International Journal of Electronics Signals and Systems

Volume 4 | Issue 1 Article 2

July 2014

SPEECH RECOGNITION BASED EMBEDDED CONTROL SYSTEM

PARVEZ HASAN
Electronics and Telecommunication Department, Goa College of Engineering, Farmagudi, Goa, India,
parvezhssn@gmail.com

V. K. JOSEPH
Electronics and Telecommunication Department, Goa College of Engineering, Farmagudi, Goa, India,
vkj@gec.ac.in

SPEECH RECOGNITION BASED EMBEDDED CONTROL SYSTEM

PARVEZ[1], PROF. V. K. JOSEPH[2]

[1,2]Electronics and Telecommunication Department, Goa College of Engineering, Farmagudi, Goa, India
E-mail: parvezhssn@gmail.com, vkj@gec.ac.in

Abstract- The purpose of this project is to operate or control Embedded system based on voice recognition, which helps to introduce hearing as well as Natural Language (NL) interface through Speech for the Human-Embedded system interaction. One of the important goals of pursued project is to introduce suitable user interface for novice user and the test plan is to design accordingly.

Index Terms- *LPC, Signal Processing, Hamming Window*

I. INTRODUCTION

The special feature of the application is the ability of the software to train itself for the above voice commands for a particular user. voice controlled Embedded system (VCES) is a Semi Autonomous system whose actions can be controlled by the user by giving specific voice commands. The graphical user interface running along with the software provides a very convenient method for the users to first train the system and then run. The speech signal is captured through microphone and processed by a software running on a PC. Linear Predictive Coding is intended to be used for extracting speech characteristics from sample and implement it through an Embedded system.

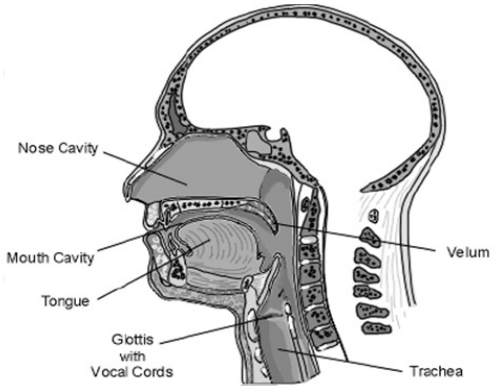
II. HUMAN SPEECH PRODUCTION

Speech is a natural form of communication for human beings. Regardless of the language spoken, all people use relatively the same anatomy to produce sound. The output produced by each humans anatomy is limited by the laws of physics. The process of speech production in humans can be summarized as air being pushed from the lungs, through the vocal tract, and out through the mouth to generate speech. In this type of description the lungs can be thought of as the source of the sound and the vocal tract can be thought of as a filter that produces the various types of sounds that make up speech.

Phonemes are de_ned as a limited set of individual sounds. There are two categories of phonemes, voiced and unvoiced sounds. Voiced sounds are usually vowels and often have high average energy levels and very distinct resonant or formant frequencies. Voiced sounds are generated by air from the lungs being forced over the vocal cords. As a result the vocal cords vibrate in a somewhat periodically pattern that produces a series of air pulses called glottal pulses. The rate at which the vocal cords vibrate is what determines the pitch of the sound produced. Unvoiced sounds are usually consonants and generally have less energy and higher frequencies then voiced sounds. The production of unvoiced sound involves air being forced through the vocal tract in a turbulent flow.

During this process the vocal cords do not vibrate, instead, they stay open until the sound is produced. The amount of air that originates in the lungs also affects the production of sound in humans. The air owing from the lungs can be thought of as the source for the vocal tract which acts as a filter by taking in the source and producing speech. Higher the volume of air, louder the sound is.

Some of the fundamental properties of the speech signal that can be successfully exploited for compression of speech include the quasi-stationary nature of the speech signal. Quasistationary means that speech can be treated as a stationary signal for short intervals of time. This allows us to use techniques which are generally used for stationary signals for processing speech signals. The amplitude of speech signal varies slowly with time, which is another characteristic that is commonly exploited for compression purpose.



III. VOICED AND UNVOICED SPEECH

Voiced sounds, e.g., a, b, are essentially due to vibrations of the vocal cords, and are oscillatory.

