International Journal of Pattern Recognition and Artificial Intelligence
Vol. 16, No. 7 (2002) 927–955

NOISY SPEECH SEGMENTATION/ENHANCEMENT WITH MULTIBAND ANALYSIS AND NEURAL FUZZY NETWORKS

CHIN-TENG LIN*, RUI-CHENG WU and GIN-DER WU
Department of Electrical and Control Engineering,
National Chiao-Tung University, Hsinchu, Taiwan, R.O.C.
* *ctlin@fnn.cn.nctu.edu.tw*

This paper addresses the problem of speech segmentation and enhancement in the presence of noise. We first propose a new word boundary detection algorithm by using a neural fuzzy network (called ATF-based SONFIN algorithm) for identifying islands of word signals in fixed noise-level environment. We further propose a new RTF-based RSONFIN algorithm where the background noise level varies during the procedure of recording. The *adaptive time-frequency* (ATF) and *refined time-frequency* (RTF) parameters extend the TF parameter from single band to multiband spectrum analysis, and help to make the distinction of speech and noise signals clear. The ATF and RTF parameters can extract useful frequency information by *adaptively* choosing proper bands of the mel-scale frequency bank. Due to the self-learning ability of SONFIN and RSONFIN, the proposed algorithms avoid the need of empirically determining thresholds and ambiguous rules. The RTF-based RSONFIN algorithm can also find the variation of the background noise level and detect correct word boundaries in the condition of variable background noise level by processing the temporal relations. Our experimental results show that both in the fixed and variable noise-level environment, the algorithms that we proposed achieved higher recognition rate than several commonly used word boundary detection algorithms and reduced the recognition error rate due to endpoint detection.

Keywords: Mel-scale frequency; multiband; spectrum analysis; self-learning ability; neural fuzzy network.

1. Introduction

An important problem in speech processing is to detect the presence of speech in noisy environment, where the word boundary is hard to detect exactly. A major source of errors in the isolated-word systems for the automatic speech recognition 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: impulse noise, fixed-level noise and variable-level 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.[15,22–24] 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. 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.[13] Especially, the zero-crossing rate is very sensitive to the additive noise.

Up to date, several other parameters were proposed such as linear prediction coefficient (LPC), linear prediction error energy[14,21] and pitch information.[7] Although the LPCs are quite successful in modeling vowels,[4] 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 Ref. 13: an energy-based algorithm with automatic threshold adjustment,[15,23] use of pitch information,[7] a noise adaptive algorithm, and a voiced activation algorithm. The reliability of these four algorithms are strongly dependent on the noise condition. In this connection, Junqua *et al.*[13] 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. Based on the TF parameter, a TF-based robust algorithm was proposed in Ref. 13 including noise classification, a refinement procedure and some preset thresholds. The TF-based robust algorithm 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 Refs. 5, 14 and 21, 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 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, for variable-level background noise, this TF-based algorithm usually results in inaccurate detection of the beginning or ending boundaries in 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 fixed- and variable-level background noise conditions. To avoid the problems of the above approaches, this paper first proposes a modified TF parameter and then uses a neural fuzzy network to detect word boundary based on this parameter. By considering multiband analysis of noisy speech

signals, we propose a new robust parameter, called the *adaptive time-frequency* (ATF) and *refined time-frequency* (RTF) parameters. The ATF and RTF parameters represent both the time and frequency features of noisy speech signals and extend the TF parameter from single-band to multiband spectrum analysis based on the mel-scale frequency bank (20 bands). A procedure is proposed such that the ATF and RTF parameters can extract more informative frequency energy than the single-band approach to compensate the time-energy information by *adaptively* choosing proper frequency bands. The ATF and RTF parameters are 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 ATF and RTF parameters, we further propose new word boundary detection algorithms by using neural fuzzy networks for identifying islands of speech signals in noisy environment. The neural fuzzy networks are called self-constructing neural fuzzy inference network (SONFIN), and recurrent self-organizing neural fuzzy inference network (RSONFIN) that we proposed previously in Refs. 11 and 12. The RSONFIN can find the variation of the background noise level and detect correct word boundaries using the temporal relations embedded in the network connections of the memory elements. Due to their self-learning ability, the SONFIN and RSONFIN can always find an economic network size in high learning speed, and avoid the need of empirically determining the number of hidden layers and nodes, by housing the human-like IF-THEN rules and expert knowledge.[16,17] Experimental results also showed that the SONFIN's and RSONFIN's performances were not significantly affected by the size of training set.

This paper is organized as follows. The ATF-based SONFIN algorithm and the structure and function of the SONFIN are briefly introduced in Sec. 2. The RTF-based RSONFIN algorithm is derived in Sec. 3. The performance evaluation and comparisons of the proposed scheme using the ATF and RTF parameters are performed extensively in Sec. 4. Finally, the conclusions of our work are summarized in Sec. 5.

2. ATF-Based SONFIN Algorithm

In this section, we generalize the single-band analysis of the TF parameter to multiband analysis based on mel-scale frequency bank and proposed ATF-based SONFIN algorithm for speech segmentation in fixed noise-level environment.

2.1. *Adaptive time-frequency (ATF) parameter*

For the human ear perceiving speech along a nonlinear scale in the frequency domain,[1] one approach is to use a uniformly space-warped frequency scale, such as the mel scale. The relation between mel-scale frequency and frequency (Hz) is described by the following equation[19]:

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

where mel is the mel-frequency scale and f is in Hz. The filter bank is then designed according to the mel scale where the filters of 20 bands are approximated by simulating 20 triangular band-pass 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 are determined by a constant mel frequency interval by Eq. (1). Consider a given time-domain noisy speech signal, $x_{\text{time}}(m,n)$, representing the magnitude of the nth point of the mth frame. We first find the spectrum, $x_{\text{freq}}(m,k)$, of this signal by Discrete Fourier Transform (128-point DFT):

$$x_{\text{freq}}(m,k) = \sum_{n=0}^{N-1} x_{\text{time}}(m,n) W_N^{kn}, \quad 0 \le k \le N-1,\ 0 \le m \le M-1 \tag{2}$$

$$W_N = \exp(-j2\pi/N) \tag{3}$$

where $x_{\text{freq}}(m,k)$ is the magnitude of the kth point of the spectrum of the mth frame, N is 128 in our system, and M is the number of frames of the speech signal for analysis. We then multiply the spectrum $x_{\text{freq}}(m,k)$ 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:

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

where i is the filter band index, k is the spectrum index, m is the frame number, and M is the number of frames for analysis.

In order to remove some undesired impulse noise in Eq. (4), 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 background noise, Noise_freq, to get the energy of almost pure speech signal, $X(m,i)$. For illustration, the smoothed and normalized frequency energies of a clean speech signal, $X(m,i)$ in Eq. (6), for 20 bands and 100 frames are shown in Fig. 1. The energy of background noise is estimated by averaging the frequency energy of the first five frames of the recording:

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

With the smoothed and normalized energy of the ith band of the mth frame, $X(m,i)$, we can calculate the total energy of the almost pure speech signal at the

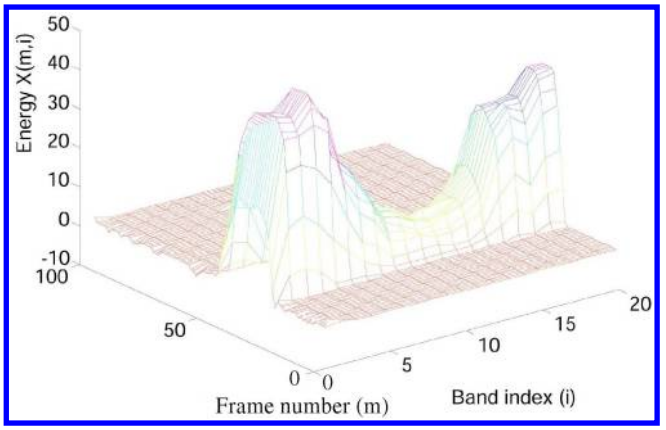


Fig. 1. Multiband spectrum analysis of the speech signal with smoothed and normalized frequency energies, $X(m,i)$, on 20 frequency bands.

ith band as $E(i)$:

$$E(i) = \sum_{m=0}^{M-1} |X(m,i)| \,. \tag{7}$$

Our goal is to select some useful bands having the maximum word signal information, it is obvious that $E(i)$ in Eq. (7) is a good indicator since the band with higher $E(i)$ contains more pure speech information. We sort $E(i)$ in descending order with $I(i)$ as the index of the corresponding frequency band, $i = 1, 2, \ldots 20$. Let $P(i) = E(I(i))$, that is, $P(1) = E(I(1)) = \max\{E(i)\}$, and $P(20) = E(I(20)) = \min\{E(i)\}$.

