JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 303-317 (2011)

Mandarin Singing-voice Synthesis Using an HNM Based Scheme[*]

HUNG-YAN GU AND HUANG-LIANG LIAO
Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 106 Taiwan

In this paper, HNM (harmonic plus noise model) is enhanced and used to design a scheme for synthesizing a Mandarin Chinese singing voice. Enhancements made include a Lagrange-interpolation based estimation of spectral envelope, piecewise linear mapping of time axes, fixed-pace placement of control points, and other modifications for analyzing HNM parameters and efficient execution. In terms of the enhancements and the signal-synthesis equations rewritten here, a Mandarin singing-voice synthesis system is built. In the system, each Mandarin syllable is recorded just once for analyzing HNM parameters. Then, the HNM parameters of a source syllable are used to synthesize singing syllables of diverse pitches and durations. This system can parse a song score file and synthesize its lyric syllables' signals in real-time. Also, the skill of portamento (pitch gliding) singing is implemented. According to the perception tests, our system can indeed synthesize signals of singing voice that are consistent in timbre, of no reverberation, and much clearer than a PSOLA (pitch synchronous overlap add) based scheme.

Keywords: singing-voice synthesis, harmonic-plus-noise model, spectral envelope, timbre consistency, reverberation

1. INTRODUCTION

Currently, a specially designed robot can play a musical instrument (*e.g.*, piano or trumpet) as well as a human can play. It would be attractive if a humanoid robot (of facial expression) could sing a song as expressive as one sung by a real singer. Toward this goal, we began to study the synthesis of a singing voice. Recently, our singing-voice synthesis system was integrated into a two-wheel robot [8]. Nevertheless, our synthesis system was just in its early phases because we had expended more effort on signal quality than on expressiveness when developing this system. Used here, signal quality actually means signal clarity. That is, a signal that is less noisy and less reverberant is better in quality. We think that the synthesis of a high signal-quality singing voice is an essential basic step. Our reason is that performance rules [17] can be further used and singing expressions such as vibrato and huskiness [12, 17, 22] can be further added to the clean synthetic signal.

Several techniques have been proposed to synthesize instrumental music signals, including additive synthesis, subtractive synthesis, FM (Frequency Modulation) synthesis, along with others [3, 13]. As to the synthesis of a singing voice, the techniques developed include phase vocoder [3, 13], formant synthesis [3, 13], linear-prediction based source-

Received January 5, 2009; revised April 21 & June 22, 2009; accepted August 13, 2009.
Communicated by Chin-Teng Lin.

[*] This research was partially supported by National Science Council of Taiwan under Grant No. NSC-95-2218-E-011-009.

filter model [3, 7, 13], sinusoidal model [10], EpR (Excitation plus Resonances) model [1, 2], PSOLA synthesis [18], corpus-based synthesis [2, 9, 11], *etc*. In this paper, however, we choose to enhance HNM (harmonic plus noise model) in the hope of obtaining higher signal quality and fluency levels. Based on the enhanced HNM, we design a signal-synthesis scheme to implement a Mandarin Chinese singing-voice synthesis system. HNM is proposed by Y. Stylianou [20, 21] and is thought to belong to the class of additive synthesis. It splits the spectrum of a signal frame into two halves of unequal widths in order to better model the spectrum. The lower frequency half is modeled as consisting of harmonic partials while the higher frequency half is modeled as consisting of noise signal components.

Although the Mandarin Chinese language is adopted here for studying singing-voice synthesis, the technique developed should be applicable to other languages, especially to syllable-prominent languages. Mandarin is a syllable-prominent language, and has only 408 different syllables if the superimposed tones are not distinguished. Each syllable of Mandarin is of the structure, C_xVC_n. The initial, C_x, may be null, a voiced consonant, or an unvoiced consonant while the final, C_n, can just be null or a nasal /n/ or /ng/. As to the nucleus, V, it may be a vowel, diphthong, or triphthong. Therefore, we take syllable as the unit for synthesis processing.

In this paper, the HNM based scheme is depicted in Fig. 1. First, a note's data is input and parsed. Then, the comprising phonemes of the note's lyric syllable are planned for their durations. In terms of these durations, a time-axis mapping function can be constructed. More detailed explanation of this is given in section 3.1. Next, a pitch contour for the note is computed. If the lyric syllable is to be sung in portamento (pitch gliding), the computation of the pitch contour becomes more complicated. This will be explained in section 3.6. In addition, to keep the timbres of synthetic syllables consistent, the HNM parameters of a lyric syllable must be adjusted in an appropriate way. This will be explained in sections 3.2 and 3.3. In the last block of Fig. 1, the signals for the unvoiced and voiced segments of a lyric syllable are synthesized with an enhanced HNM.

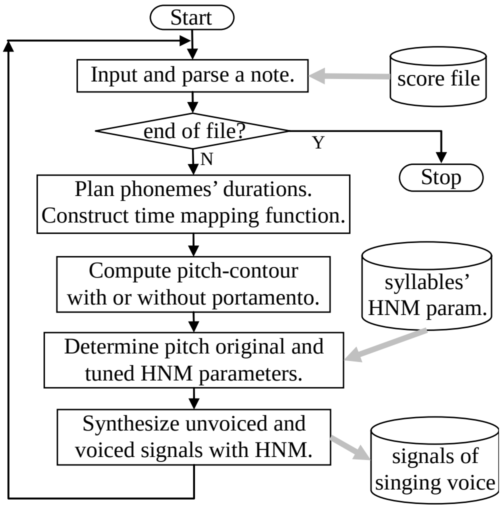

Fig. 1. Main flow of the HNM based synthesis scheme.

