interspeech 2008 incorporating SST 08

Parameter Estimation Method of F0 Control Model for Singing Voices

Yasunori Ohishi[1], *Hirokazu Kameoka*[2], *Kunio Kashino*[2], *Kazuya Takeda*[1]

[1]Graduate School of Information Science, Nagoya University
[2]NTT Communication Science Laboratories, NTT Corporation
ohishi@sp.m.is.nagoya-u.ac.jp, kameoka@eye.brl.ntt.co.jp,
kunio@eye.brl.ntt.co.jp, kazuya.takeda@nagoya-u.jp

10.21437/Interspeech.2008-32

Abstract

In this paper, we propose a novel representation of F0 contours that provides a computationally efficient algorithm for automatically estimating the parameters of a F0 control model for singing voices. Although the best known F0 control model, based on a second-order system with a piece-wise constant function as its input, can generate F0 contours of natural singing voices, this model has no means of learning the model parameters from observed F0 contours automatically. Therefore, by modeling the piece-wise constant function by Hidden Markov Models (HMM) and approximating the second order differential equation by the difference equation, we estimate model parameters optimally based on iteration of Viterbi training and an LPC-like solver. Our representation is a generative model and can identify both the target musical note sequence and the dynamics of singing behaviors included in the F0 contours. Our experimental results show that the proposed method can separate the dynamics from the target musical note sequence and generate the F0 contours using estimated model parameters.

Index Terms: Singing voice, Fundamental frequency (F0), F0 control model, Parameter optimization, Singing voice synthesis

1. Introduction

The goal of this study is to build a model that can represent both musical-note information and the dynamics of various singing behaviors (e.g., fluctuations in a musical note and continuous transitions between notes) in a sung melodic contour. Although a symbolic melodic contour (a sequence of musical notes) can be easily modeled by a discrete-time stochastic representation such as n-grams [1, 2], this representation cannot be used for modeling a sung melody because representing the singing dynamics of melodic contour, such as vibrato and overshoot, is difficult. The dynamic representation for modeling a sung melody is important for defining an appropriate melodic similarity between sung melodies, which is useful for various applications such as query-by-humming (QBH) and automatic clustering of songs [3].

On the other hand, various singing voice synthesis systems have been proposed [4, 5]. The F0 control model based on a second-order system, Eq. (1), is known to generate natural singing voices [4]. This model is represented by

$$\alpha \frac{d^2 y(t)}{dt^2} + \beta \frac{dy(t)}{dt} + \gamma y(t) = u(t), \qquad (1)$$

where $u(t)$ is the model input which is the melody component (a sum of step functions), and $y(t)$ is the generated F0 contour. By controlling parameters α, β, and γ, this model can generate various F0 contours with different local dynamics such as vibratos and overshoots. In [4], however, these parameters were determined manually based on subjective experiments. We believe that a framework is required to learn these parameters automatically from observed F0 contours in singing voice synthesis, just as speech recognition has been developed dramatically by HMM and effective learning algorithms. The establishment of this framework will result in more natural singing voice synthesis that reflect personal singing behaviors, the transcription of singing styles, and automatic singing skill evaluation.

