SIMULATION VOICE RECOGNITION SYSTEM FOR CONTROLING ROBOTIC APPLICATIONS

[1]WAHYU KUSUMA R., [2]PRINCE BRAVE GUHYAPATI V

[1]Computer Laboratory Staff., Department of Information Systems, Gunadarma University, Indonesia

[2]Postgraduate Student., Department of Electrical Engineering, Gunadarma University, Indonesia

E-mail: [1]wahyukr@staff.gunadarma.ac.id, [2]prince8888@pasca.gunadarma.ac.id

ABSTRACT

Voice recognition is a system to convert spoken words in well-known languages into written languages or translated as commands for machines, depending on the purpose. The input for that system is "voice", where the system identifies spoken word(s) and the result of the process is written text on the screen or a movement from machine's mechanical parts.

This research focused on analysis of matching process to give a command for multipurpose machine such as a robot with Linear Predictive Coding (LPC) and Hidden Markov Model (HMM), where LPC is a method to analyze voice signals by giving characteristics into LPC coefficients. In the other hand, HMM is a form of signal modeling where voice signals are analyzed to find maximum probability and recognize words given by a new input based from the defined codebook.

This process could recognize five basic movement of a robot: "forward", "reverse", "left", "right" and "stop" in the desired language. The analysis will be done by designing the recognition system based from LPC extraction, codebook model and HMM training process. The aim of the system is to find accuracy value of the recognition system built to recognize commands even the speaker voice isn't currently stored in the database.

Keywords: *Voice Recognition, Robot, LPC, HMM*

1. INTRODUCTION

Biometric systems commonly used for identify and verify an individual being to acquire the identity of the authorized individuals by comparing and checking the submitted data with the database that contains the record of authorized individuals. The process followed by verification, where the system made decision for the submitted data after being compared with the stored data.

Biometric recognition applied as identification method for humans based from specific biological characteristics they have. The use for biometric recognition has many ways and forms, and so on, which implemented on many ways and form, one of them is voice recognition. Voice recognition is the method to recognize voice spoken by a person, which divided by two classifications: voice recognition and speaker recognition [3][8]. The main difference from those methods is the purpose of the system, where voice recognition system identifies the keyword said by a speaker regardless of the speaker's identity, and speaker recognition identifies the speaker based on the elements of sound. The aim for this research is to define the methods behind the voice recognition systems and set up for an implementation for defined system.

2. VOICE RECOGNITION SCHEME

The basic principle of voices is that the voice made by friction between two or more objects which produces vibration on the air and then received by human ear. That vibration can be produced by human itself with vocal instruments.

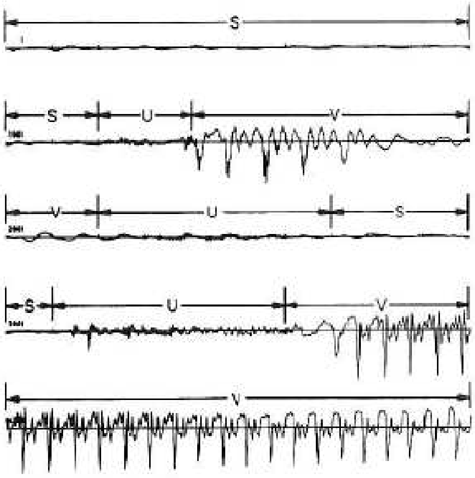
Voice signals divided based from the excitation methods:

a. Voiced excitation;

b. Voiceless excitation;

c. Transient excitation [10]

Voice signals are shown as in Figure 1.



[Figure 1: Voice Signal Samples][8]

The schemes for voice recognition systems are (a) input stage by retrieving voice samples; (b) extraction stage by building a template database based from sampled voice signals; (c) matching stage, to match any submitted data with given template and (d) identity validation, to find out the appropriate keyword then sending the command to another defined system.

There are two classification for those systems: (a) dependent voice recognition, where it requires special training from users by using sound profiles and easier to build because the voice samples are already saved on a database with vocabulary list; and (b) independent voice recognition where the system recognizes a word or sentence regardless of who spoke the word/sentence [8]. This model examines each voice input with recognized words/sentences and choose one which have the best probability value of all.

3. LINEAR PREDICTIVE CODING

Linear predictive coding or abbreviated LPC is a stronger method to analyze the coded voice files with better quality on low bit rate samples. The reasons why LPC commonly used are: (a) LPC proves better approximation coefficient spectrum; (b) LPC give shorter and efficient calculation time for signal parameters and (c) LPC has been able to get important characteristics of the input signals.

