Voks : A Vocal Instrument Family Based on Syllabic Sequencing of Vocal Samples

Grégoire Locqueville

SORBONNE UNIVERSITÉ

École doctorale 391 : Sciences Mécaniques, Acoustique, Électronique et Robotique de Paris (SMAER)

Institut Jean le Rond d'Alembert, Sorbonne Université, CNRS, UMR 7190

Voks : A Vocal Instrument Family Based on Syllabic Sequencing of Vocal Samples

Grégoire LOCQUEVILLE

Thèse de doctorat

Écrite sous la direction de :

Christophe d'ALESSANDRO

Boris DOVAL

Soutenue le 20 juin 2022, devant un jury composé de :

Nathalie HENRICH BERNARDONI	Directrice de Recherche	Rapporteure
Marcelo M. WANDERLEY	Professeur	Rapporteur
Thierry DUTOIT	Professeur	Examinateur
Sylvie GIBET	Professeure	Examinatrice
Claire PILLOT-LOISEAU	Maître de Conférences	Examinatrice
Axel RÖBEL	Directeur de Recherche	Examinateur
Christophe d'ALESSANDRO	Directeur de Recherche	Directeur de thèse
Boris DOVAL	Maître de Conférences	Co-encadrant

Résumé

Nous présentons l'étude de Voks, une famille d'instruments vocaux qui permettent le contrôle par les mains de l'intonation, du séquencement rythmique et de paramètres de qualité vocale d'une voix articulée. Cette thèse développe particulièrement la question du contrôle rythmique. Sa gestion, inspirée par la théorie cadre/contenu de production de la parole, repose sur la mise en correspondance d'impulsions résultant d'un geste de tapping biphasique, d'une part, avec des points de contrôle placés sur un échantillon de voix pré-enregistré d'autre part. Ce mode de fonctionnement met en jeu une boucle complexe impliquant à la fois la production gestuelle et la perception auditive du rythme, que la notion de centre perceptif, ou P-center, permet de modéliser. Le contrôle manuel direct de la mélodie et de l'effort vocal, ou chironomie, est réalisé par des contrôleurs continus tels que la tablette graphique et le thérémine, des interfaces qui permettent des gestes expressifs nouveaux. Pour la synthèse, deux vocodeurs différents, World et SuperVP, sont intégrés à Voks et comparés. Ils permettent le contrôle de la mélodie, du rythme et du timbre de la voix. Voks utilise en entrée des échantillons de voix munis d'étiquettes syllabiques, des données qui peuvent être fournies directement ou générées automatiquement à la volée. La famille d'instruments Voks est appliquée à la musique et à l'apprentissage de langues étrangères.

Abstract

The study of Voks is presented here. Voks is a family of vocal instruments that allows for control, using one's hands, of intonation, rhythmic sequencing, and vocal quality parameters of an articulated voice. This dissertation focuses more specifically on the question of rhythmic control, which is here based on the frame/content theory of speech production. It relies on matching impulsions produced by a biphasic tapping gesture, with control points placed on a prerecorded voice sample. This mode of operation relies on a complex loop that involves both gestural production and auditory perception of rhythm, which can be modeled by the notion of the P-center, or perceptual center. Direct manual control of melody and vocal effort, also known as chironomy, is performed using continuous controllers such as the graphic tablet and the theremin, enabling new expressive gestures. For synthesis, two distinct vocoders, World and SuperVP, are integrated into Voks and compared. They make it possible to control melody, rhythm and vocal timbre. Voks takes as input voice samples equipped with syllabic labels; those data can be either directly fed as input or generated automatically on the fly. The Voks instrument family has been applied to music and to foreign language acquisition.

Remerciements

Je tiens d'abord à remercier mes encadrants, Christophe d'Alessandro et Boris Doval, dont la gentillesse, le respect et l'ouverture d'esprit n'ont jamais fait défaut. Nos échanges, qu'ils fussent scientifiques et/ou musicaux ou non, ont toujours été enrichissants et j'ai beaucoup appris à vos côtés. Je remercie également les autres membres de mon jury : mes rapporteurses Marcelo Wanderley et Nathalie Henrich Bernardoni, dont le regard sur mon travail durant ces années m'a permis d'améliorer sa qualité, ainsi que Claire Pillot-Loiseau, Axel Röbel, Thierry Dutoit et Sylvie Gibet, pour leurs remarques passionnantes au cours de la soutenance.

Je remercie mes collègues qui sont pour beaucoup devenu.e.s des ami.e.s : Thomas, qui a partagé avec moi, tout au long de notre thèse, les joies et les peines du métier de doctorant, et Xiao ; vous avez beaucoup contribué à ma thèse, tant scientifiquement que moralement. Merci aussi à Manuel, Jean-Théo, et à toustes les membres du labo, notamment les autres doctorant.e.s qui sont toustes super cool, et aux membres du LAM, avec qui c'est toujours un plaisir de parler et découvrir de nouveaux aspects de la richesse de la musique.

J'ai perdu de vue certain.e.s ami.e.s au cours de la thèse, j'en ai rencontré, d'autres sont resté.e.s à mes côtés tout du long, mais je vous remercie toustes du fond du cœur. Merci à Ferréol, Gaëlle, Céline, Shana et toustes les autres. Si la rédaction d'une thèse est un travail dont la difficulté est apparue clairement sur le moment, les mesures d'isolement prises pour faire face au covid ont aussi eu un impact non moins réel, quoique moins facile à cerner. Durant ces moments, rester en lien avec vous a été vital, et votre soutien, essentiel.

Enfin, merci à ma famille. Papa et maman qui m'ont toujours soutenu dans tout dans l'ouverture et la bienveillance, Matthieu, Mathilde et Bertille, toujours là pour moi.

Contents

Introduction

By 2022, most people are familiar with the idea of a voice being produced without directly resorting to any human's natural voice: common computer and smartphone operating systems come equipped with text-to-speech synthesis software, and talking virtual assistants have seen widespread adoption in the 2010's.

On the other hand, electronic sound synthesizers are used routinely in a wide variety of musical genres. Those include systems that aim to imitate acoustic instruments, ones that aim to create sounds never heard before, and everything in between. Just as acoustic instrumentalists interact with their instrument using gestures to control sound as it is being produced, electronic artists often drive sound synthesis by interacting with some interface — a typical, but far from exclusive example being the piano-like keyboard.

In contrast, the concept of using gestures to control a synthesizer that imitates the voice, especially articulated voice, seems to remain foreign to most. Several reasons behind that situation can be put forward. Even if one's definition of a music instrument is to include the vocal apparatus, it certainly is a peculiar one, being no separate from the performer's body [22]. Incidentally, the most common interfaces — piano-like keyboards, pads —, as well as MIDI, the most used communication protocol for real-time control, seem to have been designed for discrete, possibly polyphonic control, whereas voice is inherently continuous and monophonic when on its own.

Another distinction, and the one that we are mostly interested in in this work, is the ability of voice to transmit two types of data. On the one hand, voice is able to convey musical and expressive information: a voice may sing a melody with its pitch and rhythm, and adopt various kinds of timbre — dimensions that many other music instruments are also able to act on. On the other hand, voice also conveys linguistic information: when speaking, one usually expects listeners to be able to parse the produced sound into text expressed in some language, a feat not easily achieved with instruments. In other words, voice may be used in two ways: as a means of *expression* or as a means of *communication*.

Those two dimensions, musical/expressive and linguistic, are far from independent or even totally separate. After all, most phonemes can be seen as fast changes in timbre. On its side, prosody — pitch, rhythm, stress — is essential in language [15], providing help in parsing speech or adding information to it, even in cases when it is not phonetic in its own right. Nevertheless, the presence of the linguistic dimension sets vocal synthesis apart from other types of synthesis, and presents it with novel challenges.

The speed and diversity of timbre changes associated with the articulated voice is at the heart of a doubly demanding task for the performer: articulated speech involves the continual selection of one among many phonetic units, and this selection has to happen fast. When designing a gestural vocal synthesizer, several goals thus compete: one may wish the instrument to be easy to learn and/or use, as well as for its output to be intelligible. One may also want its users to be able to choose utterances from as large a set as possible, and/or with as much spontaneity as possible.

Several strategies are thus possible. Each of those strategies implies a trade-off between ease of use, expressivity, proximity to natural voice, and performer freedom. One such strategy is to give control to the user over the full complexity of the linguistic dimension of the voice [39, 45], by designing a gesture for each sound that is likely to occur in a given language. Another strategy would be to deliberately reduce that complexity, by targeting only a subset of possible vocalizations, such as those composed only of vowels [48].

This dissertation is centered around a third approach for dealing with the linguistic complexity of the human voice: any text can be played with the synthesizer, but the text that is to be played has to be determined before using the synthesizer. Users get to control musical and expressive dimensions of vocal sounds without having to worry about controlling the segmental details of the synthesized vocal production, but synthesizing any text remains possible.

To be more precise, our approach is based on the hypothesis that real-time, gesture-controlled modification of *rhythm*, *pitch* and *intensity* of an existing voice sample makes for a convincing expressive vocal music instrument. Much of this doctorate has been dedicated to the design and development of a family of vocal synthesizers, named Voks, whose design adopts that approach. The singing voice is not our only target, and we also examine to what extent spoken voice can be convincingly controlled by gestures.

After a first chapter which aims to provide context around vocal production and vocal synthesis, the second chapter of this document gives a detailed description of the design principles and implementation of the instruments in the Voks family.

A special focus is put on two aspects of the synthesizer; the first is temporal control. Among several available temporal control modes, *syllabic* control provides an intuitive way to organize the vocal production temporally. This relies on an analogy between the cyclic opening/closing motion of the jaw during speech, following the frame/content theory of speech production [78], and the opening/closing gesture of the hand. This analogy relies on impulses resulting from the performer's gestures, on one side, and control points placed on the prerecorded signal, on the other side. Those two types of points live in different time dimensions — one in the time of the performance, the other in the time during which the source sample was recorded —; we introduce the notion of *time index* to make the bridge between both.

The second aspect which we put a special focus on is the choice of interface: the synthesizer has been developed in a modular way, separating the synthesizer algorithm from the control interfaces in a way that makes it relatively easy to connect different types of interfaces. Each interface comes with its own set of gestures and must be approached by the performer in a specific way, even though all control the same underlying algorithm. This warrants Voks' designation as a "family" of instruments rather than a single instrument. Chapter 3 offers a more musical

perspective, with a comparison of the different interfaces.

Voks is being considered not only as a musical instrument, but also as a tool for language learning: it is hoped that by providing control over the prosodic dimensions of a foreign language using a new modality, that of hand gestures, those dimensions become more tangible and easier to internalize. With that in mind, chapter 4 describes investigations over the ability to accurately reproduce intonation patterns using hand gestures as compared to natural voice.

Chapter 5 gets into more detail about syllabic control. Voks' syllabic control algorithm is not a trivial one, and the complex interaction between the algorithm on one side, and the performer's motor production and auditory perception on the other side, make performing and studying rhythm in Voks challenging. The control points essential to syllabic control are compared with the existing notion of P-center, related to auditory perception. A preliminary study is described, which investigates the suitability of the syllabic method in the case of speech.

The strategy adopted in Voks, that of requiring that a text be determined before the time of performance, has proven to be fruitful, but remains somewhat constraining to performers. Chapter 6 describes efforts to evolve Voks into the direction of more flexibility.

Chapter 7 evokes the design and development of Voks from an engineering perspective; technological and organizational challenges are touched upon. To provide perspective, another development realized during this doctorate, the reimplementation of the Cantor Digitalis, is described.

In chapter 8, we give an overview of the contexts in which Voks has been used, including, but not restricted to music performances.

Chapter 1

Context and State of the Art

Contents

1.1 Human voice

Voice is a complex phenomenon that involves organs from the lungs to the mouth and nose; [123] gives a rather detailed account of the anatomic and physical processes involved in voice

production. Let us here recall the general mechanisms involved.

Most vocal sounds originate in the larynx, which contains a flexible organ, a pair of so-called vocal folds. Those can be brought together, completely blocking the flow of air from the lungs, or pulled apart, letting it go through unobstructed, by specific muscles. Between those two extremes, there is an intermediary state where, if air is expelled from the lungs, the vocal folds oscillate between being closed and open, resulting in a periodic pattern whose fundamental frequency and spectrum can be varied by the speaker. It is also possible to make it so that the flow of air is turbulent. Whether the pattern at the vocal folds is periodic or turbulent, the air then goes through the pharynx, the mouth, and possibly the nasal cavity, which all impact the flow of air.

When a vowel is uttered, the air flow comes out of the respiratory system to circulate between the vocal folds. Those are put in vibration, and the resulting acoustic flow propagates through the vocal tract. The characteristics of the vocal folds as well as the geometry of the vocal tract all have an influence on the resulting signal. In particular, the geometry of the vocal tract changes with time based on the configuration of articulators. That geometry can be modified in many ways by the motion of a number of organs called *articulators*. This impacts the pressure pattern outside of the body and results in a wide variety of possible sounds.

Voice registers

Although numerous parameters influence the production and perception of the voice, vocal sounds are generally classified into three categories, or "registers", each with their specific frequency range and possible spectra. [113] Two such registers are most commonly known, among other names, as the M1, or *chest*, and M2, or *falsetto*, registers. Two other mechanisms exist, known as *fry* and *whistle*.

Other non-standard ways to use one's voice may be called "registers" as well. This is the case, for instance, of the register known as "growling", which involves vibration of the *ventricular folds*, located above the vocal folds. [7]

1.2 The source-filter model

In many vocal sounds, the cavities that the voice goes through act as a resonator through which the acoustic flow passes. A common approximation is to treat such sounds as being the result of some signal, filtered linearly: the unfiltered signal is seen as corresponding to the acoustic flow at the output of the vocal folds, and the filter is determined by the position of the articulators.

Such a model, as any other, is imperfect: it does not take into account any nonlinear phenomena happening in the vocal apparatus, and it neglects any influence that the oral and nasal cavities may have on what happens at the vocal folds. What is more, it does not fully explain sounds produced by turbulent phenomena in the mouth, such as fricative consonants. Nevertheless, it so happens that vocal sounds are well approximated by the source-filter model. The source-filter model is especially useful in applications where the physical details of voice production are not essential, such as voice synthesis and processing. [43]

Let us now get into a little more detail about each of the components of the source-filter model.

1.2.1 The source

Generally, the source is some combination of a periodic, spectrally rich signal, with some noise. Either the periodic part, the noise, neither, or both, may be zero. Looking at the vocal signal through the prism of the source-filter model, the respective amounts of harmonic signal and noise, as well as the fundamental frequency (if applicable), are examples of properties of the source.

In some cases, the source is taken to be a simple pulse train, and the spectral content of the waveform at the output of the vocal folds is taken into account in the filter. In that case, the source-filter model is clearly not a physical model, but a phenomenological one, in that it is based on the output signal and not on the underlying physical process.

1.2.2 The filter

Among the most notable features of the filter are *formants*, spectral peaks related to resonances of the cavities. The center frequency and amplitude of each formant are determined by the position of the articulators, and are in large part responsible for the timbre of the output sound. The formants are key to identifying the linguistic content of the voice signal [34].

In cases where the source is just a pulse train, the spectrum of the vocal fold signal is also encoded in the source. In that case, it can be modeled as a so-called glottal formant, even though it does not correspond to any resonance of the cavities that the air flow goes through. [36]

1.2.3 Classification of speech sounds

Vocal sounds are generally distributed into two categories: *vowels*, which involve no constriction, occlusion or vibration of the vocal tract, and *consonants*, which do.

Vowels

The two dimensions of a diagram parameterized by formant center frequencies are correlated [46] with the position of certain articulators: the first formant frequency tends to be lower when the tongue is close to the palate, hence the name *height* to denote the vertical dimension of vowel diagrams; and the second formant frequency, lower when the tongue is close to the back of the mouth, hence the name *backness* for the horizontal dimension.

Further distinctions between vowels exist: vowels are often described as either *rounded* or *non-rounded*, depending on the roundedness of lips. In some languages, including French, some vowels are *nasal*, that is, air does not only escape through the mouth during their production, but also through the nose.

Consonants

Consonants encompass a large class of sounds; the shared characteristic is constriction, occlusion or vibration of the vocal tract during sound production. Consonants are most commonly

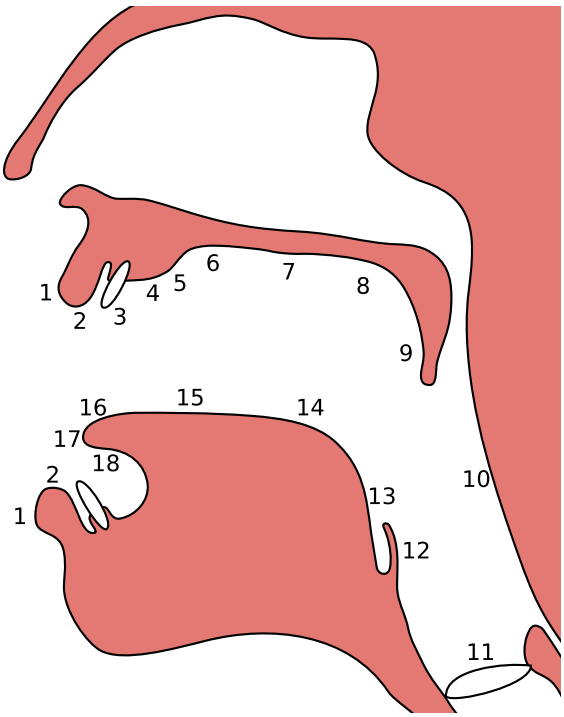


Figure 1.1: *Schematic section of the vocal tract, with the possible places of articulation labeled with numbers: 1. Exo-labial. 2. Endo-labial. 3. Dental. 4. Alveolar 5. Post-alveolar. 6. Pre-palatal. 7. Palatal. 8. Velar. 9. Uvular. 10. Pharyngeal. 11. Glottal. 12. Epiglottal. 13. Radical. 14. Postero-dorsal. 15. Antero-dorsal. 16. Laminal. 17. Apical. 18. Sub-laminal.*
Created by Wikimedia user ish shwar, *edited by Wikimedia user* Rohieb, *CC BY-SA 3.0, via Wikimedia Commons*

produced by air flow coming out of the lungs, just as vowels are. Contrary to vowels, however, consonants may or may not involve oscillation of the vocal folds.

The set of possible consonants features a wide variety of sounds. Those are commonly sorted according to three dimensions:

Place of articulation As mentioned above, what sets consonants apart from vowels is obstruction of the vocal tract. Consonants are classified based on the location of the articulators involved in the obstruction. Figure 1.1 shows the possible places of articulation.

Manner of articulation Consonants are also classified according to the way that a given articulator obstructs the vocal tract. Not every articulator may obstruct in every way: for instance, a glottal tap is considered impossible to perform.

Voicing Some consonants involve vibration of the vocal folds; others do not. The former are called *voiced*; the latter *voiceless*. Most voiceless consonants have a voiced counterpart, and vice versa, although this is not systematic.

Some consonants are special in that they do not involve expulsion of air through the lungs;

those are called *non-pulmonic*. Many languages only involve pulmonic consonants; this is the case of English, French, German, and Mandarin.

1.2.4 Phonetics

Phonetics is the study of speech sounds. An individual speech sound is called a *phone*. Two different phones may be different, even though they may be used interchangeably without altering the meaning. Such pairs of phones are then said to be *allophones* of each other, and instance of the same *phoneme*.

The *international phonetic alphabet* (IPA) [66] is a writing system that aims to be able to transcribe the sounds of all spoken languages, regardless of whether they have associated writing systems. Itself based primarily on the latin alphabet, it includes many more characters and diacritics. The international phonetic alphabet may be used either to transcribe speech *phonemically*, that is, denoting instances of the same phoneme with the same symbol, or *phonetically*, denoting each distinct sound with a different symbol. Phonemic transcriptions are conventionally written in between slashes /·/; phonetic ones are written in between square brackets [·].

The North Wind and the Sun were disputing which was the stronger, when a traveler came along wrapped in a warm cloak.
/ðə ˈnoɹθ ˌwɪnd ən (ð)ə ˈsʌn wəˇ dɪsˈpjutɪŋ ˈwɪtʃ wəz ðə ˈstɹɑŋgəˇ,
wɛn ə ˈtɹævələˇ ˌkem əˈlɑŋ ˈɹæpt ɪn ə ˈwoɹm ˈklok./
[ðə ˈnɔɹθ ˌwɪnd ən ə ˈsʌn wəˇ dɪsˈpjuɾɪŋ ˈwɪtʃ wəz ðə ˈstɹɑŋgəˇ,
wɛn ə ˈtɹævləˇ ˌkem əˈlɑŋ ˈɹæpt ɪn ə ˈwɔɹm ˈklok.]

Figure 1.2: *Orthographic, phonemic and phonetic transcriptions of the same sentence, taken from [66].*

Figure 1.2 shows an example transcription of a sentence using the IPA, both phonemically and phonetically. The two transcriptions have much in common; a few differences appear, such as the voiceless plosive /t/ turning into the voiced flap [ɾ] in the word "disputing". [66] notes that knowing to interpret the phoneme /t/ as the phoneme [ɾ] in this context relies on a number of conventions — the convention being, in this particular case, that the consonants /t/, /d/ and /n/, when intervocalic and occuring before an unstressed vowel, turn into the flap /ɾ/.

Symbols of the IPA are arranged in a chart, reproduced in figure 1.3 in which consonants are classified by place of articulation, manner of articulation and voicing, and vowels are arranged on the vowel trapezium.

Although most phones and phonemes correspond to individual speech sounds, other features of sound are also considered in phonetics. When those features alter individual phones, they are said to be *segmental*; if they span several ones, they are called *suprasegmental*.

In some languages, suprasegmental features are said to be *phonemic*, that is, changing them alters the meaning. A well-known example is tone in tonal languages, in which the fundamental frequency contour is instrumental in determining the meaning of speech. The IPA provides ways to denote such phonemic features, as well as some non-phonemic ones such as stress.

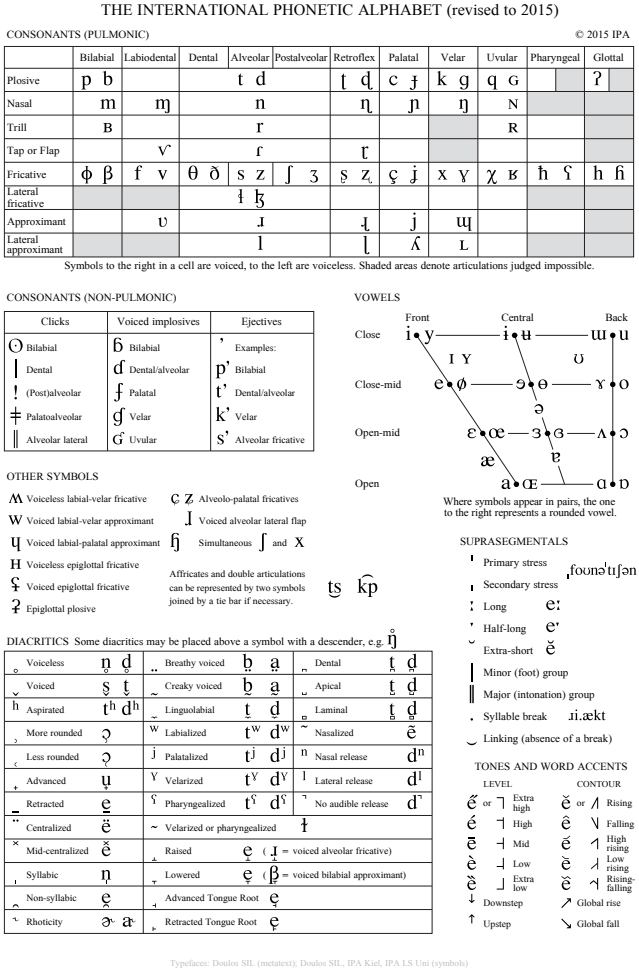


Figure 1.3: *The IPA chart,* `http://www.internationalphoneticassociation.org/content/ipa-chart`*, available under a Creative Commons Attribution-Sharealike 3.0 Unported License. Copyright © 2015 International Phonetic Association.*

1.3 Voice synthesis

Much work has been conducted on the subject of offline vocal synthesis; that is, synthesis of a sound sample based on information that is entirely provided at the beginning of the task. The typical application of offline vocal synthesis is text-to-speech, whereby audio is generated based on a text provided entirely in advance. Research in synthesis controlled with gestures in real time has been comparatively limited. This dissertation, however, is mainly concerned with performative voice synthesis, therefore a special focus is given to gesture-controlled synthesis.

1.3.1 Parametric voice synthesis

Parametric synthesizers are based on an algorithm that produces sound based on a set number of parameters that evolve in time. Such parameters generally include fundamental frequency, frequencies and amplitudes of formants, as well as voicing. They are not necessarily intuitive, and are thus generally not directly controlled by the user. Nevertheless, their comparatively small number makes dealing with them easier than with other types of synthesis.

Voice synthesis before the electronic age: Kempelen's speaking machine

Though the advent of the digital age has drastically expanded the possibilities for synthesizing the voice, producing sound that imitates the human voice without using one's vocal apparatus has been a concern since long before the existence of electronic devices.

At the end of the 18^{th} century, Wolfgang von Kempelen developed a totally mechanical "speaking machine" able to produce both consonants and vowels. In his treatise [13], Kempelen starts by discussing the theories of speech of his time, before going over the organs at play in speech production, as well as the way phonemes of European languages are produced. He then gives a detailed description of his machine and how it is able to produce a large amount of those phonemes. A short, last part acts as a manual, briefly giving instructions for the user of his machine to play each phoneme.

In a sense, von Kempelen's speaking machine mimics the vocal apparatus, with most parts of the machine playing a role broadly similar to an organ in the human body:

- Air gets injected into the system by means of a bellow, the mechanical equivalent of lungs.
- The air vibrates a reed, which acts as a glottis.
- An oboe bell acts as the mouth. By partially or totally covering the artificial mouth in various ways with their hands, the user can reproduce the formants of several vowels.
- Two small holes act as nostrils, which allow sound to escape even when the artificial mouth is totally obstructed, thus producing nasal consonants.
- In addition to those elements, a "wind chest" contains several mechanisms that can be activated to produce sounds such as fricatives. The wind chest is harder to relate to a body organ, drawing inspiration instead from those organs that can be found in churches.

Von Kempelen's machine requires the simultaneous control of different elements: activation of the bellow, of the wind chest mechanisms, and covering of the artificial mouth. A user would certainly have had to practice thoroughly before speaking with it. Even then, and despite enthusiastic accounts of demonstrations of the machine after its creation, the editors of [13] doubt its ability to produce consistently identifiable speech sounds.

Dudley's Voder

Without disregarding the inventiveness of devices such as von Kempelen's, the advent of electronics opened new perspectives for all types of synthesis, including vocal synthesis. In particular, Homer Dudley's work at Bell Labs proved influential both for creative and practical applications,

mainly through his two related inventions, the *Voder* and the *Vocoder*, the latter of which is described in section 1.4.