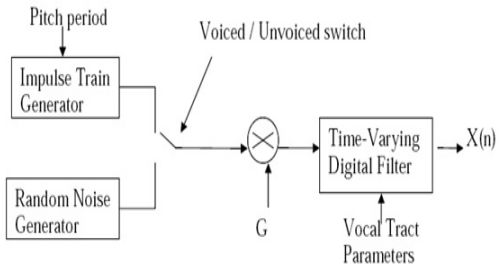
Therefore, over short periods of time, they are well modeled by sums of sinusoids. Unvoiced sounds such as s, sh, are more noise-like.

For many speech applications, it is important to distinguish between voiced and unvoiced speech. There are two simple but effective methods for doing it: Short-time power function: Split the speech signal x(n) into blocks of 10-20 ms, and calculate the power within each block:

$$P_a v = \frac{1}{L} * \sum_{n=1}^{L} x^2(n)$$

$$\text{Typically, } P_a v, voiced > P_a v, unvoiced$$

Zero-crossingrate:the signal x(n) has a zero –crossing at n0 means that
x(n0)x(n0 + 1) < 0
Unvoiced signals oscillate much faster, so they will have a much higher rate of zero-crossings.



Mathematical model of Speech production

IV. SOURCE-FILTER MODEL OF SPEECH PRODUCTION

Sound is variations in air pressure. The creation of sound is the process of setting the air in rapid vibration. Our model of speech production will have two major components:

- Excitation: How air is set in motion. Voiced sounds: Periodic air pulses pass through vibrating vocal chords. Unvoiced sounds: Force air through a constriction in vocal tract, producing turbulence.
- Its upper part (the production of voiced sounds) is very much akin to playing a guitar, you produce a sequence of impulsive excitations by plucking the strings, and then the guitar converts it into music.

The strings are sort of like the vocal cords, and the guitars cavity plays the same role as the cavity of the vocal tract. The period T is called the pitch period, and 1/T is called the pitch frequency.
On average:
male: T 8ms - pitch125Hz:
female : T4ms - pitch250Hz:
Vocal tract: Different voiced sounds are produced by changing the shape of the vocal track, this system is time-varying.

However, it is slowly varying. Changes occur slowly compared to the pitch period. In other words, each sound is approximately periodic, but different sounds are different periodic signals. This implies that we can model the vocal tract as an LTI filter over short time intervals. Moreover, since the vocal tract is a cavity, it resonates. In other words, when a wave propagates in a cavity, there is a set of frequencies which get amplified. They are called natural frequencies of the resonator, and depend on the shape and size of the resonator. Therefore, the magnitude response of the vocal tract for one voiced sound (phoneme) can be modeled, The waveform for this particular phoneme will then be the convolution of the driving periodic pulse train x(t) with the impulse response v(t) and the magnitude of its spectrum |S(f)| will be the product of X(f) and the magnitude response |V (f)|. The maxima of |S(f)| are called the formant frequencies of the phoneme. Typically, one formant per 1 kHz. Locations are dictated by the poles of the transfer function. Roll-off is such that the first 3 4 formants (range: up to 3.5 kHz) are enough for reasonable reconstruction. Thus, sampling at 3.52 kHz = 7 kHz is typically enough. Depending on the application, the sampling rate is usually 7 20 kHz. Suppose we discretised speech, and want to model the vocal tract as a digital filter. The following gives a very rough idea of how to this. If we knew the formant frequencies, we could use what we learned about designing frequency selective filters.
Poles of H(z) near the unit circle correspond to large values of H|ej|. So, we can design an all-pole filter, with poles which are close to the unit circle, corresponding to formant frequencies. The larger the magnitude response at the formant frequency, the closer the corresponding pole(s) to the unit circle.

V. HISTORICAL PERSPECTIVE OF LINEAR PREDICTIVE CODING

The history of audio and music compression begin in the 1930s with research into pulse codemodulation (PCM) and PCM coding. Compression of digital audio was started in the 1960s by telephone companies who were concerned with the cost of transmission and width. Linear Predictive Coding's origins begin in the 1970s with the development of the first LPC algorithms. Adaptive Differential Pulse Code Modulation (ADPCM), another method of speech coding, was also first conceived in the 1970s.
The history of speech coding makes no mention of LPC until the 1970s. However, the history of speech synthesis shows that the beginnings of Linear Predictive Coding occurred 40 years earlier in the late 1930s. The first vocoder was described by Homer Dudley in 1939 at Bell Laboratories. Dudley

developed his vocoder, called the Parallel Band pass Vocoder or channel vocoder, to do speech analysis and re-synthesis.