Obviously, larger background noise will add more noise component into each band, and thus reduce each $E(i)$. Thus, the number of useful bands decreases as the energy of background noise increases. We denote the number of bands useful for producing reliable frequency energy as N_a. Large N_a should be used at high SNR because most bands are corrupted seriously by the additive noise. There are two factors affecting the selection of useful bands, SNR and noise characteristics. The effects of these two factors can be detected by the total frequency energy $E(i)$ in Eq. (7).

We now propose a way to choose the number of useful bands adaptively for extracting helpful frequency information. More precisely, after ordering the band indexes according to their total frequency energy, we wished to decide the number N_a such that the first N_a bands $(I(1), I(2), \ldots, I(N_a))$ can produce helpful frequency energy. At first, we observed from our experiments that the first 18 bands (after ordering) could provide the maximum improvement for word boundary detection in clean environment. Little improvement was observed with the addition of the other two bands. We also observed that one or two bands only cannot give helpful frequency information in our test cases. Hence, we bound the N_a values

Table 1. Experimental statistics on the average number of bands whose $E(i)$ satisfies some thresholds under different noise conditions and SNRs.

	Noise			
SNR	White	Babble	Cockpit	Factory
Clean	16.8	16.8	16.8	16.8
20 dB	13.2	16.4	14.7	15.5
15 dB	11.0	16.4	13.2	15.2
10 dB	9.1	15.4	10.7	13.4
5 dB	7.6	15.0	8.1	10.4
0 dB	6.0	10.1	5.4	7.3

between 3 and 18 for the noisiest and clean environments, respectively. Within the range $(3, 18)$, N_a is tuned adaptively according to the strength of background noise; higher noise level should lead to smaller N_a value as observed in Table 1. To obtain a reliable tuning rule for N_a, we first observed from our experiments that the average frequency energy of background noise, Noise_freq [see Eq. (6)], is 83 in clean environment, and is 93 at a low SNR value (5 dB). We set the corresponding numbers of useful bands to be 18 and 3 for these two extreme cases, respectively. For computation simplicity, we assume that the relation between N_a and Noise_freq is linear. With the above experimental observations and assumption, we can derive the tuning rule for N_a as follows:

$$\frac{N_a - 18}{\text{Noise\_freq} - 18} = \frac{18 - 3}{83 - 93}\,, \quad 3 \le N_a \le 18\,, \quad N_a \text{ is an integer}\,. \tag{8}$$

Rewriting the above result into a general form, we have

$$N_a = \lfloor A \times \text{Noise\_freq} + B \rfloor\,, \quad 3 \le N_a \le 18\,, \quad A = -1.5 \quad \text{and} \quad B = 142.5 \tag{9}$$

where $\lfloor\ \rfloor$ is a function used to denote the rounding to nearest integer operation, and A and B are constants determining the slope and offset, respectively.

With the number of useful bands, N_a, decided by Eq. (9), we then sum the total energies of the first N_a bands (after ordering) to get the final frequency energy, $F(m)$, of frame m:

$$F(m) = \sum_{i=1}^{Na} X(m, I(i))\,. \tag{10}$$

The proposed adaptive time-frequency (ATF) parameter of the mth frame is the result obtained after smoothing the sum of the frequency energy $F(m)$ in Eq. (10) and time energy $T(m)$:

$$\text{ATF}(m) = \text{SMOOTHING}(T(m) + cF(m)) \tag{11}$$

where SMOOTHING is performed by a three-point median filter as in Eq. (5), constant c is a proper weighting factor, and 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:

$$x_{\text{rms}}(m) = \log \sqrt{\frac{\sum_{n=0}^{L-1} x_{\text{time}}^2(m,n)}{L}} \tag{12}$$

$$\hat{x}_{\text{rms}}(m) = \frac{x_{\text{rms}}(m-1) + x_{\text{rms}}(m) + x_{\text{rms}}(m+1)}{3} \tag{13}$$

$$\begin{aligned} T(m) &= \hat{x}_{\text{rms}}(m) - \text{Noise\_time} \\ &= \hat{x}_{\text{rms}}(m) - \frac{\sum_{m=0}^{4} \hat{x}_{\text{rms}}(m)}{5} \end{aligned} \tag{14}$$

where L is the length of the frame, which is 120 (15 ms) in our system. The procedure to calculate the ATF parameter is illustrated in Fig. 2(a). The details of the block with label "Select N_a useful bands to produce frequency energy" of this figure are shown in Fig. 2(b).

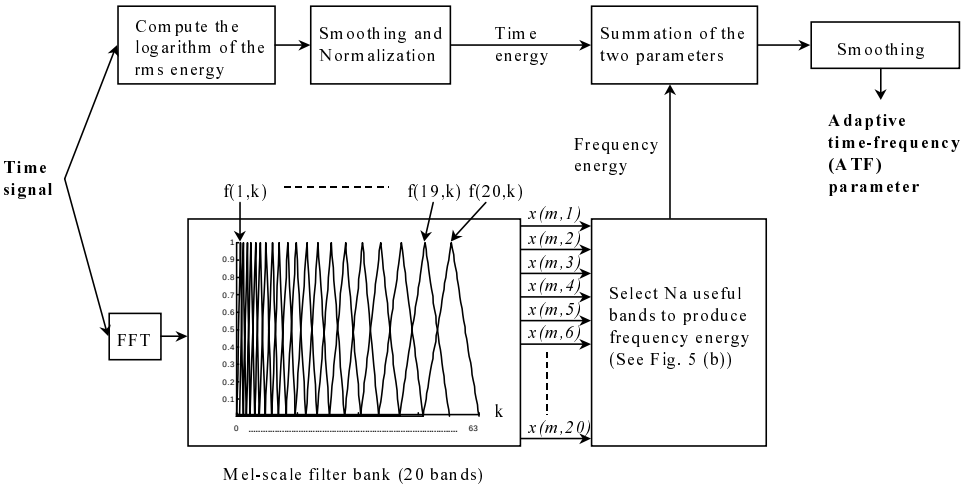
2.2. *Self-constructing neural fuzzy inference network (SONFIN)*

The neural fuzzy network that we used for word boundary detection is called the self-constructing neural fuzzy inference network (SONFIN) that we proposed previously in Ref. 11. The SONFIN is a general connectionist model of a fuzzy logic system, which can find its optimal structure and parameters automatically. The structure of the SONFIN is shown in Fig. 3(a). This six-layered network realizes a fuzzy model of the following form

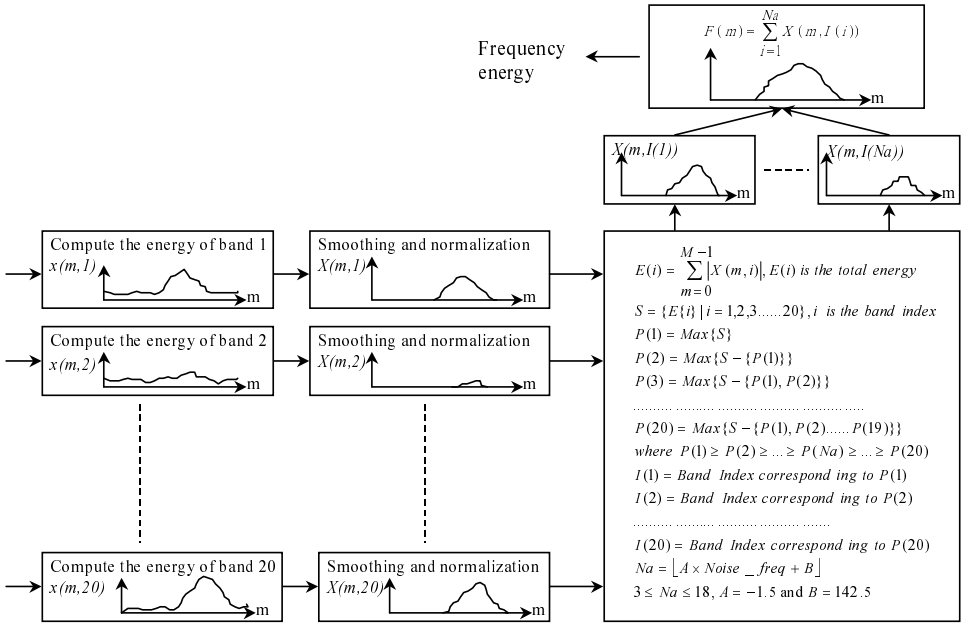
$$\begin{aligned} &\text{Rule } i: \text{ IF } x_1 \text{ is } A_{i1} \text{ and} \cdots \text{and } x_n \text{ is } A_{in} \\ &\qquad \text{THEN } y \text{ is } m_{0i} + a_{ji}x_j + \cdots, \end{aligned} \tag{15}$$

where A_{ij} is a fuzzy set, m_{0i} is the center of a symmetric membership function on y, and a_{ji} is a consequent parameter. It is noted that unlike the traditional TSK model[17,25,27] where all the input variables are used in the output linear equation, only the significant ones are used in the SONFIN; i.e. some a_{ji}s in the above fuzzy rules are zero. We shall next describe the functions of the nodes in each of the six layers of the SONFIN.

