PARAMETRIC SPEECH SYNTHESIS BASED ON GAUSSIAN PROCESS REGRESSION USING GLOBAL VARIANCE AND HYPERPARAMETER OPTIMIZATION

Tomoki Koriyama[1], *Takashi Nose*[2], *Takao Kobayashi*[1]

[1]Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology
[2]Graduate School of Engineering, Tohoku University

koriyama@ip.titech.ac.jp, tnose@m.tohoku.ac.jp, takao.kobayashi@ip.titech.ac.jp

ABSTRACT

This paper examines two issues of a statistical speech synthesis approach based Gaussian process (GP) regression. Although GP-based speech synthesis can give higher performance in generating spectral parameters than the HMM-based one, a number of issues still remain. In this paper, we incorporate global variance (GV) feature to overcome over-smoothing problem into the parameter generation. Furthermore, in order to utilize an appropriate kernel function in accordance with actual data, we propose an EM-based kernel hyperparameter optimization technique. Objective and subjective evaluation results show that using GV and hyperparameter estimation enhanced the performance in spectral feature generation.

Index Terms— statistical parametric speech synthesis, Gaussian process, global variance, kernel hyperparameter

1. INTRODUCTION

In the last decade, statistical parametric speech synthesis framework based on hidden Markov model (HMM) [1] has shown various advantages such as flexibility to transforming voice characteristics, robustness to language dependence, and realization of small foot-print systems [2]. However, there still remain problems in vocoding, which result from insufficient accuracy of acoustic modeling and over-smoothing effect. Recently, for the purpose of more accurate acoustic modeling, other approaches to the statistical parametric acoustic modeling have been proposed using deep neural networks [3] and deep belief network-Gaussian process hybrid model [4].

In this context, we have proposed an alternative speech synthesis approach based on frame-level Gaussian process (GP) regression [5, 6]. In this approach, frame-level contextual features are used as input features of GP regression [7], and acoustic features are used as output features. To make computational cost feasible, we adopted partially independent conditional (PIC) approximation [8] and showed that the GP-based approach achieved comparable or better performance compared with HMM-based approach in generation of spectral feature trajectories [6]. This could be attributed to the advantages of GPs, such as the flexibility to model complexity and the robustness against over-fitting.

Although the GP-based speech synthesis is promising, there exist a number of issues for the realization of practical systems. One of them is generation of acoustic feature trajectories from predictive distribution. We used the predictive mean sequence as the generated trajectories in the previous study [6]. However, it generally causes over-smoothing problem. In this study, therefore, we incorporate an alternative way considering global variance (GV) [9] which has been widely used to alleviate over-smoothing problem in the HMM-based speech synthesis. Another issue is selection of hyperparameter of kernel function used in GP. Fortunately, it is possible to use automatic hyperparameter optimization based on empirical Bayesian approach for GP. In this paper, we apply it to the speech synthesis framework using PIC approximation.

A part of this work was supported by JSPS Grant-in-Aid for Scien- tific Research 24300071, 25540065, and 25·8776.

2. GP-BASED SPEECH SYNTHESIS

In the speech synthesis framework based on GP regression [5,6], we consider GP in which an input variable $\mathbf{x}_n$ is a contextual feature set obtained by linguistic and phone boundary information, and an output variable y_n is an acoustic feature normalized into zero mean. The correlation between two frames, whose input variables are $\mathbf{x}$ and $\mathbf{x}'$, are defined by a kernel function $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a hyperparameter vector of the kernel function. This enables frame-level modeling of acoustic features without using dynamic features that are used and essential in the HMM-based speech synthesis. Based on GP regression framework [7], predictive distribution of test (synthetic) data $\mathbf{y}_T$ given training data $\mathbf{y}$ are inferred, which is given by a Gaussian distribution

$$p(\mathbf{y}_T|\mathbf{y}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_T; \boldsymbol{\mu}_{\mathbf{y}_T|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}_T|\mathbf{y}}) \tag{1}$$

and used for parameter trajectory generation. For example, we used mean sequence $\boldsymbol{\mu}_{\mathbf{y}_T|\mathbf{y}}$ in the previous study [5].