Like von Kempelen's machine, the Voder [39] is composed of elements that can be made to correspond to body organs (see figure 1.4). In this case, however, the signal is no longer acoustic, but electric. A buzzer produces a spectrally rich harmonic signal whose fundamental frequency can be adjusted, acting like the glottis; this signal gets modulated by filter that act as the formants in natural voice. Inharmonic noise can be added and filtered too, allowing for either purely harmonic, mixed or unvoiced sounds to be produced.

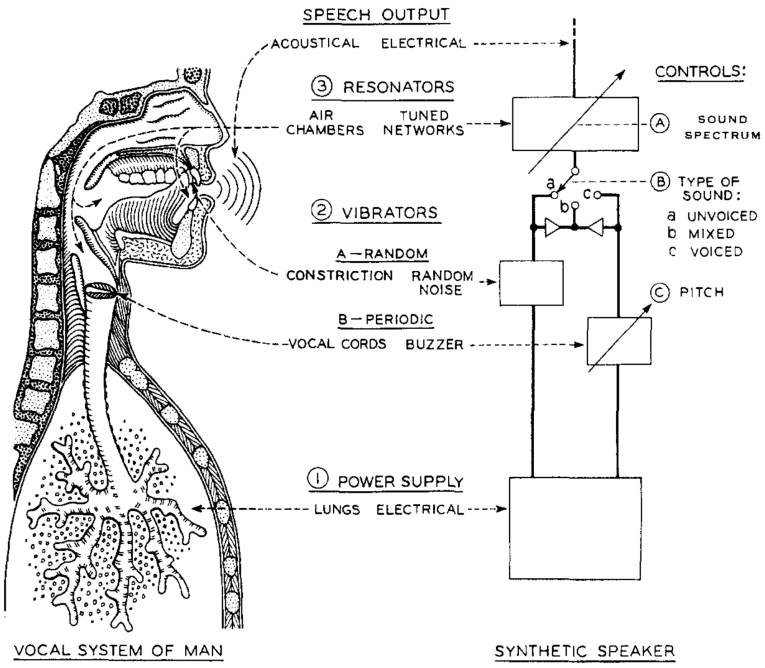


Figure 1.4: *Diagram outlining the parallel between the human vocal apparatus and Dudley's Voder. Figure reprinted from [39], Page No. 747, Copyright (1939), with permission from Elsevier*

The Voder is controlled by a dedicated keyboard to select not pitch, but phonetic content, by acting on the filters that act on the buzzing and noise signals. The pitch is controlled by a pedal. From the admission of its creator, the Voder's interface is a difficult one to master: selected operators had to follow lessons and daily practice for about a year before having developed a good technique.

The electrical nature of the Voder makes design of a control method more flexible than in a purely acoustic device such as von Kempelen's, and in his article, Dudley describes an iteration process whereby some details of control (such as the "spatial arrangement of the keys") are only determined after the global structure of the system has been set up. This prefigures the decoupling between control interface, on one hand, and synthesis algorithm, on the other hand, that digital technology has later made possible.

Glove-Talk II

Glove Talk II [45] produces sound thanks to the help of such a parametric synthesizer. Here, the gestures of a user are captured by a glove equipped with ten sensors. The captured data are then fed to neural networks trained to associate gestures to phonemes, which output the value of synthesis parameters. The fundamental frequency and volume are an exception, in that they are controlled directly without going through a network.

Cantor Digitalis

Cantor Digitalis [48] is yet another gesture-controlled parametric synthesizer. The input data comes from a graphic tablet; control parameters, such as pressure of the stylus or position along each tablet axis, are directly mapped to relatively intuitive parameters such as *pitch*, *vocal effort*, *breathiness* and *tension*. Those are converted into synthesis parameters such as formants using rules, i.e. generally a mathematical formula. The nature of vowels is determined by the position of the finger in a rectangle that represents the vowel space. That position is interpolated between points labeled with vowels for which the formant center frequencies, bandwidths and amplitudes have been finely tuned.

An extension to Cantor Digitalis, Digitartic, has been proposed to allow for synthesis of articulated voice with consonants, but was deemed too difficult to control by its creators to pursue serious investigations. [47]

1.3.2 Corpus synthesis

Corpus-based synthesis encompasses all methods that rely on a "corpus", that is, a database of existing sound samples. Such samples will often have been recorded well in advance, and labeled with additional information.

Concatenative synthesis

Perhaps the most obvious corpus-based method is concatenative synthesis [41]: depending on the input, units of prerecorded sound are selected and put together to form an utterance. The most commonly used type of units used in concatenative synthesis is that of *diphones*, *i.e.* pairs of successive phones [96, 10].

The MBROLA project is a popular concatenative synthesis system based on diphones. [40] It represents diphones not as raw audio, but coded using a custom representation which allows for smoothly connecting spectrally different pieces of audio, in addition to storing diphones more efficiently than audio.

Other units than the diphone are possible as well, such as the "vocalic sandwich" [14], a unit explicitly designed so that the phones at its boundary are "robust", meaning that splitting them and gluing them back to one another is not excessively detrimental to the quality of the synthesis. The use of concatenative synthesis has been explored for real-time voice synthesis. [3]

Among concatenative synthesis systems, the (non-real time) ISiS synthesizer is particular in

that it takes as input not only text, but also a musical score.

ISiS ISiS is a diphone-based text-to-speech synthesis system developed at IRCAM [4]. It performs non-real time text-to-speech synthesis in French based on a text provided by the user. However, and contrary to most text-to-speech synthesizers, it also requires the user to specify the melody and rhythm to be synthesized. It is also possible to make it perform intensity variations.

In practice, ISiS takes one text file as input, containing:

- The text to be synthesized, in the ASCII-compatible X-SAMPA phonetic format;
- For each syllable of the above specified text, a pitch in MIDI format. Non-integer values are allowed, enabling microtonality. Only one pitch is allowed per syllable, so glissandi are not possible.
- For each syllable, a duration value.
- Optionally, an intensity value for individual syllables.

It should be noted that the input pitches are only globally respected in the synthesized audio signal; ISiS includes microprosodic pitch variations, as a singer's natural voice would feature.

While ISiS was originally designed to synthesize a natural-sounding voice singing a specified melody, we used it for another purpose (see sections 6.2 and 6.4.2), taking advantage of its ability to generate audio with a controlled rhythm and pitch — notwithstanding the aforementioned microprosodic events.

Synthesis based on machine learning

Data from a corpus can also be used not by direct manipulation, but as training data for machine learning. Approaches exist for synthesizing audio of various types, including text-to-speech. [124] Trained statistical models have also been used for real-time modification of an existing vocal sample in order to modify high-level parameters such as speaker identity or degree of articulation. [6]

1.3.3 Gesture-controlled modification of an existing sample

Beside synthesis systems that attempt to create original textual content, some systems are focused on modifying existing samples through gestures. Among commonly modified parameters are prosodic ones, notably pitch. A discussion of such "singing instruments" is available in [20].

Synthesizers have been developed, notably a real-time versions of the CALM [27] and TD-PSOLA [73] synthesizers, which are to be coupled with a control interface to allow performers to gesturally control sound parameters.

The choice of interface for such gestural control is of primary importance and will be extensively discussed in this dissertation. A discussion of gestural control for sound synthesis in general is available in [128]. In particular, the graphic tablet has been used for control of sound synthesis for more than two decades. [116, 127] Use of the tablet for pitch control of monophonic sounds, notably vocal sounds, was further investigated in [28, 29]. The quality of pitch control,

using a graphic tablet, of a vocal sample, which has been called *chironomy*, has been evaluated and shown to be comparable to that of natural voice. [24, 25]

Such systems include *Calliphony* [75, 74], which can be used to change the pitch and rhythm of a given voice sample, and is controlled with a graphic tablet, and Vokinesis [31, 30], which includes a novel rhythm control method. Vokinesis' control method is used, with some modifications, in Voks, the main object of study of this dissertation, and as such will be described in detail in chapter 2.

1.4 The vocoder: an essential tool for voice processing and manipulation

The name *vocoder* covers several related, but distinct signal processing techniques that originate in Dudley's work at Bell Labs (section 1.4.1). Nowadays, the term has two main meanings. In musical contexts, "vocoder" refers to a specific variation on Dudley's original technique (section 1.4.2). In signal processing contexts, it refers to a more general class of systems based on some variant of the source-filter model, consisting of an *analysis* module that encodes a voice signal into an intermediate representation, and a *synthesis* module that converts such a representation back into an audible signal. The systems described in sections 1.4.3 and 1.4.4 are examples of such systems.

1.4.1 Dudley's vocoder

In [39], Dudley describes a "synthetic speaker", controlled by gestures in real-time, based on the source-filter model: a combination of a buzzing sound and a noise source form the source, and a number of filters can be selected with keys to generate different phonemes. Dudley's so-called vocoder [38] is introduced as a "voice-controlled" counterpart to that synthetic speaker.

The original vocoder consists in an analysis circuit composed of:

- A **frequency analyzer** that detects the input signal's original pitch;
- A number of **band-pass filters** whose combined bands cover the voice spectrum. The output is the amplitude of the filtered signals.

The output of those bricks can then be used to resynthesize a voice signal using the previously described "synthetic speaker":

- A buzzing sound is synthesized whose fundamental frequency is the output of the frequency analyzer;
- That sound passes through filters similar to those used for analysis, and the output amplitude of each filter is weighted by the corresponding amplitude that was measured during analysis.

The above described workflow assumes that no modification of the signal is desired. This is typically useful in audio compression, for the following reason. The typical variations of an audio signal are on the order of 10^{-4}s, whereas the typical time constant of articulator motions is

on the order of 10^{-2}s. If sampling a voice, the sampling frequency thus needs to be several orders of magnitude greater when directly sampling the voice signal than when sampling the output of the analysis.

Compression is not the only application considered at the creation of the first vocoder, though, nor is it the one mentioned in Dudley's article. In it, creative modifications of the signal made possible by the vocoder are mentioned, including modifying or replacing the fundamental frequency of the input signal, and replacing the source by something else altogether, be it electronically generated noise, to get a whispering effect, or everyday sounds.

1.4.2 The musical vocoder

The vocoder as described in musical contexts is a variant of the original vocoder [33] where the fundamental frequency of the input signal is not measured — only the output amplitude of the bandpass filters is taken into account. The output signal is obtained by filtering some source signal using the same bandpass filters weighted by the observed amplitudes.

The source signal may be any signal, but it is most commonly the output signal of a monophonic instrument such as a keyboard-controlled synthesizer. Such a source produces a signal with segmental features typical of voice, but with a flat pitch contour that endows it with a robotic quality. Other commonly used sources are polyphonic instruments (guitar, keyboard-controlled synthesizers).

It is also possible to use other signals than the voice to estimate the filters. Using a pitched signal (vocal or instrumental) as the source and drums as the filter, for instance, one obtains a signal that has the dynamics of drums, but with a pitch.

The following vocoders are not instances of Dudley's original vocoder; rather, they are technologies inspired by his invention.

1.4.3 The World vocoder

World's vocoded representation [87, 83, 84, 85] is based on a double version of the source-filter model: a voice signal $s(t)$ is modeled as the sum of white noise $w(t)$ filtered by a finite-response linear filter $h_{ap}(t)$, and an impulse train $\text{Ш}_{T_0}(t)$ with fundamental frequency $f_0 = \frac{1}{T_0}$, filtered by another finite-response linear filter $h_p(t)$:

$$s(t) = \text{Ш}_{T_0}(t) * h_p(t) + w(t) * h_{ap}(t) \tag{1.1}$$

or, expressed in the spectral domain:

$$S(\omega) = \text{Ш}_{f_0}(\omega) H_p(\omega) + W(\omega) H_{ap}(\omega) \tag{1.2}$$

Thus, to encode a speech signal, three pieces of information are needed at each point in time: fundamental frequency, and the response of both filters.

In reality, information about the filters is stored in another way, made possible by the fact that the phase of both filters is discarded. This does not make a difference for the noise filter, since the phase of white noise is already random. However, it does make a difference, theoretically, for the filter applied to the impulse train. In practice, this does not seem to be excessively detrimental to the quality of sound, as evidenced by the good quality of sound reconstructed from World's vocoded representation [86].

Discarding the phase in both filters means that they can be represented by their power response $|H_p(\omega)|^2$ and $|H_{ap}(\omega)|^2$. World uses an equivalent representation that uses the *global envelope* and an *aperiodicity ratio*, both functions of frequency, defined as:

$$\begin{aligned} E(\omega) &= |H_p(\omega)|^2 + |H_{ap}(\omega)|^2 \\ R(\omega) &= \frac{|H_{ap}(\omega)|^2}{E(\omega)} \end{aligned} \tag{1.3}$$

The power responses $|H_p(\omega)|^2$ and $|H_{ap}(\omega)|^2$ are readily recovered from $E(\omega)$ and $R(\omega)$. To summarize, at each point in time, the World representation of a speech signal is composed of:

- Fundamental frequency f_0,
- The global envelope $E(\omega)$,
- The aperiodicity ratio $R(\omega)$.

In the remainder of this document, we will refer to a triple (f_0, E, R) as a *World frame*. A voice signal can then be represented as a *stream of World frames*, that is, a function T that associates a World frame $T(t) = (f_0(t), E(t), R(t))$ to each point t in time.

Software

World is distributed as open-source software under the permissive 3-clause BSD license. It consists in C source code files:

- Analysis source files, that, given a digital audio signal, estimate f_0, $E(\omega)$ and $R(\omega)$ at each one of a sequence of regularly spaced frames. By default, frames are placed every 5 ms.
- Synthesis source files, allowing the decoding of vocoded data into a digital audio signal.

It should be noted that those source files do not allow analysis to be performed in real time: the whole audio sample has to be loaded in memory to be analyzed. In contrast, synthesis may, in principle, be performed while analysis data is being fed in real time.

1.4.4 The phase vocoder

In *phase vocoding* [35], an approach to vocoding somewhat distinct from traditional ones, the phase is regarded as relevant information. This makes it possible to deal not only with monophonic signals, but with a large class of sounds. Phase vocoding is most notably used to perform time and pitch distortions. Phase vocoding was introduced in [52]; a perhaps more accessible description

can be found in [35], and [8] specifically discusses real-time aspects of phase vocoding. Although phase vocoding was designed with audio compression in mind, uses of the technology to perform time and pitch transformations have been envisioned from its inception.

Phase vocoding consists in two steps, analysis and synthesis, between which an intermediary modification step can be added. Analysis consists in conversion from the audio signal into an intermediate representation based on a series of Fourier transforms of the windowed signal. The optional modification step consists in manipulating that representation to obtain effects that would have been difficult to achieve in the time domain. Synthesis converts that representation back into an audio signal that, under suitable conditions on the parameters of the analysis and synthesis, is indistinguishable from the original.

SuperVP

An improved version of the phase vocoder algorithm, based on [32], has been implemented under the name SuperVP. The improvements of SuperVP over the traditional phase vocoder are:

- A novel way to deal with transients [108, 111]: whereas in previous vocoders, transients were detected and dealt with on certain time periods (that is, at each time, the sound was considered to be composed either exclusively of transients, or not at all), in SuperVP, frequency bins are discriminated: two frequency bins in the same frame can be treated differently.
- Integration of a particular technique for spectral envelope estimation [109, 110], allowing preservation of timbre while performing other transformations such as pitch shifting.

SuperVP makes possible many transformations on top of the classical pitch and time modifications; of special interest to us is the ability to perform *cross-synthesis* (see section 6.4.2).

For uses in real time, SuperVP has been turned into a collection of Max/MSP objects.

1.4.5 Pitch-synchronous overlap-add (PSOLA)

In the case of periodic signals, knowledge of the fundamental frequency f_0 can be taken advantage of by using frames synchronous with the periodicity of the signal. In the PSOLA technique [89], a series of instants separated by the period of the signal $\frac{1}{f_0}$ is defined, and each frame is centered around one of those so-called *pitch marks*.

The TD-PSOLA technique, for *time-domain PSOLA*, works by replaying the frames while skipping or repeating some, in such a way that a target fundamental frequency and speed are achieved. If the frames are played with the original number of frame per second, time deformation independent from pitch change is achieved. By changing the number of frames per second, pitch can be manipulated too. Another PSOLA denomination, FD-PSOLA, for *frequency/Fourier-domain PSOLA*, covers methods that subject frames to spectral modifications before synthesis, to correct or manipulate the spectral envelope.

PSOLA has been implemented as a real-time algorithm [73], but it assumes a periodic signal. If dealing with noise, repeating the same frame at regular intervals inevitably results in a periodic

signal. Thus unvoiced vocal sounds get turned into periodic sounds: the technique cannot be used as is for time expansion. [30] proposes to adapt the method to unvoiced sounds by playing frames of random length.

1.4.6 Vocoders in this work

In this dissertation, vocoders will be used as a tool for modifying and resynthesizing a prerecorded vocal signal. Two vocoders are used in particular: World and SuperVP. A compared commentary of the benefits and drawbacks of each for our purposes is available in section 2.5.3.

1.5 Rhythm in perception and production

This document is centered around a type of vocal synthesis where rhythm is controlled by a tapping motion. As such, we are interested in rhythm from two perspectives: perception of rhythm in speech, and gestural production of rhythm.

1.5.1 Rhythm in speech as compared to music

Both speech and music feature temporal hierarchy. In music, [71] proposes a hierarchy of pulse scales at which one could perform beat detection on a given piece of meter-based music. The finest level, the *tatum*, has a period equal to the shortest duration encountered on a consistent basis in the piece. The *tactus* corresponds to what is commonly refered to as the "beat", and it is what tempo indications in beat per minute refer to. The *measure* corresponds to a longer temporal unit. The number of tatums in each tactus, and of tactuses in each measure is usually a whole number that remains the same at least for a few measures.

Speech rhythm features a different kind of hierarchy. In addition to segmental timing — that is, timing of the individual phones —, the timing of syllables is also important. Depending on language, those can in turn be organized into larger units, called *feet*. This has linguists to classify languages into *stress-timed* ones and *syllable-timed* ones, a terminology introduced by [97]. The terminology has later been extended to include *mora-timed* languages, whose timing is determined by units smaller than the syllable, the *morae*. Correlates based on the acoustic properties of the speech signal and its phonetic segmentation have later been found which demonstrate the validity of that classification. [103] As pointed out by [126], however, speech rhythm is not necessarily tied to segmental content.

1.5.2 Perceived rhythm in speech

The temporal arrangement of syllables is of primordial importance in the perception and production of rhythm in articulated voice. Syllables, however, are complex objects, built up from phonemes that mutually affect each other, and that themselves evolve in time.

The notion of *perceptual center*, or *P-center* in short, is an abstraction that makes it possible to talk about syllabic rhythm without having to deal with the whole complexity of the syllabic phenomenon. P-centers were introduced in [88], in which the following, avowedly imperfect

definition of the notion is given: *the P-center of a syllable is defined to be its "psychological moment of occurence"*. P-centers allow one to assign definite time points to syllables, in much the same way as note onsets, in a musical context, allow one to assign time points to notes.

It should be noted that although the notion of P-centers has been mostly discussed in the particular case of the English language, investigations have been led into other languages [61]. In particular, the P-center phenomenon has been shown to occur in all three categories of the common rhythm typology of languages, with evidence observed not only in English, a stress-timed language, but in Spanish and Japanese as well, respectively a syllable-timed and a mora-timed language.

P-center models

Several models have been proposed to predict P-center position. Morton *et al.* failed to relate the position of P-centers to any point that could be determined by obvious acoustic means (such as peak intensity). [88] Fowler proposes that listeners do not perceive P-centers based directly on acoustic characteristics, but rather based on the underlying articulatory processes they infer from speech sounds. [55]

Schütte proposes a signal-based approach, albeit for non-speech sounds. [115] A signal-based model for prediction of position of the non-speech counterpart of P-centers is proposed. As in the case of speech, the relationship between acoustic parameters and position of this P-center analogue is not a trivial one, but an algorithm is provided in the form of a block diagram, based on the envelope of the stimulus transient.

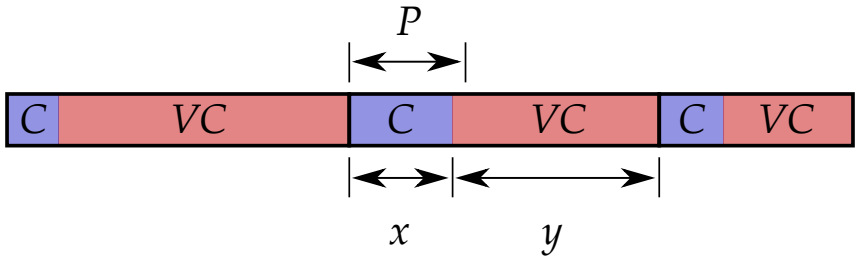


Figure 1.5: *Illustration of the model presented in [80]. Each syllable is composed of a consonantic onset (C) followed by a rhyme (VC). The P-center of a syllable is located a distance P from the beginning of the syllable, where P can be computed based on the lengths x of the onset and y of the rhyme, using equation 1.4.*

[80] proposes a much simpler model for P-center position. It is not directly based on the signal, but instead on the temporal repartition of phonemes in the syllable. The model relies on decomposing syllables into subunits smaller than the syllable but greater than the phoneme: as described in [126], a syllable is composed of an *onset* followed by a *rhyme* (which may in turn be decomposed into a syllabic nucleus and an optional coda). The model from [80] then predicts the location of the P-center of a syllable to be at a location determined by the following equation:

$$P = \alpha x + \beta y + k \tag{1.4}$$

where

- P is the duration between the beginning of the syllable and the P-center;
- x is the duration of the syllable onset;
- y is the duration of the syllable rhyme;
- α and β are constants determined experimentally. [80] gives values $\alpha = 0.65$ and $\beta = 0.25$;
- k is a time shift inherent to the fact that P-center location may only be determined relative to one another.

The terms of equation 1.4 are schematized on figure 1.5.

[62] introduces a model for the so-called "center of gravity" of a syllable weighted by its acoustic energy. Although it does not allow to predict the position of P-centers, it is claimed that this notion enables ordering of the P-centers.

[98] refutes [80]'s simple model based solely on consonant onset and rhyme. They propose a significantly more complex signal-based algorithm, which relies on detection of events of interest in the loudness curve of the stimulus.

[67] asserts that P-center position is indeed directly related to syllable onset using a protocol based on tapping (see below).

1.5.3 Studying gestural production of rhythm: the tapping protocol

To study rhythm production as well as sensorimotor synchronization, a simple protocol, the tapping protocol, has been used extensively. The tapping protocol consists in exposing subjects to a rhythmic stimulus, usually a series of audible clicks, and having them tap with their finger on a surface equipped with sensors that can detect the exact instant of their tapping.

The simplicity of the tapping protocol has allowed researchers to develop many variations around it. [106, 107] give an overview of the tests that have been conducted under the tapping protocol or variations of it.

One notable feature of taps produced synchronously as a response to a series of audible clicks is the presence of a systematic negative delay [5]; that is, subjects tend to anticipate when tapping along an auditory stimulus.

Tapping along a voice

Of special interest to us is the test conducted in [1, 2], which looks into tapping with a vocal stimulus, and relates production and perception by comparing the data produced, using the same stimuli, in tests based on the following three protocols:

- *Tapping* while hearing the stimuli to place a beat on each syllable;
- *Moving* "movable clicks" around the stimuli up until they were perceived as happening at the moment of occurence of each syllable;
- *Judging* whether fixed clicks superimposed on the stimuli coincided with the moment of

occurence of syllables.

Of those protocols, the first one involves the tapping task, a production task. The last two are perception tasks, which establishes a relation between perception and production. The methods are found to be in general agreement. A notable observation is the increased fit when the syllable in question is a stressed one.

Lateralization

A natural question to ask is whether handedness has an influence on tapping. Although this question, to our knowledge, has not directly been studied, a few observations can ben made.

First, it should be noted that handedness is not simply a matter of using one's dominant or secondary hand. [18] points out that humans can be put on a spectrum, not only between left- and right-handedness, but also between mixed handedness and strong handedness. This complexity of the handedness phenomenon calls for caution when drawing any conclusions about handedness.

The very relevance of the notion of handedness itself depends on the task to be performed: [53] makes the distinction between two types of tasks. The first type is that of so-called "ballistic" tasks; those can be performed at once without the sensory feedback playing any role. The typical example of a ballistic task is the motion of the left hand of a well-trained violinist when playing a fast series of notes: assuming the piece has been practiced thoroughly enough, the musician plays notes somewhat mechanically, without relying on haptic feedback. The second type of task is that of tasks that rely on a feedback loop between perception and control. [53] argues that handedness has no influence in the first type of task, and that it only plays a role in the quality of feedback.

Nevertheless, the effect of handedness on rhythm production tasks has been studied. [64] shows a "rhythm laterality effect" disparity between two hands (or feet) in a (musical) rhythm task consisting in marking the beat with one limb and reproducing a rhythm with another. All subjects — left- and right-handed alike — did better when their *right* hand tapped the rhythm. However interindividual disparity was greater than laterality disparity, and musician subjects managed to perform the tasks in all cases regardless.

Estimating P-centers with tapping tasks

Some researchers have attempted to use the tapping protocols to locate P-centers.

[67] uses an experimental protocol based on tapping along an auditory stimulus and adjusts for the systematic temporal shift that occurs in tapping [5] to estimate P-center location. They then compare a variety of methods with one another and their own.

Yet another estimation method has been proposed [105], based on the *phase correction response* (PCR). The PCR is the correction that arises when tapping along a sequence of clicks featuring one offset click among otherwise regularly spaced ones. [125] proposes adapting that protocol by replacing clicks with syllables: a sequence of identical syllables is played, except for one syllable; the resulting PCR is then measured to see how shifted the different syllable was, compared to the others.

1.6 Conclusion

In this section, we have described the general principles behind voice production and modelling, as well as the different general vocal synthesis techniques, with a special focus on performative voice synthesis. The notion of a *vocoder*, crucial to many voice synthesis techniques, has been explained; some examples of vocoders have been given and explained, among which World and the phase vocoder will be used by us for voice synthesis, as described in chapter 2. In addition to voice synthesis-related matters, some background regarding perceived and gesture-produced rhythm, notably vocal rhythm, has been given; such background will be relevant when dealing with the syllabic rhythm control method in use with our synthesis system, and its evaluation.

Chapter 2

Voks, a Gesture-Controlled Synthesizer for Articulated Voice

Contents

2.1 Introduction

Much of the work of this doctorate has revolved around a vocal synthesizer named *Voks*, including its design, development, use and evaluation. The synthesizer is the object of [76], which has been reproduced in the appendix of this document. Voks is based on previous work, notably that of Samuel Delalez and Christophe d'Alessandro on Vokinesis [30], a system which we now consider to be an earlier iteration of the Voks family.

Voks sets itself apart from predecessors such as Cantor Digitalis by its ability to control a truly articulated synthesized voice, featuring consonants as well as vowels. This allows for synthesis of any text. As a counterpart, the specification of the utterance to be synthesized is, by default, a somewhat involved process, thus limiting the creative possibilities in another way. An attempt to make text specification more fluid for the performer is developed in section 6.2.

