INTELLIGENT SENSING AND CONTROL OF GAS METAL ARC WELDING

H. B. Smartt and J. A. Johnson
Idaho National Engineering Laboratory
Idaho Falls, Idaho 83415-2210, U.S.A.

ABSTRACT

Intelligent sensing and control is a multidisciplinary approach that attempts to build adequate sensing capability, knowledge of process physics, control capability, and welding engineering into the welding system such that the welding machine is aware of the state of the weld and knows how to make a good weld. The sensing and control technology should reduce the burden on the welder and welding engineer while providing the great adaptability needed to accommodate the variability found in the production world.

This approach, accomplished with application of AI techniques, breaks the tradition of separate development of procedure and control technology.

INTRODUCTION

Conventional, automated processing generally involves sophisticated sensing and control techniques applied to various processing parameters. In arc welding, for example, these parameters may include current, voltage, welding speed, and various other factors deemed important to process reproducibility. The attributes actually desired in the product, such as properties and quality, are normally controlled by some form of statistical process control. Thus, we can think of conventional practice in terms of real control (implemented by hardware-based systems) applied to the process, and virtual control (implemented by people/paper systems) applied to the product.

The prime objective of intelligent sensing and control is to make a good product the first time. The approach involves integrating off-line inspection into the process via sensors for both process and product state, in combination with appropriate control technology to drive the product state to the desired point. Of course, the objective is the same as for existing conventional technology; the difference is that the time constant of a real control loop should be orders of magnitude shorter than that of a virtual loop. Thus, fewer rejects should be produced. In addition, intelligent sensing and control should be less expensive, mainly due to reduced labor cost.

We can now examine the tools necessary to implement intelligent sensing and control. These include (in no particular order) control theory, process modeling, sensing, and artificial intelligence, in addition to the normal tools of welding engineering and materials science.

Control Theory provides a formal means of developing a strategy to obtain the desired product state and suitable process dynamics. The foundation of control theory is a body of techniques that allow convergence, stability, robustness, frequency response, and other factors to be predicted and obtained. Thus, design of a controller has a mathematical engineering basis.

Process Modeling provides a means of incorporating both first principal and empirical information into a control strategy. Models may be used off-line to evaluate and tune a controller in a simulation. They may also be used to develop transfer functions of a process for use in formal controller design, or to provide maps between input and output parameters. Process models are an important bridge between what is known and what is desired.

Process Sensing is the necessary means of identifying the state of both the process and the product. Controllers operate by comparing actual process output (in terms, for example, of product attributes) to the desired output. The difference or error is used to calculate the appropriate control input to the process. Thus sensors are normally needed for each of the various parameters or attributes chosen as system outputs in the design of the controller.

Artificial Intelligence (AI) is a body of techniques that attempts to mimic biological intelligence. These various techniques, including expert systems, artificial neural networks, and fuzzy logic, are used for a variety of interesting applications including image and signal processing, selection of nominal parameters, and dynamic control. More is said about this topic in the next section.

Intelligent sensing and control involves application of all of the above tools to control of both the desired operation of a process and the attributes of the product of that process.

THE ROLE OF AI TECHNIQUES

Before turning to the question of what this means to welding, we believe it is worth while to comment on some of the AI techniques mentioned above. Although a well balanced paper would expound on all four of the above tools, we have selected AI techniques for comment because we believe that some simple observations are of sufficient value to include in this short paper. We apologize for the fact that what follows is in no way a comprehensive review of this vast field; there is a story to tell, and we have selected references accordingly.

Papers and presentations on AI methods and applications generally discuss the nuts and bolts of the machine, but gloss over what it is that the machine does. Consider, for example, artificial neural networks. Anderson and Rosenfeld [1] and more recently Simpson [2] provide excellent starting places for the study of artificial neural networks, while MacGregor presents an excellent overview of biological neurons and neural networks [3]. Anderson and Rosenfeld provide an historical view, and Simpson gives possibly the best presentation now available of the most important algorithms being used, based on classification into four groups in terms of feed forward and recursive network structures and supervised and unsupervised learning methods.