One issue of GP-based speech synthesis is computational cost. To reduce the computational cost, we have incorporated partially independent conditional (PIC) approximation [8] into GP-based acoustic modeling [6]. PIC approximation uses a latent variable $\mathbf{f}_M$ which is dependent on pseudo-data input features $\{\mathbf{x}_m | m = 1, \ldots, M\}$, where the pseudo-data size M is much smaller than the number of frames in training data, N. The joint distribution of $\mathbf{f}_M$ is given by following Gaussian distribution.

$$p(\mathbf{f}_M|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_M; \mathbf{0}, \mathbf{K}_M) \tag{2}$$

where $\mathbf{K}_M$ is a covariance (Gram) matrix whose elements are kernel values of pseudo-data input features, which is calculated by a kernel function $k(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta})$. In PIC approximation, all training data $\mathbf{y}$ is separated into several blocks $\{\mathbf{y}_{B_s}\}(s = 1, \ldots, S)$, and it is assumed that $\mathbf{y}_{B_1}, \ldots, \mathbf{y}_{B_S}$ are conditionally independent given $\mathbf{f}_M$. By using this assumption, the joint probability of all training data

$\mathbf{y} = [y_1 \dots y_N]^\top$ is defined by

$$p(\mathbf{y}|\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{f}_M, \boldsymbol{\theta}) p(\mathbf{f}_M|\boldsymbol{\theta}) d\mathbf{f}_M = \int \prod_{s=1}^{S} p(\mathbf{y}_{B_s}|\mathbf{f}_M, \boldsymbol{\theta}) p(\mathbf{f}_M|\boldsymbol{\theta}) d\mathbf{f}_M \quad (3)$$

where

$$p(\mathbf{y}_{B_s}|\mathbf{f}_M, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_{B_s}; \boldsymbol{\mu}_{\mathbf{y}_{B_s}|\mathbf{f}_M}, \boldsymbol{\Sigma}_{\mathbf{y}_{B_s}|\mathbf{f}_M}) \quad (4)$$

$$\boldsymbol{\mu}_{\mathbf{y}_{B_s}|\mathbf{f}_M} = \mathbf{K}_{B_s M} \mathbf{K}_M^{-1} \mathbf{f}_M \quad (5)$$

$$\boldsymbol{\Sigma}_{\mathbf{y}_{B_s}|\mathbf{f}_M} = \mathbf{K}_{B_s} - \mathbf{Q}_{B_s} + \sigma_\nu^2 \mathbf{I} \quad (6)$$

$$\mathbf{Q}_{B_s} = \mathbf{K}_{B_s M} \mathbf{K}_M^{-1} \mathbf{K}_{M B_s}. \quad (7)$$

Here, $\mathbf{K}_{B_s M} = \mathbf{K}_{M B_s}^\top$ represents the covariance matrix that expresses the relationship between the block s and the pseudo-data, and $\mathbf{K}_{B_s}$ and $\mathbf{K}_M$ correspond to self-covariance matrices of the block s and the pseudo-data, respectively. σ_ν^2 is a noise variance. By marginalizing out the latent variable $\mathbf{f}_M$ in (3), we get the following Gaussian distribution.

$$p(\mathbf{y}|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K}_N^{\mathrm{PIC}} + \sigma_\nu^2 \mathbf{I}) \quad (8)$$

$$\mathbf{K}_N^{\mathrm{PIC}} = \mathbf{Q}_N + \mathrm{blkdiag}(\mathbf{K}_N - \mathbf{Q}_N) = \begin{bmatrix} \mathbf{K}_{B_1} & \mathbf{Q}_{B_1 B_2} & \cdots & \mathbf{Q}_{B_1 B_S} \\ \mathbf{Q}_{B_2 B_1} & \mathbf{K}_{B_2} & & \mathbf{Q}_{B_2 B_S} \\ \vdots & & \ddots & \vdots \\ \mathbf{Q}_{B_S B_1} & \mathbf{Q}_{B_S B_2} & \cdots & \mathbf{K}_{B_S} \end{bmatrix} \quad (9)$$

where

$$\mathbf{Q}_{B_i B_j} = \mathbf{K}_{B_i M} \mathbf{K}_M^{-1} \mathbf{K}_{M B_j}. \quad (10)$$

