Loss Function Considering Multiple Attributes of a Temporal Sequence for Feed-Forward Neural Networks

Noriyuki MATSUNAGA†,††, Yamato OHTANI†, and Tatsuya HIRAHARA††, Members

SUMMARY Deep neural network (DNN)-based speech synthesis became popular in recent years and is expected to soon be widely used in embedded devices and environments with limited computing resources. The key intention of these systems in poor computing environments is to reduce the computational cost of generating speech parameter sequences while maintaining voice quality. However, reducing computational costs is challenging for two primary conventional DNN-based methods used for modeling speech parameter sequences. In feed-forward neural networks (FFNNs) with maximum likelihood parameter generation (MLPG), the MLPG reconstructs the temporal structure of the speech parameter sequences ignored by FFNNs but requires additional computational cost according to the sequence length. In recurrent neural networks, the recursive structure allows for the generation of speech parameter sequences while considering temporal structures without the MLPG, but increases the computational cost compared to FFNNs. We propose a new approach for DNNs to acquire parameters captured from the temporal structure by backpropagating the errors of multiple attributes of the temporal sequence via the loss function. This method enables FFNNs to generate speech parameter sequences by considering their temporal structure without the MLPG. We generated the fundamental frequency sequence and the mel-cepstrum sequence with our proposed method and conventional methods, and then synthesized and subjectively evaluated the speeches from these sequences. The proposed method enables even FFNNs that work on a frame-by-frame basis to generate speech parameter sequences by considering the temporal structure and to generate sequences perceptually superior to those from the conventional methods.

key words: loss function, multiple attributes of temporal sequence, feed-forward neural networks, fundamental frequency, mel-cepstrum

1. Introduction

Deep neural networks (DNNs) have been widely used in the field of speech synthesis since 2013 [1]. Today’s computers with high-performance CPUs, GPUs, or customized processing units can handle the high computational costs and complex structures of DNNs [2], [3]. However, the computational cost of a DNN is a limiting issue for poor computing environments that results in long computation times and slow responses from the speech synthesis system, which degrades usability. The performance of CPUs in embedded devices is lower than that of general-purpose PCs because of low-cost manufacturing and power saving. The computational resources available for one process are limited in applications where multiple processes, including a speech synthesis process, operate simultaneously in order to stabilize each process. Therefore, reducing the computational costs of a DNN-based speech synthesis system is of great value for deployments to poor computing environments.

Conventional speech waveform generation based on a vocoder [4], [5] is still often used due to computational limitations in poor computing environments, and high-quality speech waveform generation methods based on DNNs are even available [6]–[8]. The synthesized speech quality from the vocoder depends on the quality of the speech parameters so that the prediction performance of DNNs generating these parameters is paramount. The first primary DNN-based method for modeling these speech parameter sequences is the feed-forward neural network (FFNN) with maximum likelihood parameter generation (MLPG) [1], [9]. FFNNs generate static and dynamic features of the speech parameters and the MLPG generates speech parameter sequences by maximizing the likelihoods for these static and dynamic features over an entire utterance. Then, the MLPG requires a computational cost according to the utterance length. Because the MLPG relates the speech parameters in adjacent frames, this component of the method cannot be eliminated to reduce the overall computational cost.

The second primary DNN-based method is the recurrent neural network (RNN) that performs recursive processing while retaining previous information. RNNs automatically acquire the relationships between speech parameters of adjacent frames by the recursive structure from the training phase. Therefore, the system designed with RNNs does not require MLPG. However, these recursive structures increase computational costs, such as in a popular and high performing type of RNN, the long short-term memory RNN (LSTM-RNN) [10] that requires a high computational cost due to its use of complex recursive structures.

FFNNs are basic DNNs with simplified structures that reduce computational cost. Although FFNNs are suitable for poor computing environments, they ignore the relationships between speech parameters in adjacent frames because they only function frame-by-frame. This limitation can be overcome by improving the prediction performance of FFNNs for which neither MLPG nor recursive structures are required. Then, FFNNs alone can generate speech parameter sequences with a natural temporal structure.

In this paper, we propose a new training method that enables FFNNs to acquire parameters captured with a natural temporal structure for these sequences by
backpropagating the errors of multiple attributes from the temporal sequence through the loss function. As a preliminary study, the effectiveness of the proposed loss function in modeling the F0 sequence was investigated [11]. In this paper, the effectiveness of the proposed loss function in modeling the spectral feature sequence in addition to the F0 sequence is discussed. The use of the proposed method reduces the computational cost of generating speech parameter sequences while maintaining the quality of these sequences and achieving a low-computational cost speech synthesis system suitable for poor computing environments.

2. A New Loss Function MATS

A loss function plays a key role in DNN training by calculating the error between supervised and generated sequences to update the parameters through backpropagation. Although a general loss function calculates only this error, we propose one that calculates the error and the errors between multiple attributes extracted from these sequences. Because the parameters reflect these errors through backpropagation, the DNN generates speech parameter sequences that considers these multiple attributes. The following describes the training method concept proposed, and we name the loss function Multiple Attributes of Temporal Sequences (MATS) as is specified below.

We define the linguistic feature vector sequence $x = [x^1, \ldots, x^n, \ldots, x^T]^\top$ and the natural speech parameter vector sequence $y = [y^1, \ldots, y^n, \ldots, y^T]^\top$. Then, $\hat{y} = [\hat{y}^1, \ldots, \hat{y}^n, \ldots, \hat{y}^T]$ is a DNN-generated speech parameter sequence where $t$ and $T$ are the frame index and total frame length, respectively, and $x_t = [x_t^{(1)}, \ldots, x_t^{(k)}, \ldots, x_t^{(K)}]$ and $y_t = [y_t^{(1)}, \ldots, y_t^{(d)}, \ldots, y_t^{(D)}]$ are the linguistic and speech parameter vectors at frame $t$, respectively. The terms $k$ and $K$ are the index and length of the linguistic feature vector, respectively, and $d$ and $D$ are the index and length of the speech parameter vector, respectively.

