A Fuzzy Intelligent System for End-of-Line Test

Yi Lu[1], Tie-Qi Chen[1], Jianxin Zhang[1], Jacob Crossman[1], and Brennan Hamilton[2]

[1]Department of Electrical and Computer Engineering
The University of Michigan-Dearborn
Dearborn, MI 48128-1491, U.S.A.
Voice: 313-593-5028, Fax: 313-593-9967
yilu@umich.edu

[2]Diagnostic Systems Department
Powertrain Division
Ford Motor Company
Dearborn, MI 48128

Introduction

The success of the U.S. motor vehicle industry very much depends on the quality of the products it produces. As automotive electronic control systems have become more advanced and sophisticated in recent years, malfunction phenomena have also become increasingly more complicated. It is well recognized in the automotive industry that effective vehicle diagnostic systems will play a key role in the competitive market of the new century. In order to meet this challenge of improved quality control and diagnostics, the major US automotive companies are in the process of launching end-of-line test systems at every North American assembly plant. Part of the end-of-line test system is designed to collect and analyze Electronic Engine Controller (EEC) data while the vehicle is dynamically tested. Operators drive the vehicle through a preset profile and the vehicle is either passed or failed according to the data collected during the tests. The pass/fail decision is made based on two information sources – an EEC on-board tests and an EEC off-board test that is performed by the vehicle test system on EEC generated data. Our Fuzzy Intelligent System is focused on automating the off-board testing process to obtain faster and more reliable test results than are currently realized by line engineers.

As its name implies, our automated diagnostic system is based on fuzzy logic. The theory of fuzzy logic is aimed at the development of a set of concepts and techniques for dealing with sources of uncertainty, imprecision or incompleteness [Zad69, YOT87, Zim91]. Fuzzy systems have been successful in many applications including control systems when gradual adjustments are necessary [Ayo95, NHW91, RhK93, TaS85, Kan93, GKG94]. The nature of fuzzy rules and the relationship between fuzzy sets of differing shapes provide a powerful capability for incrementally modeling a system whose complexity makes traditional expert system, mathematical, and statistical approaches very difficult, and provide a more flexible, and richer representational scheme than other methods.

There are several issues that make fuzzy logic modeling desirable for automotive testing:

1. Fuzzy logic can effectively model incompletely or inaccurately described systems. Expert knowledge regarding faults is almost always incomplete and vague because of the complexity of modern vehicles. Engineering experts are usually aware of only a subset of the parameters that impact the behavior of a component, and, furthermore, it is often difficult to collect adequate data for the parameters of which they are aware.
2. Fuzzy logic is resistant to imprecision in system data. Collected data is often unreliable due to inconsistencies in manufacturing and the test process itself.

Seemingly minor changes in the manufacturing process such as changing a single worker or tool in the assembly process can lead to significant changes and/or noise in the collected data. This is especially a concern during the diagnostic modeling or training stage when we hope the system we are testing will remain relatively stable.

3. Fuzzy logic rules and knowledge is expressed in terms familiar to engineers – low, medium, hot, cold, high, low, etc.
4. Fuzzy logic decisions can be made quickly (in seconds) and, thus, do not slow down dependent manufacturing processes.
5. Fuzzy logic can model the information (possibly conflicting) from multiple engineering experts.

In the remainder of this paper, we present the major features of our Fuzzy Intelligent System including its abilities to automatically formulate rules (learn), accommodate expert knowledge, make diagnostic decisions from test data, and, finally, deal effectively with system uncertainty and data imprecision. The system was developed for use in Windows 9X, NT, 2000. A particular application to detection of vacuum leaks in vehicles is given at the end.