PIC approximation adopts the independence assumption for test (synthetic) data in the same way as that for the training data. The predictive distribution of test data is expressed by a Gaussian distribution. Let $\mathcal{D}_T = \{(\mathbf{x}_t, y_t) | t = 1, \dots, T\}$ be a test data set, and $\mathbf{y}_{T_s}$ be a partial vector of $\mathbf{y}_T$, which consists of test data assigned to the block s. Then, we obtain following predictive distribution for the synthetic acoustic feature trajectory.

$$p(\mathbf{y}_T|\mathbf{y}, \boldsymbol{\theta}) = \mathcal{N}(\mathbf{y}_T; \boldsymbol{\mu}_{\mathbf{y}_T|\mathbf{y}}, \boldsymbol{\Sigma}_{\mathbf{y}_T|\mathbf{y}}) \quad (11)$$

$$\boldsymbol{\mu}_{\mathbf{y}_T|\mathbf{y}} = \begin{bmatrix} \boldsymbol{\mu}_{\mathbf{y}_{T_1}|\mathbf{y}}^\top & \cdots & \boldsymbol{\mu}_{\mathbf{y}_{T_S}|\mathbf{y}}^\top \end{bmatrix}^\top \quad (12)$$

$$\boldsymbol{\Sigma}_{\mathbf{y}_T|\mathbf{y}} = \begin{bmatrix} \boldsymbol{\Sigma}_{\mathbf{y}_{T_1 T_1}|\mathbf{y}} & \cdots & \boldsymbol{\Sigma}_{\mathbf{y}_{T_1 T_S}|\mathbf{y}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}_{\mathbf{y}_{T_S T_1}|\mathbf{y}} & \cdots & \boldsymbol{\Sigma}_{\mathbf{y}_{T_S T_S}|\mathbf{y}} \end{bmatrix}. \quad (13)$$

Here, let

$$\mathbf{K}_{N T_s}^{\mathrm{PIC}} = \begin{bmatrix} \mathbf{Q}_{B_1 T_s} & \cdots & \mathbf{Q}_{B_{s-1} T_s} & \mathbf{K}_{B_s T_s} & \mathbf{Q}_{B_{s+1} T_s} & \cdots & \mathbf{Q}_{B_S T_s} \end{bmatrix} \quad (14)$$

then the partial vector $\boldsymbol{\mu}_{\mathbf{y}_{T_S}|\mathbf{y}}$ and matrix $\boldsymbol{\Sigma}_{\mathbf{y}_{T_S}|\mathbf{y}}$ are given by

$$\boldsymbol{\mu}_{\mathbf{y}_{T_s}|\mathbf{y}} = \left(\mathbf{K}_{N T_s}^{\mathrm{PIC}}\right)^\top \left(\mathbf{K}_N^{\mathrm{PIC}} + \sigma_\nu^2 \mathbf{I}\right)^{-1} \mathbf{y} \quad (15)$$

$$\boldsymbol{\Sigma}_{\mathbf{y}_{T_i T_j}|\mathbf{y}} = \mathbf{Q}_{T_i T_j} + \delta_{T_i T_j} (\mathbf{K}_{T_i T_j} - \mathbf{Q}_{T_i T_j} + \sigma_\nu^2 \mathbf{I}) - \left(\mathbf{K}_{N T_i}^{\mathrm{PIC}}\right)^\top \left(\mathbf{K}_N^{\mathrm{PIC}} + \sigma_\nu^2 \mathbf{I}\right)^{-1} \mathbf{K}_{N T_j}^{\mathrm{PIC}}. \quad (16)$$

Since the matrix $\left(\mathbf{K}_N^{\mathrm{PIC}} + \sigma_\nu^2 \mathbf{I}\right)$ is the sum of the low rank matrix $\mathbf{Q}_N$ and the block diagonal matrix, we can speed up the inversion of $\left(\mathbf{K}_N^{\mathrm{PIC}} + \sigma_\nu^2 \mathbf{I}\right)$ using the Woodbury, Sherman & Morrison formula [10]. When the maximum block size is B, the computational cost for training results in $O(S(B+M)^3))$, whereas GP without any approximation methods needs $O(N^3)$ computational cost.

