MIXTURE OF GAUSSIAN PROCESS EXPERTS FOR PREDICTING SUNG MELODIC CONTOUR WITH EXPRESSIVE DYNAMIC FLUCTUATIONS

Yasunori Ohishi†, *Daichi Mochihashi*‡, *Hirokazu Kameoka*†, *Kunio Kashino*†

†NTT Communication Science Laboratories, NTT Corporation, ‡The Institute of Statistical Mathematics

ABSTRACT

We present a generative model for predicting the sung melodic contour, i.e., F_0 contour, with expressive dynamic fluctuations, such as *vibrato* and *portamento*, for a given musical score. Although several studies have attempted to characterize such fluctuations, no systematic method has been developed for generating the F_0 contour with them in connection with musical notes. In our model, the relationship between a musical note sequence and F_0 contour is directly learned by a mixture of Gaussian process experts. This approach allows us to automatically characterize the fluctuations by utilizing the kernel function for each Gaussian process expert and predict the F_0 contour for an arbitrary musical note sequence. Experimental results show that our model can better predict the F_0 contour than a baseline method can. Additionally, we discuss the effective musical contexts and the amount of training data for the prediction.

Index Terms— Singing voice, fundamental frequency (F_0), mixture of Gaussian process experts, multiple kernel learning, Markov chain Monte Carlo method

1. INTRODUCTION

The goal of this study is to build a generative model that can characterize the singing style of a singer in sung melodic contours, i.e., F_0 contours, and predict the F_0 contour that reflects the personal singing style for an arbitrary musical score. Although no firm definition has yet been established for "singing style" in music information processing research, several studies have reported the relationship between singing styles and signal features, such as singing formant [1,2] and singing fluctuations [3–6]. Specifically, various research efforts have attempted to characterize the fluctuations, such as *vibrato* and *portamento* [5,7–9]. Vibrato is a quasi-periodic variation in the frequency of a sustained musical note and is described with the following parameters: rate, extent, waveform, regularity, time delay until vibrato onset, and the percentage of vibrato present for the duration [8–13]. Portamento is a gradual sliding of pitch smoothly and continuously from one note to another when well-separated musical notes are sung in the same breath [14,15]. Accordingly, modeling the expressive fluctuations enables us to characterize the singing style of a singer, and it is potentially very beneficial for any singing voice application, such as singing style conversion, singer identification, and the synthesis of more varied singing voices [16–20].

Several application systems that characterize the fluctuations have been reported. [21] proposed a method for calculating the vibrato rate and extent to evaluate singing skill automatically. [8] verified the vibrato features (rate, extent, and duration) for synthesizing more varied singing voices. [9] modeled vibrato and portamento as the sum of the periodic modulation plus a slow continuous variation, and used the estimated parameters for singer identification. In [7] and [22], a generative process for an F_0 contour was proposed using a second-order differential equation for singing style conversion. Although these studies attempted to characterize the fluctuations as a cue for singing skill and style, no systematic method has been developed for generating the various fluctuations in connection with the musical contexts as shown in Fig. 1. The fluctuations singers use differ depending on the musical score, and so it is important to learn this relationship. In singing voice synthesis based on hidden Markov models (HMMs), F_0 contours are characterized by Gaussian distributions for context-dependent states [18, 19, 23, 24]. However, an HMM is not always an appropriate model, because its hidden-state space is discrete in spite of the continuous and rapid changes in the fluctuations. Furthermore, a context-dependent decision-tree clustering has been employed to make robust models for new contexts, but averaging the F_0 contours of a leaf node of the tree causes an over-smoothing effect.

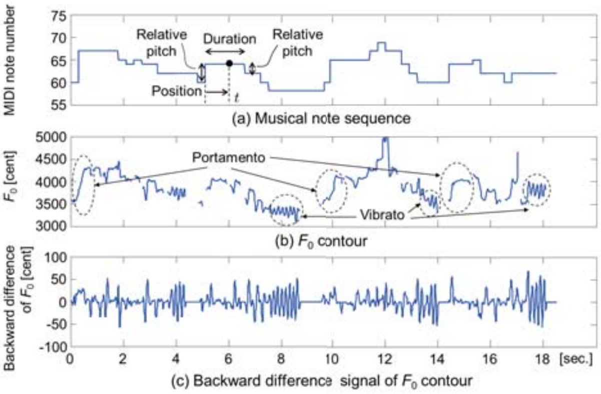

Fig. 1. Musical note sequence and singing voice F_0 contour

In this paper, we propose a generative model that uses a mixture of Gaussian process experts (MoGPEs) [25] to predict F_0 contours with expressive fluctuations for a given musical score. Each GPE directly learns the relationship between the musical contexts and an expressive fluctuation utilizing the kernel function. The MoGPEs represents the continuous transition of the fluctuations as a mixture model, and then the F_0 contour for an arbitrary musical score is generated by the predictive distribution of the MoGPEs. One merit of this representation is that since the F_0 contour is not deterministic, i.e., it varies across singing styles, the fluctuations are characterized stochastically. Thus we design the kernel functions for characterizing the fluctuations elaborately. Experimental results show the effectiveness of using the MoGPEs in terms of predicting the F_0 contour for a new musical score. We also discuss the effective musical contexts and the amount of training data for learning our model.

