Synthesis of the Singing Voice by Performance Sampling and Spectral Models

Jordi Bonada and Xavier Serra

Pompeu Fabra University, Music Technology Group, Ocata, 1, 08003 Barcelona, Spain

E-mail: {jordi.bonada,xavier.serra}@iua.upf.edu

1. Introduction

Among the many existing approaches to the synthesis of musical sounds, the ones that have had the biggest success are without any doubt the sampling based ones, which sequentially concatenate samples from a corpus database [1]. Strictly speaking, we could say that sampling is not a synthesis technique, but from a practical perspective it is convenient to treat it as such. From what we explain in this article it should become clear that, from a technology point of view, it is also adequate to include sampling as a type of sound synthesis model.

The success of sampling relies on the simplicity of the approach, it just samples existing sounds, but most importantly it succeeds in capturing the naturalness of the sounds, since the sounds are real sounds.

However, sound synthesis is far from being a solved problem and sampling is far from being an ideal approach. The lack of flexibility and expressivity are two of the main problems, and there are still many issues to be worked on if we want to reach the level of quality that a professional musician expects to have in a musical instrument.

Sampling based techniques have been used to reproduce practically all types of sounds and basically have been used to model the sound space of all musical instruments. They have been particularly successful for instruments that have discrete excitation controls, such as percussion or keyboard instruments. For these instruments it is feasible to reach an acceptable level of quality by using large sample databases, thus by sampling a sufficient portion of the sound space produced by a given instrument. This is much more difficult for the case of continuously excited instruments, such as bowed strings, wind instruments or the singing voice, and therefore recent sampling based systems consider a trade-off between performance modeling and sample reproduction (e.g. [2]). For these instruments there are numerous control parameters and many ways to attack, articulate or play each note. The control parameters are constantly changing and the sonic space covered by a performer could be considered to be much larger than for the discretely excited instruments.

The synthesis approaches based on physical models have the advantage of having the right parameterization for being controlled like a real instrument, thus they have great flexibility and the potential to play expressively. One of the main open problems relates to the control of these models, in particular how to generate the physical actions that excite the instrument. In sampling these actions are embedded in the recorded sounds.

We have worked on the synthesis of the singing voice for many years now mostly together with Yamaha Corp., part of our results having being incorporated into the Vocaloid[1] software synthesizer. Our goal has always been to develop synthesis engines that could sound as natural and expressive as a real singer (or choir [3]) and whose inputs could be just the score and the lyrics of the song. This is a very difficult goal and there is still a lot of work to be done, but we believe that our proposed approach can reach that goal. In this paper we will overview the key aspects of the technologies developed so far and identify the open issues that still need to be tackled. The core of the technologies is based on spectral processing and over the years we have added performance actions and physical constraints in order to convert the basic sampling approach to a more flexible and expressive technology while maintaining its inherent naturalness.

In the first part of the article we introduce the concept of synthesis based on performance sampling and the specific spectral models that we have developed and used for the singing voice. In the second part we go over the different components of the synthesizer and we conclude by identifying the open issues of this research work.

2. Performance based Sampling synthesis

Sampling has always been considered a way to capture and reproduce the sound of an instrument but in fact it should be better considered a way to model the sonic space produced by a performer with an instrument. This is not just a fine distinction; it is a significant conceptual shift of the goal to be achieved.

We want to model the sonic space of a performer/instrument combination. This does not mean that the synthesizer shouldn't be controlled by a performer, it just means that we want to be flexible in the choice of input controllers and be able to use high-level controls, such as a traditional music score, or to include lower-level controls if they are available. Thus taking advantage from a smearing of the traditional separation between performer and instrument.

Figure 1 shows a visual representation of a given sonic space to be modeled. The space A represents the sounds that a given instrument can produce by any means. The space B is the subset of the space A that a given performer can produce by playing that instrument. The trajectories shown in the space B represent the actual recordings that have been sampled. The reality is that this sonic space is an infinite multidimensional one but we hope to be able to get away by approximating it with a finite space. The trajectories represent paths in this multidimensional space. The size of these spaces may vary depending on the perceptually

[1] http://www.vocaloid.com

relevant degrees of freedom in a given instrument/performer combination, thus we could say that a performed drum can be represented by a smaller space than a performed violin. This is a very different concept than the traditional timbre space; here the space is defined both by the sound itself and by the control exerted on the instrument by the performer. Thus the sound space of an accomplished performer would be bigger than the space of a not so skilled one.

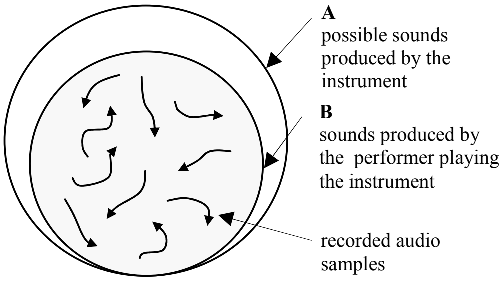
A
possible sounds produced by the instrument
B
sounds produced by the performer playing the instrument
recorded audio samples

Figure 1 Instrument sonic space

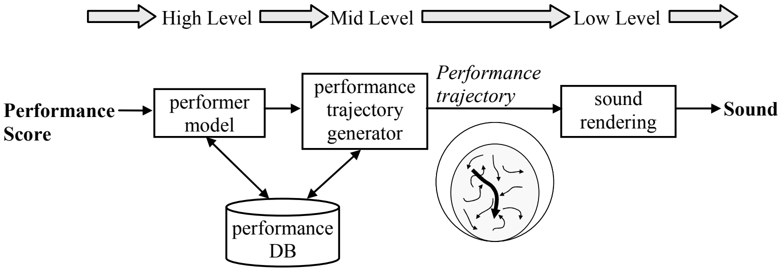

Figure 2 Block diagram of the proposed synthesizer

From a given sampled sonic space and the appropriate input controls, the synthesis engine should be able to generate any trajectory within the space, thus producing any sound contained in it. The brute force approach is to do an extensive sampling of the space and perform simple interpolations to move around it. In the case of the singing voice, the space is so huge and complex that this approach is far from being able to cover a large enough portion of the space. Thus the singing voice is a clear example that the basic sampling approach is not adequate and that a parameterization of the sounds is required. We have to understand the relevant dimensions of the space and we need to find a sound parameterization with which we can move around these dimensions by interpolating or transforming the existing samples.

Figure 2 shows a block diagram of our proposed synthesizer. The input of the system is a generalization of the traditional score, a *Performance Score*, which can include any symbolic information that might be required for controlling the synthesizer. The *Performer Model* converts the input controls into lower level performance actions, the *Performance Trajectory Generator* creates the parameter trajectories that express the appropriate paths to move within the sonic space, and the *Sound Rendering* module is the actual synthesis engine that produces the output sound by concatenating a sequence of transformed samples which approximate the performance trajectory. The *Performance Database* includes not only the performance samples but also models and measurements that relate to the performance space and that give relevant information to help in the process of going from the high level score representation to the output sound.

In the next sections we will present in more detail each of these components for the specific case of a singing voice, but considering that most of the ideas and technologies could be applied to any performer/instrument combination.

3. Modeling the singing voice

As stated by Sundberg [4], we consider the singing voice as the sounds produced by the voice organ and arranged in adequate musical sounding sequences. The voice organ encloses the different structures that we mobilize when we produce voice: the breathing system, the vocal folds, and the vocal and nasal tracts. More specifically, as shown in Figure 3, voice sounds originate from an airstream from the lungs which is processed by the vocal folds and then modified by the pharynx, the mouth and nose cavities. The sound produced by the vocal folds is called the Voice Source. When the vocal folds vibrate, the airflow is chopped into a set of pulses producing voiced sounds (i.e. harmonic sounds). Otherwise, different parts of the voice organ can work as oscillators to create unvoiced sounds. For example, in whispering, vocal folds are too much tense to vibrate but they form a narrow passage which makes the airstream become turbulent and generate noise.

The vocal tract acts as a resonator and shapes acoustically the voice source, especially enhancing certain frequencies called formants[2] (i.e. resonances). The five lowest formants are the most significant ones for vowel quality and voice color. It is important to note that the vocal tract cannot be considered a linear-phase-response filter. Instead, each formant decreases the phase around its center frequency, as can be seen in Figure 4. This property is perceptually relevant, especially for middle and low pitch utterances.