Voks allows its user to control various parameters of a synthesized voice:

- **Rhythm**, a dimension central when dealing with articulated voice, which poses specific challenges;
- **Pitch**, also crucial in both speech and singing. Pitch has been thouroughly investigated in the context of research around non-articulated voice;
- **Timbre**: although those aspects of timbre that are related to lexical content are generally fixed before a performance using Voks, users are granted direct control over a range of timbral parameters.

In this chapter, we describe Voks technically. There are two similar versions of Voks, based on different vocoders (see section 2.5). The general architecture of the World version is represented on figure 2.1.

Voks takes two types of input:

- An **audio recording** that will serve as the basis for the synthesis. Depending on the version of Voks (see section 2.5), the recording might need to be processed first, or on the contrary may be fed as is to the synthesizer. In any case, the syllabic structure of the recording needs to be written in a file, the *labeling file* — see section 2.5;
- **Gestural data** from control interfaces — complemented by data from the graphical interface, for parameters that need not be continuously changed. See section 2.2. Those gestural data are then processed based on certain rules, to give parameters that will be fed to a synthesis engine.

The synthesis engine (section 2.5) is in turn composed of a vocoder and, for the vocoder that allows it, a module to modify the spectral representation of the audio to synthesize.

2.2 Interface

2.2.1 Graphical interface

The graphical interface can be seen on figure 2.2 and is composed of several zones:

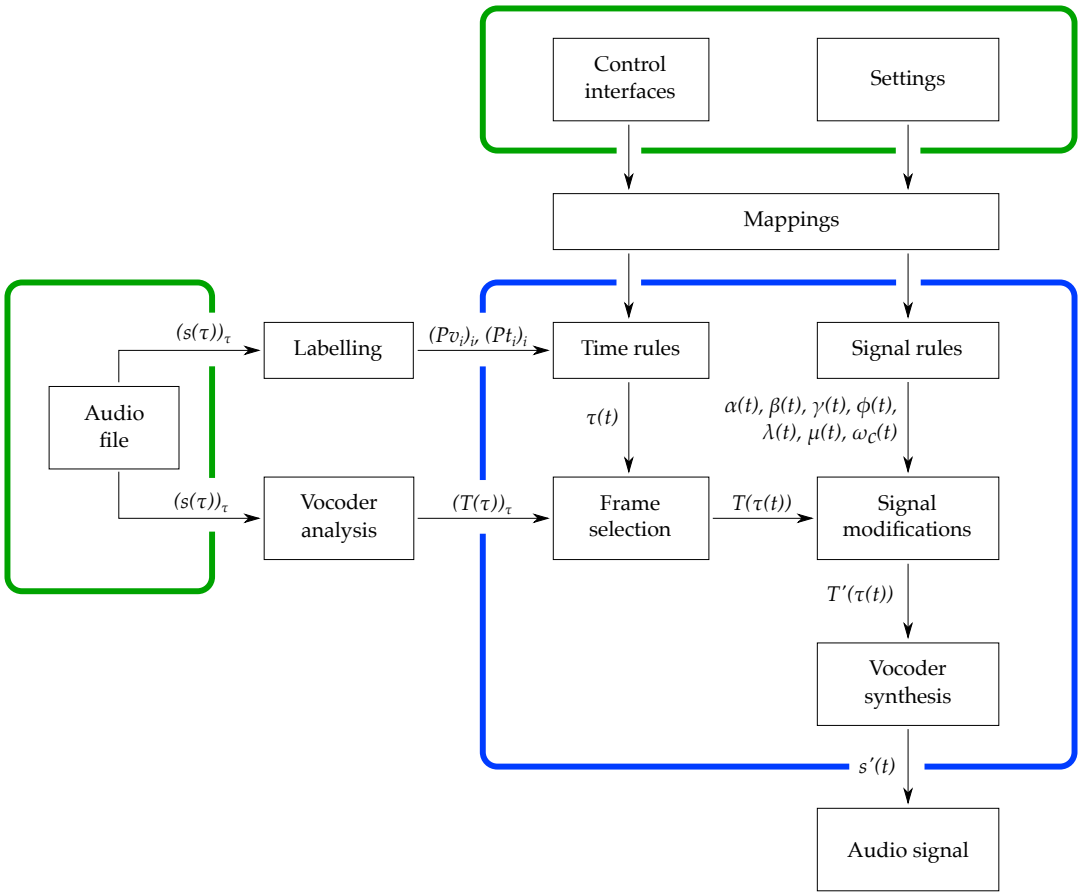


Figure 2.1: *Architecture of Voks. Green rectangles enclose inputs; the blue box encloses the system* per se. *The role of each block, together with the data that they communicate to each other, is explained in what follows.*

- The *File locations* zone includes a button to load a menu of texts. Clicking on an item in the list loads the input files corresponding to the text.
- The *Controllers* zone allows for selection of hardware gestural controllers, including a controller for rhythm, one for pitch and vocal effort, and one for other parameters.
- The *Audio settings* zone allows for selection of audio input and output devices.
- The *Parameters* zone allows for setting the value of diverse parameters and activating/deactivating diverse timbre modifications.
- The *Effects* zone allows to apply to the synthesized sound some effects customarily applied to voice, namely equalization and reverbation.
- The *Presets* zone allows for loading and saving presets, which set the value of all parameters at once.

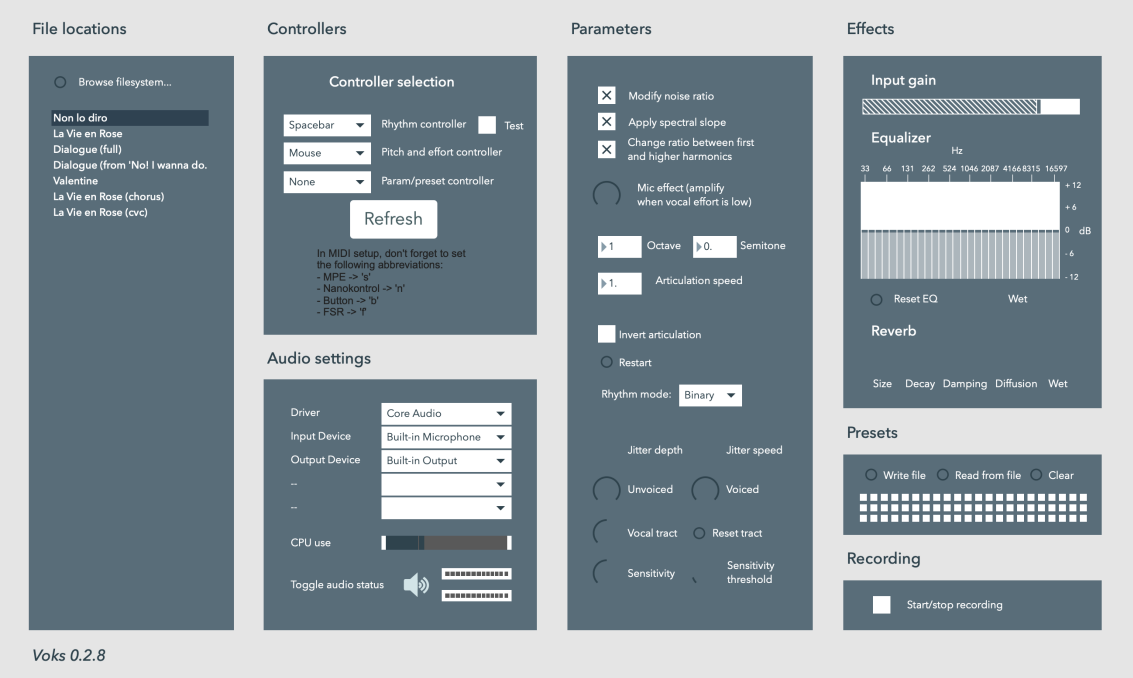


Figure 2.2: *Voks' graphical interface*

2.2.2 Control interfaces

Gestural control in Voks is not limited to any one set of interfaces. Instead, for both syllabic and continuous control, performers are presented with a list of interfaces to choose from in the graphical interface.

Two different types of interfaces are necessary to control Voks: a discrete one, to control rhythm on a syllabic level, and a continuous one, to control pitch and vocal effort. Another interface may be added for easy control of secondary parameters and fast preset recalling.

Syllabic control interfaces

In Voks' main control method, rhythm is controlled *syllabically* and *biphasically*: the performer manages the temporal progression by sending two signals for each syllable. Thus an interface is needed that is based on a cyclic motion (typically, a cycle of alternating presses and releases) during which two signals are sent for each cycles. Such interfaces include:

- Alphanumeric or musical keyboard keys,
- On/Off buttons,
- Pressure sensors,
- *Pads*: surfaces originally designed to be tapped to control digital percussions.

None of those interfaces is too exotic, and most of them have a commercialized version which is able to communicate using the MIDI protocol.

Continuous interfaces

Although rhythm control is a prominent theme in this work, control of pitch and vocal quality remains necessary in a singing instrument. Pitch control using a graphic tablet has been developed and studied in previous work, notably in the context of Cantor Digitalis [48]. With Voks, we make use of that existing method, and we explore new ones. Aside from the graphic tablet, the method for control of continuous parameters that has been most thoroughly explored is the theremin.

The diversity of interfaces — and, as a consequence, the diversity of available gesture types to control Voks — has led us to consider Voks equipped with different interfaces not as different playing modes of the same one instrument, but as several distinct instruments from the same family. Hence the devising of a new name for the two most instruments of the family: Voks equipped with the graphic tablet has been named *C-Voks*, in reference to calligraphy — the visual art of writing — and chironomy, the name which has been given in [24] to the gestural control of intonation. As for Voks equipped with the theremin, it has been named T-Voks.

Graphic tablet with stylus The graphic tablet, together with a stylus, has been used historically in performative vocal synthesis [75]; as noted in [75], its advantage over other continuous interfaces such as the joystick or the sensor-equipped glove is the greater spatial range and the reuse of gestures similar to those involved in writing, a task which a significant part of humanity is expert in.

Gestural interactions with the Wacom Intuos Pro graphic tablet are registered thanks to existing Max/MSP externals [100]. Note that those objects only work with a specific, obsolete version of the Wacom driver, which itself only runs on older versions of the Mac operating system. As such, another solution will have to be provided if the system is to be used in the long term.

The externals make the following available in Max/MSP:

- Position of the stylus along the X and Y axes;
- Pressure of the stylus on the surface of the tablet;
- Other data such as tilt angle of the stylus; those are not taken into consideration in Voks.

As in previous singing instruments, position of the stylus along the horizontal axis is mapped to pitch, and pressure, to vocal effort. As in other instruments, a mask reproducing the pattern of a piano is affixed to the surface of the tablet to help performers find their bearings.

Theremin The theremin is based on a very simple gestural control method — though visually impressive and difficult to master. Two antennas are attached to the instrument; the proximity of each hand of the performer with each antenna respectively determines the frequency and the sound level of the output sound, a continuous, harmonic waveform. Many theremins feature a way to manipulate the spectrum, but this is done via knobs, and thus cannot be controlled by the performer while they are playing.

Theremins usually include voltage outputs for performers to control other devices with their interactions with the theremin; one output corresponds to fundamental frequency/right antenna, and the other output, to sound level/right antenna. In a sense, those theremins feature a built-in

separation between interface and synthesis. However, the output sound is close to sinusoidal and simple enough that performing a signal processing analysis can give real-time pitch and volume information reliably and fast enough that it can be used to control a synthesis algorithm. The benefits of this approach are the ability to connect the theremin and the computer using only a sound card, a widely available type of device, as well as its versatility: the method is not bound to any voltage-pitch correspondence convention, and in principle, anything that produces sound with a clear pitch could even be substituted for the theremin.

In Voks, the voltage output of an Moog Etherwave theremin was initially used; a small custom electronic circuit with an Arduino microcontroller converted the voltage output into MIDI data. The fragility and material complexity of such a setup has later been avoided by directly analysing the audio. This is done with the `sigmund~` Max/MSP external, which outputs a pitch and a sound level in real time based on input audio.

The resulting instrument has been presented in [131], reproduced at the end of this document.

MPE interfaces MPE (MIDI Polyphonic Expression) is a protocol that makes use of the MIDI protocol to continuously modulate parameters attached to individual notes even when playing polyphonically. As a typical example, the MPE protocol enables musicians to bend the pitch of individual notes in a chord. MPE being based on the existing and widespread MIDI protocol implies that no new hardware is needed to decode MPE messages.

A number of interfaces that output data in the MPE format are commercially available. Although nothing in the MPE specification mandates it, most MPE interfaces come in the form of a tactile surface, the horizontal dimension of which is mapped to pitch in a manner reminiscient of the way that graphic tablets are used in vocal synthesis; what is more, the pressure of fingers of those surfaces is usually picked up as well, just like the pressure of the stylus on the tablet is picked up in synthesizers such as C-Voks and T-Voks.

Hence it seems quite natural to map MPE pitch and pressure to Voks' pitch and vocal effort inputs. Notwithstanding the MPE protocol being able to transmit polyphonic data, current versions of Voks do not feature polyphonic synthesis, thus only one of the voices is selected when receiving data for several voices at once. A possible improvement of Voks would be to allow synthesis of several simultaneous voices.

In Voks, MPE data is interpreted using the dedicated objects built into Max/MSP, and horizontal position/pressure are mapped to pitch/effort as one would expect.

Most existing MPE interfaces are *instrument-inspired controllers*, to borrow the words of [128]; specifically, controllers inspired by the design of the ubiquitous music keyboard. It would thus only seem natural to use the gesture of pressing/releasing a key to control rhythm syllabically, instead of adding another piece of hardware such as a button or pressure sensor to the equation. Indeed, MPE interfaces do register "*NoteOn/NoteOff* messages when a finger starts or stops pressing the surface, like any MIDI interface would. However, controlling rhythm with "NoteOn/NoteOff"s suffers from the following issue: if ceasing contact with the surface is to be necessary for rhythm control, there will be times when the finger is not in contact with the interface, and thus there is no horizontal position, and no pitch value to provide the synthesizer.

Several basic solutions have been envisioned for pitch to be specified even when the finger is depressed : keeping pitch constant until the finger presses the surface again, or requiring that another finger be in contact with the surface, which will be in charge of pitch specification when the "main finger" is not available. Those solutions are unsatisfactory in that they lead to discontinuity of pitch, and further thought will have to be given to reach an adequate control method for the MPE on its own.

Mobile devices Mobile devices can be used in much the same way as graphic tablets. The specifics are detailed in section 4. Let us simply note here the differences with the graphic tablet:

- The position along the X and Y axes is no longer given by the position of a stylus, but by that of the finger on the surface of the tablet.
- The pressure of the finger on the surface is not taken into account; intensity is determined on an "all-or-nothing" basis based on the presence or the absence of a contact.

Other interfaces have been tested with Voks, though not extensively:

Sensel Morph tablet The Sensel Morph tablet differs from the Wacom tablet in that it registers information not about a single point of contact of a stylus with surface, but about every point of the tablet. This is more information than needed, but it is perfectly possible to deduce information about individual points of contact. The Sensel Morph has the advantage of not requiring a stylus. It has been designed for musical applications; as a consequence, there exist a software tool, a Max/MSP library, that converts the output of the tablet into data usable in Max. Unlike the Max/MSP object that deals with the data of the Wacom tablet, it is well integrated and does not require the installation of an additional driver that might become obsolete.

The spatial resolution of the Sensel tablet seems insufficient, with audible pitch discontinuities when performing a glide. It remains to be seen whether lowpass filtering the data and adding an appropriate pitch correction mechanism can solve this problem.

Electric violin The use of the electric violin as a controller has also been briefly investigated. Like the human voice and the theremin, the violin is continuous and mostly monophonic, which makes it a good candidate as a vocal synthesis controller. Compared to the acoustic violin, the electric violin features two advantages: the acoustic sound that it produces, which might be a disturbance when controlling a synthesizer, is greatly reduced, and the output is picked up directly at the strings, thus sparing the need for an external microphone.

The sound output of a violin, be it acoustic or electric, is more complex than that of a theremin, making pitch detection harder to achieve, and indeed, our preliminary test with an electric violin have featured many pitch detection errors, which made using the violin as a controller impractical. Possible ways to improve the process are use of a more performant pitch tracking algorithm, and augmentation of the violin to track pitch by other means than the sound output [95]. Going the augmented instrument route could also allow the motion of the bow to be tracked for rhythm control.

Graphical interface substitute

Figure 2.3: *The Korg Nanokontrol*

For all its versatility, using a non-tactile graphical interface is not ideal in a live setting. Use of the mouse requires a combination of serenity and dexterity that are more easily found in a calm office than on a stage during a live performance. To use the graphical interface as little as possible during live performances, an additional control interface, the Korg Nanokontrol, can be connected to Voks. It is a MIDI control interface with buttons, sliders and knobs.

The Nanokontrol is used as a Swiss army knife to perform several tasks relatively easily:

- Setting the value of various parameters, notably timbre parameters;
- Recalling presets, which comprise the index of a text, a playing mode, and the value of the parameters.

In addition, a button of the interface can also be used as an alternative syllabic rhythm controller (see the paragraph "Syllabic control interfaces").

2.2.3 File management and presets

The first rehearsals for public demonstrations of Voks have made evident an important difference between the task of playing an isolated segment of music or speech, for instance for the purposes of shooting a section of a demo video, and playing several pieces in a row, as is the case in a concert.

During a concert, several sections can usually be heard successively, featuring diverse textual material, possibly several distinct playing modes and voice transformation effects. In a controlled environment such as in front of a camera in our laboratory, source files and labeling files can be loaded manually from disk, and playing modes and voice transformation parameters can be set one by one. The time taken by those tasks is of little importance, and mistakes made by performing them, such as loading the wrong file or selecting the wrong playing mode, are not critical and can be corrected without damage. A concert setting is much less forgiving, and performers have to be able to quickly and accurately switch from one section to the next, a task which performance-related anxiety is likely to make all the more difficult.

To spare performers the hassle of manually performing a series of tasks every time they wish

to change what or how they play, Voks has been endowed with a two-step preset system.

File selection

To play an utterance with Voks, one first needs to load a number of files from disk (those containing the audio data, and the labeling file if playing in syllabic mode). To easily switch between those, we require that a text file be provided which contains all of the information about the location of the input files. The text file, which we will call the *paths file*, is to be written in a custom, human-readable format.

The paths file syntax is as follows:

- On the first line, the total number of utterances is given, along with a path prefix that will be prepended to all file paths.
- Blocks of four lines follow: a title, followed by the file paths for the audio and labeling data, relative to the prefix provided on the first line.

When a paths file is loaded, all its title strings are extracted and displayed in a list on screen, whose elements can be selected either with the mouse or using the preset system described in the next paragraph, prompting loading of the corresponding files.

Preset management

The state of most parameters that can be selected in the graphical interface, included timbre parameters, rhythm mode, and position of the utterance in the utterance list, can be saved and recalled reasonably easily using *presets*, using the following mechanism.

The graphical interface features a grid, whose every square is a preset slot that can be bound to a preset. Binding the current state of the parameters to/recalling the state from, a preset, is done by shift-clicking (respectively, clicking) on the corresponding slot. Bindings between slots and presets are not saved upon quitting and reopening Voks; however, two dedicated buttons on the graphical interface allow for saving/recalling those in a file on disk.

This description might make it seem that a source of complexity, that related to loading audio data and labeling data files, as well as setting the value of parameters, has been replaced with another source of complexity of the same nature, that of loading a preset file and selecting the adequate preset. The crucial difference lies in the fact that with presets, the complex task (selecting a file from disk) happens only once at the beginning of a playing session, whereas without presets, one would have to load files and set parameters at every change of playing mode or text.

To further simplify the performer's task, each preset slot has been mapped to a button on the Nanokontrol, in such a way that pressing one of the preset buttons on the Nanokontrol has the same effect as clicking the corresponding slot in the graphical interface. That way, in a performance, the graphical interface does not have to be interacted with past the initial setup.

2.3 Rhythm control

Voks is based on the principle of *sampling synthesis*, that is, synthesis based on existing audio samples[1].

2.3.1 The time index

In this section, we introduce the notion of *time index*, an abstraction that proves useful when attempting to describe methods for time control in sampling synthesis. We will first deal with the case where synthesis is based on one source signal, introducing the framework and then the notion of time index itself, before generalizing to several ones.

Formal framework

Like any physical phenomenon, the production of sound takes place in time, and dealing with a sound signal mathematically warrants the use of a real variable t to denote time. The ability to record and play back sound complicates this situation. Playing back a recorded sound warrants consideration of another time, that during which the recording was made.

Given that in sampling synthesis, a recording can undergo an analysis step, whose result is subject to manipulation, followed by a synthesis step, we will refer to the time of playback as *synthesis time*, and to the time of recording as *analysis time*. To differentiate between both, notations pertaining to analysis time will involve the Latin letter τ, as is customary, whereas notations pertaining to synthesis time will involve the Greek letter t.

Definition of the time index

Using that formalism, designing a method for rhythm control in sampling synthesis amounts to solving the following problem:

> Find a function $\tau(t)$ from synthesis time to analysis time, that indicates, at each point in time, what section of the source signal to base synthesis on.

We shall call the function $\tau(t)$ the *time index*.

The highlighted sentence above is a general statement of the rhythm control problem. Depending on the purpose of the rhythm control method, different constraints may be imposed on the time index. For instance, to avoid sudden changes in timbre, or even artifacts such as clicks, one may wish to require $\tau(t)$ to be continuous. Another example is requiring that $\tau(t)$ be non-decreasing, to rule out reversed sound output.

In Voks, as in most cases of sampling synthesis, the time index $\tau(t)$ is not wholly determined in advance, but depends on input being fed in the synthesis time (as noted in [73]) — typically, on data from control interfaces. To account for this, we slightly modify the definition of the time index so that it depends on the value of the control signal(s) at all times. Rigorously speaking, the

[1] In the World version of Voks, the samples are loaded not as audio data, but as a previously computed spectral-domain representation. This does not change the fact that what is represented is ultimately an audio sample, albeit .

time index should then be notated $\tau(t, c_1(t), \ldots, c_n(t))$, where $c_1(t), \ldots, c_n(t)$ are control parameters. For the sake of clarity, however, we shall leave the dependence on the $c_i(t)$ implicit.

Note that this approach can easily be adapted to the case of synthesis based on several recordings, by replacing the single function τ by a set $(\tau_i)_i$ of function indexed by the set of recordings. This makes it possible, for instance, to resynthesize several samples at the same time.

2.3.2 Basic rhythm control methods

Using the formalism of the time index, we can describe the rhythm control methods used in Voks. The main method is the *syllabic method*. Two other, more basic modes are available, *scrub mode* and *speed mode*. All three methods are available through Voks' graphical interface.

Scrub mode

The most basic method for controlling rhythm is the one where the time index is controlled directly. That is, we take as input a control signal $p(t)$ in the interval $[0, 1]$, and the time index is given by the equation:

$$\tau(t) = T \cdot p(t) \tag{2.1}$$

where T is the length of the source sample.

This method provides a very direct access to playback of the source sample: keeping the parameter $p(t)$ constant freezes sound, increasing $p(t)$ plays the source sample with the corresponding speed, and playing backwards in time is possible by simply decreasing $p(t)$.

Speed mode

Speed mode is the first-order counterpart to scrub mode seen as a zeroth order method. It still takes as input a continuous parameter $v(t)$, but that parameter is used to control speed instead of position:

$$\tau(t) = \int_0^t v(x)dx \tag{2.2}$$

Note that the speed parameter $v(t)$ can be negative, in which case sound will be played in reverse.

This basic formula suffers from a few issues. The following paragraphs describes those and presents methods for solving them.

Bounding the time index in speed mode

When in speed mode, depending on the evolution of the control parameter $v(t)$, the time index $\tau(t)$ may very well exceed the bounds of the interval $[0, T]$, outside of which the behavior of the synthesis algorithm is undefined.

There are three natural solutions to this problem, illustrated in figure 2.4:

Method 1 Only use the actual value of the time index when it is within bounds. Otherwise, use the value of the bounds:

$$\tau_1(t) = \max\left(\min\left(\int_0^t v(x)dx, T\right), 0\right) \tag{2.3}$$

Method 2 Keep the time index from changing when it tries to get out of bounds:

$$\tau_2(t) = \int_0^t \bar{v}(x)dx \tag{2.4}$$

where, for all $t \in \mathbb{R}$,

$$\bar{v}(t) = \begin{cases} \max(v(t), 0) \text{ if } \tau_2(t) \leq 0 \\ \min(v(t), 0) \text{ if } \tau_2(t) \geq T \\ v(t) \text{ otherwise} \end{cases}$$

Method 3 Wrap the time index between bounds:

$$\tau_3(t) = \left[\int_0^t v(x)dx\right] \mod T \tag{2.5}$$

Of those three solutions, let us explain why the first one is problematic. Let us consider the case of a performer controlling rhythm by direct control over $v(t)$. The performer can adapt their playing by paying attention to the audio feedback they are hearing; in fact, due to the very general problem of accumulating error when performing an integration, not doing so would inevitably lead them to making an arbitrarily large error on the value of the time index. With the first bounding method, this is essentially what happens: when the unbounded version $\tau(t)$ of the time index (the dotted lines on figure 2.4) is out of bounds, the sound is frozen, and the performer has no indication as to when $\tau(t)$ will be back in the bounds.

In contrast, when using method 2, when the upper (respectively, lower) bound has been reached, the performer knows they just have to make $v(t)$ negative (resp. positive) to unfreeze the sound. When using method 3 ($\tau_3(t)$ on figure 2.4), the performer has full knowledge of the value of the index *up to an offset that is a multiple of* T, which is sufficient, as in this case, the synthesis itself is invariant under addition of any multiple of T.

Put differently, method 1 is not satisfying insofar as knowledge of the current value of the (bounded) time index is not sufficient to determine how the system will behave in the future.

Methods 2 and 3 each have their use cases. Method 2 features two equilibrium positions, namely, the two bounds 0 and T; that is, it is easy to stop at one of those positions by keeping $v(t)$ less than (respectively, greater than) 0. Those may help with control by providing a fallback whenever the performer has trouble stabilizing the time index at a given value. As a counterpart, once at the end of the sample, the sound freezes, and the only way to keep playing is to play the end of the sample in reverse. The time index is at a dead end.