2. GENERATIVE MODELING OF F_0 CONTOUR

First, we briefly explain the input and output features for the MoGPEs. As shown in Fig. 1, we use singing voice signals synchronized

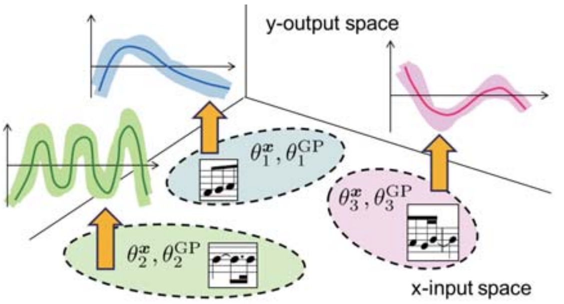
y-output space
$\theta_1^{\boldsymbol{x}}, \theta_1^{\mathrm{GP}}$
$\theta_3^{\boldsymbol{x}}, \theta_3^{\mathrm{GP}}$
$\theta_2^{\boldsymbol{x}}, \theta_2^{\mathrm{GP}}$
x-input space

Fig. 2. An example allotment of input space in the MoGPEs. Each elliptical region represents a two-dimensional full covariance Gaussian distribution. The shaded regions are owned by the same GPE.

with melodies, which are represented in standard MIDI file format. Fig. 1 (a) shows the musical note sequence of a melody. Input vector $\boldsymbol{x}_t$ consists of the position and duration of the current musical note on the second time scale, the relative pitch between the current note and the preceding note, and the relative pitch between the current note and the succeeding note, all of which are extracted from this sequence every 10 ms. It is certainly possible to add phonemes labels and dynamic marks such as *crescendo* to the input vector. On the other hand, we use the backward-differential values of the F_0 contour as the outputs $\{y_t\}_{t=1}^T$ to remove musical note information from the F_0 contour [Fig. 1 (c)]. The F_0 value is estimated every 10 ms using YIN [26]. Then, the F_0 frequency in hertz is converted to cents by

$$y_{\texttt{cent}} = 1200 \log_2 \frac{y_{\text{Hz}}}{440 \times 2^{\frac{3}{12}-5}}, \tag{1}$$

so that one equal-tempered semitone corresponds to 100 cents.

Next, we explain the standard GP regression [27]. Suppose that we have T observations as a training set $\mathcal{D} = \{\boldsymbol{x}_t, y_t\}_{t=1}^T$. When output vector $\boldsymbol{y} = [y_1, \ldots, y_T]^{\mathrm{T}}$ follows a GP, the probabilistic density function is represented as a multivariate Gaussian distribution

$$p(\boldsymbol{y}) = \mathcal{GP}(\boldsymbol{y}; \boldsymbol{0}, \boldsymbol{K} + \eta^2 \boldsymbol{I}) = \mathcal{N}(\boldsymbol{y}; \boldsymbol{0}, \boldsymbol{K} + \eta^2 \boldsymbol{I}), \tag{2}$$

where $\boldsymbol{K}$ is a Gram matrix whose element is given by $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is calculated by the kernel function, which defines "similarity" between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. η^2 and $\boldsymbol{I}$ denote the noise variance and identity matrix, respectively. The goal of GP regression is to infer the predictive distribution of y_* given a new input vector $\boldsymbol{x}_*$. The predictive distribution is given by

$$\begin{aligned} p(y_*|\boldsymbol{y}, X, \boldsymbol{x}_*) &= \mathcal{N}(y_*; \mu_*, \sigma_*^2), \\ \mu_* &= \boldsymbol{k}_*^{\mathrm{T}}(\boldsymbol{K} + \eta^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \\ \sigma_*^2 &= k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_*^{\mathrm{T}}(\boldsymbol{K} + \eta^2 \boldsymbol{I})^{-1}\boldsymbol{k}_*, \end{aligned} \tag{3}$$

where $\boldsymbol{k}_*$ is the column Gram vector, which consists of the elements $k(\boldsymbol{x}_t, \boldsymbol{x}_*)$ $(t = 1, \ldots, T)$. The exemplars in the training set contribute to the prediction based on their correlation to the new input, as measured in input space.

