Singer Module with Singing Synthesis for Humanoid Robots Applications

A. Rojo-Hernandez[1], H.M. Perez-Meana[1], N. Takayuki[2], E. Hernandez[1], and J. C. Sanchez[1]
[1]SEPI, National Polytechnical Institute, Mexico D.F., Mexico
[2]DMEIS, The University of Electro-Communications, Tokyo, Japan

Abstract— *Currently the research on robotics is a field with the most demand in various parts of the world, as it seeks to ensure they are capable of developing many of the most common tasks humans perform, such as identifying objects and taking them to recognize people by voice or face, identifying the words a person is saying, moving in free space, talking (to interact with humans) and one of the last things to be achieved: Making a singing robot. For this reason, on this task we will talk about a method in which a robot can sing using Vocaloid Editor 3.0, a speech synthesizer software which is able to generate the songs a robot will sing.*

Keywords: Speech synthesis; Phoneme; TTS (Text-to-speech); Acapella; Vocaloid Editor 3.0; VocaListener.

1. Introduction

A synthesizer is a tool designed to produce electronic musical sounds generated artificially, using techniques such as additive synthesis, subtractive, frequency modulation, or physical modeling of phase modulation, to create sounds. The synthesizer sounds are created by direct manipulation of electrical signals, through the manipulation of a digital FM waves, handling of discrete values using computers (software based synthesizers), or combining any method. In the final phase of the synthesizer, electric currents are used to produce vibrations in speakers and headphones.

The steps followed in all synthesis process are: First, a set of modules analyzes the input text to determine the structure of the sentence and the phonetic composition of each word and a secondly, another set of modules transforms this abstract linguistic representation into speech [9].

Currently there are many techniques and algorithms for synthesizing text into speech. These algorithms are called Text-To-Speech (TTS). An example of this type of algorithm can be seen in the program developed by Yamaha Corporation: Vocaloid Editor 3.0.

Given the case that Vocaloid has been developed by the Japanese company Yamaha is expected that the program is based on the same language in others words in Japanese. Therefore this research will use the Japanese language as basis for the songs. It is important to emphasize that Vocaloid Editor 3.0 will be responsible for synthesizing the song, therefore it will be syncing the timing of the notes with the phonemes and taking care of the duration of them.

2. Objective

The main objective of this paper is to propose a way or method by which it is possible to create a new module which will make the robot sing. Since the songs are synthesized using Vocaloid Editor 3.0, the aim of this project is to find a way in which the robot can play the songs that have been previously synthesized and are stored in a database, by a new module programmed in C++ that is responsible for the reproduction of these songs and can be added to existing modules of the robot.

3. Robot Plataform

To understand what is planned, it is necessary to first understand how a robot works, and how hardware and software combine themselves to correct operations, providing a better control of it.

We can see a robot like a system which is in turn divided into multiple modules that control the different parts or components of the robot. Modular organization systems provide greater control of the robot and make the error corrections.

The robot has modules that control vision (camera), audio (speakers and microphone), hands and arms, wheels among other modules. Each module has been programmed in C++ on Visual Studio environment. This work focuses on creating a new module that will be charged of making the robot sing, or in our case, play songs that have been synthesized using Vocaloid Editor 3.0. The whole system is divided in four modules: Vision, Audio, Robot and Task Modules and a server.

All modules are connected through the "Server" with GigE and have subscription information that describes required information for the processing each module. All information is gathered in the server and then the server forwards information to each module according to its subscription information.

The "Task Module" works as a controller for each scenario. This modular network architecture makes it

relatively easy to share the job in the development stage of the robot system [1]. The robot platform used in this paper is shown in Figure 1. The robot is based on the Segway RMP200 and consists of the following hardware components:

- The robot is equipped with a visual sensor.
- Two Arms capable of six degrees of freedom (6DOF robotic arm manufactured by Exact Dynamics), and 6-DOF hands.
- Omnidirectional wheels and a laser range finder (LRF) enable the robot to move freely within a room. Laser range finder (HOKYO UTM-30LX) is used for environmental mapping.
- Four on board PCs (Intel Core2Duo processor) are communicated each other through LAN.
- A sanken shotgun microphone CS-3e for audio input and Yamaha speaker NX-U10 for audio output.
- A stereo camera is used for obtaining depth information.
- The camera and microphone are mounted on Directed Perception pan-tilt unit PTU 46-47. [1] [2]

The abilities of the robot, other than the learning novel objects, are listed below [2]

1) Online SLAM(Simultaneous Localization and Mapping) and path planning.
2) Object manipulation (RRT-based path planning).
3) Simple speech interaction in English and Japanese.
4) Human detection and tracking using visual information.
5) Searching objects in the living room environments.
6) Face recognition using 2D-HMM.
7) Gesture Recognition. [1]

The new task to be performed is making the robot sing.