In contrast, method 3 ensures that sound will never freeze as long as $v(t)$ is nonzero. This is particularly useful with short samples, which can be played repeatedly by keeping $v(t)$ constant

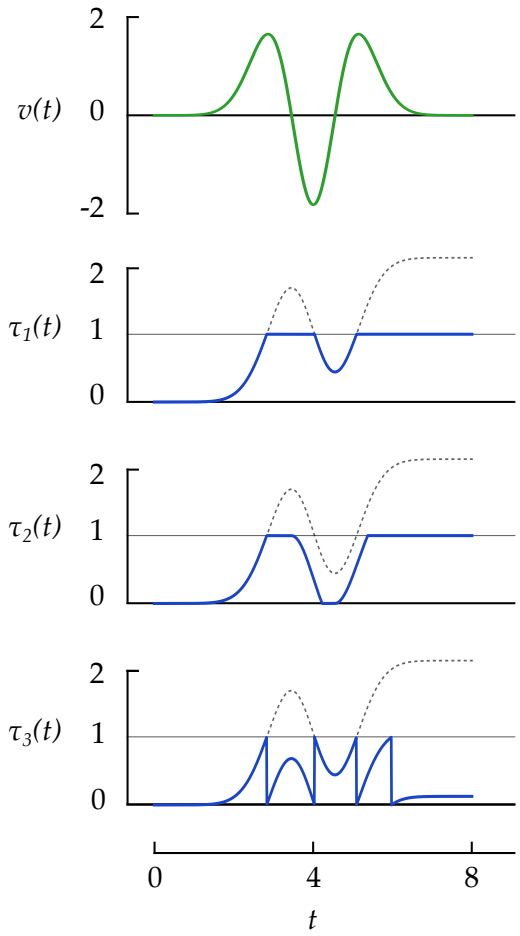


Figure 2.4: *Illustration of the three bounding methods. In green, a function used as speed (for illustration purposes; not based on an actual gesture); in blue, the resulting time indices when using each of the three bounding methods. The unbounded time index $\tau(t)$ has been superimposed in dotted lines for reference.*

or quasi-constant (meaning its variations are slow compared to the length of the source sample), leading to potentially interesting effects.

A way to combine some of the advantages of method 2 and method 3 is to use method 2, but add another, discrete input — typically the output of a button, in the form of impulses. Each time an impulse is received, playback starts back at the beginning. This solves the dead end issue mentioned above, although it still does not allow looping. Thus even with this modification to method 2 available, method 3 remains interesting in its own right.

Adaptation function

Two other issues are associated with speed mode:

1. Should a performer wish to freeze sound, i.e. make the time index constant, for some given amount of time, they have to keep the value of $v(t)$ exactly zero during that time. Using a continuous gestural control interface, it is difficult, if not impossible, to reach and maintain a specific control value. Thus in equation 2.2, the integrated term is close, but not equal to zero, leading to accumulation of error, unless the performer carefully keeps monitoring the time index and correcting the control value accordingly.
2. In general, a same change in speed will be more significant the smaller the speed is: gestures to go from a speed from 0 to 1 or to go from a speed of 10 to a speed of 11 involve the same distance, but the former is much more noticeable.

Those problems can be solved by using an alternate mapping $a(v)$ from the control parameter $v(t)$ to the actual playback speed:

$$\tau(t) = \int_0^t a(v(x))dx \tag{2.6}$$

Issue 1 warrants a mapping which reaches 0 on an interval, whereas issue 2 warrants an exponential mapping. Since the only exponential that reaches 0 on an interval is the zero function, a tradeoff has to be made. On top of those requirements, we also look for a function that reaches negative values, so as to make it possible to play sound backwards.

The hyperbolic sine function

$$\sinh : x \mapsto \frac{e^x - e^{-x}}{2} \tag{2.7}$$

is asymptotically equal to an exponential, and being an odd function, it reaches negative values too, with control for negative speed behaving the same as for positive speeds. However, it equals zero only on a point. We thus use a modified version of the hyperbolic sine that reaches 0 on an interval:

$$a(v) = \begin{cases} \sinh(v - A) - (v - A) \text{ if } v \geq A \\ \sinh(v + A) - (v + A) \text{ if } v \leq A \\ 0 \text{ otherwise} \end{cases} \tag{2.8}$$

where A is half the width of the interval where the speed should be equal to zero. The graph of this function is pictured in figure 2.5.

The $(v - A)$ / $(v + A)$ term ensures that the function approaches 0 smoothly (i.e. the function is differentiable at the ends of the zero interval). This allows the performer to play finely at small speeds.

Interaction with intensity control

A rhythm control method may be combined with other methods for controlling sound, ranging from simple sound level control to complex pitch and/or timbre modifications. In some cases, such modifications can lead to the result of synthesis being temporarily inaudible. Typically, one would expect a synthesis system featuring an "intensity" control parameter to not play any sound whenever that parameter goes to 0.

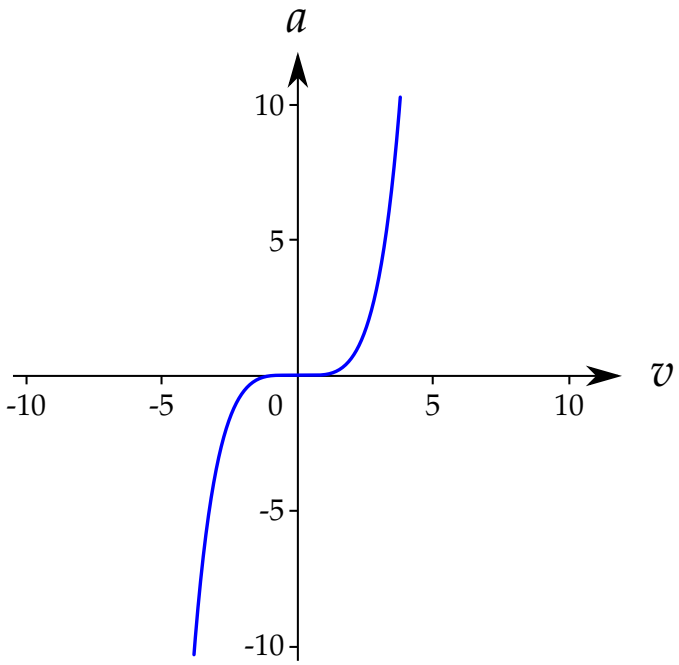


Figure 2.5: *The adaptation function defined by equation 2.8. Here the value of the parameter A is 0.5, meaning there is an interval of length 1 where the function equals 0.*

When in speed mode, this lack of auditive feedback will make the performer lose track of the absolute position of the time index when intensity is at its minimum. To counter this, the value of the speed control parameter can be artificially set to 0 whenever the parameters for sound manipulation make the sound inaudible. That way, the performer knows where in the source sample they stand upon restarting of the sound.

2.3.3 Syllabic mode

Syllabic mode is the main playing mode of Voks. It is based on the frame/content theory of speech production [78], which postulates that speech is organized around a cyclic opening/closing motion of the jaw inherited from the act of mastication. That cycle is viewed as a *frame* for segmental events, the *content*, to take place. The frame is then responsible for the organization of speech around a succession of syllables.

In light of the perspective of the frame/content theory, Voks' syllabic control can be described as a type of rhythmic control where the performer controls the frame, the temporal organization of the segmental content being determined as a consequence of that of the frame.

To control the rhythm of some sample with a frame-based method, one first needs the frame of the source sample to be identified. In practice, this is done through a *labeling* process, detailed in section 2.4; the result of the labeling process is a discrete list of temporal anchors $(\tau_i)_{i\in\mathbb{N}}$ — two for every cycle of the frame — named *control points*. As for the performer's gesture, it is assumed to be a discrete sequence of impulses $(t_i)_{i\in\mathbb{N}}$, such as key or button presses and releases (possible

interfaces for syllabic control are detailed in section 2.2.2), which we choose to regard as events resulting from a cyclic process (the repeated motion of the hand) analogous to that of the jaw when speaking.

In that context, the problem of controlling rhythm can then be rephrased in the following way:

How can the frame of the source sample, whose structure is made apparent by the control points $(\tau_i)_{i\in\mathbb{N}}$, be put in correspondance with the motion of the hand, which also has a frame-like structure made apparent by the sequence of discrete impulses $(t_i)_{i\in\mathbb{N}}$?

If the moment of occurence of each t_i were known in advance, the problem would be easily solved by setting the time index τ to be the continuous, piecewise linear function that reaches the control point τ_i at each impulse t_i:

$$\tau_{\text{ideal}}(t) = \tau_i + \left(\frac{t - t_i}{t_{i+1} - t_i}\right)(\tau_{i+1} - \tau_i) \tag{2.9}$$

where i is the index of the greatest impulse time smaller than or equal to t:

$$t_i \leq t < t_{i+1}$$

The trouble here is that the value of any t_i is only known when $t \geq t_i$. In the absence of any hypothesis on the t_i (such as a lower bound on the duration between two consecutive t_i), if each target τ_i were to be reached exactly at time t_i, this would put the time index $\tau(t)$ at unavoidable risk of discontinuity; in turn, discontinuity of the time index would make for unphysical-sounding synthesis.

To mitigate or deal with that risk, several strategies are possible:

- Switch from a discrete control input method to a continuous one;
- Try to predict the t_i before they occur;
- Aim to reach each target τ_i fast enough once the impulse at t_i has been received. This is the method currently in use in Voks.

Continuous input method

In principle, reaching the targets in time without any discontinuity of the time index can be achieved by replacing the series (t_i) of discrete impulses by a continuous parameter $\mu(t) \in [0, 1)$. $\mu(t)$ can be seen as a continuous generalization of the (t_i) which reaches 0 each time that a t_i would have previously happened, and increases continuously towards 1 before jumping back to 0 at t_{i+1}.

Given such a continuous input parameter, the time index is given by the equation:

$$\tau(t) = \tau_i + (\tau_{i+1} - \tau_i)\mu(t) \tag{2.10}$$

with i corresponding to the number of times that μ has reached 0 up until now.

The difficulty with considering such an input parameter is that with many continuous interfaces, extremal positions are usually not reached. In Vokinesis [30], the data coming from continuous interfaces is converted into what we call here the $\mu(t)$ parameter, using dedicated algorithms specific to the type of continuous interface:

- With a graphic tablet, the vertical position of the stylus is clipped in such a way that only positions of the stylus in a band the middle of the tablet give values of $\mu(t)$ strictly between 0 and 1; when the stylus gets outside of that zone, $\mu(t)$ is constant and equal to 0. This ensures that extremal positions can be reached, but the occasional constancy of $\mu(t)$ can be deemed problematic.
- A solution is proposed that requires two potentiometers: an algorithm ensures that $\mu(t)$ is never constant provided that one of the potentiometers is always in motion.
- The shape of a hand can also be taken as input: the hand is filmed and converted into a continuous parameter using algorithms trained specifically for that task.

Use of a cyclic parameter, such as the angle of a joystick, is also a possible strategy, which has been implemented in MonoReplay [77].

However, from our experience, continuous interfaces tend to require a compromise between speed, which can be achieved with small gestures, and accuracy, associated with large ones. The search is still under way for a satisfying combination of a control interface and an algorithm for converting its output into a continuous parameter $\mu(t)$, that would be both fast and accurate enough. Two strategies are being explored:

- Development of an algorithm that takes data from the Touché interface as input, a kind of hand potentiometer with a distinctive feel;
- Use of the angle coordinate of the stylus position on a circular sector (approximately, a triangle) of a graphic tablet, which would allow for both small gestures, with a small radius, and larger ones, with a large radius.

Future t_i estimation

The series of impulses (t_i) is generally not entirely random, and depending on the type of audio that they correspond to, the value of a t_i might be estimated from the previous $(t_k)_{k<i}$.

In the case of metered music, with (t_i) corresponding to beats, the hypothesis of an approximately constant beat duration can be made to estimate the next t_i. That is the strategy adopted in MonoReplay, [77] where one can choose between several playing modes; each playing mode comprises an algorithm for estimating, at each t_i, the time of occurence of the next beat t_{i+1}:

- Assuming a constant inter-impulse duration: $\hat{t}_{i+1} = t_i + T$, where T is a predefined constant.
- Assuming the inter-impulse duration to be proportional to the corresponding inter-label duration: $\hat{t}_{i+1} = t_i + \frac{1}{v}(\tau_{i+1} - \tau_i)$, where v is a predefined constant.
- More generally, assuming the inter-impulse duration to be some function of the control points and the previous impulses: $\hat{t}_{i+1} = f(t_1, \ldots, t_i, \tau_1, \ldots)$

In the case of syllabic speech and singing synthesis control, the situation is not as simple, as the syllables cannot be assumed to be of a similar length. In the case of singing, a system that takes a score as input may be able to estimate the moment of occurence of the next syllable. For speech, perhaps complex algorithms could be able to predict it too. Such techniques, however, have not been implemented, and Voks is based on the last strategy:

Waiting for the impulses

Rather than trying to predict the t_i, Voks' rhythm control method accepts the fact that it does not know when they will occur, and deals with that situation by advancing the time index fast enough that the induced delay is tolerable. The problem becomes that of choosing a curve to connect the point $(t_i, \tau(t_i))$ to a point $(t_{\text{target}}, \tau_{i+1})$ (where t_{target} is to be chosen), with the following constraints in mind:

Limitation of delay t_{target} should be reasonably close to t_i as possible; in other words, the target control point should be reached soon after the control impulse;

Physical sound The derivative of $\tau(t)$ should not be exceedingly greater than 1; in other words, the time index should progress at a rate close to the default one, so that the synthesis does not suffer from audible jumps or near-jumps.

Those two constraints conflict: forcing t_{target} to be too close to t_i, for instance, will force $\tau(t)$ to increase rapidly. Thus the choice of a curve for rhythm control is the result of a compromise. Let us now propose several options for such a curve. With each of those options, the value of the target t_{target} directly depends on a user-defined *speed parameter* v via the equation $t_{\text{target}} = t_i + \frac{1}{v}$.

The first choice, that which is represented on figure 2.6, is for the curve to be linear:

$$\tau(t) = \begin{cases} \tau_i + \frac{t-t_i}{t_{\text{target}}-t_i}\left(\tau_{i+1} - \tau_i\right) \text{ if } t \leq t_{\text{target}} \\ \tau_{i+1} \text{ otherwise} \end{cases} \tag{2.11}$$

where i is the index of the latest impulse before t: $t_i \leq t < t_{i+1}$.

Note that such a curve can cause issues: if the impulse t_{i+1} is received before $\tau(t)$ has reached τ_{i+1}, equation 2.11 leads to discontinuity of the time index (the vertical segment on figure 2.6 is an example). An example of this phenomenon can be observed on figure 2.6, at around $t = 0.7$s, where the blue curve is vertical. To avoid such a situation, the linear progression can be made to start from the current value of the time index $\tau(t_i)$ rather than the last target τ_i:

$$\tau(t) = \begin{cases} \tau(t_i) + \frac{t-t_i}{t_{\text{target}}-t_i}\left(\tau_{i+1} - \tau(t_i)\right) \text{ if } t \leq t_{\text{target}} \\ \tau_{i+1} \text{ otherwise} \end{cases} \tag{2.12}$$

When the estimated impulse times (t_i) are consistently close to each other — that is, the performer is controlling rhythm fast —, it can happen that the time index takes too long to reach its destinations (τ_i). In those cases, it can be beneficial to replace the linear progression with a convex

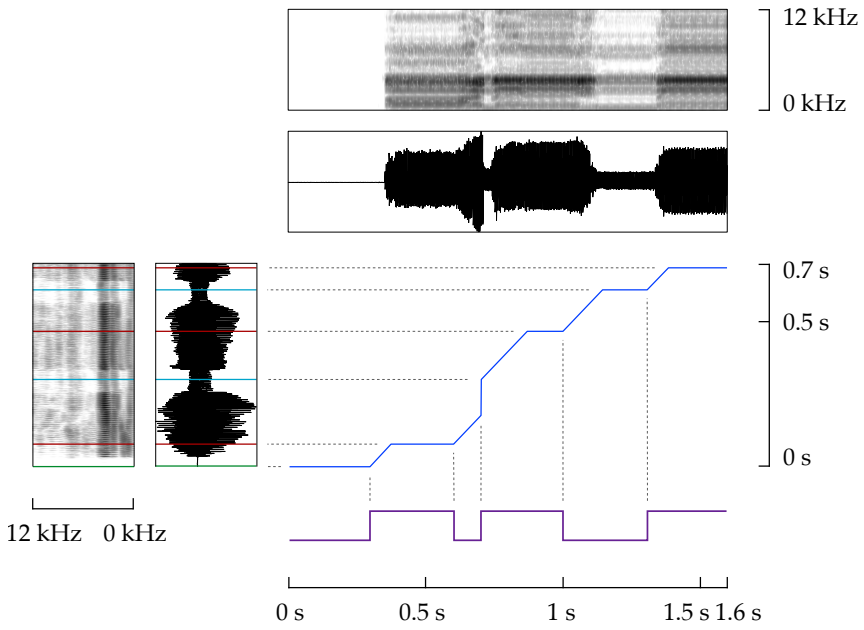


Figure 2.6: *Plot (in blue) of the time index as a function of time. Control impulses are taken to happen at times where the value of a binary control parameter, in purple, changes. The time index turns the labeled source sample, pictured vertically on the left, into the synthesized sound pictured horizontally at the top. The time index features horizontal segments, where the output sound is frozen, oblique ones, where the sound advances linearly, and a vertical segment, that is, a discontinuity of the time index that can be heard in the output sound as an effect similar to that of a skipping record.*

curve[2], one whose slope is greater at the beginning than at the end. A simple choice is that of a parabola:

$$\tau(t) = \begin{cases} \tau_{i+1} + \left(\frac{t_{\text{target}} - t}{t_{\text{target}} - t_i}\right)^2 (\tau(t_i) - \tau_{i+1}) \text{ if } t \leq t_{\text{target}} \\ \tau_{i+1} \text{ otherwise} \end{cases} \tag{2.13}$$

This last equation corresponds to Voks' rhythm control algorithm.

To conclude this description of syllabic control, let us note that although the method described here aims to put in correspondence control impulses and control points, the relation between those is not obvious, as evidenced by the relative complexity of equation 2.13 — owing to the fact that the system does not know in advance when the user will interact with the rhythm control interface.

[2]The everyday acceptation of the word "convex" conflicts with the mathematical definition of a convex function, which denotes a function f with the property that any point p on its graph is always *below* any segment whose endpoints lie on the graph of f and surround p. Here we use the everyday convention, with "bumps" being convex, and depressions being concave.

2.4 Labeling

Section 2.3 describes methods to control the rhythm of an audio sample based on the syllabic unit. To let the rhythm control algorithm know about the syllabic structure of the source sample, syllabic markers, known as *control points*, are communicated to it by means of a text file. The text file is composed of a series of numbers representing timestamps of the control points in the source sample.

Depending on the control interface used (see section 2.2.2), one or two control points may be needed for each cycle, one corresponding to presses or taps and one corresponding to releases. With one event/control point per syllable, control is said to be *monophasic*; with two, control is *biphasic*.

In the biphasic case, one type of control point corresponds to vocalic nuclei, and one to consonantic transitions, following [30]. In the monophasic case, only vocalic control points are used. Thus a biphasic labeling can be used both in the monophasic case, by ignoring consonantic points, and in the biphasic case. Therefore, we label all samples biphasically by default.

2.4.1 General heuristic for locating control points

The method for choosing the exact timestamp of the control points is a heuristic one. Control points are placed with the following considerations in mind, in order of priority:

1. Given that the algorithm for rhythm control is likely to have sound freeze at a control point, we require that control points be located on *stable* zones of the signal, that is, zones without sudden changes in spectral content. This aims to ensure that the synthesized sound remains physical-sounding, that is, that the synthesis algorithm does not introduce spectral stagnation in a zone that did not originally feature any.
2. For the same reason, control points should be placed in a zone where it would not feel too unnatural for a speaker/singer to sustain the timbre. Here the labeler's judgement is required.
3. The rhythm control algorithm is subject to inertia caused by the delay to go from one control point to the next. Thus we aim to place control points as late as possible.

Based on those considerations, the heuristic for placing control points is as follows: *control points should be placed as far on the right as possible, while remaining in a spectrally stable part of the signal.*

This consideration is applicable for both vocalic and consonantic control points, but the practical implications are different depending on the type of phone. In the next sections, we detail the rules that we, as labelers, subject ourselves to when labeling a voice sample, but before, let us make a remark regarding a detail in the above general rule.

A practical remark

Even though our general heuristic rule states that the point chosen should be as far on the right as possible, we still leave a little room on the right of the control point, in the order of a few

hundredths of second, before the transition from one diphthong to the next. This aims to keep the subsequent phone from interfering: as mentioned above, the rhythm control algorithm is likely to freeze at control points; in practice, no matter what signal processing technology is employed, freezing is achieved by considering a small window around the control point. If the point is placed on the exact boundary between phones, the window around the point will contain signal from both phones, hence the left shift in control point location.

2.4.2 Choosing where to place control points

In this section, we give guidelines for placing control points by considering the different types of possible sounds. Those guidelines are based on the above heuristic rule and our experience. As such, they are mostly applicable to French and English, the main languages with which we have been dealing, and to a lesser extent, Italian and German.

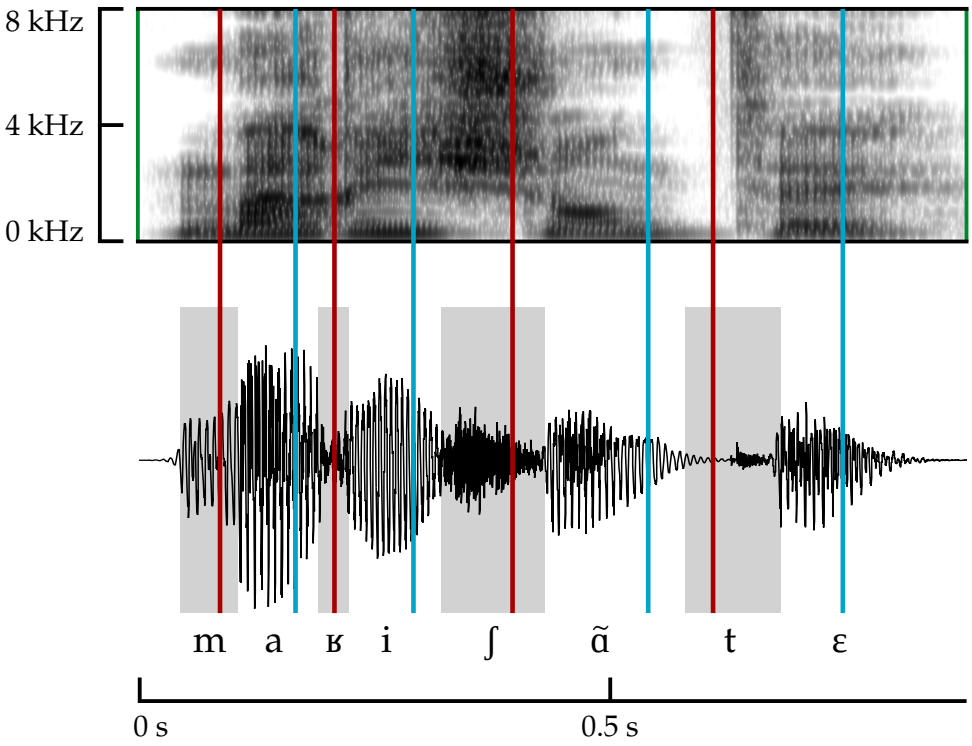


Figure 2.7: *Labeling of the French sentence* Marie chantait, *whose spectrogram and waveform are represented. Vocalic control points are in dark red; consonantic control points, in cyan; and boundary control points, in dark green. To stress the distinction between control points and phone boundaries, phones are represented on the waveform as contiguous zones with alternating gray and white backgrounds.*

Monophthongs

For vocalic control points, in the case of monophtongs, the "stable zone" referred to in the above heuristic is simply the whole vowel. Thus we place the control point on the right end of the

vowel. This is illustrated multiple times in figure 2.7, for instance with the control point in the /i/ phone.

Diphthongs

In the case of diphthongs, there is no longer a single stable zone, but a spectral transition between two vowels, possibly starting from, or ending on, a stable zone if the syllable is sustained. In that case, we chose one of the vowels. To do so, the labeler must rely not on an explicit algorithm, but on their intuition and artistic vision, and ask themselves the following question: *on what vowel should the sound freeze, should the syllable be sustained?* Once a vowel has been chosen, if there is a zone of the signal where the spectrum is static, place the control point at the right of the zone; otherwise, choose the point where the spectrum corresponds most to the chosen vowel.

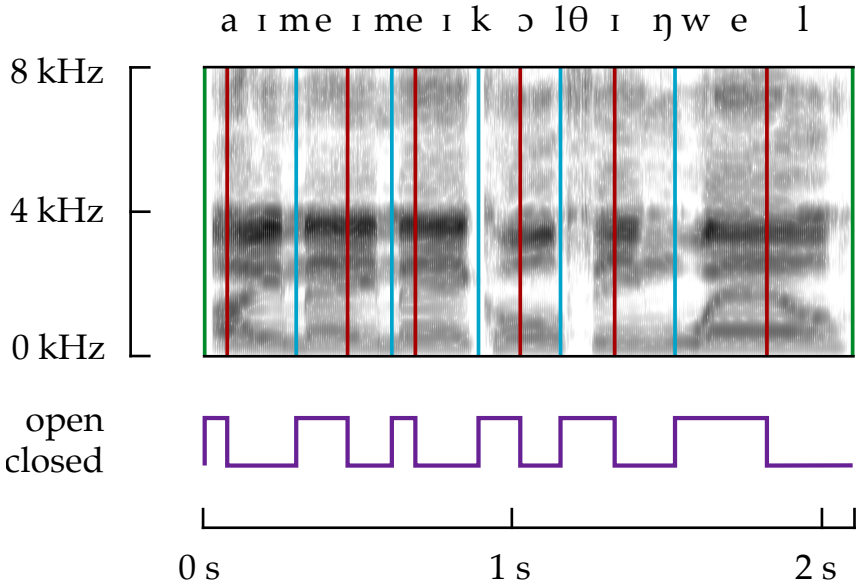


Figure 2.8: *Visual representation of the labeling of an audio recording of the English sentence* I may make all thing well. *Vocalic control points, in dark red; and consonantic control points, in cyan, are superimposed on a spectrogram of the audio sample. Additional points, in dark green, are placed at the extremities of the sample. The purple graph represents the position of the controller corresponding roughly to the state of a controller when controlling rhythm using the displayed labeling.*

An illustration of the method can be found in figure 2.8, where the first vowel of the diphthongs /aɪ/ and /eɪ/ have been chosen. In that example, the vowel /a/ in the diphtong /aɪ/ is not held, thus the control point is placed at the point that sounds closest in timbre to that of an /a/, as judged by the labeler. On the contrary, the first instance of the diphthong /eɪ/, in the word "may", features a spectrally stable zone during which an /e/ is sounded, at the right of which the control point is placed.

Larger vowel combinations

Although triphtongs are exceedingly rare in English and French, the procedure for diphthongs can also be used for triphthongs, and more generally for syllables with any number of vowels.

Nasals and fricatives

From our experience, nasal consonants are the easiest to label, as they generally feature clearly defined endpoints between which the spectrum is stable. Here the procedure is the same as that for sustained vowels: place the control point on the right of the stable zone. Although their boundaries are less clearly defined, fricatives are also composed of a spectrally stable zone at the end of which control points are to be placed.

Liquids and semivowels

In most of the languages that we considered, liquids and semivowels are generally composed of a fast change in spectral content. The control point is then placed in the middle of that change.

Plosives

Plosives feature a relatively spectrally stable zone, followed by an abrupt transient. The control point is placed at the end of the stable zone. Plosives are best labeled by looking at the sample waveform, to see the precise location of the transient. Labeling of a plosive can be seen on figure 2.7 in the /t/ phone.

