MUSICAL AND PHONETIC CONTROLS IN A SINGING VOICE SYNTHESIZER

by

Jaume Ortolà i Font

Submitted to the Polytechnics University of Valencia
in fulfillment of the thesis requirement for the degree of
Engineer in Telecommunications

Directed by Luis Vergara

October 2001

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author wishes to thank the people who have been working in the Daisy Project in the Pompeu Fabra University in Barcelona: Òscar Celma, Àlex Loscos, Jordi Bonada and our director Xavier Serra. I am also grateful to Shigeki Fuji and Hideki Kenmochi, from Yamaha Corporation, who have directed the project in Hamamatsu (Japan) and have visited us regularly in Barcelona. The rest of the people in the Music Technology Group have also to be mentioned, they have been always available to give us a hand when needed. In addition, I want to express my gratitude to Luis Vergara who has directed this thesis from the Polytechnics University of Valencia.

INTRODUCTION

This dissertation describes my work in the Daisy project, which since February 2000 is being developed by the Music Technology Group (Audiovisual Institute, Pompeu Fabra University, Barcelona) in collaboration with Yamaha Corporation.

The Music Technology Group, MTG, is a research group working on signal processing techniques for musical production and for other multimedia applications. Apart from pursuing the development of spectral audio models, MTG is dedicated to sound models for synthesis, the processing of audio based content and other issues related to Music Technology.

On the other hand, Yamaha Corporation manufactures all kinds of musical instruments and professional audio equipment for professionals and amateur enthusiasts. From its base in Hamamatsu City, southwest of Tokyo, the company is also a leading producer of audiovisual products, semiconductors and other computer-related products, electronic equipment and specialty metals.

The aim of the Daisy project is to synthesize a singing voice from a musical score. That is to say, from a given musical melody and a given lyrics in a particular language (English and Japanese in our case) our goal is to obtain an output sound as though a real singer was performing a song. Of course, this is not an easy thing, to say the least. However, with the accumulated knowledge in different fields, the use of new technologies and the increasing power of computers, this objective has become achievable nowadays. The artistic and technical disciplines relevant to this project cover an impressive variety of fields: sound recording and reproduction, music performance, music perception, phonetics, computer programming, digital signal processing… We can say that this is a really multidisciplinary enterprise.

This research project presented here is a continuation of an automatic singing voice impersonator application for karaoke developed by the Music Technology Group [Cano, Loscos, Bonada, de Boer, Serra, 2000]. That system morphed in real time the voice attributes of a user (such as pitch, timbre, vibrato and articulations) with the ones from a prerecorded singer.

Because of my education as a musician and as an engineer, I have always been willing to work in an area in which I could apply my knowledge in both fields. And from the first time I heard about the MTG group, I knew that I would like to collaborate with some of its research projects and to meet the people working there. When I joined the Daisy project team, I was extremely pleased to contribute to such a fascinating project.

I have been working not alone but very close to other people. This is teamwork and because of its great complexity it could not be any other way. Now the project is not yet finished, we are still working and improving the system. There are still many things to do. In this report, I will try to give an explanation of the whole system, but I will only provide detailed descriptions in the points directly related to my personal work. My personal assignments in the Daisy project are related basically to the musical input controls, expressiveness control and phonemes timing. These will be my principal focus in this report. I will not give full details on the synthesis process but an overall explanation of it.

Chapter One

OBJECTIVES AND BACKGROUND

1. OBJECTIVES

The human voice is clearly the most flexible and fascinating of the musical instruments. All along the history of music, it has attracted the attention of composers and has captivated audiences. Already organ builders had the dream to imitate as faithfully as possible the sound of human voice, as we can hear in the Vox Humana stop in some organs. Nowadays this dream has become achievable thanks to the digital signal processing techniques and the increasing capabilities of computers. However, although there are several systems that perform singing voice synthesis, the results are far from a good and realistic reproduction of the human voice.

Many efforts have been devoted to speech synthesis. Text-to-speech systems have a wide range of applications and there are large commercial interests in this subject. Conversely, in singing voice synthesis, the applications are limited to the entertainment and the musical production fields. For that reason, less effort have been invested and there is still a lot of room for research and improvement. We have to say also that, in spite of this, singing voice synthesis is becoming a very competitive area in the last years, with many research groups and companies that are trying to enhance the current technology. The goal of having the best singing voice synthesizer in the market is a great lure to carry out our project.

From the beginning of the project, the keywords in our task have been quality and naturalness in the synthetic output sound. We can understand quality and naturalness as usually it is understood in hi-fi recording and playback systems: a high cut-off frequency and a good resolution, i.e. 44100 Hz and 16 bits in a standard CD format. Nevertheless, we knew also from the beginning that this was not enough to our purpose. There is another concept that accounts for what we are trying to achieve: musical expressiveness. It is even more significant than mere 'sound quality'. It is the reason why we can listen to interesting and communicative musical performances in old recordings with poor 'sound quality'. This is, needless to say, a fuzzy concept, not easy to understand or to explain precisely. The reason is that expressiveness is influenced by numerous factors at different levels in the musical communication process. Anyway, we could not avoid confronting this question if we wanted to succeed in our goal of a realistic singing synthesis. Therefore, in addition

to the digital signal processing techniques that we use to synthesize the singing voice, we have to take care of other important issues related to musical control parameters and expressiveness in order to achieve the desired naturalness in our synthesis.

Summing up, we can say that the aim of the Daisy project is to achieve a realistic and natural singing voice synthesis. The fundamental feature to consider as an evaluation criterion is naturalness. The goal of my personal work in the project is to define and implement the musical input controls and the expressive controls that are needed to give musicality to the synthetic voice. Besides, I'm also responsible of the phonetic database construction and the phonetic timing in the synthesis.

2. HISTORY OF SINGING VOICE SYNTHESIS

We have to go back to the eighteenth century to find the first attempts of generating a human voice with a machine (with a mechanical model). Ch. G. Kratzenstein made the first experiments. In 1773 he was able to generate vowel sounds by using resonance pipes connected to a bellows.

At the same time, inventor Wolfgang Von Kempelen, known as the first speech researcher, had begun to design a machine for phonetic sounds production. Von Kempelen wrote the complete specification of his machine in order to allow other investigators to enhance the device. The Von Kempelen machine was the first to generate not only isolated phonemes but also complete words. It was capable of making nineteen different consonants. However, a full year of practice was required to obtain the desired sounds.

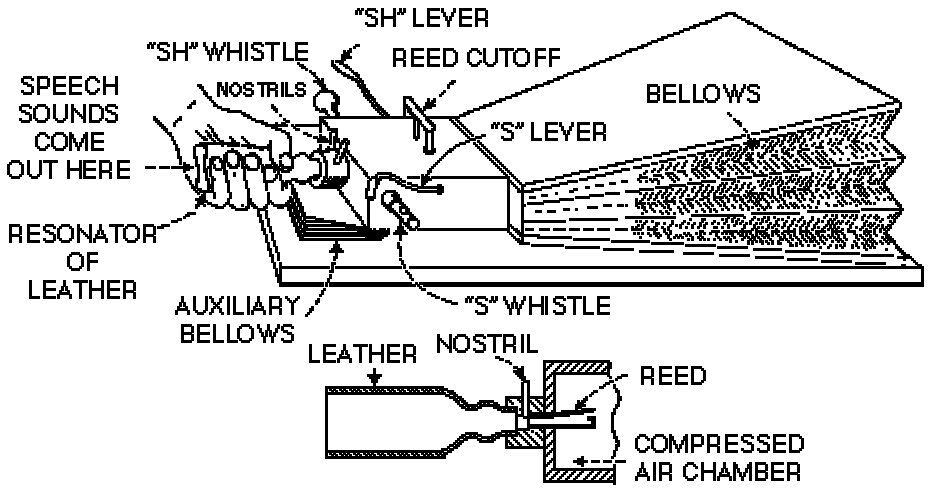

Figure 1. Von Kempelen machine (reproduction by Sir Charles Wheatstone).

During the nineteenth century, several machines were constructed with the same goal, but they didn't contribute substantial innovation to the Von Kempelen machine. We have to mention the machine designed by Joseph Faber in 1835, called *Euphonia*, which was able to generate synthetic speech and to sing the English anthem "*God Save the Queen*".

Figure 2. Euphonia, designed by Joseph Faber (1835).

Towards the beginning of the twentieth century, thanks to the electrical engineering development, it was possible to generate speech sounds by means of electrical signals. The first device was VODER, designed by Homer Dudley at Bell Labs, and was presented successfully in New York in 1939.

Although the main use of these machines by people was flippant (they were presented like an entertainment or like a curiosity), the aim of its designers was very different. For instance, Von Kempelen developed his machine as he studied human voice mechanisms. Homer Dudley based his invention on the Vocoder (Voice Coder), whose purpose was to reduce bandwidth in telephonic transmissions.

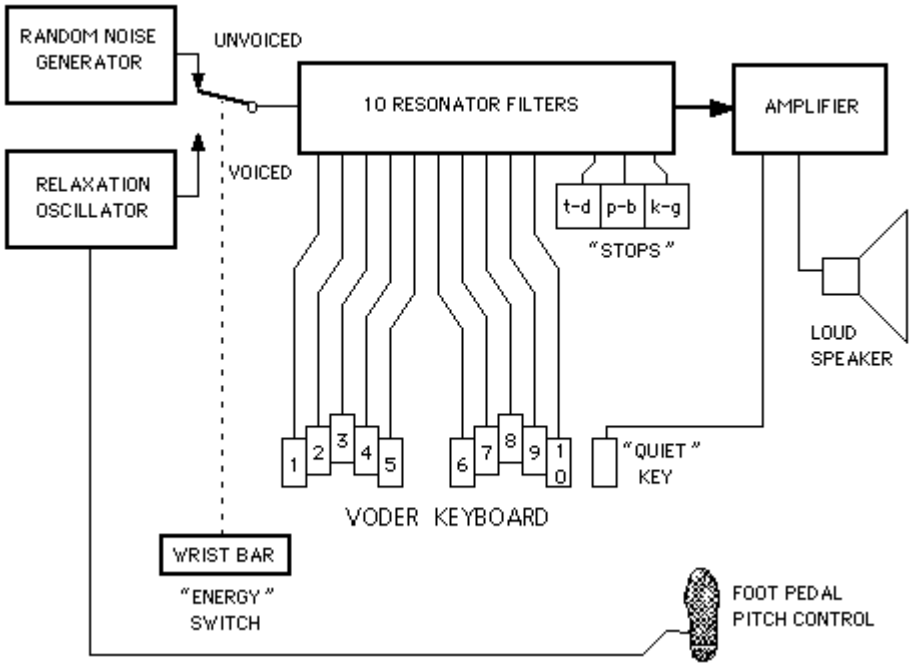

Figure 3. VODER scheme.The first singing voice synthesizer based on electrical signals.

The first use of digital technology in singing voice synthesis was in 1961. This year John Kelly at Bell Labs programmed a computer to sing a song, using the Linear Predictive Coding (LPC) technique. The tune it warbled was: "Daisy Bell (Bicycle Built for Two)". Seven years later, in 1968, for the first time, a computer (HAL 9000) costarred in a movie, Stanley Kubrick's "2001: A Space Odyssey". In the film, as HAL's memory is unplugged, it sings "Daisy Bell (Bicycle Built for Two)", the same tune John Kelly's computer had sung seven years earlier.

Some commercial software products for singing synthesis have been released in recent years. For example, Vocalwriter [VOCALWRITER, 2000] for the English language and SmartTalk 3.0 [SMARTTALK, 2000] for the Japanese language have to be mentioned. However, the systems developed until now are far from providing enough quality to meet the practical requirements of real-world commercial applications.

The goal of a singing voice synthesis indistinguishable from a real human voice is still remote. Thus, there is a lot of room for improvement in this research area, and naturalness is one of the keywords for the work to be done. Moreover, it seems that one of the main issues behind singing voice synthesis is to offer not only quality but also flexible and musically meaningful control over the vocal sound. In that sense, we may think of applications where impossible singing voices can be synthesized or where existing voices can be enhanced.

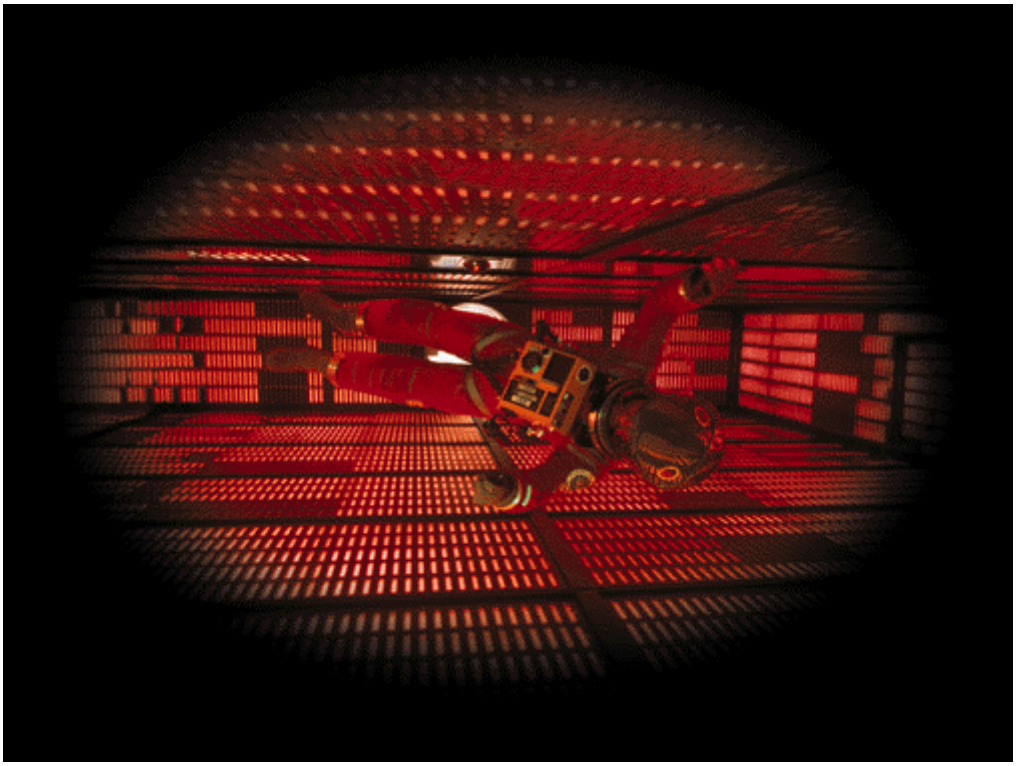

Figure 4. Stanley Kubrick's "2001: A Space Odyssey".

3. SINGING VOICE SYNTHESIS TECHNIQUES

Singing voice synthesis has been an active research field for almost fifty years [Cook, 1996]. Traditionally, the voice has been modeled as a linear system consisting of one or more sources and a set of filters that shape the spectrum of the sources. The sound source can be a periodic signal, a noisy signal, or a mixture of both, and the set of filters can be regarded as the vocal tract filters. The resulting spectrum is characterized by resonant peaks called formants. Thus a vocal synthesizer has to allow the control of the resonant peaks of the spectrum and of the source parameters.

With regard to the synthesis models used in singing voice synthesis, they can be classified into two groups: Spectral models, which can be viewed as based on perceptual mechanisms, and Physical models, which can be viewed as based on production mechanisms. Any of these two models might be considered as suitable depending on the specific requirements of the application, or may even be combined to take advantage of both approaches.

The main benefit of using Physical models is that the parameters used in the model are closely related to the ones a singer uses to control his/her own vocal system. As such, some knowledge of the real-world mechanism can be introduced in the design. The model itself can provide intuitive parameters if it is constructed with the intention that it sufficiently matches the physical system. Conversely, such a system usually has a large number of parameters. This turns the mapping of the controls of the production mechanism to the final output, and so to the listener's perceived quality, into something not trivial.

On the other hand, Spectral models are closely related to some aspects of the human perceptual mechanism. Changes in the parameters of a spectral model can be more easily mapped to a change of sensation in the listener. Yet parameter spaces yielded by these systems are not necessarily the most natural ones for manipulation. The methods based on spectral models include Frequency Modulation, FOFs, Vocoder and sinusoidal models. Acoustic tube models are an example of physical models. Linear Predictive Coding (LPC) and formant synthesizers can be considered as spectral models and also as pseudo-physical, not strictly physical because of the source/filter decomposition they use.

3.1. Spectral Subband Vocoders

From the legacy of speech signal processing came the powerful and flexible techniques known as the spectral subband vocoders (Voice CODERs). In the vocoder, the spectrum is broken into sections called subbands, and the information in each subband is analyzed. The analyzed parameters are then stored for reconstruction in another time or physical site. The parametric data representing the information in each subband can be manipulated, yielding transformations such as pitch or time shifting or spectral shaping.

3.2. Linear Prediction

Linear Predictive Coding (LPC) involves forming a digital filter that predicts the next time sample from a linear combination of previous samples. An error signal is yielded which, if fed back through the time-varying prediction filter, will yield exactly the original signal. The filter models linear correlations in the signal, which correspond to spectral features such as formants. The error signal models the input to the formant filter, and typically is periodic and impulsive for voiced speech, and noise-like for unvoiced speech. The success of LPC in representing speech signals is due to the similarity the source/filter decomposition of linear prediction and the source/filter model of the human vocal tract. However, LPC often sounds artificial because the voice has multiple possible sources of non-linear behavior.

3.3. Frequency Modulation

Frequency Modulation (FM) involves modulating the frequency of one oscillator (the carrier) with the output of another (the modulator) to create a spread spectrum consisting of sidebands surrounding the carrier frequency. In vocal modeling, carriers placed near formant locations in the spectrum are modulated by a common modulator oscillator operating at the voice fundamental frequency. The carrier frequencies must be integer multiples of the modulator frequency. For this reason, it is impossible to generate vocal sounds that smoothly vary from vowel to vowel.

3.4. FOFs

Formant Wave Functions (FOFs in French) represent time-domain waveform models of the impulse responses of individual formants. These are characterized as a sinusoid at the formant center frequency with an amplitude value that rises rapidly upon excitation and decays exponentially. The control parameters define the center frequency and bandwidth of the formant being modeled, and the rate at which the FOFs are generated and added determines the fundamental frequency of the sound. The synthesis system for controlling FOFs was developed by IRCAM and was dubbed CHANT.

3.5. Formant Filter Models

Second order resonant filters can be used to model formants directly. Fourier or LPC analysis can be used to automatically extract formant frequencies, bandwidths and source parameters from recorded speech. The Swedish Royal Institute of Technology created the MUSSE DIG with this technique.

3.6. Physical Models

Acoustic tube models simulate the vocal tract transfer function by solving the one-dimensional wave equation inside a smoothly varying tube. The one dimensional approximation is justified by noting that the length of the vocal tract is significantly larger than any width dimension, and thus the longitudinal modes dominate the resonance structure up to about 4000 Hz. Modal standing waves in an acoustic tube correspond to the formants. Speech modeling work at Bell Labs included the acoustic tube model of Kelly and Lochbaum. The SPAM and Singer systems, developed by Perry Cook, are based on the same physical model.

3.7. Spectral Modeling Synthesis

Simply performing a Fourier transform on speech data does not yield a parameterization that is useful beyond simple pitch and time manipulations. Spectral Modeling Synthesis (SMS) includes two parts: a sinusoidal model and a residual model. In the sinusoidal model, SMS uses Fourier analysis to locate and track individual sinusoidal partials in the sound signal. Individual trajectories of sinusoidal (amplitude, frequency and phase) as a function of time are extracted from the time

varying peaks in a series of Short Time Fourier Transforms (STFT). The sinusoids can be resynthesized from the track parameters, after modification or coding, by additive synthesis. Noise can be treated as rapidly varying sinusoids or explicitly as a non-sinusoidal, stochastic component. Since this is one of the techniques we use in our system, it is further explained in chapter three.

