[4] ELENA. [Online]website: http://www.dice.ucl.ac.be/neural-nets/Research/Projects/ELENA/elena.htm
[5] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugen.*, pt. II, pp. 179–188.
[6] R. N. Goldman and J. S. Weinberg, *Statistics: An Introduction*. Englewood Cliffs, NJ: Prentice-Hall.
[7] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 23, pp. 665–685, May/June 1993.
[8] T. K. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. New York: Springer-Verlag.
[9] K. J. Lang and M. J. Witbrock, "Learning to tell two spirals apart," in *Proc. Connectionist Models Summer School*, 1988, pp. 52–59.
[10] C. T. Lin and C. S. G. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. Comput.*, vol. 40, pp. 1320–1336, Dec. 1991.
[11] ——, *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall.
[12] C. J. Lin and C. T. Lin, "An ART-based fuzzy adaptive learning control network," *IEEE Trans. Fuzzy Systs.*, vol. 5, no. 4, pp. 477–496.
[13] D. Nauck, F. Klawonn, and R. Kruse, *Foundations of Neuro-Fuzzy Systems*. New York: Wiley, 1997.
[14] C. Quek and R. W. Zhou, "POPFNN-TVR: A Pseudo Outer-Product based Neural Fuzzy Network," *Neural Netw.*, vol. 9, no. 9, pp. 1569–1581, 1996.
[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams *et al.*, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, vol. 1, ch. 8.
[16] G. K. Tan, "Feasibility of Predicting Congestion States With Neural Networks," M.S. dissertation, Sch. Civil Structural Eng., Nanyang Technological University, Singapore.
[17] W. L. Tung, "Fuzzy Neural Approximate Reasoning Systems," M.S. dissertation, Sch. Appl. Sci., Nanyang Technological University, Singapore.
[18] W. L. Tung and C. Quek, "A novel approach to the derivation of fuzzy membership functions using the falcon-MART architecture," *Pattern Recognit. Lett.*, vol. 9, no. 22, pp. 941–958.
[19] W. L. Tung, "A Generalized Framework for Fuzzy Neural Architecture," Sch. Comput. Eng., Nanyang Technological University, Singapore, Tech. Rep. ISL-TR/02-01.
[20] I. B. Turksen and Z. Zhong, "An approximate analogical reasoning scheme based on similarity measures and interval valued fuzzy sets," *Fuzzy Sets Systs.*, vol. 34, pp. 323–346, 1990.
[21] L. A. Zadeh, "Calculus of fuzzy restrictions," in *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*. New York: Academic, 1975, pp. 1–39.

A Hybrid Neural Network/Genetic Algorithm Approach to Optimizing Feature Extraction for Signal Classification

G. A. Rovithakis, M. Maniadakis, and M. Zervakis

***Abstract*—In this paper, a hybrid neural network/genetic algorithm technique is presented, aiming at designing a feature extractor that leads to highly separable classes in the feature space. The application upon which the system is built, is the identification of the state of human peripheral vascular tissue (i.e., normal, fibrous and calcified). The system is further tested on the classification of spectra measured from the cell nucleii in blood samples in order to distinguish normal cells from those affected by Acute Lymphoblastic Leukemia. As advantages of the proposed technique we may encounter the algorithmic nature of the design procedure, the optimized classification results and the fact that the system performance is less dependent on the classifier type to be used.**

***Index Terms*—Classification, feature extraction, genetic algorithms, neural networks.**

I. Introduction

Feature extraction is a central task in any process that involves classification. It refers to collecting those characteristics that efficiently and uniquely describe an object. An ideal feature extractor would yield a representation that makes the job of the classifier trivial. In the literature there exists a vast number of reported feature extraction techniques that, depending on the application, rely on image analysis, statistics, system modeling etc [14].

In a recently published work [7], we have proposed a neural network architecture to efficiently identify the state (i.e., normal, fibrous and calcified) of human peripheral vascular tissue, by processing the information obtained via laser induced fluorescence spectroscopy. The feature extractor was developed as a nonlinear adaptive filter based on a High Order Neural Network (HONN), equipped with a stable learning law for determining its weights. Input to the filter was the fluorescence spectrum and the HONN was designed to approximate it. Appropriately defined norms of the weights and of the approximation error were selected to serve as features. Information regarding the use of HONNs as basic modules for the construction of dynamic system identifiers and of controllers for highly uncertain systems may be found in [8]–[12].

It has been stated in [7] that different values on the sigmoid function parameters that appear in the HONN structure correspond to different feature space topologies. Hence, the existence of a set of values that maximizes the separability among the classes is evident.

One of the major drawbacks of [7] is the lack of an automated process for selecting values for the aforementioned parameters, or stated otherwise for the design of the HONN-based feature extractor. In [7] the design employs a time consuming trial and error procedure. Since the number of parameters is generally high, following such a technique requires a certain level of experience and in any case it leads to classification results that may be far from optimal.

In this paper we propose the introduction of Genetic Algorithms (GAs) into the feature extraction process. In other words, we equip the HONN structure with a GA to force the resulting classes in the feature space to be as separable as possible, thus providing an integrated

Manuscript received May 22, 2002; revised September 4, 2002. This paper was recommended by Associate Editor P. Bhattacharya.

G. A. Rovithakis is with the Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, 54006 Thessaloniki, Greece.

M. Maniadakis is with the Foundation for Research and Technology-Hellas, Institute of Computer Science, Computer Vision and Robotics Laboratory, Science and Technology Park of Crete, 71110 Crete, Greece.

M. Zervakis is with the Department of Electronic and Computer Engineering Technical University of Crete, 73100 Crete, Greece.

Digital Object Identifier 10.1109/TSMCB.2003.811293

design solution. GAs may be seen as an adaptive technique that can efficiently search the complex space of plausible parameter combinations and locate the values which yield near optimal results. With respect to classical optimization techniques they are efficient and ideal for parallel implementation. Moreover, they do not require a mathematical description of the searching surface and are independent of the learning algorithm used to train the HONN-based feature extractor. Unfortunately, they involve time consuming procedures and their complexity grows significantly with the dimension of the search space (i.e., the number of the parameters to be determined). However, this is not a hard constraint in our case, since the feature extractor design process is performed off-line.

