July 25, 2004

Reduction of Fuzzy Systems through Open Product Analysis of Genetic Algorithm-Generated Fuzzy Rule Sets

Sanza Kazadi
Jerry Mendel, *University of Southern California*
Diana Jue-Rajasingh, *Jisan Research Institute*
Dharshan Chandramohan, *Jisan Research Institute*
David Choi, *Jisan Research Institute*, et al.

Reduction of Fuzzy Systems through Open Product Analysis of Genetic Algorithm-Generated Fuzzy Rule Sets

T. Shin
Integrated Media Systems Center
University of Southern California
3740 McClintock Ave. Suite 131
Los Angeles, CA 90089

D. Jue
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

D. Chandramohan
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

D. Choi
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

C. Seng
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

J. Yang
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

A. Bae
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

A. Lee
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

J. Lee
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

P. Lim
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

S. Kazadi
Jisan Research Insitute
28 N. Oak Avenue
Pasadena, CA 91107

J. Mendel
Integrated Media Systems Center
University of Southern California
3740 McClintock Ave. Suite 131
Los Angeles, CA 90089

***Abstract*— We explore the reduction of a fuzzy classifier designed to perform a binary classification of tracked or wheeled vehicles based on acoustic data. A genetic algorithm is used to explore the design space of the classifier, with variations performed on the number of antecedents included in the final fuzzy system. Besides the original individual set generated by the GA, we define a subset of it with a small number of antecedents as a filtered set. A novel method of extracting important system components, known as *open product analysis*, is applied to these two sets, yielding systems that perform well with a small number of antecedents. The fuzzy classifier we reduced performs well using only 10 to 30% of the antecedents that were originally used for classification.**

keywords: fuzzy classifier, open product analysis

I. Introduction

Fuzzy Rule-Based Classifier Systems (FRBCS) have been studied for quite some time. Literally hundreds of papers have been written on the topic, and these papers have generated a number of interesting ways to approach the design of the fuzzy systems. The central problem in generating fuzzy systems is the development of the fuzzy sets and membership functions of the fuzzy sets. In general, these have been developed primarily by using large amounts of human expertise in the design of the systems. Of course, when human beings are putting together fuzzy systems, it is sometimes unclear as to whether or not the system is minimally designed, as the tendency for human designers is to focus on the functionality rather than parsimony. Yet it is advantageous to create devices that are both functional and simple. As a result, it is important to consider whether or not a particular device, fuzzy or not, has been built in a way that is both simple and robust.

It is not clear, however, that there is a particular method of completing this type of reduction when it comes to the design of fuzzy systems. The authors of this work are unaware of any systematic methodology designed to reduce existing fuzzy systems in such a way that the new system is both functionally robust and significantly simpler than the system from which it was designed. Of fundamental importance are decisions relating to which antecedents are used within the rules. How might we go from a working fuzzy system to a simpler one? How do we reduce the number of antecedents impacting the rule? Generally, these questions are answered primarily by experience.

In the realm of evolutionary computation, these questions reduce essentially to a somewhat simpler question. If we wish to consider the original design as being defined by the membership functions and the fuzzy rules, the system design consists of optimizing the rule usage, the rules themselves, and the membership functions of the system. However, if the question is "simply" to decide whether or not a membership function or some of a rule's antecedents are needed for the device in question, then the question reduces further. If the inclusion of an antecedent is represented by a '1' in a binary list, then the question becomes: Which of the binary lists has a small number of '1's and still maintains good functionality? In general, there is no *a priori* method for answering this question. However, we can begin to answer the question by employing evolutionary and analytic methods.

We apply a genetic algorithm to fuzzy systems as a precursor to open product analysis in this paper. As one might guess from the good fit between fuzzy system design and optimization and the application of a genetic algorithm, such an application is not particularly new [1][2][3][4]. However, rather than designing the fuzzy system's rules and membership functions, we only use the GA to simplify an existing fuzzy system by gleaning details about the search space from multiple applications of the genetic algorithm. We analyze the output of the GA with open product analysis, a method that further reduces a fuzzy system by finding the necessary antecedents for optimal functionality of the system.