3. PARAMETER TRAJECTORY GENERATION USING PREDICTIVE DISTRIBUTION WITH GV

A simple way to generate a speech parameter trajectory from the predictive distribution is using the acoustic feature trajectory $\mathbf{y}_T$ that maximizes the likelihood of predictive distribution $p(\mathbf{y}_T|\mathbf{y}, \boldsymbol{\theta})$. Since the predictive distribution is a Gaussian distribution, the trajectory that maximizes the likelihood becomes the mean sequence $\boldsymbol{\mu}_{\mathbf{y}_T|\mathbf{y}}$. However, the resultant trajectories tend to be overly smoothed. To alleviate this effect, we incorporate global variance (GV) constraint [9] into trajectory generation.

GV is an utterance-level feature which is obtained by

$$v(\mathbf{y}_T) = \frac{1}{T} \sum_{t=1}^{T} (y_t - m(\mathbf{y}_T))^2 \quad (17)$$

$$m(\mathbf{y}_T) = \frac{1}{T} \sum_{t=1}^{T} y_t \quad (18)$$

where T is the number of frames of an utterance. To incorporate GV into GP-based speech synthesis, we define the following likelihood

$$\mathcal{L}_{\mathrm{GV}} = p(\mathbf{y}_T|\mathbf{y}, \boldsymbol{\theta})^\omega p(v(\mathbf{y}_T)|\mu_v, \sigma_v^2) \quad (19)$$

where μ_v and σ_v^2 correspond to the mean and variance of GVs of training utterances, respectively. The parameter ω adjusts the weight of Gaussian process likelihood and GV likelihood. In this study, we use $\omega = 1/T$, i.e., the dimensional ratio of v to $\mathbf{y}_T$. We generate a speech parameter trajectory by maximizing $\mathcal{L}_{\mathrm{GV}}$. Practically, in order to search the optimum trajectory, we use a steepest descent algorithm with the derivative

$$\frac{\partial \mathcal{L}_{\mathrm{GV}}}{\partial \mathbf{y}_T} = -\omega \boldsymbol{\Sigma}_{\mathbf{y}_T|\mathbf{y}}^{-1} (\mathbf{y}_T - \boldsymbol{\mu}_{\mathbf{y}_T|\mathbf{y}}) - \frac{2}{T} \cdot \frac{(v(\mathbf{y}_T) - \mu_v)}{\sigma_v^2} (\mathbf{y}_T - m(\mathbf{y}_T) \cdot \mathbf{1}). \quad (20)$$

4. OPTIMIZATION OF KERNEL HYPERPARAMETERS

Here we derive how to estimate an appropriate hyperparameter set $\boldsymbol{\theta}$ automatically for PIC approximation. In the same way as usual GPs, we take the empirical Bayesian approach, where we search optimum hyperparameter set $\boldsymbol{\theta}$ that maximizes the marginal likelihood $p(\mathbf{y}|\boldsymbol{\theta})$. Since PIC approximation includes a latent variable $\mathbf{f}_M$, EM algorithm can be applied to the maximization. Q-function is defined by

$$Q(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) = \int p(\mathbf{f}_M|\mathbf{y}, \boldsymbol{\theta}) \log \left(p(\mathbf{y}|\mathbf{f}_M, \tilde{\boldsymbol{\theta}}) p(\mathbf{f}_M|\tilde{\boldsymbol{\theta}}) \right) d\mathbf{f}_M \quad (21)$$

where $\tilde{\boldsymbol{\theta}}$ is a new hyperparameter set. Using the assumption of PIC approximation, the Q-function is decomposed into two Q-functions Q_s and Q_M as follows:

$$Q(\theta, \tilde{\theta}) = \sum_{s=1}^{S} Q_s(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) + Q_M(\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}) \quad (22)$$

$$Q_s(\theta, \tilde{\theta}) = \int p(\mathbf{f}_M|\mathbf{y}, \boldsymbol{\theta}) \log p(\mathbf{y}_{B_s}|\mathbf{f}_M, \tilde{\boldsymbol{\theta}}) d\mathbf{f}_M \quad (23)$$

$$Q_M(\theta, \tilde{\theta}) = \int p(\mathbf{f}_M|\mathbf{y}, \boldsymbol{\theta}) \log p(\mathbf{f}_M|\tilde{\boldsymbol{\theta}}) d\mathbf{f}_M. \quad (24)$$