The requirement for the kernel function is that the Gram matrix should be positive semi-definite and symmetric. On the assumption that the output signal is stationary, the squared exponential (SE) covariance function and the rational quadratic covariance function are generally used. However, as shown in Fig. 1, the output signal is not always stationary because a singer sings while using different fluctuations depending on the musical note sequence. We employ the MoGPEs to model the expressive fluctuations which have non-stationary dynamics.

In the MoGPEs, as shown in Fig. 2, the input space is probabilistically divided by a gating network [28] into regions within which specific separate experts make predictions. Each GPE learns different characteristics of the fluctuation. Of course, the learning of the experts and that of the gating network are intimately coupled. Finally, the outputs are represented as the mixture of these experts. Two types of the MoGPEs have been proposed [25, 29], and the difference between these formulations is whether or not a full generative model over inputs and outputs is defined. Since [25] defines a full generative model of the MoGPEs that has a number of potential advantages, such as the ability to deal with partial specified data and infer inverse functional mappings, we utilize this formulation. The generative process is as follows:

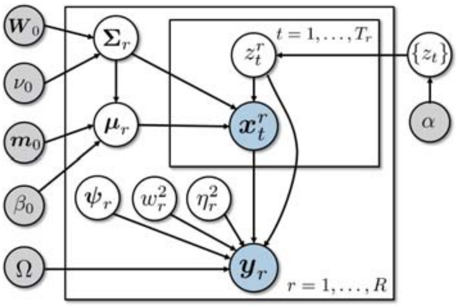

Fig. 3. Graphical representation of our model

1. Construct a partition of T observations into R groups using the Dirichlet-multinomial distribution. This assignment is denoted by using a set of the indicator variables $\{z_t\}_{t=1}^T$.
2. For each grouping of indicators $\{z_t : z_t = r\}$, sample the input space parameters $\theta_r^{\boldsymbol{x}} \equiv \{\boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r\}$. $\theta_r^{\boldsymbol{x}}$ defines a full-covariance Gaussian.
3. Given the parameters $\theta_r^{\boldsymbol{x}}$ for each group, sample the input vectors $X_r \equiv \{\boldsymbol{x}_t : z_t = r\}$.
4. For each group, estimate parameters θ_r^{GP} for the GP expert.
5. Using the input vectors X_r and parameters θ_r^{GP} for the individual groups, formulate the GP output covariance matrix and sample the set of output values, $\boldsymbol{y}_r \equiv \{y_t : z_t = r\}$, from the joint Gaussian distribution.

The graphical representation of this process is shown in Fig. 3. The full joint distribution is given by

$$\begin{aligned} &p(\{\boldsymbol{x}_t, y_t\}_{t=1}^T, \{z_t\}_{t=1}^T, \{\theta_r^{\mathrm{GP}}\}_{r=1}^R, \{\theta_r^{\boldsymbol{x}}\}_{r=1}^R|\Omega) \\ &= \prod_{r=1}^R \Big[p(\theta_r^{\boldsymbol{x}}|\Omega)p(X_r|\theta_r^{\boldsymbol{x}})p(\boldsymbol{y}_r|X_r, \theta_r^{\mathrm{GP}}, \Omega)\Big] \times p(\{z_t\}_{t=1}^T|\Omega), \end{aligned} \tag{4}$$

where R and Ω are the number of experts and hyper-parameters, respectively, and we directly represent the indicators $\{z_t\}_{t=1}^T$ and sample them to capture their dependence using the Gibbs sampler [25]. The individual distributions in Eq. (4) are defined as follows:

$$p(\{z_t\}_{t=1}^T|\Omega) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + T)} \prod_{r=1}^R \frac{\Gamma(T_r + \alpha/R)}{\Gamma(\alpha/R)}, \tag{5}$$

$$p(\theta_r^{\boldsymbol{x}}|\Omega) = \mathcal{N}(\boldsymbol{\mu}_r; \boldsymbol{m}_0, \boldsymbol{\Sigma}_r/\beta_0)\mathcal{W}(\boldsymbol{\Sigma}_r^{-1}; \boldsymbol{W}_0, \nu_0), \tag{6}$$

$$p(X_r|\theta_r^{\boldsymbol{x}}) = \mathcal{N}(X_r; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r), \tag{7}$$

$$p(\boldsymbol{y}_r|X_r, \boldsymbol{\theta}_r^{\mathrm{GP}}, \Omega) = \mathcal{GP}\left(\boldsymbol{y}_r; \boldsymbol{0}, \boldsymbol{K}_r + \eta_r^2 \boldsymbol{I}_r\right), \tag{8}$$

where α is the hyper-parameter of the Dirichlet-multinomial distribution. T_r, $\boldsymbol{I}_r$, η_r^2, and $\mathcal{W}$ denote the number of elements in X_r, the $T_r \times T_r$ identity matrix, noise variance, and Wishart distribution, respectively.

