VOICE CONTROLLED BOTS

Team : SOME_VAGUE

This project basically aims at creating a voice recognition module that could be used for different systems. For the demonstration we have also created a four-wheeled RF bot and a biped bot . Both of them will be controlled by the voice commands.

Prateek , Aditya , Praveen , Dheerendra

6/4/2013

VOICE CONTROLLED BOTS

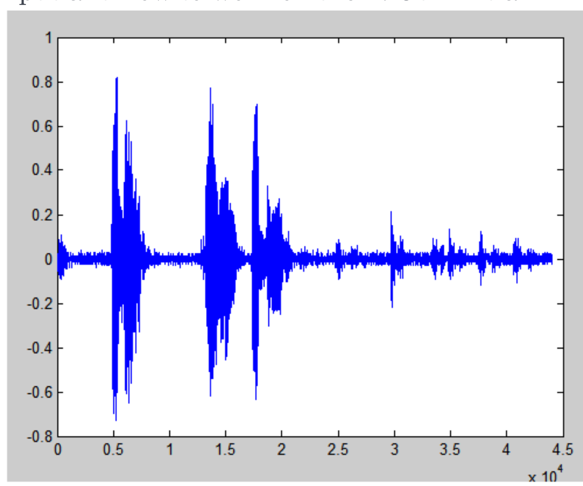
Team : SOME_VAGUE

As a part of our ITSP Project , we have created a voice recognition module and two different bots (4-wheeled RF bot and A biped). This document aims as describing of what we have done in our project , how we have done it , the difficulties faced while doing the project , components required , the things we have learnt , how this can be used in our general life purposes etc. So lets begin

VOICE RECOGITION

Our Initial Approach :

We started to learn about how to take voice as input and how to work on them. Our initial idea was to match two words by string comparison methods. But the data which we got was totally raw and unusable. The graph is shown here. And we were unable to draw any conclusion from this. So we started to research about voice processing techniques.



Before processing we extracted different words from an audio file. For this we kept the minimum sound that can be considered as noise as 0.2. Generally it is chosen as the sounds in the beginning. But sometimes it was quite low so we set the threshold as 0.2 which is good when used inside the rooms.

The next challenge was to build up a model where we can compare our audio signals

. After a lot of study we decided to use **Hidden Markov Model** for voice comparison .

Our Method for Voice recognition :

The following steps were followed for voice recognition using **Hidden Markov Model**

Obtaining the signal

The first part is obtaining signal from microphone. This is easily achieved by using the audiorecorder object of MATLAB. The getaudiodata function of audiorecorder object gives the signal data in form of column of numbers representing the audio signal.

Processing the signal into observation

Then comes the feature analysis part of processing.

Filtering / pre-emphasis

First the signal is passed through a part of code which acts as a high pass filter known as fir filter . This spectrally flattens the signal.

Voice Activation Detection

The signal is now passed to a function which finds the part of signal containing the spoken word and thus returns the signal without silence. Now for the word obtained we proceed further. Now onwards

Signal would refer to the extracted words.

Finding cepstral coefficients and delta coefficients

Using some predefined functions of MATLAB we convert the given signal into a set of observations which can be used for the model. The various operations that are performed are breaking the signal into windows , then applying the window function for each (hamming window is used) , then autocorelation coefficients are obtained, from these coefficients we obtain the cepstral coefficients by levinson durbin algorithm , further these coefficients are weighted to normalize , and the last step involves in finding the delta cepstral coefficients. Thus we get two set of coefficients which are merged to form row of 24 elements for each window and this completes the conversion into observation sequence.

Choice of model parameters

The number of states we selected was 6 after analysis of number of states versus the error made by the system / model. The data being obtained from paper by Rabiner over speech recognition by HMM model.

Next the HMM model to be used is selected to be Left-to-Right as the speech signals have parts which follow one after the other to form a meaningful word. Left-to-Right model has property that the system can move into the same state or a state ahead of it and not backward. The other type being ergodic model where system can move from any state to any other.

Training the system/model

Training the system here means determining the model parameters so that the given observation sequence used to train the model is declared highly probable by the system and others low.

Parameters

The parameters for HMM model are transition probabilities between the states represented by matrix A

(whose elements at ith row , jth column represent transition probability form ith state to jth), then comes the probabilities of occurrence of each observation in a particular word represented by B_ik meaning probability of kth observation being in ith state , then are the initial probabilities for occurrence of a given state at beginning time. In our case since we have dealt with continuous data the B_ik is represented as a gaussian distribution.

Initial estimation of the parameters

For better convergence of the estimation by K-Means algorithm we need to have good initial estimation of the parameters. The transition probabilities are made equal for the states into which the system may go. Rest are made zero. The mean and variance for the gaussian distribution are calculated from the 5 data sets obtained. The pi values for each state are set to be equally likely.

Readjusting the model parameters

The method used for performing this is based on classic work of BAUM and his colleagues .

