Classification of Battlefield Ground Vehicles Using Acoustic Features and Fuzzy Logic Rule-Based Classifiers

Hongwei Wu, *Member, IEEE*, and Jerry M. Mendel, *Life Fellow, IEEE*

***Abstract*—In this paper, we demonstrate, through the multicategory classification of battlefield ground vehicles using acoustic features, how it is straightforward to directly exploit the information inherent in a problem to determine the number of rules, and subsequently the architecture, of fuzzy logic rule-based classifiers (FLRBC). We propose three FLRBC architectures, one non-hierarchical and two hierarchical (HFLRBC), conduct experiments to evaluate the performances of these architectures, and compare them to a Bayesian classifier. Our experimental results show that: 1) for each classifier the performance in the adaptive mode that uses simple majority voting is much better than in the non-adaptive mode; 2) all FLRBCs perform substantially better than the Bayesian classifier; 3) interval type-2 (T2) FLRBCs perform better than their competing type-1 (T1) FLRBCs, although sometimes not by much; 4) the interval T2 nonhierarchical and HFLRBC-series architectures perform the best; and 5) all FLRBCs achieve higher than the acceptable 80% classification accuracy.**

***Index Terms*—Acoustic signal, Bayesian classification, fuzzy logic rule-based classification, ground vehicles, interval type-2 fuzzy logic rule-based system.**

I. Introduction

MOST distance-based classification problems are solved by using one or more of the following approaches [6].

- Assume that class-conditional probability density functions are of a certain parametric form (e.g., Gaussian mixture model or hidden Markov model); then estimate the parameters by using the maximum-likelihood or Bayesian methods; and, finally, use the resulting distributions for classification.
- Assume that the discriminant functions are of a certain parametric form (e.g., linear discriminant function, support vector machine), and then adjust the parameters to optimize some criterion function.
- Use mode-free methods, e.g., k-nearest-neighbor, multilayer neural networks, or fuzzy logic rule-based classifiers (FLRBC).

According to the *No Free Lunch Theorem* [32]: "Although some of these approaches may perform better than others for certain circumstances, there is no one approach that is universally superior to the others." Whether one approach can perform well for a specific classification problem is heavily related to questionable choices that the designer of the classifier makes based on his insights into that problem [6], [8]. There are some questions one has to face at the very beginning of a classifier design, e.g., 1) the number of components included in the Gaussian mixture model, 2) the dimensionality of the hidden states of the hidden Markovian model, 3) the kernel function and parameters chosen for the support vector machine, 4) the number of neighbors chosen for the k-nearest neighbor method, 5) the numbers of hidden layers and neurons included in, the inter-connectivity among neurons, and the activation functions chosen for the neural network model, and 6) the number of rules included in, the membership functions (MF), and the fuzzy set models (either type-1 or interval type-2) chosen for the FLRBC.

The designer may try to answer these questions based on his knowledge about the classification problem, or by treating them as machine learning problems at a higher level and then using machine learning techniques, or, by using stochastic methods (e.g., a genetic algorithm) [6], [8]. No matter which approach is utilized, *a priori* knowledge (i.e., the designer's information about the classification problem or his experience with different classifier design techniques) is exploited in one way or another. It is difficult, and generally an open problem, to make direct connections between such knowledge and the concrete configurations of classifiers. For example, when designing multilayered neural network classifiers, knowing the dimensionality of the features and the number of categories does not help much in determining the number of hidden layers, or the number of neurons in each layer (except for the first and the last layers), or the interconnectivity among the neurons.

Interestingly enough, for those classification problems where the a priori knowledge can be represented in the form of expert opinions, a FLRBC is able to incorporate such knowledge; hence, it may be able to outperform other classifiers. Because a FLRBC provides a framework to incorporate both subjective and objective information, and to process both linguistic and numeric information, it has already been been applied to many classification problems where both expert opinion (subjective knowledge) and design samples (from which objective knowledge can be extracted) are available, or where the features are represented as words (e.g., the degree of pain is *severe*) [1]–[3], [9], [10], [14], [15], [21].

In this paper, we demonstrate, through the application of classification of ground vehicles from acoustic features, how it is straightforward to directly exploit the information inherent

Manuscript received January 27, 2005; revised December 18, 2005, March 20, 2006, and June 16, 2006. This work was supported by the Department of Army Research Office under Grant DAAD19-01-1-0666. The content of the information does not necessarily reflect the position or policy of the federal government, and no official endorsement should be inferred.

H. Wu is with the Department of Biochemistry and Molecular Biology, the University of Georgia, Athens, GA 30601 USA (e-mail: hongweiw@csbl.bmb.uga.edu).

J. M. Mendel is with the Department of Electrical Engineering, the University of Southern California, Los Angeles, CA 90089-2564 USA (e-mail: mendel@sipi.usc.edu).

Digital Object Identifier 10.1109/TFUZZ.2006.889760

about this problem to determine the number of rules, and subsequently the architecture, of such a FLRBC.

The rest of this paper is organized as follows. We first describe this classification problem and data pre-processing in Sections II and III, respectively; then provide a brief review of theories of fuzzy logic rule-based systems (FLRBS) in Section IV before we propose classifier designs in Section V and present experimental results in Section VI, and finally draw conclusions in Section VII.

II. Problem Description

The acoustic emissions of ground vehicles contain a wealth of information which can be used for vehicle classification under certain circumstances (e.g., in the battlefield). The model for the acoustic emissions can be simplified to be the addition of periodic components and noise. The former accounts for the periodic movements in the engine, and the latter accounts for the propulsion process in the engine and the interactions between the vehicle and the roads [26]. Because the operation mechanisms are different for different vehicles, it is believed to be possible to distinguish among different vehicles based only on their acoustic emissions.

There has been some previous research devoted to multicategory classifications of ground vehicles based on acoustic signals [5], [20], [24], [30]. Choe *et al.* [5] first performed the discrete wavelet transform of the acoustic data to generate multiple resolution-level spectrograms, then used the statistical parameters and the energy content of the wavelet coefficients in each spectrogram as the features, and finally compared these features against the reference vehicle features in the database to determine which class the input acoustic signal belongs to. Liu [20] adopted the cochlear filter and A1-cortical wavelet transform of the acoustic signal to obtain multiple resolution-level spectrograms in the auditory frequency domain, and used these representations and vector-quantization based clustering algorithms to classify vehicles. Sampan [24] used 30 features describing the energy envelope of the acoustic signal in the time domain, and a multi-layer perceptron network and a FLRBS to classify four different categories of vehicles. Wellman [30] investigated three feature extraction methods—simple power spectrum estimates, harmonic line association (HLA), and principal component analysis (PCA)—and used an artificial neural network to classify vehicles.

In our problem, there are *nine* different *kinds* of vehicles that have to be classified into one of *four categories*—heavy-tracked, heavy-wheeled, light-tracked and light-wheeled—based on their acoustic emissions collected on a so-called *normal terrain*. Table II summarizes the category label and the number of runs for each kind of vehicle. Although the aforementioned research [5], [20], [24], [30] has shown that the classification of ground vehicles based on their acoustic emissions is attractive, it is still very challenging to accomplish this, because the features that are extracted from the acoustic measurements are time-varying and contain a lot of uncertainties (see Section III-D). This is due to variations of the environmental conditions (e.g., terrain and wind), vehicle-traveling speed, and signal-to-noise ratio (SNR) of the acoustic measurements that is subject to the variability of the distance between the vehicle and the sensor system. Consequently, any classifier that makes use of acoustic measurements of ground vehicles for classification must account for these time-variations and uncertainties [33].

TABLE I
Acronyms and their full descriptions

Acronym	Full Description
FLRBC	Fuzzy Logic Rule-Based Classifier
HFLRBC	Hierarchical FLRBC
HFLRBC-P	HFLRBC in Parallel
HFLRBC-S	HFLRBC in Series
T1	Type-1
T2	Type-2
MF	Membership Function
FS	Fuzzy Set
HLA	Harmonic Line Association
PCA	Principal Component Analysis
ACIDS	Acoustic Classification/Identification Data Set
CPA	Closest Point of Approach
SD	Standard Deviation

Even before we provide explicit details about the features we used as well as the FLRBCs, it is worthwhile for us to explain that our approach is different than the aforementioned research in the following ways.

- Our features were extracted from *short time intervals* (about one second) within which the acoustic measurements can be assumed to be stationary. We believe that to use long time intervals is too challenging because the nature of the unstationarities over such intervals are unknown. Additionally, using short time interval features for classification allows prompt decisions, which is desirable in a battlefield scenario.
- Because our features can vary from one short time interval to another, we developed both type-1 (T1) and interval type-2 (T2) FLRBCs. The T1 FLRBCs do not have a direct mechanism to model the unknown varieties of the features, but the interval T2 FLRBCs do; hence, the latter have the *potential* to outperform the former [21].
- We used one fuzzy logic rule for each kind of vehicle (for a total of nine rules) to describe the relationship between the features and vehicle category labels, then used fuzzy sets in each fuzzy logic rule to model the variations of features caused by variable operating conditions (e.g., traveling speed, wind, temperature and humidity), and finally optimized the parameters of MFs by using training samples. In this way, our a priori knowledge that there are nine possible vehicles helped to establish the architecture of each FLRBC.
- We designed three multicategory classifiers, one nonhierarchical and two hierarchical, all of which were greatly influenced by our previous work into binary classification of battlefield ground vehicles [34]–[36] (i.e., tracked vs. wheeled vehicles, heavy-tracked versus light-tracked vehicles, and heavy-wheeled vs. light-wheeled vehicles).

In the next few sections, we explain these differences in more detail. Since we use numerous acronyms throughout this paper, we first list their full descriptions in Table I.

III. Preprocessing

Preprocessing, including data analysis, prototype generation, feature extraction and uncertainty analysis, was used to establish

TABLE II
THE NUMBER OF RUNS FOR HEAVY-TRACKED, LIGHT-TRACKED, HEAVY-WHEELED AND LIGHT-WHEELED VEHICLES

Class	Vehicle	Number of Runs	Total Number
Heavy-tracked	HT-a (Vehicle 1)	15	46
	HT-b (Vehicle 2)	8	
	HT-c (Vehicle 8)	15	
	HT-d (Vehicle 9)	8	
Light-tracked	LT-a (Vehicle 4)	15	15
Heavy-wheeled	HW-a (Vehicle 3)	8	16
	HW-b (Vehicle 5)	8	
Light-wheeled	LW-a (Vehicle 6)	8	12
	LW-b (Vehicle 7)	4	
TOTAL		89	

appropriate models for the features that were extracted from the acoustic measurements. Because many readers of this journal may not be so familiar with the specifies for this application, we briefly provide some of them next.

A. Data Collection

We used the acoustic-seismic classification/identification data set (ACIDS) that consists of acoustic data collected for nine kinds of ground vehicles on the normal terrain. The number of runs for each kind of vehicle is summarized in Table II.

The acoustic data were collected by using a sensor system that is a three-element equilateral triangular microphone array with a length of 38.1 cm between microphones. As shown in Fig. 1, a *run* corresponds to a ground vehicle traveling at a constant speed toward the sensor system, passing the *closest point of approach* (CPA), and then moving away from the sensor system. Although the traveling speed of a ground vehicle is approximately constant within each run, it varies from run to run, ranging from 5 to 40 km/hr. The variation of the traveling speed, along with the environmental variations (e.g., wind and terrain), makes the acoustic emissions of the same vehicle different from run to run. Within each run, when the vehicle is far away from the sensor system (in the beginning and ending parts of a run), the acoustic measurements mainly consist of background noise, whereas when the vehicle is closer to the sensor system (in the middle part of a run), the acoustic measurements consist of acoustic emissions of the ground vehicle as well as the background noise. The variation of the distance between the traveling vehicle and the sensor system makes the SNR of the measurements variable within each run. The above two sources of variations are both embodied in the uncertainties of the features that are extracted from the acoustic measurements.

B. Prototype Generation

A complete run contains anywhere from 56 to 420 seconds worth of measurements, where the different lengths of runs are due to the different conditions under which the data were collected. At the sampling rate of 1025.641 Hz, there are, therefore, a great number of measurements in each run. As already mentioned, these measurements are nonstationary because their SNR varies within each run. These two factors make it impractical to process all measurements of a run simultaneously; hence, we segmented them into one-second blocks, and treated one block (rather than a whole run) as one *prototype*.

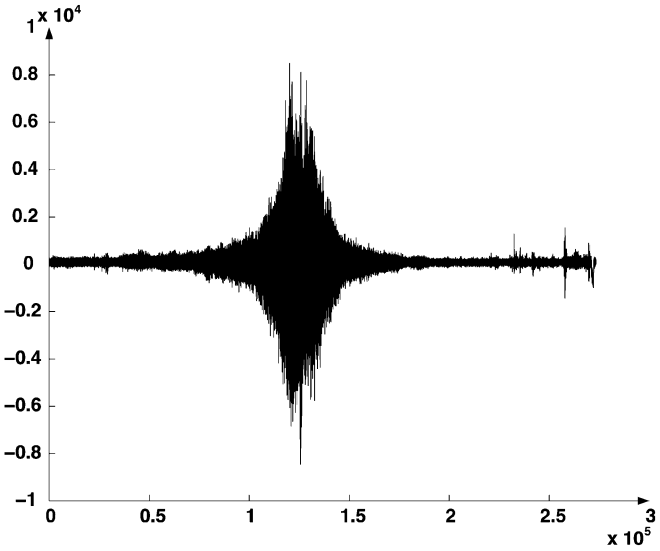


Fig. 1. Acoustic measurements of a typical run obtained from one sensor.

Since the middle part of each run has higher SNR and contains more information about the vehicle than its beginning and ending parts (see Fig. 1), we used this part of the measurements to generate prototypes, and used the following two ways to locate the measurements with high SNR.