Rule Generation: Learning

Fuzzy reasoning is performed within the context of a fuzzy system model, which consists of control and solution variables, fuzzy sets, rule (proposition) statements, and an underlying control structure. For our diagnostic problem, the control variables are the known parameters of behavior (e.g. air intake, engine speed, etc), the solution variable(s) are the possible faults (e.g. vacuum leak), and the fuzzy sets consist of values or *terms* of the control and solution variables (e.g. high, low, medium). The rules describe the system model in terms of these variables and terms. In general, if we let $X = \{x_1, x_2, \dots, x_n\}$ be a set of n control variables, $\Sigma_i = \{\alpha_i^1, \alpha_i^2, \dots, \alpha_i^{pi}\}$ be the set of p fuzzy terms associated with control variable x_i, y be a single solution variable, and $\Gamma=\{\tau_1, \tau_2, \dots, \tau_q\}$ be the set of q fuzzy terms associated with the solution variable, then a fuzzy rule can be expressed as:

$$\textbf{IF } (x_{k1} \text{ is } a_i^{k1}) \text{ AND } (x_{k2} \text{ is } a_i^{k2}) \text{ AND ... } (x_{km} \text{ is } a_i^{km}), \textbf{ THEN } y \text{ is } t_i^k$$

Here, $m \le n$, $\{x_{k1}, xk2, ..., xkm\} \subset X$, $\{a_i^{k1}, a_i^{k2}, \dots, a_i^{km}\} \subset \Sigma_i$, and $t_i^k \in \Gamma$. The goal of the fuzzy learning process in our system is to generate a set, or *knowledge base*, of these rules that together describe the behavior of the system we wish to test. A diagram of the learning process appears below:

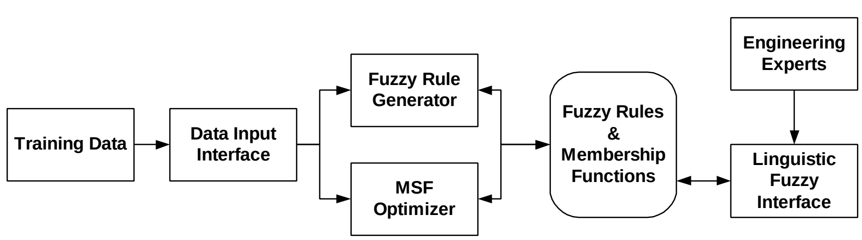


Figure 1: The fuzzy learning process. There are two ways to generate rules – automatically as defined by the steps on the left of the diagram, and through engineering knowledge as shown on the upper right of the diagram.

As can be seen from the diagram, the system can learn through one of two processes – direct translation of engineering knowledge and automatic rule generation. The first case is the simplest and is discussed first. An engineer or system operator can manually enter known operating characteristics and system fault and no-fault behaviors through our system's rule editing interface. A figure of this interface follows:

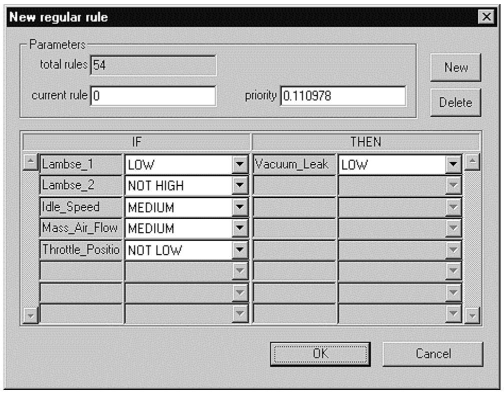
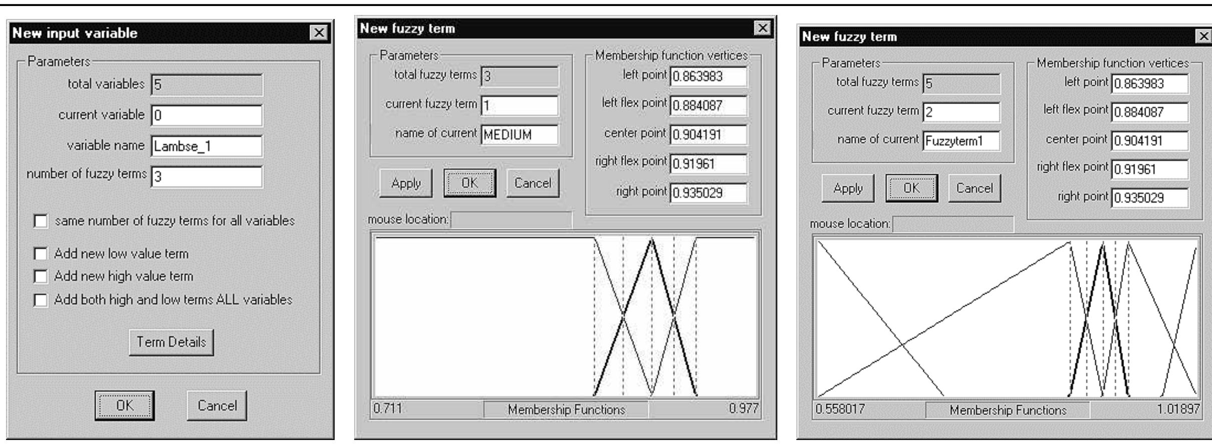


