Jurnal Sains, Nalar, dan Aplikasi Teknologi Informasi Vol. 1 No. 1 (2021)

E-ISSN: 2807-5935 DOI: 10.20885/snati.v1i1.1

SPEECH RECOGNITION APPLICATION AS AN ANIMATED OBJECT MOVEMENT CONTROLLER SYSTEM

Akuwan Saleh
Electrical Engineering Department,
Electronic Engineering Polytechnic
Institute of Surabaya, Surabaya,
Indonesia
Jalan Raya ITS Sukolilo, Surabaya
60111, +62 31 594 7280
akuwan@pens.ac.id

Ach. Aflah Jamazy
Electrical Engineering Department,
Electronic Engineering Polytechnic
Institute of Surabaya, Surabaya,
Indonesia
Jalan Raya ITS Sukolilo, Surabaya
60111, +62 31 594 7280
aflahjamazy13@gmail.com

ABSTRACT

Technological developments in the world have no boundaries. One of them is Speech Recognition. At first, words spoken by humans cannot be recognized by computers. To be recognizable, the word is processed using a specific method. Linear Predictive Coding Method (LPC) is a method used in this research to extract the characteristics of speech. The result of the LPC method is the LPC coefficient which is the number of LPC orders plus 1. The LPC coefficient is processed using Fast Fourier Transform (FFT) 512 to simplify the process of speech recognition. The results are then trained using Backpropagation Neural Network (BPNN) to recognize the spoken word. Speech recognition on the program is implemented as an animated object motion controller on the computer. The end result of this research is animated objects move in accordance with the spoken word. The optimal BPNN structure in this research is to use traingda training function, number of nodes 3, learning rate 0.05, epoch 1000, performance goal 0,00001. This structure can produce the smallest MSE value that is 0,000009957. So, this structure can recognize new words with 100% accuracy for trained data, 80% for the same respondents with trained data and reach 67.5% for new respondents.

Keywords: Speech recognition; LPC method; backpropagation neural network; animation.

1. INTRODUCTION

Technological advances in all fields are very fast. Likewise with speech recognition technology. This technology processes analog data that initially was not recognized by computer devices into data that can be recognized by computer devices. In processing data that is not recognized into data that is recognized by a computer, there are many ways that can be done. One of them is using the Linear Predictive Coding (LPC) method. The LPC method can predict a given speech as a linear combination of the previous human voice signal. By using the LPC method, the characteristics of the sound can be obtained and processed by the next method.

The next method is a method that has the ability like the human brain. This method is the Neural Network method. By using the Neural Network method, computers will be able to have the same capabilities as the human brain.

Voice recognition technology is usually applied to security systems. However, this technology can also be applied to various other things. One of them is as an animation motion control system. In general, animated movements, images, or other objects still use input from the keyboard or mouse.

In this paper, research has been carried out on speech recognition technology which is used to control the movement of animated objects on a computer.

2. RELATED WORK

Research conducted by Wang, Z and Panne, M. Van de [1] on "Walk to here: A Voice Driven Animation System" has produced a system that can recognize human voice input combined with a mouse pointer to produce the desired character animation based on motion capture data. Compares a sound-driven system with an animated, button-driven interface of equivalent capability. The result is that the voice user interface (VUI) is more efficient than using the graphical user interface (GUI) to create the same animation.

Other research conducted by Nancy Ellen Kho [9] on "COMMANIMATION: A Speech-Controlled Animation System" has resulted in a speech-controlled animation system that allows users to quickly and easily create and control new animations. The nature of this command allows speech to be used with this system, it also makes the system feel more natural and can immediately express its purpose in a variety of ways.

The paper "Implementing Speech Recognition in Virtual Reality" written by Denis V.D and Judy M V [10] presents details of implementing speaker-independent, command and control, speech recognition menu systems for virtual reality applications. Generate speech control as a very effective way to control the application. The performance accuracy of the speech interface is quite satisfactory.

3. SYSTEM DESIGN

This section discusses the steps taken, starting from data retrieval, data processing in the learning phase, data processing in the speech recognition phase, to making animations that will be used as application objects. The system design block diagram is shown in Figure 1.

3.1 Data Capture / Voice Recording

Initial data were taken from 10 respondents (6 male and 4 female) aged 21 to 23 years. The data is in the form of words in Indonesian for control commands, namely the word '**atas**', '**bawah**', '**kanan**', and '**kiri**'. The purpose of this data retrieval to extract features using Linear Predictive Coding. The feature extraction is then used as input on neural network learning. Each respondent said each twice, Then the initial data is as in Table 1. The data were taken using Wavesurfer software. An informal user study indicates that for the test scenarios, the voice-user interface (VUI) is faster than an equivalent graphical user interface (GUI) [1].

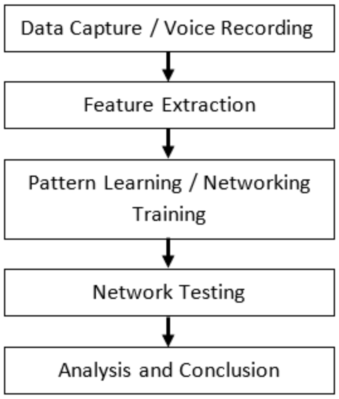

Data Capture / Voice Recording
Feature Extraction
Pattern Learning / Networking Training
Network Testing
Analysis and Conclusion

Figure 1. Block diagram of the system design.

Table 1. Initial data

The spoken word	Number of respondents
Atas	2x6 male respondents 2x4 female respondents
Bawah	2x6 male respondents 2x4 female respondents
Kanan	2x6 male respondents 2x4 female respondents
Kiri	2x6 male respondents 2x4 female respondents
Total number	80 sounds

3.2 Feature Extraction

The feature extraction process is shown in Figure 2. Sampling is a process for taking continuous signal data for any given period. In conducting the data sampling process, the Nyquist rule applies, namely that the sampling frequency (sampling rate) must be at least 2 times higher than the maximum frequency to be sampled. Rules nyquist sampling is used because if the signal is less than 2 times the maximum frequency of the signal to be sampled, then there will be aliasing effects. Aliasing is an effect where the resulting signal has a different frequency from the original signal [2].