- **CPA-based prototypes:** We first located the CPA (where the measurements have the maximum magnitude), and then slid a 1024-point rectangular window (about one second) with 50% of window overlap to the left and right of the CPA, to obtain 80 CPA-based prototypes from each run.
- **Non-CPA-based prototypes:** We first estimated the background noise level of a run as the energy of its first data block, and then compared each succeeding data block with this background noise level. If the energy of a succeeding data block was above the background noise level for a certain threshold[1], then this data block was considered to be a non-CPA-based prototype.

During our classifier evaluation experiments, we mainly used CPA-based prototypes because: 1) the number of CPA-based prototypes for each kind of ground vehicle is proportional to its number of runs, and since the number of runs available for each of the nine kinds of vehicles is different (see Table II), the knowledge of the *a priori* probability of each kind of vehicle was implicitly used during the training and testing of a classifier; and 2) CPA-based prototypes are more robust to noise than non-CPA-based prototypes, because the estimation of the energy level for each data block was sensitive to the noise. However, because CPA information may not be available during the real-time operation period, we also used non-CPA-based prototypes during our evaluation experiments to test the performance of classifiers for the more realistic battlefield scenario that decisions must be made before a vehicle reaches the CPA.

C. Feature Extraction

We used the harmonic line association (HLA) feature vector [23], [30], i.e., the magnitudes of the second through 12th har-

[1]After examining the energy distribution of all runs, the threshold was set at 8 dB.

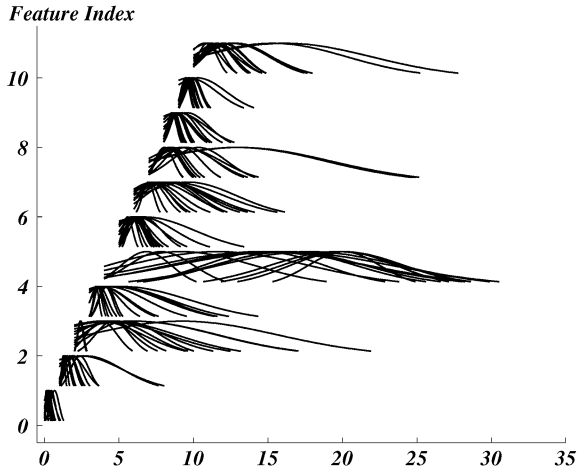


Fig. 2. Feature distributions of the light-tracked vehicle, LT-a. Each curve corresponds to a feature distribution within one of the 15 available runs, which is characterized by a Gaussian function centered at $m_{i,\mathrm{run}}$ and $\pm\sigma_{i,\mathrm{run}}$ about $m_{i,\mathrm{run}}$. For a better visualization effect, the origin for showing the distribution of x_i is shifted to the point (i, i) in the figure.

monic frequency components,[2] for vehicle classification, because it is low-dimensional, has physical meaning, outlines the magnitude spectrum envelope, and is simple to extract based on the algorithm in [23]. In the rest of this paper, the features are denoted $x_1, \ldots, x_{11}$, and their corresponding feature spaces are denoted $X_1, \ldots, X_{11}$.

The most important step in the HLA method is the estimation of the fundamental frequency f_0, because only after this occurs can the harmonic frequencies be located and their magnitudes be taken as features. The algorithm used to estimate f_0 in the HLA method was first developed in [23]. Compared with the maximum-likelihood estimation method [17] that requires f_0 to be initially known within an accuracy of 0.5 Hz, the HLA method in [23] does not rely very strongly on a mathematical model for the acoustic emissions of ground vehicles, and requires less *a priori* knowledge about f_0; hence, we chose the HLA method.

D. Uncertainties of the Features

Since all classifiers (described in Section V) were designed (trained) using CPA-based prototypes, our analysis about the uncertainties of the features also focused on such prototypes. In each individual feature space X_i, we first calculated the mean and standard deviation (SD) (denoted $m_{i,\mathrm{run}}$ and $\sigma_{i,\mathrm{run}}$, respectively) for each run to reflect the uncertainties of features *within* runs, and then calculated the mean of $m_{i,\mathrm{run}}$, SD of $m_{i,\mathrm{run}}$, mean of $\sigma_{i,\mathrm{run}}$, and SD of $\sigma_{i,\mathrm{run}}$ for each kind of vehicle to reflect the uncertainties of features *across* runs. For the light-tracked vehicle LT-a, Fig. 2 shows the feature distributions (characterized by $m_{i,\mathrm{run}}$ and $\sigma_{i,\mathrm{run}}$), and Table III summarizes the four statistics across runs. Observe from Table III that: 1) the SD of $m_{i,\mathrm{run}}$ is not negligible when compared to the mean of $m_{i,\mathrm{run}}$, 2) the SD of $\sigma_{i,\mathrm{run}}$ is also not negligible when compared to the mean of $\sigma_{i,\mathrm{run}}$, and 3) the SD of $\sigma_{i,\mathrm{run}}$ is of the same order of magnitude as the SD of $m_{i,\mathrm{run}}$.

[2]On one hand, we used the 11-dimensional features because the studies presented in [23], [30] have used the same approach and achieved fair performance. On the other hand, our study presented in this paper was focused only on the classifier designs. The interested readers can refer to [25] for a study on the minimum sufficient number of features for this application.

TABLE III
STATISTICS OF THE FEATURES (x_i) FOR THE LIGHT-TRACKED VEHICLE, LT-A

Statistics	x_1	x_2	x_3	x_4	x_5	x_6
mean of $m_{i,\mathrm{run}}$	0.2649	0.7564	2.5723	1.1011	11.4690	1.0919
mean of $\sigma_{i,\mathrm{run}}$	0.1878	0.5498	2.2688	1.7858	4.3333	1.0356
SD of $m_i$run	0.1421	0.3805	2.0130	0.5851	4.7285	0.3303
SD of $\sigma_{i,\mathrm{run}}$	0.1016	0.5557	2.1362	1.5766	1.3903	0.9048
Statistics	x_7	x_8	x_9	x_{10}	x_{11}	
mean of m_i(run)	1.7425	2.4722	0.8606	0.7360	2.2884	
mean of σ_i(run)	1.8200	2.0462	0.7220	0.5234	1.9249	
SD of m_i(run)	0.7300	1.5985	0.2953	0.2339	1.5684	
SD of σ_i(run)	0.8007	1.7276	0.4625	0.4062	1.5735	

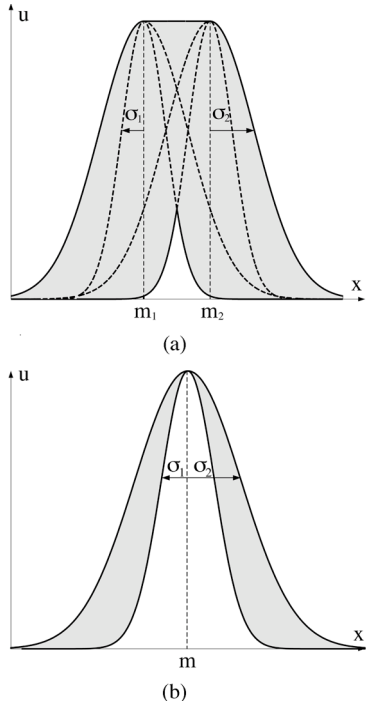


Fig. 3. Interval T2 fuzzy sets. (a) Gaussian primary MF with an uncertain mean and an uncertain SD. (b) Gaussian primary MF with an uncertain SD.

These results demonstrate that feature uncertainties exist both within and across runs. They also provide a basis for choosing appropriate fuzzy set models for the features, i.e., the ideal models must be able to account for the *simultaneous variations in the* $m_{i,\mathrm{run}}$ *and* $\sigma_{i,\mathrm{run}}$ of the features. T1 FLRBCs cannot directly model such simultaneous variations, because the T1 MFs only provide a point value at each feature measurement. Interval T2 FLRBCs, on the other hand, can model such simultaneous variations, because the interval T2 MFs provide a range of values at each feature measurement. Based on our feature uncertainty analysis, we chose to model the antecedent that is associated with each feature as an interval T2 fuzzy set whose MF is a Gaussian function that has both an uncertain mean and an uncertain SD, as shown in Fig. 3(a). We also chose to model each input as an interval T2 fuzzy set whose MF is a Gaussian function that is centered at the measured feature but has an uncertain SD, as shown in Fig. 3(b), so as to account for feature extraction uncertainties.

IV. Preliminary Knowledge of Fuzzy Logic Rule-Based Systems

Because we establish nonhierarchical and hierarchical FLRBCs (in Section V) all based on FLRBSs, it is appropriate to briefly review the operations and optimization procedure of such a system.

A FLRBS consists of four components: rule-base, fuzzifier, fuzzy inference engine and output processor [21]. The *rule-base* contains M rules that are extracted from expert knowledge, mathematical models, or data. Each rule describes a relation from the domain $X_1 \times \cdots \times X_p \subseteq R^p$ to the range $Y \in R$, and can be expressed as the following IF–THEN statement:

R^j: IF x_1 is F_1^j and $\cdots$ and x_p is F_p^j, THEN y is $G^j, j = 1, \ldots, M$

in which R^j represents the jth rule, F_k^j is the antecedent associated with the kth input variable x_k $(k = 1, \ldots, p)$, and G^j is the consequent associated with the output variable y. Given a vector of measurements, $\mathbf{x}' \equiv [x_1', \ldots, x_p']^T$, the *fuzzifier* converts them into fuzzy sets $A_{x_1'}, \ldots, A_{x_p'}$, one for each dimension. The *fuzzy inference engine* then computes the *firing degree* for each rule which describes how much the input fuzzy sets, $A_{x_1'}, \ldots, A_{x_p'}$, match the antecedents, $F_1^j, \ldots, F_p^j$. The *output processor* computes a crisp output, $y(\mathbf{x}')$, by using the firing degrees and the consequents or all rules.

In summary, through fuzzification, inference and output processing, the FLRBS maps the input $\mathbf{x}'$ to an output $y(\mathbf{x}')$ according to the rules. The literature about FLRBSs is very extensive, and we refer the readers to, e.g., [14], [21], and [29]. Because the input fuzzy sets, antecedents and consequents can be either T1 or interval T2 fuzzy sets, the specific computations of fuzzification, inference and output processing are different for T1 and interval T2 FLRBSs. We summarize these computations next.

A. Type-1 FLRBS

In a T1 FLRBS, the input fuzzy sets, antecedents, and consequents are all T1 fuzzy sets, and output processing only consists of defuzzification. For consistency with our later descriptions of FLRBCs, we illustrate the operations of and optimization procedure for the T1 FLRBS based on the following assumptions.

- Each input fuzzy set, $A_{x_k'}(k = 1, \ldots, p)$, is a T1 fuzzy set whose MF is a Gaussian function that is centered at x_k' with SD σ_k, i.e.,

$$\mu_{A_{x_k'}}(x_k) = \exp\left\{-\frac{(x_k - x_k')^2}{2\sigma_k^2}\right\} \equiv \phi(x_k; x_k', \sigma_k) \tag{1}$$

  where $\phi(x; m, \sigma)$ represents a Gaussian function for the variable x, with mean m and SD σ. Using such a measurement model leads to a *non-singleton* fuzzifier.
- Each antecedent, F_k^j, is a T1 fuzzy set whose MF is a Gaussian function that is centered at m_k^j with SD σ_k^j, i.e.,

$$\mu_{F_k^j}(x_j) = \phi(x_k; \mu_k^j, \sigma_k^j). \tag{2}$$

- The center-of-sets defuzzification method [21], [27] was used. In this method, each consequent fuzzy set G^j is represented only by its centroid $g^j \in Y$.

a) Operations:: For a T1 FLRBS, the inference engine obtains one firing degree for each rule. We used the Mamdani inference method implemented via the product t-norm, so that the the firing degree of the jth rule, $f^j(\mathbf{x}')(j = 1, \ldots, M)$, is computed by using the sup-star composition as [21], [29]

$$\begin{aligned} f^j(\mathbf{x}') &= \prod_{k=1}^{p} \sup_{x_k} \mu_{A_{x_k'}}(x_k)\mu_{F_k^j}(x_k) \\ &= \prod_{k=1}^{p} \exp\left\{-\frac{\left(x_k' - m_k^j\right)^2}{2\sigma_k^2 + 2\left(\sigma_k^j\right)^2}\right\} \\ &\equiv \prod_{k=1}^{p} \phi\left(x_k'; m_k^j, \sqrt{\sigma_k^2 + \left(\sigma_k^j\right)^2}\right). \end{aligned} \tag{3}$$

For a T1 FLRBS, our output processing consists of center-of-sets defuzzification [21], [27], which leads to the crisp output, $y(\mathbf{x}')$, computed as

$$y(\mathbf{x}') = \frac{\sum_{j=1}^{M} f^j(\mathbf{x}')g^j}{\sum_{j=1}^{M} f^j(\mathbf{x}')}. \tag{4}$$

b) Optimization:: The parameters of a T1 FLRBS include the p input fuzzy set parameters $\sigma_k, 2pM$ antecedent parameters m_k^j and σ_k^j, and M consequent parameters g^j, where $k = 1, \ldots, p$ and $j = 1, \ldots, M$. There are therefore a total of

$$p + M(2p + 1) \tag{5}$$

design parameters. These parameters were tuned so as to minimize a given objective function J (whose structure is provided in Section V) by using the following steepest descent algorithm:

$$\theta_{\text{new}} = \theta_{\text{old}} - \alpha \left.\frac{\partial J}{\partial \theta}\right|_{\theta_{\text{old}}}. \tag{6}$$

In (6) θ is a generic symbol standing for all parameters that are optimized, and $\alpha > 0$ is the learning parameter. Note that $\partial J/\partial\theta$ is computed as:

$$\frac{\partial J}{\partial \theta} = \frac{\partial J}{\partial y}\frac{\partial y}{\partial \theta} \tag{7}$$

where y is the crisp output of the T1 FLRBS. The calculation of $\partial J/\partial y$ depends on the specific form of J, and the formulas for $\partial y/\partial\theta$ are summarized in Appendix A (see, for example, [21], [29], and [37] for the details of these formulas).

