FORMANT ANALYSIS AND SYNTHESIS USING HIDDEN MARKOV MODELS

Alex Acero

Microsoft Research
One Microsoft Way
Redmond, Washington 98052, USA
http://research.microsoft.com/srg

ABSTRACT

This paper describes a unifying framework for both formant tracking and speech synthesis using Hidden Markov Models (HMM). The feature vector in the HMM is composed by the first three formant frequencies, their bandwidths and their delta with time. Speech is synthesized by generating the most likely sequence of feature vectors from a HMM, trained with a set of sentences from a given speaker. Higher formant tracking accuracy can be achieved by finding the most likely formant track given a distribution of the formants of every sound. This data-driven formant synthesizer bridges the gaps between rule-based formant synthesizers and concatenative synthesizers by synthesizing speech that is both smooth and resembles the speaker in the training data.

1. INTRODUCTION

Both rule-based formant synthesis [2] and concatenative synthesis [4][5] yield unnatural speech, although for different reasons. Concatenative synthesizers sound quite natural within a unit, but overall naturalness can be low due to the presence of discontinuities at unit boundaries. Rule-based formant synthesizers never exhibit such discontinuities, but their simplified model of each sound never results in a high level of naturalness either.

We propose a technique, based on Hidden Markov Models (HMM) with differential coefficients, that results in speech synthesizers that have the compactness and smoothness of rule-based formant synthesizers, yet the synthesized speech is significantly more natural and resembles (like in the case of concatenative synthesizers) the original speaker used to train the model. The goal is to have a *data-driven formant synthesizer*.

To train a statistical model for formants, we need to be able to track them. Automatic formant trackers [8] have attracted a great deal of interest, mostly because formant frequencies are critical in speech perception. Many such algorithms often miss a formant when one is present, insert a formant when there is none, or mislabel them (such as label F1 as F2, or F3 as F2), mostly because of incorrectly pruning the correct formant at the frame level. The proposed HMM model used to generate formants for speech synthesis will also let us accurately track formants, which will be considered hidden variables.

Section 2 describes how to generate formants from a HMM. HMM-based formant tracking is presented in Section 3. Estimation of the HMM parameters is shown in Section 4. Section 5 deals with smoothing a raw formant track using HMMs. Section 6 describes the rest of the model needed to synthesize speech and Section 7 the conclusions and future work.

2. FORMANT GENERATION

In this section we will describe a method based on HMMs that generates parameters to drive a speech synthesizer. A method to synthesize speech based on HMM with dynamic features and cepstral vectors was presented in [7]. We will now outline this method.

Let λ be a *N*-state left-to-right HMM with a feature vector **x** of dimension *M*. We now desire to generate a sequence **X** of *T* feature vectors

$$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)' \tag{1}$$

from this HMM that maximizes the overall likelihood:

$$p(\mathbf{X}|\lambda) = \sum_{\mathbf{q}} p(\mathbf{X}|\mathbf{q},\lambda)\, p(\mathbf{q}|\lambda) \tag{2}$$

over all possible state sequences

$$\mathbf{q} = (q_1, q_2, \cdots, q_T)' \tag{3}$$

In practice, the Viterbi approximation is used in (2), and the state sequence $\hat{\mathbf{q}}$ can be either provided by the prosody module of a TTS system, or maximized independently of **X** as

$$\hat{\mathbf{q}} = \arg\max_{\mathbf{q}} \left[\ln p(\mathbf{q}|\lambda) \right] \tag{4}$$

in an iterative way [7]. With this state sequence $\hat{\mathbf{q}}$, the Viterbi approximation in (2) yields

$$\hat{\mathbf{X}} = \arg\max_{\mathbf{X}} \left[\ln p(\mathbf{X}|\hat{\mathbf{q}},\lambda) \right] \tag{5}$$

Now let's assume that the output distribution of each state *i* is modeled by one Gaussian density function with a mean μ_i and covariance matrix Σ_i. The HMM model λ is the set of all means and covariance matrices for all *N* states:

$$\lambda = (\mu_1, \Sigma_1, \mu_2, \Sigma_2, \cdots, \mu_N, \Sigma_N) \tag{6}$$

Therefore the log-likelihood in (5) is given by

$$\ln p(\mathbf{X}|\hat{\mathbf{q}},\lambda) = -\frac{TM}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{T} \ln|\Sigma_{q_t}| - \frac{1}{2}\sum_{t=1}^{T} (\mathbf{x}_t - \mu_{q_t})' \Sigma_{q_t}^{-1} (\mathbf{x}_t - \mu_{q_t}) \tag{7}$$

Maximizing **X** in (7) leads to the trivial solution $\hat{\mathbf{X}} = (\mu_{q_1}, \mu_{q_2}, \ldots, \mu_{q_T})'$, a piecewise function which has discontinuities at state boundaries and thus not likely to represent well the physical phenomena of speech.