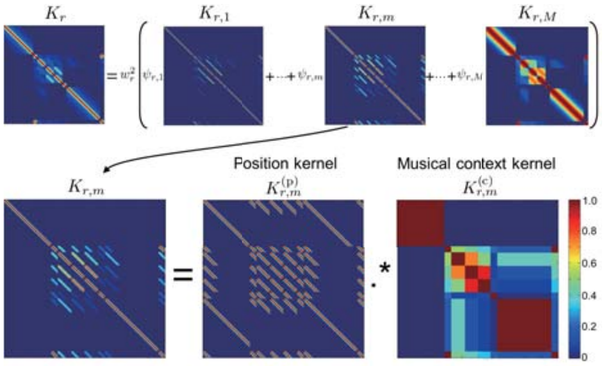

Fig. 4. Gram matrix based on multiple kernel learning

The Gram matrix $\boldsymbol{K}_r$ is calculated using the input vectors X_r and the parameters θ_r^{GP}. Using multiple kernel leaning [30, 31], we represent the function as the linear combination of kernel functions:

$$k_r(\boldsymbol{x}_i, \boldsymbol{x}_j) = w_r^2 \sum_{m=1}^{M} \psi_{r,m} k_{r,m}(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{9}$$

where θ_r^{GP} represents the amplitude w_r^2 and the weight $\psi_{r,m}$ of each kernel, and then $\boldsymbol{x}_i, \boldsymbol{x}_j \in X_r$, $\sum_{m=1}^{M} \psi_{r,m} = 1$. M is the number of kernel functions. For the input vector consisting of various features, we define $k_{r,m}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as the product of two kernels

$$k_{r,m}(\boldsymbol{x}_i, \boldsymbol{x}_j) = k_{r,m}^{(\mathrm{p})}(x_i^{(\mathrm{p})}, x_j^{(\mathrm{p})}) k_{r,m}^{(\mathrm{c})}(\boldsymbol{x}_i^{(\mathrm{c})}, \boldsymbol{x}_j^{(\mathrm{c})}), \tag{10}$$

where $k_{r,m}^{(\mathrm{p})}(x_i^{(\mathrm{p})}, x_j^{(\mathrm{p})})$ and $k_{r,m}^{(\mathrm{c})}(\boldsymbol{x}_i^{(\mathrm{c})}, \boldsymbol{x}_j^{(\mathrm{c})})$ denote the position kernel and musical context kernel, respectively, as shown in Fig. 4. The position kernel represents the similarity between the positions in the musical notes, whereas the musical context kernel represents the similarity between the musical contexts. Here vector $\boldsymbol{x}_i$ is divided into $x_i^{(\mathrm{p})}$ for the position and $\boldsymbol{x}_i^{(\mathrm{c})}$ for the musical context, and then the kernel functions are calculated for each variable. To capture the continuity and periodicity of the expressive fluctuations, the SE covariance function and the periodic covariance function are used for the position kernel:

$$k_{r,m}^{(\mathrm{p})}(x_i^{(\mathrm{p})}, x_j^{(\mathrm{p})}) = \exp\left(-\frac{(x_i^{(\mathrm{p})} - \boldsymbol{x}_j^{(\mathrm{p})})^{\mathrm{T}}(x_i^{(\mathrm{p})} - x_j^{(\mathrm{p})})}{2{l_m^{(\mathrm{p})}}^2}\right),$$

$$k_{r,m}^{(\mathrm{p})}(x_i^{(\mathrm{p})}, x_j^{(\mathrm{p})}) = \exp\left(-2\sin^2\left(\frac{l_m^{(\mathrm{p})}}{2\pi}(x_i^{(\mathrm{p})} - x_j^{(\mathrm{p})})\right)\right).$$

For the musical context kernel, we use the SE covariance function

$$k_{r,m}^{(\mathrm{c})}(\boldsymbol{x}_i^{(\mathrm{c})}, \boldsymbol{x}_j^{(\mathrm{c})}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}_i^{(\mathrm{c})} - \boldsymbol{x}_j^{(\mathrm{c})})^{\mathrm{T}}\boldsymbol{\Lambda}(\boldsymbol{x}_i^{(\mathrm{c})} - \boldsymbol{x}_j^{(\mathrm{c})})\right),$$

$$\boldsymbol{\Lambda}^{-1} = \mathrm{diag}({l_{m,1}^{(\mathrm{c})}}^2, {l_{m,2}^{(\mathrm{c})}}^2, \ldots, {l_{m,D_{\mathrm{c}}}^{(\mathrm{c})}}^2),$$