For this various variables are defined as follows--

alpha(t,i)--- (forward variable) The probability of partial observation O1,O2,O3.... Ot (until time t) and ith state at time t, given the model .

beta(t,i)--- (backward variable) The probability of partial observation O(t+1) till end with ith state at time t.

gamma(t,i)--- The probability of observation sequence given the model with state ith at time t.

eta(t,i,j)--- The probability of observation sequence given the model with state ith at time t and jth at time t+1.

delta(t,i)--- The best score(highest probability) along a single path , at time t , which accounts for first t observations and end in state ith.

Now we have following adjustments--

pi(i) = expected frequency (number of times) in state ith at time (t=1)= gamma(1,i)

A(i,j) = (expected number of transitions from state ith to jth)/(expected number of transitions from state ith) = (summation over t from 1 to T-1 (eta(t,i,j)))/(summation over t from 1 to T-1 (gamma(t,i)))

mean(i)= (summation over t from 1 to T (gamma(t,i)*O(t))) /(summation over t form 1 to T (gamma(t,i)))

variance(i)=(summation over t from 1 to T (gamma(t,i)*(O(t)-mean(i))*(transpose(O(t)-mean(i)))))/(summation over 1 to T (gamma(t,i))).

Thus the new model parameters are obtained and the HMM model for the word is obtained.

Identifying the word

The probabilities for a given sequence is obtained with the different HMMs and highest among those is declared as the required word (probability must be greater than a threshold).

Obtaining the probability

The algorithm used to find the probability is viterbi algorithm. It finds the highest probability among the best state sequence for the given observation sequence.

Note : We have also attached some of the documents which have the detailed descriptions of the algorithms which we have used in our system.

Possible Uses of Voice recognition

As extension to our project

The model which we have used is basically developed for four basic commands but it can be extended to many more systems. It is a robust model which can be used for even more than 100 words which sound dissimilar. And using this we can build A voice controlled home, Tvs , washing machines etc all controlled by voices. Imagine How good it will be.

RF 4-Wheeled Bot

This is a simple XLR8 bot Which is Remote Controlled and the remote will be attached to the Arduino which will be controlled from the Matlab.

Components Used :

Mechanical:

- 12V DC Lead Acid battery – 1 pc
- 12V DC motors 200rpm – 4 pc
- Tires (7cm diameter) – 4pc
- Wooden Chassis – 20 X 25 cm
- L-Clamps
- Nut-bolts
- Styrofoam (for body)

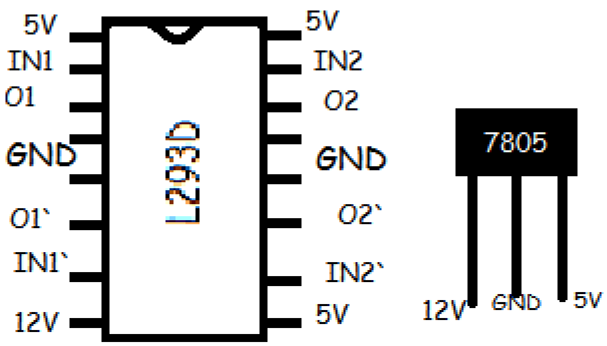
Electronics :

- An RF transmitter and receiver circuit (Used in China Cars)
- ARDUINO
- PCB
- IC – 7805 , L293D
- Heat Sink
- IC holder
- Wires and bug Strips
- LED's

How we made it ?

This is the simplest bot which one can make of himself.

Make a motor driver circuit using L293D and 7805. The input (IN) will be received from the rf circuit and the output(0) will be passed to the motors. 5V will be supplied from 7805



Fog : Pin diagrams of Both ICs

We have used the simple differential mechanism for the turning. You can get the explanation of differential mechanism from Google. The basic is , for turning left the tires turns backwards and the right tires turns forward this will turn the bot left And the same will follow opposite for turning right.

There is also a beautiful **LED simulation** on this bot which denotes the turning of the bot. There are LEDs connected to all sides and those LED glows which side the bot is turning.

THE REMOTE :

We studied the design of remote of the Rf Circuit so that we can use it with Arduino. For transmission there are some wires whose voltage changes. We identified those wires and Connected them with Arduino. For the motion we changed the voltage on those wires which made the motion of the bot.

THE BIPED BOT

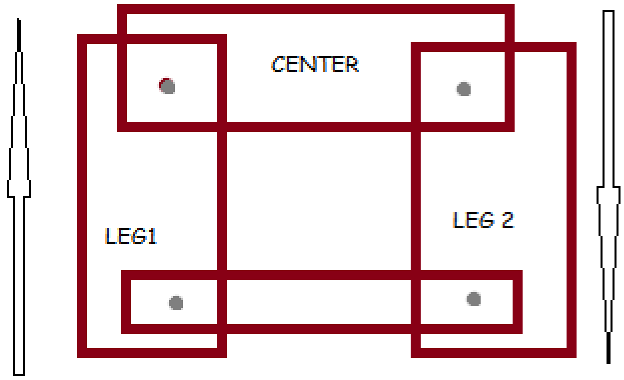
This is a bot which has 3-servo leg based walking mechanism . It walks by lifting one leg then turning on other leg and then lifting 2^{nd} leg and turning on 1^{st} leg.

