Building a Japanese MIDI-to-Singing song synthesis using an English male voice

Hung-Che Shen^{1,*}

¹I-Shou University, Digital Media Design Department, Kaohsiung, Taiwan

Abstract. This work reports development of a MIDI-to-Singing song synthesis that will produce audio files from MIDI data and arbitrary Romaji lyrics in Japanese. The MIDI-to-Singing system relies on the Flinger (Festival **singer**) for singing voice synthesis. Originally, this MIDI-to-Singing system was developed by English. Based on some Japanese pronunciation rules, a Japanese MIDI-to-Sing synthesis system was developed and derived. For a language transfer like Festival synthesized singing, two major tasks are the modifications of a phoneset and a lexicon. Originally, MIDI-to-Sing song synthesis can create singing voices in many languages, but there is no existing Japanese festival diphone voice available right now. We therefore used a voice transformation model in festival to develop Japanese MIDI-to-Singing synthesis. An evaluation of a song listening experiment was conducted and the result of this voice conversion showed that the synthesized singing voice successfully migrate from English to Japanese with high voice quality.

1 Introduction

The goal of this research is to synthesize natural singing Japanese song from an English Text-to-Speech voice. With the term MIDI-to-Singing, we mean the production of human-like singing voice based on a given MIDI format music. The MIDI-to-Singing system is an extension from a speech-to-singing synthesis, which converts a speaking voice reading the lyrics of a song to a singing voice given its musical score. Therefore, a different language singing can be easily derived by modifying the original speech features unique to them.

Singing voice synthesis enables computers to “sing” any song. Since 2007, it has become especially popular in Japan because of Yamaha’s VOCALOID singing synthesizer [1]. There can be found a lot of original musical compositions in the video sites such as YouTube or Niko Niko Douga. There is now a growing demand for more flexible systems that can sing songs with various voices as evidenced by the many singer libraries being created and released on the Internet by users for UTAU [2] singing voice synthesis software.

Without a score editor environment for end-users, it should be noted that a concatenation-based singing synthesizer was already proposed by Macon et al. [3] in 1992. They finally released open source for this method that is called Flinger [4]. It is written by Mike Macon based on the Festival Speech Synthesis System [5], developed at the University of Edinburgh. Flinger needs the OGIsLPC plugin as the signal processing “backend” that was developed at OGI [6]. This takes diphone waveform files, coded as residual-excited LPC parameters, and performs the necessary concatenation, pitch/duration change, and smoothing of the diphone waveforms.

---

* Corresponding author: shungch@isu.edu.tw

A MIDI-to-Singing system requires the integration of two major components. One is a MIDI sequencer with a lead-sheet view. The other is a singing voice synthesis (SVS). Based on a musical graphical interface such as a lead sheet, the users can compose music/lyrics in a MIDI sequencer.

In Festival, a new language is mainly comprised of constructing a phoneset and building a lexicon. The phoneset is basically a set of syllables and includes the linguistic characteristics of how vowels and consonants are pronounced. A lexicon is a database of words with their respective pronunciations, using the phones from the phoneset. During the past few years, we have been using songs for educational learning based on our previous MIDI-to-Singing system [7]. To extend the applications of singing voice synthesis, a language transfer will be needed. A language transfer refers to speakers applying knowledge from one language to another language. This paper assumes that a MIDI-to-Singing language transfer between English and Japanese is feasible. In order to demonstrate Japanese MIDI-to-Singing, some songs are available online as shown in Fig. 1.

Fig. 1. Japanese MIDI-to-Singing Demo Websites: <http://sing.dmd.isu.edu.tw/en/>.

This paper is organized as follows. In Section 2, we present Japanese syllable and word syntax which explains the language transfer in Festival Text-to-Speech. After that, a MIDI-to-Singing song synthesis is described in Section 3. Finally, experimental results are given in Section 4. Section 5 concludes this paper.

