Journal of The Korea Society of Computer and Information
Vol. 20, No. 5, May 2015
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2015.20.5.053

HMM 기반 TTS와 MusicXML을 이용한 노래음 합성

칸 나지브 울라*, 이 정 철*

Singing Voice Synthesis Using HMM Based TTS and MusicXML

Najeeb Ullah Khan*, Jung-Chul Lee *

요 약

노래음 합성이란 주어진 가사와 악보를 이용하여 컴퓨터에서 노래음을 생성하는 것이다. 텍스트/음성 변환기에 널리 사용된 HMM 기반 음성합성기는 최근 노래음 합성에도 적용되고 있다. 그러나 기존의 구현방법에는 대용량의 노래음 데이터베이스 수집과 학습이 필요하여 구현에 어려움이 있다. 또한 기존의 상용 노래음 합성시스템은 피아노 롤 방식의 악보 표현방식을 사용하고 있어 일반인에게는 익숙하지 않으므로 읽기 쉬운 표준 악보형식의 사용자 인터페이스를 지원하여 노래 학습의 편의성을 향상시킬 필요가 있다. 이 문제를 해결하기 위하여 본 논문에서는 기존 낭독형 음성합성기의 HMM 모델을 이용하고 노래음에 적합한 피치값과 지속시간 제어방법을 적용하여 HMM 모델 파라미터 값을 변화시킴으로서 노래음을 생성하는 방법을 제안한다. 그리고 음표와 가사를 입력하기 위한 MusicXML 기반의 악보편집기를 전단으로, HMM 기반의 텍스트/음성 변환 합성기를 합성기 후단으로서 사용하여 노래음 합성시스템을 구현하는 방법을 제안한다. 본 논문에서 제안하는 방법을 이용하여 합성된 노래음을 평가하였으며 평가결과 활용 가능성을 확인하였다.

▸ Keywords : 텍스트/음성변환, 은닉 마코프 모델, 노래음 합성, 악보편집기

Abstract

Singing voice synthesis is the generation of a song using a computer given its lyrics and musical notes. Hidden Markov models (HMM) have been proved to be the models of choice for text to speech synthesis. HMMs have also been used for singing voice synthesis research, however, a huge database is needed for the training of HMMs for singing voice synthesis. And commercially available singing voice synthesis systems which use the piano roll music notation, needs to adopt the easy to read standard music notation which make it suitable for singing learning applications. To overcome this problem, we use a speech

•제1저자 : 칸 나지브 울라 •교신저자 : 이정철
•투고일: 2015. 4. 9, 심사일 :2015. 5. 10, 게재확정일 :2015. 5. 15.
* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan)

database for training context dependent HMMs, to be used for singing voice synthesis. Pitch and duration control methods have been devised to modify the parameters of the HMMs trained on speech, to be used as the synthesis units for the singing voice. This work describes a singing voice synthesis system which uses a MusicXML based music score editor as the front-end interface for entry of the notes and lyrics to be synthesized and a hidden Markov model based text to speech synthesis system as the back-end synthesizer. A perceptual test shows the feasibility of our proposed system.

▸ Keywords : TTS, HMM, Singing Voice Synthesis, Score Editor

I. Introduction

Singing voice synthesis is the generation of a song using a computer given its lyrics and musical notes. In recent years singing voice synthesizers have gained a lot of popularity. The Vocaloid singing synthesizer [1] developed by Yamaha has been used for commercial purposes by professional musicians [2]. Another such software is UTAU singing synthesizer [3]. In these software the lyrics and music score are input using a piano roll notation. The piano roll notation makes the entry of the music score easier; however, it is difficult to read. We use a standard music notation based score editor, which enable our singing voice synthesis system to be used in a singing training application. To get a new voice in the commercially available synthesizers mentioned above, a large database of singing data is required. Hence there is a need for more flexible approaches to singing voice synthesis. Recently deep learning [4] has led to very promising results in the speech recognition and synthesis however these methods also need a huge amount of data, besides being not very adaptable. The hidden Markov models (HMM) have been proved quite suitable for natural text to speech synthesis, voice conversion and speaker adaptation [5]. An open source toolkit for training and synthesis of speech based on HMMs has also been developed [6]. Figure 1 shows the block diagram of text to speech synthesis system based on HMMs.