Sequences of the short-term segment, which is the speech parameter vector sequence for several frames, $Y = [Y_1, \ldots, Y_t, \ldots, Y_T]$ and $\hat{Y} = [\hat{Y}_1, \ldots, \hat{Y}_t, \ldots, \hat{Y}_T]$ are formed by slicing $y$ and $\hat{y}$ into a short-term $[t + L, t + R]$ respectively where $Y_t = [y_{t+L}, \ldots, y_{t+L}, \ldots, y_{t+R}]$ is a short-term segment at frame $t$, $L \leq 0$ is a backward lookup frame count, $R \geq 0$ is a forward lookup frame count, and $\tau$ is a short-term lookup frame index ($L \leq \tau \leq R$). In an FFNN, a sequence $\hat{y_t}$ corresponding to $x_t$ is independently predicted regardless of the adjacent frames. We introduce loss functions of the time-domain attribute (TD), local variance (LV) [12], and local covariance (LC) to relate adjacent frames in $Y_t$ that propagate over the long-term frames during the training phase because $Y_t$ and $Y_{t+\tau}$ overlap. Under the MATS loss function, the explicitly define that the short-term relationship between the speech parameters implicitly propagates to the long-term, allowing the FFNN to train temporary sequences similar to LSTM-RNNs.

In addition, the MATS loss function considers the implicit long-term relationship as well as the explicit long-term relationship such as the global variance (GV) [13] and global covariance (GC), and the relationship of the speech parameters between dimensions by introducing a loss function of the dimensional-domain (DD) attribute. The mean squared error between $y$ and $\hat{y}$, called the loss function of direct coupling (DC) in this paper, can be applied.

The MATS loss function is expressed by the weighted summation of these loss functions described above as

$$L(y, \hat{y}) = \sum_i \omega_i L_i(y, \hat{y}),$$

where $i = \{DC, TD, DD, LV, LC, GV, GC\}$ is the loss function identifier and $\omega_i$ is the weight of the loss function with identifier $i$. The primary role of each loss function is as follows. The $L_{TD}$ manages temporal changes of speech parameters instead of the MLPG in the minimum generation error training and the trajectory training [14], [15] by backpropagating the error of the derivative of the adjacent time frame vector. The $L_{LV}$ manages the temporal fluctuation of speech parameters within several frames [16] by backpropagating the variance error within the short-term segment. The $L_{LV}$ manages the scale of the amplitude of the entire speech parameter sequences by backpropagating the variance error of the entire frame. The $L_{LC}$ manages the shape of the distribution of speech parameters in several frames by backpropagating the covariance error within the short-term segment. The $L_{DC}$ manages the distribution shape of the entire speech parameter sequences by backpropagating the covariance error of the entire sequence. The $L_{DD}$ manages the relationship of speech parameter vectors between the dimensions by propagating the error of the feature vector converted from the speech parameter vector. The $L_{DD}$, $L_{DC}$, and $L_{LC}$ are for only multiple dimensional speech parameters. The $L_{TD}$ is same as the $L_{DD}$ when the $L_{TD}$ does not relate speech parameters in adjacent frames.

2.1 Direct Coupling Loss Function

The direct coupling loss function, $L_{DC}(y, \hat{y})$, teaches the DNN the approximate the shape for the speech parameter sequence and is defined as the mean squared error (MSE) of the difference between $y$ and $\hat{y}$ as

$$L_{DC}(y, \hat{y}) = MSE(y, \hat{y}) = \frac{1}{TD} \sum_{t=1}^{T} \sum_{d=1}^{D} \frac{1}{D} \sum_{\tau=1}^{\tau} (y_{t,\tau}^{(d)} - \hat{y}_{t,\tau}^{(d)})^2, \quad (2)$$

The $L_{DC}$ is a limited version of the $L_{TD}$ in the next section and considers only the MSE of the static feature. The $L_{TD}$ is used for considering the MSE of the static and dynamic features. The usage of calculating the MSE of the static and dynamic features by the $L_{DC}$ and $L_{TD}$, respectively, is basically not allowed. The $L_{DC}$ and $L_{TD}$ are defined separately to distinguish between considering and not considering the MSE of the dynamic feature.
2.2 Time-Domain Attribute Loss Function

The TD attribute loss function teaches the DNN the relationship between speech parameters in adjacent frames by extracting static and dynamic features. Here \( Y_{TD} = [Y_{1}^T, Y_{2}^T, \ldots, Y_{M}^T] \) is a sequence in a \( D \times M \) matrix of TD attributes at each short-term \([t + L, t + R]\). The loss function \( L_{TD}(y, \hat{y}) \) is defined as the mean squared error of the difference between \( Y_{TD} \) and \( \hat{Y}_{TD} \) as

\[
L_{TD}(y, \hat{y}) = \text{MSE}(Y_{TD}, \hat{Y}_{TD}) = \frac{1}{T_{TD}} \sum_{t=1}^{T} \sum_{d=1}^{D} \sum_{m=1}^{M} \left( \sum_{\tau=L}^{R} (y_{t+\tau}^d w_{m}^\tau - \hat{y}_{t+\tau}^d w_{m}^\tau) \right)^2,
\]

where \( W = [W_1^T, \ldots, W_m^T, \ldots, W_M^T] \) is a \((L+R+1) \times M\) coefficient matrix that relates adjacent frames in the short-term \([t + L, t + R]\), \( W_m^\tau = [w_{m}^{(0)}, \ldots, w_{m}^{(0)}(\tau), \ldots, w_{m}^{(R)}(\tau)] \) is a coefficient vector, and \( m \) and \( M \) are an index and the total number of coefficient vectors, respectively.

To extract the static and dynamic features, \( W = [W_1^T, W_2^T] \) comprises two coefficient vectors of \( W_1 = [0^{(L)}, \ldots, 0^{(L)}(1), 0^{(1)}(1), \ldots, 0^{(1)}(R)] \) for obtaining the static feature at frame \( t \), and \( W_2 = [0^{(L)}, \ldots, 0^{(L)}(1), -w_{1}^{(0)}(1), 0^{(1)}(1), \ldots, 0^{(1)}(R)] \) for obtaining the delta between frames \( t \) and \( t-1 \). This implementation calculates the static and dynamic features just as an arbitrary generation error training and trajectory training [14, 15], but with a smaller memory. Although both trajectory methods consider the static and dynamic features, the subject of the training loss differs as it is calculated from speech parameters obtained by applying the MLPG to the generated static and dynamic features from the model. On the other hand, in our proposed method, the training loss is calculated from the static and dynamic features extracted directly from the generated speech parameters of the model.

