Estudi de casos pel control de síntesi d'instruments musicals virtuals mitjançant la veu cantada.

Case studies for controlling virtual instruments synthesis using the singing voice

Memòria científica de l'Estada de Recerca a Music Technology Area, Faculty of Music, McGill University, Montreal, Canada
(Octubre-Desembre 2005)

Beca AGAUR BE-2005. Generalitat de Catalunya

Jordi Janer
Music Technology Group, Universitat Pompeu Fabra, Barcelona
jordi.janer@iua.upf.edu

January 12, 2006

Contents

1 Introducció

L'objectiu d'aquesta recerca és aprofundir en l'estudi de la veu cantada com a eina per controlar la síntesi d'un instrument musical virtual. Aquest procés inclou el disseny i la implementació d'algoritmes d'anàlisis i de síntesi basats en tècniques espectrals de processament digital del senyal. Un cop digitalitzat el senyal acústic de la veu, s'analitzarà el senyal i s'extreuran una sèrie de paràmetres expressius d'alt nivell de l'acció del cantant. Aquest procés es realitza síncronament a interval petits de temps, de manera que a cada instant disposarem de diversos paràmetres que ens modelaran l'acció del cantant. En una segona fase, aquests paràmetres s'assignaran als controls d'un sintetitzador de so, l'instrument musical virtual.

Aquest document és una memòria cinetífica de la recerca realitzada durant una estada a la Music Technology Area (Sound Processing and Control Lab), Faculty of Music, McGill University, Montréal, Canada. El periòde de l'estada fou d'octubre del 2005 a desembre del 2005. Els professors responsables al centre visitant foren el Dr. Gary Scavone i el Dr. Philippe Depalle.Part de la recerca presentada en aquesta memòria has estat realitzada amb la colaboracó del Dr. Vincent Verfaille.

L'estructura d'aquest document és:

- Descriptció dels objectius de recerca
- Aspectes generals del control de veu
- Anàlisis del timbre utilitzant Cepstrum Discret
- Estudi de casos: Clarinet, Baix i Visualització

NOTA: Aquest document està escrit en anglès per tal de poder ser revisat pels investigadors del centre visitant.

2 Introduction

The aim of this research is to study the singing voice for controlling virtual musical instrument synthesis. It includes analysis and synthesis algorithms based on spectral audio procesing. After digitalizeing the acoustic voice signal in the computer, we extract a number of expressive descriptors of the singer. This process is achieved synchronously, thus we track all the nuance of the singer performance. In a second stage, the extracted parameters are mapped to a sound synthesizer, the so-called digital musical instruments.

This document is a scientific report of the research carried out during a stay at the Music Technology Area (Sound Processing and Control Lab), Faculty of Music, McGill University, Montréal, Canada. The period of the stay was from October 2005 to December 2005. The supervisors in the visiting institution were Dr. Gary Scavone and Dr. Philippe Depalle at the Music Technology Area, Faculty of Music, McGill University. Also, this research described here has been partially done in collaboration with Dr. Vincent Verfaille.

The structure of this document:

- Description of the research objectives
- General issues on the voice control
- Timbre analysis approach using the Discrete Cesptrum
- Case studies: Clarinet, Bass guitar, Visual feedback

3 Objectives

The initial goals of the research stay described in the grant application document (*Sol. licitud Beca, March 2005*) have been further developed during the months previous the starting date. The research objectives and schedule were specified with Dr. Philippe Depalle during the DAFX'2005 Conference (20-22/09/2005) in Madrid. Finally, the research focused mainly on the control of a virtual clarinet synthesizer, and not on the control of a bass guitar as specified in the research proposal document. This decision permited a stronger collaboration with the resarchers at the visiting institution, since they are currently developing a Wind Instrument Synthesizer. The objectives are summarized in the following list:

1. Preliminar study for the integration of UPF Voice-Features Extraction Module (Voctro) and McGill's Clarinet Synthesizer (Ssynth) using Open Sound Control (OSC)
2. Formant tracking research.
3. User experiments with students in music preformance.
4. Publication of the results in conference proceedings.

4 General issues on the voice control

Singing voice qualities have been exploited in the history of music since ancient times. The emotion transmitted by an opera singer is indubitable, probably due to the fact that the voice is *per se* an organic musical instrument. It is not the aim of this research to study in detail the characteristics and qualities of the singing voice as a musical instrument. Rather, we address the high degree of expression and the nuances of the singing voice in order to exploit it as a musical controller. Music Technology has tackled the singing voice mainly from the analysis / synthesis perspective, putting efforts in "inventing" a computer that sings as a human. Another topic, in which the singing voice is involved, is *score following* systems [9].

In the work presented here, the singing voice acts as a controller. The system examines the characteristics of the captured acoustical signal, which at its turn, drives the parameters of a synthesis engine. Other approaches of "sound-controlled sound synthesis" are found in the literature. Tristan Jehan [6] presents a system developed at MIT which uses the timbre characteristics of a continuous input stream to recreate the output with characteristics of another pre-analyzed instrument. In another approach by Miller Puckette et al. [10],

the goal is to map a low-parametric timbre space of an input audio stream onto a pre- analyzed output audio stream in real-time. Also related to this work, we should include *PitchToMidi* systems, which were first introduced in the 80's as hardware devices. For our purposes, though, MIDI presents mainly two limitations. First, it is an event-based protocol, and the voice -as many other instruments- varies its sounds in a continuous manner. Second, the available bandwidth offers an insufficient time resolution.

Our previous work on voice control, we proposed to use Open Sound Control for sending extracted parameters of the voice to a sinthesizer [4]. Using this system configuration we can send synchronous parameters at a frame rate speicified by the analysis module. Since the analysis is based on spectral techniques, the frame rate will depend on the FFT(Fast Fourier Transform) configuration parameters. We refer the reader to [4], for further information on the feature extraction module.

When bulding a musical controller, it should be ergonomic and universal, in the sense that it should be comfortable for a majority of users. Using the voice as input device, however, presents a limitation: the individual vocal range. Each person has a typical vocal range, usually two ocatves. In singers, the tessitura, is the part of the vocal range which the singer is most comfortable producing: for tenors, C3 to C5; for baritones, A2 to A4; for basses, E2 to E4; for sopranos, C4 to C6; for mezzo-sopranos, A3 to A5; for altos, E3 to G5.

	note range	Hz
bass	E2 to E4	82.41 to 329.63
bariton	A2 to A4	110.00 to 440.00
tenor	C3 to C5	130.81 to 523.25
alto	E3 to G5	164.81 to 783.99
mezzo-soprano	A3 to A5	220.00 to 880.00
soprano	C4 to C6	261.63 to 1046.50

Table 1: Singers typical pitch range.

We can partially overcome this problem by letting the user to configure the system. Optionally, methods to automatically do this configuration for naive users can be studied.

5 Timbre analysis methods

One of the relevant parameters that we will want to extract from the voice are those that characterize the timbre. In speech or singing voice, a description of timbre refers mainly to two aspects, the quality of the voice (hoarseness, breathiness), and those related to the vocal tract. The formants are the resonances caused by the vocal tract, acting as an acoustic filter to the glottal pulsed signal. The resulting output voice signal can be explained, thus, by a source-filter model. Our goal will consist in identifying the formants (frequencies, amplitude and bandwidth) that are present in the spectral envelope. The two largest

spaces in the vocal tract are the throat and mouth. Therefore, thay produce the two lowest reson ant frequencies, or formants. These formants are designated as F1 (the throat/pharynx) and F2 (the mouth). The formants characteristics are directly related to the sound production of voiced sounds. We argue that controlling the palette of voiced sounds can be highly useful in terms of expressive musical control.

