The NICT Entry for the Blizzard Challenge 2009: an Enhanced HMM-based Speech Synthesis System with Trajectory Training considering Global Variance and State-Dependent Mixed Excitation

Ranniery Maia[†], *Tomoki Toda*[†,‡], *Shinsuke Sakai*[†], *Yoshinori Shiga*[†], *Jinfu Ni*[†], *Hisashi Kawai*[†]
Keiichi Tokuda[†,††], *Minoru Tsuzaki*[†,‡†], *Satoshi Nakamura*[†]

[†]National Institute of Information and Communications Technology, Japan
[‡]Nara Institute of Science and Technology, Japan
[††]Nagoya Institute of Technology, Japan
[‡†]Kyoto City University of Arts, Japan

{ranniery.maia,shinsuke.sakai,jinfu.ni,yoshinori.shiga,satoshi.nakamura}@nict.go.jp
tomoki@is.naist.jp, tokuda@nitech.ac.jp, tsuzaki@kcua.ac.jp

Abstract

This paper describes the NICT speech synthesis system submitted to the Blizzard Challenge 2009: a hidden Markov model (HMM)-based synthesizer constructed by training trajectory HMMs considering global variance. To improve naturalness of the synthesized speech a mixed excitation approach based on closed-loop residual modeling through the training of state-dependent filters is employed. According to the official results the system in question performs well in terms of naturalness and intelligibility although synthesized speech does not sound very similar to the original speaker.

Index Terms: speech synthesis, Blizzard Challenge, HMM-based speech synthesis, trajectory HMM, residual modeling.

1. Introduction

By following recent tendency towards the development of HMM-based speech synthesizers the National Institute of Information and Communications technology (NICT) has submitted a system based on the statistical parametric synthesis technology to the Blizzard Challenge 2009 (BC2009) [1]. The system basically corresponds to the one submitted last year jointly with ATR [2] with two significant enhancements. The first one concerns HMM modeling through the utilization of the technique of trajectory training with embedded global variance [3]. The second improvement consists in the use of a decision tree-based state clustering method for state definition [4] in the excitation modeling employed by the system [5]. According to the official listening test results, the submitted system performs well in terms of naturalness and intelligibility although the scores obtained for similarity to the original speaker were not good.

This paper is organized as follows. Section 2 briefly outlines the Blizzard Challenge 2009. Section 3 describes the submitted systems. Section 4 shows the voice building procedure while Section 5 discusses the results of the official listening test. Finally, Section 6 shows our conclusions.

2. The Blizzard Challenge 2009

The Blizzard Challenge is an event promoted in order to better understand and compare different techniques for building corpus-based speech synthesizers on the same data. The challenge consists of building some voices from a released data and synthesizing a prescribed set of test sentences, which are eventually evaluated through extensive listening tests by volunteers, speech experts, and paid native speakers.

For the BC2009 [1] two databases were released:

- UK English: 15 hours of a male speaker released by The Centre for Speech Technology Research (CSTR) at the University of Edinburgh, UK;
- Mandarin Chinese: 6000 sentences of a female speaker released by iFlyTek Co., Ltd, China.

For this year the required tasks were divided into *hub tasks* and *spoke tasks*. Hub tasks for English were:

- task EH1: build a voice from the full English database (about 15 hours);
- task EH2: build a voice from the specified ARCTIC subset of the full English database (approximately 1 hour).

The spoke tasks for English in which we took part were:

- task ES2: build a voice from the full English database suitable for synthesizing speech to be transmitted via the telephone channel;
- task ES3: build a voice from the full English database suitable for synthesizing the computer role in a human-computer dialog.

We did not take part in any Mandarin Chinese task.

Rule for this year corresponded to the non-utilization of the remainder of any database to build subset voices.

3. The submitted system

We submitted the same system for tasks EH1, ES2 and ES3, henceforth *System 1*. Further, the exact same technology was employed to construct the system submitted to task EH2, henceforth *System 2*. Thus, the only difference between *System 1* and *System 2* is that the former was trained with the full database whereas the latter employed only the ARCTIC subset.

Systems 1 and 2 are basically the same synthesizer described in [2] (submitted by ATR/NICT last year) with two main enhancements, namely: (1) trajectory training considering global variance [3]; (2) top-down clustering to define states of the utilized excitation model [4]. In the next sections each of these main improvements are treated with details.

3.1. GV-constrained trajectory training

The GV-constrained trajectory training method [3] provides a unified framework for training and synthesis using a common criterion considering global variance (GV) of [6]. We employ such method for refining the state output probability densities of conventional HMMs.