For modeling the fundamental frequency (F0) sequences, \( W_2 \) is introduced because the relative changes in F0 are deeply related to Japanese accent perceptions [17]. Focusing on \( Y_{TD} - \hat{Y}_{TD} \) at frame \( t \) in (3), when the loss is assumed to be zero, the equation can be rearranged into the recurrence formula,

\[
\hat{y}_t = y_t - \frac{w_2}{w_1 + w_2} y_{t-1} + \frac{w_2}{w_1 + w_2} \hat{y}_{t-1}.
\]

Because the loss function considers (4), setting the value of \( W \), as described above, enables recursive training as with an RNN. In addition, the strength of recursion can be controlled by adjusting the values of \( w_1 \) and \( w_2 \). For example, \( \hat{y}_t \) is determined by the immediately preceding generated speech parameter \( \hat{y}_{t-1} \) and the difference of the natural speech parameter \( y_t - y_{t-1} \) when the value of \( w_2 \) is sufficiently larger than that of \( w_1 \).

2.3 Dimensional-Domain Attribute Loss Function

The DD attribute loss function teaches the DNN the relationships between the speech parameters for each dimension by transforming the generated speech parameter into another. Here, \( Y_{DD} = [y_1^T, y_2^T, \ldots, y_m^T, \ldots, y_N^T] \) is a sequence of DD attributes, and the loss function \( L_{DD}(y, \hat{y}) \) is defined as the mean squared error of the difference between \( Y_{DD} \) and \( \hat{Y}_{DD} \) as

\[
L_{DD}(y, \hat{y}) = \text{MSE}(Y_{DD}, \hat{Y}_{DD}) = \frac{1}{T_{N}} \sum_{t=1}^{T} \sum_{n=1}^{N} \left( \sum_{d=1}^{D} (y_{t}^d w_{n}^d - \hat{y}_{t}^d w_{n}^d) \right)^2,
\]

where \( \hat{W} = \left[ \hat{W}_1^T, \hat{W}_2^T \right] \) is a \((L+R+1) \times N\) coefficient matrix that relates the speech parameters from each dimension, \( \hat{W}_n = [w_{n}^{(0)}, \ldots, w_{n}^{(0)}(1), \ldots, w_{n}^{(1)}(1), \ldots, w_{n}^{(1)}(R)] \) is \( n \)th coefficient vector, and \( n \) and \( N \) are an index of the coefficient vector and the number of coefficient vectors, respectively.

The coefficient matrix \( \hat{W} \) varies according to the modeled speech parameters. For example, in the modeling of a mel-cepstrum (mcep), the coefficient matrix \( \hat{W} \) represents the frequency transformation formula, and for a spectrum, the coefficient matrix \( \hat{W} \) represents the mel-filter bank.

2.4 Local Variance Loss Function

The LV loss function teaches the DNN a variation and amplitude scale for the speech parameters in the short-term [12]. Here, \( Y_{LV} = [v_1^T, \ldots, v_L^T] \) is a sequence of variances in the short-term \([t + L, t + R]\), and the loss function \( L_{LV}(y, \hat{y}) \) is defined as the mean absolute error (MAE) of the difference between \( Y_{LV} \) and \( \hat{Y}_{LV} \) as

\[
L_{LV}(y, \hat{y}) = \text{MAE}(Y_{LV}, \hat{Y}_{LV}) = \frac{1}{T_{D}} \sum_{t=1}^{T} \sum_{d=1}^{D} |v_{t}^d - \hat{v}_{t}^d|.
\]

The term \( \hat{v}_t = [\hat{v}_t^{(1)}, \ldots, \hat{v}_t^{(D)}] \) is a \( D \)-dimensional variance vector at frame \( t \), and \( v_t^d \) is the \( d \)th variance at frame \( t \) given as

\[
\hat{v}_t^d = \frac{1}{R + L + 1} \sum_{\tau=L}^{R} (y_{t+\tau}^d - \bar{y}_t^d)^2,
\]

where \( \bar{y}_t^d \) is the \( d \)th mean at frame \( t \) expressed as

\[
\bar{y}_t^d = \frac{1}{R + L + 1} \sum_{\tau=L}^{R} y_{t+\tau}^d.
\]

2.5 Local Covariance Loss Function

The LC loss function teaches the DNN a correlation between the speech parameters for each dimension in the
short-term. Here, \( Y_{LC} = [c_1, \cdots, c_t, \cdots, c_T] \) is a sequence of covariances in the short-term \([t + L, t + R]\), and the loss function \( L_{LC}(y, \tilde{y}) \) is defined as the mean absolute error of the difference between \( Y_{LV} \) and \( \tilde{Y}_{LV} \) as

\[
L_{LC}(y, \tilde{y}) = \text{MAE}(Y_{LV}, \tilde{Y}_{LV}) = \frac{1}{TD^2} \sum_{t=1}^{T} \sum_{d_1=1}^{D} \sum_{d_2=1}^{D} |c_t^{(d_1,d_2)} - \bar{c}_t^{(d_1,d_2)}|.
\] (9)

Here, \( c_t \) is a \( D \times D \) covariance matrix at frame \( t \) given as

\[
c_t = \begin{bmatrix} c_t^{(d_1,d_2)} \end{bmatrix}
= \frac{1}{-L + R + 1} \left( Y_t - \bar{Y}_t \right)^\top \left( Y_t - \bar{Y}_t \right),
\] (10)

where \( c_t^{(d_1,d_2)} \) is an element of \( d_1 \)-th row and \( d_2 \)-th column of \( c_t \), \( \bar{Y}_t \) is a mean vector in the short-term \([t + L, t + R]\), and \( Y_t - \bar{Y}_t = [y_{t+L}, \cdots, y_{t+R} - \bar{Y}_t, \cdots, y_T - \bar{Y}_t] \) is the broadcast operation.

2.6 Global Variance Loss Function

The GV loss function teaches the DNN a variation and amplitude scale of the entire speech parameter sequence [13]. Here, \( Y_{GV} = [V^{(1)}, \cdots, V^{(d)}, \cdots, V^{(D)}] \) is a variance vector of the entire speech parameter sequence, and the loss function \( L_{GV}(y, \tilde{y}) \) is defined as the mean absolute error of the difference between \( Y_{GV} \) and \( \tilde{Y}_{GV} \) as

\[
L_{GV}(y, \tilde{y}) = \text{MAE}(Y_{GV}, \tilde{Y}_{GV}) = \frac{1}{D} \sum_{d=1}^{D} |V^{(d)} - \bar{V}^{(d)}|.
\] (11)

where \( V_d \) is the \( d \)-th variance given as

\[
V_d = \frac{1}{T} \sum_{t=1}^{T} \left( y_t^{(d)} - \bar{y}^{(d)} \right)^2,
\] (12)

and \( \bar{y}^{(d)} \) is the \( d \)-th mean given as

\[
\bar{y}^{(d)} = \frac{1}{T} \sum_{t=1}^{T} y_t^{(d)}.
\] (13)