LPC process block diagram given in Figure 2, which contains six steps:

Step 1: Pre-emphasis

Step 2: Frame selection

Step 3: Window process

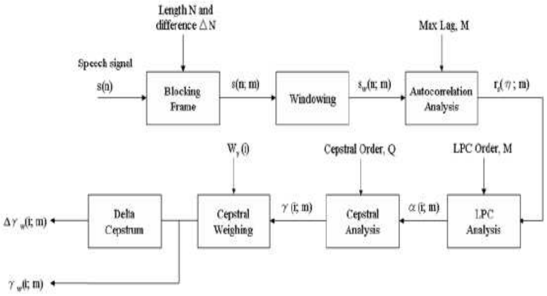
Step 4: Autocorrelation analysis

Step 5: LPC analysis

Step 6: Cepstral coefficients conversion

Step 7: Cepstral weight

Step 8: Delta cepstrum definition [8]

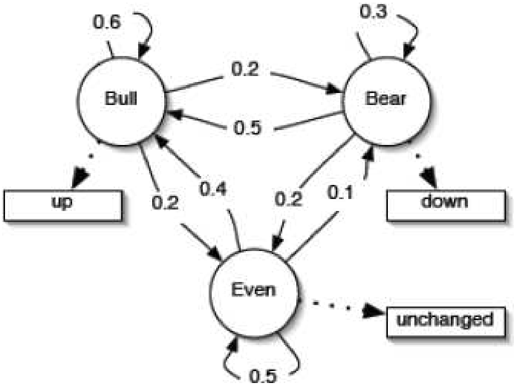


[Figure 2: LPC Process Diagram][8]

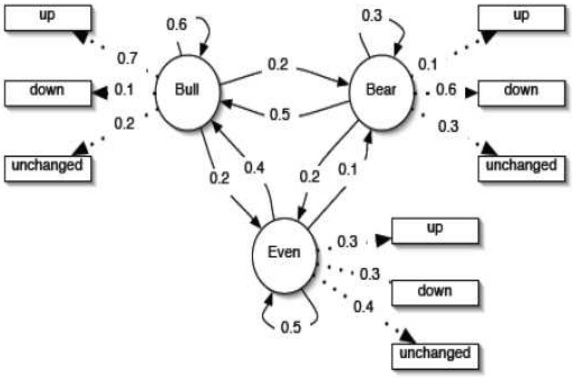
4. HIDDEN MARKOV MODEL

Hidden Markov Model or HMM is a statistical model from a system that assumed as Markov processes with unknown parameters. The aim for HMM is to find hidden parameters inside recognized signal patterns. HMM states are observed by identifiable variables which influenced by those hidden states [1][6].

The simplest way to understand how HMM works is represented in Figure 3 and 4 below:



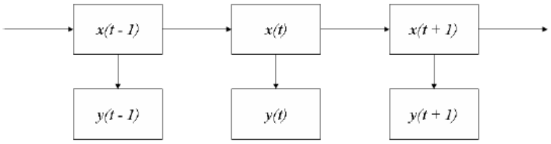
[Figure 3: Markov Process Diagram][3]



[Figure 4: HMM Process Diagram][3]

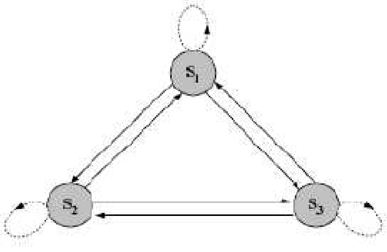
There are random variables used by HMM process: (a) x(t) that contains x(t) that contains values of a hidden variable on t time session, (b) y(t) that contains values of a known variable on t time session. Value of y(t) is depend on the value state of x(t), and the value of x(t) depend on its previous state x(t-1).

Figure 5 shows how HMM process defined as dependent states which every state x(t) depend on values from the previous state and also influence values of the next state.

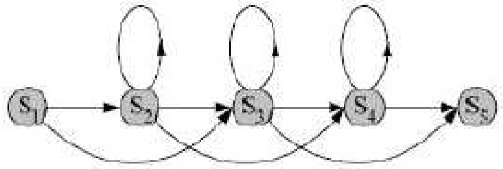


[Figure 5: HMM Architecture Diagram][2]

Also, there are two HMM types for describe HMM: (a) ergodic model where the change of one state to another is all possible or reversible on a loop or known as *state cycle*, and (b) left-to-right model where the state changes in order from the leftmost to rightmost state with irreversible process. Those models shown in Figure 6(a) for ergodic model and Figure 6(b) for left-to-right model.