Each node in Layer 1, which corresponds to one input variable, only transmits input values to the next layer directly. Each node in Layer 2 corresponds to one linguistic label (small, large, etc.) of one of the input variables in Layer 1. In other words, the membership value which specifies the degree to which an input value belongs to a fuzzy set is calculated in Layer 2. A node in Layer 3 represents one fuzzy logic rule and performs precondition matching of a rule. The number of nodes in Layer 4 is equal to that in Layer 3, and the result (firing strength) calculated in Layer 3 is normalized in this layer. Layer 5 is called the consequent layer. Two types of nodes are used in this layer, and they are denoted as blank and shaded circles in Fig. 3(a), respectively. The node denoted by a blank circle (blank node) is



(a)



(b)

Fig. 2. (a) Flowchart for computing the ATF parameter. (b) Adaptive band selection procedure in (a) for computing frequency energy.

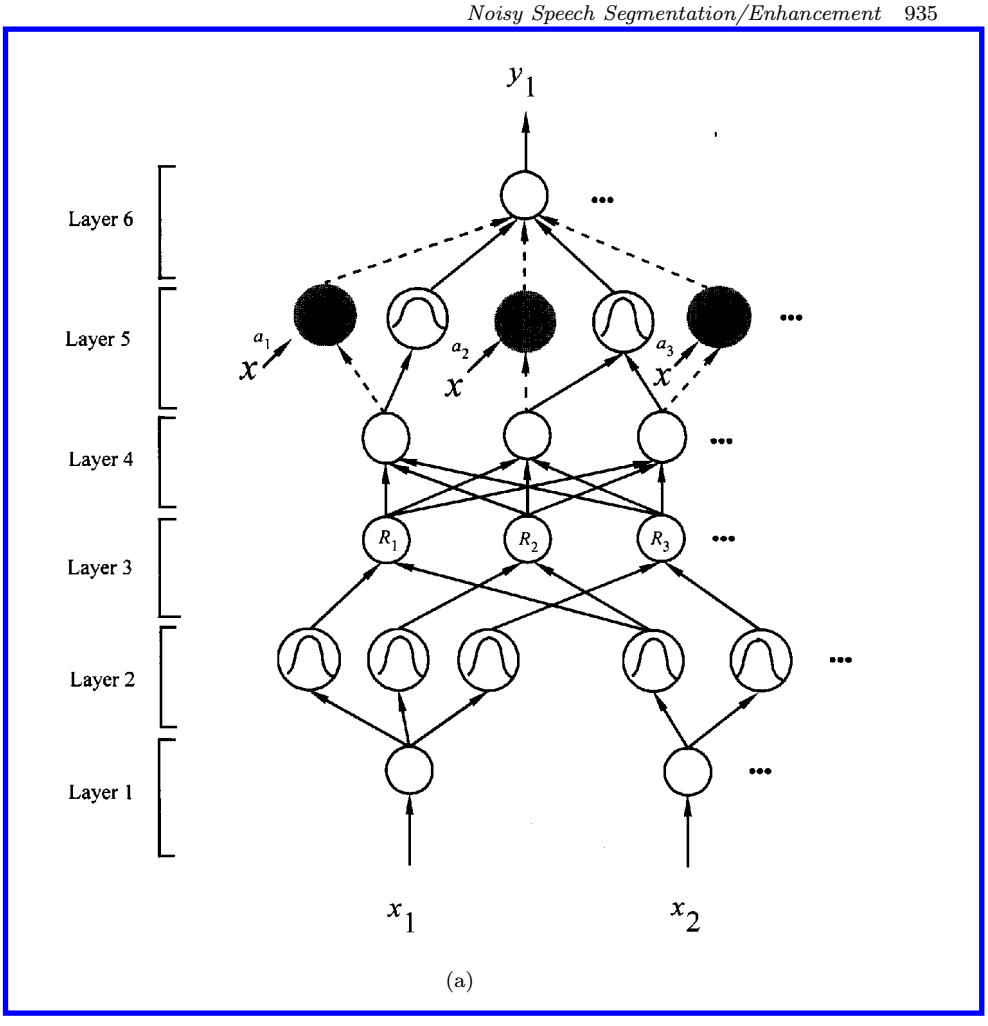


Fig. 3. (a) Network structure of the SONFIN. (b) Flowchart of the SONFIN-based word boundary detection procedure.

the essential node representing a fuzzy set of the output variable. The shaded node is generated only when necessary. One of the inputs to a shaded node is the output delivered from Layer 4, and the other possible inputs (terms) are the selected significant input variables from Layer 1. Combining these two types of nodes in Layer 5, we obtain the whole function performed by this layer as the linear equation on the THEN part of the fuzzy logic rule in Eq. (15). Each node in Layer 6 corresponds to one output variable. The node integrates all the actions recommended by Layer 5 and acts as a defuzzifier to produce the final inferred output.

Two types of learning, structure and parameter learning, are used concurrently for constructing the SONFIN. The structure learning includes both the precondition and consequent structure identification of a fuzzy IF-THEN rule. For the parameter learning, based upon supervised learning algorithms, the parameters of the linear equations in the consequent parts are adjusted to minimize a given cost function. The SONFIN can be used for normal operation at any time during the learning process without repeated training on the input–output patterns when online operation is required. There are no rules in the SONFIN initially, and 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) Optimal consequent structure identification, (D) Parameter identification. Processes A–C belong to the structure learning phase and process D belongs to the parameter learning phase. The details of these learning processes can be found in Ref. 11.

2.3. *ATF-based SONFIN algorithm for word boundary detection*

The procedure of using the SONFIN for word boundary detection is illustrated in Fig. 3(b). The input feature vector of the SONFIN is a combination of the average energy of background noise (Noise_time), adaptive time-frequency (ATF) parameter and zero-crossing rate. 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 the SONFIN. Here the noise energy, Noise_time, as in Eq. (14), is the average of the logarithm of the rms energy on the first five frames of "relative silence" at the beginning of the recording. Before entering the SONFIN, the three input parameters are normalized to be in $[0, 1]$. For each input vector (corresponding to a frame), the output of SONFIN indicates whether the corresponding frame is a word signal or noise. For this purpose, we used two (output) nodes in Layer 6 of the SONFIN, where the output vector of $(1, 0)$ stands for word signal, and $(0, 1)$ for noise.

The SONFIN was trained by a set of 80 training patterns, which were randomly selected from four noise conditions with different SNRs. These training patterns are classified as word signal or noise by using waveform, spectrum displays and audio output. Among the 80 training patterns, 40 patterns are from word sound category with the desired SONFIN output vector being $(1, 0)$, and the other 40 from noise category with the desired SONFIN output vector being $(0, 1)$. We usually used the

frames around the word-noise transition area as the training patterns, because these ambiguous training patterns lead the SONFIN to a more accurate word boundary in noisy environment. After training, there were only 14 rules generated in the SONFIN. As shown in Fig. 3(b), the outputs of the SONFIN are processed by a decoder. The decoder processes the SONFIN's output vector $(1, 0)$ as value 100 standing for word signal, and $(0, 1)$ as value 0 standing for noise. In addition, we let the output waveform of the decoder pass through a three-point median filter to eliminate the undesired "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 send part of the original signal corresponding to the allocated word-signal island to our word recognition system.

3. RTF-Based RSONFIN Algorithm

In this section, we propose a new *refined time-frequency* (RTF) parameter obtained by smoothing the sum of the time energy and frequency energy, where the frequency energy is contributed by several adaptively chosen frequency bands.

3.1. *Refined time-frequency (RTF) parameter*

Based on the discussion and illustrations in Sec. 2.1, we now propose a way to adaptively extract helpful frequency information from word signals. From Eqs. (1)–(6), we adopt the maximum $X(m, i)$ to get the final frequency energy, $F(m)$, of frame m:

$$F(m) = \max[X(m, i)]_{i=1,2,\ldots,20} . \tag{16}$$

The proposed refined time-frequency (RTF) parameter of the mth frame is the result obtained after smoothing the sum of the frequency energy $F(m)$ in Eq. (16) and time energy $T(m)$:

$$\mathrm{RTF}(m) = \mathrm{SMOOTHING}(T(m) + cF(m)) \tag{17}$$

where SMOOTHING is performed by a three-point median filter as in Eq. (5), and the constant weighting factor c is optimally set as 0.8 in our experiments. 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:

$$x_{\mathrm{rms}}(m) = \log \sqrt{\frac{\sum_{n=0}^{L-1} x_{\mathrm{time}}^2(m, n)}{L}} \tag{18}$$

$$\hat{x}_{\mathrm{rms}}(m) = \frac{x_{\mathrm{rms}}(m-1) + x_{\mathrm{rms}}(m) + x_{\mathrm{rms}}(m+1)}{3} \tag{19}$$

$$\begin{aligned} T(m) &= \hat{x}_{\mathrm{rms}}(m) - \mathrm{Noise\_time} \\ &= \hat{x}_{\mathrm{rms}}(m) - \frac{\sum_{m=0}^{4} \hat{x}_{\mathrm{rms}}(m)}{5} \end{aligned} \tag{20}$$

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. 4(a). The details of the block with label "Produce frequency energy" of this figure are shown in Fig. 4(b).