LPC is a descendent of this channel vocoder. The analysis/synthesis with research into pulse code scheme used by Dudley is the scheme of compression that is used in many types of speech compression such as LPC.

The idea of using LPC for speech compression came up in 1966 when Manfred R. Schroeder and B.S. Atal turned their attention to the following: for television pictures, the encoding of each picture element ("pixel") as if it was completely unpredictable is of course rather wasteful, because adjacent pixels are correlated. Similarly, for voiced speech, each sample is known to be highly correlated with the corresponding sample that occurred one pitch period earlier. In addition, each sample is correlated with the immediately preceding samples because the resonances of the vocal tract. Therefore short durations of speech show an appreciable correlation.

VI. LINEAR PREDICTION

The success with which a signal can be predicted from its past samples depends on the autocorrelation function, or equivalently the bandwidth and the power spectrum, of the signal. As illustrated in figure 2.1, in the time domain, a predictable signal has a smooth and correlated fluctuation, and in the frequency domain, the energy of a predictable signal is concentrated in narrow band/s of frequencies. In contrast, the energy of a non predictable signal, such as white noise, is spread over a wide band of frequencies.

For a signal to have a capacity to convey information it must have a degree of randomness. Most signals, such as speech, music and video signals are, partially predictable and partially random. These signals can be modeled as the output of a filter excited by an uncorrelated input. The random input models the unpredictable part of the signal, whereas the filter models the predictable structure of the signal. The aim of linear prediction is to model the mechanism that introduces the correlation in a signal.

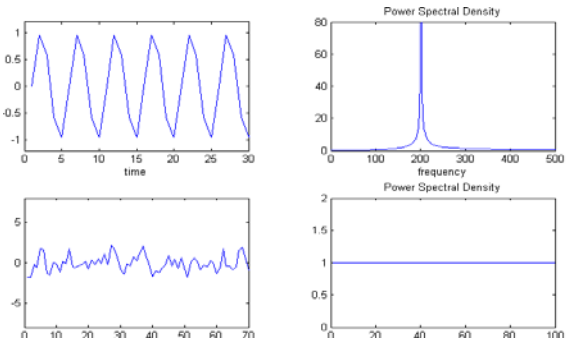


Fig. 1. Time and Power Spectral Density Representation

VII. THEORY OF LINEAR PREDICTION

It is one the most powerful speech analysis technique. This method has become predominant technique for estimating the basis speech parameters, e.g. pitch, formants, spectra, vocal tract area functions, and for representing speech for low bit rate transmission and storage Linear prediction involves predicting the future values of a stationary random process from the observation of past values of the process. By minimizing the sum of squared difference (over a finite interval) between the actual speech samples and the linearly predicted ones, a unique set of predictor coefficient can be determined.

Consider, in particular, a one step forward linear predictor, which forms the prediction of the value x(n) by a weighted linear combination of the past values x(n-1),x(n-2) ... … x(n-p).

Hence linearly predicted value of x(n') is

$$x(n') = -\sum_{k=1}^{k=p} a_p(k) * x(n-k)$$

Where the - (k) represent the weights in the linear combination. These weights are called prediction coefficients of the one step forward linear predictor of order P. The negative sign in the definition of x(n) is for mathematical convenience.

The difference between the value x(n) and the predicted value of x(n)is called the forward prediction error, denoted by

$$fp(n) = x(n) - x(n') = x(n) + \sum_{k=1}^{k=p} a_p(k) * x(n-k)$$

For information bearing signals, the prediction error fp(n) may be regarded as the information, or the innovation, content of the sample. To calculate the optimum prediction coefficients for our prediction filter we minimize the mean square error i.e.

$$\sum (x(n) - x(n'))^2$$

is minimum

Two approaches can obtain the LPC coefficients characterizing an all-pole H(z) model. The least mean square method selects to minimize the mean energy in e(n) over a frame of signal data, while the lattice filter approach permits instantaneous updating of the coefficients.

The first of the two common least squares technique is the autocorrelation method, which multiplies the speech signal by a window w(n) so that x'(n)=w(n)*x(n) has a finite duration.

The first of the two common least squares technique is the autocorrelation method, which multiplies the speech signal by a window w(n) so that x'(n)=w(n)*x(n) has a finite duration.