In a broad sense, and according to whether the focus is put on the system or its output, synthesis models can be classified into two main groups: spectral models and physical models. Spectral models are mainly based on perceptual mechanisms of the listener while physical models focus on modeling the production mechanisms of the sound source. Any of these two models are suitable depending on the specific requirements of the application or they might be combined for taking advantages of both approaches. Historical and in-depth overviews of singing voice synthesis models are found in [5][6][7].

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 must be brought on the design. The model itself can provide intuitive parameters if it is constructed so that it sufficiently matches the physical system. Conversely, such a system usually has a large number of parameters and the mapping of those quite intuitive controls of the production mechanism to the final output of the model, and so to the listener's perceived quality, is not a trivial task. The controls would be related to the movements of the vocal apparatus elements such as jaw opening, tongue shape, sub-glottal

[2] There also exist the antiformants or antiresonances, i.e. frequency regions in which the amplitudes of the voice source are attenuated. These are especially present in nasal sounds because nasal cavities absorb energy from the sound wave.

air pressure, tensions of the vocal folds, etc. The first digital physical model of the voice was based on simulating the vocal tract as a series of one-dimensional tubes [8], afterwards extended by means of digital waveguide synthesis [9] in SPASM [10]. Physical models are evolving fast and becoming more and more sophisticated, 2-D models are common nowadays providing increased control and realism [11][12], and the physical configuration of the different organs during voice production is being estimated with great detail combining different approaches. For example, 2D vocal tract shapes can be estimated from Functional magnetic resonance imaging (fMRI) [13], X-ray computed tomography (CT) [14], or even audio recordings and EGG signals by means of genetic algorithms [15].

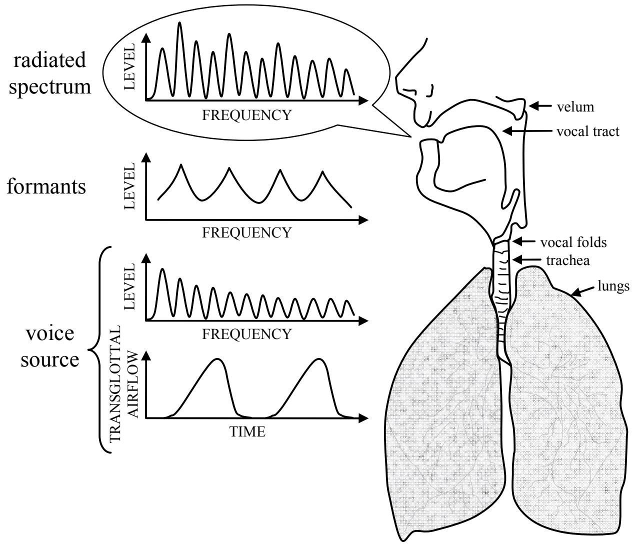

Figure 3 The voice organ

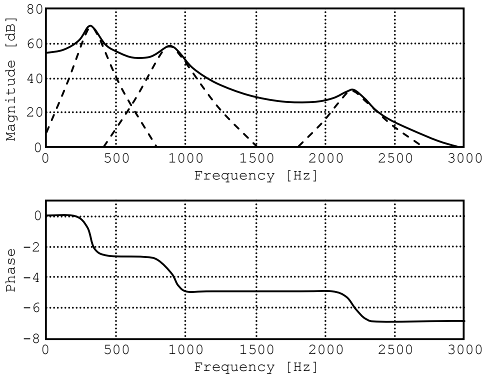

Figure 4 Vocal tract transfer function.

Alternatively, spectral models are 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. Typical controls would be pitch, glottal pulse spectral shape, formant frequencies, formant bandwidths, etc. Of particular relevance is the sinusoidal based system in [16].

A typical example of combining both approaches is the one of the formant synthesizers (e.g. [17], CHANT system [18]), considered to be pseudo-physical models because even though they are mainly spectral models they make use of the source / filter decomposition which considers voice to be the result of a glottal excitation waveform (i.e. the voice source) filtered by a linear filter (i.e. the vocal tract). The voice model we present here would be part of this group.

3.1 EpR Voice Model

The EpR[3] Voice Model [19] is based on an extension of the well known source/filter approach [20]. It models the magnitude spectral envelope defined by the harmonic spectral peaks of the singer's spectrum. It consists of three filters in cascade (see Figure 5). The first filter models the voice source frequency response with an exponential curve plus one resonance. The second one models the vocal tract with a vector of resonances which emulate the voice formants. The last filter stores the amplitude differences between the two previous filters and the harmonic envelope. Hence, EpR can perfectly reproduce all the nuances of the harmonic envelope.

Each of the parameters of this model can be controlled independently. However, whenever a formant frequency is modified, the residual filter envelope is scaled taking as anchor points the formant frequencies, thus preserving the local spectral amplitude shape around each formant.

3.2 Audio Processing

Since our system is a sample based synthesizer in which samples of a singer database are transformed and concatenated along time to compose the resulting audio, high quality voice transformation techniques are a crucial issue. Thus we need audio processing techniques especially adapted to the particular characteristics of the singing voice, using EpR to model voice timbres, and preserving the relationship between formants and phase.

SMS

We initially used SMS[4] [21] as the basic transformation technique [19]. SMS had the advantage of decomposing the voice into harmonics and residual, respectively modeled as sinusoids and filtered white noise. Both components were independently transformed, so the system yielded a great flexibility. But although the results were quite encouraging in voiced sustained parts, in transitory parts and consonants, especially in voiced fricatives, harmonic and residual components were not perceived as one, and moreover transients were significantly smeared. When modifying harmonic frequencies, for example in a transposition transformation, harmonic phases were computed according to the phase rotation of ideal sinusoids at the estimated frequencies. Hence, the strong phase synchronization (or phase-coherence) between the various harmonics resulting of the formant and phase relationship was not preserved, producing phasiness[5] artifacts, audible especially at low pitch utterances.

[3] EpR stands for Excitation plus Resonances

[4] SMS stands for Spectral Modeling Synthesis.

[5] A lack of presence, a slight reverberant quality, as if recorded in a small room