If the C_x segment of a syllable is a short unvoiced consonant, *e.g.* /b/, /d/, its synthetic signal will be copied directly from the corresponding segment in the recorded syllable. Here, when we write /b/, /d/, we refer to the unvoiced and non-aspirated Mandarin versions of these phonemes. If the C_x segment is a long unvoiced consonant, *e.g.* /s/, /p/, its synthetic signal will be generated as noise signal with HNM. Otherwise, the C_x is a voiced consonant (*e.g.* /m/, /r/) and will be considered together with the remaining phonemes. As the remaining phonemes of a syllable are all voiced, their synthetic signal will be generated as harmonic partials plus noise signals with HNM. The methods for synthesizing harmonic and noise signals are explained in sections 3.4 and 3.5, respectively.

2. PARSING OF SCORE FILE

For our singing voice synthesis system, a special file format is used to store a song's score. In detail, each line of the score file except the first line contains a musical note's information, *i.e.*, pitch symbol, number of beats, and lyric syllable. The information in the first line consists of song name, tempo (*e.g.*, 120 means 120 beats per minute), and duty ratio (*e.g.*, 85 means 85% of a note's duration is used for singing). Parsing is used to slice out the 3 fields in a line and to interpret the meanings of these fields.

The pitch symbol of a note is of the format "XYZ" (*e.g.* G3#). "X" denotes the tone name, "Y" denotes the tone range, and "Z" is "#" (sharp) or "b" (flat). Through interpretation, the pitch symbol is converted to a numeric value of pitch frequency. After all notes' pitch frequencies are determined, automatic key shifting is performed. Key shifting must be done in order to translate the pitch range of the score file to match the pitch range of the person who utters the Mandarin syllables used as the synthesis units. Thus, the key of the synthetic song is not the original key defined in the score file. Here, automatic key shifting is done in the following steps: (a) find the maximum and minimum values from the notes' pitch frequencies; (b) take the average of the maximum and minimum values found; (c) compute the ratio of the person's mean pitch frequency to the average value computed in the previous step; (d) multiply each note's pitch frequency with the ratio computed.

After the number of beats for a note is parsed, the tempo value in the first line can be used to compute the time length of a note. A note, however, is usually not sung in its full duration because some small ratio of this duration is reserved for breathing or transiting to its following note. Next, the lyric syllable of a note is parsed. Usually, each note has a unique lyric syllable assigned to it. Sometimes, though, two or three consecutive notes may be assigned the same lyric syllable. This means that the syllable should be sung in portamento. This situation is indicated in the score file with a convention. When a note is to be assigned the same lyric syllable as its preceding note, the third field for this note will contain a special character such as "|".

3. SYNTHESIS OF SIGNAL WAVEFORM

Note that each Mandarin Chinese syllable has only one recorded utterance here. Therefore, it becomes important that the timbres of synthetic syllables are kept consistent. The solution for keeping timbre consistent can be found in the literature [3, 19]. Accurate estimation of spectral amplitude envelope is the key point. Nevertheless, the amplitude envelope estimation method adopted in the original HNM [20, 21] is a complicated global approximation method that may not be accurate enough. If the spectral envelope is estimated with sufficient accuracy, the values of a syllable's HNM parameters can then be adequately adjusted to obtain consistent timbre when the pitch frequency of the syllable is changed. In addition, note that a syllable's duration needs to be lengthened or shortened according the number of beats assigned to its corresponding note, and linear time warping usually results in lower perceived fluency. Therefore, how to warp the time axis of a synthetic syllable in order that a more fluent syllable signal can be synthesized is also a problem. A solution method to this problem, however, is not found in the original HNM. Furthermore, direct copying of HNM parameters from an analysis frame to a control point [3, 13] is adopted in original HNM. Here, in contrast, we determine the HNM parameter values for a control point by interpolation.

3.1 Planning of Phoneme Duration

When a syllable begins with a short unvoiced phoneme, *e.g.* /bau/, the time length of the synthetic short-unvoiced phoneme is planned as its source length in the recorded syllable. In contrast, when started with a long unvoiced phoneme, the length of the synthetic long-unvoiced phoneme is planned by multiplying its source length with a factor *Fu*. *Fu* is computed as the synthetic syllable's length divided by the recorded syllable's length. Its value, however, is confined to within the range from 0.6 to 1.2. The values, 0.6 and 1.2, are determined empirically. They are used to reserve a minimum duration to have the phoneme perceived, and limit the maximum duration to mimic a real singer's duration planning for a long unvoiced phoneme.

For the planning of voiced phonemes' durations, consider the syllable, /man/, for example. Suppose in the recorded signal of /man/, the three phonemes, /m/, /a/, and /n/, occupy *Rm*, *Ra*, and *Rn* ms, respectively, and $Rv = Rm + Ra + Rn$. Also, suppose that *Dm*, *Da*, and *Dn* represent the time lengths of the three phonemes within the synthetic syllable, and $Dv = Dm + Da + Dn$. First, we let the initial values of *Dm* and *Dn* be $0.85 \times Dv \times (Rm/Rv)$ and $0.85 \times Dv \times (Rn/Rv)$, respectively. Next, we plan the values of *Dm*, *Da*, and *Dn* with a procedure [6] that iteratively decreases the values of *Dm* and *Dn*, and increases the value of *Da* till the ratio, Da/Dv, is greater than a defined value (*e.g.* 0.5). This procedure is designed according to the observation that the consonant-to-vowel duration ratio will become smaller when the syllable is sung within a song than uttered in isolation. Note that in recording the signals of Mandarin syllables, each syllable is uttered in isolation. Another phenomenon created by recording syllables in isolation is that the durations, *Rm* and *Rn*, of a recorded syllable may differ largely but the synthetic durations, *Dm* and *Dn*, should be planned to have about equal lengths.