Many attempts have been reported in the literature to employ GAs for the construction and/or training of neural networks [18]–[24]. A survey on using evolutionary algorithms for classification problems can be found in [25]. Works on blending GAs for classification problems are toward classifier design for optimizing their performance (e.g., minimizing the classification success ratio), given the feature space topology. In our work, the GA searches for those values in the HONN-parameter space that correspond to low variance, minimum overlapping classes, whose centers are distant in the feature space. In contrast to other works, our approach aims at modifying the feature extraction process, as to directly affect the feature space topology. To our knowledge, feature extractor optimization using GAs has not been previously reported.

The advantages of the proposed approach are

- the design of the feature extractor is performed in an algorithmic way, thus reducing significantly the designer's effort, while optimizing the classification success rate;
- the system performance becomes less dependent on the classifier type. Hence, we don't have to incorporate the classifier into the feature extractor design process, thus resulting in a faster design procedure.

The paper is organized as follows. In Section II we overview the GAs, the structure and the properties of the HONNs and we review the diagnostic system architecture used to peripheral vascular tissue classification. Section III, presents the proposed design algorithm. The method is applied on two classification problems and the results are presented in Section IV. Finally, we conclude in Section V.

II. Preliminaries and Problem Definition

A. Genetic Algorithms Overview

Based on the Darwinian survival of the fittest, genetic algorithms are global search and optimization techniques. Inherently parallel, they operate on a set of candidate solutions that formulate a population, whose size is maintained constant. Each solution is usually coded as a binary string called a chromosome[1] . The chromosomes of the initial population are randomly generated. Each iteration of the GA called a generation, involves three stages:

- the current population is first evaluated and ranked with the aid of a fitness function;
- chromosomes that possess the highest fitness values are probabilistically selected to construct the "parents" pool;
- from the selected "parents," the GA reproduces "children" performing the genetic operations of crossover and mutation.

The GA terminates when an acceptable solution is found, or when a predetermined number of generations is reached.

Considering the genetic operations, crossover exchanges information between two chromosomes. After randomly selecting the cut position which separates each parent chromosome into two regions, the parents exchange their right regions to produce two new, called children, keeping unaltered the chromosome length. A crossover probability controls the application of the aforementioned operation. On the other hand, mutation replaces a randomly selected bit of a chromosome with its complement. Similar to crossover, a mutation probability controls its application. However, mutation has significantly lower probability when compared with crossover.

Owing to their inherently parallel structure, GA simultaneously search all candidate solutions defined in the population having as the only feedback their fitness evaluation. In this way, multiple spaces are investigated in parallel. Moreover, since GA are global search methods, are not easily entrapped in local minima, while typically their convergence is slow.

GAs have been employed to successfully address a plethora of practical optimization problems [13]. For more information on GAs the interested reader is referred to [15]–[17].

B. High Order Neural Networks (HONNs)

HONNs are fully interconnected nets, containing high order connections of sigmoid functions in their neurons. If we define as x, y its input and output respectively, with $x \in \Re^n$ and $y \in \Re^q$ the input-output representation of a HONN is given by

$$y = \hat{W} S(x) \tag{2.1}$$

where $\hat{W}$ is a $n \times q$ matrix of adjustable synaptic weights and $S(x)$ is a q-dimensional vector $S_i(x)$, $i = 1, 2, \ldots, q$ of the form

$$S_i(x) = \prod_{j \in I_i} [s(x_j)]^{d_j(i)} \tag{2.2}$$

where I_i, $i = 1, 2, \ldots, q$ are a collections of q not-ordered subsets of $\{1, 2, \ldots, n\}$ and $d_j(i)$ are nonnegative integers. In (2.2) $s(x_j)$ is a monotone increasing, smooth, function, which is usually represented by sigmoidals of the form

$$s(x_j) = \frac{\mu}{1 + e^{-l x_j}} + \lambda \tag{2.3}$$

for all $j = 1, 2, \ldots, n$. In (2.3), the parameters μ, l represent the bound and maximum slope of sigmoid's curvature while λ is the vertical shift.

For the HONN model described above, it can be seen [10], that there exist integers q, $d_j(i)$ and optimal weight values $W^\star$ such that for any smooth but unknown function $f(x)$ and $\forall \varepsilon \geq 0$, $|f(x) - W^\star S(x)| \leq \varepsilon$, $\forall x \in \mathcal{M}$ where $\mathcal{M} \subset \Re^n$ is a compact region. In other words, for sufficient high order terms, there exist weight values $W^\star$ such that the HONN structure $W^\star S(x)$, can approximate $f(x)$ to any degree of accuracy, in a compact domain.

Remark 1: Observe that HONNs possess a linear-in-the-weights property. Depending on the form of its regressor terms, (2.1) may represent various well known neural network structures. Indicatively, we mention RBFs [3]–[5], Shifted Sigmoidal neural networks [1], [2] and the CMAC network [6].

C. Diagnostic System Architecture

In a recently published work [7], we have proposed a diagnostic system which was comprised of two modules. The first was a nonlinear adaptive filter implemented by using HONNs as the basic building block. It receives as input the entire distribution of the recorded fluorescence spectra of the tissue sample and outputs feature values which encode its state. The second module was a classifier whose purpose was to categorize the received feature vector to one out of three predetermined classes (i.e., normal, fibrous, and calcified). The diagnostic system architecture is pictured in Fig. 1.

It has been shown in [7] that the nonlinear adaptive filter

$$\dot{z} = -\alpha z + f(x) - W^T S(x), \quad \alpha > 0 \tag{2.4}$$

equipped with the update law

$$\dot{W} = -\gamma W + z S(x), \quad \gamma > 0 \tag{2.5}$$

[1]Real-coded chromosomes are also possible.

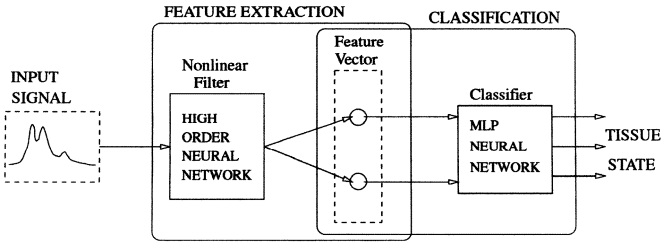

Fig. 1. Diagnostic system architecture.