Perhaps the most popular algorithm for artificial neural networks is the feed forward network using backpropagation for learning of the interconnection weights. In one of their introductory papers on backpropagation, Rumelhart et al. [4] employ the XOR Boolean logic problem as a test case. In this problem, there are two binary inputs and one binary output; if the inputs are both 1s or 0s, the output will be 0 and if the two inputs are not both the same (i.e., 1,0 or 0,1) the output is 1. A feed forward network having two binary inputs (X1 and X2), three artificial neurons in the hidden layer, and one output neuron, where all neurons include the nonlinear sigmoid activation function, is taught the XOR problem. This network is able to learn a good solution to the problem in several hundred iterations of supervised learning. The neural network output OUT in the above example may be calculated algebraically from:

$$OUT = f(V1*f(X1*W11 + X2*W12) + V2f(X1*W21 + X2*W22)) \qquad (1)$$

where f is the sigmoid function operator.

The resulting output is plotted as a function of X1 and X2 in Figure 1, a plot of the input to output mapping function learned by the network. The function is continuous and is a good solution for binary inputs, but a poor solution for intermediate inputs. The main point to be made is that an artificial neural network is a mechanism for generating an input to output mapping function, given a set of discrete (not necessarily binary) data points. This point is discussed by Gallant and White [5] and more recently Cardaliaguet and Euvrard [6]. As is noted below, fuzzy logic systems are also mechanisms for generating an input to output mapping function, and a strong argument may be made that expert systems are also such mechanisms, though in this latter case the mapping function is not continuous.

Expert systems and fuzzy logic systems both differ from artificial neural networks in that they use conditional logic statements as the input data. The difference between the two methods is that expert systems normally give yes/no or black/white types of output, whereas fuzzy logic systems admit degrees of maybe or levels of gray as inputs and outputs.

Consider the XOR problem discussed above. The conditional logic statements were stated as: **if** the inputs are both 1s or 0s, **then** the output will be 0; and **if** the two inputs are not both the same (i.e.. 1,0 or 0,1), **then** the output is 1. A fuzzy logic solution [7] is shown in Figure 2. This solution has the interesting features that completely ambiguous inputs (X=0.5, Y=0.5) give an ambiguous output (Z=0.5), and the plot is symmetrical. This is a much more logical answer than that obtained by the artificial neural network above. Indeed, it is an enlightening exercise to attempt to reproduce the symmetry of the fuzzy logic solution with an artificial neural network (but one that we will leave to the reader).

At least two additional observations may be made about the above example. One, that method of evaluating fuzzy logic allows one to write the solution as an algebraic function, in the same manner as was possible for the artificial neural network solution. (This is not possible for the standard fuzzy logic paradigm [8], which requires some scheme of defuzzification to calculate a solution. The conventional approach normally generates non-continuous functional solutions.) For this reason, this new method is called "continuous fuzzy logic" to differentiate it from the more standard paradigm.. Two, the solution to continuous fuzzy logic may be represented in a network form, Figure 3, in which the inputs X and Y and the output Z are as discussed above. The elements in the network hidden layer are directly associated with the mathematical evaluation of the individual **if** conditions. The hidden elements-to-output element weights are directly associated with the action to be taken. The output element contains a summation operator. (The reader may want to compare this network with the corresponding structure for conventional fuzzy logic shown in Kosko, figure 11.8, p. 392 [8].) It is probably not appropriate to call this a neural network, but it is proper to refer to it as a connectionist network, thus recognizing that it belongs to a superset that contains artificial neural networks.

In the case of an artificial feed forward neural network, the number of elements required in the hidden layer(s) is determined by the number of degrees of freedom required to adequately map the control law. Determining the actual number required is generally accomplished on a trial and error basis. For a fuzzy

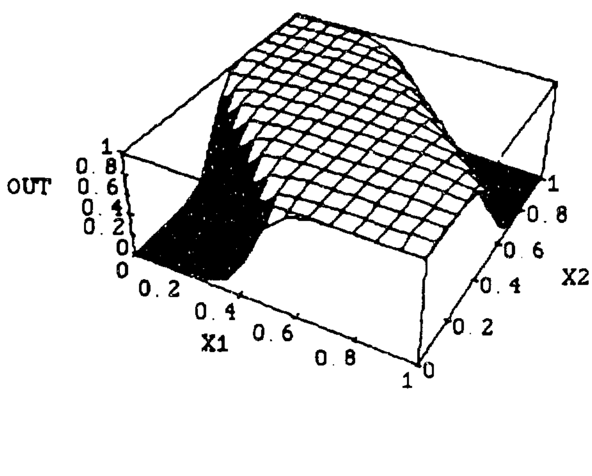

Figure 1. Plot of functional input-to-output mapping learned by feed forward neural network as solution to XOR problem.