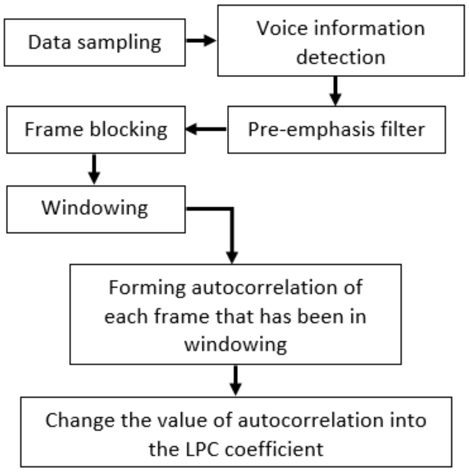

Data sampling
Voice information detection
Frame blocking
Pre-emphasis filter
Windowing
Forming autocorrelation of each frame that has been in windowing
Change the value of autocorrelation into the LPC coefficient

Figure 2. Block diagram of feature extraction.

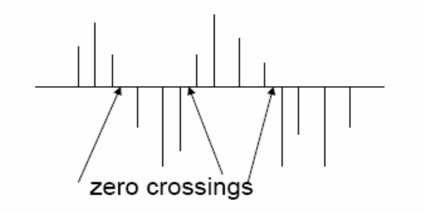


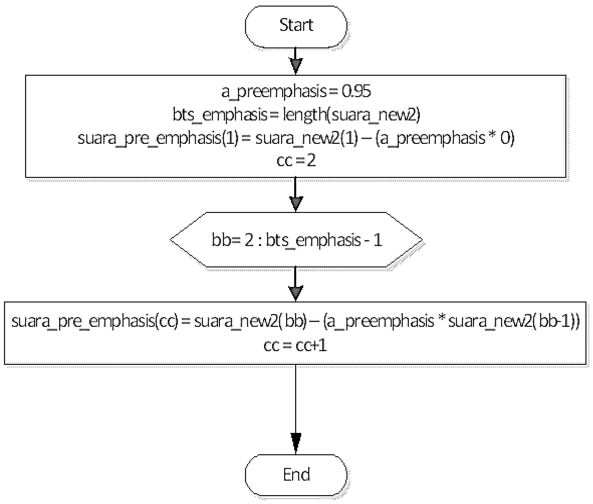
Figure 3. Definition of zero-crossings rate.

After going through the sampling process, information is detected that separates voice, unvoiced, and silence. This detection is carried out by utilizing Zero Crossing Rate (ZCR) and Short-Term Energy (STE). The zero-crossing rate is the rate of sign change along a signal to determine the voiced and unvoiced sounds of an input speech signal. The zero-crossing finds the rate at which the signal changes from positive to negative and vice-versa. This feature of Voice Activity Detection has been used for speech recognition and music information retrieval. ZCR is an important parameter for voiced, unvoiced signal classification. It is often used as a part of the front-end processing in Automatic Speech Recognition system. Speech signals are broadband signals; thus, its results are less precise when interpreted on an average [3]. The zero-crossing rate is an indicator of the frequency at which signal energy is concentrated in the spectrum.

The speaker's voice is generated due to the existence of periodic sounds carried by the air through the sound and usually indicates a low zero crossing rate. Table 2 shows the energy data and ZCR. $\mathbf{M_n}$ is energy and $\mathbf{Z_n}$ is ZCR [8].

Table 2. Energy Data and ZCR

	Voiced	Unvoiced	Silence
$\mathbf{M_n}$	High	Middle	Low
$\mathbf{Z_n}$	Low	High	Middle



```
a_preemphasis = 0.95
bts_emphasis = length(suara_new2)
suara_pre_emphasis(1) = suara_new2(1) – (a_preemphasis * 0)
cc = 2
bb= 2 : bts_emphasis - 1
suara_pre_emphasis(cc) = suara_new2(bb) – (a_preemphasis * suara_new2(bb-1))
cc = cc+1
```

Figure 4. Pre-emphasis filter flow chart.

From table 2 if the zero-crossing rate is high, then the speech signal is unvoiced, and if the zero-crossing rate is low, then the speech signal is voice. For voiced speech, the energy is below 3 KHz, for unvoiced speech, the energy is above 3 KHz. While the average of Zero Crossing Rate for unvoiced speech is 49 per 10 msec interval, the average of Zero Crossing Rate for voiced speech is 10 msec interval. Meanwhile, unvoiced sound is generated due to sufficient narrowing of the sound channel to cause airflow which eventually results in noise and shows a high zero crossing rate [4].

After experiencing the information detection process, the data will be processed with pre-emphasis filter. From Figure 4, it can be explained that the sound data undergoes pre-emphasis filter as a whole. This filter maintains high frequencies in a spectrum, which are generally eliminated during the sound production process. The purpose of this filter is to reduce the noise ratio in the signal, so as to improve signal quality. Equation (1) is a representation of the pre-emphasis filter.

$$y(n) = s(n) - as(n-1) \quad (1)$$

where y (n) is the signal from the pre-emphasis filter, while s (n) = the signal before the pre-emphasis filter. The filtered signal is divided into several frames in the frame blocking process. Frame blocking is done because the voice signal must be processed in short segments (short frame). This process causes the signal to be discontinuous.

