A Recurrent Neural Fuzzy Network for Word Boundary Detection in Variable Noise-Level Environments

Gin-Der Wu and Chin-Teng Lin, *Senior Member, IEEE*

***Abstract*—This paper discusses the problem of automatic word boundary detection in the presence of variable-level background noise. Commonly used robust word boundary detection algorithms always assume that the background noise level is fixed. In fact, the background noise level may vary during the procedure of recording. This is the major reason that most robust word boundary detection algorithms cannot work well in the condition of variable background noise level. In order to solve this problem, we first propose a *refined time–frequency* (RTF) parameter for extracting both the time and frequency features of noisy speech signals. The RTF parameter extends the (time–frequency) TF parameter proposed by Junqua *et al.* from single band to multiband spectrum analysis, where the frequency bands help to make the distinction between speech signal and noise clear. The RTF parameter can extract useful frequency information. Based on this RTF parameter, we further propose a new word boundary detection algorithm by using a recurrent self-organizing neural fuzzy inference network (RSONFIN). Since RSONFIN can process the temporal relations, the proposed RTF-based RSONFIN algorithm can find the variation of the background noise level and detect correct word boundaries in the condition of variable background noise level. As compared to normal neural networks, the RSONFIN can always find itself an economic network size with high-learning speed. Due to the self-learning ability of RSONFIN, this RTF-based RSONFIN algorithm avoids the need for empirically determining ambiguous decision rules in normal word boundary detection algorithms. Experimental results show that this new algorithm achieves higher recognition rate than the TF-based algorithm which has been shown to outperform several commonly used word boundary detection algorithms by about 12% in variable background noise level condition. It also reduces the recognition error rate due to endpoint detection to about 23%, compared to an average of 47% obtained by the TF-based algorithm in the same condition.**

***Index Terms*—Cepstrum, linear prediction coefficient (LPC), mel-scale filter bank, recurrent network, space partition, time–frequency (TF).**

I. Introduction

An important problem in speech processing is to detect the presence of speech in noisy environments. A major source of errors in isolated-word automatic speech recognition systems is the inaccurate detection of the beginning and ending boundaries. In many applications, the problem is further complicated by nonstationary backgrounds where there may exist concurrent noises due to movements of desks, door slams, etc. These background noises can be broadly classified into three classes:

1) impulse noise;
2) fixed-level noise;
3) variable-level noise.

In order to solve this problem, many researchers proposed robust word boundary detection algorithms in the presence of noise. However, they focused only on the impulse noise and fixed-level background noise.

Among the three classes of background noises, the impulse noise can be solved by the parameter of time duration. The problem of fixed-level background noise was first attacked by commonly used robust word boundary detection algorithms [1]–[4]. These algorithms usually use energy (in time domain), zero crossing rate, and time duration to find the boundary between the word signal and background noise. However, it has been found that the energy and zero-crossing rate are not sufficient to get reliable word boundaries in noisy environments, even if more complex decision strategies are used [5]. To date, several other parameters were proposed such as linear prediction coefficient (LPC), linear prediction error energy [6], [7] and pitch information [8]. Although the LPCs are quite successful in modeling vowels [9], they are not particularly suitable for nasal sounds, fricatives, etc. The reliability of the LPC parameter depends on the noisy environments. The pitch information can help to detect the word boundary, but it is not easy to extract the pitch period correctly in noisy environments.

Four endpoint detection algorithms were compared in [5]: an energy-based algorithm with automatic threshold adjustment [3], [4], use of pitch information [8], a noise adaptive algorithm, and a voiced activation algorithm. These four algorithms are strongly dependent on the noise condition. The reliability of the parameters used by the four algorithms also depends on the noise condition. In the connection, Junqua *et al.* [5] proposed the time–frequency (TF) parameter. They used the frequency energy in the fixed frequency band 250–3500 Hz to enhance the time–energy information. The TF parameter is the result obtained after smoothing the sum of the time energy and frequency energy. The frequency energy helps us to make the distinction between speech and noise. Based on the TF parameter, a robust algorithm was proposed in [5] to get more precise word boundaries in noisy environments. This TF-based robust algorithm in-

Manuscript received May 10, 1999; revised February 29, 2000 and August 21, 2000. This work was supported by the Lee and MTI Center for Networking Research and the National Science Council, R.O.C., under Grant NSC 89-2218-E-009-042. This paper was recommended by Associate Editor T. Sudkamp.

The authors are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: ctlin@fnn.cn.nctu.edu.tw).

Publisher Item Identifier S 1083-4419(01)00090-5.

1083–4419/01$10.00 © 2001 IEEE

cludes noise classification, a refinement procedure, and some preset thresholds.

However, the TF-based robust algorithm in [5] needs to empirically determine thresholds and ambiguous rules which are not easily determined by humans. Some researchers used the neural network's learning ability to solve this problem. In [6], [7], and [10], multilayer neural networks are used to classify the speech signal into voiced, unvoiced, and silence segments. In the neural network approach, the decision rules are in the form of input–output (I/O) layer mappings and can be learned by the training procedure (supervised learning). However, the proper structure of the network (including numbers of hidden layers and nodes) is not easy to decide.

Although the aforementioned TF-based algorithm outperforms several commonly used algorithms for word boundary detection in the presence of noise, it could work well only for the impulse noise and fixed-level background noise. For variable-level background noise, this TF-based algorithm usually results in inaccurate detection of the beginning or ending boundaries in the recording interval. There was little research about specific algorithms for processing the variable-level background noise. The reason may be that most laboratory systems used reasonably *fixed* background noise level in a given recording interval. The desired spoken word is present in the recording interval. The existing robust algorithms usually set thresholds from the first few frames of the recording interval. Then the algorithms used these preset thresholds to determine the word boundary of the speech signal. These thresholds are fixed during the recording interval. In the real world, the background noise level is not always fixed and may gradually vary over the recording interval. It is not reasonable to make these preset thresholds fixed over the recording interval. If the variation of background noise level is large, these fixed preset thresholds will result in incorrect location of word boundaries.

The main aim of this paper is to develop a new robust word boundary detection algorithm to attack the problem in variable-level background noise condition. To develop a more robust word boundary detection algorithm and avoid the problems of the above approaches, this paper first proposes a modified TF parameter and then uses a recurrent neural fuzzy network to detect word boundaries based on this parameter. In the TF parameter proposed by Junqua *et al.* [5], the frequency information is extracted on a single frequency band (250–3500 Hz). Since the frequency energy, i.e., magnitudes of the spectrum of different speech signals focus on different frequency bands, more accurate frequency information can be obtained by considering multiband analysis of noisy speech signals. With this motivation, we propose a new robust parameter, called the *refined time–frequency* (RTF) parameter, for word boundary detection in noisy environments. Like the TF parameter, the RTF parameter represents both the time and frequency features of noisy speech signals. However, the RTF parameter extends the TF parameter from single-band to multiband spectrum analysis based on the mel-scale frequency bank (20 bands). The 20 frequency bands are spaced on a nonlinear frequency scale (mel scale). A procedure is proposed such that the RTF parameter can extract more informative frequency energy than the single-band approach to compensate the time–energy information by *adaptively* choosing proper frequency bands. The RTF parameter is the result obtained after smoothing the sum of the time energy and frequency energy. It makes the word signal more obvious than the TF parameter that uses a single frequency band.

Based on the RTF parameter, we further propose a new word boundary detection algorithm by using a recurrent self-organizing neural fuzzy inference network (RSONFIN) that we proposed in [11]. Since this RSONFIN can process the temporal relations automatically and implicitly, the proposed RTF-based RSONFIN algorithm can find the variation of the background noise level and detect correct word boundaries in the condition of variable background noise level. The temporal relations embedded in the network are built by adding some feedback connections representing the memory elements to a feedforward *neural fuzzy* network.

Due to the self-learning ability of RSONFIN, the proposed RTF-based RSONFIN algorithm avoids the need of empirically determining ambiguous decision rules in normal word boundary detection algorithms. The RSONFIN can always find itself an economic network size with high learning speed, and so it avoids the need for empirically determining the number of hidden layers and nodes in normal neural networks. Also, since the RSONFIN houses the human-like IF–THEN rules in its network structure, expert knowledge can be put into the network as a priori knowledge, which can usually increase its learning speed and detection accuracy [12], [13]. This new algorithm has been tested over a variety of noise conditions and has been found to perform well not only in a variable background noise level condition but also in a fixed background noise level condition. Our results also showed that the RSONFIN's performance is not significantly affected by the size of the training set.

This paper is organized as follows. The RTF parameter is derived in Section II. The structure and function of the RSONFIN are briefly introduced in Section III. In Section IV, the RTF-based RSONFIN word boundary detection algorithm is proposed. The performance evaluation and comparisons of the proposed algorithm using RSONFIN are performed extensively in Section V. Finally, the conclusions of our work are summarized in Section VI.

II. RTF PARAMETER

Accurate location of the endpoint of an isolated word is important for reliable and robust word recognition. In general, the word boundary is susceptible to noise corruption because the additive noise obscures the distinction between the word signal and noise. The general solution is to compensate the strength of the word signal in noisy environments. It has been found that the frequency energy of a noisy speech signal can enhance the normally used time energy to make the distinction between word signal and background noise more obvious. In [5], Junqua *et al.* extracted the frequency energy of the signal on a single frequency band (250–3500 Hz) to form the TF parameter. In this section, we generalize the single-band analysis of the TF parameter to multiband analysis based on the mel-scale frequency bank and propose a new RTF parameter. The RTF parameter is obtained by smoothing the sum of the time energy and frequency energy, where the frequency energy is contributed by several adaptively chosen frequency bands.