In this section we tackle the description of the vocal tract with two different approaches: spectral centroid and discrete cepstrum-based formant tracking. Both approaches differ in complexity as well as in accuracy, as we will see.

Finally, for the experiments with the clarinet synthesizer, we chose to use the centroid as a measure of the timbre (vowel). This decision was forced by the fact that at the time of running the experiments the formant tracking algorithm described in section 5.2 worked only offline due to its implementation in Matlab. In order to optimize the the formant tracking algorithm and to run it in real-time, we will reimplement in C++ language.

5.1 Spectral Centroid

A low level descriptor commonly related to the brightness of a sound is the spectral centroid, defined in equation **??**. Dealing with harmonic or pseudo-harmonic signals, we can use a simplified formula. It takes only values of the harmonic spectral peaks 1 and is referred in the literatura as Harmonic Spectral Centroid[8].

$$c_f = \frac{\sum_{k=0}^{L} a_k f_k}{\sum_{k=0}^{L} a_k} \tag{1}$$

Our objective is to calculate the spectral centroid of the voice signal, and map it, in a second step to the timbre dynamics (*brightness*) in the synthesizer.

5.2 Formant Tracking based on Discrete Cepstrum Analysis

Several methods of the spectral envelope have been proposed such as Linear Prediction Coefficients, Cepstrum or the Discrete Cepstrum. Linear Prediction is an all-pole representation. Its main disadvantage is that for high pitched voices the spectral envelope fits only the first partials, which will depend on the fundamental frequency. On the other hand, spectral zeros are not estimated, which might be important on some cases. Cepstral Analysis models poles and zeros with equal importance. However, with small number of coefficients, Cepstral Analysis oversetimates the bendwidth of the formants. Recent studies [11] have shown that the most appropriate technique for periodic signals was the Discrete Cepstrum, first presented by Galas [3]. This method takes a number of spectral values, in general, the amplitude of the harmonic partials, and calculates a spectral response that fits all the specified points. For voices with high fundamental frequency, this methods apears to be more convenient.

Our objective in the identification of formants will be to get some continuous parameters for the timbre. Actually, for our purpose of using the voice as

a controller, the problem can be reduced to the identification of the first two formants frequencies. It will give us an estimation of the sung vowel, and at the same time, the resulting values will be continuous, since the vocal tract position varies slowly compared to the frame rate.

5.2.1 Method description

Thsi method was implemented in Matlab for a rapid prototyping. Real-time performance will require an optimized implementation in C++. The implemented method for estimating the frequencies of the first two formants can be decomposed in the following steps:

1. Spectral Analysis. Find harmonic partials
2. Discrete Cesptrum Analysis. Params (λ and order)
3. Find local maxima (candidates) in the Discrete Cesptrum Envelope
4. Add frequency candidates: inflection points and sub-band centroid
5. Compute cost probability for each frequency candidate. $P_{cost,F1}$ and $P_{cost,F2}$
6. Compute transition probability for each frequency candidate. $P_{trans,F1}$ and $P_{trans,F2}$
7. Compute F_1 and F_2 trajectories
8. Post-processing

Regarding the first step, since this method takes the amplitude of the partials as input data, a previous harmonic spectral analysis is required. For this purpose, we used the SMSTools application[1] developed at the Universitat Pompeu Fabra, Barcelona. The pitch estimation used in this application is based on the Two-way mismatch method[1].

5.2.2 Discrete Cesptrum Analysis

We implemented the algorithm proposed by Galas and Rodet [3] in Matlab. The principle is to calculate a set of cepstral coefficients so that the spectral response matches certain values, usually the harmonic partials.

Given L amplitudes a_k corresponding to the frequencies f_k, we find the real cepstrum coefficients $c_0, ..., c_n$, whose frequency response minimizes the error function in equation2.

$$\epsilon = \sum_{k=1}^{L} |\log a_k - \log |S(f_k)||^2 \tag{2}$$

$$\log |S(f_k)| = c_0 + 2 \sum_{i=1}^{n} c_i \cos(2\pi f_i) \tag{3}$$

[1] `http://www.iua.upf.es/mtg/clam/`

Assuming $|S(f_k)|$ is real and simmetrical, its Fourier Transfor is reduced to a sum of cosinus. We have to minimize ϵ to find the vector c that contains the spectral coefficients. We can write in form of a matrix:

$$\epsilon = ||\mathbf{a} - \mathbf{Mc}||^{\mathbf{2}} \quad \text{with} \quad a = \begin{bmatrix} \log(a_1) \\ \vdots \\ \log(a_L) \end{bmatrix} \tag{4}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 2\cos(2\pi f_1) & 2\cos(2\pi f_1 2) & \cdots & 2\cos(2\pi f_1 n) \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & 2\cos(2\pi f_L) & 2\cos(2\pi f_L) & \cdots & 2\cos(2\pi f_L n) \end{bmatrix} \tag{5}$$

By zeroing the partial derivatives of the equation (**??**), the solution can be expressed as:

$$\mathbf{c} = \left(\mathbf{M}^{\mathrm{T}}\mathbf{M}\right)^{-\mathbf{1}} \mathbf{M}^{\mathrm{T}}\mathbf{a} \tag{6}$$

However, the resulting matrix $\mathbf{MM}^{\mathrm{T}}$ is generally bad conditioned and singular if $(p \geqslant L)$, yielding meaningless results in some cases. Note that for high pitch the value L might be low. To overcome this problem Galas and Rodet[3] proposed to replace each amplitude-frequency pair a_k, f_k by a cluster of points in order to add compelxity to the minimization problem and finding a more consistent solution. In some cases, such as when in a broad frequency region no point is specified, this method was not satisfactory. Cappé and Laroche [7] proposed a regularization technique that imposes additional constraints to the logarithmic envelope. This constraint seeks to find a *smooth* envelope, in other words, this is a penalty function that will be small if the estimated envelope is smooth and high if it contains high variations. Finally, the error function (equation 2) is defined as:

$$\epsilon_r = (1-\lambda)\sum_{k=1}^{L} |\log a_k - \log|S(f_k)||^2 + \lambda\left(\int_0^1 \left[\frac{d}{df}\log|S(f_k)|\right]^2 df\right) \tag{7}$$

The parameter λ is the regularization parameter and controls the importance of the smoothness constraint. Typical values of λ is 10^{-4}. We can compute the final solution vector $\mathbf{c}$ a:

$$\mathbf{c} = \left[\mathbf{M}^T\mathbf{M} + \frac{\lambda}{\mathbf{1}-\lambda}\mathbf{R}\right]^{-\mathbf{1}} \mathbf{M}^T\mathbf{a} \quad \text{with} \quad \mathbf{R} = \mathbf{8}\pi^{\mathbf{2}} \begin{bmatrix} 0 & & & & \\ & 1^2 & & & \\ & & 2^2 & & \\ & & & \ddots & \\ & & & & n^2 \end{bmatrix} \tag{8}$$

5.2.3 Cost Functions. Heuristic Rules

For finding the formant frequencies we select the candidates from the set of found local maxima in the Discrete Ceptrum Envelope. Then, for each candidate maximum, we compute two cost probabilties, one for F1 and one for F2

vowel	F1	F2
/a/	750	1300
/e/	500	1800
/i/	300	2000
/o/	500	1000
/u/	300	700

Table 2: Typical first two formant frequencies for five vowel sounds evaluated

($P_{cost,F1}$ and $P_{cost,F2}$). These cost probabilities are computed using heuristic rules wihch observe the discrete cesptrum envelope and the typical patterns for vowel sounds formant frequencies. The final cost probablity for each maxima will be the multiplication of all rule probabilites. Vocal tract articulation varies in time, but it varies slowlier than the frame rate. It means that the first two formant frequency trajectories will vary slowly as well. In order to get smooth trajectories, and make the algorithm more robust, we use fuzzy logic to the cost functions (P_{cost}). In particular, the cost functions will be Gaussian functions whose parameters σ, μ depend on each rule.