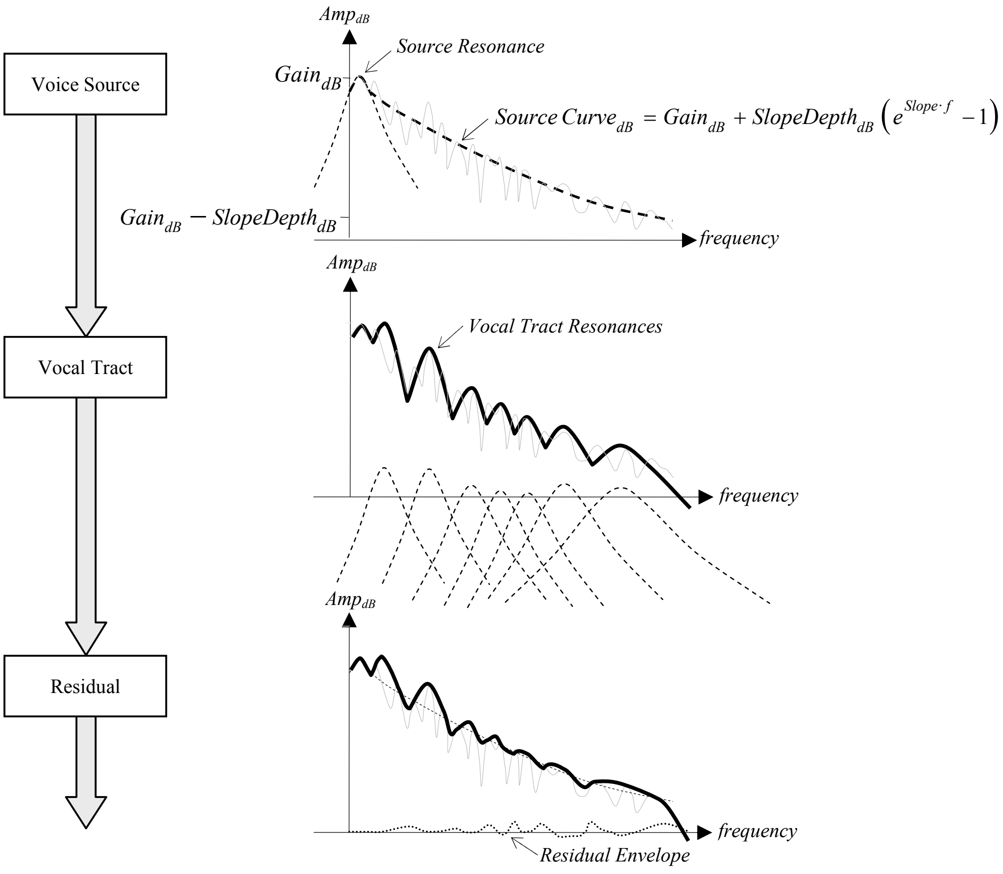

Figure 5 EpR Voice Model

Phase-Locked Vocoder

Intending to improve our results, we moved to a spectral technique based on the phase-locked vocoder [22] where the spectrum is segmented into regions, each of which contains a harmonic spectral peak and its surroundings. In fact, in this technique each region is actually represented and controlled by the harmonic spectral peak, so that most transformations basically deal with harmonics and compute how their parameters (amplitude, frequency and phase) are modified, in a similar way to sinusoidal models. These modifications are applied uniformly to all the spectral bins within a region, therefore preserving the shape of the spectrum around each harmonic. Besides, a mapping is defined between input and output harmonics in order to avoid shifting in frequency the aspirated noise components and introducing artifacts when low level peaks are amplified. We improved the harmonic phase continuation method by assuming a perfectly harmonic frequency distribution [23]. The result was that when pitch was modified (i.e. transposition), the unwrapped phase envelope was actually scaled according to the transposition factor. The sound quality was improved in terms of phasiness but not sufficiently, and the relation between formants and phase was not really preserved.

Several algorithms have been proposed regarding the harmonic phase-coherence for both phase-vocoder and sinusoidal modeling (for example [24] and [25]), most based on the idea of defining pitch-synchronous input and output onset times and reproducing at the output onset times the phase relationship existing in the original signal at the input onset times. However, the results are not good enough because the onset times are not synchronized to the voice pulse onsets, but assigned to an arbitrary position within the pulse period. This causes unexpected phase alignments at voice pulse onsets which doesn't reproduce the formant to phase relations, adding an unnatural 'roughness' characteristic to the timbre (see Figure 6). Different algorithms detect voice pulse onsets relying on the minimal phase characteristics of the voice source (i.e. glottal signal) (e.g. [26][27]). Following this same idea we proposed in [28] a method to estimate the voice pulse onsets out of the harmonic phases based on the property that when the analysis window is properly centered, the unwrapped phase envelope is nearly flat with shifts under each formant, thus being close to a maximally flat phase alignment (MFPA) condition. By means of this technique, both phasiness and roughness can be greatly reduced to be almost inaudible.

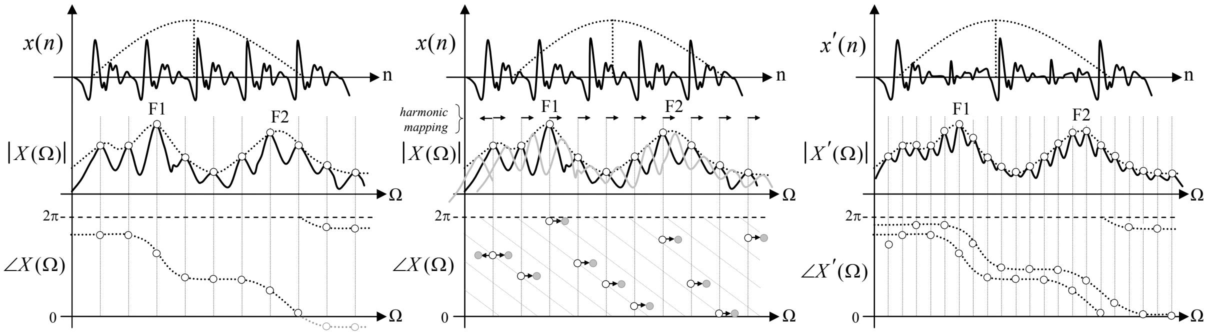

Figure 6 Spectrums obtained when the window is centered at the voice pulse onset (left figure), and between two pulse onsets (middle figure). In the middle figure harmonics are mapped to perform one octave down transposition. In the right figure, spectrum of the transformed signal with the window centered at the voice pulse onset. The 'doubled' phase alignment adds an undesired 'roughness' characteristic to the voice signal. Besides, we don't see only one voice pulse per period as expected, but two with strong amplitude modulation.

VPM

Both SMS and phase-vocoder techniques are based on modifying the frequency domain characteristics of the voice samples. Several transformations can be done with ease and with good results, for instance transposition and timbre modification[6]. However, certain transformations are rather difficult to achieve, especially the ones related to irregularities in the voice pulse sequence, which imply to add and control subharmonics [29]. Irregularities in the voice pulse sequence are inherent to roughness or creaky voices and appear frequently in singing, sometimes even as an expressive recourse such as growl.

[6] Here timbre modification is understood as modification of the envelope defined by the harmonic spectral peaks.

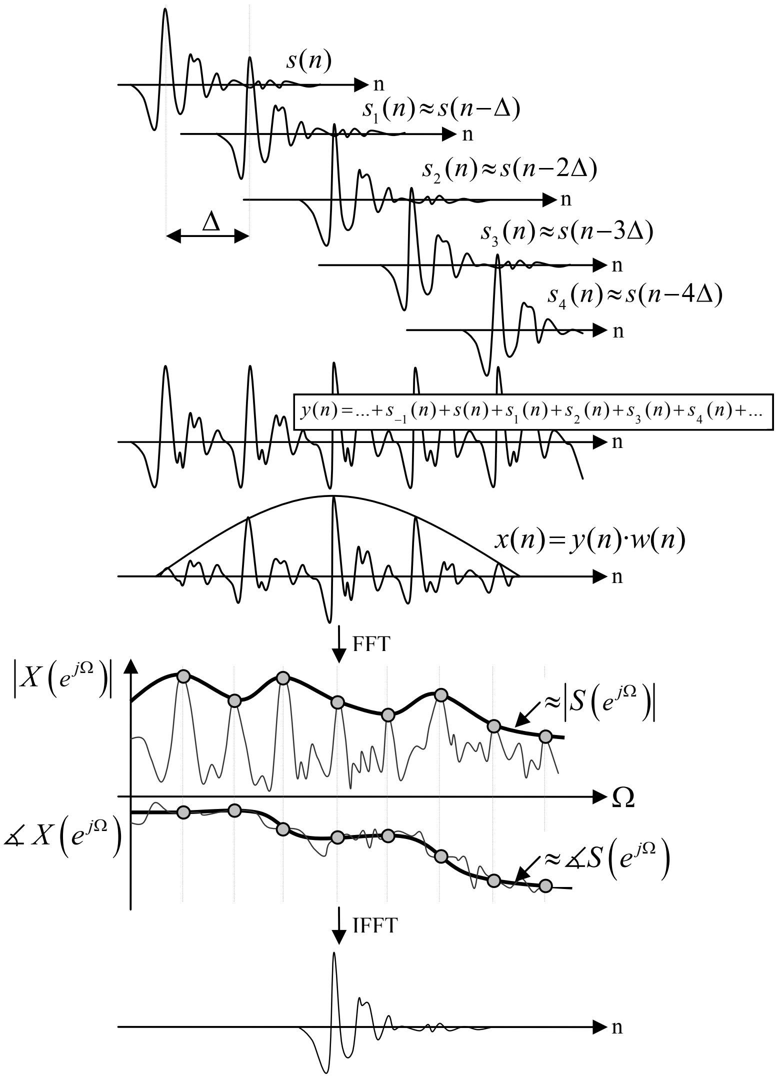
$s(n)$
n
$s_1(n)\approx s(n-\Delta)$
n
$s_2(n)\approx s(n-2\Delta)$
n
Δ
$s_3(n)\approx s(n-3\Delta)$
n
$s_4(n)\approx s(n-4\Delta)$
n
$y(n)=\ldots+s_{-1}(n)+s(n)+s_1(n)+s_2(n)+s_3(n)+s_4(n)+\ldots$
n
$x(n)=y(n)\cdot w(n)$
n
FFT
$\left|X\left(e^{j\Omega}\right)\right|$
$\approx\left|S\left(e^{j\Omega}\right)\right|$
Ω
$\measuredangle X\left(e^{j\Omega}\right)$
$\approx\measuredangle S\left(e^{j\Omega}\right)$
IFFT
n

Figure 7 Voice Pulse Modeling: the spectrum of a single filtered voice pulse can be approximated as the spectrum obtained by polar interpolation of harmonic peaks when the analysis window is centered on a voice pulse onset.