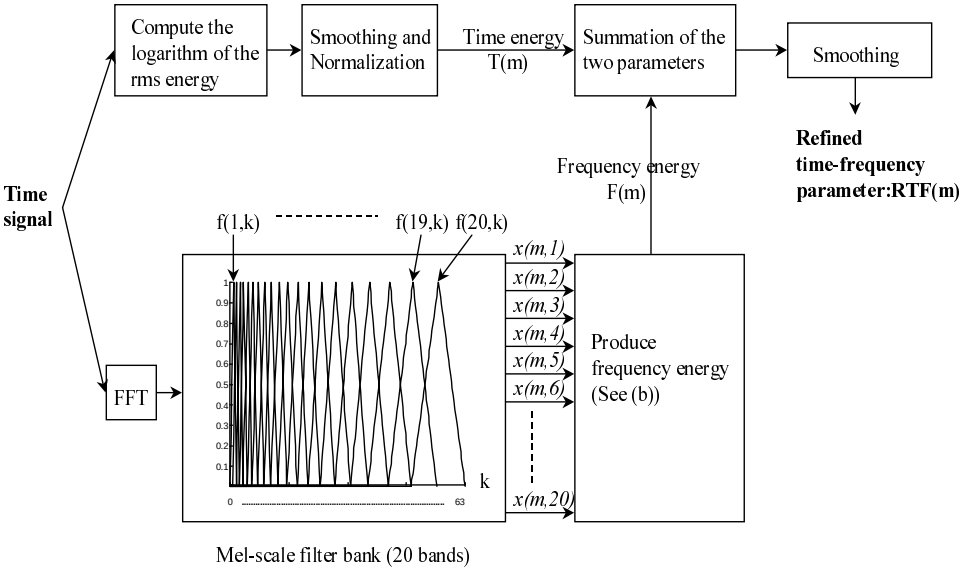


Fig. 4. (a) Flowchart for computing the RTF parameter. (b) Procedure for producing the frequency energy in (a).

3.2. *Recurrent self-organizing neural fuzzy inference network (RSONFIN)*

Based on this RTF parameter, we further propose a new recurrent self-organizing neural fuzzy inference network (RSONFIN) for word boundary detection that we proposed previously in Ref. 12. 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 RSONFIN realizes the following dynamic fuzzy reasoning[6]:

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

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

The structure of the RSONFIN is shown in Fig. 5. It is a five-layered neural fuzzy network embedded with dynamic feedback connections (the feedback layer in Fig. 5) that bring the temporal processing ability into a feedforward neural fuzzy network. The following describes the function of each layer, 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.

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

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: one from Layer 2 and the other 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,

$$a^{(3)} = a^{(6)} \cdot \prod_i u_i^{(3)} = a^{(6)} \cdot e^{-[D_i(\mathbf{x}-\mathbf{m}_i)]^T[D_i(\mathbf{x}-\mathbf{m}_i)]} \tag{21}$$

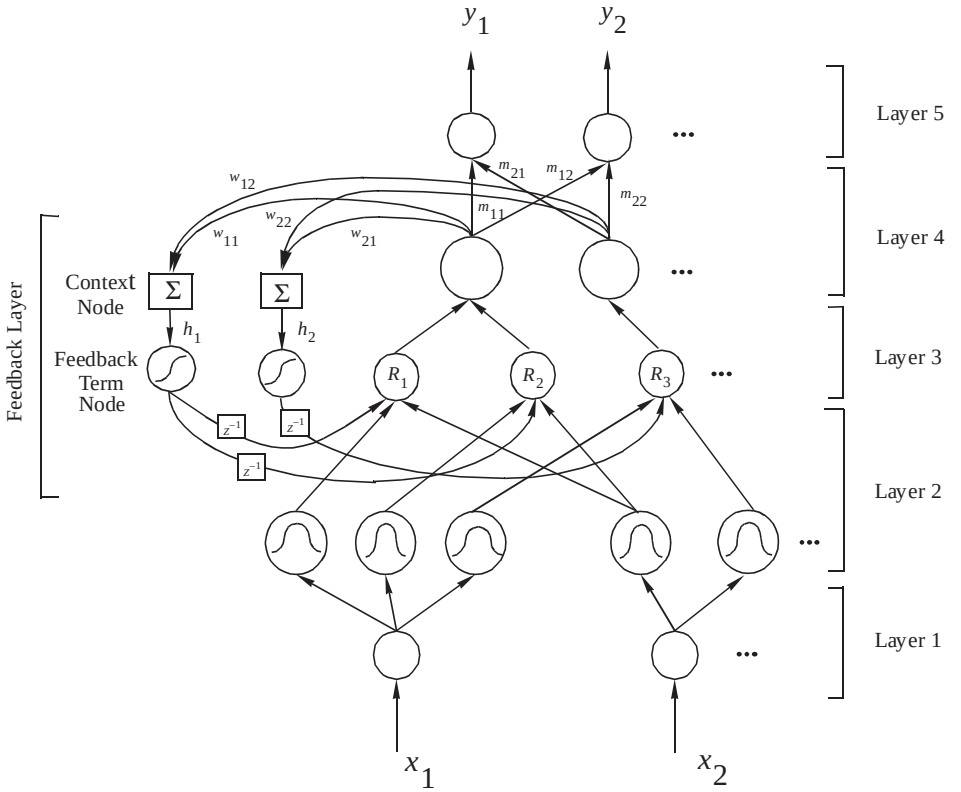


Fig. 5. Structure of the Recurrent Self-Organizing Neural Fuzzy Inference Network (RSONFIN).

where $D_i = \text{diag}(1/\sigma_{i1}, 1/\sigma_{i2}, \ldots, 1/\sigma_{in})$, $\mathbf{m}_i = (m_{i1}, m_{i2}, \ldots, m_{in})^T$, and $a^{(6)}$ is the output of the feedback term node described in the feedback layer part. 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 LMOM (local mean of maximum) defuzzification operation.[3] The function of each output term node performs the following fuzzy OR operation to integrate the fired rules which have the same consequent part.

$$a^{(4)} = \sum_i u_i^{(4)} . \tag{22}$$

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 function performed in this layer is

$$y_j = a^{(5)} = \frac{\sum_i u_i^{(5)} \hat{m}_{ji}}{\sum_i u_i^{(5)}} \tag{23}$$

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. 5, two types of nodes are used in this layer, the square node named as *context node* and the circle node named as *feedback term node*, where each context node is associated with a feedback term node. The context node functions as a defuzzifier,

$$h_j = \sum_i a_i^{(4)} w_{ji} \tag{24}$$

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. In Eq. (24), the simple weighted-sum is calculated.[10,26] 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.[10,28] With the chosen membership function, the feedback term node evaluates the output by

$$a^{(6)} = \frac{1}{1 + e^{-h_i}} \,. \tag{25}$$

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.

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 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 input/output linguistic nodes) in the RSONFIN initially. They are created dynamically as learning proceeds upon receiving online incoming training data by

performing the four learning processes simultaneously: (A) Input/output space partitioning, (B) Construction of fuzzy rules, (C) Feedback structure identification, (D) Parameter identification. The processes A–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 Ref. 12.

3.3. *RTF-based RSONFIN for word boundary detection*

As mentioned in Sec. 2.3, a procedure of using the RSONFIN for word boundary detection in variable background noise level condition is illustrated in Fig. 6. 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. The input nodes in Layer 1 of RSONFIN consist of the Noise_time, RTF parameter and ZCR. We use two output nodes in Layer 5 of RSONFIN standing for word signal and noise, separately.

The RSONFIN was trained by a speech waveform with 15 seconds. 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. 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. After training, as shown in Fig. 6, the outputs of RSONFIN are processed by a decoder and then passed 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 the original signal corresponding to the allocated word-signal island as the word signal, and the other ones as the background noise.