In this paper, we explore a method of reduction of a fuzzy system by applying it to a fuzzy classifier that classifies acoustic data as having been produced by either a tracked or wheeled vehicle. In Section 2, we describe how a simple genetic algorithm is applied to exploring the search space of possible derivative fuzzy systems. Section 3 examines the output of the genetic algorithm described on Section 2 and the critieria for selectively choosing GA outputs to create the "filtered set." Section 4 illustrates how conditional probability matrices can be built from the data so as to extract grouping information for system design. Finally, Section 5 offers a discussion and our concluding remarks.

II. Genetic Algorithm-Based Feature Reduction

A. Basic fuzzy rule-based classifier

To explore the minimization of a fuzzy logic system we used a type-1 fuzzy binary classifier designed by Wu and Mendel [5], which classifies vehicles as tracked or wheeled based on acoustic data. Of the 89 runs collected, 61 were from tracked vehicles and 28 were from wheeled vehicles. The data collected from these runs was segmented into one-second blocks, and each block was processed into feature vectors of 11 dimensions. One set of 11 antecedents corresponds to one fuzzy rule, and a total of 9 rules are used inside the FRBCS. However, there was no justifiable reason for choosing precisely 11 antecedents for each fuzzy rule and precisely 9 fuzzy rules for the system.

B. Genetic Algorithm

To select features for the classifier system, we utilized the search space that a single point mutation, single point crossover, and probability-based reproduction genetic algorithm (GA) provides. The GA is a method that optimizes a population of randomly initialized binary features, each of which encodes the set of fuzzy antecedents used in the classifier (1's represents antecedents that are turned "on" and are used in the classifier; 0's represent antecedents that are turned "off" and are not used in the classifier). As a measurement of how well the classifier would work with the inclusion of particular features, the fitness was formulated to include the accuracy of classification by the classifier and the number of features used to achieve this accuracy. Mutation, crossover, and initial population characteristics, which include population size and generation number, are optimized to produce more individuals with better fitness over a shorter amount of time.

1) Fitness Function: The fitness function evaluates a possible solution candidate for our set of objectives. Our objectives include the following: maintenance of high classification accuracy and performance and a reduction in the number of included features to achieve this performance. The fitness must increase as the performance increases and the number of antecedents decreases.

$$f(l) = \sum_{k=1,k\neq l}^{89} [\alpha \sum_{i=1}^{N_l} C_{k,i} - P_l] \tag{1}$$

The equation (1) represents the general form of our fitness function that can be applied to the lth design of the FRBCS. $C_{k,i}$ represents the accuracy of the classifier that uses raw acoustic data that the 1's in the individuals of the GA hold the place for. $C_{k,i}$ takes the value of '1' if the ith block of data from the kth run is classified correctly and '0' if the block of data is classified incorrectly; the value can be seen as classifier accuracy. The total number of these binary values, N_l, depends on the type of vehicle from which the block of data comes from. N_l is 80 for l less than or equal to 61, which corresponds to tracked vehicles; N_l is 160 for the remaining vehicles, which are wheeled. The second summation within the first summation represents the number of correctly classified blocks from the kth run of data. P_l represents the number of antecedents used in the lth design of the FRBCS. The constant α can be considered a weighting factor put on the correctness of the classifier, and it is set to 10,000, a value large enough to preserve the accuracy of the classifier at any cost while attempting to reduce the number of antecedents required. Even if the number of required antecedents decreases dramatically, the fitness cannot increase if only one more block is misclassified.