In the training phase a labeled speech database is used to train context dependent Hidden semi Markov models, modeling the spectrum, F0 and duration of each context dependent unit. In the synthesis part an input text string is processed by the language processing module to output the sequence of context dependent models and the spectrum and excitation parameters are generated from the HMM parameters using a parameter generation algorithm [7]. Finally the speech is synthesized using a synthesis filter.

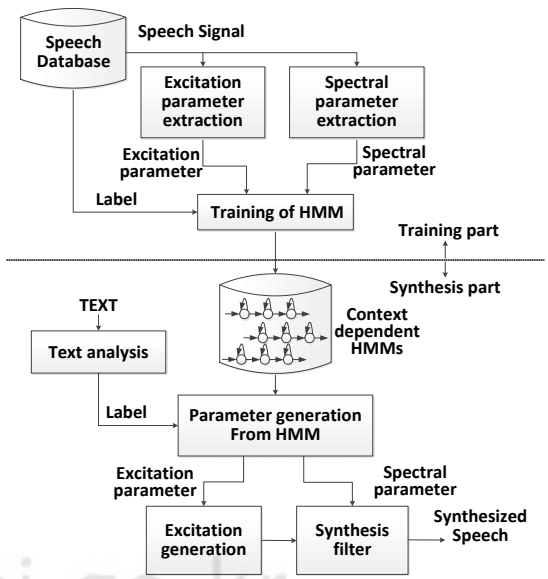

Fig. 1. HMM based text to speech synthesis

In recent years, this approach has been applied many times to the singing voice synthesis problem. In [8-10], the authors have used a singing voice database for training context dependent models. In [11], pitch and speaker adaptation techniques have been used to cope with the limited size of singing voice database. In [12], the authors have attempted to modify a speaking voice based on the lyrics and score to get the singing voice, however the system does not utilizes a speech database.

A large database of labelled data is needed for the training of HMMs for singing voice synthesis. The construction of a large labelled singing voice database is a difficult task. Therefore, in this paper, we use a speech database for training context dependent HMMs, and using these context dependent HMMs in conjunction with the information about pitch and duration from the music score.

The contribution of this paper includes the development of a score editor for singing voice synthesis purpose, implementation of the synthesis part of the HMM based text to speech system and incorporating the duration and pitch information from the music score to generate the parameters for synthesizer.

The rest of the paper is organized as follows. Section II describes the system architecture followed by a description of the front-end score editor in section III. The backend HMM based text to speech system (TTS) is described in section IV. Experimental results are given in section V. Finally conclusions and future work are given in section VI.

II. System Architecture

1. Acoustic Model Training

The singing voice synthesis system is a modification of the HMM based text to speech synthesis system. In this work, the same setup is used for the training part as that used for the text to speech synthesis system shown in figure 1. The open source toolkit HTS [6] was used for the training of the models. A male speaker from the CMU arctic speech database [13] was used as the training database. The database consists of around 1150 utterances. Spectral parameters such as Mel frequency cepstral coefficients and excitation parameters such as log F0 and their dynamic features including delta and delta-delta were used as the observation vectors. The structure of the observation vectors is shown in the figure 2.

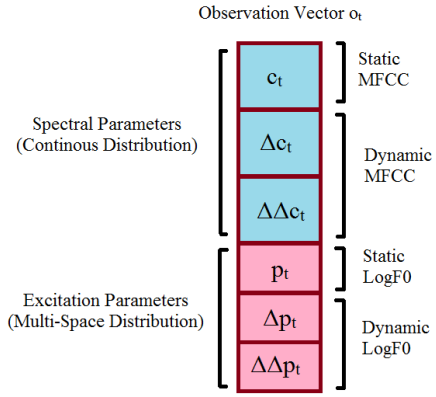

Fig. 2. Observation vector consisting of static and dynamic features [6]

To model the log F0 we cannot use the discrete or continuous density HMMs since F0 has continuous values in the voiced region and undefined values in the unvoiced regions. To model the log F0, multi-space probability distributions[14] are used to solve this problem. Since spectral parameters are modeled using continuous distributions and excitation parameters are modeled using multi-space distributions, to keep synchronization between the two, while using different distributions, a multi-stream HMM is used in which different probability distributions are used for different parts of the observation vector.