Dealing with this issue we proposed in [28] an audio processing technique especially adapted to the voice, based on modeling the radiated voice pulses in frequency domain. We call it VPM which stands for Voice Pulse Modeling. We have seen before that voiced utterances can be approximated as a sequence of voice pulses linearly filtered by the vocal tract. Hence the output can be considered as the result of overlapping a sequence of filtered voice pulses. It can be shown that the spectrum of a single filtered voice pulse can be approximated as the spectrum obtained by interpolation of the harmonic peaks (in polar coordinates, i.e. magnitude and phase), when the analysis window is centered on a voice pulse onset (see Figure 7 and Figure 9). The same applies if the timbre is modeled by the EpR model exposed in §3.1. Since the phase of the harmonic peaks is interpolated, special care must be taken regarding phase unwrapping, in order to avoid discontinuities when different integer numbers of 2π periods are added to a certain harmonic in consecutive frames (see Figure 8). Finally, once we have a single filtered voice pulse we can reconstruct the voice utterance by overlapping several of these filtered voice pulses arranged as in the original sequence. This way, introducing irregularities to the synthesized voice pulse sequence becomes straightforward.

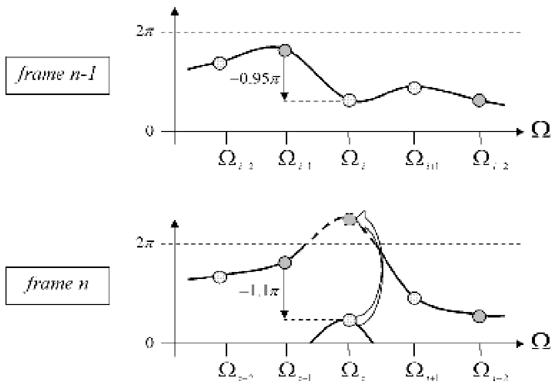

Figure 8 Phase unwrapping problem: different numbers of 2π periods are added to the ith harmonic in consecutive frames.

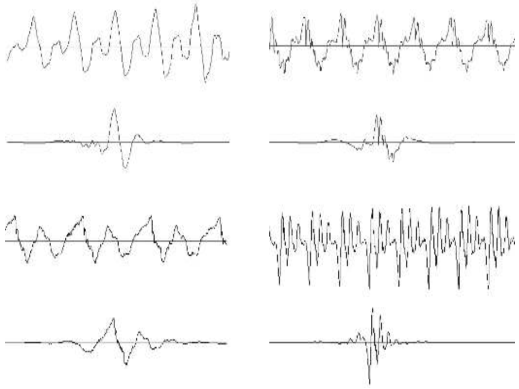

Figure 9 Synthesis of single pulses from the analysis of recorded voiced utterances.

Clearly we have disregarded all the data contained in the spectrum bins between harmonic frequencies, which contain noisy or breathy characteristics of the voice. Like in SMS, we can obtain a residual by subtracting the VPM resynthesis to the input signal, therefore ensuring a perfect reconstruction if no transformations are applied. It is well known that the aspirated noise produced during voiced utterances has a time structure, which is perceptually important, and is said to be correlated to the glottal voice source phase [30]. Hence, in order to preserve this time structure, the residual is synthesized using a PSOLA method synchronized to the synthesis voice pulse onsets. Finally, for processing transient-like sounds we adopted the method in [31] which by integrating the spectral phase is able to robustly detect transients and discriminate which spectral peaks contribute to them, therefore allowing translating transient components to new time instants.

VPM is similar to old techniques such as FOF [32] and VOSIM [33], where voice is modeled as a sequence of pulses whose timbre is roughly represented by a set of ideal resonances. However, in VPM the timbre is represented by all the harmonics, allowing capturing subtle details and nuances of both amplitude and phase spectra. In terms of timbre representation we could obtain similar results with spectral smoothing by applying restrictions to the poles and zeros estimation in AR, ARMA or PRONY models. However, with VPM we have the advantage of being able to smoothly interpolate different voice pulses avoiding problems due to phase unwrapping, and decomposing the voice into three components (harmonics, noise and transients) which can be independently modified. A block diagram of VPM analysis and synthesis processes is shown in Figure 10.

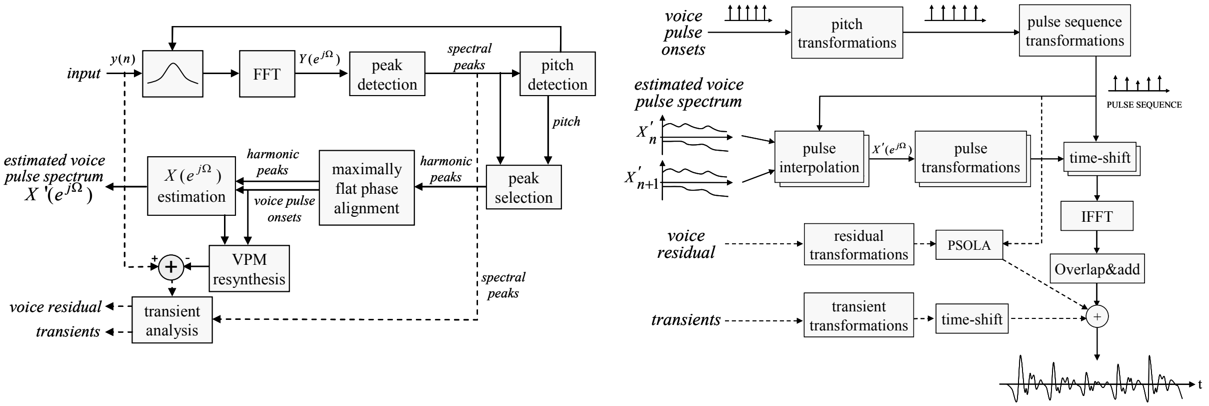

Figure 10 VPM analysis and synthesis block diagram

4. Performer Model

From the input score, the Performer Model is in charge of generating lower level actions, thus it is responsible of incorporating performance specific knowledge to the system. Some scores might include a fair number of performance indications and in these cases the Performer Model would have an easier task. However we are far from completely understanding and being able to simulate the music performance process and therefore this is still one of the most open ended problems in music sound synthesis. The issues involved are very diverse, going from music theory to cognition and motor control problems, and current approaches to performance modeling only give partial answers. The most successful practical approaches have been based on developing performance rules by an analysis by synthesis approach [34][35] and more recently machine learning techniques have been used for generating these performance rules automatically [36]. Another useful method is based on using music notation software which allows interactive playing and adjustment of the synthetic performance [37].

For the particular case of the singing voice system presented here the main characteristics which are determined by the Performer Model are the tempo, the deviation of note duration from the standard value, the vibrato occurrences and characteristics, the loudness of each note, and how notes are attacked, connected and released in terms of musical articulation (e.g. legato, staccato, etc.). It is beyond the scope of this research to address higher levels of performance control such as the different levels of phrasing and their influence in all the expressive nuances.

5. Performance Database

In this section, we explain how to define the sampling grid of the instrument's sonic space and the steps required to build a database of it. First of all we must identify the dimensions of the sonic space we want to play with and the transformations we can do with the samples. For each transformation we need to study as well the tradeoff between sound quality and transformation range, preferably in different contexts (*for*

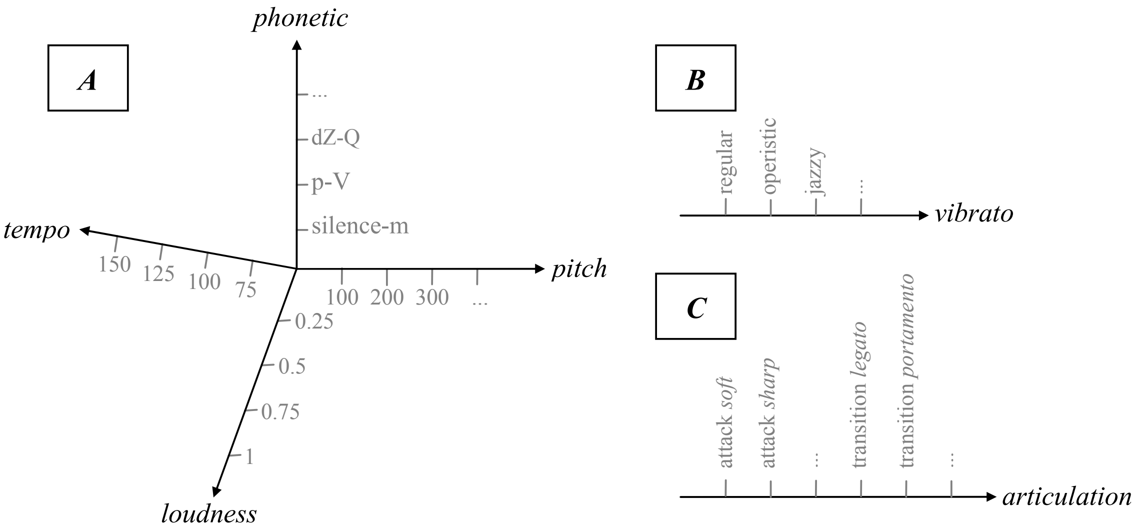

Figure 11 Subspaces of the singing voice sonic space.

example, we can think of a case where transpositions sound acceptable for the interval [-3,+4] semitones at low pitches, but the interval becomes [-2,+2] at high pitches). Besides we should also consider the sample concatenation techniques to be used in synthesis, and the consequent tradeoff between sample distance and natural sounding transitions. Once we have set the sampling grid to be a good compromise for the previous tradeoffs, we have to come up with detailed scripts for recording each sample, trying to minimize the score length thus maximize the density of database samples in the performance being recorded. These scripts should be as detailed as possible in order to avoid ambiguities and assure that the performance will contain the target samples.