After the values of *Dm*, *Da*, and *Dn* are determined, a mapping function from the phonemes in the synthetic syllable to the corresponding phonemes in the recorded syllable can be established. This mapping function is shown by the solid line depicted in Fig. 2, *i.e.* a piecewise linear time warping function. According to the constructed mapping function, a control point placed on the time-axis of a synthetic syllable can then be mapped to locate its corresponding analysis frames in the recorded syllable. For example, the dashed lines in Fig. 2 represent the mappings from synthetic-phoneme boundaries to their corresponding source-phoneme boundaries.

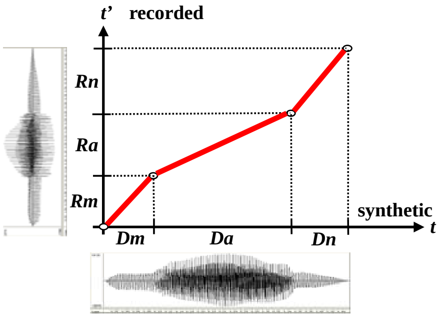

Fig. 2. A piecewise linear time mapping function.

In this paper, each source syllables is recorded at a sampling rate of 22,050Hz. In analyzing HNM parameters, frame size is set to 512 sample points (23.2ms) and frame shift is set to 256 sample points. In synthesizing a singing voice, however, the concept of "control point" [3, 13] is adopted. The term "control point" is used instead of "frame" because the HNM parameters for a control point within the synthetic voiced segment are obtained by interpolating the parameters from its two corresponding analysis frames, *i.e.* not directly copying parameters from an analysis frame into a control point (note that original HNM uses direct copying). The details of the interpolation method are described in section 3.2. Here, in the synthetic voiced segment, adjacent control points are always placed 100 sample points (4.5ms) apart. A fixed pace, 100 sample points, is adopted because more accurate control of spectrum progression is intended. For example, consider the situation where a syllable of a diphthong kernel is to be synthesized with pitch gliding and under piecewise linear time mapping. As to the number 100, it is selected by trading off accurate spectral envelope with computation burden. In contrast, when synthesizing the long-unvoiced segment, the HNM parameters of a control point are obtained by just copying the HNM parameters from one corresponding analysis frame within the recorded unvoiced segment.

3.2 Pitch-original HNM Parameters

To determine the HNM parameter values for a control point within the synthetic voiced segment, the first step is to do time-position mapping according to the constructed mapping function as shown in Fig. 2. Suppose a control point's time position, *ts*, on the synthetic time axis is mapped to t_r frames on the time axis of the recorded syllable. Then, we use the HNM parameters analyzed from the two frames numbered $\lfloor t_r \rfloor$ and $\lfloor t_r \rfloor + 1$ to interpolate HNM parameters for the control point. Currently, we do the interpolation in a linear way:

$$\overline{A}_i = (1-w)\times A_i^n + w\times A_i^{n+1}, n = \lfloor t_r \rfloor, w = t_r - n, \tag{1}$$

$$\overline{F}_i = (1-w)\times F_i^n + w\times F_i^{n+1}, \tag{2}$$

$$\overline{\theta}_i = w\times(\hat{\theta}_i^{n+1} - \theta_i^n) + \theta_i^n, \tag{3}$$

where A_i^n, F_i^n, and θ_i^n denote the amplitude, frequency, and instantaneous phase of the *i*th harmonic partial in the *n*th analysis frame, and $\overline{A}_i$, $\overline{F}_i$, and $\overline{\theta}_i$ denote the amplitude, frequency, and instantaneous phase of the *i*th harmonic partial for the control point. Note that, in Eq. (3), $\hat{\theta}_i^{n+1}$ represents the unwrapped phase of θ_i^{n+1} versus θ_i^n, *i.e.* $\hat{\theta}_i^{n+1} = puw(\theta_i^{n+1}, \theta_i^n, F_i^{n+1}, F_i^n)$. The phase θ_i^{n+1} is unwrapped in order that the phase difference is controlled to within the range from $-\pi$ to π. Here, the phase unwrapping is done as:

$$puw(\theta_i^{n+1},\ \theta_i^n,\ F_i^{n+1},\ F_i^n) = \theta_i^{n+1} - 2\pi\left(\frac{F_i^n + F_i^{n+1}}{2}\right)\frac{256}{22{,}050} - M \times 2\pi, \tag{4}$$

$$M = \left\lfloor \frac{1}{2\pi} \times \left(\theta_i^{n+1} - 2\pi\left(\frac{F_i^n + F_i^{n+1}}{2}\right)\frac{256}{22{,}050} - \theta_i^n + \theta_c \right) \right\rfloor,$$

$$\theta_c = \begin{cases} \pi, & \text{if } \theta_i^{n+1} - 2\pi\left(\frac{F_i^n + F_i^{n+1}}{2}\right)\frac{256}{22{,}050} \geq \theta_i^n \\ -\pi, & \text{otherwise} \end{cases},$$

where 256 is the frame shift in sample points and 22,050 is the sampling frequency. Since θ_i^n and θ_i^{n+1} are analyzed from adjacent signal frames, the extra accumulated phase must also be considered, which is estimated here as $256/22{,}050 \times 2\pi(F_i^n + F_i^{n+1})/2$.

In original HNM, the noise signal components are represented with 10 cepstrum coefficients [15, 20]. Therefore, for each control point, 10 cepstrum coefficients should be derived. Here, the cepstrum coefficients from the two mapped analysis frames are linearly interpolated to derive the cepstrum coefficients for the control point.