4. TEXT-TO-SPEECH AND SINGING VOICE SYNTHESIS

In the 1970's, research on speech synthesis was already linked to digital technology. In this period, the first Text-to-speech systems (TTS) were developed, which incorporate digital signal processing and natural language processing techniques. One of the main issues to solve in text-to-speech synthesis is prosody, rhythm and intonation problems. Currently, some of the most well known TTS systems are: *Festival Speech Synthesis System*, *MBROLA*, *AT&T Lab's Next-Generation TTS* i *DECtalk*.

First of all, we can ask ourselves whether singing is significantly different from speech. At first sight, it could seem that singing voice is only a particular case of human speech, since singers have trained their vocal mechanism to produce musical sounds. Nevertheless, almost immediately basic differences are observed between speech and singing voice. These divergences make us consider singing voice and speech as two separate research areas.

Certainly, the goals in speech synthesis and in singing voice synthesis are dissimilar to a certain extent. Intelligibility is the main goal in speech synthesis and quality is the main goal in singing voice synthesis. Below there is a list of differences between the two topics:

-Voiced/unvoiced ratio. The ratio between voiced, unvoiced sounds, and silence is about (60%, 25%, 15%) in the case of normal speech. In singing, the percentage of phonation time can increase up to 95% in the case of opera music.

-Loudness. The dynamic range as well as the average loudness is greater in singing than in speech. The spectral characteristics of a voiced sound change with the loudness.

-Fundamental frequency: In speech, the fundamental frequency variations express an emotional state of the speaker or add intelligibility to the spoken words. This frequency range of f0 is very small compared to singing where it can be up to three octaves.

-Vibrato. Two types of vibrato exist in singing. The classical vibrato in opera music corresponds to periodic modulation of the phonation frequency, and in popular music the vibrato implies an added amplitude modulation of the voice source. In speech, no vibrato exists.

-Formants: Since in singing the intelligibility of the phonemic message is often secondary to the intonation and musical expression qualities of the voice, there is an alteration of the formants position, and therefore the perceived vowel is slightly modified.

5. MUSICAL CONTROL LANGUAGES

In this section, musical control and representation languages used in our system are presented. First, an introduction to the standard MIDI protocol is provided. Afterwards, the Metrix format for musical scores representation, which is used in our system, is also described.

5.1. MIDI standard protocol

Since the design of the first musical synthesizers, towards the beginning of the sixties, different musical control systems have been employed. In the early days, each company implemented its own protocols and representation formats according to the designed synthesizer. In 984, MIDI standard was defined and it seemed to put a stop to chaos. Synthesizers manufacturers agreed to define and use a unique communication protocol for interconnecting controllers, synthesizers and sequencers. The first specification was Musical Instrument Digital Interface 1.0. and it has undergone several modifications and additions (General MIDI, Standard MIDI file, MIDI Time Code), but it still maintain its original name and version number. The MIDI specification defines a communication protocol, from the physical layer interface to the data representation and transmission.

The creators of MIDI chose a simple, readily available five-pin plug to put on all MIDI-compatible instruments and cables, called a Din plug. Since MIDI uses a single wire in the cable to send information, the musical data that MIDI sends travels in only one direction over a single cable. However, MIDI was devised to allow information to go both directions between two instruments, by simply using two cables. At the same time, MIDI can also pass data on to a third, fourth, fifth instrument, or as many synthesizers as you can afford. To accomplish this, it was decided to have three different MIDI connectors on each instrument: one to receive data IN, one to send the data OUT and one to pass incoming data on THROUGH (spelled "THRU") to another MIDI instrument.

MIDI uses serial transmission. MIDI sends information at a rate of 31,250 bits per second. All MIDI messages use 8 bits for the information. To guarantee perfect accuracy when MIDI data is being transmitted, two additional bits are used in every byte, bringing the total up to ten bits per MIDI message byte. Which means that it takes 320 microseconds for each message byte.

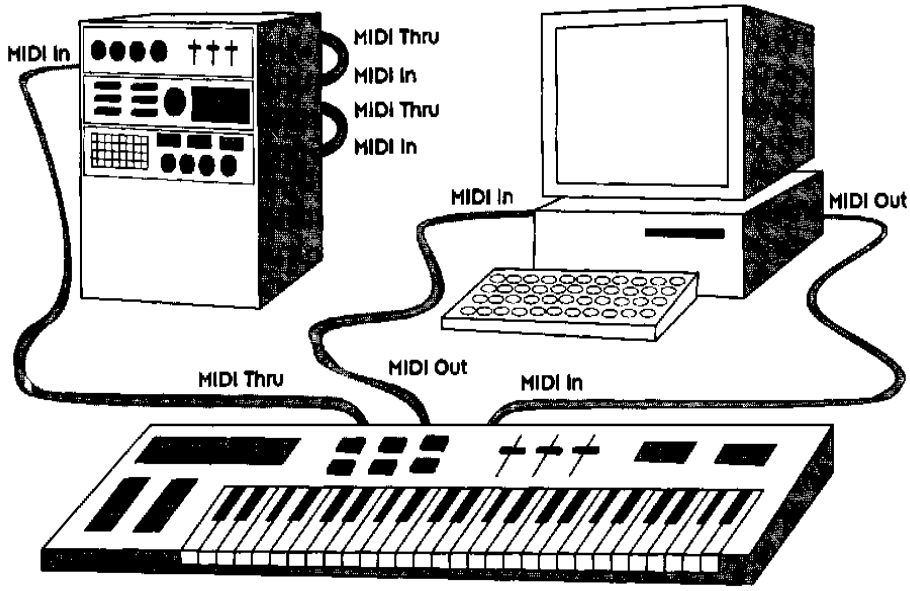

Figure 5. MIDI devices connection.

A MIDI message can have a variable length. In any case, the first byte is always the status byte. The status bytes have a '1' in its more significant bit and the data bytes have a '0'. Since the first bit is used to identify status and data bytes, there are 7 bits left to send information. All the system users know this value (2^7=128) and it is one of its limitations (128 notes, 128 instruments per bank, 128 values for every parameter...)

There are two different categories of messages: channel message and system message. In the first case, the status byte includes the type of message and the channel (0-15) affected by the message. On the other hand, a system message affects all the channels.

Channel messages

A status byte in a channel message has this structure:

1	**m**	**m**	**m**	**c**	**c**	**c**	**c**

where:

1: is the status byte mark.

m: bits with the type of channel message sent (0-7).

c: bits with the channel number (0-15).

There are seven different orders that can be sent in a channel message:

- Note On
- Note Off
- Control Change
- Program Change
- Aftertouch
- Polyphonic aftertouch
- Pitch bend

Below some of these messages are explained.
Note On and Note Off:

These are the two orders more frequently used in the MIDI protocol. Its meaning is to begin the note sound (note on) and to stop the sound (note off). The note on message has this bit structure:

1	0	0	0	**n**	**n**	**n**	**n**

The four bits 'nnnn' is the channel number which note is send through. This status byte is followed by two data bytes. The first indicates the note and the second indicates the loudness (key velocity). The note will stop sounding when a note off message is received. The note off message has the same structure that the note on message. It is common, instead of sending a note off message, to send a note on message with key velocity equals to zero, which is equivalent. The MIDI note numbers begin at 0, which is the lower note. The note number 60 is C3.

Control Change:

The Control Change messages are related to actions performed from some device: a pedal, a joystick, a slider, etc. Each order has to parts. The first one defines the control that has to be modified and the second the value. There are some controls that are defined by the MIDI specification and listed below; the most of them are available for the manufacturer to define its own controls.

- 1 Mod wheel
- 2 Breath controller

- 4 Foot controller
- 5 Portamento time
- 6 Data entry knob
- 7 Main Volume
- 8 Balance
- 10 Pan
- 11 Expression
- 64 Sustain
- 65 Portamento
- 66 Sostenuto

The Control Change messages have usually one data byte and a range of values between 0 and 127.

System messages

System messages are not sent to a particular channel, since they control global parameters of the synthesizer.

System Exclusive messages

System Exclusive messages are messages defined by each manufacturer.

5.2. Metrix

Metrix is a musical description language designed for controlling a synthesizer based on spectral models [Amatriain, 1999]. Within the Metrix specification, the Musical *Score Description Language* (MSDL) is a control language based on text, without low-level synthesis parameters, which is understandable to a user with some musical knowledge.

Metrix data can be stored in a standard ASCII text file and can be edited with any word processor. The score has two separate parts: header and body. In the header, general score attributes are defined:

```
Score_Info{
      Tempo:      120
      Meter:      4/4
}
```

After that, the instrument or instruments used in the score are defined:

```
Instrument_Info {
      DaisyVoice
}
```

The body contains all the musical events that are to be synthesized. Events are written between the reserved words *begin* and *end*. The body is a list of ordered events. Each event has a time stamp, an instrument declaration and a list of parameters with its corresponding values. Each parameter has associated a data type. A standard event has this syntax:

$$T\ V\ (P{:}PV)^{+}$$

where,

T is Time stamp

V is Instrument declaration

P is Parameter

PV is Parameter value

An event is shown in the next example. At time 0.5 seconds, the previously defined instrument (DaisyVoice) has to synthesize a note with four different parameters (NoteNumber, Duration, Lyrics, Loudness) and its values (G3, t0.5, "j u:", 0.6):

```
t0.5  DaisyVoice  NoteNumber: G3
                  Duration: t0.5
                  Lyrics: "j u:"
                  Loudness: 0.6
```

Below there is a complete example of a score in Metrix format. It is the beginning of the song "As Time Goes By".

```
Score_Info {
       Tempo: 80
       Meter: 4/4
}
Instrument_Info {
       DaisyVoice
}

begin
t0.5          DaisyVoice    NoteNumber: G3
                            _Duration: t0.5
                            Lyrics: "j U"
                            _Loudness: 0.6

t1            DaisyVoice    NoteNumber: Ab3
                            Lyrics: "m V s"

t1.5          DaisyVoice    NoteNumber: G3
                            Lyrics: "r I"

t2            DaisyVoice    NoteNumber: F3
                            Lyrics: "m e m"
                            Loudness: 0.5

t2.5          DaisyVoice    NoteNumber: Eb3
                            Lyrics: "b @ r"
                            Loudness: 0.4

t3            DaisyVoice    NoteNumber: F3      Duration: t1
                            Lyrics: "D I s"
                            Loudness: 0.3
end
```

The Metrix specification has been modified and adapted according to the needs of the Daisy system. A set of new parameters, required for representing all the features of a singing voice, has been added. Besides, a new element has been added to the Metrix syntax in order to facilitate the edition process of the score. Parameters with an underscore will be applied over all the next notes of the song unless something else is stated explicitly. In the next example, the first note defines a general behavior that will affect implicitly the next notes. That is to say, all notes will have the same duration and the same syllable.

```
begin
t0.5        DaisyVoice  NoteNumber: G3
                        _Duration: t0.5
                        _Lyrics: "m u: n"
t1          DaisyVoice  NoteNumber: Ab3
t1.5        DaisyVoice  NoteNumber: G3
t2          DaisyVoice  NoteNumber: F3
t2.5        DaisyVoice  NoteNumber: Eb3
end
```

6. RESEARCH ON MUSICAL EXPRESSIVENESS

Systematic studies on musical expressiveness appeared for the first time at the beginnings of the 20th century and it has become a research field of increasing interest and promising achievements in the future. In the musical tradition, musicians teach and learn about expressiveness not always in a systematic way but most of the time intuitively. Although most of the music performers are unable to put into words what they do with their instruments, expressiveness in the context of musical communication is something that can be learned or improved by oneself or with the help of a teacher. It is very known the case of Franz List, probably the most portentous pianist of the history, who was totally incapable of describing his own technique [Kaemperer, 1968]. In fact, as is known in the present day, he explained to his pupils the contrary to what he really did. He possessed his technique by instinct but without actually understanding it.

The required background to carry out research on musical expressiveness involves a lot of different disciplines. It ranges from psychology of perception and music theory to physics and engineering. The main difficulty is how to handle methodically a subject that is viewed traditionally as purely intuitive and spontaneous. Of course, there is spontaneity in music performance, but in the case of professional musicians it is always the result of a conscientious preparation. For that reason, it can be expected to learn systematically the expressive resources used by singers, or musicians in general, and to apply them in a synthesis engine, allowing users to control the results to a certain extent.

6.1. Musical communication model

In order to better understand the expressiveness concept, we can take a look at figure 6 in which the musical communication process with its principal elements is depicted. The musical communication process, from the composer's mental representation to the listener's perception, can be divided in three basic transformations:

1. Mapping of the composer's mental representation of a musical idea in a score (T_1)
2. The score analysis and interpretation by the performer (T_2)
3. Perception of the music by the listener (T_3)

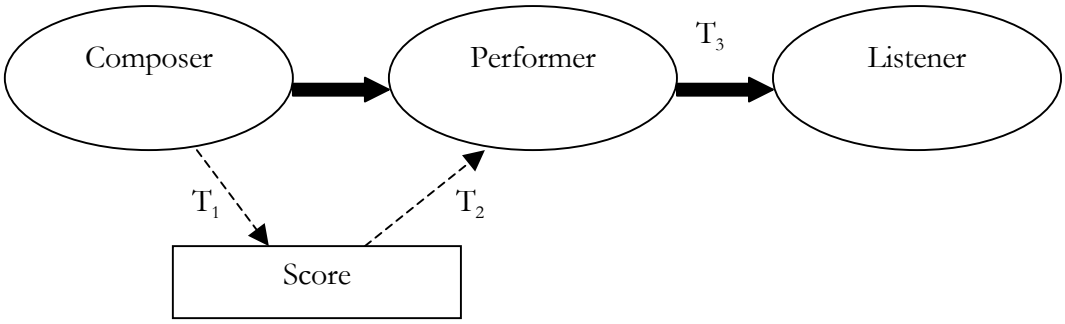

Figure 6. Musical communication process.

It has to be remarked that "precision" can be lost in each transformation from the composer's original idea. In T_1 this can be due to the constraints of a score for representing exactly the composer's intention. In T_2 the performer brings in his own interpretation of the score, adding his own expressiveness and his own view of the score, more or less faithful to the composer's purpose. Finally, in T_3 the listener does his own interpretation of the music according to his musical background and his previous experiences.

In this context, we can see that the expressiveness concept is spread over different layers. Expressive indications are represented one way or another in the score but in a very vague and imprecise manner. From that inaccurate representation, the performer is responsible of introducing a wide range of micro variations in order to do more understandable the musical message to the listener. The purpose of the research on musical expressiveness is to understand how these variations brought in by the performer work and accomplish its goals. In addition, the problem of musical expressiveness includes also how to generate automatically expressive performances according to a determinate musical style, a particular singer style or emotion from a musical score without, or with few, expressive indications.

Music theorists have tried to relate expression patterns to musical structure. In most cases, the size and shape of expressive gestures are assumed to be more or less closely linked to the phrase or grouping structure at various levels (from the complete piece to the single note) of the piece being performed. Thus, the main interest is in finding the regularities and mechanisms that connect musical structure and expressive gestures. That is to say, to model the functions accomplished by

the performer in the musical communication process. These functions can be clearly separated in two: analysis and interpretation.

In the analysis process, the musical structure of the piece has to be apprehended. This normally means to find out the grouping structure, that is, the most basic component of musical understanding, expressing a hierarchical segmentation of the piece into units such as motives, phrases, sections, etc. [Lerdahl and Jackendoff, 1983]. At least in tonal music, it is also necessary to analyze harmony. This means the study of the structure, progression, and relation of the chords.

In the interpretation process, the performer introduces a set of micro variations that affect different physical parameters of the notes such as duration, loudness, vibrato, attack type, articulation type, etc. These deviations from the written score have to be done in accordance with the previous analysis and also with the musical style or the composer's style.

Performers can carry out these tasks in a more or less conscious or intuitive way according to their background and personal experience. However, in scientific studies the purpose is to find out the mechanisms involved in musical performance and to express them as unambiguously as possible. With this objective, several computational models have been developed during the last two decades. All of them have diverse approaches to the subject, which is due to the different original disciplines of the authors. In the sections below, a summary is given of the most relevant models.

6.1. Analysis-by-Synthesis (Sundberg and Friberg, 1983)

Johan Sundberg and Anders Friberg belong to the Royal Institute of Technology of Stockholm (Department of Speech, Music and Hearing). Since the beginnings of the 80's, this research group has been working on computational models in order to synthesize musical expressive performances.

The model developed by Sundberg and Friberg is a rule-based system, which transforms a score without expressiveness into a meaningful musical performance. The set of defined rules models some of the performance features used by the musicians: dynamics, phrasing, articulation, etc.

The rules that generate expressive performances are obtained by an *analysis-by-synthesis* method. This method has been adapted from one the most employed techniques in the research field of speech synthesis. This is a description of the method. First of all, a hypothesis is formulated with regard to some performance principle. After that, synthetic performances of different musical examples are generated according to the hypothesis. In order to validate the supposition, an expert musician evaluates the resultant syntheses. The hypothesis can be now modified and *tuned* until appropriate results are obtained.

The advantage of this method is that musical perception is used directly in obtaining the rules. It is like a musician who plays and listen to himself/herself in order to improve his/her performance. There are some disadvantages. Conclusions are based on a small set of experts in the studied domain. In addition, they have to be exceptionally skilled: capable of focusing in specific aspects of interpretation and extremely sensitive to little changes and perceptual deviations.

6.2. Analysis by Machine Induction (Widmer, 1992)

Gerhard Widmer is a researcher at the Austrian Research Institute for Artificial Intelligence (Machine Learning Group). His work is related to machine learning and data mining techniques and, in particular, its application to the musical research field.

Widmer's approach is based on using artificial intelligence methods in order to discover general expressiveness principles in musical performances. It tries to answer two questions. Is there a general musical knowledge that allows understanding and explaining expressive variations? And, if this knowledge exists, can it be made explicit and be modeled in a computational model?

The research is divided in two related tasks in order to answer the two questions. On one hand, actual performances by professional musicians are analyzed through automatic machine learning techniques in order to understand the principles of a expressive performance. On the other hand, a computational model capable of generating expressive performances is developed with the knowledge acquired in the learning phase.

In practice, the problem has to be restricted. Particularly, the research has focused on the study of dynamics and articulation in piano performances. As the objective is to learn from actual performances, it is essential that the recollected data is consistent and unbiased. They have to be generated by expert musicians. [Widmer, 2000]

The computational model used by Widmer is based on theories by [Lerdahl-Jackendoff, 1983] and [Narmour, 1977] which explain certain features of structural listening and the understanding of tonal music. Lerdahl and Jackendoff describe the way the listener creates a musical structure from what he listens to. The authors combine Chomsky's linguistic theory, Shenker's musical theory and some psychological aspects in order to model the formal structure. Narmour formulates a theory on melodic analysis in order to explain the cognitive relation between musical pitches, and proposes a syntactic codification for the analyzed melody.

6.3. The Global Music Interpretation (Clynes, 1983)

Manfred Clynes is a pianist and neurobiologist, and became famous when invented the word *cyborg*. He has an extremely diverse approach to the research on musical expressiveness. His findings, which link music and neurobiology, are in most cases very personal and have generated strong controversy among musicologist.

His most interesting research has focused in developing mathematical functions in order to model some global controls on expressiveness. Thus, Clynes have developed functions for modeling amplitude envelopes and vibrato envelopes of the notes, according to note durations and intervals between notes. Each note has its own amplitude shaping and its own vibrato depth and vibrato rate envelopes. These mathematical functions are useful for musical instruments capable of continuous modulation as the voice, the strings and wind instruments, but are not suitable to others such as the piano, the organ or the guitar.

On the other hand, Clynes proposes a theory named "composer's pulse". According to this controversial theory, he claims to have discovered a set of micro deviations from the written score that have to be applied in a performance in order to reproduce properly the style of determinate well-known composers. Each composer has his own set of deviations that clearly identifies him.

