifier quickly and efficiently to the computer for further analysis.

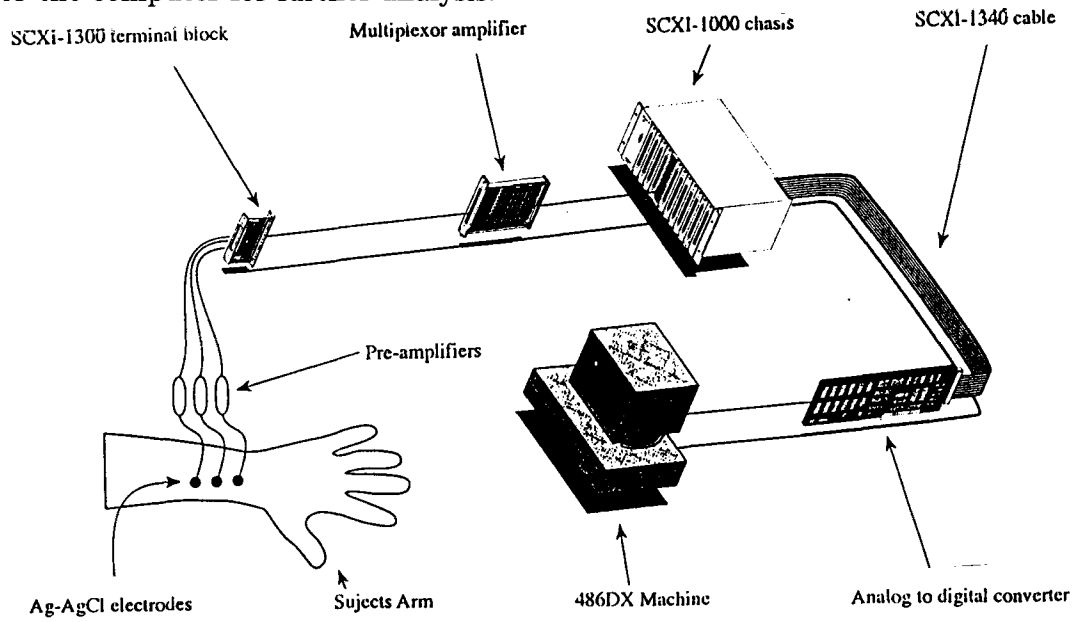

Figure 3: The System of hardware used for data collection

The computer we use is an IBM clone with eight megabytes RAM and an Intel 80486DX running at 66 MHz. The environment we are using is Windows 3.1. Our compiler is Borland's C++ compiler, version 4.0. This compiler allows us tailor our software for Windows 3.1 and the machine we are running it on.

4.2 Data Analysis

The procedure for collecting the data is as follows. We ask the subject to place their arm in a relaxed position by their side and then we:

1. Examine the subjects arm and locate the appropriate muscles (in our case, it is the flexor digitorum superficialis).

2. Place the preamplifiers over these muscles.

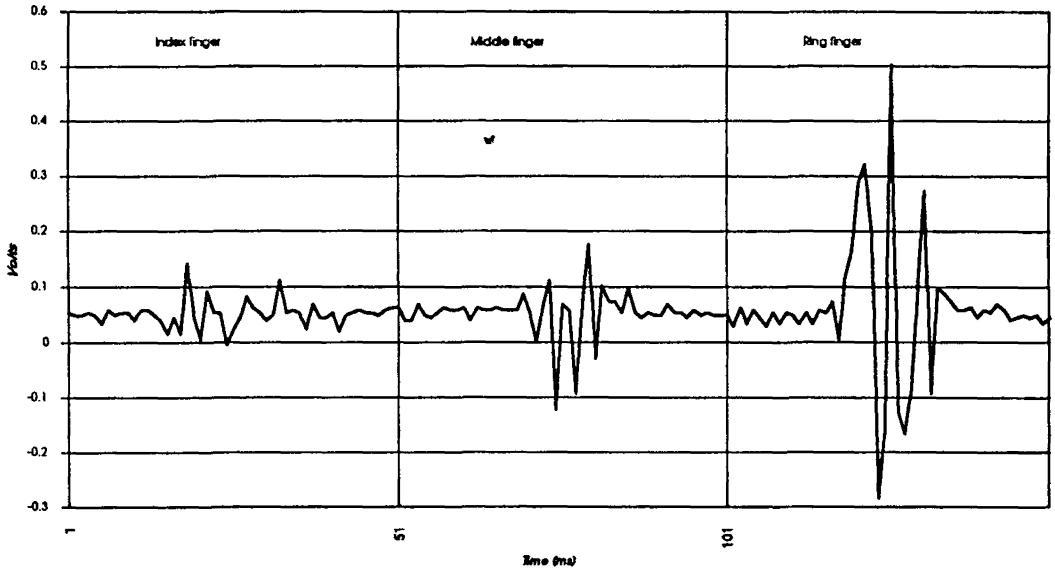

Figure 4: The myoelectrical signal received from flexor digitorum superficialis

3. Ask the subject to perform a test sequence that consists of flexing the ring finger, index finger and middle finger three times each with a two second interval between each movement.

4. Make a preliminary examination of the resulting data. to determine whether the sites that we have chosen to record from are appropriate. If they are, then we go on to step 5, otherwise we return to step 2.

5. Once the electrodes are in the appropriate place, data collection begins for each of the movements described in step 3.

We try to collect about 400 samples. During data collection, the hardware is sampled at 100 pts/sec. Data was collected from three normal individuals using the procedure described above. The data set was split in two parts, a training set and a test set. The training set has 300 samples (100 samples for each finger) in it and the test set has 45 samples (15 samples for each finger). As the set labels imply, the training set is used to train the classification procedures and the test set is used to evaluate the performance of the classification procedures.

4.3 Data Analysis

Once we have collected the entire data set, we must identify the MES's that occurred during the flexion of the finger(s). The graph in Figure 4 shows sample MES produced by the first individual's flexor digitorum superficialis for the indicated movements.

Although these readings are from a specific muscle, the behavior of the other muscles is similar, with slight differences (i.e., different shapes for the movements and more or less

noise). To identify the MES's that occurred during the flexion of a finger, we keep a moving window of 7 data points. When the average of the data points in that window goes above the standard deviation for the entire raw data set, call this value α, we remove the next λ data points. This value λ is found by counting the number of data points beyond the starting point which go above α for each sample. These individual values are then averaged together and the resultant is rounded to the nearest power of two. The purpose of rounding the data is necessary so that it may be used by the FFT algorithm (described later). We then fully rectify the data (take the absolute value of) and transform the new data set from the time domain to the frequency domain. One of the most popular transformation is Fourier's transform and this is what we use on our entire data set.

4.4 Fourier's Transform

Fourier Analysis, developed by Jean-Baptiste - Joseph Fourier has played an important role in applied mathematics. A well known theorem in mathematics guarantees that a non-periodic function can be approximated to any desired degree by means of any class of periodic functions. WE use Fourier's transformation for approximating a non- periodic function by means of Fourier analysis. Fourier coefficients have the property that they are orthogonal, so that the coefficients may be determined independently of one another. The role of Fourier analysis is important from one other viewpoint. The signals received from the muscles are continuous functions in the time-domain. In time-domain signal classification is difficult mainly due to lack of an appropriate measure to differentiate between them. Fourier analysis transforms the signals from the time-domain to the frequency-domain, thereby making it easy to work with Fourier coefficients for the purpose of classification.

5 Classification Approaches.

The main purpose for collecting the data is to develop a procedure that can be used for classifying a signal for its intended action. As we described earlier, all three of the individuals were asked to move the index finger, the middle finger, and the ring finger and the signals generated were collected. Given the Fourier coefficient of a signal our goal is to find if we can detect whether the individual wanted to move the index finger, the middle finger, or the ring finger.

5.1 Statistical Analysis

We have performed several statistical analyses to the data. The principle component analysis was performed to evaluate the role of all 16 Fourier coefficients. The results are presented in Tables 13 and 14 for individuals 1 and 2 respectively. In both cases we note that it is sufficient to consider only first 12 principle coefficients. In other words we find that only 12 linear combinations of the 16 coefficients are sufficient for any other analysis; use of only these 12 newly defined variables will reduce the complexity of analysis without sacrificing