A. Auditory-Based Mel-Scale Filter Bank

Loosely speaking, it has been found that the perception of a particular frequency f by the auditory system is influenced by the energy in a critical band of frequencies around f [14]. Hence, an auditory-based spectrum obtained by summing the energies in each critical band is a perceptually relevant characterization. It is also known that critical band filtering of the speech spectrum using parallel band-pass filters functionally represents an aspect of auditory processing. There is evidence from auditory psychophysics that the human ear perceives speech along a nonlinear scale in the frequency domain. One approach to simulating the subjective spectrum is to use a filter bank, spaced uniformly on a nonlinear, warped frequency scale, such as the mel scale. The relation between mel-scale frequency and frequency (Hz) is described by the following equation [15]:

$$\mathrm{mel} = 2595 \log(1 + f/700) \tag{1}$$

where mel is the mel-frequency scale and f is in Hz. The filter bank is then designed according to the mel scale as shown in Fig. 1(a), where the filters of 20 bands are approximated by simulating 20 triangular bandpass filters, $f(i,k)$ $(1 \le i \le 20, 0 \le k \le 63)$, over a frequency range of 0–4000 Hz. Hence, each filter band has a triangular bandpass frequency response, and the spacing as well as the bandwidth is determined by a constant mel frequency interval by(1). The value of the triangular function, $f(i,k)$ in the figure, also represents the weighting factor of the frequency energy at the kth point of the ith band.

With the mel-scale frequency bank given in Fig. 1(a), we can now calculate the energy of each frequency band for each time frame of a speech signal. Consider a given time–domain noisy speech signal, $x_{\mathrm{time}}(m,n)$, representing the magnitude of the nth point of the mth frame. We first find the spectrum $x_{\mathrm{freq}}(m,k)$ of this signal by Discrete Fourier Transform (128-point DFT)

$$x_{\mathrm{freq}}(m,k) = \sum_{n=0}^{N-1} x_{\mathrm{time}}(m,n) W_N^{kn}$$
$$0 \le k \le N-1, \quad 0 \le m \le M-1 \tag{2}$$
$$W_N = \exp(-j2\pi/N) \tag{3}$$

where

$x_{\mathrm{freq}}(m,k)$ magnitude of the kth point of the spectrum of the mth frame;

N 128 in our system;

M number of frames of the speech signal for analysis.

We then multiply the spectrum $x_{\mathrm{freq}}(m,k)$ by the weighting factors $f(i,k)$ on the mel-scale frequency bank and sum the products for all k to get the energy $x(m,i)$ of each frequency band i of the mth frame

$$x(m,i) = \sum_{k=0}^{N-1} |x_{\mathrm{freq}}(m,k)| f(i,k)$$
$$0 \le m \le M-1 \quad 1 \le i \le 20 \tag{4}$$

where

i filter band index;

k spectrum index;

m frame number;

M number of frames for analysis.

We found in our experiments that the energy $x(m,i)$ obtained in (4) usually had some undesired impulse noise and was covered by the energy of background noise. Hence, we further smooth it by using a three-point median filter to get $\hat{x}(m,i)$

$$\hat{x}(m,i) = \frac{x(m-1,i) + x(m,i) + x(m+1,i)}{3}. \tag{5}$$

Finally, the smoothed energy $\hat{x}(m,i)$ is normalized by removing the frequency energy of the beginning interval Noise_freq to get $X(m,i)$ where the energy of the beginning interval is estimated by averaging the frequency energy of the first five frames of the recording

$$\begin{aligned} X(m,i) &= \hat{x}(m,i) - \text{Noise\_freq} \\ &= \hat{x}(m,i) - \frac{\sum_{m=0}^{4} \hat{x}(m,i)}{5}. \end{aligned} \tag{6}$$

Since our goal is to extract the word signal information from the noisy speech waveform as much as possible so that we can use it to make the distinction between the word signal and background noise clear, we need a parameter to stand for the amount of word signal information of each band. It is understood that $X(m,i)$ in (6) cannot represent the frequency energy of exactly pure speech signal, since the part of the word signal covered by background noise is also removed in the normalization procedure. However, $X(m,i)$ is still a good indicator for the amount of speech information, since the more the word signal information is covered by the noise, the smaller the $X(m,i)$ is. In other words, the larger the $X(m,i)$ is, the more word signal information the ith band has. Hence, we use the smoothed and normalized energy of the ith band of the mth frame $X(m,i)$ to stand for the amount of the word signal information in band i of the mth frame. We can extract useful frequency information for word boundary detection by adopting the bands having large $X(m,i)$.

B. Effect of Additive Noise

Before we consider the adaptive choices of suitable bands for extracting useful frequency information from word signals, we first make some observations on the effect of additive noise on each frequency band. In Fig. 2(a), we try to add white noise (0 dB) to the clean speech signal to see the effects of adding white noise on each band. For illustration, the smoothed and normalized frequency energies of a speech signal $X(m,i)$ in (6) for 20 bands $(i = 1, 2, \ldots, 20)$ and 166 frames $(m = 0, 1, \ldots, 165)$ are shown in Figs. 2(b) and 2(c). We found that the energy of the first word signal $(m = 30, 41, \ldots, 50)$ mainly focuses on the fifth bands. Since the 8th–20th bands are seriously corrupted by the additive white noise, these bands have little information of word signal. In order to detect the boundaries of the first word signal correctly, we shall adopt the fifth band to make the distinction between the first word signal and noise clear. In addition, the energy of second word signal $(m = 70, 71, \ldots, 90)$ mainly focuses on the seventh band, and the energy of third word signal $(m = 120, 121, \ldots, 140)$ mainly focuses on ninth band. We

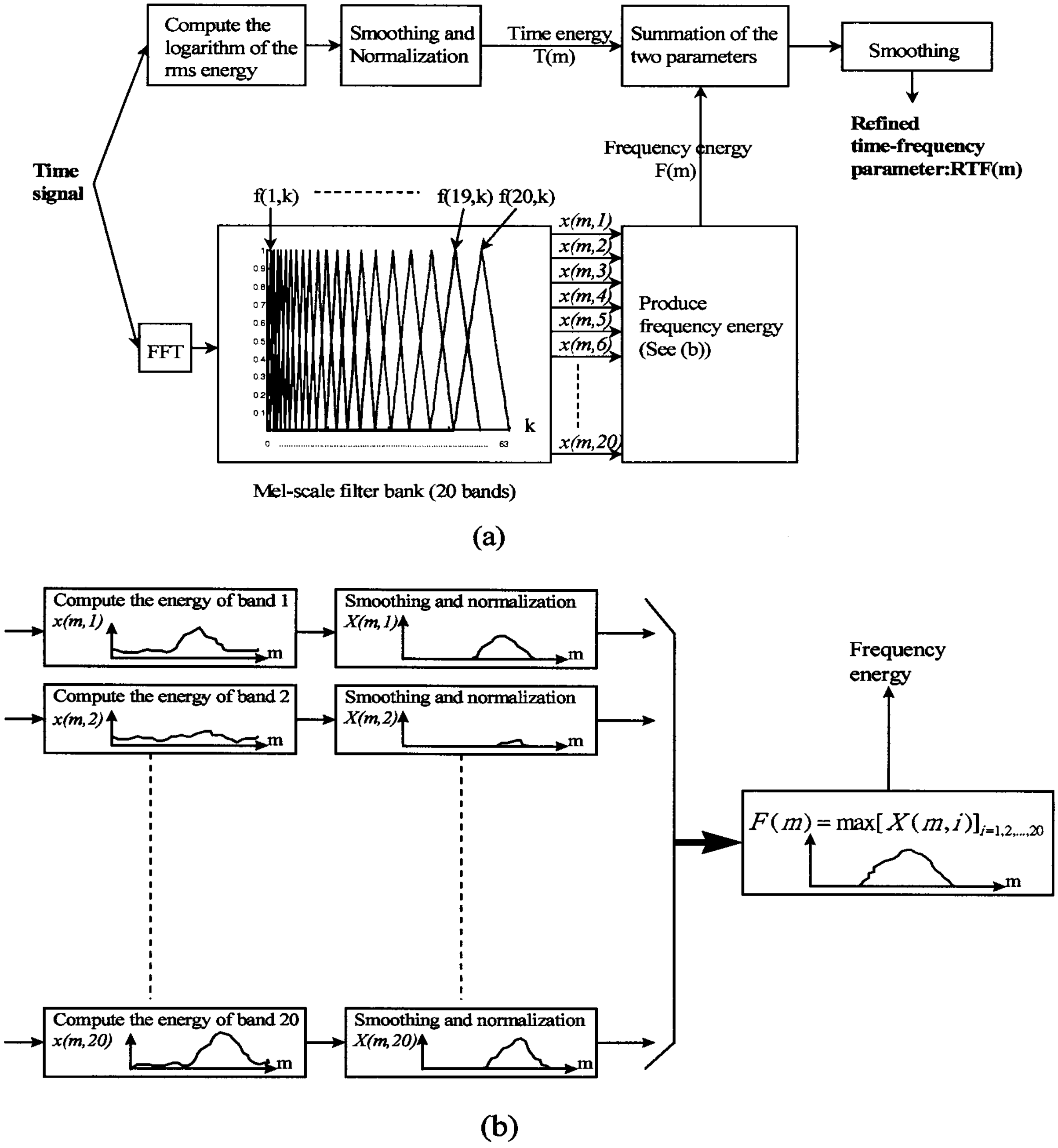

Compute the logarithm of the rms energy
Smoothing and Normalization
Time energy T(m)
Summation of the two parameters
Smoothing
Refined time-frequency parameter:RTF(m)
Time signal
Frequency energy F(m)
f(1,k)
f(19,k) f(20,k)
FFT
x(m,1)
x(m,2)
x(m,3)
x(m,4)
x(m,5)
x(m,6)
x(m,20)
k
0
63
Produce frequency energy (See (b))
Mel-scale filter bank (20 bands)
(a)
Compute the energy of band 1
x(m,1)
m
Smoothing and normalization
X(m,1)
Compute the energy of band 2
x(m,2)
Smoothing and normalization
X(m,2)
Compute the energy of band 20
x(m,20)
Smoothing and normalization
X(m,20)
Frequency energy
F(m) = max[X(m,i)]_{i=1,2,...,20}
(b)

Fig. 1. (a) Flowchart for computing the RTF parameter. (b) Procedure for producing the frequency energy in (a).

shall adopt the seventh and ninth bands to make the second and third word signals clear from the noise. Obviously, some bands have large frequency energy $X(m, i)$ and should be adopted to be useful bands. However, these useful bands may change under different word signals. This is because different word signals focus their frequency energy on different bands; some focus on low frequency bands, and others on high frequency bands.

