REFERENCES

Ikeda, S., Ochiai, M., and Sawaragi, Y. (1976). Sequential GMDH Algorithm and Its Application to River Flow Prediction. IEEE Trans. Syst. Man Cybern. SMC-6(7):473-479.

Ivakhnenko, A. G., and Ivakhnenko, N. A. (1974). Long-Term Prediction by GMDH Algorithms Using the Unbiased Criterion and the Balance-of-Variables Criterion. Sov. Autom. Control 7(4):40-45.

Ivakhnenko, A. G., and Ivakhnenko, N. A. (1975). Long-Term Prediction by GMDH Algorithms Using the Unbiased Criterion and the Balance-of-Variables Criterion, Part 2. Sov. Autom. Control 8(4):24-38.

Ivakhnenko, A. G., Ivakhnenko, N. A., Vysotskiey, V. M., and Cheberkus, V. I. (1976). Long-term Prediction by GMDH Algorithms Using the Unbiased Criterion and the Balance-of-Variables Criterion, Part 3. Sov. Autom. Control 9(2):28-42.

Ivakhnenko, A. G., Krotov, G. I., and Visotsky, V. N. (1979). Identification of the Mathematical Model of a Complex System by the Self-Organization Method. Theoretical Systems Ecology: Advances and Case Studies, E. Halfon (Ed.). Academic Press, New York, Chap. 13.

2

Adaptive Learning Networks: Development and Application in the United States of Algorithms Related to GMDH

ROGER L. BARRON* / General Research Corporation, Subsidiary of Flow General, Inc., McLean, Virginia

ANTHONY N. MUCCIARDI,† FRANCIS J. COOK,† JOSEPH NELSON CRAIG,† and ANDREW R. BARRON‡ / Adaptronics, Inc., Subsidiary of Flow General, Inc., McLean, Virginia

I. INTRODUCTION

The GMDH family of modeling algorithms used today discovers the structure (functional form) of empirical models as well as performing the traditional task of fitting model coefficients to bases of observational or postulated data. Forty years ago scientists began seeking such inductive algorithms in their quest for underlying principles governing the activity of the central nervous system. It was believed—with good reason—that a grasp of these principles would yield improvements in feedback control systems and in the design of automatic calculating machines.

In the United States, algorithms related to GMDH are traceable directly to studies in the 1940s of the behavior of neurons and neuron aggregates. In the USSR, more emphasis was directed (particularly in the 1960s) toward the mathematics of cybernetic systems than toward the emulation of neurons. Both lines of development began to flow together for the 1970s, and in the past five years, scientists in Japan have also had an impact on the course of GMDH work.

Present affiliations:
*Barron Associates, Inc., Annandale, Virginia
†General Research Corporation, Subsidiary of Flow General, Inc., McLean, Virginia
‡Information Systems Laboratory, Stanford University, Stanford, California

Recent U.S. activity has emphasized use of a predicted squared error criterion for prevention of model overfitting. Using this criterion, it is not necessary to divide the data base into groups: all available data can be used for model fitting, and overfitting is prevented at each stage of model synthesis, including establishment of the structures of individual elements within the network (polynomial model). Because the data base is not grouped into subsets, much of the current U.S. work in the field of this book cannot strictly be said to be GMDH activity. Nevertheless, the GMDH imprint has been indelible, and the authors feel it appropriate to outline the principal U.S. line of development—generally referred to as the adaptive learning network (ALN) approach—in a volume devoted substantially to the Soviet treatment, known as GMDH.

This chapter will trace the origins and development of the ALN method in relation to GMDH, summarize the predicted squared error criterion and its use in ALN synthesis, outline other principal distinctions between ALN and GMDH algorithms, and report on representative applications of the ALN method. No attempt will be made to describe details of the GMDH algorithm; these are treated elsewhere in this volume, as is the derivation of the predicted squared error criterion. In some cases the authors of the present chapter quote at length from earlier papers that are not available in readily accessible publications.

II. BACKGROUND: DEVELOPMENT OF ADAPTIVE LEARNING NETWORK TECHNIQUES PRIOR TO 1971

In the 1940s, the physical and information sciences had already felt the impact of probabilistic models replacing many of the earlier deterministic representations of natural phenomena. The mathematics of these probabilistic models appealed to interdisciplinary scientists seeking to explain the actions of the human nervous system and apply their findings to the improvement of communication and control processes. Wiener et al. [1] in their 1943 paper "Behaviour, Purpose, and Teleology" showed the "possibility of treating such teleological notions as 'goal,' 'purpose,' 'evolution,' etc. in a quantitative manner. Philosophers had been debating these terms for centuries, and it was quite startling in the context of the times to be shown that a quantitative treatment was indeed possible. As an outgrowth of this way of thinking, Wiener in 1948 founded the science of Cybernetics [2], meaning, in his words, the study of control and communication in animal and machine [3].

Continuing from Ref. 3:

> Wiener's development of this theme was couched in a highly abstract form for the times and not easily accessible to the engineer. His ideas were put at the disposal of the engineer by the British psychiatrist, W. R. Ashby [4,5] . . . [who] demonstrated . . . that it is possible, working from purely cybernetic ideas, to develop machines that would show such peculiar animal-like characteristics as purpose, goal, and

> survival potential in an a priori unknown environment. It is important to note that a brain was not being constructed, but rather a machine which exhibited behavior which is presumed to be caused, in animals, by a nervous system.
>
> Ashby amplified the definition of Cybernetics as follows: "Many a book has borne the title 'Theory of Machines,' but it usually contains information about things, about levers and cogs. Cybernetics, too, is a 'theory of machines' but it treats not things but ways of behaving. It does not ask, 'What is this thing?' but, 'What does it do?' It is thus functional and behavioristic." . . . The point of Cybernetics, then, is the explicit possibility of quantitatively treating such behavioral parameters independently of the mechanisms presumed to give rise to them.
>
> A number of scientific people . . . were caught up in the ferment of these ideas. The U.S. Department of Defense became interested, and in the period 1958-60, through one of its elements, the U.S. Air Force, initiated the Bionics (another word for applied Cybernetics) program. We will not go into detail on the various types of learning and self-organizing machines studied since this period. One type, known as a probability state variable (PSV) machine, was ultimately selected for intensive development. . . .

The concepts for a PSV machine grew out of the work of R. J. Lee in the 1950s and early 1960s on artificial neurons and neuron networks [6-9]. Theoretical work began in 1961 on the use of PSV machines for control system applications [10,11], and a laboratory prototype of a PSV controller was first constructed in 1964 [12]. Successful flight testing of an elementary PSV controller took place in 1969 [13,14].

A PSV controller is self-directed toward a performance goal, using internal "reward" and "punishment" (selective reinforcement) actions to influence its behavior. These rewards and punishments are interpreted by the controller in the context of the prior decisions that produced them, and the controller modifies the statistics of its internal states accordingly.

To acquire sufficient feedback information for the purposes of identification and control of a plant, PSV controllers must interact with that plant. This interaction takes the form of small experiments conducted at a rate that is generally comparable to the bandwidth of the closed-loop system. There are many interesting theoretical and practical topics associated with this class of controller; unfortunately, these are outside the scope of this paper.

References 10 to 33 detail many of the aerospace applications of PSV self-organizing controllers (SOCs). Other uses of PSV control have been made, chiefly in systems that must continuously reallocate resources in the context of rapid environmental changes; there is, however, little in the open literature on these other applications.

U.S. cybernetics research in the 1960s tended to focus on self-organizing control processes, but in 1965, attention began to be directed toward aspects of empirical modeling (a design process). Although self-organizing control systems cannot have an explicit "teacher" and must rely on self-assessment of performance, empirical modeling processes usually can be guided by an explicit, stored data base and gauge their performance by means of a goodness-of-fit criterion. In 1963 it was not known how to discover the structure of an empirical model from the data base, except to the extent that identification of parameter values within an assumed overall structure might cause certain terms in the model to become negligible. The structure was postulated by the analyst, and a multiparameter numerical search was used to find the values of the model parameters [34,35].

The first empirical modeling application addressed from a cybernetics viewpoint appears to have been the high-speed prediction (in 1963) of trajectories of atmospheric ballistic reentry vehicles [34,35]. A network of 72 algebraic elements was prestructured using multilinear, two-input elements of the form

$$y = w_0 + w_1 x_i + w_2 x_j + w_3 x_i x_j \qquad (1)$$

where x_i and x_j represent the inputs; $w_0, \ldots, w_3$ are constants; and y is the output of the element. A "guided" random search was employed to find the values of the 288 parameters in this network. The performance objective for the search was the minimization of average absolute error on all points in the training data set. After training, an independent testing set was used to verify that the model performed properly on new data. A typical data base consisted of radar tracking data for 50 trajectories, with half the data used in the training set and the balance in the testing set. Prediction accuracies for these models were comparable to those usually obtained through serial integration of differential equations of motion, and solution speeds were several orders of magnitude faster than via integration.

From the perspective afforded by the 1960s trajectory prediction model synthesis work, L. O. Gilstrap, Jr. wrote as follows in 1971 [36]:

> One of the more critical problems in the design of intelligent machines is how to construct a large enough space of possible transformations or mappings in the performance unit of a learning machine. If a learning system is to find a suitable mapping of inputs into outputs, that mapping must be within its range of possible mappings.
> This problem is most acute in systems with many interacting variables. One important solution to this problem is provided by a method for approximating nonlinear hypersurfaces. Just as arbitrary (but reasonably well-behaved) functions of one variable can be approximated by a polynomial, so can arbitrary functions of many variables be approximated by a suitable high-degree multinomial.* Constructing a space of

*Kolmogorov-Gabor polynomial.

> hypersurfaces must be done indirectly, however. To show this we define first multi-linear, homogeneous, and complete multinomials.
>
> A multi-linear multinomial is a polynomial in m variables in which all possible product pairs, product triples, ..., and the m-way products appear, but no variable appears to a degree higher than first.
>
> As an example, the multi-linear multinomial in three variables is
>
> $$y = a_{000} + a_{100} x_1 + a_{010} x_2 + a_{001} x_3 + a_{110} x_1 x_2 + a_{101} x_1 x_3 + a_{011} x_2 x_3 + a_{111} x_1 x_2 x_3$$
>
> Note that $\partial y / \partial x_1$ is not a function of x_1, and similarly for the other variables in a multi-linear multinomial. Also, if none of the coefficients is zero, there are $N_M = 2^m$ terms in a multi-linear multinomial in m variables.
>
> A homogeneous multinomial of degree d in m variables is a polynomial such that the exponents of all the variables that make up each term sum to d.
>
> As an example, the homogeneous multinomial of degree two in three variables is
>
> $$y = a_{200} x_1^2 + a_{020} x_2^2 + a_{002} x_3^2 + a_{110} x_1 x_2 + a_{101} x_1 x_3 + a_{011} x_2 x_3$$
>
> Note that if none of the coefficients is zero, the number of terms in a homogeneous multinomial of degree d in m variables is
>
> $$N_H = \frac{(d + m - 1)!}{d!(m - 1)!}$$
>
> A complete multinomial of degree n in m variables is the sum of all homogeneous multinomials from zero^th degree through n^th degree.
>
> The number of terms in a complete multinomial of degree n in m variables, provided none of the coefficients is zero, is
>
> $$N_C = \frac{(n + m)!}{n!m!}$$
>
> From the magnitudes of N_M, N_H, and N_C for even relatively small values of n and m, it is apparent that it is not practical to use multinomials directly to approximate nonlinear hypersurfaces. For example, in the case of m = 24 variables, for which a fifth degree (n = 5) surface is to be fitted, $N_C = (24 + 5)!/24!5! = 118,755$ coefficients would be required to specify the surface. If a multi-linear multinomial in these

