Jazz Piano Trio Synthesizing System Based on HMM and DNN

Takeshi Hori
Meiji University
cs51003@meiji.ac.jp

Kazuyuki Nakamura
Meiji University
knaka@meiji.ac.jp

Shigeki Sagayama
Meiji Univerisity
sagayama@meiji.ac.jp

ABSTRACT

In this paper, we discuss a computational model of an automatic jazz session, which is a statistically trainable. Moreover, we describe a jazz piano trio synthesizing system that was developed to validate our model.

Most previous mathematical models of jazz session systems require heuristic rules and human labeling of training data to estimate the musical intention of human players in order to generate accompaniment performances. In contrast, our goal is to statistically learn the relationship between a piano, a bass, and a drum player from performance MIDI data as well as information contained in lead sheets, for instance tonic key and chord progression. Our system can generate the performance data of bass and drums from only piano MIDI input, by learning the interrelationship of their performances and time series characteristics of the three involved instruments. The experimental results show that the proposed system can learn the relationship between the instruments and generate jazz piano trio MIDI output from only piano input.

1. INTRODUCTION

We previously developed an automatic accompaniment system called Eurydice [1, 2], that can handle tempo changes and note insertion/deviation/substitution mistakes in the accompanied human performance as well as repeats and skips. Although this system can deal with various mistakes, the musical score of the accompaniment is fixed in Eurydice, since the system follows the player's performance by matching between the performance and the music score.

To appropriately accompany improvised performances, the accompaniment system needs to estimate the musical intention of the improvising player. Such systems are called jazz session systems. Previous jazz session systems [3–7] require heuristic rules and human labeling of training data to estimate the musical intention of human players. In contrast, our approach is to statistically learn the relationship between performances of the different instruments from performance MIDI data as well as information contained in lead sheets, in order to avoid the need of parameters set by the developer.

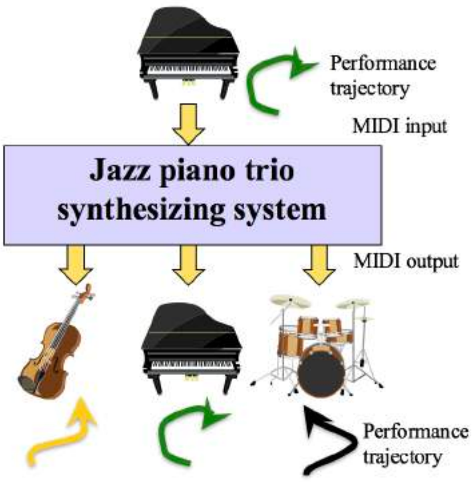

Figure 1. The input and output of the system described in this paper. The instrument performances are modeled as trajectories in a musical performance feature space, as described in section 2.1.

We previously developed an offline jazz piano trio synthesizing system [8] utilizing a database of performances from which appropriate accompaniment performances of bass and drums are selected based on characteristics of the piano MIDI input. Experimental results demonstrated the system's ability to meaningfully accompany piano performances to a certain degree.

This paper describes an offline jazz piano trio synthesizing system based on free generation of accompaniment performances without using the performance database. Instead, the system can learn the interrelationship of the instrument performances and their time series characteristics using hidden Markov models and deep neural networks. The approach follows a recent trend to uses deep learning (especially long short-term memory recurrent neural networks (LSTM-RNN [9]) in automatic music generation. The composition system DeepBach [10] uses a bidirectional LSTM to generate Bach-style music. Chu et al. [11] use a LSTM to generate POP music.

As illustrated in Fig. 1, the modeled jazz piano trio consists of a piano, whose performance MIDI data is used as input and a bass as well as drums whose performance MIDI data is generated by the computer.

2. MATHEMATICAL MODEL OF THE JAZZ SESSION SYSTEM

2.1 Trajectory model

We describe a musical performance as a trajectory in musical feature space [12]. At every point of time in the performance, the musical features of the current performance state (tempo, harmony, etc.) identify a position in the musical feature space. Consequently, a jazz musician ”moves” through this feature space while performing, resulting in a performance trajectory.