2 Derive a Japanese diphone voice

In Japanese, a syllable is formed by a vowel only or a combination of a consonant and a vowel. One syllable can only be one to three letters (Latin alphabet) long. A syllable cannot be a single consonant. In addition, a word can be formed by a series of syllables. Words

may never end on a consonant, with the exception of consonant /n/. Once a word is broken into syllables it can be fed into a speech synthesis system, such as Festival, and then be pronounced. Japanese has five vowels. Vowels are the characters A, E, I, O, U. The vowels are pronounced roughly like these sounds in standard American English: “A” as in “father,” “E” as in “set,” “I” as in “bee,” “O” as in “toe” and “U” as in “good”. Vowels in Japanese may be long or short. Long vowels are held for twice as long as short vowels, but except for the length, there is no difference in pronunciation between short and long vowels. Japanese Vowels can be located in the initial, the medial and final part of words. Japanese also has what is called a “syllabic nasal.” This sounds like the English “N” in “night” when it is followed by an “S,” “Z,” “T,” “D,” “N,” or “R” sound.

Japanese Romanization (Romaji) is the standard way of transliterating Japanese into the Latin alphabet. Consonants and vowels are always broken up into the same “blocks” of sound allowing very easy parsing of words. This basic phonetic alphabet is known as hiragan as shown in the Table 1.

Table 1. Japanese Hiragana.

VC	k	s	t	n	h	m	y	r	w
a	ka	sa	ta	na	ha	ma	ya	ra	wa
i	ki	shi	chi	ni	hi	mi	(y)i	ri	(w)i
u	ku	su	tsu	nu	hu	mu	yu	ru	(w)u
e	ke	se	te	ne	he	me	e	re	(w)e
o	ko	so	to	no	ho	mo	yo	ro	(w)o

There are a few oddities in the hiragana table. The blocks “(y)i”, “(w)i”, “(w)u”, “(w)e”, and “(w)o” are pronounced as the vowel components only, e.g., “(w)o” is simply “o”; (w) indicates the “w” is fully cut out from the sound. There is no “tu” sound. Instead we use the “tsu” sound. “Tsu” can give some people trouble, but using the notation above “tsu” would be pronounced as the ts in “cats” with a short “u” added to the end; (ca)ts-u. The Japanese Romaji consists of fourteen consonants—k, s, t, n, h, m, y, r, w, g, z, d, b and p. Japanese pronunciation follows a simple set of rules for syntax. Each vowel and basic consonant/vowel combination corresponds to a phonetic character in the Japanese alphabet. All syllables must end on a vowel. A syllable cannot be a single consonant.

Assuming that a voice set exists, constructing a language in Festival requires modifying a phoneset, building a lexicon, and adding prosody (defining the letter to sound rules). Because a word is usually broken into syllables in singing, when syllables are fed into a singing synthesis system, some considerations required in text-to-speech are ignored such as prosody. Two basic processes need to be addressed only contains:

- Modifying a phoneset,
- Modifying a lexicon,

2.1 Modifying a phoneset

A phoneme (or phone) is one of the units of sound that distinguish one word from another in a particular language. Syllables are considered the smallest unit of speech, while a phone

is only a speech sound. For the Japanese language, there exists a 1-1, 1-2 and 1-3 relationship between syllables and phones. In Festival the phones are stored in the phoneset. A phoneset is a set of symbols which may be further defined in terms of features, such as vowel/consonant, place of articulation for consonants, type of vowel etc. This, like everything else in Festival, is a function-based structure in Scheme. The phoneset for a language has been defined in many cases, and it is wise to follow convention when it exists. For Japanese, that's probably ok as the Japanese phoneset is mostly a subset of the English phoneset. In Festival, the phones' features and their values must be defined with the phoneset. This, like everything else in Festival, is a function-based structure in Scheme. At synthesis time, each Japanese phone must be mapped to an equivalent (one or more or modified) US phone. This is done though phoneset members for the Japanese as shown in Table 2.

Table 2. Phoneset members for the Japanese.