This problem arises because the slopes at state boundaries do not match the slopes of natural speech. To avoid these discontinuities, we would like to match not only the target formants at each state, but also the formant slopes at each state. To do that, we augment the feature vector $\mathbf{x}_t$ at frame *t* with the delta vector $\mathbf{x}_t - \mathbf{x}_{t-1}$. Thus we increase the parameter space of λ with the corresponding means δ_i and covariance matrices Γ_i of these delta parameters, and assume statistical independence among them. The corresponding new log-likelihood has the form

$$\begin{aligned} \ln p(\mathbf{X}|\hat{\mathbf{q}},\lambda) &= K - \frac{1}{2}\sum_{t=1}^{T}\ln|\Sigma_{q_t}| - \frac{1}{2}\sum_{t=2}^{T}\ln|\Gamma_{q_t}| \\ &- \frac{1}{2}\sum_{t=1}^{T}(\mathbf{x}_t - \mu_{q_t})'\Sigma_{q_t}^{-1}(\mathbf{x}_t - \mu_{q_t}) \\ &- \frac{1}{2}\sum_{t=2}^{T}(\mathbf{x}_t - \mathbf{x}_{t-1} - \delta_{q_t})'\Gamma_{q_t}^{-1}(\mathbf{x}_t - \mathbf{x}_{t-1} - \delta_{q_t}) \end{aligned} \tag{8}$$

Maximization of (8) requires solving several sets of linear equations. If Γ_i and Σ_i are diagonal covariance matrices, it results in a set of linear equations for each of the *M* dimensions

$$\mathbf{BX} = \mathbf{c} \tag{9}$$

where **B** is a tridiagonal matrix (all values are zero except for those in the main diagonal and its two adjacent diagonals), which leads to a very efficient solution [6]. For example, the values of **B** and **c** for *T*=3 are given by

$$\mathbf{B} = \begin{pmatrix} \frac{1}{\sigma_{q_1}^2} + \frac{1}{\gamma_{q_2}^2} & -\frac{1}{\gamma_{q_2}^2} & 0 \\ -\frac{1}{\gamma_{q_2}^2} & \frac{1}{\sigma_{q_2}^2} + \frac{1}{\gamma_{q_2}^2} + \frac{1}{\gamma_{q_3}^2} & -\frac{1}{\gamma_{q_3}^2} \\ 0 & -\frac{1}{\gamma_{q_3}^2} & \frac{1}{\sigma_{q_3}^2} + \frac{1}{\gamma_{q_3}^2} \end{pmatrix} \tag{10}$$

$$\mathbf{c} = \begin{pmatrix} \frac{\mu_{q_1}}{\sigma_{q_1}^2} - \frac{\delta_{q_2}}{\gamma_{q_2}^2} & \frac{\mu_{q_2}}{\sigma_{q_2}^2} + \frac{\delta_{q_2}}{\gamma_{q_2}^2} - \frac{\delta_{q_3}}{\gamma_{q_3}^2} & \frac{\mu_{q_3}}{\sigma_{q_3}^2} + \frac{\delta_{q_3}}{\gamma_{q_3}^2} \end{pmatrix}' \tag{11}$$

where just one dimension is represented, and the process repeated for all dimensions with a computational complexity of *O*(*TM*). The case of mixture of Gaussians and/or non-diagonal covariance matrices, significantly more involved, is presented in [7].

The maximum likelihood sequence $\hat{\mathbf{x}}_t$ are close to the targets μ_i while keeping the slopes close to δ_i for a given state *i*, thus generating a continuous function. In general if a state duration is long enough, the feature will reach its target, whereas for a short duration the target will likely not be reached. This matches well with the locus theory of speech production. Because of the delta coefficients, The solution depends on all the parameters of all states and not just the current state.