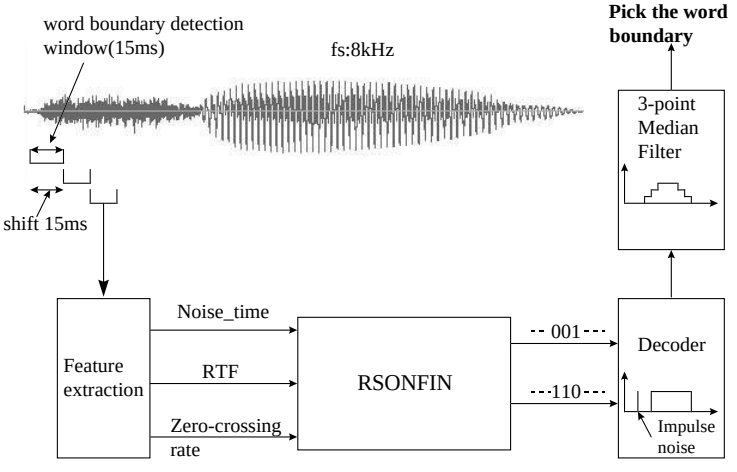


Fig. 6. The RTF-based RSONFIN algorithm for automatic word boundary detection.

4. Experimental Results

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 cepstrum (Modified TDC — MTDC)[2,8,18,20] is used as the speech feature. A Gaussian clustering algorithm is used in the classifier. In the training phase, the training was done on clean speech pronounced in a clean environment (without background noise). 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 Ref. 18. The frame window used for obtaining the MTDC features is 30 ms in length, and is with 15 ms overlapping between two frames in the recognition procedure. And in the word boundary detection procedure, the frame length is set to be 15 ms in order to get a more accurate endpoint location. The speech data used for our experiments 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. The noise signals are taken from the noise database provided by the NATO Research Study Group on Speech Processing (RSG.10) NOISE-ROM-0.[29] Among the noisy database, we take four typical types of noise: multitalker babble noise, cockpit noise, noise on the floor of car factory and white noise.

To set up the noisy speech database for testing, we added the prepared noisy signals to the recorded speech signals for testing with different signal-to-noise-ratios (SNRs) including 0 dB, 5 dB, 10 dB, 15 dB, 20 dB and ∞ dB. The duration of each utterance is about one second (including silence). A total of 600 utterances were used in our experiments. 300 utterances are in the condition of increasing background noise level, and 300 utterances are in the condition of decreasing background noise level.

4.1. *Evaluations of ATF-based SONFIN algorithm*

In this section, we show the performance of the ATF-based SONFIN algorithm, the ATF-based robust algorithm, TF-based robust algorithm, TF without robust algorithm, and the performance of hand-labeling (i.e. manually determined boundaries). The recognition rates of the three algorithms for four types of noise with different SNRs are shown in Fig. 7. We also examined the recognition error rates averaged across the four noise conditions due to word boundary detection as a function of SNRs as shown in Fig. 8. These results show that, by using the same three parameters (Noise_time, ATF and zero-crossing rate), the SONFIN outperforms the robust algorithm by about 2% in recognition rate. As a total, the ATF-based SONFIN had higher recognition rate than the TF-based robust algorithm in Ref. 13 by about 5%. Also, the ATF-based SONFIN reduced the recognition error

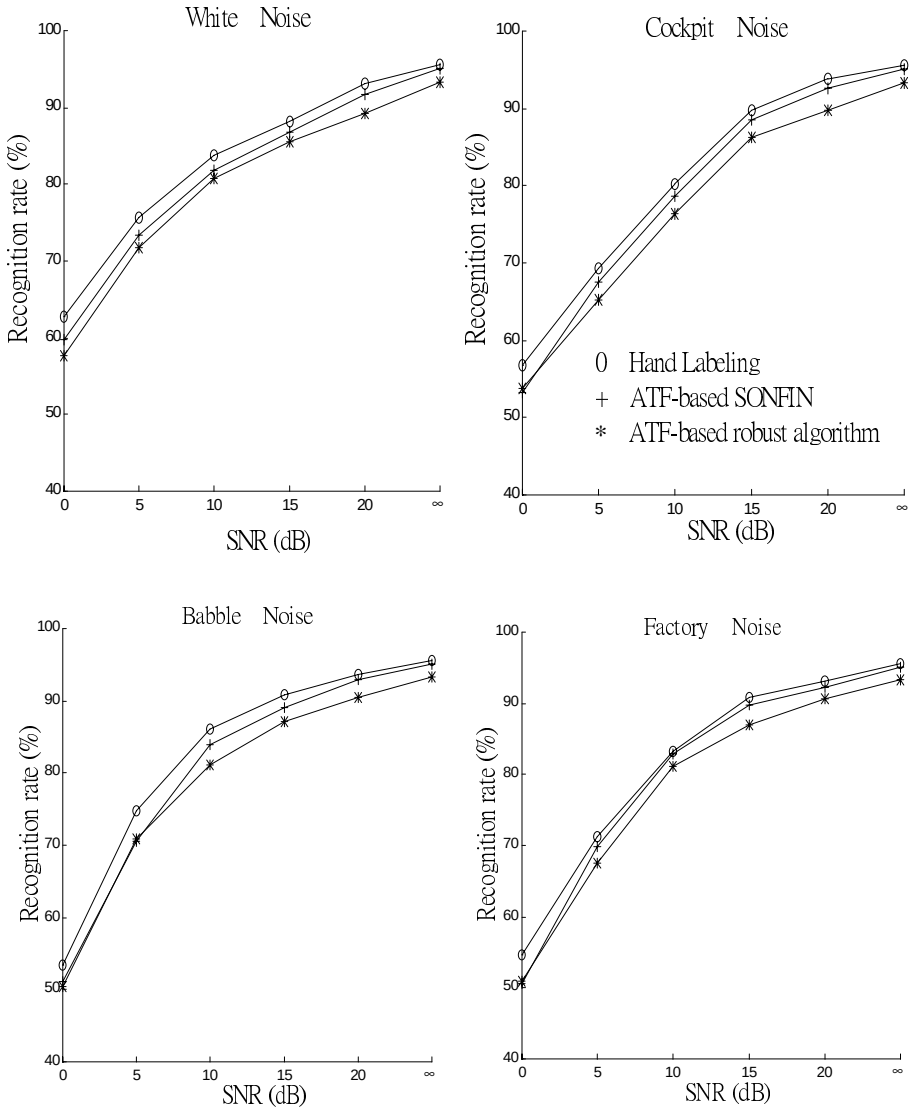


Fig. 7. Recognition rates of three word boundary detection algorithms (ATF-based SONFIN, ATF-based robust algorithm, and hand labeling) in an MTDC-based recognition system across six SNRs and four noise conditions.

rate due to endpoint detection to about 10%, compared to about 20% obtained with the ATF-based robust algorithm, about 30% obtained with the TF-based robust algorithm, about 40% obtained with the TF without robust algorithm, and about 50% obtained with the modified version of the Lamel *et al.* algorithm.[15,23] We also found that the SONFIN could approach the result of hand labeling, which is usually considered as the optimum result for reference.