In conventional HMMs, the state transitions are determined by the state transition probabilities,

which imply a geometric distribution for state durations however, such a distribution is inappropriate for modelling speech since the probability of state duration decreases exponentially with increase in duration. To deal with this problem, HMMs with explicit state duration distribution modelled by a Gaussian distribution called hidden semi-Markov models were used.

Means and variances for left to right triphone HMMs were estimated in a maximum-likelihood manner using the Baulm-Welch algorithm. Duration models were trained using an efficient algorithm based on the statistics obtained from the Baulm-Welch algorithm [15]. Since the training database is limited, it is not possible to get the robust estimates of the HMM parameters for all the triphones. To solve this problem, state tying was used to cluster similar states in to a single one and robustly estimate the parameters of the clusters. A decision tree clustering method based on the minimum description length (MDL) [16] criterion was used. Since the Mel frequency cepstral coefficients (MFCC), logarithmic fundamental frequency (F0) and duration have different context-dependency, separate clustering is performed for each of these parameters [17].

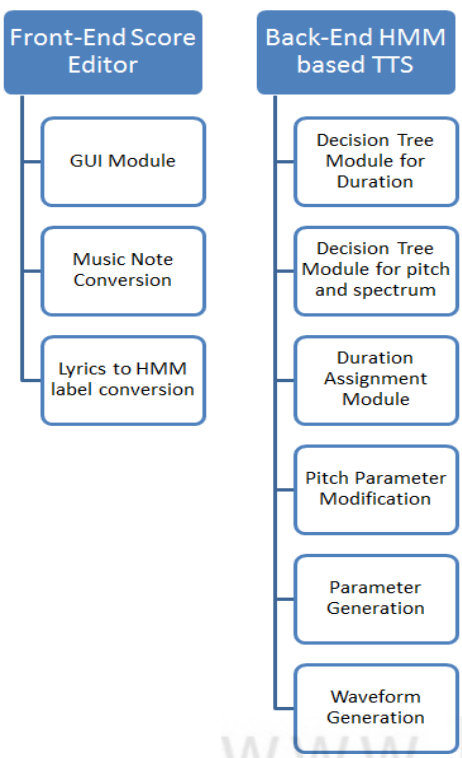

Fig. 3. Architecture of the proposed singing voice synthesis system

2. Singing Voice Synthesis

The synthesis part is implemented entirely in the C# programming language using a modular structure. The synthesis part is composed of a front-end score editor and a back-end synthesizer which utilizes the trained HMMs as shown in figure 3. The modular structure of the system makes the addition of new modules and the modification of the existing modules easier. Each of the modules is described in the subsequent sections.

III. Front-End Score Editor

In the synthesis part shown in figure 3 a score editor based on MusicXML was used as the front-end user interface for the singing voice synthesis system. MusicXML is a digital sheet music interchange and distribution format for common Western music notation based on extensible markup language (XML). The goal of MusicXML is to interchange data between different music software programs such as score editors, optical music recognition software and digital music stands etc [18]. Thus music score created in any MusicXML compatible score editor can be imported into our score editor for singing voice synthesis. The GUI module in the music score editor based on the Microsoft's Windows Presentation Foundation is used to input the musical notes and lyrics. The score editor GUI is shown in figure 4.

Given the music score and lyrics the score editor determines the triphone transcription for each syllable using a dictionary lookup. The Musical note conversion module converts the notes information to

quantities usable by the back-end HMM based TTS. The duration of each note is determined in units of frame length as a function of the note type (i.e. half, quarter etc.), time signature and tempo of the score as follows