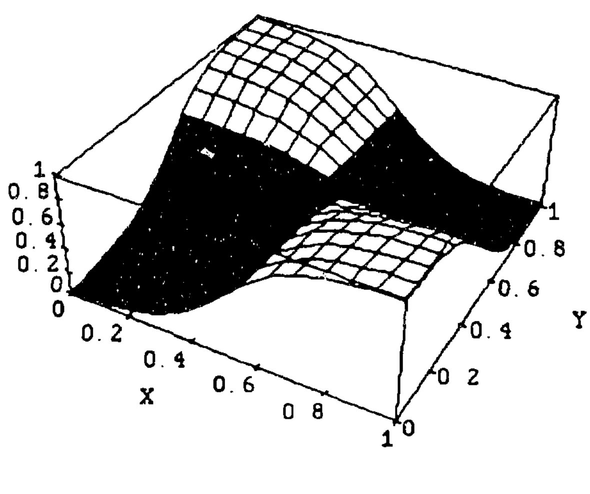

Figure 2. Continuous fuzzy logic solution to the XOR problem.

network, a hidden layer element is associated with each conditional logic rule, so establishing the number of hidden layer elements is no problem.

By now you realize that the fuzzy logic XOR solution is a mapping function. An expert system solution is also easily generated, consisting either of isolated 'peaks and holes' or of a stepped surface, depending upon how you go about it. So we see that all three methods discussed are, in fact, mapping function generators.

The next logic step involves recognition that input to output mapping functions are in fact the same as transfer functions. Transfer function based analysis and design methods form a major portion of dynamic systems analysis [9], signal analysis [10], and control theory; thus, the route to formal integration of AI techniques into these other disciplines is available, but unfortunately few have attempted to exploit it. Fortunately, Ydstie [11], and especially Narendra [12] have, and the reader may start with them to explore this interesting topic.

It may be noted that the mapping function generated by the fuzzy logic system, Figure 2, lacks the faceted appearance normally seen in such functions, for example Kosko figure 9.4a, p. 343 [8]. Figure 2 was generated using a continuous fuzzy logic algorithm [13] that is not normally seen, but which has apparently been derived at least three times by various workers. Continuous fuzzy logic readily allows generation of continuous mapping functions, at least in part, by elimination of the defuzzification step required by the more standard algorithms [8]. An important side benefit is a significant reduction in the amount of computer code required for implementation. But more important from a control standpoint is the effect of using a continuous function as a control law.

Figure 4 shows a mapping function generated using continuous fuzzy logic that describs the control law for a one-dimensional tracking problem similar to seam tracking in welding. The logic used to generate the mapping function is based on the input X being the tracking error, the input Y being the first derivative with respect to time of the tracking error, and the output Z being the control input to the system. Thus, this controller operates as a proportional-derivative controller.

Next, we will examine control of the tracking problem using continuous fuzzy logic.

By making changing the mapping function, the resulting controlled system dynamics may be tuned in a manner that is equivalent to a conventional controller. For example, Figure 5 shows tracking error (gap) as a function of time. As the function's partial derivative with respect to X is increased, the

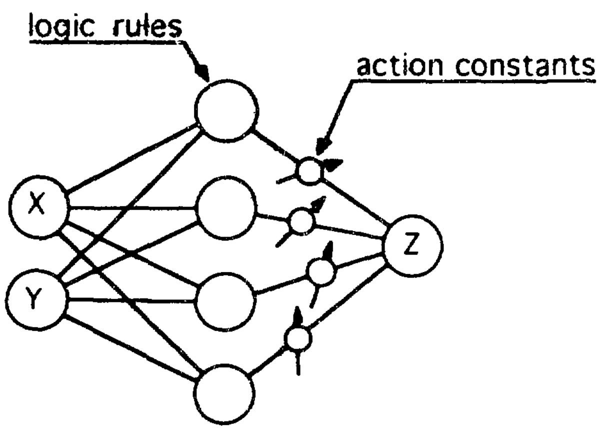

Figure 3. Network representation of continuous fuzzy logic.