Figure 2: The rule editing interface for our system.

The dialog on the left shows all of the control variables in an "if" column on the left and the solution variable in the "then" column to the right. A user can use the drop-down selection boxes to change the condition (fuzzy term) for each control variable and to change the rule consequence (output fuzzy term). The system allows both editing of existing rules (as seen here) and the creation of new rules using the "New" button. The dialog also allows the user to change a rule's priority. The priority is a weight applied in the inference process that modifies the overall belief (or truth) value of a rule. In general, if more than one rule fires, the rule(s) with the highest priority will dominate in the calculation of the solution variable. By default, rules added by an engineering expert are given higher priority than rules generated by automatic learning because they are based on a wider variety of knowledge and experience than the automatically generated rules.

The system also allows the user to edit the properties of the control and output variables, including the number, type, and shape of it's fuzzy terms. Every fuzzy term of each system variable is defined by a special *membership function*. The membership function is used to calculate the belief that a particular data sample belongs to a given fuzzy term. The shape of these membership functions are controlled by their function type (e.g. linear, Gaussian) and a small set of *critical parameters*. Our Fuzzy Intelligent System allows the user to view and modify the fuzzy term types and critical parameters through a simple visual interface shown in figure 3.

Figure 3(a) shows the input variable and fuzzy membership function edit dialogs. Using these, the user can specify variable names, the number of fuzzy terms, membership function critical parameters, and special conditions for the boundary terms. In fuzzy logic theory, the boundaries of the minimum and maximum terms extend (as shoulders) to $-\infty$ and $+\infty$ respectively as seen in figure 3(b). In real engineering applications, data that falls very far outside of the natural range of a parameter is generally an indication of a fault. Unfortunately, if we generate a rule with an antecedent involving a boundary term and a consequence of "no-fault" (e.g. IF x_1 is LOW THEN y is NO-FAULT, where LOW and NO-FAULT are friendly names for α_i^1 and τ_i respectively), we may fire a rule that says data reading outside of usual operating conditions is normal. We allow the



(a) **(b)** **(c)**

Figure 3: Input variable editing interface (a), the membership function editing interface (b) of

user to remedy this situation by specifying that special boundary terms be created by the system just outside of the normal range of a given parameter. These boundary terms indicate LOW-OUT-OF-RANGE and HIGH-OUT-OF-RANGE, and they extend to -∞ and +∞ respectively, as theory requires. Next, letting α_i^L be the left boundary term, α_i^R be the right boundary term, and τ_R be the consequence "out-of-range", the system automatically generates the following rules for each variable having the special boundary terms:

1. **IF** (x_L is α_i^L), **THEN** y is τ_R 2. **IF** (x_L is α_i^R), **THEN** y is τ_R

Figure 3(c) shows the updated fuzzy terms after boundary conditions have been added to the variable in 3(b). If desired, the system will add only one of the boundary terms.

The interface components we have described thus far allow the user to specify a set of rules that define the test system behavior and to modify low level fuzzy set structure for system variables. These abilities are useful in particular situations when engineering heuristics outweigh available objective data. However, in most cases, this manual method of knowledge base construction is tedious. With our Fuzzy Intelligent System we provide a procedure for automatic rule generation that is combined seamlessly with the engineering heuristic knowledge described above to form a powerful learning system.