B. Interval Type-2 FLRBS

In an interval T2 FLRBS, the input fuzzy sets, antecedents and consequents are all interval T2 fuzzy sets, and the output

TABLE IV
FORMULAS FOR $\underline{f}_k^j(x'_k)$ AND $\overline{f}_k^j(x'_k)$ [21]

$$\underline{f}_k^j(x'_k) \equiv \sup_{x_k} \underline{\mu}_{\tilde{A}_{x'_k}}(x_k)\underline{\mu}_{\tilde{F}_k^j}(x_k)$$

x'_k Location	$\underline{f}_k^j(x'_k)$
$x'_k \le \frac{m_{1,k}^j+m_{2,k}^j}{2} - \frac{\sigma_{1,k}^2\left(m_{2,k}^j-m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}$	$\phi\left(x'_k; m_{2,k}^j, \sqrt{\sigma_{1,k}^2+\left(\sigma_{1,k}^j\right)^2}\right)$
$x'_k \ge \frac{m_{1,k}^j+m_{2,k}^j}{2} + \frac{\sigma_{1,k}^2\left(m_{2,k}^j-m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}$	$\phi\left(x'_k; m_{1,k}^j, \sqrt{\sigma_{1,k}^2+\left(\sigma_{1,k}^j\right)^2}\right)$
otherwise	$\exp\left\{-\frac{\left(m_{2,k}^j-m_{1,k}^j\right)^2}{8\left(\sigma_{1,k}^j\right)^2} - \frac{\left(2x'_k-m_{1,k}^j-m_{2,k}^j\right)^2}{8(\sigma_{1,k})^2}\right\}$

$$\overline{f}_k^j(x'_k) \equiv \sup_{x_k} \overline{\mu}_{\tilde{A}_{x'_k}}(x_k)\overline{\mu}_{\tilde{F}_k^j}(x_k)$$

x'_k Location	$\overline{f}_k^j(x'_k)$
$x'_k \le m_{1,k}^j$	$\phi\left(x'_k; m_{1,k}^j, \sqrt{\sigma_{2,k}^2+\left(\sigma_{2,k}^j\right)^2}\right)$
$x'_k \ge m_{2,k}^j$	$\phi\left(x'_k; m_{2,k}^j, \sqrt{\sigma_{2,k}^2+\left(\sigma_{2,k}^j\right)^2}\right)$
otherwise	1

processing consists of type-reduction followed by defuzzification. For consistency with our later descriptions of FLRBCs, we illustrate the operations of and optimization procedure for the interval T2 FLRBS based on the following assumptions.

- Each input fuzzy set, $\tilde{A}_{x'_k}(k=1,\ldots,p)$, is an interval T2 fuzzy set whose MF is a Gaussian function centered at x'_k with an uncertain SD σ where $\sigma \in [\sigma_{1,k}, \sigma_{2,k}]$, i.e., the lower and upper MFs of $\tilde{A}_{x'_k}$ are given as [21] [see, also, Fig. 3(b)]

$$\underline{\mu}_{\tilde{A}_{x'_k}}(x_k) = \phi(x_k; x'_k, \sigma_{1,k}) \tag{8}$$

$$\overline{\mu}_{\tilde{A}_{x'_k}}(x_k) = \phi(x_k; x'_k, \sigma_{2,k}). \tag{9}$$

- Each antecedent, $\tilde{F}_k^j$, is an interval T2 fuzzy set whose MF is a Gaussian function with an uncertain mean m where $m \in [m_{1,k}^j, m_{2,k}^j]$, and an uncertain SD σ where $\sigma \in [\sigma_{1,k}^j, \sigma_{2,k}^j]$, so that its lower and upper MFs are [37] [see, also, Fig. 3(a)]

$$\underline{\mu}_{\tilde{F}_k^j}(x_k) = \begin{cases} \phi\left(x_k; m_{2,k}^j, \sigma_{1,k}^j\right), & \text{if } x_k \le \frac{m_{1,k}^j+m_{2,k}^j}{2} \\ \phi\left(x_k; m_{1,k}^j, \sigma_{1,k}^j\right), & \text{if } x_k > \frac{m_{1,k}^j+m_{2,k}^j}{2} \end{cases} \tag{10}$$

$$\overline{\mu}_{\tilde{F}_k^j}(x_k) = \begin{cases} \phi\left(x_k; m_{1,k}^j, \sigma_{2,k}^j\right) & \text{if } x_k \le m_{1,k}^j \\ \phi\left(x_k; m_{2,k}^j, \sigma_{2,k}^j\right), & \text{if } x_k > m_{2,k}^j \\ 1, & \text{otherwise} \end{cases} \tag{11}$$

- A special case[3] of the center-of-sets type-reduction method—height type-reduction— was used [19], [21]. In this method, each consequent fuzzy set $\tilde{G}^j$ is represented as a crisp number $g^j \in Y$.

[3]In the original center-of-sets type-reduction method, each consequent fuzzy set $\tilde{G}^j$ is represented by its centroid which is an interval $[g_l^j, g_r^j]$. Because the consequent fuzzy sets in our classification problem correspond to the vehicle category labels and are represented by crisp numbers, this original type-reduction method was tailored to fit in with the special situation.

c) Operations: For an interval T2 FLRBS, the inference engine obtains two firing degrees for each rule—the *lower* and *upper firing degrees*—by using the extended sup-star composition [21], i.e., the lower and upper firing degrees of the j-th rule, $\underline{f}^j(\mathbf{x}')$ and $\overline{f}^j(\mathbf{x}')$, are computed[4] as

$$\underline{f}^j(\mathbf{x}') = \prod_{k=1}^{p} \sup_{x_k} \underline{\mu}_{\tilde{A}_{x'_k}}(x_k)\underline{\mu}_{\tilde{F}_k^j}(x_k) \equiv \prod_{k=1}^{p} \underline{f}_k^j(x'_k) \tag{12}$$

$$\overline{f}^j(\mathbf{x}') = \prod_{k=1}^{p} \sup_{x_k} \overline{\mu}_{\tilde{A}_{x'_k}}(x_k)\overline{\mu}_{\tilde{F}_k^j}(x_k) \equiv \prod_{k=1}^{p} \overline{f}_k^j(x'_k). \tag{13}$$

The formulas for computing $\underline{f}_k^j(x'_k)$ and $\overline{f}_k^j(x'_k)$ are summarized in Table IV.

For an interval T2 FLRBS, output processing consists of type-reduction followed by defuzzification. Type-reduction obtains an interval output, $[y_l(\mathbf{x}'), y_r(\mathbf{x}')]$, by using the iterative procedures developed by Karnik and Mendel [11], [21], that use the just-computed lower and upper firing degrees as well as the consequent g^j values. The end-points of the type-reduced set, $y_l(\mathbf{x}')$ and $y_r(\mathbf{x}')$, can be expressed[5] as (see Appendix B)

$$y_l(\mathbf{x}') = \frac{\sum_{j=1}^{M} g^j\left\{\delta_l^j \overline{f}^j(\mathbf{x}') + \left(1-\delta_l^j\right)\underline{f}^j(\mathbf{x}')\right\}}{\sum_{j=1}^{M}\left\{\delta_l^j \overline{f}^j(\mathbf{x}') + \left(1-\delta_l^j\right)\underline{f}^j(\mathbf{x}')\right\}} \tag{14}$$

$$y_r(\mathbf{x}') = \frac{\sum_{j=1}^{M} g^j\left\{\delta_r^j \overline{f}^j(\mathbf{x}') + \left(1-\delta_r^j\right)\underline{f}^j(\mathbf{x}')\right\}}{\sum_{j=1}^{M}\left\{\delta_r^j \overline{f}^j(\mathbf{x}') + \left(1-\delta_r^j\right)\underline{f}^j(\mathbf{x}')\right\}}. \tag{15}$$

[4]Due to space limitations, we do not provide derivations here. They are in [19], [21].

[5]The reasons that we introduce δ_l^j and δ_r^j into the formulas of $y_l(\mathbf{x}')$ and $y_r(\mathbf{x}')$ are mainly for the convenience of partial derivative formulas, as shown in (C.17)–(C.20)

Defuzzification of the interval T2 FLRBS obtains a crisp output, $y(\mathbf{x}')$, from the type-reduced set as

$$y(\mathbf{x}') = \frac{1}{2}[y_l(\mathbf{x}') + y_r(\mathbf{x}')]. \tag{16}$$

Optimization: The parameters of an interval T2 FLRBS include the $2p$ input fuzzy set parameters $\{\sigma_{1,k}, \sigma_{2,k}\}, 4pM$ antecedent parameters $\{m^j_{1,k}, m^j_{2,k}\}$ and $\{\sigma^j_{1,k}, \sigma^j_{2,k}\}$, and M consequent parameters g^j, where $k = 1, \ldots, p$ and $j = 1, \ldots, M$. There are, therefore, a total of

$$2p + M(4p + 1) \tag{17}$$

design parameters. These parameters were also tuned by using the steepest descent algorithm in (6). The partial derivatives $\partial J/\partial\theta$ of the interval T2 FLRBS were also computed as in (7), where the formulas for $\partial y/\partial\theta$ for all of the parameters of the interval T2 FLRBS are summarized in Appendix C (see [37] for details of these formulas).

V. Classifier Designs

In this section, we design three architectures for the FLRBCs—nonhierarchical, hierarchical in parallel and hierarchical in series—each of which is implemented based on T1 or interval T2 fuzzy set and fuzzy logic theories. The performance of these FLRBCs are base-lined against a Bayesian classifier that is also described below.

In the rest of this paper, without any explicit indications, we use lower-case letters (e.g., $\mathbf{x}$ or x_k) to represent numbers, and upper-case letters (e.g., A) to represent random variables (for the Bayesian classifier) or fuzzy sets (for FLRBCs).

A. Bayesian Classifier

The Bayesian classifier was established based on assumptions about the conditional probability density function (pdf) of the features in each category. This classifier consists of nine conditional probability models, each of which corresponds to one kind of vehicle, and is described by a Gaussian pdf

$$p(\mathbf{x}|V_j) \sim N(\mathbf{x}; \mathbf{m}_j, \Sigma_j), \qquad j = 1, \ldots, 9 \tag{18}$$

where $\mathbf{x} \equiv [x_1, x_2, \ldots, x_{11}]^t$ represents the feature vector and V_j represents the jth kind of vehicle. $\mathbf{m}_j$ and Σ_j are the mean and covariance matrix of the multivariate Gaussian distribution associated with V_j, and are estimated as the sample mean and sample covariance of the training prototypes (how prototypes were divided into the training and testing sets is described in Section VI) of V_j.

Note that one can use more complicated models, e.g., Gaussian mixture models, to describe the feature distributions of each kind of vehicle. As mentioned earlier in Section I, the performance of a classifier is heavily dependent upon the designer's experience and a priori knowledge about the problem. We chose the above model because it is compatible with using one rule per vehicle in a FLRBC.

Given an unlabeled feature vector $\mathbf{x}'$ as the input, our Bayesian classifier first computes the log-likelihood $L(\mathbf{x}'|V_j)$ for each kind of vehicle as

$$\begin{aligned} L(\mathbf{x}'|V_j) &= \log p(\mathbf{x}'|V_j) \\ &= -\log|\Sigma_j| - (\mathbf{x}' - \mathbf{m}_j)^t \Sigma_j^{-1} (\mathbf{x}' - \mathbf{m}_j) \end{aligned} \tag{19}$$

and then compares all likelihoods to determine $V_{\max}$ that is associated with the maximum likelihood. Finally, our Bayesian classifier assigns $\mathbf{x}'$ to the same category as $V_{\max}$.

B. Nonhierarchical FLRBC

The nonhierarchical FLRBC is based on encoding the labels of the four categories into two-dimensional vectors, with $[\text{positive}, \text{positive}]^t$, $[\text{positive}, \text{negative}]^t$, $[\text{negative}, \text{positive}]^t$, and $[\text{negative}, \text{negative}]^t$ corresponding to the heavy-tracked, light-tracked, heavy-wheeled, and light-wheeled vehicles, respectively. It is constructed as a traditional FLRBS but with two-dimensional consequents and outputs. The rule base of the non-hierarchical FLRBC has nine rules, one for each kind of vehicle, i.e.,

R^j: IF x_1 is F^j_1 and $\ldots$ and x_{11} is F^j_{11}, THEN y is $[g^j_1, g^j_2]^t, j = 1, \ldots, 9$

where R^j represents the rule for the jth kind of vehicle V_j, and the consequent $[g^j_1, g^j_2]^t$ is a two-dimensional vector encoding the category label of V_j. Given an unlabeled feature vector $\mathbf{x}' \equiv [x'_1, \ldots, x'_{11}]^t$ as the input, this FLRBC obtains a two-dimensional output vector $[y_1(\mathbf{x}'), y_2(\mathbf{x}')]^t$ through fuzzification, inference and output processing[6], and makes a final decision for $\mathbf{x}'$ based only on the signs of $[y_1(\mathbf{x}'), y_2(\mathbf{x}')]^t$ according to Table V.