In the E-step, we calculate the posterior distribution of pseudo-data variables by

$$p(\mathbf{f}_M|\mathbf{y},\boldsymbol{\theta}) = \mathcal{N}(\mathbf{f}_M;\boldsymbol{\mu}_{\mathbf{f}_M|\mathbf{y}},\boldsymbol{\Sigma}_{\mathbf{f}_M|\mathbf{y}}). \quad (25)$$

From Eqs. (2) and (4), we can derive the parameters

$$\boldsymbol{\mu}_{\mathbf{f}_M|\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{f}_M|\mathbf{y}}\mathbf{K}_M^{-1} + \sum_{s=1}^{S}\mathbf{K}_{MB_s}\boldsymbol{\Sigma}^{-1}_{\mathbf{y}_{B_s}|\mathbf{f}_M}\mathbf{y}_{B_s} \quad (26)$$

$$\boldsymbol{\Sigma}^{-1}_{\mathbf{f}_M|\mathbf{y}} = \mathbf{K}_M^{-1} + \mathbf{K}_M^{-1}\sum_{s=1}^{S}\mathbf{K}_{MB_s}\boldsymbol{\Sigma}^{-1}_{\mathbf{y}_{B_s}|\mathbf{f}_M}\mathbf{K}_{B_sM}\mathbf{K}_M^{-1}. \quad (27)$$

When the block size is B and pseudo-data size is M, we need $\mathcal{O}(S(B+M)^3)$ calculations for the each E-step, which is equal to the number of calculations of the model training with PIC approximation.

In the M-step, we employ generalized EM algorithm because it is difficult to find the exact hyperparameter $\tilde{\boldsymbol{\theta}}^*$ that maximizes Q-function. Specifically, we increase the value of Q-function using a gradient-based method. Furthermore, we may use stochastic gradient descent (SGD) algorithm because the Q-function is represented as an average form

$$Q(\theta,\tilde{\theta}) = \frac{1}{S}\sum_{s=1}^{S}\left(SQ_s(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}) + Q_M(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}})\right). \quad (28)$$

For each step of SGD, we choose a block randomly and update hyperparameters. The i-th hyperparameter at time k is updated as follows.

$$\tilde{\theta}_i^{(k+1)} = \tilde{\theta}_i^{(k)} + \eta_i^{(k)}\left(S\frac{\partial Q_s(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}^{(k)})}{\partial\tilde{\theta}_i^{(k)}} + \frac{\partial Q_M(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}}^{(k)})}{\partial\tilde{\theta}_i^{(k)}}\right) \quad (29)$$

where $\eta_i^{(k)}$ represents a step size of i at time k.

5. EXPERIMENTS

5.1. Kernel definition and hyperparameter setting

Based on the previous study [5, 6], we defined the following kernel function.

$$k(\mathbf{x}_m,\mathbf{x}_n) = \sum_{i\in\{-1,0,+1\}}\sum_{j\in\{-1,0,+1\}} w_m^{(i)}w_n^{(i)}k_p(\mathbf{p}_m^{(i)},\mathbf{p}_n^{(j)})k_c(\mathbf{c}_m^{(i)},\mathbf{c}_n^{(j)}) + \delta_{utt(m)utt(n)}k_\tau(\tau_m,\tau_n) + \delta_{mn}\theta^2_{\text{floor}}. \quad (30)$$