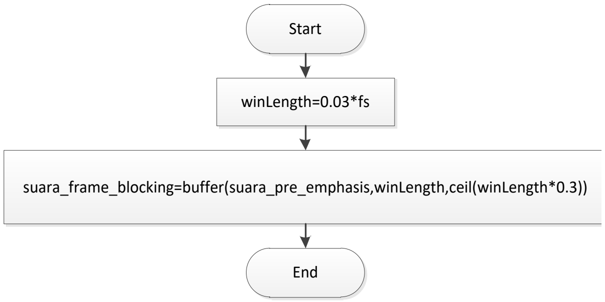

Start
winLength=0.03*fs
suara_frame_blocking=buffer(suara_pre_emphasis,winLength,ceil(winLength*0.3))
End

Figure 5. Frame blocking flowchart.

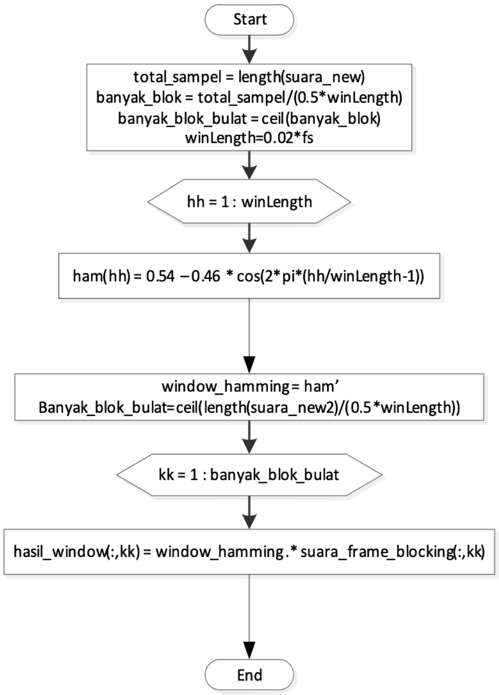

Start
total_sampel = length(suara_new)
banyak_blok = total_sampel/(0.5*winLength)
banyak_blok_bulat = ceil(banyak_blok)
winLength=0.02*fs
hh = 1 : winLength
ham(hh) = 0.54 – 0.46 * cos(2*pi*(hh/winLength-1))
window_hamming= ham'
Banyak_blok_bulat=ceil(length(suara_new2)/(0.5*winLength))
kk = 1 : banyak_blok_bulat
hasil_window(:,kk) = window_hamming.* suara_frame_blocking(:,kk)
End

Figure 6. Window hamming flowchart.

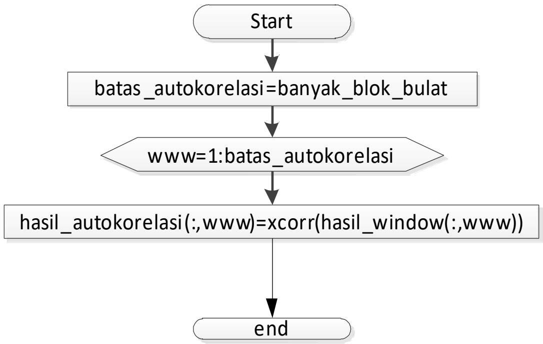

Start
batas_autokorelasi=banyak_blok_bulat
www=1:batas_autokorelasi
hasil_autokorelasi(:,www)=xcorr(hasil_window(:,www))
end

Figure 7. Flowchart of autocorrelation formation.

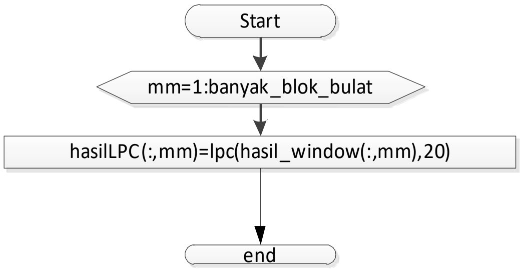


Figure 8. Flowchart of LPC coefficient searching.

To overcome the consequences of the frame blocking process, windowing is carried out, namely window hamming. The windowing process flow chart used is shown in Figure 6.

Equation (2) is a representation of the window function of the sound signal.

$$x(n) = x_t(n).w(n) \quad (2)$$

where x(n) is the sample value of the windowing signal, $x_t(n)$ is the sample value of the nth signal frame, w(n) is the window function, and n is the frame size. Equation (3) shows the Hamming window function.

$$w(n) = 0.54 - 0.4 \cos\left(\frac{2.\pi.n}{M-1}\right) \quad (3)$$

where w(n) is the window function, n is 0.1, ..., M-1, and M is the length of the frame [2].

After windowing, the autocorrelation value of the signal is searched and converted to the LPC parameter. The flow chart for the formation of autocorrelation is shown in Figure 7.

These parameters vary, some are called the LPC coefficient, the reflection coefficient (PARCOR), the cepstral coefficient or other desired transformations. p + 1 autocorrelation result in each frame is converted into the LPC am coefficient for m = 1, 2, ..., p. The flowchart of LPC coefficient search process is shown in Figure 8.

Seeing in Figure 8, the order used is a value of 20 so that the resulting LPC coefficient is 20 + 1 for each frame.

A common method for solving autocorrelation coefficients to LPC coefficients is the Durbin method. The Durbin method algorithm is shown in equations (4), (5), (6), (7), (8), and (9).

$$\mathrm{E}(0)=\mathrm{r}(0) \tag{4}$$

$$\mathrm{k_m}=\frac{\mathrm{r}(\mathrm{m})-\sum_{\mathrm{j}=1}^{\mathrm{m}-1}\mathrm{a}_{\mathrm{j}}^{(\mathrm{m}-1)}.\mathrm{r}(|\mathrm{m}-\mathrm{j}|)}{\mathrm{E}(\mathrm{m}-1)} \tag{5}$$

$$\mathrm{a}_{\mathrm{m}}^{(\mathrm{m})}=\mathrm{k_m} \tag{6}$$

$$\mathrm{a}_{\mathrm{j}}^{(\mathrm{m})}=\mathrm{a}_{\mathrm{j}}^{(\mathrm{m}-1)}-\mathrm{k_m}\mathrm{a}_{\mathrm{m}-\mathrm{j}}^{(\mathrm{m}-1)}\ ,\ 1\leq j\leq m\text{-}1 \tag{7}$$

$$\mathrm{E}(\mathrm{m})=(1-\mathrm{k}_{\mathrm{m}}^{2}).\,\mathrm{E}(\mathrm{m}-1) \tag{8}$$

$$\mathrm{a_j}=\mathrm{a}_{\mathrm{j}}^{(\mathrm{m})}\ ,\ 1\leq \mathrm{j}\leq \mathrm{m} \tag{9}$$

where r(0) is the result of autocorrelation, E(m) is error, $\mathrm{k_m}$ is the coefficient reflection, and is the prediction coefficient for $1 \leq \mathrm{j} \leq \mathrm{m}$ [5].

