2011 International Conference on Information and Electronics Engineering
IPCSIT vol.6 (2011) © (2011) IACSIT Press, Singapore

Speech Recognition Using Linear Predictive Coding and Artificial Neural Network for Controlling Movement of Mobile Robot

Thiang [1 +] and Suryo Wijoyo [1]

[1] Electrical Engineering Department, Petra Christian University
Jalan Siwalankerto 121-131, Surabaya 60236, Indonesia

Abstract. This paper describes about implementation of speech recognition system on a mobile robot for controlling movement of the robot. The methods used for speech recognition system are Linear Predictive Coding (LPC) and Artificial Neural Network (ANN). LPC method is used for extracting feature of a voice signal and ANN is used as the recognition method. Backpropagation method is used to train the ANN. Voice signals are sampled directly from the microphone and then they are processed using LPC method for extracting the features of voice signal. For each voice signal, LPC method produces 576 data. Then, these data become the input of the ANN. The ANN was trained by using 210 data training. This data training includes the pronunciation of the seven words used as the command, which are created from 30 different people. Experimental results show that the highest recognition rate that can be achieved by this system is 91.4%. This result is obtained by using 25 samples per word, 1 hidden layer, 5 neurons for each hidden layer, and learning rate 0.1.

Keywords: speech recognition, linear predictive coding, artificial neural network, backpropagation.

1. Introduction

Nowadays, speech recognition system is used to replace many kinds of input devices such as keyboard and mouse, therefore the primary objective of the research is to build a speech recognition system which is suitable to be used to control industrial machine processes. The speech recognition system has also been implemented on some particular devices. Some of them are personal computer (PC), digital signal processor, and another kind of single chip integrated circuit.

A framework to address the quantization issues which arise in fixed-point isolated word recognition was introduced in [1]. The system was developed using C++ language which is implemented on a PC. Reference [2] introduced a speech recognition system using SPHINX-II, an off-the-shelf speech recognition package [3]. In reference [4] and [5], speech recognition system has been tried to be implemented on a FPGA and an ASIC. Reference [6] introduced a speech recognition system using fuzzy matching method which was implemented on PC. As the feature extraction method, the system used Fast Fourier Transform (FFT). The best average recognition rate that was achieved by the system was 92%. In reference [7], the speech recognition system was implemented on MCS51 microcontroller. The method used to recognize the word in a speech signal was Linear Predictive Coding (LPC) combined with Euclidean Squared Distance. Next, in reference [8], the speech recognition was implemented on another microcontroller by implementing Linear Predictive Coding (LPC) and Hidden Markov Model (HMM) method.

This paper describes further work about the speech recognition which was implemented on a PC. The methods implemented on this system are LPC and Artificial Neural Network (ANN). As in the previous work [7,8], the method used as feature extraction method in this system is LPC. The difference with previous work is ANN, which is used as the recognition method. The purpose of implementing ANN as recognition method is to improve the recognition rate. There are several words that used as commands for controlling movement of the robot. All the commands that are used to control movement of the robot are Indonesian language, such as “Maju” for forward movement, “Mundur” for backward movement, “Kanan” for turning right, “Kiri” for turning left, “Stop”, etc.

+ Corresponding author. Tel.: + 62-31-2983446; fax: +62-31-8436418.
E-mail address: thiang@petra.ac.id.

2. Speech Recognition System Design

2.1. Feature Extraction Using Linear Predictive Coding

Voice signal sampled directly from microphone, is processed for extracting the features. Method used for feature extraction process is Linear Predictive Coding using LPC Processor.

The basic steps of LPC processor include the following [9, 10]:

1. *Preemphasis*: The digitized speech signal, $s(n)$, is put through a low order digital system, to spectrally flatten the signal and to make it less susceptible to finite precision effects later in the signal processing. The output of the preemphasizer network, , is related to the input to the network, $s(n)$, by difference equation:

$$\tilde{s}(n) = s(n) - \tilde{a}s(n-1) \tag{1}$$

2. *Frame Blocking*: The output of preemphasis step, $\tilde{s}(n)$, is blocked into frames of N samples, with adjacent frames being separated by M samples. If $x_l(n)$ is the l^{th} frame of speech, and there are L frames within entire speech signal, then

$$x_l(n) = \tilde{s}(Ml + n) \tag{2}$$

where $n = 0,1,\ldots,N-1$ and $l = 0,1,\ldots,L-1$

3. *Windowing*: After frame blocking, the next step is to window each individual frame so as to minimize the signal discontinuities at the beginning and end of each frame. If we define the window as $w(n)$, $0 \le n \le N-1$, then the result of windowing is the signal:

$$\tilde{x}_l(n) = x_l(n)w(n) \tag{3}$$

where $0 \le n \le N-1$

Typical window is the Hamming window, which has the form

$$w(n) = 0.54 - 0.46\cos\left[\frac{2\pi n}{N-1}\right] \qquad 0 \le n \le N-1 \tag{4}$$