2.7 Global Covariance Loss Function

The GC loss function teaches the DNN a correlation between the speech parameters for each dimension in the entirety of the sequence. Here, \( Y_{GC} \) is a \( D \times D \) covariance matrix of the complete speech parameter sequence, and the loss function \( L_{GC}(y, \tilde{y}) \) is defined as the mean absolute error of the difference between \( Y_{GC} \) and \( \tilde{Y}_{GC} \) as

\[
L_{GC}(y, \tilde{y}) = \text{MAE}(Y_{GC}, \tilde{Y}_{GC}) = \frac{1}{D^2} \sum_{d_1=1}^{D} \sum_{d_2=1}^{D} |C^{(d_1,d_2)} - \bar{C}^{(d_1,d_2)}|,
\] (14)

where \( Y_{GC} \) is

\[
Y_{GC} = \begin{bmatrix} C^{(d_1,d_2)} \end{bmatrix} = \frac{1}{T} (y - \bar{y})^\top (y - \bar{y}),
\] (15)

\( C^{(d_1,d_2)} \) is an element of \( d_1 \)-th row and \( d_2 \)-th column of \( Y_{GC} \), \( \bar{y} = [\bar{y}^{(1)}, \cdots, \bar{y}^{(d)}, \cdots, \bar{y}^{(D)}] \) is the \( D \)-dimensional mean vector, and \( y - \bar{y} = [y_1 - \bar{y}, \cdots, y_t - \bar{y}, \cdots, y_T - \bar{y}] \) is the broadcast operation.

3. Experimental Conditions

Japanese speech data obtained from a female professional speaker were curated into training and test data sets comprising of 2,000 and 100 utterances, respectively. From this data and corresponding transcriptions, phonetic alignments were manually refined by skilled labelers following automatic generation. The frame-level linguistic features for the DNN included 283 linguistic contexts based on the Japanese instantiation of the HMM/DNN-based speech synthesis system (e.g., the place of articulation, manner of articulation, number of morae in an accent phrase, and frame in the current phoneme) [18]. The linguistic features were normalized in advance with the robust normalization method to remove outliers [19].

Logarithmic F0 values were extracted every 5 ms from the speech data sampled at 48 kHz and 16 bits using the WORLD system [5]. To model the log F0 sequences, silent and unvoiced frames were interpolated. The mcep values were also extracted using this system with a frequency parameter of 0.55 and a dimension of 60 every 5 ms from the spectrum of the speech data sampled at 48 kHz and 16 bits.

Table 1 lists the architectural features of the DNNs used in the experiment, which intentionally does not include a generative adversarial network for comparing the prediction performance of generator networks [22]. The number of parameters of each DNN was nearly the same as in FFNN MATS, and the log F0 and mcep were modeled individually. Each DNN was trained by randomly selecting the training data that had the silent frames removed over 20 epochs, and the mini-batch size was the utterance length of each training sample.

The FFNN MATS system consists only of an FFNN with the MATS loss function. Post-processing is not required because this loss function boosts the FFNN’s prediction performance. Optimum settings for the MATS loss function are determined heuristically by preliminary experiments after the effects of each loss function are checked independently. The \( \mathbf{W} \) of the \( L_{DD} \) is \( 60 \times 1025 \) matrix and represents the frequency transformation formula \( \text{(freqt)} \) in the Speech Signal Processing Toolkit [23]. The parameter of the \( \text{freqt} \) are \( m_1 = 59, m_2 = 1024, a_1 = 0.55 \) and \( a_2 = 0 \).

In this experiment, log F0 and mcep were normalized to z-scores based on these global means and variances calculated from the training data set.

The FFNN MSE system consists of an FFNN with the
and the cepstrum enhancement post-filter [24], representing mean squared error (MSE) criterion loss function, MLPG, and the cepstrum enhancement post-filter [24], representing the most basic configuration [1]. The static and dynamic speech parameter sequences, modeled by FFNN and MLPG, were applied to the predicted feature sequences. The static and dynamic features of log F0 and mcep were normalized to the z-scores based on these global means and variances calculated from the training data set. The variances of the static and dynamic features used in MLPG were calculated from the training data set and remained the same in all frames. The mcep coefficients were enhanced by a post-filter with a factor of 1.4 after the MLPG. The cepstrum enhancement post-filter is implemented in the merlin and the demo script of the HTS, and the factor value is the same as those default value [18], [24], [25].

The FFNN MGE system consists of an FFNN with the minimum generation error (MGE) criterion [14], MLPG, and the cepstrum enhancement post-filter [24], which is an advanced version of the FFNN MSE system. Log F0 and mcep were normalized to the z-scores based on these global means and variances calculated from the training data set. The variances of the static and dynamic features used in MGE and MLPG were calculated from the speech parameters normalized to the z-score from the training data set and remained the same in all frames. A post-filter enhanced the mcep coefficients with a factor of 1.4 after the MLPG.

The LSTM MSE system consists of an LSTM-RNN with an MSE criterion loss function and cepstrum enhancement post-filter [10], [24]. The LSTM-RNN directly models the speech parameter sequences without the MLPG because its recurrent structure considers the temporal structure of the sequence. The timestep is set to the length of the utterance. In this experiment, log F0 and mcep were normalized to the z-scores based on these global means and variances calculated from the training data set, and the mcep coefficients were enhanced by a post-filter with a factor of 1.4 after the MLPG.

One hundred utterances not included in the training data set were used for evaluation. To objectively evaluate the predicted log F0 sequences, the absolute errors for each frame of the sequences $E_{DC}$, the square root of the global variance $E_{GV}$, and the modulation spectrum $E_{MS}$ were calculated after converting from a log scale to a linear scale and drawn using box plots [26]. The $E_{MS}$ of the log F0 sequence serves as an index for the roughness of the sequence as the high-frequency components increase with the occurrence of discontinuities in the sequence.

To objectively evaluate the predicted mcep sequences, $E_{DC}$, $E_{GV}$, and $E_{MS}$ for all dimensions were calculated and drawn using box plots. The $E_{MS}$ of the mcep sequence serves as an index for the complexity of the sequence as high-frequency components decrease with the occurrence of oversmoothing.