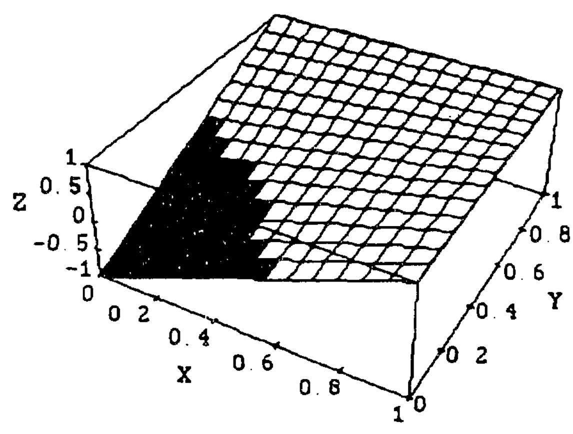

Figure 4. Control law for one-dimensional tracking generated using continuous fuzzy logic.

convergence rate increases. The effect is to increase the time for convergence until overshoot is obtained as shown in Figure 5. Figure 6 shows the steady state solution obtained (in this problem a small positive error (gap) is desired); the steady state value may be varied as the amplitude of the mapping function value is changed for $X = Y = 0$.

There are lessons to be learned from this simple example. One, all controllers need to be properly tuned, even intelligent ones. There has been considerable hype in the popular press to the effect that fuzzy logic controllers are the panacea for applications where the system transfer functions are not readily available, but "expert" knowledge is available. A recent IEEE video tape on fuzzy logic [14] disclosed that a certain commuter train in Japan that uses a fuzzy logic controller required approximately eight years of controller development including some 350,000 computer simulations for proper tuning. This situation would not be tolerated by industry in this country; the solution is development of engineering tools for controller tuning applicable to fuzzy logic and other AI-based controllers. Two, the partial derivatives of the mapping function with respect to the inputs are equivalent to the gains in a conventional controller. Mapping functions should thus be continuous. Considerable care must be exercised with conventional fuzzy logic controllers to obtain continuous mapping functions. Three, most learning methods for artificial neural networks teach the network the amplitude of the mapping function at a given coordinate. Methods to teach control laws should also be capable of teaching mapping function partial derivatives with respect to inputs at a given coordinate.

This discussion of applying artificial neural networks or fuzzy logic systems as controllers is now summarized. Consider a more generalized formulation of the controller transfer function as:

$$G(s) = K_1(\ldots)_1 + K_2(\ldots)_2 + K_3(\ldots)_3 + \ldots + K_n(\ldots)_n \quad (2)$$

Replacing such a controller by a connectionist network may be accomplished by formulating a network having an input dimension, D, associated with each of the above n gains. Also associated with each input will be a signal preprocessor $(\ldots)_n$. Using such a formulation, given linear activation or membership functions, the derivatives of the network mapping function, F ,with respect to the network input dimensions D are:

$$\partial F/\partial D_n = K_n \quad (3)$$

where K_n is the local gain associated with the nth input dimension. The connectionist network will have an output dimension associated with each of the conventional controller outputs.

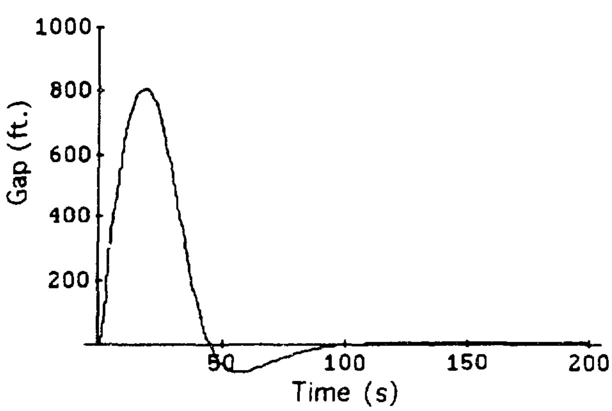

Figure 5 Tracking error (gap) showing overshoot for a high value of mapping function derivative with respect to error input.

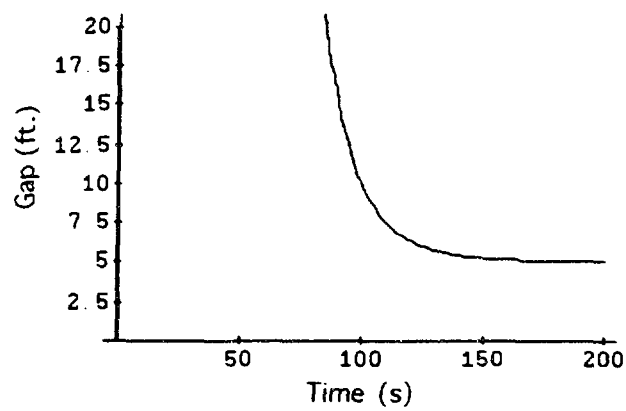

Figure 6 Tracking error (gap) showing small steady state gap for a low value of mapping function amplitude for inputs of zero.