3.1.1. Observation vectors

Let us assume a D-dimensional static feature vector $\boldsymbol{c}_i = [c_i(1) \quad \cdots \quad c_i(D)]^\top$ at frame i. We use a speech parameter vector $\boldsymbol{o}_i = [\boldsymbol{c}_i^\top \quad \Delta\boldsymbol{c}_i^\top \quad \Delta\Delta\boldsymbol{c}_i^\top]^\top$ consisting of not only the static feature vector but also dynamic feature vectors $\Delta\boldsymbol{c}_i$, $\Delta\Delta\boldsymbol{c}_i$ as the observation vector. The sequences of vectors $\boldsymbol{o}_i$ and $\boldsymbol{c}_i$ over an utterance are written as $\boldsymbol{o} = [\boldsymbol{o}_1^\top \quad \cdots \quad \boldsymbol{o}_T^\top]^\top$ and $\boldsymbol{c} = [\boldsymbol{c}_1^\top \quad \cdots \quad \boldsymbol{c}_T^\top]^\top$, respectively. Moreover, we also use a GV vector $\boldsymbol{\nu}(\boldsymbol{c}) = [\nu_{\boldsymbol{c}}(1) \quad \cdots \quad \nu_{\boldsymbol{c}}(D)]^\top$ of the static feature vector sequence $\boldsymbol{c}$ as the other observation vector, calculated by

$$\nu_{\boldsymbol{c}}(d) = \frac{1}{T}\sum_{i=1}^{T}\left(c_i(d) - \langle c(d)\rangle\right)^2 \tag{1}$$

$$\langle c(d)\rangle = \frac{1}{T}\sum_{\tau=1}^{T} c_\tau(d) \tag{2}$$

3.1.2. Objective function

Given the HMM state sequence $\boldsymbol{q} = (q_1, \ldots, q_T)$, the HMM parameter set $\boldsymbol{\lambda}$ is optimized by maximizing the following objective function $\mathcal{L}_{\boldsymbol{q}}$ for the GV-constrained trajectory training,

$$\mathcal{L}_{\boldsymbol{q}} = P\left[\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\lambda}\right] P\left[\boldsymbol{\nu}(\boldsymbol{c})|\boldsymbol{q}\boldsymbol{\lambda}, \boldsymbol{\lambda}_\nu\right]^{\omega T} \tag{3}$$

where $P[\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\lambda}]$ is the probability density function in a trajectory HMM [7], $P[\boldsymbol{\nu}(\boldsymbol{c})|\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_\nu]$ is the probability density function of the GV, and λ_ν is the set of GV model parameters. The likelihood balance between these two probability density functions is controlled by the GV weight ω.

Definition of $P[c|\boldsymbol{q}, \boldsymbol{\lambda}]$: In the traditional HMMs, the probability density function of $\boldsymbol{o}$ given an HMM state sequence $\boldsymbol{q} = (q_1, \ldots, q_T)$ is written as

$$P[\boldsymbol{o}|\boldsymbol{q}, \boldsymbol{\lambda}] = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{\boldsymbol{q}}, \boldsymbol{U}_{\boldsymbol{q}}) = \prod_{i=1}^{T}\mathcal{N}(\boldsymbol{o}_i; \boldsymbol{\mu}_{q_i}, \boldsymbol{U}_{q_i}) \tag{4}$$

where $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{U})$ denotes a Gaussian distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix $\boldsymbol{U}$. The mean vector $\boldsymbol{\mu}_{\boldsymbol{q}}$ and the covariance matrix $\boldsymbol{U}_{\boldsymbol{q}}$ are given by

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^\top \quad \cdots \quad \boldsymbol{\mu}_{q_T}^\top\right]^\top \tag{5}$$

$$\boldsymbol{U}_{\boldsymbol{q}} = \text{diag}\left[\boldsymbol{U}_{q_1}, \cdots, \boldsymbol{U}_{q_T}\right] \tag{6}$$

The traditional HMM is reformulated as a trajectory HMM by imposing an explicit relationship between static and dynamic features, $\boldsymbol{o} = \boldsymbol{W}\boldsymbol{c}$, where $\boldsymbol{W}$ is a $3DT$-by-DT window matrix. The probability density function of $\boldsymbol{c}$ in the trajectory HMM given the state sequence $\boldsymbol{q}$ is then given by

$$P[\boldsymbol{c}|\boldsymbol{q}, \boldsymbol{\lambda}] = \frac{1}{Z_{\boldsymbol{q}}}P[\boldsymbol{o}|\boldsymbol{q}, \boldsymbol{\lambda}] = \mathcal{N}(\boldsymbol{c}; \overline{\boldsymbol{c}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}) \tag{7}$$

where

$$\overline{\boldsymbol{c}}_{\boldsymbol{q}} = \boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}} \tag{8}$$

$$\boldsymbol{P}_{\boldsymbol{q}}^{-1} = \boldsymbol{W}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{W} \tag{9}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} \tag{10}$$

$$Z_{\boldsymbol{q}} = \frac{\sqrt{(2\pi)^{DT}|\boldsymbol{P}_{\boldsymbol{q}}|}}{\sqrt{(2\pi)^{3DT}|\boldsymbol{U}_{\boldsymbol{q}}|}}e^{-\frac{1}{2}\left(\boldsymbol{\mu}_{\boldsymbol{q}}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{r}_{\boldsymbol{q}}^\top\boldsymbol{P}_{\boldsymbol{q}}\boldsymbol{r}_{\boldsymbol{q}}\right)} \tag{11}$$

Note that the mean vector $\overline{\boldsymbol{c}}_{\boldsymbol{q}}$ is equivalent to the ML estimate of the static feature vector sequence generated from the HMM by the conventional parameter generation algorithm [8].