This FLRBC architecture was implemented using both T1 and interval T2 fuzzy sets, as described in Sections 4.1 and 4.2, respectively. For our application there are 11 features ($p = 11$) and nine rules ($M = 9$); hence, based on (5)[7] there are a total of the 227 parameters in the T1 nonhierarchical FLRBC. Because a rule corresponds to one kind of vehicle, the consequent and antecedent parameters of each rule were initialized during optimization based on the category label and the training prototypes of its corresponding vehicle. More specifically, the consequent $[g^j_1, g^j_2]^t$ of a rule was initialized as $[+1, +1]^t$ ($[+1, -1]^t$, $[-1, +1]^t$ or $[-1, -1]^t$) if the vehicle belonged to the heavy-tracked (light-tracked, heavy-wheeled or light-wheeled) category. The antecedent parameters of a rule, m^j_k and σ^j_k, were initialized as the sample mean and sample SD of the training prototypes of the corresponding vehicle, and the input parameters, $\sigma_k (k = 1, \ldots, 11)$, were initialized as the average σ^j_k over all nine rules.

[6]The computations for the output processing [i.e., (4) for the T1 implementations, and (14)–(16) for the interval T2 implementations] are applied twice, one for each dimension, with g^j being substituted by g^j_n ($n = 1$ or 2) for the computations of the nth dimension.

[7]Equation (5) has been changed to $p + M(2p + 2)$ when it is used, since there are two consequent parameters for each rule in this non-hierarchical FLRBC architecture as well as in sub-system 2 of the HFLRBC-S architecture as detailed in Section V-D.

Based on (17)[8] there are a total of 436 parameters in the interval T2 nonhierarchical FLRBC. These parameters were initialized during optimization based on the final optimal parameters of the competing (just-designed) T1 nonhierarchical FLRBC, according to the following:

$$\begin{aligned} g_1^j(0) &\equiv g_1^j(\text{T1 optimal}) \\ g_2^j(0) &\equiv g_2^j(\text{T1 optimal}) \end{aligned} \tag{20}$$

$$\begin{aligned} m_{1,k}^j(0) &\equiv m_k^j(\text{T1 optimal}) - \gamma\sigma_k^j(\text{T1 optimal}) \\ m_{2,k}^j(0) &\equiv m_k^j(\text{T1 optimal}) + \gamma\sigma_k^j(\text{T1 optimal}) \end{aligned} \tag{21}$$

$$\begin{aligned} \sigma_{1,k}^j(0) &\equiv (1-\gamma)\sigma_k^j(\text{T1 optimal}) \\ \sigma_{2,k}^j(0) &\equiv (1+\gamma)\sigma_k^j(\text{T1 optimal}) \end{aligned} \tag{22}$$

$$\begin{aligned} \sigma_{1,k} &\equiv (1-\gamma)\sigma_k(\text{T1 optimal}) \\ \sigma_{2,k} &\equiv (1+\gamma)\sigma_k(\text{T1 optimal}) \end{aligned} \tag{23}$$

where $\gamma \in (0,1)$ and was set equal to 0.1 during the experiments. Note that in these equations the left-hand sides are the initial values of the parameters of the interval T2 nonhierarchical FLRBC, and the right-hand sides are the optimal values of the parameters of the previously-designed T1 non-hierarchical FLRBC.

During training all of the 227 the parameters of the T1 non-hierarchical FLRBC and the 436 parameters of the T2 non-hierarchical FLRBC were optimized by using the steepest descent algorithm in (6) to minimize the following classification error objective function:

$$J = \sum_{\mathbf{x}\in V_{\text{train}}} \frac{1}{2}\{[d_1 - y_1(\mathbf{x})]^2 + [d_2 - y_2(\mathbf{x})]^2\} \tag{24}$$

where V_{train} represents the training set, and $[d_1, d_2]^t$ is the desired classification result for $\mathbf{x}$, i.e., $[d_1, d_2]^t = [+1,+1]^t$ ($[+1,-1]^t, [-1,+1]^t$ or $[-1,-1]^t$) if $\mathbf{x}$ belongs to the heavy-tracked (light-tracked, heavy-wheeled, or light-wheeled) category.

C. Hierarchical FLRBC in Parallel

The hierarchical FLRBC in parallel (HFLRBC-P) consists of four sub-systems, as shown in Fig. 4. The first three sub-systems, which operate in parallel, are for the binary classification of *tracked* versus *wheeled* (T/W), *heavy-tracked* versus *light-tracked* (HT/LT), and *heavy-wheeled* versus *light-wheeled* (HW/LW) categories, respectively. The fourth sub-system makes final decisions based on the decisions from the first three sub-systems.

Each of the first three sub-systems is a FLRBS. Sub-system 1 contains nine rules for the classification of tracked versus wheeled categories, each of which corresponds to one kind of vehicle; sub-system 2 contains five rules for the classification of heavy-tracked versus light-tracked categories, each of which corresponds to one kind of *tracked* vehicle (see Table II); and sub-system 3 contains four rules for the classification of heavy-wheeled versus light-wheeled categories, each of which corresponds to one kind of *wheeled* vehicle. Despite these differences, the rules of these sub-systems all have the same form, namely

R^{j_i}: IF x_1 is $F_1^{j_i}$ and ... and x_{11} is $F_{11}^{j_i}$, THEN y is g^{j_i}, $j = 1,\ldots,M_i$ and $i = 1,\ldots,3$

where M_i is the number of rules of sub-system i, R^{j_i} represents the jth rule of sub-system i, and the consequent g^{j_i} is a scalar. Given an unlabeled feature vector $\mathbf{x}'$ as the input, each of the first three sub-systems obtains an intermediate output $y_i(\mathbf{x}')(i = 1,2,3)$ through fuzzification, inference and output processing. Sub-system 1 computes two weights, $w_t(\mathbf{x}') \in [0,1]$ and $w_h(\mathbf{x}') \in [0,1]$, by using the the following logistic function with parameter $\beta_1 > 0$ (β_1 is optimized during training):

$$\begin{aligned} w_t(\mathbf{x}') &\equiv [1 + \exp\{-\beta_1 y_1(\mathbf{x}')\}]^{-1} \\ w_w(\mathbf{x}') &\equiv 1 - w_t(\mathbf{x}'). \end{aligned} \tag{25}$$

Weights $w_t(\mathbf{x}')$ and $w_h(\mathbf{x}')$ are associated with *tracked* and *wheeled* categories, respectively. Similarly, sub-system 2 computes two *conditional* weights, $w_{ht|t}(\mathbf{x}') \in [0,1]$ and $w_{lt|t}(\mathbf{x}') \in [0,1]$, which are conditioned on $\mathbf{x}'$ belonging to the tracked category and are associated with *heavy-tracked* and *light-tracked* categories, respectively, as ($\beta_2 > 0$ and is optimized during training)

$$\begin{aligned} w_{\text{ht}|t}(\mathbf{x}') &\equiv [1 + \exp\{-\beta_2 y_2(\mathbf{x}')\}]^{-1} \\ w_{\text{lt}|t}(\mathbf{x}') &\equiv 1 - w_{\text{ht}|t}(\mathbf{x}'). \end{aligned} \tag{26}$$

Sub-system 3 also computes two *conditional* weights, $w_{hw|w}(\mathbf{x}') \in [0,1]$ and $w_{lw|w}(\mathbf{x}') \in [0,1]$, which are conditioned on $\mathbf{x}'$ belonging to the wheeled vehicle and are

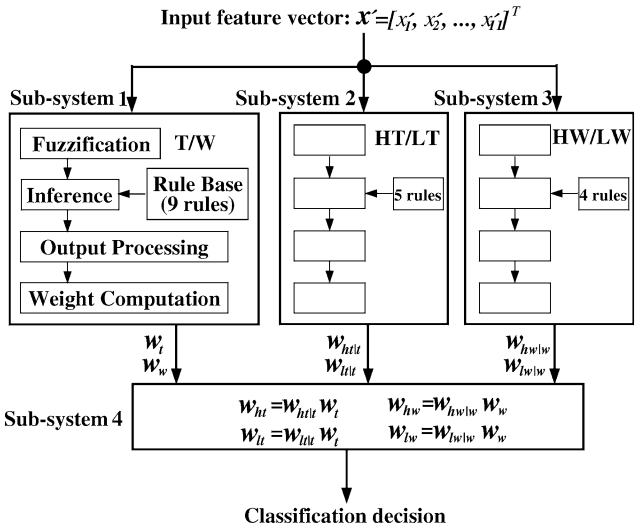


Fig. 4. HFLRBC-P that consists of four sub-systems.

[8]Equation (17) has been changed to $2p + M(4p+2)$ when it is used, since there are two consequent parameters for each rule in this non-hierarchical FLRBC architecture as well as in sub-system 2 of the HFLRBC-S architecture as detailed in Section V-D.

associated with *heavy-wheeled* and *light-wheeled* categories, respectively, as ($\beta_3 > 0$ and is also optimized during training)

$$w_{\mathrm{hw}|w}(\mathbf{x}') \equiv [1+\exp\{-\beta_3 y_3(\mathbf{x}')\}]^{-1}$$
$$w_{\mathrm{lw}|w}(\mathbf{x}') \equiv 1 - w_{\mathrm{hw}|w}(\mathbf{x}'). \quad (27)$$

Sub-system 4 then computes four weights, $w_{\mathrm{ht}}(\mathbf{x}')$, $w_{\mathrm{lt}}(\mathbf{x}')$, $w_{\mathrm{hw}}(\mathbf{x}')$ and $w_{\mathrm{lw}}(\mathbf{x}')$, that are associated with the heavy-tracked, light-tracked, heavy-wheeled, and light-wheeled vehicle categories, respectively, as

$$w_{\mathrm{ht}}(\mathbf{x}') = w_{\mathrm{ht}|t}(\mathbf{x}')w_t(\mathbf{x}')$$
$$w_{\mathrm{lt}}(\mathbf{x}') = w_{lt|t}(\mathbf{x}')w_t(\mathbf{x}')$$
$$w_{\mathrm{hw}}(\mathbf{x}') = w_{hw|w}(\mathbf{x}')w_w(\mathbf{x}')$$
$$w_{\mathrm{lw}}(\mathbf{x}') = w_{lw|w}(\mathbf{x}')w_w(\mathbf{x}'). \quad (28)$$

These weights can be interpreted as *approximations* to the posterior probabilities for the four categories, e.g., $w_t(\mathbf{x}')$ is the approximate probability of $\mathbf{x}'$ belonging to the tracked category, $w_{ht|t}(\mathbf{x}')$ is the approximate conditional probability of $\mathbf{x}'$ belonging to the heavy-tracked category given that it belongs to the tracked category, and $w_{\mathrm{ht}}(\mathbf{x}')$ is the approximate probability of $\mathbf{x}'$ belonging to the heavy-tracked category. Sub-system 4 makes a decision for the input feature vector based on the four weights in (28), i.e., by assigning $\mathbf{x}'$ to the category with the highest weight.

The HFLRBC-P architecture was implemented also using both T1 and interval T2 fuzzy sets. In the T1 HFLRBC-P, the antecedents and fuzzifier were modeled using the same assumptions we made in Section IV-A for the T1 FLRBS; and, in the interval T2 HFLRBC-P, the antecedents and fuzzifier were modeled using the same assumptions we made in Section IV-B for the interval T2 FLRBS. The computations of sub-systems 1–3 are as described in Section 4. Note, also, that sub-system 4 is the same for both the T1 and interval T2 HFLRBC-Ps.

By applying (5) to sub-systems 1–3, and adding the three parameters for $\beta_i (i = 1, \ldots, 3)$, it is straightforward to show that there are a total of 450 parameters for the T1 HFLRBC-P. Since each rule corresponds to one kind of vehicle, the consequent and antecedent parameters of this rule were initialized during optimization based on the label and prototypes of its corresponding vehicle. More specifically, in sub-system 1, the consequent $g^{j_1} (j_1 = 1, \ldots, 9)$ of one rule was initialized as $+1$ (or -1) if its corresponding vehicle belonged to the tracked (or wheeled) category; in sub-system 2, the consequent $g^{j_2} (j_2 = 1, \ldots, 5)$ of one rule was initialized as $+1$ (or -1) if its corresponding vehicle belonged to the heavy-tracked (or light-tracked) category; and in sub-system 3, the consequent $g^{j_3} (j_3 = 1, \ldots, 4)$ of one rule was initialized as $+1$ (or -1) if its corresponding vehicle belonged to the heavy-wheeled (or light-wheeled) category. In each sub-system, the antecedent and input parameters were initialized in the same way as described in Section V-B for the T1 nonhierarchical FLRBC, and the parameters for the logistic function, $\beta_i (i = 1, 2, 3)$, were initialized as 1.

By applying (17) to sub-systems 1–3, and adding the three parameters for $\beta_i (i = 1, \ldots, 3)$, it is straightforward to show that there are a total of 879 parameters for the interval T2 HFLRBC-P. These parameters were initialized during optimization based on the optimal parameters of the competing (just-designed) T1 HFLRBC-P as in (20)–(23). Additionally, $\beta_i (i = 1, 2, 3)$, were initialized as

$$\beta_i(0) = \beta_i(\mathrm{T1\ optimal}) \quad (29)$$

where $\beta_i(0)$ represents the initial value for the interval T2 HFLRBC-P, and $\beta_i(\mathrm{T1\ optimal})$ represents the optimal value for the T1 HFLRBC-P.