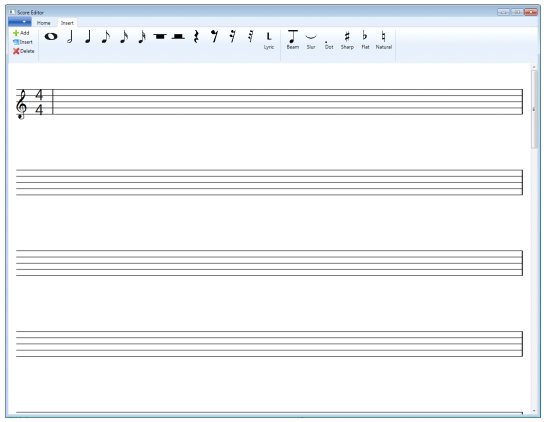

Fig. 4. Music Score Editor

$$D_{(\#\ \text{of frames})} = \frac{samplingRate}{frameLength}\left(\frac{beatType * duration}{4 * divisions}\right)\left(\frac{60}{tempo}\right) \quad (1)$$

where beatType is the type of beat in the denominator of a time signature, duration is the duration of note in units of divisions per quarter note as represented in MusicXML and the tempo is the number of beats per minute. Details of MusicXML representation can be found in [18, 19].

MusicXML uses the step element (A, B, C, D, E, F, or G), an optional alter element (-1 for flat, 1 for sharp), and an octave element (4 for the octave starting with middle C). The MusicXML pitch was converted to MIDI pitch values which were then converted to fundamental frequency using the following formula assuming that A4 is 440Hz

$$f_0 = 2^{(m-69)/12}.440Hz, \quad (2)$$

where m is the MIDI pitch value. For example the middle C has MusicXML value of C4 which is converted to a MIDI value of 60 and using the above formula we get an F0 of 261.6Hz.

Thus the output of the score editor for each note is a sequence of triphone HMM labels, duration for each syllable and the target F0 values.

IV. Back-End HMM Based TTS

1. Decision Tree Modules

A set of clustered states for triphone models and decision trees for the duration, pitch and spectrum are received from the training part of the system, and a sequence of triphone labels to be synthesized are received from the score editor. The decision tree module determines the state's mean and variance for use of the subsequent modules. For duration modeling single state HMMs are used, with each state containing multiple mixtures modeling the duration of the states in the corresponding phoneme models. For pitch and spectrum modeling multiple state HMMs are used. Due to two different HMM topologies for duration and pitch plus spectrum two different tree parsing modules are used.

2. Duration Assignment Module

Before the generation of parameters for a waveform synthesizer, duration assignment to each state is necessary. The duration assignment to states is done in such a way that the duration of each syllable given by the note duration from the front-end score editor is satisfied.

The method described here is a modification of the speech duration control given in [15]. Given a note with duration D and phoneme HMMs corresponding to the note $\lambda_1, \lambda_2, \cdots, \lambda_N$ each with S states and duration probability distribution for model i and state s as p_{is}, the number of frames for each state are determined so as to maximize the log likelihood

$$\log(q_1,\ldots,q_D|\lambda_1,\ldots,\lambda_N,D)=\sum_{i=1}^{N}\sum_{s=1}^{S}\log(p_{is}(d_{is})), \quad (3)$$

with the constraint that the durations which maximize the probability should sum to the total duration of the musical note, i.e.

$$D=\sum_{i=1}^{N}\sum_{s=1}^{S}d_{is}, \quad (4)$$

where d_{is} is the duration of state s and model i. For the case of Gaussian duration models, duration pdf for model i and state s is given by

$$p_{is}(d_{is})=\frac{1}{\sqrt{2\pi\sigma_{is}^2}}e^{-\frac{(d_{is}-\mu_{is})^2}{2\sigma_{is}^2}} \quad (5)$$

and the above constrained optimization problem can be solved using the Lagrange multiplier as follows

$$L(d_{1.1},\ldots,d_{N.S},\lambda)=\sum_{i=1}^{N}\sum_{s=1}^{S}\log p_{is}(d_{is})+\lambda\left(\sum_{i=1}^{N}\sum_{s=1}^{S}d_{is}-D\right) \quad (6)$$

where

$$\log p_{is}(d_{is})=-\frac{1}{2}log(2\pi)-\frac{1}{2}log(\sigma_{is}^2)-\frac{1}{2\sigma_{is}^2}(d_{is}-\mu_{is})^2. \quad (7)$$