3.3 Pitch-tuned HNM Parameters

After the parameters $\overline{A}_i$, $\overline{F}_i$, and $\overline{\theta}_i$ for pitch-original harmonic partials are computed, the parameters $\tilde{A}_k, \tilde{F}_k$ and $\tilde{\theta}_k$ for pitch-tuned harmonic partials should be computed on each control point within the voiced segment. Note that the pitch-height defined by $\overline{F}_i$, $i = 1, 2, \ldots$, is the original pitch predetermined in recording time. Thus, the pitch-height of a control point must be tuned in order to follow the pitch height defined by the corresponding music note. For example, let the pitch defined by the harmonic frequencies, $\overline{F}_i$, be 100Hz, and we need a pitch-height of 150Hz. Apparently, a simple tuning method is to set the values of $\tilde{A}_k, \tilde{F}_k$ and $\tilde{\theta}_k$ as $\tilde{F}_k = \overline{F}_k \cdot 150/100$, $\tilde{A}_k = \overline{A}_k$, and $\tilde{\theta}_k = \overline{\theta}_k$. This is illustrated in Fig. 3. From this figure, it can be seen that the pitch can indeed be tuned from 100Hz to 150Hz. The formant frequencies, however, are also scaled up. For example, the first formant is shifted from 240Hz on the solid-line spectral curve to 360Hz on the dashed-line spectral curve in Fig. 3. The shifting of formant frequencies will cause the timbre be distinctly changed. As a result, the timbre of a synthetic syllable will not be consistent and will vary with the scaling factors (*e.g.* 150/100) set for different control points.

To have consistent timbre, one principle is to keep the spectral envelope unchanged [3]. This implies that the amplitude $\tilde{A}_k$ of the pitch-tuned harmonic partial located at frequency $\tilde{F}_k$ (*i.e.*, *k* times of the pitch frequency to be tuned) must be computed according to an estimated spectral envelope. Therefore, it is important to accurately estimate the spectral-envelope curve. In the past, a few solution methods for this problem have been

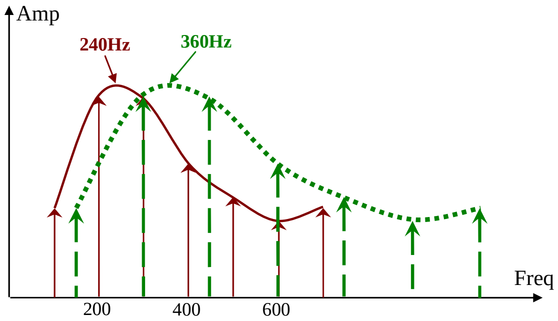

Fig. 3. Pitch tuning with spectral envelope scaled simultaneously.

proposed [16, 19]. Here, considering the two factors of efficient computing and sufficient accuracy, we decide to estimate the spectral envelope by Lagrange interpolating the sequence of pairs, $(\bar{F}_i, \bar{A}_i)$ *i.e.* a local approximation method. In detail, for the *k*th harmonic frequency $\tilde{F}_k$, we first find a pitch-original harmonic frequency $\bar{F}_j$, from $\bar{F}_1$, $\bar{F}_2$, $\bar{F}_3, \ldots$, that is nearest to and less than $\tilde{F}_k$. Then, the four pitch-original harmonic partials of the frequencies, $\bar{F}_{j-1}, \bar{F}_j, \bar{F}_{j+1}$, and $\bar{F}_{j+2}$, are used to perform order-three Lagrange interpolation [4] to compute the value of $\tilde{A}_k$:

$$\tilde{A}_k = \sum_{m=j-1}^{j+2} \bar{A}_m \times \prod_{\substack{h=j-1 \\ h\neq m}}^{j+2} \frac{\tilde{F}_k - \bar{F}_h}{\bar{F}_m - \bar{F}_h}. \tag{5}$$

As to the order of interpolation, the sound synthesized by order-two interpolation is perceived to be slightly less delicate. Therefore, order-three interpolation is chosen.

An illustration of this method of pitch tuning without changing spectral envelope is shown in Fig. 4. In this figure, the pitch is scaled up by a factor of 1.25 but the timbre is preserved. Similarly, the phase $\tilde{\theta}_k$ of the pitch-tuned harmonic partial located at frequency $\tilde{F}_k$ can also be interpolated with the four pitch-original partials of frequencies, $\bar{F}_{j-1}$, $\bar{F}_j$, $\bar{F}_{j+1}$, and $\bar{F}_{j+2}$. The phases of the four partials, $\bar{\theta}_{j-1}, \bar{\theta}_j, \bar{\theta}_{j+1}$, and $\bar{\theta}_{j+2}$, however, must be unwrapped beforehand to prevent phase discontinuities. That is, the unwrapped phases,

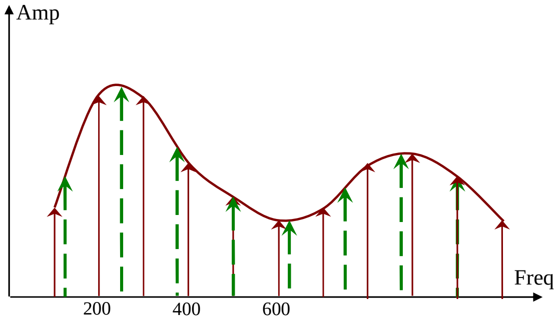

Fig. 4. Spectral envelope is kept while tuning pitch.

$\hat{\theta}_{j-1} = \bar{\theta}_{j-1}$, $\hat{\theta}_j = puw(\bar{\theta}_j, \hat{\theta}_{j-1}, 0, 0)$, $\hat{\theta}_{j+1} = puw(\bar{\theta}_{j+1}, \hat{\theta}_j, 0, 0)$ and $\hat{\theta}_{j+2} = puw(\bar{\theta}_{j+2}, \hat{\theta}_{j+1},$ $0, 0)$, are used instead in the interpolation processing.