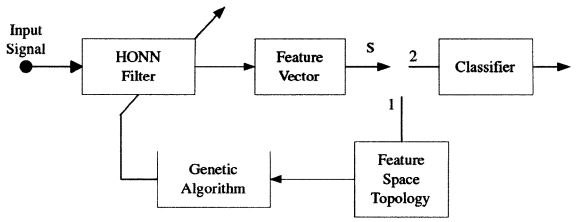

Fig. 2. Proposed feature extraction module parameters selection process.

guarantees the uniform ultimate boundedness of its output z with respect to the arbitrarily small set

$$\mathcal{Z} = \left\{ z \in \Re : |z| \leq \frac{\epsilon}{\alpha} + \frac{1}{2}\sqrt{\left(\frac{\epsilon}{\alpha}\right)^2 + \frac{2\gamma|W^*|^2}{\alpha}} \right\} \tag{2.6}$$

as well as the boundedness of the HONN weights W, $\forall x \geq 0$. In the aforementioned relations, $y = f(x)$ is the recorded fluorescence spectrum, with $y \geq 0$ the fluorescence intensity and $x \geq 0$ the wavelength, α, γ are design constants and $\epsilon \geq 0$ is an unknown but small bound on the HONN reconstruction error. The extracted feature vector was defined as

$$V = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} \int_0^x \left(\sum_{i=1}^{q} w_i^2(\tau)\right) d\tau \\ \int_0^x e^2(\tau)\, d\tau \end{bmatrix}. \tag{2.7}$$

In (2.7), $w_i, i = 1, 2, \ldots, q$ is the ith HONN weight, while e is defined as $e = y - \hat{y}$.

The obtained feature vectors were classified to the three classes (i.e., normal, fibrous and calcified) with the aid of an appropriately constructed Multilayer Perceptron (MLP) neural network.

III. Diagnostic System Parameter Selection

The major drawback in designing a diagnostic system as in [7] is the requirement for manually selecting a great number parameters. More precisely, the feature extraction module includes

- structural parameters
  - HONN order;
  - sigmoid function parameters μ, l, λ;
  - parameter $\alpha > 0$ that appears in the nonlinear adaptive filter (2.4);
  - parameter $\gamma > 0$ that appears in the weights update law (2.5).
- learning parameters (i.e., HONN weights W for which (2.5) serves as selection algorithm);
- GA operational parameters (as defined in Section III-A).

Similarly, the classification module includes

- structural parameters (i.e., parameters necessary for building the classifier);
- learning parameters required for the classifier learning process.

For the case of using a MLP neural net classifier, the sigmoid function parameters, the required number of layers, as well as the number of neurons per layer, are all structural parameters. The learning rate that appears in back-propagation learning algorithm and the total number of training iterations, are learning parameters.

Selecting all these parameters through a trial and error procedure is very time consuming and requires vast engineering experience. Moreover, the final result (i.e., classifier's success rate), may be far from optimal. Obviously, an automated design process aiming at optimizing the classifier's decision, will serve as a significant addition to the already proposed diagnostic system.

The experience obtained from [7] indicates that the feature space topology may be significantly altered by differently selecting the feature extraction module parameters. Hence, there exists a set of values that maximizes the separability among the three classes (i.e., normal, fibrous, and calcified). Such a knowledge will practically reduce the designer's effort in training the classifier, while optimizing its performance. Another positive side-effect is that the diagnostic system performance becomes less dependent on the classifier type to be used. As a consequence, we don't have to engage the classifier into the feature extraction module parameter selection process. In this way a significant speed up can be achieved. Unfortunately, since the parameters appear nonlinearly in the feature extraction module structure, determining their appropriate values is not a straightforward task.

In order to facilitate the design of such a system and automate the process of parameter selection, we propose the use of GAs embedded into the feature extraction module.In particular, we use GAs for the selection of the structural parameters of the feature extractor. Our approach aims to overcome the problems introduced by the nonlinearity in the parameters property, which implies an infinite set of different parameter combinations that lead to the same feature space topology. The proposed design procedure is pictured in Fig. 2.

To enable the feature extraction module parameter selection process, the switch S that appears in Fig. 2 is set to position 1. After the parameters have been determined, the switch is set to position 2, to proceed to classifier construction and ultimately, to diagnostic system operation.

A. Genetic Algorithm Description

It becomes apparent that the GA plays a key role in the parameter selection process. In Fig. 3 we can visualize the GA in block diagram form.

Notice the circular topology of the algorithm. Each cycle is called a generation. In what follows we will describe in more detail the basic ingredients of the GA.

First, the number of chromosomes is specified. This number remains constant for all future generations throughout the termination of the genetic process. The chromosomes are real-valued and represent the sigmoid function parameters μ, λ, l as well as the gains α, γ that appear in (2.4) and (2.5) respectively. Each parameter in the chromosome forms a gene. All genes in a chromosome are initialized randomly from values that belong in pre-specified intervals. In this way an initial population is established. Each chromosome corresponds to a specific HONN structure with unspecified weight values. These weights are determined in the HONN training phase that utilizes (2.5).

In the sequel, all chromosomes are evaluated with the aid of a properly selected fitness function. We aim at highly separable classes in the feature space. Two classes i, j are called highly separable if the following properties hold:

P_1: The intra-distance between any two features in a class, is minimum.

P_2: The inter-distance between any two classes i and j is maximum.

P_3: The overlap of class i and class j is minimum.

The intra-distance measures the compactness of a class. Hence, the area E_i each class occupies serves as a logical measure. Notice that the re-

quirement of having minimum intra-distance becomes significant as the number of classes grows, which further leads to improving the generalization of the procedure. To measure the inter-distance, we first determine the centers c_i, c_j of the classes i, j respectively, and then calculate the inter-distance as the Euclidean distance $\|c_i - c_j\|$. Satisfaction of P_1, P_2 and P_3 is equivalent of having low variance, minimum overlapping classes, whose centers are distant in the feature space.