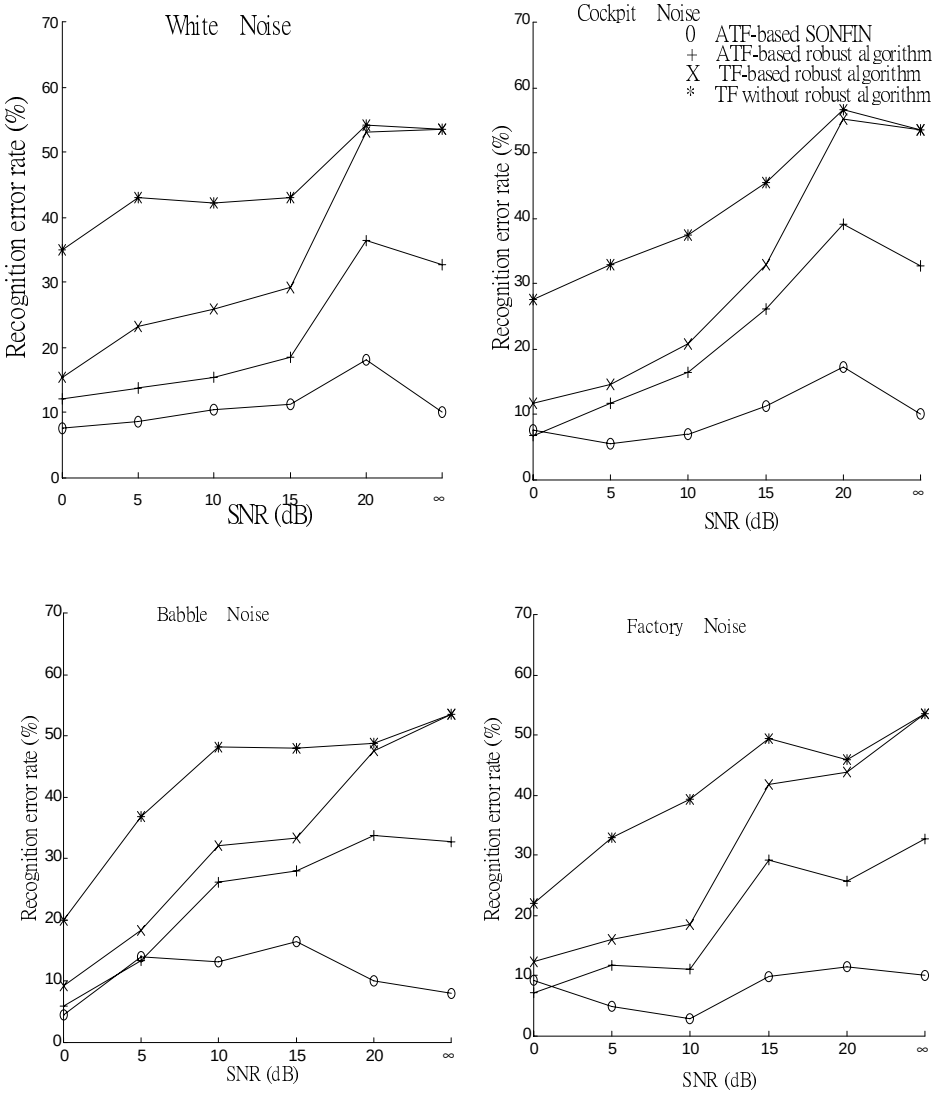


Fig. 8. Recognition error rates of four word boundary detection algorithms (ATF-based SONFIN, ATF-based robust algorithm, TF-based robust algorithm, and TF without robust algorithm) in an MTDC-based recognition system across six SNRs and four noise conditions.

After learning, the SONFIN generated ten membership functions (ten *fuzzy* categories) in the input dimension (variable) "Noise_time" representing the energy of environment noise [see Eq. (14)]. The SONFIN also automatically classified the other two features, ATF and zero-crossing rate, into 11 and 14 *fuzzy* categories by using only 14 rules. Each input node in the SONFIN is only connected to its related rule nodes through its term nodes resulting in a small number of weights to be tuned.

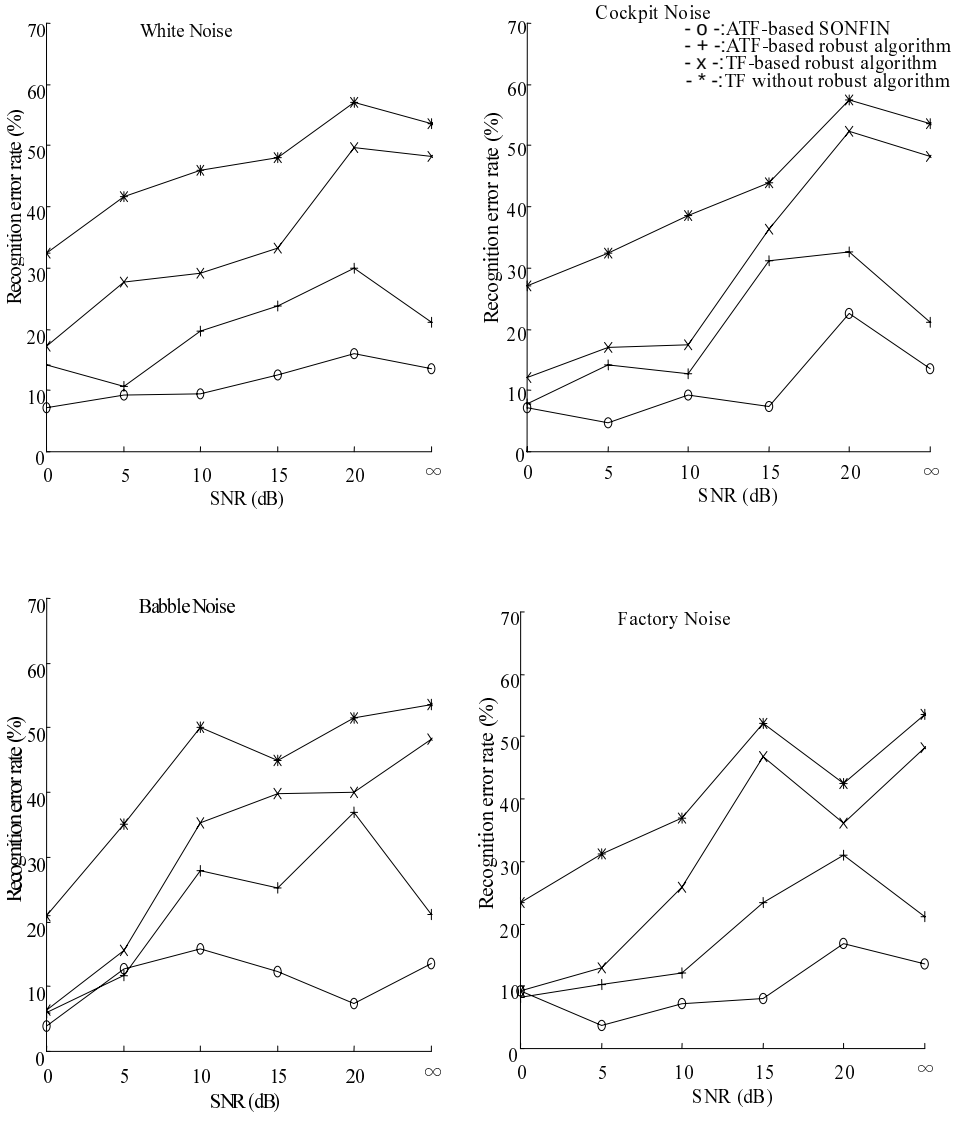


Fig. 9. Recognition error rates of four word boundary detection algorithms (ATF-based SONFIN, ATF-based robust algorithm, TF-based robust algorithm, and TF without robust algorithm) in an MFCC-based HMM recognition system across six SNRs and four noise conditions.

In order to see the performance of our algorithms on other speech features and recognizer, we replace the MTDC-based recognizer used in the previous experiments by the MFCC (mel-frequency cepstral coefficient)-based HMM recognizer with temporal filter in another set of experiments, where the temporal filter is used to remove the noise components in the feature extraction phase. The number of coefficients of each frame used in this HMM recognizer is 26, including MFCCs,

energy, delta MFCCs and delta energy, and the analysis order is 24. Each Mandarin digits is modeled by a 5-state, left-to-right, continuous density HMM. In the HMM, each state is split into two streams, and a mix of Gaussian density with two mixture components in each stream is assigned to each state observation probability. The recognition error rates averaged across the four noise conditions due to word boundary detection as a function of SNRs are shown in Fig. 9. The results show that the conclusions on the good performance of the proposed word boundary detection algorithms still hold on the common speech features and recognizer.

Although the SONFIN has the advantages of small network size, high learning speed, and high learning accuracy, its merits are obtained at the expense of longer CPU time.

4.2. *Evaluations of RTF-based RSONFIN algorithm*

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. Each experiment consists of 60,000 samples and the SNR is 10 dB. There are totally seven words in the recording interval, which are Mandarin digits of "1, 2, 3, 4, 5, 6, 7".

4.2.1. *Increasing background noise level*

In Fig. 10, the word boundaries detected by hand labeling *in clean environments* are shown by dotted lines. The noise in the last half segment of recording interval is larger than the noise in the first half segment. Word boundaries detected by the TF-based algorithm are shown by solid lines in Fig. 10(a), where two word segments are found. The first word segment is determined properly but the other six word boundaries are missing. The major reason for this error is that the TF-based algorithm cannot detect the variation of the background noise level and does not decide proper thresholds to find word boundaries. The word boundaries detected by the TF-based RSONFIN algorithm are shown by solid lines in Fig. 10(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. 10(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.