Definition of $P[\boldsymbol{\nu}(\boldsymbol{c})|\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_\nu]$: The probability density of the GV is modeled by

$$P[\boldsymbol{\nu}(\boldsymbol{c})|\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_\nu] = \mathcal{N}(\boldsymbol{\nu}(\boldsymbol{c}); \boldsymbol{\nu}(\overline{\boldsymbol{c}}_{\boldsymbol{q}}), \boldsymbol{\nu}_v) \tag{12}$$

Note that the mean vector $\boldsymbol{\nu}(\overline{\boldsymbol{c}}_{\boldsymbol{q}})$ is defined as the GV of the mean vector of the trajectory HMM. Hence, the GV likelihood $P[\boldsymbol{\nu}(\boldsymbol{c})|\boldsymbol{q}, \boldsymbol{\lambda}, \boldsymbol{\lambda}_\nu]$ works as a penalty term to make the GV of the generated parameters close to that of the natural ones.

3.1.3. Model parameter estimation

Given the HMM state sequence $\boldsymbol{q}$, the GV weight ω, and the GV covariance matrix $\boldsymbol{\Sigma}_v$, the parameter set $\boldsymbol{\lambda}$ including the mean vectors and diagonal covariance matrices at all HMM states (from 1 to S),

$$\boldsymbol{m} = \left[\boldsymbol{\mu}_1^\top \quad \cdots \quad \boldsymbol{\mu}_S^\top\right]^\top \tag{13}$$

$$\boldsymbol{\Sigma}^{-1} = \left[\boldsymbol{U}_1^{-1} \quad \cdots \quad \boldsymbol{U}_S^{-1}\right]^\top \tag{14}$$

are simultaneously updated by maximizing $\mathcal{L}_{\boldsymbol{q}}$. This is done iteratively by using the following gradients,

$$\frac{\partial\mathcal{L}_{\boldsymbol{q}}}{\partial\boldsymbol{m}} = \boldsymbol{\Phi}_{\boldsymbol{q}}^\top\boldsymbol{U}_{\boldsymbol{q}}^{-1}\boldsymbol{W}\left(\boldsymbol{c} - \overline{\boldsymbol{c}}_{\boldsymbol{q}} + \omega\boldsymbol{P}_{\boldsymbol{q}}\overline{\boldsymbol{x}}_{\boldsymbol{q}}\right) \tag{15}$$

$$\begin{aligned}\frac{\partial\mathcal{L}_{\boldsymbol{q}}}{\partial\boldsymbol{\Sigma}^{-1}} = \frac{1}{2}\boldsymbol{\Phi}_{\boldsymbol{q}}^\top\text{on-diag}\Big[&\boldsymbol{W}(\boldsymbol{P}_{\boldsymbol{q}} + \overline{\boldsymbol{c}}_{\boldsymbol{q}}\overline{\boldsymbol{c}}_{\boldsymbol{q}}^\top - \boldsymbol{c}\boldsymbol{c}^\top)\boldsymbol{W}^\top \\ -2\boldsymbol{\mu}_{\boldsymbol{q}}(\overline{\boldsymbol{c}}_{\boldsymbol{q}} - \boldsymbol{c})^\top\boldsymbol{W}^\top &+2\omega\boldsymbol{W}\boldsymbol{P}_{\boldsymbol{q}}\overline{\boldsymbol{x}}_{\boldsymbol{q}}(\boldsymbol{\mu}_{\boldsymbol{q}} - \boldsymbol{W}\overline{\boldsymbol{c}}_{\boldsymbol{q}})^\top\Big]\end{aligned} \tag{16}$$

where each element of a DT-dimensional vector $\overline{\boldsymbol{x}}_{\boldsymbol{q}}$ is

$$\overline{\boldsymbol{x}}_{\boldsymbol{q}} = \left[\overline{\boldsymbol{x}}_{\boldsymbol{q},1}^\top \quad \cdots \quad \overline{\boldsymbol{x}}_{\boldsymbol{q},T}^\top\right]^\top \tag{17}$$

$$\overline{\boldsymbol{x}}_{\boldsymbol{q},i} = \left[\overline{x}_{\boldsymbol{q},i}(1) \quad \cdots \quad \overline{x}_{\boldsymbol{q},i}(D)\right]^\top \tag{18}$$

$$\begin{aligned}\overline{x}_{\boldsymbol{q},i}(d) = &-2\left(\overline{c}_{\boldsymbol{q},i}(d) - \langle\overline{c}_{\boldsymbol{q}}(d)\rangle\right)\cdot \\ &\left[\boldsymbol{\nu}(\overline{\boldsymbol{c}}_{\boldsymbol{q}}) - \boldsymbol{\nu}(\boldsymbol{c})\right]^\top\boldsymbol{p}_\nu(d)\end{aligned} \tag{19}$$

The d-th column of $\boldsymbol{\Sigma}_\nu^{-1}$ is $\boldsymbol{p}_\nu(d)$. The matrix $\boldsymbol{\Phi}_{\boldsymbol{q}}$ is a $3DT \times 3DN$ matrix whose elements are 0 or 1 determined according to the state sequence $\boldsymbol{q}$. The notation on-diag[$\cdot$] denotes the extraction of only diagonal elements from a square matrix.

Although a single observation sequence is assumed (i.e., a single utterance) to simplify explanations, multiple observation sequences are actually used in the training process.

3.1.4. Parameter generation

The objective function $\mathcal{L}_{\boldsymbol{q}}$ is also used for parameter generation. Given the HMM state sequence $\boldsymbol{q}$, the ML estimate of the static feature vector sequence determined by maximizing $\mathcal{L}_{\boldsymbol{q}}$ is given by $\overline{\boldsymbol{c}}_{\boldsymbol{q}}$. Therefore, the generated trajectory is analytically calculated by (8) even if we consider the GV in the parameter generation process. This is because the GV-constrained trajectory training optimizes the HMM parameters so that the GV of the generated trajectory is close to the natural one.