the accuracy of the results. We note that the principle components are distinct for both individuals, (see Tables 15 and 16). For example, the first principle component for the first individual is given by $(0.133one + 0.168two + \ldots - 0.202sixt)$ and for the second individual by $(0.312one + 0.407two + \ldots - 0.005sixt)$. where $one, two, \ldots, sixt$ denote the first, second, ..., sixteenth Fourier coefficients respectively. From these observations we conclude that a result obtained for the first individual is not applicable for the second individual with appropriate changes. This observation persists in all analyses, statistical as well as neural networks; no two individuals generate the same or similar myoelectric signals from a particular muscle for the same indented task. We have used all 16 Fourier coefficients in the rest of the analyses presented in this paper. Detailed analyses based upon smaller number of coefficients or their linear combinations will be discussed in another paper. Using all 16 coefficients we performed two types of statistical classification analyses; the quadratic discriminate analysis and the 3 nearest-neighbor discriminate analysis, for each individual separately.(See [11] for more details about these statistical procedures.) We have used the quadratic discriminate analysis because a preliminary test shows that the associated covariance matrices are statistically significantly different. Performance of these classification procedures are described in Tables 5, 6, 7, 8, 9, 10, 11, and 12. From these tables we immediately conclude that it is possible to identify the signal received from the ring finger accurately. The signals of the middle and the index fingers are not easy to distinguish, particularly that of the middle finger which are most often confused with the signals of the index finger.

6 Artificial Neural Network Analysis

We used three types of neural networks on the data sets collected for three individuals. These networks are Kohonen's "LVQ" network [7], the feed-forward backpropagation network, and the supervised growing-cell-structure network [10]. Each network has 16 inputs, which represent the transformed data and 1 output, which takes a Boolean value and represents an intended finger movement. Thus, each movement had a dedicated network to identify it. The architecture of each network, discussed below, is given in Table 4.

6.1 The Kohonen Network

With Kohonen's LVQ network we achieved better results than given by the statistical procedures. The training and test set results are presented in Tables 2 and 3, respectively. The average number of code book vectors chosen varied between 7 and 13 depending on the person and the movement. We found that 7 code book vectors gave generally good results. Like their statistical counterpart, these results are also inconsistent in the sense that the same network that gave good results for one individual and one finger gave poor performance for another finger.

6.2 The Feed-Forward Backpropagation Network

The actual implementation of the FFBP network that we have used is the "Aspirin/Migraines Neural Network Software" package by Leighton of MITRE Corporation. This package makes it easy to design a standard FFBP network. During the training of the feed forward back propagation network we varied the number of neurons in the middle layer from 3 to 8. We trained the network until a mean-square-error of 0.01 was obtained or a maximum of 90,000 epochs was reached. We found that a network with 8 nodes in the hidden layer did better on average than other sizes in most of the training and testing. The results are summarized in Tables 2 and 3, respectively.

6.3 The Growing-Cell-Structure Network

The main difficulty with the two previous paradigms is that they have static architecture's and adaptive parameters. A large amount of our time in researching these paradigms was in finding optimal architecture's and parameters. Ultimately, though, it would be impractical to hardwire the structure of the network, as this structure seems to be user and movement specific.

A promising new paradigm, proposed by Fritzke [10] entitled "supervised-growing-cell-structures" overcomes the difficulties of the two previous models. Instead of approaching the problem from the top down as both the FFBP and LVQ do, supervised growing-cell-structures (SGCS) evolves a network from the bottom up, growing it into the problem domain. The only static features of the SGCS algorithm are the adaptive parameters. These parameters seem to play a role in the networks size and speed of convergence, but not the final accuracy. However, by loosely estimating the density of the entire sample space and the sample space of each class, we were able to estimate exceptionally good values of the parameters for the SGCS network. We slightly modified the insertion rules of the SGCS algorithm to produce as small a network as possible. The networks generated by the SGCS had, on average, 12 neurons. The accuracy of the network averaged about 96% after 30 training epochs.

Tables 2 and 3 summarize the performance of GCS network.

7 Conclusion

Summary results in Tables 2 and 3 show that all three networks are able to learn the training samples well. All achieve better than 90% accuracy in corroctly distinguishing the signals. In general the performance of Kohonen's LVQ is the worst. This is not surprising because the 3 nearest-neighbor approach was unable to give a good performance, and as noted earlier the signals from the middle and the index finger are apparantly similar. In terms of the performance on the test set, a true measure of the performance of any classification procedure, Kohonen's LVQ is far below an acceptable level; it classifies the middle finger with merely 66% accuracy. The FFBP and SGCS networks are able to give good test set performance; Overall accuracy of the GCS network is 95%, 97%, and 96% for subjects 0, 1, and 2 respectively,

whereas, the overall accuracy of the FFBP is 88.15%, 91.85%, and 88.14%, respectively. The superior performance of the growing cell structure is evident for these sets of experiments. We plan to revalidate these results for larger test sets and with other subjects, including the amputees.