2) Selection Criteria: In addition to the first selection metric, fitness, we use the false alarm rate (FAR) of one left-out run of data as the second criteria. The FAR represents the wrongness of the classifier defined by the current individual during a generation of the GA. In each generation, the FAR is calculated for only the individual that has the highest fitness. When the GA continues to the next generation, this individual's fitness is compared to the fittest individual of the previous generations. If the new generation's individual has a higher fitness than the previous generations' individual, then the FAR for the new one is calculated and compared to the FAR of the previous one. If the latter is lower than the former the new fittest individual is set to the best one. Eventually, at the end of the GA the solution with both the highest fitness and lowest FAR is considered the best individual of the genetic algorithm and is output.

III. Filtering GA-Generated Individuals

Each time the GA is applied once, the best individual is produced according to the two metrics mentioned in the previous sections. Because of the randomness inherent in

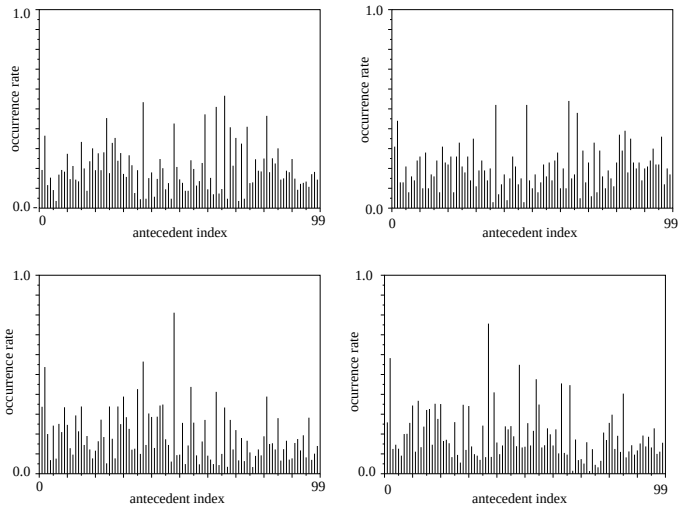

Fig. 1. Occurrence rates of antecedents: (a) design 30, (b) design 60, (c) design 80, (d) design 89.

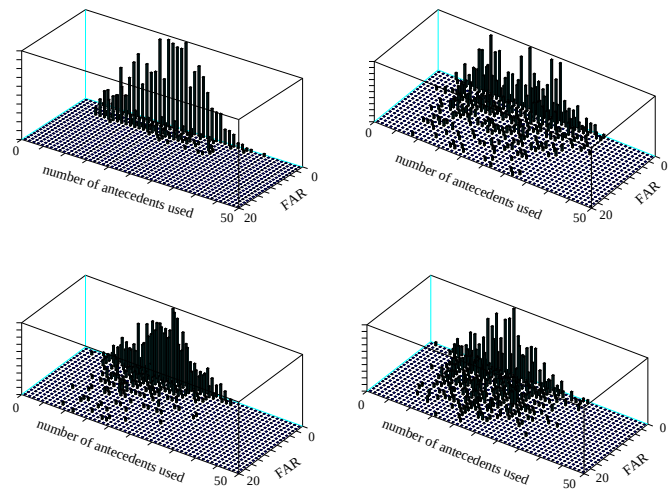

Fig. 2. Distribution in number of antecedents: (a) design 30, (b) design 60, (c) design 80, (d) design 89.

the GA, such as random initialization and several probability parameters, a large set of GA outputs is needed for accurate statistics. In this study, we collect 1,000 output individuals from the GA, which is an amount of data large enough for statistical analysis.

A. Investigation of GA-generated Individuals

The easiest way to analyze the output individuals produced by the GA is by examining the occurrence rate of each antecedent out of the total 99 antecedents. By simply summing up the number of 1's in each antecedent, we can know the frequency of a certain antecedent being used. As stated previously, a total of 89 different designs is available. For brevity, the occurrence rate of only four designs out of 89 are represented in Fig. 1.