In the case of singing voice we decided to divide the sonic space into three subspaces *A*, *B* and *C,* each with different dimensions (see Figure 11). Subspace *A* contains the actual samples which are transformed and concatenated at synthesis. Instead, subspaces *B* and *C* contain samples which once properly modeled specify how samples from *A* should be transformed to show a variety of specific expressions.

In subspace *A*, the phonetic axis has a discrete scale set as units of two allophones combinations (i.e. di-llophones) plus steady-states of voiced allophones. Using combination of two or more allophones is in fact a common practice in concatenative speech synthesis (e.g. [38]). However, not all di-allophones combinations must be sampled but only a subset which statistically covers most frequent combinations. For example, Spanish language has 29 possible allophones [39] and, after analyzing the phonetic transcription of several books containing more than 300000 words, we found out that 521 combinations out of the 841 theoretically possible ones were enough to cover more than 99.9% of all occurrences. The pitch axis must be adapted to the specific range of the singer. In our case, the sound quality seems to be acceptable for transpositions up to +/- 6 semitones, and the sampling grid is set accordingly, being often 3 pitch values enough to cover singer's range (excluding falsetto). The loudness axis is set to the interval [0,1], assuming 0 to be the softest singing and 1 the loudest one. In our experiments, we decided to record *very soft, soft, normal* and *loud* qualitative labels for steady-states and just *soft* and *normal* for articulations. Finally, for tempo we recorded *normal* and *fast* speeds, valued as 90 and 120 BPM[7] respectively. Summarizing, for each di-allophone we sampled 3 pitches, 2 loudness and 2 tempo contexts, summing 12 different locations in the sonic space.

Subspace *B* contains different types of vibratos. We don't intend to be exhaustive but to achieve a coarse representation of how the singer performs vibratos. Afterwards, at synthesis, samples in *B* are used as parameterized templates for generating new vibratos with scaled depth, rate and tremolo characteristics. More precisely, each template stores voice model controls envelopes obtained from analysis, each of which can be later used in synthesis to control voice model transformations of samples in *A*. For vibratos, as shown in Figure 12, the control envelopes we use are the EpR (gain, slope, slope depth) and pitch variations relative to their slowly varying mean, plus a set of marks pointing the beginning of each vibrato cycle and used as anchor points for transformations (for instance time-scaling can be achieved by repeating and interpolating vibrato cycles) and for estimations (for example vibrato rate would be the inverse of the duration of one cycle). Besides, vibrato samples are segmented into *attack, sustain* and *release* sections. During synthesis, these templates are applied to flat (i.e. without vibrato) samples from subspace *A*, and the EpR voice model ensures that the harmonics will follow the timbre envelope defined by the formants while varying their frequency (see Figure 13). Depending on the singing technique adopted by the singer, it is possible that formants vary in synchrony with the vibrato phase. Hence, in the specific case the template was obtained from a sample corresponding to the same phoneme being synthesized, we can use as well the control envelopes related to the formants location and shape for generating those subtle timbre variations.

The third subspace *C* represents musical articulations. We model it with three basic types: note attacks, transitions and releases. For each of them we set a variety of labels trying to cover most basic recourses. Note we are considering here the musical note level and not higher levels such as musical phrasing or style, aspects to be ruled by the Performer Model exposed in section §4. Same as vibratos, samples here become parameterized templates applicable to any synthesis context [23].

Several issues have to be considered related to the recording procedure. Singers usually get bored or tired if the scripts are repetitive and mechanical. Therefore it is recommended to sing on top of stimulating musical backgrounds. This in addition ensures an accurate control of tempo and tuning and increases the

[7] BPM means beats per minute. In this case it corresponds to the number of syllables per minute, considering a syllable is sung for each note.

feeling of singing. Besides, using actual meaningful sentences increases further more the feeling of singing. We did so for Spanish language, using a script which computed the minimum set of sentences required to cover the di-allophone subset out of the Spanish books. An excerpt of the Spanish recording script is shown in Figure 14. It took about three or four hours per singers to follow the whole scripts. Another aspect to consider is that voice quality often degrades after singing for a long time; therefore pauses are necessary now and then. Finally another concern is that high pitches or high loudness levels are difficult to hold continuously and exhaust singers.

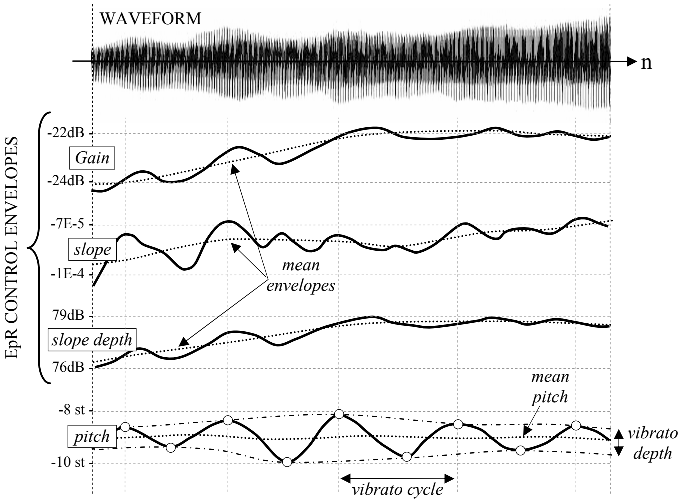

Figure 12 Vibrato template with several control envelopes estimated from a sample. The pitch curve is segmented into vibrato cycles. *Depth* is computed as the difference between local maximum and minimum of each cycle, while *mean pitch* is computed as their average.

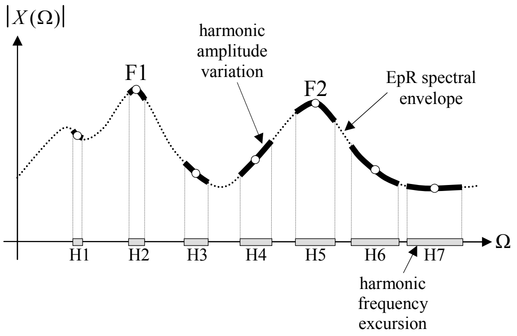

Figure 13 Along a vibrato, harmonics are following the EpR spectral envelope while their frequency is varying. For instance, when fifth harmonic (H5) oscillates, it follows the shape of the second formant (F2) peak.

Text	**en zigzag como las oscilaciones de la temperatura**
Phonetic transcription	Sil-e-n-T-i-G-T-a-G-k-o-m-o-l-a-s-o-s-T-i-l-a-T-j-o-n-e-s-d-e-l-a-t-e-m-p-e-r-a-t-u-r-a-Sil
Articulations to add	Sil-e T-i i-G G-T T-a G-k o-l s-o s-T i-l a-T T-j j-o o-n e-s a-t t-e m-p p-e e-r t-u

Figure 14 Excerpt from the Spanish recording script. The first line shows the sentence to be sung, the second one the corresponding SAMPA[8] phonetic transcription, and the third one the list of phonetic articulations to be added to the database

The creation of the singer database is not an easy task. Huge numbers of sound files have to be segmented, labeled and analyzed, especially when sampling subspace *A*. That's why we put special efforts in automating the whole process and reducing manual time-consuming tasks [40] (see Figure 15). Initially, the recorded audio files are cut into sentences. For each of them we add a text file including the phonetic transcription and

[8] SAMPA (Speech Assessment Methods Phonetic Alphabet) is a computer-readable phonetic alphabet based on the International Phonetic Alphabet (IPA), and originally developed in the late eighties by an international group of phoneticians.