3.4 Synthesis of Harmonic Signal

For the harmonic signal, $H(t)$, between the nth and $(n + 1)$th control points, its sample values are computed with these equations (rewritten by us):

$$H(t) = \sum_{k=0}^{L} a_k^n(t) \times \cos(\phi_k^n(t)),\ t = 0, 1, \dots, 99, \tag{6}$$

$$a_k^n(t) = \tilde{A}_k^n + \frac{t}{100}(\tilde{A}_k^{n+1} - \tilde{A}_k^n), \tag{7}$$

$$\phi_k^n(t) = \phi_k^n(t-1) + 2\pi \times f_k^n(t)/22{,}050,\ \ \phi_k^n(0) = \hat{\theta}_k^n, \tag{8}$$

$$f_k^n(t) = \tilde{F}_k^n + \frac{t}{100} \times (\tilde{F}_k^{n+1} - \tilde{F}_k^n), \tag{9}$$

where L is the number of harmonic partials, 100 is the number of samples between the nth and $(n + 1)$th control points, 22,050 is the sampling rate, $a_k^n(t)$ is the time-varying amplitude of the kth partial at time t from the start of the nth control point, $\phi_k^n(t)$ is the cumulated phase for the kth partial, $f_k^n(t)$ is the time-varying frequency for the kth partial, and $\hat{\theta}_k^n$ = $puw(\tilde{\theta}_k^n, \hat{\theta}_k^{n-1}, 0, 0)$, *i.e.* unwrapped phase of $\hat{\theta}_k^n$ versus $\hat{\theta}_k^{n-1}$. In Eqs. (7) and (9), linear interpolation is used, which seems sufficient according to perception testing.

According to Eq. (9), the instantaneous harmonic frequency is interpolated in a linear manner. Therefore, the cumulated phase at the boundary time point, $\phi_k^n(100)$, would not be continuous to the initial phase of the next control point. These kinds of discontinuities, *i.e.* $\phi_k^n(100) \neq \phi_k^{n+1}(0)$, will induce amplitude discontinuities to signal waveform, and cause clicks to be heard. To avoid these kinds of discontinuities, a basic method has been adopted in the literature [20, 21]. That is, the amount of mismatched phase, ξ_k^n, at the boundary point, t = 100, is computed beforehand. Then, this amount is divided and shared among the 100 sample points between two adjacent control points. Accordingly, the phases of the signal samples (especially those around the boundary point) will advance smoothly. Here, we compute the amount of mismatched phase as

$$\xi_k^n = puw(\phi_k^n(100),\ \phi_k^{n+1}(0),\ 0,\ 0) - \phi_k^{n+1}(0) \tag{10}$$

where the phase unwrapping function, $puw(x, y, 0, 0)$, is as defined in Eq. (4). According to our derivation $\phi_k^n(100)$ can be directly computed as

$$\phi_k^n(100) = \phi_k^n(0) + \frac{2\pi}{22{,}050}\left(\frac{101}{2} \times \tilde{F}_k^{n+1} + \frac{99}{2} \times \tilde{F}_k^n\right). \tag{11}$$

The formula of Eq. (11) is obtained by recursively evaluating Eqs. (8) and (9). Then, by dividing and sharing ξ_k^n with the samples between two adjacent control points, Eq. (6) is modified to:

$$H'(t) = \sum_{k=0}^{L} a_k^n(t) \times \cos\left(\phi_k^n(t) - \frac{t}{100} \times \xi_k^n \right), t = 0, 1, \ldots, 99. \tag{12}$$

Let L_n be the number of harmonic partials on the *n*th control point. The value of L_n is computed as dividing the MVF (maximum voiced frequency) by the pitch frequency, *i.e.* $L_n = MVF(n)/\tilde{F}_1^n$. In general, L_n may not be equal to L_{n+1}. Hence, we set the value of L, *i.e.* the number of partials, in Eqs. (6) and (12) to the greater of L_n and L_{n+1}. Suppose here that L_n is less than L_{n+1}. Then, the parameter values for the extended partials on the *n*th control point must be defined. Here, considering the continuity of signal-waveform, we simply let $\tilde{A}_k^n = 0$, $\tilde{F}_k^n = \tilde{F}_k^{n+1}$, $\tilde{\theta}_k^n = \tilde{\theta}_k^{n+1}$, for $k = 1 + L_n, 2 + L_n, \ldots, L_{n+1}$.

3.5 Synthesis of Noise Signal

For the noise signal, we synthesize it as a summation of sinusoidal components according to the signal model of HNM [20]. Let G_k be the frequency of the *k*th sinusoid. As G_k does not change with time, we define $G_k = 100 \times k$ (Hz) according to the thesis of Stylianou [20]. For the *n*th control point, however, the index k of G_k is not started from 1 and its starting value, K_s^n, is determined according to the MVF of this control point, *i.e.* $K_s^n = \lceil \mathrm{MVF}(n)/100 \rceil$. In contrast, the end value of the index k is always a fixed value, $K_e = \lfloor 11{,}025/100 \rfloor$. The MVF value of an analysis frame is determined during HNM parameter analysis [20].

Let B_k^n be the noise amplitude for the *k*th sinusoid on the *n*th control point. To determine its value, the 10 cepstrum coefficients, on the *n*th control point, representing the noise spectral envelope are first appended with zero values and inversely transformed (inverse discrete Fourier transform) to the spectral domain [15, 20]. Then, exponentiation is taken to obtain the corresponding spectral magnitude coefficients, X_j, $j = 0, 1, \ldots, 2{,}047$. According to the magnitudes X_j, the value of B_k^n can be obtained by linearly interpolating the two adjacent magnitudes, X_i and X_{i+1}, whose frequencies surround the frequency of G_k.