The SGCS network is easier to use, more dynamic and more accurate than previous attempts. In addition, the time taken for SGCS to converge was several orders of magnitudes smaller than the back propagation network and comparable to LVQ.

By using the SGCS algorithm coupled with our simple statistical techniques, we were able to achieve excellent results with very little subjective interference. The only part of our method that requires human interference is in the judgment of the preamplifiers. The rest of the procedure is completely automatic. An operator need only collect the data and pass it on to our system. From that point on, the analysis of the data, the architecture of the network and adaptive he network are all determined automatically. With respect to the design of an artificial limb this automatic process is a very desirable characteristics. There would not be any need for custom built hardware for a specific user of the system. The operator of the system would not need to learn about a specific neural network type and the heuristics required for training it. Thus, there is a very low learning curve for using our method in practice.

References

[1] Sears, H. and J. Shaperman [1991], "Proportional Myoelectric Hand Control: An Evaluation," American Journal of Physical Medicine and Rehabilitation, 70, pp. 20-28

[2] Kelly, F, P. Parker and R. Scott [1990], "The Application of Neural Networks to Myoelectric Signal Analysis: A Preliminary Study," IEEE Transactions On Biomedical Engineering, 37, pp. 221-230

[3] Basmajian , J.V., and C. J. Deluca, Muscles Alive, Their Function Revealed by Electromyography. Baltimore, MD: Williams and Wilkins, 1985

[4] Graupe, D., J. Salahi and D. Zhang, "Stochastic analysis of myoelectric temporal signatures for multifunctional single-site activation of prosthesis and orthoses," Journal of Biomedical Engineering, 7, pp. 18-29

[5] Gray, H., Gray's Anatomy, 36th ed, (P. Williams and R. Warwick ed.). Philadelphia: W.B. Saunders Company, 1980, pg. 576.

[6] Melzack, R. [1992], "Phantom Limbs," Scientific American, April, pp. 120-126

[7] Kohonen, T. [1982], "Self-organized formation of topologically correct feature maps," Biological Cybernetics, 43, pp. 59-69

[8] Hudgins, B. [1993], "A New strategy for Myoelectric Control," IEEE Transactions on Biomedical Engineering, 40, pp. 82-93

[9] Rummelhart, D.E. and J.L. McClelland, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge, MA: MIT Press, 1986, pp. 445-459

[10] Fritzke, B. [1993], "Kohonen Feature Maps and Growing Cell Structures - a Performance Comparison," Advances in Neural Information Processing Systems, 5, pp. 123-130

[11] Johnson, Richard A., and Dean W. Wichern,[1982], Applied Multivariate Statistical Analysis, Prentice Hall, Englewood Cliffs, New Jersey.

[12] Jacobsen, S.C., E.K. Iversen, D.F. Knutti, R.T. Johnson & K.B. Biggers [1986], "Design of the Utah/M.I.T. Dextrous Hand."

8 Appendix

Supervised Growing Cell Structure					
Subject 0	Accuracy	Subject 1	Accuracy	Subject 2	Accuracy
Ring	97.00	Ring	99.00	Ring	98.33
Middle	94.33	Middle	97.00	Middle	97.00
Index	97.33	Index	98.33	Index	97.33
Supervised Growing Cell Structure					
Subject 0	Accuracy	Subject 1	Accuracy	Subject 2	Accuracy
Ring	97.00	Ring	97.67	Ring	97.00
Middle	89.33	Middle	93.00	Middle	89.33
Index	91.67	Index	94.00	Index	86.00
Supervised Growing Cell Structure					
Subject 0	Accuracy	Subject 1	Accuracy	Subject 2	Accuracy
Ring	99.00	Ring	99.00	Ring	99.00
Middle	93.00	Middle	89.33	Middle	93.00
Index	99.00	Index	99.00	Index	95.00

Table 2: Neural Networks Performance on the Training Set

Supervised Growing Cell Structure					
Subject 0	Accuracy	Subject 1	Accuracy	Subject 2	Accuracy
Ring	95.56	Ring	100.00	Ring	97.78
Middle	93.33	Middle	95.56	Middle	95.56
Index	97.78	Index	95.56	Index	95.56
Learning Vector Quantization					
Subject 0	Accuracy	Subject 1	Accuracy	Subject 2	Accuracy
Ring	93.33	Ring	100.00	Ring	97.78
Middle	66.67	Middle	64.44	Middle	66.37
Index	93.33	Index	68.89	Index	82.79
Feed-forward Back propagation Network					
Subject 0	Accuracy	Subject 1	Accuracy	Subject 2	Accuracy
Ring	86.67	Ring	100.00	Ring	86.67
Middle	82.22	Middle	80.00	Middle	84.44
Index	95.56	Index	95.56	Index	93.33