The autocorrelation sequence describes the redundancy in the signal x(n)

$$r_{ss}[l] = \left(\frac{1}{n}\right)\sum_{n=0}^{N-1}(s[n] - s[n-l])$$

Where x(n), n = {-P, (-P) + 1, . . . ,N - 1} are the known samples and then is a normalizing factor

VIII. COVARIANCE METHOD

An alternative to using a weighting function or window for defining is to fix the interval over which the mean squared error is computed to the range 0 < m < N-1. And use the unweighted speech directly. That is

$$E_n = \sum_0^{N-1} e_n{}^2(m)$$

where $\Phi_n(i,k)$ is defined as

$$\Phi_n(i,k) = \sum_{m=0}^{N-1} x_n(m-1) * x_n(m-k) \begin{cases} 1 \le i \le p \\ 0 \le k \le p \end{cases}$$

or by a change of variable Using the extended speech interval to define the covariance values, (i,k) , the matrix form of the LPC analysis equation becomes,

$$\Phi_n(i,k) = \sum x_n(m) * x_n(m+i-k) \begin{cases} 1 \le i \le p \\ 0 \le k \le p \end{cases}$$

The resulting covariance matrix is symmetric but not Toeplitz, and can be solved efficiently be a set of techniques called the Cholesky decomposition.

A question may arise as to whether to use the autocorrelation method or the covariance method in estimating the predictor parameters. The covariance method is quite general and can be used with no restrictions. The only problem is that of stability of the resulting filter. In the autocorrelation method on the other hand, the filter is guaranteed to be stable, but problems of the parameter accuracy can arise because of the necessity of the windowing (truncating) the time signal. This is usually a problem if the signal is a portion of an impulse response. For example, if the impulse response of an all-pole filter is analyzed by covariance method, the filter parameters can be computed accurately from

$$\begin{bmatrix} \emptyset(1,1) & \cdots & \emptyset(1,p) \\ \vdots & \ddots & \vdots \\ \emptyset(p,1) & \cdots & \emptyset(p,p) \end{bmatrix} \begin{matrix} a_1 \\ \vdots \\ a_p \end{matrix} = \begin{matrix} \emptyset(1,0) \\ \vdots \\ \emptyset(p,0) \end{matrix}$$

only a _nite number of samples of the signal. Using the autocorrelation method, one cannot obtain the exact parameters values unless the whole infinite impulse response is used in the analysis. However, in practice, very good approximations can be obtained by truncating the impulse response at a point where most of the decay of the response has already occurred.

IX. THE AUTOCORRELATION METHOD

In this method, the speech segment is assumed zero outside the interval 0 £ m £ N -1.
Thus the Speech Sample can be expressed as

$$x_n(m) = \begin{cases} x(n+m) * w(m), 0 \le m \le N-1 \\ 0, \quad \text{otherwise} \end{cases}$$

Another least square technique called covariance method windows the error signal, instead of the actual speech signal. In the autocorrelation method the filter is guaranteed to best able, but problems of the parameter accuracy can arise because of the necessity of the windowing (truncating) the rime signal. This is usually a problem if the signal is a portion of an impulse response. For example, if the impulse response of an all-pole filter is analysed by covariance method, the filter parameters can be computed accurately from only a finite number of samples of the signal. Using the autocorrelation method, one cannot obtain the exact parameters values unless the whole infinite impulse response is used in the analysis. However, in practice, very good approximations can be obtained by truncating the impulse response at a point where most of the decay of the response has already occurred.

X. WORD CAPTURING

The signals coming from the microphone is processed only when you speak something. The program waits until the sample value exceeds some threshold value (which can be adjusted buy the user). When the program is triggered by a significant sample, a number of following samples are captured to process. After that to determine the actual boundaries of the word spoken, 'edge detection' is performed. Here the centre of gravity of the energy distribution of the signal iscalculated and then from that point intervals where the amplitude level lies below a threshold level are removed. Finally we can have a set of voice samples corresponding to a particular word free of silent periods.

In this process the noises introduced in the signals because of disturbance in the surrounding also get eliminated and also the software initially measures the noise present in the environment and subtracts this threshold value from the signals to be recorded.

XI. WINDOWING

Windowing of a simple waveform, like cos(ωt) causes its Fourier transform to develop non-zero values (commonly called spectral leakage) at frequencies other than ω. The leakage tends to be

worst (highest) near ω and least at frequencies farthest from ω.