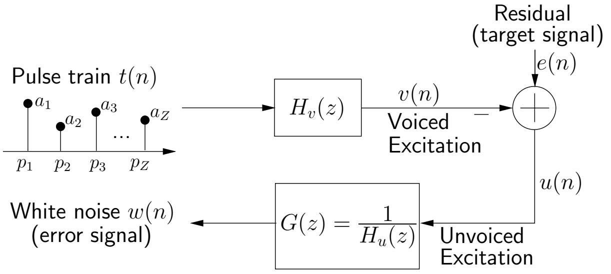

Figure 1: *Training part of the excitation model utilized by the NICT system.*

3.2. Tree-based state definition for excitation modeling

Figure 1 depicts the training part of the excitation model described in [5], which is applied to our system. Filters

$$H_v(z) = \sum_{l=-M/2}^{M/2} h(l)z^{-l} \tag{20}$$

$$H_u(z) = \frac{K}{1 - \sum_{l=1}^{L} g(l)z^{-l}} \tag{21}$$

vary according to each HMM state and their coefficients are optimized using a residual signal ML criterion [5]. The excitation training process can be enumerated through the following steps: (1) state definition; (2) residual segment classification according to the defined states; (3) iterative filter calculation for each cluster of residual segments using the procedure described in [5]. Our improvement on excitation modeling concerns an analytic method to define states in Step 1.

3.2.1. Clustering criterion: residual ML

Assuming that the noise sequence $w(n)$ which is output by filter $G(z)$ in Figure 1 is a Gaussian process, the log likelihood of the signal $u(n)$, also a Gaussian process, is given by

$$\log P[\mathbf{u}|\mathbf{H}_u] = -\frac{N}{2}\log 2\pi + \frac{1}{2}\log|\mathbf{G}^\top\mathbf{G}| - \frac{1}{2}\mathbf{u}^\top\mathbf{G}^\top\mathbf{G}\mathbf{u} \tag{22}$$

where N is the number of samples of the entire database and

$$\mathbf{u} = \begin{bmatrix} u(0) & \cdots & u(N-1) \end{bmatrix}^\top \tag{23}$$

$$\mathbf{G} = \begin{bmatrix} \tilde{\mathbf{g}}^{(0)} & \cdots & \tilde{\mathbf{g}}^{(N-1)} \end{bmatrix} \tag{24}$$

$$\tilde{\mathbf{g}}^{(m)} = \Bigg[\underbrace{0 \cdots 0}_{m \text{ terms}} \; \tfrac{1}{K} \; \tfrac{g(1)}{K} \; \cdots \; \tfrac{g(L-1)}{K} \; \underbrace{0 \cdots 0}_{N-m-1 \text{ terms}}\Bigg]^\top \tag{25}$$

The second term in the right side of (22) can be written as

$$\frac{1}{2}\log|\mathbf{G}^\top\mathbf{G}| = \frac{1}{2}\sum_{n=0}^{N-1}\log\left|1 - \sum_{l=1}^{L} g(l)e^{jw_n l}\right|^2 - N\log K \tag{26}$$

and because $G(z)$ is minimum-phase, the first term in the right side of (26) is zero [9]. Further, if $w(n)$ is a white noise sequence with variance one and mean zero, the third term in the right side of (22) can be approximated as follows

$$\mathbf{u}^\top\mathbf{G}^\top\mathbf{G}\mathbf{u} = K^2NE\{w^2(n)\} \approx K^2N \tag{27}$$

Therefore, the likelihood of $e(n)$ given the excitation model is simply a function of the unvoiced filter gain component K,

$$\log P[\mathbf{e}|\mathbf{H}_v, \mathbf{H}_u, \mathbf{t}] = -\frac{N}{2}\log 2\pi - N\left(\log K + \frac{K^2}{2}\right) \tag{28}$$

3.2.2. Clustering procedure

By taking into account the state-dependency of the filter coefficients, (28) can be re-written as

$$\log P[\mathbf{e}|\mathbf{H}_v, \mathbf{H}_u, \mathbf{t}] = -\frac{N}{2}\log 2\pi + \sum_{j=1}^{\mathcal{S}} \mathfrak{L}_j \tag{29}$$

where

$$\mathfrak{L}_j = -N_j\left(\log K_j + \frac{K_j^2}{2}\right) \tag{30}$$

is the likelihood of $e(n)$ under state s_j, N_j is its corresponding number of samples, K_j is the corresponding unvoiced filter gain, and $\mathcal{S}$ is the number of states (or clusters for tied states).

From Figure 1, initially voiced filter coefficients are computed, followed by the determination of $u(n)$, finally leading to gain component K_{s_j}. The process of splitting one cluster into two thus can be sketched as follows:

1. split s_j into s_{j_1} and s_{j_2} given a candidate question;
2. calculate voiced filter coefficients, $\mathbf{h}_{s_{j_1}}$ and $\mathbf{h}_{s_{j_2}}$, for the new clusters s_{j_1} and s_{j_1}, respectively;
3. compute unvoiced filter coefficients with corresponding gain components, $\mathbf{g}_{j_1}$, K_{j_1}, $\mathbf{g}_{j_2}$, and K_{j_2}, respectively for s_{j_1} and s_{j_2}.