After obtaining the LPC coefficients, the coefficients are stored in a *.mat file and then processed using fast Fourier transform. The term fast is used because the FFT formulation is much faster than the previous Fourier transform calculation method. Fast Fourier Transform (FFT) is important for a wide variety of applications, from digital signal processing and solving partial differential equations to algorithms for multiplying large numbers of integers [6]. Fast Fourier Transform (FFT) flowchart is shown in Figure 9.

3.3 Pattern Learning/Networking Training

Patterns learning using the neural network backpropagation training method. However, before entering the neural network method, voice data will be extracted first using the linear predictive coding method. The backpropagation network training rule consists of two stages, feedforward and backward propagation. The feedforward process is carried out according to the theoretical basis, starting from the input to the hidden layer to the output layer. After reaching the output layer, error checking is performed. After getting the feedforward results and the errors have been known, backward propagation is carried out to update all the weights and bias values. The first thing to do is get the delta_in value. It then calculates the delta value and updates the weight and bias values accordingly.

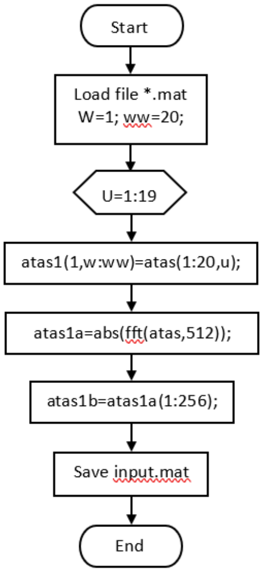


Figure 9. Flowchart of FFT.

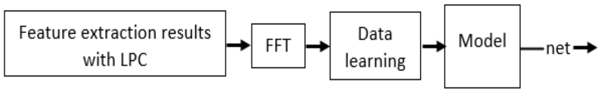


Figure 10. Block diagram of pattern learning.

These steps are carried out until the overall weight and bias have been updated starting from the hidden layer to the input. Because this process moves backward from output to input, it is called backward propagation. If the feedforward and backward propagation processes are completed until the entire training data, it is called one iteration or one epoch if the number of these two processes is carried out continuously until it reaches the maximum iteration specified or has reached the maximum error seen from the MSE value. The MSE value is obtained by calculating the average value of the accumulated error squares.

The network is given a set of training examples called training sets. This training set is described by a feature vector called an input vector which is associated with an output that is the target of the training [7].

The backpropagation network training algorithm consists of 3 stages, namely:

1. The feed forward stage (feedforward).
2. The backpropagation stages.
3. The renewal stage of weights and biases.

The order of the backpropagation algorithm is as follows:

1. Initialize the initial weight consisting of the weight from the input to the hidden layer, the bias weight, the hidden layer weight if any, the weight from the hidden layer to the output node using the random method.
2. Set maximum epoch or iteration, target error, and learning rate.
3. As long as you haven't reached the maximum epoch or MSE is less than the target error, do:

   Learning Stage:

   a. Each input unit is multiplied by its respective weight, which is added to the respective bias weight values. When written in the formula, it will be shown in equation (10).

   $$z\_in_j = b_j + \sum_{i=1}^{n} x_i v_{ij} \tag{10}$$

   b. Use the activation function to get an output signal. The activation function used is shown in equation (11).

   $$z = f(z\_in_j) = \frac{1}{1+e^{-z\_in_j}} \tag{11}$$

   c. Do it to reach the output unit.

   d. In the output unit of each output of the hidden layer multiplied by the weight each is added with the weight of the output unit bias. The equation used is shown in the equation (12).

   $$y\_in_k = b_k + \sum_{i=1}^{p} z_i w_{ij} \tag{12}$$

   e. Perform an activation function to get an output signal. The activation function is shown in the equation (13).

   $$y = f(y\_in_k) = y\_in_k \tag{13}$$

f. Calculate the error value of the output signal. The equation used is equation (14).

$$Error = out\_real - out\_proc \quad (14)$$

g. With out_real is the actual output value, and out_proc is the output value of the process.

h. Add the squared error.

i. Calculate the delta value of the output error using the equation (15).

$$delta = error * f'(y\_in) \quad (15)$$

j. Calculate changes in the value of bias and weights to the output unit. To find out the bias change, the way is by multiplying the value of A with the Delta value. Whereas to determine the change in the weight of the method is by multiplying the value of A with the Delta value and the Z value of the weight, or in other words it can be by multiplying the results of changes in bias with the Z value of the weight.

k. Calculate the Delta value for each hidden layer node. To calculate it, the delta_in value is sought. The trick is the previous layer delta value multiplied by the corresponding weight. After meeting delta_in values, the corresponding delta_in value is multiplied by f '(z_in). The result is a delta value.

l. Calculate changes in the value of bias and weights to the hidden layer.

m. Add changes to each weight and bias with the weight of the weight and the original bias so that the weight and bias value is obtained.

n. Calculate MSE.

4. Do it until all data training is tested.
5. Test all the dataset [6].

At this stage input and target initialization is carried out. Initial initialization in the form of words from respondents, while the target initialization in the form of a boundary value tailored to the function of activation of the bipolar sigmoid. Table 3 shows the input and target initialization data used in the study.