where D_{c} is the number of dimensions of $\boldsymbol{x}_i^{(\mathrm{c})}$. The parameters and the hyper-parameters are $\Theta = \{z_1, \ldots, z_T, \theta_1^{\boldsymbol{x}}, \ldots, \theta_R^{\boldsymbol{x}}, \theta_1^{\mathrm{GP}}, \ldots, \theta_R^{\mathrm{GP}}\}$ and $\Omega = \{\alpha, \boldsymbol{m}_0, \boldsymbol{W}_0, \beta_0, \nu_0, l_1^{(\mathrm{p})}, \ldots, l_M^{(\mathrm{p})}, l_{1,1}^{(\mathrm{c})}, \ldots, l_{M,D_{\mathrm{c}}}^{(\mathrm{c})}\}$, respectively. For information on how to set up the hyper-parameters, see the footnote of Section 4.

Finally, we derive the predictive distribution of y_* for a new input vector $\boldsymbol{x}_*$. The predictive distribution is given by

$$p(y_*|\{y_t, \boldsymbol{x}_t\}_{t=1}^{T}, \boldsymbol{x}_*, \Theta, \Omega) \tag{11}$$
$$= \sum_{r=1}^{R} p(y_*|\boldsymbol{y}_r, X_r, \boldsymbol{x}_*, z_* = r, \theta_r^{\mathrm{GP}}) p(z_* = r|\boldsymbol{x}_*, \theta_r^{\boldsymbol{x}}),$$

where the first term and the second term can be written as

$$p(y_*|\boldsymbol{y}_r, X_r, \boldsymbol{x}_*, z_* = r, \theta_r^{\mathrm{GP}}) = \mathcal{N}(y_*; \mu_{r,*}, \sigma_{r,*}^2), \tag{12}$$
$$\mu_{r,*} = \boldsymbol{k}_{r,*}^{\mathrm{T}}(\boldsymbol{K}_r + \eta_r^2 \boldsymbol{I}_r)^{-1}\boldsymbol{y}_r,$$
$$\sigma_{r,*}^2 = k_r(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}_{r,*}^{\mathrm{T}}(\boldsymbol{K}_r + \eta_r^2 \boldsymbol{I}_r)^{-1}\boldsymbol{k}_{r,*},$$
$$p(z_* = r|\boldsymbol{x}_*, \theta_r^{\boldsymbol{x}}) = \frac{p(\boldsymbol{x}_*|z_* = r, \theta_r^{\boldsymbol{x}})p(z_* = r)}{p(\boldsymbol{x}_*)}, \tag{13}$$
$$p(\boldsymbol{x}_*|z_* = r, \theta_r^{\boldsymbol{x}}) = \mathcal{N}(\boldsymbol{x}_*; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r),\ p(z_* = r) = 1/T_r.$$

Therefore, the predictive distribution of Eq. (11) can be rewritten as

$$p(y_*|\{y_t, \boldsymbol{x}_t\}_{t=1}^{T}, \boldsymbol{x}_*, \Theta, \Omega) = \mathcal{N}(y_*; \mu_*, \sigma_*^2), \tag{14}$$
$$\mu_* = \sum_{r=1}^{R} c_r \mu_{r,*},\ \sigma_*^2 = \sum_{r=1}^{R} c_r^2 \sigma_{r,*}^2,\ c_r \equiv p(z_* = r|\boldsymbol{x}_*, \theta_r^{\boldsymbol{x}}).$$

Based on this predictive distribution, the F_0 contour, $\{f_t\}_{t=1}^{T_m}$, is reproduced by

$$f_t = o_t + c - \bar{o}, \quad o_t = \textstyle\sum_{n=1}^{t} \mu_{*,n}, \quad \bar{o} = \sum_{t=1}^{T_m} o_t / T_m, \tag{15}$$

where c and T_m are the pitch and duration of a target musical note, respectively.

The GP regression has been applied to speech synthesis [32,33], F_0 contour prediction of speech [34], voice conversion [35], and music performance rendering [36]. In [33], the relationship between the input features, which consist of linguistic information, and the output, which consists of the spectral feature calculated at each frame, was learned. This method outperformed conventional HMM-based speech synthesis. Our model is related to these approaches but differs from them in that it capitalizes on characterizing the dynamics of the fluctuations elaborately utilizing the MoGPEs and various kinds of kernel functions, which was not considered in these earlier approaches. Furthermore, using GP regressions as experts, we gain the advantage that computation for each expert is cubic only in the number of data points in its region, rather than in the entire dataset.