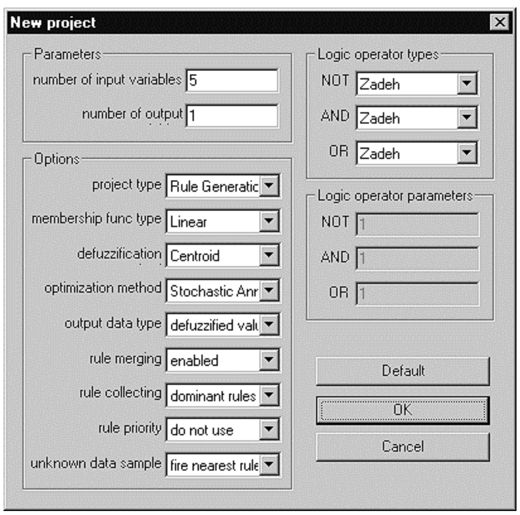


Figure 4: Fuzzy Intelligent System high-level options

The automatic learning process is controlled by a set of high level parameters supplied by the user through the dialog shown below. This dialog is the main options dialog of the system and allows the user to specify such key parameters as system mode (rule generation or inference), membership function type, defuzzification methods, and even logical operator types. This wide range of options gives the system the flexibility it needs to be molded to fit a wide range of different diagnostic tasks.

Once the user has specified the parameters that will govern the process, automatic rule generation can begin. The algorithm for automatically generating fuzzy rules is effective and efficient. It goes through a number of iterations of clustering-extracting and re-clustering. The clustering-extraction procedure first separates the training data into clusters and calculates critical parameters for the membership functions of each control variable that geometrically "bend" them toward the cluster centers. One fuzzy rule is generated for each cluster using a "winner-take-all" method, where the consequence with the largest number of data samples in the training set becomes the consequence of the rule for that cluster. If any new rule generated during the learning process conflicts with or is identical to an existing rule, the new rule is discarded. After fuzzy rules are generated for all of the clusters, procedures to merge rules, delete useless rules, and re-cluster are applied. Each iteration of clustering, critical parameter choice, merging and re-clustering may change the control variable membership functions; therefore, each iteration may generate different rules. The process ends when the membership functions – thus, the rule base – is sufficiently stable. For more complete details on this process, the reader is referred to [LuC97].

Fuzzy Inference: Testing

Fuzzy Inference (or reasoning) is the process of determining (inferring) the correct output, $y = \tau_i$, given a data sample $D = \{d_1, d_2, \ldots, d_n\}$, where τ_i is the output fuzzy term that best corresponds to D given the set of fuzzy rules available. The inference process is what is actually used by end-of-line test engineers to test vehicles in real time at the assembly plants. Below we present a high-level diagram of the inference process:

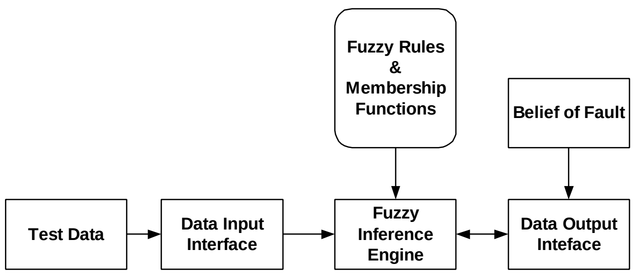


Figure 5: The fuzzy inference process.

The Fuzzy Intelligent System inference process uses a min-max technique to calculate the value for the output variable in a few steps. First, the incoming data sample is compared with the antecedent of every rule in the knowledge base, and a belief (or truth) value is calculated describing how closely the data sample conforms to each rule. The belief value for each rule is calculated by taking the minimum predicate truth of the rule applied to the data sample. Next, we (optionally) multiply the belief value for each rule by its priority to get a modified belief. The modified belief value is used to find the rule with the maximum belief for each consequence. This is the maximization part of the min-max process. After the min-max procedure is completed, we are left with a belief (0-1) for each consequence, or term of the solution variable. Defuzzification is used to calculate a final solution value from the solution variable term beliefs. Our system allows the user to specify a defuzzification method. The default, centroid method, defuzzifies the output using a weighted average of the solution variable. For a more theoretical discussion of the defuzzification process, see [LuC97].