6.4. The Kinematics of Musical Expression (Todd, 1990)

Neil Todd, researcher at the University of Manchester (Department of Psychology), relates kinematical movement with musical expressiveness. His theory is based on the analogy between physical movement in the space (an athletic runner for example) and musical expression [Todd, 1995]. His work is centered in the tempo changes that occur at the beginning and at the end of melodic phrases. Tempo is accelerated a little at the beginning and especially is slowed down at the end of a phrase. In order to explain this tempo variation, an analogy with a runner is made. In the initial state, the runner, who is motionless, accelerates, maintain a steady speed during the race, and finally slow down and stop. Todd's work shows that the speed functions in the runner's case and in musical phrasing are very similar.

6.5. Case-based reasoning

Another approach to musical expressiveness is to use a case-based reasoning system (CBR) in order to infer the suitable expressiveness for a musical score. This technique try to solve problems by observing the way in which similar problems have been solved previously. CBR can be applied in environments where a lot of solved cases are available. The Saxex system [Arcos, 1997] uses CBR in order to add musical expressiveness to a melody played by a saxophone. This approach, compared to a rule-based system, is more flexible as for the modification and extension of the applied musical knowledge, since it is enough to add new expressive samples to the database for expanding the potential of expressive musical fragments generation.

Chapter Two

SYSTEM OVERVIEW AND MUSICAL CONTROLS

1. SYSTEM OVERVIEW

The inputs of our system are the musical score and the lyrics of the song. The output is a sound waveform, a synthetic singing voice. The complexity of the system is enormous because there are many parameters in different levels that influence the quality and the naturalness of the synthesized sound. In order to simplify and to understand better what we are doing, we have divided the system in two parts, as you can see in figure 7.

The first part is the expressive module. The inputs of this module are the lyrics and the musical score without expressiveness (i.e., only quantized duration and pitch values). We need the phonetic transcription of the lyrics in a standard computer format. This transcription can be made easily in an automatic way with an English phonetic dictionary and some rules to change connections between words, as is explained in section four. The musical information can be extracted from a standard MIDI file or from our own text file format, as we will explain later.

The expressiveness module applies a set of rules to transform the input melody and to make it musically meaningful. The output of this module is a very detailed musical score with all the needed features to characterize an expressive performance by a singer. This detailed instructions for performance is what we call MicroScore, and it includes all the needed parameters: general parameters common to all musical instruments and specific parameters for the singing voice. MicroScore is described in the corresponding section.

The second module of the system is the synthesis module. This is responsible for the final synthesis and it takes MicroScore as its input. Internally the synthesizer has a phonetic and timbre database with all the needed data extracted from recordings of real singers. In this report we will not explain in depth how the synthesis process works, but we will give some details on the construction of the databases, especially the chose of phonetic articulations in English and some other musical features like note to note transitions, vibratos, attacks, etc. Besides, we will explain also the first step made by the synthesis engine: to convert the MicroScore data in another internal

format (called DaisyScore) that is directly readable by the synthesis loop. This step involves some important decisions related to phonetics and musical expressiveness.

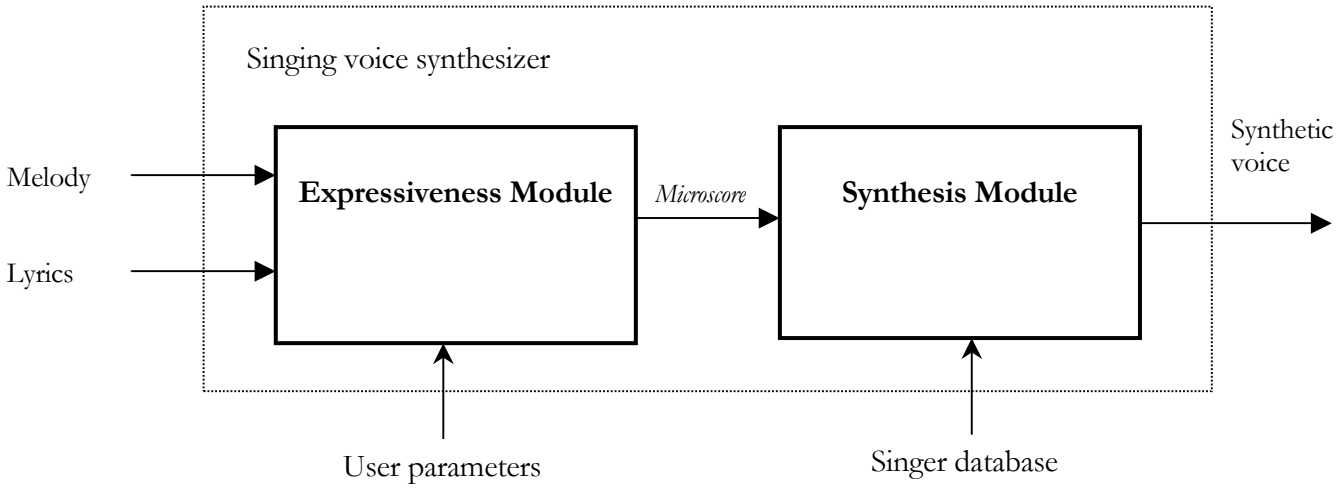

Figure 7. System Overview.

2. MUSICAL CONTROLS

2.1. MicroScore specification

As we have said previously, the MicroScore contains detailed information of musical performance for controlling the singing voice synthesizer. A musical score is a list of ordered events with its corresponding parameters. With this parameters we try to cover so much situations as possible in a musical performance made by a singer. The most of the parameters are related to each musical note (note parameters) and a few affect all the song (control parameters).

Note parameters

Pitch
MIDI number (0...127)

Begin Time
Time in seconds and precision of milliseconds from the beginning of the song when the note starts.

Duration
Note duration in seconds with precision of milliseconds.

Lyrics
Lyrics of the note in SAMPA phonetic transcription. There are two special cases: one note for n syllables and n notes for one syllable (*melisma*). We think that the first case is very unusual and we can write n notes (the same note) for n syllables. So it would be unnecessary to write more than one syllable per note. In the second case, the solution can be to distribute the phonemes along the notes. For example, syllable 'tas' in three notes: ta-a-as.

Attack parameters
Type of attack {*normal, sharp, soft, sexy, high_attack*…}
Attack depth (normalized value)
Attack duration (normalized value)

Release parameters
Type of release
Release depth (normalized value)
Release duration (normalized value)

Vibrato parameters
Type of vibrato
Delay time (in seconds)
Vibrato depth (envelope, in cents)
Vibrato rate (envelope, in Hz)
Vibrato tremolo depth (in dB)

Transition parameters
Type of transition *(molto legato, legato, staccato, molto staccato, portamento, glissando…)*
Transition duration (normalized value).

Pitch Envelope
We need an envelope for the pitch to perform a fine-tuning intonation and other deviations from the tempered pitch inside the note. The envelope length is time-normalized on the note duration

Dynamic Envelope
We want to characterize note's dynamics, such as *ppp, pp, p, mf, f, ff, fff* in a perceptual measure, from 0 to 1. Also we can define an envelope with any number of changing values inside the note duration.

Opening of vowels
Envelope with any number of points: (time, normalized value)

Hoarseness
Envelope with any number of points: (time, normalized value)

Whispering
Envelope with any number of points: (time, normalized value)

Location of the note
Location of note in a musical phrase.
{*0: Inside Phrase, 1: Top of Phrase, 2: End of Phrase, 3: Top of Subphrase, 4: End of Subphrase*}

Control Parameters

Singer
Index of singer

Language
Index of language

Gender
Index of gender {*male, female, child*}

2.1. MicroScore implementation: Metrix and MIDI

MicroScore, as is described in the previous section, is a general and abstract structure than is stored in memory before being read by the synthesis module. Besides, it can be stored in a file or be

transmitted through a communication system. Two different formats have been implemented for these purposes.

For testing and research purposes, the Metrix format have been implemented and used. The Metrix specification has been modified and adapted according to the needs of the Daisy system. A set of new parameters, required for representing all the features of a singing voice, has been added. Below, we show a list of parameters and its value types according to the Daisy project requirements. This list can be easily modified and extended.

Note Parameters:

Parameters	**Value type**	**Observations**
Pitch	Integer	Midi number (0-127) or A3, G#4...
Duration	Time	Seconds
Lyrics	String	
DynamicEnvelope	Envelope	(time, normalized value)
PitchEnvelope	Envelope	(time, cents)
AttackType	Index	
AttackDepth	Float	dB
AttackDuration	Float	
ReleaseType	Index	
ReleaseDepth	Float	dB
ReleaseDuration	Float	
TransitionType	Index	
TransitionDuration	Float	Normalized value
VibratoType	Index	
VibratoDelay	Float	
VibratoDepth	Envelope	(time, cents)
VibratoRate	Envelope	(time, Hz)
VibratoTremoloDepth	Float	dB
Consonant advancement	Float	Normalized value
Hoarseness	Envelope	(time, normalized value)
Whisper	Envelope	(time, normalized value)
Opening of vowels	Envelope	(time, normalized value)
Location	Index	0: Inside Phrase 1: Top of Phrase 2: End of Phrase 3: Top of Sub. 4: End of Subphrase

Control parameters:

Parameters	Value type	Observations
Singer	Index	
Language	Index	
Gender	Integer	0: Male, 1: Female, 2: Child

Some types need further explanation. 'Time' type is anyone of the Metrix time formats. 'Index' type is an enumerated type. Thus, we can write a label or the integer value associated to this label. In 'Envelope' type we can define any number of points we want, or a single discrete value. E.g. [(0, 0.8) (0.5, 0.7) ... (1, 0.2)] or 0.5

On the other hand, a MIDI implementation has been developed in order to meet commercial requirements. Since the standard MIDI protocol do not handle so much parameters as we consider necessary to control properly a singing voice synthesizer, a MIDI specification with System Exclusive messages has been designed in order to contain all the parameters defined in our MicroScore specification. System exclusive messages are not standard but machine and manufacturer-dependant messages. This specification is not given in this dissertation.

In order to be able to handle all these different file implementations a common memory structure is used before the synthesis module and an input/output interface is responsible of making the proper transformations from and to the different formats. This architecture is also useful for saving the results of applying the expressive controls. See figure 8.

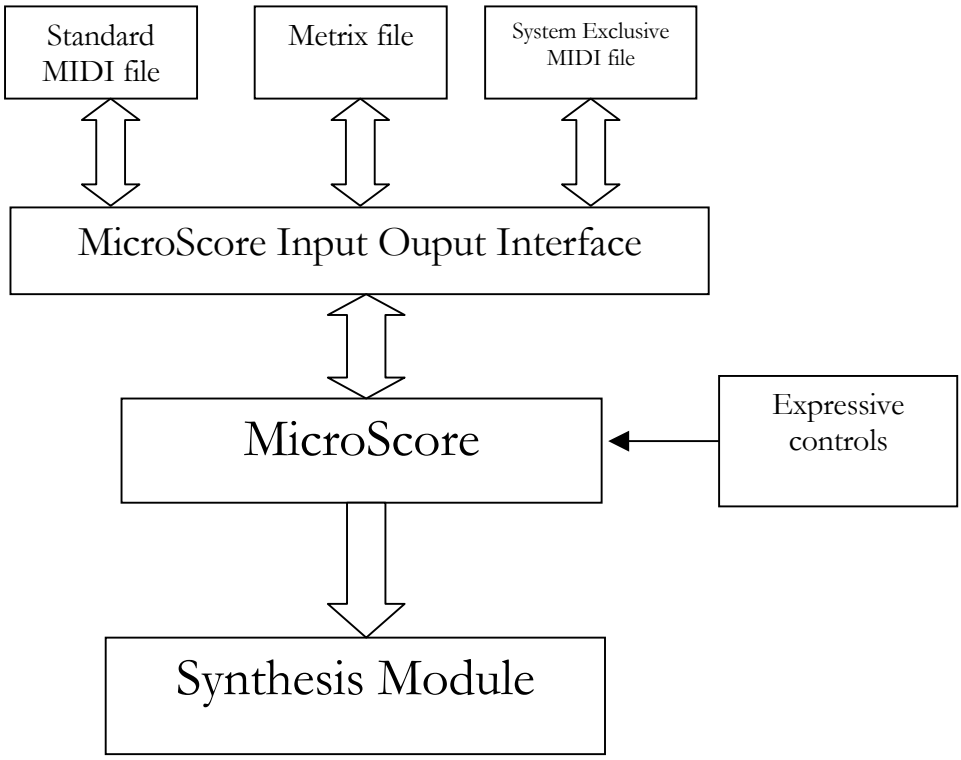

Figure 8. MicroScore Input Output interface.

In addition, in a commercial MIDI application is considered necessary to be able to handle not just a MIDI file but a MIDI streaming flow. For that reason, all the system, from the input to the synthesis loop, has been designed for meeting the real-time requirement. However, since for synthesizing properly a syllable it is necessary to know at least the starting phonemes of the next syllable and the pitch of the next note, a delay of one note event has to be introduced in the system. This is not a problem whether what we want to process is a streaming flow from a sequencer, but it makes impossible for the system to handle in real time a streaming from a human-controlled device such as a keyboard.

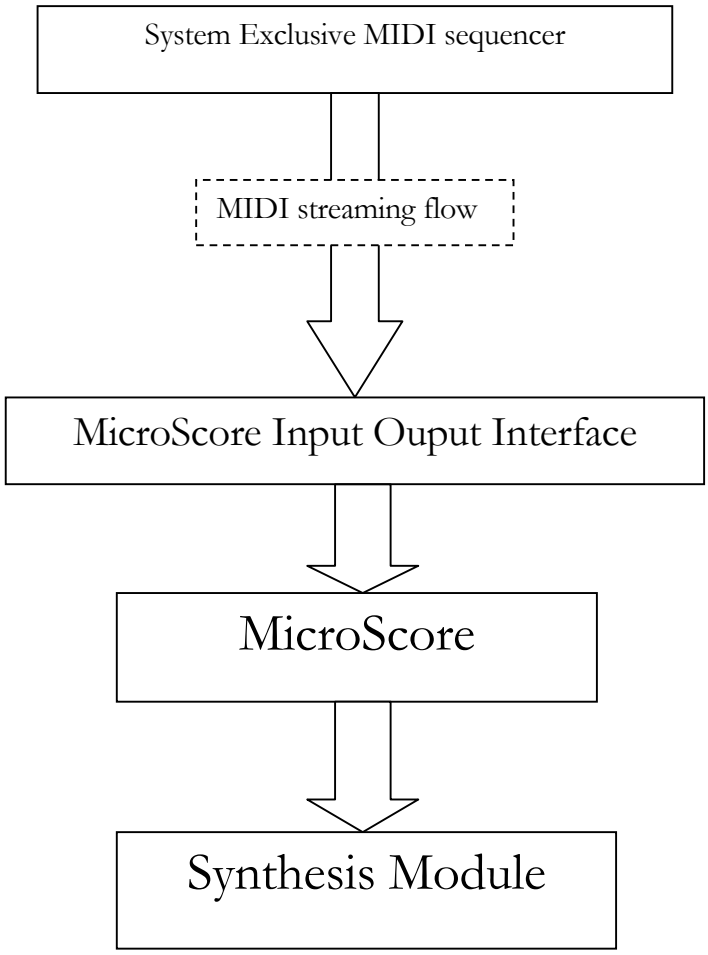

Figure 9. MicroScore input/output interface in streaming mode

3. EXPRESSIVENESS MODULE

The expressiveness module is in charge of applying certain deviations to the input score in order to make it more natural or expressive. Following the research made by Sundberg and Friberg on musical expressiveness [Friberg, 1991], we have implemented a rule-based system for expressive control. These rules were designed as a general tool for any kind of instruments, and need adaptations and proper tuning to be suitable for the singing voice [Berndtsson, 1996]. So far, only a few rules have been successfully tested in our system. We still need more experimentation to include rules specific to voice parameters such as vibrato, breathiness, hoarseness or special kinds of attack.

Some of these rules need additional information as harmonic analysis or grouping structure. This information is introduced manually for now, but could be automated in the future. For automatic harmonic analysis see [Temperley, 1997] and for automatic structural analysis see [Cambouropoulos].

Apart from this rule-based system, some other features have been added to control continuous parameters that are specifically relevant to the singing-voice synthesis.

3.1. Expressive rule-based system

In this section, we list and explain briefly the musical expressiveness rules that we have implemented and are available to generate automatic performances in different styles and emotions. Each rule is controlled by a normalized control parameter between 0 and 1, where 0 means not to apply the rule and 1 means to apply it at its maximum degree. A musical style or emotion can be established by a set of rules with its control parameters [Bresin, 1999].

The rules are separated in two groups: macro level rules which are applied at musical phrase level and micro level rules which are applied at note level. The structural analysis of the score is necessary for applying the macro level rules.

MACRO LEVEL RULES

Phrase Arch.

Tempo curves in form of arches with an initial accelerando and a final ritardando are applied to the phrase structure as defined in the score. The sound level is coupled with the tempo variation so as to create *crescendo* and *diminuendo*. This rule needs a formal analysis to determine where insert boundaries between phrases and subphrases.

Affected parameters: Dynamics and Tempo change

Phrase Final Note.

This rule marks phrases on two hierarchical levels: phrase and subphrase. The last note in a phrase and the last note in the piece are lengthened. After the last note of a phrase or subphrase a micro-pause is inserted.

Affected parameters: Tempo change and articulation.

Harmonic Charge.

This rule marks the distance of the current chord to the root of the current key. Sound level and duration are increased in proportion to the harmonic charge value. The increases and decreases of these parameters are gradual with linear interpolation between chord changes (made by amplitude smoothing rule).

Affected parameters: Dynamics, tempo change and vibrato.

Final Ritardando

The tempo at the end of the piece is decreased according to a square-root function of nominal time (or score position).

Affected parameters: Tempo change

Synchronization rules

There are two options:

- synchronize accompaniment to melody
- synchronize melody to accompaniment. This can be at bar level or at phrase level (with a maximum error margin in *ms*).

Affected parameters: Tempo change.

MICRO LEVEL RULES

Duration Contrast

This rule makes short notes shorter and softer and long notes longer and louder.

Affected parameters: Tempo change, Dynamics.

Duration Contrast Articulation

This rule inserts micropause between two notes. This rule can be used for the purpose of articulation to perform notation conventions (*legato, staccato*…)

Affected parameters: Articulation.

Double Duration

For two notes having the duration ratio 2:1, the short note will be lengthened and the long note shortened.

Affected parameters: Tempo change.

High Loud

This rule increases the loudness in proportion to the pitch height.

Affected parameters: Dynamics.

Melodic Charge

This rule accounts for the "remarkableness" of the tones in relation to the underlying harmony. Sound level, duration and vibrato extents are increased in proportion to the melodic charge value.

Affected parameters: Dynamics, tempo change, vibrato.

Punctuation

The melody can be divided into small musical gestures normally consisting of a few notes, then, the identified gestures are performed by inserting micro-pauses at the boundaries.

Affected parameters: Tempo change, articulation.

Leap Articulation

This rule inserts a micro-pause between the notes in a melodic leap. The length of the micro-pause is proportional to the magnitude of the leap. Included at Punctuation rule as a subrule.

Affected parameters: Dynamics

Leap Tone Duration

The first note in an ascending melodic leap is shortened and the second note lengthened if the preceding and succeeding intervals are by step (less than a minor third). In a descending leap the first note is lengthened and the second shortened. The amount in *ms* is only dependent on the interval size of the leap.

Affected parameters: Tempo change

Faster Uphill

Note duration in an ascending melodic line is shortened.

Affected parameters: Tempo change

Repetition Articulation

A micro-pause is inserted between two notes of same pitch, without altering the interonset duration.

Affected parameters: Dynamics, articulation.

Inégalles or swing

Add swing rhythm for Jazz style.

Affected parameters: tempo change

Mixed Intonation

The pitch deviation from equal temperament is made dependent on the interval from previous note and the note's relation to the root of the current chord (melodic intonation and harmonic intonation)

Affected parameters: Pitch.