Based on the above discussion and illustrations, we now propose a way to adaptively extract helpful frequency information from word signals. Since $X(m, i)$ is a good indicator for the amount of speech information, we adopt the maximum $X(m, i)$ to get the final frequency energy $F(m)$ of frame m

$$F(m) = \max[X(m, i)]_{i=1,2,\ldots,20}. \tag{7}$$

The proposed RTF parameter of the mth frame is the result obtained after smoothing the sum of the frequency energy $F(m)$ in (7) and time energy $T(m)$

$$\mathrm{RTF}(m) = \mathrm{SMOOTHING}(T(m) + cF(m)) \tag{8}$$

where SMOOTHING is performed by a three-point median filter as in (5), and the constant c is set as 0.8. The evaluation of this weighting factor c is given in Subsection. The time energy $T(m)$ is given by smoothing and normalizing the logarithm of the root-mean-square (rms) energy of the time–domain speech signal

$$x_{\mathrm{rms}}(m) = \log \sqrt{\frac{\sum_{n=0}^{L-1} x_{\mathrm{time}}^2(m, n)}{L}} \tag{9}$$

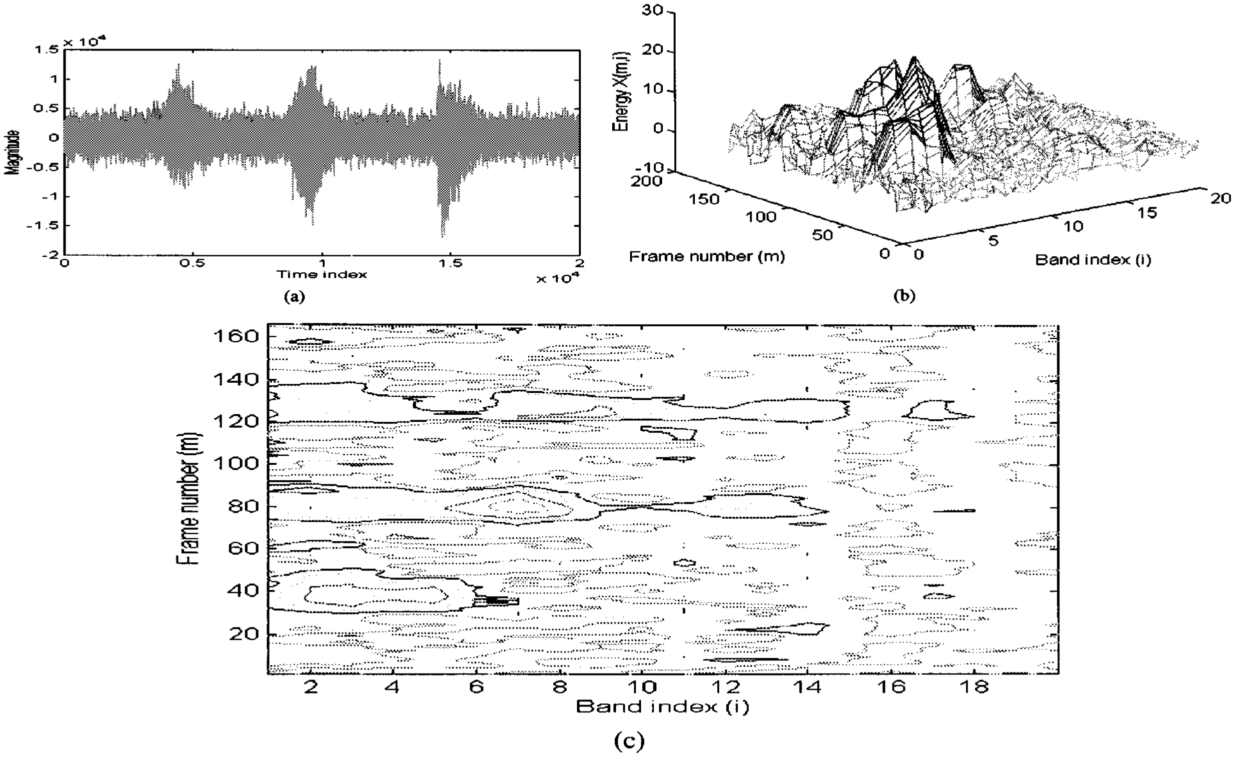


Fig. 2. (a) Speech waveform recorded in additive white noise of 0 dB. (b) Smoothed and normalized frequency energies $X(m,i)$ on 20 frequency bands. (c) Contour of (b).

$$\hat{x}_{\rm rms}(m) = \frac{x_{\rm rms}(m-1) + x_{\rm rms}(m) + x_{\rm rms}(m+1)}{3} \tag{10}$$

$$\begin{aligned} T(m) &= \hat{x}_{\rm rms}(m) - \text{Noise\_time} \\ &= \hat{x}_{\rm rms}(m) - \frac{\sum_{m=0}^{4} \hat{x}_{\rm rms}(m)}{5} \end{aligned} \tag{11}$$

where L is the length of the frame, which is 120 (15 ms) in our system. The procedure to calculate the RTF parameter is illustrated in Fig. 1(a). The details of the block with label "Produce frequency energy" of this figure are shown in Fig. 1(b).

Up to now, we have proposed the RTF parameter to indicate the amount of word signal information. Based on this RTF parameter, we further propose a new word boundary detection algorithm by using a RSONFIN. Since RSONFIN can process the temporal relations, it can find the variation of the background noise level and detect correct word boundaries in the condition of variable background noise level.

III. Recurrent Self-Organizing Neural Fuzzy Inference Network (RSONFIN)

The recurrent neural fuzzy network that we used for word boundary detection is called the RSONFIN that we proposed previously in [11]. The RSONFIN is constructed from a series of dynamic fuzzy rules. The temporal relations embedded in the network are built by adding some feedback connections representing the memory elements to a feedforward *neural fuzzy* network. Each weight as well as node in the RSONFIN has its own meaning and represents a special element in a fuzzy rule. There are no hidden nodes, i.e., no membership functions and fuzzy rules, initially in the RSONFIN. They are created online via concurrent structure identification (the construction of dynamic fuzzy IF–THEN rules) and parameter identification (the tuning of the free parameters of membership functions). The structure learning together with the parameter learning forms a fast-learning algorithm for building a small, yet powerful, dynamic neural fuzzy network. The number of generated rules and membership functions is small even for modeling a sophisticated system.

A. Structure of the RSONFIN

In this section, the structure of the RSONFIN shown in Fig. 3 is introduced. The RSONFIN consists of nodes, each of which has some finite fan-in of connections from other nodes and some fan-out of connections to other nodes. Basically, it is a five-layered neural fuzzy network embedded with dynamic feedback connections (the feedback layer in Fig. 3) that bring the temporal processing ability into a feedforward neural fuzzy network. To give a clear understanding of the network structure, the function of the node in each layer is described below. In the following descriptions, the symbol $u_i^{(k)}$ denotes the ith input of a node in the kth layer; correspondingly, the symbol $a^{(k)}$ denotes the node output in layer k.

Layer 1: No computation is done in this layer. Each node in this layer is called an input linguistic node and corresponds to one input variable. The node only transmits input values to the next layer directly. That is

$$a^{(1)} = u_i^{(1)}. \tag{12}$$

Layer 2: Nodes in this layer are called input term nodes, each of which corresponds to one linguistic label (small, large,