When the values of K_s^n and B_k^n for the *n*th control point are known, the noise-signal samples between the *n*th and (*n* + 1)th control points can be computed with the equations (rewritten here):

$$N(t) = \sum_{k=K_s}^{K_e} b_k^n(t) \times \cos(\gamma_k^n + t \times 2\pi \times G_k / 22{,}050), t = 0, 1, \ldots, 99, \tag{13}$$

$$b_k^n(t) = B_k^n + \frac{t}{100} \times (B_k^{n+1} - B_k^n), \tag{14}$$

$$\gamma_k^n = \gamma_k^{n\text{-}1} + 100 \times 2\pi \times G_k/22{,}050, \tag{15}$$

where K_s is set to the lesser of K_s^n and K_s^{n+1} and where γ_k^n is the initial phase for the *k*th sinusoid on the *n*th control point. In Eq. (14), the time-varying amplitude, $b_k^n(t)$, is linearly interpolated.

For the synthesis of the long unvoiced segment in Fig. 1, Eqs. (13), (14) and (15) can still be used to generate signal samples. Nevertheless, the lower bound of the summation index, *k*, in Eq. (13) will now be fixed to 1. This is equivalent to setting all the MVF values to the constant, 0Hz, for all the control points within the unvoiced segment.

3.6 Synthesis of Portamento Singing

Usually a lyric syllable is assigned one musical note. A syllable, however, may occasionally be assigned two (or three) notes. When a syllable is assigned more than one note, it should be sung in portamento. That is, the pitch-contour of the syllable should transit smoothly from the former note's pitch to the latter note's pitch in the middle portion. An example pitch-contour is shown in Fig. 5. The duration of the voiced segment of a syllable is divided into three time intervals of equal lengths. The left and right intervals are planned to sing stable pitches of the two notes in order that they can be explicitly perceived. The middle interval is used to transit the pitch smoothly.

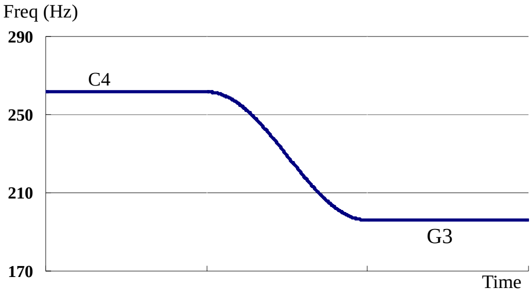

Fig. 5. Example pitch-contour for a syllable sung in portamento.

In this paper, the pitch-contour of a lyric syllable is planned before its pitch-tuned HNM parameters are calculated. Suppose that the two notes to be sung in portamento are of the pitch frequencies P_a and P_b. We first divide the control points within the voiced segment of the syllable into three groups. Then, the control points within the first and third intervals are directly assigned the pitches of P_a and P_b respectively. For the *n*th control point in the second interval, however, its pitch, P^n, is defined with a cosine based function:

$$P^n = \frac{(P_a + P_b)}{2} + \frac{(P_a - P_b)}{2} \times \cos(\frac{n}{M} \times \pi) \tag{16}$$

where M is the number of control points in the second interval. This cosine based transition function is designed according to our heuristic. It has a good property that the slopes at the left and right ends are both zero. Hence, the transition part of the pitch contour connects smoothly to the left and right stable parts.

4. SYSTEM IMPLEMENTATION AND TESTING

4.1 System Implementation

Mandarin Chinese has only 408 different syllables if the superimposed tones are not distinguished [23]. Hence, we just record and save each of these syllables once for analyzing their HNM parameters. Each of these syllables is uttered in isolation and in a level tone by a female in a soundproof room. Then, an HNM analysis program is developed to analyze these syllables. The analysis method is based on the one proposed by Stylianou [20], but some modifications are made. For example, the frequency values of harmonic peaks in a spectrum are more precisely estimated with parabolic interpolation, and the frequency values of harmonic peaks are all saved for latter use in Eq. (2). In addition, an analysis frame's MVF is more strictly defined as its following four harmonic candidates must all be checked to be not harmonic peaks. If any one of its following four harmonic candidates is checked to be a harmonic peak, the current MVF will not be admitted. This MVF checking rule is useful and can prevent detecting lower and incorrect MVF values for such syllables with vowel /i/ as their nucleus.

In developing the program for synthesizing a Mandarin singing voice, the methods described in section 2 are used to parse an input score file and the methods described in section 3 are used to synthesize the signal waveforms for the lyric syllables. Since the number of computations is considerable, the synthesis program is difficult to run in real-time on an ordinary personal computer (*e.g.*, a 2.6MHz Pentium CPU based). Nevertheless, we intend to synthesize singing voices and play the signal waveforms in real-time. This is because our synthesis program will be integrated into a humanoid robot to show the skill of singing. Therefore, we tried to find possible bottlenecks. As a result, a major bottleneck is found to be the FFT (fast Fourier transform) [13, 15] operation for transforming cepstrum coefficients back to the spectrum domain for determining the noise signals' amplitudes. When the FFT length is changed from 4,096 to 1,024 points, the synthesis speed is largely improved and achieves 3 times real-time speed. That is, a synthetic singing voice of 3 seconds in length needs only 1 second of CPU time to synthesize. Here, the frequency spacing of 21.53Hz (22,050/1,024) between two adjacent bins is thought to be sufficient because the frequencies of adjacent sinusoidal components are 100Hz apart.