The first term represents the core of this kernel. The superscripts -1, 0, and $+1$ of the variables correspond to the preceding, current, and succeeding phones of the current frame. Using those adjacent phones, the values of kernel function get smooth around phone boundaries. $k_p(\cdot)$ and $k_c(\cdot)$ correspond to relative position kernel and phone context kernel. $\mathbf{p}_n^{(i)} = (p_{n,1}^{(i)}, p_{n,2}^{(i)}, p_{n,3}^{(i)})$ represents the relative frame position information in the phone. $p_{n,1}^{(i)}$ is the normalized position, and $p_{n,2}^{(i)}$ and $p_{n,3}^{(i)}$ are position differences from the beginning and end of the phone, respectively. We used sum of squared exponential (SE) kernel for the relative position kernel $k_p(\mathbf{p}_m^{(i)},\mathbf{p}_n^{(j)})$ as follows:

$$k_p(\mathbf{p}_m^{(i)},\mathbf{p}_n^{(j)}) = \sum_{k=1}^{3}\theta^2_{pa,k}\exp\left(-\frac{\left(p_{m,k}^{(i)}-p_{n,k}^{(j)}\right)^2}{2\theta^2_{pb,k}}\right) \quad (31)$$

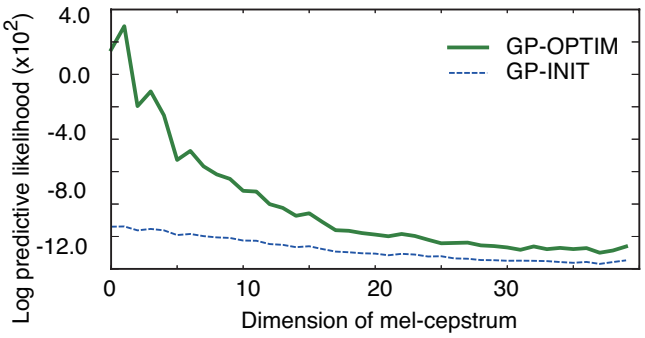

Fig. 1. Log likelihood of predictive distribution for each dimension using optimized/non-optimized hyperparameters

where $\theta_{pa,k}$ and $\theta_{pb,k}$ correspond to relevance and scale parameters for input feature $p_{n,k}^{(j)}$. For the phone context kernel $k_c(\mathbf{c}_m^{(i)},\mathbf{c}_n^{(j)})$, we used a linear kernel in the same way as [5] defined by

$$k_c(\mathbf{c}_m^{(i)},\mathbf{c}_n^{(j)}) = \sum_{k=1}^{3P}\theta^2_{c,k}c_{m,k}^{(i)}c_{n,k}^{(j)} \quad (32)$$

where $\theta_{c,k}$ is a relevance parameter, and $\mathbf{c}_n^{(j)}$ is a binary-valued distinctive phonetic feature (DPF) [11] vector including preceding, current, and succeeding phonemes. P is the number of DPFs and we used $P = 13$ in according with the previous study [5]. A weight parameter $w_n^{(i)}$ was used to emphasize the effect of closer phones to the n-th frame, and defined by a sine window in the same way as [6].

The second term of the right side of (30) is introduced to model short time correlation within the same utterance, where $utt(n)$ is the utterance index of frame n. The input feature of kernel $k_\tau(\cdot)$, τ_n, is the time in the utterance. We used the following SE kernel

$$k_\tau(\tau_m,\tau_n) = \theta^2_{\tau a}\exp\left(-\frac{(\tau_m-\tau_n)^2}{2\theta^2_{\tau b}}\right) \quad (33)$$

The last term in (30) is a flooring value to keep $\mathbf{K}_M$ positive definite because $\mathbf{K}_M^{-1}$ needs to be calculated.

In summary, the hyperparameter set used in this study became

$$\boldsymbol{\theta} = (\theta_{pa,1},\ldots,\theta_{pa,3},\theta_{pb,1},\ldots,\theta_{pb,3},\theta_{c,1}\ldots,\theta_{c,3P}, \theta_{\tau a},\theta_{\tau b},\theta_{\text{floor}},\sigma_\nu). \quad (34)$$