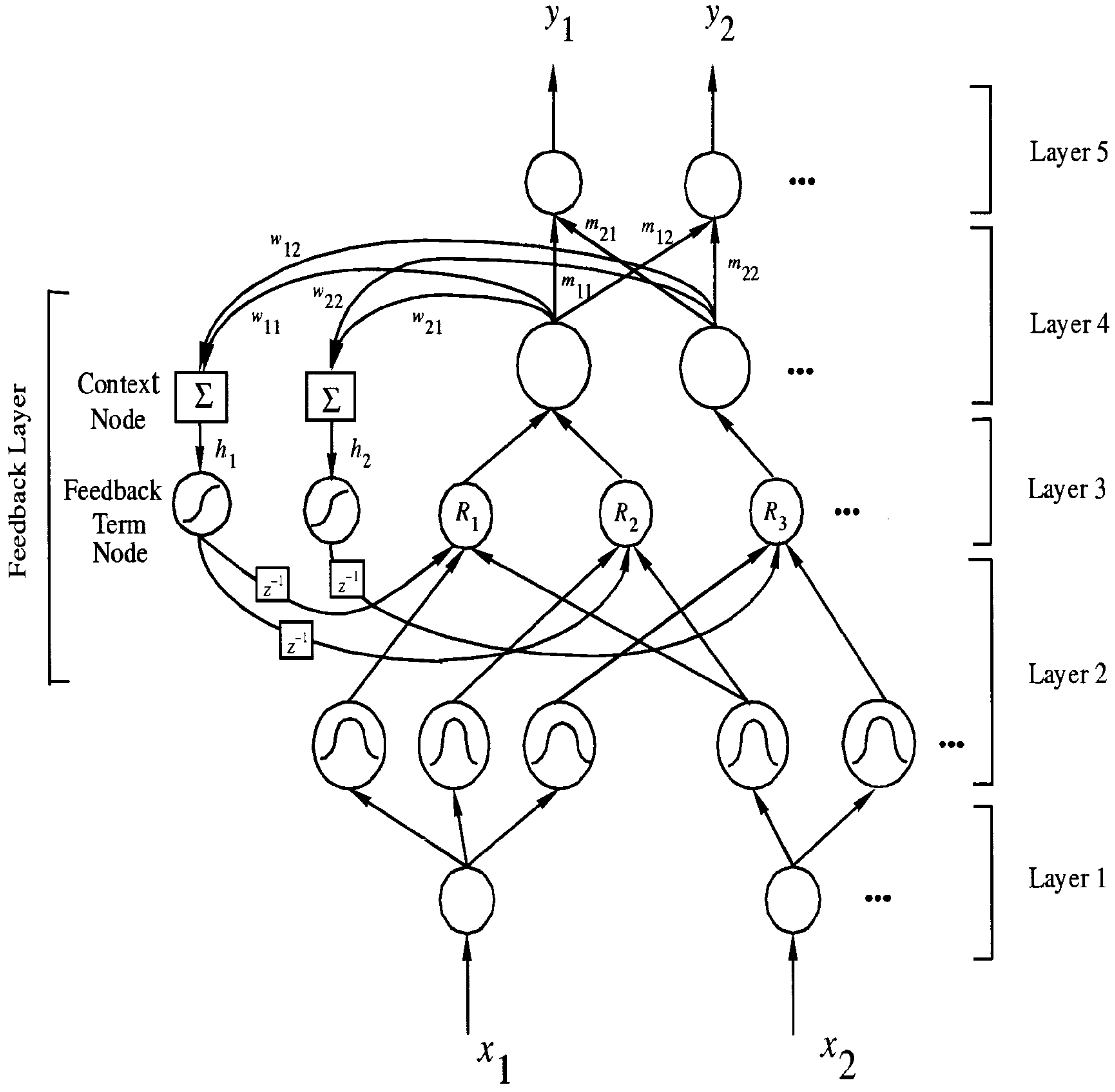


Fig. 3. Structure of the RSONFIN.

etc.) of an input variable. Each node in this layer calculates the membership value specifying the degree to which an input value belongs to a fuzzy set. A local membership function is used in this layer. There are many qualified candidates for the types of membership functions, such as triangular-, trapezoidal-, or Gaussian-membership functions. Here, a Gaussian- membership function is employed. The reason is that a multidimensional Gaussian- membership function can be easily decomposed into the product of one-dimensional (1-D) membership functions. With this choice, the operation performed in this layer is

$$a^{(2)} = \exp\left\{ -\frac{\left(u_i^{(2)} - m_{ij}\right)^2}{\sigma_{ij}^2} \right\} \tag{13}$$

where m_{ij} and σ_{ij} are, respectively, the center and the width of the Gaussian-membership function of the jth term of the ith input variable x_i.

Layer 3: Nodes in this layer are called rule nodes. A rule node represents one fuzzy logic rule and performs precondition matching of a rule. The fan-in of a fuzzy node comes from two sources: 1) from layer 2; and 2) from the feedback layer. The former represents the rule's spatial firing degree, and the latter the rule's temporal firing degree. We use the following AND operation on each rule node to integrate these fan-in values:

$$\begin{aligned} a^{(3)} &= a^{(6)} \cdot \prod_i u_i^{(3)} \\ &= a^{(6)} \cdot e^{-[D_i(\mathbf{x}-\mathbf{m}_i)]^T[D_i(\mathbf{x}-\mathbf{m}_i)]} \end{aligned} \tag{14}$$

where $D_i = \mathrm{diag}(1/\sigma_{i1}, 1/\sigma_{i2}, \ldots, 1/\sigma_{in})$, $\mathbf{m}_i = (m_{i1}, m_{i2}, \ldots, m_{in})^T$, and $a^{(6)}$ is the output of the feedback term node which will be described in the feedback layer part in this section. Obviously, the output $a^{(3)}$ of a rule node represents the firing strength of its corresponding rule.

Layer 4: This layer is called the consequent layer and the nodes in this layer are called output term nodes. Each output term node represents a multidimensional fuzzy set (described by a multidimensional Gaussian function) obtained during the clustering operation in the structure learning phase. Only the center of each Gaussian membership function is delivered to the next layer for the local mean of maximum (LMOM) defuzzification operation [16], so the width is used for output clustering only. Different nodes in Layer 3 may be connected to a same node in this layer, meaning that the same consequent is specified for different rules. The function of each output term node performs the following fuzzy OR operation:

$$a^{(4)} = \sum_i u_i^{(4)} \tag{15}$$

to integrate the fired rules which have the same consequent part.

Layer 5: Each node in this layer is called an output linguistic node and corresponds to one output linguistic variable. This layer performs the defuzzification operation. The nodes in this layer together with the links attached to them accomplish this task. The function performed in this layer is

$$y_j = a^{(5)} = \frac{\sum_i u_i^{(5)} \hat{m}_{ji}}{\sum_i u_i^{(5)}} \tag{16}$$

where $u_i^{(5)} = a_i^{(4)}$ and $\hat{m}_{ji}$, the link weight, is the center of the membership function of the ith term of the jth output linguistic variable.

Feedback Layer: This layer calculates the value of the internal variable h_i and the firing strength of the internal variable to its corresponding membership function, where the firing strength contributes to the matching degree of a rule node in Layer 3. As shown in Fig. 3, two types of nodes are used in this layer: 1) the square node named as *context node*; and 2) the circle node named as *feedback term node*, where each context node is associated with a feedback term node. The number of context nodes (and thus the number of feedback term nodes) are the same as that of output term nodes in Layer 4. Each context node and its associated feedback term node corresponds to one output term node. The inputs to a context node are from all the output term nodes, and the output of its associated feedback term node is fed to the rule nodes whose consequent is the output term node corresponded to this context node. The context node functions as a defuzzifier

$$h_j = \sum_i a_i^{(4)} w_{ji} \tag{17}$$

where the internal variable h_j is interpreted as the inference result of the hidden (internal) rule, and w_{ji} is the link weight from the ith node in Layer 4 to the jth internal variable. The link weight w_{ji} represents a fuzzy singleton in the consequent part of a rule, and also a fuzzy term of the internal variable h_j. For an internal variable, a fuzzy singleton instead of a fuzzy membership function is used as its fuzzy term; a fuzzy membership function on an internal variable does not make much sense in the network due to the use of the LMOM defuzzification operation, where only the center of the Gaussian membership function is used. This is different from the situation for the input and output linguistic variables, where the widths of fuzzy membership functions are used for clustering the input and output training data. In (17), the simple weighted sum is calculated [17], [18]. Instead of using the weighted sum of each rule's outputs as the inference result, the conventional average weighted sum $h_j = \sum_i a_i^{(4)} w_{ji} / \sum_i a_i^{(4)}$ can also be used [18], [19].

As to the feedback term node, unlike the case in the space domain where a local membership function is used, a global membership function is adopted on the universe of discourse of the internal variable to simplify network structure and meet the global property of the temporal history. Here, the global property means that for a cluster in the space domain its history path (memorized by the internal variables) can be anywhere in the space at different times, and so a global membership function, which covers the universe of discourse of the internal variable, is used to rank the influence degree each internal variable contributes to a rule. In this paper, the membership function $f(u) = 1/(1 + e^{-u})$ is used for each internal variable. With this choice, the feedback term node evaluates the output by

$$a^{(6)} = \frac{1}{1 + e^{-h_i}}. \tag{18}$$

This output is connected to the rule nodes in Layer 3, which connect to the same output term node in Layer 4. The outputs of feedback term nodes contain the firing history of the fuzzy rules.

With the aforementioned node functions in each layer, the RSONFIN realizes the following dynamic fuzzy reasoning [20] (see the equation at the bottom of the page), where

x_i	input variable;
y_i	output variable;
$A_{i1}, A_{in}, G, B_{i1}$, and B_{i2}	fuzzy sets;
h_i	internal variable;
w_{1i} and w_{mi}	are fuzzy singletons, and
n and m	are the numbers of input and internal variables, respectively.

B. Learning Algorithms for the RSONFIN

Two types of learning, structure and parameter learning, are used concurrently for constructing the RSONFIN. The structure learning includes the precondition, consequent, and feedback structure identification of a fuzzy IF–THEN rule. Here the precondition structure identification corresponds to the input space partitioning. The consequent structure identification is to decide when to generate a new membership function for the output variable based upon clustering. As to the feedback structure identification, the main task is to decide the number of internal variables with its corresponding feedback fuzzy terms and the connection of these terms to each rule. For the parameter learning, based

Rule i: IF $x_1(t)$ is A_{i1} and $\ldots$ and $x_n(t)$ is A_{in} and $h_i(t)$ is G
THEN $y_1(t+1)$ is B_{i1} and $y_2(t+1)$ is B_{i2} and $h_1(t+1)$ is w_{1i} and $\ldots$ and $h_m(t+1)$ is w_{mi}

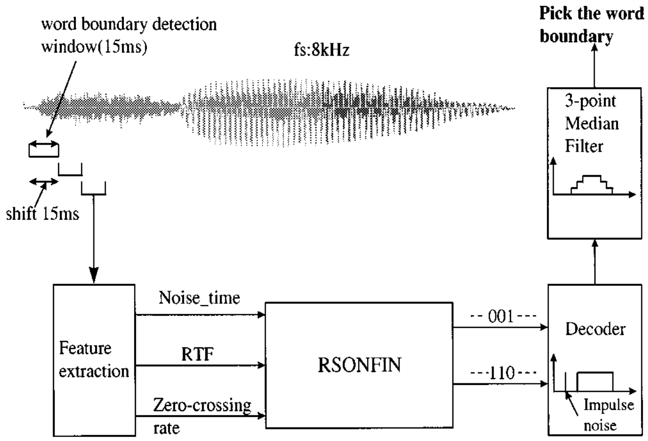


Fig. 4. RTF-based RSONFIN algorithm for automatic word boundary detection.

upon supervised learning, an ordered derivative learning algorithm is derived to update the free parameters in the RSONFIN. There are no rules, i.e., no nodes in the network except the I/O linguistic nodes, in the RSONFIN initially. They are created dynamically as learning proceeds upon receiving online incoming training data by performing the following learning processes simultaneously,

A. **Input–Output space partitioning.**
B. **Construction of fuzzy rules.**
C. **Feedback structure identification.**
D. **Parameter identification.**

In the above, processes A, B, and C belong to the structure learning phase and process D belongs to the parameter learning phase. The details of these learning processes are described in [11].