Setting derivatives of the Lagrange L to zero with respect to d_{is} and λ we get the following set of equations

$$L_{d_{is}}=-\frac{1}{\sigma_{is}^2}(d_{is}-\mu_{is})+\lambda=0, \quad (8)$$

$$L_{\lambda}=\sum_{i=1}^{N}\sum_{s=1}^{S}d_{is}-D=0. \quad (9)$$

Solving (8) and (9), the durations for each state are given by

$$d_{is}=u_{is}+\rho\sigma_{is}^2, \quad (10)$$

$$\text{where } \rho=\frac{\left(D-\sum_{i=1}^{N}\sum_{s=1}^{S}\mu_{is}\right)}{\sum_{i=1}^{N}\sum_{s=1}^{S}\sigma_{is}^2}. \quad (11)$$

The above optimization gives non-integer state durations and rounding off results in violation of the equality constraint imposed by the note duration. To avoid this problem all the real valued durations are floored to the nearest integers and the remaining frames are uniformly distributed among the states so as to meet the note duration constraint. If the number of frames are not completely divisible by the number of states in the note, the frames are assigned to states in a left to right manner.

3. Pitch Parameter Modification

In singing the pitch value of each syllable or phoneme is dictated by the musical notes with local variations. A simple approach to the pitch modification problem is to use the note's pitch as the mean for all the voiced states of the phonemes belonging to the note. The variances of the states should not be altered so that the parameter generation algorithm can generate pitch contours which have the local variations.

4. Speech Parameter Generation

The parameter generation algorithm generates the spectral and F0 contour given the modified HMM parameters using maximum likelihood given the dynamic feature constraints [7].

Given the durations of each state of the models, assigned by the duration control module, and considering the fact that we are dealing with a left to right topology HMMs, the state sequence Q is already known to the parameter generation algorithm. The problem of parameter generation for a model λ is then that of finding static parameter

vector sequence C which maximizes the log probability of C given the state sequence Q, i.e.

$$C = \arg\max \log P(C|Q,\lambda). \tag{12}$$

Given the Gaussian observation distribution the above maximization correspond to choosing the mean vector of each state as the observation vector. However, in real speech the parameters are slowly changing thus we need to maximize the above probability subject to dynamic feature constraints, giving

$$\triangle c_t = \sum_{\tau=-1}^{1} w^{(1)}(\tau) c_{t+\tau}, \tag{13}$$

$$\triangle^2 c_t = \sum_{\tau=-1}^{1} w^{(2)}(\tau) c_{t+\tau}. \tag{14}$$

where c_t is the static feature vector. Representing (13) and (14) in matrix form W, the observation sequence O including both the static and dynamic features can be written as

$$O = WC. \tag{15}$$

The maximization problem can be re-written in terms of O as

$$C = \arg\max \log P(O|Q,\lambda), \tag{16}$$

$$C = \arg\max \log P(WC|Q,\lambda). \tag{17}$$

The log probability is given by

$$\log P(O|Q,\lambda) = -\frac{1}{2}[WC]^T U^{-1}[WC] + [WC]^T U^{-1} M + K, \tag{18}$$

where U is the covariance matrix sequence, M is the mean sequence vector and K is a constant. Setting the derivative of (18) with respect to C equal to zero we get

$$W^T U^{-1} WC = W^T U^{-1} M^T. \tag{19}$$

To get the generated parameter vector C of size M X 1 we need to find the inverse of 3M X 3M matrix on the left. However the matrix W is a sparse matrix consisting of non-zero values in a band and hence band Cholesky decomposition[20] was used to implement the matrix inverse.

5. Waveform Generation

Based on the generated spectral and excitation parameters and the durations for each state of the HMM, a pulse train is generated for voiced frame and Gaussian noise is generated for unvoiced frames by the excitation generator. A Mel log spectral approximation (MLSA) filter with pade order of 4 and with the MFCC parameters provided by the parameter generation algorithm is used to synthesize a speech waveform. The MLSA filter transfer function can be given as

$$D(z) = \exp\left(\sum_{m=0}^{M} b(m)\Phi_m(z)\right) \tag{20}$$

where

$$\Phi_m(z) = \begin{cases} 1, & \text{for } m = 0 \\ \dfrac{(1-\alpha^2)z^{-1}}{1-\alpha z^{-1}} \widetilde{Z^{-(m-1)}}, & \text{for } m \geq 1 \end{cases} \tag{21}$$

$$\tilde{Z}^{-1} = \frac{z^{-1}-\alpha}{1-\alpha z^{-1}}. \tag{22}$$

The transfer function of the MLSA filter can be realized as shown in figure 5.