If the signal under analysis is composed of two sinusoids of different frequencies, leakage can interfere with the ability to distinguish them spectrally. If their frequencies are dissimilar and one component is weaker, then leakage from the larger component can obscure the weaker's presence. But if the frequencies are similar, leakage can render them irresolvable even when the sinusoids are of equal strength.

The rectangular window has excellent resolution characteristics for signals of comparable strength, but it is a poor choice for signals of disparate amplitudes. This characteristic is sometimes described as low-dynamic-range.

At the other extreme of dynamic range are the windows with the poorest resolution. These high-dynamic-range low-resolution windows are also poorest in terms of sensitivity; this is, if the input waveform contains random noise close to the signal frequency, the response to noise, compared to the sinusoid, will be higher than with a higher-resolution window. In other words, the ability to find weak sinusoids amidst the noise is diminished by a high-dynamic-range window. High-dynamic-range windows are probably most often justified in wideband applications, where the spectrum being analysed is expected to contain many different signals of various strengths.

In between the extremes are moderate windows, such as Hamming and Hann. They are commonly used in narrowband applications, such as the spectrum of a telephone channel. In summary, spectral analysis involves a trade off between resolving comparable strength signals with similar f.

- HAMMING WINDOW

The "raised cosine" with these particular coefficients was proposed by Richar.W.Hamming. The window is optimized to minimize the maximum (nearest) side lobe, giving it a height of about one-fifth that of the Hann window, a raised cosine with simpler coefficients.

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

$$w_0(n) \stackrel{\text{def}}{=} w(n + \tfrac{N-1}{2})$$

$$= 0.54 + 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$$

where, N represents the width, in samples of a discrete-time window function. Typically it is an integer power-of-2, such as $2^{10} = 1024$.
n is an integer, with values $0 \le n \le N\text{-}1$. So these are the time-shifted forms of the windows: $w(n) = w_0\left(n - \frac{N-1}{2}\right)$,
where $w_0(n)$ is maximum at n=0.
Some of these forms have an overall width of N−1, which makes them zero-valued at n=0 and n=N−1. That sacrifices two data samples for no apparent gain, if the DFT size is N. When that happens, an alternative approach is to replace N−1 with N in the formula.

Each figure label includes the corresponding noise equivalent bandwidth metric (B), in units of DFT bins. As a guideline, windows are divided into two groups on the basis of B. One group comprises $1 \le B \le 1.8$, and the other group comprises $B \ge 1.98$. The Gauss and Kaiser windows are families that span both groups, though only one or two examples of each are shown.

XII. PRE-EMPHASIS

In processing electronic audio signals, pre-emphasis refers to a system process designed to increase (within a frequency band) the magnitude of some (usually higher) frequencies with respect to the magnitude of other (usually lower) frequencies in order to improve the overall signal-to-noise ratio by minimizing the adverse effects of such phenomena as attenuation distortion or saturation of recording media in subsequent parts of the system. That is the mirror of the de-emphasis. The whole system is called emphasis. The frequency curve is decided by special time constants. The cut off frequency can be calculated from that value. Pre-emphasis is commonly used in telecommunications, digital audio recording, record cutting, in FM broadcasting transmissions, and in displaying the spectrograms of speech signals.

In high speed digital transmission, pre-emphasis is used to improve signal quality at the output of a data transmission. In transmitting signals at high data rates, the transmission medium may introduce distortions, so pre-emphasis is used to distort the transmitted signal to correct for this distortion. When done properly this produces a received signal which more closely resembles the original or desired signal, allowing the use of higher frequencies or producing fewer bit errors.

This operation is necessary for removing DC and low frequency components of the incoming speech signal. It also makes the signal spectrum flatter. Pre-

emphasis is done using a first order FIR filter which can be described by the transfer function,

$H(z) = 1-a$

Here we used a = 0.9.