3.2. Continuous parameters rules

The rule system developed by Sunberg and Friberg is suitable to discrete parameters, that is to say, parameters that are constant during the duration of a whole note. This is enough for some instruments such as the piano or the organ. However, in some instruments such as the strings or the singing voice there are parameters that can change along the note duration. We call them continuous parameters since they can be affected by a continuous modulation. These parameters are extremely relevant in other to obtain a natural and realistic musical synthesis, especially in the singing voice case.

The two continuous controls that have been implemented have an effect on the individual amplitude shape of a note and the pitch contour between two notes. The first one is explained in the dynamic model section and the second one in the pitch model. The vibrato parameters could also be modulated by continuous controls. This has not yet been implemented, since further studies are needed to find out the rules that could control it.

Dynamics model

In order to shape individually the amplitude of each note, we have implemented a rule developed by Manfred Clynes [Clynes, 1987] and called *Predictive Amplitude Shaping* (PAS). This rule designs individual but globally controlled shapes for each note, related to musical structure. There is a global amplitude contour for every note within a musical fragment, which is modified slightly according to the interval to the next note.

PAS does this by using a family of Beta-like functions, which are defined as:

$$x^{p_1}(1-x)^{p_2} \quad for \;\; 0 \le x \le 1$$

normalized to unity maximum amplitude by

$$N = \frac{p_1^{\,p_1}\, p_2^{\,p_2}}{(p_1 + p_2)^{(p_1+p_2)}}$$

giving an amplitude envelope shape as a function of time t, (0 < t < T)

$$A(t) = \frac{G}{N}\left(\frac{t}{T}\right)^{p_1} * \left(1-\frac{t}{T}\right)^{p_2}$$

for a note of duration T and amplitude G.

They are a wide family of curves specified by only two parameters p_1 and p_2, $(p_1, p_2 \ge 0)$ which encompasses the shapes of notes encountered. A basic shape is chosen for a particular voice (p_{i1}, p_{i2}) and placed on each note, and is skewed forward or backward on the note depending on the slope of the pitch time curve from the present note to the next note. The shape of each note is given by

$$p_1 = p_{i1} e^{bs\exp(-aT)}, \qquad p_2 = p_{i2} e^{-bs\exp(-aT)}$$

where

s is the number of semitones to the next note,

T is the duration of the present tone in milliseconds,

a and b are constants relating to duration and pitch respectively

(.00269 mlliseconds^{-1} and .20 semitone^{-1}).

The modified shape implicitly predicts what is to follow, somewhat how in speech the form of a syllable predicts how the next syllable might be formed. This engenders a sense of continuity, each separate expectation being capable of culmination (figure 10).

It turns out that applying this principle to such music remarkably results in phrasing, which is subtly and globally controllable by the composer or interpreter.

Figure 10. Result of applying Predictive Amplitude Shaping

Pitch model

The pitch contour of the singing voice has to be carefully generated in order to obtain a faithful synthesis. So we have designed a mathematical model for reproducing the smooth pitch transitions between notes. This model allows us to control the transition duration and the tuning deviations at the end and the beginning of the notes in accordance with the musical context. In the note-to-note transitions, the synchronization between phonetics and musical rhythm is assured by reaching always the target pitch at the onset of the vowel of each syllable.

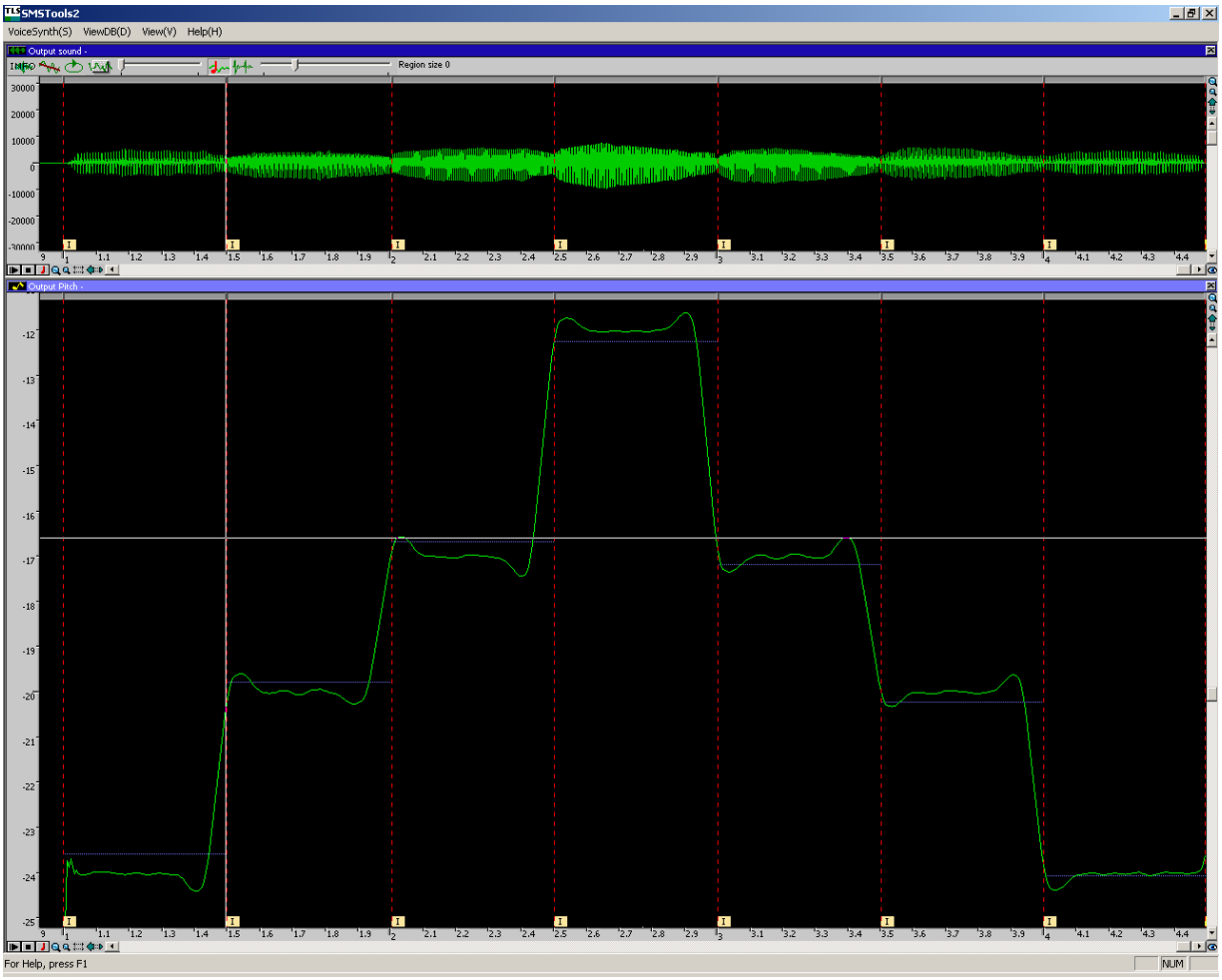

Figure 11. Example of a generated pitch contour.

Remarks on vibrato

The same way as Predictive Amplitude Shaping rule, Manfred Clynes proposes similar rules for controlling the vibrato parameters. Now we have only a first approximation of these rules, not a full implementation. Manfred Clynes principally uses them for string instruments and his results are not directly applicable to the singing voice. We will need to refine the parametric control constants and adapt them to the human voice and to the desired musical styles. Comparison with real measurements on vibrato will be very useful to us [Prame, 1994 &1997].

According to Clynes we need to design individual vibrato on each note. This rule (or set of rules) is named by Manfred Clynes as Organic Vibrato (OV), with equations that describe the vibrato amplitude, frequency, rise time, fall time, and differential placement on each note of vibrato, where this is used. Predictive elements are present in these equations also. The equations combine static and dynamic parameters to design vibrato differently on each note, organically related to musical structure. They are also globally controllable for each voice as needed by the music. Vibrato can give added conviction and communicative power, but only when organically applied (that means related to structure, not as overall 'lacquer'). Different pieces of music require different kinds of vibrato.

The vibrato is influenced by several factors.

Static influences:

- increase in vibrato amplitude as pitch is decreased.
- decrease in vibrato frequency with decreasing pitch

Dynamic influences

- increase in vibrato amplitude with increasing duration of a note
- decrease of vibrato frequency with increasing duration of a note.
- increase in vibrato amplitude if next note will be higher in pitch
- increase in vibrato frequency if next note is higher in pitch
- similar decreases if next note is lower in pitch

The position of the vibrato on a note is changed according to the same beta function variables p1 and p2 that apply to the shaping function for that note. Accordingly the position of the beginning is moved later on the note if p1 is larger, and earlier if it is smaller. The end of the vibrato on the note is also shifted according to p2, so that it comes later if p2 is smaller.

This means that even though the position values are entered in the vibrato controls as base values, every note will in general have a different starting and ending point for the vibrato. So you see it is a combination of a number of factors.

4. ENGLISH GRAPHEME-TO-PHONEME CONVERSION

We have developed an English grapheme-to-phoneme conversion procedure based on a phonetic dictionary search and a set of rules for the linking between words. The transcription procedure takes as its input an orthographic text divided into syllables. Each syllable has to be attached to a musical note. For example:

We've | on- | ly | just | be- | gun

In addition, it has to be allowed to spread a syllable in more than one note, forming a *melisma*. In the below example, the syllable "on" is spread in three musical notes and the syllable "be" in two notes:

We've | on- | - | - | ly | just | be- | - | gun

The procedure consists of uniting the syllables into words, searching them in a phonetic dictionary (See Appendix A) and obtaining their phonetic transcription. After that, some modifications can be done at the linking between words. Of course, these alterations are not to be applied if there is a silence between notes in the musical score. At the moment, we are doing these ones:

- Deletion of consonants in clusters:

band shell /b { n d S e l/ ➔ /b { n S e l/

left field /l e f t f i: l d/ ➔ /l e f f i: l d/

- Devoicing voiced consonants before voiceless sounds

It was clear /I t w Q z k l I r/ ➔ /I t w Q s k l I r/

- Treatment of common words (the)

the only /D i: @U n l i:/ (before vowel or diphthong)

the car /D V c Q r/ (before consonant)

- Assimilation in some common cases:

/d + j/ = /dZ/ *Would you?*/ w U d j u:/ ➔ / w U dZ u:/

/t + j/ = /tS/ *I want you.* / aI w Q n t j u:/ ➔ / aI w Q n tS u:/

The assimilation effect appears when the first consonant of a phonetic articulation becomes more like, or assimilates to, the second one. In general, we think that it is not necessary to take into account the assimilation of consonants at the linking of words. The reason is that this phenomenon should be already registered properly in the articulation database. Anyway, this issue is still under study. In some contexts could be better to do a modification in the phonetic transcription as in the above examples.

Finally, the resulting transcription is divided into syllables:

w i: v | @U | n l i: | dZ V s | b I | g V n |

In the melismatic version, the result will be:

w i: v | @U | @U | @U | n l i: | dZ V s | b I | I | g V n |

With this procedure the English phonetic transcription matter is almost solved. Only two problematic points remain:

1. - What to do when we have different phonetic transcriptions for one written form. For example: read (/r i: d/ or /r e d/?) wind (/ w I n d/ or /w aI n d/?). A complete solution for this problem involves automatic syntactical and semantical analysis. Besides, a dictionary with further information is needed. Even though, a lot of cases will be unsolvable. So the practical thing is let the user choose among the options.

2. - How to divide properly the phonetic transcription into syllables. Without syllabic information in the dictionary, there is no rule to do this division. For example:

apply: *V|plaI* or *Vp|laI* or *Vpl|aI* ? 1st option is the correct one.

upload: *V|pl@Ud* or *Vp|l@Ud* or *Vpl@Ud* ? 2nd option is the correct one.

Since the syllables of a word should be connected in singing as they are in speech, there will be no difference in the synthesis regardless of the syllabic division. The reason is that the phonetic timing is always done with the rule of synchronizing the vowel onset with the note onset. The only problematic case will appear when we have a silence inserted between the syllables of a word and for now we have no solution. This can be considered as a very rare case. However, if we consider the syllabic division as an important issue, the solution is a dictionary with syllabic information, which we don't have available, or maybe a set of approximate rules.

SYNTHESIS MODULE

1. SPECTRAL MODELING SYNTHESIS

The Spectral Modeling Synthesis (SMS) is a synthesis by analysis technique based on modeling the sounds as stable sinusoids (partials) plus noise (residual component). This sinusoidal plus residual model can be seen as a generalization of the *STFT* and the S*inusoidal* representations as we can decide what part of the sound to model as sinusoidal and what part to leave as STFT. [Serra 1996; Serra and Smith 1990].

The input sound *s(t)* is modeled by,

$$s(t) = \sum_{r=1}^{R} A_r(t) \cos\left[\theta_r(t)\right] + e(t) \qquad (1)$$

where $A_r(t)$ and $\theta_r(t)$ are the instantaneous amplitude and phase of the r^{th} sinusoid, respectively, and *e(t)* is time-varying noise component.

This estimation of the sinusoidal component is generally done by first computing the STFT of the sound, then detecting the spectral peaks (and measuring the magnitude, frequency and phase of each one), and organizing them as time-varying sinusoidal tracks. By using the fundamental frequency information in the peak continuation algorithm, we can identify the harmonic partials.

The sinusoidal plus residual model assumes that the sinusoids are stable partials of the sound with a slowly changing amplitude and frequency. With this restriction, we are able to add major constraints to the detection of sinusoids in the spectrum and omit the detection of the phase of each peak. The instantaneous phase that appears in the equation is taken to be the integral of the instantaneous frequency $\omega_r(t)$, and therefore satisfies

$$\theta_r(t) = \int_o^t \omega_r(\tau)\, d\tau \qquad (2)$$

where $\omega(t)$ is the frequency in radians, and r is the sinusoid number. When the sinusoids are used to model only the stable partials of the sound, we refer to this part of the sound as the deterministic component.

The residual component is obtained by first generating the sinusoidal component with additive synthesis, and then subtracting it from the original waveform. This is possible because the instantaneous phases of the original sound are matched and therefore the shape of the time domain waveform preserved. A spectral analysis of this time domain residual is done by first windowing it, window which is independent of the one used to find sinusoids, and thus we are free to choose a different time-frequency compromise. Finally, the FFT is computed.

Within this model we can either leave the residual signal, $e(t)$, to be the difference between the original sound and the sinusoidal component, resulting into an identity system, or we can assume that $e(t)$ is a stochastic signal. In this case, the residual can be described as filtered white noise,

$$e(t) = \int_0^t h(t, \tau)\, u(\tau)\, d\tau \qquad (3)$$

where $u(t)$ is white noise and $h(t,\tau)$ is the response of a time varying filter to an impulse at time t. That is, the residual is modeled by the time-domain convolution of white noise with a time-varying frequency-shaping filter.

The calculation of the residual component can be optimized by subtracting the sinusoids directly in frequency domain (see figure 11). This can be done subtracting the spectrum resulting of the convolution of each sinusoid with the transform of the same window used in the residual spectral analysis.

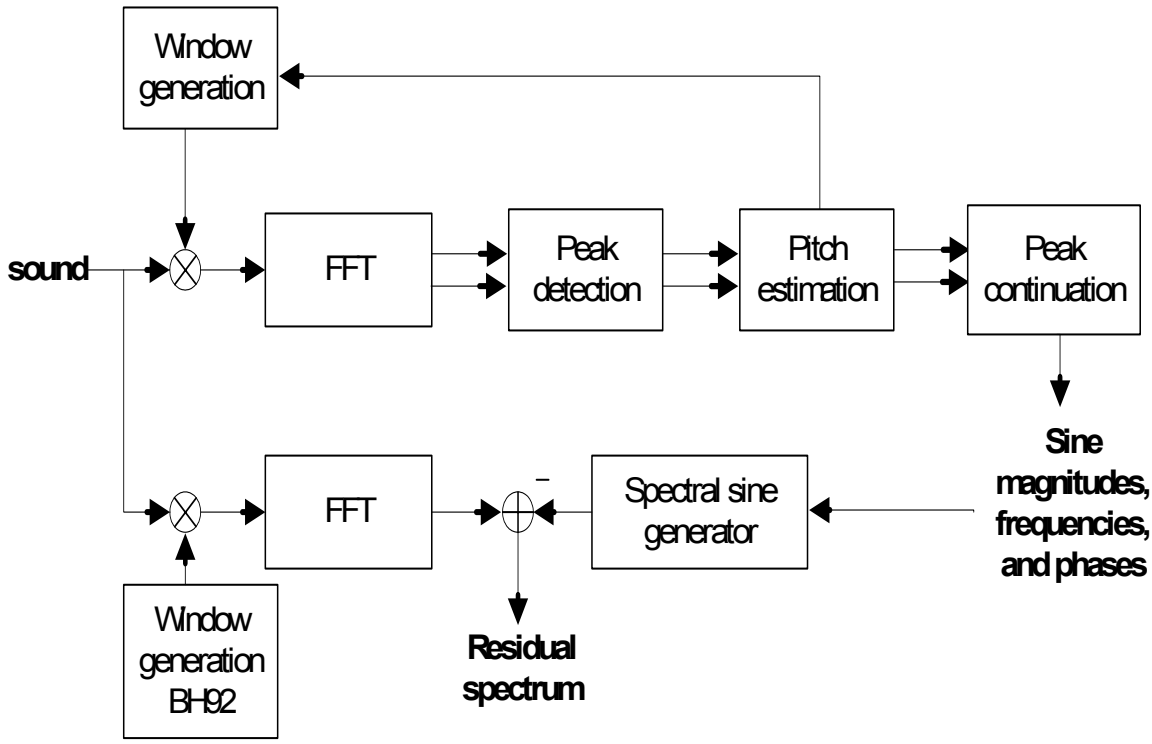

Figure 12. Block diagram of the SMS analysis.

From the output of the analysis techniques presented we can obtain several features of the input sound and its frequency-domain representation. These features are then used in the transformation block in order to modify the characteristics of the sound in a meaningful way. Transformations such as morphing, pitch shifting or spectral shape modification can be performed using this approach. All these transformations can be done in the frequency domain. Afterwards, the output sound can be synthesized.

The sinusoidal component is generated using some type of additive synthesis approach and the residual, if present, is synthesized using a subtractive synthesis approach using an IFFT approach, efficient implementations may be provided. Figure 12 shows a block diagram of the final part of the synthesis process.

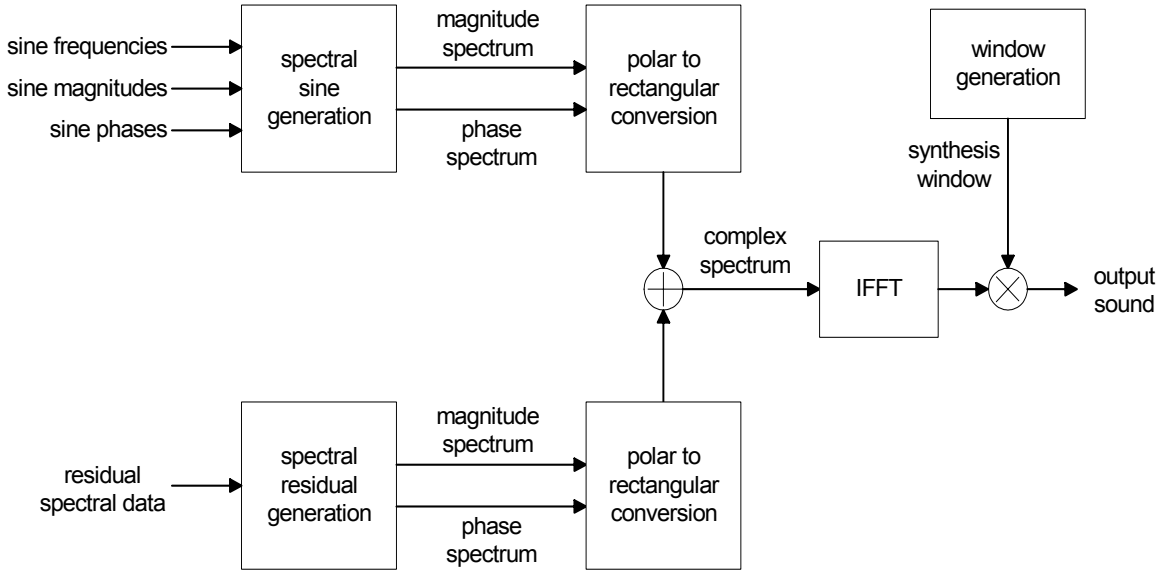

Figure 13. Block diagram of the SMS synthesis.