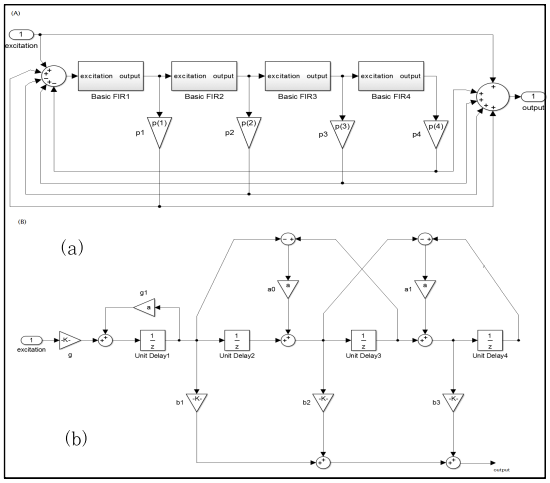

Fig. 5. (a) MLSA Filter structure (b) Basic filter

Table 1. System Parameters

Parameter	Value
Speaker Database	CMU_arctic_rms
Sampling Rate	16000 samples/sec
Frame Length	400 samples
Frame Shift	80 samples
MFCC Order	34
Frequency Warping	0.42
Delta win $w^{(1)}$	-0.5, 0.0, 0.5
Delta-Delta win $w^{(2)}$	1.0, -2.0, 1.0
States per HMM	5

V. Experimental Results

The parameters used in the experiment are listed in table 1. A speaker in the CMU speech database named "rms" was used for training 5 state HMMs using the methods described in section II-1. A frame length of 400 samples was used to calculate 34 MFCC coefficients using the Mel-generalized cepstral analysis method [21] with no pole weighting i.e. $\gamma=0$ and the frequency warping coefficient of 0.42. First and second order dynamic features were calculated for MFCC and log F0. The observation vector consisting of static and dynamic features of 34 MFCC and 1 log F0 coefficient had a length of 105.

After training the models, musical notes and lyrics for a test song "twinkle twinkle little star" were input into the score editor as shown in figure 6. The musical tempo was set to 100 beats per minute, with a 4/4 time signature. These specifications lead to a quarter note being equal to 120 frames in duration. State durations generated using the proposed method are shown in table 2 for the first two notes of the song. As can be seen the sum of all the state's durations belonging to the phonemes of a note equals the duration of the note.

The generated F0 contours for the first four notes are shown in figure 7. Figure 7 also shows the notes pitch values and the generated F0 contour without any modification. Figure 8 shows the waveform and spectrogram of the natural singing voice for the lyrics "twinkle twinkle little star". The corresponding waveform and spectrogram of the singing voice synthesized using an MLSA filter is shown in figure 9. The natural and synthesized waveform differ in spectrum, however, they match in durations.

A mean opinion score (5:excellent, 4: good, 3:fair, 2:poor, 1:bad) test was performed. Ten people were asked to rate two songs generated

Table 2. State duration assignment

syllable/note	phonemes	s1	s2	s3	s4	s5	phone duration	syl/note Duration
twin/quarter	t	5	4	10	6	7	32	120
	w	7	5	6	4	7	29	
	ih	20	6	8	2	5	41	
	ng	4	4	4	3	3	18	
kle/quarter	k	4	5	5	4	6	24	120
	ax	4	4	4	5	4	21	
	l	5	14	3	2	10	34	
	pau	5	11	17	5	3	41	

from lyrics and music score using the proposed system. The result of the test is shown in table 3. As can be seen in table 3, the perception test showed comparatively good quality singing voice given the simplicity of the algorithms.