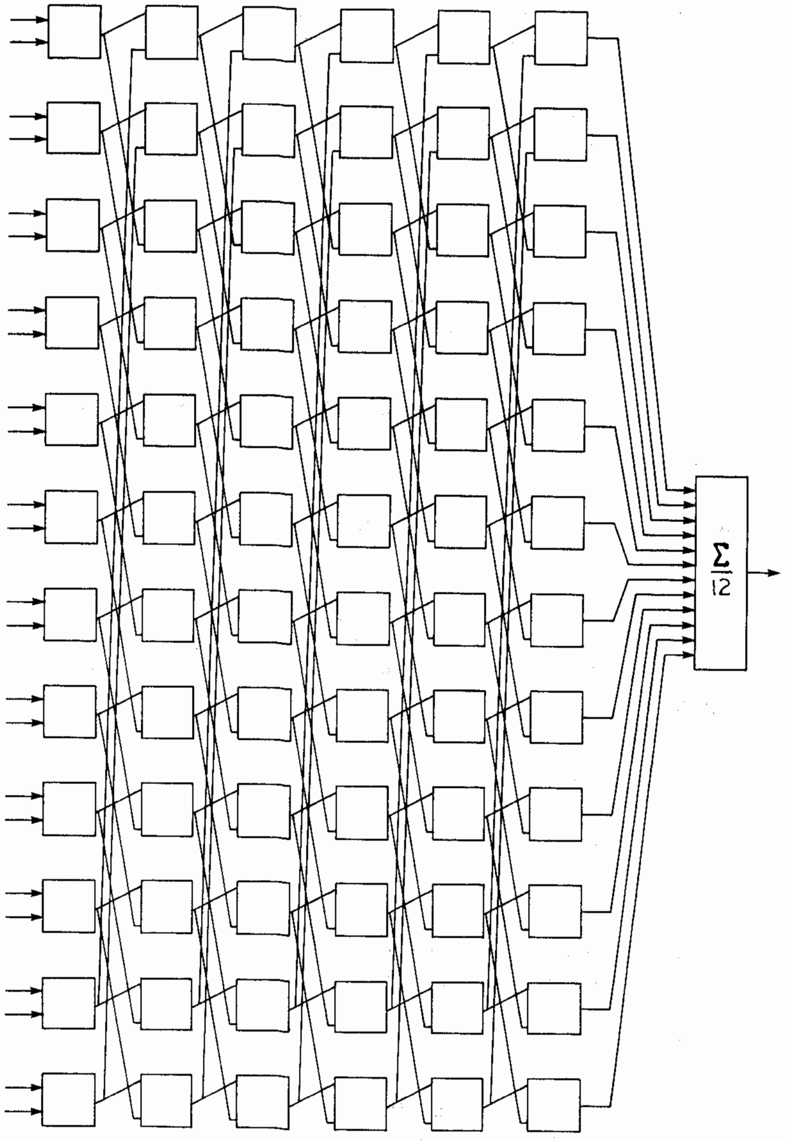


Fig. 1 Uniform spiral 72-element network.

24 variables were desired, then $N_M = 2^{24} \approx 1.6 \times 10^7$ coefficients would have to be specified.

However, high degree multinomials in many variables can be generated using a basic building block element that computes a multi-linear multinomial in two variables:

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2$$

where x_1 and x_2 are inputs, y is the output, and the w's are arbitrary coefficients. . . .

If three of these scaled units are connected into a triangular network . . ., all the terms in a multi-linear multinomial in the four input variables, $x_1, \ldots, x_4$, appear in the output, y_3. However, it should be noted that the coefficients appearing in y_3 are not all independent. As stated earlier, a multi-linear multinomial in four variables requires $2^4 = 16$ independent coefficients, while y_3 has only 12 independent coefficients (the four w's in each of the three units, ... While this [lack of independence for all coefficients in the multinomial] might appear to be a defect in the generation of larger multinomials, it turns out to be the key factor in the practical realization of high degree multinomials in many variables. Although physical systems in many variables can exhibit strong interactions in these variables, the interactions are constrained in many respects, and a multinomial with relatively few degrees of freedom in the coefficients can be used to describe the interactions.

The degree of the multinomial produced by a network is dependent on the connectivity pattern of the building block elements. Figure 1 shows a rectangular network of 72 [multi-linear, two-input] elements connected in a feedforward fashion. In a rectangular net, the highest degree term in any one variable can be as high as one less than the number of columns in the net. The cross-product terms will appear to higher degree; e.g., in the net in Fig. 1, cross products with 12 variables appear in the net output, so that the multinomial produced by the net is a sum of two complete multinomials of degree five in 12 variables and two multi-linear multinomials of degree six.

It is important to obtain complete mixing of inputs in a net if there is no prior knowledge as to which cross-product terms in the multinomial are needed. When the products of all pairs of inputs to a net appear in the net output, the net is said to have sufficiently rich connectivity. The multinomial produced by a sufficiently rich connectivity generally contains all the zeroth, first, and second degree terms, i.e., is at least a quadratic multinomial. The net shown in Fig. 1 does not satisfy the sufficiently rich condition. It is a cylindrically connected net with a uniform spiral of pitch two; i.e., the outputs from any row element are connected to the elements in the same row and to the elements two rows lower. The last two elements are connected back to the first two rows, much as if the net were wrapped around a cylinder and then connected according to a uniform pattern. Rectangular nets with 12 rows and six columns with an alternating pitch of one and two do satisfy the sufficiently rich condition, although nets of pitch one or two alone do not. Note also that a sufficiently rich rectangular net with 72 elements and 24 inputs similar to that of Fig. 1 would actually produce a multinomial with more than the 118,755 terms in a fifth degree multinomial in 24 variables, and it is apparent why no effort was made to write out the output of the net of Fig. 1. . . .

There are several questions that might be posed of this method of approximating multinomials, even assuming that the correct coefficients can be found, such as:

1. How good is the approximation to an arbitrary multinomial?
2. Is the set of network coefficients unique?

The first question is difficult to answer theoretically, but, on purely practical grounds, various empirical data bases containing from eight to 24 variables have been satisfactorily approximated. Since it is impractical to realize fifth-degree functions of as many as 24 variables, there is nothing to compare results with. About the best that we can do is to compute the distribution of errors over the hypersurface. If the distribution of errors is acceptable, then the network approximation can be used, much as any function-generator or multi-variable table can be.

In general, the set of network coefficients is not unique, but uniqueness of the coefficients is important only when the coefficients have some physical meaning, and it is the values of the coefficients that are desired, rather than the approximation to the multinomial. Since network coefficient values are partly a result of the connectivity pattern of the net, they cannot have much physical meaning, and the question of uniqueness is somewhat irrelevant.

Having indicated that it is possible to generate multinomials, the next problem is how to find the four coefficients in each of the network elements. Classical matrix inversion is clearly of no value here, since the coefficients of the multinomial terms are nonlinear combinations of the coefficients in the multi-linear multinomials in the elements of the net. (In the network of Fig. 1, there are four coefficients per element and 72 elements in the net, for a total of 288 coefficients to be found.) The most satisfactory procedure for finding these coefficients should be independent of the initial values used in the search and should converge quickly; such a procedure is provided by . . . guided random search. . . .

Search processes can be divided into two groups, deterministic and random. The latter group appears to be the most suitable for finding

multinomial coefficients. R. L. Barron [14] has summarized a comparison of a guided random search with the gradient search (one of the deterministic searches) as follows:

1. The guided random search converges faster for spaces of high dimensionality.
2. The guided random search is effective for multi-modal surfaces (the gradient search is basically suited only for unimodal surfaces).
3. It [the guided random search] is an effective search for time-varying surfaces.
4. The guided random search is more effective in coping with sensor or measurement noise than the gradient search.
5. The guided random search can be used to search all parameters simultaneously.
6. The guided random search can be mechanized simply, and in software versions generally requires less storage and less computation time than the gradient search.

In view of the advantages of guided random searches, we will discuss only the random searches in what follows. Much of the discussion has been excerpted from an earlier paper by R. L. Barron [37].

Let $X = (x_1, x_2, \ldots, x_n)$ be a point in an n-dimensional space being searched. Associated with each point in space is some type of index or score, S(X), which is a measure of the value or utility of that point. Typically, the objective of any search is to find the point, X_m, which yields the maximum (or minimum) value of S. Frequently, there are constraints on the permissible choices for X; among the simplest of these constraints is an upper and lower bound on each of the variables being searched. One consequence of a constraint of this type is that the maximum or minimum found within a permissible region may not correspond to the theoretically optimum point, since that point might lie outside the permissible region. Also, there is, in general, no guarantee that the point, X_m, in a bounded space is unique, but uniqueness is not always an essential characteristic of a solution to a search problem.

The rule for halting the search process, called the stop rule, may depend on such practical matters as having a fixed time in which to perform the search, or may be determined from the score function itself. For example, if a minimum value of the score function is sought and if that minimum value is zero, then the search can be halted if a . . . selected set of coefficients produces a score that is arbitrarily close to zero.

In an unguided (or parallel) search, the sequence of trial points, X^1, X^2, ..., X^k, is selected according to a fixed formula or algorithm which does not take into account the results of each trial point. In a guided search, the score from one trial point is used to guide the selection of the next trial point, i.e.,

$$X^{k+1} = X^k + \Delta X^{k+1}$$

where ΔX^{k+1} is selected on the basis of S^k, the score obtained on the kth trial, and the search becomes an iterative procedure.

Most of the guided random searches are modifications to the basic, unguided random search. In the unguided random search, points are selected at random from the total space being searched; scores are noted for all of the points and, after k trials, the best estimate for X_m is simply the point corresponding to the maximum score obtained during the k trials. Brooks [38] has shown that the probability, p(f), that X_m lies in a certain fraction, f, of the total space after k trials is

$$p(f) = 1 - (1 - f)^k$$

which approaches unity as k increases indefinitely. The expected number of trials to achieve a given level of confidence that the maximum lies in a fraction, f, of the total space can be obtained from the above equation:

$$k = \frac{\log [1 - p(f)]}{\log (1 - f)}$$

In the basic, unguided random search, the sampling of points is obtained from a rectangular distribution, and no use is made of information gained on prior trials.

Although the unguided random search is slow compared to all guided searches, it is independent of the modality of S, i.e., it can be used to find the global maximum of S and is not subject to "trapping" by local maxima. Because of this desirable feature of the unguided random search, several modifications to the algorithm have been devised to improve the rate at which it converges. The simplest of these modifications is the change from sampling from a rectangular distribution for each variable to sampling from a normal distribution centered about the point corresponding to the maximum score obtained from the beginning of the search to the current trial. This search is guided, but it makes minimal use of prior information. Additional modifications to this algorithm include reversal, hill climbing, acceleration, and smoothing of the terminal search. These modifications assume continuity of S but are so designed that the search reverts to an unguided search if continuity does not hold and no correlations can be found in the accumulated information.

Reversal is based upon the principle that the opposite of downhill is actually uphill. Hence, if a given step, ΔX^k, produces a worsening of the score, ΔX^{k+1} is set to $-\Delta X^k$; if this step also produces no improvement in score, then the step ΔX^{k+2} is taken at random.

Once a direction is found that produces an improvement in score, either by a random trial or by reversal, the continuation of trials in the same direction is . . . hill climbing. Although an uphill direction selected at random will not, in general, coincide with the direction of the maximum slope, improvement in performance may be noted for several steps in any uphill direction. Since the expected number of experiments [at the beginning of the search] required to find an uphill direction (not necessarily the maximum slope), is about one-half [the number of experiments performed], then hill climbing is seen to be a means for exploiting this limited information acquired by random trials.