3. INFERENCE

All the necessary parameter updates are straightforwardly carried out using a Markov Chain Monte Carlo (MCMC) and Expectation Maximization (EM) scheme [37]. Specifically, Gibbs sampling is used for the inference of $\{z_t\}_{t=1}^{T}$ and $\{\theta_r^{\boldsymbol{x}}\}_{r=1}^{R}$ [25]. Then we consider the multivariate Gaussian distribution of Eq. (8) as the sum of independent Gaussians, and then employ the EM algorithm for the inference of $\{\theta_r^{\mathrm{GP}}\}_{r=1}^{R}$ [31,38].

4. EVALUATION

We tested our model in terms of the predictive performance of the output for a new input vector. In this experiment, we used the F_0 contours and the MIDI signals of melodies annotated manually in the AIST annotation [39] (RWC-MDB-P-2001 [40], Song No. 38, 39, 42, 44, 45, 46, 64, 72, 74, and 76). These songs are sung by an identical singer. Although the F_0 contour should essentially be estimated from the acoustic signal, we used these data to evaluate the upper limit of the performance of our model.

Input vector $\boldsymbol{x}_t$ and the backward-differential value y_t are calculated every 10 ms from the musical note sequence and the F_0 contour, respectively. Since the F_0 values are not obtainable in the unvoiced segments, we removed those segments. We used song No.

Table 1. Average RMSEs for MoPRs and our model (MoGPEs)

Num. of experts	5	10	20	30	40	50
MoPRs	65.6	71.8	59.7	73.1	79.6	56.5
MoGPEs	**26.4**	**25.0**	**24.0**	**23.9**	**23.1**	**22.3**

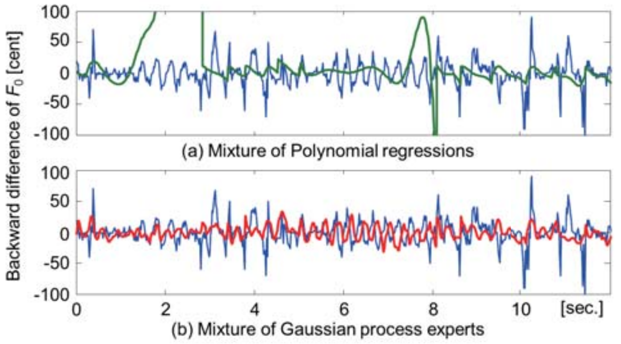

Fig. 5. Comparison of the prediction results: blue, green, and red lines correspond to the actual outputs, the result of the MoPRs, and the result of the MoGPEs, respectively.

38, 39, 42, 44, and 45 for training and song No. 46, 64, 72, 74, and 76 for testing. The training and test data consist of 649.3 sec. and 625.6 sec. of the singing voices, respectively. Using the training data, parameters Θ are estimated, and then the outputs are predicted for the input vectors of the test data. We initialize the indicators as the assignments obtained by k-means clustering of all the input vectors of the training data. Using the input vectors assigned to each expert, the input space parameters $\theta_r^{\boldsymbol{x}}$ are initialized. The number of kernel functions is set at $M = 216$, and the initial values of θ_r^{GP} are set at $w_r^2 = 100$, $\psi_{r,1} = 1/M, \ldots, \psi_{r,M} = 1/M$, $\eta_r^2 = 10$ $(r = 1, \ldots, R)$. The settings of the hyper-parameters are shown in the footnote [1]. The parameter inference was run for 50 iterations.

As the evaluation measure, we use the root mean square error (RMSE) between the actual outputs and the predictive means μ_* for the input vectors in the test data as follows:

$$\mathrm{RMSE} = \sqrt{\textstyle\sum_{t=1}^{T_t} (y_t - \mu_{*,t})^2 / T_t}, \tag{16}$$

where T_t is the signal length of the test data. As a conventional method, we defined a mixture of polynomial regressions (MoPRs) in which the relationship between the input and the output is modeled as a mixture of nth order polynomials. Specifically, we replace the GP regression of Eq. (8) with the polynomial regression [41].

Tab. 1 shows the average RMSEs when the number of experts is changed. Here we only used song No. 35 for the training. In the MoPRs we set n at 8, because it was difficult sometimes to stably estimate the parameters of the higher order polynomials. Our model outperformed the simple method consisting of the polynomial regressions, and the performance of our model improved as the number of experts increased. Fig. 5 compares the prediction results of our model and the MoPRs. We confirmed the effectiveness of using our model to capture the periodic dynamics of the expressive fluctuations.