Note that the overall occurrence rate is low. Most antecedents are turned on with the probability much lower than 0.5, which means that only a few antecedents survive in the final best individual after a GA evolution. Nevertheless, it is not easy to specifically select a small number of antecedents out of 99 since the probability difference between most antecedents is not statistically significant. Thus it is highly probable that there is a grouping property or correlation between different antecedents, which will be discussed in the following section.

Together with the fitness, the FAR of one left-out run of data is the second selection metric in the GA. The 3-D histograms that consider both the number of antecedents used and FAR are shown in Fig. 2. Of course, from the GA evolution most data takes on a very low FAR for all four designs. We can see that the number of antecedents used varies over a large range while the maximum number is still less than half of the total of 99, which means that many different solution individuals are produced by the GA. The solution space obtained from the GA is very complicated, i.e. multimodal rather than unimodal, which is again the reason why it's difficult to specify the best solution uniquely.

B. Filtering output individuals

Of course all output data from the GA can be possible solutions since it comes from the GA evolution based on the specified objective function. However, we need to filter this original GA output further, assuming that a considerable part of it is stuck in a local optima, which is in fact one drawback of many optimization tools that include GAs. Among all 99 antecedents within the nine fuzzy rules there must be core antecedents that play the most important role in classification. If one of a few of these core antecedents is not used in calculating the FAR of a GA output individual, then many other antecedents would be used for compensation, resulting in the use of many unnecessary antecedents. As can be seen in Fig. 2, the output individuals that have a relatively high number of antecedents are obtained from the GA. In order to extract optimal solutions more efficiently, we need to selectively collect output individuals that have fewer turned-on antecedents than a threshold set at 20. For distinction from the original set of individuals, we term these selected individuals the "filtered set."

IV. Feature Selection via Open Product Analysis

As we have seen from histograms of the raw output individuals of the GA, the probabilities of antecedent occurrence vary very little, and consequently we learn little about the interplay between the parts of the system. This is because while the GA eliminates system components, it can only evaluate the system as a whole. Hence, redundancies built in to the system or cross talk between the components of the system (particularly the rules) remains unaccounted for. However, since the effects of one rule on the functionality of others is neglected in the process of designing rule sets, redundancies are often built into the system; it becomes significantly difficult to eliminate

all unnecessary components of the system using only a GA. It would seem that after the GA has randomly weeded out individual antecedents that demonstrate detrimental effects on system performance, that the structure of the system ought to be examined to determine the effects of interplay between system components. To accomplish this we use a statistical methodology called Open Product Analysis, which uses a conditional probability matrix to determine the relationship between different rules and reconstruct the system to take advantage of these relationships.

A. Conditional probability matrix

Suppose that we have N binary vectors of dimension M that are either original individuals from the GA or filtered ones. Among these N vectors the lth one can be represented by $\mathbf{n}(l) = (n_1(l), \ldots, n_m(l))$. Then, suppose that we take

$$A_{\mathbf{n}(l)} = (\mathbf{n}(l))^T (\mathbf{n}(l)) = \begin{pmatrix} n_1(l)\, n_1(l) & \cdots & n_1(l)\, n_m(l) \\ \vdots & \ddots & \vdots \\ n_m(l)\, n_1(l) & \cdots & n_m(l)\, n_m(l) \end{pmatrix}. \quad (2)$$

Suppose that we now sum all M of these matrices formed from the M vectors and normalize them. Then we have

$$B \equiv \frac{1}{N} \sum_{l=1}^{n} A_{\mathbf{n}(l)}$$

Since each $n_i \in \{0, 1\}$, this reduces to

$$B = \begin{pmatrix} P(n_1) & P(n_1, n_2) & \cdots & P(n_1, n_m) \\ P(n_2, n_1) & P(n_2) & \cdots & P(n_2, n_m) \\ \vdots & \vdots & \ddots & \vdots \\ P(n_m, n_1) & P(n_m, n_2) & \cdots & P(n_m) \end{pmatrix}. \quad (3)$$