In case of a jazz session comprising multiple instruments, one can observe a correlation between the trajectories of the different instruments during a ”good session”. Therefore, our aim is to implement a statistically trainable model that is able to learn the correlation between the instruments' trajectories. This correlation is used to generate the performance trajectories of accompaniment instruments (bass and drums) from the performance trajectory of the main instrument (piano).

2.2 The problems and solution strategies

Since the amount of available jazz performance data was relatively small, three problems had to be addressed in order to use machine learning methods effectively. Firstly, the number of musical features that could possibly be used to describe performances is large, resulting in a very high-dimensional feature space. To keep the dimensionality relatively low, we chose a set of features especially suited for jazz music. Secondly, since the data in feature space is sparse, an interpolation method is required to generate continuous performance trajectories through this space. This problem is addressed by segmenting the feature space into subspaces using a continuous mixture hidden Markov model. Thirdly, the jazz session system needs to learn the non-linear dependency between the performance trajectories of the different instruments, which is achieved using neural networks.

2.2.1 High dimensionality

We selected 68 features that are extractable from the music performance at every bar. The features were chosen based on opinions of jazz musicians, and are the following:

- *Piano-specific features*
  - The range between the highest and lowest notes.
  - The number of notes contained in the current diatonic scales, as well as the numbers of tension notes, avoid notes, and blue notes.
  - The ratios of diatonic notes, tension notes, avoid notes and blues notes to the total number of notes in the current bar.
- *Bass-specific features*
  - The range between the highest and lowest notes.
- *Drums-specific features*
  - The number of notes played by the hi-hat, snare drum, crash cymbal and the total number of all other notes.
- *Common features*
  - The number of notes, the number of simultaneously played notes, and the average MIDI velocities (loudness).
  - The ratio of off-beat notes to all notes in the current bar.
  - The ratios of all features (except for the off-beat-ratio) with respect to the previous bar as well as the ratios with respect to the complete performance.

We refer to these features as ”style features” in the following.

2.2.2 Trajectory continuity

To generate performance trajectories from anywhere in feature space, one has to discretize or interpolate the available music performance data. In the proposed jazz session system, this is achieved by using clustering in the musical feature spaces of the accompaniment instruments. As a first step, the musical feature vectors are clustered using a continuous mixture hidden Markov model. The observation probabilities related to each hidden state of the HMM are described utilizing a Gaussian mixture model (GMM). Based on preliminary analysis of the data, the number of hidden states was chosen to be 4, using 4 GMM clusters per state. As a result, the trajectory model is approximated by a stochastic state transition model. A musical performance corresponds to a Viterbi path in the HMM and the interaction between instruments to co-occurrence of state transitions.

To increase precision, we further segment each HMM state internally. We are using a GMM to cluster the vectors of deviation from the HMM state's centroid.

The GMMs approximate a distribution as a mixture of normal distributions, and their updating algorithm applies the auxiliary function method. In case of the HMM, the state transition probabilities and the states' GMMs are optimized jointly using the auxiliary function method.

The auxiliary function has to fulfill Jensen's inequality, formulated as follows:

$$F(\sum_i \phi_i f_i(\boldsymbol{x};\theta)) \quad \geq \quad \sum_i \phi_i F(f_i(\boldsymbol{x};\theta)) \qquad (1)$$

$$=: \quad G(\theta,\phi) \qquad (2)$$

$$s.t. \sum_i \phi_i \quad = \quad 1, \qquad (3)$$

where $G(\theta,\phi)$ is the auxiliary function, $\phi = (\phi_1, \cdots, \phi_n)$ are the auxiliary variables, θ is a parameter, and $F(\cdot)$ a convex function. $\boldsymbol{x}$ is either a style feature vector (when obtaining the HMM states), a vector of deviation from an HMM state's centroid (for state internal clustering).

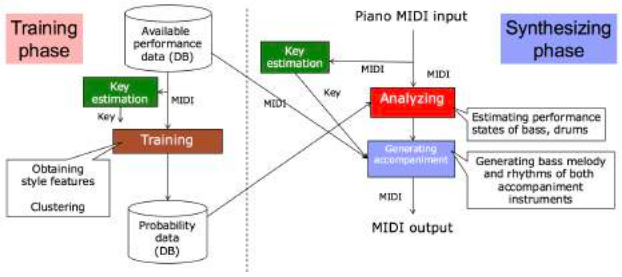

Figure 2. The training and synthesizing phases of the proposed system. During training, the instruments' interrelationship is learned and used to synthesize "improvised" accompaniment performances.