While the choice of dynamic features was critical to the smoothness of the synthesized speech, the synthesized speech of [7] exhibited formant bandwidths that are wider than the natural ones. This prompted us to investigate using formants as features instead of cepstrum. In our experiments we used 3-state tree-clustered context-dependent phone HMMs [4], where each state is modeled with one Gaussian density function with diagonal covariance matrices. A total of 24 parameters per state are then needed: 3 formant means, 3 formant variances, 3 bandwidth means, 3 bandwidths variances, as well as the corresponding delta parameters.

The first three formants and corresponding bandwidths generated are not sufficient to produce speech. In Section 6 we will analyze what additional parameters are needed to generate speech and how they can also be learned from data. In Section 4 we will see how to train this HMM model from a set of recordings.

3. FORMANT TRACKING

Formant trackers typically have two steps: 1) computation of formant candidates for every frame, and 2) determination of the formant track, generally using continuity constraints.

One way of obtaining formant candidates at a frame level is to compute the roots of a p^{th} order LPC polynomial. There are standard algorithms to compute the complex roots of a polynomial with real coefficients [6]. Each complex root z_i can be represented as

$$z_i = \exp(-\pi b_i + j2\pi f_i) \tag{12}$$

where f_i and b_i are the formant frequency and bandwidth respectively of the i^{th} root. Real roots are discarded and complex roots are sorted by increasing *f*, discarding negative values. The remaining pairs (f_i, b_i) are the formant candidates.

Traditional formant trackers discard roots whose bandwidth is higher than a threshold [8], say 200Hz, and formant alignment from one frame to another is generally done using heuristics. This implies that a given frame could have no formants, only one formant (either first, second or third), two, three or more..

In the proposed approach, no formant candidates are eliminated at the frame level. If the first *n* formants were desired, a maximum of *r* *n*-tuples are considered where *r* is given by

$$r = \begin{pmatrix} p/2 \\ n \end{pmatrix} \tag{13}$$

A Viterbi search is then carried out to find the most likely path of formant *n*-tuples given the HMM model. While the Viterbi search should search for the most likely formant track *and* the most likely state segmentation, in practice a sub-optimal search was carried out with a fixed state segmentation that has been computed by a standard HMM using mel-frequency cepstrum.

Let's understand intuitively how this formant tracker works. The *a priori* distribution for formant targets is used to determine which formant candidate to use. Formant continuity is imposed through the *a priori* distribution of the formant slopes. This algorithm produces *n* formants for every frame, including silence.

Since we are interested in obtaining the first three formants (*n*=3) and F3 is known to be lower than 4kHz, it is advantageous to downsample the signal to 8kHz to avoid obtaining formant candidates above 4kHz, and to let us use a lower order analysis which offers fewer numerical problems when computing the roots. In our experiments we have used *p*=14 which results in a maximum of *r*=35 triplets for the case of no real roots. We computed these LPC coefficients from 20-millisecond Hanning windows spaced every 10 milliseconds using the autocorrelation method.

4. HMM PARAMETER ESTIMATION

This section describes how to estimate the HMM model parameters from a set of recordings. The model parameters are trained through the standard EM algorithm.

For the initial model, we set all the state distributions to be the same. The 3 formant means were set to 500Hz, 1500Hz and 2500Hz respectively, and the bandwidth means were set to 100Hz for all three formants. The 3 formant standard deviations were set to 500Hz and the 3 bandwidth standard deviations were set to 100Hz. The means of the delta-formant and delta-bandwidth were all set to 0Hz. The standard deviations for the delta-formant and delta-bandwidth were all set to 100Hz. These values worked well empirically and didn't appear to be critical.

State segmentation for all utterances was done with a traditional HMM using mel-frequency cepstrum as feature. This segmentation was not altered in the training procedure, which results in a sub-optimal likelihood.

Given the initial model and the state segmentation, the three formants were tracked for every utterance in the training set according to the method in Section 3 (the Estimate step). Then means and variances for the formants, bandwidths, delta-formants and delta-bandwidths are re-estimated (the Maximize step) as the sample means and variances. This process is iterated till convergence. We noticed that 2 or 3 iterations were sufficient in our speaker-dependent experiments.

5. SMOOTHING FORMANT TRACKS

The formant track obtained through the method of Section 3 can be rough, and it may be desired to smooth it. Smoothing without knowledge about the speech signal would result in either blurring the sharp transitions that occur in natural speech, or maintaining ragged formant tracks where the underlying physical phenomena vary slowly with time. Ideally we would like a larger adjustment to the raw formant when the error in the estimate is large relative to the variance of the corresponding HMM state.