[Figure 5(a): Ergodic Model][2]



[Figure 5(b): Left-to-Right Model][2]

A. *Elements*

There are some HMM elements that needed to deal with as follows:

a. N, indicates total states given due to the model implementation.

b. M, indicates total of unique observation symbols in every states. The observation symbol could be character sets or numbers.

c. Transition state distribution, stated by the formula below:

$$a_{ij} = P\left[q_{t+1} = j | q_t = j\right], \quad 1 \leq i, j \leq N$$

d. Observation symbol probability distribution, given by this equation:

$$b_j(k) = P\left(o_t = v_k | q_t = j\right), \ 1 \leq j \leq N, \ 1 \leq k \leq M$$

e. Initialization state distribution model, defined as this:

$$\pi_i = P\left[q_t = i\right], \ 1 \leq i \leq N$$

[1][9]

There are three algorithms to solve each cases stated on the following:

a. Forward algorithm that solves given model parameters which have output probability as a certain series of number.

b. Viterbi algorithm that solves given model parameters which have hidden state series with maximum probability to give output as a given certain series of number.

c. Baum-Welch algorithm that solves state transition with given output series or dataset to found the best probability of state transition groups together with the output probability [6].

B. Quantization Vectors

Vector quantization or known as VQ is a clustering technique for process time series signals to several clusters. Each cluster represents data that have little difference on spectral characteristics and owned by a specific population. The gravitation center of each cluster assigned for specific index and assumed as representative of cluster population on signal process [7].

By assuming VQ as redudance shifting that minimize required bits to identify windows structure inside of the signal, VQ could be use for generating a codebook that defined as a voice database by quantizing weight cepstral coefficient vectors from all references. The main benefits from VQ are (a) reduce the amount of spectral analysis information; (b) reduce the calculations for define the similarity of spectral vector analysis and (c) discrete spoken voice representation make recognition process more efficient.

Two types of recognition algorithms below were commonly used on VQ:

a. K-Means

b. Binary split

K-Means have easier and simpler method to be implemented on a HMM model, so it become common when used for some applications because of using a set of learning vector as codebook vector.

Hence there are some steps on K-Means algorithm:

Step 1: Initialization

The algorithm starts by choosing M vector as the codeword initial set on the codebook.

Step 2: Nearest neighbor

For every learning vectors L, define the nearest codeword on the corresponding codebook and assign the vector to the proper cell.

Step 3: Centroid update

The system updates any codewords in each cell using centroid method from learning vector assigned to that cell.Step 4: LPC analysis

Step 4: Iteration process

Repeat two steps above (nearest neighbor and centroid) until the mean distance value has below the preset threshold value [2].

C. Forward-Backward Algorithm

This algorithm based from the dynamic programming model that make calculations for small samples and save its results, then can be re-used when those results become important. The method is more efficient than repeating all steps given from the beginning.

The algorithm itself divided on two process: forward algorithm and backward algorithm. Hence it is the forward algorithm order shown below:

a. Initialization

$$\alpha_1(j) = \pi_i e_j(o_1) \quad j = \overline{1, N}$$

b. Recursion

$$\alpha_{k+1}(j) = \left[\sum_{i=1}^{N} \alpha_k(i) a_{ij}\right] \cdot e_j(o_{k+1}) \quad j = \overline{1, N};\ k = \overline{1, L-1}$$

c. Termination

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_L(i)$$

The same order also applied for backward algorithm as given:

a. Initialization

$$\beta_1(i) = 1 \quad i = \overline{1, N}$$

b. Recursion

$$\beta_k(i) = \left[\sum_{j=1}^{N} a_{ij} e_j\right] \cdot e_j(o_{k+1}) \beta_{k+1}(j) \quad i = \overline{1, N};\ k = \overline{L-1, 1}$$

c. Termination

$$P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i e_i(o_1)\, \beta_1(i)$$

[2]

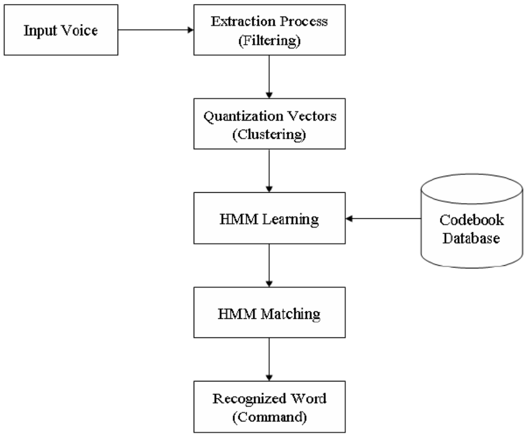
5. PROPERTY SETUP

The system made for this research is a software-based system that have aim for (a) model the parent system with a voice control; (b) model possible voice types to be recognized by the system and (c) generate plots that will help determine whether the voice recognition model satisfy given requirements.

These processes included inside the system for apply the signal processing technique: (a) extraction process, (b) VQ or vector quantization and (c) HMM learning with recognition algorithm.

