INTERNATIONAL JOURNAL OF RESEARCH SCIENCE & MANAGEMENT

SPEECH SYNTHESIS USING HIDDEN MARKOV MODEL AND APPLICATION OF VOICE CONVERSION

Tejas Arun Shinde[1]*, B G Subramanian [2]
[1*2]School of Electronics Engineering (SENSE), Vellore Institute of Technology, Chennai-600127, India
Correspondence Author: shinde.tejas2013@vit.ac.in

Keywords: Speech synthesis, LPC, HMM, Voice conversion.

Abstract

Speech synthesis, an electronics system talks like human has grown in popularity over the last few years. This paper gives a general overview of techniques used in speech synthesis. One instance of these techniques, called hidden Markov model (HMM) based speech synthesis, has recently been demonstrated to be very effective in synthesizing acceptable speech. This paper also contrasts these techniques with the more conventional technique, calledlinear predictive coding (LPC) that has dominated speech synthesis over the last decade. The problem in implementation, advantages and drawbacks of these synthesis techniques are highlighted. Finally, advanced techniques for future developments are described. This paper gives brief idea about application of voice conversion using a codebook method.

Introduction

Speech is the primary means of communication between people. Speech synthesis, automatic generation of speech waveforms, has been under development for several decades. Recent progress in speech synthesis hasproduced synthesizers with very high intelligibility but the sound quality and naturalness still remain a major problem. It has wide applications like, in-car navigation systems, e-book readers, voice-over functions for the visually impaired, and communication aids for the speech impaired. Singing speech synthesizers, communicative robots, speech-to speech translation systems and spoken dialog systems are recent famous application of speech synthesis.

There are two main phases in speech synthesis. The first one is analysis of speech, where the input is transcribed into a phonetic or some other linguistic representation, and the second one is the generation of speech waveforms, where the acoustic output is produced from this phonetic and prosodic information. These two phases are usually known as high- and low-level synthesis. A simplified version of the procedure is presented in Figure 2.2. The input might be for example data from a word processor, standard ASCII from e-mail, a mobile text-message, or scanned text from a newspaper. The character string is then pre-processed and analyzed into phonetic representation which is usually a string of phonemes with some additional information for correct intonation, duration, and stress. Speech sound is finally generated with the low-level synthesizer by the information from high-level one.

Speech production and vocoder

The speech production process is approximated using a digital filter shown in Fig. 1.Implementation this filter is based on the source filter theory of voice production and istherefore called the source filter model. This model uses a white excitation (pulse train or noise) [1]as input and filtered with asingle resonance filter to model the acoustic speech pressurewave, where spectral envelopes of the glottal flow,vocal tract resonance, and lip radiation effect are modelled all together by the single resonance filter. This modelcomprises: voicing information, fundamental frequency and spectral envelope represented by, e.g., linear predictive coefficients [2], and speech waveforms canbe reasonably reconstructed from the sequence of these acoustic parameters.

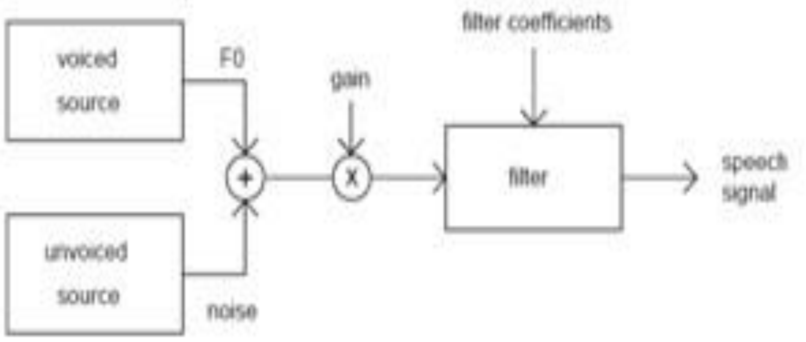

Fig 1 Source-filter model of speech

Methodology

Linear predictive coding

Linear predictive analysis is one of the most powerful speech analysis techniques [3]. It is based on the source filter model of speech production. This method has become the predominant technique for estimating the basic speech parameters, e.g. pitch, formants, spectra, vocal tract area functions, and for representing speech for low bit rate transmission or storage.