4.2 System Testing – Signal Timbre and Clarity

To show the ability of the HNM-based synthesis scheme, spectrograms for the signals of the syllable /wan/ are analyzed with the package, WaveSurfer, and shown in the lower part of Fig. 6. The spectrogram at the left side of Fig. 6 is for the recorded syllable /wan/ while the spectrogram at the right side is for a synthetic syllable /wan/ sung in portamento. When the two spectrograms at the two sides are compared, it can be found that the formant traces have same curve shape and same frequency height. This explains why they will have same timbre. Also, as seen in the upper part of Fig. 6, the pitch height and shape of the synthetic syllable are very different from those of the recorded syllable. Nevertheless, the clarity and naturalness of the synthetic singing signal are still kept in a high level. For demonstration, we have set up a web page [5]. From this web page, the signal waveforms for the two syllables in Fig. 6 can be heard and compared.

4.3 System Testing – Perception of Reverberation and Fluency

Usually, the undesired effect of reverberation may be heard from synthetic audio signals. Here, two Mandarin song files, denoted as *SA* and *SB*, were synthesized for testing the reverberation effect. *SA* was synthesized with the HNM-based scheme studied here, and *SB* was synthesized with our PSOLA-based scheme studied previously. The details

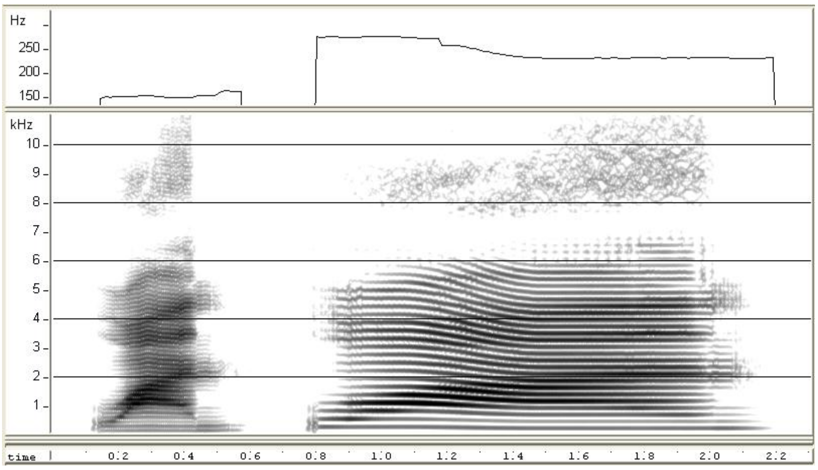

Fig. 6. Pitch-contours and spectrograms for the recorded and synthetic syllables of /wan/.

of PSOLA are referred to relevant literature [14, 18]. The two files, *SA* and *SB*, can also be downloaded from a web page [5]. We invited 12 persons to participate the perception tests. Each person was allowed to listening to *SA* and *SB* an unrestricted number of times. Then, he or she was asked to give a score about the reverberation level of *SB* when compared to *SA*. The score, 2 (– 2), is defined as *SB* (*SA*) is apparently more reverberant than *SA* (*SB*) while the score, 1 (– 1), is defined as *SB* (*SA*) is slightly more reverberant than *SA* (*SB*). Otherwise, the score, 0, should be given if they cannot be distinguished. After the perception tests, the averaged score was computed to be 1.25 and its standard deviation was 0.92. Therefore, the HNM-based scheme is better than the PSOLA-based scheme in reducing the effect of reverberation.

About the reverberation effect found in PSOLA, we think there are two possible causes. The first is that the pitch markers (or peaks) labeled for a syllable may not be synchronized well with their corresponding glottal epochs and may stagger around correct synchronization points. We find that the speech waveforms of some recorded syllables with /a/ kernel are especially difficult to label in regards to their pitch markers. The other cause is that when the pitch of the synthetic syllable is tuned to be considerably higher than the pitch of the recorded syllable, three or more adjacent pitch periods of the recorded syllable will be overlapped and added. This will also result in reverberation.

Furthermore, the PSOLA based scheme requires considerable labor to manually check and correct the automatically labeled pitch markers. In contrast, the HNM based scheme requires no such kind of labor. As to the issue of real-time execution, both schemes can be executed in real-time but the PSOLA based scheme is much faster than the HNM based scheme. In this study, however, signal quality is the major concern.

Note that a piecewise-linear time mapping function is proposed in section 3.1 to promote the fluency level of a synthetic song. Here, fluency is defined as the lyric syllables of a music sentence are heard as fluently connected and not just concatenation of independent syllables. To show the effectiveness of the proposed mapping function, two more song files, denoted as *SC* and *SD*, were synthesized using another female's recorded syllables to analyze HNM parameters. Here, *SC* was synthesized with a linear time mapping, and *SD* was synthesized with a piecewise linear time mapping. Then, the same 12 persons as mentioned were invited to compare the fluency level of *SD* with *SC*. Each person was allowed listening to *SC* and *SD* multiple times. Then, he or she was asked to give a score concerning the fluency level of *SD* when compared to *SC*. The score, 2 (– 2), is defined as *SD* (*SC*) is apparently more fluent than *SC* (*SD*) while the score, 1 (– 1), is defined as *SD* (*SC*) is slightly more fluent than *SC* (*SD*). Otherwise, the score, 0, should be given. As a result, the averaged score was computed to be 1.08 and its standard deviation was 0.49. Therefore, the piecewise linear time mapping can indeed significantly promote the fluency level.

5. CONCLUDING REMARKS

In this paper, a piecewise linear function is proposed to map a control point on the synthetic time axis to two adjacent analysis frames of a recorded syllable. Although such a time-axis mapping method may not be the best, it can promote the fluency level of a synthetic singing syllable significantly, according to the results of perception tests. Next, control points are placed in a fixed pace for synthesizing voiced segment. This can provide more accurate control of spectrum progression. As to the estimation of the spectral amplitude envelope for a signal frame, an order three Lagrange interpolation based method is proposed. This is because we must consider computing efficiency and envelope accuracy simultaneously in order to synthesize singing voice in real-time. Although Lagrange interpolation is simple and may seem inaccurate, the timbres of the synthetic syllables are very close to their corresponding recorded syllables according to perception testing. In addition, we have added a MVF checking rule to prevent detecting an erroneous MVF value from a signal frame.