APPLICATION TO WELDING

Recent examples of applications to welding cover the fine technical details [15,16]. We will look instead at what we consider the global problems to be, and speculate on a possible solution.

Typically, welding systems either allow the welder access to process parameters at the expense of placing a significant burden on the welder to program the system, or the system is very easy to program but the welder has limited access to the process parameters. This creates one of two problems. If the welder has easy access to the process parameters, the assumption is made that he/she has the knowledge about the physics of welding necessary to make changes to those parameters during welding to correct an undesirable situation; this is generally not true. On the other hand, systems that are easy to program generally do not provide the adaptability necessary to correct an undesirable situation during welding. The solution to both problems is identical. Do not allow the welder to change anything, but use an off-line statistical process control program to modify machine settings. Which is exactly where this paper started! Now let us suggest an alternative.

The objective may now be restated as that of developing a welding system that is very easy for the welder to program, but which has considerable adaptability to allow in-process corrections to be made for undesirable situations. (More to the point, the system should be able to predict and prevent undesirable situations.)

The alternativee approach is novel, and perhaps disturbing, for the first thing we need to do is to throw away welding procedures.

Consider the approach to development of a welding procedure. If we were doing it for the first time, we would probably want to review fundamental knowledge about the physics of heat and mass transfer in welding, microstructural development during solidification and solid state transformations, effect of thermal gradients and phase transformations on residual stresses and distortion, and also general knowledge about welding processes. From this set of data we could presumably derive a set of conditional logic statements that would at least define the qualitative characteristics of the procedure. We could also develop models describing the physics of the process. If we were able to build these rules and models into the control logic, and we can do so using fuzzy logic and artificial neural networks, then perhaps the welding machine could use this knowledge to actually develop the procedure as the weld was being made.

In order to accomplish the type of control we have been discussing, it is necessary to have a variety of sensors on the welding machine. Selection of the sensors should take into account the source of heat and mass transferred to the base metal -- melting of the base metal, dilution of the filler metal, solidification of the weld bead, microstructural development in the weld bead and heat affected zone, physical properties development, and thermomechanical distortion and residual stresses in the weldment all follow from the heat and mass transferred by the process to the weld.

Unfortunately, with few exceptions, sensors do not exist to detect weld microstructure and properties. This lack is a limit on the ultimate capabilities of sensing and control of arc welding, even for conventional control approaches. The development of advanced sensors is a significant research opportunity.

EXAMPLE

Consider the application in Einerson's paper [16]. For reasons related to the specific end product involved, it is desired to control the weld cooling rate and fill of the weld joint. The conditional logic rules are simple:

1. **If** the reinforcement is too low, **then** increase the ratio of electrode speed to welding speed.
2. **If** the reinforcement is too high, **then** decrease the ratio of electrode speed to welding speed.
3. **If** the cooling rate is too low, **then** decrease the heat transfer rate to the weld.
4. **If** the cooling rate is too high, **then** increase the heat transfer rate to the weld.

It may be shown [17] that the weld bead reinforcement (G), defined as the transverse cross-sectional area added to the weld bead by the addition of filler metal, is given by $G = \pi dS/4R$ where d is electrode diameter, S is electrode speed, and R is welding speed, for 100% deposition efficiency. The amount of heat transferred to the weld per unit length (H) is given by $H = \eta EI/R$ where η is the heat transfer efficiency, E is voltage and I is current.

The sensing requirements are defined by the logic of the problem. Two sensors are required, one to measure the transverse cross-sectional area of the weld joint and a second to measure the cooling rate of the weld bead. It is also necessary to know the welding speed, electrode speed, current, and voltage, the values of which are normally readily available in an automated arc welding system.

The difficult aspect of this example is that reinforcement and heat transfer rates are both functions of welding speed and, in gas metal arc welding, current is a function of voltage and electrode speed. This problem may be handled by deriving a model of these relationships [17], by teaching them to an artificial neural network [16], by using a look-up table, or perhaps other means. We may comment that the relationships between conventional parameters and heat and mass transfer rates are not obvious to the average welder or welding engineer. This is exactly the kind of problem that prevents the welder from adjusting heat input to the weld while maintaining constant fill rate, for example. In this work, the relationships have been used as the training set for a feedforward artificial neural network.