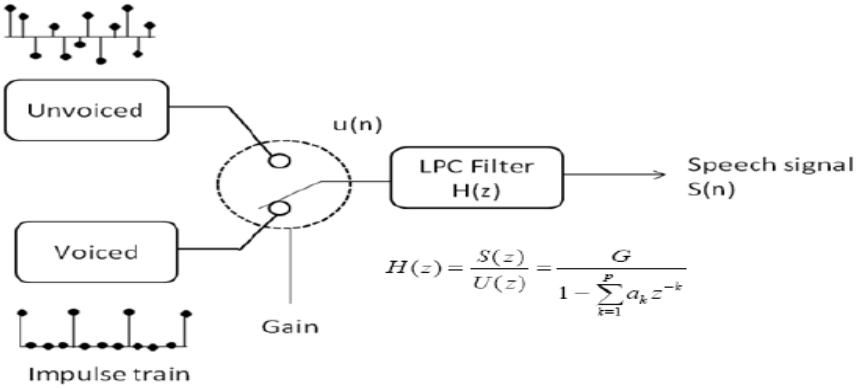

Fig 2LPC Speech Production Model

The Fig 2 shows a LPC speech production model [2]. In this case, the composite spectrum effects of radiation, vocal tract, and glottal excitation are represented by a time varying digital filter whose steady-state system function is of the form

$$H(z)=\frac{S(z)}{U(z)}=\frac{G}{1-\sum_{k=1}^{p}a_k z^{-k}}$$

This system is excited by an impulse train for voiced speech or a random noise sequence for unvoiced speech. Thus, the parameters of this model are: voiced/unvoiced classification, pitch period for voiced speech, gain parameter G, and the coefficients a_k of the digital filter. The LPC has two key components: analysis or encoding and synthesis or decoding. The analysis part of LPC involves examining the speech signal and breaking it down into segments or frames. The LPC analysis is done in each frame which involves

1. Determination of LPC filter coefficients
2. Gain parameter
3. Voiced/Unvoiced classification)
4. Pitch (for voiced frame)

The LPC analysis is done by the sender. Once these parameters are determined, they can be transmitted. For reconstitution of the segment of the speech LPC synthesis of the segment is done. Here if the frame is a voiced frame then the input to the filter is an impulse train with period equal to that of the pitch period. If the frame is unvoiced then the input to the filter is a random white noise.

The input signal is sampled at a rate of 8000 samples per second. This input signal is then divided into a number of segments or frames. The size of each frame is between 20 and 30 milliseconds. Each frame is analyzed and transmitted to the receiver.

Determination of LPC coefficients

For the system of Fig., the speech samples *s(n)* are related to the excitation *u(n)* by the simple difference equation

$$s(n)=\sum_{k=1}^{p}a(k)s(n-k)+Gu(n)$$

A linear predictor with prediction coefficients, α_k is defined as a system whose output is

$$\tilde{s}(n)=\sum_{k=1}^{p}\alpha_k s(n-k)$$

The system function of a *pth* order linear predictor is the polynomial

$$p(z)=\sum_{k=1}^{p}\alpha_k z^{-k}$$

The prediction error, *e(n)*, is defined as

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} \alpha_k s(n-k)$$

It can be seen that the prediction error sequence is the output of a system whose transfer function is

$$A(z) = 1 - \sum_{k=1}^{p} a_k z^{-k}$$

If the speech signal obeys the model exactly, and if $\alpha_k = a_k$ then *e (n) = G u (n).* Thus, the prediction error filter, *A(z),* will be an inverse filter for the system, *H(z),* i.e.,

$$H(z) = \frac{G}{A(z)}$$

Determine a set of predictor coefficients α_k directly from the speech signal which has good estimate of the spectral properties of the speech signal. The basic approach is to find a set of predictor coefficients that will minimize the mean-squared prediction error over a short segment of the speech waveform.