Table 3. MOS test results

Song	MOS
1	3.22
2	3.04

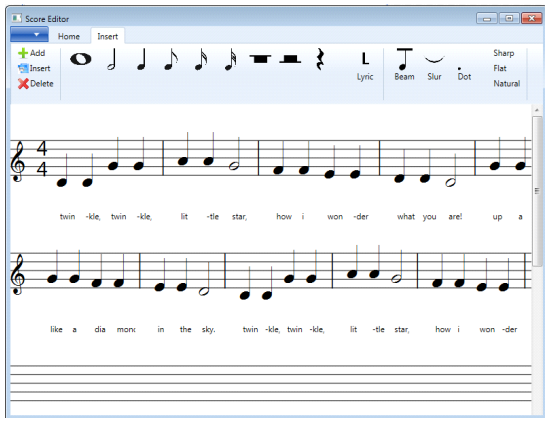

Fig. 6. Notes and lyrics for the test song "twinkle twinkle"

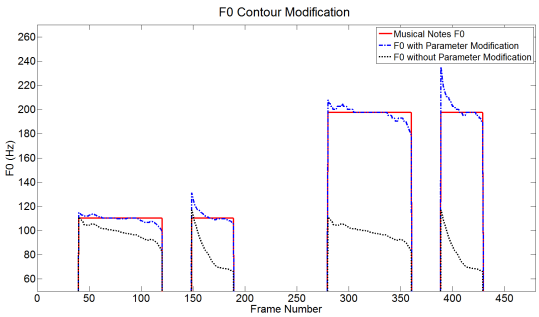

Fig. 7. Pitch contours generated for speech and singing voice

VI. Conclusions and Future Work

In this work, we proposed a TTS based singing synthesizer using spoken speech database. HMM models trained on speech data have been integrated with a score editor to turn a TTS system into singing voice synthesis system. The score editor is based on the traditional music notation which is easy to read. Simple algorithms such as maximum likelihood based state duration assignment and replacement of speech's pitch model's mean values with the note's pitch were used.

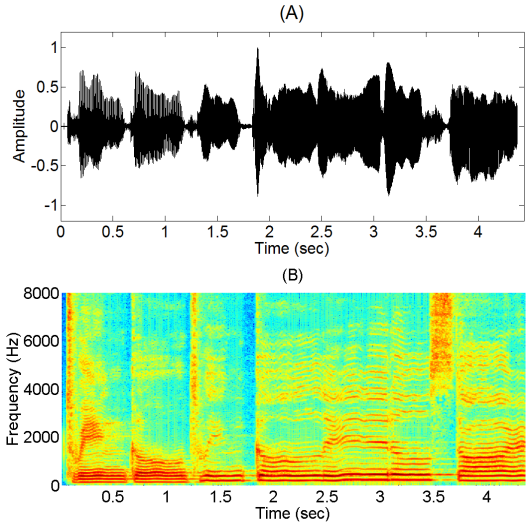

Fig. 8. Natural singing voice (A) waveform, (B) spectrogram

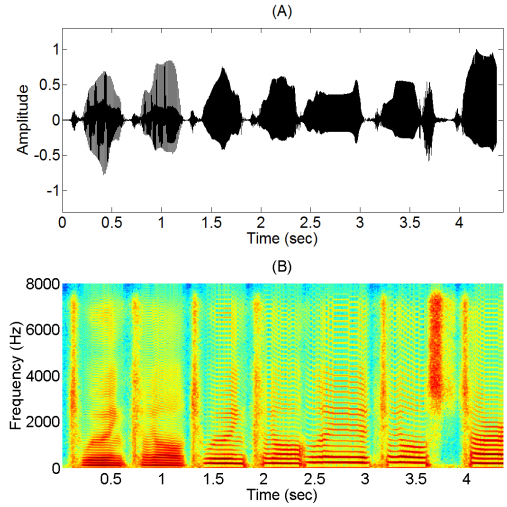

Fig. 9. Synthesized singing voice using MLSA filter (A) waveform, (B) spectrogram

In future more sophisticated duration and pitch control algorithms can be used to transform the speech trained models to be used as the synthesis units for singing voice. Natural singing voice can be analyzed to derive heuristics for the control algorithms. Along with pitch and duration, the spectrum modification can also be added. Being based on a layered architecture the score editor can be extended to provide a graphical interface for manual correction of the pitch contours.