A first approximative list of the proposed heuristic rules cover:

- Amplitude relation to the maximum spectral value
- Formant frequency in valid range $[200 - 2700Hz]$
- Typical sub-band energy distribution

In order to provide more robustness to the formant frequencies estimation algorithm, it seems apporpiate looking at the energy distribution in different frequency sub-bands. From previous research on speech and the singing voice[2] we can assume that the first two formant frequencies will be bounded to a certain frequrency range. For the first formant it is from $200Hz$ to $1500Hz$; and for the second formant from $700Hz$ to $2700Hz$.

Our goal is to roughly identify the produced vowel sound, but since it is not a speech recognition system, we have some degree of freedom left. It is used for focusing on robustness issues rather than on accuracy. In this sense we chose only five typical vowels, which will give the reference formant frequencies for our estimator. We considered five vowel sounds that are common of different languages. The set of vowel sounds are /a/,/e/,/i/,/o/ and /u/. In the table 2, we can observe the typical values for the formant frequencies of the aforementioned vowels.

On the other hand, we divide the spectrum in four frequency sub-bands with the following cut-off frequencies: 1:$200-500Hz$, 2:$500-900Hz$, 3:$900-1700Hz$ and 4:$1700-2700Hz$. In each band, we derive heuristic rules based on the probablity that a formant frequency is present within the sub-band. These rules will rely on the typical energy patterns of the vowel sounds described before (Table. 2). Two values are computed for each sub-band: mean energy and centroid.

The proposed rules for the sub-band energy distribution are summarized in the table 3:

band	vowel	F1	F2
1	/i/	$E_1 > E_2 + E_3$	
1	/e/	$E_1 > E_2 + E_3$	
1	/a/	$E_1 > E_2 + E_3$	
2	/u/	$E_4 > E_3$	
2	/u/		$E_1 + E_2 > E_3 + E_4$
3	/u/		$E_1 + E_2 > E_3 + E_4$
4	/o/		$E_1 + E_2 > E_3 + E_4$
4	/u/		$E_1 + E_2 > E_3 + E_4$

Table 3: Sub-band energy rules used for the costs functions.

5.2.4 Validation tests

Once we have set the heuristic rules, we proceed in the validation of the system that will help us in the final parameter tuning. Mainly, we will want to find optimal values for the Discrete Cesptrum Order, and the lambda variable. Other parameters that can be tuned are the amount of cloud point added to the original partial values.

Synthetic test signal The procedure consists in generating a synthetic signal with a particular spectral envelope. This envelope has two resonances, corresponding to two artificial formants situated a the frequencies $F1 = 750Hz$ and $F2 = 1300Hz$. For each formant, the filter designed is a biquad structure with two poles whose characteristiccs are specified by the resonance frequency in radians ϕ_c and the damping factor R. When R is close to one, it can be defined as $R = exp^{-\pi BT}$, being T the sampling period in seconds[2].

$$y(n) = b_0 x(n) - 2Rcos(\phi_c)y(n-1) - R^2 y(n-2) \tag{9}$$

The theoretical bandwidth of the implemented filters are $B1 = 25Hz$ and $B2 = 60Hz$. In the figure, we observe the block diagram of the synthetic signal.

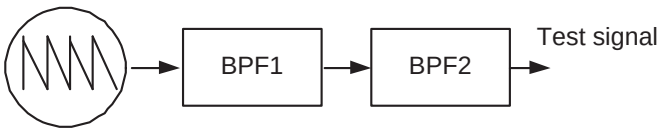

Figure 1: The sawtooth oscillator and two ressonant filters with center frequencies of $700Hz$ and $1300Hz$. A anti-aliasing low-pass filter is added at the end.

The oscillator is a saw-tooth which produces a spectrum with all the harmonics and spectral slope of $-6dBOctave$. Our test signal will be a saw-tooth

[2] http://ccrma.stanford.edu/~jos/filters/Resonator_Bandwidth_Terms_Pole.html

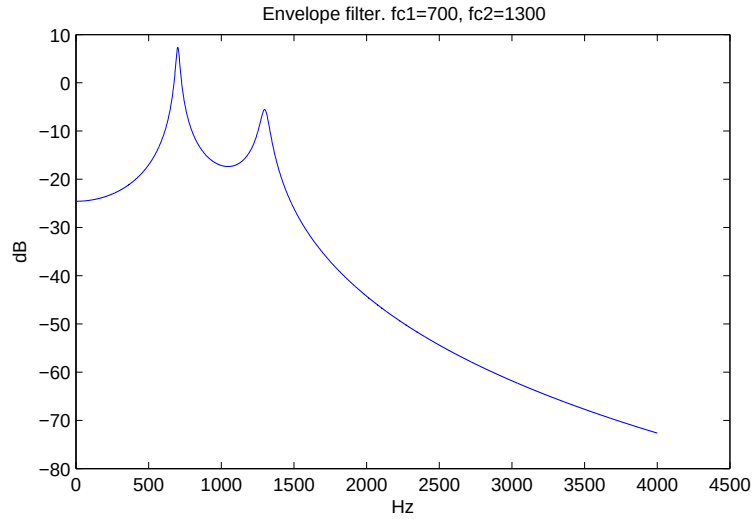

Figure 2: Spectral response of the filter two biquad in cascade, one at $700Hz$ and the other at $1300Hz$.

with increasing frequencies since our goal is to test the efficiency of the formant frequency estimation algorithm in terms of fundamental frequecy. Therefore, we generated a test signal with a duration of 2 seconds, whose fundamental frequency varies from $100Hz$ to $500Hz$ linearly. This type of signals are called *chirp*, and the instantaneous phase ϕ_i is calculated as the integral of a linear function, which is the instantaneous frequency f_i.
In the following figures, we can observe the estimated frequencies of the first two formants and the relative error to the correct filter center frequencies ($F1 = 750Hz$ and $F2 = 1300Hz$).

An error signal is computed for different order of the discrete cepstrum ($p = \{20, 30, 40, 50\}$). The following figure (Fig. 5.2.4) shows the error (y-axis) and the evolution when changing the fundamental frequency of the test signal (filtered sawtooth). The error is computed as shown in the equation 10. IT measures the difference between the estimated formant frequencies and the original ones ($fc1 = 700Hz$, $fc2 = 1300Hz$).

$$err = sqrt(((\hat{f1} - fc1)/fc1).^2 + ((\hat{f2} - fc2)/fc2).^2) \tag{10}$$

Looking at the error figure, we observe a peak for a fundamental frequency of $385Hz$. This peak is motivated by the subsampluing of the spectrum. More precisely, when the algorithm looks for the formant frequencies candidates, looks first for local maxima, and in a seconds step for inflection points in order to have more room for spectrum sub-sampling errors. In this particular case, in addition to the spectrum subsampliung, the test signal spectrum (filtered saw-tooth) does not present any inflection point in the region $100 - 300Hz$. Therefore, the algorithm lack of candidates and gives erronous result. We consider that this situation is not found in real voice signal where the spectrum will not have a perfect smooth decay.