Hidden markov model(HMM)

A hidden Markov model (HMM) is denoted by $\lambda=(A, B,\pi)$ is characterized by the initial state distribution, $\pi$the state transition matrix **A**, and the emission probability matrix **B**.

$B (l_t)=N (l_t ; \mu_i ,\sum_i)$

$B = \frac{1}{\sqrt{(2\pi)^2\sum}}\exp \{1/2(l_t - u_i)^T \sum_i^{-1}(l_t - u_i)\}$

Where$[ui]_{d*1}$ and $[\sum_i]_{d*d}$is covariance matrix, d is the dimension of theacoustic parameters; and $o_t$is an observation vector, whichconsists of the vocoder parameters at frame t.

Let set of speech parameters

$L= [L_1^T,L_2^T,\ldots\ldots\ldots,L_T^T]^T$

Wis corresponding linguistic specifications whichis used for training of HMM.$l=[l_1^T,l_2^T,\ldots l_T^T]^T$and w bespeech parameters and corresponding linguistic specification that we want to generate at synthesis time.

Training

λmax =arg max $p(L|\lambda,W)$

$p (L|\lambda,W)=\sum_{q=1}^{T} \pi_{q0}\prod_{t=1}^{T} a_{qt-1qt}b_{qt}(L_t)$

Synthesis

l_{max}=arg max $p(l|\lambda max , w)$

$q=(q_1,q_2,\ldots\ldots..q_T)$

Three basic problems

Having formally defined hidden Markov models, we may now turn to the three classical problems that must be solved in order to apply hidden Markov modelling to real-world tasks [4]. These problems are the following:

Evalution

Evaluation or computing *P*(*Observations* | *Model*). This allows us to find out how well a model matches a given observation sequence. The main concern here is computational efficiency. Forward backward algorithm is one of best solution for such problem.

We start with forward algorithm for markov chain*and then extend it for HMM* .Let westart the *forward variable* for Markov chains:

$\alpha_t(i) = P(S_t = i \mid \lambda)$ For all $i \in L^{t+1}$ and $t \geq 0$ A computationally efficient formula can be derived for the forward variable is given as $\sum_{i \in L} \alpha_i(t-1) a_{ij}$ .It depends on α_i (*t*-1) and the transition probabilities a_{ij}.

In order to finally solve the evaluation problem to handle hidden Markov models. We first redefine the forward variableand backward variable as

$$\alpha_t(i) = P(L_1, L_2 .... L_t, q_t = S_i / \lambda)$$

$$\beta_t(i) = P(L_{t+1}, L_{t+2}, ... L_T / q_t = S_i, \lambda)$$

We can solve for the forward variable inductively in the same manner as before; the emission probabilities just need to be multiplied in. We also introduce a terminating condition: once we have the forward probabilities for all states at time *t* (the final time frame) we sum over the state space to yield the quantity that we were looking for in the first place P(L| λ).This is the most fastest way of computing the likelihood of HMM, with the running time of the order N^2

1. **Initialization**

$$\alpha_1(i) = \pi_i b_i(L_1) \qquad 1 \leq i \leq N$$

$$\beta_T(i) = 1$$

2. **Induction**

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} b_j(L_{t+1})\right] \qquad 1 \leq j \leq N$$

$$1 \leq i \leq N$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(L_{t+1}) \beta_{t+1}(j)$$

3. **Termination**

$$P(L / \lambda) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i)$$

For all t=T

$$P(L / \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Decoding: finding the best path

The forward algorithm allows the probability of an HMM to be evaluated with respect to a given observation sequence, but it does not give any indication as to the underlying state sequence. There are several possible ways in solving the second problem i.e. to find the optimal state sequence from the given observation sequence. The difficulty lies within the optimality of the sequence. This optimality criterion maximizes the expected number of correct individual states. To implement this we define a new variablei.e. the probability of being in the state S_i at the time t with the observation sequence L and the model λ . The above equation can be expressed in simple terms in forward and backward variables as

$$\gamma_t(i) = P(q_t = S_i / L, \lambda)$$

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{P(L / \lambda)}$$

Since $\alpha_t(i)$ accounts for the partial observation sequence $L_1, L_2....L_t$, and $\beta_t(i)$ accounts for the rest of the sequence $L_{t+1}, L_{t+2},...L_T$ given state S_i at t.