Table 3. Input Initialization and Target

Input	Target
Atas	between -1 to -0.5
Bawah	between -0.5 to 0
Kanan	between 0 to 0.5
Kiri	between 0.5 to 1

Table 3 shows the target values of each voice input. Targets are made within boundaries because the characteristics of each sound are different so that in order to make the network easier in the learning process, the targets are made within boundaries.

3.4 Network Testing

In the voice recognition phase, the program will retrieve new data by recording a new voice and directly extracting the voice. The results of the feature extraction are stored in a .mat file. After the features are obtained, the next step is that these features are processed FFT and entered into the neural network to be identified in the trained network.

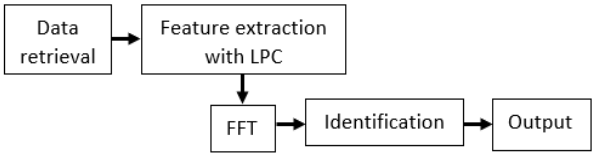


Figure 11. Block diagram network testing.

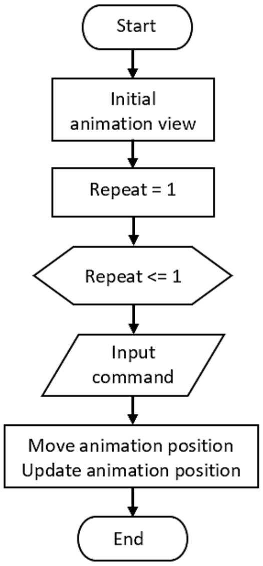


Figure 12. Flow diagram of making animation.

Network testing is done with several experiments, namely:

1. Trained network tested to recognize trained data,
2. Trained networks are tested to recognize data blind type 1 or untrained data but comes from the same respondent as trained data,
3. The trained network is tested to recognize data type 2 blinds or untrained data but comes from different respondents with trained data,
4. The trained network was tested to recognize direct voice input by trained respondents.
5. The trained network is tested to recognize direct sound input by new respondents
6. Trained networks are tested to recognize voice input that is not in accordance with pronunciation procedures.

The result of network identification is the animation that moves according to the word spoken. The animation is a picture displayed in Figure in MATLAB.

In making animation, an image is displayed on a figure. By utilizing the image command provided by the MATLAB software, an image that can be moved and rotated is displayed as desired by the animation maker. And by integrating speech recognition technology, animation will be driven using voice input.

Figure 12 shows a flowchart for the animation. In the flow chart, the animation is displayed in a stationary state in the middle of the field. Then the animation waits for the user to enter the command. When the command has been entered, the animation changes position according to the command and when after the animation moves, the position is updated so that when there is further input, the animation moves from its last position and not from the initial position when the animation was executed.

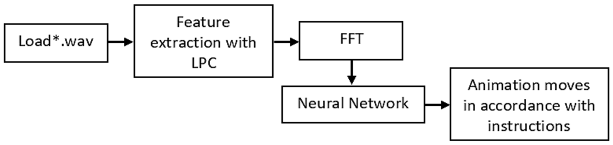

Load*.wav
Feature extraction with LPC
FFT
Neural Network
Animation moves in accordance with instructions

Figure 13. Block diagram offline speech recognition application.

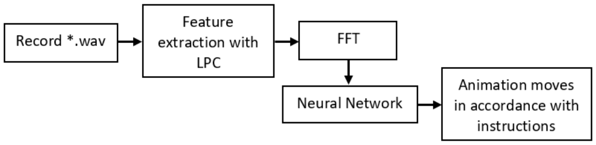

Record *.wav
Feature extraction with LPC
FFT
Neural Network
Animation moves in accordance with instructions

Figure 14. Block diagram online speech recognition. application.

3.5 Integrate Speech Recognition Application with Animation

After creating a program to extract the characteristics of the sound and practice the characteristics of each sound and animation program, an integration between these programs is carried out. The integration of this application can be divided into two, namely offline and online.

3.5.1 Offline Speech Recognition Application

The offline Speech Recognition application is made with the concept of calling the recorded sound database. After the voice is called, the voice undergoes a feature extraction process, FFT and Neural Network. The output is fed into the animation as input.

3.5.2 Online Speech Recognition Application

The online Speech Recognition application is an application made with a slightly different concept from the offline speech recognition application. The difference lies at the beginning of the process, namely recording sound and immediately processing it with feature extraction, FFT, and Neural Network. The output is fed into the animation as input.

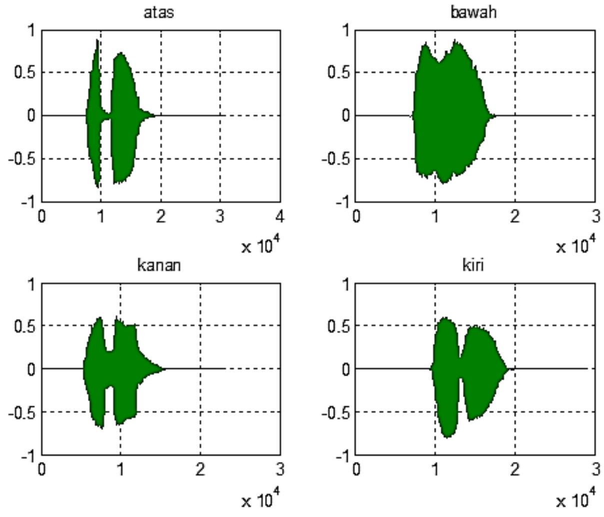


Figure 15. The recording of the word 'atas', 'bawah', 'kanan', and 'kiri'.