The information as to whether a given direction is uphill or downhill is of limited value, and it should be exploited as quickly and efficiently as possible. This can be done by lengthening the average step size as long as performance continues to improve. Both arithmetic and geometric progressions of step sizes have been used in random searches during the hill climbing phase. This increase in step size each step is called acceleration of the search. In a geometric acceleration, e.g., doubling the step size each step, large overshoots can occur, and Matyas [39] has employed a deceleration to come as close as possible to the highest point in the given random direction. . . . Matyas also employed a bound on the largest possible step size. . . .

Although bounding the maximum possible step size does not speed up the random search, control of the average step size does appear to provide some improvement in the terminal search. Scaling of the average step size as a function of the score provides smoothing of the search in the region near the maximum as well as improving speed of search. In searching for minima using, for example, least squares score functions, average step size can be set proportional to the best-to-date score to provide automatic and continuous scaling of step size. The constant of proportionality can be adjusted for each problem. In searching for maxima, scaling can be inversely proportional to the score or can be any convenient, monotonically decreasing function of score.

More recently, Mucciardi [40] has described further refinements of the guided random search that improve the ability to shift modes.

A bibliography on random search is presented in Ref. 41. Further summaries of the pre-1971 development of ALN techniques in the United States are contained in Refs. 42 and 43.

In 1968, one of the authors of this chapter had the privilege of meeting with A. G. Ivakhnenko in the USSR while attending technical conferences in that country. Translations of several of Ivakhnenko's works on the theory of self-organizing systems and his earliest writings on GMDH had just begun to appear in the United States, although not in publications generally accessible to the scientific community. Ivakhnenko was invited to submit a paper on GMDH for publication in the *Transactions on Systems, Man, and Cybernetics* of the Institute of Electrical and Electronics Engineers, Inc. The paper he produced, "Polynomial Theory of Complex Systems" [44], greatly stimulated interest in GMDH outside the USSR following its appearance in 1971.

The writings of Ivakhnenko on GMDH, which have appeared primarily in the bimonthly Kiev journal *Avtomatika*, are extensive.* Reference 45 is a succinct presentation of GMDH research by Ivakhnenko and his associates through the late 1970s.

III. APPLICATION AND REFINEMENT OF GMDH: DEVELOPMENT OF ADAPTIVE LEARNING NETWORK TECHNIQUES FROM 1971 TO 1978

The greatest significance of GMDH is in its capacity for "discovery" of the functional forms of empirical models. This capacity greatly lessens the need for analyst involvement in the model synthesis process and reduces the time (calendar and computer) that must be expended.

In 1970, Armco Steel Corporation became interested in the application of ALN techniques to processes in the steel industry. Their sponsorship of basic and applied research and development (R&D) in ALN areas was a significant factor in the development of GMDH empirical modeling in the United States, because the industrial emphasis of the Armco support demanded improvement in cost-effectiveness of the ALN methodology. Predicated on Ivakhnenko's paper [44], a "Polynomial Network Training Routine (PNETTR), Version I" was programmed and evaluated. Following on its success, Mucciardi formulated in 1971-1972 a Version II algorithm that incorporated several procedures that enhanced the basic GMDH. Version II was used extensively for approximately eight years, and proved itself in a great variety of applications [46].

The primary characteristics of PNETTR II not found in prior GMDH programs were:

1. The Mucciardi-Gose clustering procedure [47] was used to ensure the representativeness of the data groups (subsets), as discussed further below.
2. Three independent data subsets were employed—Fitting (F), sometimes called the Training subset; Selection (S), also referred to as the Testing subset; and Evaluation (E), used after model synthesis to predict the accuracy of the model.

*These works are available in the English-language translation of *Avtomatika*, titled *Soviet Automatic Control*, available through Scripta Publishing Company, 7961 Eastern Avenue, Silver Spring, MD 10910.

3. The original candidate inputs considered in layer 1 of the model were reintroduced as candidate inputs to each following layer (together with the outputs of the immediately preceding layer), thereby enlarging the combinational possibilities during evolution of the model.
4. The first minimum in the S subset error rate was used as the stopping criterion in model synthesis unless a significantly lower minimum was reached within a predetermined number of additional layers.
5. After preliminary synthesis of the model was obtained substantially in the manner taught by Ivakhnenko, the coefficients (weights) within the *entire* model were optimized (keeping the structure fixed) by means of a multiparameter search routine.
6. Except in instances which suggested that overfitting would have resulted, multiple, parallel subnetworks were created, with their outputs combined by summation, each subnetwork having been trained and tested on the error residuals (in F and S, respectively) from the aggregate of all lower-order subnetworks.

The use of a clustering algorithm was found to be valuable in applying the GMDH cross-validation procedure. It was also found that the identified cluster structure was highly useful in subsequent data screening to ensure that the model was being interrogated under conditions for which it had been trained [48]. It came to be more fully recognized than before that data cluster structures are, in themselves, valuable models if the behavior of the modeled system can be unambiguously correlated to the various input (observational) data clusters. The utility of a known prior cluster structure for data screening and data modeling was found to stem in part from the finite boundaries of data clusters—these boundaries signal immediately if the model is being called upon to extrapolate (risky) or interpolate (usually safe) [49]. Finally, it was found that the identification of cluster membership for an unknown input data point can sometimes be used as a "pointer" to lead the decision process to ALN-type models tailored for the particular clusters.

It is emphasized that data clusters are inherently bounded regions, whereas polynomial networks synthesized by GMDH and ALN methods extend to infinity along each axis of the modeled spaces. Because of their importance to the subject of empirical modeling and the discovery of data structures, attention is directed here to some of the details of cluster analysis.

The main idea in cluster analysis is to "discover" the groups, or clusters, of points that lie close together in the data space. Closeness is defined by a distance measure. The most suitable clustering algorithms are those that make the weakest assumptions about the statistical structure of the data, are order independent (i.e., the data may be introduced in any arbitrary sequence without materially altering the inferences made by the algorithm about the structure of the data base), and are recursively updatable (i.e., each time an additional input data point is introduced, the structure can be updated without calling forth the prior data except in terms of their statistical properties).

After a clustering analysis has been employed to examine the geometric (spatial) interrelationships of observed data, the results obtained include:

1. Structure of the Data Space: the number of distinct clusters (classes) that are discriminable based on a set of N measured parameters
2. Identification of Noisy Data: clusters or isolated points that are far removed from the main data body
3. Detection of Nonstationary Conditions: consistent observations of new data, which fall in the periphery or just outside existing clusters, whose effect is to cause cluster migration
4. Discovery of New Operating Regions: new data that form clusters which do not overlap with existing clusters (i.e., are statistically distinct via a multivariate F-test), but are not far enough removed to be considered noise
5. Avoidance of Extrapolation Error: screening new observations before interrogating a model to ensure that new data fall within or near those regions of N-space for which the model was synthesized
6. Establishment of Model Confidence Regions: modeling the error of a model to assign probabilities of error to future model outputs on new observations

A clustering analysis can therefore be used as an information filter [48] to detect the foregoing process descriptors from experimental data. These six aspects of data structural analysis are discussed below; further details can be found in the quoted references.

The Mucciardi-Gose CLUSTR algorithm [47] will now be outlined (refer to Fig. 2). The first data sample (a vector observation of N components) is introduced and the first cell (cluster) is centered at this point. The cells are hyperellipsoids, and their initial radii (principal axes) are preselected. The birth of each cell defines a new cluster in the space. The next sample is presented and it either falls within the boundary of the existing cell, within a "guard zone" surrounding the cell, or outside the guard zone so that a second cell is generated and centered at this point. Similarly, all succeeding points either fall within the cells in existence at that time, within their guard zones, or determine the generation of new cells. When a point falls within a cell, the location of the cell (its mean) and radii are changed to accommodate this new point. The cells thus locate themselves at the dense regions (modes) of the data and assume shapes that conform to the spread of data about these modes.

The CLUSTR algorithm requires the following parameters for each cell, which control the birth and growth rate of the cells: (1) shape factors, σ_i, proportional to the distances from the cell center to the cell boundaries in each dimension (these describe the N-dimensional shape of the cell); (2) a cell size factor, τ, equal to the radius of a hypersphere containing the same hypervolume as the cell; and (3) a guard zone size factor, θ, which is the ratio of distances from the cell center to the outer and inner boundaries of the guard zone (see Fig. 2). The centers and shapes of the cells are adaptively updated as the data are sequentially introduced, and a provision is made for merging cells containing few points with their nearest large neighbor (if sufficiently close) after all the data have been examined.

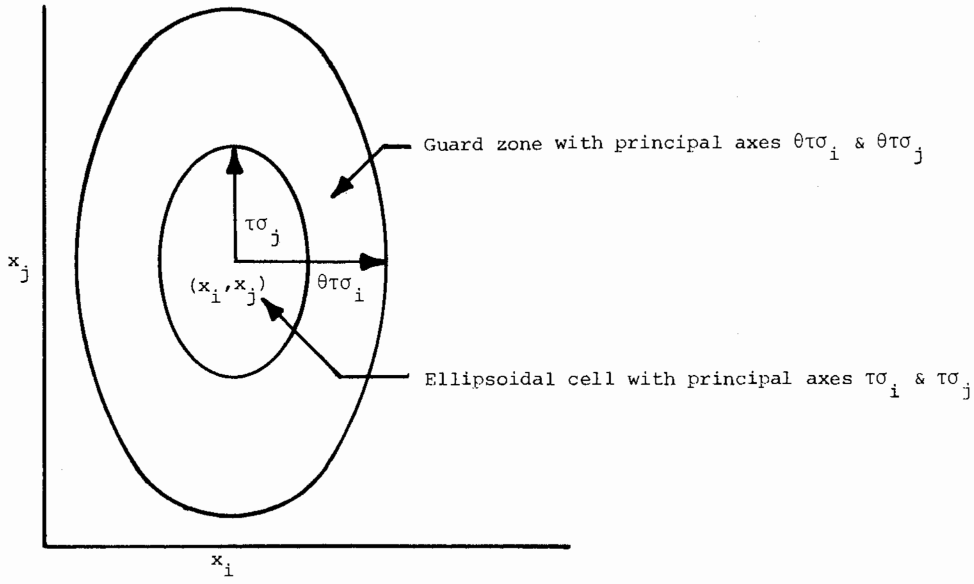

Guard zone with principal axes $\theta\tau\sigma_i$ & $\theta\tau\sigma_j$
$\tau\sigma_j$
x_j
(x_i, x_j)
$\theta\tau\sigma_i$
Ellipsoidal cell with principal axes $\tau\sigma_i$ & $\tau\sigma_j$
x_i

Fig. 2 Cell descriptors.

Points falling within cell m satisfy

$$\sum_{i=1}^{N} \left[\frac{x_i - \bar{x}_{mi}(t)}{\sigma_{mi}(t)} \right]^2 \leq \tau^2$$

where $\bar{x}_{mi}(t)$ and $\sigma_{mi}(t)$ are the center coordinate and shape factor of the cell in dimension i at some time t, and x_i is the ith component of the input vector X. Points X satisfying the inequality are likely to lie inside or on the boundary of a hyperellipsoid in N dimensions with principal axes $\tau\sigma_{mi}$.