Several modifications have been done to the basic SMS procedures to adapt them to the requirements of the singing voice application. The major changes include the tuning of all parameters to the particular case of the singing voice. This includes the extraction of the most appropriate higher-level parameters for the case of the singing voice.

2. THE EXCITATION PLUS RESONANCE VOICE MODEL (EPR)

Our singing voice synthesizer is based on an extension of the well-known source/filter approach [Childers, 94], which we call EpR (***E****xcitation* ***p****lus* ***R****esonances)*. The excitation can be either voiced, unvoiced, or a combination of both. Besides, in the case of a voiced phonation we model a harmonic source plus a residual source. In figure 13 we can see a graphic with the three types of excitation generated in our system and the corresponding filters. For the case of a voiced phonation, the filter applied to each excitation tries to mimic the singing voice spectral shape using a frequency domain filtering function that can be decomposed into two filters in cascade: an exponential decay curve plus several resonances. After filtering, the voiced residual excitation needs to be transposed because it is a filtered SMS residual recording and has traces of the original pitch. Otherwise, in the case of an unvoiced phonation, we apply a filter that just changes the tilt curve and the gain of the STFT of an original recording.

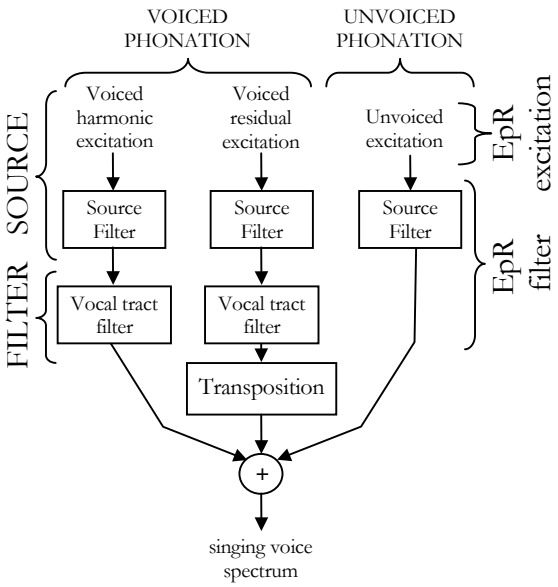

Figure 14. EpR Voice model.

2.1. The EpR excitation

Voiced harmonic excitation

The inputs that control the voiced harmonic excitation generation are the desired pitch and gain envelopes. The resulting excitation signal is obtained by generating a delta train in the time domain thus allowing to achieve period resolution and to use some simple excitation templates. This can be useful to generate jitter or different types of vocal disorders. This delta train can be seen as a glottal source wave previous to a convolution with the differentiated glottal pulse.

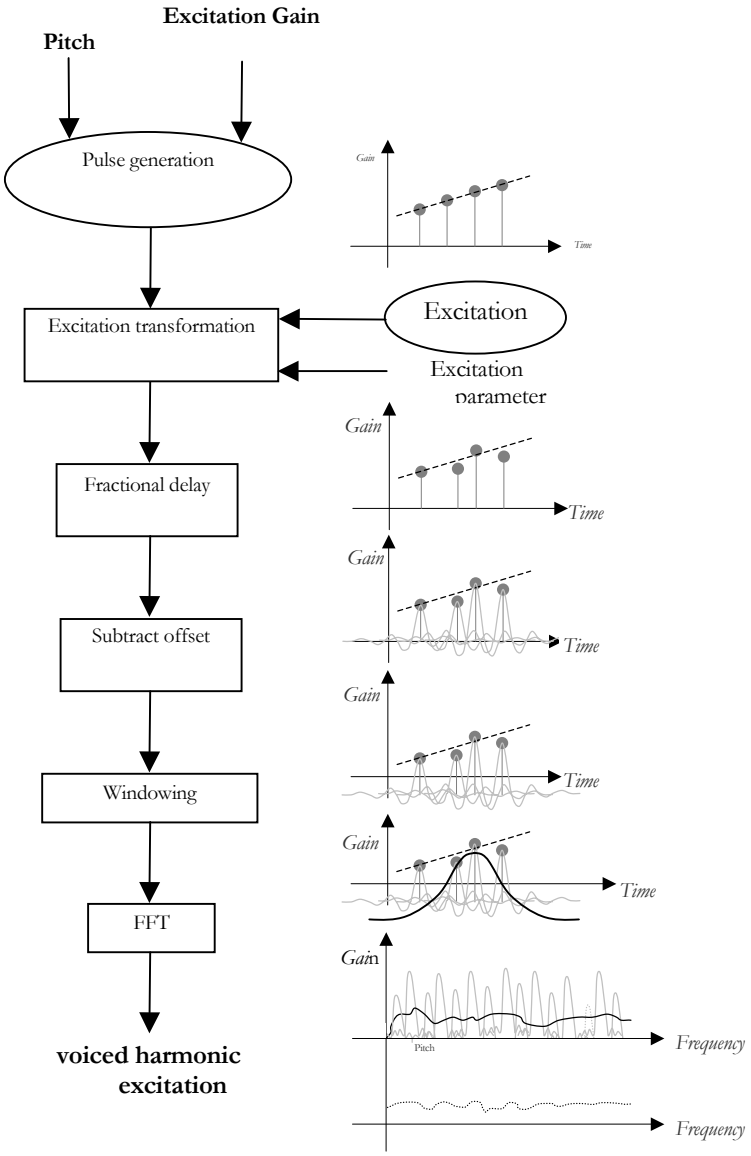

Figure 15. The EpR harmonic voice excitation.

A fractional delay filter is needed to position the excitation deltas between samples, since we have to go beyond the sampling rate resolution. The filter is implemented using a windowed sinc like function situated at each pulse location with the offset subtracted [Smith, Gosset, 1984]. Finally the windowing and the FFT are applied. The result is a spectrum approximately flat that contains the harmonics approximately synchronized in phase. If no excitation template is applied, the spectrum will be perfectly flat and the phase synchronization precise.

Voiced residual excitation

The voiced residual excitation is obtained from the residual of the SMS analysis of a long steady state vowel recorded from a real singer. The SMS residual is inverse-filtered by its short-time average spectral shape envelope to get an approximately flat excitation magnitude spectrum.

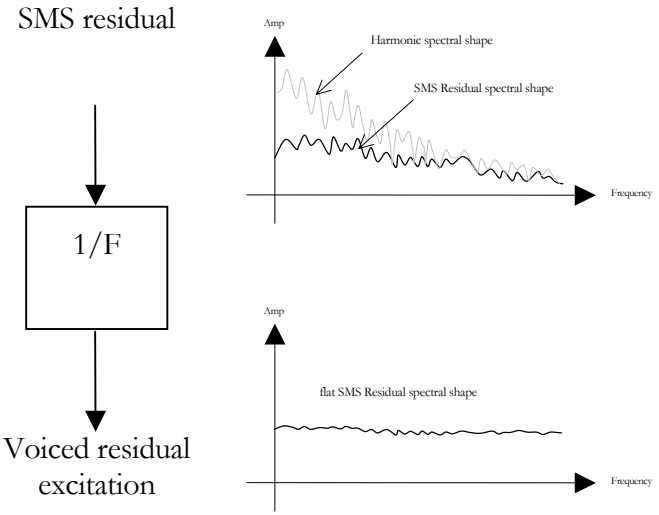

Figure 16. The voiced residual excitation.

Unvoiced excitation

The excitation in the unvoiced parts is left unmodeled, using directly the original recording of a singer's performance.

2.2. The EpR filter

The EpR filter can be decomposed in two cascade filters. The first of them models the differentiated glottal pulse frequency response, and the second the vocal tract (resonance filter).

The EpR source filter

The EpR source is modeled as a frequency domain curve and one source resonance applied to the input frequency domain flat excitation described in the previous section. This source curve is defined by a gain and an exponential decay as follows:

$$Source_{dB} = Gain_{dB} + SlopeDepth_{dB}\left(e^{Slope \cdot f} - 1\right) \qquad (4)$$

This curve is obtained from an approximation to the harmonic spectral shape (*HSS*) determined by the harmonics identified in the SMS analysis

$$HSS(f) = envelope_{i=0..n}\left[f_i, 20\log(a_i)\right] \qquad (5)$$

where i is the index of the harmonic, n is the number of harmonics, f_i and a_i are the frequency and amplitude of the i^{th} harmonic.

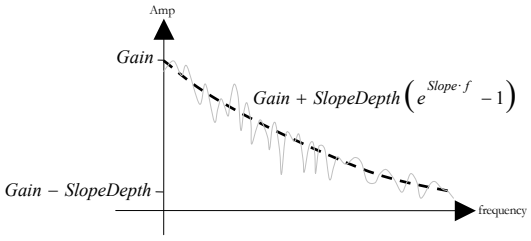

Figure 17. The EpR source curve.

On top of the source curve, we add a second resonance in order to model the low frequency content of the spectrum below the first formant. This resonance affects the synthesis in a different way than the vocal tract resonances, as will be explained later.

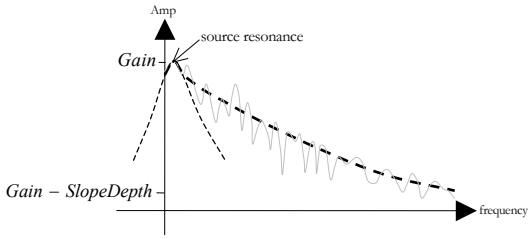

Figure 18. The EpR source resonance.

The source resonance is modeled as a symmetric second order filter (based on the Klatt formant synthesizer [Klatt, 1980]) with center frequency *F*, bandwidth *Bw* and linear amplitude *Amp*. The transfer function of the resonance *R(f)* can be expressed as follows

$$H(z) = \frac{A}{1 - Bz^{-1} - Cz^{-2}}$$

$$R(f) = Amp \frac{H\left(e^{j2\pi\left(0.5+\frac{f-F}{fs}\right)}\right)}{H\left(e^{j\pi}\right)} \qquad (6)$$

where

$$fs = \text{Sampling rate}$$

$$C = -e^{-\frac{2\pi Bw}{fs}}$$

$$B = 2\cos(\pi)e^{-\frac{\pi Bw}{fs}}$$

$$A = 1 - B - C$$

The amplitude parameter (*Amp*) is relative to the source curve (a value of 1 means the resonance maximum is just over the source curve).

The EpR vocal tract filter

The vocal tract is modeled by a vector of resonances plus a differential spectral shape envelope. It can be understood as an approximation to the vocal tract filter. These filter resonances are modeled in the same way as the source resonance (see equation 6), where the lower frequency resonances are somewhat equivalent to the vocal tract formants.

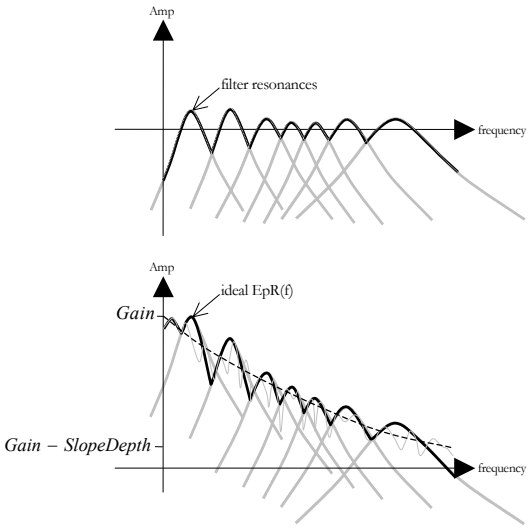

Figure 19. The EpR filter resonances.

The EpR filters for voiced harmonic and residual excitations are basically the same, but just differ in the gain and slope depth parameters. This approximation has been obtained after comparing the harmonic and residual spectral shape of several SMS analysis of singer recordings. Figure 19 shows these differences.

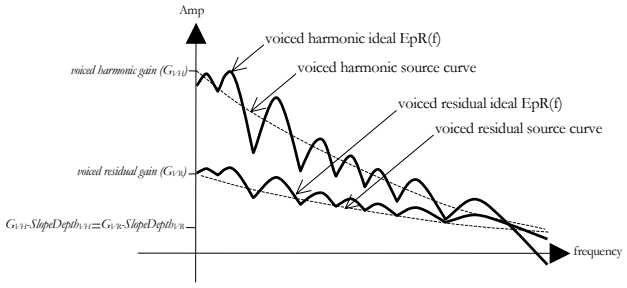

Figure 20. Differences between harmonic and residual EpR filters.

The differential spectral shape envelope actually stores the differences (in dB) between the ideal EpR model (*iEpR*) and the real harmonic spectral shape (*HSS*) of a singer's performance. We calculate it as a 30 Hz equidistant step envelope.

$$DSS(f) = envelope_{i=0..}\left[30i, HSS_{dB}(30i) - iEpR_{dB}(30i)\right] \quad (7)$$

The EpR phase alignment

The phase alignment of the harmonics at the beginning of each period is obtained from the EpR spectral phase envelope. A time shift is applied just before the synthesis, in order to get the actual phase envelope at the synthesis time (usually it will not match the beginning of the period). This phase alignment is then added to the voiced harmonic excitation spectrum phase envelope. The EpR spectral phase model states that each vocal tract resonance produces a linear shift of π on the flat phase envelope with a bandwidth depending on the estimated resonance bandwidth. This phase model is especially important for the intelligibility and in order to get more natural low pitch male voices.

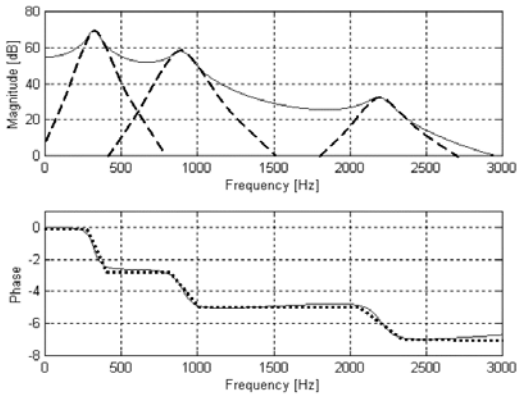

Figure 21. Phase alignment.

The EpR filter implementation

The EpR filter is implemented in the frequency domain. The input is the spectrum that results out from the voiced harmonic excitation or from the voiced residual excitation. Both inputs are supposed to be approximately flat spectrums, so we just need to add the EpR resonances, the source curve and the differential spectral shape to the amplitude spectrum. In the case of the voiced harmonic excitation we also need to add the EpR phase alignment to the phase spectrum.

For each frequency bin we have to compute the value of the EpR filter. This implies a considerable computational cost, because we have to calculate the value of all the resonances. However, we can optimize this process by assuming that the value of the sum of all the resonances is equal to the maximum amplitude (dB) of all the filter and excitation resonances (over the source curve). Then

we can even do better by only using the two neighbors resonances for each frequency bin. This is not a low-quality approximation of the original method because the differential spectral shape envelope takes care of all the differences between the model and the real spectrum.

If we want to avoid the time domain voiced excitation, especially because of the computational cost of the fractional delay and the FFT, we can change it to be directly generated in the frequency domain. From the pitch and gain input we can generate a train of deltas in frequency domain (sinusoids) that will be convolved with the transform of the synthesis window and then synthesized with the standard frame based SMS synthesis, using the IFFT and overlap-add method. However the voice quality may suffer some degradation due to the fact that the sinusoids are assumed to have constant amplitude and frequency along the frame duration.

The EpR filter transformation

We can transform the EpR by changing its parameters:

- excitation gain, slope and slope depth
- frequency, amplitude and bandwidth of the resonances

However, we have to take into account that the differential spectral shape is related to the resonances position. Therefore, if we change the frequency of the resonances we should stretch or compress the differential spectral shape envelope according to the resonances frequency change (using the resonances center frequency as anchor points).

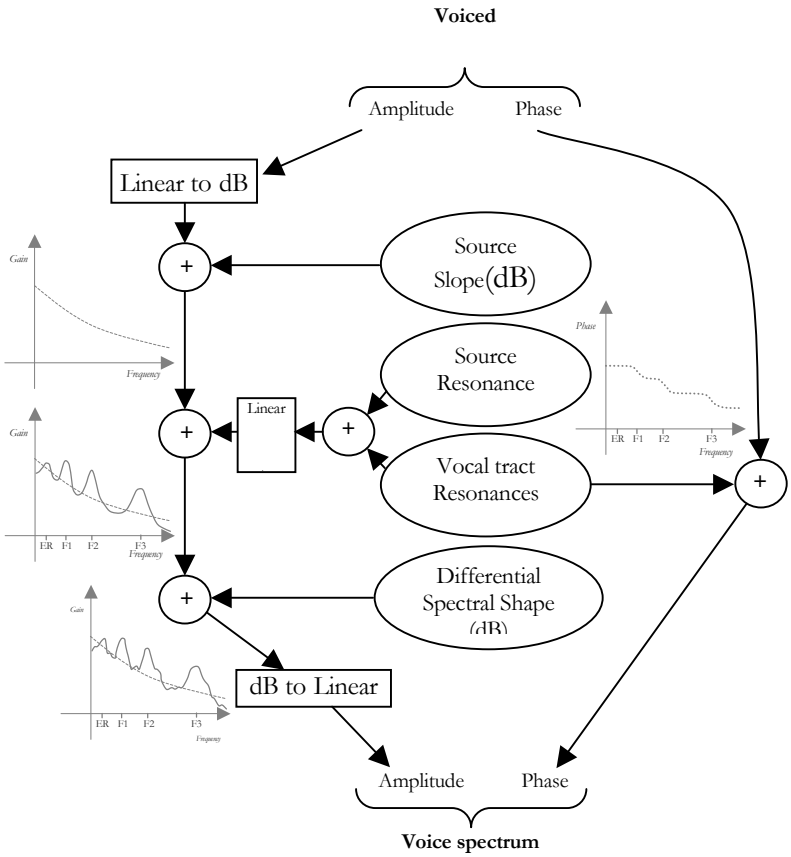

Figure 22. Frequency domain implementation of the EpR model.

3. SYNTHESIS SCORE AND PHONEMES TIMING

DaisyScore is the internal data format of the synthesis module that is directly readable by the synthesis loop. It has been designed to accept a streaming data flow as its input and to get a synthesis in real time. In DasisyScore, information is divided in different tracks and is ordered temporally. The synthesis loop has to read data in every track according to the time that is being synthesized.