Let v_{ij} denote the overlap of class i with class j. Minimizing the fitness function

$$F = \sum_{i,j\ i\neq j} \left[\left(\frac{E_i + E_j}{\|c_i - c_j\|^k} \right)^p + v_{ij} \right] \tag{3.1}$$

where p, k are some positive constants, obviously leads to the simultaneous satisfaction of the separability conditions $P_1 - P_3$, thus yielding highly separable classes in the feature space. Prior to calculating (3.1), the border of each class should be determined. Efficient polygonal approximation algorithms can be used for this purpose, as also explained in the examples.

After evaluation, each chromosome is ordered according to its fitness value. A percentage (e.g., the top 35%) are selected to serve as the candidate parents, thus forming the mating pool, while the rest are disregarded.

Since the chromosomes are real-valued, real-valued crossover, and mutation operations are applied. First, a probability $p_c \in [0, 1]$ for applying crossover is selected. Consequently, for each gene, a real number $r_{ci} \in [0, 1]$ is randomly selected. If $r_{ci} > p_c$ the parents exchange the values of their corresponding i-genes. Similar to the crossover, a probability $p_m \in [0, 1]$ for applying mutation is first defined. Consequently, a real number $r \in [0, 1]$ is randomly selected. If $r < p_m$ then the mutation operation is applied to a randomly selected gene of the chromosome. Let v be the value of the selected gene. A random perturbation is added on v according to the formula

$$v' = v + \Delta v = v + 0.5\bar{r}v \tag{3.2}$$

where $\bar{r} \in [-0.5, 0.5]$ is also randomly selected. In this way the real-valued mutation operation is performed.

According to the elitism operation, the best fit chromosome (i.e., the one with the highest fitness value) is copied to the next generation. Hence, the search is directed toward the currently best solution. Elitism is the last step of our search. In the sequel, a new population is established and a new generation begins. The algorithm terminates whenever a chromosome is found to possess a fitness value lower than an a priori defined threshold or whenever a pre-specified number of generations has been reached. The algorithm termination conditions are checked in the specifications verification phase, which is depicted in Fig. 3.

Summarizing our approach, we select all sructural parameters except HONN order using the GA (guidelines for the selection of HONN order are presented in [7]). The learning parameters of the HONN are determined through (2.5). The operational parameters p_m, p_c, p and k, as well as the stopping criteria (fitness threshold and maximum number of generations), are defined by the user through a trial and error procedure.

IV. Results

A. Example 1

The aforementioned diagnostic scheme was tested on 31 fibrous, 33 calcified and 31 normal tissue samples obtained from humans with the procedure described in [7]. Again a fifth order HONN-based nonlinear adaptive filter was implemented to perform feature extraction, but in this case its parameters were derived with the aid of a GA, aiming at improving the separability among the three different classes. More precisely, since we have a one-dimensional (1-D) input signal (i.e., the fluorescence spectrum), the following HONN structure was used:

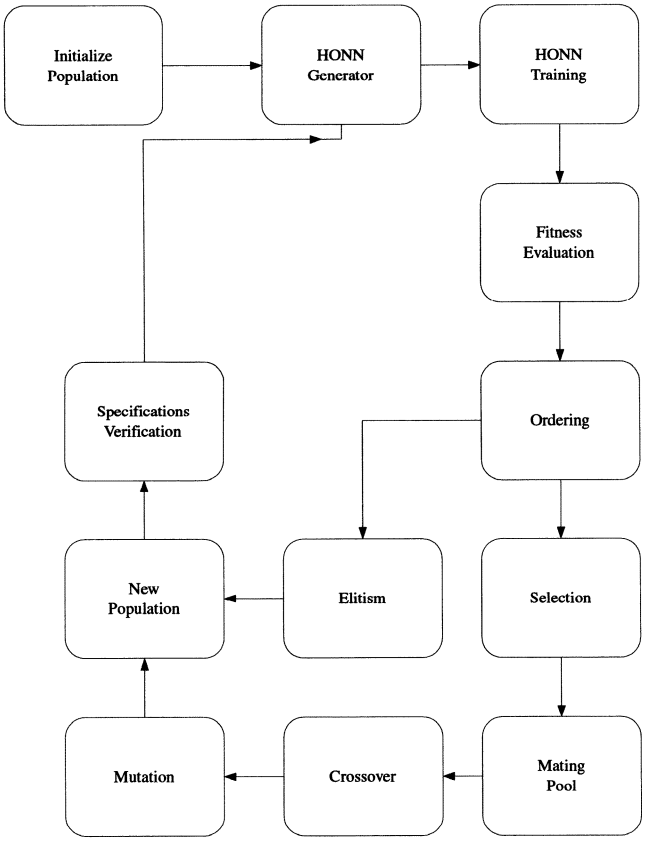

Fig. 3. Genetic algorithm in block diagram form.

$$y = \sum_{i=1}^{5} w_i s_i^i(x) \tag{4.1}$$

with

$$s_i(x) = \frac{\mu_i}{1 + e^{-l_i x}} + \lambda_i, \quad i = 1, 2, \ldots, 5. \tag{4.2}$$

After a trial and error procedure, we have selected $p_m = 0.35$, $p_c = 0.75$, $p = 0.5$ and $k = 0.2$. Although the value of p_m seems high, we observe that all genes in the chromosome are real-valued and thus the mutation operation actually performs (when activated) a very small perturbation. Hence, even though frequent, mutation doesn't introduce significant change in the gene value, thus preserving its purpose of being equivalent to fine-tuning. The GA terminates if the threshold of -1 in the fitness value is exceeded, or the maximum number of 20 generations is reached. In our case the second criterion was enabled, since a -1 fitness value was proved to be far from attainable, even for a large number of allowed generations. Table I presents the final values of the feature extraction module parameters as were provided by the GA.

The parameters α, γ that do not appear in Table I, were found to be $\alpha = 0.348\,119$, $\gamma = 0.083\,893$. Notice that the GA had to determine in total 17 real-valued parameters. The obtained two-dimensional (2-D) feature space is presented in Fig. 4.