Cells are prevented from overlapping, at least during the initial part of the growth phase, by the use of a guard zone. A point falling in the guard zone is not allowed to generate a new cell. All such points are temporarily stored and tagged for later processing. If the size of the guard zone is carefully chosen, it will prevent the birth of new cells at points which are close enough to current cells so that the cells would be likely to cover common regions of space after attaining their full growth. The reason for temporarily storing these points is that the closest cell may grow due to the influence of newer data. Thus, at some later time, a point that was originally in the guard zone of a cell may be contained inside the cell. If this does not happen after all the data have been examined, each point remaining in storage is used to update the cell nearest it. Mucciardi and Gose [47] provide techniques for computing the cell descriptors as a function of dimension, N. The results reported by the CLUSTR algorithm are:

The number of clusters (i.e., cells)
Their (N-dimensional) location
Their (N-dimensional) shape
The identity of the data points in each cell
The amount of overlap existing (if any) between cells

The CLUSTR program also computes the probability of observing new data in the overlap regions. One of the uses of this information is for cluster merging. When all clusters have been computed, a multidimensional F-test is used in CLUSTR to determine if some of them are not statistically distinct. The generalized multivariate F-test for two N-dimensional distributions containing n_1 and n_2 points and with mean vectors and covariance matrices X_1, X_2 and S_1, S_2, respectively, is

$$F = \frac{n_1 + n_2 - N - 1}{N(n_1 + n_2 - 2)^2} \; \frac{n_1 n_2}{n_1 + n_2} \; (X_1 - X_2)^T (S_1 + S_2)^{-1} (X_1 - X_2)$$

This statistic may be compared to $F_{\alpha;N,n_1+n_2-N-1}$ at the $\alpha = 0.05$ level of significance. If F is less than the tabular value, the hypothesis that the two distributions are not statistically distinct is accepted. Any clusters that are not distinct are merged automatically. The final result is a parsimonious description of the data base structure. This description is useful in establishing an unbiased partitioning of data bases, in defining the effective boundaries of prior data regions, and in determining exact relationships between a new data point and prior data clusters, regardless of the dimensionality of the space.

The structure of the multivariate data space can be inferred from the results given above. For example, one cluster containing the majority of the data, surrounded by clusters containing a few points each, is the usual result for a unimodal data structure. A bimodal data structure produces either two clusters containing all the data, or two clusters containing a majority of the data, surrounded by smaller satellite clusters, and so on.

If the data are from K classes and are to be used for synthesizing a pattern recognition system, it is very helpful to perform K cluster analyses, one for the features of each class. The degree of overlap between classes is a measure of the irreducible error based on the parameters measured, and this information can be used both for parameter selection and the design of the classifier.

The CLUSTR algorithm plays an important role in determining the quality and consistency of the observations in a multivariate data base. Sensor information might be degraded due to equipment failure or to temporary interruptions, such as dust or steam shielding an optical sensor. Therefore, it is very important to screen data in the input vector. Screening can be performed in the following way. A clustering analysis is first conducted on a set of data free of fault conditions to find the regions of the operating space in which the valid sensor data are clustered. Then, in an operational mode (as new data are observed), the cluster to which a new input vector is "closest" is determined (see below). The input vector is accepted as legitimate if it falls within the "nearest" cluster. If not within, this event signals either a fault condition or a time shift of the process.

Once a prior data cluster has been found, a normalized metric, D^2, may be computed for the distance between any new multidimensional point, X, and the mth cluster by the following equation:

$$D^2(m, X) = \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}_{mi}}{\sigma_{mi}} \right)^2$$

where $\bar{x}_{mi}$ is the mean value of x_i for the mth cluster and σ_{mi} is the length of the ith principal axis of this cluster. D^2 is computed from each cluster, and thus the value of D^2 is found for that cluster, m*, for which the normalized distance to point X is a minimum. If $D^2(m^*, X)$ is less than unity, X is inside cluster m*; if this distance equals unity, X is on the m* boundary; and if X is greater than unity, X is outside cluster m*.

A decision to classify an input vector as a noisy, or fault, condition would probably be rendered if the data point is far from the nearest cluster. This is reasonable, since a distant point can result from one or more of its components taking on an extreme value with respect to the main portion of the data.

On the other hand, new observations which are just outside or in peripheral portions of clusters which represent the past observed operating regions probably suggest that a time shift is occurring. Depending on the physical process involved, the time shift may take the form that (1) all operating regions are moving simultaneously at the same rate, or (2) all operating regions are moving simultaneously but at different rates (some of which may be zero). The CLUSTR program can be used to detect which of the two types of time shift is taking place and to determine the rate of movement of each operating region (cluster) with respect to the coordinate axes of the data space.

If the frequency of data observations far removed from the main clusters increases and if a check confirms that the sensors are reporting properly, the possibility that new operating region(s) are being observed has to

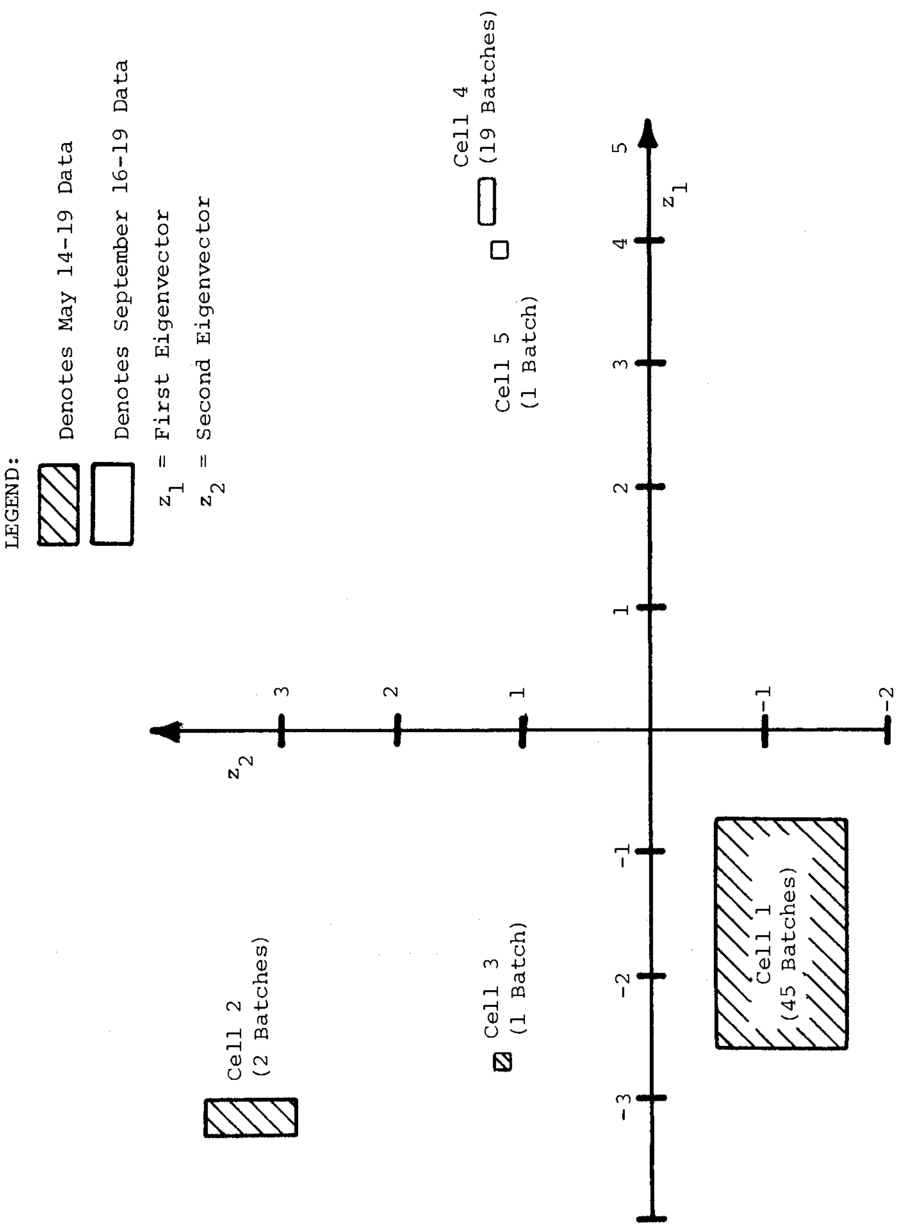


Fig. 3 Data structure for an industrial process. (From Ref. 49.)

be considered. New regions could signal, for example, previously unobserved states of equilibrium in a chemical process. Incoming data points should be screened to avoid extrapolation errors. Usually, the points that are interior to prior data regions can be safely admitted for model interrogation, while points that are exterior should be kept in memory for use in updating the model.

To aid the analyst in visualization of cluster structures of high dimensionality, one may reduce the displayed dimensionality of the cluster structure (with attendant loss of completeness) by computing eigenvectors (the principal components z_1, z_2, ..., z_n) of the entire prior-available data base, then transforming the boundaries of each X-space cluster into the corresponding Z-space boundaries. The kth eigenvector parameter is derived as a linear transformation of the x's:

$$z_k = \sum_{i=1}^{N} u_{ki}(x_i - \bar{x}_i)$$

where u_{ki} is the ith coefficient of the kth eigenvector of the N × N covariance matrix and $\bar{x}_i$ is the mean value of the ith parameter. The clusters may now be plotted as they appear when projected onto the z_1-z_2 plane, and, as a further convenience, cluster ellipses may be drawn as rectangles. The width of the jth cluster expressed as a rectangle along the kth z axis is given by

$$z_{k,\text{lower}} = \sum_{i=1}^{N} u_{ki}[(x_i - s_{ji}) - \bar{x}_{ji}]$$

and

$$z_{k,\text{upper}} = \sum_{i=1}^{N} u_{ki}[(x_i + s_{ji}) - \bar{x}_{ji}]$$

where $\bar{x}_{ji}$ and s_{ji} are the mean and standard deviation values of the jth cluster along the ith coordinate.

The first two eigenvectors (z_1 and z_2) are often adequate to reveal separations between the data clusters, because these eigenvectors account, cumulatively, for much of the variance in the data base (usually more than half). An example is an industrial crystallization process for which data clusters are drawn in Fig. 3. Data on initial conditions of 48 product batches from this process were recorded during the interval May 14-19, 1975; these data clustered as shown in the crosshatched rectangles. Data on another 20 batches were then recorded during the period September 16-19, and these later data were found to cluster as shown in the open rectangles. From Fig. 3 it is readily concluded that the process operating conditions, measuring instruments, and/or the process itself were sufficiently different in September to be far removed in data space from the situations observed in June. This conclusion was not so immediately obvious to the analyst when inspecting a 30-dimensional data file!

The CLUSTR data screening procedure, using the D^2 distance metric, also detected that the September data from this process did not belong to the June data distribution. Accordingly, to avoid extrapolation, models trained on June data were not interrogated with the September data until model updating had been performed.

Once the statistical structure of the data base has been established as described above, the data base can be rationally divided into two or more parts for GMDH model synthesis (PNETTR II). The first part, the design data, consisting of F and S subsets, is used for model synthesis (i.e., model structure and coefficient determination). The second part, the evaluation data, used after the model has been found, is employed in a final test to ensure that the error rate that was obtained on the design data will occur approximately for all future data within the same regions in the N-dimensional parameter space. Notice that the X vectors are clustered as one group and then divided into subsets by selection (typically at random) within each cluster.

Once a model has been found, its output is an estimate of the dependent quantity:

$$\hat{y} = f(x_1, \ldots, x_N)$$

whose true value is y. For each observation j, let Δy_j denote the error committed by the model:

$$\Delta y_j = y_j - \hat{y}_j$$

The X vectors can now be divided according to their associated errors. That is, label as class 1 the set of X vectors for which the model produces the lowest error:

$$X_j \in \{X\}_1 \quad \text{if} \quad 0 \le |\Delta y_j| \le \epsilon_1$$

and so on until the class k set has been established as comprising the remaining observation vectors for which the model is least accurate (i.e., has the largest error):

$$X_j \in \{X\}_k \quad \text{if} \quad \epsilon_{k-1} < |\Delta y_j| \le \epsilon_k$$

The data base has thus been divided into sets based on the modeling error, and now the model error can in turn be modeled as a multiclass pattern recognition problem. That is, given an X, what is the probability that the estimate of the model will be in error between ϵ_{k-1} and ϵ_k? This is expressed as

$$P(\epsilon_{k-1} < |\Delta y_m| \le \epsilon_k | X_m) = P(X_m | \epsilon_{k-1} < |\Delta y_m| \le \epsilon_k) \times P(\epsilon_{k-1} < |\Delta y_m| \le \epsilon_k)$$

The conditional probability distribution function (PDF) for the kth class, $P(X_m | \epsilon_{k-1} < |\Delta y_m| \le \epsilon_k)$, can be found from a k-class cluster analysis in which each of the classes is clustered separately [50].