IV. RSONFIN FOR WORD BOUNDARY DETECTION

In this section, we shall develop a robust algorithm based on the RSONFIN to find the word boundary in the variable background noise level condition by using this network's learning ability for temporal relationships. We shall also introduce another existing word boundary detection algorithm for performance comparisons in the next section.

A. RTF-Based RSONFIN Algorithm

With the learning ability of temporal relations, a procedure of using the RSONFIN for word boundary detection in variable background noise level condition is illustrated in Fig. 4. The input feature vector of the RSONFIN consists of the average of the logarithmic root-mean-square (rms) energy on the first five frames of recording interval (Noise_time), RTF parameter, and zero-crossing rate (ZCR). The three parameters in an input feature vector are obtained by analyzing a frame of signal. Hence, there are three (input) nodes in Layer 1 of RSONFIN. Before entering the RSONFIN, the three input parameters are normalized to be in $[0, 1]$. For each input vector (corresponding to a frame), the output of RSONFIN indicates whether the corresponding frame is a word signal or noise. For this purpose, we used two (output) nodes in Layer 5 of RSONFIN, where the output vector of $(1, 0)$ standing for word signal and $(0, 1)$ for noise.

The RSONFIN was trained by a speech waveform with 15 s. This speech waveform is added by white noise with increasing and decreasing energy, and then each frame is transformed to be the desired input feature vector of the RSONFIN (Noise_time, RTF parameter, and zero-crossing rate). These training vectors are classified as word signal or noise by using waveform, spectrum displays, and audio output. Among these training vectors, some are word sounds with the desired RSONFIN output vector being $(1, 0)$, and all others are noises with the desired RSONFIN output vector being $(0, 1)$. Although the zero-crossing rate (ZCR) is not reliable for speech segmentation in noisy environments, it is still an important parameter in clean environments. Hence, we also adopt it as an input parameter of RSONFIN. In the training phase, the RSONFIN will tune the proper weighting of ZCR automatically to reach the optimum performance of speech segmentation not only in noisy environments but also in clean environments.

The RSONFIN after training is ready for word boundary detection. As shown in Fig. 4, the outputs of RSONFIN are processed by a decoder. The decoder decodes the RSONFIN's output vector $(1, 0)$ as value 100 standing for word signal and $(0, 1)$ as value 0 standing for noise. We observed that the decoding waveform, i.e., the outputs of the decoder, contains impulse noise sometimes. Hence, we let the output waveform of the decoder pass through a three-point median filter to eliminate the isolated "impulse" noise. Finally, we recognize the word-signal island as the part of the filtered waveform whose magnitude is greater than 30, and duration is long enough (by setting a threshold value). We then regard the parts of original signal corresponding to the allocated word-signal island as the word signal, and the other ones as the background noise.

B. TF-Based Algorithm

In this section, we introduce the TF-based algorithm proposed in [5] for performance comparison with the proposed RSONFIN-based algorithm. TF-based algorithm used the TF parameter and was shown to outperform several commonly used algorithms for word boundary detection in the presence of noise. The TF parameter uses the frequency energy in the fixed frequency band 250–3500 Hz to enhance the time–energy information. The TF parameter is the result obtained after smoothing the sum of the time energy and frequency energy. This frequency energy helps us to make the distinction between speech and noise. The TF-based robust algorithm first performs a noise classification procedure to determine noise level (high, medium, or low) and the noise category (high or low zero-crossing rate) by using ten frames of "relative" silence at the beginning of the recording, and computing an average of the logarithmic rms energy and the zero-crossing rate on these frames. A set of empirically determined threshold values are used to perform the noise classification. After noise classification, the TF-based robust algorithm applies a noise adaptive procedure to determine the word boundary. It uses the TF parameter with some thresholds to find the islands of reliability boundary. Finally, the refinement procedure, which also depends on the noise classification results, is applied to the initial boundary. It tries to find the earliest boundary by subtracting an adjustment value (typically 20 ms) from the beginning boundary to obtain a new boundary

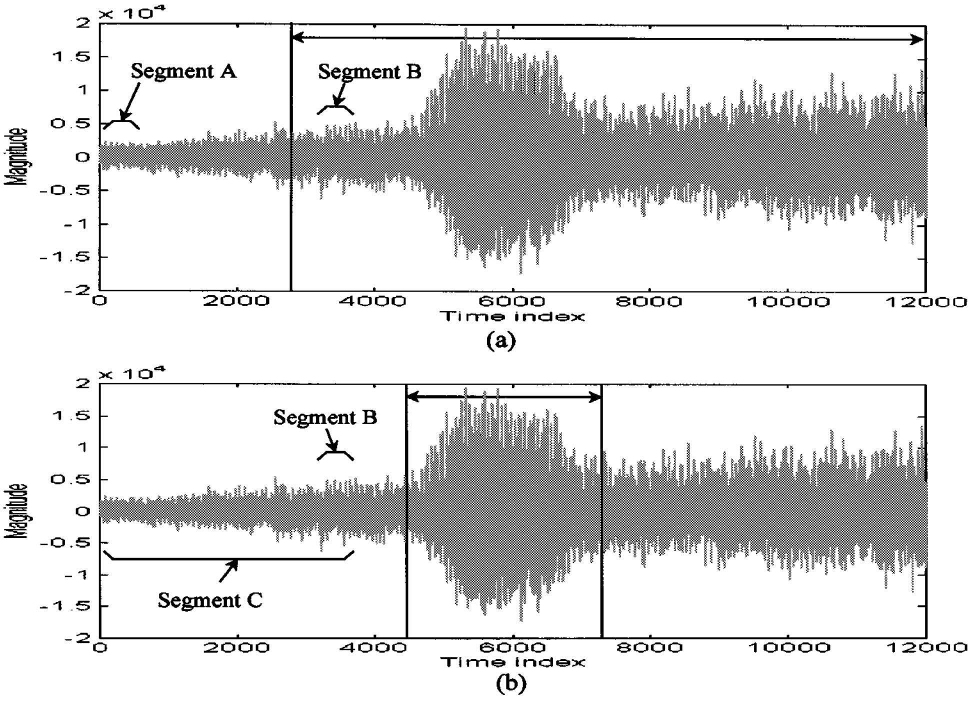


Fig. 5. Speech waveforms recorded in additive increasing-level white noise (12000 speech samples and $\mathrm{SNR} = 5$ dB). (a) Word boundaries detected by the TF-based algorithm are shown by solid lines. Segment B is detected as word signal incorrectly, and the ending boundary is missing. (b) Word boundaries detected by the RTF-based RSONFIN algorithm (ten rules) are shown by solid lines. Segment B is detected as background noise correctly.

(maximum up to 100 ms from the beginning island of the reliability boundary). Some thresholds are then used to determine the final beginning boundary. It tries to find the latest boundary by adding an adjustment value (typically 50 ms) from the ending boundary to obtain a new boundary (maximum up to 150 ms from the ending island of the reliability boundary). Then, the refinement procedure uses some thresholds to determine the final ending boundary. The thresholds in the refinement procedure include the logarithm of the time–domain rms energy and zero-crossing rate.

C. Test Environments and Noise Speech Database

In the word boundary detection procedure, the frame length is set to 15 ms in order to get more accurate endpoint location. The sampling rate of our system is 8 KHz. We take the white noise from the NATO Research Study Group on Speech Processing (RSG.10) NOISE-ROM-0 [21] for speech contamination in our experiments. The original NOISE-ROM-0 data were sampled at 19.98 KHz and stored as 16-bit integers. In our experiments, they are prepared for use by downsampling to 8 KHz and applying attenuation to them. The attenuation was applied to enable the addition of noise without causing an overflow of the 16-bit integer range. The speech data used for our experiments are the set of isolated Mandarin digits. The recording sampling rate is 8 KHz and stored as 16-bit integer.

D. Analysis in Variable Background Noise Level Condition

Fig. 5(a) shows a typical example of the increasing background noise level. A desired spoken word is presented in this interval, and its word boundaries detected by the TF-based algorithm are shown by solid lines. Although this algorithm outperforms several commonly used algorithms for word boundary detection in the presence of noise, we found that the located beginning boundary is wrong, and the ending boundary is missing. The major reason is that the TF-based algorithm always sets thresholds from the first few frames of the recording interval [segment A in Fig. 5(a)]. These preset thresholds determined by segment A are used to stand for the background noise level in all the recording interval and to find the word boundaries. In other words, segment B in Fig. 5(a) is determined to be word signal according to the noise property in segment A. In fact, segment B is the background noise. Since the background noise level changes in all the recording interval, it is not reasonable to use these preset thresholds determined by segment A to judge whether segment B is word signal or background noise. In addition, this TF-based algorithm cannot tune the preset thresholds determined by segment A properly according to the variation of background noise level, so the preset thresholds are proper in segment A and improper in segment B. Improper thresholds will result in incorrect location of word boundaries.