Standard vowels	Long vowels	Unvoiced reduced vowels	Consonants
(a + s 3 3 0 0 0 0)	(a: + 1 3 3 0 0 0 0)	(I + a 1 1 0 0 0 0) (U + a 1 3 + 0 0 0 0)	(k - 0 0 0 0 s v -) (s - 0 0 0 0 f a -) (sh - 0 0 0 0 f p -) (t - 0 0 0 0 s a -) (ch - 0 0 0 0 a p -) (ts - 0 0 0 0 f a -) (n - 0 0 0 0 n a +) (h - 0 0 0 0 f g -) (f - 0 0 0 0 f b -) (m - 0 0 0 0 n l +)
(i + s 1 1 0 0 0 0)	(i: + 1 1 1 0 0 0 0)	Geminates (Qk - g 0 0 0 0 s v -) (Qs - g 0 0 0 0 f a -) (Qsh - g 0 0 0 0 f p -)	
(u + s 1 3 + 0 0 0 0)	(u: + 1 1 3 + 0 0 0 0)		
(e + s 2 1 0 0 0 0)	(e: + 1 2 1 0 0 0 0)		
(o + s 2 3 0 0 0 0)	(o: + 1 2 3 0 0 0 0)		

The description of phone features is shown in Fig. 2. The descriptions describe the following phone features, for vowels: length (short, long, diphthong, schwa), height (close, close-mid, mid, open-mid, open), frontness (front, central, back), and lip rounding (rounded, unrounded). For consonants: consonant type (stop, affricate, fricative, nasal, lateral, approximant), place of articulation (labial, alveolar, velar, palatal, post-alveolar, labio-dental, dental, glottal), and consonant voicing (voiced, unvoiced).

```

;;; Phone Features
;;; vowel or consonant
(vc + -)
;; vowel length: short long diphthong schwa geminate (for consonants)
(vlmg s l d a g 0)
;; vowel height: high mid low
(vheight 1 2 3 0)
;; vowel frontness: front mid back
(vfront 1 2 3 0)
;; lip rounding
(vrnd + - 0)
;; consonant type: stop fricative affricate nasal lateral approximant
(ctype s f a n l r 0)
;; place of articulation: labial alveolar palatal labio-dental
;; ..... dental velar glottal
(cplace l a p b d v g 0)
;; consonant voicing
(cvox + - 0)
)

```

Fig. 2. Phone features.

However, scanning down the phoneset one will notice an arrangement of syllables. For example, “(a + 1 2 2 - 0 0 0)” is the phone from the Radio phoneset included with festival. This phone represents the first ‘a’ can be the phone of a Japanese Romaji word “ai” or “au”.

In the phone above, the “a” represents the name of the phone. This name is used in the lexicon to refer to which phones need to be uttered by the speech synthesis system to form words. Next comes a ‘+’ or ‘-’ symbol where the positive affirms that the phone represents a vowel, and the negative indicates the phone represents a consonant. All of the remaining settings can take 0 for a value which indicates that that setting is not applicable. Next in the phone definition there is “l 2 2” which define three linguistic settings for vowels. The first entry corresponds to vowel length can be ‘s’ for short, ‘l’ for long, ‘d’ for diphthong, ‘s’ for schwa, or 0 for not applicable. Next, the second setting indicates vowel height and can be an integer from 0 for not applicable, 1 for high, 2 for medium, and 3 for low. Lastly for the vowel section, the third setting corresponds to vowel frontness and can have the values 1 for high, 2 for mid, and 3 for low. For the following setting, in the example above, “-” represents lip rounding. When a human pronounces a phoneme the lip position can affect which phoneme is uttered. This setting can take the values “+” indicating that lip rounding is present, or “-” which indicates the contrary. For the last three settings, the ones for consonants, we start with consonant type which can be ‘s’ for stop, ‘f’ for fricative, ‘a’ for affricate, ‘n’ for nasal, ‘l’ for lateral, or ‘r’ for approximant. Next, the second setting provides for the place of articulation and takes the values ‘l’, ‘a’, ‘p’, ‘b’, ‘d’, ‘v’, or ‘g’. The place of articulation represents how the mouth or throat says the consonant. Lastly, consonant voicing is represented by the final setting with values ‘+’ for its presence, or ‘-’ for its absence.