During training, all the parameters of the HFLRBC-P were tuned simultaneously by using the steepest descent algorithm in (6) to minimize the following classification error objective function:

$$J = \sum_{\mathbf{x}\in V_{\mathrm{train}}} \frac{1}{2}\{[w_{\mathrm{ht}}(\mathbf{x}) - d_{\mathrm{ht}}(\mathbf{x})]^2 + [w_{\mathrm{lt}}(\mathbf{x}) - d_{\mathrm{lt}}(\mathbf{x})]^2 + [w_{\mathrm{hw}}(\mathbf{x}) - d_{\mathrm{hw}}(\mathbf{x})]^2) + [w_{\mathrm{lw}}(\mathbf{x}) - d_{\mathrm{lw}}(\mathbf{x})]^2\} \quad (30)$$

where $[d_{\mathrm{ht}}(\mathbf{x}), d_{\mathrm{lt}}(\mathbf{x}), d_{\mathrm{hw}}(\mathbf{x}), d_{\mathrm{lw}}(\mathbf{x})]$ are the desired weight values for the heavy-tracked, light-tracked, heavy-wheeled and light-wheeled categories, i.e., $[d_{\mathrm{ht}}(\mathbf{x}), d_{\mathrm{lt}}(\mathbf{x}), d_{\mathrm{hw}}(\mathbf{x}), d_{\mathrm{lw}}(\mathbf{x})] = [1,0,0,0], [0,1,0,0], [0,0,1,0]$ or $[0,0,0,1]$ when $\mathbf{x}$ is a feature vector of the heavy-tracked, light-tracked, heavy-wheeled, or light-wheeled vehicle, respectively.

D. Hierarchical FLRBC in Series

The hierarchical FLRBC in series (HFLRBC-S) consists of three sub-systems, as shown in Fig. 5. Sub-system 1 is for the binary classification of *tracked* versus *wheeled* categories, sub-system 2 is for the four-category classification of *heavy-tracked*, *light-tracked*, *heavy-wheeled* and *light-wheeled* vehicles and utilizes the output of the first sub-system as well as the input feature measurements, and, sub-system 3 makes final decisions based on the outputs of the second sub-system.

Both sub-systems 1 and 2 are complete and separate FLRBSs. Sub-system 1 consists of nine rules, each of which is associated with one kind of vehicle and has the following form:

R^{j_1}: IF x_1 is $F_1^{j_1}$ and ... and x_{11} is $F_{11}^{j_1}$, THEN y is $g^{j_1}, j_1 = 1, \ldots, 9$,

where R^{j_1} represents the j-th rule of sub-system 1, and the consequent g^{j_1} is a scalar. Sub-system 2 also consists of nine rules, each of which is associated with one kind of vehicle and has the following form:

R^{j_2}: IF y is $F_0^{j_2}$ and x_1 is $F_1^{j_2}$ and ... and x_{11} is $F_{11}^{j_2}$, THEN z is $[g_1^{j_2}, g_2^{j_2}]^t, j_2 = 1, \ldots, 9$,

where R^{j_2} represents the j-th rule of sub-system 2, and the consequent $[g_1^{j_2}, g_2^{j_2}]^t$ is a two-dimensional vector. Additionally, the antecedent $F_0^{j_2}$ for the intermediate output of the first

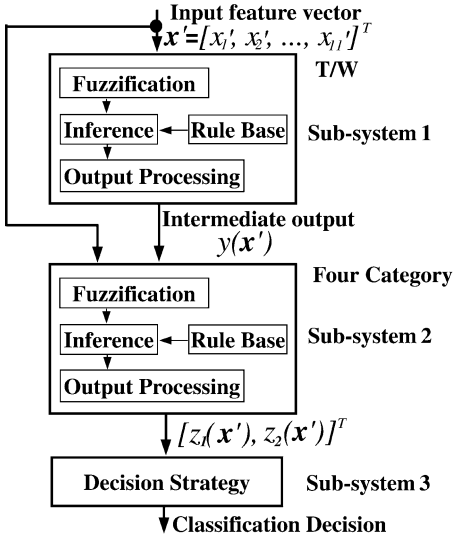


Fig. 5. HFLRBC-S that consists of three sub-systems.

TABLE V
CLASSIFICATION DECISION FOR $\mathbf{x}'$ BASED ON $[y_1(\mathbf{x}'), y_2(\mathbf{x}')]^t$

Decision	$y_1(\mathbf{x}')$	$y_2(\mathbf{x}')$
heavy-tracked	positive	positive
light-tracked	positive	negative
heavy-wheeled	negative	positive
light-wheeled	negative	negative

sub-system was modeled as a T1 fuzzy set with logistic MF, i.e.,

$$\mu_0^{j_2}(y) \equiv [1 + \exp\{-\beta^{j_2} y\}]^{-1} \tag{31}$$

where β^{j_2} was initialized during optimization as $+1$ (or -1) if R^{j_2} was associated with a tracked (or wheeled) vehicle.

Given an unlabeled feature vector $\mathbf{x}'$ as the input, sub-system 1 obtains an output $y(\mathbf{x}')$ through fuzzification, inference and output processing. This output together with the feature vector $\mathbf{x}'$ are then used as the inputs to sub-system 2, which obtains the crisp output vector $[z_1(\mathbf{x}'), z_2(\mathbf{x}')]^t$ also through fuzzification, inference and output processing. Sub-system 3 makes the final classification decision for $\mathbf{x}'$ based on the signs of $[z_1(\mathbf{x}'), z_2(\mathbf{x}')]^t$ according to Table V [with the replacements $y_i(\mathbf{x}') \to z_i(\mathbf{x}')$ for $i = 1, 2$].

The HFLRBC-S architecture was also implemented using both T1 and interval T2 fuzzy sets. In the T1 HFLRBC-S, except for the antecedent and fuzzifier of sub-system 2 for the intermediate output $y(\mathbf{x}')$, all the other antecedents and fuzzifier were modeled using the same assumptions we made in Section 4.1 for the T1 FLRBS; and the fuzzifier of sub-system 2 encoded the intermediate output of the first sub-system $y(\mathbf{x}')$ into a singleton fuzzy set.

In the interval T2 HFLRBC-S, except for the antecedent and fuzzifier of sub-system 2 for the intermediate output $y(\mathbf{x}')$, all the other antecedents and fuzzifier were modeled using the same assumptions we made in Section 4.2 for the interval T2 FLRBS; and the fuzzifier encoded the intermediate output $y(\mathbf{x}')$ into an interval T1 fuzzy set $[y_l(\mathbf{x}'), y_r(\mathbf{x}')]$, which is the type-reduced set of the first sub-system. The computations of sub-systems 1 and 2 are as described in Section IV. Note, also, that sub-system 3 is the same for both the T1 and interval T2 FLRBC-Ss.

By applying (5) (see footnote 7) to sub-systems 1 and 2, and adding the p parameters for $\beta^{j_2}(j_2 = 1, \ldots, 9)$, it is straightforward to show that there are a total of 454 parameters in the T1 HFLRBC-S. Since each rule corresponds to one kind of vehicle, the consequent and antecedent parameters of this rule were initialized during optimization based on the label and the prototypes of its corresponding vehicle, i.e., in sub-system 1, the consequent $g^{j_1}(j_1 = 1, \ldots, 9)$ was initialized as $+1$ (or -1) if its corresponding vehicle belonged to the tracked (or wheeled) category, and the antecedent and input parameters were initialized in the same way as described in Section V-B for the T1 nonhierarchical FLRBC. In sub-system 2, the consequent $[g_1^{j_2}, g_2^{j_2}]^t$ was initialized as $[+1, +1]^t$ ($[+1, -1]^t, [-1, +1]^t$ and $[-1, -1]^t$) if its corresponding vehicle belonged to the heavy-tracked (light-tracked, heavy-wheeled, or light-wheeled) category; the parameter of the antecedent $F_0^{j_2}, \beta^{j_2}(j_2 = 1, \ldots, 9)$, was initialized as $+1$ (or -1) if its corresponding vehicle belonged to the tracked (or wheeled) category; and, the remaining antecedent and input parameters were initialized in the same way as described in Section V-B for the T1 nonhierarchical FLRBC.

By applying (17) (see footnote 8) to sub-systems 1 and 2, and adding the p parameters for $\beta^{j_2}(j_2 = 1, \ldots, 9)$, it is straightforward to show that there are a total of 872 parameters in the interval T2 HFLRBC-S. These parameters were initialized based on the optimal parameters of the competing T1 HFLRBC-S in a similar way to (20)–(23). Additionally, in sub-system 2, the parameter of the antecedent $F_0^{j_2}, \beta^{j_2}(j_2 = 1, \ldots, 9)$, was initialized as

$$\beta^{j_2}(0) = \beta^{j_2}(\text{T1 optimal}) \tag{32}$$

where $\beta^{j_2}(0)$ represents the initial value for the interval T2 HFLRBC-S, and $\beta^{j_2}(\text{T1 optimal})$ represents the optimal value for the T1 HFLRBC-S.

During training, all the parameters of the HFLRBC-S were optimized by using the steepest descent algorithm in (6) to minimize the objective function in (24) [with the replacements $y_i(\mathbf{x}') \to z_i(\mathbf{x}')$ for $i = 1, 2$].

VI. EXPERIMENTS AND RESULTS

We performed various experiments to evaluate the Bayesian classifier and FLRBCs. Here we only provide the details and results for the *10-fold cross-validation* and *leave-two-runs-out* experiments. Additional experiments can be found in [37]. Note that this is an application where[9] 80% or better correct classification is considered to be excellent.

A. 10-Fold Cross Validation Experiment

In the 10-fold cross validation experiment, we first randomly divided the CPA-based prototypes of all runs into 10 even folds, and then performed 10 designs for each classifier. In the t-th design, the prototypes of the t-th fold were used for testing, and the prototypes of the remaining nine folds were used for training.

[9]This number was provided to us by our ARL sponsors.

The pseudo code for this 10-fold cross validation experiment is:

```
Index the CPA-based prototypes of all runs;
//There are 80 CPA-based prototypes/run ×89
runs in total.
Randomly permute the index of all CPA-based
    prototypes;
Divide all CPA-based prototypes into 10 even
    folds according to their indices;
for t = 1 : 10//10 designs in total
    { Use CPA-based prototypes of the tth
      fold for testing;
    Use CPA-based prototypes of the remaining
      folds for training;
    Estimate the parameters for the Bayesian
      classifier by using the training data;
    Test the Bayesian classifier using testing
      prototypes, and save the testing
      classification error rate as p_b(t);
    Initialize the parameters of the T1 FLRBC;
    for n = 1 : 1000
    //train and test for 1000 epochs
      { Optimize the T1 FLRBC using training
          prototypes;
        Test the T1 FLRBC using testing
          prototypes, and save the testing
          classification error rate as e_1(t,n);
      }
    Save the minimum of e_1(t,n) for n = 1,...,1000
      as p_1(t);
    Save the optimal parameters as θ_1(t);
    Initialize the parameters of the interval
      T2 FLRBC based on θ_1(t);
    for n = 1 : 1000
    //train and test for 1000 epochs
      { Optimize the interval T2 FLRBC using
          training prototypes;
        Test the interval T2 FLRBC using
          testing prototypes, and save the
          testing classification error rate
          as e_2(t,n);
      }
    Save the minimum of e_2(t,n) for n = 1,...,1000
      as p_2(t);
    Save the optimal parameters as θ_2(t);
  }
Compute the mean and SD of p_b(t), p_1(t) and p_2(t)
  for t = 1,...,10.
```

TABLE VI
AVERAGE AND SD OF THE TESTING CLASSIFICATION ERROR RATES ACROSS THE 10 DESIGNS OF THE 10-FOLD CROSS-VALIDATION EXPERIMENT, AND THE NUMBER OF DESIGNS THAT THE TESTING CLASSIFICATION ERROR RATE OF ONE (T1 OR INTERVAL T2) IMPLEMENTATION IS SMALLER THAN THE OTHER. NOTE THAT THE DESIGNS WHERE THE TESTING CLASSIFICATION ERROR RATES OF THE T1 AND INTERVAL T2 IMPLEMENTATIONS ARE TIED ARE NOT COUNTED

Classifier	Average	SD	T1 vs. interval T2
Bayesian	22.0084%	0.009007	
T1 Non-hierarchical	14.1994%	0.012605	2
interval T2 Non-hierarchical	13.6517%	0.009838	7
T1 HFLRBC-P	13.8624%	0.014776	3
interval T2 HFLRBC-P	13.3146%	0.009335	6
T1 HFLRBC-S	11.4747%	0.008324	1
interval T2 HFLRBC-S	10.8427%	0.008678	7

The results of the 10-fold cross-validation experiments are summarized in Table VI. Observe that: 1) while the SD values of the testing classification errors of all classifiers are of the same magnitude, the mean values of the testing classification errors of all FLRBCs are much smaller than that of the Bayesian classifier, which suggests that all FLRBCs perform much better than the Bayesian classifier; 2) for most of the designs, the interval T2 FLRBCs have smaller testing classification errors than their competing T1 FLRBCs; 3) in terms of the mean value of the testing classification error, the interval T2 HFLRBC-S architecture performs the best; and 4) all FLRBCs are able to achieve higher than the 80% acceptable classification accuracy.

B. Leave-Two-Runs-Out Experiment

Leave-two-runs-out means that we randomly picked one tracked *run* and wheeled *run* so as to use their prototypes for testing, and then used the prototypes of the remaining runs for training. There are 61 tracked runs and 28 wheeled runs (see Table II); hence, this experiment could be performed $61 \times 28 = 1,708$ times, each experiment corresponding to one possible combination of one tracked run and one wheeled run. However, because 1 708 is a rather large number, we only performed this experiment 200 times, and then used the bootstrap method [39] to estimate the confidence interval of the mean of the classification error rate.

During this experiment, we established the concepts of *non-adaptive* and *adaptive operational modes*, where the former refers to the classification decision for one data block $\mathbf{x}'$ being based only on $\mathbf{x}'$ itself, whereas the latter refers to the classification decision for $\mathbf{x}'$ being based on all available data blocks prior to and including $\mathbf{x}'$. We used the following simple majority voting rule: the adaptive decision of $\mathbf{x}'$ is the same as the majority of the nonadaptive decisions for the data blocks prior to and including $\mathbf{x}'$. Theoretical analyses that are based on the assumption that local decisions for different observations are independent, as well as experiments, have shown that the simple majority vote can greatly improve the performance of decision-making [4], [7], [12], [13], [18], [22], [28], [31], [38]. We did not use any other fusion method because of the computational simplicity of the majority vote, something that is very important when our classifiers would be used in a battlefield environment. In our case, when the local decisions are dependent, because they use some of the same measurements, we still observed that the simple majority-voting based adaptive working mode achieved much better performance than the non-adaptive working model. Other ways to combine multiple classifiers are described in [16].