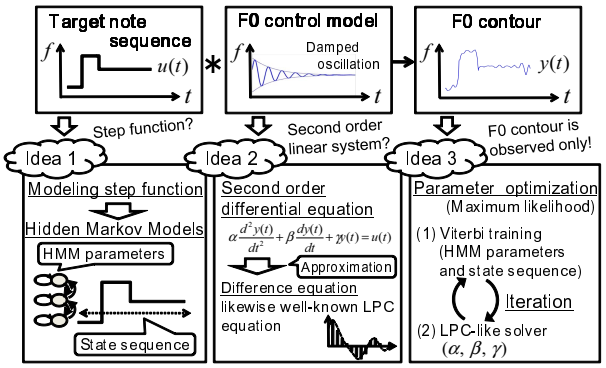

Figure 1: *Schematic view of F0 contour model for singing voices*

Therefore, we propose a novel representation of F0 contours that provides a computationally efficient algorithm for automatically estimating the parameters of the F0 control model. By incorporating into the traditional F0 control model HMM and a difference approximation of the differential equation, we can optimally estimate model parameters from observed F0 contours. The rest of this paper consists of the following sections. Section 2 presents the F0 control model for singing voices. Section 3 describes the maximum likelihood estimation for model parameters. Section 4 deals with the experimental evaluations of the proposed algorithms, and Section 5 is with the conclusions and future directions.

2. F0 control model for singing voices

Figure 1 shows a schematic view of our proposed F0 control model. The following are its key ideas:

Idea 1 Target note sequence $u(t)$ is modeled by HMM.

Idea 2 The differential equation of Eq. (1) is approximated by a difference equation, that enables the use of an LPC-like solver to estimate of α, β, and γ.

Idea 3 The parameters of each phase are optimized by the iteration of the following two steps based on maximum likelihood estimation.

Step 1 α, β, and γ are updated by solving the difference equation in Eq. (3).

Step 2 The state sequence and HMM parameters are updated using Viterbi training.

2.1. Difference approximation of differential equations

We assume that the F0 contour is governed by a second-order linear system. The first and second order differentials in Eq. (1) are approximated by

$$\frac{d\mathbf{y}}{dt} \simeq \mathbf{A}\mathbf{y}, \qquad \frac{d^2\mathbf{y}}{dt^2} \simeq \mathbf{B}\mathbf{y}, \tag{2}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N)^{\mathrm{T}}$ denotes the observed F0 contour and $\mathbf{A}$ and $\mathbf{B}$ are $N \times N$ differential operator matrices. Therefore, we can approximate the differential equation by a linear matrix equation:

$$(\alpha\mathbf{B} + \beta\mathbf{A} + \gamma\mathbf{I})\mathbf{y} = \mathbf{u}, \tag{3}$$

where $\mathbf{u} = (u_1, u_2, \ldots, u_N)^{\mathrm{T}}$ denotes the model input and $\mathbf{I}$ is the $N \times N$ unit matrix.

2.2. Modeling target note sequence

Target note sequence u_n is modeled by HMM which consists of a set of I states $\{S_1, \ldots, S_I\}$. This model generates a sample point for a state transition, as shown in Fig. 2:

$$u_n = m_{q_n}, \ \ (m_{S_i} \in \mathbb{R}, q_n \in \{S_1, \ldots, S_I\}), \tag{4}$$

where m_{S_i} denotes the mean of the Gaussian in state S_i. Using $\mathbf{u} = (u_1, \ldots, u_N)^{\mathrm{T}}$ and $\mathbf{m} = (m_{q_1}, \ldots, m_{q_N})^{\mathrm{T}}$, the model input is represented as $\mathbf{u} = \mathbf{m}$. We assume this model is a homogeneous Markov chain, and then represent the transition probability from states S_j to S_i as $\mathbb{P}(S_i|S_j)$. Here, transition probability $\mathbb{P}(S_i|S_j)$ is a constant value.

2.3. Interpretation of model parameters

From Sections 2.1 and 2.2, the set of model parameters is thus $\mathbf{\Theta} \triangleq \{\alpha, \beta, \gamma, q_1, \ldots, q_N, m_{S_1}, \ldots, m_{S_I}\}$. Coefficients α, β, and γ of the differential equation in Eq. (3) represent the dynamics of various singing behaviors. By state sequence $q_1, \ldots, q_N$ and means $m_{S_1}, \ldots, m_{S_I}$ of the Gaussians in HMM states, target note sequence $\mathbf{u}$ is represented. Here, $q_1, \ldots, q_N$ and $m_{S_1}, \ldots, m_{S_I}$ can represent the inter-onset interval and the pitch corresponding to each note in the sung melodic contour, respectively. Therefore, the target note sequence does not correspond to the original musical note sequence described in the musical score.