For the evaluation of (3.1), the border of each class and their overlap should be determined. At first, the total feature space of dimension [0, 1] × [0, 1] is linearly transformed to a pixel space of dimension [0, 1000] × [0, 1000]. Then, the minimal convex hull that encloses the features of each class is estimated by using the Jarvis's march algorithm

TABLE I
FEATURE EXTRACTION MODULE PARAMETERS

Parameter	i=1	i=2	i=3	i=4	i=5
l_i	51.164447	139.170507	102.682577	101.138340	128.867458
μ_i	0.275674	0.4337635	0.357738	0.287027	0.862819
λ_i	0.075381	0.350902	0.597339	0.044160	0.444960

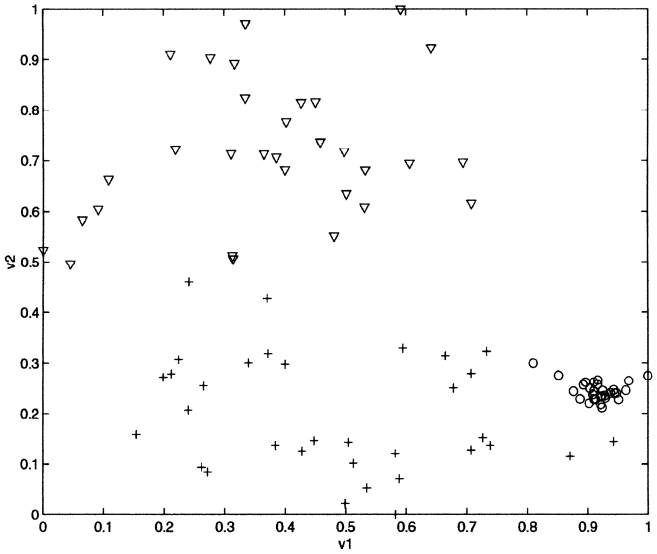

Fig. 4. Two-dimensional feature space as obtained from the feature extraction module. Triangles correspond to fibrous plaque, crosses correspond to calcified plaque and circles correspond to normal artery tissue.

TABLE II
CLASSIFICATION PERFORMANCE FOR DIFFERENT COMBINATIONS OF LEARNING AND TEST SETS

Test samples per class	Success ratio (Normal)	Success ratio (Fibrous)	Success ratio (Calcified)	Average
12	100%	100%	100%	100%
14	100%	100%	96.97%	98.99%

TABLE III
ESTIMATED CLASSIFICATION SUCCESS RATIO

Estimated success ratio (Normal)	Estimated success ratio (Fibrous)	Estimated success ratio (Calcified)	Average
100%	100%	93.97%	97.98%

[27]. Having estimated the border of each class, the number of pixels that are enclosed in each convex hull is computed and used as an estimation of the area covering the respective class. In the same way, by counting the number of common pixels within classes, we obtain a measure of their overlap.

Prior to classifier tuning, a training and a test set are built. From the 31 normal, 31 fibrous and 33 calcified feature vectors that constitute the classification set, we first exclude those that define the boarders among the different classes and from the rest we randomly select a pre-specified number of feature vectors from each class. These vectors formulate the test set and the remaining the training set. The latter is used to tune the classifier, while the former to measure its performance.

To investigate the robustness of the classifier, different combinations of training and test sets are used. Using up to 14 test samples per class, the average success ratio remains the same (100%) and slowly decreases from that point on. The results are presented in Table II.

Similar to [7], the Jack-knife (or leave one out) method [14] is used to obtain the classification estimates, which are summarized in Table III.

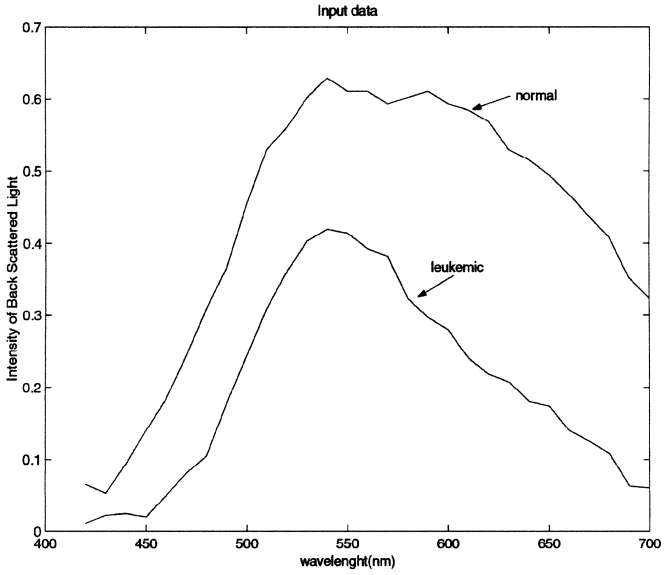

Fig. 5. Typical examples of input spectra.

TABLE IV
FEATURE EXTRACTOR PARAMETERS WHEN THE FEATURE VECTOR IS FORMED BY FEATURES 1 AND 2

Parameter	i=1	i=2	i=3	i=4	i=5
l_i	156.997797	155.727475	159.000824	165.369427	164.379711
μ_i	0.039598	0.264669	0.884548	0.880459	0.314988
λ_i	0.282632	0.177221	0.230811	0.005463	0.611011

TABLE V
FEATURE EXTRACTOR PARAMETERS WHEN THE FEATURE VECTOR IS FORMED BY FEATURES 1 AND 3

Parameter	i=1	i=2	i=3	i=4	i=5
l_i	148.991547	163.712271	162.921232	152.375866	169.372377
μ_i	0.323732	0.234931	0.113834	0.180484	0.158040
λ_i	0.438856	0.460067	0.084841	0.474685	0.814173

TABLE VI
FEATURE EXTRACTOR PARAMETERS WHEN THE FEATURE VECTOR IS FORMED BY FEATURES 1 AND 4

Parameter	i=1	i=2	i=3	i=4	i=5
l_i	165.552538	159.570299	164.610549	151.215857	173.860286
μ_i	0.806970	0.524055	0.039644	0.189489	0.16687
λ_i	0.076388	0.367168	0.014039	0.212836	0.057436

Comparing the outcome of the Jack-knife method obtained in the present work with the one derived in [7], we notice an increase by 3.26% in the average. This result comes to verify our initial gesture that the proposed hybrid neural network/genetic algorithm approach leads to performance improvement, while reducing the designer's effort. The increase reported is mainly attributed to the use of the GA to automatically manipulate the parameters of the HONN-based feature extractor for optimal performance. However, credit should be given to the different fitness function used (with respect to the one used in [7]), whose optimization is equivalent of having low variance, minimum overlapping classes whose centers are distant in the feature space. In addition, the increased performance is also partially attributed to the number of the feature extractor parameters controlled by the GA. In [7], all sigmoid functions used had identical parameters μ, λ, l. In this paper we have allowed each sigmoid function to have its own set of parameters as is obvious from Table I. The computational time required to achieve the aforementioned results was approximately 35 min/generation.