Note that $P(n_i, n_j) = P(n_i)\, P(n_j | n_i)$. Then we can define the matrix C as follows:

$$C_{ij} \equiv \frac{B_{ij}}{B_{ii}} = \begin{pmatrix} 1 & P(n_2|n_1) & \cdots & P(n_m|n_1) \\ P(n_1|n_2) & 1 & \cdots & P(n_m|n_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(n_1|n_m) & P(n_2|n_m) & \cdots & 1 \end{pmatrix}. \quad (4)$$

Finally, this matrix provides the conditional probabilities between any two binary random variables associate with antecedents in this study.

The conditional probability matrices for two kinds of individual sets, original and filtered, are illustrated as 3-D bar plots in Fig. 3 for design 30. Since the shape of original matrices is too complicated to be represented, only a submatrix corresponding to 50 antecedents is illustrated. Apparently there exist several ridges and valleys over certain rows, which means that many other antecedents highly tend to also be turned on or turned off when a specific one is already turned on. On the whole, the variation of conditional probabilities in the filtered set is larger than that of the original set, which means that the latter tell us the connectedness between antecedents more clearly.

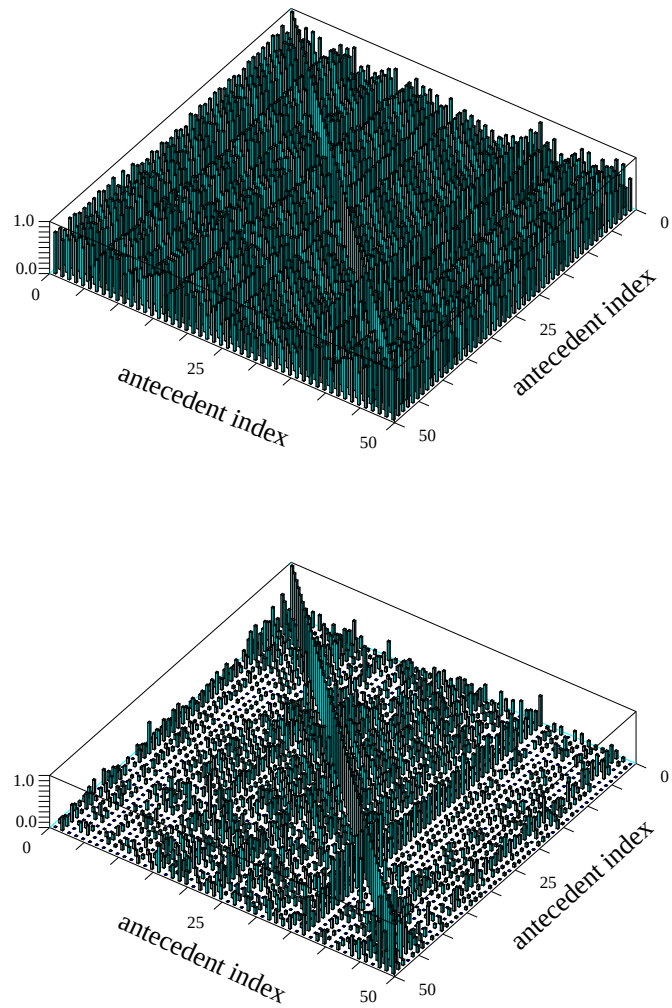

Fig. 3. Conditional probability distribution: (a) original set (design 30), (b) filtered set (design 30).