2.2 Modifying a lexicon

A lexicon is basically a large dictionary of words with the corresponding syllables to pronounce the word correctly. The lexicon had to be converted to the phonetic alphabet and to the format that is required by the Festival system. The lexicon format comes in three parts, a compiled lexicon, an addenda, and a rule system for handling unknown words. If any word appears that is not part of the lexicon, the pronunciation will be found by letter-to-sound-rules. Additionally, there exist many websites with examples of correct pronunciation of a subset of Japanese words. An addenda of words augment what is in the lexicon, but may not have been compiled and saved into the lexicon itself. This is useful for testing new additions to the lexicon. Some typical example entries of English are (“walkers” n (((w oo) 1) ((k @ z) 0))) for the word “walker” and (“present” v (((p r e) 0) ((z @ n t) 1))) for the word “present”.

A pronunciation in Festival requires not just a list of phones but also a syllabic structure. We already discussed about our phones with their features. The lexicon structure that is basically available in Festival takes both a word and a part of speech to find the given pronunciation. An example of an entry in the compiled lexicon is (“honda” nil (((h > n) 1) ((d ^ 0))). Note that this represents the word “Honda” and breaks it into the phones named “Hon,” and “Da.” Numbers next to the phone names can affect the duration of the phone pronounced. Explicit marking of syllables a stress value is also given (0 or 1). Lexicon entries of Japanese are shown in Fig. 3.

```
(“a” nil (((A) 0) ((A A) 2)))
(“i” nil (((j j) 0) ((i: i:) 2)))
(“u” nil (((u) 0) ((u) 2)))
(“e” nil (((E) 0) ((E) 2)))
(“o” nil (((>) 0) ((oU) 2)))
(“sa” nil (((s s) 0) ((A A) 2)))
```

Fig. 3. Lexicon entries of Japanese.

3 A MIDI-to-Singing song synthesis

Festival is a general-purpose concatenative text-to-speech (TTS) system that uses the residual-LPC synthesis technique, and is able to transcribe unrestricted text to speech. Assuming that a voice set exists, constructing a language in Festival requires creating a phoneset, building a lexicon, and defining the letter to sound rules. The voice set is the actual sounds that festival outputs. Since Festival provides some example voice sets, the focus of my research has been on the theoretical construction of the other components which have greater relevance to the Japanese language. Building a new voice set using a local Japanese speaker can be accomplished using the application FestVox.

3.1 OGI residual-LPC synthesizer (Festival plug-in)

This OGI residual-LPC synthesizer, which has to be considered as a plug-in for Festival, has been developed at OGI (Oregon Graduate Institute of Science and Technology, Portland, OR), and provides a new signal processing engine, and new voices, not included in the Festival distribution. Specifically, new pitchmark and LPC analysis algorithms, together with some scheme scripts that enable the creation of new voices in the OGIResLPC synthesizer are included. It is freely available for research and educational use. OGI has expanded its range of languages: there are TTS systems for English, Spanish and Welsh.

OGIResLPC is a drop-in module for the Festival TTS system created by CSTR at the University of Edinburgh (<http://www.cstr.ed.ac.uk/projects/festival>). This version of OGIResLPC has been designed to work with Festival version 1.2.0, released September 1997. It should work with any version 1.2.x newer than this, and can possibly be made to work with other versions of Festival, but this would require some changes to the code and knowledge of Festival internals. It provides waveform synthesis of speech with reasonable quality, but has not been extensively optimized in any way. It is meant to serve as a simple baseline synthesizer in the CSLU Toolkit and for other experiments.

3.2 Score editor

The Score Editor provides an environment in which the user can input notes, lyrics, and optionally some expressions. The Editor is designed especially for MIDI-to-Singing system. A lead sheet is a form of musical notation that specifies the essential elements of a MIDI song: the melody, lyrics and harmony. The melody is written in modern Western music notation, the lyric is written as text below the staff and the harmony is specified with chord symbols above the staff.