4. *Autocorrelation Analysis*: The next step is to auto correlate each frame of windowed signal in order to give

$$r_l(m) = \sum_{n=0}^{N-1-m} \tilde{x}_l(n)\tilde{x}_l(n+m) \qquad m = 0,1,\ldots,p \tag{5}$$

where the highest autocorrelation value, p, is the order of the LPC analysis.

5. *LPC Analysis*: The next processing step is the LPC analysis, which converts each frame of $p + 1$ autocorrelations into LPC parameter set by using Durbin's method. This can formally be given as the following algorithm:

$$E^{(0)} = r(0) \tag{6}$$

$$k_i = \frac{r(i) - \sum_{j=1}^{i-1} \alpha_j^{i-1} r(|i-j|)}{E^{i-1}} \qquad 1 \le i \le p \tag{7}$$

$$\alpha_i^{(i)} = k_i \tag{8}$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \qquad 1 \le j \le i\text{-}1 \tag{9}$$

$$E^{(i)} = (1 - k_i^2)E^{i-1} \tag{10}$$

By solving (6) to (10) recursively for $i = 1,2,\ldots,p$, the LPC coefficient, a_m, is given as

$$a_m = \alpha_m^{(p)} \tag{11}$$

6. LPC Parameter Conversion to Cepstral Coefficients: LPC cepstral coefficients, is a very important LPC parameter set, which can be derived directly from the LPC coefficient set. The recursion used is

$$c_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) \cdot c_k \cdot a_{m-k} \qquad 1 \le m \le p \tag{12}$$

$$c_m = \sum_{k=m-p}^{m-1} \left(\frac{k}{m}\right) \cdot c_k \cdot a_{m-k} \qquad m > p \tag{13}$$

The LPC cepstral coefficients are the features that are extracted from voice signal and these coefficients are used as the input data of Artificial Neural Network. In this system, voice signal is sampled using sampling frequency of 8 kHz and the signal is sampled within 0.5 seconds, therefore, the sampling process results 4000 data. Because we choose LPC parameter N = 240, m = 80, and LPC order = 12 then there are 576 data of LPC cepstral coefficients. These 576 data are used as the input of artificial neural network.

2.2. Recognition Using Artificial Neural Network

An Artificial Neural Network is used as recognition method. Architecture of ANN used in this system is a multilayer perceptron neural network. The network has 576 input neurons, which receive input of LPC cepstral coefficient. Number of hidden layer varies from 1 to 4 layers and number of neuron each layer varies from 5 to 30 neurons. The output of the network is code of recognized word. Figure 1 shows architecture of ANN that used in this system. If the artificial neural network has *m* layers and receives input of vector ***p***, then the output of the network can be calculated by using the following equation:

$$\boldsymbol{a}^m = \boldsymbol{f}^m(\boldsymbol{W}^m \boldsymbol{f}^{m-1}(\boldsymbol{W}^{m-1}\boldsymbol{f}^{m-2}(\cdots \boldsymbol{W}^2 \boldsymbol{f}^1(\boldsymbol{W}^1 \boldsymbol{p} + \boldsymbol{b}^1) + \boldsymbol{b}^2) + \boldsymbol{b}^{m-1}) + \boldsymbol{b}^m) \tag{14}$$

Where $\boldsymbol{f}^m$ is log-sigmoid transfer function of the m[th] layer of the network that can be defined as following equation:

$$f(n) = \frac{1}{1+e^{-n}} \tag{15}$$

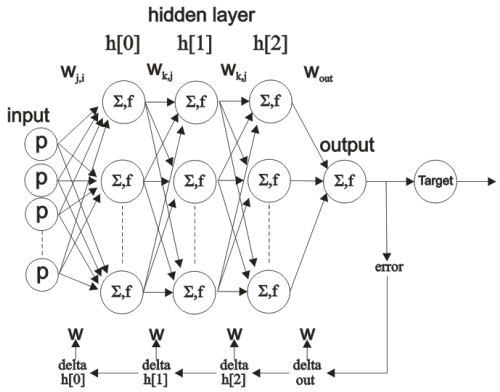

Figure 1. Architecture of Artificial Neural Network Used for Speech Recognition

$\boldsymbol{W}^m$ is weight of the m[th] layer of the network, and $\boldsymbol{b}^m$ is bias of the m[th] layer of the network. Equation (14) is known as the feed forward calculation.