B. Extraction of new designs

As mentioned previously the unique optimal solution with reduced complexity cannot be obtained from the simple use of the GA. The corresponding solution space turned out to be multimodal; hence there can exist many solutions even for fixed complexity or number of antecedents used. As a possible example of utilizing the conditional probability matrix derived in the previous subsection, we propose one way of deriving new designs under various conditions on complexity. By assigning a certain threshold to conditional probabilities, we can reasonably adjust the number of antecedents in a solution individual. The specific procedure for building this is as follows:

1) Choose a threshold value $t \in [0, 1]$. Set k equal to the index of the antecedent of highest frequency. Create a vector $\overrightarrow{v}$ of m numbers, all set to -1.
2) Set the kth element of the vector equal to 1.
3) Set all elements j of the vector equal to 1 if $C_{kj} > t$.
4) Change remaining elements of vector $\overrightarrow{v}$ from -1 to 0.

This builds up a binary vector $\overrightarrow{v_t}$ indicating the antecedents included in the solution. It is clear that if $t' < t$, then $v^i_{t'} = 1$ if $v^i_t = 1$.

Once a threshold value is determined, a single solution for each set of output individuals is obtained using the above scheme. Of course this solution is different from those present in the original and the filtered sets of individuals obtained from the simple use of the GA. Using this solution, both the average FAR over all 89 runs of data and the single FAR of one left-out data can be computed. The relation between FAR and total number of antecedents associated with many threshold values is illustrated in Fig. 4. In each design, two plots in the first column are obtained from the original individual set of 99 antecedents, and the plots in the second column are from the filtered data set with the antecedent threshold set at 20 antecedents. In the first row of figures for each design, the FAR is calculated over the single run of data left out for validation. In the second row of figures for each design, the FAR is averaged over all of 89 runs of data.

As expected, the FAR decreases, though not monotonically, as the number of antecedents increases. We can see that on the whole the classifier obtained from the filtered set of individuals outperforms the classifier from the original set for lower numbers of antecedents. These results indicate that the filtered data set reflects information on core antecedents out of the total 99 antecedents more accurately while the original set is influenced, though not much, by noisy individuals, i.e. ones with many unneccesary antecedents at the cost of core antecedents. The solution created from the filtered data set is able to classify one left-out run of data perfectly with about 10 to 30 antecedents, which is about 10%- 30% of the original number of antecedents. Also, as seen in design 80's solution the average FAR reaches a low around 20 antecedents. However, the FAR increases as more antecedents are included in the calculation because of adverse interplay between antecedents, creating a dip in the graph. The antecedents used at the low point of the dip form the core antecedents that are essential to classification.