3. Maximum likelihood estimation for a F0 control model

Given data set $\mathbf{y} = (y_1, \ldots, y_N)^{\mathrm{T}}$, we estimate model parameters $\mathbf{\Theta}$ using maximum likelihood. If $\mathbf{y}$ follows the difference equation ideally, the parameters should be solved directly with Eq. (3). However, in fact, considering the error between the estimated target note sequence $\hat{\mathbf{u}} = \mathbf{W}\mathbf{y}$ and $\mathbf{u}$, we define

$$\hat{\mathbf{u}} - \mathbf{u} = \mathbf{W}\mathbf{y} - \mathbf{m} = \boldsymbol{\epsilon}, \qquad (\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})), \tag{5}$$

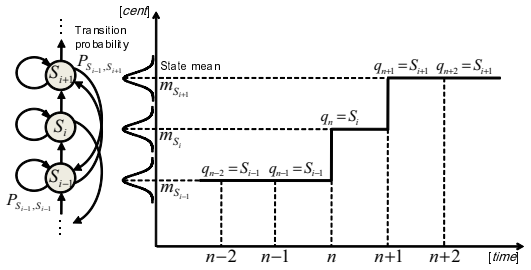

Figure 2: *Target note sequence modeled by HMM*

where $\mathbf{W} \equiv \alpha\mathbf{B} + \beta\mathbf{A} + \gamma\mathbf{I}$, and $\boldsymbol{\epsilon}$ is assumed to be an i.i.d. zero-mean Gaussian white noise. The log likelihood function is therefore given by

$$\begin{aligned} L(\mathbf{\Theta}) &\triangleq \log \mathbb{P}(\mathbf{y}|\mathbf{\Theta}) \\ &= -\frac{N}{2}\log(2\pi) - \log \sigma^N |\mathbf{W}^{-1}| - \frac{1}{2\sigma^2}(\mathbf{W}\mathbf{y} - \mathbf{m})^{\mathrm{T}}(\mathbf{W}\mathbf{y} - \mathbf{m}). \end{aligned} \tag{6}$$

The posterior probability of parameters $\mathbf{\Theta}$ is then given by

$$\mathbb{P}(\mathbf{\Theta}|\mathbf{y}) \propto \mathbb{P}(\mathbf{y}|\mathbf{\Theta})\mathbb{P}(\mathbf{\Theta}). \tag{7}$$

Defining $U(\mathbf{\Theta}) \triangleq \mathbb{P}(\mathbf{y}|\mathbf{\Theta})\mathbb{P}(\mathbf{\Theta})$,

$$\begin{aligned} \log U(\mathbf{\Theta}) &= L(\mathbf{\Theta}) + \log \mathbb{P}(\alpha, \beta, \gamma) \\ &\quad + \log \mathbb{P}(m_{S_1}, \ldots, m_{S_I}) + \log \mathbb{P}(q_1, \ldots, q_N), \end{aligned} \tag{8}$$

where α, β, γ, and m are assumed to follow uniform distributions. Since $\mathbb{P}(q_1, \ldots, q_N)$ is assumed to be a homogeneous Markov chain, log prior probability $\mathbb{P}(q_1, \ldots, q_N)$ is given by

$$\log \mathbb{P}(q_1, q_2, \ldots, q_N) = \log \mathbb{P}(q_1)\mathbb{P}(q_2|q_1) \ldots \mathbb{P}(q_N|q_{N-1}). \tag{9}$$

Transition probability $\mathbb{P}(S_i|S_j)$ is a constant value determined in advance. Here, we define $P_{S_i,S_j} \equiv \log \mathbb{P}(S_i|S_j)$ briefly. Hence, Eq. (9) is described as

$$\log \mathbb{P}(q_1)\mathbb{P}(q_2|q_1) \ldots \mathbb{P}(q_N|q_{N-1}) = P_{q_1,q_0} + \sum_{n=2}^{N} P_{q_n,q_{n-1}}. \tag{10}$$