The PDF for class k can be approximated by fitting a Gaussian distribution to each of the C_k major clusters for that class, using the data within the jth such region ($1 \le j \le C_k$) to estimate the mean and covariance matrix of that region and creating a weighted sum of the statistically distinct regions:

$$P(X_m | \epsilon_{k-1} < |\Delta y_m| \le \epsilon_k) \cong \sum_{j=1}^{C_k} w_{kj} A_{kj} \exp\left[-\frac{1}{2}(X_m - X_{kj})^T S_{kj}^{-1} (X_m - X_{kj})\right]$$

where w_{kj} is the fraction of class k data in the jth region and A_{kj} is a constant inversely proportional to the square root of the determinant of S_{kj}.

As each new data point is observed and an estimate of y_i is rendered, the confidence in the estimate can be assigned via the last two equations above. This process can be easily made adaptive if the cluster structure is updated as soon as the true value of the error for X_i is available [51]. If Δy_i fell within the kth error band that was predicted, the regions of class k in which X_i falls is modified by updating its mean and covariance matrix due to the influence of this new point. If, on the other hand, Δy_i was not in the predicted kth error band, but fell instead in the nth error band, a new operating region for class n has been found. In this way, the probability of correctly classifying X—that is, the probability of assigning the correct model confidence—is made an adaptive process.

IV. POST-1978 DEVELOPMENT OF ADAPTIVE LEARNING NETWORKS

The utility of PNETTR II concealed, for a time, its deficiencies, which were:

1. Strong tendency to overfit the design data
2. Poor correlation between model structure and underlying "physics" of the modeled process

3. Need to partition the data base, which reduced the quantity of data available for model fitting
4. Limitation of model elements ("partial descriptors"—[44]) to bivariate, quadratic form
5. Limitation of model to a single output variable (requiring creation of multiple, unrelated models for the case of multiple outputs)
6. Inadequate mechanisms for influencing model synthesis and behavior on the basis of estimated errors in old and new observational data

In the 1970s, the work of Akaike [52, 53] in Japan influenced additional developments in ALN synthesis algorithms. Akaike introduced an information criterion that incorporated two fundamental types of terms: one that signified the error performance of a model and a second term that conveyed a measure of the complexity of that model. Akaike suggested that the optimum model was that which, in any given instance, produced the minimum sum of error and complexity terms.

A. R. Barron ([54] and in Chap. 4) has carried the reasoning further. He suggests that two related questions have had a major role in the evolution of adaptive learning network synthesis programs. They are:

1. What is the criterion for determining ALN structure?
2. What is the expected performance of the ALN when it is presented with new data?

Ideally, there is one answer that resolves both of these questions; for if we know that a first structure will perform better on new data than another, we should adopt the first. The performance measure recommended by A. R. Barron is the expected squared error on new observations. Two estimates of the expected squared error, derived by him, are discussed below.

A natural way to estimate the expected squared error is by withholding a subset of observations during ALN training and evaluating the (empirical) average squared error on this subset. If the evaluation subset is kept independent of the creation of the ALN, and is representative of the universe of potential observations, it then provides a feasible measure of the performance of the model. However, if this subset is used, as in the GMDH cross-validation treatment, to help select the structure of the ALN, the withheld subset no longer provides an independent measure of future performance. In fact, the selection subset used in GMDH and in PNETTR II influences the future performance through the ALN it selects. Thus a selection data subset is not ideal for its original purpose—selecting structure according to an independent measure of performance on as yet unseen data. In PNETTR II a third group of data, the evaluation subset, did not participate in creation of the ALN and was used to estimate the expected squared error of the final model.

The use of two or three subsets of data permits using cross-validation to avoid overfitting and also (using three subsets) provides an estimate of the expected squared error. However, use of the cross-validation technique requires attention to the partitioning of the original observations into representative data groups and reduces by a factor of 2 to 3 the quantity of data available for model synthesis.

A. R. Barron has shown that the expected squared error on new data can be evaluated analytically with only mild statistical assumptions. The result is a criterion that can be applied to training data to predict the future performance of the ALN, thereby eliminating the need for data base partitioning. Specifically, it is assumed that the model errors are zero-mean, pairwise uncorrelated, and have common variance σ^2. The errors need not be Gaussian random variables and need not have the same distribution. The model may be nonlinear in the input variables. The derivation assumes linearity in the coefficients. Individual elements of ALNs are linear in their coefficients, but networks of elements are not. Nevertheless, the criterion is believed to be a useful and realistic result for ALNs.

Under the assumptions noted above, expected squared error on future data is given by (see Chap. 4)

$$\sigma^2 + \frac{\sigma^2 \text{ trace } (\underline{R}_F \underline{R}_T^{-1})}{n} \tag{2}$$

where $\underline{R}_T$ is the "training covariance" matrix composed of average cross-products between the transformed input variables used in the ALN (the average being over the n training observations) and $\underline{R}_F$ is the corresponding "future covariance" matrix (the average being over any set of observations with errors uncorrelated with training errors).

The two terms in formula (2) correspond to two factors contributing to error on future data. The first term, σ^2, is the expected squared error of the "true" or optimum (but unknown) model. The second term is the expected squared difference between the trained model and the true model when evaluated on data not used for training. If the "covariance structure" of training observations is nearly the same as for future observations, then $\underline{R}_F \underline{R}_T^{-1} \cong \underline{I}$, the identity matrix of dimension k. In that case the expected squared error (2) reduces to

$$\sigma^2 + \frac{\sigma^2 k}{n} \tag{3}$$

where k is the number of estimated coefficients. Thus models of high complexity and many estimated coefficients are not expected to perform well unless there are enough training observations that the second term in (3) is negligible. Small training data bases necessitate simpler models.

Since the true model, its error variance (σ^2), and number of parameters k are regarded as unknown, formula (3) is not yet in the form of a usable criterion for selecting the structures of ALNs. We need a family of estimates of (3) which for each ALN indicates how well it will perform on new data. Then we adopt that ALN which we estimate will perform best. The suggested estimate of an ALN's performance, used in the most recent ALN synthesis algorithm, PNETTR IV, is the predicted squared error (PSE):

$$\text{PSE} = \text{TSE} + \frac{2\sigma_P^2 k}{n} \tag{4}$$

where TSE is the (empirical) average squared error of the ALN on the training data and σ_P^2 is a prior estimate of σ^2 (which does not depend on the ALN being examined). The fixed σ_P^2 in the penalty term is used because we do not want PSE to underestimate future squared error when the particular ALN considered is incorrect (e.g., an overly complex ALN with low error on training data). The factor 2 appears because the training squared error (TSE) is biased below σ^2 by a factor of $\sigma^2 k/n$. From analysis of the statistical properties of PSE, it is found that prior knowledge of σ^2 need not be accurate, although having $\sigma_P^2 \geq \sigma^2$ is helpful to avoid overfitting [54]. Typically, it is reasonable to assume that σ^2 is less than the variation in the dependent variable y, which is given by

$$\sigma_0^2 = \frac{1}{n} \sum_i (y_i - \bar{y})^2$$

where $\bar{y} = \Sigma y_i/n$. If no prior value is provided, PNETTR IV uses $\sigma_P^2 = \sigma_0^2/2$. Experience has verified that this choice gives acceptable results.

The PSE criterion resembles criteria proposed by Mallows [55], and Akaike [52, 53]. The differences and reasons for preferring the PSE criterion are discussed by A. R. Barron. Implicit in the derivations of Mallows and Akaike are quantities such as (3) [but not (2)]. Bibby and Toutenburg [56] have also examined expected performance on future observations [and (2) can be derived from their equations 1.5.5 and 1.5.13]; however, they do not derive a criterion for selecting among many models.

The result (2) can be very useful after an ALN has been selected and is being applied to new data for which one does not know the true value of the dependent variable y and wishes to know how accurate is the ALN estimate. Result (2) indicates that one should monitor the "covariance structure" R_F (the matrix of average cross-products) of the new input data. If trace $(\underline{R}_F\underline{R}_T^{-1})$ is less than or comparable to k, the ALN should be satisfactory. However, if trace $(\underline{R}_F\underline{R}_T^{-1})$ tends to be much greater than k, the ALN may not be suitable for the new data. In such a case the ALN should be adapted or retrained.

The work of A. R. Barron led him to the development of versions III and IV of PNETTR. Because PNETTR III and PNETTR IV do not use data groups (subsets of data for cross validation), they are not, strictly speaking, examples of the group method of data handling. Instead, polynomial ALNs are created via PNETTR III and IV using information-theoretic criteria to govern the selection of terms in the elements and which elements remain in the model. Table 1 summarizes versions II to IV of the ALN polynomial network training routine.

Numerous applications of the ALN synthesis methodology have been made. The preponderance of these applications have arisen in signal, image, and time-series estimation (forecasting, target detection, target discrimination, and the like), but uses have also appeared in discrete modeling solutions. Because of the emphasis on signal processing and analyses of time series, considerable attention has been devoted to the development of computer routines for extraction of signal (time-series) parameters ("features"). Practitioners of ALN modeling now have at their disposal a battery of related computational aids, including time-domain and frequency-domain feature extraction, clustering algorithms (hyperellipsoidal and hypercubical), multimodal Bayes decision-rule synthesis algorithms, and so on.

These same numerous applications have generally been conducted under the assumption that the model to be found,

$$y = f(x_1, \ldots, x_N)$$

expresses an analytic relationship between the dependent quantity y and the feature vectors x_1 through x_N. The Kolmogorov-Gabor (K-G) polynomial discussed at length elsewhere in this volume represents such an analytic relationship. The need to generalize to a class of nonanalytic polynomial representations has been recognized by Cook and Craig [57]. The generalization admits singularities into the analytic background represented by the K-G polynomial. Using this representation, basic building blocks of ALNs can be constructed by PNETTR IV in such a manner that the resulting networks can represent symmetric and asymmetric singularities, discontinuities, and jumps embedded in analytic backgrounds.

It is particularly noteworthy that elementary forms of the ALN synthesis algorithms now exist in new instruments that exploit advances in microprocessor (integrated circuit) technology to achieve completely self-contained ALN capabilities in portable devices. Although it will be some time before the power and generality of these compact devices rival the capabilities of, say, PNETTR IV installed in mainframe scientific computers, it is visualized that the gap in capabilities between portable and fixed-base ALN systems will be gradually closed over the next several years.

One other future trend is discernible at the time of this writing. Traditionally, artificial intelligence (AI) techniques have been oriented almost totally toward realization of AI via emulation of the behavior of human experts. GMDH and ALN techniques provide a useful augmentative or alternative approach for the realization of artificial intelligence; that is, some AI capabilities may be achieved by methods that are relatively alien to human thought processes but exploit the arithmetic and logical powers of computers.