In terms of the enhancements and the signal-synthesis equations rewritten here, we have built a Mandarin Chinese singing voice synthesis system. In this system, each Mandarin syllable needs only one recorded utterance, and its analyzed HNM parameters are used to synthesize singing syllables of diverse durations and pitches. Also, by eliminating the computational bottleneck in transforming cepstrum coefficients back to noise spectrum, the system can now run smoothly in real-time. Furthermore, perception tests have been conducted to compare the singing-voice signals synthesized respectively by the HNM and PSOLA based schemes. The results show that the HNM based scheme proposed here can indeed be used to synthesize a Mandarin singing voice of consistent timbre and much higher signal quality (much clear and without reverberation) than our PSOLA based scheme studied previously. As to the singing expression factors, vibrato and huskiness, we will study them in a further study and integrate them into the synthesis scheme presented here.

REFERENCES

1. J. Bonada and A. Loscos, “Sample-based singing voice synthesizer by spectral concatenation,” in *Proceedings of the Stockholm Music Acoustics Conference*, 2003, pp. 1-4.
2. J. Bonada and X. Serra, “Synthesis of the singing voice by performance sampling and spectral models,” *IEEE Signal Processing Magazine*, Vol. 24, 2007, pp. 67-79.

3. C. Dodge and T. A. Jerse, *Computer Music: Synthesis, Composition, and Performance*, Schirmer Books, New York, 1997.
4. J. D. Faires and R. Burden, *Numerical Methods*, Books/Cole Publishing Company, Pacific Grove, CA, 1998.
5. H. Y. Gu and H. L. Liau, http://guhy.csie.ntust.edu.tw/trhnm/sing.html.
6. H. Y. Gu and Y. Z. Zhou, "An HNM based scheme for synthesizing Mandarin syllable signal," *International Journal of Computational Linguistics and Chinese Language Processing*, Vol. 13, 2008, pp. 327-341.
7. Y. E. Kim, "Singing voice analysis/synthesis," Ph.D. Thesis, Massachusetts Institute of Technology, 2003.
8. W. C. Lee, H. Y. Gu, K. L. Chung, *et al.*, "The realization of a music reading and singing two-wheeled robot," in *Proceedings of IEEE Workshop on Advanced Robotics and its Social Impacts*, 2007, pp. 67-72.
9. C. Y. Lin, T. Y. Lin, and J. S. R. Jang, "A corpus-based singing voice synthesis system for mandarin Chinese," in *Proceedings of the 13th ACM international Conference on Multimedia*, 2005, pp. 359-362.
10. M. W. Macon, L. Jensen-Link, J. Oliverio, M. A. Clements, and E. B. George, "A singing voice synthesis system based on sinusoidal modeling," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 435-438.
11. Y. Meron, "High quality singing synthesis using the selection-based synthesis scheme," Ph.D. Thesis, Department of Information and Communication Engineering, University of Tokyo, 1999.
12. Y. Meron and K. Hirose, "Synthesis of vibrato singing," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000, pp. 745-748.
13. F. R. Moore, *Elements of Computer Music*, Prentice-Hall, Englewood Cliffs, NJ, 1990.
14. E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech Communication*, Vol. 9, 1990, pp. 453-467.
15. D. O'Shaughnessy, *Speech Communications: Human and Machine*, IEEE Press, Piscataway, NJ, 2000.
16. A. Robel and X. Rodet, "Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation," in *Proceedings of International Conference on Digital Audio Effects*, 2005, pp. 1-6.
17. X. Rodet, "Synthesis and processing of the singing voice," in *Proceedings of the 1st IEEE Benelux Workshop on Model Based Processing and Coding of Audio*, 2002, pp. 99-108.
18. N. Schnell, G. Peeters, S. Lemouton, P. Manoury, and X. Rodet, "Synthesizing a choir in real-time using pitch synchronous overlap add," in *Proceedings of International Computer Music Conference*, 2000, pp. 102-108.
19. D. Schwarz and X. Rodet, "Spectral envelope estimation and representation for sound analysis-synthesis," in *Proceedings of International Computer Music Conference*, 1999, pp. 351-354.
20. Y. Stylianou, "Harmonic plus noise models for speech, combined with statistical methods, for speech and speaker modification," Ph.D. Thesis, Ecole Nationale Supèrieure

des Télécommunications, Paris, France, 1996.

21. Y. Stylianou, “Modeling speech based on harmonic plus noise models,” *Nonlinear Speech Modeling and Applications*, G. Chollet, *et al*., eds., 2005, pp. 244-260.
22. F. Thibuult and P. Depalle, “Adaptive prpcessing of singing voice timbre,” in *Proceedings of Canadian Conference on Electrical and Computer Engineering*, 2004, pp. 871-874.
23. Mandarin Romanization Table, Bureau of Consular Affairs, Ministry of Foreign Affairs, R.O.C., Taiwan, http://www.boca.gov.tw/ct.asp?xitem=1608&ctnode=193.

Hung-Yan Gu (古鴻炎) received the B.S. and M.S. degrees in Computer Engineering from National Chiao Tung University in 1983 and 1985, respectively, and the Ph.D. degree in Computer Science and Information Engineering from National Taiwan University in 1990. Currently, he is an Associate Professor in the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei. Also, he is one of the Section Editors of International Journal of Computational Linguistics and Chinese Language Processing. His research interests include speech signal processing, computer music synthesis, and information hiding.

Huang-Liang Liao (廖皇量) was born in 1980. He received the B.S. degree in Information Engineering from Tatung University, Taipei, in 2003, and the M.S. degree in Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, in 2006.