Sometimes a data sample does not conform to any rule, as in the case when a fault the system has never seen before is presented. In these cases, no rule fires, and we say the solution is "unknown." The interpretation of the "unknown" value is application specific; however, in a diagnostic application, it indicates that a particular data sample is "suspect" (probably faulty) though the exact fault cannot be determined by the fuzzy system. This fact allows us to train the system in situation where data describing faults is difficult to obtain or unreliable. We can train the system using only known good data and consider all "unknown" samples to be indications of a fault. Alternatively, the user can specify that the nearest rule fire when a data sample is "unknown." The nearest rule is defined as the rule with the minimum Euclidean distance to the data sample (a rule's geometric position is considered to be the center of the cluster it defines).

Given a set (possibly large) of data samples, our Fuzzy System performs inference on all of them simultaneously in a matter of seconds. This is true even for knowledge bases with hundreds of rules. Furthermore, during inference, our system can collect detailed inference statistics such as which rule(s) fired, the firing frequency, the belief values for each rule, and more. Engineers and system developers can use this information to help interpret unusual cases, such as misclassifications.

Application to End-of-Line Diagnostics

Our Fuzzy Intelligent System has been used successfully to detect vacuum leaks in end-of-line testing in automotive vehicle plants. In this application, engineering experts chose five parameters important to vacuum leak detection as control variables for the fuzzy system: Lambse1, Lambse2, Throttle Position, Mass Air Flow, and Idle Engine Speed. Each control variable was associated with three, linear terms – LOW, MEDIUM, and HIGH. There was a single solution variable labeled Vacuum Leak with the same three fuzzy terms as the control variables. We trained our system on hundreds of data samples from two different vehicle models. We then tested the system on data samples not in the training set. Results of these tests are summarized in the table below

Vehicle Model	*# sample with VL*	*% VL correctly classified*	*# samples w/o VL*	*% samples w/o VL correctly classified*
Model I	**42**	**95.2**	**1651**	**98.0**
Model II	**37**	**89.1**	**1254**	**99.3**

Figure 6: Results using the fuzzy system to detect vacuum leaks (VL) in end-of-line assembly plant vehicle tests.

(model names are not used to protect proprietary information):

We get almost perfect classification of good samples. Samples from vehicles with vacuum leaks are classified correctly about 90-95% of the time. These are excellent results and have encouraged us to look for new applications for our Fuzzy Intelligent System in areas such as general vehicle diagnostics using powertrain control module (PCM) signals. For more thorough description of our Fuzzy Intelligent System and the vacuum leak detection application see [LuC97].

References

[Ayo95] M. Ayoubi, " Neuro-fuzzy structure for rule generation and application in the fault diagnosis of technical processes," in *Proc. 1995 American Control Conference, Seattle,* (USA), pp. 2757-2761, 1995.

[LuC97] Yi Lu and Tie Qi Chen, "Fast Rule Generation and Membership Function Optimization for a Fuzzy Diagnosis System," *The Tenth International conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,* Georgia, USA, June, 1997

[NHW91] H. Nomura, H. Ichihashi, T. Watanabe, "A Self-Tuning Method of Fuzzy Reasoning Control by Descent Method," *Proc. of 4th IFSA Congress, Brussels,* Vol. Eng. pp. 155-158, 1991.

[RhK93] F. C.-H. Rhee and R. Krishnapuram, " Fuzzy rule generation methods for high-level computer vision," *Fuzzy Sets and Systems*, vol. 60, pp. 245-258, 1993.

[YOT87] R.R. Yager, S. Ovchinnikov, R. M. Tong, and H. T. Ngyen, "*Fuzzy Sets and Applications,*" New York, John Wiley & Sins, 1987.

[Zad69] Lotfi Zadeh, "Toward a Theory of Fuzzy Systems," *Technical Report*, Electronics Research Laboratory, The University of California, Berkeley, California, ERL-69-2, 1969.

[Zim91] H. J. Zimmermann, "*Fuzzy Set Theory and Its Applications,*" Massachusetts, Kluwer Academic Publishers, 1991.