The pseudo-code for the leave-two-runs-out experiment is:

```
for t = 1 : 200 //200 times in total
  { Randomly pick one tracked run and one
      wheeled run out of the pool, and use
      their Non-CPA-based prototypes for
      testing;
    Use CPA-based prototypes of the remaining
      runs for training;
    Estimate the parameters of the Bayesian
      classifier using training prototypes;
    Test the Bayesian classifier using testing
      prototypes, and save the classification
      error rate for the non-adaptive mode as
      p_b(t), and for the adaptive mode as p_b^a(t);
    Initialize the parameters of the T1 FLRBC
      using training prototypes;
    for n = 1 : 400 //train and test for 400
    epochs
      { Optimize the T1 FLRBC using training
          prototypes;
        Test the T1 FLRBC for the
        non-adaptive
          mode using testing prototypes, and
          save the testing classification
          error rate as e_1(t,n);
      }
    Save the minimum of e_1(t,n) for n = 1,...,400
      as p_1(t);
    Save the optimal parameters as θ_1(t);
    Test θ_1(t) for the adaptive mode, and save
      the classification error rate as p_1^a(t);
    Initialize the parameters of the interval
      T2 FLRBC based on θ_1(t);
    for n = 1 : 400
    //train and test for 400 epochs
      { Optimize the interval T2 FLRBC using
          training prototypes;
        Test the interval T2 FLRBC for the
          non-adaptive mode using testing
          prototypes, and save the testing
          classification error rate as e_2(t,n);
      }
    Save the minimum of e_2(t,n) for n = 1,...,400
      as p_2(t);
    Save the optimal parameters as θ_2(t);
    Test θ_2(t) for the adaptive mode, and save
      the classification error rate
      as p_2^a(t)
      ;
  }
Compute the mean of the classification error
  rates for t = 1,...,200, and apply the
  bootstrap method to calculate their 95%
  confidence intervals.
```

TABLE VII
AVERAGE AND 95% CONFIDENCE INTERVAL (CI) OF THE AVERAGE OF THE CLASSIFICATION ERROR RATES FOR THE LEAVE-TWO-RUNS-OUT EXPERIMENTS

Classifier		Non-Adaptive	Adaptive
Bayesian	Average	34.3451%	22.3464%
	CI	[31.9935%, 36.7703%]	[18.9374%, 25.9359%]
T1	Average	17.6786%	8.2966%
Non-hierarchical	CI	[16.6279%, 18.7804%]	[6.8075%, 9.8960%]
interval T2	Average	16.7695%	7.3657%
Non-hierarchical	CI	[15.7636%, 17.7849%]	[5.9490%, 8.8957%]
T1	Average	21.2602%	9.9866%
HFLRBC-P	CI	[19.9713%, 22.5182%]	[8.1770%, 11.8454%]
interval T2	Average	19.2968%	8.4532%
HFLRBC-P	CI	[18.0315%, 20.5908%]	[6.5671%, 10.4044%]
T1	Average	17.1055%	6.6670%
HFLRBC-S	CI	[16.1410%, 18.0746%]	[5.3733%, 8.0909%]
interval T2	Average	16.8712%	6.2602%
HFLRBC-S	CI	[15.9415%, 17.8165%]	[5.0485%, 7.6902%]

TABLE VIII
NUMBER OF DESIGNS OF THE LEAVE-TWO-RUNS-OUT EXPERIMENTS FOR WHICH THE CLASSIFICATION ERROR RATE OF ONE (NONADAPTIVE OR ADAPTIVE) MODE IS STRICTLY SMALLER THAN THE OTHER. NOTE THAT THE DESIGNS WHERE THE CLASSIFICATION ERROR RATES OF THE NONADAPTIVE AND ADAPTIVE MODES ARE TIED ARE NOT COUNTED

Classifier	Non-Adaptive	Adaptive
Bayesian	34	163
T1 Non-hierarchical	29	167
interval T2 Non-hierarchical	22	174
T1 HFLRBC-P	30	168
interval T2 HFLRBC-P	20	179
T1 HFLRBC-S	18	179
interval T2 HFLRBC-S	16	180

Note that for the leave-two-runs-out experiment the training prototypes were presented 400 epochs rather than 1000 epochs, as was done for the 10-fold cross validation experiments. This is because in the leave-two-runs-out experiment more prototypes were used for training than in the 10-fold cross validation experiments. On the one hand, more training data mean a longer training process; on the other hand, we did not want the classifiers to be over-fitted to the training data.

The results of this experiment are summarized in Tables VII–X. Observe the following.

- Comparing the the nonadaptive and adaptive working modes (see Tables VII and VIII) for each classifier, the average classification error rate of the adaptive mode is much smaller than that of the nonadaptive mode. Even the upper-bound of the 95% confidence interval of the adaptive mode is smaller than the lower-bound of the 95% confidence interval of the nonadaptive mode. The number of designs for which the classification error rate of the adaptive mode is smaller than that of the nonadaptive mode is larger than the number of designs for which the classification error rate of the nonadaptive mode is smaller than that of the adaptive mode. This indicates that the performance of each classifier in the adaptive mode is much better than in the nonadaptive mode.
- Comparing the Bayesian classifier and the FLRBCs, for both the non-adaptive and adaptive working modes (see Tables VII and IX), the average classification error rates of the Bayesian classifier are greater than those of each of the FLRBCs. Even the lower-bounds of the 95% confidence intervals of the Bayesian classifier are greater than the upper-bounds of the 95% confidence intervals of each

TABLE IX
NUMBER OF DESIGNS OF THE LEAVE-TWO-RUNS-OUT EXPERIMENTS FOR WHICH THE CLASSIFICATION ERROR RATE OF A PARTICULAR CLASSIFIER IS NO LARGER THAN THE MINIMUM CLASSIFICATION ERROR RATE FROM ALL SEVEN CLASSIFIERS BEING COMPARED

Classifier	Non-Adaptive	Adaptive
Bayesian	2	25
T1 Non-hierarchical	40	74
interval T2 Non-hierarchical	53	83
T1 HFLRBC-P	12	63
interval T2 HFLRBC-P	46	96
T1 HFLRBC-S	48	83
interval T2 HFLRBC-S	54	91

TABLE X
NUMBER OF DESIGNS OF THE LEAVE-TWO-RUNS-OUT EXPERIMENTS FOR WHICH THE CLASSIFICATION ERROR RATE OF ONE (T1 OR INTERVAL T2) IMPLEMENTATION IS STRICTLY SMALLER THAN THE OTHER. NOTE THAT THE DESIGNS WHERE THE CLASSIFICATION ERROR RATES OF THE COMPETING T1 AND INTERVAL T2 IMPLEMENTATIONS ARE TIED ARE NOT COUNTED

Classifier	Non-Adaptive	Adaptive
T1 Non-hierarchical	65	41
interval T2 Non-hierarchical	97	68
T1 HFLRBC-P	49	48
interval T2 HFLRBC-P	142	87
T1 HFLRBC-S	84	53
interval T2 HFLRBC-S	91	59

of the FLRBCs. The numbers of designs that the classification error rates of the Bayesian classifier are the smallest among all the seven classifiers are rather small (two for the nonadaptive mode, and 25 for the adaptive mode, out of 200 designs). This indicates that the Bayesian classifier is worse than the FLRBCs.

- Comparing the T1 and interval T2 implementations of FLRBCs (see Tables VII and X), for each FLRBC architecture and for both the nonadaptive and adaptive working modes, the average classification error rates of the interval T2 implementation are smaller than those of the competing T1 implementation, and the numbers of designs for which the classification error rates of the interval T2 implementation are smaller than those of the competing T1 implementation are larger than the numbers of designs for which the classification error rates of the T1 implementation are smaller than those of the competing interval T2 implementation. This indicates that the interval T2 implementation of each FLRBC architecture performs somewhat better than its competing T1 implementation.
- For the nonadaptive working mode, in terms of the average classification error rate and the number of designs with the smallest classification error rate (see Tables VII and X), the interval T2 nonhierarchical and HFLRBC-S architectures perform the best.
- For the adaptive working mode, in terms of the average classification error rate (see Table VII), the T1 and interval T2 HFLRBC-S perform the best; whereas, in terms of the number of designs with the smallest classification error rate (see Table IX), the interval T2 HFLRBC-P and the interval T2 HFLRBC-S perform the best.
- In terms of the average classification error rate (see Table VII), all FLRBCs are able to achieve higher than the 80% acceptable classification accuracy even in their nonadaptive working modes.

VII. CONCLUSION

When designing FLRBCs for multicategory classification of ground vehicles using their acoustic features, we established one fuzzy logic rule for each kind of vehicle, modeled the variations of the features within each kind of vehicle by using fuzzy sets, and optimized the parameters by using the training samples. We applied this method of design to three FLRBC architectures (nonhierarchical, hierarchical in parallel and hierarchical in series), and described their structures, operations, initializations and optimizations. To evaluate these FLRBCs, we also constructed a Bayesian classifier, and compared the performance of all classifiers through 10-fold cross validation and leave-two-runs-out experiments. Our experimental results showed that: 1) for each classifier the performance in the adaptive mode that uses simple majority voting was much better than in the non-adaptive mode; 2) all FLRBCs performed substantially better than the Bayesian classifier; 3) interval T2 FLRBCs performed better than their competing T1 FLRBCs, although sometimes not by much; 4) the interval T2 non-hierarchical and HFLRBC-S architectures performed the best; and 5) all FLRBCs achieved higher than the 80% acceptable classification accuracy.

Surprising to us is the fact that, although we have observed lots of uncertainties and time-variations in the acoustic features, we have not observed substantial improvements of interval T2 FLRBCs over T1 FLRBCs. In an extended work in which the classification was performed by using the acoustic features of *multiple terrains* that contain more uncertainties, we have observed a more significant improvement of the T2 FLRBC over the T1 FLRBC (mean of the classification error rates over more than 800 experiments: 9.13% versus 12.8%).

These observations raise numerous important research questions, including but not limited to, the following.

- In which sense are interval T2 FLRBCs considered to outperform T1 FLRBCs, e.g., in terms of the classification error rate, or generalizability, or robustness?
- How much uncertainty must be present in a problem so that it is worthy of trading the complexity of interval T2 fuzzy logic systems for better performance (than T1 fuzzy logic systems)?
- How should the uncertainties contained in the feature measurements be quantified?

To the best of our knowledge, these questions remain open and, because they are beyond the scope of this paper, we do not provide discussions about them.

Finally, we believe that the FLRBC designs described in this paper provide the following general methodology that can also be applied to other applications such as face recognition, optical character recognition, speaker recognition, etc.

1) Establish one fuzzy logic rule for each naturally distinguishable sub-category.
2) Model variations of features within each sub-category by using fuzzy sets.
3) Optimize the parameters of the MFs by using training data.

Step 1) is novel because it immediately establishes the number of rules for a FLRBC and consequently its architecture. This may provide the FLRBC with an advantage over, e.g., a neural

network classifier, because knowledge about each naturally distinguishable subcategory (e.g., our nine kinds of vehicles) does not help to establish the number of layers, or interconnectivity of all the neurons. How to accomplish this for neural networks is an open question.

APPENDIX A
PARTIAL DERIVATIVES FOR A T1 FLRBS

The partial derivatives that are used to update the parameters of a T1 FLRBS include $\partial y/\partial g^j$, $\partial y/\partial m_k^j$, $\partial y/\partial \sigma_k^j$ and $\partial y/\partial \sigma_k$ ($k = 1, \ldots, 11$, and $j = 1, \ldots, M$), where g^j is the consequent parameter, m_k^j and σ_k^j are the antecedent parameters, and σ_k is the input parameter.

Observe from (4) that $y(\mathbf{x})$ is a direct function of the consequent parameters, g^j, and firing degrees $f^j(\mathbf{x})$; and, from (3) that $f^j(\mathbf{x})$ is a direct function of the antecedent parameters, m_k^j and σ_k^j, and input parameters, σ_k. Consequently, by using the chain rule, it is straightforward to formulate the derivatives of y with respect to these consequent, antecedent, and input parameters as

$$\frac{\partial y}{\partial g^j} = \frac{f^j(\mathbf{x})}{\sum_{m=1}^{M} f^m(\mathbf{x})} \tag{A.1}$$

$$\frac{\partial y}{\partial f^j} = \frac{g^j - y(\mathbf{x})}{\sum_{m=1}^{M} f^m(\mathbf{x})} \tag{A.2}$$

$$\frac{\partial y}{\partial m_k^j} = \frac{\partial y}{\partial f^j}\frac{\partial f^j}{\partial m_k^j} = \frac{\partial y}{\partial f^j} f^j(\mathbf{x}) \frac{x_k - m_k^j}{\sigma_k^2 + \left(\sigma_k^j\right)^2} \tag{A.3}$$

$$\frac{\partial y}{\partial \sigma_k^j} = \frac{\partial y}{\partial f^j}\frac{\partial f^j}{\partial \sigma_k^j} = \frac{\partial y}{\partial f^j} f^j(\mathbf{x}) \frac{\sigma_k^j \left(x_k - m_k^j\right)^2}{\left[\sigma_k^2 + \left(\sigma_k^j\right)^2\right]^2} \tag{A.4}$$

$$\frac{\partial y}{\partial \sigma_k} = \frac{\partial y}{\partial f^j}\frac{\partial f^j}{\partial \sigma_k} = \frac{\partial y}{\partial f^j} f^j(\mathbf{x}) \frac{\sigma_k \left(x_k - m_k^j\right)^2}{\left[\sigma_k^2 + \left(\sigma_k^j\right)^2\right]^2}. \tag{A.5}$$

APPENDIX B
TYPE-REDUCTION

Type-reduction is an extension of a T1 defuzzification procedure, and represents a mapping of a T2 fuzzy set into a T1 fuzzy set. The type-reduced set of an interval T2 fuzzy set is an interval T1 fuzzy set, which is completely characterized by the left and right end-points of its support [11], [21].