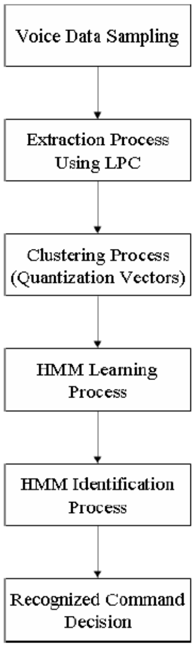
A. HMM Process

Figure 6 represents the flow diagram for set up designed system in proper parameters as shown below:


Input Voice
Extraction Process (Filtering)
Quantization Vectors (Clustering)
HMM Learning
Codebook Database
HMM Matching
Recognized Word (Command)

[Figure 6: HMM Recognition Scheme][10]

Also, there are operation steps or operational procedures that occur inside the system, given by Figure 7.


Voice Data Sampling
Extraction Process Using LPC
Clustering Process (Quantization Vectors)
HMM Learning Process
HMM Identification Process
Recognized Command Decision

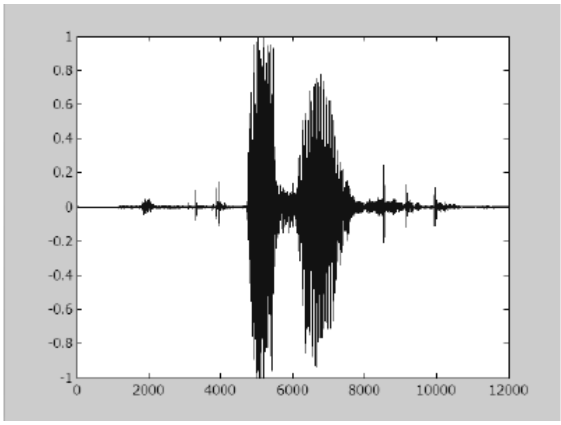
[Figure 7: System Operational Procedures]

B. Voice Sampling Method

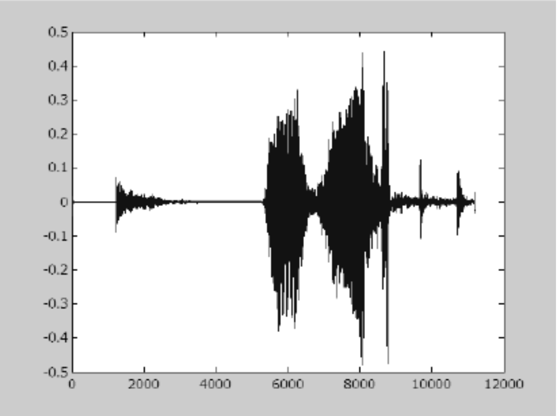
The first and the most important stage to perform a voice recognition system scheme is voice sampling method where the sample voices recorded through voice-sensitive recording device to generate digitized waveform of the sampled voice signal.

The system is set to be able to recognize five types of voice with wave sound and special audio files which samples separated based on the speaker: male and female speakers.

For example, Figure 8 shows one of the voice given by male speakers and Figure 9 shows one of the voice given by female speakers.



[Figure 8: Waveform from the male speaker]



[Figure 9: Waveform from the female speaker]

The system initialized by program the codebook with five sample voice signatures that used as basis for recognition process.

After all voice samples generated, the next step is to extract the characteristics from the provided signal where the signal usually already filtered to reduce noise level and decrease the error ratio for recognizing noise-interfered signal from the environment.

Hence there are the known process order:

a. Signals are grouped on frames with N sample size with estimated ms sampling time on given sampling frequency.

b. Each frame windowed with Hamming window method to minimize signal discontinuities on the start and end part of the frame, then autocorrelated with order value M.

c. LPC analysis, where autocorrelation value on each frame converted to LPC coefficients and calculated with Levinson-Durbin recursive process, then converted to cepstral coefficients with Q cepstral coefficients.

d. Cepstral weight used to minimize sensitivity against the noise, and the last delta cepstrum done for represents cepstral from voice spectrum.

The execution of those process inside the programming environment formed a function named *hmmfeatures*, which uses calculation for the length of signal, then determine how many frames built using the command below it. Afterwards, the framing, windowing, autocorrelation, LPC analysis, cepstral coefficient calculation, cepstrum weighing and cepstrum difference is executed in order to extract voice parameters inside the signal.

C. Quantization Vectors

The whole process above is how to find the observation vector that needed to build required quantization vector. Key point of the system is the clustering process that using K-Means algorithm.

K-Means algorithm based from two steps: (a) observation vector distribution and (b) clustering process on the highest distribution area. Quantization vector processing written in a function named *kmeans* where the dimension vector given by two dimensional array, then the extracted parameters used for randomly initializing centroid and to create centroid. Afterwards, there is a loop to done clustering process so that the system generates quantized vectors that could be processed as a codebook by using *hmmcodebook* function.