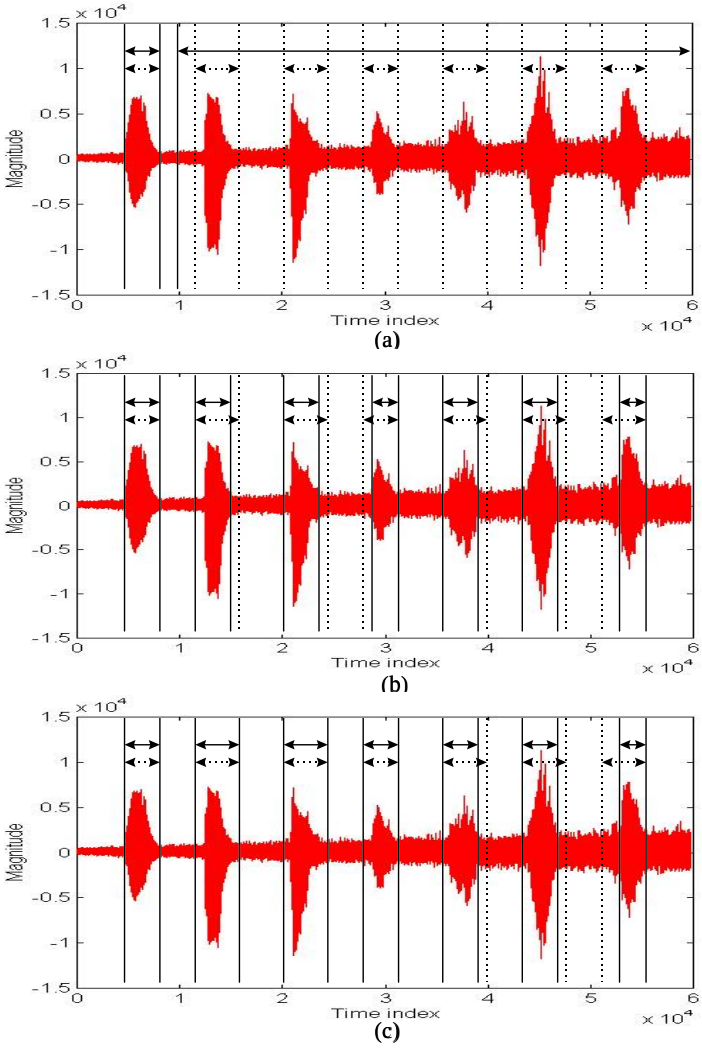


Fig. 10. Speech waveform recorded in additive increasing-level white noise including 60,000 samples with the SNR being 10 dB. The word boundaries detected by hand labeling in clean environments are shown by dotted lines. (a) The word boundaries detected by the TF-based algorithm are shown by solid lines,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

4.2.2. *Decreasing background noise level*

In Fig. 11, 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. 11(a), where only five word segments are found, and the fourth and fifth words are missing. Although

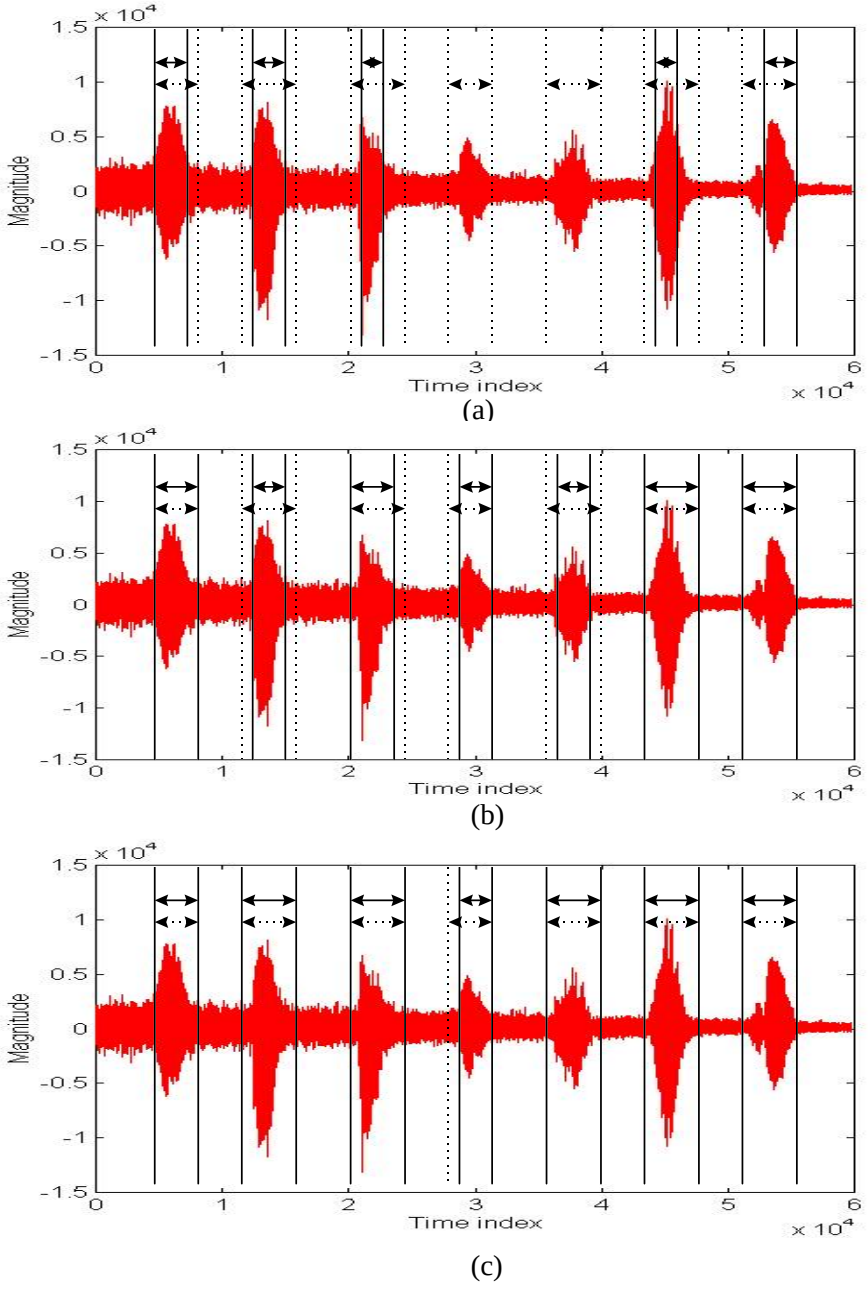


Fig. 11. Speech waveform recorded in additive decreasing-level white noise including 60,000 samples with the SNR being 10 dB. The word boundaries detected by hand labeling in clean environments are shown by dotted lines. (a) The word boundaries detected by the TF-based algorithm are shown by solid lines, and we notice that the fourth and fifth words are not detected at all. (b) The word boundaries detected by the TF-based RSONFIN algorithm (17 rules) are shown by solid lines. (c) The word boundaries detected by the RTF-based RSONFIN algorithm (10 rules) are shown by solid lines.

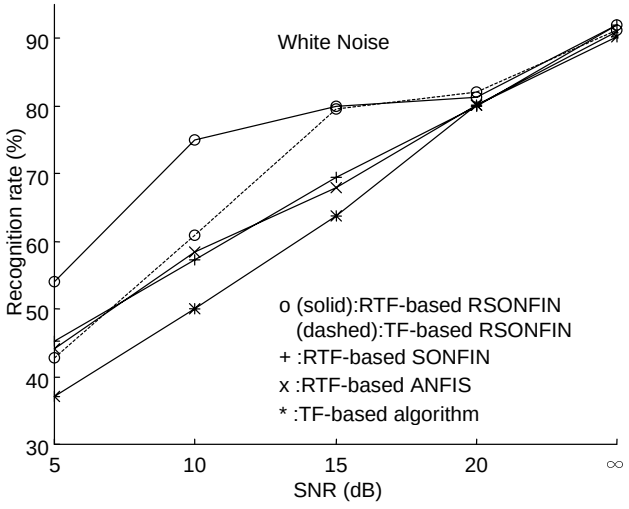
the seventh word is detected, the beginning part of this word is missing. The word boundaries detected by the TF-based RSONFIN algorithm are shown by solid lines in Fig. 11(b), where seven word segments are found. This algorithm can really sense the variation of the background noise level and detect 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. 11(c). These word boundaries are more accurate than those detected by the TF-based RSONFIN algorithm.

4.2.3. *Speech recognition in variable background noise level conditions*

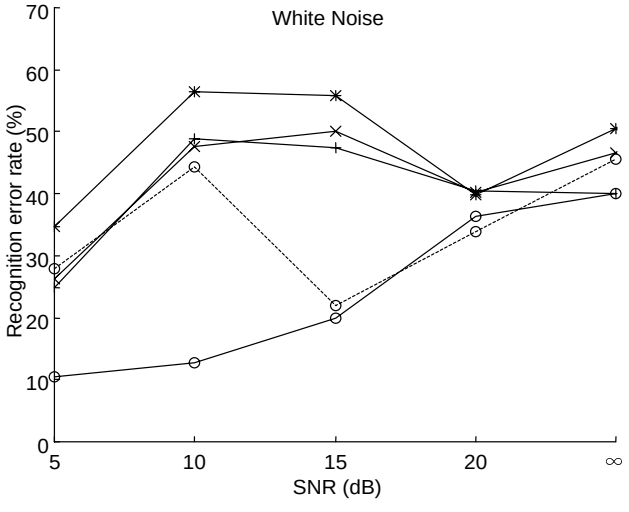
We also compare the performance of RSONFIN to that of two other neural fuzzy networks. They are SONFIN and the adaptive-network-based fuzzy inference system (ANFIS).[9] As a result, there are five word boundary detection algorithms used for testing in the following. The recognition rates of the five algorithms for added white noise with different SNRs are shown in Fig. 12(a). The resulting recognition error rates of the five algorithms are given in Fig. 12(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 TF-based algorithm by about 4%. With the temporal relations captured and embedded in the RSONFIN, the TF-based RSONFIN algorithm outperforms 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 Ref. 13 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 Ref. 13.