The functions $f_i(\cdot)$ encode the class assignments (HMM states or deviation clusters, respectively). In the auxiliary function method, the following two iterative update steps are applied alternately:

$$\phi^* = \arg\max_{\phi} G(\theta, \phi), \tag{4}$$

$$\theta^* = \arg\max_{\theta} G(\theta, \phi). \tag{5}$$

2.2.3 non-linear instrument performance interrelationship

A LSTM is used to learn the relationship between the musical features of the piano performance and the HMM segments of the bass and drums performance feature spaces. To learn the relationship between the piano features and vector of deviation from the states' centroids, a deep belief network (DBN) is used. The training process of these networks is described in the following section.

3. JAZZ PIANO TRIO SYNTHESIZING SYSTEM

Based on the stochastic state transition model which approximates a performance trajectory model, as described in the previous section, we developed an automatic jazz accompaniment system. The system receives MIDI-format data of only the piano performance, and as output generates MIDI-format data of the performance of the complete jazz piano trio (bass and drums added to the piano).

Fig. 2 illustrates the training and synthesizing phases of our system, which are described in the following.

3.1 Training phase

The training phase consists of following 6 steps:

1. *Key estimation*
   Estimating the tonic key of every bar of the performances in the training data.

2. *Feature extraction*
   Extracting performance features and computing the style features of all instruments at every bar.

3. *Data segmentation*
   Segmenting the performance feature spaces of bass and drums using a continuous mixture HMM.

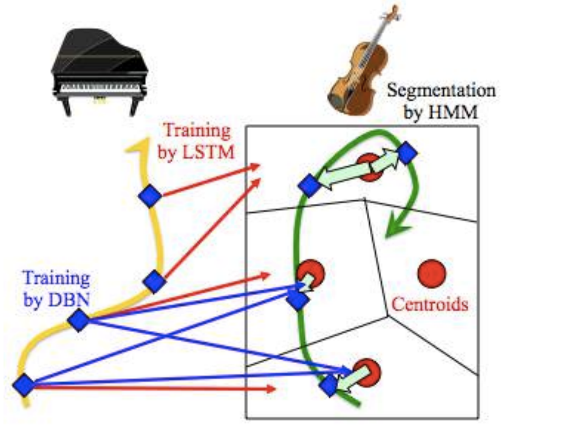

Figure 3. The training result of the proposed system. A LSTM matches piano performance vectors to the other instruments HMM states and a DBN refines the matching by providing deviation vectors from states' centroids.

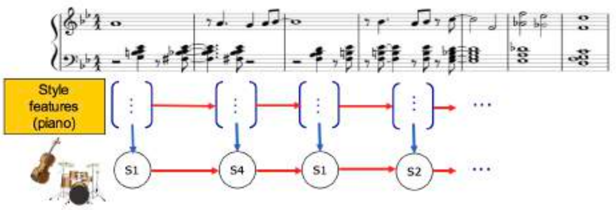

Figure 4. The LSTM matches style feature vectors of the piano performance to HMM states ($S_1, \cdots, S_4$) of the other instruments.

4. *Performance correlation (coarse)*
   Training a LSTM network to match piano performance feature vectors to the other instruments' feature space segments derived in (3.).

5. *Clustering of centroid deviation vectors*
   Using a Gaussian mixture model (GMM) to cluster the vectors of deviation from the centroids of the HMM states derived in (3.).

6. *Performance correlation (fine)*
   Training a DBN [13] to match piano performance features vectors to the centroid deviation clusters derived in (5.).

Fig. 3 illustrates the training process described above.

3.1.1 Long short-term memory (LSTM)

A LSTM [9,14] is a type of recurrent neural network which can effectively handle time series data. In our system, a LSTM is used to match piano performance features to the HMM states which are obtained as clusters in other feature space of the two other instruments (see section 2.2.2). This essentially corresponds to identifying non-linear clusters in the piano performance feature space which are related to those of the other instruments. This dependency is illustrated as red arrows in Fig. 3. The LSTM is trained using MIDI data of jazz piano trio performances.