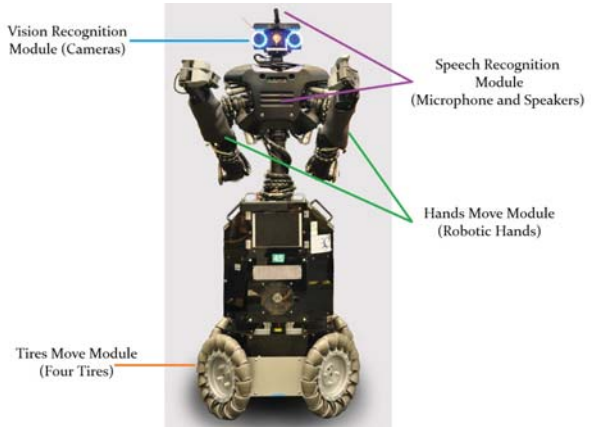

Fig. 1: Robot's Parts 大五郎 (DIGORO).

4. System Singer

The Singer Module is the new module that has been programmed and, as the name mentions, it will be in charge of the task of singing. The Singer Module needs to use the Audio Module; the Audio Module is divided into two parts: The Speech Recognition Nodule and the Jukebox Module.

The Speech Recognition Module works with everything related with speech process. The Jukebox Module is responsible for making the reproduction of the songs that will be performed by the robot. We will talk more about these modules later. The proposed path for making the robot to be able to sing is as follows.

The user asks the robot to sing any song using the Speech Recognition Module. The robot detects and sends the command to the Singer Module, this is the one that is responsible for processing and executing the command. Finally the robot starts to play the song that has been requested. In Figure 2 a block diagram of the system is presented. The process is performed as follows.

According to the Figure 2 this process was made in this way because the robot doesn't have ears, vocal cords, or other Senses and the way in which the robot is able to perform this task in a way is similar to a human is because the robot replaces ears with speakers and the vocal cords with the microphone.

The first time we tried to do a synthesis of the songs in real time, Vocaloid Editor 3.0 required a complex process for speech synthesis and it was not possible do it in real time, therefore the songs that the robot will reproduce have been previously synthesized using Vocaloid Editor 3.0 (For more references go to section 7, 7.1 and 7.2). In subsequent sections, each of the steps involved in all the modules will be explained in detail.

Fig. 2: General block diagram of proposed system.

5. Speech Recognition Module

The Speech Recognition Module used by the robot is a software system called Julius. It is responsible for speech processing and speech recognition. Within the recognition Julius makes use of the internet for search of vocabulary.

The Speech Recognition Module detects the words that are being said and with the help of the internet, it provides ten possible options of what was said.

5.1 Julius

Julius is a high-performance, two pass large vocabulary continuous speech recognition (LVCSR) decoder software for speech-related researchers and developers. It can perform almost real-time decoding on most current PCs in 60k word dictation tasks using word 3-gram and context-dependent on Hidden Markov Model (HMM). Major search techniques are fully incorporated. It is also modularized carefully to be independent from model structures. Various HMM types are supported, such as shared-state triphones and tied-mixture models, with any number of mixtures, states, or phones. Standard formats are adopted to cope with other free modeling toolkits.

Julius works with HTK which has the control of HMM. In order to execute the Julius recognizer, you need a language model and an acoustic model for your language. Julius adopts acoustic models in HTK ASCII format, pronunciation dictionary in HTK-like format, and word 3-gram language models in ARPA standard format (forward 2-gram and reverse 3-gram as trained from speech corpus with reversed word order). Its important to mention that Julius is only distributed with Japanese models. [15]

5.2 Hidden Markov Model Toolkit (HTK)

The Hidden Markov Model Toolkit (HTK) is a portable toolkit for building and manipulating hidden Markov models. HTK is primarily used for speech recognition research; although it has been used for numerous other applications include research on speech synthesis, character recognition and DNA sequencing.