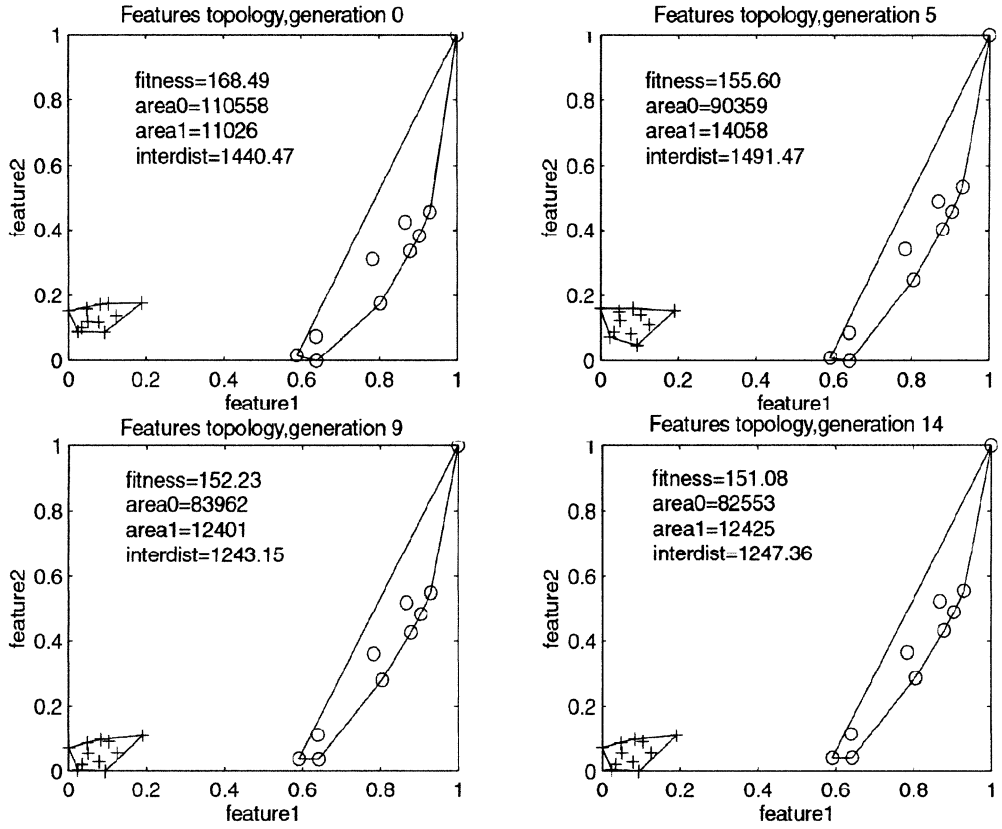

Fig. 6. Evolution of the 2-D feature space as obtained from the feature extraction module when features 1, 2 are combined. Crosses correspond to leukemic lymphoblasts and circles to normal lymphocytes.

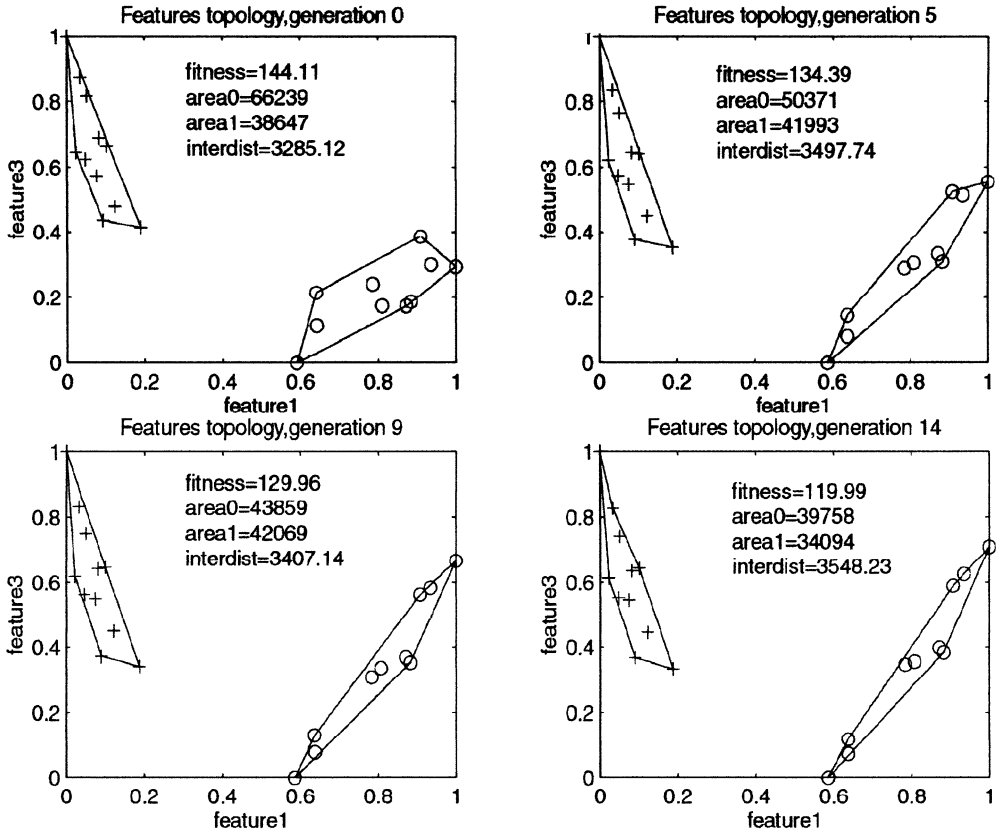

Fig. 7. Evolution of the 2-D feature space as obtained from the feature extraction module when features 1, 3 are combined. Crosses correspond to leukemic lymphoblasts and circles to normal lymphocytes.