By defining the data length and load the voice data, the system generates a codebook given by two dimensional array contains voice data and the results stored by K-Means algorithm. Also, there is a distance variable where used for count distance of errors made when the codebook is being generated.

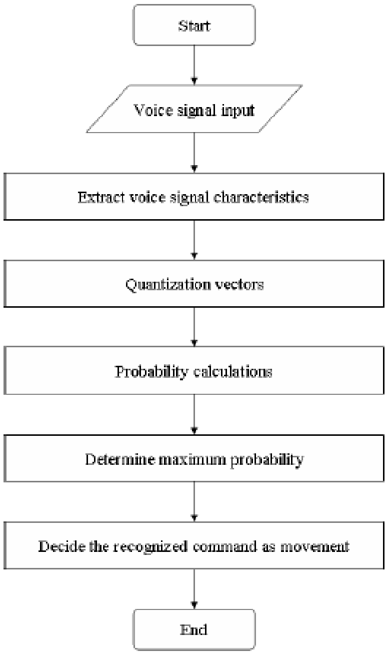
D. HMM Learning and Recognition

The forward-backward algorithm on HMM section used to obtain log-likelihood values by building *hmmrecog* and *hmmlogp* function, respectively. HMM recognition process by the *hmmrecog* function has codebook parameter and delta cepstrum pattern as input, where the function itself produces the log parameter for signal detection probability and also generate one of five values given to decided the voice type based from the codebook database.

E. Programming Interface

To develop the recognition system on the proper enviroment, the MATLAB programming with graphical interface is chosen since it is easier to write HMM related codes and functions [5].

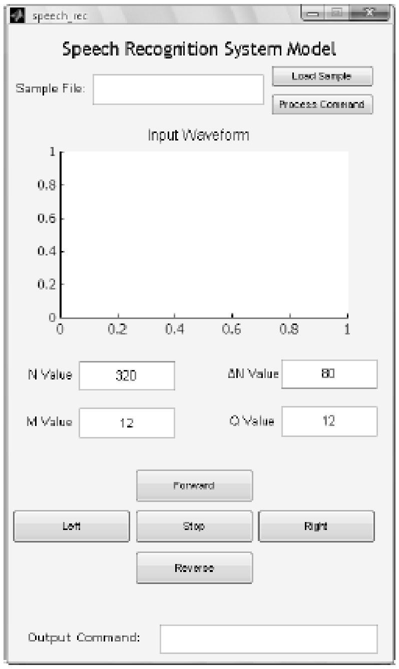
Figure 10 shows that how a voice had been recognized by the system as a command for another mechanical systems where the voice data processed with codebook database acts as basis for recognition scheme.


Start
Voice signal input
Extract voice signal characteristics
Quantization vectors
Probability calculations
Determine maximum probability
Decide the recognized command as movement
End

[Figure 10: Flowchart of the recognition process][8]

F. Program Properties

Under the MATLAB graphical interface environment, the system for voice recognition scheme has been set up as shown in Figure 11. The implemented extension system is a mechanical system that can made five basic elements of movement: forward, reverse, turn left, turn right and stop.



Sample File:

N Value 320

ΔN Value 80

M Value 12

Q Value 12

Output Command:

[Figure 11: System initialization][2][4]

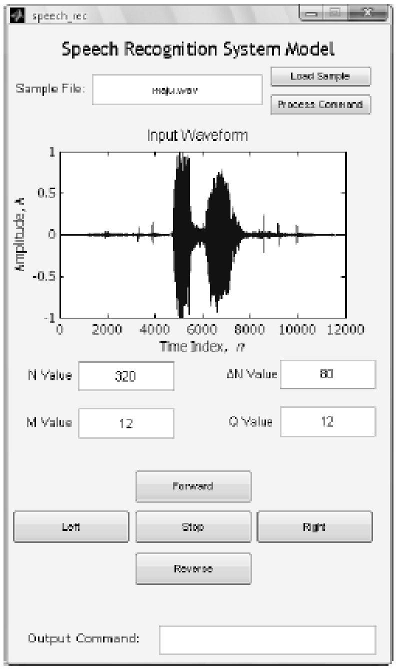
The codebook used for the system is based from TI46 codebook model with some changes applied to fit on five samples given as recognition base. Voice data that had been set stored in an audio file with ".mat" extension as an array, with ".label" and ".case" parameters to perform matching process [5].

6. RESULTS AND ANALYSIS

This result analysis is done by an Intel Core 2 Duo T5850, 2.16 GHz processor, 2 GB of RAM and 250 GB hard disk with Windows XP Service Pack 3 and MATLAB 7.4 installed.

A. System Test

By executing "Load Sample" button object and fill the sample voice data path, the system displays waveform of the sample signal as seen in Figure 12.