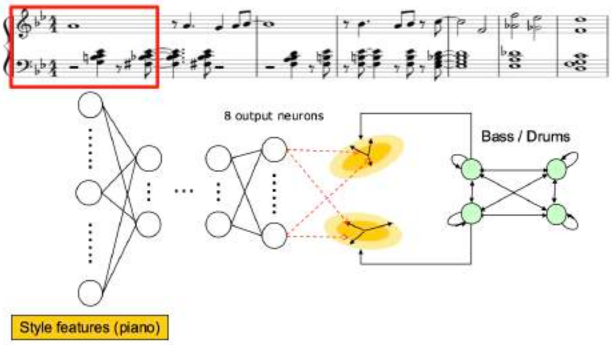

Figure 5. The DBN matches style feature vectors of the piano performance to deviation vectors from the centroids of the HMM states.

It is then used to estimate performance states of the accompanying instruments from the piano performance as shown in Fig. 4.

3.1.2 Deep belief network (DBN)

The used DBN consists of 5 restricted Boltzmann machine (RBM) layers and a classification output layer. Our jazz session system utilizes this DBN to match piano performance feature vectors to vectors of deviation from centroids of the HMM segments. As described in section 2.2.2, a GMM is applied to obtain 8 clusters of deviation vectors for each HMM state. The DBN effectively assigns a set of 4 deviation vectors (one for each HMM state) to every vector in the piano performance feature space (see Fig. 5).

We use Gaussian-Bernoulli RBM layers [15] in our system's DBN, because this type of RBM is able to handle continuous values, which our case are the style feature values extracted from the instrument performances. To update the RBM layers, we use the persistent contrastive divergence (PCD) method [16], because of its efficiency.

3.1.3 Combined results

The LSTM network and DBN in combination compute the tuple of HMM state and the associated centroid deviation vector for each piano performance vector. This tuple identifies a unique vector in the performance feature space of an accompaniment instrument, corresponding to a discretized step of the performance trajectory of this instrument. Two separate sets of networks are trained for the two accompaniment instruments (bass and drums).

3.2 Synthesizing phase: Generating bass and drums performances

In the synthesizing phase, we use the neural networks described in the previous section to estimate the performance states of bass and drums from a given piano performance feature vector. In the context of a complete performance, this corresponds to estimating the two accompaniment performance trajectories based on the piano performance trajectory.

To develop a method to automatically generate improvised performances from given performance trajectories, we consulted several jazz musicians. The devised method comprises the following three steps for the bass performance:

1. Constructing a melody without rhythm, based on the current harmony information, including the chord and information about usable notes (jazz scales, etc.).

2. Deciding on a rhythm pattern based on the current performance features. The rhythm is generated using a rhythm tree as explained in section 3.2.2.

3. Combining the melody with the rhythm.

In case of the drums, only the second step is required, where the rhythm is generated for every individual element (snare drum, hi-hat, etc.). The performance generation method is applied for every bar of the musical performance.

3.2.1 Generation of the bass melody

Since the melody of the bass generally doesn't include large interval jumps, we select melody notes close to the respective previous note. In detail, we compute the probability distribution of intervals occurring in the bass performances of the training data. This distribution is approximated as a Gaussian distribution with standard deviation $\sigma_{S,C}$ and obtained for every state S of the HMM and deviation vector cluster C. In the melody generation process, the probability of the ith MIDI note $x_{t,i}$ of bar number t is obtained as follows:

$$x_{t,i} \sim \mathcal{N}(\frac{1}{N_{t,i-1}} \sum_{j=0}^{i-1} x_{t,j}, (\frac{\sigma_{S,C}}{2})^2), \tag{6}$$

where the average of all notes played in bar t before note $x_{t,i}$ is the mean of the Gaussian distribution. From this distribution, the notes are randomly sampled for every melody step.