Trills

One text in Italian has been labeled; it features the trill /r/. On a signal, trills manifest as a very short repeated sound. Thus to "freeze sound" on a trill in such a manner that the resulting signal would sound physical, that short sound would need to be repeated some more. Our syllabic algorithm is not able to perform such a repetition — only to keep the spectral envelope constant. We work around the problem by placing the consonantic control point in the preceding vowel, as if there were no consonant (see the rule for absence of consonant below). The result is impossibility to stop during a trill; if the performer attempts to stop regardless, the sound will freeze at the preceding vowel.

Consonant clusters

The question of consonant clusters is the same as that of diphthongs. If a control point is to be placed in a consonant cluster, the labeler must exercize their judgement to decide which consonant they would most like sound to freeze. Once the consonant has been decided, the rule corresponding to the corresponding consonant can be chosen in the ones listed above.

Absence of consonant

When two vowels belonging to different syllables are in succession without a consonant in between, control is effectively locally monophasic, that is, the biphasic component of the gesture (typically, release of a key or button) is to be ignored for that syllable. This is achieved by placing the consonantic control point at the same location as the preceding vocalic control point.

2.4.3 Boundary points

In addition to the standard control points, two markers can be placed at the endpoints of the considered utterance. These will determine what the initial value of the time index $\tau(t)$ is, and what value it should stop on. These do not need be specified — if they are not, Voks will default to the beginning and end of the sample —, but specifying them makes it possible to isolate a segment from a longer recording.

2.4.4 Labeling in practice

```
-1 0.0
0 0.100
1 0.215
0 0.275
1 0.380
-1 0.529
```

Figure 2.9: *Example labeling file. The lines starting with -1 denote boundary points; those starting with 0 denote consonantic control points; those starting with 1, vocalic ones.*

Labelings are to be stored in a text file with a specific, simple syntax (see figure 2.9), with one control point one each line. The labeling can be done by simply opening the sample in an audio editor such as Audacity, placing the cursor on the desired control point location, and manually copying the corresponding timestamp in the text file, but the process can be automated without too much hassle. The exact automation process depends on the specifics, but as an example, with Audacity, it is possible to create, in parallel to audio tracks, *marker tracks*, where points can be placed. The timestamps of all points in a marker track can then be exported into a text file. Naturally, the syntax of that file is not the same as that required by Voks, but it is easy to write a script to convert from one format to the other.

A basic feature of audio editors has proven useful when labeling in practice: the ability to loop sound. Looping a very short segment of sound, although it introduces clearly audible artifacts, provides the labeler with a helpful approximation of what a sound freeze in that segment would sound like, considering that the time index might stop at control points.

2.5 Synthesis

There are two different versions of Voks, each based on a distinct audio engine. The World vocoder was initially used, but presented some issues, after which a version based on SuperVP was implemented.

2.5.1 World version

The World version of Voks is based on two software bricks that we developed based on the World library [87].

The first software brick performs the World analysis of a given audio signal. Since, in World, the analysis phase does not take place synchronously, this needs not use any real-time technology. Thus it just consists in an executable which loads a given audio signal, runs the analysis functions included in the World library, and writes their output on disk.

The file format used for storing the result of analyses is the Jitter matrix format. As explained in what follows, the synthesis is performed in Max/MSP. The Jitter matrix format, native to Max/MSP, is thus a natural way to deal with the analysis data. In addition to loading and feeding the data to the synthesis module, the Jitter format makes it possible to edit data in real-time. This is how the manipulations described in the following section 2.6 are performed in the World version of Voks.

Synthesis is performed by a Max external that we developed. At each point t in time, the computed time index $\tau(t)$ is used to select a frame in the Jitter matrices representing the analyzed source sample; that frame is fed to our Max external, which uses the synthesis function from the original World library to generate audio in real time.

Issues with the World engine

The choice of using World as a vocoder was made as an alternative to VRT-PSOLA, the modified version of RT-PSOLA used in Voks' predecessor, Vokinesis. VRT-PSOLA suffered from occasional sound quality issues, namely audible clicks, which made it ill-suited to use in live contexts. There were no such issues with our modified version of World, and the overall sound quality was informally judged to be satisfying.

However, our version of World suffers from other shortcomings. The main issue is related to a conflict between our goal of designing a gesture-controlled synthesizer, and the original design of World, which was not intended to be used in real time. This difficulty was worked around somewhat by packaging the analysis algorithm into a standalone executable, and modifying the synthesis algorithm so that it could be integrated into a real-time Max patch. Nevertheless, the induced workflow remains cumbersome in a real-time setting.

2.5.2 SuperVP version

The SuperVP version relies on a single Max external, `supervp.scrub~`, from the Max/MSP SuperVP package. This external directly takes as input the time index $\tau(t)$ and synthesizes the

audio corresponding to $\tau(t)$ based on a sample loaded beforehand.

Since the external does not expose its internal representation, the manipulations described in the following section 2.6 are either:

- Applied using the parameters that the SuperVP external does expose, when possible;
- Applied to the audio signal output by the SuperVP external after synthesis, when possible;
- Or not performed at all.

2.5.3 Comparisons of the two vocoders

Both SuperVP and World offer a comparable, satisfactory sound quality. World's strength lies in the direct access to the spectral representation from within Max. With SuperVP, any timbre modifications must be performed either through the setting of a limited number of parameters, those made accessible through the Max interface, or directly on the synthesized audio.

SuperVP prevails when it comes to usability. Our version of World requires steps that seriously hinder the artistic process: performing an analysis using a separate executable tool, loading the result of the analysis in Voks. Admittedly, this workflow is not entirely inherent to World, and is a consequence of the choices we made while packaging it. However, while improvements to this workflow should not be ruled out, World will always need an offline analysis step.

2.6 Other expressive controls

In addition to rhythm, Voks allows for manipulation of pitch and timbre of the source sample. This section describes those manipulations. Their effect on the produced sound can be visualized on figure 2.10. Note that the specific manipulations differ slightly depending on the audio engine used. This is due to the fact that with World, the intermediate representation of the vocoder is accessible, whereas in SuperVP, we have to resort to direct manipulation of the synthesized signal.

Each one of the modifications is driven by a control parameter, denoted by a different Greek letter. Rules are implemented that convert into those parameters data coming from both gestural control interfaces and the software's graphical interface. This is represented in figure 2.1, with the *Signal modifications* block taking as input those parameters, as well as a stream of World frames, and outputting a stream of modified World frames. Note that this applies only to the version of Voks based on World (in fact, it would apply to any version based on a vocoder giving direct access to the vocoded representation). In the SuperVP version, expressive modifications are obtained by a combination of:

- Providing parameters directly to SuperVP, for those transformations that SuperVP makes accessible through its interface;
- Modifying the synthesized signal.

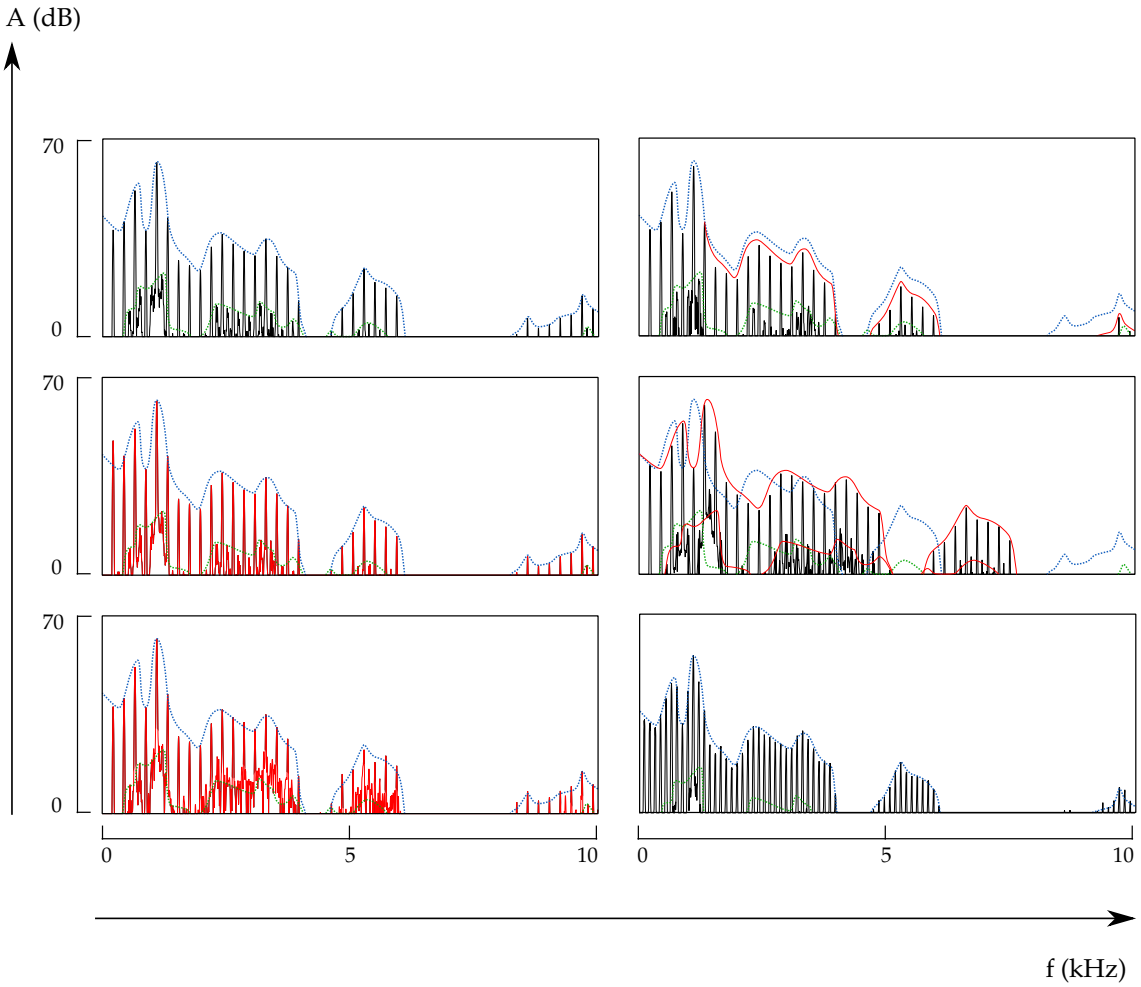


Figure 2.10: *Power spectrum of the vowel /a/ played : (a) without timbre changes, (b) with simulation of a shift in the glottal formant frequency (see section 2.6.2), (c) with amplification of the nonperiodic part by a factor of 10, (d) by applying a spectral slope with a cutoff frequency of 1000Hz, (e) with simulation of a stretching of the vocal conduit by a factor of 0.8, (f) with a fundamental frequency one octave lowerer.*
On figure (a), an envelope, respectively blue and green, has been drawn manually to outline the periodic and aperiodic parts of the signal. The same, unmodified envelopes appear on the other figures. On figures (b) and (c), the deviations with respect to the original envelopes have been marked in red; on figure (d), a new version of the envelope has also been drawn manually in red. On figure (e), in addition to the original envelopes, a version of those that appears that has been stretched by a factor $\frac{1}{0.8}$*. On figure (f), the envelopes of figure (a) remain relevant.*

2.6.1 Melodic control

Control interfaces provide a pitch parameter $\pi(t)$ expressed as a MIDI semitone value; that is to say, it relates to fundamental frequency according to the following equation:

$$f(t) = 440 \cdot 2^{\frac{\pi(t)-69}{12}} \text{ Hz} \tag{2.14}$$

with $f(t)$ the fundamental frequency. Thus going up (respectively down) n equally-tempered semitones amounts to adding (resp. removing) n to $\pi(t)$, with the value 69 corresponding to the reference A pitch at 440 Hz.

With World as an audio engine, fundamental frequency management is easy: fundamental frequency is specified directly to the World synthesis algorithm. When using SuperVP, the situation is slightly more challenging: the `supervp.scrub` object, by default, leaves unchanged the fundamental frequency of the source sample. It is possible, however, to specify a *transposition* value, in cents (hundredths of equally-tempered semitones). Direct pitch control is thus achievable provided that the pitch of the original sample is specified. Voks provides two ways to do so:

- Setting a single value for the fundamental frequency once through the graphical interface. This assumes that the source sample has a single fundamental frequency all throughout. This is achievable either with some preprocessing of the sample, by using a non-real time algorithm to flatten the pitch, or by recording a monotonous sample in the first place.
- If the source sample has a varying fundamental frequency, specify it continuously during synthesis. This is more restrictive, as it requires, at each time during synthesis, to know the pitch of the section source sample being played. A method for specifying a varying pitch remains to be developed.

2.6.2 Timbre control

Glottal formant

To change the tension of the synthesized voice, a shift of the glottal formant is simulated. The glottal formant is a peak in the spectral envelope at frequencies close to the first harmonic [37]. A shift of that formant can thus be simulated by varying the ratio $\frac{H_{>1}}{H_1}$ between the weights of the first and higher harmonics, a quantity introduced by [17] under the name of *harmonic richness factor*, which [42] relates to loudness.

Although neither World nor SuperVP allow direct manipulation of the harmonics, one may access the first and higher harmonics separately in World, by weighting the envelope $E(\omega)$ by "masking functions" that respectively take the values 1 at $\omega = f_0$, where f_0 is the fundamental frequency of the synthesized sound, and 0 at $\omega = nf_0$, for all $n \in \mathbb{N}^*$, and conversely. In practice, we rather use sigmoids: although do not take the value 0 or 1 anywhere, they come close enough for our purposes, and they are very regular.

The equations that define the masking functions we use are thus:

$$E_1(\omega) = 1 - \tanh\left(8\left(\frac{\omega}{2\pi f_0} - \frac{3}{2}\right)\right) \tag{2.15}$$

$$E_{\text{sup}}(\omega) = 1 + \tanh\left(8\left(\frac{\omega}{2\pi f_0} - \frac{3}{2}\right)\right) \tag{2.16}$$

$$= 2 - E_1(\omega) \tag{2.17}$$

The envelope $\left|H'_p(\omega)\right|^2$ of the periodic part of the sound to be synthesized can then be synthesized in real time:

$$\left|H'_p(\omega)\right|^2 = \left((1-\gamma(t))E_1(\omega) + \gamma(t)E_{\text{sup}}(\omega)\right)|H_p(\omega)|^2 \tag{2.18}$$

based on the following data:

- The envelope $|H_p(\omega)|^2$ of the source sample, computed from the corresponding World frame, based on equation 1.3,
- A continuous control parameter $\gamma(t)$, between 0 and 1, that specifies the relative prevalence of the first and the higher harmonics. $\gamma(t) = 1$ corresponds to the case where the first harmonic is muted, $\gamma(t) = 0$, to that where the higher harmonics are muted, and $\gamma(t) = \frac{1}{2}$, to that where all harmonics are left unchanged.

Spectral slope

To change the spectral content at high frequencies, we apply a "spectral slope", that is, a first-order low-pass filter. The control parameter is the cutoff radian frequency ω_c of the filter. It should be noted that this effect may only decrease the energy of the signal, not enrich it.

With World, again, the modification is applied by manipulating the coefficients of the envelope, based on the following equation:

$$H'(\omega) = \frac{H(\omega)}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)^2}} \tag{2.19}$$

With Voks as an audio engine, a first-order low-pass filter with cutoff pulsation ω_c is approximated with a first-order discrete filter, and applied directly to the signal.

Aperiodicity

A notable feature of the World vocoder is its ability to separate signals as the sum of a periodic part, whose envelope we denote by $H_p(\omega)$, and an aperiodic part, whose envelope we denote by $H_{ap}(\omega)$. In Voks' World version, we expose the control parameter α, to balance those two parts:

$$H'_{ap}(\omega) = 2\alpha H_{ap}(\omega) \tag{2.20}$$

$$H'_p(\omega) = 2(1-\alpha)H_p(\omega) \tag{2.21}$$

where

- $H'_p(\omega)$ and $H'_{ap}(\omega)$ are respectively the periodic and aperiodic parts of the source sample, introduced in section 1.4.3,
- α is a continuous control parameter, between 0 and 1. When $\alpha = 1$, the periodic part disappears, simulating whispering. When $\alpha = 0$, the aperiodic part disappears. When $\alpha = \frac{1}{2}$, the periodic and aperiodic parts are left unchanged.

Vocal tract length

Compressing or stretching the envelopes $H_p(\omega)$ and $H_{ap}(\omega)$ in the frequency domain leads to an impression of change in the length of the vocal tract:

$$H'_p(\omega) = H_p(\lambda\omega) \tag{2.22}$$

$$H'_{ap}(\omega) = H_{ap}(\lambda\omega) \tag{2.23}$$

where λ is a continuous control parameter between 0 and ∞.

When λ is close to 1 ($0.85 \lesssim \lambda \lesssim 1.2$), the synthesized voice remains realistic: a value lower than 1 makes a male voice closer to a female or juvenile voice; a value higher than 1 produces the opposite effect. Values of λ further from 1 lead to more extreme effects, such as a "chipmunk voice" for $\lambda << 1$.

Sound level

Sound level is easily changed. In the World version, just like for other timbre modifications, we apply changes in sound level by modifying the envelopes H_p and H_{ap}:

$$H'_p(\omega) = GH_p(\omega) \tag{2.24}$$

$$H'_{ap}(\omega) = GH_{ap}(\omega) \tag{2.25}$$

with G a real, positive-or-zero control parameter.

In the SuperVP version, the gain is applied to the output signal.

Vocal effort

The timbre modifications described in the previous sections consist in directly manipulating some specific aspect of the signal, based on one control parameter. In that respect, they may be described as "low-level". In an artistic perspective, higher-level manipulations may prove useful. In Voks, one high-level manipulation, that of vocal effort, is implemented.

"Vocal effort" is the phrase we use to characterize a voice when singing loudly. The obvious dimension affected by vocal effort is sound level, but timbre is also affected by it. In Voks, vocal effort is implemented as the combination of several of the aforementioned changes:

- Glottal formant shift,

- Spectral slope,
- Variation of the relative prevalence of the periodic and aperiodic part (only in the World version),
- Global sound level.

To express dependence of this high-level manipulation into lower-level ones, we make the parameters γ, ω_c, α and G no longer depend directly on time, but on a new continuous control parameter $e(t)$, whose value is comprised between 0 and 1. This parameter e is what we call "vocal effort".

Chapter 3

Voks in Practice

Contents

The previous chapter has described the technical underpinnings of Voks. As will be discussed in chapter 8, Voks has been used in a variety of contexts, and as such observations regarding its use in practice can be made. In this chapter, we make such practical observations, focusing in particular on the choice of interface and on the available playing modes.

3.1 Interfaces

Experiencing the control of Voks with different interfaces has highlighted some of their specificities. In this section, an account is given of those specificities, with a special focus on the tablet and the theremin, which have been used with Voks in musical contexts more extensively than other interfaces.

T-Voks has been used extensively by Xiao Xiao only — meaning only she has been practicing consistently with the aim of becoming truly skilled at it, and only she has performed with it in

front of a live audience. Xiao is an accomplished musician: she has been studying the piano for 26 years and picked up singing lessons in 2021, and has perfect pitch. She has been learning the theremin with meticulous effort for about five years, and can thus also be considered a reasonably skilled thereminist.

Given how much harder the theremin is to learn than the graphic tablet (see section 3.1.1), it makes sense, as a first approach, to focus on the situation of an accomplished thereminist picking up an augmented version of the instrument they already know.

3.1.1 Pitch control

Theremin

Invented in 1920 [58, 122], the theremin is widely considered to be one of the very first electronic instruments, and as such, it has been attracting interest from performers and composers alike since around the 1930 [58]. Interestingly, the (non-augmented) theremin is often compared to voice, probably owing to the strictly monophonic, strictly continuous nature of its output, and to its timbre. Like most common music instruments, the theremin is associated with a repertoire, a small though active community of performers, playing techniques, and schools of thought. As a consequence, T-Voks can benefit from the expertise of theremin players trained during the "many years of study and practice [...] required to master the theremin" [69].

Our discussion with thereminists has made evident the need of some of them for a theremin-like instrument with more expressive capabilities — notably, a sound spectrally richer than that of the default theremin. Eric Wallin's work on such a digital synthesizer with a theremin-like control method, the D-Lev[1], is worth mentioning.

A fundamental specificity of the theremin as compared to the other interfaces considered is the fact that the gestures for pitch control take place in three-dimensional space, without any guide to the performer's hand. This makes it difficult to even maintain a constant pitch, with the overall movements of the body potentially parasiting the hand movements. To adapt to such challenges, some thereminists advocate specific playing techniques, such as "hand positions" reminiscient of violin left hand positions — nevertheless, playing techniques of thereminists are very diverse and individual. The techniques for pitch control in T-Voks, however, are not specific to it, and learning them is part of the general theremin learning process.

Figure 3.1 shows Xiao Xiao playing T-Voks, that is, Voks controlled with a theremin.

Graphic tablet

Although the graphic tablet has been used for at least one decade and a half as a controller for pitched synthesis [28], it does not enjoy, as a music instrument, the same prestige as the theremin, touched upon in the previous section.

In contrast to the theremin, users of the graphic tablet as a music controller have a surface which has two benefits. First, it provides a tangible object on which the performer's wrist and

[1] https://www.d-lev.com/

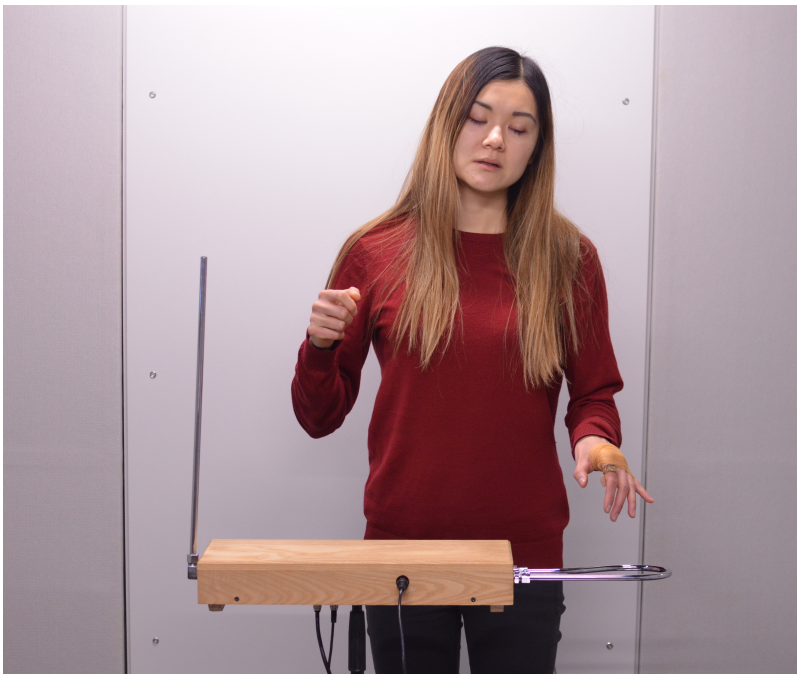


Figure 3.1: *Xiao Xiao using a theremin and a force-sensing resistor to control Voks. The combination of software and hardware is named T-Voks.*

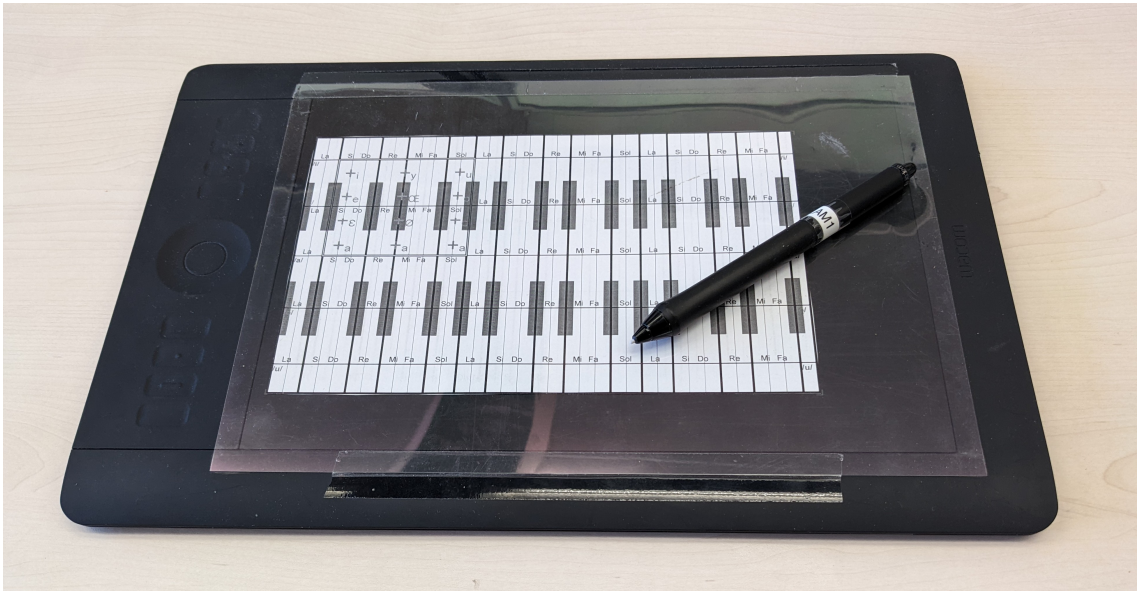


Figure 3.2: *The Wacom Intuos Pro graphic tablet, with its stylus. A mask has been affixed to the surface of the tablet to locate pitches semitone by semitone.*

stylus can rest, making the motion of any part of the body above the wrist having effectively no impact on pitch control. Second, it provides a visual aid to for performers to reach the desired

notes accurately.

The stylus associated to the graphic tablet also puts it to an advantage over other controllers based on a two-dimensional surface, such as mobile interfaces and certain MPE controllers. First, it benefits from the expertise gained when learning to write [24]. The presence of a second dimension on an interface that controls a one-dimensional parameter also grants performers with more flexibility to design optimal gestures [75] — that is, unless the other dimension is used to control some other parameter (as is the case in scrub and speed mode).

With the graphic tablet, intensity is controlled by the pressure of the stylus on the surface. Compared to intensity control in the theremin, the associated gesture happens on a smaller space scale (of the order of the millimeter for the graphic tablet, as opposed to a few decimeters for the theremin). In fact, control using a graphic tablet may involve isometric gestures, that is, changes in e.g. stylus pressure without any visible displacement of fingers or the stylus. This contrasts with the theremin gestures, which are isotonic, that is, they necessarily involve displacement of the hands or fingers. As a result, intensity control with the graphic tablet is faster, but less fine, than with the theremin.[2]

Mobile interface

A modified version of Voks has been developed to be controlled with mobile interfaces to assess the relevance of performative speech synthesis in foreign language acquisition. The design and technical details of the apparatus are detailed in chapter 4. The version of Voks controlled with a mobile interface has only been used in the specific context of the studies regarding foreign language acquisition; nevertheless, some observations can already be made from that limited experience.

Note that although there are commercially available styluses for mobile interfaces, none has been tested with Voks yet. With a stylus, the control gestures on a mobile interface would presumably be similar to those on a graphic tablet, though the influence of device specifications such as resolution on the quality of control, as well as size, would have to be investigated.

