VOICE CONTROLLED ROBOT

Supervised By:

Dr. Rohan Munasinghe

Group Members:

Vithana P.V.K.C 020402 (ENTC)
Hettiarachchi I.M. 020144 (ENTC)
Prasanga M.W.B. 020304 (EE)
Piyasinghe L.P. 020298 (EE)
**Introduction**

Voice Controlled Robot (VCR) is a mobile robot whose motions can be controlled by the user by giving specific voice commands. The speech recognition software running on a PC is capable of identifying the 5 voice commands ‘Run’, ‘Stop’, ‘Left’, ‘Right’ and ‘Back’ issued by a particular user. After processing the speech, the necessary motion instructions are given to the mobile platform via a RF link. Following is the system overview.

![Speech Recognition Diagram](image)

The speech recognition software is speaker 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 robot.

**Speech Recognition**

In this unit we capture the speech signals coming from the microphone attached to the PC. The software running on PC processes the signals to recognize the voice commands ‘Run’, ‘Stop’, ‘Left’, 'Right' and ‘Back’. The software also provides a facility to train itself for the above commands. For training we use Artificial Neural Networks. The software is written using MATLAB.

To identify words, we use LPC (Linear Predictive Coding) which is a popular method of extracting speech characteristics from sample values. First, a spoken word is recognized and then a set of parameters called ‘Cepstral Co-efficients’ was calculated from the voice data samples belonging to that word based on LPC. Then those co-efficients are processed by the trained neural network to decide whether that word is any of the above specific commands. If the word is identified as one of those commands, then a relevant signal is sent to the mobile platform via RS232. The overall operation is illustrated in the diagram below:
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 by 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 center of gravity of the energy distribution of the signal is calculated 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.

LPC Processing

The steps we followed for the extraction of speech characteristics from captured samples using LPC is described in the following block diagram.
a) **Pre-emphasis:**
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,

\[ H(z) = 1 - az^{-1} \]

Here we used \( a = 0.9 \). FIR filtering was applied to the signal in the time domain using the MATLAB function ‘filter’.

b) **Frame Blocking:**
Each signal is now converted into a set of fixed length frames, with some number of samples in each frame overlapping. If the frame length is \( L \) and each frame is shifted by \( M \) samples away from the adjacent frame, then \( n^{th} \) frame can be denoted by,

\[ x_n[i] = s[(n-1)M + i] \]

where \( n = 1,2,...,N \) and \( i = 1,2,...,L \).

c) **Windowing:**
Each individual frame is windowed to minimize the signal discontinuities at the borders of each frame. We used the ‘Hamming Window’ for this purpose. The set of samples for each frame is multiplied by the time domain version of the Hamming window with size equal to the frame length.

d) **LPC Calculation:**
First step of calculating LPC parameters is to get the autocorrelation vector for each frame. If the order of the autocorrelation is \( P \), then the autocorrelation vector, ‘\( r \)’ can be given by,

\[ r(m) = \sum_{n=0}^{N-m-1} x(n) x(n + m) \]

where \( m = 0,1,2,...,P \) and \( x(i) \) s (\( i = 1,2,...,L \)) are sample values in the windowed frame. Then Hermitian Toeplitz matrix of \( r \) is computed as shown below:

\[
R = \begin{bmatrix}
   r[0] & r[1] & ... & r[P-1] \\
   r[1] & r[0] & ... & r[P-2] \\
   ... & ... & ... & ... \\
   r[P-1] & r[P-2] & ... & r[0]
\end{bmatrix}
\]

Finally the LPC parameter matrix, ‘\( a \)’ is calculated by matrix multiplication of inverse of \( R \) and \( r \).

\[ a = R^{-1} * r \]

In our software we used order(\( P \)) as 10.

e) **Cepstral Co-efficient Calculation:**
Cepstral co-efficients are Fourier transform representation of the log magnitude spectrum. Use of Cepstral co-efficients makes the application more robust and reliable.
As low order LPC parameters are too sensitive to the spectral slope and low order parameters are sensitive to noise, Cepstral co-efficients are weighted by a tapered window. So weighted set of Cepstral co-efficients, \( C_m \) are found by,
\[
C(m) = \left\{ \begin{array}{ll}
\sum_{k=1}^{m-1} kC(k)a(m-k)/m, & m \in [1, P] \\
\sum_{k=1}^{m-1} kC(k)a(m-k)/m, & m > P
\end{array} \right.
\]

For each frame belong to the captured word, this sequence of operations are performed and finally we end up with a number of sets of Cepstral co-efficients. Next values for all the frames are averaged to get a single set of Cepstral co-efficients for that spoken word.