Let's define **X** as the real formant track, which is hidden, and **Y** as the observed formant track. The joint probability can be expressed as

$$p(\mathbf{Y},\mathbf{X}|\lambda) = p(\mathbf{Y}|\mathbf{X},\lambda)\,p(\mathbf{X}|\lambda) = p(\mathbf{X}|\lambda)\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t) \qquad (14)$$

where we have assumed that $p(\mathbf{Y}|\mathbf{X},\lambda)$ does not depend on λ, and that $\mathbf{y}_t$ depends only on $\mathbf{x}_t$. Furthermore, we can model $p(\mathbf{y}|\mathbf{x})$ as a Gaussian density function

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi)^{M/2}\prod_{j=1}^{M}\upsilon_j}\exp\left\{-\frac{1}{2}\sum_{j=1}^{M}\frac{(\mathbf{y}[j]-\mathbf{x}[j])^2}{\upsilon_j^2}\right\} \qquad (15)$$

The maximum likelihood estimate of **X** in (14) results in another set of tridiagonal linear equations similar to that in (9). The corresponding **B** and **c** for the example of *T*=3 are now given by

$$\mathbf{c} = \begin{pmatrix} \frac{y_1}{\upsilon_1^2} + \frac{\mu_{q_1}}{\sigma_{q_1}^2} - \frac{\delta_{q_2}}{\gamma_{q_2}^2} & \frac{y_2}{\upsilon_2^2} + \frac{\mu_{q_2}}{\sigma_{q_2}^2} + \frac{\delta_{q_2}}{\gamma_{q_2}^2} - \frac{\delta_{q_3}}{\gamma_{q_3}^2} & \frac{y_3}{\upsilon_{31}^2} + \frac{\mu_{q_3}}{\sigma_{q_3}^2} + \frac{\delta_{q_3}}{\gamma_{q_3}^2} \end{pmatrix}'$$

$$\mathbf{B} = \begin{pmatrix} \frac{1}{\upsilon_1^2} + \frac{1}{\sigma_{q_1}^2} + \frac{1}{\gamma_{q_2}^2} & -\frac{1}{\gamma_{q_2}^2} & 0 \\ -\frac{1}{\gamma_{q_2}^2} & \frac{1}{\upsilon_2^2} + \frac{1}{\sigma_{q_2}^2} + \frac{1}{\gamma_{q_2}^2} + \frac{1}{\gamma_{q_3}^2} & -\frac{1}{\gamma_{q_3}^2} \\ 0 & -\frac{1}{\gamma_{q_3}^2} & \frac{1}{\upsilon_3^2} + \frac{1}{\sigma_{q_3}^2} + \frac{1}{\gamma_{q_3}^2} \end{pmatrix}$$

We have observed that the estimates provided by the LPC roots are more accurate if the corresponding bandwidth is small. Thus we empirically set the estimate's precision $1/\upsilon_i^2 = \alpha / b_i^2$, with $0 \le \alpha \le \infty$ and b_i being the formant's bandwidth of frame *i*. The parameter α controls the degree of smoothing and varies between raw formants ($\alpha = \infty$) to synthetic formants ($\alpha = 0$).

Figure 1 shows an utterance from a male speaker with the smoothed formant tracks ($\alpha = 1$). Figure 2 compares the raw, smoothed and synthetic formants, where the match is quite good except perhaps for F3. When no real formant is visible from the spectrogram, the algorithm tends to assign a large bandwidth.

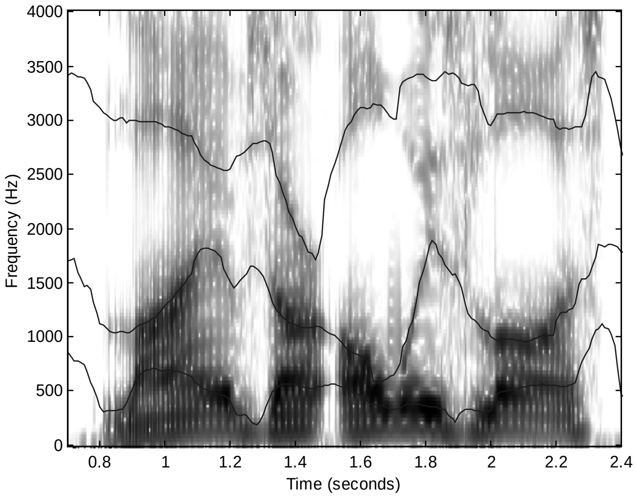

Figure 1. Spectrogram and 3 smoothed formants (α=1)