Now, we use the proposed RTF-based RSONFIN algorithm to repeat the same experiment. After training, there were only 10 rules generated in the RTF-based RSONFIN algorithm [see Fig. 6(a)]. The number of fuzzy sets on the variables, RTF parameter, zero-crossing rate and Noise_time, are seven, seven, and five, respectively see [Figs. 6(b)–(d)]. The word boundaries detected by the RTF-based RSONFIN algorithm are shown by solid lines in Fig. 5(b). We found that the beginning and ending boundaries were detected properly. The major reason is that the RSONFIN can learn the temporal relations automatically and

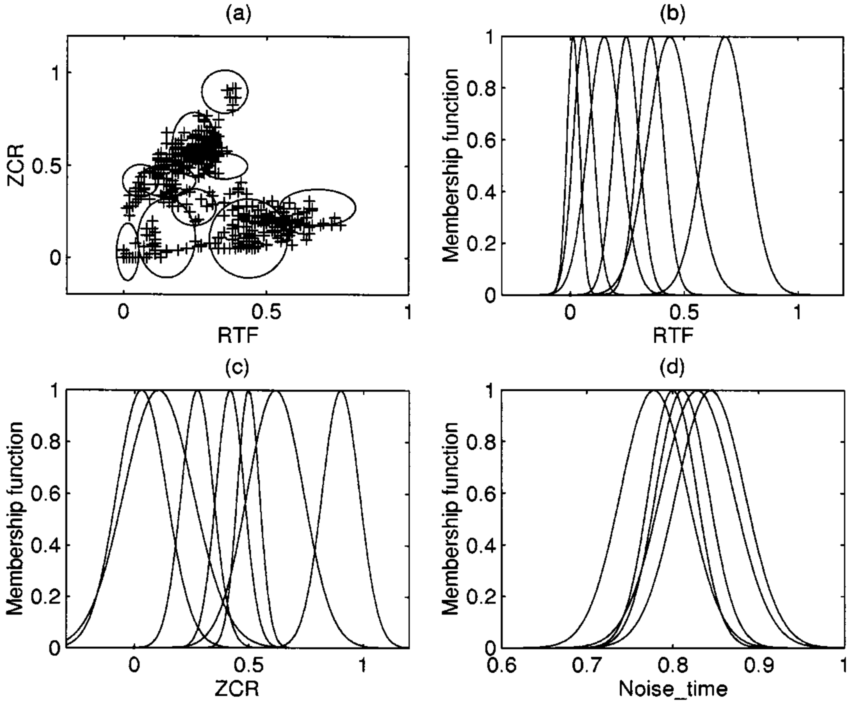


Fig. 6. (a) Input training patterns for the RTF-based RSONFIN algorithm and the final assignment of ten rules. (b) Distribution of the membership functions on the dimension of "RTF" variable. (c) Distribution of the membership functions on the dimension of "ZCR" variable. (d) Distribution of the membership functions on the dimension of "Noise_time" variable.

implicitly and is trained by a speech waveform which is in variable background noise level condition. Based on the temporal relations embedded in the RSONFIN, this algorithm can trace the variations of the background noise level and detect correct word boundaries. In other words, segment B is determined to be background noise according to the noise property in segment C in Fig. 5(b), where segment C is from the beginning of recording interval to segment B (including segment B). In the next section, we shall do experiments in several kinds of background noise level conditions in order to show the performance of the RTF-based RSONFIN algorithm.

V. Experiments

In this section, we test the performance of the proposed RTF-based RSONFIN algorithm in two experiments. In the first experiment, we demonstrate the segmentation results in two kinds of variable background noise level conditions and compare them to those obtained by hand labeling in clean environments. In the second experiment, the performance of the proposed algorithm on a large set of speech signals is evaluated through a speech recognizer, and the resulting recognition correct rate and error rate are reported.

A. Speech Segmentation in Variable Background Noise Level Conditions

In order to compare the effects of the TF and RTF parameters, we use the TF parameter instead of the RTF parameter in the RTF-based RSFONFIN algorithm to form another word boundary detection algorithm, called TF-based RSONFIN algorithm for performance comparison. Based on the same previous training procedure, there were 17 rules generated in the TF-based RSONFIN algorithm, and the number of fuzzy sets on the variables, Noise_time, TF parameter and zero-crossing rate, are nine, 16 and 12, respectively. However, there were only 10 rules generated in the RTF-based RSONFIN algorithm, and the number of fuzzy sets on the variables, Noise_time, TF parameter and zero-crossing rate, are five, seven, and seven, respectively. In this subsection, three word boundary detection algorithms (TF-based algorithm, TF-based RSONFIN algorithm and RTF-based RSONFIN algorithm) are tested in two kinds of background noise level conditions; increasing and decreasing background noise level conditions. There are totally seven words in the recording interval, which are Mandarin digits of "1, 2, 3, 4, 5, 6, 7".

1) Increasing Background Noise Level: In this experiment, the speech waveforms recorded in additive increasing-level white noise consists of 60 000 samples, and the SNR is 10 dB. We first make some observations on the effect of the increasing background noise level on the speech signal in Fig. 7(a), where the word boundaries detected by hand labeling *in clean environments* are shown by dotted lines. Obviously, the noise in the last half segment of recording interval is larger than the noise in the first half segment of recording interval. The noise makes the distinction between word signal and background noise ambiguous. Word boundaries detected by the TF-based algorithm are shown by solid lines in Fig. 7(a), where two word segments are found. Since the background noise level varies slowly in the beginning, the first word segment is determined properly. However, the ending boundary of the second word segment is missing. In fact, there are six words in this part. The major reason for this error is that the TF-based algorithm

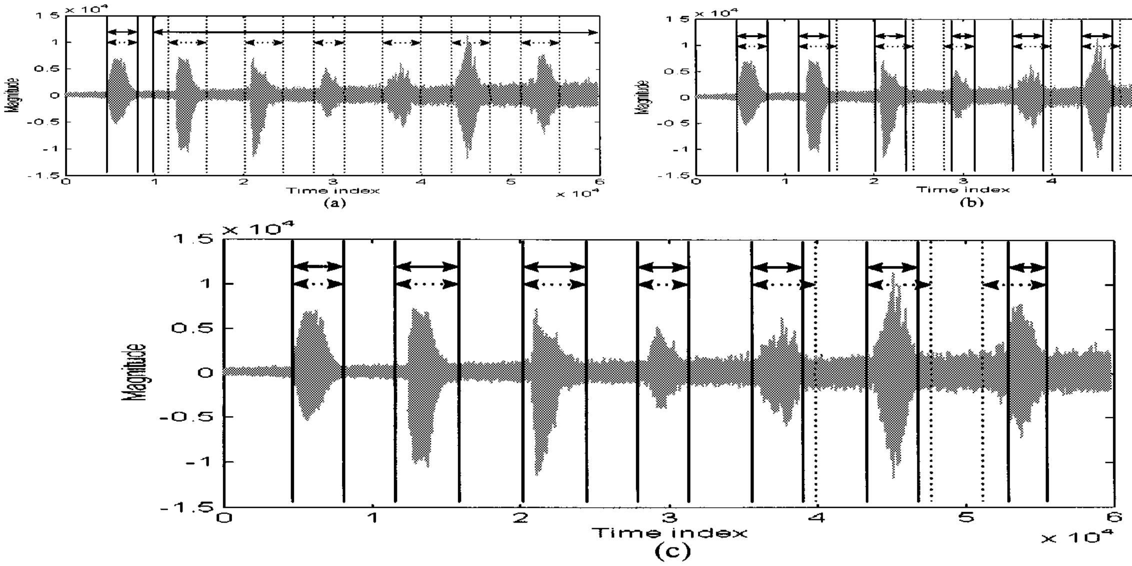


Fig. 7. Speech waveform recorded in additive increasing-level white noise including 60 000 samples with the SNR being 10 dB. The word boundaries detected by hand labeling in clean environments are shown by dotted lines. (a)Word boundaries detected by the TF-based algorithm are shown by solid lines, and we notice that the second word ending boundary is missing. (b)Word boundaries detected by the TF-based RSONFIN algorithm (17 rules) are shown by solid lines. (c) Word boundaries detected by the RTF-based RSONFIN algorithm (ten rules) are shown by solid lines.

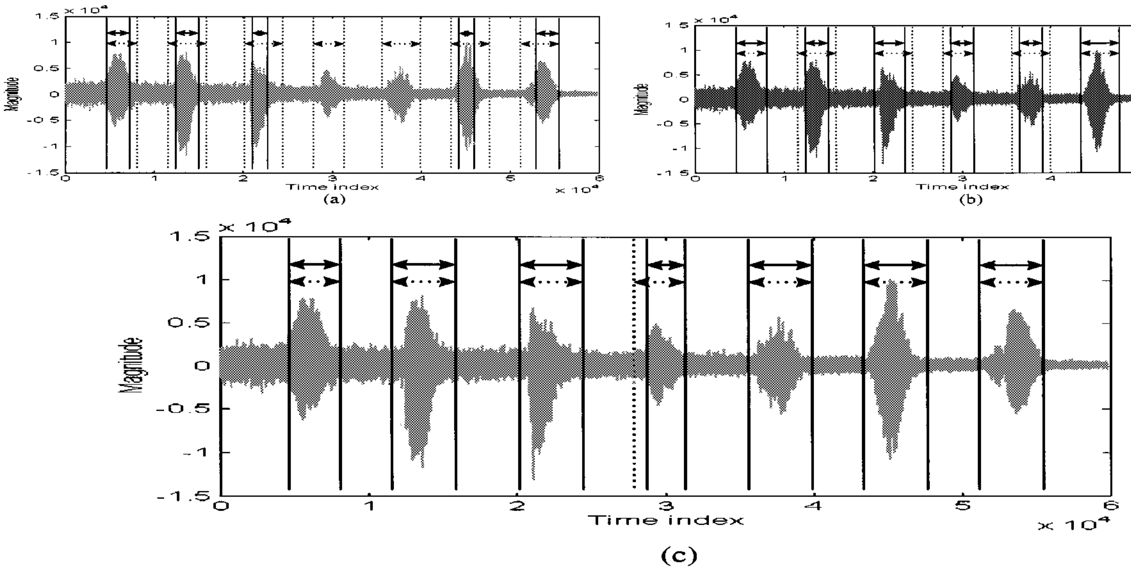


Fig. 8. Speech waveform recorded in additive decreasing-level white noise including 60 000 samples with the SNR being 10 dB. The word boundaries detected by hand labeling in clean environments are shown by dotted lines. (a)Word boundaries detected by the TF-based algorithm are shown by solid lines, and we notice that the fourth and fifth words are not detected at all. (b)Word boundaries detected by the TF-based RSONFIN algorithm (17 rules) are shown by solid lines. (c) Word boundaries detected by the RTF-based RSONFIN algorithm (ten rules) are shown by solid lines.

cannot detect the variation of the background noise level and does not decide proper thresholds to find word boundaries.