Sample File: maju.wav

N Value 320

ΔN Value 80

M Value 12

Q Value 12

Output Command:

[Figure 12: Voice sample input]

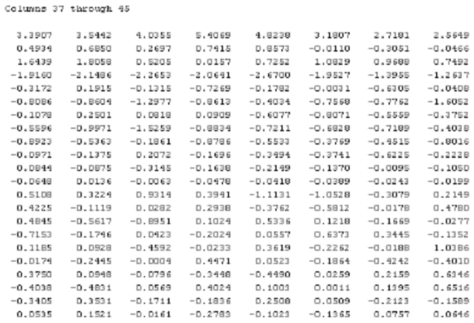
In the same time, the system also generated about 3,288 LPC coefficients as identification values from the sample given. Some values given inside Table 1 for example.

Table 1: LPC Coefficients from a Sample File

1.715	20.958	0.9371	0.7024	13.574	24.917
0.6489	21.805	20.237	22.006	14.051	0.5542
0.4999	21.634	20.355	23.006	34.481	28.116
-0.7421	22.044	18.191	11.283	14.129	3.098
20.921	13.567	18.154	24.516	26.434	24.072
10.098	19.926	30.763	29.008	13.614	13.933
-0.0832	13.224	18.105	1.77	18.754	17.394
12.548	12.809	13.874	13.178	13.023	1.216
0.2929	0.9075	0.795	0.8071	12.543	0.9186
0.4945	0.7778	0.965	0.9075	0.7501	0.7455
0.1057	0.2515	-0.0052	0.1445	0.3654	0.1404
0.0679	0.1055	0.1856	0.1579	0.1129	0.1033
0.3808	-0.389	-0.6967	0.2102	0.8947	0.7266
15.316	0.6874	0.01	-0.3093	-0.8232	0.025
16.634	0.7678	0.0686	0.7063	0.2555	-0.6188
29.465	12.806	-0.538	-0.2031	0.9848	0.7597
-0.7354	-0.1384	0.5474	0.414	-0.0222	-0.3966
0.9827	10.332	0.4541	-0.8575	-0.7538	-0.2411
14.056	0.9468	0.2238	0.0325	-0.0153	-0.0886
0.0261	0.0663	0.0185	-0.0426	-0.0509	-0.0468
0.6146	0.251	-0.0502	0.2297	0.0557	-0.1054
0.2832	0.2352	0.0649	-0.1075	-0.081	0.0407
0.1457	-0.0555	-0.0535	0.1853	-0.002	-0.0759
0.0376	0.0589	0.0262	-0.0364	-0.0273	-0.0094

There are another values from LPC process besides LPC coefficients, called observation vectors where used for clustering process.

Values of the observation vectors given by Figure 13.

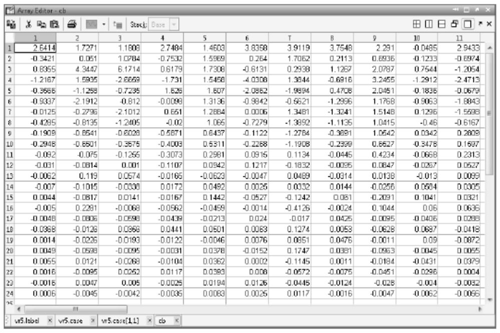


```
  Columns 37 through 45

    3.3907    3.5442    4.0355    5.4069    4.8238    3.1807    2.7181    2.5649
    0.4934    0.6850    0.2697    0.7415    0.8573   -0.0110   -0.3051   -0.0466
    1.6439    1.8058    0.5205    0.0157    0.7252    1.0829    0.9688    0.7492
   -1.9160   -2.1486   -2.2653   -2.0641   -2.6700   -1.9527   -1.3955   -1.2637
   -0.3172    0.1915   -0.1315   -0.7269   -0.1782   -0.0031   -0.6305   -0.0408
   -0.8086   -0.8604   -1.2977   -0.8613   -0.4034   -0.7568   -0.7762   -1.6052
   -0.1078    0.2501    0.0818    0.0909   -0.6077   -0.8071   -0.5559   -0.3752
   -0.5596   -0.9971   -1.5259   -0.8834   -0.7211   -0.6828   -0.7189   -0.4038
   -0.8923   -0.5363   -0.1861   -0.8786   -0.5533   -0.3769   -0.4515   -0.8016
   -0.0971   -0.1375    0.2072   -0.1696   -0.3494   -0.3741   -0.6225   -0.2228
    0.0844   -0.0875   -0.3145   -0.1638   -0.2149   -0.1370   -0.0095   -0.1050
   -0.0648    0.0136   -0.0063   -0.0478   -0.0418   -0.0389   -0.0243   -0.0199
    0.5108    0.3224    0.9314    0.3941   -1.1131   -1.0528   -0.3079    0.2149
    0.4225   -0.1119    0.0282    0.2938   -0.3762   -0.5812   -0.0178    0.4780
    0.4845   -0.5617   -0.8951    0.1024    0.5336    0.1218   -0.1669   -0.0277
   -0.7153   -0.1746    0.0423   -0.2024    0.0557    0.6373    0.3445   -0.1352
    0.1185    0.0928   -0.4592   -0.0233    0.3619   -0.2262   -0.0188    1.0386
   -0.0174   -0.2445   -0.0004    0.4471    0.0523   -0.1864   -0.4242   -0.4010
    0.3750    0.0948   -0.0796   -0.3448   -0.4490    0.0259    0.2159    0.6346
   -0.4038   -0.4831    0.0569    0.4024    0.1003    0.0011    0.1395    0.6516
   -0.3405    0.3531   -0.1711   -0.1836    0.2508    0.0509   -0.2123   -0.1589
    0.0535    0.1521   -0.0161   -0.2783   -0.1023   -0.1365    0.0757    0.0646
```