3.2.2 Rhythm pattern generation

To generate the rhythms of the accompaniment instruments, we use idea of the rhythm tree [17]. In the rhythm tree, the root node corresponds to a whole note, which spans the whole bar. The tree then branches into nodes representing half notes or combinations of a half note and a half pause, and then further into quarter, eighth, and sixteenth note nodes. In an iteration of the rhythm pattern generation process, a note can be replaced by an associated child node. The number of iterations equals the number of tree branching layers (4 if branching until reaching sixteenth notes). A "rhythm characteristic" is a set of probability values that determine whether or with which child node a note is replaced in an iteration. The rhythm characteristic probabilities are extracted from the training data for every HMM state and deviation vector. In the synthesizing phase, the rhythm characteristics are then used to sample a rhythm pattern that fits to the performance state obtained as the tuple of HMM state and deviation vector from the neural networks.

4. EXPERIMENTAL EVALUATION

We trained our jazz piano trio synthesizing system using MIDI data of jazz piano trio performances of 12 different songs. We used the Python library Theano to implement the neural networks, whose properties are listed in table 1. The 12 training songs contain a total of about 2000 bars of which about 150 bars were used as validation set. The matching error of the LSTM, which matches piano performance vectors to performance states of the other instruments, was approximately 26% in case of the bass and 49% in case of the drums. The matching error of the DBN, which matches piano feature vectors to HMM state centroid deviation vector clusters, was approximately 28% in case of the bass and 38% in case of the drums.

Table 1. System Parameters

HMM parameters	
The number of state	4
The number of mixture	4
Epoch	100
GMM parameters (deviation from center)	
The number of mixture	8
Epoch	200
LSTM parameters	
Learning rate	0.0001
The number of middle layer units	30
Momentum	0.9
Drop out	0.5
Epoch	5000
DBN parameters	
Learning rate	0.00001
The number of hidden units (RBM)	50
Momentum	0.9
Weight decay	0.001
Objective average activation	0.01
Coefficient for sparse regularization	0.1
The number of layers	5
Prior epoch	10000
Epoch	100000

We let the system generate accompaniment performances for another song and asked 4 students to rate the result using a five-point grading scale (2 students had more than ten years of musical experience). We then compared the scores with evaluation results of previously used methods based on the selection of performance data from a database [18]. These methods include:

- Random: Only harmony constraints are met and both melody and rhythm pattern are drawn randomly from the performance database.

- Nearest neighbor: For each bar, the system searches the database to find the piano performance with the closest resemblance to the input piano performance, and the associated accompaniment performances are used for the respective bar.

- NMF: The piano performance feature space is clustered using non-negative matrix factorization (NMF) and the nearest neighbor method is applied within the resulting clusters.

- NMF + co-occurrence + trigram: The other instruments' feature spaces are clustered as well (using NMF). To identify the instrument performance interrelationship, co-occurrence matrices of the different instruments' performance feature clusters are obtained. Additionally, for each instrument, trigram probabilities are computed. The accompaniment performances are then chosen by maximizing the trigram probability combined with the co-occurrence probability.

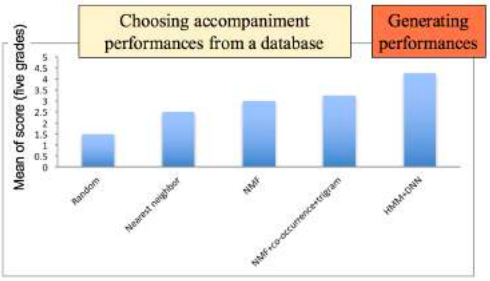

Figure 6. A result of comparative evaluation experimental. Every evaluation score is the average of the grades given by the four research participants.

The evaluation results are shown Fig. 6. A significant improvement over the previously used methods was observed. While the best previous method achieved a score of 3.25, the proposed system reached a score of 4.25.

5. CONCLUSION

We modeled a jazz piano trio session as the set of performance trajectories of the trio's instruments. To be able to learn from a small amount of data, we approximated the trajectory model by a stochastic state transition model. We used a LSTM network as well as a DBN to learn the correlation between the performance of the different jazz piano trio instruments from the performance MIDI data of jazz songs. Based on the estimated performance states of the accompaniment instruments, the system generates melody and rhythm of the accompaniment.