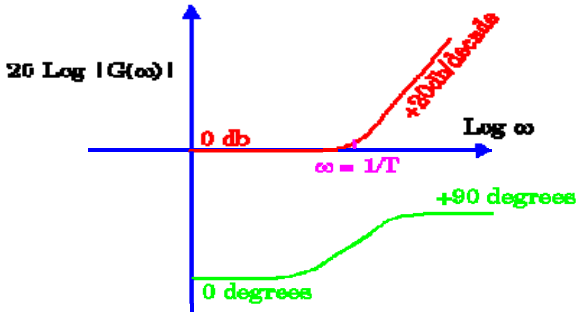
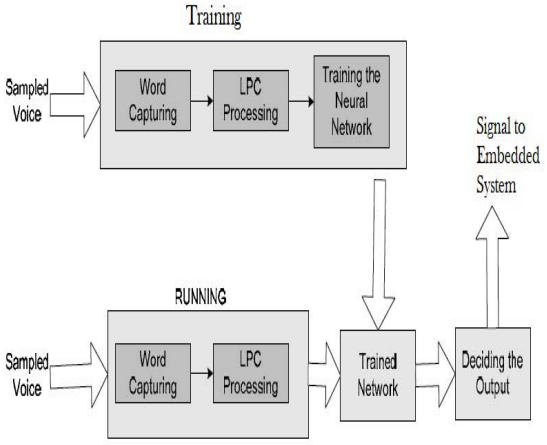


Fig. 2. Pre emphasis Transfer Function

XIII. IMPLEMENTATION

Speech Recognition System - The objective is to develop a software that it will be able to recognize the human voice and then controlled the embedded system based on training mode algorithm and running mode algorithm, i.e The ability of a computer to identify and respond to the sounds produced in human speech. Speech Controlled Embedded system is a system whose motions/functions can be controlled by giving specific voice commands. After processing the speech, the necessary command instructions are sent to the Embedded system via Parallel Port Interface or by using a special function IC embedded in the system The speech recognition software is microphone dependant. The special feature of the application is the ability of the software to train itself for the above voice commands for a particular user. The graphical user interface running along with the software provides a very convenient method for the users to train. It also provides many other facilities in operating the Embedded system.



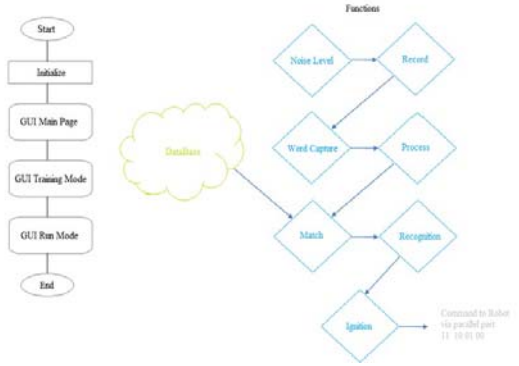
The flowchart of the algorithm implemented is as shown in fig.

The algorithm is as follows:

- During the training phase, the user inputs his sound files.
- This file is then passed through a signal conditioning block where noise cancellation take place, by adjusting the mic noise level.
- After the voice is recorded "word capturing", the process of word filtering take place.
- For better filtering we make use of "Hamming window" as the voice captured is of narrow band frequency.
- On the word captured linear predictive coding(LPC) is performed for getting the cepstral coefficient.
- The comparison is done by comparing the training mode word captured and running mode word captured.
- The interface between embedded system and pc is dove via parallel port connection. and programming the microcontroller of the circuit.

Process

- Return a frequency domain data of the word captured.
- Save waveform.
- Creating a System object to read from a multimedia file.
- Create an FIR digital filter System object used for pre-emphasis.
- Creating a Hamming window object.
- Creating an autocorrelation System object with lags in the range [0:12] scaled by the length of input.
- Create a System object which computes the reflection coefficients.



Setup plots for visualization.

- Read audio input.
- Pre-emphasis..
- Buffer and Window
- Autocorrelation
- Compare the stored data and return the difference value

ACKNOWLEDGMENT

The authors are grateful to Dr. R.B. Lohani, Principal, Goa College of Engineering and Dr. H.G. Virani, Head of Department, Electronics and

Telecommunication Department, Goa College of Engineering for constant support and encouragement.

REFERENCES

[1] Bo Lu, A speech recognition system based on multiple neural networks, Aug 2010.

[2] Aiping Ning, A Speech Recognition System Based on Fuzzy Neural Network Optimized by Time Variant PSO, Sept 2010.

[3] Jing Zhang, A speech recognition method based clustering neural network integration, April2011.

[4] Chenghui Yang, Based on Arti_cial Neural Networks for voice recognition word segment, May 2011.

[5] Morgan, N, Neural networks for statistical recognition of continuous speech, May 1995.

[6] Song Yang,A high performance neural-networks-based speech recognition system, 2001.

◈◈◈