The first track is called phonetic track and it includes all the information related to the phonetic units needed in the synthesis. It contains the phonetic segments and its duration values. Phonetic segments can be stationary if we don't change the phoneme or articulations if we go from one phoneme to another. Here we have an example of the data in the phonetic track.

```
NoteOn -      NoteOff      Duration     FromPhU -    ToPhU
=============================================================================
0.000000      -      0.950000     (0.950000)   Sil
0.950000      -      1.159799     (0.209799)   U
1.159799      -      1.165604     (0.005805)   Sil
1.165604      -      1.226780     (0.061176)   m
1.226780      -      1.279025     (0.052245)   m      -      {
1.279025      -      1.435780     (0.156755)   {
1.435780      -      1.522855     (0.087075)   {      -      s
1.522855      -      1.628411     (0.105556)   s
1.628411      -      1.634216     (0.005805)   Sil
1.634216      -      1.716950     (0.082734)   r
1.716950      -      1.809830     (0.092880)   r      -      I
1.809830      -      1.868142     (0.058312)   I
1.868142      -      1.931997     (0.063855)   I      -      m
1.931997      -      1.982585     (0.050588)   m
1.982585      -      2.023220     (0.040635)   m      -      e
2.023220      -      2.282325     (0.259105)   e
2.282325      -      2.334570     (0.052245)   e      -      m
2.334570      -      2.423981     (0.089412)   m
2.423981      -      2.464616     (0.040635)   m      -      b
2.464616      -      2.482585     (0.017969)   b
2.482585      -      2.523220     (0.040635)   b      -      @
2.523220      -      2.697256     (0.174036)   @
2.697256      -      2.703061     (0.005805)   Sil
2.703061      -      2.715170     (0.012109)   d
2.715170      -      2.779025     (0.063855)   d      -      I
2.779025      -      4.426485     (1.647460)   I
4.426485      -      4.554195     (0.127710)   I      -      s
4.554195      -      4.744195     (0.190000)   s
4.744195      -      4.750000     (0.005805)   Sil
Total is 4.750000
```

The next track is the note track and it contains information about the current state of the note. We defined four labels for the note state: 'Attack', 'Release', 'Note to note' and 'None'. In case that phonetic track has an articulation between current note and next note, then we insert a 'NoteToNote' block to indicate the change of the pitch between the two notes. The duration of the pitch change is the 'NoteToNote' duration. In case that there is no phonetic articulation between current note and next, then we insert a 'Release' at the end of the current note and an "Attack" one at the beginning of next note. Otherwise, we label the note segment with "None". Here is an example of note track (related with the previous phonetic track):

```
NoteOn       -      NoteOff      Duration     NoteType
================================================================
0.000000     -      0.950000     (0.950000)   NONE
0.950000     -      0.980000     (0.030000)   ATTACK
0.980000     -      1.124799     (0.144799)   NONE
1.124799     -      1.159799     (0.035000)   RELEASE
1.159799     -      1.165604     (0.005805)   NONE
1.165604     -      1.195604     (0.030000)   ATTACK
1.195604     -      1.250000     (0.054396)   NONE
1.250000     -      1.576536     (0.326536)   NONE
1.576536     -      1.628411     (0.051875)   RELEASE
1.628411     -      1.634216     (0.005805)   NONE
1.634216     -      1.664216     (0.030000)   ATTACK
1.664216     -      1.775000     (0.110784)   NONE
1.775000     -      1.900000     (0.125000)   NONE
1.900000     -      2.000000     (0.100000)   NOTE_TO_NOTE
2.000000     -      2.400000     (0.400000)   NONE
2.400000     -      2.500000     (0.100000)   NOTE_TO_NOTE
2.500000     -      2.666006     (0.166006)   NONE
2.666006     -      2.697256     (0.031250)   RELEASE
2.697256     -      2.703061     (0.005805)   NONE
2.703061     -      2.733061     (0.030000)   ATTACK
2.733061     -      2.750000     (0.016939)   NONE
2.750000     -      4.581695     (1.831695)   NONE
4.581695     -      4.744195     (0.162500)   RELEASE
4.744195     -      4.750000     (0.005805)   NONE
Total is 4.750000
```

In addition to these two tracks, DaisyScore has several tracks that are envelopes with an indeterminate number of points. These envelopes are useful to model the continuous change of some parameters. Until now, the parameters that have been implemented as an envelope are Pitch, Dynamics, Vibrato depth and Vibrato rate. Vibrato tremolo, Opening and Hoarseness are not yet implemented.

Envelopes example:

```
- Envelope: Pitch
0.000000     -      -1000.000000
0.950000     -      -1000.000000
1.153894     -      -1000.000000
1.153994     -      -900.000000
1.651531     -      -900.000000
1.651631     -      -1000.000000
1.999900     -      -1000.000000
2.000000     -      -1200.000000
2.499900     -      -1200.000000
2.500000     -      -1300.000000
2.702961     -      -1300.000000
2.703061     -      -1200.000000
4.749900     -      -1200.000000
1000.000000  -      0.000000


- Envelope: Dynamics
0.000000     -      0.000000
0.950000     -      0.400000
1.153794     -      0.400000
1.153994     -      0.450000
1.651431     -      0.450000
1.651631     -      0.400000
1.999800     -      0.400000
2.000000     -      0.450000
2.499800     -      0.450000
2.500000     -      0.200000
2.702861     -      0.200000
2.703061     -      0.400000
3.726530     -      0.300000
4.749795     -      0.200000
1000.000000  -      0.000000


- Envelope: Opening
0.000000     -      0.000000
0.950000     -      0.500000
1.153973     -      0.500000
1.153994     -      0.500000
1.651581     -      0.500000
1.651631     -      0.500000
1.999965     -      0.500000
2.000000     -      0.500000
2.499950     -      0.500000
2.500000     -      0.500000
2.703040     -      0.500000
2.703061     -      0.500000
4.749795     -      0.500000
1000.000000  -      0.000000

- Envelope: Hoarseness
0.000000     -      0.000000
0.950000     -      0.000000
1.153973     -      0.000000
1.153994     -      0.000000
1.651581     -      0.000000
1.651631     -      0.000000
1.999965     -      0.000000
2.000000     -      0.000000
2.499950     -      0.000000
2.500000     -      0.000000
2.703040     -      0.000000
2.703061     -      0.000000
4.749795     -      0.000000
1000.000000  -      0.000000
```

```
- Envelope: VibDepth
0.000000     -      0.000000
0.950000     -      0.000000
1.153994     -      0.000000
1.651631     -      0.000000
2.000000     -      0.000000
2.500000     -      0.000000
2.703061     -      0.000000
3.521836     -      0.250000
4.238265     -      1.000000
4.729531     -      0.000000
1000.000000  -      0.000000


- Envelope: VibRate
0.000000     -      0.000000
0.950000     -      0.000000
1.153994     -      0.000000
1.651631     -      0.000000
2.000000     -      0.000000
2.500000     -      0.000000
2.703061     -      5.000000
4.197326     -      5.100000
4.340612     -      5.200000
4.463429     -      5.400000
4.612855     -      5.800000
1000.000000  -      0.000000
```

DaisyScore Construction

In the construction of DaisyScore, the most important points to have into account are the duration of stationary and articulation parts of phonemes and the timing of pitch change.

First of all, we have to decide which is the main phoneme in a syllable, normally a vowel, whose duration we can vary according to note duration. After that, the length of articulations between consonants and vowels is extracted from the phonetic database. Some consonants have also a stationary part with a duration value by default. However, if these fixed values are used everywhere, some problems appear regarding the naturalness of the synthesis, especially in short notes. In some phonemes groups, the duration of the stationary part of a consonant may depend on the note duration. Different articulation lengths may be needed also for different note duration values. In that case, we can ask for a duration value different from that in the phonetic database. Then the synthesis engine will have to stretch or to shrink the articulation length.

In regard to the timing of pitch change, in general, the new pitch in a syllable is approached during the consonant and the target pitch is reached at the vowel onset. To avoid rhythm errors, it is necessary to subtract duration of all consonants from the duration of the preceding vowel.

In order to construct correctly the phonetic track, we have to follow these steps:

1. Write all the phonetic segments in pairs of phonemes (different phonemes in an articulation, the same phonemes in a stationary segment).

2. Obtain duration values of the articulation from the phonetic database and estimate the duration of stationary parts of consonants if necessary.

3. Find the vowel of the syllable. If there is an articulation from another phoneme to this vowel, find the exact point at which the vowel starts during the articulation. This point has to coincide with the note onset.

4. Subtract duration of all consonants of the next syllable from the current note duration. We have to be careful if there is silence between notes.

5. Calculate the vowel duration by subtracting all the articulations from the note duration.

6. If we obtain a negative duration for the vowel, we have to reduce progressively the stationary lengths and the articulation lengths until we get a positive value for the vowel.

Lyrics: You must remember

Phonetic transcription (SAMPA): j U - m V s t - r I - m e m - b @ r

Phonetic track:

Sil-j	j-	u:	u:-u:	u:-m	m–m	m-	V	V-V	V-s	s-s	s-t	t-Sil	Sil-Sil	Sil-r	r-r	r-	I

Note Track:

Attack	None	NoteToNote	None	Release	None	Attack	None

Figure 23. Example of phonetic and note track.

Chapter Four

SINGER DATABASE

The creation of the database is one of the critical steps in most speech and singing voice synthesizers. There are no singing voice databases available, thus we recorded our own samples by first defining a set of musical exercises to be sung by the singer (i.e. different type of attacks, releases, note to note transitions, etc). Needless to say, the voice had to be recorded in a dry and noiseless environment to get the best possible SMS analysis.

In terms of what information is needed to be stored in the database, we can either attempt to record as many samples as possible (all possible pitches, attacks…) or start from fewer samples and obtain the rest through transformations from the recorded ones. This second approach would give more flexibility to the system at the possible expense of sound quality. Our approach has been this second one.

1. DATABASE SECTIONS

The database is organized into two parts; timbre and voice DB.

Timbre DB

The timbre database stores the voice model (EpR) for each of the voiced phonemes. For each of these, we store several samples at different pitches and at different dynamics. When a phoneme with intermediate pitch or dynamics is required, the EpR parameters of the neighboring samples are interpolated to synthesize the phoneme at the desired pitch.

Voice DB

The voice database includes the time varying characteristics of the voice. It is divided into several categories: steady states, phonetic articulations, note attacks, note-to-note articulations, note releases and vibratos. It is also possible to have different templates for the different pitches of the same type of sound.

Steady states

There are steady states stored for each of the phonemes. They model the behavior of the stationary part of a note. To do so, we store the time-varying evolution of the EpR parameters (if the phoneme is voiced) and the SMS residual along the steady state.

Phonetic articulations

All the articulations are segmented in two regions, the end of one sound and the beginning of the next. If the regions are different in terms of voiceness, each region is analyzed with different specific parameters. Otherwise, if both regions are voiced or unvoiced the segmentation is used in synthesis to control the onset of the articulation. The EpR parameters are estimated only in the voiced part regions.

Vibratos

They characterize the vibrato characteristics by storing the behavior of the voice excitation parameters. In particular, the fundamental frequency evolution, gain and source curve changes. Each of them is segmented into attack, body and release. There can be different template labels according to some musical or expression classification.

Note attacks, note transitions and note releases

They model the excitation behavior along their duration, using the Pitch Model. They are phoneme independent since they do not model the EpR changes. They can be organized into different musical evocative classification.

2. RECORDING SCRIPTS

In this section, we describe the recordings from real singers that we have made and also the choice of phonetic articulations to be recorded. This latter point is especially delicate because of the huge complexity of English phonetics. During the project development, we made some recording sessions with two professional singers, a male and a female. In the next paragraphs, a description of the recording scripts is given. This is the final version after several refinements and optimisations. A three-hour studio session is needed for a singer to make a complete recording of all the required features for constructing a database.

Attack and release

Different types of note attacks have been recorded: normal, sharp, soft, and sexy. Each of those was recorded at three different pitches. Besides, each of these attacks was recorded at different note duration: half note and quarter note. All the different types of attacks were recorded using the /{/ phoneme sound. About notes releases, we recorded two different types: sharp, and soft. Apart from that, the different contexts chosen to record the releases were the same as the ones used in the attacks: three different pitches, two notes with different duration values, and one phoneme.

Dynamics

We recorded three different degrees of dynamics: soft, normal, and loud. For each of these categories we recorded all vowel sounds.

Note to note transitions

The note-to-note transition recordings are based in a scale in which the root is repeated at each interval. We recorded the following different types of transitions between notes: staccato, legato, glissando, and portamento. Each one of these transition types was recorded for the half note case. For the quarter note length case we only recorded staccato and legato transitions. All these note-to-note transitions were recorded with the same phonetic unit /{/.

Openings

Although that was not clearly specified in the recording scripts, we think it is very important for the EpR model to have different levels of mouth openings in the database. We decided to record this openings for every vowel sounds. The way we did so was asking the singers to attack the phoneme with the mouth as open as they could and try to relax the face muscles while sustaining the note so that the mouth close gradually and little by little along the note.

Singing Style

We recorded different ways of performing the same song with both singers. In the case of Donna the song she chose was "Breaking up is hard to do", written by Neil Sedaka. We got six different singing styles with her: normal, flat, happy, sad, jazzy, and drunk. She did really well. We could not obtain such good results with Alex. He recorded four different performances of "Midnight Hour" written by Wilson Picket: normal, flat, happy, and sad.

Vibrato

Due to the fact that neither Donna nor Alex were school trained singers we did not record as many different types of vibrato as we wanted. Both singers did a recording of their natural vibrato at three different pitch and for each of all the vowel sounds. Donna also recorded a classic vibrato and what we called increasing vibrato. These were recorded at three pitches also.

Timbres

In order to obtain samples of all the needed vowels and diphthongs, we asked the singers to sing a set of words containing these phonemes in seven different pitches. See Appendix C.

Steady States

We asked the singers to sing the same words that in the timbres section, but only in five different pitches and with a longer duration. See Appendix C.

Phonetic Articulations

We have studied which combinations of English phonemes are possible to articulate (See Appendix A). In order to make Alex and Donna sing these phonetic articulations, we had to do a list of words that include the articulations we were looking for. To do so, we used the CMU pronouncing dictionary. The Carnegie Mellon University Pronouncing Dictionary is a machine pronunciation dictionary for North American English that contains over 100 thousand words and their transcriptions.

In order to choose the most important and useful phonetic articulations to be included in our English phonetic database, we have ordered all the possible articulations according to its use frequency. A statistical analysis was made of 76000 English songs with around three million words. This large lyrics collection was obtained from the website http://www.getlyrics.com/. We have found that it is much more significant to make our statistical studies on use frequency from lyrics of real songs rather than directly from the CMU phonetic dictionary, as we have made until now. We find out very different results. For example, the articulation /D {/ appears in only two words in the dictionary (that, than) but it is the 44th more used articulation in the songs.

The analysis procedure has been to convert the songs in a text file format and then to do an automatic phonetic transcription with the CMU dictionary (without taking into account liaisons between words). After that, we have counted and ordered the articulations according to its use frequency. At each articulation of the ordered list, we calculate the covered percentage of all possible articulations by using this articulation and all the previous ones. In Appendix B there is a sample of the results we obtained.

With this information, now we know how many articulations we need to cover a fixed percentage of all the possible articulations. See next table.

Number of articulations	Covered percentage
71	50 %
308	90 %
395	95%
573	99%
785	99.9%
1129	100%

After that, we had to select over one thousand words that contained these diphonemes and we recorded them from the singers in two different pitches. See Appendix C.

CONCLUSIONS AND FUTURE WORK

With the purpose of demonstrating the potential of our system, two small databases with a male singer and a female singer have been created. With the female voice, we have synthesized fragments of two different songs ("We've only just begun", by The Carpenters, and "Natural woman", by Carole King). With the male voice, we have synthesized the same song by The Carpenters and some choruses for accompanying the female songs.

The system evaluation was made by a group of independent advisors disconnected from the project. Taking Vocalwriter synthesizer as a comparison point for the English synthesis, our demo songs were considered more intelligible and more natural. In contrast, our syntheses showed a lack of timbre uniformity. In conclusion, although the synthesis obtained was not comparable with a real singer performance, considerable improvement has been made.

Nevertheless, the system presents important drawbacks. Unnatural artifacts appear in the synthesis, especially in the voiced consonants phonemes. These artifacts emerge from the fact that our synthesis engine is built on top of a sinusoidal plus residual decomposition. For these phonemes, it is difficult to determine what should be included in the sinusoidal component and what not. The low register timbres of a male voice suffer from unnaturalness. This problem may come from the simplicity of the EpR phase model. Sharp attacks, specially the ones belonging to plosive phonemes are smeared. This is due to the fact the sinusoidal model cannot deal with unstable partials in an accurate manner. Some solutions to this problem are proposed in [Fitz, Haken, Christensen, 2000; Verma, Meng, 2000].

The creation of a complete voice database is for now a laborious work. More automated ways have to be developed to facilitate and speed up the database creation process. Obviously the larger the singer database the better synthesis quality we will be able to achieve. Because of some significant coarticulation effects in English pronunciation, it will be necessary to modify the database format in order to handle not only diphonemes but also multiphonemes, for example the triphonemes made up of a plosive, a liquid and a vowel (/p r {/, /p r e/, /p l i:/, etc.). This will increase the number of phonetic articulations from one thousand up to two thousand, a common figure in text-to-speech databases.

Concerning the expressive musical controls of the system, good results have been obtained by applying flexible and powerful mathematical models to the pitch and amplitude contours of the

notes. Besides, a rule-based system for expressive enhancement has been implemented and tested. Further research is needed on automatic control of some voice characteristics as vibrato or timbre variations. The next goal will be to provide the users with a full set of expressive controls in different levels from the global level (sing happy, sing sad...) to the note level (sharp attack, smooth attack, legato, etc.) covering some different musical styles. At this point, the main difficulty will be to define properly interfaces understandable and useful for the users.

Regarding the practical user applications that the system can have, the first step will be to develop a software product for amateurs. Most likely this program will be distributed freely on the net in order to show the potential of the technology and make it known in the music world. When the quality of the synthesis is improved as we expect in the near future, a professional product for composers and music producers will be developed. This product could be either hardware equipment or a software program.

Certainly there is room for improvement in every step of the system. Even so, assuming naturalness as the essential feature for evaluating the quality of a singing synthesizer, results are promising and prove the suitability of our approach.

Appendix A

ENGLISH PHONETIC TRANSCRIPTION

SAMPA (Speech Assessment Methods Phonetic Alphabet) is a phonetic alphabet designed to be machine-readable.

English phonemes classification in SAMPA notation

	Voiced	**unvoiced**
vowels	I e Q { V U @	-
diphthongs	i: eI aI OI u: @U 3: A: O: I@ e@ U@	-
plosives	b d g	p t k
nasals	m n N	-
liquids	l r	-
fricatives	v D z Z	f T s S h
affricates	dZ	tS
semivowels	w j	-

English phonemes with examples and its transcription

Plosives

Phoneme	Example	Phonetic transcription
p	pin	pIn
b	bin	bIn
t	tin	tIn
d	din	dIn
k	kin	kIn
g	give	gIv

Affricates

Phoneme	Example	Phonetic transcription
tS	chin	tSIn
dZ	gin	dZIn

Fricatives

Phoneme	Example	Phonetic transcription
f	fin	fIn
v	vim	vIm
T	thin	TIn
D	this	DIs
k	sin	sIn
z	zing	zIN
S	shin	SIn
Z	measure	meZ@
h	hit	hIt

Nasals

Phoneme	Example	Phonetic transcription
m	mock	mQk
n	knock	nQk
N	thing	TIN

Liquids

Phoneme	Example	Phonetic transcription
r	wrong	rQN
l	long	lQN

Semivowels

Phoneme	Example	Phonetic transcription
w	wasp	wQsp
j	yacht	jQt

Vowels

Phoneme	Example	Phonetic transcription
I	pit	pIt
e	pet	pet
{	pat	p{t
Q	pot	pQt
V	cut	kVt
U	put	pUt
@r	bird	b@rd

Dipthongs

Phoneme	Example	Phonetic transcription
i:	ease	i:z
eI	raise	reIz
aI	rise	raIz
OI	noise	nOIz
u:	lose	lu:z
@U	nose	n@Uz
aU	rouse	raUz
3:	furs	f3:z
A:	stars	stA:z
O:	cause	kO:z
I@	fears	fI@z
e@	stairs	ste@z
U@	cures	kjU@z

The Carnegie Mellon University Pronouncing Dictionary is a machine pronunciation dictionary for North American English that contains over 100 thousand words and their transcriptions. The current phonemes set they use has 39 phonemes. These are:

Phoneme	Word Example	Phonetic Transcription
AA	odd	AA D
AE	at	AE T
AH	hut	HH AH T
AO	ought	AO T
AW	cow	K AW
AY	hide	HH AY D
B	be	B IY
CH	cheese	CH IY Z
D	dee	D IY
DH	thee	DH IY
EH	Ed	EH D
ER	hurt	HH ER T
EY	ate	EY T
F	fee	F IY
G	green	G R IY N
HH	he	HH IY
IH	it	IH T
IY	eat	IY T
JH	gee	JH IY
K	key	K IY
L	lee	L IY
M	me	M IY
N	knee	N IY
NG	ping	P IH NG
OW	oat	OW T
OY	toy	T OY
P	pee	P IY
R	read	R IY D
S	sea	S IY
SH	she	SH IY
T	tea	T IY
TH	heta	TH EY T AH
UH	hood	HH UH D
UW	two	T UW
V	vee	V IY
W	we	W IY
Y	yield	Y IY L D
Z	zee	Z IY
ZH	seizure	S IY ZH ER

Correspondences between the CMU and the SAMPA phonetic symbols.