Next, we use the TF-based and RTF-based RSONFIN algorithms to repeat the same experiment. The word boundaries detected by the TF-based RSONFIN algorithm are shown by solid lines in Fig. 7(b), where seven word segments are found. Based on the temporal relations embedded in the RSONFIN, TF-based RSONFIN algorithm can find the variation of the background noise level and detect all word signals in the increasing background noise level condition. However, the boundaries of some word signals are not determined properly. The word boundaries detected by the RTF-based RSONFIN algorithm are shown by solid lines in Fig. 7(c). These word boundaries are more accurate than those detected by the TF-based RSONFIN algorithm. This is because that the RTF parameter can extract more informative frequency energy than the TF parameter to compensate the time–energy information by *adaptively* choosing proper frequency bands.

2) Decreasing Background Noise Level: In this experiment, the speech waveforms recorded in additive decreasing-level white noise consists of 60 000 samples, and the SNR is 10 dB. The effect of the decreasing background noise level on the speech signal can be observed in Fig. 8(a). Obviously, the noise in the first half segment of recording interval is larger than the noise in the last half segment of recording interval. Word boundaries detected by the TF-based algorithm are shown by solid lines in Fig. 8(a), where only five word segments are found, and the fourth and fifth words are missing. Although the seventh word is detected, the beginning part of this word is missing.

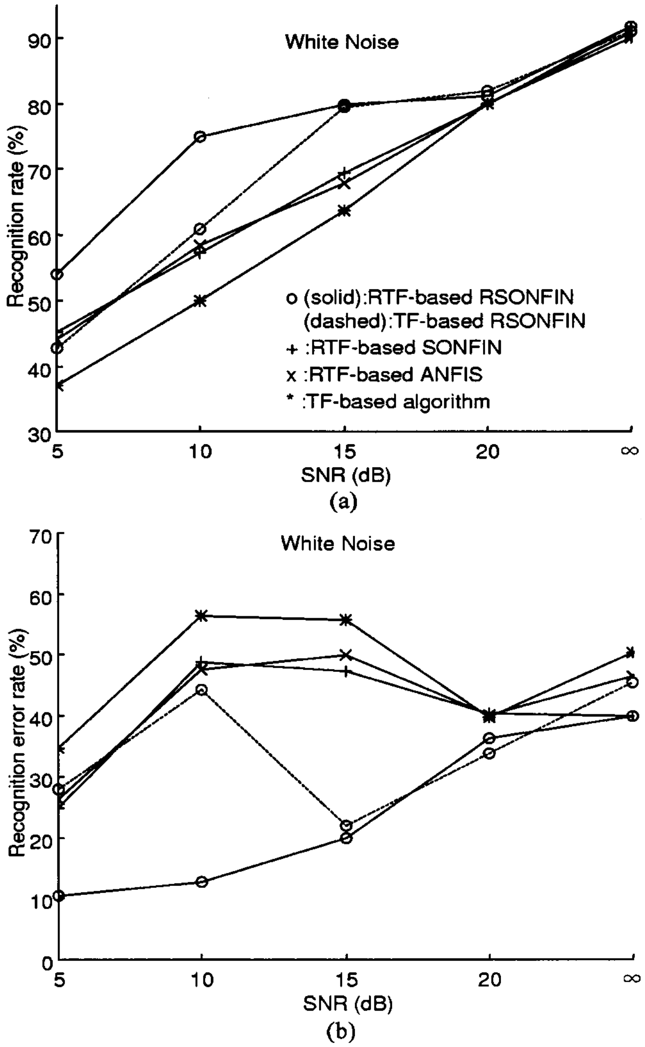
We then use the TF-based and RTF-based RSONFIN algorithms to repeat the same experiment again. The word boundaries detected by the TF-based RSONFIN algorithm are shown by solid lines in Fig. 8(b), where seven word segments are found. This algorithm can really sense the variation of the background noise level and detect all word signals. However, the boundaries of some word signals are not determined properly. The word boundaries detected by the RTF-based RSONFIN algorithm are shown by solid lines in Fig. 8(c). These word boundaries are more accurate than those detected by the TF-based RSONFIN algorithm.

B. Speech Recognition in Variable Background Noise Level Conditions

Since inaccurate detection of word boundary is harmful to recognition, the performance of the word boundary detection process can be also examined by the recognition rate of a speech recognizer. The speech recognizer used in this experiment consists of two parts, feature extractor and classifier. In the feature extractor, the modified two-dimensional (2-D) cepstrum (modified TDC—MTDC) [22]–[25] is used as the speech feature. The MTDC can simultaneously represent several types of information contained in the speech waveform: static and dynamic features, as well as global and fine frequency structures. To represent an utterance, only some MTDC coefficients need to be selected to form a feature vector instead of the sequence of feature vectors. The MTDC has the advantage of simple computation and is suitable for noisy speech recognition due to its choices of robust coefficients. In the classifier, a Gaussian clustering algorithm is used. The training is done on clean speech pronounced in a clean environment (without background noise). In the training phase, each model is trained by a mixture of four Gaussian distribution density functions. We use a total of 1000 utterances for training. The details of the above isolated word recognition system can be found in [25].

The speech data used for our experiment are the set of isolated Mandarin digits. They are ten digits spoken by 10 speakers and each speaker pronounced 20 times of the ten digits. The recording sampling rate is 8 KHz and stored as 16-bit integer. To set up the noisy speech database for testing, we add the prepared noisy signals to the recorded speech signals with different signal-to-noise-ratios (SNR's) including 5 dB, 10 dB, 15 dB, 20 dB and ∞ dB. The duration of each utterance used for testing the performance of the word boundary detection algorithm is about one second (including silence). A total of 600 utterances are used in our experiment; 300 utterances are in the condition of increasing background noise level, and 300 utterances are in the condition of decreasing background noise level.

In addition to the three word boundary detection algorithms used in the previous speech segmentation tests, we also compare the performance of RSONFIN to that of two other neural fuzzy networks. They are the self-constructing neural fuzzy inference network (SONFIN) that we proposed previously in [26] and the adaptive-network-based fuzzy inference system (ANFIS) [27]. We use the SONFIN and ANFIS to replace the RSONFIN in the RTF-based RSONFIN algorithm to form the RTF-based SONFIN and RTF-based ANFIS algorithms, respectively. As a result, there are five word boundary detection algorithms used for testing in the followings.



(a)

(b)

Fig. 9. (a) Recognition rates and (b) error rates of five word boundary detection algorithms (RTF-based RSONFIN, TF-based RSONFIN, RTF-based SONFIN, RTF-based ANFIS, and TF-based algorithms) in the condition of variable background noise level.

The recognition rates of the five algorithms for added white noise with different SNR's are shown in Fig. 9(a). In addition, we also consider the error rate which is the ratio of the recognition errors due to incorrect word boundary detection (taking recognition scores obtained with hand-labels as a reference) to the total number of recognition errors of the detection algorithm. More precisely, let the recognition errors obtained by using hand labeling be E_{hl}, and the recognition errors obtained by using automatic word boundary detection algorithm be E_{al}. Then the recognition error rate is given by $(E_{\mathrm{al}} - E_{\mathrm{hl}})/E_{\mathrm{al}}$. The resulting recognition error rates of the five algorithms are given in Fig. 9(b).

From the above results, we find that the performance of the RTF-based SONFIN algorithm is similar to that of the RTF-based ANFIS algorithm, and they both outperform the

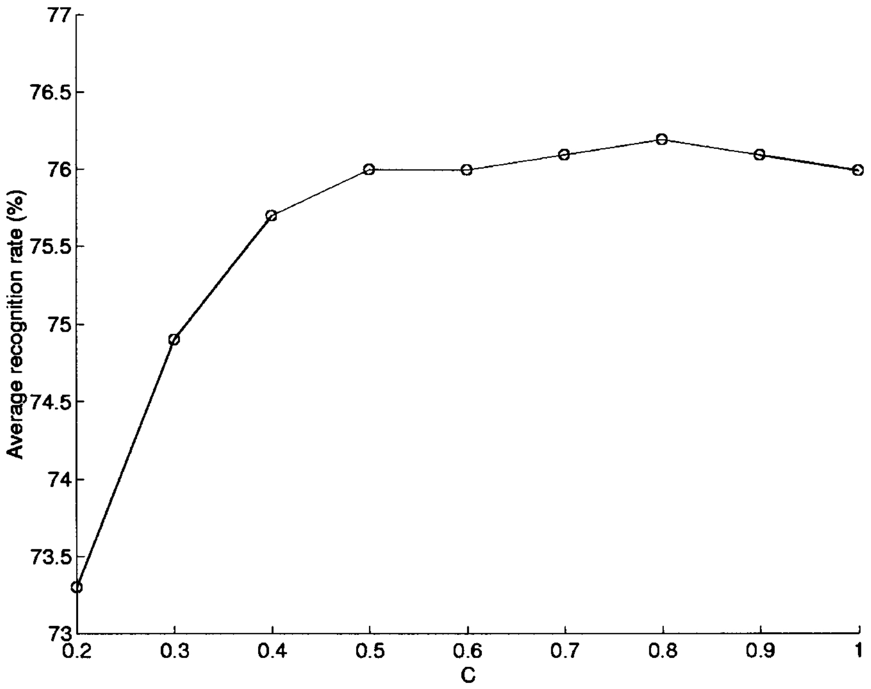


Fig. 10. Average recognition rates of the RTF-based RSONFIN algorithm with respect to various values of the parameter c in (8).