Mobile interfaces have a shape comparable to the graphic tablet. However, not using a stylus makes for less comfortable and less accurate gestures. The size of most mobile interfaces — and all interfaces tested — is also smaller than that of the Wacom Intuos Pro, the graphic tablet which we have mostly been using. This makes controlling Voks with a finger on a mobile interfaces even less convenient compared to the graphic tablet.

As a counterpart to those downsides, the wide availability of mobile interfaces would make it easy for a large part of the population to use Voks casually, without the need to invest in new hardware.

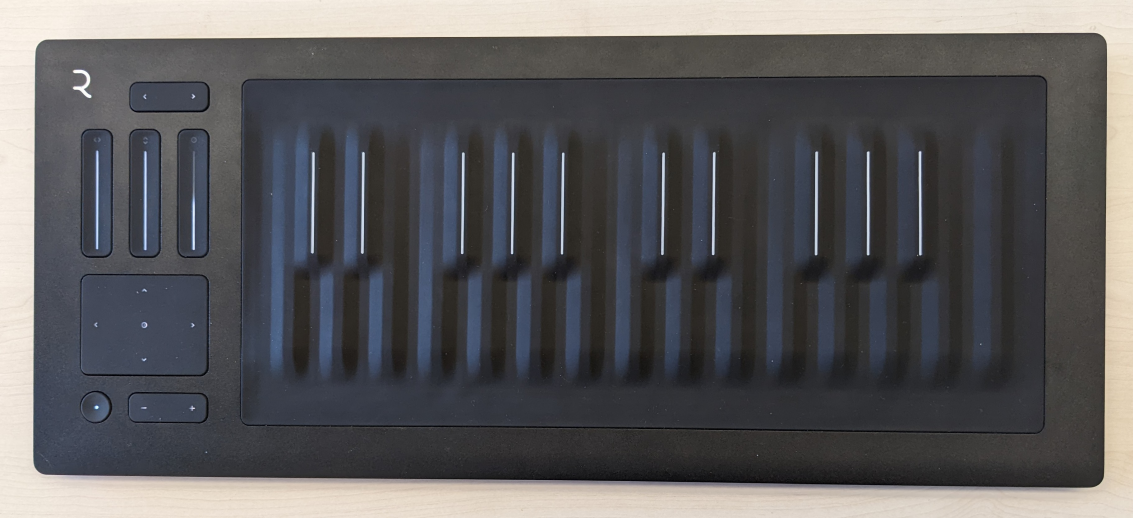


Figure 3.3: *The Seaboard interface, with its uneven surface*

Seaboard

MPE, standing for *MIDI Polyphonic Expression*, is a protocol that builds upon the MIDI protocol. It allows interfaces and synthesizers to communicate in a way that plain MIDI does not. In particular, it can handle individual modulation of pitch, intensity and other parameters at the note level. MPE interfaces thus seem well-suited to control of a continuous-pitch synthesizer like Voks. Several so-called "MPE interfaces" have been commercialized. Those usually include a 2D surface, which may be totally flat, or feature differences of elevation to help performers find their way on the interface without looking at it.

Voks has been tested with a ROLI Seaboard, an MPE interface composed of a soft surface with hollows and bumps that make up a continuous version of the traditional piano keyboard. On the side, a smaller 2D rectangle and sliders provide a way to modulate control parameters. Although only the Seaboard has been tested, the following remarks remain generally applicable to other MPE interfaces.

Pitch control is assigned to the position of a finger on the surface of the interface. As with previously mentioned interfaces, it is possible to change pitch continuously by sliding one's finger on the surface, making glissando and vibrato possible. As Voks currently only features one voice, fingers other than the first to come into contact with the surface are ignored. Rhythm can be controlled with a separate interface such as a key from a computer keyboard. Controlling Voks in this way is not dramatically different from controlling it with another 2D surface such as a tablet or mobile interface.

The piano keyboard-like design of the Seaboard, however, suggests another way to control rhythm, that which is used in keyboard instruments: using the presses and releases of the finger

[2]It is in fact possible to set up one's theremin in such a way that the spatial range of the intensity-controlling gestures is on the order of a few millimeters. This course of action is not common, however, and such a high sensitivity is seen by players as a hurdle more than anything else.

that determines pitch. This method has been implemented; testing has quickly brought to light the following issues:

- Finger articulation in keyboard instruments is not exactly analogous to syllabic articulation as implemented in Voks. If several fingers alternatively come into contact with the interface — as is usually the case when playing a melody on a keyboard —, there is no guarantee that presses and releases are in strict alternation. It is perfectly possible, for instance, for two fingers to come in contact with the interface without the first one leaving the surface in between. *A priori*, this leads to a problem, as by default, Voks' rhythm control algorithm that there is always a release between each two successive presses, and a press between each two successive releases. Undefined behavior may arise, and a modification to the algorithm is warranted.
- If the same finger is to control pitch and rhythm, the pitch is left undetermined every time the finger leaves the surface. A natural solution would be to take into account the last position of the finger as it was still pressed, but this solution leaves some new potential issues open: constancy of pitch when no finger is pressed, discontinuity of pitch when the finger is pressed again.

Those issues may be solved by modifying the software slightly, but finding the right modifications to apply would require more in-depth testing than what has been done yet.

Comparison

To sum up, from our experience, the graphic tablet offers an ease of pitch control that makes it easier to learn than the theremin, without sacrificing an unreasonable amount of expressivity, though the differing characteristics of the two interfaces makes it relevant for both to coexist. Those observations will have to be supplemented with a formal comparison of interfaces for continuous pitch control in the future.

3.1.2 Rhythm control

Rhythm can be managed with a simple binary interface such as a musical or alphanumeric keyboard key. If playing with a theremin, naturally, one needs an interface more portable than a whole keyboard; one that can be held in one hand, the one that also controls intensity. A simple force-sensing resistor (FSR), as in figure 3.4, or a small button, are adequate. Such interfaces, however, must be connected to the computer running Voks via a small, custom electronic circuit, which has historically caused practical issues such as interference with the theremin and unreliability.

Xiao Xiao's experience with T-Voks has given rise to a number of observations related to adjoining a syllabic rhythm control interface to T-Voks.

The adjunction of an interface for syllabic control, simple though it may seem, causes a significant increase in the thereminist's mental load, at least at first. Even an experienced thereminist needs some training to manage both the parameters they are used to, namely pitch and intensity, and rhythm.

Figure 3.4: *A force-sensing resistor used as a rhythm interface, to be used combined with a theremin. The sensor is held in place in the hand that also controls intensity, with the combination of a ring and a fabric strip.*

Traditional thereminists still exert some amount of rhythmic control through modulation of the intensity parameter. They usually make the intensity lower during the transition between notes, to an extent and in a manner dependent on artistic choices. If part of a piece, for instance, is to be played *legato*, such a modulation will be slow and mild, if not inexistent. If, on the other hand, the thereminist wishes to play *staccato*, inter-note changes of intensity will be fast and intense. In the practice of the theremin as well as in music in general, the manner in which the transition from note to note is sounded is named *articulation*.

The transition from the traditional theremin to T-Voks implies a change in the management of articulation. In the traditional theremin, articulation is entirely managed with changes in intensity. The addition of syllabic rhythm control to T-Voks forces thereminists to rethink the way that they manage articulation, and think more on the scale of musical phrases than of individual notes.

With that said, some control of intensity on the level of a note is necessary even with articulated speech. In her use of T-Voks, Xiao Xiao's has felt the need to soften the transitions between syllables, which she achieved with slight wrist movements, superimposed onto larger, phrase-level movements of the left arm.

Finally, the delay between the gesture (in the case of T-Voks, the press of a button or of a force-sensing resistor) and the triggering of the next syllable is another factor that participates in the mental load associated with learning to control T-Voks.

3.1.3 Vocal quality

Timbre changes

All timbre transformations can be used for expressive purposes. There is a fundamental distinction, however, between the timbre changes related to vocal effort, and other timbre changes.

Vocal effort is directly connected to gestural parameters: when using a graphic tablet, it is related to stylus pressure; when using a theremin, it is connected to one of the theremin antennae. As such, vocal effort is determined by fine gestures synchronized with pitch and rhythm gestures.

In contrast, timbre changes such as vocal tract length change can only be controlled using the graphical interface, or sliders separate from the main interface. The parameters driving those timbre manipulations are thus more difficult to control simultaneously. Such parameters can be set by the performer:

- Either in between musical phrases, when their attention can be entirely dedicated to the parameter change;
- Or while playing a musical phrase; from our experience, however, changing the parameter requires enough of the performer's attention that it is difficult to do anything musically interesting with the main interface. Thus the parameter change, as long as it is taking place, has to be the main point of attention of the performance. As an example, changing the vocal tract length of a virtual singer is possible, but that operation will have to be the main point of focus; playing something interesting while changing the vocal tract length is a difficult feat to pull off.

3.2 Source sample requirements

A word should be said about the source sample, whose timbre greatly affects the quality of the synthesis in ways that we do not yet fully understand. It remains to be seen exactly what vocal parameters make for a good source sample, and why.

3.2.1 Vocal effort

The spectral manipulations related to vocal effort tend to reduce vocal effort, but it is harder to increase it. Thus our initial assumption was that samples with high vocal effort — assuming they are otherwise clearly articulated — would make for good source samples. Experience, however, has shown otherwise, and samples from professional-level singers chanting with a high vocal effort tended to result in synthesis with low intelligibility. On the other hand, a non-musician speaker has recorded samples with relatively low vocal effort which ended up being perfectly adequate for synthesis.

3.2.2 Register

Another observation is that use of the falsetto register to record samples has, at times, resulted in low intelligibility of the synthesized sound. Naturally, Voks' algorithm itself does not differentiate between source sample registers — nor does it care what the pitch of the source sample is. However, the more different performance pitch is from source sample pitch, the more likely it is for the synthesis to be of lesser quality. Thus when playing music written for a falsetto voice, a compromise might need to be found between faithfulness and intelligibility.

3.3 Playing modes

The diverse parameters and interfaces make several playing modes available. Some of them result in articulated voice, others do not; some of them are natural-sounding, others result in sounds that could not have been produced with a natural voice.

3.3.1 Imitation of natural, articulated voice

One natural way to play Voks, which could be considered the default, is to imitate natural voice. This is done by using syllabic mode and no timbre changes other than that of intensity. This playing mode can emulate either spoken voice or singing voice, as well as anything in between, including *Sprechstimme* (see chapter 8, section 8.2.2). It should be noted, however, that whereas it is possible to emulate a singing voice fairly convincingly, it is usually easy to tell apart a natural speaking voice from a synthesized speaking voice. This might be due to the presence of clear pitch targets in music, which, when controlling pitch with a tablet, even correspond to visual marks. In spoken voice, on the contrary, the pitches are not specified explicitly, and microprosody plays a more important role.

Voks' available timbre modifications make it possible to apply effects to the voice while performing the same gestures as in the default playing mode.

3.3.2 Whispering

By removing the periodic part from the World spectral representation of a sample, a whispering effect can be applied to an originally normal-sounding voice, which becomes *unvoiced*. The amount of voicing being a continuous parameter, intermediaries are also possible: a small amount of voicing gives a soft voice with a lot of breath. However, even small amounts of voicing give significantly different-sounding results from no voicing at all. From our experience, there seem to be a discrete perceptual separation between whispered and non-whispered voice.

The inverse operation, that is, suppression of all non-harmonic content of a voice, is also possible, but it is not particularly spectacular. The perceived effect can be described as sounding similar to that of a low-pass filter.

3.3.3 Vocal tract length changes

Changes in the length of the virtual vocal tract, if moderate (with a stretch factor between about 0.8 and 1.2), shift the position original sample on the masculine/feminine spectrum. In this way, an originally female voice can be turned into a convincing male one, and vice versa.

More extreme changes give more radical effects. Shrinking the virtual vocal tract can give a baby or chipmunk voice, while stretching it gives yet another, impossible voice.

3.3.4 Babbling

While scrub and speed modes can be used to play a text linearly, nonnatural-sounding effects can also be obtained with them. By actually scrubbing the surface of the tablet in scrub mode, a fast, nonsensical succession of phonemes is obtained. Those are alternatively reversed and not reversed, depending on the direction of the stylus along the vertical axis of the tablet, making it certain that the voice heard makes no sense. An advantage of scrub mode is the direct correspondence between position of the stylus and segmental content, which provides the performer with close control over the synthesized sound.

Using speed mode, one can get a related effect: with a speed fast enough, the synthesized voice ceases to be intelligible, and a babbling effect is obtained. That effect can be obtained with either a positive speed or a negative one. With a negative speed, the voice is guaranteed to not be intelligible. As a counterpart, some phonemes might sound non-physical, although the sound remains very much voice-like.

3.3.5 Sustained sound

The performer can keep the time index constant using either one of two methods:

- In rhythm mode, by not performing any rhythm control gesture;
- In speed mode, by staying on the zone in which the speed parameter is zero.

The time index can also theoretically be kept constant in scrub mode, if the stylus is kept on a specific vertical coordinate, but this is difficult to achieve, as subtle hand movements are likely to affect the stylus position along the vertical axis.

Keeping the time index constant results in a timbre which is fixed by default. Depending on the value of the time index, the timbre is that of a vowel or a consonant — the choice is up to the performer: vowels and nasal consonants are good phonemes to keep the time index constant if wanting to sustain a clear, pitched sound, whereas fricatives result in sustained noise.

With this playing mode, Voks behaves like a rudimentary version of Cantor Digitalis, where pitch and intensity can be controlled with chironomy. Other timbre changes are still possible. Vocal tract length modifications are particularly spectacular here, as the constancy of all other timbral parameters makes it most noticeable.

As with Cantor Digitalis, all harmonic content of the sound can be removed, leaving only some noise to be heard. Harmonic manipulations such as vocal tract length change can still be applied: the sound then gets further from voice and ventures into electroacoustic aesthetic territory.

Chapter 4

Potential of Performative Vocal Synthesis as an Educational Tool

Contents

4.1 Introduction

Beyond the artistic domain, Voks is also being considered as an educational tool, in particular in foreign language acquisition. In the context of the GEPETO project, fellow researchers Xiao Xiao, Claire Pillot-Loiseau, Nicolas Audibert, Barbara Kühnert, and Lise Buchman have been interested in Voks' educational potential and have conducted some preliminary studies to assess its educational relevance. To conduct those studies, Xiao and we have developed a version of Voks controlled with mobile interfaces, much more widespread than the graphic tablet and the theremin.

In this chapter, we describe the design of this new version of Voks, which for brevity we will call E-Voks ("E" standing for "education") in the remainder of the chapter.

4.2 Adapting Voks' interface for education

For the purpose of using Voks as an educational tool, the synthesizer has been overhauled; the vocoder has been left unchanged, but the whole interface has been redesigned so that the synthesizer could be controlled using a mobile interface.

Though we would hope for a pedagogical version of Voks to ultimately fully run on a mobile device, note that the version described here does not; only the interface is a mobile one, and the synthesis engine runs on a separate computer. Such an arrangement allows to run experiments using a device like the one that we would like to ultimately use, without having to go through the trouble of porting the whole synthesizer to run on mobile devices.

4.2.1 Architecture

To make it feasible to independently develop a new graphical interface for Voks, the part of its code which has to do with audio — featuring the vocoder, the rhythm control algorithms and the pitch and timbre modifications — has been separated from the interface — graphical interface and gestural control interfaces. The interface modules have been removed and replaced with a layer with OSC-receiving capabilities. OSC (Open Sound Control) is a protocol analogous to, but more versatile than MIDI, allowing for communication between unrelated programs.

As a result of the replacement of Voks' interface with an OSC layer, the following are controlled by receiving OSC messages:

Another program has been developed by Xiao Xiao in Javascript. This program is a web server with OSC-sending capabilities. This means that any device which is able to display a web page is a potential control interface; interactions of a user with that webpage are communicated to the server, which in turn sends OSC data to control the synthesizer. Figure 4.1 shows the global architecture.

- File paths for the source audio and labeling files;
- Discrete and continuous rhythm control signals;
- Pitch;

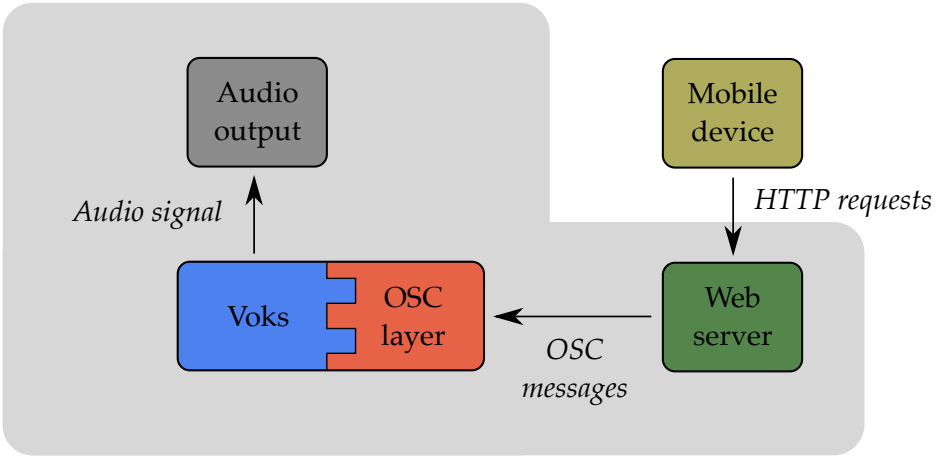


Figure 4.1: *Functional diagram of E-Voks. Blocks in the gray area run on a computer.*

- Timbre modification parameters.

4.2.2 Interface

The interface of E-Voks is shown in figure 4.2. Although a web interface can, in principle, be interacted with using any web device, the interface of E-Voks was designed specifically to be displayed on a touch screen. As such, a large part of the screen is directly dedicated to chironomy. The remaining part of the screen consists of buttons to control the system in non-chironomic ways.

Chironomy area

The "chironomy area" (in cyan on figure 4.2) is the zone on which the user will move their finger to control intonation. It is a rectangle split vertically into smaller rectangles of equal width. Each rectangle corresponds to a syllable of the currently loaded sentence; the IPA transcription of said syllable is displayed in the center of the rectangle. The syllables are arranged from left to right, and thus each horizontal position is mapped to an analysis time, or time index (see section 2.3.1).

Although not labeled as such, the vertical axis corresponds to pitch: each vertical position is mapped to a pitch value on a two-octave-wide scale, with higher positions corresponding to higher pitches. Regularly spaced horizontal reference lines are located one tone apart from each other.

Using this two-dimensional time-pitch mapping, users can control E-Voks in scrub mode (see section 2.3.2).

The displaying capabilities of the touch screen (as opposed to a screen-less graphic tablet) are taken advantage of by displaying the trace of the user's finger as they are moving it on the screen. Depending on the mode selected, the trace either disappears a few tenths of a second after the gesture, or remains visible until further action is taken by the user.

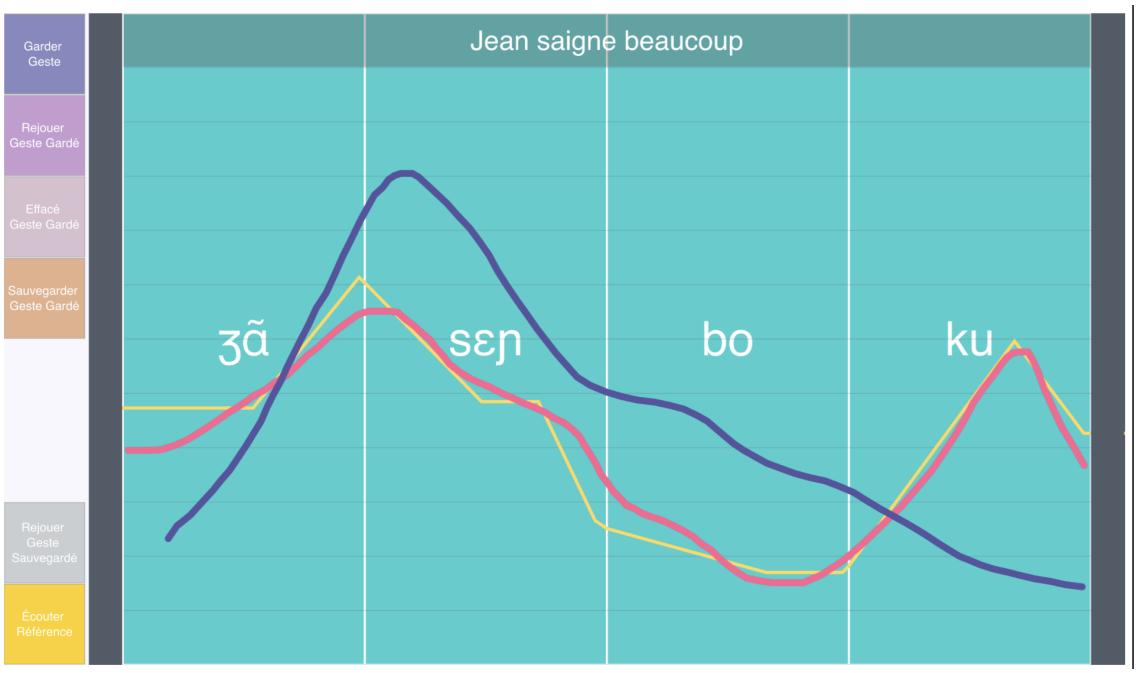


Figure 4.2: *Screenshot of E-Voks. Picture reproduced from [129]. The yellow curve is a reference trace, the purple curve is a remaining trace from one of the user's previous attempts at reproducing the sentence; the fuchsia curve is the result of the user's current attempt (by default, fades after 1.5 seconds). Translation of the text on the buttons on the left panel, from top to bottom: 1.* Keep gesture, *2.* Replay kept gesture, *3.* Erase kept gesture, *4.* Write gesture on disk, *5.* Replay saved gesture, *6.* Listen to reference.

Traces can be saved. Saved hand-generated traces, as well as reference traces, can be loaded and recalled. Upon loading a trace, the whole trajectory gets displayed for the user to follow. Different types of traces are illustrated in figure 4.2. Note that the reference trace does not come from a gesture, but has been generated following a process described in section 4.4.1. This explains the contrast between the appearance of the yellow trace in figure 4.2, made of connected segments, and the other, manually generated traces, which are actual curves.

Traces are expected to be a helping tool for the user, but note that they do not contain all the information of a realization of a sentence. That is, even perfectly following the same trace twice can lead to different auditory results, and perfectly following the trace generated by some previous chironomic realization of a sentence does not guarantee an identical auditory result either. This is due to the trace only including intonational information, not temporal information; to get a sound identical to some performance, one would have to follow the corresponding trace *with the same speed as in said performance*. To aid with this, the interface features a replay button which plays back the trace with a cursor showing the speed.

Control panel

On the left of the interface is a panel with a number of virtual buttons (on the left of figure 4.2, in various colors). The buttons allow users to listen to, save and load gestures, as well as to listen to reference audio.

Note that the version of E-Voks described here, whose interface is shown on figure 4.2, has been developed in view of the test. As such, the next sentence is automatically brought up upon saving a gesture — with an additional informative interface on the computer running Voks to show which sentence is currently being considered. When not running experiments, an interface on the computer allows users to choose sentences.

4.3 Disambiguation test

The work described in the remainder of this chapter is for the most part the result of Xiao Xiao's efforts. The pilot study has been published in [129], reproduced in the appendix of this document. The analysis based on GAMMs will appear in the proceedings from the 2022 edition of the Journées d'Études sur la Parole [130], also reproduced in the appendix.

The term *chironomy* has been proposed in [21] to refer to control of sound using one's gestures, especially in the context of vocal synthesis. It provides a new modality through which prosody can be controlled with satisfying accuracy. It can be argued that such a modality is more tangible than that of natural voice, which relies on internal, largely invisible processes.

Using chironomy in foreign language learning is hypothesized to be beneficial in three regards:

- Chironomy may help learners perceive features of the target language that would have gone unnoticed with the natural voice modality;
- By producing vocal sounds without using their natural voice, learners may be less subject to unconsciously applying patterns specific to their native language;
- The automated quality of segmental content production in Voks may help learners focus on suprasegmental features, thus separating the problem of production of segmental content on one hand and prosody on the other hand.

To test Voks' potential as a way for foreign language learners to control and experiment with prosody in their target language, a pilot study has been carried out, focusing on the case of prosody in French, which makes use of the paradigm of *syntactic disambiguation*.

4.3.1 Disambiguation tasks

The study is based on *disambiguation tasks*. A disambiguation task consists in presenting subjects with audio samples of semantically distinct sentences whose segmental content happens to be the same (see table 4.1), and asking them to discriminate between two possible meanings. Meaning selection may be done by asking subjects to select one among several orthographic transcriptions — provided those are not ambiguous —, or one among several accompanying so-called "disam-

biguating contexts", as is done in [99].

Prosody has been shown in [99] to be an important cue in disambiguation tasks. Such tasks thus constitute an appropriate test to assess the quality of the prosody obtained with a system such as E-Voks. More explicitly, comparing the performance of subjects when performing disambiguation tasks based on samples recorded using natural voices on one hand, and vocal synthesis on the other hand, gives an indirect indication on the quality of prosody of vocal synthesis as compared to natural vocal performance.

4.3.2 Aims of the study

The pilot study recounted in [129] focuses on intonation; the goal is to compare the quality of intonation as realized chironomically with that realized with one's natural voice. The influence of the following additional variables is measured:

- Presence/absence of a visual guide;
- Automated/manual timing control;
- Language proficiency of the subject (two categories: native/non native).

The influence of the choice of language is not measured; the experiment focuses on the case of French.

4.4 Protocol

Ten subjects participated in the test. Five were native speakers, the other five were learners at an advanced level. They were presented with:

A computer A MacBook Air laptop running the testing interface. The interface integrates E-Voks as well as a simple graphical interface to guide subjects during vocal imitations;

A touch screen tablet A 9.7in Samsung Galaxy S2 tablet, connected to the computer, displaying E-Voks' interface and controlling synthesis;

A microphone An AKG C414 XLS microphone, connected to the computer, used in the vocal tasks;

A pair of headphones For listening to reference audio and their own vocal and synthesized productions.

4.4.1 Corpus

The corpus consists of three parts:

- Pairs of ambiguous sentences in French, that is, sentences that have the same phonetic representation, but distinct meanings. Thus the only audible differences between sentences of a pair are prosodic features;
- Audio recordings of the sentences in the ambiguous sentence corpus;

- Traces to help with synthesizing the sentences in Voks.

For the text, a corpus of 6 pairs of ambiguous sentences in French has been selected; the sentences, along with a phonetic transcription and a translation in English, are listed in table 4.1.

Phonetic transcription	Ambiguous sentences	English translation
/typaʁɛtʁɛsusjø/	Tu parais très soucieux.	You seem very worried.
	Tu paraîtrais soucieux.	You would seem worried.
/ʒɑ̃lɛvsɔ̃vɛʁ/	J'enlève son verre.	I remove his glass.
	Jean lève son verre.	Jean lifts his glass.
/ʒɑ̃sɛɲboku/	Jean saigne beaucoup.	Jean is bleeding a lot.
	J'enseigne beaucoup.	I teach a lot.
/ʒɑ̃pɔʁtɛ̃ʒuʁnal/	Jean porte un journal.	Jean carries a newspaper.
	J'emporte un journal.	I take away a newspaper.
/ʒɑ̃kadʁlafoto/	Jean cadre la photo.	Jean centers the photo.
	J'encadre la photo.	I frame the photo.
/selamɔʁsyʁ/	C'est la morsure.	It's the bite.
	C'est la mort sûre.	It's death for sure.