Table 1 Adaptive Learning Network Synthesis Algorithms

Algorithm	Performance criterion	Element structure	Layer structure	Other attributes
PNETTR II (1971)— A. G. Ivakhnenko, A. N. Mucciardi	Coefficients adjusted and structure selected so as to minimize (empirical) average squared error on, respectively, fitting and selection subsets of training data	Two inputs: linear or quadratic (preassigned); all terms included	Possible inputs include 16 best outputs from preceding layer and key (preassigned) original input variables; complete elements for all pairs of possible inputs are computed	Clustering procedure is used to ensure representativeness of data subsets
PNETTR III (1979)— A. R. Barron	Coefficients and structure determined so as to minimize sum of error and model complexity terms; criterion evaluated using all training data	Two or three inputs: linear, quadratic, or cubic; various subsets of terms considered (best automatically selected); input pairs are screened prior to computing complete elements to predict which pairs will perform well	Possible inputs include variable number of outputs from immediately preceding layer (typically 64 from first layer, fewer from following layers) and all original input variables	Multiple networks may be trained for as many dependent variables; possible interactions (links) between networks are treated; faster execution and lower cost than PNETTR II while considering a greater variety of ALN structures
PNETTR IV (1982)— A. R. Barron	Coefficients and structure determined so as to minimize error (PSE) criterion; criterion evaluated using all training data	Same as PNETTR III but with improved screening on input pairs; includes cube roots and exponentials of inputs (including some outputs of preceding layers); rates of the exponentials and critical points of cube roots are determined automatically from data	Possible inputs include variable number of outputs from all preceding layers (typically 30 from first, 25 from second, ..., 5 from sixth and following layers) and all original input variables	Same as PNETTR III, but includes extensive automatic analysis of classifier ALN (confusion matrices, ROC curves, nonsymmetric density estimations of ALN outputs, results of Bayes' classifiers); final networks and classifiers outputted as ready-to-use subroutines; user friendly; fast; inexpensive

The challenge for ongoing research will be to meld an effective union between the traditional (rule-based) and ALN (inductive) approaches toward AI [58].

V. APPLICATIONS OF ADAPTIVE LEARNING NETWORKS

Table 2 summarizes recent U.S. applications of ALNs, and the following sections of this chapter detail a few of these applications with a view to illuminating the principles involved.

A. Scene Classification of LANDSAT Multispectral Scanner Data

The adaptive learning network (ALN) methodology was used to classify a LANDSAT scene into three terrain classes: water, forest, and nonforest. The U.S. Army Engineer Waterways Experimental Station provided training and evaluation sets of LANDSAT multispectral scanner data taken from the 13 October 1975 LANDSAT 2 scene located 40 km northwest of Vicksburg, Mississippi.

The training data consisted of 545 classified radiance vectors. Each radiance vector contained a mean radiance for each of four spectral bands which was obtained from 3 × 3 pixel arrays. The number of vectors in the water, forest, and nonforest classes of the training data was 8, 156, and 381, respectively. The geographical distribution of the 545 3 × 3 pixel arrays was not made available.

The ALN classifier synthesized from these training data was used to classify 52,000 pixels in an independent evaluation set. The classification results were essentially 100% accurate, based on a comparison with the actual terrain conditions. Considering the large evaluation set compared to the training data set, together with the speed and simplicity of the derived ALN, the ALN methodology was deemed to be ideally suited to rapid classification of large LANDSAT scenes [59, 60].

B. Target Recognition for Missile Guidance

The ALN synthesis methodology has been used to create a ground target image classification algorithm for infrared images representative of those obtained with seekers in tactical air-launched missiles. Using features extracted from transforms of the original image, the classification algorithm achieves range- and aspect-angle-independent separation of images that contain a specific target type from images that do not contain that type. A receiver operating characteristic (ROC) analysis of the algorithm, using 385 sample images, shows 95% detection rate, 5% false-alarm rate, and a small (< 1%) false-dismissal rate. This study examined the potential for ALNs to recognize specific targets of interest in infrared images. Particular

Table 2 Representative U.S. Applications of Adaptive Learning Networks

- Process control
  - Hot strip steel mill runout table cooling sprays
  - Crystallization processes
  - Fermentation processes
- Radar
  - Reentry vehicle trajectory prediction
  - Radar imagery target classification
  - Detection and identification of tactical targets
  - Radar pulse classification
- Passive acoustic and seismic analyses
  - Ocean platform detection and classification
  - Seismic discrimination
- Infrared
  - Target acquisition and aim-point selection
  - LANDSAT scene classification
- X-ray
  - X-ray image analysis for bomb detection in luggage
- Ultrasonics and acoustic emission
  - Ultrasonic imaging
  - Feedwater nozzle inspection
  - Turbine rotor inspection
  - Ultrasonic pipe inspection
  - Monitoring of crack-growth activity
- Eddy currents
  - Automatic bolthole inspection
  - Recirculating steam generator tubing inspection
  - Once-through steam generator tubing inspection
- Missile guidance
  - Air-to-air guidance law synthesis
- Materials
  - Radiation embrittlement modeling
  - Modeling of single-particle erosion of heat shields
  - Weld strength estimation
- Multisensor signal processing
  - Physical security systems
- Microprocessor-based hardware
  - "Smart" ultrasonic flaw discriminator
- Biomedical modeling
  - Sleep stage classification
  - Crash injury modeling
- Econometric forecasting
  - Steel shipment forecasting
  - Cost-estimating relationships

emphasis was placed on designing a target recognition algorithm that is independent of the target range and aspect angle [61, 62].

C. Missile Guidance Laws

The feasibility of using ALN techniques to provide passive implementation of modern optimal guidance laws has been demonstrated via simulations. A modified proportional navigation (MPN) optimal guidance law was used to establish the ALN training data base. (The details of MPN guidance laws are described in Ref. 63.) The resulting ALN guidance law was found to be superior to constant-gain proportional navigation and to have performance comparable to that of the ideal, but passively unrealizable MPN guidance law over the envelope of MPN launch conditions [57, 64, 65].

The MPN law was defined to be the one that provides desirable commands for missile acceleration, but for implementation the MPN law requires knowledge of passively unobservable range to target and range rate; and an estimate of the completely unobservable time to go, t_{go}, is also required. Thus the basic idea was to use MPN to produce a training data base of intercept engagements and have the ALN model the acceleration commands in this data base using, as ALN inputs, only passive-seeker observables. To perform as well as MPN, the ALN had to learn to infer the unobservable information (combinations of range, range rate, and t_{go} appearing as optimal gains in the MPN formulation) from passive observables in the missile-target engagements incorporated in the training data base. (A description of the training data base is given in Ref. 64.)

The ALN implementation of MPN showed excellent performance. In an independent six-degree-of-freedom simulation of 266 engagements of a maneuvering target in which the missile was under the control of the passive ALN guidance law, the missile "hit" the target in 212 engagements (79.7%). The corresponding figure for the active MPN guidance law was 200 engagements (75.2%), and for classical passive proportional navigation it was 159 engagements (59.8%). Because of the success of the ALN in modeling MPN guidance while requiring only passively observable inputs, this new approach is viewed as offering significant improvements over classical guidance techniques for passive systems and a promising approach for active and hybrid systems.

D. Development of a Distributed, Adaptive, Intrusion Detection System

A laboratory prototype of an adaptive, fixed-site, physical security system has been developed which incorporates a distributed microprocessor network, fiber-optic data links, and adaptive signal processing technologies [66-69]. The system achieves a high probability of detection and a low nuisance alarm rate by using ALN-based detection algorithms to perform multisensor signal processing. The decision logic simultaneously integrates the outputs of geophones and three other types of sensors (SPIR, RACON, and MILES). The detection algorithm is an "alerted classifier"; that is, energy in any of the detectors above a threshold initiates the classification of the combined waveforms as intruder or nonintruder induced. The ALN classifier is trained on representative data. The system features the ability to adapt to site-specific characteristics. The adaptation takes place by training/retraining the classifier on data collected by the ALN system at that site. This capability resides in the software resident on the system hardware. The system also has a capability of recognizing that its environment has changed and alerting the system operator accordingly.

E. Establishing Signal Processing and Pattern Recognition Techniques for In-Flight Discrimination Between Crack-Growth Acoustic Emissions and Other Acoustic Waveforms

Signal processing and pattern recognition algorithms have been developed to discriminate crack-growth acoustic emissions from other innocuous, extraneous acoustic sources. Laboratory experiments were performed to record thousands of crack-growth and noise waveforms on aircraft structural aluminum plates of different geometries and alloy compositions. The problem was separated into four stages, each solved in an automatic mode: detection of signal in background noise, windowing of various parts of the signal, feature extraction, and classification. The algorithms were designed keeping the limitations and requirements of real-time implementation in mind. The ALN methodology was used to select the most important features from the candidate feature list and to derive nonlinear classification functions. Results indicate that optimum combinations of temporal and spectral features result in significantly improved acoustic emission signal identification [70].

F. Quantitative Nondestructive Evaluation of Materials

Adaptive learning networks were first applied to quantitative nondestructive evaluation (NDE) in 1973. The combination of digital signal processing as a preprocessing step and ALNs to model time- and frequency-domain waveform features has led to a new level of performance of NDE systems. Prior to the introduction of ALNs, the detection and assessment of material defects (such as cracks in pipe welds) was based almost solely on the amplitude of a pulse-echo or through-transmission ultrasonic response.

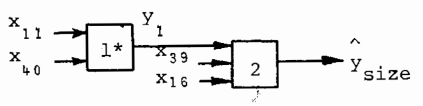
Typical candidate features that are computed from the ultrasonic (i.e., "RF") signal for possible use as ALN inputs are given in Table 3. Usually, the time-domain features are derived from the analytic envelope of the signal, estimated via the Hilbert transform. The frequency-domain features are obtained from the Fourier transform.

Typical ALNs for determining defect type ("characterization") and size are shown in Fig. 4. These models were derived using the PNETTR IV

Table 3 Typical Ultrasonic RF Waveform Features

Rise time 25-90%	Computed from the Hilbert transform analytic envelope
Pulse width at 60% amplitude	
Pulse width at 25% amplitude	
Fall time 90-25%	
Polarity at 25% on rising edge	Computed from the Hilbert transform
Polarity at 60% on rising edge	
Polarity at 90% on rising edge	
Polarity at peak	
Average phase between 25% points	
Average phase between 60% points	
Average phase between 90% points	
Energy band, analytic power spectrum at 0.156 MHz	
Energy band, analytic power spectrum at 0.313 MHz	
Energy band, analytic power spectrum at 0.469 MHz	
Energy band, analytic power spectrum at 0.625 MHz	
Energy band, analytic power spectrum at 0.781 MHz	
Energy band, analytic power spectrum at 0.938-1.094 MHz	
Energy band, analytic power spectrum at 1.250-1.406 MHz	
Energy band, analytic power spectrum at 0.156-0.625 MHz	
Energy band, analytic power spectrum at 0.781-1.250 MHz	
Energy band, analytic power spectrum at 1.406-2.500 MHz	
Energy band, RF power spectrum at 3.44-3.75 MHz	
Energy band, RF power spectrum at 3.75-4.38 MHz	
Energy band, RF power spectrum at 4.38-5.00 MHz	
Energy band, RF power spectrum at 5.00-5.63 MHz	
Energy band, RF power spectrum at 5.63-6.25 MHz	
Energy band, RF power spectrum at 3.44-4.06 MHz	
Energy band, RF power spectrum at 4.06-4.84 MHz	
Energy band, RF power spectrum at 4.84-6.25 MHz	
Energy band, RF power spectrum at 3.44-4.53 MHz	
Energy band, RF power spectrum at 4.53-6.25 MHz	
Ratio of RF energy bands: feature 27/feature 28	
Ratio of RF energy bands: feature 28/feature 29	
Ratio of RF energy bands: feature 27/feature 29	
Moment of the RF power spectrum, center of mass	
Moment of the RF power spectrum, standard deviation	
Moment of the RF power spectrum, skewness	
Moment of the RF power spectrum, kurtosis	
Moment of the analytic signal, center of mass	
Moment of the analytic signal, standard deviation	
Moment of the analytic signal, skewness	
Moment of the analytic signal, kurtosis	

(a) Sizing Model



*Mathematical form of element blocks

Single-Feature Blocks:

$$y_i = a_o + a_1x_1 + a_2x_1^2 + a_3x_1^3$$

Two-Feature Blocks:

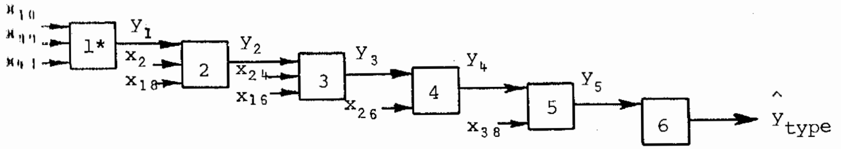
$$y_i = a_o + a_1x_1 + a_2x_2 + a_3x_1x_2 + a_4x_1^2 + a_5x_2^2 + a_6x_1^3 + a_7x_2^3$$

Three-Feature Blocks:

$$y_i = a_o + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_1x_2 + a_5x_1x_3 + a_6x_2x_3$$

$$+ a_7x_1^2 + a_8x_2^2 + a_9x_3^2 + a_{10}x_1^3 + a_{11}x_2^3 + a_{12}x_3^3$$

(b) Characterization Model



*Note: Input features are shown in Table 3. Model structure and coefficients (a's) learned from the training data.