5. Conclusions

In this paper, we have proposed the reliable parameters in noisy environment including *adaptive time-frequency* (ATF) and *refined time-frequency* (RTF) parameters. Comparative study has shown that the ATF and RTF parameters are very beneficial for several SNRs and noise conditions (including clean speech, for which very good results were obtained). The ATF-based SONFIN and RTF-based RSONFIN algorithms have been tested and performed well in fixed and variable noise-level conditions, respectively. Our experiments showed that the proposed scheme (ATF-based SONFIN) achieved higher recognition rate by about 2% than the ATF-based robust algorithm, and thus by about 5% than the TF-based robust algorithm. On the other performance index, the ATF-based SONFIN reduced the recognition error



(a)



(b)

Fig. 12. (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.

rate due to endpoint detection to about 10%, compared to about 20% obtained with the ATF-based robust algorithm. Based on the RTF parameter, 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%,

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 the same condition.

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

(1) The proposed ATF and RTF parameters can extract both the time and frequency features of noisy speech signals through multiband spectrum analysis, and can extract more informative frequency energy than the TF parameter by *adaptively* choosing proper frequency bands.
(2) The recurrent property of the RSONFIN makes it more suitable for dealing with temporal problems, thus 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 SONFIN and 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 SONFIN and RSONFIN, our proposed algorithms avoid the need of empirically determining ambiguous decision rules in normal word boundary detection algorithms. Also, since the SONFIN and RSONFIN house 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.

References

1. J. B. Allen, "Cochlear modeling," *IEEE Acoust. Speech Sign. Process. Mag.* **2** (1985) 3–29.
2. Y. Ariki, S. Mizuta and T. Sakai, "Spoken-word recognition using dynamic features analyzed by two-dimensional cepstrum," *IEE Proc.* **136**, 2 (1989) 133–140.
3. H. R. Berenji and P. Khedkar, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. Neural Networks* **3**, 5 (1992) 724–740.
4. J. R. Deller, J. G. Proakis and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan, NY, 1993.
5. T. Ghiselli-Crippa and A. El-Jaroudi, "A fast neural net training algorithm and its application to voiced-unvoiced-silence classification of speech," *ICASSP '91* **1** (1991) 441–444.
6. V. Gorrini and H. Bersini, "Recurrent fuzzy systems," *Proc. IEEE Int. Conf. Fuzzy Systems*, Vol. 1, 1994, pp. 193–198.
7. M. Hamada, Y. Takizawa and T. Norimatsu, "A noise robust speech recognition," *Proc. ICSLP '90*, 1990, pp. 893–896.
8. H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.* **2**, 4 (1994) 578–589.
9. J. S. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. Neural Networks* **3**, 5 (1992) 714–723.

10. J. S. R. Jang and C. T. Sun, "Functional equivalence between radial basis function networks and fuzzy inference system," *IEEE Trans. Neural Networks* **4**, 1 (1993) 156–159.
11. C. F. Juang and C. T. Lin, "An online self-constructing neural fuzzy inference network and its application," *IEEE Trans. Fuzzy Syst.* **6**, 1 (1998) 12–32.
12. C. F. Juang and C. T. Lin, "A recurrent self-organizing neural fuzzy inference network," *IEEE Trans. Neural Networks* **10**, 4 (1999) 828–845.
13. J. C. Junqua, B. Mak and B. Reaves, "A robust algorithm for word boundary detection in the presence of noise," *IEEE Trans. Speech Audio Process.* **2** (1994) 406–412.
14. S. J. Kia and G. G. Coghill, "A mapping neural network and its application to voiced-unvoiced-silence classification," *Proc. First New Zealand Int. Two-Stream Conf. Artificial Neural Networks and Expert Systems*, 1993, pp. 104–108.
15. L. F. Lamel, L. R. Rabiner, A. E. Rosenberg and J. G. Wilson, "An improved endpoint detector for isolated word recognition," *IEEE ASSP Mag.* **29** (1981) 777–785.
16. C. T. Lin, *Neural Fuzzy Control Systems with Structure and Parameter Learning*, World Scientific, 1994.
17. C. T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neural-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall, Englewood Cliffs, NJ, May, 1996.
18. C. T. Lin, H. W. Nein and J. Y. Hwu, "GA-based noisy speech recognition using two-dimensional cepstrum," *IEEE Trans. Speech Audio Process.* **78**, 6 (2000) 664–675.
19. D. O'Shaughnessy, *Speech Communication*, Addision-Wesley, 1987, p. 150.
20. H. F. Pai and H. C. Wang, "A study on two-dimensional cepstrum approach for speech recognition," *Comput. Speech Lang.* **6** (1992) 361–375.
21. Y. Qi and B. R. Hunt, "Voiced-unvoiced-silence classification of speech using hybrid features and a network classifier," *IEEE Trans. Speech Audio Process.* **1** (1993) 250–255.
22. L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell Syst. Tech. J.* **54**, 2 (1975) 297–315.
23. B. Reaves, "Comments on an improved endpoint detector for isolated word recognition," *IEEE Trans. Sign. Process.* **39** (1991) 526–527.
24. M. H. Savoji, "A robust algorithm for accurate endpointing of speech," *Speech Commun.* **8** (1989) 45–60.
25. M. Sugeno and K. Tanaka, "Successive identification of a fuzzy model and its applications to prediction of a complex system," *Fuzzy Sets Syst.* **42**, 3 (1991) 315–334.
26. H. Takagi and I. Hayashi, "NN-driven fuzzy reasoning," *Int. J. Approx. Reas.* **5**, 3 (1991) 191–212.
27. T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst. Man Cybern.* **15**, 1 (1985) 116–132.
28. T. Takagi and M. Sugeno, "Derivation of fuzzy control rules from human operator's control actions," *Proc. IFAC Symp. Fuzzy Information, Knowledge Representation and Decision Analysis*, July 1983, pp. 55–60.
29. 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.* **12** (1993) 247–251.

Chin-Teng Lin received the B.S. degree in control engineering from the National Chiao-Tung University, 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, National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., where he is currently a Professor and Chairman of Electrical and Control Engineering Department. He served as the Deputy Dean of the Research and Development Office of the National Chiao-Tung University from 1998 to 2000.

He is the co-author of Neural Fuzzy System — A Neuro-Fuzzy Synergism to Intelligent System (Prentice Hall), and the author of Neural Fuzzy Control Systems with Structure and Parameter Learning (World Scientific). Dr. Lin has published over 67 journal papers in the areas of soft computing, neural networks, and fuzzy systems, including about 45 *IEEE Transactions* papers.

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 System, Man, Cybernetics Society. Dr. Lin has been the Executive Council Member (Supervisor) of Chinese Automation Association since 1998. He is the Executive Council Member of the Chinese Fuzzy System Association T (CFSAT), from 1995 to 2002. He is the Society President of Chinese Fuzzy Systems Association T since 2002. He is the Chairman of IEEE Robotics and Automation Society, Taipei Chapter since 2000, and the association editor of *IEEE Transactions on Systems, Man, Cybernetics* since 2001. Dr. Lin won the Outstanding Research Award granted by National Science Council (NSC), Taiwan, R.O.C. Since 1997, he won 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 (CIE) in 2000. Dr. Lin was also elected to be one of the 38th Ten Outstanding Young Persons in Taiwan, R.O.C., 2000. Dr. Lin currently serves as the association editor of *IEEE Transactions on Systems, Man, Cybernetics, Part B, IEEE Transactions on Fuzzy Systems*, and the *Journal of Automatica*.

His current research interests are fuzzy systems, neural networks, intelligent control, human-machine interface, image processing, pattern recognition, video and audio (speech) processing, and intelligent transportation system (ITS).

Rui-Cheng Wu received the B.S. degree in nuclear engineering from the National Tsing-Hua University, Taiwan, R.O.C. in 1995, and M.S. degree in control engineering from the National Chiao-Tung University, Taiwan, R.O.C. in 1997. He is currently pursuing the Ph.D. in electrical and control engineering at the National Chiao-Tung University, Taiwan, R.O.C.

His current research interests are audio signal processing, speech recognition/enhancement, fuzzy control, neural networks and linear control.

Gin-Der Wu received the B.S. degree in engineering science from the National Cheng-Kung University, Taiwan, R.O.C. in 1996, and Ph.D. in control engineering from the National Chiao-Tung University, Taiwan, R.O.C. in 2000.

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