[Figure 13: Observation vectors from a sample file]

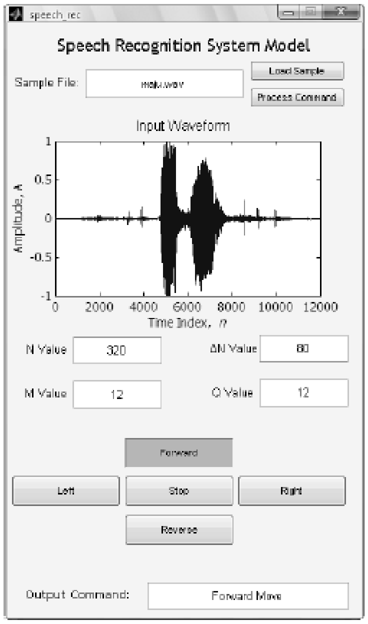
The process continues with HMM training stage, where forward-backward algorithm used to get log-likelihood values using five hidden state models. The result from HMM training process given in Figure 14.



	1	2	3	4	5	6	7	8	9	10	11
1	2.6414	1.7271	1.1808	2.7484	1.4603	3.8358	3.9119	3.7548	2.291	-0.0485	2.9433
2	-0.3421	0.051	1.0784	-0.7532	1.5969	0.264	1.7062	0.2113	0.6936	-0.1233	-0.6974
3	0.8355	4.3447	6.1714	0.6179	1.7308	-0.6131	0.2938	1.1267	2.0787	0.7544	-1.2054
4	-1.2167	1.5935	-2.6669	-1.731	1.5458	-4.0308	1.3844	-0.6916	3.2455	-1.2912	-2.4713
5	-0.3666	-1.1258	-0.7235	1.626	1.607	-2.0862	-1.9894	0.4708	2.0451	-0.1836	-0.0679
6	-0.9337	-2.1912	-0.812	-0.0098	1.3136	-0.9842	-0.5621	-1.2996	1.1768	-0.9063	-1.8843
7	-0.0125	-0.2796	-2.1012	0.651	1.2884	0.0006	1.3481	-1.3241	1.5148	0.1296	-1.5698
8	-0.4285	-0.8135	-1.2405	-0.02	1.065	-0.7279	-1.3892	-1.1135	1.0415	-0.46	-0.6167
9	-0.1909	-0.8541	-0.6028	-0.5871	0.6437	-0.1122	-1.2784	-0.3891	1.0542	0.0342	0.2809
10	-0.2948	-0.6601	-0.3675	-0.4003	0.5311	-0.2268	-1.1908	-0.2399	0.6627	-0.3478	0.1697
11	-0.092	-0.075	-0.1255	-0.3073	0.2981	0.0915	0.1134	-0.0445	0.4234	-0.0668	0.2313
12	-0.031	-0.0814	0.001	-0.1107	0.0942	0.1217	-0.1832	-0.0095	0.0847	-0.0267	0.0527
13	-0.0062	0.119	0.0674	-0.0165	-0.0623	-0.0047	0.0489	-0.0314	0.0138	-0.013	0.0099
14	-0.007	-0.1015	-0.0336	0.0172	0.0492	0.0025	0.0332	0.0144	-0.0256	0.0684	0.0305
15	0.0044	-0.0817	0.0141	-0.0167	0.1442	-0.0527	-0.1242	0.061	-0.2091	0.1041	0.0321
16	-0.005	0.2281	-0.0068	-0.0562	-0.0459	-0.0014	-0.4126	-0.0024	0.1044	0.06	0.0636
17	-0.0048	-0.0806	-0.0698	-0.0439	-0.0213	0.024	-0.017	0.0425	-0.0095	-0.0406	0.0288
18	-0.0368	-0.0126	0.0966	0.0441	0.0501	0.0063	0.1274	0.0053	-0.0628	0.0687	-0.0418
19	0.0014	-0.0226	-0.0193	-0.0122	-0.0046	0.0076	0.0851	0.0476	-0.0011	0.09	-0.0872
20	0.0049	-0.0598	-0.0095	-0.0031	0.0378	-0.0152	0.1747	0.0381	-0.0963	-0.0045	0.0055
21	0.0055	0.0121	-0.0268	-0.0104	0.0362	0.0002	-0.1145	0.0011	-0.0184	-0.0431	0.0379
22	0.0016	-0.0095	0.0252	0.0117	0.0383	0.008	-0.0572	-0.0075	-0.0451	-0.0296	0.0004
23	-0.0016	0.0047	0.005	-0.0025	0.0194	0.0126	-0.0445	-0.0124	-0.028	-0.004	-0.0032
24	0.0006	-0.0045	-0.0042	-0.0035	0.0083	0.0025	0.0117	-0.0016	-0.0047	-0.0062	-0.0056