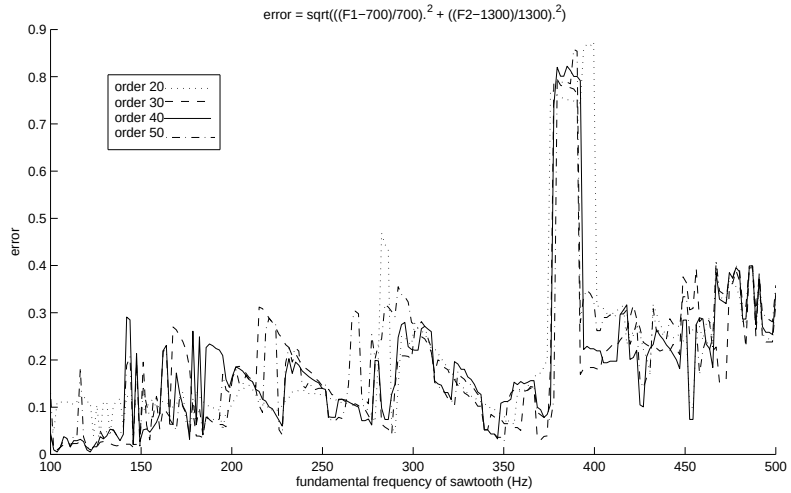

Figure 3: Measurement of the relative error of the center frequencies $fc1 = 700$, $fc2 = 1300$. The test signal fundamental frequency varies from $100Hz$ to $500Hz$ along time axis.

The conclusions of the test are that a higher coefficent order does not imply a more accurate in the estimation of the formant frequencies. It is important to notice that for certain values of fundamental frequency, the formant frequencies estimation fails completely. This is due to the sub-sampling of the spectrum for high pitched signals (fundamental frequencies). In that case, the harmonic partials of the signal are not affected by the formant filters, and thus, the discrete ceptrum lacks of information. This is one of the major problems of formant tracking and there is no straight-forward solution for this. A way to address this issue would be to perform a paralell analysis with the residual signal, after descomposing the spectra in two components sinusoidal and residual[12]. However, to partially overcome this issue, we propose to use additional frequency candidates such as inflection points in the discrete cepstrum, and sub-band centroid frequencies.

Real voice tests As a second step, we run some tests using real voice recordings. Looking at the results achieved, we consider the algorithm to be satisfactory. The sound examples consist of one vowel sequence test of a male voice at $120Hz$, and a short real singer excerpt. In the figure 6, we see the formant frequency candidates (crosses) and the local maxima (diamonds). A closer look allows us to identify five different regions that correspond to the five sung vowels.

The figures Fig. 7 and Fig. 8 depict the waveform of the sound example and the estimated first two formant trajectories.

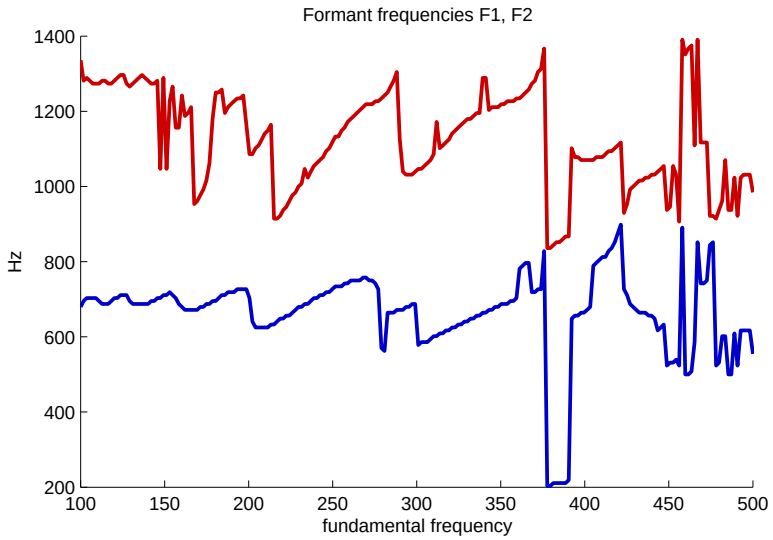

Figure 4: Estimation of the first two formants with the test signal. The test signal fundamental frequency varies linearly from $100Hz$ to $500Hz$ along time axis. Order of the discrete cesptrum is 30.

6 Case studies

One of the main goals of the resarch stay was to run experiments with music students of the Faculty of Music, McGill University. These experiments constributed to evaluate the system and to derive new control strategies to integrate.

- Clarinet synthesis
- Bass guitar
- Visual Feedback

6.1 Clarinet synthesis control: Ssynth

We have run an experiment for controlling with the voice a wind instrument synthesizer under development at McGill University's Music Technology Area (*Ssynth*). Carrying out this experiment was the main motivation for doing the research stay at the visiting lab. We ran several experiments that aim to compare the voice with a MIDI windcontroller for modifying a perceptual attribute of the synthsizer. In our case, our goal is to control the timbre dynamics of a clarinet sound. Although a valid experiment requires at least a dozen of participants, for our purpose a smaller number of participants was sufficient. We had five subjects in total, three singers using the voice as a controller; and two saxophone performers using the windcontroller. All participants come from the Faculty of Music, McGill University, Montreal.

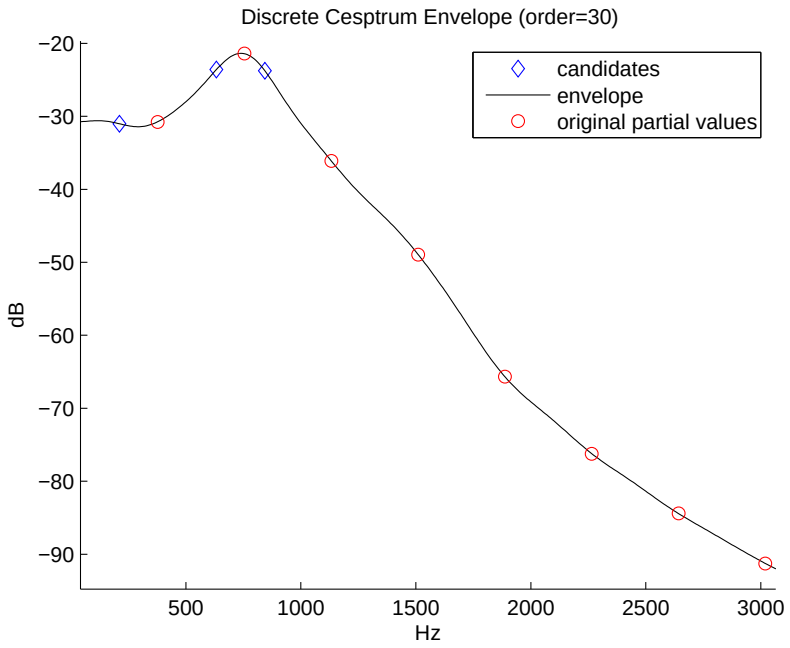

Figure 5: Discrete Cesptrum Envelope for a $order = 30$. Red circles correspond to original harmonic partials of the signa spectrum. Blue diamonds correspond to the selected candidates. Fundamental frequency of the test signal is $385Hz$.

6.1.1 System description

As we already mentioned, these experiments use two different musical controllers for one unique synthesizer. The final goal is to compare the capabilities of the singing voice with the ones of an existing controller dedicated to wind insturments such as the Yamaha MIDI WX5. Obviously, this comparison is not fair, in the sense that the singing voice has inherent limitations for synthesis control. For instance, playing a rapid note sequence can be easily performed with the windcontroller fingering but is hard to achieve by singing. We are aware of that, thus, our mission is to explore the qualities of the voice for other expressive resources.

The experiment set-up consists of two computers, one MIDI interface, a microphone and a windcontroller (Yamaha WX5). As shown in the figure 9, for voice control, the microphone capture the acoustic voice signal that inputs a laptop PC soundcard. The voice analysis stand-alone application (Voctro), extracts a number of parameters from the voice and converts them as Open Sound Control(OSC) messages. Then, OSC messages are sent over the local network to the synthesizer. For the windcontroller, the WX5 send MIDI messages to a MIDI interface that is conected via USB to a PowerPC G5. In the PowerPC, the MIDI messages are converted to OSC messages that meet the requirements of the internal protocol of the synthesizer.