In this case we are supposed to find the maximum value of γ in that particular row and find the index value thereby we can track the sequence from the given observation sequence.

Training: estimating model parameters

This method is to adjust the parameters $(A, B, \pi)$to maximize the probability of the given observation sequence for the given model. There is no known way to solve the problem analytically so that probability will be maximized. In fact, given any finite observation sequence as training data, there is no optimal way of estimating the model parameters.

$l_{max} = \arg(\max)_l\ p(l|\lambda_{max}, w)$----------------1

$= \arg(\max)_l \sum_q p(l,q|\lambda_{max}, w)$--------2

$= \arg(\max)_{l,q}\ p(l,q|\lambda, w)$---------------3

$= \arg(\max)_{l,q}\ p(l|q, \lambda_{max})P(q|\lambda_{max}, w)$--4

$= \arg(\max)_l\ p(l|q_{max}, \lambda_{max})$-----------5

$= \arg(\max)_l \prod_{t=1}^{T'} N(l_t; u_{qmax}, \sum_{qmax,t})$--6

$q_{max} = \arg(\max)q\ P(q|\lambda_{max}, w)$--------------7

The maximization problem of (7) i.e. q_{max}can be solved bystate duration probability distributions. We can maximize (5) by maximizing p (l, q | λ) with respect to l as the predetermined state sequence q_{max} is given.

Implementation

Training and Synthesis Procedure for one word

1.To appreciate LPC for speech synthesis we collect ten samples of the same word are obtained from ten different persons. A given sample of the word is divided into non – overlapping frames of size 256 samples. A frame is analyzed to get LPC features of the order 15 and this LPC vector is the symbol we consider. In second case instead of using whole word we divide it into phonemes and it is used to get LPC features. Excitement signal with respect to the LPC we obtained is determined and stored for every word orphonemes and this signal has to be used during synthesis. Therobust pitch estimation method is available so the excitement signal can be replaced by the impulse train having pitch period as the period. However, in this work we have first used the excitement signal and then impulse train. At synthesize side using LPC vector of input and impulse train we obtain synthesise voice. Here observed that speech quality and naturalness is not adequate.

2.To get good quality of sound we train our parameter with HMM. For CHMM training an 1 state left to right model is assumed with arbitrary initial parameters. With the observation sequence obtained for the given word model is updated using the estimation formula referred in HMM section. Similarly it is repeated for all the samples of words. Then, the common voice model is obtained by averaging all the model parameters obtained for individual words .Taking the result of first iteration as the initial value for second iteration similar process is repeated. This process is repeated for several iterations and at last we get state sequence which is used as speech parameter as LPC parameters and the stored excitement signal, the speech is reconstructed frame by frame speech synthesis. These frames are concatenated as a complete word.

3. **Voice conversion**

Voice transformation is performed in two steps, training stage and transformation stage. We approachvoiceconversion based on codebook using vector quantisation method.

Vector quantization:-

VQ is a technique used for signal processing as well as data compression.

Working-

It works by dividing a large set of vectors into groups having approximately the same number of points closest to them. Each group is represented by its Leader point.

A targeted speechsample is digitized into several no of frames with sampling time 32 ms. Frames are divided intogroups. Each group is compared with 15 LPC coefficients. According to the comparison, groups are reconstituted based on minimum distance with LPC.Further, no of groups are reduced by merging groups in certain ratio.Resultant group is termed as Codebook.

Experiments were performed with the database used in order to train the mapping rules consists of the long speech in English language by one male speaker. Speech signals were digitized at 8 KHz sampling frequency and the order of LPC was set to 15.The source signal are transformed using the mapping rules such that the synthesized speech possesses the personalities of the target speaker. Two evaluation tests for the proposed method were carried out, the capability of converting excitation signal, and the quality of transformed speech.