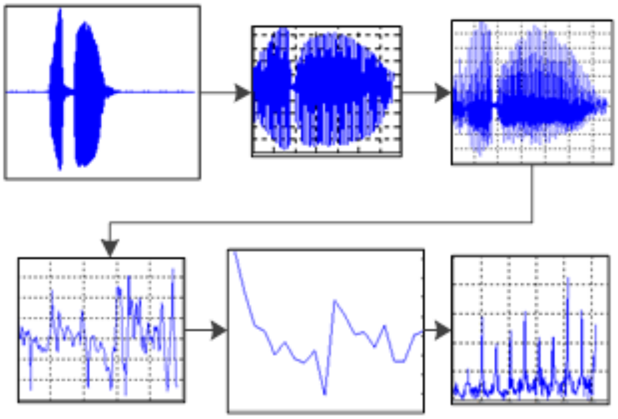


Figure 16. The feature extraction LPC and FFT.

4. RESULTS AND ANALYSIS

Data retrieval is carried out as many as 10 samples for one type of sound. The sampling frequency value used at the time of recording is 22050 Hz. The results obtained are in the form of analog sound signals. Figure 15 is the result of data retrieval one sample for each word. Data that has been taken is stored in the form of a file with the format * .wav.

4.1 Feature Extraction Testing

The results of the feature extraction of a word spoken are shown in Figure 16. Figure 16 explained that the initial sound input is processed through the extraction of LPC and FFT characteristics so that input from the training network is in the form of FFT output.

Before the data is inserted into the network, a network structure formation is carried out which includes the determination of training functions and determination of network parameters. When searching and determining matching competition training and can be used, another parameter initialization is carried out, namely:

- The number of hidden layer = 1,
- Learning Rate = 0.1,
- Performance Goal = 0.00001,
- Number of nodes = 3,
- Max. Epoch = 100000.

Table 4 shows the results of the determination of training functions

Table 4. Determination of training function

Training Function	Performance	Gradient	Epoch	Time
Traingd	0.000009987	0.0200	1408	0:00:08
Traingda	0.0000099974	0.00205	1860	0:00:11
Traingdm	0.4340	$8.22x10^{-6}$	177	0:00:01
Traingdx	0.0979	Infinity	415	0:00:04
Trainlm	0.2941	$1.82x10^{-9}$	7	0:00:00

Training functions in MATLAB are various. However, which is included in the backpropagation neural network includes *traingd*, *traingda*, *traingdm*, *traingdx*, and *trainlm*. The results obtained are Traingd and Traingda training functions produce very small performance values. Judging from the two training terms, a lower gradient decline, namely Traingda so that in this study Training Function Traingda was chosen. After the determined Training Function, namely Traingda, the next step is to determine the suitable and good network parameters to apply. The parameters sought are the number of *nodes*, *learning rates*, *number of epochs*, and *performance goals* so that the MSE value is obtained or the smallest error value with a relatively short time.

Changes in the value of these parameters include:

- Number of nodes = 1, 3, 5, 10
- Learning Rate = 0.01, 0.05, 0.1
- Number of Epoch = 1000, 10000, 100000
- Performance Goal = 0.1, 0.001, 0.00001

Determination of these parameters is shown in Table 5 to 8. From table 5, the node was selected 3 due to the increasing number of nodes used, the more iterations needed. In table 6, when iterates or repetition to 386, learning rate 0.05 is able to achieve the smallest MSE value.

From the data shown in Table 7, it can be explained that the number of EPOCH is the maximum number of network training. Of the three quantities of Epoch, all of them achieved a smaller MSE value of the specified error. However, the best network is the network with iterations / loops at least so that the number of Epoch 1000 is due in addition to the least amount of Epoch, the MSE value is achieved before iteration reaches maximum value.

Table 5. Determination of the number of nodes

Nodes	L. Rate	Epoch	P. Goal	MSE	Gradient	Iteration	Time
1	0.05	1000	0.000010	0.0000 19064	0.00412	1000	00:00:08
3	0.05	1000	0.000010	0.0000 09957	0.00495	386	00:00:02
5	0.05	1000	0.000010	0.0000 09902	0.00457	517	00:00:03
10	0.05	1000	0.000010	0.0000 09837	0.00918	583	00:00:04

Table 6. Determination of learning rate

Nodes	L. Rate	Epoch	P. Goal	MSE	Gradient	Iteration	Time
3	0.01	1000	0.000010	0.0000 09772	0.00351	966	00:00:06
3	0.05	1000	0.000010	0.0000 09957	0.00495	386	00:00:02
3	0.1	1000	0.000010	0.0000 09994	0.00369	866	00:00:05

Table 7. Determination of the number of epochs

Nodes	L. Rate	Epoch	P. Goal	MSE	Gradient	Iteration	Time
3	0.05	1000	0.000010	0.00000 9957	0.00495	386	00:00:02
3	0.05	10000	0.000010	0.00000 9950	0.00248	1604	00:00:10
3	0.05	100000	0.000010	0.00000 9962	0.00558	750	00:00:04

Table 8. Determination of performance goal

Nodes	L. Rate	Epoch	P. Goal	MSE	Gradient	Iteration	Time
3	0.05	1000	0.1	0.0986	0.493	33	00:00:00
3	0.05	1000	0.001	0.00099 2550	0.0312	395	00:00:02
3	0.05	1000	0.00001	0.00000 9957	0.00495	386	00:00:02