After calculating $\mathfrak{L}_{j_1}$ and $\mathfrak{L}_{j_2}$ from K_{j_1} and K_{j_2}, respectively, according to (30), likelihood increment due to the split can be measured by

$$\mathfrak{L}_{\text{inc}} = \mathfrak{L}_{\text{after}} - \mathfrak{L}_{\text{before}} = \mathfrak{L}_{j_1} + \mathfrak{L}_{j_2} - \mathfrak{L}_{s_j} \tag{31}$$

3.2.3. Approximations to decrease computational complexity

The determination of voiced filters and unvoiced filter gain components for s_{j_x} implies optimization of filter coefficients and pulse trains for the new clusters, according to the algorithm described in [5]. In order to decrease computational complexity this iterative optimization is replaced by single calculation of voiced filters followed by linear prediction analysis of the unvoiced excitation signal $u(n)$ under segments belonging to s_{j_x} to derive the gain component K_{j_x}.

Assuming the diagram of Figure 1, voiced filter coefficients for cluster s_{j_x} can be obtained by least squares,

$$\mathbf{h}_{j_x} = \left(\sum_{i \in s_{j_x}} \mathbf{A}_i^\top \mathbf{A}_i\right)^{-1} \sum_{i \in s_{j_x}} \mathbf{A}_i^\top \mathbf{e}_i \tag{32}$$

where

$$\mathbf{A}_i = \begin{bmatrix} \tilde{\mathbf{t}}_i^{(0)} & \cdots & \tilde{\mathbf{t}}_i^{(M)} \end{bmatrix} \tag{33}$$

$$\tilde{\mathbf{t}}_i^{(m)} = \Bigg[\underbrace{0 \cdots 0}_{m \text{ terms}} \; t_i(0) \cdots t_i(N_i - 1) \; \underbrace{0 \cdots 0}_{M-m \text{ terms}}\Bigg]^\top \tag{34}$$

$$\mathbf{e}_i = \Bigg[\underbrace{0 \cdots 0}_{\frac{M}{2} \text{ terms}} \; e_i(0) \cdots e_i(N_i - 1) \; \underbrace{0 \cdots 0}_{\frac{M}{2} \text{ terms}}\Bigg]^\top \tag{35}$$

with $t_i(n)$ and $e_i(n)$ being respectively pulse train and residual segments with N_i samples belonging to cluster s_{j_x}. Segments are obtained according to alignment performed at the HMM state level. After that, the gain K_{j_x} is calculated from

$$K_{j_x} = \sqrt{r_{j_x}(0) - \sum_{l=1}^{L} g_{j_x}(l) r_{j_x}(l)} \tag{36}$$

with

$$r_{j_x}(l) = \sum_{i \in s_{j_x}} \sum_{n=0}^{N_i - 1} u_i(n) u_i(n-l), \quad l = 0, \ldots, L \qquad (37)$$

being the sum of autocorrelation sequences of all segments of $u_i(n) = e_i(n) - h_{j_x}(n) * t_i(n)$, where $i \in s_{j_x}$. The unvoiced filter coefficients of cluster s_{j_x}, $\{g_{j_x}(1), \ldots, g_{j_x}(L)\}$, are determined from $r_{j_x}(l)$ using Levinson-Durbin [9].

3.2.4. Stop criterion

The minimum description length (MDL) [10] is utilized as stop criterion. Let the description length of excitation model $\mathbf{\Gamma}_i$ with cluster set $\{s_1, \ldots, s_{\mathcal{S}_i}\}$, where $\mathcal{S}_i$ is the number of clusters, where each of them has a voiced and unvoiced filter, be

$$\ell_i = \frac{N}{2} \log 2\pi - \sum_{j=1}^{\mathcal{S}_i} \mathfrak{L}_{s_j} + \frac{(M+L+2)\mathcal{S}_i}{2} \log N \qquad (38)$$

The first and second terms in the right side of (38) correspond to the likelihood of $\mathbf{\Gamma}_i$ whereas the third term measures its complexity [10]. The difference of description length between the model after the split $\mathbf{\Gamma}_{i+1}$, and the model before the split $\mathbf{\Gamma}_i$ is

$$\Delta\ell = \ell_{i+1} - \ell_i = -\mathfrak{L}_{\text{inc}} + \frac{M+L+2}{2} \log N \qquad (39)$$

The clustering process is stopped if $\Delta\ell > 0$.

4. Building voices for the BC2009

4.1. Database segmentation and labeling

We utilized the same labels constructed for the system submitted last year. Thus, procedure for segmentation and full context label construction can be seen in [2].

4.2. Speech parameter extraction

Speech parameters modeled by HMM consisted of $\log F_0$ and spectral parameter vectors, extracted from the database at every 5 ms. We extracted F_0 using the Snack Sound Toolkit [11].