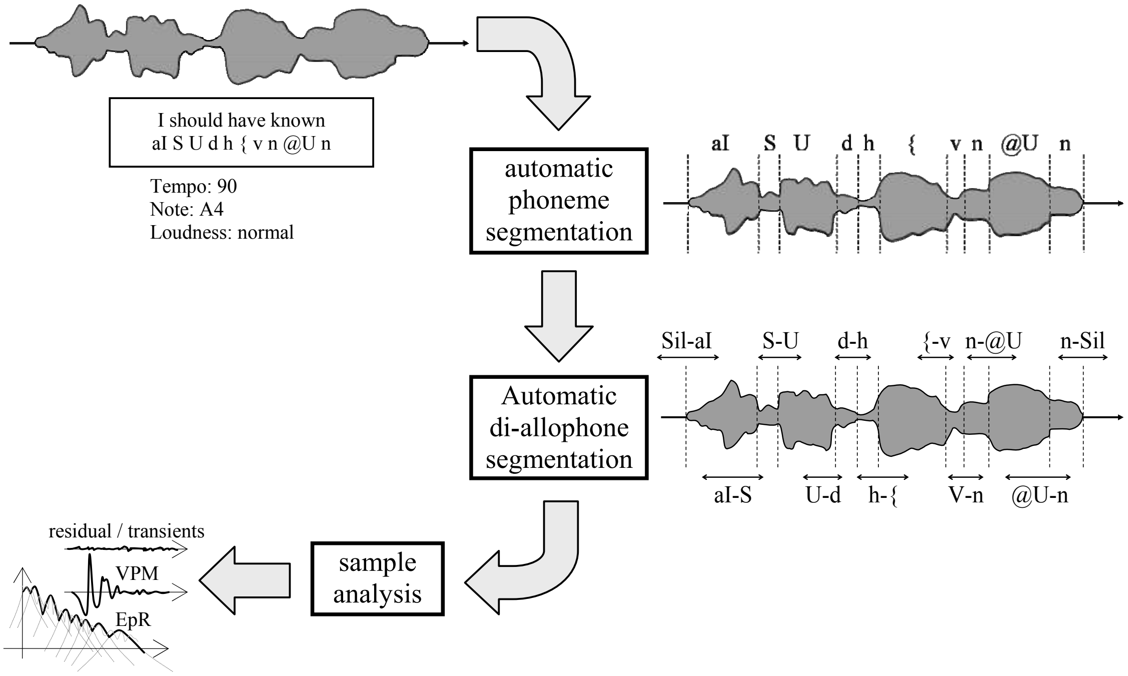

Figure 15 Singer database creation process

tempo, pitch and loudness values set in the recording scripts. Next we perform an automatic phoneme segmentation adapting a speech recognizer tool to work as an aligner between the audio and the transcription. Then we do an automatic sample segmentation fixing the precise boundaries of the di-allophone units around each phoneme onset, following several rules which depend on the allophone families. For example, in the case of unvoiced fricatives such as the English ‘s’, we end the *-s articulation just at the ‘s’ onset, and the beginning of the s-* articulation is set at the beginning of the ‘s’ utterance. We do it like this because in our synthesis unvoiced timbres are not smoothly concatenated, thus it is better to synthesize the whole ‘s’ from a unique sample.

In our recording scripts we set tempo, pitch and loudness to be constant along each sentence. One reason is that we want to capture loudness and pitch variations inherent to phonetic articulations (see Figure 16), and make them independent of the ones related to musical performance. Another reason is that we want to constrain the singer’s voice quality and expression to ensure a maximal consistency between different samples among the database, intending to help hiding the fact that the system is concatenating samples and increasing the sensation of a continuous flow. Another significant aspect is that we detect gaps, stops and timbre stable segments (see Figure 17). The purpose is to use this information for better fitting samples at synthesis, for example avoiding using those segments in the case of fast singing instead of time-compressing samples. Details can be found in [40].

The last step is to manually tune the resulting database and correct erroneous segmentations or analysis estimations. Tools are desired to automatically locate most significant errors and help the supervisor to perform this tuning without having to check every single sample.

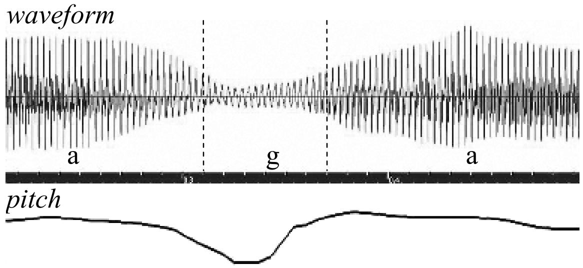

Figure 16 Loudness and Pitch variations inherent to phonetic articulations. In this figure we can observe a valley in waveform amplitude and pitch along a-g-a transition.

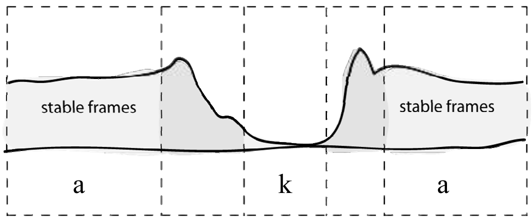

Figure 17 Gaps, stops and timbre stable segments detection

6. Performance Trajectory Generation

The Performance Trajectory Generator module converts performance actions set by the Performer Model into adequate parameter trajectories within the instrument's sonic space. We characterize singing voice Performance Trajectories in terms of coordinates in subspace A: phonetic unit sequences plus pitch and loudness envelopes. We assume that the tempo axis is already embedded in the phonetic unit track as both timing information and actual sample selection. An essential aspect when computing phonetic timing is to ensure the precise alignment of certain phones (mainly vowels) with the notes [41][4].

We saw in the previous section how the singing voice sonic space was split into three subspaces, the first one *A* including the actual samples to be synthesized, and the rest *B* and *C* representing ways of transforming those samples so to obtain specific musical articulations and vibratos. Hence, the Performance Trajectory Generator actually applies models from subspaces *B* and *C* to the coarse Performance Score coordinates in *A,* in order to obtain detailed trajectory functions within subspace *A*.

A representative example is shown in Figure 18, starting with a high-level performance score (top left) of two notes with the lyrics *fly me*. The former note is an Ab2 played forte and attacked softly. The latter a G2 played piano, ending with a long release and exhibiting a *wet* vibrato. The note transition is smooth (i.e. legato). From this input an internal score (top right) is built which describes a sequence of audio samples, a set of musical articulation templates, vibrato control parameters, and loudness and pitch values. Lowering another level, musical articulation templates are applied and the resulting Performance Trajectory (bottom right) is obtained. In this same figure, we can observe another possible input performance score (bottom left), in this case a recorded performance by a real singer, obviously a low-level score. From it we could directly generate a performance trajectory with a similar expression by performing phoneme segmentation and