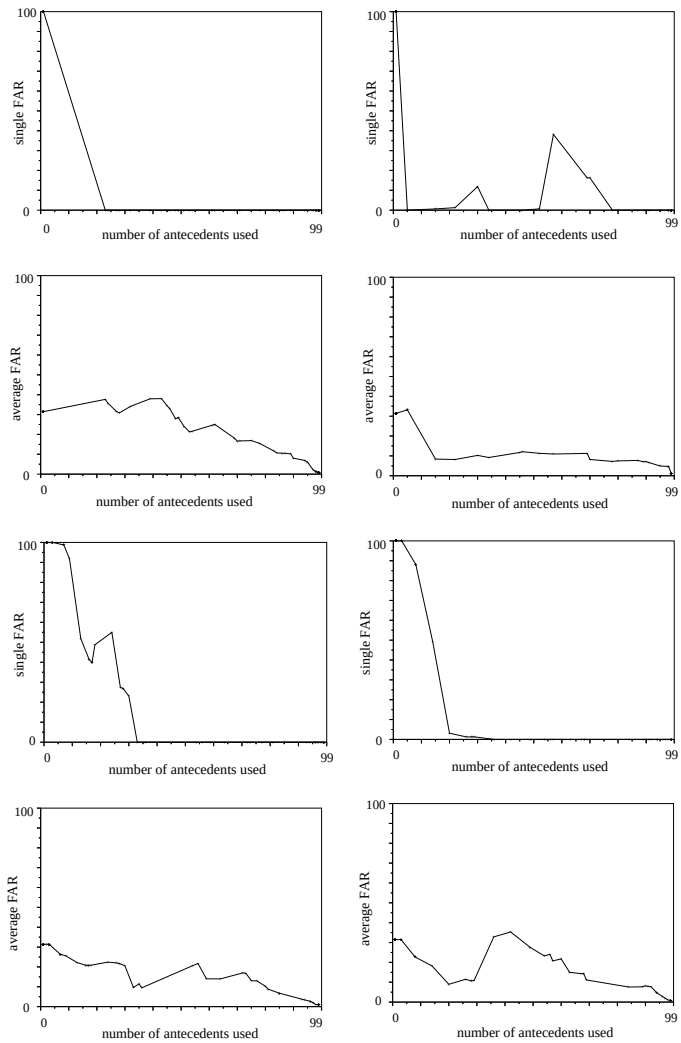

Fig. 4. Relationship between number of antecedents and FAR: (a) design 30, (b) design 80.

V. Concluding Remarks

The naive expectation of the fuzzy engineer is that one can use a stochastic optimization technique such as the genetic algorithm to both create a fuzzy system and to reduce the fuzzy system once it is created. Indeed, this seems to be a rather simple task given the appropriately designed optimization technique. However, we've seen in this paper that such a blind use of the genetic algorithm cannot be generally effective for a number of important reasons. These reasons include the structure of the search space, which we have seen has nontrivial structure which tends to strand genetic algorithm populations in suboptimal locations. This structure, heretofore unknown, has significant impact on the effect of any algorithm which seeks to find more parsimonious implementations of the fuzzy system.

As a result, we've created a new methodology for both analyzing the search space and developing simpler fuzzy systems that are capable of carrying out the same calculations as the original system. This method uses the varied final solutions developed from the genetic algorithm to extract enough information about the search space that parsimonious design is possible. It is remarkable that such a large part of the original fuzzy system can be left out of the design with perfect performance on the single run and excellent performance on all runs. Such a revelation indicates that fuzzy systems may be extremely inflated in their design, indicating the potential for successful application of similar analyses on other fuzzy systems.

Still, this is a first step, and much work needs to be

done yet before this methodology can be applied to fuzzy systems in general. It is not yet clear whether or not the individual solutions which have come out of the open product analysis can be joined by other individual solutions starting with other antecedents. Moreover, if there are other solutions, how much do these overlap with the existing solutions? Is it advantageous to use multiple design / reduce / design / reduce cycles? Finally, is it possible to ground this work in an even more solid theoretical background, and how might such a theoretical approach help us to apply this technique to more general problems? These questions must be answered before a satisfactory understanding of this method and its place in fuzzy systems can be established.

ACKNOWLEDGEMENTS

T. Shin, J. Lee, and P. Lim were supported by the Integrated Media Systems Center at the University of Southern California under NSF Grant # EEC-9529152.

REFERENCES

[1] O. Cordon, F. Gomide, F. Herrera, F. Hoffman, and L. Magdalena, *Ten years of genetic fuzzy systems : current framework and new trends*, **Fuzzy Sets and Systems**, 141, 5˜31, 2003.

[2] T.-P. Hong and J.-B. Chen, *Finding relevant attributes and membership functions*, **Fuzzy Sets and Systems**, 103, 389˜404, 1999.

[3] A. Gonzalez and R. Perez, *Selection of relevant features in a fuzzy genetic learning algorithm*, **IEEE Trans. System Man Cybernet. B**. 31 (3), 417˜425, 2001.

[4] H. -M. Lee, C.-M. Chen, J.-M. Chen, and Y.-L. Jou, *An efficient fuzzy classifier with feature selection based on fuzzy entropy*, **IEEE Trans. Systems Man Cybernet. B**. 31 (3), 426˜432, 2001.

[5] H. Wu and J. Mendel, *Binary classification of ground vehicles based on the acoustic data using fuzzy logic-rule based classifier*, USC-SIPI report, 2002.