The *Ssynth* is an additive synthesizer that uses a database filled with pre-

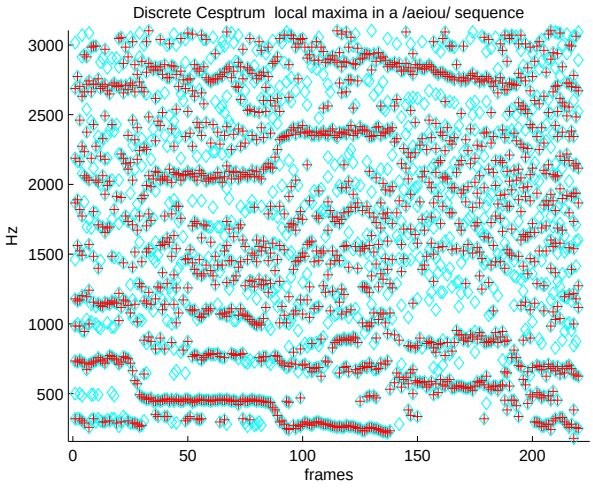

Figure 6: Voice sound example: sequence of five vowels. Frequency candidates are marked as crosses and local maxima of the discrete cepstrum as diamonds.

recorded and pre-analyzed notes. It allows to generate sounds that interpolate betrween to original recorded instrument, dynamics and pitch. This document reports only the research carried out that focuses on the voice control. For further details on the synthesizer used in teh experiment, we refer the reader to [13] or to the website of the Music Technology Area [3]. Concerning the parameters extracted from the voice signal were: energy, pitch and harmonic spectral centroid. The goal was to control the dynamics of the synthesized sound using the "vowel-space". For this purpose, we designed a one-to-one mapping: Energy controls the loudness level of the synthesizer, Pitch the fundamental frequency of the synthesizer and the Harmonic Spectral Centroid controls the dynamics.

Overview of the system describing the different parts.

Voctro - Voice Controller VOCTRO is a system developed at UPF that aims to use the singing voice as a musical controller. It consists of a feature extraction module that analyzes the voice and a communication module that converts these feature into Open Sound Control (OSC) messages, which are send to a client computer via UDP through a network. Note that technically, a single local computer can perform both voice analysis and sound synthesis.

Ssynth Virtual Wind Instrument The virtual instrument used in the experiment is a wind instrument synthesizer developed at McGill University by Vincent Verfaille and others which is based on work done at Ircam for the ESCHER project, a previous project developed at Ircam. More information on the Escher system is found in Marcelo Wanderley's PhD Thesis [13]. It addresses mapping issues as well as gestural control experiments. The synthesizer runs on PureData [4].

[3] `http://www.music.mcgill.ca/musictech/`
[4] `http://www.puredata.org`

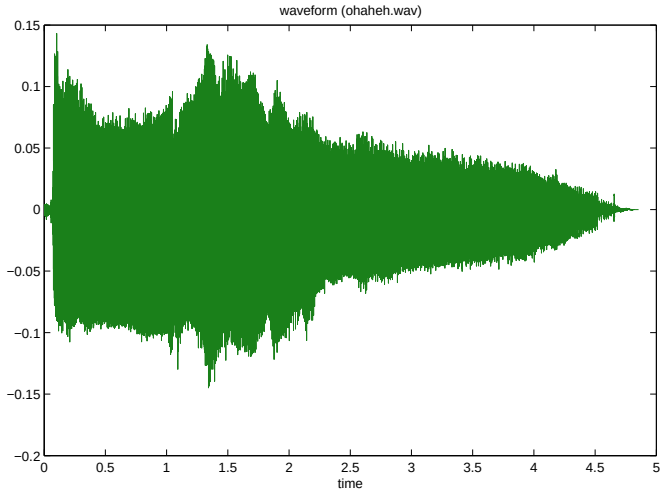

Figure 7: Real singing excerpt. Waveform.

Latency An important issue of Digital Music Instruments is the latency. Although audfio interfaces and computer processor speds have increased in the last yeras, we still have to deal with a noticeable latency. In our experiment, we only focus on the sustained part of a note. Therefore, we have room for latency, which is mainly noticeable during the note attack. Studies indicate that it should be kept below $30ms$ for percussive insturments. In our set-up the measured latencies are listed in the table 4.

	Controller	**Synthesis**	**Latency**
1	Voctro	test tone	30ms
2	WX5	Ssynth	110ms
3	Voctro	Ssynth	120ms

Table 4: In 1), the internal latency in Pd was set to 12ms. In 2) and 3), the internal latency in Pd was 50ms due to the required processing.

Other issues of the clarinet The clarinet has a wider pitch range than the common pitch range of the human voice. Usually, humans are capable of producing sounds in a pitch range of usually 2 octaves (table with different registers). In the typical clarinet model (B♭), the note range is of almost 4 octaves, going from E3($147Hz$) to C♯7 ($1976Hz$). Depending on the selected piece and the singer register, it will imply to transpose the analyzed singer pitch to a higher frequency. This opens another issue to be tackled in future work, which relates to find other parameters of the voice for changing the register of the synthesized instrument. This would be a similar control as in wind instrument

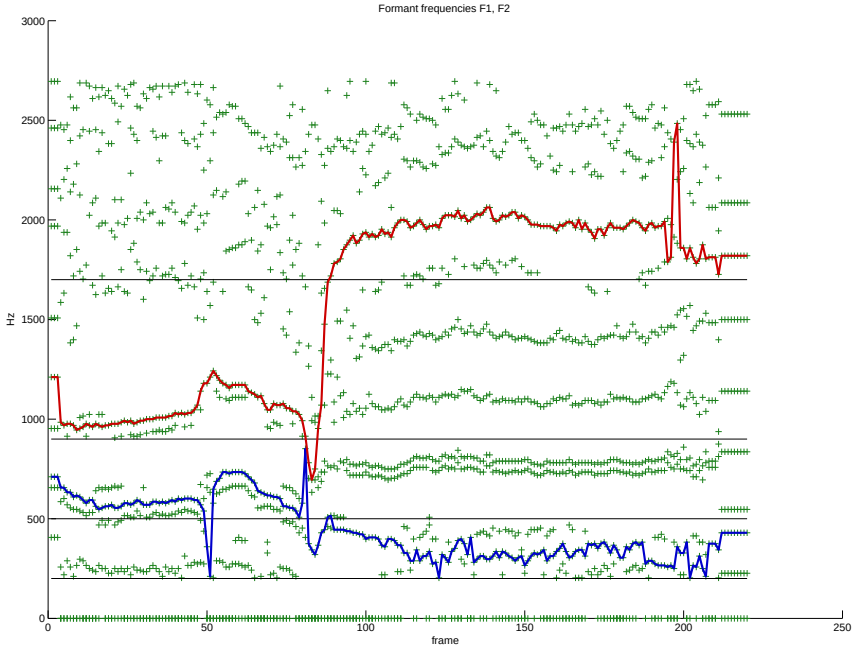

Figure 8: Real singing excerpt (oh-ah-eh). First two formant trajectories (solid lines) are estimated based on the frequency candidates found for each frame (crosses).

where a key allow a change of octave. Using phonetic information might be interesting but must be further studied.

Mapping Concerning the control of perceptual parameters of the synthesized sound, we ask the subjects to play the same melody with different mappings, allowing different degree of control of the synthesis parameters. The synthesizer input controls are high level parameters, also knwon as *abstract parameters* [13]. In this experiment we are concerned only with the first mapping layer, which is responsible for converting controller parameters into *abstract parameters*. In the case of the MIDI WindController, the control parameters are MIDI messages, with three type of data.In the case of the singing voice, the controller parameters are OSC messages at a rate of 86 fps and contain the parameters specified in the following table.