CMU	**SAMPA**
AA	Q
AE	{
AH	V
AO	O:
AW	aU
AY	aI
B	b
CH	tS
D	d
DH	D
EH	e
ER	@r
EY	eI
F	f
G	g
HH	h
IH	I
IY	i:
JH	dZ
K	k
L	l
M	m
N	n
NG	N
OW	@U
OY	OI
P	p
R	r
S	s
SH	S
T	t
TH	T
UH	U
UW	u:
V	v
W	w
Y	j
Z	z
ZH	Z

Appendix B

PHONETIC ARTICULATIONS STATISTIC

	covered percentage	articulation	example
1	2,1286	**n d**	understand
2	4,1929	**D V**	the
3	6,2569	**{ n**	can't
4	8,0601	**V n**	merchant
5	9,7416	**j u:**	music
6	11,2517	**O: r**	store
7	12,7260	**I N**	adding
8	14,1070	**s t**	store
9	15,3792	**t u:**	to
10	16,6340	**I n**	in
11	17,7557	**n t**	merchant
12	18,7699	**I t**	it's
13	19,7512	**V v**	of
14	20,5883	**e n**	pennies
15	21,4126	**Q n**	gonna
16	22,1952	**V m**	some
17	22,9511	**m aI**	my
18	23,6960	**m i:**	me
19	24,4321	**e r**	airplanes
20	25,1362	**V s**	youngest
21	25,8339	**Q r**	darlings
22	26,5260	**V l**	natalie
23	27,2130	**V t**	but
24	27,8667	**{ t**	natalie
25	28,4757	**r i:**	three
26	29,0766	**l aI**	lightning
27	29,6727	**r V**	astronauts
28	30,2672	**t s**	astronauts
29	30,8598	**O: l**	all
30	31,4490	**w I**	with
31	32,0183	**l V**	lucky
32	32,5850	**n @U**	know
33	33,1417	**I l**	building
34	33,6979	**@U n**	don't
35	34,2533	**I z**	is
36	34,8044	**t V**	natalie
37	35,3360	**aI m**	i'm
38	35,8612	**l i:**	natalie
39	36,3779	**s V**	some
40	36,8943	**k V**	nickels
41	37,4022	**w V**	one
42	37,9095	**w e**	well
43	38,4045	**b i:**	ruby
44	38,8795	**D {**	than
45	39,3526	**f O:**	afford
46	39,8201	**I s**	this
47	40,2808	**e l**	yourselves
48	40,7414	**aI t**	night
49	41,1973	**w i:**	weeks
50	41,6413	**v @r**	fever
51	42,0814	**w Q**	was
52	42,5198	**n V**	gonna
53	42,9578	**e t**	better
54	43,3952	**j O:**	your
55	43,8222	**t r**	astronauts
56	44,2422	**Q t**	bought
57	44,6554	**e v**	every
58	45,0622	**aU n**	down
59	45,4668	**r I**	bring
60	45,8708	**h {**	has
61	46,2739	**k {**	can't
62	46,6753	**D e**	them
63	47,0755	**w eI**	away
64	47,4738	**b V**	buck
65	47,8557	**l d**	cold
66	48,2371	**l I**	slip
67	48,6150	**b I**	been
68	48,9923	**aI n**	wind
69	49,3623	**eI k**	take
70	49,7315	**r aI**	try
71	50,0876	**s i:**	see
72	50,4418	**t @r**	better
73	50,7854	**h I**	his
74	51,1234	**m V**	remedies
75	51,4571	**@U l**	hole
76	51,7841	**I r**	hero
77	52,1100	**g Q**	gonna
78	52,4357	**s @U**	so
79	52,7588	**T I**	anything
80	53,0816	**d i:**	remedies

81	53,4036	**d I**	adding
82	53,7215	**n i:**	pennies
83	54,0347	**t I**	still
84	54,3478	**eI n**	airplanes
85	54,6596	**i: z**	these
86	54,9643	**n s**	once
87	55,2642	**r eI**	brace
88	55,5608	**U d**	should
89	55,8559	**d @U**	don't
90	56,1489	**f r**	from
91	56,4412	**dZ V**	jungle
92	56,7303	**aU t**	out
93	57,0188	**g e**	get
94	57,3061	**d V**	dust
95	57,5865	**I k**	music
96	57,8669	**s e**	yourselves
97	58,1400	**aI d**	tried
98	58,4130	**e d**	head
99	58,6802	**k I**	kids
100	58,9446	**i: l**	feelings
101	59,2071	**t aI**	tired
102	59,4671	**l e**	letters
103	59,7232	**l @U**	follow
104	59,9786	**I D**	with
105	60,2311	**s I**	civilized
106	60,4829	**V p**	up
107	60,7314	**m eI**	made
108	60,9778	**w @r**	words
109	61,2232	**t eI**	take
110	61,4686	**I f**	before
111	61,7112	**t i:**	t
112	61,9522	**k r**	cry
113	62,1923	**g @U**	goes
114	62,4322	**d u:**	do
115	62,6716	**e s**	yes
116	62,9091	**t e**	tears
117	63,1462	**k s**	explain
118	63,3806	**aI l**	child
119	63,6147	**r e**	remedies
120	63,8470	**Q z**	was
121	64,0779	**l eI**	airplanes
122	64,3069	**S i:**	she
123	64,5354	**h i:**	he
124	64,7628	**k O:**	calling
125	64,9893	**{ k**	back
126	65,2156	**w aI**	wind
127	65,4380	**V d**	remedies
128	65,6563	**aI k**	strikes
129	65,8744	**r @U**	growing
130	66,0920	**n z**	airplanes
131	66,3086	**r t**	charted
132	66,5221	**r u:**	through
133	66,7351	**s eI**	save
134	66,9473	**d z**	words
135	67,1582	**d r**	dream
136	67,3667	**p l**	airplanes
137	67,5747	**b r**	brace
138	67,7821	**D I**	this
139	67,9892	**U r**	sure
140	68,1948	**eI t**	hateful
141	68,3958	**i: t**	eat
142	68,5968	**r z**	there's
143	68,7970	**S V**	missions
144	68,9967	**l {**	last
145	69,1960	**n Q**	not
146	69,3931	**h Q**	alcohol
147	69,5902	**m I**	million
148	69,7851	**p r**	prize
149	69,9797	**{ s**	astronauts
150	70,1724	**m {**	man
151	70,3643	**aI v**	i've
152	70,5510	**m e**	nightmare
153	70,7360	**n e**	mercenary
154	70,9196	**b l**	blowing
155	71,1020	**l O:**	lost
156	71,2829	**N k**	blankets
157	71,4634	**h @r**	her
158	71,6439	**d eI**	day's
159	71,8241	**s k**	risk
160	72,0037	**I m**	trims
161	72,1833	**d aU**	down
162	72,3626	**i: v**	leave
163	72,5415	**m z**	games
164	72,7202	**i: p**	sleep
165	72,8979	**b {**	back
166	73,0753	**n aI**	night
167	73,2523	**{ z**	has
168	73,4289	**O: N**	strong
169	73,6041	**D eI**	they
170	73,7791	**h @U**	home
171	73,9537	**I v**	lives
172	74,1277	**n aU**	now
173	74,3016	**b aI**	by
174	74,4753	**i: d**	eden
175	74,6477	**l z**	dolls
176	74,8195	**f i:**	fever
177	74,9903	**aI z**	prize
178	75,1609	**D @r**	other

179	75,3299	**{ d**	bad
180	75,4988	**@r l**	girl
181	75,6673	**g r**	growing
182	75,8351	**f aI**	five
183	76,0015	**s p**	explain
184	76,1677	**h e**	head
185	76,3334	**V z**	does
186	76,4991	**d @r**	understand
187	76,6638	**j U**	yourselves
188	76,8274	**i: n**	gasoline
189	76,9906	**n I**	nickels
190	77,1534	**{ v**	have
191	77,3156	**@r n**	iron
192	77,4774	**eI s**	brace
193	77,6367	**p V**	open
194	77,7945	**i: m**	dream
195	77,9522	**s aI**	inside
196	78,1066	**eI m**	games
197	78,2606	**e m**	them
198	78,4145	**@r z**	dollars
199	78,5681	**p i:**	european
200	78,7216	**@r d**	words
201	78,8749	**V b**	rub
202	79,0277	**w O:**	walls
203	79,1797	**k Q**	cost
204	79,3279	**eI b**	baby
205	79,4761	**eI z**	day's
206	79,6198	**@U z**	grows
207	79,7634	**U k**	pocketbook
208	79,9067	**r {**	scraps
209	80,0498	**V g**	again
210	80,1922	**r d**	afford
211	80,3326	**v I**	unswerving
212	80,4717	**u: z**	music
213	80,6090	**eI d**	played
214	80,7461	**m O:**	more
215	80,8816	**k t**	doctor's
216	81,0164	**b Q**	borrowed
217	81,1508	**I d**	kids
218	81,2848	**t Q**	livestock
219	81,4183	**l Q**	lot
220	81,5498	**b e**	better
221	81,6808	**g I**	begin
222	81,8100	**e k**	respects
223	81,9384	**Q d**	god
224	82,0664	**k l**	claimed
225	82,1942	**V w**	away
226	82,3215	**{ m**	am
227	82,4485	**aU @r**	our
228	82,5744	**V k**	buck
229	82,6967	**b eI**	baby
230	82,8152	**aI f**	wife
231	82,9326	**b @r**	robert
232	83,0498	**d e**	dead
233	83,1668	**h aU**	how
234	83,2831	**z V**	reason
235	83,3992	**T r**	through
236	83,5152	**@r s**	understand
237	83,6304	**f l**	afloat
238	83,7451	**g @r**	girl
239	83,8570	**p Q**	pocketbook
240	83,9678	**f @r**	first
241	84,0777	**n l**	only
242	84,1859	**@r i:**	nursery
243	84,2934	**p I**	spilling
244	84,4003	**l u:**	blue
245	84,5067	**l U**	look
246	84,6129	**V f**	afford
247	84,7189	**v V**	civilized
248	84,8247	**u: l**	cool
249	84,9292	**V D**	other
250	85,0333	**t {**	understand
251	85,1372	**N z**	darlings
252	85,2406	**n dZ**	dangerous
253	85,3436	**k i:**	lucky
254	85,4462	**r Q**	robert
255	85,5479	**f eI**	famous
256	85,6487	**I p**	slip
257	85,7489	**O: k**	talk
258	85,8491	**w U**	wouldn't
259	85,9485	**k U**	could
260	86,0479	**f I**	filled
261	86,1469	**m Q**	mocks
262	86,2460	**Q k**	doctor's
263	86,3437	**r O:**	strong
264	86,4410	**f V**	hateful
265	86,5377	**h u:**	who
266	86,6332	**f t**	lifts
267	86,7256	**m p**	simple
268	86,8178	**s l**	slip
269	86,9091	**h aI**	hide
270	86,9999	**I g**	begin
271	87,0897	**@U m**	home
272	87,1787	**p @r**	slippers
273	87,2676	**t O:**	store
274	87,3564	**s @r**	nursery
275	87,4446	**d aI**	die
276	87,5327	**d Q**	dolls

277	87,6204	**p eI**	pain
278	87,7079	**d {**	dance
279	87,7950	**Q p**	drop
280	87,8816	**Q l**	follow
281	87,9679	**u: v**	moved
282	88,0529	**V tS**	much
283	88,1373	**{ l**	alcohol
284	88,2205	**u: n**	spoons
285	88,3033	**m b**	stumbling
286	88,3857	**p e**	pennies
287	88,4679	**f Q**	follow
288	88,5488	**O: t**	astronauts
289	88,6297	**v r**	everything
290	88,7101	**Q m**	common
291	88,7902	**j {**	yeah
292	88,8697	**eI v**	save
293	88,9488	**g U**	good
294	89,0259	**eI l**	tale
295	89,1025	**n u:**	new
296	89,1790	**k eI**	came
297	89,2554	**b aU**	bound
298	89,3317	**{ p**	captured
299	89,4065	**@U k**	soaked
300	89,4806	**i: s**	east
301	89,5547	**aI s**	christ's
302	89,6280	**m @U**	motions
303	89,7010	**V N**	youngest
304	89,7737	**s O:**	saw
305	89,8464	**{ N**	blankets
306	89,9191	**@U v**	over
307	89,9917	**k w**	quite
308	90,0640	**aI @r**	tired
309	90,1362	**z I**	music
310	90,2081	**w @U**	won't
311	90,2797	**k @U**	cold
312	90,3510	**k j**	curiosity
313	90,4221	**r aU**	crown
314	90,4929	**n eI**	names
315	90,5636	**h eI**	hateful
316	90,6343	**g V**	youngest
317	90,7050	**I S**	missions
318	90,7755	**r n**	born
319	90,8459	**p {**	pamphlets
320	90,9161	**s {**	salvage
321	90,9861	**f e**	fell
322	91,0560	**s w**	unswerving
323	91,1257	**r m**	form
324	91,1930	**O: s**	lost
325	91,2603	**@U s**	postmarked
326	91,3270	**s m**	small
327	91,3934	**m @r**	merchant
328	91,4596	**tS eI**	change
329	91,5255	**O: n**	gone
330	91,5911	**t @U**	untold
331	91,6563	**l f**	shelf
332	91,7210	**p U**	poor
333	91,7854	**p s**	ships
334	91,8498	**S U**	should
335	91,9138	**eI S**	consolation
336	91,9769	**@r t**	robert
337	92,0397	**@r aU**	surrounding
338	92,1024	**i: k**	weeks
339	92,1646	**p t**	apt
340	92,2263	**l @r**	dollars
341	92,2880	**j V**	youngest
342	92,3497	**{ f**	drafted
343	92,4110	**I tS**	witch
344	92,4723	**r k**	dark
345	92,5334	**i: T**	anything
346	92,5943	**@U p**	open
347	92,6552	**v z**	yourselves
348	92,7161	**V T**	nothing
349	92,7769	**m u:**	moved
350	92,8374	**k aI**	sky
351	92,8974	**z i:**	frenzied
352	92,9571	**d O:**	door
353	93,0166	**N g**	youngest
354	93,0756	**k e**	cares
355	93,1329	**@U I**	growing
356	93,1887	**h V**	hush
357	93,2443	**u: m**	room
358	93,2997	**i: r**	we're
359	93,3532	**tS @r**	centuries
360	93,4067	**u: t**	brutal
361	93,4596	**l w**	always
362	93,5124	**f {**	family
363	93,5648	**i: V**	european
364	93,6165	**tS I**	children
365	93,6668	**l t**	felt
366	93,7164	**j e**	yes
367	93,7657	**i: tS**	each
368	93,8149	**z d**	civilized
369	93,8639	**g l**	glued
370	93,9126	**dZ i:**	jesus
371	93,9613	**aI I**	trying
372	94,0098	**O: f**	off
373	94,0581	**D @U**	though
374	94,1059	**s aU**	sound

375	94,1536	**n @r**	nursery
376	94,2014	**g eI**	games
377	94,2490	**b OI**	boy
378	94,2961	**@r k**	work
379	94,3417	**e f**	deaf
380	94,3871	**g O:**	gone
381	94,4324	**@U t**	notice
382	94,4776	**m T**	something
383	94,5227	**@U d**	borrowed
384	94,5677	**m w**	someone's
385	94,6118	**k @r**	conquer
386	94,6550	**s u:**	souvenirs
387	94,6974	**e p**	steps
388	94,7393	**b @U**	bowl
389	94,7813	**S @U**	show
390	94,8218	**@r V**	dangerous
391	94,8622	**U l**	full
392	94,9022	**b O:**	born
393	94,9418	**s Q**	sodom
394	94,9811	**v aI**	divided
395	95,0202	**u: d**	you'd
396	95,0592	**v i:**	v
397	95,0982	**i: w**	anyone
398	95,1372	**t w**	between
399	95,1760	**m j**	music
400	95,2145	**S I**	ships
401	95,2528	**T aU**	without
402	95,2909	**U t**	put
403	95,3288	**@r e**	caressing
404	95,3661	**tS V**	merchant
405	95,4031	**v e**	adventure
406	95,4400	**u: s**	truce
407	95,4763	**j Q**	backyard
408	95,5127	**T O:**	thoughts
409	95,5479	**t aU**	hometown
410	95,5826	**e z**	desert
411	95,6172	**p O:**	portion
412	95,6516	**D i:**	these
413	95,6861	**S eI**	shamelessly
414	95,7205	**Q tS**	watches
415	95,7547	**dZ @r**	dangerous
416	95,7887	**dZ I**	jilt
417	95,8226	**{ S**	dash
418	95,8563	**aI V**	giant
419	95,8898	**@r g**	forgive
420	95,9231	**l aU**	flower
421	95,9563	**e D**	together
422	95,9895	**aU d**	crowded
423	96,0226	**f j**	confuse
424	96,0557	**t l**	lastly
425	96,0886	**v d**	moved
426	96,1211	**{ g**	bag
427	96,1536	**V dZ**	manage
428	96,1860	**I T**	without
429	96,2183	**S {**	shattered
430	96,2505	**i: b**	everybody
431	96,2822	**j @r**	familiar
432	96,3138	**dZ e**	gesture
433	96,3450	**Q b**	robert
434	96,3761	**k S**	affection
435	96,4070	**aU T**	mouth
436	96,4370	**m d**	claimed
437	96,4665	**t U**	took
438	96,4957	**T i:**	wealthy
439	96,5249	**p @U**	postmarked
440	96,5539	**e T**	death
441	96,5828	**r l**	darlings
442	96,6113	**j I**	years
443	96,6394	**h O:**	hall
444	96,6674	**eI dZ**	page
445	96,6952	**S aI**	shires
446	96,7228	**Q s**	cost
447	96,7503	**l p**	help
448	96,7778	**e S**	possession
449	96,8051	**l s**	dulce
450	96,8322	**OI z**	toys
451	96,8591	**r s**	yourselves
452	96,8860	**@r T**	earth
453	96,9126	**g z**	rags
454	96,9390	**f u:**	fool
455	96,9654	**eI I**	saying
456	96,9915	**Z V**	casual
457	97,0175	**n f**	confuse
458	97,0434	**k u:**	cool
459	97,0690	**@U b**	robes
460	97,0944	**dZ Q**	jars
461	97,1198	**@r I**	wandering
462	97,1447	**I dZ**	individuality
463	97,1696	**f aU**	found
464	97,1942	**g {**	gasoline
465	97,2186	**O: z**	claws
466	97,2428	**I h**	behind
467	97,2670	**m t**	hometown
468	97,2912	**@r v**	unswerving
469	97,3153	**eI p**	escape
470	97,3395	**f @U**	fold
471	97,3634	**tS {**	challenged
472	97,3872	**l v**	yourselves

473	97,4110	**z @r**	razor
474	97,4346	**e g**	egg
475	97,4581	**OI s**	voice
476	97,4812	**V S**	hush
477	97,5041	**f U**	full
478	97,5269	**n tS**	centuries
479	97,5493	**S e**	shelf
480	97,5717	**@r f**	perform
481	97,5939	**p aI**	piety
482	97,6160	**l j**	million
483	97,6380	**u: T**	youth
484	97,6600	**p aU**	pounce
485	97,6819	**S Q**	shards
486	97,7035	**d l**	handling
487	97,7248	**N I**	slinging
488	97,7460	**l r**	chivalry
489	97,7671	**@U S**	motions
490	97,7883	**{ tS**	scratch
491	97,8090	**m aU**	amounts
492	97,8296	**b j**	bugles
493	97,8503	**v OI**	voice
494	97,8706	**U m**	woman
495	97,8905	**tS aI**	child
496	97,9104	**aU s**	house
497	97,9302	**{ b**	collaborate
498	97,9497	**v eI**	vases
499	97,9691	**k aU**	cowboy
500	97,9883	**@r m**	firm
...	...	...	...