Fig. 4 Adaptive learning network models to (a) estimate the size of surface-connected cracks, and (b) discriminate between crack and inclusion-type defects in turbine rotor bores.

algorithm. The following are representative examples of ALN NDE applications.

1. Under-Fastener Cracks in Aircraft Components

The feasibility of adaptively synthesized nonlinear signal processing techniques for characterization of ultrasonic waveforms to detect and evaluate under-fastener fatigue cracks in aluminum was demonstrated in 1976. Reliable detections and accurate size measurements over the entire flaw size range of 10 to 270 mils represented by the tested specimens was achieved. Previous NDE techniques provided no detection capability below approximately 35 mils and no size measurement capability whatever [71].

2. Multilayered Adhesively Bonded Materials

The ultrasonic waveforms obtained from pulse-echo testing of multilayered, adhesively bonded materials is a complex function of the number and composition of the layers and the adhesive bonds, multipath reflections, the incident angle of the main ultrasonic beam with respect to the material surface, and the presence of numerous spurious reflectors. ALNs can successfully detect bond-line defects, classify the types of these defects (disbonds, delaminations, porosity, etc.), and report their sizes and locations [72].

3. Austenitic Pipe Welds

ALNs have been synthesized for detecting, locating, and classifying flaws produced by intergranular stress-assisted corrosion cracking in austenitic pipe welds. These cracks are a critical problem in the operation of reactors for nuclear power generation. The detection of the cracks is complicated by the presence of numerous geometrical reflectors and by the high attenuation of acoustic signals propagated through stainless steel. The geometrical reflectors consist mainly of the stainless steel grain structure and other surfaces induced in the machining and welding steps. True defects can be accurately discriminated by ALNs, and information regarding flaw sizes, locations, and orientations can be extracted from the ultrasonic waveforms. As a direct outgrowth of this project and related work, a nondestructive evaluation "smart" automatic pipe inspection system for use during in-service ultrasonic inspections of nuclear power plant piping has been developed [73].

4. Turbine Rotor Boresonic Inspection

Advanced signal processing techniques have been applied to the evaluation of ultrasonic data collected from nuclear power plant turbine rotors. The data have been analyzed and algorithms developed to increase the transducer output signal-to-noise ratio by both temporal and spatial beamforming techniques. A smart signal processing system based on ALNs for turbine rotor bore inspection has been developed [74].

5. Spot Weld Strength

An NDE system has been developed to measure and document the strengths of resistance spot welds in automobile bodies. Experimental instrumentation was used to measure and record a variety of welding variables, including electrical, mechanical, and acoustic emission signals. Sheet metal coupons, welded and monitored in laboratory and plant environments, were tested destructively to determine their tensile shear strengths. Using a data base consisting of the waveforms recorded during the welding process and the net strengths obtained from the destructive tests, ALN techniques were used to synthesize models to predict net strength from the instrumented signals. The feasibility of estimating the strengths of resistance spot welds from measurements made during the welding operation was established based on the results of modeling the strength of more than 600 welds [75].

6. Molten Metal Inspection

In the history of nondestructive evaluation of materials, attention has focused almost exclusively on detection (and, more recently, characterization) of flaws in materials inspected in their solid states. Procedures and a hardware system for detection of flaws in molten aluminum have been developed. With this system, flaws can be detected and removed before the metal solidifies, greatly reducing costs of production and significantly improving product reliability [76].

G. Modeling of Behavior of Materials

The behavior of materials has been characterized using the ALN method. The following applications are representative.

1. Single-Particle Heat Shield Erosion Analysis

The applicability of ALNs to modeling single-particle heat shield erosion test data has been investigated. The erosion mass loss from carbon-carbon composites was modeled successfully in terms of material properties, manufacturing process variables, and test parameters. In addition, crater core depth and radius were successfully modeled [77].

2. Radiation Embrittlement

Radiation embrittlement toughness curves and the variability in fracture properties of nuclear reactor steels have received research attention in the nuclear power industry. A large data base for unirradiated materials has been assembled to permit analysis of fracture-toughness characteristics of pressure vessel steels. A reduced irradiated data base has also been acquired. ALNs have been used to model successfully a measure of toughness-test response which, in addition to being a function of temperature and neutron fluence, depends on irradiation time, neutron energy density, radiation temperature, material properties, reflecting the microstructure, and chemical impurity content [78].

REFERENCES

1. N. Wiener, A. Rosenblueth, and J. Bigelow, Behaviour, Purpose, and Teleology, 1943. [Appears in Modern Systems Research for Behavioral Scientists, W. Buckley (Ed.), Aldine, Chicago, 1968, pp. 221-226.]
2. N. Wiener, Cybernetics, Wiley, New York, 1948.
3. C. W. Gwinn and R. L. Barron, Recent Advances in Self-Organizing and Learning Controllers for Aeronautical Systems, Proc. AGARD Symp. Appl. Digital Computers to Guidance and Control, London, June 2-5, 1970.
4. W. R. Ashby, Design for a Brain, Wiley, New York, 1952.
5. W. R. Ashby, Introduction to Cybernetics, Wiley, New York, 1956.
6. R. J. Lee, Self-Programming Information and Control Equipment, Melpar, Inc., Falls Church, Va., 1959.
7. R. J. Lee, Generalization of Learning in a Machine, Proc. 14th Natl. Meet. ACM, Sept. 1-3, 1959.
8. L. O. Gilstrap, Jr., and R. J. Lee, Learning Machines, Proc. Bionics Symp., Air Force Wright Air Development Division, WADD TR 60-600, Sept. 1960, pp. 437-450.
9. R. J. Lee, L. O. Gilstrap, Jr., R. F. Synder, and M. J. Pedelty, Theory of Probability State Variable Systems, 6 vols., Adaptronics, Inc., Final Technical Report, Air Force Avionics Laboratory, ASD-TDR-63-664, Dec. 1963.
10. M. A. Ostgaard and L. M. Butsch, Adaptive and Self-Organizing Flight Control Systems, Aerosp. Eng., Sept. 1962.
11. R. J. Lee and R. F. Snyder, Functional Capability of Neuromime Networks for Use in Attitude Stabilization Systems, Adaptronics, Inc., Final Technical Report, Air Force Aeronautical Systems Division, ASD-TDR-63-549, 1963, AD 429 116.
12. R. L. Barron et al., Self-Organizing Spacecraft Attitude Control, Adaptronics, Inc., Final Technical Report, Air Force Flight Dynamics Laboratory, AFFDL-TR-65-141, 1965, AD 475 167.
13. Anonymous, Design, Fabrication, and Flight Testing of Self-Organizing Flight Control System, Adaptronics, Inc., Final Technical Report, Air Force Flight Dynamics Laboratory, AFFDL-TR-70-77, June 1970.
14. R. L. Barron, Adaptive Flight Control Systems, in Principles and Practice of Bionics, AGARD Conf. Proc. 44 (1968), Technivision Services, Slough, England, 1970, pp. 119-167.
15. R. L. Barron, Self-Organizing Control of Aircraft Pitch Rate and Normal Acceleration, Adaptronics, Inc., Final Technical Report, Air Force Flight Dynamics Laboratory, AFFDL-TR-66-41, 1966, AD 801 157.

16. R. L. Barron, Self-Organizing and Learning Control Systems, AD 811 244, in Cybernetic Problems in Bionics (1966 Bionics Symposium), Gordon and Breach, London, 1968, pp. 147-203.
17. R. L. Barron et al., Analysis and Synthesis of Advanced Self-Organizing Control Systems, Adaptronics, Inc., Final Technical Report, Air Force Avionics Laboratory, AFAL-TR-67-93, Apr. 1967, AD 813 918.
18. R. L. Barron and R. M. McKechnie III, Design Principles for Self-Organizing Control System Flight Hardware, Proc. 19th Ann. NAECON, May 1967, pp. 465-473.
19. R. L. Barron et al., Synthesis of a Spacecraft Probability State Variable Adaptive Control System, Adaptronics, Inc., Final Project Report, NASA Goddard Space Flight Center, June 1967.
20. R. L. Barron, Self-Organizing Controller (Mark V), U.S. Patent No. 3,519,998, Re. 671, 743, July 7, 1970.
21. R. L. Barron, Self-Organizing Control: The Next Generation of Controllers, Control Eng., Part I: The Elementary SOC, Feb. 1968, pp. 70-74; Part II: The General Purpose SOC, Mar. 1968, pp. 69-74.
22. R. L. Barron, Analysis and Synthesis of Advanced Self-Organizing Control Systems—II, Adaptronics, Inc., Final Technical Report, Air Force Avionics Laboratory, AFAL-TR-68-236, Sept. 1968, AD 840 295.
23. R. L. Barron, Self-Organizing Controller with Constrained Performance Assessment, U.S. Patent No. 3,591,778, Nov. 5, 1968.
24. R. L. Barron et al., Application of Self-Organizing Control Techniques to High-Performance Missiles, Adaptronics, Inc., Final Technical Report, Naval Air Systems Command, NASC, Apr. 1969, AD 852 869.
25. R. L. Barron et al., Self-Organizing Control of Advanced Turbine Engines, Adaptronics, Inc./Hamilton Standard Division of United Aircraft Corporation, Final Technical Report, Air Force Aeropropulsion Laboratory, AFAPL-TR-69-73, Aug. 1969, AD 857 616.
26. R. L. Barron, Self-Organizing Controller (Mark III), U.S. Patent No. 3,460,096, Aug. 5, 1969.
27. D. Cleveland, R. L. Barron, et al., Research and Development on Self-Organizing Control Systems for Air Launched Missiles, Adaptronics, Inc., Final Technical Report, Naval Air Systems Command, Apr. 1970, AD 867 918.
28. R. L. Barron and J. R. Gouge, Jr., Redundant Self-Checking, Self-Organizing Control System, U.S. Patent No. 3,593,307, July 13, 1971.
29. Anonymous, Application of Self-Organizing Control to Remotely Piloted Vehicles, Adaptronics, Inc., Final Technical Report, Air Force Aeronautical Systems Division, ASD XR-72-19, Apr. 1972.
30. R. L. Barron, K. S. Kelleher, and G. C. Vieth, Jr., Self-Programming Antenna Tracking System, U.S. Patent No. 3,680,126, July 25, 1972.
31. R. L. Barron and R. A. Gagnon (Major, USAF), Application of Self-Organizing Control to Remote Piloting of Vehicles, Remotely Manned Systems, Proc. NASA/Cal. Tech. First Natl. Conf. Remotely Manned

Systems, E. Heer (Ed.), Sept. 13-15, 1972, Pasadena, Calif., pp. 409-422. (Publ. 1973)

32. D. Cleveland, R. L. Barron, J. R. Binkley, Jr., and L. O. Gilstrap, Jr., RPV/Self-Organizing Control Demonstration System, Vol. I: SOC Equation Development, Logic Configurations, and Control Modes; Vol. II: Hardware Description, System Operation and Maintenance, and RPV Simulation, Adaptronics, Inc., Final Technical Report, Aerospace Medical Research Laboratory, AMRL TR-73-66, June 1973.