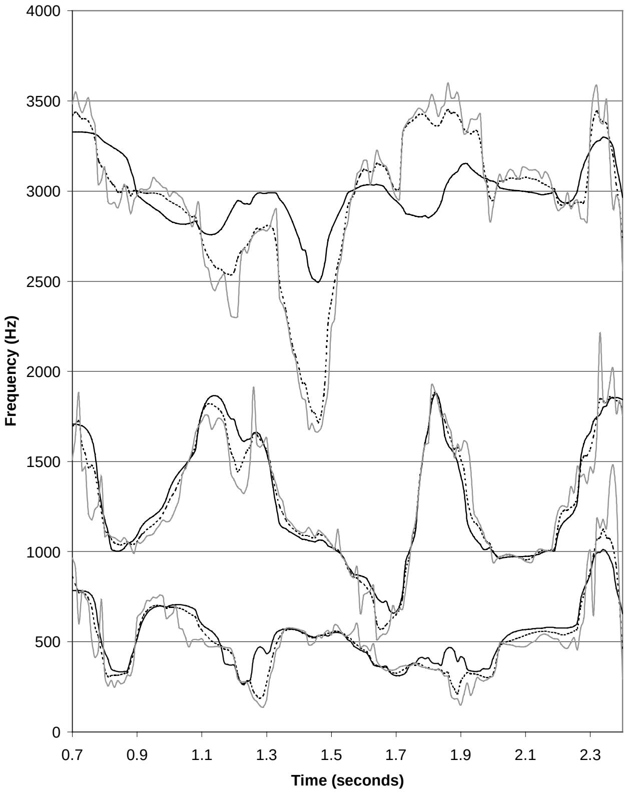

Figure 2. Raw formants (α=∞,ragged gray line), smoothed formants (α=1,dashed line) and synthetic formants (α=0,smooth solid line).

6. EXCITATION MODELING

Traditional formant synthesizers [2] have, in addition to the first 3 formants and bandwidths, other parameters, mostly dealing with source modeling. To obtain the excitation $e[n]$, inverse filtering was performed on the input speech $s[n]$ through a cascade of three second-order filters obtained from the smoothed formants and bandwidths of Sections 3 and 5. The spectrogram of $e[n]$ can be seen in Figure 3.

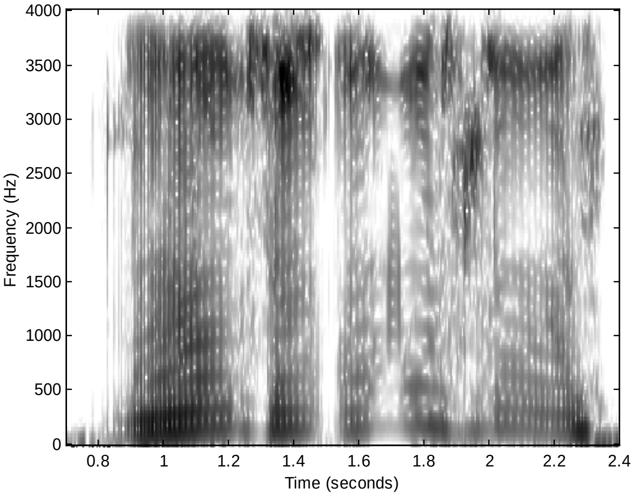

Figure 3. Spectrogram of the excitation $e[n]$ after removing the smoothed formant structure from $s[n]$.

To build a speech synthesizer we thus need to specify a model for $e[n]$. A first approximation $\tilde{e}[n]$ can be computed by passing either white noise (unvoiced speech) or an impulse train (voiced speech) through an LPC filter, that has been estimated from $e[n]$. The reconstructed speech, obtained by passing $\tilde{e}[n]$ through the smoothed formant track, exhibited some of the mechanic sound quality typical of LPC vocoders including buzzy voiced fricatives. In order to synthesize these LPC coefficients directly, the feature vector in the HMM was augmented with the gain and log-area ratios, which were trained in a standard way. The speech generated through this method was very intelligible but had lost some of the voice quality of the original speaker, perhaps because the averaging process was not done in an appropriate parameter domain.