The total number of hyperparameters included in $\boldsymbol{\theta}$ resulted in 49. We set initial values of the hyperparameters based on preliminary experimental results. For the relevance hyperparameters $\theta^2_{pa,k}$ and $\theta^2_{c,k}$, we used equally divided values as $\theta^2_{pa,k} = 1/3(k = 1,2,3)$ and $\theta^2_{c,k} = 1/3P(k = 1,\ldots,3P)$. Moreover $\theta_{\tau a}$ and σ^2_ν were set to 1.0 in order to let the core kernel, short-time correlation kernel, and, noise term have equal relevance. On the other hand, we used a relatively small value for the flooring value 0.01 for the hyperparameter θ^2_{floor}. The scale hyperparameter for the normalized position, $\theta_{pb,1}$, was set to 0.289 which is equal to the standard deviation of the uniform distribution on $[0,1)$. The other scale hyperparameters $\theta_{pb,2}$ and $\theta_{pb,3}$ were set to 0.010s to consider correlations of frames of shorter range than $\theta_{pb,1}$, and the scale hyperparameter for short-time correlation kernel, $\theta_{\tau b}$, was set to 0.020s.

5.2. Experimental conditions

We used speech data of a Japanese female voice actress. The speaker uttered 503 phonetically balanced sentences with a reading style.

Table 1. Mel-cepstral distances between original and synthetic speech [dB].

Generation	HMM	GP-INIT	GP-OPTIM
w/o GV	5.38	5.11	5.03
w/ GV	5.95	5.59	5.55

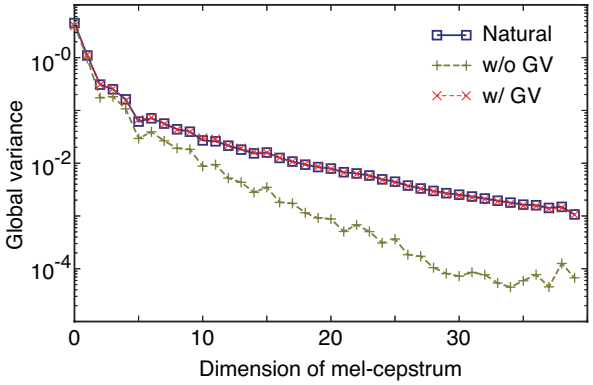

Fig. 2. Global variances of natural and generated mel-cepstrum sequences. These values show GV means over all test utterances. The optimized hyperparameters were used for generation.

These sentences were taken from ATR Japanese speech database set B [12]. In this study, we modeled and generated spectral features only. Speech samples were synthesized using generated spectral features, while F0s, aperiodicity features, and phone durations were taken from the original speech.

The number of training sentences was 450. The remaining 53 sentences, which were not included in the training data, were used as test data. The phone boundary information was annotated manually. Speech signals were sampled at a rate of 16kHz, and the frame shift was 5ms. The 0-39th mel-cepstral coefficients derived from the spectral envelop extracted by STRAIGHT [13] were used as the spectral features. The maximum number of frames B of each block was set to 1000. The number of pseudo data sets M was set to 200 and the samples of pseudo data sets were randomly chosen from the training data. The number of iterations of EM algorithm in hyperparameter optimization was 5.

For comparison, we also evaluated the HMM-based speech synthesis using minimum generation error (MGE) training [14]. The model topology was 5-state, left-to-right, no-skip hidden semi-Markov model (HSMM). The output distribution in each state was modeled with a single Gaussian pdf, with diagonal covariance matrices. The feature vector included delta and delta-delta dynamic features as well as the static one. Triphones were used for the context set for the HMM training. In the decision-tree-based context clustering for parameter tying, MDL was used as a stopping criterion [15].

5.3. Objective evaluation

To evaluate the effectiveness of the hyperparameter estimation objectively, log predictive likelihoods and spectral distortions are calculated. Figure 1 shows the average log predictive likelihoods of original utterances obtained by the predictive distribution $p(\mathbf{y}_T|\mathbf{y}, \boldsymbol{\theta})$. GP-INIT and GP-OPTIM represent the GP models using the initial kernel hyperparameters and optimized ones. It can be seen that the hyperparameter optimization consistently increased the predictive likelihood especially in lower dimensions. Table 1 shows the spectral distortions between original and synthetic speech. The table includes the generation methods without (w/o) and with (w/) GV. In both generation method, the hyperparameter optimization reduced the distortions. As a well-known effect of using GV, the trajectory generation considering GV had larger distortions than without GV case, whereas the global variances of the generated trajectories got closer to natural speech, by comparing average global variance as shown in Fig. 2. It should also be noted that the distortions of GP-based method were smaller than HMM-based ones.