Backpropagation algorithm is used as the training method of the designed artificial neural network. The backpropagation algorithm includes the following steps:

1. Initialize weights and biases to small random numbers.
2. Present a training data to neural network and calculate the output by propagating the input forward through the network using (14).
3. Propagate the sensitivities backward through the network:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t}-\mathbf{a}) \tag{16}$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T \mathbf{s}^{m+1}, \text{ for } m = M-1, \ldots, 2, 1 \tag{17}$$

Where

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \cdots & 0 \\ 0 & \dot{f}^m(n_2^m) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \text{ and } \dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m} \tag{18}$$

4. Update the weights and biases

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) - \alpha \mathbf{s}^m(\mathbf{a}^{m-1})^T \tag{19}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) - \alpha \mathbf{s}^m \tag{20}$$

5. Repeat step 2 – 4 until error is zero or less than a limit value.

3. Experimental Result

Experiments were done in order to test performance of the designed system. Experiments were done in various training parameter value of artificial neural network, i.e., various numbers of hidden layers, various number of neuron per layer, and various value of learning rate. The experiments were also done in various numbers of samples as the training data set. The system was tested using voice which is sampled directly from microphone. Table 1 shows summary of experimental results of the system using 4, 10, 15, 20, and 25 samples training data for each command. Summary of all experimental results are shown at table 2.

Table 1. Recognition Rate of the System Trained Using 4, 10, 15, 20, and 25 Samples per Command

System Trained Using 4 Samples	**Number of Hidden Layer**							
	1		**2**		**3**		**4**	
	Learning Rate		**Learning Rate**		**Learning Rate**		**Learning Rate**	
	0,01	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**
5 Neurons	55.7%	40%	24.3%	70%	14.3%	31.4%	14.3%	14.3%
10 Neurons	57.1%	55.7%	41.4%	57.1%	18.6%	42.9%	14.3%	14.3%
20 Neurons	60%	51.4%	52.9%	62.9%	38.6%	61.4%	20%	37.1%
30 Neurons	58.6%	70%	**72.9%**	42.9%	38.6%	47.1%	28.6%	47.1%
System Trained Using 10 Samples	**Number of Hidden Layer**							
	1		**2**		**3**		**4**	
	Learning Rate		**Learning Rate**		**Learning Rate**		**Learning Rate**	
	0,01	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**
5 Neurons	**82.9%**	51.4%	54.3%	31.4%	14.3%	51.4%	14.3%	14.3%
10 Neurons	64.3%	64.3%	62.9%	58.6%	44.3%	50%	14.3%	47.1%
20 Neurons	64.3%	67.1%	54.3%	62.9%	32.9%	48.6%	41.3%	14.3%
30 Neurons	70%	65.7%	65.7%	50%	50%	71,4%	65.7%	68.6%
System Trained Using 15 Samples	**Number of Hidden Layer**							
	1		**2**		**3**		**4**	
	Learning Rate		**Learning Rate**		**Learning Rate**		**Learning Rate**	
	0,01	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**
5 Neurons	55.7%	52.9%	54.3%	72.9%	38.6%	68.6%	14.3%	14.3%
10 Neurons	75.7%	68.6%	77.1%	72.9%	62.9%	74.3%	14.3%	72.9%
20 Neurons	67.1%	64.3%	68.6%	65.7%	52.9%	58.6%	25.7%	68.6%
30 Neurons	57.1%	51.4%	60%	**78.6%**	70%	71,4%	55.7%	75.7%
System Trained Using 20 Samples	**Number of Hidden Layer**							
	1		**2**		**3**		**4**	
	Learning Rate		**Learning Rate**		**Learning Rate**		**Learning Rate**	
	0,01	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**
5 Neurons	75.7%	71.4%	80%	80%	41.4%	67.1%	14.3%	25.7%
10 Neurons	84.3%	72.9%	**85.7%**	82.9%	54.3%	75.7%	25.7%	47.1%
20 Neurons	78.6%	80%	64.3%	81.4%	72.9%	84.3%	67.1%	82.9%
30 Neurons	70%	81.4%	77.1%	67.1%	60%	82.9%	65.7%	64.3%
System Trained Using 25 Samples	**Number of Hidden Layer**							
	1		**2**		**3**		**4**	
	Learning Rate		**Learning Rate**		**Learning Rate**		**Learning Rate**	
	0,01	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**	**0,01**	**0,1**
5 Neurons	82.9%	**91.4%**	74.3%	67.1%	41.4%	65.7%	14.3%	14.3%
10 Neurons	90%	90%	74.3%	72.9%	64.3%	72.8%	25.7%	27.1%
20 Neurons	82.9%	78.6%	78.6%	74.3%	62.9%	61.4%	47.1%	64.3%
30 Neurons	80%	72.9%	72.9%	77.1%	68.6%	77.1%	78.6%	71.4%