Substituting Eqs. (6) and (10) into Eq. (8), objective function J is defined as

$$\begin{aligned} J \equiv &- N\log\sigma + \log|\mathbf{W}| \\ &- \frac{1}{2\sigma^2}(\mathbf{W}\mathbf{y} - \mathbf{m})^{\mathrm{T}}(\mathbf{W}\mathbf{y} - \mathbf{m}) + P_{q_1,q_0} + \sum_{n=2}^{N} P_{q_n,q_{n-1}}. \end{aligned} \tag{11}$$

However, there is no closed-form solution for maximizing J with respect to $\mathbf{\Theta}$. We therefore find it most convenient to perform the following steps iteratively until J reaches convergence.

3.1. Update of α, β, and γ

We maximize J with respect to α, β, and γ while keeping $q_1, \ldots, q_N$, and $m_{S_1}, \ldots, m_{S_I}$ fixed. In particular cases where $\mathbf{A}$ and $\mathbf{B}$ are both triangular matrices whose entries on the main diagonal are 1, $\log|\mathbf{W}| = $ const. holds if α, β, and γ are constrained to $\alpha + \beta + \gamma = $ const. Therefore, neglecting term

$\log|\mathbf{W}|$ may not be a highly unrealistic assumption. The partial derivative of J with respect to α is given by

$$\frac{\partial J}{\partial \alpha} = \mathrm{Tr}\Big(\frac{\partial J}{\partial \mathbf{W}}\frac{\partial \mathbf{W}^{\mathrm{T}}}{\partial \alpha}\Big) = \mathrm{Tr}\Big((\mathbf{W}\mathbf{y}-\mathbf{m})\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\Big), \quad (12)$$

where $\mathrm{Tr}(\mathbf{X})$ denotes the sum of the diagonal elements of matrix $\mathbf{X}$. Setting this derivative to zero, one obtains

$$\alpha\mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) + \beta\mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) + \gamma\mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) = \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}). \quad (13)$$

Similarly, differentiating partially with respect to β and γ and setting these to zero, one obtain Eq. (14) and Eq. (15)

$$\alpha\mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) + \beta\mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) + \gamma\mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) = \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}), \quad (14)$$

$$\alpha\mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}) + \beta\mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}) + \gamma\mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}) = \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}). \quad (15)$$

Therefore, the normal equation is obtained by

$$\begin{bmatrix} \mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}) \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}) \end{bmatrix}. \quad (16)$$

The updating values of α, β, and γ are given by

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{B}\mathbf{y}\mathbf{y}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{A}\mathbf{y}\mathbf{y}^{\mathrm{T}}) & \mathrm{Tr}(\mathbf{y}\mathbf{y}^{\mathrm{T}}) \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}) \\ \mathrm{Tr}(\mathbf{m}\mathbf{y}^{\mathrm{T}}) \end{bmatrix}. \quad (17)$$

The estimate of variance σ^2 is given by

$$\sigma^2 = \frac{1}{N}(\mathbf{W}\mathbf{y}-\mathbf{m})^{\mathrm{T}}(\mathbf{W}\mathbf{y}-\mathbf{m}). \quad (18)$$

3.2. Update of $q_1, \ldots, q_N$

We maximize J with respect to $q_1, \ldots, q_N$ while keeping α, β, γ, and $m_{S_1}, \ldots, m_{S_I}$ fixed. In Eq. (11), we define $\hat{u}_n \equiv (\mathbf{W}\mathbf{y})_n$ and effectively solve this problem using a Viterbi algorithm (Dynamic Programming). First, we define quantity $\delta_k(S_i)$ based on the optimal state sequence up to time k and state S_i:

$$\delta_k(S_i) \equiv \max_{q_1,\ldots,q_{k-1}} \Big[-\frac{1}{2\sigma^2}\sum_{n=1}^{k-1}\big(\hat{u}_n - m_{q_n}\big)^2 - \frac{1}{2\sigma^2}\big(\hat{u}_k - m_{S_i}\big)^2 + P_{q_1,q_0} + \sum_{n=2}^{k-1} P_{q_n,q_{n-1}} + P_{S_i,q_{k-1}}\Big]. \quad (19)$$