HTK consists of a set of library modules and tools available in C language source form. The tools provide sophisticated facilities for speech analysis, HMM training, testing and results analysis. The software supports HMMs using both continuous density mixture Gaussians and discrete distributions and can be used to build complex HMM systems. [8]

5.3 Singer Module

The Singer Module is responsible for processing the command received from the Speech Recognition Module.

Figure 3 shows a block diagram of the edge module. We can see that the first step is to receive the command of the Speech Recognition Module for later processing.

The first thing that is done in the processing is to find the proper command because the Speech Recognition Module delivers ten possible options, so the first step is to analyze each of these options and identify if any of them corresponds to an command known.

Once the command has been identified the process continue to process it. In the processing section it is necessary to separate the command into two parts, by name of the song and command.

When the program has separated the name of the song and the command to run, the next step is to analyze the command. It is important to note that the commands will be delivered to the robot in Japanese as well as the names of the songs.

In order to identify the command it is necessary to search which command has been given. In this case there are two different commands with diferent Japanese representations, but in this case, we just to use two representations: hiragana and kanji (Katakana just on case of words in english).

1) Play the song (song's name をうたってください, を歌ってください song's name o uttate kudasai)
2) Stop the song (やめてください, 止めてください yamete kudasai)

After having identified the command, the module proceeds to search the name of the song in a database, if found, the command is executed: e.g. If the given command was "あああをうたってください" ("aaa o utatte kudasai) will be sought in the the song database "あああ" ("aaa") and as the command has already been previously identified "うたってください" ("utatte kudasai") in this case, it will play the song; then finally, it proceed to play the song that has already been found. The graphical process of this module is indicated in Figure 3.

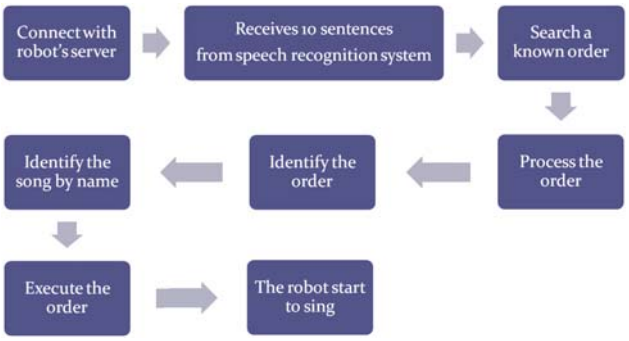

Fig. 3: The Singer Module process block diagram.

It's important to mention the Japanese's representation of the commands because there are differently presentations and therefore it's more difficult to identify a command both of this representations are given by the voice recognition within the ten options that it produces, ie within ten options it generates could be possible that two or more sentences have the same meaning but with different representation, however the singer module take into account just the first correct representation that you find.

The identification process commands and song's title. Because we only take into account two possible repre-

sentations for commands, in total we have four possible correct commands, two for play and two for the stop.

- Command 1: をうたってください (uttate kudasai "play").
- Command 2: を歌ってください (uttate kudasai "play").
- Command 3: やめてください (yamete kudasai "stop").
- Command 4: 止めてください (yamete kudasai "stop").

In the flowchart of Figure 4 you can see the process undertaken to identify commands and songs, we can see that through a string comparison, it performs a search for known commands as well as the search of the database of song title. In other words, when the ten options are received by the Singer Module, it takes the first option and compares it one of the four possible commands. If the comparison is negative, then it takes the second option and a comparison process is repeated again. The process is repeated until the comparison is positive in any of the ten choices, or until all have been compared with the ten options and none is positive, meaning that the user is required to repeat the command and the process start again.

In the event that one of the comparisons is positive, the next step is to clean the sentence eliminating unnecessary information such as the name of the robot. When you already have the free judgment garbage information, then it proceeds to compare the title song with song titles available in the database. This procedure is performed on the basis of comparisons similar to the identification of the command. So if it was compared to all existing titles in the database and comparison have been negative, then it is necessary that the user repeats the command and to restart process. However, in the event that it finds the title of the song in the database, it proceeds to extract the song ID and the ID of the command, and these two numbers will be sent to Jukebox which is the module responsible for playing or pausing songs.

As the goal was to create a module that was able to control playback of songs acording the information received by speech recognition, we don't implemented a more elaborate algorithm for identifying commands and songs. Another reason for not implementing a more efficient searching algorithm is because the database that we created is small and it would be a computational process that we don't really need.

6. Jukebox Module

As mentioned previously Jukebox Module is an audio processing manager. In this case, it represents the songs which have been previously synthesized with Vocaloid Editor 3.0.