33. R. L. Barron and D. Cleveland, Self-Organizing Control System, U.S. Patent No. 3,794,271, Feb. 26, 1974.

34. R. F. Snyder, R. L. Barron, et al., Advanced Computer Concepts for Intercept Prediction, Vol. I: Conditioning of Parallel Networks for High-Speed Prediction of Re-entry Trajectories, Adaptronics, Inc., Final Technical Report, Army Nike-X Project Office, Redstone Arsenal, Ala., Nov. 1964.

35. L. O. Gilstrap, Jr., An Adaptive Approach to Smoothing, Filtering, and Prediction, Proc. 1969 NAECON, Dayton, Ohio, May 1969, pp. 275-280.

36. L. O. Gilstrap, Jr., Keys to Developing Machines with High-Level Artificial Intelligence, ASME Paper 71-DE-21. (Presented at ASME Design Eng. Conf., New York, Apr. 19-22, 1971.)

37. R. L. Barron, Inference of Vehicle and Atmosphere Parameters from Free-Flight Motions, AIAA J. Spacecraft Rockets 6(6):641-648 (1969). (Paper presented at AIAA Guidance, Control Flight Dyn. Conf., Huntsville, Ala., Aug. 1967.)

38. S. H. Brooks, A Discussion of Random Methods for Seeking Maxima, Oper. Res. 6:244-251 (1958).

39. J. Matyas, Random Optimization, Avtomatika Telemekhanika (Automation and Remote Control) 26:246-253 (1965).

40. A. N. Mucciardi, Neuromime Nets as the Basis for the Predictive Component of Robot Brains, Cybernetics, Artificial Intelligence, and Ecology, H. W. Robinson and D. E. Knight (Eds.), Spartan Books, Bensalem, Pa., 1972, pp. 159-193. (Presented at 4th Ann. Symp. Am. Soc. Cybern., Washington, D.C., Oct. 1970.)

41. L. Devroye, A Bibliography on Random Search, Technical Report SOCS-79.9, McGill University, Montreal, May 1979.

42. R. L. Barron, Adaptive Transformation Networks for Modeling, Prediction, and Control, Proc. IEEE/ORSA Joint Natl. Conf. Major Syst., Anaheim, Calif., Oct. 25-26, 1971.

43. R. L. Barron, Theory and Application of Cybernetic Systems: An Overview, Proc. IEEE 1974 Natl. Aerosp. Electron. Conf., May 1974, pp. 107-118.

44. A. G. Ivakhnenko, Polynomial Theory of Complex Systems, IEEE Trans. Syst. Man Cybern. SMC-1(4):364-378 (1971).

45. A. G. Ivakhnenko, G. J. Krotov, and V. N. Visotsky, Identification of the Mathematical Model of a Complex System by the Self-Organization

Method, Theoretical Systems Ecology, Academic Press, New York, 1969, pp. 325-352.

46. R. L. Barron, Learning Networks Improve Computer-Aided Prediction and Control, Comput. Des., Aug. 1975, pp. 65-70.

47. A. N. Mucciardi and E. E. Gose, An Automatic Clustering Algorithm and Its Properties in High-Dimensional Spaces, IEEE Trans. Syst. Man Cybern. SMC-2(2):247-254 (1972).

48. A. N. Mucciardi, Information Filtering Using the CLUSTR Algorithm, Proc. Computer Image Processing and Recognition Conf., Univ. of Missouri, Columbia, 1972, Vol. 2, pp. 15-3-1 to 15-3-8.

49. R. L. Barron, Three Approaches for Extrapolation with Black Box Models in Process Control Systems, Proc. 11th Conf. on Use of Digital Computers in Process Control, Louisiana State University, Baton Rouge, La., Feb. 25-27, 1976.

50. A. N. Mucciardi, New Computational Techniques in the Evaluation of Drug-Induced EEG Changes, Psychotropic Drugs and the Human EEG: Modern Problems in Pharmacopsychiatry, Vol. 8, Turan M. Itil (Ed.), Karger, Basel, 1974, pp. 350-377.

51. A. N. Mucciardi, Self-Organizing Probability State Variable Parameter Search Algorithms for Systems That Must Avoid High-Penalty Operating Regions, IEEE Trans. Syst. Man Cybern. SMC-4(4):350-362 (1974).

52. H. Akaike, Statistical Predictor Identification, Ann. Inst. Stat. Math. 22:203-217 (1970).

53. H. Akaike, Information Theory and an Extension of the Maximum Likelihood Principle, in Proceedings of the Second International Symposium on Information Theory, B. N. Petrov and F. Csaki (Eds.), Akadémia Kiádo, Budapest, 1972, pp. 267-281.

54. A. R. Barron, Properties of the Predicted Square Error: A Criterion for Selecting Variables, Ranking Models, and Determining Order, Adaptronics, Inc., McLean, Va., 1981.

55. C. L. Mallows, Some Comments on C_p, Technometrics 15:661-675 (1973).

56. J. Bibby and H. Toutenburg, Prediction and Improved Estimation in Linear Models, Wiley, New York, 1970, pp. 13-15.

57. F. J. Cook and J. N. Craig, Adaptive Learning Networks and Image Processing for Missile Guidance, Proc. Soc. Photo-Optical Instrum. Eng., Vol. 238: Image Processing for Missile Guidance, 1980, pp. 293-301.

58. R. L. Barron, Adaptive Learning Network Algorithms: Bringing a New Innovation to Market, seminar presentation to Rensselaer Polytechnic Institute, Oct. 14, 1982.

59. P. Horvath and F. J. Cook, Scene Classification of Landsat Multispectral Scanner Data by Means of the Adaptive Learning Network Methodology, Proc. IEEE Comput. Soc. Conf. on Pattern Recognition and Image Processing, Aug. 1981, pp. 473-477.

60. F. J. Cook, Analysis of LANDSAT Radiance Variance Sets, Adaptronics, Inc., Final Technical Report, Army Engineer Waterways Experimental Station, Contract DACW39-78-M-4955, Dec. 1978.
61. J. N. Craig, F. J. Cook, and M. F. Whalen, A Priori Training of Guidance and Control Algorithms for Tactical Missiles, Task II—Target Acquisition and Aim-Point Selection, Adaptronics, Inc., Final Technical Report, Air Force Armament Technology Laboratory, AFATL-TR-80-95, Aug. 1980.
62. J. N. Craig, M. F. Whalen, and F. J. Cook, Target Recognition for Missile Guidance Using Adaptive Learning Networks, Proc. Soc. Photo-Optical Instrum. Eng., Vol. 238: Image Processing for Missile Guidance, 1980, pp. 309-315.
63. T. L. Riggs, Jr., Linear Optimal Guidance for Short Range Air-to-Air Missiles, Proc. IEEE 1979 Natl. Aerosp. Electron. Conf., p. 757.
64. J. N. Craig, R. L. Barron, and F. J. Cook, A Priori Training of Guidance and Control Algorithms for Tactical Missiles, Task I—Air-to-Air Guidance Law Implementation, Adaptronics, Inc., Final Technical Report, Air Force Armament Technology Laboratory, AFATL-TR-80-102, Sept. 1980.
65. J. N. Craig, R. L. Barron, and F. J. Cook, A New Class of Guidance Laws for Air-to-Air Missiles, 3rd Meet. Coordinating Group on Modern Control Theory, Redstone Arsenal, Ala., Oct. 1981.
66. F. A. Bick, M. Inbar, F. J. Cook, and F. J. Kline, The Utilization of Emerging Technologies in Physical Security Systems: Phase I Status Report—System Feasibility, Effects Technology, Inc./Adaptronics, Inc. Draft Final Technical Report for Defense Nuclear Agency, Contract DNA001-80-C-0271, Nov. 1980.
67. F. J. Kline and F. J. Cook, The Utilization of Emerging Technologies in Physical Security Systems: Adaptive Learning Network Performance Demonstration—Phase II, Adaptronics, Inc., Draft Final Technical Report for Effects Technology, Inc., Nov. 1980.
68. A. R. Hunt, F. J. Cook, G. S. Lapman, A. H. Sanders, and F. E. Lanham, Development of a Distributed, Adaptive, Intrusion Detection System: Hardware and Software Operation, Proc. Conf. on Crime Countermeasures and Security Technology, May 11-13, 1983.
69. F. J. Kline, F. J. Cook, A. R. Hunt, and F. E. Lanham, Development of a Distributed, Adaptive, Intrusion Detection System: Algorithm Operation and Field Test Results, Proc. Conf. on Crime Countermeasures and Security Technology, May 11-13, 1983.
70. P. Horvath and F. J. Cook, Establishing Signal Processing and Pattern Recognition Techniques for Inflight Discrimination Between Crack-Growth Acoustic Emission and Other Acoustic Waveforms, in Review of Progress in Quantitative Nondestructive Evaluation, D. O. Thompson and D. E. Chimenti (Eds.), Plenum Press, New York, 1982, pp. 463-473.

71. R. Shankar, A. N. Mucciardi, D. Cleveland, W. E. Lawrie, and H. L. Reeves, Adaptive Nonlinear Signal Processing for Characterization of Ultrasonic NDE Waveforms; Task 2: Measurement of Subsurface Fatigue Crack Size, Adaptronics, Inc., Final Technical Report, Air Force Materials Laboratory, AFML-TR-76-44, Apr. 1976.
72. M. H. Loew, J. M. Fitzgerald, A. N. Mucciardi, R. K. Elsley, and G. A. Alers, Exploratory Development of Adhesive Bond Flaw Detection, Adaptronics, Inc., Final Technical Report, Air Force Materials Laboratory, AFML-TR-78-206, Dec. 1978.
73. A. N. Mucciardi, ALN 4000 Ultrasonic Pipe Inspection System, in Nondestructive Evaluation Program: Progress in 1981, Electric Power Research Institute Report NP-2088-SR, Jan. 1982.
74. M. F. Whalen et al., Advanced Signal Processing of Turbine Rotor Bore Waveforms, Adaptronics, Inc., Final Technical Report, Electric Power Research Institute, Contract RP502-8, Aug. 1981.
75. D. Cleveland, B. Decina, J. M. Jamieson, and A. N. Mucciardi, Development of Adaptive Learning Network Models to Predict the Tensile Shear Strengths of Resistance Spot Welds, Adaptronics, Inc., Final Technical Report, General Motors Corp., Contract MD 874157, Sept. 1981.
76. Anonymous, ALN 5000 Final Software Documentation for the 4M System$^{\text{T.M.}}$, June 1982; 4M System$^{\text{T.M.}}$ Users Manual, June 1982; 4M System$^{\text{T.M.}}$ Hardware Documentation, June 1982; Adaptronics, Inc. for Reynolds Metals Company.
77. J. N. Craig and F. J. Cook, Single Particle Erosion Modeling Using Adaptive Learning Networks, Adaptronics, Inc., Final Technical Report, Defense Advanced Research Projects Agency, Contract MDA903-80-C-0291, 1981.
78. J. N. Craig and F. J. Cook, Application of ALN Modeling to Radiation Embrittlement, Adaptronics, Inc., Final Technical Report, Electric Power Research Institute, Contract RP1553-2, 1981.