Components used :

- 3 servos one with high torque.
- Wooden blocks and wooden sheets
- Wires , tapes and L-Clamps
- Nut Bolts
- Bondtite and Fevicol

Walking mechanism:

The main body is made is made of two parallelograms which can move parallel to each other.



The nuts attached are not fixed while they provide an axle kind of thing for the parallelograms to move. One servo is attached to each of the legs. And the High torque servo is attached at the center. The Servos are attached very compactly to each other so that the center servo can't turn. So when it is given a command to turn it lets the one of the legs to lift while the other rests on the ground. Now the servo on which it is standing turns to move the body.

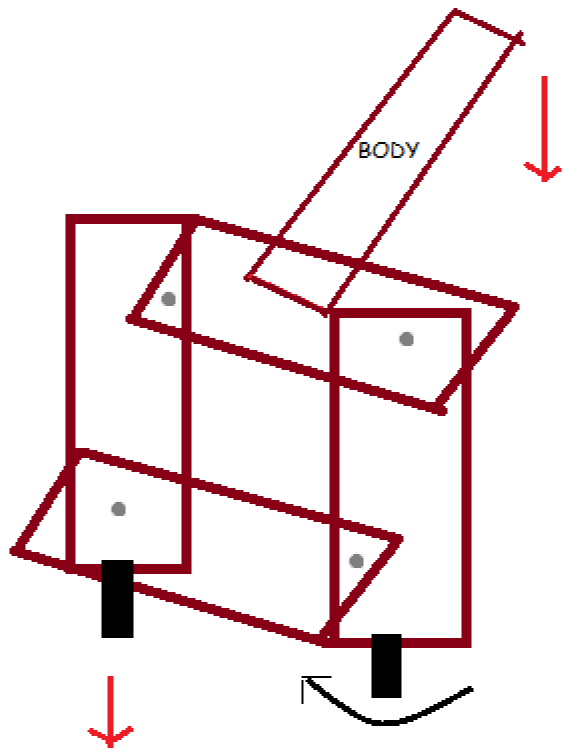
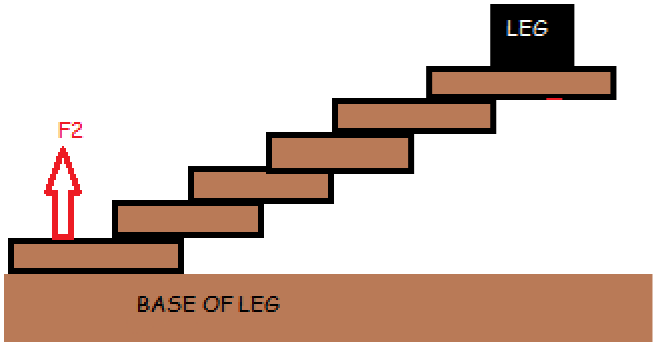


Fig : Turning of the leg and the torque getting balanced

This seems to be simple but the major problem which comes is of balancing the body on legs. So to balance the torque, as one leg lifts the body turns. And the weight of the body is adjusted so that the torque becomes equal on both sides of the leg which is on ground. To do this we have kept wooden blocks at the far end on the top of the bot. The heavy blocks provides good torque and balances the body.

The other major factor which allows the balancing of the bot is wide legs with staircase to sustain the torque. Due to the stair case kind of thing of the legs the sidewise toppling gets prevented. The force F1 which comes on the bot get transferred at the far end of the base so the bot becomes more stable and hence toppling is prevented.



The Rib of the body is tightly attached as it has to hold most of the body weight. And it also holds the upper weights. So it is backbone of the bot.

The Videos of both the walking bots is attached.

TEAM SOME_VAGUE :

Aditya Kumar Akash

120050046 , CSE Department

Adityakuamrakash@gmail.com

Prateek Chandan

120050042 , CSE Department

Prateekchandan5545@gmail.com

Praveen Agrawal

12D020030 , Chem Department Dual

Praveen.agrawal0@gmail.com

Dheerendra Singh Rahtor

120050033 , CSE department

Dheeru.rathor14@gmail.com

Mentor :

Naveen Gunturu

Manager:

Prajwal raut

ACKNOWLEDGEMENT

For the completion of our project we thank STAB and the Robotics club for their extreme support and the following people :

Our Mentor and Manager , Sudarshan , Jayesh and some of our friends.

For most of the Matlab Works we took help from its documentation present at www.mathworks.in

For the Biped ideas we took help from www.youtube.com

We also went to many of the online sources to learn about the voice processing . You can get it from our facebook group https://www.facebook.com/groups/somevague/ We have all our discussions and useful links over there.

To get the codes and the software get the link from our wiki page.

The used documents is attached with this.

THANKS.