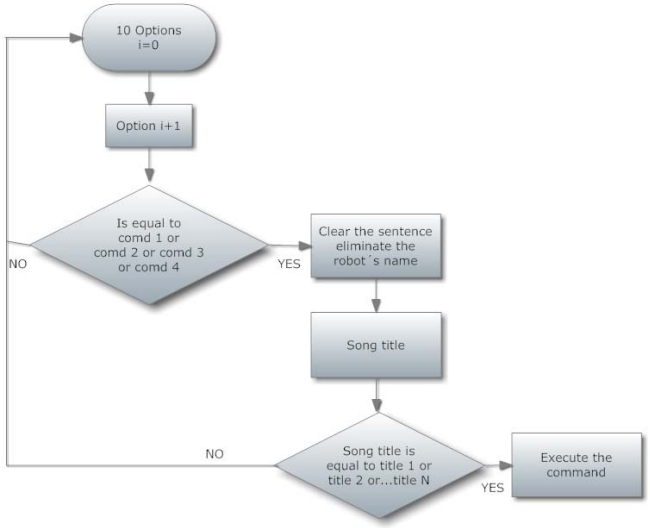

Fig. 4: Command and song title identify flowchart.

Jukebox is able to play files in wav, mp3, wmv, wma and mpg. Basically it has two functions: playing and stopping audio files in the formats mentioned above. These two functions or commands use one ID number to facilitate the execution of commands. In turn, each audio file has an ID number that is assigned in the command in which they were added to the database.

Jukebox receives the command's ID and the song title ID after the song begins playing or is stopped according to what prompted. Virtually, Jukebox plays or stops a song when the command has been executed.

7. Vocaloid Editor 3.0

Vocaloid (ボーカロイド Bōkaroido) Editor 3.0 is an speech application software that is able to sing. It was developed by the Yamaha Corporation in collaboration with the Music Technology Group at the University Pompeu Fabra in Barcelona, Spain.

The software provides the user with the ability to synthesize songs simply by typing the lyrics and melody. It uses the voice synthesize technology specially recorded from dubbing actors or singers. To create a song, the user must enter the melody and lyrics. An interface of a piano roll is used to incorporate the melody and lyrics that can be put into each note [10].

Figure 5 shows the main screen of Vocaloid Editor 3.0. To the left, you can see the piano roll that helps create notes and some notes being made example.

7.1 VocaListener

VocaListener is a plug-in that can be integrated to the Vocaloid Editor 3.0 program. VocaListener focuses on the combination with the Vocaloid Editor 3.0 software

Fig. 5: The main screen of Vocaloid Editor 3.0

and facilitates the work and the burden of making a song with Vocaloid Editor 3.0, and at the same time, improves the quality of the song. VocaListener is an automated online-generated file system and is running on a specific server, allowing users quickly get a good quality of CSA (Vocaloid Sequence) in a quick way, to present it before loading the original file and the voice wave lyrics file. This technology can greatly reduce the work to make a good Vocaloid VSQ file [3].

In Figure 6 we can see at the top an example of how to introduce the song (Acapella Song) and VocaListener generate the voice signal that is showed in the image, therefore the notes are created for this input signal. We can also see an example of the notes as they are created and how you entered the letters of the same manner that the letters should be introduced in Japanese using hiragana.

7.2 Synthesizing a song using VocaListener

There are several ways to synthesizing a song using Vocaloid Editor 3.0 but the way you have best results is when using VocaListener.

To synthesize a song using VocaListener, it is necessary to have the song's acapella version in wav format. Then the file is exported to VocaListener and this generates the signal corresponding to the voice, this signal is what helps us generate the notes on the size and scale necessary to go manually entering only the lyric of the song, remembering that we have to respect the hiragana letters and we must introduced the lyric in Japanese.

Since it has been introduced the letter and notes were generated need to listen and edit manually synthesizing as VocaListener not a perfect program that can generate so silent moments that cause the sequence of notes lose