The proposed jazz session system received a higher evaluation score than our previous system. This confirms that the use of subspace clustering is an effective means to increase the precision of the used machine learning methods. However, there is still room for improvement, as the amount of available training data was relatively small and further investigation of the system and style features might be fruitful.

Another possibility to improve the system might be the use of context clustering. In our system, the context will be the musical structure or the harmony context (chord progression dependencies). Yoshimura et al. [19] have observed that context clustering can be useful to increase the effectiveness of machine learning methods utilized in natural speech synthesizing systems.

Furthermore, one could investigate how to obtain more training data, replace parametric clustering methods with non-parametric ones, or enable the system to handle audio input data instead of MIDI data. Moreover, our future goal is to develop a real-time jazz session system.

6. REFERENCES

[1] H. Takeda, N. T, S. Sagayama *et al.*, "Automatic accolnpaniment system of MIDI performance using HMM-based score followng," *IPSJ SIG Technical Reports*, pp. 109–116, 2006.

[2] E. Nakamura, H. Takeda, R. Yamamot, Y. Saito, S. Sako, S. Sagayama *et al.*, "Score following handling performances with arbitrary repeats and skips and automatic accompaniment," *IPSJ Journal*, vol. 54, no. 4, pp. 1338–1349, 2013.

[3] S. Wake, H. Kato, N. Saiwaki, and S. Inokuchi, "Cooperative musical partner system using tension - parameter : JASPER (jam session partner)," *IPSJ Journal*, vol. 35, no. 7, pp. 1469–1481, 1994.

[4] I. Hidaka, M. Goto, and Y. Muraoka, "An automatic jazz accompaniment system reacting to solo," *IPSJ SIG Technical Reports*, vol. 1995, no. 19, pp. 7–12, 1995.

[5] M. Goto, I. Hidaka, H. Matsumoto, Y. Kuroda, and Y. Muraoka, "A jazz session system for interplay among all playersI. system overview and implementation on distributed computing environment," *IPSJ SIG Technical Reports*, vol. 1996, no. 19, pp. 21–28, 1996.

[6] I. Hidaka, M. Goto, and Y. Muraoka, "A jazz session system for interplay among all playersII. implementation of a bassist and a drummer," *IPSJ SIG Technical Reports*, vol. 1996, no. 19, pp. 29–36, 1996.

[7] M. Hamanaka, M. Goto, H. Aso, and N. Otsu, "Guitarist simulator: A jam session system statistically learning player's reactions," *IPSJ Journal*, vol. 45, no. 3, pp. 698–709, 2004.

[8] T. Hori, K. Nakamura, and S. Sagayama, "Selection and concatenation of case data for piano trio considering continuity of bass and drums," *Autumn Meeting Acoustical Society of Japan*, vol. 2016, pp. 701–702, 2016.

[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[10] G. Hadjeres and F. Pachet, "Deepbach: a steerable model for bach chorales generation," *arXiv preprint arXiv:1612.01010*, 2016.

[11] H. Chu, R. Urtasun, and S. Fidler, "Song from pi: A musically plausible network for pop music generation," *arXiv preprint arXiv:1611.03477*, 2016.

[12] T. Hori, K. Nakamura, and S. Sagayama, "Statistically trainable model of jazz session: Computational model, music rendering features and case data utilization," *IPSJ SIG Technical Reports*, vol. 2016, no. 18, pp. 1–6, 2016.

[13] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[14] A. Graves, "Supervised sequence labelling," in *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, 2012, pp. 5–13.

[15] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," in *Advances in neural information processing systems*, 1992, pp. 912–919.

[16] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[17] M. Tsuchiya, K. Ochiai, H. Kameoka, and S. Sagayama, "Probabilistic model of two-dimensional rhythm tree structure representation for automatic transcription of polyphonic midi signals," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*. IEEE, 2013, pp. 1–6.

[18] T. Hori, K. Nakamura, and S. Sagayama, "Automatic selection and concatenation system for jazz piano trio using case data," *Proc. ISCIE International Symposium on Stochastic Systems Theory and its Applications*, vol. 2017, 2017.

[19] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum , pitch and duration in hmm-based speech synthesis," *The IEICE Transactions*, vol. 83, no. 11, pp. 2099–2107, 2000.