Table 2. Average RMSEs for the number of musical contexts

Num. of musical contexts	1	2	3
MoGPEs	23.8	23.1	**22.3**

Table 3. Average RMSEs for the number of songs in training data

Num. of songs	1	3	5
MoGPEs	22.3	20.7	**20.5**

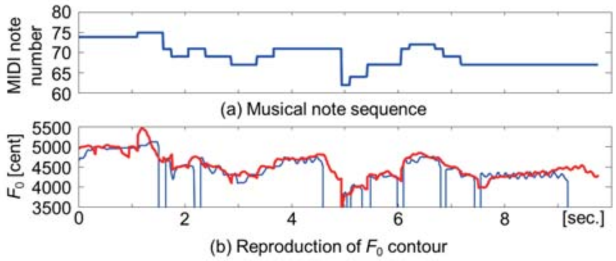

Fig. 6. Reproduction of F_0 contour: blue and red lines correspond to an actual F_0 contour and the reproduction, respectively.

Next, we discussed the effective musical contexts and the amount of training data. To find the effective musical contexts, we trained our model by selecting some of musical contexts described in Section 2, and then we predicted the outputs. The number of experts is set at $R = 50$. As shown in Tabs. 2 and 3, as the number of contexts and the number of songs in the training data increased, the performance of our model improved.

Fig. 6 shows the reproduction result for an F_0 contour obtained by Eq. (15). The reproduced F_0 contour is exactly the same as the actual F_0 contour, but it is still inadequate for predicting the individual fluctuations. To further improve the predictive performance, it is necessary to adjust the number of experts and the amount of training data. We can sidestep the model selection problem by using an infinite number of experts and employing the gating network related to the Dirichlet process to specify a spatially varying Dirichlet process [25, 29, 42]. Adding the long-term musical note sequence, the dynamic marks, and the articulation in the musical score to the input vector as binary variables is also future work. Finally, the F_0 contour reproduced by our model should be compared with the F_0 contour generated by HMM-based singing voice synthesis.

5. CONCLUSIONS

We proposed a generative model for predicting the sung melodic contour with the expressive dynamic fluctuations for an arbitrary musical note sequence. Specifically, each GPE directly learns the relationship between the musical contexts and an expressive fluctuation utilizing the kernel function, and then the MoGPEs represents the continuous transition of the fluctuations as a mixture model. The experimental results showed that our model is promising for predicting the F_0 contour for a given musical score.

To further improve the predictive performance, we plan to adjust the number of experts and the hyper-parameters, and simultaneously characterize the dynamics of the F_0 contour and the voice volume contour utilizing the MoGPEs. Although our model has great potential for recognizing and converting singing styles, we have not tested it yet. In future work, we will evaluate its ability to automatically classify singing styles and identify singers using Eq. (14), and apply our model to singing style conversion using a speech manipulation system such as STRAIGHT [43].

[1] The hyper-parameters are fixed to $\alpha = 1$, $\beta_0 = 0.1$, $\nu_0 = D + 1$, where D is the number of dimensions of the input vector. $\boldsymbol{m}_0$ is set to the mean value over all the input vectors for training data. $\boldsymbol{W}_0$ is set to the inverse matrix of the covariance matrix over all the input vectors divided by ν_0. The hyper-parameters of the position kernel are set to $\{l_m^{(\mathrm{p})}, m = 1, \ldots, 108 | 0.05, 0.11, 0.23, 0.5\}$ for the SE covariance functions and to $\{l_m^{(\mathrm{p})}, m = 109, \ldots, 216 | 0.13, 0.15, 0.17, 0.2\}$ for the periodic covariance functions. The hyper-parameters of the musical context kernel are set to $\{l_{m,1}^{(\mathrm{c})} | 1, 2.2, 5\}$, $\{l_{m,2}^{(\mathrm{c})} | 1, 2.2, 5\}$, and $\{l_{m,3}^{(\mathrm{c})} | 0.1, 0.55, 3\}$.

6. REFERENCES

[1] J. Sundberg, "Singing and timbre," *Music room acoustics*, vol. 17, pp. 57–81, 1977.

[2] J. Sundberg, *The Science of the Singing Voice*, Northern Illinois University Press, 1987.

[3] K. Kojima et al., "Variability of vibrato -a comparative study between Japanese traditional singing and bel canto-," in *Proc. Speech Prosody 2004*, pp. 151–154.

[4] I. Nakayama, "Comparative studies on vocal expressions in Japanese traditional and western classical- style singing, using a common verse," in *Proc. ICA 2004*, pp. 1295–1296.

[5] T. Saitou et al., "Development of an F0 control model based on F0 dynamic characteristics for singing-voice synthesis," *Speech Communication*, vol. 45, no. 3-4, pp. 405–417, 2005.

[6] V. Verfaille et al., "Perceptual evaluation of vibrato models," in *Proc. CIM 2005*.

[7] T. Saitou et al., "Speech-To-Singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices," in *Proc. WASPAA 2007*, pp. 215–218.