[Figure 14: HMM codebook contents]

The process is completed by determining the result of the recognition process and give command to the mechanical system, as shown in Figure 15.



Sample File: maju.wav

N Value 320

ΔN Value 80

M Value 12

Q Value 12

Output Command: Forward Move

[Figure 15: Voice recognition result]

B. *Performance Test*

This test performed for characteristics analysis for 5 male and 5 female samples as input with the same keyword given. The test procedure split by two procedures: (a) LPC coefficient test and (b) codebook database test.

The performance test involving 5 samples with 25 voice data files given 25 files correctly recognized or resulting the 100\% accuracy, where the same test involving 5 samples with 25 voice data files given 17 files correctly recognized or resulting the 68\% accuracy.

7. CONCLUSION

This research has been outlined the work on analysis of voice recognition in voice-controlled robot devices. As a review of thesis objectives, the concept of in the area of voice recognition system especially for voice recognizing method are studied. Also, the model for the recognition system is designed in order to analyze hidden states and HMM effectiveness by using computer simulation.

In this research, the recognition system are modeled in MATLAB graphical user interface (GUI) with the representation of input signals and given commands (forward, reverse, turn left, turn right, and stop).

From the simulation result there are several conclusions as follows :

a. In the voice recognition proses, the following steps used such as voice input, extraction using LPC, clustering, HMM training and HMM recognition.
b. The accuracy result of 25 voice sample data that have voice database gives 100% accuracy, where other 25 voice sample data that did not have voice database gives 68% accuracy.
c. Based from accuracy test performed, voice database is significantly affected the recognition accuracy where larger probability of recognition given by larger voice sample data stored in the database.
d. The accuracy test also shown the system had recognized the command "turn left" and "turn right" more accurate with all input samples contained "turn left" and "turn right" commands given correct results, which indicates the system would better to recognize left and right turn command rather than another commands.

8. FUTURE WORK

In the future, there is still needs some improvement of the simulation model in order to provided a more resemble compared to the real world.

Several suggested future works that can be done are listed below:

a. In this thesis, the implementation of the robot controlling system still on software simulation. The future work can be done on building the hardware system for the simulation and investigate more hidden states inside the HMM model with further iterations. Also, the developed system in future may be a real-time processing with input directly given from the microphone.
b. The future work can be done by using another algorithm model, such as Viterbi or Baum-Welch algorithm to analyze the effect of every hidden states and characteristics of the voice signals.
c. In fact, LPC and HMM voice signal processing algorithms given in this thesis use similar methods of Texas Instruments' TI46 voice recognition model based on English numbers, another codebook model and expansion of control commands will apply.

Therefore, in future work needs some improvement on the related algorithms and other possible moves that can be done by a voice-controlled robot.

REFERENCES:

[1] W. H. Adbulla and N. K. Kasabov, *The Concept of Hidden Markov Model in Speech Recognition*, 1999.

[2] A. Hidayatno and Sumardi, "Pengenalan ucapan kata terisolasi dengan metode hidden markov model melalui ekstraksi ciri linear predictive coding," *Penelitian Hibah Bersaing DIKTI Depdiknas*, vol. 2, 2006.

[3] X. Huang, A. Acero, and H. W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.* Prentice Hall PTR, 2001.

[4] P. Marchand and O. T. Holland, *Graphics and GUIs with MATLAB.* CRC Press, 2003.

[5] I. T. U. of Copenhagen, "Speech coding and recognition course," November 2005, *http://www.itu.dk/courses/TKG/E2005/exercises.html.*

[6] V. A. Petrushin, *Hidden Markov Models : Fundamentals and Applications*, 2000.

[7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*. Prentice Hall, 2007.

[8] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[9] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[10] S. Saito and K. Nakata, *Fundamentals of Speech Signal Processing*. Academic Press, 1985.