Ten practiced listeners participated in a multi-stimulus listening test using the hidden reference and anchor (MUSHRA) method [27] to evaluate the naturalness of the synthesized speeches. Six groups of synthesized speeches were used for the subjective evaluation of the predicted log F0 sequences. Of these, four were evaluation groups consisting of synthesized speeches generated from the predicted F0 from each system with the spectra, aperiodicity ratios, and durations matching those of the natural speech. Another group was the reference consisting of the re-synthesized natural speeches. The final group was the anchor consisting of synthesized speeches generated from the predicted log F0 by the FFNN MSE system without MLPG and with the spectra, aperiodicity ratios, and durations matching those of the natural speech. The listeners were instructed to focus on the accent and intonation when observing the stimuli, and feedback was not provided.

Similarly, the six groups of synthesized speeches were used for the subjective evaluation of the predicted mcep sequences. The four groups consisting of the synthesized speeches generated from the predicted mcep from each system with the F0, aperiodicity ratios, and durations matching those of natural speech were used for evaluation, and another was the reference group consisting of re-synthesized natural speeches. The final group was the anchor consisting of synthesized speeches generated by the predicted mcep from the FFNN MSE system without the cepstrum enhancement post-filter with the F0s, aperiodicity ratios, and durations matching those of the natural speech. The listeners were instructed to focus on the individuality of the voice color and spatial expanse while listening to the stimuli, and feedback was not provided.
4. Experimental Results

4.1 Fundamental Frequency

The loss functions available for training log F0 sequences are the $L_{DC}$, $L_{TD}$, $L_{LV}$ and $L_{GV}$. In order to check the behavior of these loss functions in modeling a log F0 sequence, preliminary experiments were conducted with about 40 combinations of the parameters of these loss functions. Based on the preliminary experimental results, the $L_{TD}$, $L_{LV}$ and $L_{GV}$ were determined to be necessary for modeling a log F0 sequence. Figure A.1 in the Appendix shows a part of the preliminary experimental results. In order to improve the discontinuity of the log F0 sequence mainly, the parameters were adjusted so that the $L_{TD}$ worked as main and the $L_{LV}$ and $L_{GV}$ worked as auxiliary. Finally, the parameters were $\omega_{TD} = 1$, $w_{1} = 1$, $w_{2} = 20$, $\omega_{GV} = 1$, $\omega_{LV} = 2$, $L = -8$, and $R = 8$. The F0 sequence was jagged when only the $L_{DC}$ was used. To suppress this jaggedness and smooth the log F0 sequence, the loss of the dynamic feature was reduced by the $L_{TD}$. First, the loss of the static feature was minimized by the $L_{DC}$, then the loss of the dynamic feature was minimized by the $L_{TD}$. However, the log F0 sequence did not become sufficiently smooth. When the $L_{DC}$ was not used and the losses of the static and dynamic features were minimized by the $L_{TD}$, then the log F0 sequence smoothed.

Also, by increasing the value of $w_{2}$ above that of $w_{1}$, the $E_{MS}$ was reduced and the F0 sequences smoothed, although the $E_{GV}$ increased. The $L_{TD}$ makes the log F0 sequences perceptually natural and smooth, but the intonation became monotonous because it degraded the GV. The $L_{GV}$ recovered this GV that had been reduced by the $L_{TD}$, which was further improved when the value of $\omega_{GV}$ was 1, although increasing the value of $\omega_{GV}$ led to the collapse of the F0 sequence. The value of $\omega_{GV}$ should not be larger than the other weights’ values. The $L_{LV}$ does not alone drastically change the log F0 sequences, but provided a weak constraint to the $L_{TD}$ and $L_{GV}$.

Figure 1 shows examples of generated log F0 sequences, and all were generally similar to those of the target sequence. The slight discontinuity shown in the sequence of the FFNN MATS system is not perceptible. The MLPG smoothed out very discontinuous sequences from the FFNN MSE and FFNN MGE systems, which became smoother than the other sequences. In part of the sequence of the LSTM MSE system in some test samples, the position of the peaks and depths mismatched those of the target. The all square roots of GV of the generated sequence were 0.03 to 0.04 smaller than that of the target. The modulation spectrum of the sequence from the FFNN MATS system was about 30 dB larger than that of the target sequence within the band of 10 Hz and higher. This result does not impact the natural perception of F0 because the components in this frequency band are sufficiently smaller than the main components of the log F0 sequence in the band of 10 Hz and lower. The same can be said for the error of the modulation spectrum error of the log F0 sequence from LSTM MSE system in the band of 60 Hz and higher.

Figure 2 shows the objective evaluation results of the log F0 sequences in the test dataset. The median of $E_{DC}$ from the FFNN MATS system was approximately 0.06, which was nearly the same as the others. The median of $E_{GV}$ from the FFNN MATS system was approximately 0.11, which was approximately 0.15 to 0.20 smaller than the others. The median of $E_{MS}$ from the FFNN MATS system was 14 dB, which was approximately 12 dB larger than that of the FFNN MSE system. This is due to the error in the band of 10 Hz and higher, as described above.

Figure 3 shows the subjective evaluation results on the prosody quality. By comparing the scores from the Tukey-Kramer method, those of the reference and anchor groups differed from those of all evaluation groups at a 1% significance level. In addition, the scores of the FFNN MATS and FFNN MSE systems differed from that of the LSTM MSE system at a 1% significance level. The quality of some samples from the FFNN MATS and FFNN MSE systems was nearly the same to that of the target.

![Figure 1](image1.png)

**Fig. 1** Examples of the log F0 sequence. The first column represents the log F0 trajectory, the second column represents the square root of the global variance, and third column represents the modulation spectrum. The gray, orange, and blue solid lines and bars represent the target log F0, generated log F0 without MLPG, and generated log F0, respectively.
4.2 Mel-Cepstrum

The loss functions available for training mcep sequences are the $L_{DC}$, $L_{TD}$, $L_{DD}$, $L_{LV}$, $L_{LC}$, $L_{GV}$ and $L_{GC}$. In order to check the behavior of these loss functions in modeling mcep sequences, preliminary experiments were conducted with about 100 combinations of the parameters of these loss functions. Based on the preliminary experimental results, the $L_{TD}$, $L_{DD}$, $L_{LV}$, $L_{LC}$ and $L_{GV}$ were determined to be necessary for modeling mcep sequences. 

![Box plots of the log F0 sequence errors for all test samples.](image)

The first column represents the absolute error for each frame of the sequence $E_{DC}$, the second column represents the absolute error of the square root of the global variance $E_{GV}$, and the third column represents the absolute error of the modulation spectrum $E_{MS}$.

![Subjective scores of the listening test by the MUSHRA method on the prosody quality with 95% confidence intervals.](image)

The log F0 sequences include the natural log F0 sequences as a reference group, the generated log F0 sequences by the FFNN MSE without MLPG as an anchor group, and generated log F0 sequences by four systems as evaluation groups.