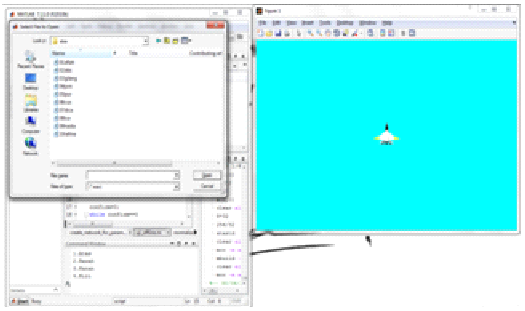
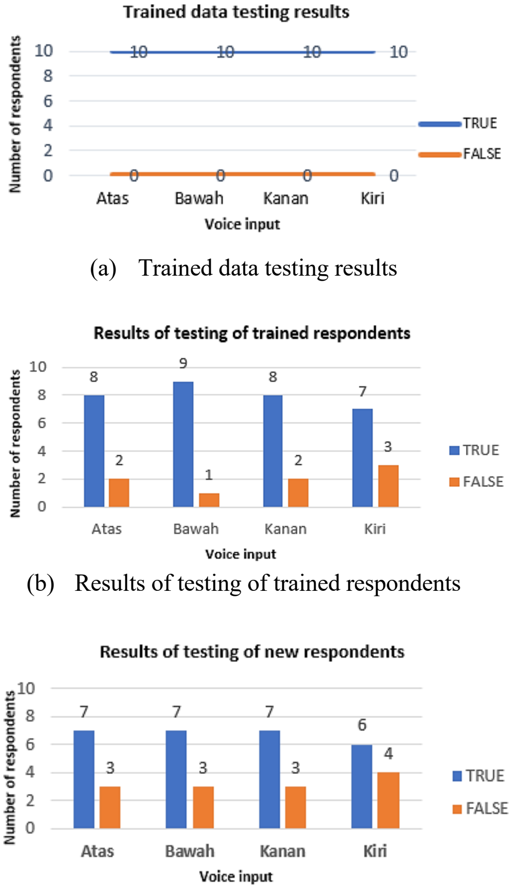


Figure 17. The initial display of animated object control programs.



(a) Trained data testing results

(b) Results of testing of trained respondents

(c) Results of testing of new respondents

Figure 18. Offline testing data charts.

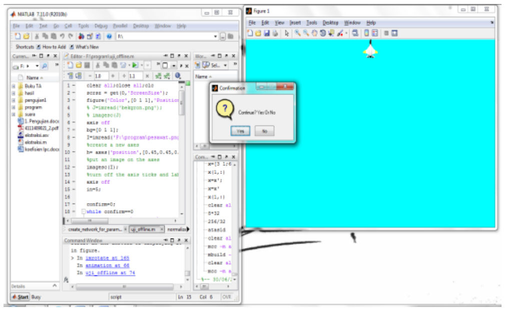
From the data shown in Table 8, it can be explained that the performance of the Goal affects both the poorly trained network. The smaller the value of performance goals, the better the network is. So that the Performance Goal 0.00001 is chosen because it produces the smallest MSE value.

After getting the parameters suitable to apply to the application, Trained network testing is done offline, and online. The response from a trained network is in the form of animations displayed in Figure MATLAB. The animation displayed in the form of a picture of a plane. This animation moves position towards in accordance with the order spoken. Figure 17 shows the initial appearance of the program integrated with animation on MATLAB.

4.2 Testing Offline Speech Recognition Application

Testing with 10 respondents as a sample was tested offline carried out on trained data, new data respondents were trained, and new data on new respondents. According to network accuracy for trained data of 100% or can recognize the entire input data. While the accuracy of the network for new data is trained respondents of 80% and for new data new respondents amounted to 67.5%. As shown in Figure 18.

The Speech Recognition application offline on testing is used to test the sound data database from the same person as trained sound data and voice data from different people with trained sound data. When the program was first run, the appearance appeared as in Figure 17. In the program, users are asked to select the sound database on the computer. The database consists of a command voice '**atas**', '**bawah**', '**kanan**', '**kiri**'. After one of the votes is selected, the response will appear according to the selected sound shown in Figure 19.

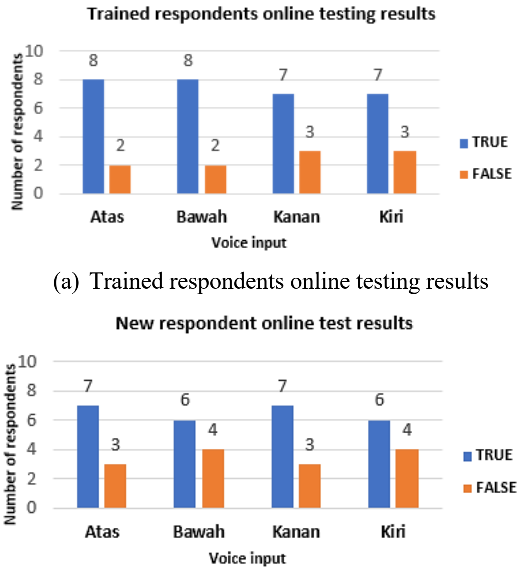


```
clear all;close all;clc
scrsz = get(0,'ScreenSize');
figure('Color',[0 1 1],'Position
% J=imread('bekgron.png');
% imagesc(J)
axis off
bg=[0 1 1];
I=imread('F:\program\pesawat.png
%create a new axes
b= axes('position',[0.45,0.45,0.
%put an image on the axes
imagesc(I);
%turn off the axis ticks and lab
axis off
in=5;

confirm=0;
while confirm==0
```

```
in figure.
> In imrotate at 165
  In animation at 66
  In uji_offline at 74
```

Figure 19. Animated response by input command 'atas'.



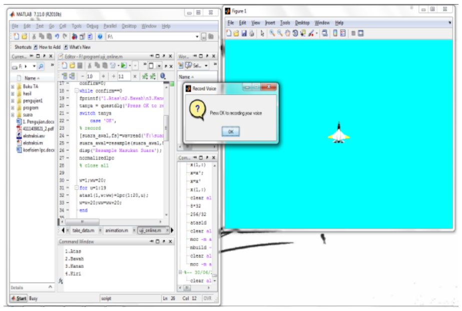
(a) Trained respondents online testing results

(b) New respondent online test results

Figure 20. Online testing data charts.