TF-based algorithm by about 4%. With the temporal relations captured and embedded in the RSONFIN, the TF-based RSONFIN algorithm outperfoms the RTF-based SONFIN and RTF-based ANFIS algorithms by about 3%. In addition, since the RTF parameter can extract useful frequency energy through multiband spectrum analysis, the RTF-based RSONFIN algorithm outperforms the TF-based RSONFIN algorithm by about 5%. As a total, the RTF-based RSONFIN algorithm has higher recognition rate than the TF-based algorithm in [5] by about 12%. Also, the RTF-based RSONFIN algorithm reduces the recognition error rate due to endpoint detection to about 23%, compared to about 34% obtained by the TF-based RSONFIN algorithm, about 40% obtained by the RTF-based SONFIN or RTF-based ANFIS algorithms, and about 47% obtained by the TF-based algorithm in [5].

Finally, to evaluate the weighting factor c in (8), we show the average recognition rates of the RTF-based RSONFIN algorithm with respect to various values of c in Fig. 10. Since the value $c = 0.8$ results in the highest average recognition rate, we set $c = 0.8$ in (8).

VI. Conclusion

Three major characteristics of the proposed RTF-based RSONFIN word boundary detection algorithm can be seen.

1) The proposed RTF parameter can extract both the time and frequency features of noisy speech signals through multiband spectrum analysis. Since this RTF parameter can extract more informative frequency energy than the TF parameter to compensate the time–energy information by *adaptively* choosing proper frequency bands, the RTF-based RSONFIN algorithm with fewer (10 in our experiments) rules outperforms the TF-based RSONFIN algorithm with more (17 in our experiments) rules.
2) The recurrent property of the RSONFIN makes it more suitable for dealing with temporal problems. Since the RSONFIN can recognize the temporal relations automatically and implicitly, the proposed algorithm can find the variation of the background noise level and detect correct word boundaries in the condition of variable background noise level.
3) No predetermination, like the number of hidden nodes, must be given to the RSONFIN, since it can find its optimal structure and parameters automatically and quickly. This avoids the need of empirically determining the number of hidden layers and nodes in normal neural networks. Due to this self-learning ability of RSONFIN, our proposed RSONFIN-based algorithm avoids the need of empirically determining ambiguous decision rules in normal word boundary detection algorithms. Also, since the RSONFIN houses the human-like IF-THEN rules in its network structure, expert knowledge can be put into the network as a priori knowledge, which can usually increase its learning speed and detection accuracy.

The RTF-based RSONFIN algorithm has been tested over a variety of noise conditions and has been found to perform well not only in variable background noise level condition but also in fixed background noise level condition. Our results show that the RTF-based RSONFIN algorithm achieved higher recognition rate than the TF-based algorithm by about 12% in variable background noise level conditions. It also reduced the recognition error rate due to endpoint detection to about 23%, compare to about 34% obtained by the TF-based RSONFIN algorithm, about 40% obtained by the RTF-based SONFIN or RTF-based ANFIS algorithms, and about 47% obtained by the TF-based algorithm in the same condition.

REFERENCES

[1] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell Syst. Tech. J.*, vol. 54, no. 2, pp. 297–315, Feb. 1975.

[2] M. H. Savoji, "A robust algorithm for accurate endpointing of speech," *Speech Commun.*, vol. 8, pp. 45–60, 1989.

[3] L. F. Lamel, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilson, "An improved endpoint detector for isolated word recognition," *IEEE Acoust., Speech, Signal Process. Mag.*, vol. 29, pp. 777–785, Aug. 1981.

[4] B. Reaves, "Comments on an improved endpoint detector for isolated word recognition," *IEEE Trans. Signal Processing*, vol. 39, pp. 526–527, Feb. 1991.

[5] J. C. Junqua, B. Mak, and B. Reaves, "A robust algorithm for word boundary detection in the presence of noise," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 406–412, July 1994.

[6] Y. Qi and B. R. Hunt, "Voiced-unvoiced-silence classification of speech using hybrid features and a network classifier," *IEEE Trans. Speech Audio Processing*, vol. 1, pp. 250–255, Apr. 1993.

[7] S. J. Kia and G. G. Coghill, "A mapping neural network and its application to voiced-unvoiced-silence classification," in *Proc. 1st New Zealand Int. Two-Stream Conf. Artificial Neural Networks Expert Systems*, 1993, pp. 104–108.

[8] M. Hamada, Y. Takizawa, and T. Norimatsu, "A noise robust speech recognition," in *Proc. Int. Conf. Spoken Language Processing*, 1990, pp. 893–896.

[9] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*. New York: Macmillan, 1993.

[10] T. Ghiselli-Crippa and A. El-Jaroudi, "A fast neural net training algorithm and its application to voiced-unvoiced-silence classification of speech," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, 1991, pp. 441–444.

[11] C. F. Juang and C. T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. Neural Networks*, vol. 10, pp. 828–845, July 1999.

[12] C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, May 1996.

[13] C. T. Lin, *Neural Fuzzy Control Systems with Structure and Parameter Learning*. Singapore: World Scientific, 1994.

[14] J. B. Allen, "Cochlear modeling," *IEEE Acoust., Speech, Signal Processing Mag.*, vol. 2, pp. 3–29, 1985.

[15] D. O'Shaughnessy, *Speech Communication*. Reading, MA: Addison-Wesley, 1987.

[16] H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks*, vol. 3, pp. 724–740, 1992.

[17] H. Takagi and I. Hayashi, "NN-driven fuzzy reasoning," *Int. J. Approx. Reason.*, vol. 5, no. 3, pp. 191–212, 1991.

[18] J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference system," *IEEE Trans. Neural Networks*, vol. 4, pp. 156–159, 1993.

[19] T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," in *Proc. IFAC Symp. Fuzzy Information, Knowledge Representation, Decision Analysis*, July 1983, pp. 55–60.

[20] V. Gorrini and H. Bersini, "Recurrent fuzzy systems," in *Proc. IEEE Int. Conf. Fuzzy Systems*, vol. 1, 1994, pp. 193–198.

[21] A. Varga and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Commun.*, vol. 12, pp. 247–251, 1993.

[22] Y. Ariki, S. Mizuta, and T. Sakai, "Spoken-word recognition using dynamic features analyzed by two-dimensional cepstrum," *Proc. Inst. Elect. Eng.*, vol. 136, no. 2, Apr. 1989.

[23] H. F. Pai and H. C. Wang, "A study on two-dimensional cepstrum approach for speech recognition," *Comput. Speech Lang.*, vol. 6, pp. 361–375, 1992.

[24] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Processing*, vol. 2, pp. 578–589, Oct. 1994.

[25] C. T. Lin, H. W. Nein, and J. Y. Hwu, "GA-based noisy speech recognition using two-dimensional cepstrum," *IEEE Trans. Speech Audio Processing*, to be published.

[26] C. F. Juang and C. T. Lin, "An on-line self-constructing neural fuzzy inference network and its application," *IEEE Trans. Fuzzy Syst.*, vol. 6, pp. 12–32, Feb. 1998.

[27] J. S. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. Neural Networks*, vol. 3, pp. 714–723, Sept. 1992.

Gin-Der Wu received the B.S. degree in engineering science from the National Cheng-Kung University, Taiwan, R.O.C., in 1996. He is currently pursuing the Ph.D. degree in electrical and control engineering at the National Chiao-Tung University, Hsinchu, Taiwan, R.O.C.

His current research interests are speech recognition and enhancement in noisy environments, adaptive signal processing, neural networks, and fuzzy control.

Chin-Teng Lin (S'88–M'91–SM'99) received the B.S. degree in control engineering from the National Chiao-Tung University (NCTU), Hsinchu, Taiwan, R.O.C., in 1986 and the M.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, in 1989 and 1992, respectively.

Since August 1992, he has been with the College of Electrical Engineering and Computer Science, NCTU, where he is currently a Professor of electrical and control engineering. Since 1998, he has served as the Deputy Dean of the Research and Development Office of the National Chiao-Tung University. His current research interests are fuzzy systems, neural networks, intelligent control, human–machine interface, and video and audio processing. He is the co-author of *Neural Fuzzy Systems—A Neuro-Fuzzy Synergism to Intelligent Systems* (Englewood Cliffs, NJ: Prentice Hall), and the author of *Neural Fuzzy Control Systems with Structure and Parameter Learning* (Singapore: World Scientific). He has published over 50 journal papers in the areas of neural networks and fuzzy systems.

Dr. Lin is a member of Tau Beta Pi and Eta Kappa Nu. He is also a member of the IEEE Computer Society, the IEEE Robotics and Automation Society, and the IEEE Systems, Man, and Cybernetics Society. Since 1995, he has been an Executive Council Member of the Chinese Fuzzy System Association (CFSA). Since 1998, he has been the Supervisor of the Chinese Automation Association. He was the Vice Chairman of the Taipei Chapter of the IEEE Robotics and Automation Society in 1996 and 1997. He won the Outstanding Research Award granted by the National Science Council (NSC) in 1997–1998 and 1999–2000, the Outstanding Electrical Engineering Professor Award granted by the Chinese Institute of Electrical Engineering (CIEE) in 1997, and the Outstanding Engineering Professor Award granted by the Chinese Institute of Engineering in 2000.