**Artificial Neural Network:**

To recognize the speech mobile robot should have intelligence. Hence artificial neural network is used to make intelligence through the learning process. In here we used Multi Layer Feedforward Network and error back propagation learning algorithm to train the network. Figure 4 is shown the network model.

This network consists of two Hidden layers with the Input & Output layers. Liner transfer function used to the input layer and sigmoid transfer function applied to hidden layer as well as output layer. Following section describe the mathematical background of this network and learning algorithm. Figure 5 shows the block diagram of the backpropagation training algorithm.
Input Layer transfer function
\[ y = mx \]

Hidden Layers & Output Layers transfer function
\[ y = \frac{1}{1 + \exp(-\lambda x)} \]

---

This neural network builds on MATLAB computing environment and it process data in real time and the program give the outputs according to recognized voice command. First of all calculated set of Cepstral co-efficients for that spoken word is fed to input layer of ANN and calculate the output. During the training process we input the set of voice command samples with the relevant output network to be produced. For each iteration network calculate the actual network output and compare with the specified output and error is backpropagate to minimized the error in next iteration.

---

**Tuning Process**

The above Neural network we used is capable of identifying two voice commands. In order to identify more commands we used three separate networks. When using this kind of process we need to set output threshold levels according to the way that it had trained. To do this we give known voice patterns to the already obtained neural network and set the threshold levels accordingly.
Graphical User Interface

The main objective of the GUI is to provide a user friendly Interface which integrates all the required functionalities. That is it should be able to facilitate all phases of voice recognition algorithm including acquiring voice samples, then train the neural network and finally running voice recognition process in real time.

There are several important areas within the GUI. They are

- Mode Selection Panel (Used to select required Mode that user want to run)
  - Acquisition Mode – Obtain voice samples separately for each command and then calculate Cepstral co-efficients of each voice sample and store them
  - Training Mode – Use cepstral co-efficients to build the Neural Network. So Neural Network coefficients are obtained.
  - Tuning Mode – Use known voice samples to calculate threshold levels needed.
  - Running Mode - Each voice input is sampled real time and use the neural network to obtain the output and then use threshold levels obtained in the Tuning mode to recognize the voice input
Options Panels (Acquisition, Tuning and Running)
  - Used to set options needed at each Mode

Recognition Indications
  - This is enabled at Running Mode and used to indicate the status of the latest voice input

Graph Window
  - Used to display acquired sound pattern

Emergency Stop
  - Used to issue a STOP command if needed

Hardware Construction

The mobile robot platform consists of three wheels and forward two wheels are coupled to the two geared DC motors. The rest wheel can free rotate with its own axis. The robots capable of forward, backward, turn left and turn right motions and onboard motor driver circuit with L298 IC is used to make the those motion control with the aid of PWM (Pulse Width Modulation) switching signals.

Most popular low cost PIC16F877A microcontroller is the main controller within the mobile robot which is communicates with the remote host computer through the wireless RF link. Remote host is acting as master and microcontroller on robot acting as slave to transmit asynchronous serial data with RS232 data format between these two devices. Hence according to the received command from the host, microcontroller generates relevant PWM switching signals according of controlling algorithm. This microcontroller has special a capability of generate the HPWM (Hardware PWM) which is help to generate the two PWM signal simultaneously without affecting execution of other part of program.

To establish the communication link between the mobile robot with the remote host computer, single chip Radiometric BIM2-433 -64 RF communication IC is used which is operating in the 433 MHz frequency band.

The power supply for the mobile robot is big issue. Because DC motors consume huge power to overcome friction during the operation. Hence Sealed Led Acid rechargeable two 6V & 4 AMPhr batteries are used to power the mobile robot.
Conclusion

The voice recognition software has an accuracy around 75% in correctly identifying a voice command. But it is highly sensitive to the surrounding noises. There is a possibility of misinterpreting some noises as one of the voice commands given to the robot. Also the accuracy of word recognition reduces in face of the noise. The sound coming from motors has a significant effect on accuracy.

There are some drawbacks in the mobile platform. The rechargeable 6V batteries carried onboard makes it too heavy. Hence we had to use powerful motors to drive the robot making the power consumption higher. So we had to recharge the batteries quite often. The mobile platform had some problems in turning due to the heaviness of itself. The back freewheel used to get stuck when turning specially in reverse motion. Hence we suggest that steering mechanism will be a better option.