The resulting controller is shown in Figure 9, from [16]. The fuzzy logic controller contains simple rules that specify engineering practice. The resulting control law is tuned, as was discussed above. The artificial neural network contains (experimental) knowledge about the physics of heat and mass transfer in the process. The resulting mapping function effectively linearizes the process with respect to heat and mass transfer rates to the base metal. This system requires only that the welder set the desired weld bead cooling rate. The sensors measure the weld joint ahead of the torch and the weld bead cooling rate behind the weld pool, and the system responds to the measurement as governed by the conditional logic rules given above. This is a simple example in which knowledge of both process physics and engineering practice have been used to develop a control law and linearize a process, but the concept can be extended to include other factors.

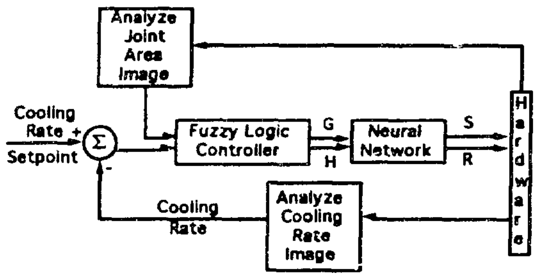

Figure 9. Process control scheme block diagram, where G is the reinforcement, H is heat input, S is electrode speed, and R is travel speed.

CONCLUSIONS

There is much more to this story. However, we will summarize by saying that what is needed is sensing and control technology that reduces the burden on the welder and welding engineer while providing the great adaptability needed to accommodate the variability found in the production world. Conventional approaches to automation of welding have been reasonably successful, but there are still significant opportunities for additional development.

It may be time to consider breaking the traditional approach that separates procedure development methods from control technology. A marriage of these two topics, accomplished with application of AI techniques, may be in order.

Finally, advanced sensor development is still needed for control of weld microstructure and properties.

ACKNOWLEDGMENTS

Appreciation is expressed to C. J. Einerson, P. L. Taylor, U.S. Wallace, and N. M. Carlson for their support. This work was supported by the U.S. Department of Energy, Office of Energy Research, Office of Basic Energy Sciences, under DOE Idaho Field Office Contract DE-AC07-76ID01570.

REFERENCES

1. J. F. ANDERSON and E. ROSENFELD, *Neurocomputing*, MIT Press, Cambridge, MA (1988).

2. P. K. SIMPSON, *Artificial Neural Systems*, Pergamon Press, New York (1990).

3. R. J. MACGREGOR, *Neural and Brain Modeling*, Academic Press, Inc., San Diego, CA (1987).

4. D. E. RUMELHART, G. E. HINTON, and R. J. WILLIAMS, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, Vol. 1, ed. D. E. Rumelhart and J. L. McClelland, MIT Press, Cambridge, MA, pp. 318-362 (1986).

5. A. R. GALLANT and H. WHITE, IEEE Int. Conf. on Neural Networks, San Diego, CA, July, pp. 657-664 (1988).

6. P. CARDALIAGUET and G. EUVRARD, Neural Networks, Vol. 5, pp. 207-220 (1992).

7. H. B. SMARTT, J. A. JOHNSON, C. J. EINERSON, and G. A. CORDES, *Intelligent Engineering Systems Through Artificial Neural Networks*, ASME Press, New York, pp. 711-716 (1991).

8. B. KOSKO, *Neural Networks and Fuzzy Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1992.

9. R. H. CANNON, *Dynamics of Physical Systems*, McGraw-Hill Publishing Co. (1967).

10. K. BEAUCHAMP and C. YUEN, *Digital Methods for Signal Analysis*, George Allen & Unwin, London (1979).

11. B. E. YDSTIE, Computers Chem. Engng. Vol. 14, No. 4,5 pp. 583-599 (1990).

12. K. S. NARENDRA and K. PARTHASARATHY, IEEE Trans. on Neural Networks, Vol. 1, No. 1, pp. 4-27 (1990).

13. T. TAKAGI and M. SUGENO, IEEE Trans. Systems, Man, and Cybernetics SMMC 15, pp. 116-132 (1985)

14. Institute of Electrical and Electronics Engineers (1991).

15. K. ANDERSEN, G. E. COOK, J. F. SPRINGFIELD, and R. J. BARNETT, *Intelligent Engineering Systems Through Artificial Neural Networks*, ASME Press, New York, pp. 717-727 (1991).

16. C. J. EINERSON, H. B. SMARTT, J. A. JOHNSON, P. L. TAYLOR, K. L. MOORE, 3rd Int. Conf. on Trends in Welding Research, Gatlinburg, TN, June (1992).

17. H. B. SMARTT, C. J. EINERSON, A. D. WATKINS, R. A. MORRIS, in *Advances in Welding Science and Technology*, ASM, pp. 461-465 (1986).