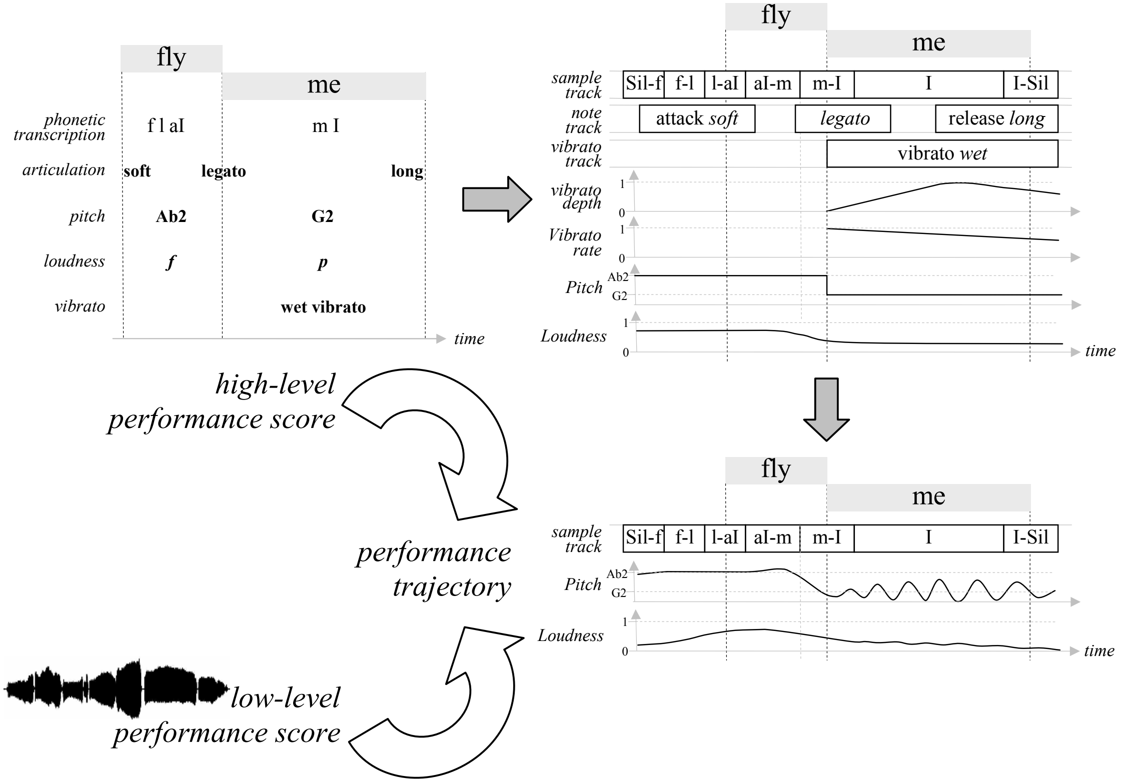

Figure 18 From performance score to performance trajectory

computing pitch and loudness curves.

The optimal sample sequence is computed so that the overall transformation required at synthesis is minimized. In other words, we choose the samples which better match locally the performance trajectory. With this aim, we compute the matching distance as a cost function derived from several transformation costs: temporal compression or expansion applied to the samples to fit the given phonetic timing, pitch and loudness transformations needed to reach the target pitch and loudness curves, and concatenation transformations required to connect consecutive samples.

7. Sound rendering

The Sound Rendering engine is the actual synthesizer that produces the output sound. Its input consists of the Performance Trajectory within the instrument sonic space. The rendering process works by transforming and concatenating a sequence of database samples. We could think of many possible transformations. However we are mostly interested in those directly related to the sonic space axes, since they allow us to freely manipulate samples within the sonic space, and therefore match the target trajectory with ease (see Figure 19).

However, feasible transformations are determined by the spectral models we use and their parameterization. In our case, spectral models have been specially thought for tackling the singing voice (as

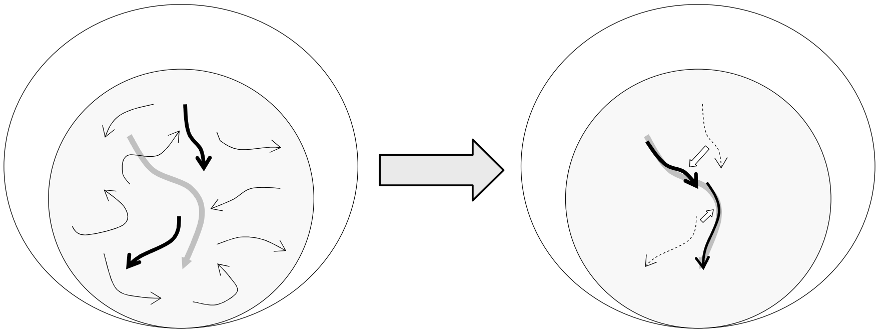

Figure 19 Matching a performance trajectory by transforming samples. On the left side we see the target trajectory in gray and the samples in black. The two selected samples are drawn with wider width. On the right side we see how these samples are transformed and approximate the target trajectory.

seen in section §3) and allow transformations such as transposition, loudness and time-scaling, all of them clearly linked to the *A* sonic subspace axes. Several recourses related to musical articulation are already embedded in the Performance Trajectory itself, thus no specific transformations with this purpose are needed by the rendering module. Still, other transformations not linked to our specific sonic space axes and particular to the singing voice are desired, such as the ones related to voice quality and voice phonation, which might be especially important for achieving expressive and natural sounding rendered performances. For example, we could think of transformations for controlling breathiness or roughness qualities of the synthetic voice. In particular, roughness transformation would be very useful in certain musical styles, such as blues, to produce growling utterances. We explore several methods for producing these kinds of alterations using our voice models in [29][42].

If we restricted our view to the sonic space we deal with, we would reach the conclusion that transformed samples do connect perfectly. However, this is not true, because the actual sonic space of the singing voice is much richer and complex than our approximation. Thus, transformed samples almost never connect perfectly. Many voice features are not described precisely by the coordinates in *A* subspace, and others such as voice phonation modes are just ignored. For example, phonetic axis describes the timbre envelope as phoneme labels, so rather coarsely. Hence, when connecting samples with the same phonetic description, formants won't match precisely. Another reason for imperfect connections is that the transposition factor applied to each sample is computed as the difference between the pitch specified in the recording scripts and the target pitch, with the aim of preserving the inner pitch variations inherent to phonetic articulations (see section §5), and therefore pitch rarely match at sample joints.

In order to smoothly connect samples, we compute the differences found at joint points for several voice features and transform accordingly surrounding sample sections by specific correction amounts. These correction values are obtained from spreading out the differences around the connection points, as shown in

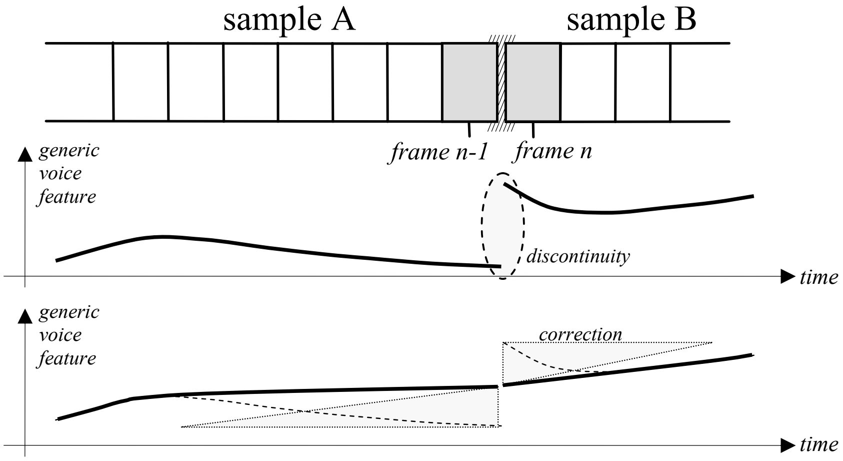

Figure 20 Sample concatenation smoothing

Figure 20. We apply this method to our voice models, specifically to the amplitude and phase envelopes of the VPM spectrum, to the pitch and loudness trajectories, and to the EpR parameters, including the controls of each formant [43]. Results are of high quality. However, further work is needed to tackle phonation modes and avoid audible discontinuities such as those found in breathy to non-breathy connections.

8. Conclusions

In this article we have introduced the concept of synthesis based on performance sampling. We have explained that although sampling has been considered a way to capture and reproduce the sound of an instrument, it should be better considered a way to model the sonic space produced by a performer with an instrument. With this aim we have presented our singing voice synthesizer, pointing out the main issues and complexities emerging along its design.

The singing voice is probably the most complex instrument and the richest one on expressive nuances. After introducing its particular characteristics, we have detailed several spectral models we developed during the last few years which specifically tackle them, and we have pointed out the most relevant problems and difficulties we found.

Then we have discussed the key aspects of the proposed synthesizer and described its components. We have distinguished two main processes. The former consists of transforming an input score into a performance trajectory within the sonic space of the target instrument, i.e. the singing voice. The latter actually generates the output sound by concatenating a sequence of transformed samples which approximates the target performance trajectory. We have put special emphasis on the issues involved in the creation of the synthesizer's database, starting with the definition of the singing voice sonic space and ending with our efforts in automating the creation process.

Although the current system is able to generate convincing results in certain situations, there is still much room for improvements, especially in the areas of expression, spectral modeling and sonic space design. However, we believe we are not so far from the day when computer singing will be barely distinguishable from human performances.

References

[1] Schwarz, D., “Corpus-based Concatenative Synthesis”, IEEE Signal Processing Magazine, vol. 24, no. 1, Jan. 2007.

[2] Lindemann, E., “Music Synthesis with Reconstructive Phrase Modeling”, IEEE Signal Processing Magazine, vol. 24, no. 1, Jan. 2007.