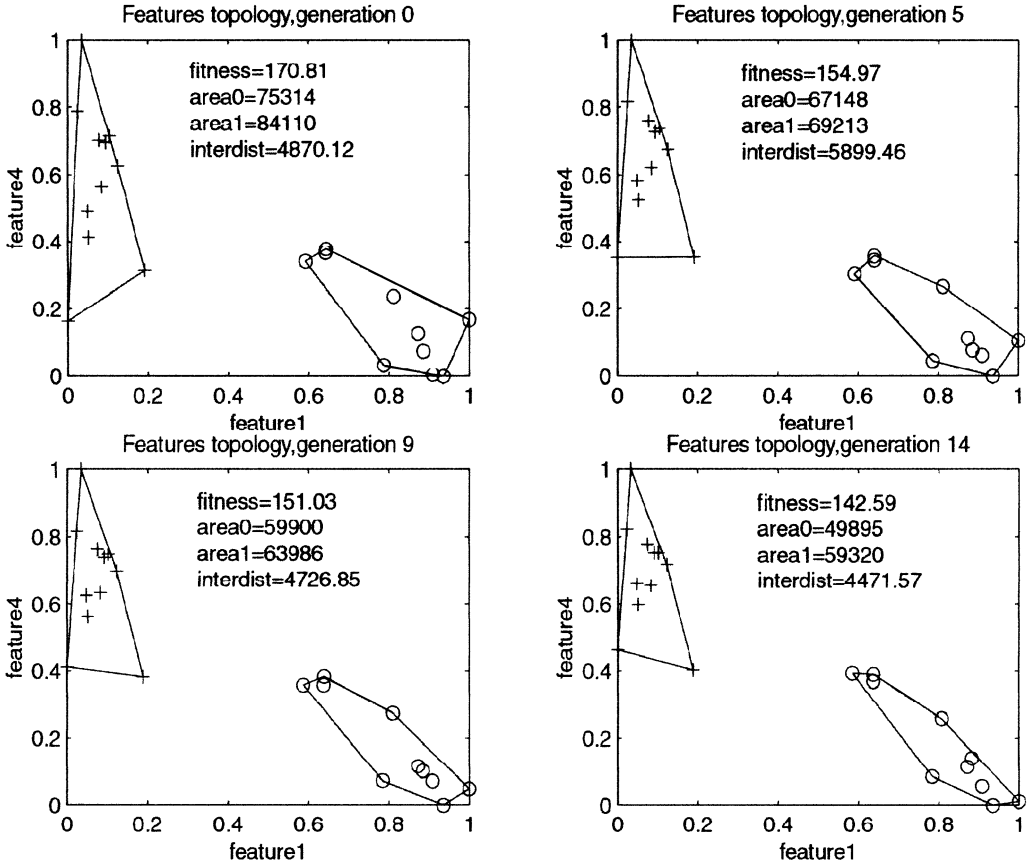

Fig. 8. Evolution of the 2-D feature space as obtained from the feature extraction module when features 1, 4 are combined. Crosses correspond to leukemic lymphoblasts and circles to normal lymphocytes.

B. Example 2

The classification architecture presented in previous Sections is now tested on the classification of spectra measured from the cell nucleii in blood samples in order to distinguish normal cells from those affected by Acute Lymphoblastic Leukemia. The image acquisition system is based on an all-optical imaging monochromator [26], integrated with a common optical microscope and a CCD camera. The system is capable of acquiring spectral images, with 2 nm tuning step, in the range of 400 nm-1000 nm. A stack of calibrated spectral images can be collected through synchronized spectral scanning. In this application, blood samples are scanned in the range of 420 nm up to 700 nm, with a step of 10 nm, providing 29 spectral images per sample. By scanning the spectral axis in specific spatial locations, several spectral responses are gathered from these data, with each spectrum representing the response of the nucleus of a blood cell. Each spectrum is thus composed of 29 points, or wavelengths in the range of [420 nm-700 nm]. Ten spectra were measured from the nucleus of Lymphocytes, which form a certain type of white blood cells and are commonly met in normal blood samples. Moreover, eleven spectra were measured from the nucleus of Lymphoblasts, which consistute the disease state of normal Lymphocytes and are met in blood from individuals with Acute Lymphoblastic Leukemia. Thus, out test sample for classification is composed of 10 normal and 11 leukemic spectral responses, each sampled over 29 equally spaced specral locations. A typical example of each type of these spectra is presented in Fig. 5.

Similarly to Example 1, a fifth order HONN (4.1), (4.2) was employed to construct the feature extractor, whose parameters were selected with the aid of the GA previously described, to improve the separability of the classes according to the definition provided at Section III.A.

Besides the separability issue, in this example we test if and how the feature space topology is altered when using different features. In the figures that follow, feature 1 and feature 2 are those defined in (2.7) and were also used in Example 1. Moreover, we used a third feature $v_3 = \max_t\{|e(t)|\}$ and a forth feature $v_4 = (\int_0^x |e(\tau)|\tau d\tau)/(\int_0^x |e(\tau)|d\tau)$ which denote the maximum and the temporal mean location of the approximation error respectively. Both of them are expected to increase as the HONN becomes less efficient in approximating the signal.

After a trial and error procedure, we selected $p_m = 0.35, p_c = 0.75$, $p = 0.5, k = 0.2$. In this example, the GA terminates whenever a maximum number of 14 generations has been reached. No further improvement was observed even though we allowed much more generations in our experimentations. Tables IV–VI give the final values of the feature extractor parameters that are better suited for Example 2 as they were provided by the GA, when different combinations of features are employed.

The parameters α, γ that do not appear in Table IV, were found to be $\alpha = 0.894\,444$, $\gamma = 0.296\,163$.

The parameters α, γ that do not appear in Table V, were found to be $\alpha = 0.585\,845$, $\gamma = 0.250\,110$.

The parameters α, γ that do not appear in Table VI, were found to be $\alpha = 0.592\,247$, $\gamma = 0.232\,779$.

Figs. 6–8, illustrate the evolution of the feature space topology for various combinations of features. Feature 1 was kept fixed. The algorithm achieved zero overlap, while making the classes more dense, for all feature combinations. In this example the combination of features 1 and 3 gave better results. However,the computation time required

was approximately 2.3, 2.80, 2.83 min per generation, when features (v_1, v_2), (v_1, v_3) and (v_1, v_4) were combined, respectively.