Spectral parameters were also extracted at every 5 ms. Periodograms, represented by power density spectrum extracted using STRAIGHT [12], were derived from each 5-ms frames, from which eventually two kinds of coefficients were extracted: (1) mel-cepstral coefficients as described in [13]; (2) mel-generalized cepstral coefficients as described in [14]. We trained two different systems by utilizing the full database and ARCTIC subset using coefficients (1) and (2), henceforth *System-MCEP* and *System-MGC*, respectively, in order to choose the best one to submit to the BC2009. The number of coefficients extracted from each frame, for both systems, was 40 (including the 0-th coefficient). For System-MGC, before HMM modeling mel-generalized coefficients were changed into the line spectral pair domain (MGC-LSP), as described in [14]. After an informal listening test performed with 8 speech synthesis expert listeners and 1 speech expert we decided to submit *System-MCEP*, i.e., we chose mel-cepstral coefficients derived from STRAIGHT spectrum for spectral parameterization.

4.3. Synthesizer training

Initially, hidden semi-Markov models (HSMMs) were trained using multi-stream observation vectors composed of mel-cepstral coefficients and F_0, with their corresponding deltas and delta-deltas, using the same conditions as the ones described in [2]. The trained HSMMs were then approximated by HMMs by copying their state output probability densities followed by transition probability training. After that, sub-optimum HMM state sequences for each training utterance were determined by Viterbi alignment. Based on the determined state sequences, HMM parameters were optimized by GV-constrained trajectory training conducted as follows. First, HMM parameters were updated by maximizing solely the trajectory likelihood, i.e., by setting the GV weight ω in (3) to zero. This optimization process is equivalent to the standard trajectory training [7]. After that, HMM parameters were further updated by setting ω to a proper value ($\omega = 0.5$ for our system). The diagonal covariance matrix $\mathbf{\Sigma}_\nu$ of the GV probability density function was trained from the GV vectors of all training utterances. Note that HMM state sequences were not updated.

Table 1: *Log-scaled trajectory likelihood (Traj) given by (7), GV likelihood (GV) given by (12), and GV-constrained trajectory likelihood (Traj + GV) given by (3) (when* $\omega = 1.0$*) of each model. These likelihoods are normalized by dividing the total likelihoods by the number of dimensions of the static feature vector and the number of frames.*

a) System 1 (submitted to tasks EH1, ES2 and ES3)

Likelihoods for mel-cepstrum	Traj	GV	Traj + GV
Standard HMM	0.81	-7.44	-6.63
Trajectory HMM	1.24	-7.34	-6.10
GV-Trajectory HMM	1.19	4.32	5.51

Likelihoods for log F_0	Traj	GV	Traj + GV
Standard HMM	0.78	0.75	1.53
Trajectory HMM	0.94	0.79	1.73
GV-Trajectory HMM	0.94	0.87	1.81

b) System 2 (submitted to task EH2)

Likelihoods for mel-cepstrum	Traj	GV	Traj + GV
Standard HMM	0.81	-13.11	-12.30
Trajectory HMM	1.25	-12.93	-11.68
GV-Trajectory HMM	1.19	4.56	5.75

Likelihoods for log F_0	Traj	GV	Traj + GV
Standard HMM	0.84	0.91	1.75
Trajectory HMM	1.00	1.09	2.09
GV-trajectory HMM	0.99	1.26	2.25

To verify the effect of GV-constrained trajectory training, we checked the log-scaled trajectory, GV, and GV-constrained trajectory likelihoods for mel-cepstrum and log-scaled F_0 in the training data, respectively, as described in [3]. Table 1 shows reasonable results for both mel-cepstrum and F_0 components, from which it can be inferred that: (1) trajectory training yields significant improvements in the trajectory likelihoods; (2) GV-constrained trajectory training yields dramatic improvements in the GV likelihoods while not causing significant reductions to the trajectory likelihoods.

4.4. Excitation training

Residual signals were derived from the speech database by inverse filtering using the mel-cepstral coefficients of [13], extracted from speech at every 5 ms. Full context models were used to segment the residual signals at the HMM state level. Pulse trains were derived from pitch marks and eventually optimized for the residual segments through the procedure described in [5]. Residual segments were then clustered using the procedure described in Section 3.2. The MDL criterion as

Table 2: *Number of terminal nodes at the end of the clustering process for the system submitted to tasks EH1, ES2 and ES3 (System 1), and system submitted to task EH2 (System 2). Number of logical models is 362515 and 38490, respectively.*

HMM state	S_1	S_2	S_3	S_4	S_5	Total
System 1	51	51	48	51	10	211
System 2	19	31	32	30	23	135

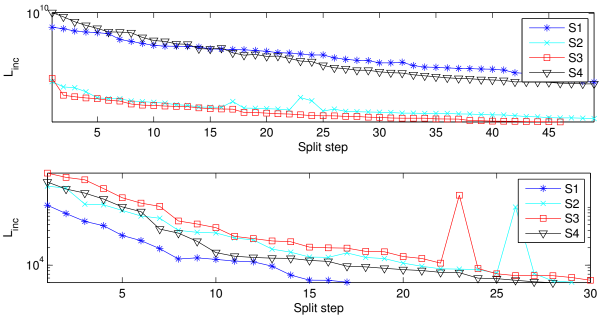

Figure 2: $\mathfrak{L}_{inc}$ *for each split iteration for HMM states* S_1 *to* S_4. *Top: system submitted to tasks EH1, ES2 and ES3 (System 1). Bottom: system submitted to task EH2 (System 2).*

shown in Section 3.2.4 was used to stop tree growth. In order to assure that training of the full database voice would be finished on time, only pentaphone and silence related questions were utilized for the clustering process. Once states were defined as terminal nodes of the obtained decision trees, voiced and unvoiced filters for excitation models of System 1 and System 2 were calculated using the procedure described in [5].