Using a recurrence equation, we obtain

$$\delta_k(S_i) = \max_{S_j}\big[\delta_{k-1}(S_j) - \frac{1}{2\sigma^2}\big(\hat{u}_k - m_{S_i}\big)^2 + P_{S_i,S_j}\big]. \quad (20)$$

Once we have completed the final maximization over $k = N$, we can find the most probable state sequence $q_1, \ldots, q_N$ using a back-tracking procedure.

3.3. Update of $m_{S_1}, \ldots, m_{S_I}$

We maximize J with respect to $m_{S_1}, \ldots, m_{S_I}$ while keeping α, β, γ, and $q_1, \ldots, q_N$ fixed. Differentiating partially with respect to m_{S_i} and setting this to zero, we obtain

$$m_{S_i} = \frac{1}{|\mathcal{C}_i|}\sum_{n\in\mathcal{C}_i}\hat{u}_n, \quad (21)$$

where we define set $\mathcal{C}_i = \{n | q_n = S_i\}$ and $|\mathcal{C}_i|$ denotes the element count. This procedure is often called Viterbi training, because the parameters for each state in HMM are learned using the optimal state sequence obtained by the Viterbi algorithm. Finally, target note sequence $\mathbf{u}$ is built by $q_1, \ldots, q_N$ and $m_{S_1}, \ldots, m_{S_I}$.

3.4. Initialization

To prevent objective function J from convergence to the local solution, we set up two initializations. In the first initialization, we estimate the initial state sequence from observed data $\mathbf{y}$ using the Viterbi algorithm. We can rewrite objective function J as

$$J_{init} \equiv -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(\mathbf{y}-\mathbf{m})^{\mathrm{T}}(\mathbf{y}-\mathbf{m}) + P_{q_1,q_0} + \sum_{n=2}^{N} P_{q_n,q_{n-1}}. \quad (22)$$

We set the state sequence that maximizes J_{init} as initial state sequence $q_1^{init}, \ldots, q_N^{init}$.

In the second initialization, we obtain the mean of the Gaussian in each HMM state using the initial state sequence. Partially differentiating with respect to m_{S_i} and setting this to zero, we obtain

$$m_{S_i}^{init} = \frac{1}{|\mathcal{C}_i^{init}|}\sum_{n\in\mathcal{C}_i^{init}} y_n, \quad (23)$$

where we define set $\mathcal{C}_i^{init} = \{n | q_n^{init} = S_i\}$. We set $m_{S_1}^{init}, \ldots, m_{S_I}^{init}$ as the initial means in each HMM state.

3.5. Generation of F0 contours

Using estimated α, β, γ, $q_1, \ldots, q_N$, and $m_{S_1}, \ldots, m_{S_I}$, we generate the F0 contour. From Eq. (5), generated F0 contour $\hat{\mathbf{y}}$ is given by

$$\hat{\mathbf{y}} = \mathbf{W}^{-1}\mathbf{u}. \quad (24)$$

If $\mathbf{W}$ is a singular matrix, we may calculate $\mathbf{W}^{-1}$ after adding small values to the diagonal elements of $\mathbf{W}$.

4. Evaluation

In this section, we present preliminary experimental evaluation using the proposed method. We estimated from the observed F0 contour $\mathbf{y}$ the target note sequence $\mathbf{u}$ and the dynamics of singing behaviors represented by α, β, and γ. We then generated F0 contour $\hat{\mathbf{y}}$ using these estimated parameters.