From table 1, we can see that in this application, increasing number of hidden layer of an ANN does not always increase the recognition rate. Moreover, the recognition rate tends to decrease along with increasing number of hidden layer. This also happens to the increasing of learning rate. Increasing number of neuron per layer gives effect of increase of recognition rate. But, after increasing number of neuron per layer

reaches a limit value, the recognition rate does not increase and tends to decrease. From table1, we can also see that increasing number of samples for training ANN results increase of recognition rate. But, if number of samples greater than a limit value, recognition rate does not improve, moreover, the recognition rate tends to decrease. Overall, this system can run well and the best recognition rate that could be achieved is 91.4%. This result is achieved by using ANN with 1 hidden layer, 5 neurons per layer, learning rate of training process of 0.1 and ANN is trained using 25 samples per command. In previous work [7] which used LPC and Euclidean Distance as feature extraction and recognition method, the highest recognition rate that can be achieved is 78.57%. In reference [8], LPC and Hidden Markov Model (HMM) are used as feature extraction and recognition method, the highest recognition rate that can be achieved is 87%. Thus, this research proves that ANN give better recognition rate comparing with Euclidean Distance and HMM.

Table 2. The Best Recognition Rate Based on Number of Sample

Number of Sample	Number of Hidden Layer	Neuron per Hidden Layer	Learning Rate	Recognition Rate (%)
1	1	5	0.01	64.3
4	2	30	0.01	72.9
6	3	5	0.1	77.1
10	1	5	0.01	82.9
15	2	30	0.1	78.6
20	2	10	0.01	85.7
25	1	5	0.1	91.4
30	1	20	0.1	85.7

4. Conclusion and Discussion

From experimental results, it can be concluded that LPC and ANN can recognize the speech signal well. The highest recognition rate that can be achieved is 91.4%. This result is achieved by using LPC and ANN with 1 hidden layer, 5 neurons per layer, learning rate of training process of 0.1 and ANN is trained using 25 samples per command. Compare with previous work, ANN has been proven that it gave better recognition rate. For further work, in order to get better recognition rate, another recognition method such as neuro-fuzzy, fuzzy type-2 method can be applied in this system.

5. References

[1] Y.M. Lam, M.W. Mak, and P.H.W. Leong. Fixed point implementations of Speech Recognition Systems. *Proceedings of the International Signal Processing Conference*. Dallas. 2003

[2] Soshi Iba, Christiaan J. J. Paredis, and Pradeep K. Khosla. Interactive Multimodal Robot Programming. *The International Journal of Robotics Research* (24). 2005, pp 83 – 104.

[3] Huang, X. et al. The SPHINX-II speech recognition system: an overview. *Computer Speech and Language* 7(2). 1994, pp 137–148.

[4] Treeumnuk, Dusadee. Implementation of Speech Recognition on FPGA. *Masters research study, Asian Institute of Technology*, 2001. Bangkok: Asian Institute of Technology.

[5] Sriharuksa, Janwit. An ASIC Design of Real Time Speech Recognition. *Masters research study, Asian Institute of Technology*, 2002. Bangkok: Asian Institute of Technology.

[6] Thiang, Lukman Herlim. Limited Word Recognition Using Fuzzy Matching. *Proceedings of International Conference on Opto-Electronics and Laser Applications*. Jakarta, 2002.

[7] Thiang. Implementation of Speech Recognition on MCS51 Microcontroller for Controlling Wheelchair. *Proceedings of International Conference on Intelligent and Advanced System*. Kuala Lumpur, Malaysia, 2007.

[8] Thiang, Dhanny Wijaya. Limited Speech Recognition for Controlling Movement of Mobile Robot Implemented on ATmega162 Microcontroller. *Proceedings of International Conference on Computer and Automation Engineering*, Bangkok, Thailand, 2009.

[9] Lawrence Rabiner, and Biing Hwang Juang, *Fundamentals of Speech Recognition*. Prentice Hall, New Jersey, 1993.

[10] Ethnicity Group. “Cepstrum Method”. 1998 http://www.owlnet.rice.edu/~elec532/PROJECTS98/speech/cepstrum/cepstrum.html