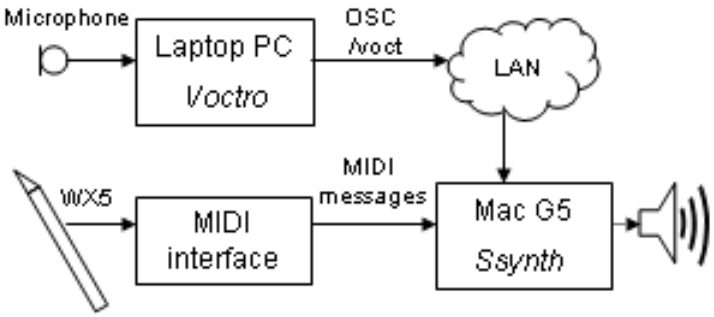

Figure 9: Set-up diagram. Microphone connected to a laptop sending OSC over a network. Windcontroller connected to a USB MIDI interface directly to the Mac G5 synthesizer computer.

Control Parameters	Control Parameters
WX5	Voctro
breath	energy
fingering	pitch
lip pressure	brightness

Abstract Parameters
Ssynth
Timbre dynamics
Loudness
Fundamental Frequency

Protocol The goal of this experiment is to evaluate the voice timbre, in this case the "vowel-space" for controlling a perceptual attribute of the sytnhesized sound (brightness). This task is *a priori* not intuitive for a singer, since it would tend to control the clarinet dynamics with the loudness, which relates to the actual acoustical behaviour of clarinet . Nevertheless, our goal is to exploit the voice expressive capabilities, not in a traditional context, but rather to control sound generation in a wider scope. In this sense, the singer has to learn this new context to perform tasks that he or she is not used to.

As we menrioned, this experiment is a first attempt to compare the voice witj existing musical controllers. In our case, a MIDI Windcontroller (Yamaha WX5). For the protocol, we defined eight tasksto accomplish by the participants. More specifically, we look at the control of a perceptual atirbute of the synthesizer, the *brightness*, which is related to the timbre dynamics of the clarinet. For each task the procedure consists in listening a reference musical phrase of a real clarinet recording. Then, the subject has to repeat the same phrase by focusing on the *brightness* of the synthesized sound.

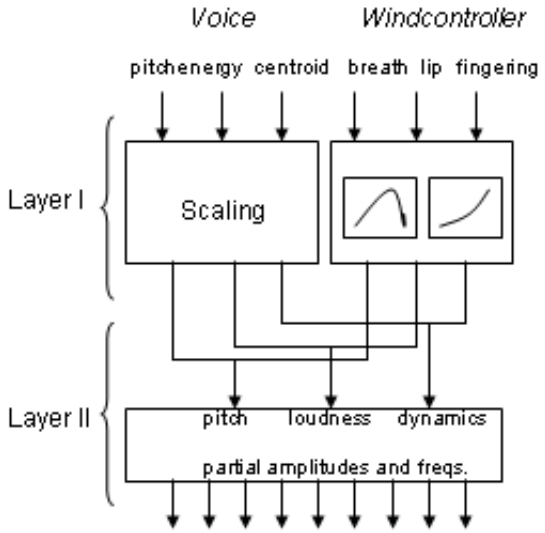

Figure 10: The diagram shows the different mapping layers. Both controllers share the synthesizer but the control signals follow different paths.

The participant could listen to the reference phrase multiple times before performing his task with one of the controllers, the voice or the windcontroller. In both cases, the participant listened to the synthesized sound through headphones. For singers, it reduces the sound level of his own voice and can better concentrate on the synthesized sound. In the table 5

6.1.2 Discussion

Here we provide some comments or ideas that arouse during or after the experiments. This helps us to understand the posed problems and to find new strategies for a better voice control of virtual insturments.

Control validity and conditioning As we noticed, the direct mapping between input and output fundamental frequency, sound level and brightness is not the ideal solution. The first reason is jitter on the fundamental frequency. The second reason is the attack quality. The third reason is the intuitive aspect of brightness control.

Pitch jitter The fundamental frequency in the singing voice presents a much higher jitter than the clarinet. It is due to the different acoustical characteristics of both sound generation system. The acoustical operation of the vocal folds is much complex than a bore. This jitter can be perceived as not natural in a clarinet sound. We aim at identifying pitch contour patterns that can be automatically mapped to the synthesizer. As observed in figures **??** and **??**, the estimated pitch signal is much more stable and flat in the clarinet than in the voice.

Task	**One note**
	insert score (D)!
1	crescendo
2	decrescendo
	Multiple notes
	insert score (D-D-D-D-D-D)!
3	pp-p-mp-mf-f-ff
4	ff-f-mf-mp-p-pp
	Alternate notes
	insert score (D-F-D-F-D-F)!
5	pp-p-mp-mf-f-ff
6	ff-f-mf-mp-p-pp
	Scale I
	insert score (SolM scale up)!
7	(vincent has the paper!)
	Scale II
	insert score (SolM scale down)!
8	

Table 5: The subjects were required to achieve eight tasks. Each task was a a sequence of notes with varying dynamics.

Amplitude Envelope and Attack Obviously, amplitude envelope is an important characteristic of any instrument timbre. For this experiment we decided to focus only on the timbre in the sustain part of the sound. However, the attack does not seem realistic for several reasons: the shape of the amplitude envelope (not smooth enough), and the fundamental frequency values during the first frames (the estimation is unstable during attack, resulting in big fundamental frequency variations). A solution consists in using a threshold on amplitude before starting a note, then using an expander to provide realistic smooth attack. By doing so, the instabilities of fundamental frequency happen mainly when the amplitude is under the threshold, and cannot be heard.

Brightness control Subect 2 mentioned that the 'vowel-space' doesn't seem to be appropiate for controlling the dynamics of wind instruments. However, This mapping is not intuitive as long as a singer tries to do exactly what he learned to do before, e.g., use vowel to sing words. But when the singer starts to really explore, he may be able to use vowels as a way to control brightness, as jazz singers do when scatting. Maybe our instructions are not clear enough, maybe our singers are not the 'good' subjects for this experiment). If this is the case, we can adopt two different positions for the future subjects:

1. changing the mapping to a more "intuitive", i.e. using the spectral slope as an estimation of the glottal excitation for controlling the dynamics

2. assuming that the mapping is not intuitive in order to evaluate the use of

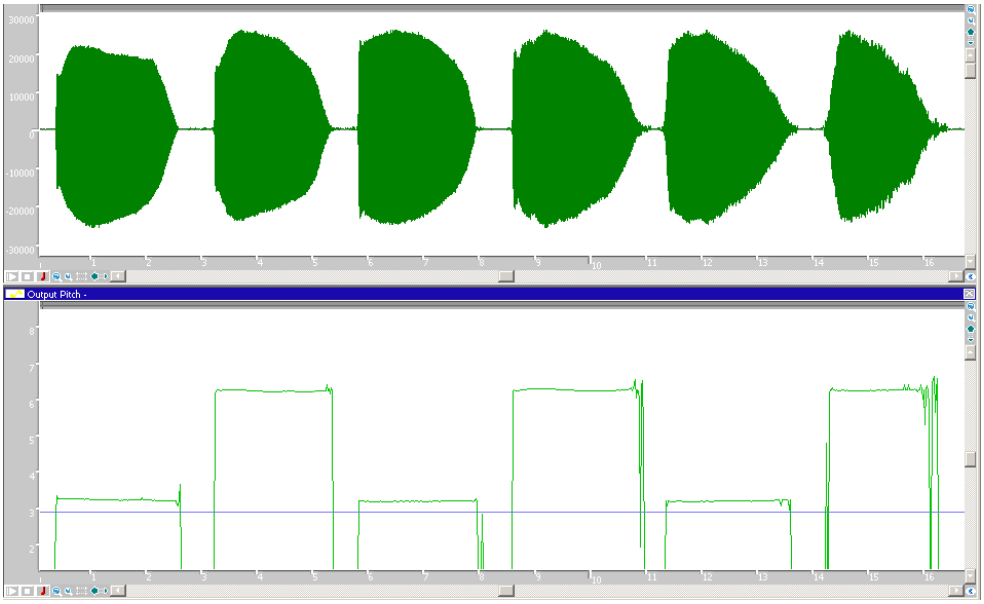

Figure 11: Estimated pitch of a real clarinet note sequence.

the "vowel-space" in a non-intuitive context/task.