4.1. Singing voice database

Firstly, we built a database consisting of 24 sound samples of singing voices recorded from three pairs of subjects: professional male and female classical music singers; professional male and female singers of popular songs; and amateur male and female singers. Without musical accompaniment, each subject sang songs with Japanese lyrics and hummed while listening to the melody (guide tones) with headphones. The songs were "*Twinkle, twinkle, little star*", and "*Ode to Joy*".

4.2. Experimental conditions

The F0 contour was estimated every 10 ms using YIN [6] and represented in cents, so that one equal-tempered semitone corresponded to 100 cents. Differential operators of Eq. (2) are given by

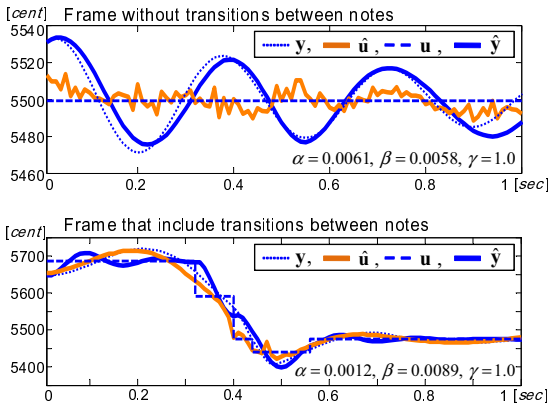

Figure 3: *Estimation results of model parameters ($N = 1$ s)*

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{2\Delta} & 0 & \dots & 0 \\ -\frac{1}{2\Delta} & 0 & \frac{1}{2\Delta} & \dots & 0 \\ 0 & -\frac{1}{2\Delta} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -\frac{2}{\Delta^2} & \frac{1}{\Delta^2} & 0 & \dots & 0 \\ \frac{1}{\Delta^2} & -\frac{2}{\Delta^2} & \frac{1}{\Delta^2} & \dots & 0 \\ 0 & \frac{1}{\Delta^2} & -\frac{2}{\Delta^2} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -\frac{2}{\Delta^2} \end{bmatrix}, \tag{25}$$

where Δ denotes the sampling period of 10 ms. Initial HMM states are located every 100 cents. The initial value of σ^2 is 2500. We estimated model parameters in every frame of length N. Representing the F0 contour in frame f as $\mathbf{y}^{(f)}$, parameters $\alpha^{(f)}$, $\beta^{(f)}$, $\gamma^{(f)}$, and $\mathbf{u}^{(f)}$ were then estimated using the proposed method. F0 contour $\hat{\mathbf{y}}^{(f)}$ is generated by Eq. (24). Here, the F0 contour of the guide tones in frame f is represented as $\mathbf{g}^{(f)}$. In this experiment, we consider $\mathbf{g}^{(f)}$ as the actual target note sequence. The frame is shifted by 100 ms.

4.3. Evaluation measure

We used the root mean square (RMS) to evaluate the estimation accuracy of $\mathbf{u}^{(f)}$ and $\hat{\mathbf{y}}^{(f)}$ of each frame:

$$\mathrm{RMS}_{\mathbf{u}^{(f)}} = \frac{1}{N}||\mathbf{u}^{(f)} - \mathbf{g}^{(f)}||_2, \ \mathrm{RMS}_{\hat{\mathbf{y}}^{(f)}} = \frac{1}{N}||\hat{\mathbf{y}}^{(f)} - \mathbf{y}^{(f)}||_2, \tag{26}$$

where $||\cdot||_2$ denotes the Euclidean norm. We then defined the percentage of correctly estimated frames as:

$$\text{Percentage of correctly estimated frames} = \frac{\text{Frames which } \mathrm{RMS}_{\mathbf{u}^{(f)}} \text{ AND } \mathrm{RMS}_{\hat{\mathbf{y}}^{(f)}} \leq 50\,\text{cent}}{\text{Total frames}} \times 100. \tag{27}$$