In Figure 19, the animation moves from the initial position up, after that there is a confirmation popup to continue the program or exit. If the user selects 'Yes' then the program will repeat again from the beginning but the position of the animation is fixed and if the user selects 'No' then the program will come out.

4.3 Testing Online Speech Recognition Application



```
confirm=0;
while confirm==0
fprintf('1.Atas\n2.Bawah\n3.Kana
tanya = questdlg('Press OK to re
switch tanya
    case 'OK',
% record
[suara_awal,fs]=wavread('F:\suar
suara_awal=resample(suara_awal,
disp('Resample Masukan Suara');
normalizedlpc
% close all

w=1;ww=20;
for u=1:19
atas1(1,w:ww)=lpc(1:20,u);
w=w+20;ww=ww+20;
end
```

```
1.Atas
2.Bawah
3.Kanan
4.Kiri
```

MATLAB 7.11.0 (R2010b)
Editor - F:\program\uji_online.m
Figure 1
Record Voice
Press OK to recording your voice
OK

Figure 21. Initial speech recognition online display.

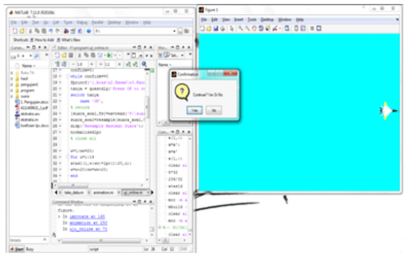


Figure 22. Animated response by input command ‘kanan’.

Online network testing with 10 respondents as a sample was carried out on trained respondents and new respondents. According to the network accuracy for trained respondents was 75%, while the accuracy of the network for new respondents was 65% as shown in graph Figure 20.

When the program is first run, it will be displayed as well as in Figure 21. There is a Popup *'Press OK to Recording Your Voice*'. If the user selects OK then what must be done is to pronounce one of the commands.

The resulting response is the animation moves the position to the direction according to the order spoken. Problemed with the user said the word 'kanan', the response is shown by Figure 22, the animated object moves from the initial position to the right position after that there is a confirmation popup to continue the program or exit.

5. CONCLUSION

From testing data and analysis, it can be concluded that Speech recognition in the form of commands from spoken Indonesian words, namely ‘atas’, ‘bawah’, ‘kanan’ and ‘kiri, has been identified with a good level of accuracy. This shows that the pattern recognition approach which consists of two steps, namely learning speech patterns and speech recognition through comparison of patterns is running well. The optimal training function for use in this study is the traingda training function. The accuracy of the network in recognizing trained data is 100%, the accuracy of the network in recognizing new sound data from trained respondents is 80%, the accuracy of the network in recognizing new sound data from new respondents reached 67.5%. The accuracy of the network in recognizing new sound data spoken by respondents is trained online, reaching 75%, the accuracy of the network in recognizing new sound data of new respondents online is 65%. The animated object in the form of a plane was successfully controlled according to 'atas','bawah', 'kanan' and 'kiri' voice command. The success of this study will be used as the basis for the next research on the mobile robot movement control system using sound signals or voice commands.

6. REFERENCE

[1] Wang, Z., and Panne, M.V. D. 2006. "Walk to here": A Voice Driven Animation System, Eurographics. *ACM SIGGRAPH Symposium on Computer Animation*, 243-250, doi: http://dx.doi.org/10.2312/SCA/SCA06/243-250.

[2] Aditya, R. 2012. *Prototipe Pengenalan Suara sebagai Penggerak Dinamo Starter pada Mobil*. Universitas Gunadarma, Depok. 5-8.

[3] Sharma, B., Talukar P.H. 2016, Zero Crossing Rate of The Voice and Unvoiced Speech Signal of Assamese Words. *International Journal of Scientific & Engineering Research*. 7, 12, 402-405, ISSN 2229-5518.

[4] Bachru R.G., Kopparthi S., Adapa B., and Barkana B.D. *Separation of Voiced and Unvoiced using Zero Crossing Rate and Energy of the Speech Signal*. Electrical Engineering Department School of Engineering, University of Bridgeport. pp. 2-4. [online] https://moam.info/separation-of-voiced-and-unvoiced-using-zero-crossing-rate-and-_59b58c381723ddd8c6ad46d6.html (accessed May 8, 2019).

[5] Rohman, N.S. and Hidayatno, A. 2012. Aplikasi Pencirian Dengan Linear Predictive Coding Untuk Pembelajaran Pengucapan Nama Hewan Dalam Bahasa Inggris Menggunakan Jaringan Saraf Tiruan Propagasi Balik, *Jurnal Ilmiah Teknik Elektro (TRANSMISI)*. Universitas Diponegoro, Semarang. 14, 4. 150-158, doi: https://doi.org/10.12777/transmisi.14.4.150-158.

[6] Nugraha, D.A. 2008. *Software Pembelajaran Neural Network Dengan Algoritma Backpropagation*. Politeknik Elektronika Negeri Surabaya, Surabaya, 23-26.

[7] Agustin, M. 2012. P*enggunaan Jaringan Syaraf Tiruan Backpropagation untuk Seleksi Penerimaan Mahasiswa Baru pada Jurusan Teknik Komputer di Politeknik Negeri Sriwijaya*. Universitas Diponegoro, Semarang, 29-31.

[8] Wuhan University. *Speech Signal Processing* (Chapter 3 Speech Analysis). School of Electronic Information. [Online] https://s3-ap-southeast-1.amazonaws.com/tv-prod/documents%2F4414-SPEECH+ANALYSIS.pdf (accessed April 16, 2020).

[9] Nancy E. K. 2004. *COMMANIMATION: A Speech-Controlled Animation System*. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology.

[10] Denis V.D and Judy M. V. 2002. Implementing Speech Recognition in Virtual Reality. *Proceedings of DETC' 02 ASME 2002 Design Engineering Technical Conferences and Computer and Information in Engineering Conference*. Montreal, Canada. 1-5, DETC2002/CIE-34390.