![Examples of 15th mcep sequence.](image)

The figure A-2 in the Appendix shows a part of the preliminary experimental results. In order to improve the over smoothing of the mcep sequence mainly, the parameters were adjusted so that the $L_{GV}$ and $L_{LV}$ worked as main and the $L_{TD}$, $L_{DD}$ and $L_{LC}$ worked as auxiliary. Finally, the parameters were $\omega_{TD} = 2$, $w_1 = 1$, $w_2 = 2$, $\omega_{DD} = 2$, $\omega_{GV} = 1$, $\omega_{LV} = 3$, $\omega_{LC} = 3$, $L = -4$, and $R = 4$. When only the $L_{DC}$ was used, the mcep sequence was over-smoothed, the GV was small, and the modulation spectrum lacked high-frequency components. The $L_{GV}$ reduced the $E_{GV}$ to improve the GV, which resulted in a clear pronunciation, but it sometimes suddenly increased the power of the synthesized speech locally. Increasing the value of $\omega_{GV}$ reduced the $E_{GV}$ and $E_{MS}$, but the $E_{DC}$ was increased, and the value of $\omega_{LV}$ should not be larger than the other weights’ values. The $L_{LV}$ reduced the $E_{MS}$ to improve the modulation spectrum, which suppressed the muffled synthetic speech. Increasing the value of $\omega_{LV}$, $L$, and $R$ reduced the $E_{GV}$ and $E_{MS}$, but the fine structure of the mcep sequence became unnaturally jagged. The optimum value of $\omega_{LV}$ was around 2 to 4, and the optimum short-term length $R - L + 1$ was around 5 to 9. The $L_{GC}$, $L_{LC}$, and $L_{DD}$ did not drastically change the mcep sequences individually, but they provided a constraint for the $L_{GV}$ and $L_{LV}$. The $L_{GC}$ and $L_{LC}$ suppressed the unnaturally jagged fine structure of the mcep sequence by the $L_{LV}$, where the $L_{LC}$ was more effective than the $L_{GC}$ because the mcep sequence’s distribution in the entire utterance cannot be fit by a single Gaussian. The $L_{DD}$ suppressed the sudden increase in the power of part of the synthesized speech, and the $L_{TD}$ smoothed the mcep sequences and stabilized the synthesized speech, which is the opposite effect of $L_{LV}$. Stability of the synthesized speech means that it is uniform throughout the utterance without a sudden power or voice color fluctuation. In this experiment, we adopted the parameters that focus on the stability of the synthesized speech, even at the sacrifice of improving the modulation spectrum.

Figure 4 shows examples of the generated 15th mcep sequence, which did not have a complicated fine structure...
as did the target. However, the trajectories were roughly consistent, and the square roots of GV were approximately 10% smaller than that of the target. In the FFNN MSE, FFNN MGE, and LSTM MSE systems, the change ratio of the squared roots of GV was consistent with the enhancement factor, and the modulation spectra were approximately 20 dB smaller than that of the target in the band of 20 Hz and higher.

Figure 5 shows the objective evaluation results of the mcep sequences in the test dataset of the FFNN MATS system. The median of $E_{DC}$ was approximately 0.1, which was slightly smaller than the other systems, and the median of $E_{GV}$ was approximately 0.02, or about 1.7 times smaller than the others. The median of $E_{MS}$ was nearly the same as those of the FFNN MGE and LSTM MSE systems at 15 dB, and the 25th percentile of $E_{MS}$ was smaller than the other systems at 7 dB. These box plots indicate that nearly half of the test dataset of the FFNN MATS system had less error with the target than that of the other systems.

Figure 6 shows the subjective evaluation results of voice quality. From comparing the scores with the Tukey-Kramer method, the reference and anchor groups differed from those of all evaluation groups at a 1% significance level, and the score of the FFNN MATS groups differed from those of the other evaluation groups at a 5% significance level. The synthesized speeches from the FFNN MATS system differed from those of the others with a suppression of the muffling sounds.

### 4.3 Duration of Speech Parameter Generation Process

Table 2 shows the duration of the prediction and post-processing for speech parameter generation. Each system was implemented in Python, and the calculated processing times were averaged over 100 cycles for a 1,000-frame sequence with frame period of 5 ms. All processing times were less than 5,000 ms, which represents the length of the input sequence. The FFNN MATS system generated speech parameters in 50–120 ms in the low-spec CPU environment, about 20 ms in the high-spec CPU environment, and 1 ms in the GPU environment. The LSTM MSE system generated speech parameters in 900–1080 ms in the low-spec CPU environment, 240–320 ms in the high-spec CPU environment, and 190–210 ms in the GPU environment. Due to the op-
timization of Python’s LSTM module, the processing times for the GPU environment were about 40–60 ms slower than those of the high-spec CPU environment. The FFNN MSE and FFNN MGE systems were about 3 to 60 times slower in the low- and high-spec CPU environments compared to the FFNN MATS and LSTM MSE systems. Benefitting from the parallel processing of the GPUs, the FFNN MSE and FFNN MGE systems generated speech parameters in about 300–400 ms. These process times, however, are longer than those of the FFNN MATS and LSTM MSE systems. The processing time of the MLPG, which is majority of that of the FFNN MSE and FFNN MGE systems, can be reduced by using the recursive MLPG, which was proposed for a low-computational cost speech synthesis system. The processing time of the recursive MLPG is faster than that of the conventional MLPG used in the experiments, but it does not reach 0 ms because of the delay of the preceding frames. Thus, there is no doubt that the FFNN MATS system, which works frame-by-frame and does not require any post-processing at all, is fastest.

5. Discussion

As a solution to enable the generation of a speech parameter sequence by capturing the temporal structure, even with the FFNN, we propose a new training method that considers multiple attributes of the temporal sequence. Comparative evaluations with conventional temporal sequence training methods demonstrated that the FFNN could generate perceptually superior log F0 and mcep sequences by leveraging our training method.

For the evaluation of F0, although these sequences from each system were smooth, the perception of the accents was different and results from the training of the dynamic feature of F0, the relative change in F0, which is deeply involved in pitch perceptions in Japanese [17]. The DNNs generate the F0 sequence with the natural Japanese accent structure by explicitly training the dynamic feature of F0. The FFNN MATS and FFNN MSE systems directly train the dynamic feature of F0, and the FFNN MGE system indirectly trains the dynamic feature of F0 through the MLPG, while the LSTM MSE system does not.