The type-reduction method we used for the interval T2 FLRBCs is a special case of the center-of-sets type-reduction, which is expressed as:

$$\begin{aligned} Y_{TR}(\mathbf{x}) &= [y_l(\mathbf{x}), y_r(\mathbf{x})] \\ &= \int_{f^1 \in [\underline{f}^1(\mathbf{x}), \bar{f}^1(\mathbf{x})]} \cdots \\ &\quad \times \int_{f^M \in [\underline{f}^M(\mathbf{x}), \bar{f}^M(\mathbf{x})]} 1 \Big/ \frac{\sum_{j=1}^{M} f^j g^j}{\sum_{j=1}^{M} f^j} \end{aligned} \tag{B.6}$$

where $Y_{\mathrm{TR}}(\mathbf{x})$ represents the type-reduced set, and $y_l(\mathbf{x})$ and $y_r(\mathbf{x})$ are the left and right end-points of $Y_{\mathrm{TR}}(\mathbf{x})$.

The left end-point, $y_l(\mathbf{x})$, is determined by using the Karnik–Mendel iterative procedure [21], which we repeat here for the convenience of the readers.

1) Reorder the rules so that $g^1 \le g^2 \le \cdots \le g^M$.
2) Initialize f^j by setting $f^j = [\underline{f}^j(\mathbf{x}) + \bar{f}^j(\mathbf{x})]/2$ for $j = 1, \ldots, M$, and compute
$$y' = \sum_{j=1}^{M} f^j g^j \Big/ \sum_{j=1}^{M} f^j.$$
3) Find $L (1 \le L \le M-1)$ so that $g^L \le y' \le g^{L+1}$.
4) Set $f^j = \bar{f}^j(\mathbf{x})$ for $1 \le j \le L$, and $f^j = \underline{f}^j(\mathbf{x})$ for $L+1 \le j \le M$, and then compute
$$y'' = \sum_{j=1}^{M} f^j g^j \Big/ \sum_{j=1}^{M} f^j.$$
5) Check if $y'' = y'$. If yes, then $y_l(\mathbf{x}) = y''$ and stop. Otherwise, go to Step 6).
6) Set y' equal to y'', and go to Step 3).

It is easy to show [21], [11] that $y_l(\mathbf{x})$ is obtained from this algorithm when $f^j = \bar{f}^j(\mathbf{x})$ for those rules for which $g^j < y_l(\mathbf{x})$, and $f^j = \underline{f}^j(\mathbf{x})$ for those rules for which $g^j > y_l(\mathbf{x})$. We can therefore define an *indicator function* for each rule, $\delta_l^j (j = 1, \ldots, M)$, that establishes whether a rule's upper or lower firing degree is used during the computation of $y_l(\mathbf{x})$, i.e.,

$$\delta_l^j \equiv \begin{cases} 1, & \text{if } g^j \le y_l(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}. \tag{B.7}$$

Consequently, $y_l(\mathbf{x})$ can be expressed as

$$y_l(\mathbf{x}) = \frac{\sum_{j=1}^{M} g^j \left\{\delta_l^j \bar{f}^j(\mathbf{x}) + \left(1 - \delta_l^j\right) \underline{f}^j(\mathbf{x})\right\}}{\sum_{j=1}^{M} \left\{\delta_l^j \bar{f}^j(\mathbf{x}) + \left(1 - \delta_l^j\right) \underline{f}^j(\mathbf{x})\right\}}. \tag{B.8}$$

The right end-point, $y_r(\mathbf{x})$, is determined by using a similar Karnik–Mendel iterative procedure except for the following three changes.

1) In Step 3), find $R (1 \le R \le M-1)$ so that $g^R \le y' \le g^{R+1}$.
2) In Step 4), set $f^j = \underline{f}^j(\mathbf{x})$ for $1 \le j \le R$, and $f^j = \bar{f}^j(\mathbf{x})$ for $R+1 \le j \le M$, and then compute $y'' = \sum_{j=1}^{M} f^j g^j / \sum_{j=1}^{M} f^j$.
3) In Step 5), check if $y'' = y'$. If yes, then $y_r(\mathbf{x}) = y''$ and stop. Otherwise, go to Step 5).

$y_r(\mathbf{x})$ is obtained from this algorithm when $f^j = \bar{f}^j(\mathbf{x})$ for those rules for which $g^j > y_r(\mathbf{x})$, and $f^j = \underline{f}^j(\mathbf{x})$ for those rules for which $g^j < y_r(\mathbf{x})$. We can therefore define another indicator function for each rule, $\delta_r^j (j = 1, \ldots, M)$ that establishes whether a rule's upper or lower firing degree is used during the computation of $y_r(\mathbf{x})$, i.e.,

$$\delta_r^j \equiv \begin{cases} 1, & \text{if } g^j \ge y_r(\mathbf{x}) \\ 0, & \text{otherwise} \end{cases}. \tag{B.9}$$

Consequently, $y_r(\mathbf{x})$ can be expressed as

$$y_r(\mathbf{x}) = \frac{\sum_{j=1}^{M} g^j \left\{\delta_r^j \bar{f}^j(\mathbf{x}) + \left(1-\delta_r^j\right) \underline{f}^j(\mathbf{x})\right\}}{\sum_{j=1}^{M} \left\{\delta_r^j \bar{f}^j(\mathbf{x}) + \left(1-\delta_r^j\right) \underline{f}^j(\mathbf{x})\right\}}. \quad \text{(B.10)}$$

APPENDIX C
PARTIAL DERIVATIVES FOR AN INTERVAL T2 FLRBS

The partial derivatives that are used to update the parameters of an interval T2 FLRBS include $\partial y/\partial g^j, \partial y/\partial m_{1,k}^j, \partial y/\partial m_{2,k}^j, \partial y/\partial \sigma_{1,k}^j, \partial y/\partial \sigma_{2,k}^j, \partial y/\partial \sigma_{1,k}$ and $\partial y/\partial \sigma_{2,k}$, where g^j is the consequent parameter, $\{m_{1,k}^j, m_{2,k}^j\}$ and $\{\sigma_{1,k}^j, \sigma_{2,k}^j\}$ are the antecedent parameters, and $\{\sigma_{1,k}, \sigma_{2,k}\}$ are the input parameters.

Observe from (16) that $y(\mathbf{x})$ is a direct function of the two end-points of the type-reduced set, $y_l(\mathbf{x})$ and $y_r(\mathbf{x})$; from (14) and (15), that both $y_l(\mathbf{x})$ and $y_r(\mathbf{x})$ are direct functions of the consequent parameters, g^j, and lower and upper firing degrees $\underline{f}^j(\mathbf{x})$ and $\bar{f}^j(\mathbf{x})$; and, from (12), (13) and Table IV, that both $\underline{f}^j(\mathbf{x})$ and $\bar{f}^j(\mathbf{x})$ are direct functions of the antecedent parameters, $\{m_{1,k}^j, m_{2,k}^j\}$ and $\{\sigma_{1,k}^j, \sigma_{2,k}^j\}$, and the input parameters, $\{\sigma_{1,k}, \sigma_{2,k}\}$. By using the chain rule, we formulate the derivatives of y with respect to these consequent, antecedent and input parameters, as

$$\frac{\partial y}{\partial g^j} = \frac{\partial y}{\partial y_l}\frac{\partial y_l}{\partial g^j} + \frac{\partial y}{\partial y_r}\frac{\partial y_r}{\partial g^j} \quad \text{(C.11)}$$

$$\frac{\partial y}{\partial \theta_k^j} = \frac{\partial y}{\partial y_l}\left(\frac{\partial y_l}{\partial \underline{f}^j}\frac{\partial \underline{f}^j}{\partial \theta_k^j} + \frac{\partial y_l}{\partial \bar{f}^j}\frac{\partial \bar{f}^j}{\partial \theta_k^j}\right) + \frac{\partial y}{\partial y_r}\left(\frac{\partial y_r}{\partial \underline{f}^j}\frac{\partial \underline{f}^j}{\partial \theta_k^j} + \frac{\partial y_r}{\partial \bar{f}^j}\frac{\partial \bar{f}^j}{\partial \theta_k^j}\right) \quad \text{(C.12)}$$

$$\frac{\partial y}{\partial \theta_k} = \frac{\partial y}{\partial y_l}\sum_{j=1}^{M}\left(\frac{\partial y_l}{\partial \underline{f}^j}\frac{\partial \underline{f}^j}{\partial \theta_k} + \frac{\partial y_l}{\partial \bar{f}^j}\frac{\partial \bar{f}^j}{\partial \theta^j}\right) + \frac{\partial y}{\partial y_r}\sum_{j=1}^{M}\left(\frac{\partial y_r}{\partial \underline{f}^j}\frac{\partial \underline{f}^j}{\partial \theta_k} + \frac{\partial y_r}{\partial \bar{f}^j}\frac{\partial \bar{f}^j}{\partial \theta^j}\right) \quad \text{(C.13)}$$

where θ_k^j represents the antecedent parameters $\{m_{1,k}^j, m_{2,k}^j, \sigma_{1,k}^j, \sigma_{2,k}^j\}$ and θ_k represents the input parameters $\{\sigma_{1,k}, \sigma_{2,k}\}$. Based on (14)–(16), we have

$$\frac{\partial y}{\partial y_l} = \frac{\partial y}{\partial y_r} = \frac{1}{2} \quad \text{(C.14)}$$

$$\frac{\partial y_l}{\partial g^j} = \frac{\delta_l^j \bar{f}^j(\mathbf{x}) + \left(1-\delta_l^j\right)\underline{f}^j(\mathbf{x})}{\sum_{m=1}^{M}\left\{\delta_l^m \bar{f}^m(\mathbf{x}) + (1-\delta_l^m)\underline{f}^m(\mathbf{x})\right\}} \quad \text{(C.15)}$$

$$\frac{\partial y_r}{\partial g^j} = \frac{\delta_r^j \bar{f}^j(\mathbf{x}) + \left(1-\delta_r^j\right)\underline{f}^j(\mathbf{x})}{\sum_{m=1}^{M}\left\{\delta_r^m \bar{f}^m(\mathbf{x}) + (1-\delta_r^m)\underline{f}^m(\mathbf{x})\right\}} \quad \text{(C.16)}$$

$$\frac{\partial y_l}{\partial \underline{f}^j} = \frac{\left(1-\delta_l^j\right)\left[g^j - y_l(\mathbf{x})\right]}{\sum_{m=1}^{M}\left\{\delta_l^m \bar{f}^m(\mathbf{x}) + (1-\delta_l^m)\underline{f}^m(\mathbf{x})\right\}} \quad \text{(C.17)}$$

$$\frac{\partial y_r}{\partial \underline{f}^j} = \frac{\left(1-\delta_r^j\right)\left[g^j - y_r(\mathbf{x})\right]}{\sum_{m=1}^{M}\left\{\delta_r^m \bar{f}^m(\mathbf{x}) + (1-\delta_r^m)\underline{f}^m(\mathbf{x})\right\}} \quad \text{(C.18)}$$

$$\frac{\partial y_l}{\partial \bar{f}^j} = \frac{\delta_l^j\left[g^j - y_l(\mathbf{x})\right]}{\sum_{m=1}^{M}\left\{\delta_l^m \bar{f}^m(\mathbf{x}) + (1-\delta_l^m)\underline{f}^m(\mathbf{x})\right\}} \quad \text{(C.19)}$$

$$\frac{\partial y_r}{\partial \bar{f}^j} = \frac{\delta_r^j\left[g^j - y_r(\mathbf{x})\right]}{\sum_{m=1}^{M}\left\{\delta_r^m \bar{f}^m(\mathbf{x}) + (1-\delta_r^m)\underline{f}^m(\mathbf{x})\right\}} \quad \text{(C.20)}$$

$$\frac{\partial \bar{f}^j}{\partial m_{1,k}^j} = \begin{cases} \bar{f}^j(\mathbf{x})\frac{x_k - m_{1,k}^j}{\sigma_{2,k}^2 + \left(\sigma_{2,k}^j\right)^2} & \text{if } x_k \le m_{1,k}^j \\ 0 & \text{otherwise} \end{cases}. \quad \text{(C.21)}$$