V. Conclusions

We have presented an algorithmic approach to designing a feature extractor, aiming at highly separable classes in the feature space. For that purpose we have used a HONN-based nonlinear adaptive filter equipped with a stable learning law for determining its weights and a GA for selecting mostly the sigmoid function parameters that appear in the HONN structure. The performance of the proposed system was tested on two different problems. Besides the algorithmic nature of the design process, the proposed methodology yields improved classification results, when compared to the ones obtained in [7]. Another advantage of the approach is that the system performance becomes less dependent on the classifier.

References

[1] N. E. Cotter, "The stone-weierstrass theorem and its applications to neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 290–295, 1990.

[2] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, 1989.

[3] M. M. Gupta and D. H. Rao, Eds., *Neuro-Control Systems: Theory and Applications*. New York: IEEE Press, 1994.

[4] *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds., Van Nostrand and Reinhold, New York, 1993.

[5] T. Poggio and F. Girosi, "Regularization algorithms for learning that are equivalent to multilayer networks," *Science*, vol. 247, pp. 978–982, 1990.

[6] S. H. Lane, D. A. Handelman, and J. J. Gelfand, "Theory and development of higher-order CMAC neural networks," *IEEE Contr. Syst. Mag.*, pp. 23–30, 1992.

[7] G. A. Rovithakis, M. Maniadakis, M. Zervakis, G. Filippidis, G. Zacharakis, A. N. Katsamouris, and T. G. Papazoglou, "Artificial neural networks for discriminating pathologic from normal peripheral vascular tissue," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 10, pp. 1088–1097, 2001.

[8] G. A. Rovithakis, "Tracking control of multi input affine nonlinear dynamical systems with unknown nonlinearities using dynamical neural networks," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 179–189, 1999.

[9] ——, "Robust neural adaptive stabilization of unknown systems with measurement noise," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, pp. 453–458, 1999.

[10] G. A. Rovithakis and M. A. Christodoulou, *Adaptive Control with Recurrent High-Order Neural Networks*. London, U.K.: Springer-Verlag, 2000.

[11] G. A. Rovithakis, "Performance of a neural adaptive tracking controller for multi-input nonlinear dynamical systemsin the presence of additive and multiplicative external disturbances," *IEEE Trans. Syst. Man Cybern. A*, vol. 30, pp. 720–730, 2000.

[12] J. Glosh and Y. Shin, "Efficient higher-order neural networks for classification and function approximation," *Int. J. Neural Syst.*, vol. 3, no. 4, pp. 323–350, 1992.

[13] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, pp. 66–72, July 1992.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Willey, 2001.

[15] K. DeJong, "Learning with genetic algorithms: An overview," *Mach. Learn.*, vol. 3, pp. 121–138, 1988.

[16] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.

[17] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. New York: IEEE Press, 1995.

[18] X. Yao, "A review of evolutionary artificial neural networks," *Int. J. Intell. Syst.*, vol. 4, pp. 203–222, 1993.

[19] B. A. Whitehead and T. D. Choate, "Evolving space-filling curves to distribute radial basis functions over an input space," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 15–23, 1994.

[20] J. R. McDonnell and D. Waagen, "Evolving recurrent perceptrons for time-series modeling," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 24–38, 1994.

[21] V. Maniezzo, "Genetic evolution of the topology and weight distribution of neural networks," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 39–53, 1994.

[22] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Trans. Neural Networks*, vol. 5, no. 1, pp. 54–65, 1994.

[23] H. Ishigami, T. Fukuda, T. Shibata, and F. Arai, "Structure optimization of fuzzy neural networks by genetic algorithm," *Fuzzy Sets Syst.*, vol. 71, pp. 257–264, 1995.

[24] D. Floreano and F. Mondada, "Evolution of homing navigation in a real mobile robot," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, no. 3, pp. 396–407, 1996.

[25] A. Freitas, "A survey of evolutionary algorithms for data mining," in *Evolutionary Computation*, A. Glosh and S. Tsutsui, Eds. Berlin, Germany: Springer-Verlag.

[26] C. Balas, "A novel optical imaging method for the early detection, quantitative grading and mapping of cancerous and precancerous lesions of cervix," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 1, pp. 96–104, 2001.

[27] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. New York: Springer-Verlag, 1985.

Integral Variable Structure Control of Nonlinear System Using a CMAC Neural Network Learning Approach

Chin-Pao Hung

***Abstract*—This work presents a novel integral variable structure control (IVSC) that combines a cerebellar model articulation controller (CMAC) neural network and a soft supervisor controller for use in designing single-input single-output (SISO) nonlinear system. Based on the Lyapunov theorem, the soft supervisor controller is designed to guarantee the global stability of the system. The CMAC neural network is used to perform the equivalent control on IVSC, using a real-time learning algorithm. The proposed IVSC control scheme alleviates the dependency on system parameters and eliminates the chattering of the control signal through an efficient learning scheme. The CMAC-based IVSC (CIVSC) scheme is proven to be globally stable inasmuch all signals involved are bounded and the tracking error converges to zero. A numerical simulation demonstrates the effectiveness and robustness of the proposed controller.**

***Index Terms*—CMAC, integral variable structure control, learning control, neural network control, nonlinear system, sliding mode.**

I. Introduction

In 1991, Chern presented the integral variable structure control (IVSC) to solve the steady-state error problem and enhance the robustness of the traditional variable structure control (VSC) [1]. Chern then applied this control scheme to a robot manipulator [2], brushless dc servo system [3], [4], [8], dc motor [5], and induction motor [6], [9], [11] to demonstrate the feasibility of IVSC. Related research has also proposed a method of designing IVSC [7]. Other applications include the engine system [10], the uninterruptible popwer supply (UPS) system [13], and the voltage regulator system [12]. The IVSC solved

Manuscript received March 25, 2002; revised September 3, 2002. This paper was recommended by Associate Editor H.-X. Li.

C.-P. Hung is with the Department of Electrical Engineering National Chin-Yi Institute of Technology, Chung-Shan Road, Taiping City, Taiwan, R.O.C. (e-mail: c.p.hung@seed.net.tw; hcb@lsc.net.tw).

Digital Object Identifier 10.1109/TSMCB.2003.811768