For the evaluation of mcep, the FFNN MATS system demonstrated outstanding scores in the prediction error of the mcep sequences and the naturalness of the synthesized speeches. By contrast, the other systems were nearly the same except for the error of the modulation spectrum, which results from the backpropagated error. As this error is the basis for training its parameters, the DNN can obtain the parameters to precisely generate the mcep sequence by training under the stringent condition of minimizing multiple errors. The FFNN MATS system is trained based on minimizing the errors of TD, DD, GV, LV, and LC, while the other systems are trained only based on minimizing the DC error. Because minimizing the DC error is a loose training condition for modeling the mcep sequence, the MLPG and RNNs cannot capture the complex structure of the mcep sequence.

DNNs with a simple structures and elimination of post processing are necessary for high-speed speech synthesis processing in embedded devices with low computational power and applications where the computational resources for speech synthesis are limited. From this perspective, the FFNN MATS system has simpler DNN structure than the LSTM MSE system and does not require the MLPG and cepstrum enhancement post-filter used in the FFNN MSE and FFNN MGE systems. Despite the simplification of processing to reduce such computational costs, the quality of synthesized speech in the FFNN MATS system was superior to other systems. In addition, for the duration required to generate the speech parameters, the FFNN MATS system was the fastest of all by generating a five-second speech sequence within hundreds of milliseconds, even on a CPU designed for previous generation laptop computers. This result suggests that the system offers a more considerable margin to generate the speech parameters compared to the other systems, even in poorer computing environments than utilized in the experiment.

The first thing we learned through this experiment is that the \( L_{GC} \) did not work to encompass the \( L_{GV} \) despite the fact that mathematically the diagonal of the GC is the same as the GV. This happens because the losses of the non-diagonal elements, which constitute a large part of the GC, are more dominant than the losses of the diagonal elements. The same is true for the \( L_{LC} \). The second thing is the strength of association between the perception of naturalness of speech and the features of speech parameters. The explicit and focused training of the dynamic feature of the log F0 greatly contributed to improve the naturalness of synthesized speech. This is because the dynamic feature of the log F0 directly affects the Japanese accent perception. The reduction of errors related to the mcep, the square root of the GV and the modulation spectrum, contributed to improve the naturalness of synthesized speech. Despite this improvement, there was still a large difference in the naturalness between the raw and synthesized speeches. Discovering and modeling which features of the mcep are deeply related to the perception of naturalness of speech is the key to further improving the naturalness of synthesized speech.

Ablation experiments were not conducted to reveal which combination of loss functions was the most effective in modeling each speech feature sequence. Through ablation experiments require an enormous number of conditions to be tested, e.g., a combination of 13 parameters varying in ten steps in seven loss functions. It would be difficult to conduct such experiments. However, the experimental results show that the combinations of loss functions determined in the manner described at the beginning of Sects. 4.1 and 4.2 are sufficiently effective in modeling of the F0 and mcep sequence.

The use of multiple loss functions increases the time for training models of speech features, but the increase in time is small. For example, the time per an epoch for training the mcep model in the FFNN MATS system with 2000 utterance is about 13 seconds when only the \( L_{DC} \) is
used, and about 24 seconds when seven loss functions are used. This time is much smaller than the time per epoch to train the mcep model in the LSTM MSE system, which is about 10 minutes. Thus, the increase in model training time that comes with using multiple loss functions need not be a concern.

The strategy of our proposed method is opposite to popular strategies that allow DNNs to acquire the relationship between the input and output automatically. The proposed training method requires manually defining multiple attributes of the temporal sequences, such as GV and LV, in the loss function and adjusting their parameters. Although this manual definition and adjustment processes are cumbersome, the manipulation is intuitive because the behaviors of these parameters are easier to understand than that of the DNN parameters, such as the number of nodes, the number of layers, the type of activation function, and the network structure. By changing the configuration of the proposed loss function, log F0 sequences can be accommodated with a gradual temporal change as well as mcep sequences with complex structures. The experimental results demonstrate that the FFNN generates perceptually superior speech parameters without post-processing by using optimal loss function settings for each speech parameter. Settings the optimal loss function requires training of models with hundreds of combinations of loss function settings and listening tests of the many speeches synthesized by those models. The automatic determination of the optimal loss function of the DNN is a future work.

6. Conclusion

In this paper, we proposed a new training method by back-propagating the errors of multiple attributes of the temporal sequence through the loss function. Subjective evaluation results demonstrated that the MATS loss function enables the FFNN without the MLPG to generate perceptually superior speech parameters compared to the FFNN with the MLPG and LSTM-RNNs. This approach achieves a DNN-based speech synthesis system with lower computational cost compared to conventional systems while operating at high-speeds in poor computing environments.

References

[1] H. Zen, A. Senior, and M. Schuster, “Statistical Parametric Speech Synthesis Using Deep Neural Networks,” Proc. ICASSP, pp.7962–7966, 2013.

[2] K. Nakamura, S. Takaki, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Computational complexity reduction method for CNN-based singing voice synthesis,” Proc. ASJ2019A, pp.939–940, Sept, 2019.

[3] Z. Kons, S. Shechtman, A. Sorin, C. Rabinovitz, and R. Hoory, “High quality, lightweight and adaptable TTS using LPCNet,” Proc. INTERSPEECH 2019, pp.176–180, Sept, 2019.

[4] H. Banno, H. Hata, M. Morise, T. Takahashi, T. Ino, and H. Kawahara, “Implementation of realtime straight speech manipulation system,” Acoust. Sci. & Tech., vol.28, no.3, pp.140–146, 2007.

[5] M. Morise, F. Yokomori, and K. Ozawa, “WORLD: a vocoder-based high-quality speech synthesis system for real-time applications,” IEICE Trans. Inf. & Syst., vol.E99-D, no.7, pp.1877–1884, 2016.

[6] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A Generative Model for Raw Audio,” Proc. ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, p.125, Sept. 2016.

[7] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient Neural Audio Synthesis,” Proc. International Conference on Machine Learning (ICML), Stockholm, Sweden, pp.2415–2424, 2018.

[8] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: A Flow-based Generative Network for Speech Synthesis,” Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp.3617–3621, May 2019.

[9] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech synthesis based on hidden Markov models,” Proc. IEEE, vol.101, no.5, pp.1234–1252, May 2013.

[10] H. Zen and H. Sak, “Unidirectional Long-Short Term Memory Recurrent Neural Network with Recurrent Output Layer for Low-Latency Speech Synthesis,” Proc. ICASSP, pp.4470–4474, 2015.