Table 3: Neural Networks Performance on the Test Set

Supervised Growing Cell Structure

Subject 0	Training Cycles	No. of Nodes	Subject 1	Training Cycles	No. of Nodes	Subject 2	Training Cycles	No. of Nodes
Ring	30	13	Ring	30	13	Ring	30	13
Middle	30	11	Middle	30	11	Middle	30	11
Index	30	13	Index	30	11	Index	30	11

Learning Vector Quantization

Subject 0	Training Cycles	No. of Nodes	Subject 1	Training Cycles	No. of Nodes	Subject 2	Training Cycles	No. of Nodes
Ring	300	7	Ring	300	7	Ring	300	7
Middle	300	7	Middle	300	7	Middle	300	7
Index	300	7	Index	300	7	Index	300	7

Feed-forward Back propagation Network

Subject 0	Training Cycles	Nodes in Hid. Layer	Subject 1	Training Cycles	Nodes in Hid. Layer	Subject 2	Training Cycles	Nodes in Hid. Layer
Ring	1804	8	Ring	3039	8	Ring	2422	8
Middle	6463	8	Middle	4870	8	Middle	5668	8
Index	6777	8	Index	2602	8	Index	4690	8

Table 4: Architecture of the Neural Networks

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	100	15	0
	86.96	13.04	0.00
2	0	93	21
	0.00	80.87	18.26
3	0	0	115
	0.00	0.00	100.00
Total	100	108	136
Percent	28.99	31.30	39.42

Table 5: The performance of the 3 nearest neighbor classification procedure on the training data of the first subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	12	3	0
	80.00	20.00	0.00
2	0	14	0
	0.00	93.33	0.00
3	0	0	15
	0.00	0.00	100.00
Total	12	17	15

Table 6: The performance of the 3 nearest neighbor classification procedure on the test data of the first subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	112	3	0
	97.39	2.61	0.00
2	2	107	6
	1.74	93.04	5.22
3	0	3	112
	0.00	2.61	97.39
Total	114	113	118

Table 7: The performance of the quadratic discriminant procedure on the training data of the first subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	100	15	0
	86.95	13.04	0.00
2	0	93	21
	0.00	80.87	18.26
3	0	0	115
	0.00	0.00	100.00
Total	100	118	136

Table 8: The performance of the quadratic discriminant procedure on the test data of the first subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	100	15	0
	95.00	4.00	1.00
2	89	11	11
	0.00	89.00	11.00
3	0	8	104
	0.00	8.00	92.00
Total	100	108	136
Percent	28.99	31.30	39.42

Table 9: Performance of the 3-nearest-neighbor classification procedure on the training data of the second subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	15	0	0
	100.00	0.00	0.00
2	11	1	0
	73.33	6.67	0.00
3	0	7	6
	0.00	50.00	42.86
Total	12	17	15

Table 10: Performance of the 3-nearest-neighbor classification procedure on the test data of the second subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	99	1	0
	99.00	1.00	0.00
2	1	76	23
	1.00	76.00	23.00
3	0	3	97
	0.00	3.00	97.00
Total	100	80	120
Percent	33.04	32.75	34.20

Table 11: Performance of the quadratic discriminant procedure on the training data of the second subject

	Number of Observations and Percent classified into Class		
From Classs	1	2	3
1	15	0	0
	100.00	0.00	0.00
2	15	0	0
	100.00	00.00	0.00
3	14	0	0
	100.00	0.00	100.00
Total	14	1	15
Percent	31.11	35.56	33.33

Table 12: Performance of the quadratic discriminant procedure on the test data of the second subject