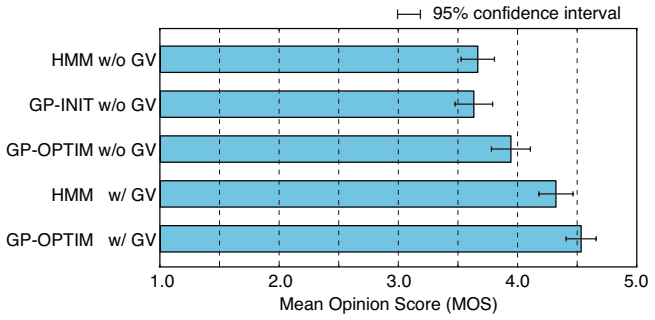

Fig. 3. MOS on naturalness of synthetic speech.

5.4. Subjective evaluation

The naturalness of the synthetic speech was evaluated by a mean opinion score (MOS) test. The number of participants was six. Each participant listened to the synthetic speech samples and rated the naturalness of synthetic speech on a five-point scale: 5: excellent, 4: good, 3: fair, 2: poor, and 1: bad. Fifteen sentences were randomly chosen from 53 sentences for each participant. Figure 3 shows the mean opinion scores (MOSs). The error bars indicate 95% confidence intervals. Note that every MOS score was higher than ordinary TTS ones because the original prosodic features were used as it were in the waveform generation. From the figure, it is seen that hyperparameter optimization increased the score in the case of "w/o GV." In addition, considering GV increased the scores in the GP-based approach as well as in the HMM-based one, and GP-OPTIM with GV gave the highest score among the methods. There was a significant difference between HMM with GV and GP-OPTIM with GV at a 5% significance level.

6. CONCLUSIONS

In this paper, we proposed a generation method of acoustic feature trajectory considering GV in GP-based statistical speech synthesis approach. Moreover hyperparameter optimization for PIC approximation using EM algorithm are introduced. The proposed method which uses GV and hyperparameter optimization outperformed the conventional HMM-based approach by subjective evaluation. In our future work, we will add contexts such as accent and part of speech that are used in general TTS systems and sub-phone or state information. Then, the relevance's of these contexts will be automatically determined by the hyperparameter optimization process. Furthermore, we will examine GP-based models of F0 and duration features using both prosodic contexts.

7. REFERENCES

[1] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. EUROSPEECH*, 1999, pp. 2347–2350.

[2] H. Zen, K. Tokuda, and A.W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[4] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, "F0 contour prediction with a deep belief network-Gaussian process hybrid model," in *Proc. ICASSP*, 2013, pp. 6885–6889.

[5] T. Koriyama, T. Nose, and T. Kobayashi, "Frame-level acoustic modeling based on Gaussian process regression for statistical nonparametric speech synthesis," in *Proc. ICASSP*, 2013, pp. 4589–4593.

[6] T. Koriyama, T. Nose, and T. Kobayashi, "Statistical nonparametric speech synthesis using sparse Gaussian processes," in *Proc. INTERSPEECH*, 2013, pp. 1072–1076.

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT press Cambridge, MA, 2006.

[8] E. Snelson and Z. Ghahramani, "Local and global sparse Gaussian process approximations," in *Proc. AISTATS*, 2007.

[9] T. Toda and K. Tokuda, "A speech parameter generation algorithm considering global variance for HMM-based speech synthesis," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[10] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge Univ. Press, 1992.

[11] T. Fukuda and T. Nitta, "Orthogonalized distinctive phonetic feature extraction for noise-robust automatic speech recognition," *IEICE Trans. Inf. & Syst.*, vol. 87, no. 5, pp. 1110–1118, 2004.

[12] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, Aug. 1990.

[13] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, "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, no. 3-4, pp. 187–207, 1999.

[14] Y. J. Wu and R. H. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. ICASSP*, 2006, vol. 1, pp. 889–892.

[15] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *Acoustical Science and Technology*, vol. 21, no. 2, pp. 79–86, 2000.