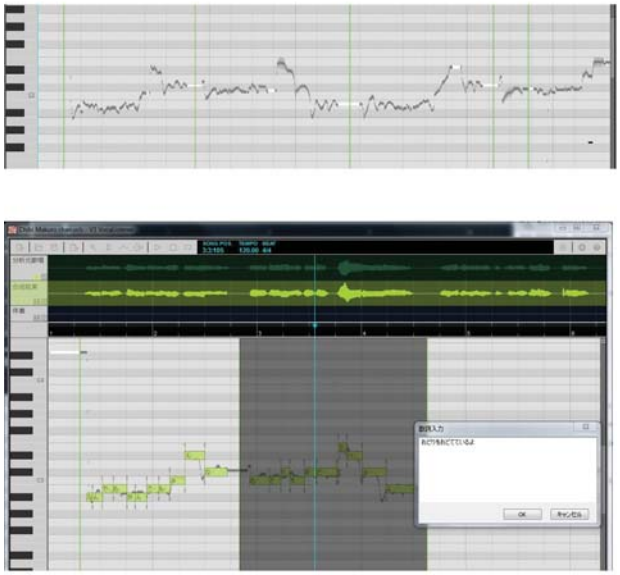

Fig. 6: VocaListener's Main Screen.

quality.

When you finish editing the synthesis according to our requirements is exported to Vocaloid Editor 3.0 the sequence of notes which is what allows us to manipulate the voice of the interpreter, i.e. the interpreter we can choose to be a male or female voice. Finally this synthesized song is exported database where Jukebox can take it to be reproduce.

8. Experimental Results

Being able to sing as part as a module adds an additional function to the robot, which in this case as the name says is a feature that allows the robot to sing. This function as seen above is activated by voice no matter who is the speaker, but the commands are specific, i.e. for the module is activated you need to ask the robot to sing a song, we have to remember that the robot base language is Japanese.

The module makes singing the prayers processing provided by speech recognition that facilitates the rapid identification of errors. We have different types of possible errors are detected. Then we'll talk a little about them.

- Error 1: The Singer Module first searches the command is known, therefore if within ten options provided by speech recognition known no command module discards this information and start the process again so the user will have to repeat the command.

Table 1: Commands executed.

	Commands Numbers	Times Repeat the Command	Type Error	Type Command
Execute	91	1/1	0	0
Error	9	2 (3 times repeat, first and second time causes an error on third time the command is executed)	4 (song)	4 (play)
		7 (2 times repeat, first time cause error second time the command is executed)	4(command)	2 (stop)
				2(play)
			3 (song)	3(play)

- Error 2: The following error has to do with the song and here are two possibilities:
  - Error 2.1:
    Taking into account that the command has been detected as known in this case we proceed to find the song but if the song is not in the database the Singer Module will detect this error and then this process will reboot and the user will have to repeat the by the robot.
  - Error 2.2 If the song is in the database but speech recognition is not optimal, i.e. has the wrong name, then reboot process and the user will have to repeat the command to the robot. These errors have to do with the command to sing (play), in case the command you ask the robot is the stopping (stop) then there can only be one type of error is that the command is not identified as known and the user will have to repeat it.

To test the operation of the Singer Module became the next test. He gave the commands system randomly hoping that properly executed which run ninety-one times correctly, i.e. that it was only necessary once for both Speech Recognition Module singing as operate properly. These results we can observe in the table 1.

The remaining nine times an error was generated from those already mentioned above. In this case two of those nine times that there was error was necessary to repeat the command three times, i.e. the first and the second time the command was given and error was detected until the third time that the command was properly recognized and enforced, for this case, the error occurred in the command "play" and due to a mistake in the name of the song because the name was not detected correctly.

On the other hand seven times was necessary to repeat the command twice, i.e. the first time you said the command was a mistake and was present until the second time the command was said that this was identified and executed properly. Four times the error is present because the command was not correctly identified by speech recognition. It is noteworthy that were twice as was the command "play" and twice the command "stop" respectively which were not identified correctly.

Finally three times was no error in the command "play" because the song is not found as the name of the song provided by the speech recognition did not match that of the database.

From this we conclude that on average 90.09% commands are executed correctly and also on average it can fail 9.90% of the time, that the best you only need to repeat the command a second time for this to run properly.

9. Conclusions

Using the Singer Module we can obtain a new function that the robot will be able to perfom perform in this case is to sing a song when prompted to do so.

The Singer Module as we saw earlier in the process needs the help of Speech Recognition Module and Jukebox Module, it is important to mention that for the Speech Recognition Module its really complex identify long sentences as is the command they need to activate the Singer Module, so one of the qualities of Singer Module is processing information that provides speech recognition and extract the important parts of the sentences that receives and analyze, identify errors if even present and otherwise execute the command you have received.

As mentioned in the results section the 90% of the commands that are given to the system run successfully, these are good results since we can say that the processing module performs Singer is optimal and efficient.