Appendix C

RECORDING SCRIPTS SAMPLE

Vowels (sing each line as an ascending arpeggio)

pat	get	pot	pit	cut	put	heart
ball	pat	get	pot	pit	cut	put
food	ball	pat	get	pot	pit	cut
bird	food	ball	pat	get	pot	pit
peach	bird	food	ball	pat	get	pot
heart	peach	bird	food	ball	pat	get
put	heart	peach	bird	food	ball	pat
cut	put	heart	peach	bird	food	ball
pit	cut	put	heart	peach	bird	food
pot	pit	cut	put	heart	peach	bird
get	pot	pit	cut	put	heart	peach

Diphthongs (sing each line as an ascending arpeggio)

raise	rise	noise	nose	rouse	fears	stairs
cures	raise	rise	noise	nose	rouse	fears
stairs	cures	raise	rise	noise	nose	rouse
fears	stairs	cures	raise	rise	noise	nose
rouse	fears	stairs	cures	raise	rise	noise
nose	rouse	fears	stairs	cures	raise	rise
noise	nose	rouse	fears	stairs	cures	raise
rise	noise	nose	rouse	fears	stairs	cures

Voiced Consonants (sing each line as an ascending arpeggio)

mum	nun	thing	feel	wrong	bin	din
give	mum	nun	thing	feel	wrong	bin
din	give	mum	nun	thing	feel	wrong
bin	din	give	mum	nun	thing	feel
wrong	bin	din	give	mum	nun	thing
feel	wrong	bin	din	give	mum	nun
thing	feel	wrong	bin	din	give	mum
nun	thing	feel	wrong	bin	din	give

vim	this	measure	zing	gin	wasp	yacht
yacht	vim	this	measure	zing	gin	wasp
wasp	yacht	vim	this	measure	zing	gin
gin	wasp	yacht	vim	this	measure	zing
zing	gin	wasp	yacht	vim	this	measure
measure	zing	gin	wasp	yacht	vim	this
this	measure	zing	gin	wasp	yacht	vim

Unvoiced Consonants

short & long-sustained
[f] fill, fat, felt, funky, feel, fog, full, fool, first
[T] thin, thatch, therapy, thunder, theme, thought, enthusiasm, thirteen
[s] sin, salad, send, sun, seize, sought, sob, soot, soon, sir
[S] shin, shadow, shed, shut, sheep, short, shop, should, shoe, shirt
[h] hill, hat, head, hulk, heat, haunt, hot, hook, hoot, hurt
[tS] chin, chat, check, cheek, chortle, chop, choose, chirp

Steady states (sing in 5 different pitches: GDGDG, 2 seconds)

vowels:

pat	pet	pot	pit	cut	put	heart	peach	bird	food	ball

diphthongs:

raise	rise	noise	nose	rouse	fears	stairs	cures

voiced consonants

mum	nun	thing	feel

vim	this	*mea-*	*sure*	zing	gin

Articulations (sing in two different pitches with a difference of an octave)

II		*Ie*		*IQ*	beyond
I{		*IV*		*IU*	
I@		*Ib*	bib	*Id*	kid
Ig	big	*Ip*	chip	*It*	pit
Ik	brick	*Im*	limit	*In*	chin
IN	thing	*Il*	bill	*Ir*	hear
Iv	civic	*ID*	with	*Iz*	whiz
IZ	vision	*If*	if	*IT*	myth
Is	miss	*IS*	fish	*Ih*	
IdZ	bridge	*ItS*	pitch	*Iw*	
Ij		*Ii:*		*IeI*	
IaI		*IOI*		*Iu:*	
I@U		*IaU*		*I3:*	
IA:		*IO:*		*II@*	
Ie@		*IU@*			

eI		*ee*		*eQ*	
e{		*eV*		*eU*	
e@	stairs	*eb*	web	*ed*	bed
eg	legacy, leg	*ep*	kept	*et*	get
ek	neck	*em*	them	*en*	entry
eN	length	*el*	cell, elegant	*er*	errant, hair
ev	never	*eD*	weather	*ez*	desert *(sandy area)*
eZ	measure	*ef*	effluent, chef	*eT*	death
es	mess	*eS*	flesh	*eh*	
edZ	edge	*etS*	fetch	*ew*	
ej		*ei:*		*eeI*	
eaI		*eOI*		*eu:*	
e@U		*eaU*		*e3:*	
eA:		*eO:*		*eI@*	
ee@		*eU@*			

Q_I		*Q_e*		*Q_Q*	
Q{		*Q_V*		*Q_U*	
Q@		*Q_b*	job	*Q_d*	nod
Q_g	fog	*Q_p*	top	*Q_t*	caught
Q_k	mock	*Q_m*	bomb	*Q_n*	John
Q_N	bong	*Q_l*	solemn	*Q_r*	card
Q_v	poverty	*Q_D*	father	*Q_z*	was
Q_Z	mirage	*Q_f*	cough	*Q_T*	brothel
Q_s	fossil	*Q_S*	wash	*Q_h*	yahoo
Q_dZ	logic	*Q_tS*	botch	*Q_w*	
Q_j		*Q_i:*		*Q_eI*	
Q_aI		*Q_OI*		*Q_u:*	
Q_@U		*Q_aU*		*Q_3:*	
Q_A:		*Q_O:*		*Q_I@*	
Q_e@		*Q_U@*			

{I		*{e*		*{Q*	
{{		*{V*		*{U*	
{@		*{b*	tab	*{d*	bad
{g	baggage	*{p*	tap	*{t*	cat
{k	back	*{m*	bamboo	*{n*	band
{N	bang	*{l*	salad	*{r*	caramel
{v	cavalier	*{D*	gather	*{z*	jazz
{Z	casual	*{f*	staff	*{T*	Catherine
{s	cascade	*{S*	cash	*{h*	
{dZ	tragedy	*{tS*	catch	*{w*	
{j		*{i:*		*{eI*	
{aI		*{OI*		*{u:*	
{@U		*{aU*		*{3:*	
{A:		*{O:*		*{I@*	
{e@		*{U@*			

BIBLIOGRAPHY

Amatriain, X, 1998. "METRIX: A Musical Data Definition Language and Data Structure for a Spectral Modeling Based Synthesizer," *Proceedings of 98 Digital Audio Effects Workshop.*

Berndtsson, G. 1996. "The KTH Rule System for Singing Synthesis," *Computer Music Journal, 20:1, 1996.*

Bresin, Roberto & Friberg, Anders, 1999. "Synthesis and decoding of emotionally expressive performance". *Proceedings of the IEEE 1999 Systems, Man and Cybernetics Conference* (SMC'99), Tokyo, IV-(317-322)

Bresin, R. & Friberg, A., 1998. "Emotional expression in music performance: synthesis and decoding", in *TMH-QPSR, Speech Music and Hearing Quarterly Progress and Status Report, 3-4/1998, Stockholm, pp. 85-94*

Bunch, Meribeth. Dynamics of the Singing Voice. *Springer.* Wien, New York, 1997.

Cambouropoulos, Emilios. Publications. [on-line] http://www.ai.univie.ac.at/~emilios/

Cano, P.; A. Loscos; J. Bonada; M. De Boer; X. Serra; 2000. "Voice Morphing System for Impersonating in Karaoke Applications," *Proceedings of the 2000 International Computer Music Conference.* San Francisco: Computer Music Association.

Childers, D.G. 1994. "Measuring and Modeling Vocal Source-Tract Interaction," *IEEE Transactions on Biomedical Engineering 1994.*

Clynes, M, 1987. "What can a musician learn about music performance from newly discovered microstructure principles (PM and PAS)?," *Action and Perception in Rhythm and Music. Royal Swedish Academy of Music No. 55, 1987.*

Cook, P. 1996. "Singing Voice Synthesis History, Current Work, and Future Directions," *Computer Music Journal, 20:2 1996.*

Cook, P.R.1998 "Toward the Perfect Audio Morph? Singing Voice Synthesis and Processing". *Proceedings of 98 Digital Audio Effects Workshop. Barcelona 1998.*

Dubnov, Shlomo & Rodet, Xavier. 1998. Study of spectro-temporal parameters in musical performance, with applications for expressive instrument synthesis in *Pro. IEEE International Conference on Systems, Man, and Cybernetics, (San Diego, USA , Novembre 1998)* [on-line] http://www.ircam.fr/equipes/analyse-synthese/listePublications/articlesRodet/

Dubnov, Shlomo. 1997. Emotion - Is it measurable? Appeared in: *KANSEI - The Technology of Emotion, AIMI International Workshop, Genova 1997* [on-line] http://www.ircam.fr/equipes/analyse-synthese/listePublications/articlesDubnov/index.html

Fitz, K.; L. Haken, and P. Christensen. 2000. "A New Algorithm for Bandwidth Association in Bandwidth-Enhanced Additive Sound Modeling," *Proceedings of the 2000 International Computer Music Conference.*

Friberg, A, 1991. "Generative Rules for Music Performance: A Formal Description of a Rule System," *Computer Music Journal 15:2, 1991.*

Friberg, Anders, 1995 "A Quantitative Rule System for Musical Performance" Summary of thesis [on-line] http://www.speech.kth.se/music/publications/thesisaf/sammfa2nd.htm

Friberg, Anders, 1998. "Musical Punctuation on the Microlevel: Automatic Identification and Performance of Small Melodic Units". *Journal of New Music Research, 1998, Vol. 27, No.3, pp-271-292.*

Friberg, A., 1995. "Matching the rule parameters of Phrase arch to performances of Träumerei: A preliminary study", in A. Friberg and J. Sundberg (eds.), *Proceedings of the KTH symposium on Grammars for music performance May 27,1995, pp. 37-44.*

Goldáraz Gaínza, J. Javier, 1992. Afinación y temperamento en la música occidental. *Alianza Música.*

Kaemperer, Gerard. 1968. Techniques pianistiques. *Alphonse Leduc.* Paris.

Klatt, D.H. 1980. "Software for a cascade/parallel formant synthesizer," *Journal of the Acoustical Society of America, 971-995, 1980.*

Lerdahl, F. and Jackendoff, R. 1983. A generative theory of tonal music. Cambridge. *The MIT Press.*

Narmour, E. 1990. The analysis and cognition of basic melodic structures: the implication-realization model. Chicago: *Univ. Chicago Press.*

Olive, Joseph P., 1997. "The Talking Computer: Text to Speech Synthesis". In HAL's legacy, 2001's computer as dream and reality. Edited by David G. Stork. *The MIT Press.* [on-line] http://mitpress.mit.edu/e-books/Hal/chap6.java/six1.html

Prame, E. 1994. Measurements of the vibrato rate of ten singers. *J. Acoust. Soc. Am. 96 (1994), 1979-1984.*

Prame E. 1997. Vibrato extent and intonation in professional Western lyric singing. *J. Acoust. Soc. Am. 102 (1997), 616-621.*

Roda, A.; Canazza, S. 199. Adding expressiveness in musical performance in real time. *Multimedia Computing and Systems, 1999. IEEE International. Volume: 2 , 1999 , Page(s): 1026 -1027 vol.2*

SAMPA, 2000. Speech Assessment Methods Phonetic Alphabet. Machine-readable phonetic alphabet. http://www.phon.ucl.ac.uk/home/sampa/home.htm

Seashore, C. E. (1936) Psychology of the vibrato in voice and instrument. Studies in the Psychology of Music, Vol. III. Iowa: University Press.

Seashore, C. E. (1967) Psychology of Music. New York: Dover. (Originally published in 1938).

Selfridge-Field, Eleanor.1997. Beyond Midi, The Handbook of Musical Codes. *MIT Press.*

Serra, X. and J. Smith. 1990. "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System based on a Deterministic plus Stochastic Decomposition," *Computer Music Journal* 14(4):12-24.

Smith, J.O., Gosset, P. 1984. "A flexible sampling-rate conversion method," *Proceedings of the ICASSP, San Diego, New York, vol. 2, pp 19.4.1-19.4.2. IEEE Press 1984.*

Sundberg, Johan, 1987. The Science of the Singing Voice. De Kalb, Illionois: Northern Illinois University Press.

Sundberg, Johan, 1989. "Synthesis of Singing by Rule." In M. Mathews and J. Pierce, eds. *Current Directions in Computer Music Research. Cambridge*, Massachusetts: MIT Press, pp. 45-55.

SMARTTALK, 2000. SmartTalk 3.0. Japanese Speech-Synthesis Engine with Singing Capability. http://www.oki.co.jp/OKI/Cng/Softnew/English/sm.htm

Sproat, Richard et al., 1997. Multilingual Text-to-Speech Synthesis. The Bell Labs Approach. Kluwer Academic Publishers, Boston.

Temperley, D. 1997. An algorithm for harmonic analysis. Music Perception, 15, 31-68. [on-line program] http://www.cs.cmu.edu/~sleator/harmonic-analysis/

Temperley, D. 1999. What's Key of Key? The Krumhansl-Schmuckler Key-Finding Algorithm Reconsidered. Music Perception, 17, 65-100. [on-line program] http://www.link.cs.cmu.edu/music-analysis/

Temperley, D. & Sleator D., 1999. "Modeling Meter and Harmony: A Preference-Rule Approach", Computer Music Journal vol. 23, issue 1 Spring 1999.

Todd, N.P. McA., 1992. The dynamics of dynamics: A model of musical expression. *Journal of Acoustical Society of America*, 91, 3540-3550.

Todd, N.P. McA., 1995. The kinematics of musical expression. *Journal of Acoustical Society of America*, 97, 1940-1949.

Todd, Neil. 1985. A Model of Expressive Timing in Tonal Music. Music Perception 3/1, 33-58.

Verma, T. S.,T. H. Y. Meng. 2000. "Extending Spectral Modeling Synthesis With Transient Modeling Synthesis" *Computer Music Journal* 24:2, pp.47-59.

VOCALWRITER, 2000. Vocalwriter. Music and Vocal synthesis. http://www.kaelabs.com/

Widmer, Gerhard. 1995. Modelling the Rational Basis of Musical Expression. Computer Music Journal 19(2), pp.76-96, MIT Press. [on-line] http://www.ai.univie.ac.at/cgi-bin/tr-online?number+93-20

INTERNET LINKS

RESEARCH GROUPS

Royal Institute of Technology. Stockholm.
Department of Speech, Music and Hearing
http://www.speech.kth.se/music/

IRCAM
The Sound Analysis/Synthesis team, headed by Xavier Rodet
http://www.ircam.fr/equipes/analyse-synthese/

The Center of Computational Sonology of the University of Padova.
http://www.dei.unipd.it/ricerca/csc/intro.html

The University of Edinburg.
The Music Informatics Research Group.
http://www.dai.ed.ac.uk/groups/aimusic/

Austrian Research Institute for Artificial Intelligence. Machine Learning Group.
http://www.ai.univie.ac.at/oefai/ml/ml-mus.html

Artificial Intelligence Research Institute
Spanish Scientific Research Council (CSIC)
http://www.iiia.csic.es/Projects/music/

NICI. University of Nijmegen.
Music Mind Machine.
http://www.nici.kun.nl/mmm/

Laboratorio di Informatica Musicale
Università di Genova - Teatro dell'Opera di Genova Carlo Felice
The Laboratorio di Informatica Musicale carries out projects on scientific and music research, design, development and experimentation of technologies and systems for music, dance, theatre, edutainment and museums.
http://musart.dist.unige.it/

MIT Speech Communication Group
http://www.mit.edu:8001/afs/athena.mit.edu/course/other/speech/www/index.html

National Center for Voice and Speech
An inter-disciplinary, multi-site team of investigators dedicated to studying the powers, limitations and enhancement of human voice and speech
http://www.ncvs.org/

MUSICAL EXAMPLES

Musical Emotions
http://www.speech.kth.se/~roberto/emotion/

Vibrato and portamento
http://www.nici.kun.nl/mmm/sound-examples/dh-95-b.html

SOFTWARE

Superconductor - Manfred Clynes
Lifelike classical computer music
http://www.superconductor.com/

Dr. Speech software is a comprehensive speech/voice assessment and training software system that's easy-to-use, portable, and affordable. This software is intended for use with professionals in voice and speech fields
http://www.drspeech.com/

Director Musices
Software for Automatic Music Performance
http://www.speech.kth.se/music/performance/download/

SINGING VOICE SYNTHESIS

SMARTTALK Version 3.0,
the First Japanese Speech-Synthesis Engine with Singing Capability
http://www.oki.co.jp/OKI/Cng/Softnew/English/sm.htm

LYRICOS: Synthesis of Singing Voice
http://users.ece.gatech.edu/~macon/Sing/index.html

Hui-Ling Lu
Toward a high-quality singing synthesizer
http://ccrma-www.stanford.edu/~vickylu/research/index.htm

Singing Synthesis
Perry Cook
http://www.cs.princeton.edu/~prc/SingingSynth.html

SPEECH SYNTHESIS

AT&T Labs-Research
Next-Generation Text-to-Speech
http://www.research.att.com/projects/tts/

Examples of Synthesized Speech
http://www.ims.uni-stuttgart.de/phonetik/gregor/synthspeech/examples.html

Linear Predictive Vocoder as a Model for Human Speech Production -- A tutorial - This tutorial explains the principle of the human speech production with the aid of a Linear Predictive
Vocoder (LPC vocoder) and the use of interactive learning procedures.
http://www.kt.tu-cottbus.de/speech-analysis
Model of the Human Speech Production (Java applet)
http://www.kt.tu-cottbus.de/speech-analysis/simulation.html

LDC Catalog
The LDC's Catalog contains 180 corpora of language data.
http://www.ldc.upenn.edu/Catalog/index.html

The Festival Speech Synthesis System
Festival is a general multi-lingual speech synthesis system developed at CSTR. It offers a full text to speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme-based command interpreter for general control.
http://www.cstr.ed.ac.uk/projects/festival/

CHATR (Generic Speech Synthesis System)
http://www.itl.atr.co.jp/chatr/

The MBROLA Project
Towards a Freely Available Multilingual Synthesizer
http://tcts.fpms.ac.be/synthesis/

Bell Laboratories
Text-to-Speech Synthesis
http://www1.bell-labs.com/project/tts/

Speech Synthesis Systems
The aim of this page is to present a cross-section of various speech synthesis systems. Some of these are academic, others are commercial. They represent many different techniques and will hopefully give the reader some idea of what is currently possible with speech synthesis technology.
http://www.cs.bham.ac.uk/~jpi/museum.html

PHONETICS

SAMPA
computer readable phonetic alphabet
http://www.phon.ucl.ac.uk/home/sampa/home.htm

Welcome to the Speech at CMU Web Page.
Carnegie Mellon University is dedicated to speech technology research, development, and deployment,
and we hope this page will be a vehicle to make our work available online.
CMU has a historic position in computational speech research, and continues to test the limits of the art.
http://www.speech.cs.cmu.edu/
The CMU Pronouncing Dictionary
http://www.speech.cs.cmu.edu/cgi-bin/cmudict

The American Heritage® Dictionary of the English Language (Fourth Edition)
Over 90,000 entries feature 10,000 new words and senses, 70,000 audio word pronunciations, 900 full-page color illustrations, language notes and word-root appendixes.
http://www.bartleby.com/61/

American Spoken English in Real Life (Spoken ESL)
to know what Americans naturally say in real life and to talk so that they easily understand what you say
For children, illiterates, workers, students all ages and to get rid of, not have a foreign accent
http://americanspokenenglish.com/