The thing is that if we simplify too much the mapping or the task (for example asking the singer to sing louder so that, naturally, it sounds brighter), then we may limit the possibilities of control (e.g., how having a louder sound that is darker, if loudness and brightness controls are not independent?).

Considering that there are two steps when designing a digital music instrument (being able to play identically, ie interpolating, and the extrapolating), we think that we missed the first step! The DMI is made of sensors (here the voice analysis), a synthesizer and mappings. The mappings are not good enough yet.

Tuning the model and improvements

Tuning errors There was an error in the database: all notes were referred one octave below. Another problem occurs when the note in not perfectly in tune, and has overblows. The database was changed to correct both problems, by using the mean fundamental frequency of the sustain part.

Vocal range We should discuss the possibility of record different clarinet reference sounds for each voice register in order to have the singer singing in tune with the reference recording with similar effort.

Analytical methods In a further step, we need to discuss methods for evaluating quantitatively the sound results of the experiment. The sound examples recorded can be manually aligned in time without problem since they consist of separated notes. However, how can we compare those gestures (voice control of fundamental frequency, sound level, spectral centroid) between two performers,

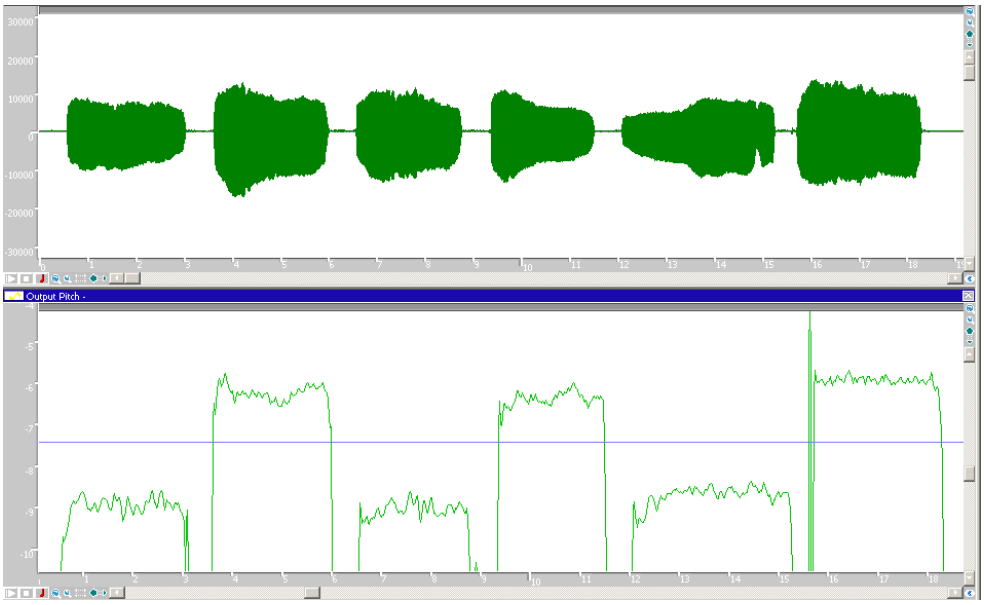

Figure 12: Estimated pitch of a singing voice note sequence.

especially when duration are different? If we time-warp the control signals, does it still makes sense? This issues constitute an important part of the future work. Several algorithms and techniques such as pattern recognition, or curve fitting can assist us in the analysis in order to find a proper mapping.

Notes on Subjects Here we reproduce some impressions of two subjects singers that participated in the experiment:

Subject 1 It was the first time we could run the experiment, and not all the tasks foreseen were completed. The subject was not confortable with the fact that the synthesizer was not in tune[5] with the pitch she was singing. This came from a bug in the database that was corrected afterwards. On the other hand, the analyzed pitch suffers from jittering that affects negatively to the synthesized sound. It makes difficults the identification of a clarinet sound. We proposed to introduce a LPF of the pitch control signal for future experiments as a first solution. Also, we will record the voice when singing along the clarinet recordings in order to have a time-matched pitch curves to work with. The goal will be to identify specific patterns in the voice that can be mapped to the synthesis pitch for the clarinet.

During the training part, it was hard for Irene to understand the actual mapping of the system, i.e. to identify the voice parameters that were responsible for controlling the synthesis brightness. This lead us to explain in advance the vowel-space control for future sessions. She was open and happy to experiment with the system, even when she thought that the problems in the final results were motivated for her singing (!).

The duration of the session was 90 minutes.

[5]octave error and slight tuning error. See next part for explanations.

Subject 2 After the indication of Subject 1, we tuned the system and solved the problem with the tuning. However, we left the pitch control signal without any LPF in order to have another opinion on this issue. Dave actually didn't mention problems pitch jittering during the experiment.

In this session, we complete all the eight required tasks. We explained him how the system was controlled and what the objective was. During the training part, Dave was able to control relatively the brightness but not in a gradual manner. This can have its origin basically in the centroid algorithm, since it gives a rapid varying signal since it uses only the harmonic peaks instead of the actual spectral bins. He divided the dynamics (brightness) parameter in three regions, one controlled by the sound /u/, the second by the sound /a/ and the third by the sound /i/. It allowed him ton control the brightness consistently.

The principal comment of Dave was that the mapping was absolutely not intuitive for a singer. He meant, that "in singing the vowels are part of the linguistics and not of the music". It makes us to reconsiderate seriously to use the vowel space for timbre control. He explained that it would seem a lot more obvious and intuitive to control the clarinet dynamics with the "vocal effort". In terms of analysis parameters, this corresponds mainly to the glottis excitation level and probably a bit to the mouth opening. A combination of spectral slope and centroid might be a good descriptor.The duration of the session was 40 minutes.

Conclusions We addressed the voice control of a clarinet synthesizer input parameters. More precisely, the control of the timbre dynamics of the sound, ranging from a *pp* to a *ff*. Here, only the sustain part of a note was considered. Future research will tackle other timbre aspects such as amplitude envelope. The clarinet synthesizer is an additive model with a database containing multiple notes and dynamics. The case study compares two controllers: a Voice Controller and a MIDI Windcontroller. In the experiment, subjects (singers and saxophonists, respectively) had to match the dynamics of a pre-recorded reference sequence.

- For singers, using the centroid was not "intuitive" as a timbre dynamics control. They had to learn it.
- For saxophonists, timbre dynamics was controlled mainly by the lip pressure and not really coupled with the breath pressure. This was not "intuitive" for the performers.
- The pitch jitter present in the voice signal makes difficult to perceive the synthesized sound as clarinet. Post-process of the analyzed pitch signal should be considered.