Result

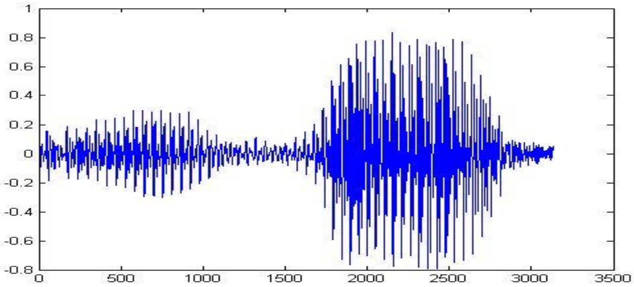

Fig.3 original voice sample of "TEJAS"

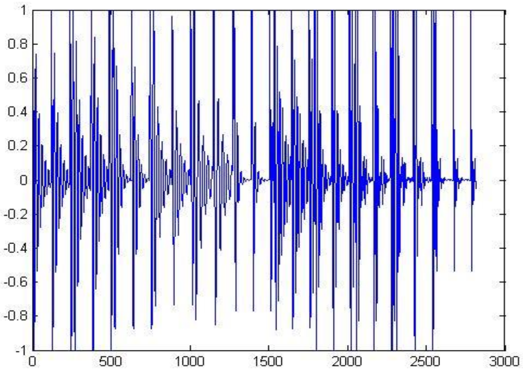

Fig.4 synthesise TEJAS by LPC

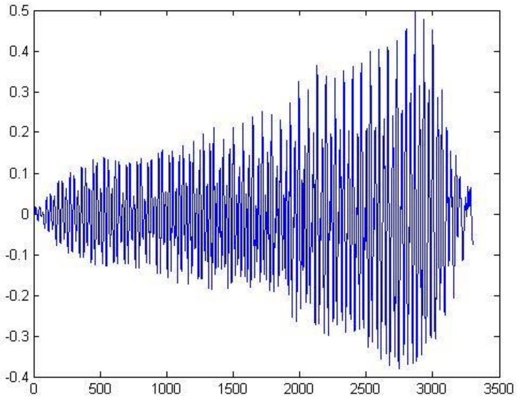

Fig.5 Original sample "ek"

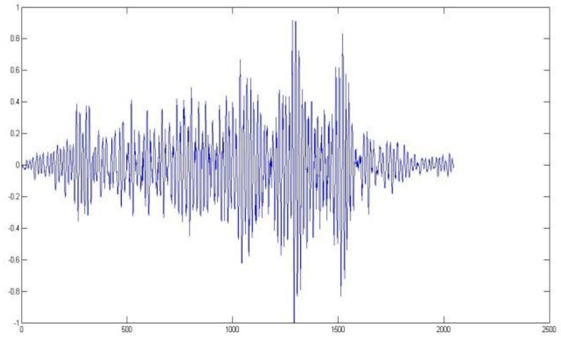

Fig.6 Synthesise sample "ek" by HMM training

Conclusion

An effective speech synthesis using LPC and HMM with voice conversion has been implemented. Further we are planning to train HMM model with dynamic features constraints to give more naturalness and smoothness of synthesis sound.

Acknowledgment

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. This project has been a great learning experience for us and its timely completion would not have been possible without the help our school faculties and facilities provided by VIT management.

References

1. Keiichi Tokuda, Yoshihiko Nankaku,Tomoki Toda,Heiga Zen, Junichi Yamagishi and Keiichiro Oura , "Speech Synthesis Based on Hidden Markov Models", Proceedings of the IEEE, Vol. 101, No. 5, May 2013.
2. S.Martincic – Ipsic and I. Ipsic, "Croatian HMM based speech synthesis", Rijeka, Croatia, 2006.
3. John Makhoul - "Linear Prediction: A Tutorial Review", Proceedings of the IEEE, Vol. 63, No. 4, April 1975.
4. L.R.Rabiner and R.W.Schafer, "A Tutorial on Hidden Markov Model and selected applications in Speech Recognition " Proceedings of the IEEE, Vol. 77, No. 2, February 1989 .