Based on Table IV

$$\frac{\partial \underline{f}^j}{\partial m_{1,k}^j} = \begin{cases} 0, \\ \quad \text{if } x_k \le \frac{m_{1,k}^j + m_{2,k}^j}{2} - \frac{\sigma_{1,k}^2\left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}^j(\mathbf{x})\frac{x_k - m_{1,k}^j}{\sigma_{1,k}^2 + \left(\sigma_{1,k}^j\right)^2}, \\ \quad \text{if } x_k \ge \frac{m_{1,k}^j + m_{2,k}^j}{2} + \frac{\sigma_{1,k}^2\left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}^j(\mathbf{x})\left[\frac{m_{2,k}^j - m_{1,k}^j}{4\left(\sigma_{1,k}^j\right)^2} + \frac{2x_k - m_{1,k}^j - m_{2,k}^j}{4\sigma_{1,k}^2}\right] \\ \quad \text{otherwise} \end{cases} \quad \text{(C.22)}$$

$$\frac{\partial \bar{f}^j}{\partial m_{2,k}^j} = \begin{cases} \bar{f}^j(\mathbf{x})\frac{x_k - m_{2,k}^j}{\sigma_{2,k}^2 + \left(\sigma_{2,k}^j\right)^2} & \text{if } x_k \ge m_{2,k}^j \\ 0 & \text{otherwise} \end{cases} \quad \text{(C.23)}$$

$$\frac{\partial \underline{f}^j}{\partial m_{2,k}^j} = \begin{cases} \underline{f}^j(\mathbf{x})\frac{x_k - m_{2,k}^j}{\sigma_{1,k}^2 + \left(\sigma_{1,k}^j\right)^2}, \\ \quad \text{if } x_k \le \frac{m_{1,k}^j + m_{2,k}^j}{2} - \frac{\sigma_{1,k}^2\left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ 0, \\ \quad \text{if } x_k \ge \frac{m_{1,k}^j + m_{2,k}^j}{2} + \frac{\sigma_{1,k}^2\left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}^j(\mathbf{x})\left[\frac{m_{1,k}^j - m_{2,k}^j}{4\left(\sigma_{1,k}^j\right)^2} + \frac{2x_k - m_{1,k}^j - m_{2,k}^j}{4\sigma_{1,k}^2}\right], \\ \text{otherwise} \end{cases} \quad \text{(C.24)}$$

$$\frac{\partial \bar{f}^j}{\partial \sigma_{2,k}^j} = \begin{cases} \bar{f}^j(\mathbf{x})\frac{\sigma_{2,k}^j\left(x_k - m_{1,k}^j\right)^2}{\left[\sigma_{2,k}^2 + \left(\sigma_{2,k}^j\right)^2\right]^2} & \text{if } x_k \le m_{1,k}^j \\ \bar{f}^j(\mathbf{x})\frac{\sigma_{2,k}^j\left(x_k - m_{2,k}^j\right)^2}{\left[\sigma_{2,k}^2 + \left(\sigma_{2,k}^j\right)^2\right]^2} & \text{if } x_k \ge m_{2,k}^j \\ 0 & \text{otherwise} \end{cases} \quad \text{(C.25)}$$

$$\frac{\partial \bar{f}^j}{\partial \sigma_{2,k}} = \begin{cases} \bar{f}^j(\mathbf{x})\frac{\sigma_{2,k}\left(x_k - m_{1,k}^j\right)^2}{\left[\sigma_{2,k}^2 + \left(\sigma_{2,k}^j\right)^2\right]^2} & \text{if } x_k \le m_{1,k}^j \\ \bar{f}^j(\mathbf{x})\frac{\sigma_{2,k}\left(x_k - m_{2,k}^j\right)^2}{\left[\sigma_{2,k}^2 + \left(\sigma_{2,k}^j\right)^2\right]^2} & \text{if } x_k \ge m_{2,k}^j \\ 0 & \text{otherwise} \end{cases} \quad \text{(C.26)}$$

$$\frac{\partial \underline{f}^j}{\partial \sigma_{1,k}^j} = \begin{cases} \underline{f}^j(\mathbf{x}) \frac{\sigma_{1,k}^j \left(x_k - m_{2,k}^j\right)^2}{\left[\sigma_{1,k}^2 + \left(\sigma_{1,k}^j\right)^2\right]^2}, \\ \quad \text{if } x_k \le \frac{m_{1,k}^j + m_{2,k}^j}{2} - \frac{\sigma_{1,k}^2 \left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}^j(\mathbf{x}) \frac{\sigma_{1,k}^j \left(x_k - m_{1,k}^j\right)^2}{\left[\sigma_{1,k}^2 + \left(\sigma_{1,k}^j\right)^2\right]^2}, \\ \quad \text{if } x_k \ge \frac{m_{1,k}^j + m_{2,k}^j}{2} + \frac{\sigma_{1,k}^2 \left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}_1^j(\mathbf{x}) \frac{\left(m_{2,k}^j - m_{1,k}^j\right)^2}{4\left(\sigma_{1,k}^j\right)^3}, \\ \quad \text{otherwise} \end{cases} \tag{C.27}$$

$$\frac{\partial \underline{f}^j}{\partial \sigma_{1,k}} = \begin{cases} \underline{f}^j(\mathbf{x}) \frac{\sigma_{1,k} \left(x_k - m_{2,k}^j\right)^2}{\left[\sigma_{1,k}^2 + \left(\sigma_{1,k}^j\right)^2\right]^2}, \\ \quad \text{if } x_k \le \frac{m_{1,k}^j + m_{2,k}^j}{2} - \frac{\sigma_{1,k}^2 \left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}^j(\mathbf{x}) \frac{\sigma_{1,k} \left(x_k - m_{1,k}^j\right)^2}{\left[\sigma_{1,k}^2 + \left(\sigma_{1,k}^j\right)^2\right]^2}, \\ \quad \text{if } x_k \ge \frac{m_{1,k}^j + m_{2,k}^j}{2} + \frac{\sigma_{1,k}^2 \left(m_{2,k}^j - m_{1,k}^j\right)}{2\left(\sigma_{1,k}^j\right)^2}; \\ \underline{f}^j(\mathbf{x}) \frac{\left(2x_k - m_{1,k}^j - m_{2,k}^j\right)^2}{4\sigma_{1,k}^3}, \\ \quad \text{otherwise} \end{cases} \tag{C.28}$$

In all the above equations, $k = 1, \ldots, 11$ and $j = 1, \ldots, M$.

REFERENCES

[1] S. Abe, *Pattern Classification: Neuro-Fuzzy Methods and Their Comparison*. New York: Springer-Verlag, 2001.

[2] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.

[3] J. C. Bezdek, M. R. Pal, J. Keller, and R. Krishnapuram, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Norwell, MA: Kluwer, 1999.

[4] D. Chen and X. Cheng, "Majority vote based on weak classifiers," *Vol. 3 WCCC-ICSP 2000*, pp. 1560–1562, 2000.

[5] H. Choe, R. Karlsen, G. Gerhart, and T. Meitzler, "Wavelet based ground vehicle recognition using acoustic signals," in *Proc. SPIE*, 1996, vol. 2762, pp. 434–445.

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

[7] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 10, pp. 993–1001, Oct. 1990.

[8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning, Data Mining, Inference and Prediction*. New York: Springer-Verlag, 2001.

[9] F. Hoppner, F. Klawonn, R. Kruse, and T. Runkler, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. New York: Wiley, 1999.

[10] A. Kandel, *Fuzzy Mathematical Techniques With Applications*. Boston, MA: Addison-Wesley, 1986.

[11] N. Karnik and J. M. Mendel, *Centroid Type-2 Fuzzy Set. Inform. Sci.*, vol. 132, pp. 195–220, 2001.

[12] J. Kittler and F. M. Alkoot, "Sum versus vote fusion in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 110–115, Jan. 2003.

[13] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Recogn. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[14] G. J. Klir and T. A. Folger, *Fuzzy Sets, Uncertainty, and Information*. Upper Saddle River, NJ: Prentice-Hall, 1987.

[15] L. I. Kuncheva, *Fuzzy Classifier Design*. Heidelberg, Germany: Physica-Verlag, 2000.

[16] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ: Wiley, 2004.

[17] D. Lake, Efficient maximum likelihood estimate for multiple and coupled harmonics Army Research Laboratory, Tech. Rep. ARL-TR-2014, 1999.

[18] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 27, no. 5, pp. 553–568, Sep. 1997.

[19] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, Oct. 2000.

[20] L. Liu, "Ground vehicle acoustic signal processing based on biological hearing models," M.S. thesis, Inst. Syst. Research, Univ. Maryland, College Park, MD, 1999.

[21] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River, NJ: Prentice-Hall, 2001.

[22] P. S. Parmpero and A. C. P. L. F. de Carvalho, "Recognition of vehicles silhouette using combination of classifiers," in *Proc. IEEE Int. Conf. IEEE World Congr. Comput. Intell.*, 1998, pp. 1723–1726.

[23] J. A. Robertson and B. Weber, Artificial neural networks for acoustic target recognition Tech. Rep. Joint Rep. ARL and ITT Research, 1993.

[24] S. Sampan, "Neural fuzzy techniques in vehicle acoustic signal classification," Ph.D. dissertation, Dept. Elect. Eng., Virginia Poly. Inst. State Univ., Blacksburg, VA, 1997.

[25] T. Shin, D. Jue, D. Chandramohan, D. Choi, C. Seng, J. Yang, A. Bae, A. Lee, J. Lee, P. Lim, S. Kazadi, and J. M. Mendel, "Reduction of fuzzy systems through open product analysis of genetic algorithm-generated fuzzy rule sets," in *Proc. IEEE Int. Conf. Fuzzy Systems*, Budapest, Hungary, 2004, vol. 2, pp. 1043–1048.

[26] N. Srour and J. Robertson, Remote netted acoustic selection system: Final report Army Research Laboratory, Tech. Rep. ARL-TR-706, May 1995.

[27] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 1, pp. 7–31, Feb. 1993.

[28] M. van Erp, L. Vuurpijl, and L. Schomaker, "An overview and comparison of voting methods for pattern recognition," in *Proc. 8th Int. Workshop Frontiers Handwriting Recogn.*, 2002, pp. 195–200.

[29] L.-X. Wang, *A Course in Fuzzy Systems and Control*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1996.

[30] M. C. Wellman, N. Srour, and D. B. Hills, Acoustic feature extraction for a neural network classifier Army Research Laboratory, Tech. Rep. ARL-TR-1166, Jan. 1997.

[31] S. Wesolkowski and K. Hassanein, "A comparative study of combination schemes for an ensemble of digit recognition neural networks," in *Proc. IEEE Int. Conf. Comput. Cybern. Simul.*, 1997, pp. 3534–3539.

[32] D. H. Wolpert, *The Relationship Between pac, the Statistical Physics Framework, the Bayesian Framework, and the vc Framework*, D. H. Wolpert, Ed. Reading, MA: Addison-Wesley, 1995, sec. The Mathematics of Generalization, pp. 117–214.

[33] H. Wu and J. M. Mendel, "Data analysis and feature extraction for ground vehicle identification," in *Proc. Meeting MSS Specialty Group on Battlefield Acoustic and Seismic Sensing, Magnetic and Electric Field Sensors*, Laurel, MD, 2001, Appl. Phys. Lab., Johns Hopkins Univ..

[34] H. Wu and J. M. Mendel, Binary classification of ground vehicles based on the acoustic data using fuzzy logic rule-based classifiers Signal and Image Process. Inst., Univ. Southern Calif., Los Angeles, CA, Tech. Rep. 356, 2002.

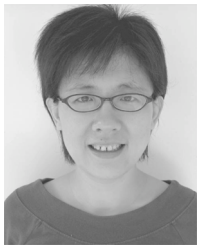
[35] H. Wu and J. M. Mendel, "Classification of ground vehicles from acoustic data using fuzzy logic rule-based classifiers: Early results," in *Proc. SPIE*, Orlando, FL, 2002, vol. 4743, pp. 62–72.

[36] H. Wu and J. M. Mendel, "Classifier design for binary classifications of ground vehicles," in *Proc. SPIE*, Orlando, FL, 2003, vol. 5090, pp. 122–133.

[37] H. Wu and J. M. Mendel, Multi-category classification of ground vehicles based on the acoustic data using fuzzy logic rule-based classifiers Signal Image Process. Inst., Univ. Southern Calif., Los Angeles, CA, Tech. Rep. 360, 2003.

[38] H. Wu and J. M. Mendel, "Quantitative analysis of spatio-temporal decision fusion based on the majority voting technique," in *Proc. SPIE Int. Symp. Defense and Security*, Orlando, FL, 2004, vol. 5434, pp. 13–24.

[39] A. M. Zoubir and B. Boashash, "The bootstrap and its aplication in signal processing," *IEEE Signal Process. Mag.*, vol. 15, no. 1, pp. 56–76, Jan. 1998.



Hongwei Wu (S'02–M'04) received the Ph.D. degree in electrical engineering and M.S.E.E. degree from University of Southern California in 2004 and 2002, respectively, M. Eng. in pattern recognition and intelligent systems, and B. Eng. in automatic control from Tsinghua University, China in 1999 and 1997, respectively.

She is currently affiliated with the Computational Biology Institute of Oak Ridge National Laboratory and Computational Systems Biology Laboratory of the University of Georgia, Athens, as a Postdoctoral Research Associate. Her research interests include bioinformatics and computational biology (particularly in computational reconstruction and modeling of biological networks, predictions of functional modules, and identification of orthologous gene groups for microbial genomes), computational intelligence, signal processing, and pattern recognition.

Dr. Wu is a member of the IEEE Computational Intelligence Society.

Jerry M. Mendel (S'59–M'61–SM'72–F'78–LF'04) received the Ph.D. in electrical engineering from the Polytechnic Institute of Brooklyn, NY.

Currently he is Professor of Electrical Engineering at the University of Southern California, Los Angeles, where he has been since 1974. He has published over 470 technical papers and is author and/or editor of eight books, including *Uncertain Rule-based Fuzzy Logic Systems: Introduction and New Directions* (Prentice-Hall, 2001). His present research interests include: type-2 fuzzy logic systems and their applications to a wide range of problems, including target classification, smart oil field technology, and computing with words.

Dr. Mendel is a Distinguished Member of the IEEE Control Systems Society. He was President of the IEEE Control Systems Society in 1986, and is presently Chairman of the Fuzzy Systems Technical Committee and a member of the Administrative Committee of the IEEE Computational Intelligence Society. Among his awards are the 1983 Best Transactions Paper Award of the IEEE Geoscience and Remote Sensing Society, the 1992 Signal Processing Society Paper Award, the 2002 TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award, a 1984 IEEE Centennial Medal, and an IEEE Third Millenium Medal.