[11] N. Matsunaga, Y. Ohtani, and T. Hirahara, “Loss Function considering Temporal Sequence for Feed-Forward Neural Network – Fundamental Frequency Case,” Proc. 10th ISCA Speech Synthesis Workshop, pp.143–148, Vienna, Austria, 2019.

[12] T. Nose, V. Chunwijitrta, and T. Kobayashi, “A parameter Generation Algorithm Using Local Variance for HMM-Based Speech Synthesis,” Proc. IEEE, vol.8, no.2, pp.221–228, 2014.

[13] T. Toda and K. Tokuda, “Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis,” Proc. INTERSPEECH 2005, pp.2801–2804, Lisbon, Portugal, Sept. 2005.

[14] Z. Wu and S. King, “Minimum trajectory error training for deep neural networks combined with stacked bottleneck features,” Proc. Interspeech 2015, pp.309–313, 2015.

[15] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, “Trajectory training considering global variance for speech synthesis based on neural networks,” Proc. Int. Conf. Acoust., Speech, Signal Process., pp.5600–5604, Shanghai, China, March 2016.

[16] M. Morise, Y. Toyoda, and K. Ozawa, “Influence of exaggerated temporal fluctuation on singing voice of perception of humanity,” [Translated from Japanese.], IPSJ Special Interest Group Technical Report, vol.2017-MUS-115, no.55, pp.1–6, June 2017.

[17] T.C. Ishi, N. Minematsu, and K. Hirose, “Identification of Japanese accent in continuous speech considering pitch perception,” IEICE Technical Report SP, 101(270), pp.23–30, 2001.

[18] HTS, http://hts.sp.nitech.ac.jp/

[19] N. Matsunaga, Y. Ohtani, and T. Hirahara, “Normalized Method of Linguistic Feature Suitable for Fundamental Frequency in Japanese Text to Speech Using Deep Learning,” Trans. IEICE, vol.J102-D, no.10, pp.721–729, Oct. 2019.

[20] V. Nair and G.E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” Proc. 27th International Conference on Machine Learning, pp.807–814, Haifa, Israel, 2010.

[21] D.P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” arXiv:1412.6980, 2014.

[22] Y. Saito, S. Takamichi, and H. Saruwatari, “Statistical Parametric Speech Signal Processing Toolkit, http://speech.ics.ohu.ac.jp/sp-tk.sourceforge.net/.

[23] E. Lockhart, F. Stimberg, A. van den Oord, S. Dieleman, and K. Kavukcuoglu, “Efficient Neural Audio Synthesis,” Proc. International Conference on Machine Learning (ICML), Stockholm, Sweden, pp.2415–2424, 2018.

[24] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Incorporating a mixed excitation model and postfilter into HMM-based text-to-speech synthesis,” Systems and Computers in Japan, vol.36, no.12, 2005.

[25] Z. Wu, O. Watts, and S. King, “Merlin: An Open Source Neural Network Speech Synthesis System,” Proc. 9th ISCA
Noriyuki Matsunaga received the B.E. and M.E. degrees in engineering from Toyama Prefectural University in 2009 and 2011, respectively. He is currently a researcher at research and development department at the AI, Inc. He is a member of the Acoustical Society of Japan.

Yamato Ohtani received the B.E. degree in engineering from Osaka University, Osaka, Japan, in 2005 and the M.E. and Ph.D. degrees in engineering from the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST) in 2007 and 2010, respectively. He was an intern researcher at ATR Spoken Language Communication Research Laboratories, Kyoto, Japan from 2006 to 2009. He was a researcher at Toshiba Corporation 2010 to 2017. He is currently the head of research and development department at AI Inc. His research interests include statistical approaches to speech processing such as speech synthesis and voice transformation. He received the Awaya Prize Young Researcher Award from ASJ in 2015. He is a member of ASJ, IPSJ, ISCA and IEEE.

Tatsuya Hirahara received the B. Eng., M. Eng., and Dr. Eng. degrees in Electronic Engineering from Hokkaido University, Sapporo, Japan, in 1978, 1980, and 1983, respectively. He worked for NTT-ECL Yokosuka, ATR Auditory and Visual Perception Labs., NTT Basic Research Labs., ATR-International Corporate Planning Division, NTT Communication Science Labs., and ATR Human Information Science Lab. He was also a visiting researcher at the Speech Communication Group of the MIT-RLE and the Eaton-Peabody Laboratory of Auditory Physiology at Massachusetts Eye and Ear Infirmary. Since 2006, he has been Professor of Faculty of Engineering at Toyama Prefectural University. Dr. Hirahara is a member of the Acoustical Society of Japan, the Acoustical Society of America, and the Virtual Reality Society of Japan.
Appendix:

Fig. A.1 Examples of log F0 sequences and box plots of log F0 sequence errors for all test samples in preliminary experiments. The first column represents the log F0 trajectory, the second column represents the absolute error for each frame of the sequence $E_{DC}$, the third column represents the absolute error of the square root of the global variance $E_{GV}$, and the fourth column represents the absolute error of the modulation spectrum $E_{MS}$. 
Examples of mcep sequences and box plots of mcep sequence errors for all test samples in preliminary experiments. The first column represents the 15th mcep trajectory, the second column represents the absolute error for each frame of the sequence $E_{DC}$, the third column represents the absolute error of the square root of the global variance $E_{GV}$, and the fourth column represents the absolute error of the modulation spectrum $E_{MS}$.

**Fig. A.2**
| Frame index | $E_{dc}$ | $E_{gv}$ | $E_{ms}$ [dB] |
|-------------|---------|---------|-------------|
| (1,1)       |         |         |             |
| (1,2)       |         |         |             |
| (1,4)       |         |         |             |
| (1,1,0.5)   |         |         |             |
| (1,1,1)     |         |         |             |
| (1,1,2)     |         |         |             |
| (2,1,2,1,1,1,-2,2) | | | |
| (2,1,2,1,1,1,-2,2) | | | |
| (2,1,2,1,1,1,-3,3) | | | |
| (2,1,2,1,1,1,-3,3) | | | |
| (2,1,2,1,1,1,-4,4) | | | |
| (2,1,2,1,1,1,-4,4) | | | |
| (2,1,2,1,1,2,-2,2) | | | |
| (2,1,2,1,1,2,-2,2) | | | |
| (2,1,2,1,1,2,-3,3) | | | |
| (2,1,2,1,1,2,-3,3) | | | |
| (2,1,2,1,1,2,-4,4) | | | |
| (2,1,2,1,1,2,-4,4) | | | |

Fig. A-2 (Continued)