[3] Bonada, J., Loscos, A., Blaauw, M., “Unisong: A Choir Singing Synthesizer”, Proceedings of the 121st AES Convention, San Francisco, USA, October 2006.

[4] Sundberg, J, “The science of the Singing Voice”, Northern Illinois University Press, 1987.

[5] Kob, M., “Singing Voice Modeling as We Know It Today”, Acta Acustica united with Acustica, vol. 90, no. 4, pp. 649-661, 2004.

[6] Rodet, X., “Synthesis and Processing of the Singing Voice”, 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio, Leuven, Nov 2002.

[7] Cook, P.R., “Singing Voice Synthesis: History, Current Work, and Future Directions”, Computer Music Journal, vol. 20, no. 3, Fall 1996.

[8] Kelly, J., Lochbaum, C., “Speech Synthesis”, in Proc. of the 4th Int. Congr. Acoustics, 1962, pp. 1-4.

[9] Smith, J.O., “Physical Modeling Using Digital Waveguides”, Computer Jusic Journal, vol. 16, no. 4, pp. 74-87, 1992.

[10] Cook, P., “SPASM: A Real-Time Vocal Tract Physical Model Editor/Controller and Singer; The Companion Software Synthesis System”, Computer Music Journal, vol. 17, no. 1, pp. 30-44, 1992.

[11] Mullen, J., Howard, D.M., Murphy, D.T., “Acoustical Simulations of the Human Vocal Tract Using the 1D and 2D Digital Waveguide Software Model”, Proc. of the 4th Int. Conference on Digital Audio Effects, Naples, Italy, Oct 2004, pp. 311-314.

[12] Mullen, J., Howard, D. M., Murphy, D.T., “Waveguide Physical Modeling of Vocal Tract Acoustics: Flexible Formant Bandwidth Control from Increased Model Dimensionality”, IEEE Transactions on Audio, Speech and Language Processing, vol. 14, no. 3, pp. 964-971, May 2006.

[13] Story, B.H., Titze, I.R., Hoffman, E.A., “Vocal Tract Area Functions from Magnetic Resonance Imaging”, Journal of Acoustics Society of America, vol. 104, no. 1, pp. 471-487, 1996.

[14] Story, B.H., “Using Imaging and Modeling Techniques to Understand the Relation Between Vocal Tract Shape to Acoustic Characteristics”, in Proc. Stockholm Music Acoustics Conf., 2003, SMAC-03, pp. 435-438.

[15] Cooper, C., Murphy, D., Howard, D., Tyrrell, A., “Singing Synthesis with an Evolved Physical Model”, IEEE Transactions on Audio, Speech and Language Processing, vol. 14, no. 4, July 2006.

[16] Macon, M., Jensen-Link, L, Oliverio, J., Clements, M., George, E., “A Singing Voice Synthesis System Based on Sinusoidal Modeling”, Proc. Intl. Conference on Acoustics, Speech, and Signal Processing ICASSP, vol. 1, pp. 435-438, 1997.

[17] Klatt, D.H., “Review of Text-to-Speech Conversion for English”, Journal of Acoustics Society of America, vol. 82, no. 3,pp. 737-793, 1987.

[18] Bennett, G., Rodet, X., “Synthesis of the Singing Voice”, in *Current Directions in Computer Music Research*, Mathews and Pierce eds., pp. 19-44, MIT Press, 1989.

[19] Bonada, J., Loscos, A., Cano, P., Serra, X, Kenmochi, H., “Spectral Approach to the Modeling of the Singing Voice”, Proceedings of the 111th AES Convention, New York, USA, Sept 2001.

[20] Childers, D. G., “Measuring and Modeling Vocal Source-Tract Interaction”, IEEE Transactions on Biomedical Engineering, vol. 41, no. 7, pp. 663-671, Jul. 1994.

[21] Serra, X., "A System for Sound Analysis-Transformation-Synthesis based on a Deterministic plus

Stochastic Decomposition", PhD thesis, CCRMA, Dept. of Music, Stanford University, USA, 1989.

[22] Laroche, J., Dolson, M., "New Phase-Vocoder Techniques for Real-Time Pitch-Shifting, Chorusing, Harmonizing, and other Exotic Audio Effects", Journal of the Audio Engineering Society, vol. 47, no. 11, pp. 928-936, Nov. 1999.

[23] Bonada, J., Loscos, A., Mayor, O., Kenmochi, H., “Sample-based Singing Voice Synthesizer using Spectral Models and Source-Filter Decomposition”, Proceedings of 3rd Intl. Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications, Firenze, Italy, 2003.

[24] Laroche, J., “Frequency-Domain Techniques for High-Quality Voice Modification”, Proc. of the 6th Int. Conference on Digital Audio Effects, London, UK, September, 2003.

[25] DiFederico, R., “Waveform Preserving Time Stretching and Pitch Shifting for Sinusoidal Models of Sound”, Proc. of the 1st Int. Conference on Digital Audio Effects, Barcelona, Spain, November, 1998.

[26] Smits, R., Yegnanarayana, B., “Determination of Instants of Significant Excitation in Speech using Group Delay Function”, IEEE Transactions on Speech and Audio Processing, 1995.

[27] Yegnanarayana, B., Veldhuis, R., “Extraction of Vocal-Tract System Characteristics from Speech Signal”, IEEE Transactions on Speech and Audio Processing, vol. 6, no. 4, pp. 313-327, 1998.

[28] Bonada, J., “High Quality Voice Transformations based on Modeling Radiated Voice Pulses in Frequency Domain”, Proc. of the 7th Int. Conference on Digital Audio Effects, Naples, Italy, Oct 2004.

[29] Loscos, A., Bonada, J., “Emulating Rough And Growl Voice in Spectral Domain”, Proceedings of 7th Int. Conference on Digital Audio Effects, Naples, Italy, Oct 2004.

[30] Kob, M., “Physical Modeling of the Singing Voice”, PhD thesis, Institute of Technical Acoustics, Aachen University, Germany, 2002.

[31] Röbel, A., “A New Approach to Transient Processing in the Phase Vocoder”, Proc. of the 6th Int. Conference on Digital Audio Effects, London, UK, September, 2003.

[32] Rodet, X., Potard, Y., Barrière, J.B.B., “The CHANT Project: from the Synthesis of the Singing Voice to Synthesis in General”, Computer Music Journal, 8(3):15-31, 1984.

[33] Kaegi, Werner, Tempelaars, “VOSIM – a New Sound Synthesis System”, Journal of the Audio Engineering Society, vol. 26, no. 6, pp. 418-425, 1978.

[34] Bresin, R., Friberg, A., “Emotional Coloring of Computer-Controlled Music Performances”, Computer Music Journal, vol. 24, no. 4, 44-63, 2000.

[35] Sundberg, J., Askenfelt, A., Frydén, L., “Musical performance: A Synthesis-by-Rule Approach”, Computer Music Journal, 7, 37-43, 1983.

[36] Widmer, G., Goebl, W., “Computational Models of Expressive Music Performance: the State of the Art”, Journal of New Music Research 33(3), 203-216, 2004.

[37] Laurson, M., Norilo, V., Kuuskankare, M., “PWGLSynth: A Visual Synthesis Language for Virtual Instrument Design and Control”, Computer Music Journal, vol. 29, pp. 29-41, Fall 2005.

[38] Black, A., “Perfect Synthesis for All of the People All of the Time”, IEEE TTS Workshop 2002, Santa Monica, USA, 2002.

[39] Llisterri, J., Mariño, J.B., “Spanish Adaptation of SAMPA and Automatic Phonetic Transcription”, ESPRIT PROJECT 6819 (SAM-A Speech Technology Assessment in Multilingual Applications), 1993.

[40] Bonada, J., Loscos, A., Blaauw, M., “Improvements to a Sample-Concatenation Based Singing Voice Synthesizer”, Proceeding of the 121st AES Convention, San Francisco, USA, October 2006.

[41] Ross, J., Sundberg, J., “Syllable and Tone Boundaries in Singing”, 4th Pan European Voice Conference, Stockholm, Sweden, Aug 2001.

[42] Fabig, L., Janer, J., “Transforming Singing Voice Expression - The Sweetness Effect”, Proc. of the 7th Int. Conference on Digital Audio Effects, Naples, Italy, 2004.

[43] Bonada, J., Loscos, A., Kenmochi, H., "Sample-based Singing Voice Synthesizer by Spectral Concatenation", Proceedings of the Stockholm Music Acoustics Conference, Stockholm, Sweden, 2003.