The user can type-in lyrics in normal writing and the Editor automatically converts the lyrics into phonetic symbols by looking into a built-in pronunciation dictionary. If the word consists of two or more syllables, the Editor automatically decomposes it into syllables. The user can easily add vibrato in the Editor. A screenshot of the MIDI-to-Singing process and score editor is shown in Fig. 4(a) and 4(b).

Fig. 4. (a) Process of a MIDI-to-Singing system. (b) Score editor of MIDI-to-Singing.

4 Discussions

The task of synthesizing singing has, of course, a lot in common with that of synthesizing speech. For singing, the immense problems of reliably modelling intonation and syllable length are already solved (or rather by-passed) by the composer. In singing, Japanese provides relatively a greater ease for the English speaker learning Japanese than the English system would to Japanese singers. Not taking into account language variation (accent and dialect), there are five vowels and 17 consonant phonemes in Japanese while there are 20 vowels and 24 consonants in English language. Therefore, a Japanese MIDI-to-Singing system can be accomplished through an English MIDI-to-Singing system.

To evaluate the effectiveness of the proposed Japanese MIDI-to-Singing song synthesis, we conducted subjective experiments. Ten Japanese songs sung by MIDI-to-Singing system and sung by VOCALOID are used for evaluation. In order to evaluate the enhancement of modification strategies we adopted in singing synthesis, we scheduled a preference test to evaluate our system. The 10 kid MIDI songs are from Mama lisa's website: <http://www.mamalisa.com/>.

10 pairs of singing utterances were generated by 2 systems and 10 people took part in our experiment to judge which system got the better performance. These pairs of singing voices are played in a randomized order and the listeners gave their scores for each pair. The score is in 3 levels: the first is better, the second is better and they are almost the same. The result is listed in Table 3. As seen from Table 3, obviously the system with our proposed strategies got a lower performance than VOCALOID.

Table 3. Preference test score.

Prefer MIDI-to-Singing	Prefer VOCALOID
45%	55%

5 Conclusions

Japanese provides relatively a greater ease for the English speaker learning Japanese than the English system would to Japanese speakers. Not taking into account language variation (accent and dialect), there are five vowels and 17 consonant phonemes in Japanese while there are 20 vowels and 24 consonants in English language. Therefore, a Japanese MIDI-to-Singing system can be accomplished through an English MIDI-to-Singing system.

This paper outlines the theoretical issues that demonstrate that it is possible for English-to-Japanese transform of a singing synthesis project. Although the development of a Japanese MIDI-to-Singing system is still in working and far from complete. However, the project is likely to be an iterative process whereby the pronunciation of the system improves over time as the settings of the phones are made more precise, and the number of entries in the lexicon is increased. In the future, speech synthesis could also be fully integrated with singing synthesis. It will be challenging to develop new voice synthesis systems that could seamlessly generate any voice produced by a human or virtual singer/speaker.

We would like to thank Dr. Mike Macon for his invaluable contributions especially in developed Flinger to demo a MIDI-to-Singing synthesis. We would also like to thank Oregon Graduate Institute of Science and Technology for having made the OGILPC Plug-In for the Festival TTS.

References

1. H. Kenmochi, H. Ohshita, Proc. of Interspeech, (2007)
2. "UTAU," Available online: <http://utau2008.web.fc2.com/>, (2018)
3. M. Macon, Jensen-Link, Oliverio, Clements, Geourge, Proceedings of ICASSP 97, **435** (1997)
4. Flinger. Not available online: <http://www.cslu.ogi.edu/tts/flinger> URL (2007)
5. Festival Speech Synthesis, Speech Tools & documentation, <http://www.festival.org/>.
6. OGILPC PlugIn for Festival. www: <http://cslu.cse.ogi.edu/tts/download/index.html> (1997)
7. H. C. Shen, and C. N. Lee, 6th International WOCMAT & New Media Conference, Chungli, Taoyuan, Taiwan, (2010)