We note that speech recognition improves if the person giving the commands have a complete mastery of the language in this case Japanese because as mentioned in section Julius Speech Recognition Module is a program designed especially for the recognition of Japanese. Furthermore Vocaloid Editor 3.0 was chosen as base software for creating songs as this is a software dedicated to speech synthesis based on letters and notes here.

For the creation o the songs, those were recorder in "acapella" versions in order to use the tool VocaListener and generate more quality songs. Vocaloid Editor 3.0 is one of the best programs for creating synthesized

speech, artificial voice in other words, so far has not been possible for humans to create a perfect speech cast by a robot or a machine, Vocaloid Editor 3.0 is as close to a human voice as the software allows modification of some features of voice such as vibrato, pitch, stress of the pronunciations, changing dynamics and tone of voice.

As future work is to further expand the database of songs and create songs in other languages such as English or Spanish. In addition to adding a new process to Singer Module that is dance, i.e. the robot generates some movements either your head or hands, to be in line with the song that played. If the database can be expanded for future work would be to implement a more efficient search algorithm to improve the processing of commands and execute commands faster.

Acknowledgments

This work was completed in Nagai's Laboratory of The University of Electro-Communications. For this reason I want to thank all members of Nagai's Laboratory and specially my professor Nagai who kindly have help me during the last year.

Thanks so much to JUSST program, JUSST program Staff, and JASSO scholarship for supporting this research.

Also, special thanks to my advisor Enrique Escamilla Hernandez from National Polytechnic Institute of Mexico because he gave me the oportunity to come to Japan and do this work.

Finally thanks to Rizki Satya Utami and Fukui Miyuki by giving me permission to record their voices.

References

[1] M. Attamimi, A. Mizutani, T. Nakamura, K. Sugiura, T. Nagai, N. Iwahashi, H. Okada and T. Omori, "Learning Novel Objects Using Out-of-Vocabulary Word Segmentation and Object Extraction for Home Assistant Robots," *in Proc. ICRA*, 2010, p.745-750.

[2] H.Okada, T. Omori, N. Iwahashi, K. Sugiura, T. Nagai, N. Watanabe, A. Mizutani, T. Nakamura and M. Attamimi, "Team eR@sers 2009 in the @Home League Team Description Paper," *in Proc. RoboCup International Symposium*, 2009.

[3] T. Nakano, M. Goto, "VocaListener: A singing-to-singing synthesis system based on iterative parameter estimation,"*in Proc. of SMC*, 2009, p. 343-348.

[4] M. Goto, "Active Music Listening Interfaces Based on Signal Processing," *in Proc. of ICASSP*, 2007, p. 1441-1444.

[5] M. Goto, "A Chorus-Section Detection Method for Musical Audio Signals and Its Application to a Music Listening Station," *IEEE Transactions on Audio, Speech and Language Processing*,2006, Vol. 14, No.5, pp. 1783-1794.

[6] M. Goto, "Music Listening in the Future: Augmented Music-Understanding Interfaces and Crowd Music Listening," *in Proc. of AES 42nd International Conf. on Semantic Audio*,2011, p. 21-30.

[7] M. Goto, T. Saitou, T. Nakano and F. Fujihara, "Singing Information Processing Based on Singing Voice Modeling," *in Proc. of ICASSP*, 2010, p. 5506-5509.

[8] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. C. Woodland, *The Fundamentals of HTK*, in The HTK Book version 3.4, December 2006, ch. 1, pp. 1-13.

[9] 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 Proc. of WASPAA*,2007, p. 215-218.

[10] H. Kenmochi and H. Ohshita, "VOCALOID - commercial singing synthesizer based on sample concatenation," *in Proc. of ICASSP*,2007, p. 4011-4010.

[11] H.Kenmochi, Yamaha Corp., "Singing synthesis as a new musical instrument,"*in Proc. Of ICASSP*, 2012, p. 5385-5388.

[12] H. Fujihara, "Singer identification based on accompaniment sound reduction and reliable frame selection," *in Proc. of ISMIR*, 2005, p. 329-336.

[13] H. Fujihara and M. Goto, "A music information retrieval system based on singing voice timbre," *in Proc. of ISMIR*, 2007, p. 467-470.

[14] T. Nakano, "Voice Drummer: A music notation interface of drum sounds using voice percussion input," *in Proc. of UIST*, 2005, p. 49-50.

[15] (On line), [22 de February de 2013], Available online:http://julius.sourceforge.jp //en_index.php