[8] N. Migita et al., "A study of vibrato features to control singing voices," in *Proc. ICA 2010*, pp. 23–27.

[9] L. Regnier, *Localization, Characterization and Recognition of Singing Voices*, Ph.D. thesis, IRCAM / UPMC in Paris, France, 2013.

[10] M. Metfessel, *The Vibrato in Artistic Voices*, University of Iowa Press.

[11] C. E. Seashore, "A musical ornament, the vibrato," in *Psychology of Music*. pp. 33–52, McGraw-Hill Book Company.

[12] D. Myers et al., "Vibrato and pitch transitions," *J. Voice*, vol. 1, no. 2, pp. 157–161, 1987.

[13] Y. Horii, "Acoustic analysis of vocal vibrato: A theoretical interpretation of data," *J. Voice*, vol. 3, no. 1, pp. 36–43, 1989.

[14] E. Pollastri, "Some considerations about processing singing voice for music retrieval," in *Proc. ISMIR 2002*.

[15] J. Potter, "Beggar at the door: the rise and fall of portamento in singing," *Music and Letters*, vol. 87, no. 4, pp. 523–550, 2006.

[16] T. Nakano et al., "VOCALISTENER2: A singing synthesis system able to mimic a user's singing in terms of voice timbre changes as well as pitch and dynamics," in *Proc. ICASSP 2011*, pp. 453–456.

[17] H. Kenmochi et al., "Singing synthesis as a new musical instrument," in *In Proc. ICASSP 2012*.

[18] K. Oura et al., "Pitch adaptive training for HMM-based singing voice synthesis," in *Proc. ICASSP 2012*, pp. 5377–5380.

[19] T. Nose et al., "A style control technique for singing voice synthesis based on multiple-regression HSMM," in *Proc. INTERSPEECH 2013*, pp. 378–382.

[20] H. Doi et al., "Evaluation of a singing voice conversion method based on many-to-many eigenvoice conversion," in *Proc. INTERSPEECH 2013*, pp. 1067–1071.

[21] T. Nakano et al., "An automatic singing skill evaluation method for unknown melodies using pitch interval accuracy and vibrato features," in *Proc. Interspeech 2006*, pp. 1706–1709.

[22] Y. Ohishi et al., "A stochastic model of singing voice F0 contours for characterizing expressive dynamic components," in *Proc. INTERSPEECH 2012*.

[23] T. Yoshimura et al., "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. EUROSPEECH 1999*.

[24] H. Zen et al., "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[25] E. Meeds et al., "An alternative infinite mixture of Gaussian process experts," in *Proc. NIPS 2006*.

[26] A. de Cheveigné et al., "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Mass, USA, 2006.

[28] R. A. Jacobs et al., "Adaptive mixture of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[29] C. E. Rasmussen et al., "Infinite mixtures of Gaussian process experts," in *Proc. NIPS 2002*.

[30] F. Bach et al., "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. ICML 2004*, pp. 6–13.

[31] K. Yoshii et al., "Infinite kernel linear prediction for joint estimation of spectral envelope and fundamental frequency," in *Proc. ICASSP 2013*, pp. 463–467.

[32] G. E. Henter et al., "Gaussian process dynamical models for nonparametric speech representation and synthesis," in *Proc. ICASSP 2012*, pp. 4505–4508.

[33] T. Koriyama et al., "Statistical nonparametric speech synthesis based on Gaussian process regression," in *Proc. INTERSPEECH 2013*, pp. 912–916.

[34] R. Fernandez et al., "F0 contour prediction with a deep belief network-Gaussian process hybrid model," in *Proc. ICASSP 2013*, pp. 6885–6889.

[35] N. C. V. Pilkington et al., "Gaussian process experts for voice conversion," in *Proc. INTERSPEECH 2011*, pp. 2761–2764.

[36] K. Teramura et al., "Gaussian process regression for rendering music performance," in *Proc. ICMPC 2008*.

[37] C. Andrieu et al., "An introduction to MCMC for machine learning," *Machine Learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[38] M. Feder et al., "Parameter estimation of superimposed signals using the EM algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 4, pp. 477–489, 1988.

[39] M. Goto, "AIST annotation for the RWC music database," in *Proc. ISMIR 2006*.

[40] M. Goto et al., "RWC music database: Popular, classical, and jazz music databases," in *Proc. ISMIR 2002*.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[42] A. Tayal et al., "Hierarchical double Dirichlet process mixture of Gaussian processes," in *Proc. AAAI 2012*.

[43] H. Kawahara et al., "TANDEM-STRAIGHT: A temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation," in *Proc. ICASSP 2008*, pp. 3933–3936.