Table 4.1: *The corpus of ambiguous sentences*

Recordings of the sentences from the corpus are needed for two purposes:

- To serve as reference sentences that subjects will be asked to imitate;
- To serve as input material to the synthesizer.

Every one of the sentences from the corpus has been recorded by the same native French speaker. Note that each pair of ambiguous sentences gave rise to two, not one, distinct recordings. Each recording of a sentence was used both as the reference audio for the subjects to listen to, and as input for Voks for the task of reproducing said sentence.

This implies that Voks was not fed the same input for two sentences from the same pair. This might seem to introduce a bias, with the way that the sentence was originally pronounced impacting the synthesized output. This risk is mitigated by the fact that the prosody is entirely recreated during synthesis. In particular, the output pitch, which is the main object of this study, directly corresponds to the subject's gesture; thus the pitch of the input sample cannot interfere with the measurements.[1]

For the purposes of the test, a trace was needed for each sentence, to provide a reference for the user. A natural choice would be to use the pitch contour of the reference recordings as measured using a signal processing algorithm. However, as noted in [82], the raw pitch contour of a speech audio sample features various microprosodic phenomena which do not directly participate in the perception of pitch, and whose imitation using Voks would be difficult, requiring exceedingly

[1]Strictly speaking, the pitch of the input sample can have an effect on timbre. For two recordings by the same speaker in the same conditions, we postulate that effect to be negligible for our purposes.

fast hand movements.

Instead of being based solely on the pitch contour of the reference samples as *produced* by speakers, the traces are obtained based on a *perception* model [23]: a *prosogram* [82] of each sentence, that is, a stylized representation of its perceived pitch, is computed using the Prosogram software [82]. The resulting prosogram, a simple "curve" composed of straight line segments, is used as the reference trace.

4.4.2 Tasks

The experiment consists in asking subjects to imitate a series of sentences from ambiguous pairs under different modalities. It is split into four successive phases; in each of those phases, all sentences from the corpus described in section 4.4.1 appear once, in random order.

1. Subjects are first presented with the written text of each sentence, which they are asked to record themselves reading. No reference audio is played in this phase; subjects only rely on the text and their own knowledge of the language.
2. Then subjects are presented with the reference recording of each sentence, and are asked to imitate it using E-Voks. No reference trace is provided in this phase.
3. The task in the third phase is the same as in the second one, except that a reference trace is now displayed on E-Voks interface. The process to produce the reference traces was described in section 4.4.1.
4. In the last phase, subjects are asked to reproduce the reference recordings again, but using their natural voice with the microphone.

Table 4.2 summarizes the four conditions.

Task number	Task type	Reference audio	Reference trace
1	Vocal	No	
2	Synthesis	Yes	No
3	Synthesis	Yes	Yes
4	Vocal	Yes	

Table 4.2: *The conditions of each of the four tasks of the test.*

Perceptual correctness assessment

To interpret the subjects' production, in addition to the data analysis that will be described in the next section, a perceptual assessment has been conducted to evaluate the ability of listeners to discriminate between the two meanings, in line with the notion of disambiguation. This is necessary to answer the question of whether gesture-controlled prosody is able to remove the ambiguity caused by two sentences featuring the same phonetic content.

The test was conducted online with 37 native French speakers as subjects. The stimuli were the productions of subjects for one specific ambiguous sentence pair, namely "*Jean cadre la photo*"/"*J'encadre*

la photo". Each subject was tasked with listening to all the stimuli, in random order, and, for each stimulis, classifying it into one of the two possible meanings.

4.5 Analysis

The imitations were analyzed in several ways:

- The pitch contour of imitations and reference sentences was extracted, and the similarity between each pitch contour and its respective reference was computed using measures from [25]. A statistic analysis was subsequently performed on the similarity measures using Bayesian Multilevel Linear Models.
- Generalized additive mixed models (GAMM) were used to compare the pitch of both versions of six of the sentences, as well as to compare the pitch of native subjects' imitations as compared to non-native subjects.

The perceptual assessment and similarity measure between pitch contours are reported in [129]; the statistical modeling using GAMM is to appear in [130].

4.5.1 Pitch processing

On the following two sets of data, the pitch is extracted and similarity measures between pitch contours are computed:

- Vocal imitations (task 4 in table 4.2) compared with guided gestural imitations — guided imitation being assumed to give the best results for gestural imitations.
- Guided gestural imitations (that is imitations involving a reference gestural trace to follow) compared with non-guided ones, to observe the effect of the guide.

In the first set, the subjects' deviation from the timing of the reference is corrected (see paragraph "Synthesized imitation processing" below) so as to only compare intonation. In the second set, the original timing is conserved, so as to evaluate the influence of the guide on timing.

Due to the difference in nature between vocal imitations and synthesized imitations, for which gestural data is directly available, both are processed in a slightly different way. The result is processable data of the same nature for both task types.

Pitch contour extraction and normalization

Vocal imitation processing For each subject performance, the pitch contour is first automatically determined, then manually verified, using the Praat software [120]. The performances being the result of a human effort, they contain inevitable differences in timing compared to the reference audio, which cannot be corrected by a simple method such as a general shift in time. To deal with this, the temporal axis of the vocal imitations is distorted using a Dynamic Time Warping algorithm [90]. The result is a pitch curve which is temporally aligned with the reference recording. To get a series of discrete data points, the curve is then resampled every 10 ms.

Synthesized imitation processing Here no analysis is required to determined the pitch of the utterance. The pitch contour is already available as a series of data points. More specifically, the captured data is a sequence $(t_i, \tau_i, p_i)_i$ of N data points, for some N, where:

- The i are natural numbers ranging from 0 to $N-1$;
- The t_i correspond to times during the test;
- The τ_i correspond to the time index (see chapter 2, section 2.3.1 for a definition of the time index) of the synthesis at time t_i;
- The p_i correspond to the pitch at time t_i.

In other words, the data available is the discretization of the 2D trajectory of the (time index, pitch) pair.

From that data, two slightly different pitch contours are obtained and used in the analysis:

- The pitch contour of the imitation, simply scaled globally so that its length matches that of the reference;
- The pitch contour of the imitation, warped so that the time index values are linear. This is the same type of warping as that which is applied to vocal imitations. The difference is that here, the time index is directly available, so no specific dynamic time warping algorithm is necessary.

4.5.2 Similarity measures

The similarity between the subjects' imitations and the references was measured both quantitatively, with two measures of similarity between pitch contours, and qualitatively, with a perceptual study.

The measures of similarity are the measures used in [25]. As in [25], we take advantage of the presence of intensity data by weighting the values by the intensity (w_i) of the reference samples.

Correlation coefficient

The use of the correlation coefficient to compare pitch contours is developed in [60]. The correlation coefficient is defined as the normalized sum of the product of the deviation of each contour from the mean — with an additional weighting by intensity:

$$r(f,g) = \frac{\sum_i w_i f_i g_i}{(\sum_i |w_i f_i|^2)^{\frac{1}{2}} \, (\sum_i |w_i g_i|^2)^{\frac{1}{2}}}$$

where f and g are pitch contours, consisting in series of data points f_i and g_i.

It is comprised between -1 and 1, with a value of 1 indicating similar curves. We then scale the domain of this value nonlinearly so that it is not bounded anymore, but distributed on the whole real line, following an approximately Gaussian distribution. This is done by means of the Fisher transform [50]:

$$r' = \operatorname{arctanh}(r) = \frac{1}{2} \ln \frac{1+r}{1-r} \tag{4.1}$$

Root Mean Square Error (RMSE)

The root mean square error is the 2-norm of the difference between the two pitch contours, normalized by the number N of data points.

$$\text{RMSE}(f, g) = \left(\frac{|f - g|^2}{N} \right)^{\frac{1}{2}} \tag{4.2}$$

where again f and g are pitch contours as above.

The result is a nonnegative value, with 0 corresponding to the maximum similarity. Again, we nonlinearly scale the domain of this value so that it is distributed along the whole real line:

$$\text{RMSE}'(f, g) = \ln(\text{RMSE}(f, g)) \tag{4.3}$$

Statistical modeling

The correlation and RMSEs were fed into Bayesian Mixed Models, a class of statistical models which sees an observed value as the sum of the result of effects of different factors and some noise, allowing for observation of possible significant differences in the scores between selected sets of data. Two comparisons were performed:

- **Vocal imitations** were compared to **chironomic imitations with a reference trace**. Timing was corrected to correspond to the reference.
- Chironomic imiations **with** and **without** a reference trace were compared. Timing was left unchanged.

4.5.3 Use of Generalized Additive Mixed Models

The pitch contour functions were then directly fitted to two separate Generalized Additive Mixed Models (GAMMs), another class of models which are able to deal with series of data points such as pitch contours, which it models as the sum of arbitrary smooth functions of some parameters and random effects. The first GAMM distinguishes between each sentence of the ambiguous pairs, making it possible to observe the influence of sentence meaning on prosody. The second GAMM distinguishes between native and non-native speakers.

4.6 Results summary

The analyses have led to a number of conclusions:

- Subjects were able to assign the correct meaning to synthesized sentences. It is thus possible to use gesture-controlled intonation to remove the ambiguity between pairs of sentences with the same phonetic content.
- Gestural imitations and vocal imitations were comparably close to the reference in terms of the distances and statistical models used, provided that (1) the gestural imitations are performed with the help of a visual guide (a trace), and that (2) the timing specific to the

imitation is not taken into account. This last condition reflects the difficulty of controlling rhythm using E-Voks' current scrub mode of temporal control.

- Natural voice remains the best performing modality for imitation of pitch contours.
- The natives' pitch contours were not found to be significantly closer to the reference than the non-natives'; however, in the perceptual test, the natives' non-guided chironomic imitations were disambiguated correctly more frequently than the natives' ones.

4.7 Limitations of the study

The study is a preliminary one, and it suffers from a number of limitations.

4.7.1 Status of the study

This being only a preliminary study, it must be interpreted with care. The number of subjects was limited — only ten —, owing to the fact that the study took place in the middle of the COVID-19 pandemic; the size of the corpus was limited too. What is more, the study does not in itself test whether chironomic intonation is a good educational tool, and must be supplemented with further studies.

4.7.2 Rhythm control

In this study, the same interface is shared by rhythm control in scrub mode, and pitch control. The values used are the projection of the finger position along the horizontal and the vertical axis, respectively; however, the human way of thinking about trajectory of a point on a surface is not in terms of projection along axes, but rather in terms of direction and speed.

The typical problem is the difference in the distance that the subject's finger needs to cover for a similar increase in time index, depending solely on whether the pitch is being subject to fast changes. As a consequence, there is a natural human tendency to slow down synthesis with sudden pitch rises and drops. Peaks and inverse peaks, with the sudden change in trajectory that they imply, also make controlling rhythm all the more difficult in this setting.

Possible solutions include the use of a different rhythm control method (Voks' existing other modes, speed and syllabic mode, are good candidates), the separation between the interface for rhythm control and that for pitch control, for instance by using one tablet in each hand, and not letting the user control rhythm (*resp.* pitch) in studies about pitch like this one (*resp.* studies about rhythm).

Beside the need of an adequate rhythm control method when testing intonation control, testing the quality of rhythm control itself is also necessary, as rhythm is, together with intonation, one of the most important dimensions of prosody.

4.7.3 Absence of intensity control

Although the intensity of reference stimuli is taken into account to weigh the pitch contours when computing similarity measures, subjects have no control over the intensity of their performances, contact with a mobile interface being of a binary nature. Adding a way to control intensity allows users a finer prosody control, which may help with learning.

4.8 Conclusion

This study has provided a motivation and a testing ground for the development of a first version of Voks geared towards educational use. Despite its preliminary nature, its findings make the case for chironomy, especially guided, being a relevant alternative control modality beside natural voice. The modest nature of this study should serve as an incentive for efforts in three directions. First, larger studies must be conducted to corroborate the main hypothesis that this study has tested, that is, that speech intonation can be satisfactorily produced with possibly guided chironomic gestures. Second, the impact of chironomic training on learning performance must be measured.

Lastly, the synthesizer itself must be adapted further to this application, with a special focus on the issue of rhythm control. If chironomic intonation does prove to be an adequate pedagogical tool, further improvements, such as making it possible to run the synthesizer from a mobile interface without relying on a separate computer, will have to be made to make it usable in real-world contexts.

Chapter 5

Evaluation of Syllabic Rhythm Control

Contents

5.1 Introduction

Syllabic performative vocal synthesis of articulated voice has been tested out in real settings, in the context of musical performances. Those performances, however, would best be complemented with a quantitative study of the system. Conducting such a study has proven challenging. In this chapter, we describe preliminary efforts related to studying syllabic rhythm control in performative vocal synthesis.

5.1.1 Evaluation of gestural pitch control

C-Voks' pitch control method, which it shares with its relative Cantor Digitalis ("Cantor" for short), has been evaluated before as part of the work around Cantor [21]. In chapter 4, we have described investigations into the potential of a similar pitch control method in education.

T-Voks' pitch control method, on the other hand, is essentially that of the theremin. A formal evaluation of theremin pitch control, and a comparison with graphic tablet pitch control, would be relevant both for research centered around the theremin and performative vocal synthesis; however, no such study has yet been conducted.

5.1.2 Rhythm control in Voks

The other essential aspect of gestural control in Voks is its novel syllabic rhythm control method. This chapter recounts efforts to evaluate this method. This evaluation can be seen as the rhythm-related counterpart to the melody-related study described in [21], and indeed, parallels may be drawn between both. In both cases, it is evaluated to what extent certain targets are hit — in the case of melody, those targets are pitch values; in the case of rhythm, they are points in time.

Challenges related to studying rhythm

At a conceptual level, rhythm can sometimes be thought of in a rather simple way as the layout of discrete events on a regular, one-dimensional grid representing time. In the practical world, however, many factors complicate this view.

Temporal grid The hypothesis of an underlying regular temporal grid is often not valid. Only certain music genres are based on an exactly regular pulse, generally through the use of a traditional metronome, a "click track", or digitally controlled virtual instruments. Admittedly, meter-based music is still, by definition, based on an approximation of such a regular grid, which rubato, tempo variations, and more generally, artistic license, temporally distort.

Isochrony Things break down further when considering the rhythm of speech. The notion of *speech isochrony* corresponds to the idea that speech is organized around temporally regular units. Depending on the language, those units may be syllables (in which case the language is said to be *syllable-timed*), they may comprise several syllables (in *stress-timed* languages; the unit is then called the *foot*), or each syllable may be comprised of several unit's worth of length (*mora-timed* languages; unit is then called the *mora*).

To what extent is the isochrony hypothesis valid? As noted by [126] (section 1.2.1), the fact that isochrony is consistently taught in classrooms for languages of different isochrony classes (isochrony of morae in Japanese classes for native speakers, isochrony of feet when teaching English as a foreign language) is enough to maintain that the notion has some basis of truth. This consideration should however be heavily tempered: many studies, the author of [126] notes, have failed to directly observe isochrony. Thus various sources of fluctuation from a regular rhythm, related notably to the hierarchical organization of speech, have to be taken into account to model speech rhythm.

Temporal location of auditory events Even assuming the existence of an underlying temporal grid, there is no obvious general way to associate, to any auditory event, a "moment of occurence". There is actually no *a priori* reason to assume that the moment of occurence as conceived by the person who produces the sound is the same as the moment of occurence as perceived by the listener.

Some sounds are easier to temporally locate than others. While the moment of occurence of sounds such as clicks, with clear, short transients, is easily identified, things get more complex in other cases, especially speech sounds.

5.1.3 Complexity of Voks' rhythm control method

Voks' rhythm control method can be seen as a recipe for matching control points attached to the source samples with control impulses provided by the performer. The continuous nature of an audio sample clashes with the discrete nature of the control impulses, which, as noted at the end of section 2.3.3, makes any such algorithm far from trivial. In particular, due to the requirement of continuity of the time index, the performer cannot expect the target control point to be reached instantaneously as soon as the gesture triggering advancement to that point has been performed.

This has been observed in practice: when playing with Voks, performers need to factor in a delay between their rhythmic gestures and their audible result.

Complex process

Gestural rhythm control is first and foremost a *production* task. One of the most basic forms of rhythm production is the act of tapping one's finger on a surface, which has been studied extensively [106, 107].

The tapping gesture has much in common with the way rhythm is controlled in Voks. In tapping experiments, however, only minimal and simple feedback is provided — typically, only the haptic feedback of the contact between finger and surface, sometimes with the addition of simple auditory feedback such as clicks. In contrast, the conversion process between gesture and the resulting vocal rhythm, schematized in figure 5.1, is by no means straightforward (see section 2.3). To predict the outcome of their action, the person controlling rhythm thus has to rely much more on rhythm *perception* than in simple tapping experiments.

Speech rhythm, musical rhythm

Articulated vocal synthesis can be used to synthesize speech and singing alike, but spoken and sung voice differ in fundamental ways. As regards temporal organization, music is, contrary to speech, *isochronic* [51] — in a specific sense of the word —: rhythm in music is organized around a relatively regular pulse. Another important time-related difference is rate: depending on language, speech syllable rates usually range from around 5 to 8 syllables per second [19], whereas songs will seldom consistently maintain syllable rates higher than that on average (barring rap-related genres, where high syllable rates may be explicitly seeked out). For reference, repeated eighth notes at 120 BPM already correspond to a rate of 4 syllables per second, less than speech

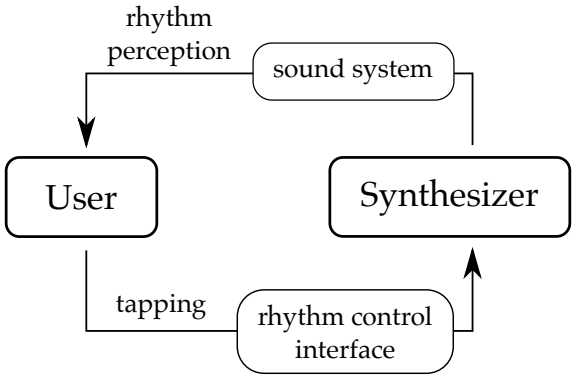


Figure 5.1: *Feedback loop involved in production of a vocal rhythm using Voks*

rate in most languages.

5.1.4 Current state of the study

This section details a study investigates the precision of rhythm control in the specific case of speech synthesis under various conditions. The aim was twofold: set up a protocol, including a graphical interface, for future, more serious tests about rhythm control, and identify general trends to select the most relevant conditions for more thorough testing with a lower number of conditions.

This study has made evident a number of observations, but suffered from its small scale as well as other issues detailed in section 5.4. As such, it can only be considered a preliminary one.

5.2 Protocol

The aim of the study is to evaluate accuracy of syllabic rhythmic control. Subjects need to be provided with an exact specification for rhythm following which they are to control Voks synthesis. Contrary to the case of musical rhythm, which may be represented accurately in symbolic form, there exist no system of notation for transcribing speech rhythm exactly. Thus we specify rhythm to subjects not in symbolic form, but as auditory stimuli that they are to reproduce.

The specification of rhythm using auditory stimuli is done in two distinct ways: *asynchronously* and *synchronously*. In other words, the test consists in two distinct task types: in the *imitation* task, subjects listen to the stimulus, and then attempt to reproduce it using synthesis. In the *synchronization* task, subjects attempt to reproduce the rhythm of the stimulus while they are hearing it, in a synchronized fashion.

Using a stimulus of the same nature as that of the subjects' production, rather than, say, clicks, whose temporal location can be determined without ambiguity, avoids a number of issues. First, when stimuli of radically different nature are heard simultaneously, effects appear that distort temporal perception — such effects have been leveraged in linguistic tests [54]. Second, it is not clear what it would even mean for sounds of different nature to be synchronized.

5.2.1 Subjects

Four subjects took part in the test. All had some musical experience and some amount of familiarity with performative vocal synthesis.

5.2.2 Corpus

The stimuli consisted of eight sentences in French featuring every possible number of syllables from 2 to 9. Each of those eight sentences had been recorded by two native French speakers, amounting to a total of 16 possible stimuli.

To limit the length of the experiment, for each subject and each sentence, a speaker was chosen at random, in such a way that each subject was randomly assigned 4 sentences spoken by the first speaker, and 4 sentences spoken by the second. Moreover, each subject A had a corresponding "dual" subject B whose assigned sentences were all different from A's sentences. That is, if some subject had as stimuli, for instance: sentence 1 spoken by speaker 1, sentence 2 by speaker 2, sentence 3 by speaker 2, and so on, then there was another subject who was assigned sentence 1 spoken by speaker 2, sentence 2 by speaker 1, sentence 3 by speaker 1, and so on.

This kind of random drawing ensured that

1. Subjects were exposed to both speakers equally, and
2. Both speakers were equally represented in the corpus.

5.2.3 Hardware and interface

Subjects were presented with an *Arturia BeatStep* MIDI interface, featuring pads and knobs, represented in figure 5.2, and a laptop whose screen displayed a graphical interface, represented in figure 5.2a. For a part of the task, subjects were also presented with a *Wacom Intuos Pro* graphic tablet.

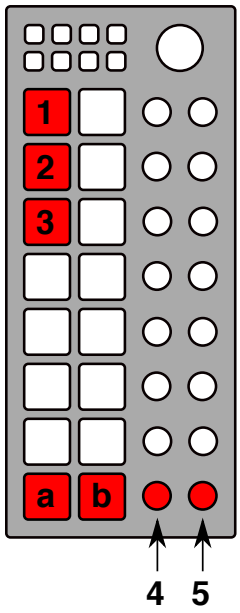
Subjects could hear both the stimuli and their own performances through a sound system. Depending on practical contingencies, sound was output either using headphones or loudspeakers. In all cases, audio output was stereo, which was necessary in some parts of the experiment.

Pads labelled 1, 2 and 3 in figure 5.2b were used as buttons, and simply mirrored buttons 1, 2 and 3 on the graphical interface. To avoid unfortunate mistakes, button 4 of the graphical interface was not mirrored on the MIDI interface, but by a key on the computer keyboard, slightly further away. Pads *a* and *b* were used to control rhythm during voice synthesis. In addition, knobs labeled 4 and 5, linked to graphical elements L and R, could be used to independently adjust the sound levels of the left and right loudspeaker/ear.

5.2.4 Conditions

The conditions of the test were comprised of a base condition, and variations around the base.

(a) *Graphical interface. Translation: 1.* Play original audio*; 2.* Record*; 3.* Listen to performance*; 4.* Save and proceed*; ii.* Dominant hand, No tablet, One finger



(b) *Schematic representation of the MIDI interface*

Figure 5.2: *Interfaces used in the test*

Base condition

During the base condition of the experiment, subjects were asked to do the following:

1. Listen to a stimulus whose playback was triggered upon pressing button 1.
2. Practice imitating the stimulus by controlling the rhythm of voice synthesis using pad *a*.
3. Once they felt ready, press button 2 to launch recording, perform one imitation of the stimulus and press button 2 again to stop recording.
4. Listen to their performance by pressing button 3.
5. Save their performance once they were happy with it, by pressing button 4.

Subjects were allowed to listen to the stimulus, record themselves and listen to their own performance as many times as needed and in any order between two presses on button 4 ("Save and proceed"), although they naturally could not listen to their own performance before recording one for the first time.

That base condition was then subject to a number of variations.

Monophasic vs biphasic control, and labeling

In the base condition, Voks' rhythm control algorithm relies on two control points per syllables — one corresponding to the vocalic nucleus, and one to the consonantic transition. Those control points are chosen using the heuristic method described in 2.4 (chapter 2). As a consequence, control in this condition is *biphasic*; that is, at every syllable, two interaction events between the subject and the interface are taken into account:

- A subject's **tapping** of a pad triggers a progression of the time index towards the next vocalic control point;
- A subject's **releasing** of a pad triggers a progression towards the next consonantic control point.

An alternative condition consists in making control *monophasic*. In other words, in the monophasic condition, only the instant when users tap the tab is taken into account.

Control points for monophasic control The heuristic method described in 2.4 was designed with two control points per syllable in mind. For monophasic control, we need a method for associating not two, but one timestamp to each syllable.

Intuitively, one would expect subjects to be most at ease when their tapping is simultaneous to the *psychological moment of occurence* of a syllable. Those are precisely the words used in [88] to introduce the notion of P-center. Thus we choose to use a version P-centers as control points in monophonic synthesis.

Among the many existing models for predicting the location of P-centers, the one described in [80] has the advantage of being easily computed in an automated manner. Thus the location of control points for monophasic control is taken to be that described in [80].

Number of fingers, dominant vs nondominant hand

Those conditions do not relate to the synthesis method itself, but to the way subjects interact with it. In the base condition, subjects were asked to only use one finger from their dominant hand. To determine the influence of those constraints, this condition was then altered in the two following, obvious ways: in one variation, subjects were asked to use two fingers, on adjacent pads a and b (see figure 5.2b); in another variation, they were asked to use their non-dominant hand instead of their dominant one.

Concurrent control of another parameter

In addition to control of rhythm, pitch is controlled using a graphic tablet. In order to let the dominant hand perform the pitch control task, we switch to the nondominant hand for rhythm control.

Imitation vs synchronization

In the base protocol, subjects could hear the stimulus *once* every time they pressed button 1, and were asked to *imitate* it afterwards. In the *synchronization* condition, pressing button 1 triggered playback of the stimulus in a loop, and they were asked to play it synchronously. Both the stimulus and the performance (including the output audio, as well as the gesture(s) captured by the control interface(s)) were recorded.

To help subjects discriminate their own productions from the stimuli, the stimuli were output through one loudspeaker (or ear, in cases where headphones were used), the subject performance through another. This is the reason for allowing subjects to independently adjust the sound level of each side.

Natural voice

In addition to performances using the synthesizer, subjects were asked to perform synchronized and unsynchronized imitations using their natural voice into a microphone, in order to compare Voks' abilities with those of human voice. To avoid recording the stimuli with the subjects' own voices, this condition had subjects systematically listen to them using headphones.

5.2.5 Analysis

Once a rhythm has been produced by subjects using synthesis, it needs to be evaluated. Although the gesture data of the imitation gesture can be recorded, there is no such data for the reference or for the vocal imitation. As such, the only type of data common to all imitations and to the reference stimuli is data related to the signal. Thus we locate a number of interest points on the signals. The interest points of a stimulus and the corresponding synthesized production should be the same (with only their location changing); what stimuli and subject productions have in common is the phonetic content, so we choose as interest points the phonetic boundaries, which contain a great deal of information about the temporal repartition of said phonetic content.

Note that although comparing the subjects' taps and releases to control points in the stimuli could seem to be a natural thing to do, there is no obvious relation between the two; hence the use of phone boundaries, a reasonably objective way to assess alignment in both the stimuli and the imitations.