VII. Acknowledgment

This work was supported by the research fund of the University of Ulsan, Ulsan, South Korea.

REFERENCES

[1] H. Kenmochi and H. Ohshita, "VOCALOID -commercial singing synthesizer based on sample concatenation," in *Proc. INTERSPEECH*, pp. 4009-4010, 2007.

[2] H. Kenmochi, "Singing synthesis as a new musical instrument," in *Proc. ICASSP*, pp. 5385-5388, 2012.

[3] "UTAU" available: *http://utau-synth.com/*

[4] J. Xu, H. Li, and S. Zhou, "An Overview of Deep Generative Models," *IETE Technical Review*, pp. 1-9, 2014.

[5] G.J. Lim and J.C. Lee, "Improvement of Naturalness for a HMM-based Korean TTS using the prosodic boundary information," *Journal of the Korea Society of Computer and Information*, v.17, no.9, pp. 75-84, Sep. 2012.

[6] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, et al., "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. ISCA Workshop Speech Synthesis*, pp. 294-299, 2007.

[7] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, pp. 1315-1318, 2000.

[8] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system," in *Proc. INTERSPEECH*, 2006.

[9] K. Oura, A. Mase, T. Yamada, S. Muto, Y. Nankaku, and K. Tokuda, "Recent development of the HMM-based singing voice synthesis system-Sinsy," in *Proc. ISCA Workshop Speech Synthesis*, pp. 211-216, 2010.

[10] K. Nakamura, K. Oura, Y. Nankaku, and K. Tokuda, "HMM-Based singing voice synthesis and its application to Japanese and English," in *Proc. ICASSP*, pp. 265-269, 2014.

[11] K. Shirota, K. Nakamura, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Integration of speaker and pitch adaptive training for HMM-based singing voice synthesis," in *Proc. ICASSP*, pp. 2559-2563, 2014.

[12] T. Saitou, M. Goto, M. Unoki, and M. Akagi, "Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices," in *Applications of Signal Processing to Audio and Acoustics*, pp. 215-218, 2007.

[13] J. Kominek and A. W. Black, "The CMU Arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, pp. 223-224, 2004.

[14] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Multi-space probability distribution HMM," *IEICE TRANSACTIONS on Information and Systems*, vol. 85, pp. 455-464, 2002.

[15] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Duration modeling for HMM-based speech synthesis," in *Proc. ICSLP*, pp. 29-31, 1998.

[16] K. Shinoda and T. Watanabe, "MDL-based context-dependent subword modeling for speech recognition," *The Journal of the Acoustical Society of Japan* (E), vol. 21, pp. 79-86, 2000.

[17] T. YoshimuraÝ, K. TokudaÝ, T. MasukoÝÝ, T.

KobayashiÝÝ, and T. KitamuraÝ, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," in *Proc. Eurospeech*, pp. 2347-2350, 1999.

[18] "MusicXML" available: http://www.musicxml.com/

[19] N. U. Khan and J.C. Lee, "Development of a Music Score Editor based on MusicXML," *Journal of the Korea Society of Computer and Information*, vol. 19, pp. 77-90, 2014.

[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C*, vol. 2: Citeseer, 1996.

[21] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis-a unified approach to speech spectral estimation," in Proc. *ICSLP*, pp. 1043-1046, 1994.

저 자 소 개

Najeeb Ullah Khan

2012: B.S. from International Islamic University, Islamabad

2013~Now: Master Course University of Ulsan

Interesting area: Digital signal processing, Speech recognition/synthesis,

Email: electronicengr@gmail.com

Jung Chul Lee

1984: B.S. from Dept of Electronic Engineering, Seoul National University

1988: M.S. from Dept of Electronic Engineering, Seoul National University

1998: Ph.D. from Dept of Electronic Engineering, Seoul National University

Now : Associate Professor at University of Ulsan

Interesting area: Digital signal processing, Speech signal processing/synthesis,

Email: jungclee@ulsan.ac.kr