Figure 2 shows the evolution of $\mathfrak{L}_{\text{inc}}$ given by (31) for HMM state positions $\{S_1, \ldots, S_4\}$ at each split step, for systems 1 and 2. It can be seen that $\mathfrak{L}_{\text{inc}}$ decreases as trees grows, although some outliers occur, probably due to the clustering approximations of Section 3.2.3. Table 2 shows the number of terminal nodes at the end of the process.

Figure 3 shows 3-D plots of impulse responses of $H_v(z)$ of the achieved excitation models, and excitation models trained by the conventional phonetic decision trees approach, used in [2]. It can be noticed that filters of *System 1* obtained through the proposed algorithm achieve convergence more than the conventional method. This shows that the clustering was successful in grouping similar residual segments under the same cluster. However, the same result was not achieved for *System 2*.

5. System performance

The BC2009 evaluated the submitted systems under three main categories: (1) naturalness; (2) similarity to the original speaker; and (3) word error rate of semantically incorrect sentences [15]. Aside from those, a new criterion was created to evaluate task ES3, which corresponded to an *appropriateness degree* of the synthesized speech as answers to some specified questions in a conversational domain.

5.1. Listening test results

All the box plots shown in this section represent the opinions of English native speakers paid to conduct the listening tests. Figure 4 shows naturalness scores for the systems submitted to tasks EH1 and EH2, whereas Figure 5 shows the scores of appropriateness of the synthesized speech for the systems submitted to task ES3. It can be seen that the results were reasonably good for *System 1*, submitted to EH1 and ES3, whereas *System 2* did not achieve the expected naturalness score in task EH2. Figure 6 shows the degrees of similarity to the original speaker for tasks EH1 and EH2, where it can be noticed that our systems performed below average. Figure 7 shows WER for system submitted to tasks EH1 and EH2. The NICT system achieves the best performance for task EH1 whereas the result for task EH2 was surprisingly below what we expected.

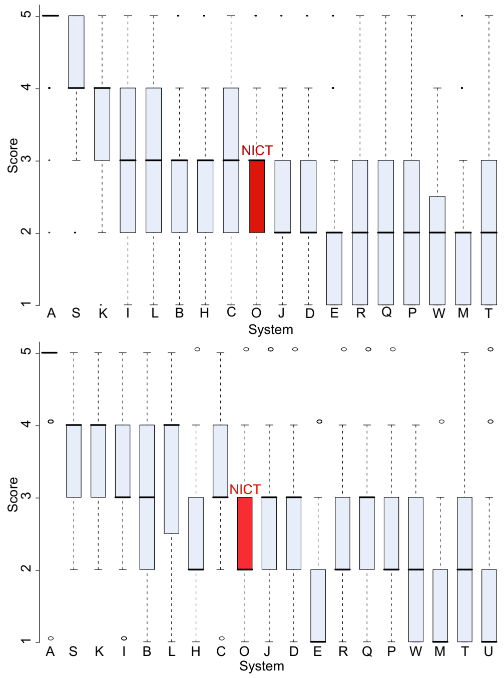

Figure 4: *Naturalness scores according to paid native speakers for systems submitted to tasks EH1 (top) and EH2 (bottom).*

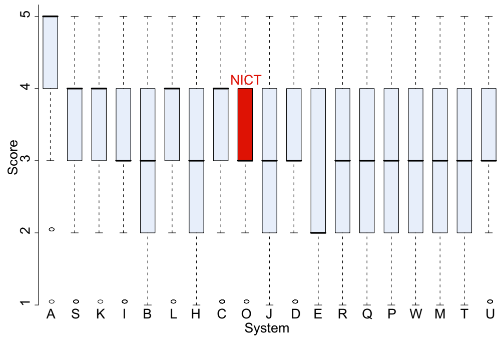

Figure 5: *Appropriateness of the speech synthesized by systems submitted to task ES3 as answers to some specified questions, according to paid native speakers.*

5.2. Discussion on the performance

In average, according to the official results, *System 1* (full database) performed better than *System 2 (ARCTIC subset)*. This fact did not happen for the system we submitted last year [2], which achieved similar performance despite the difference in data size. Perhaps our top-down clustering approach for filter state definition worked better for *System 1* due to its greater amount of training data. This can be inferred from Figure 3 where the improvement for *System 1* is more evident than the improvement obtained for *System 2*, when compared with the baseline method for state definition. Lastly, we faced some problems to synthesize some test sentences using the released Festival utterance files. Maybe it would have been more appropriate to generate full context labels directly from text.