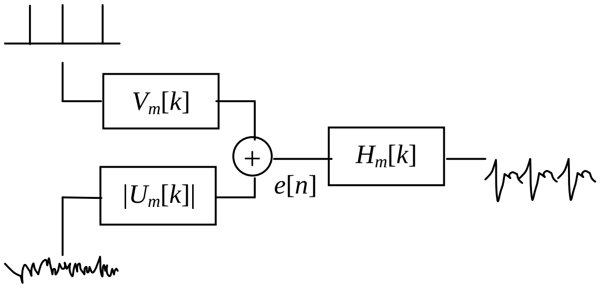

Figure 4. Mixed excitation model.

The mixed excitation model [1] decomposes $e[n]$ as a sum of a voiced component and an unvoiced component (See Figure 4) with $H_m[k]$ being the formant filter. The voiced component is generated by passing an impulse train through a filter $V_m[k]$ expressed in the frequency domain. The unvoiced component is white random noise passed through a filter $|U_m[k]|$. To reduce the number of parameters, we set the phase of $V_m[k]$ to 0, which did not result in any audible difference. A possible explanation is that the phase of $e[n]$ evolves slowly with frequency, since the formant model had already removed the rapid 180° phase shifts around formants. To reduce the number of parameters even further, we only kept R amplitude values, which were computed from triangular filters computed on a Bark scale (similarly to [3]) for both $V_m[k]$ and $U_m[k]$. Magnitude spectra were then reconstructed by linear interpolation from these R values. For a sampling rate of 11kHz, the use of 10 values for $V_m[k]$ and 4 for $U_m[k]$ didn't result in any noticeable audible difference in analysis-resynthesis for a few analyzed utterances.

Replacing the smoothed formants by synthetic formants while keeping this reduced frequency-domain mixed excitation results in speech that sounds similar to the original. A preliminary evaluation of the synthetic excitation using the mixed excitation model also shows a great deal of promise.

7. CONCLUSIONS AND FUTURE WORK

We have presented a unifying framework for both formant tracking and speech synthesis using Hidden Markov Models (HMM). A set of recordings are used to train a HMM which can then generate formants. This data-driven formant synthesizer has the smoothness and compactness of formant synthesizers, and at the same time maintains the voice quality of the original speaker.

In the future we will investigate other methods for obtaining formant candidates such as peak picking in an LPC-spectrum or smoothed spectrum, or matching all possible formants to the current spectrum. We will also evaluate the use of mixture Gaussian models. Finally, a more extensive evaluation will be conducted in the future.

REFERENCES

[1] Acero A. "A Mixed-Excitation Frequency Domain Model for Time-Scale Pitch-Scale Modification of Speech". *International Conference on Spoken Language Processing*. Sydney, pp. 1923-1926. Dec, 1998.

[2] Allen J., Hunnicutt S., and Klatt D. *From text to speech: the MITalk system*. MIT Press, Cambridge, MA, 1987.

[3] Davis S. B. and Mermelstein P. "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences". *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 28, pp. 357-366, Aug. 1980.

[4] Huang X., Acero A., Adcock J., Hon H., Goldsmith J., Liu J., and Plumpe M. "Whistler: A Trainable Text-to-Speech System". *International Conference on Spoken Language Processing*. Philadelphia, pp. 2387-2390. Oct, 1996.

[5] Hunt A. J. and Black A. W. "Unit Selection in a Concatenative Speech Synthesis System Using a Large Speech Database" . *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Atlanta, pp. 373-376. May, 1996.

[6] Press W. H. et al. *Numerical Recipes in C, The art of Scientific Computing*. Cambridge University Press, 1988.

[7] Tokuda K., Masuko T., Yamada T., Kobayashi T. and Imai S. "An Algorithm for Speech Parameter Generation from Continuous Mixture HMMs with Dynamic Features". *Proceedings of the Eurospeech Conference*, Madrid, pp. 757-760. Sep, 1995.

[8] Yegnanarayana B. and Veldhuis R. N. J. "Extraction of Vocal-Tract System Characteristics from Speech Signals". *IEEE Trans. on Speech and Audio Processing*, vol. 6, pp. 313-327, July 1998.