	Eigenvalue	Diffence	Proportion	Cumulative
PRIN1	4.13223	1.48381	0.258265	0.25826
PRIN2	2.64843	0.78963	0.165527	0.42379
PRIN3	1.85880	0.26271	0.116175	0.53997
PRIN4	1.59610	0.15891	0.099756	0.63972
PRIN5	1.43719	0.27649	0.089824	0.72955
PRIN6	1.16069	0.38561	0.072543	0.80209
PRIN7	0.77508	0.07938	0.048443	0.85053
PRIN8	0.69570	0.09866	0.043482	0.89401
PRIN9	0.59705	0.18052	0.037315	0.93133
PRIN10	0.41653	0.06575	0.026033	0.95736
PRIN11	0.35077	0.20656	0.021923	0.97929
PRIN12	0.14421	0.06334	0.009013	0.98830
PRIN13	0.08087	0.02966	0.005054	0.99335
PRIN14	0.05121	0.01531	0.003201	0.99655
PRIN15	0.03590	0.01667	0.002244	0.99880
PRIN16	0.01924	0.001202	1.00000	1.00000

Table 13: This table shows the contributions of principle components for the first subject

	Eigenvalue			
	Eigenvalue	Diffence	Proportion	Cumulative
PRIN1	4.88723	3.12543	0.305452	0.30545
PRIN2	1.76180	0.43414	0.110112	0.41556
PRIN3	1.32766	0.14269	0.082979	0.49854
PRIN4	1.18497	0.03767	0.074061	0.57260
PRIN5	1.14730	0.19056	0.071706	0.64431
PRIN6	0.95674	0.04727	0.059796	0.70411
PRIN7	0.90948	0.07352	0.056842	0.76095
PRIN8	0.83596	0.12983	0.052247	0.81320
PRIN9	0.70612	0.08350	0.044133	0.85733
PRIN10	0.62262	0.13660	0.038914	0.89624
PRIN11	0.48603	0.01166	0.030377	0.92662
PRIN12	0.47437	0.08350	0.029648	0.95627
PRIN13	0.39088	0.22208	0.024430	0.98070
PRIN14	0.16879	0.09230	0.010549	0.99125
PRIN15	0.07649	0.01294	0.004780	0.99603
PRIN16	0.06355	0.003972	1.00000	1.00000

Table 14: This table shows the contributions of principle components for the second subject

	Component			
	PRIN1	PRIN2	PRIN3	PRIN4
ONE	0.133517	0.391984	-0.114964	0.075709
TWO	0.168700	0.452274	0.179171	-0.270731
THREE	0.305212	-0.082277	-0.023586	-0.018082
FOUR	-0.138472	0.455474	-0.092862	0.332805
FIVE	0.261620	0.385817	0.193080	-0.308420
SIX	-0.291837	0.069541	-0.132131	0.466962
SEVEN	0.206292	0.286032	-0.300071	0.269862
EIGHT	-0.191683	0.221129	0.399957	-0.024205
NINE	0.273489	0.011392	-0.114363	0.250378
TEN	-0.259305	0.289276	0.079329	0.093446
ELEV	0.331744	0.015168	0.382376	0.134842
TWEL	-0.270589	0.193101	-0.378105	-0.203664
THIR	0.299223	0.045764	0.042744	0.316685
FOURT	-0.267017	0.136507	-0.120736	-0.352781
FIFT	0.285464	0.027700	-0.393210	-0.236502
SIXT	-0.202099	0.018564	0.401485	0.107420

Table 15: This table shows the linear combination that gives first four principle components for the second subject

	Component			
	PRIN1	PRIN2	PRIN3	PRIN4
ONE	0.311973	0.144342	-0.177358	-0.112386
TWO	0.407449	-0.136848	-0.050824	0.043374
THREE	0.402663	0.061577	-0.037973	-0.000387
FOUR	-0.411069	-0.007371	-0.045541	-0.167813
FIVE	0.403087	-0.166039	0.008230	-0.016807
SIX	-0.123184	-0.087077	0.342982	0.526608
SEVEN	-0.194639	0.380749	0.055508	-0.215998
EIGHT	-0.204597	-0.487601	-0.223899	-0.018598
NINE	0.315406	0.068736	-0.172473	0.218996
TEN	0.046246	-0.226235	0.390144	-0.385950
ELEV	-0.098436	0.129361	-0.008676	0.095647
TWEL	0.136369	0.478689	0.360969	-0.126722
THIR	-0.070479	0.266506	0.136724	0.556379
FOURT	0.063146	-0.309088	0.532393	0.123502
FIFT	-0.096240	0.112679	-0.408566	0.205399
SIXT	-0.004852	0.240155	0.100671	-0.208818

Table 16: This table shows the linear combination that gives first four principle components for the second subject