4.4. Experimental results and discussions

Fig. 3 shows two estimation results. Frame length N was set to 1 s. For frames without transitions between notes, model parameters could be estimated appropriately. However, due to its steplike nature, estimating $\mathbf{u}^{(f)}$ was difficult for frames including transitions. Fig. 4 shows the estimation performance. The top figure shows the percentage of correctly estimated frames for different values of N. The middle and bottom figures show the percentage of correctly estimated frames for different singers and songs, respectively, while N was set to 1 s. For frames without note transitions, the percentage of correctly estimated frames improves as N decreases. On the other hand, for frames that include note transitions, this percentage improves as N increases. However, estimation performance for these frames is poor. Percentages of correctly estimated frames vary with different types of singers and singing styles. Especially estimation performance of humming is better than that of singing because

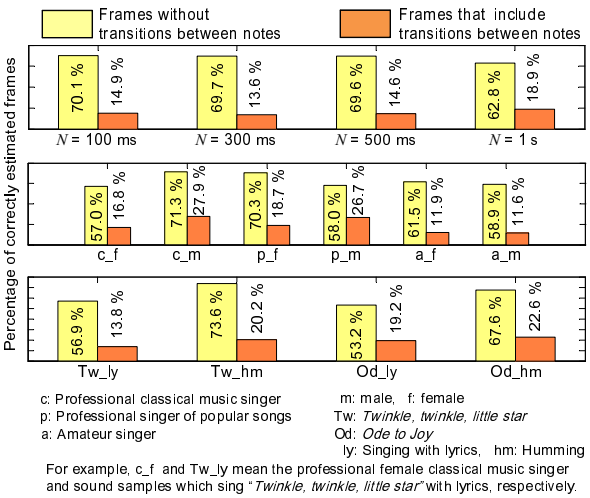

Figure 4: *Estimation performance based on RMS. Frame length N was set to 100 ms, 300 ms, 500 ms, and 1 s in top. Middle and bottom figure show estimation performance for different singers and songs, respectively.*

the F0 contour of singing voices with lyrics includes various fluctuations of vowels and consonants during phonation. As a future work, we still have to improve our method to cope with various fluctuations which cannot be captured by a second-order system.

5. Conclusion

In this paper, we discussed a parameter optimization method of the F0 control model for singing voices. Our proposed method estimates model parameters with 70% accuracy for frames without note transitions. However, correct estimation is difficult for frames that do include transitions between notes. Therefore, we plan to estimate them while automatically tuning the frame length and to estimate the auto regressive model parameters instead of proposed parameters to cope with various fluctuations. Future work will evaluate our model's ability to automatically detect particular singing behaviors such as vibrato, overshoot, and singing voice synthesis that reflect personal singing behaviors.

6. Acknowledgements

The authors would like to thank Dr. Masataka Goto at AIST for his valuable feedback and suggestions.

7. References

[1] M. Grachten *et al.*, "Melodic similarity: Looking for a good abstraction," in *Proc. ISMIR*, 2004.

[2] R. B. Dannenberg, W. P. Birmingham, *et al.*, "A comparative evaluation of search techniques for query-by-humming using the musart testbed," *JASIST*, vol. 58, no. 5, pp. 687–701, 2007.

[3] Y. Ohishi *et al.*, "A stochastic representation of the dynamics of sung melody," in *Proc. ISMIR*, 2007.

[4] T. Saitou, M. Unoki, and M. Akagi, "Development of an F0 control model based on F0 dynamic characteristics for singing-voice synthesis," *Speech Communication*, vol. 46, pp. 405–417, 2005.

[5] J. Bonada *et al.*, "Synthesis of the singing voice by performance sampling and spectral models," *IEEE Signal Processing Magazine*, vol. 24, pp. 67–79, 2007.

[6] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *JASA*, vol. 111, no. 4, pp. 1917–1930, 2002.