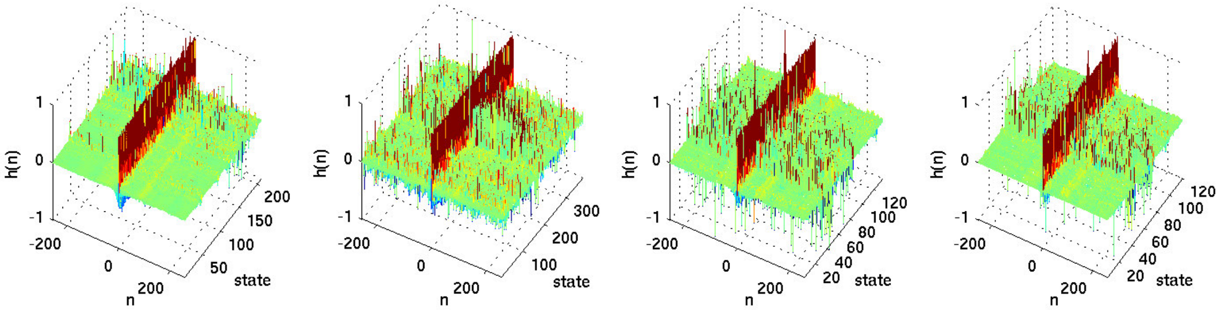

Figure 3: *Impulse responses of voiced filters $H_v(z)$ derived using state configurations yielded by the proposed residual clustering algorithm and by the baseline phonetic trees method. Respectively from left to right, filters calculated using: full database and proposed algorithm, full database and phonetic trees, ARCTIC subset and proposed algorithm, and ARCTIC subset and phonetic trees.*

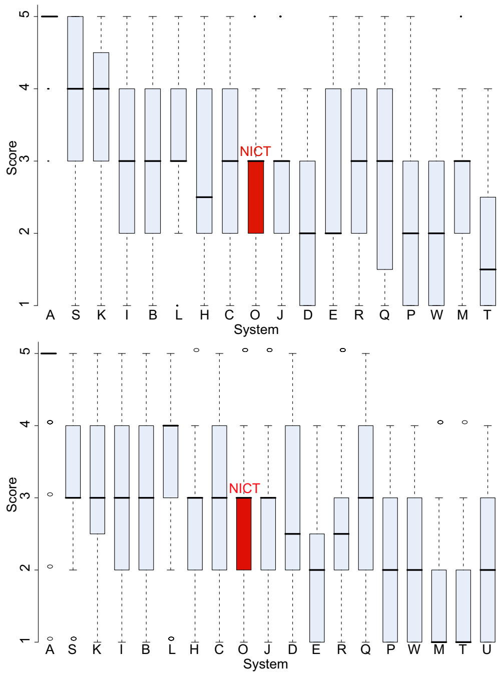

Figure 6: *Similarity scores according to paid native speakers for systems submitted to tasks EH1 (top) and EH2 (bottom).*

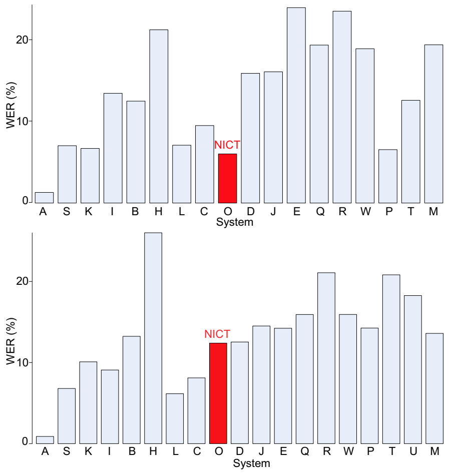

Figure 7: *WER according to paid native speakers for systems submitted to tasks EH1 (top) and EH2 (bottom).*

6. Conclusion

This paper described the NICT entry for the Blizzard Challenge 2009. According to the results, the system achieved fair results in terms of naturalness, similarity and intelligibility. The difference in performance for tasks EH1 and EH2 was considerable.

7. References

[1] http://www.synsig.org/index.php/Blizzard_Challenge_2009.

[2] R. Maia, J. Ni, S. Sakai, T. Toda, K. Tokuda, T. Shimizu, and S. Namakura, "The NICT/ATR speech synthesis system for the Blizzard Challenge 2008," in *Blizzard Challenge Workshop*, 2008.

[3] T. Toda and S. Young, "Trajectory training considering global variance for HMM-based speech synthesis," in *ICASSP*, 2009.

[4] R. Maia, T. Toda, K. Tokuda, S. Sakai, and S. Nakamura, "A decision tree-based clustering approach to state definition in an excitation modeling framework for HMM-based speech synthesis," in *INTERSPEECH*, 2009.

[5] R. Maia, T. Toda, H. Zen, Y. Nankaku, and K. Tokuda, "An excitation model for HMM-based speech synthesis based on residual modeling," in *SSW6*, 2007.

[6] T. Toda and k. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Transactions*, vol. E90-D, pp. 816–824, Mar. 2007.

[7] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Computer Speech and Language*, vol. 21, pp. 153–173, 2007.

[8] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *ICASSP*, 2000.

[9] J. D. Markel and A. H. Gray, Jr., *Linear prediction of speech*. Springer-Verlag, 1986.

[10] J. Rissanen, "Universal coding, information, prediction, and estimation," *IEEE Transactions on Information Theory*, vol. IT-30, July 1984.

[11] http://www.speech.kth.se/snack.

[12] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, Apr. 1999.

[13] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *ICASSP*, 1992.

[14] K. Koishida, G. Hirabayashi, K. Tokuda, and T. Kobayashi, "a 16kbit/s wideband CELP-based speech coder using mel-generalized cepstral analysis," *IEICE Transactions on Information & Systems*, vol. E83-D, pp. 876–883, Apr. 2000.

[15] V. Karaiskos, S. King, R. Clark, and C. Mayo, "The Blizzard Challenge 2008," in *Blizzard Challenge Workshop*, 2008.