The centroid is not an ideal voice parameter for synthesis control, since it depends at the same time on the timbre (vowel) as well as on the excitation slope, which is related to voice level. Better control could be achieved by using Formant Frequencies as control parameter which is independent of the voice level. For the future work, we foresee two different analysis strategies:

- Quantitiative: DSP methods.
- Qualitative: Web-based survey.

Figure 13: Recording session with a bass guitar performer.

6.2 Bass guitar synthesis: VoctroBass

Previous research of voice-controlled bass guitar synthesis has been already reported [5]. Opposite to the case study of the clarinet synthesis preented in the previous section, we don't aim to compare two controllers. Rather, here we strive to record a bass guitar performer playing musical phrases while singing the same phrases. The objective is to identify *expressive gestures* in both signals, bass guitar and voice. In a second step, we will use voice analysis techniques to extract these *expressive gestures* and then, to map them to the bass guitar synthesizer. In this category, we include type of attack, intonation, pitch and amplitude modulations, etc. In addition we conect the microphone the the voice-controlled synthesizer, so that in real-time we can listen to the real bass guitar and the synthesized one. Note that the performer only listens to his own voice through headphones.
The set-up consists of a Digital Audio Workstation that records three audio channels: bass guitar line output, voice signal and synthsized bass guitar. The voice-controlled bass guitar synthsizer runs on a laptiop PC, and a Mac PowerPC G5 serves as DAW.
Many thanks to Andreas Bergsland who participated in the recording sessions.

6.3 Visual Feedback: VoctroVisuals

(This research is a collaboration with Thor Magnusson [6])

The third case study reported here deals, not with sound synthesis but with visual representation of voice signals. Obviously, this is a more experimental approach. Nevertheless, it can unveil at the same time new paradigms for new musical application and new interfaces for musical expression. The voice appears as a very attractive solution as input device, specially for performance situation, where the audience needs to understand what the performer is doing. We developed a first prototype, where incoming voice phrases are drawn on a blank screen. When a phrase stops, the screen is cleared. This system consists of two independent modules: voice analysis and Graphical interface. The Voctro application described previously in this report is used to extract the voice parameters and to send them to the graphical module via Open Sound Control

[6] `http://www.ixi-software.net`

(OSC). The transmission protocol used by OSC is UDP, thus, both modules can run on different computers. The graphical module is a Java application that draws an element (square o circle) each time a new OSC message is received from the analysis module. The actual frame rate is specified by the video rate, typically $24fps$.

Although it is a very early prototype and a lot of improvements have to be done, we believe this opens a wide range of applications in the artistic, educational and human interaction field. In the figure Fig. 14, we depicted a screenshot of the application. The drawn objects have a certain size depending on the voice signal energy, and the position is specified by the analyzed voice pitch (Y)and the spectral centroid (X).

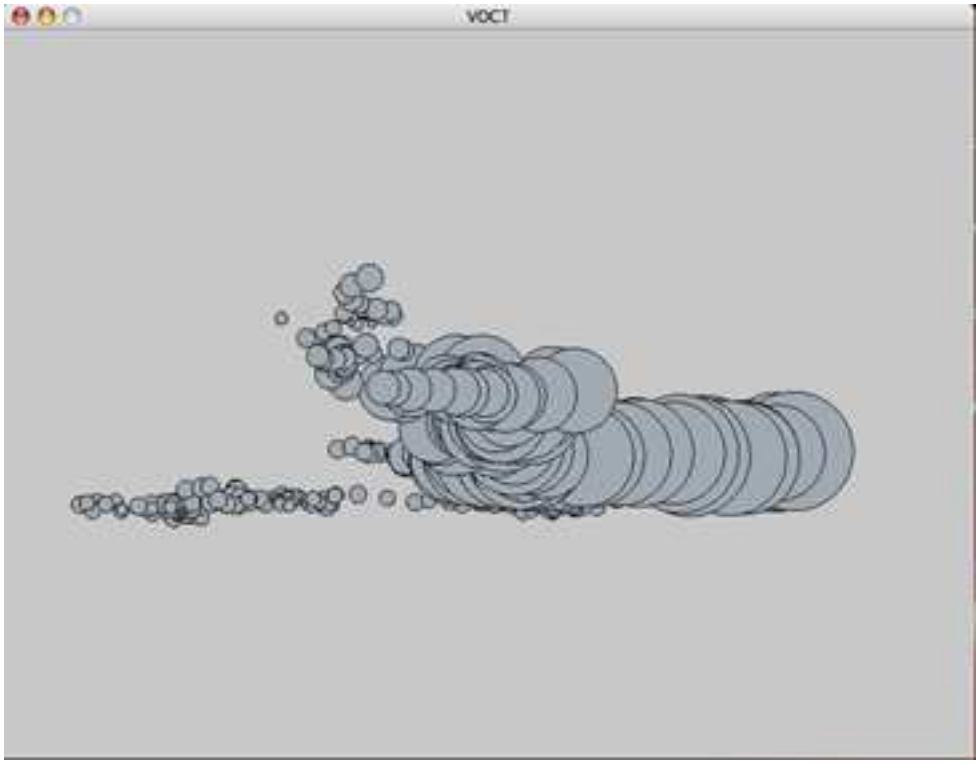

Figure 14: Visual representation of a vocal segment

7 Acknowledgments

Many thanks to singers, and clarinet and saxophone players from the Schulich School of Music, McGill University, that participated to the tests. Also to the researchers and students at the Music Technolgy Area, McGill University, specially to Vincent Verfaille and Philippe Depalle.

References

[1] P. Cano. Fundamental frequency estimation in the SMS analysis. In *Proceedings of COST G6 Conference on Digital Audio Effects 1998*, Barcelona, 1998.

[2] G. Fant. *Acoustic Theory of Speech Production.* Gravenhage, Mouton, 1960.

[3] T. Galas and X. Rodet. An improved cepstral method for deconvolution of source-filter systems with discrete spectra: Application to musical sounds. In *Proc. Int. Comp. Music Conf. (ICMC'90), Glasgow*, pages 82–8, 1990.

[4] J. Janer. Feature extraction for voice-driven synthesis. In *118th Conv. Audio Eng. Soc., Barcelona*, Barcelona, 2005.

[5] J. Janer. Voice-controlled plucked bass guitar through two synthesis techniques. In *Int. Conf. on New Interfaces for Musical Expression, Vancouver*, pages 132–135, Vancouver, Canada, 2005.

[6] T. Jehan and B. Schoner. An audio-driven perceptually meaningful timbre synthesizer. In *Proc. Int. Comp. Music Conf. (ICMC'01), Havana*, 2001.

[7] J. L. O. Cappé and E. Moulines. Regularized estimation of cepstrum envelope from discrete frequency points. In *Proc. IEEE Workshop on Applications of Digital Signal Processing to Audio and Acoustics*, 1995.

[8] G. Peeters. A large set of audio features for sound description (similarity do classification) in the cuidado project. In *CUIDADO IST Project Report*, 2004.

[9] M. Puckette. Score folowing using the sung voice. In *Proceedings, International Computer Music Conference*, San Francisco, 1995.

[10] M. Puckette. Low-dimensional parameter mapping using spectral envelopes. In *Proceedings, International Computer Music Conference*, Miami, 2004.

[11] D. Schwarz and X. Rodet. Spectral envelope estimation and representation for sound analysis-synthesis. In *Proc. Int. Comp. Music Conf. (ICMC'99), Beijing*, Beijing, 1999.

[12] X. Serra and J. O. Smith. A sound decomposition system based on a deterministic plus residual model. *J. Acoust. Soc. Am., sup. 1*, 89(1):425–34, 1990.

[13] M. Wanderley. *Intéraction Musicien-Instrument : application au contrôle gestuel de la synthèse sonore.* PhD thesis, Université Paris VI, IRCAM, 2001.