Software tools exist for automatically finding out the temporal location of phonetic boundaries in a speech recording; those are called *aligners*. When that task is performed with the phonetic content given as an additional input, the alignement is said to be *forced*. All our alignments will be forced, for both the stimuli and the subject productions are based on known texts.

For this study, we use the WebMAUS forced aligner [70], which consists of a web interface allowing researchers to batch force align large numbers of audio recordings. Using WebMAUS, the temporal position of phoneme boundaries in each production (as well as in the stimuli) has been determined. In practice, the first and last boundaries (corresponding to the beginning of the first phoneme, and to the end of the last one, respectively) of each sentence have been ignored, due to the difficulty of controlling the endpoints for synthesis.

For each production, depending on the nature of the corresponding task (synchronization or imitation), a value has been computed:

Synchronization

In the case of synchronization tasks, each boundary of the subject production can be directly compared to the corresponding boundary in the stimulus. Thus a mean distance (in second) between production and stimulus boundaries was computed.

Note that there are two ways to compute a mean distance: we will denote by m_{abs} the mean of the absolute value of the distances between stimulus and production phoneme boundaries, and by m_{or} the mean the oriented distances (taking the value to be positive when the production is ahead of the stimulus, and negative when it is behind). Both ways of computing the mean have been used.

$$m_{\text{abs}} = \frac{1}{n+1}\sum_{i=0}^{n}|b_i - \hat{b}_i|$$

$$m_{\text{or}} = \frac{1}{n+1}\sum_{i=0}^{n}(b_i - \hat{b}_i)$$

where n is the number of phonemes in the sentence, b_i is the time of the boundary between the i-th and the $(i+1)$-th phonemes in the produced sentence (with b_0 is the starting time of the first phoneme, and b_n the ending time of the last phoneme). The $\hat{b}_i$ denote the analogues of the b_i for the stimuli.

Imitation

In the imitation paradigm, any value we compute must be invariant under temporal translations of the production relative to the stimulus. Thus the values m_{abs} defined above cannot be used as such. Instead, we treat the values as if they had been produced with some systematic error (whose value in itself is of no interest), and we use the least squares method to correct for that error: first we define the following cost function:

$$c(x_0, \ldots, x_n) = \sum_{i=0}^{n}(x_i - \hat{b}_i)^2$$

and we aim to find the value of s that minimizes

$$c(b_0 - s, \ldots, b_n - s)$$

The minimum is reached when $\frac{d}{ds}c(b_0 - s, \ldots, b_n - s)$ is zero, that is,

$$\sum_{i=0}^{n} -2(b_i - s - \hat{b}_i) = 0$$

rearranging:

$$s = \frac{1}{n+1}\sum_{i=0}^{n}(b_i - \hat{b}_i) = m_{\text{or}}$$

Thus the global shift s that minimizes the squares of the differences between the "ground truth" values $\hat{b}_i$ and the shifted produced values $(b_i - s)$ is the mean m_{or} of the oriented distances defined above. We can then compute the unoriented shifts after the global adjustment:

$$m_{\text{adj}} = \frac{1}{n+1}\sum_{i=0}^{n}|(b_i - \hat{b}_i) - m_{\text{or}}|$$

5.3 Results

This being only a preliminary study, we tried to identify global trends by computing the mean and standard deviations of the quantities defined in the previous section, for various condition combinations. Those have been plotted in figures 5.3, 5.5 and 5.4. Each of the plots shows seven means and standard deviations, each computed on all eight sentences for all subjects, respectively under the following conditions (details about the conditions can be found in section 5.2.4):

Nat. No synthesis, only the subjects' natural voice.

Cont. Pts Base condition; only monophasic tasks; control points defined using the heuristic method of section 2.4.

P-cent. Base condition; only biphasic tasks; control points defined to be the P-centers using [88]'s model.

Base Base condition.

Hnd 2 Using the subjects' nondominant hand.

2 Fngs Using two fingers.

Tblt Simultaneously controlling pitch.

Those preliminary tests suggest the following:

- Subjects "do better" with their natural voice than with the synthesizer; that is, mean shifts are smaller with natural voice. However, they have the same order of magnitude, with mean shifts in the base synthesis condition being around twice the mean shifts in the natural voice condition.
- Use of the subjects' secondary hand seems slightly detrimental to the performance in some instances, as is the use of two fingers instead of one.
- The results of monophasic and biphasic modes are comparable.
- Adjunction of pitch control is detrimental to the performance.

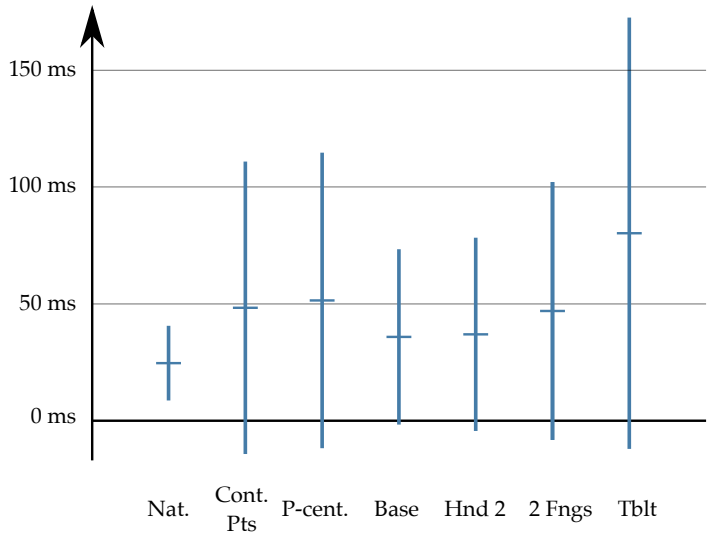


Figure 5.3: *Mean and standard deviation of the absolute shift m_{abs} for imitation tasks for all sentences with all subjects. See text for condition details.*

5.4 Discussion

The small scale of the study makes it difficult to draw precise quantitative conclusions. The experience of setting up and conducting it, as well as the issues that arose, allow us to make a number of observations nevertheless.

5.4.1 Existence of speed regimes

Controlling the rhythm of Voks in the context of this test has made evident a fundamental difference between rhythm control of an articulated voice in music and in speech. In our musical experience with Voks, the mental relationship between gesture and synthesis was rather direct: one press/release corresponded to one syllable, with a delay inherent to the rhythm control algorithm; with a little training, performers were able to compensate that delay.

With everyday speech, which is faster than most singing, the typical duration of a syllable gets small enough that its order of magnitude is the same as that of the delay of the synthesizer. Thus it is no longer feasible to mentally associate one gesture to one syllable. In fact, the pad used for rhythm control is often tapped close to the beginning of the syllable triggered by the previous tap. As a result, performers naturally adopt another strategy whereby syllables are not controlled atomically: the series of taps as a whole dictates a sentence-wide tempo. From the subjects' admission, the stimuli featured subtle rhythmic variations which they could perceive but did not manage to reproduce.

This observation makes interpreting the results of a study such as this one difficult, but is

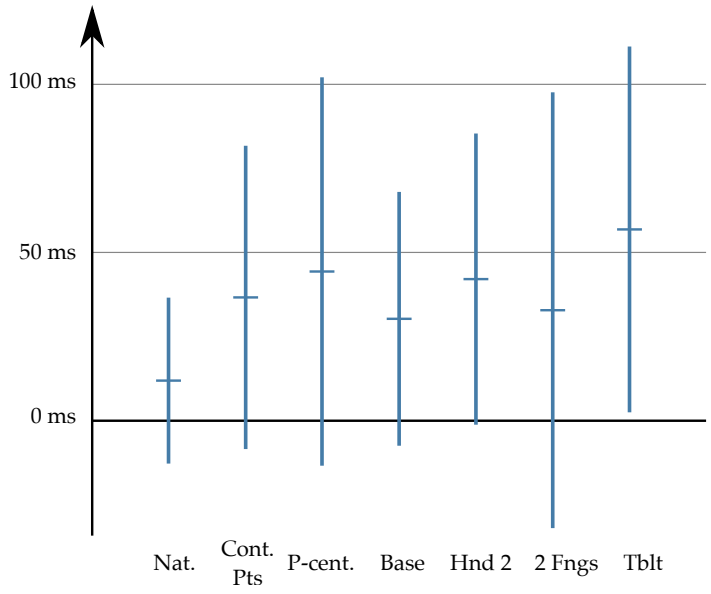


Figure 5.4: *Mean and standard deviation of the oriented shift m_{or} for synchronization tasks for all sentences with all subjects. See text for condition details.*

interesting in its own right.

5.4.2 Issues related to control with more than one finger

Another observation arises from the previous one. When considering whole sentences to control rhythm with taps, rather than acting on the note level, the performer has to take the total number of syllables into account. If using one finger, this is not an exceedingly complex task — though errors sometimes arise, with performers stopping before the whole sentence has been played, or conversely, tap one time more than required. If alternating between two fingers, however, performers must plan which finger will be last to tap, an information dependent on the parity of the number of syllables of the sentence to be played.

Thus alternating between fingers to control spoken voice has two consequences:

- An additional, non-trivial planning task is required of performers;
- A dissymmetry is introduced between sentences of even or odd length. Such a dissymmetry might or might not affect the synthesized sound.

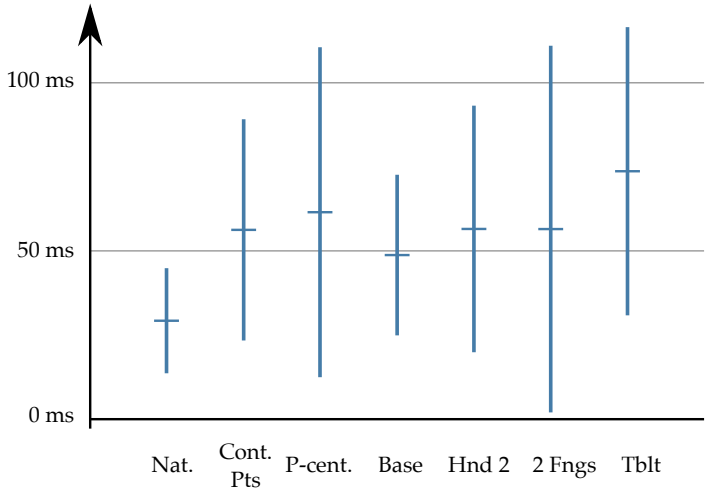


Figure 5.5: *Mean and standard deviation of the absolute shift m_{abs} for synchronization tasks for all sentences with all subjects. See text for condition details.*

Chapter 6

Beyond Fixed-Text Synthesis

Contents

6.1 Introduction

One of the overarching goals of performative vocal synthesis (PVS) is to be able to specify with ease any text of a given language, at the time of synthesis. To our knowledge, a system enabling this has not yet been devised; all existing performative systems make trade-offs on one aspect of PVS or the other:

- Glove-Talk I [44] restricts the space of accessible utterances to those composed of a set of predefined morphemes; Cantor Digitalis [48] restricts it to utterances exclusively composed of non-nasalized vowels.
- Voder [39] and Glove-Talk II [45] require months of training, and even then, the output, although intelligible, is slow and does not resemble a human voice. Digitartic [47] has also been judged too difficult by its creators.
- Voks (see chapter 2) can be used to play any text, but the text is not specified in real-time. Even though text-to-speech integration frees performers from the need to prepare material prior to the performance, the full sentence still has to be known before it can be synthesized.

Up to this point in the present document, only synthesis based on a predetermined text has been discussed. This is a key limitation of Voks, and it shapes the way that performers interact with the instrument: from our experience, the need to go through the tedious process of data preparation before playing a new text with Voks tends to make us stick to the same text and requires the design of uncommon strategies for improvizing, and limits the effective vocabulary of the performer to a few dozens sentences. Although there is nothing wrong with written music, the idea of granting the user of a synthesizer more textual flexibility is appealing. As mentioned in the introduction, however, such a flexibility tends to come with costs — difficulty of control, slowness — which make this a challenging problem.

In this chapter, we describe two attempts to tackle this problem. The first, Voks-TTS, is an extension of Voks which automatically generates prepared data, without fundamentally changing the way that Voks itself works. The second, F-Voks is a more ambitious — and, as of yet, immature — new synthesizer. Part of that synthesizer has been developed by ourselves in the context of this doctorate; another part has been developed by intern Paul Chable under the supervision of Christophe d'Alessandro.

6.2 Automatic data preparation with Voks-TTS

The process of data preparation by a user in Voks consists in two tasks:

1. Record themselves uttering the desired text.
2. Manually determine the location of control points and input them in a text file to be given to the system as input.

Task 1 is difficult to carry out in the context of a performance. On one hand, it needs to be performed in a reasonably quiet environment, a condition seldom satisfied in such a context. On the other hand, the very act recording, which by definition cannot be carried out in silence, could disrupt a performance, unless it is somehow incorporated into the performance itself.

Task 2 must be performed by an expert user, able to infer the location of control points from the look of a spectrogram. Even in the presence of such a user, the task can still take several minutes per second of audio, which is impractical in a live performance.

This section describes Voks-TTS, a system that combines the real-time aspect of Voks with the

flexibility of a text-to-speech system to make specifying a text during a performance possible. Although the text is not yet truly specified in real time, that system represents a further step in the direction of real-time text specification.

6.2.1 Principle

Voks-TTS bypasses the need for recording and preparation of an audio sample. It integrates a text field in which the user can type text, then automatically generates and prepares an audio sample for synthesis with Voks. The user can then play the artificial sample just like they would a pre-recorded one.

6.2.2 Implementation

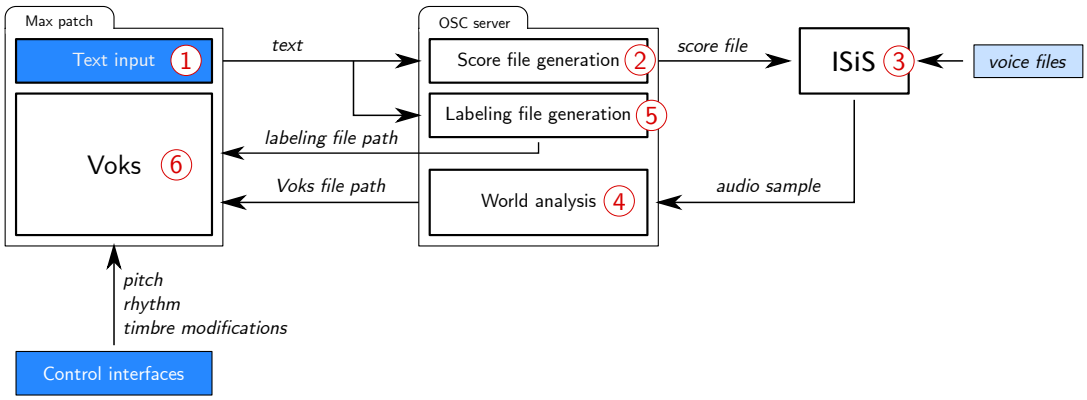


Figure 6.1: *Voks-TTS architecture. Blue boxes represent inputs. The "voice files" box corresponds to files that determine the nature of the artificial voice; here we just use voice files provided with ISiS by default.*

Voks-TTS' architecture is represented on figure 6.1.

Voks-TTS is composed of three building blocks: a Max patch, which contains the same audio engine as Voks, as well as a user-editable text field, a Python server, and IRCAM's singing synthesizer ISiS (see section 1.3.2). The Max patch and the Python server communicate via the OSC protocol; the Python server is, in turn, able to run ISiS to generate audio. Voks-TTS works in the following way:

1. To generate a new sample to be sung, users must first type text in the Max interface text field. The text must be in phonetic format (more precisely, X-SAMPA format), which eliminates possible phonetic ambiguities — and happens to correspond to the text part of ISiS' input format.
2. Once confirmation is sent by the user, the text is sent via OSC to the Python server, which generates a score in ISiS' input format, including both the text and a "melody", the specified melody being a regular sequence of notes with the same pitch. This means that the temporal organization of syllables is known, which will be useful in step 5.

3. ISiS is called on the score file and generate the corresponding audio sample.
4. The audio sample is analyzed using World, as is done when preparing a recorded sample for Voks.
5. The audio sample is automatically labeled. Knowledge of the sample's syllabic organization, due to the fact that it was generated with a controlled rhythm, allow us to know where vocalic nuclei as well as consonantic transitions lie. The exact control points are chosen in the middle of those zones. This does not provide the same accuracy as a manual labeling would, but it gives a playable sample.
6. The location of the World analysis files and the labeling file are sent to the Max patch, where they are fed to Voks as input files. The user can then control Voks with gestural interfaces, as described in 2.2.2, to play the generated sample.

6.3 Disyllabic representation and control

Free-text performative vocal synthesis involves two connected, though separate, problems:

- That of designing an **input method** for the user to specify the text to be synthesized;
- That of **synthesizing sound** based on a symbolic representation of a text being received in real-time.

The symbolic representation of text is thus of central importance. In this section, we introduce a paradigm for free-text real-time vocal synthesis, based on a specific symbolic representation, the *disyllable* representation. We have developed a synthesizer based on that representation.

Intern Paul Chable, under the supervision of Christophe d'Alessandro, has then designed a gestural input method for free-text. By connecting it with the disyllable synthesizer, he has developed F-Voks, a system which synthesizes text on the fly based on the performer's gestures.

6.3.1 The disyllable representation

To make textual improvisation possible, we need to choose a representation of text that operates at a level smaller than the whole sentence. Being the basic blocks of our mental model of speech, phonemes seem to be at the adequate level of abstraction. However, the continuous nature of the speech phenomenon, conflicting with the discrete nature of that model, causes coarticulation phenomena [72]; those make it difficult to synthesize text based on a naïve phoneme-based representation.

To overcome that problem, diphone synthesis [16] uses a representation whose atomic symbols are *diphones*, pairs of consecutive phone symbols, to account for mutual influence of consecutive phonemes around the boundary between two phonemes. A diphone synthesizer then takes as input a sequence of diphones such that the second phoneme of each diphone is the same as the first phoneme of the next diphone.

In the case of vocal synthesis controlled syllabically, it is simpler, in terms of implementation, to group speech by syllables instead of phonemes. That way, the performer gestures are more

directly related to the units used in the symbolic representation. For that reason, we use a variant of diphone representation that uses one symbol per syllable, with each symbol straddling two syllables. Thus each symbol represents the transition from one syllable to the next, and we call the symbols *disyllables*, by analogy with diphones.

Technically, disyllables are short sequences of phonemes, just like diphones, but they no longer need be 2 phonemes long. Instead, we require that they start and end with either a vowel (so that they start/land in a vocalic nucleus) or silence (for the beginning/end of a phrase). In fact, both the diphone and the disyllable representation are instances of the "vocalic sandwich" introduced in [14], with the set S of "phonemes where splicing is acceptable" being the set of every phonemes in the diphone case, and the set of vowels in the disyllable case.

6.3.2 Generating and labeling disyllables

Combinatorics of disyllables

The number of possible diphones is bounded by n^2, where n is the number of phonemes in the considered language. In contrast, since the number of phonemes in a disyllable is, at least in theory, not bounded, the number of possible disyllables is infinite. In practice, additional constraints specific to the language considered enables us to restrict the space of possible disyllables: in French, for example, one is unlikely to encounter consonant clusters of more than four consonants. In addition, diphthongs (barring diphthongs resulting from semivowels, which we treat as consonants here) are seen as constitutive of several distinct syllables.

Nevertheless, the number of possible disyllables remains high. This being a preliminary work, we chose to further restrict the disyllables that we will be working with, in two ways:

- Only considering disyllables of the form VCV (where *V* means either a *vowel* or silence, and *C* means either a *(semi)-consonant* or nothing in the case of a diphthong);
- Only considering a restricted set of vowels and consonants.

Disyllable generation

In diphone synthesis, diphones usually originate from recordings by a human speaker, tasked with reading a text containing all needed diphones. This would also, in principle, be an option in the disyllable approach; for the sake of simplicity, however, and keeping in mind the preliminary nature of this work, we chose to use a machine instead of a human to record the disyllables. Thus all needed disyllables were generated using the ISiS [112] vocal synthesizer.

ISiS was asked to generate samples with a constant pitch. In ISiS, pitch can only be specified at the note level; any microprosodic events are managed by the synthesizer itself, and it provides no interface to control them. To play samples using a system in the Voks family, samples with a known, flat pitch are needed. Thus the generated samples were further preprocessed with World to adjust pitch.

Disyllable labeling

The disyllable synthesis approach relies on "gluing" audio samples at points where such a gluing is possible, and as imperceptible as possible; that is, stable and similar parts of the signal. To that end, a *gluing* point has to be identified in the starting and ending vowel of each disyllable, such that it is stable spectrally, and the timbre at that point corresponds clearly to the corresponding vowel. The task of identifying such points on the audio signal is called *labeling*.

If using actually recorded audio samples, labeling needs to be done either manually or using semi-automated digital signal processing algorithms. In our case, however, ISiS' mode of operations can be taken advantage of: since ISiS' interface provides a way to control the temporal progression of its output, gluing points can be chosen in a systematic way.

6.3.3 Gestural grammar for disyllable synthesis

The symbolic representation is only an intermediate format for gestural input interfaces to communicate with a synthesis engine. How gestural data is to be converted into that format remains to be defined. This section describes a *gestural grammar* that specifies a way to turn gestures into a stream of disyllables, and the implementation of that gestural grammar.

The phonetic interface

Up until this point of this work, the interfaces considered have provided a few accessible parameters, allowing users to control, with their hand, the position of a point in some low-dimensional space. The complexity inherent to the linguistic aspect of speech production warrants the use of an interface more elaborate than that, that captures the movements of the hand. This is possible in three dimensions, with sensors attached to gloves, as has been done in Glove-Talk [44, 45], or cameras and image recognition algorithms [104]. This is also possible in two dimensions, using a tablet sensitive to touch, provided that it captures more than the position of a single pointer.

Capturing the movements of the hand in two dimensions with a tablet may not provide access to as much data as in three dimensions. However, it presents the advantage of not requiring the setup of cameras or sensors, relying instead on a single, commercially available device, the graphic tablet. In addition, a two-dimensional surface constitutes something tangible for users to interact with, making it easier to orient themselves spatially than if their hand were to move in the air.

The Sensel Morph is a commercially available tablet sensitive to touch. It captures the pressure applied to each point of a grid of about 355×614 points on a 138.5 mm×240 mm surface. The Sensel real-time output data is readily exploited in Max thanks to the existence of an external developed and actively maintained by Max's parent company, Cycling '74. The external provides access both to the matrix of pressures on each point, as well as higher-level data deduced from that matrix. That higher-level data is what we use for real-time text specification. The higher-level data consists of a list of control points, each paired with its position on the tablet, its bounding box, and other data.

The text selection method

In line with the notion of disyllable, our gestural grammar will allow performers to specify text syllable by syllable — that is, to each possible syllable will correspond one gesture. In our simplified setting which features no consonant clusters, all text is a sequence of alternating vowels and consonants. Thus a syllable is determined by one consonant and one vowel, and a syllabic gesture needs to contain the information for one consonant and one vowel.

Vowel selection

Principle As noted in section 1.2.3, oral vowels are traditionally seen as lying in a three-dimensional space whose dimensions are height, backness and roundedness. The roundedness dimension is relatively narrow, with vowels being merely either rounded or non-rounded. As a consequence of this narrowness, the three dimensions are easily collapsed into two. In terms of representation, this makes it possible to represent vowels on a page, as is usually done. In terms of synthesis control, the example of Cantor Digitalis has shown that a two-dimensional space is sufficient to specify French oral vowels.

Thus we use for vowels an approach similar to that of Cantor Digitalis, with the position of a finger in a rectangle controlling the vowel of each syllable. A few differences set this control method apart from Cantor Digitalis' one, though:

- In Cantor Digitalis, the hand selecting vowels is not assigned any other task. Here only the thumb of a hand is able to select vowels; the other fingers are used to select consonants as described in section 6.3.3.
- In Cantor Digitalis, vowels are selected among a two-dimensional continuum. In our case, although the selectable vowels are distributed on a space that mimics the vowel continuum, there is actually only a finite number of accessible ones: the vowel control surface is divided into zones each assigned to a vowel. Section 6.5.4 features a discussion about possible solutions to this limitation.
- Nasal vowels are included. This is described in the next section.

Nasal vowels The oral/nasal distinction can be considered to be an additional dimension of vowel space. However, the intent to keep the space two-dimensional, as well as the relatively low number of nasal vowels (that is, compared to the total number of vowels in French phonology) lead us to adopt another approach: we simply arrange nasal vowels on an area contiguous to the area dedicated to oral vowels.

The correspondence between oral and nasal vowels is thus not taken into account. If controlling vowels in a continuous space (see section 6.5.4), this would make it impossible to transfer continuously between an oral vowel and its nasal counterpart. The compensation for this sacrifice is simplicity and the preservation of a 2-dimensional control space for vowels.

The final vowel space, including both oral and nasal vowels, is represented on figure 6.2.

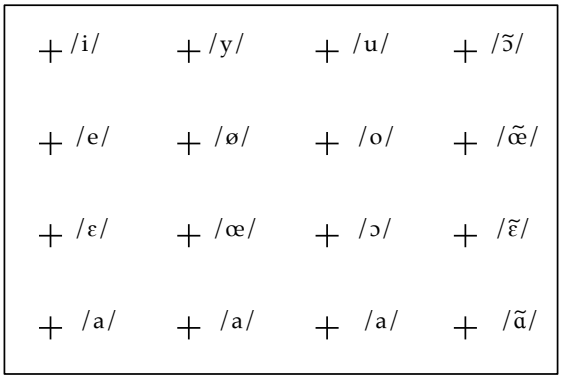


Figure 6.2: *F-Voks' discrete vowel space*

Consonant selection

As described previously, vowels are distributed in a continuous, low-dimensional space. In contrast, (pulmonic) consonants are classified according to a number of discrete categories: place of articulation, manner of articulation, voicing.

To reflect that categorization, we classify gestures of all fingers, save for the thumb (busy with vowel selection already), on the tablet surface, depending on the following characteristics:

- Number of fingers in contact with the tablet (thumb excluded);
- Position of the fingers in contact, when static;
- Direction, if any of the motion of the fingers on the surface.

Each gesture is then associated to a consonantic phoneme from the French phonetic inventory. The following general principles are followed:

- "Static" configurations, i.e. those corresponding to simple contact of the fingers on the tablet, are associated with occlusive consonants;
- Fingers moving on the tablet are associated with fricatives and liquids;
- Fingers in the same position/direction are associated to the same place of articulation;
- The number of fingers corresponds to voicing, with one finger corresponding to unvoiced consonants, and two fingers to voiced ones. Nasals are assigned gestures with three fingers.

An exception to that last rule is the phoneme /l/. Along with /ʁ/, it has no unvoiced counterpart in the French inventory, so we choose to do as if /l/ were the unvoiced counterpart of /ʁ/.

Using those rules, all consonants from the French inventory can be produced (excluding semivowels /j/, /w/ and /ɥ/). The absence of consonant is denoted by contact with one finger as close to the thumb as possible.

The gesture/phoneme dictionary for consonants is given in figure 6.3. Each entry of that table consists in a schematic representation of a gesture and the corresponding IPA phoneme.

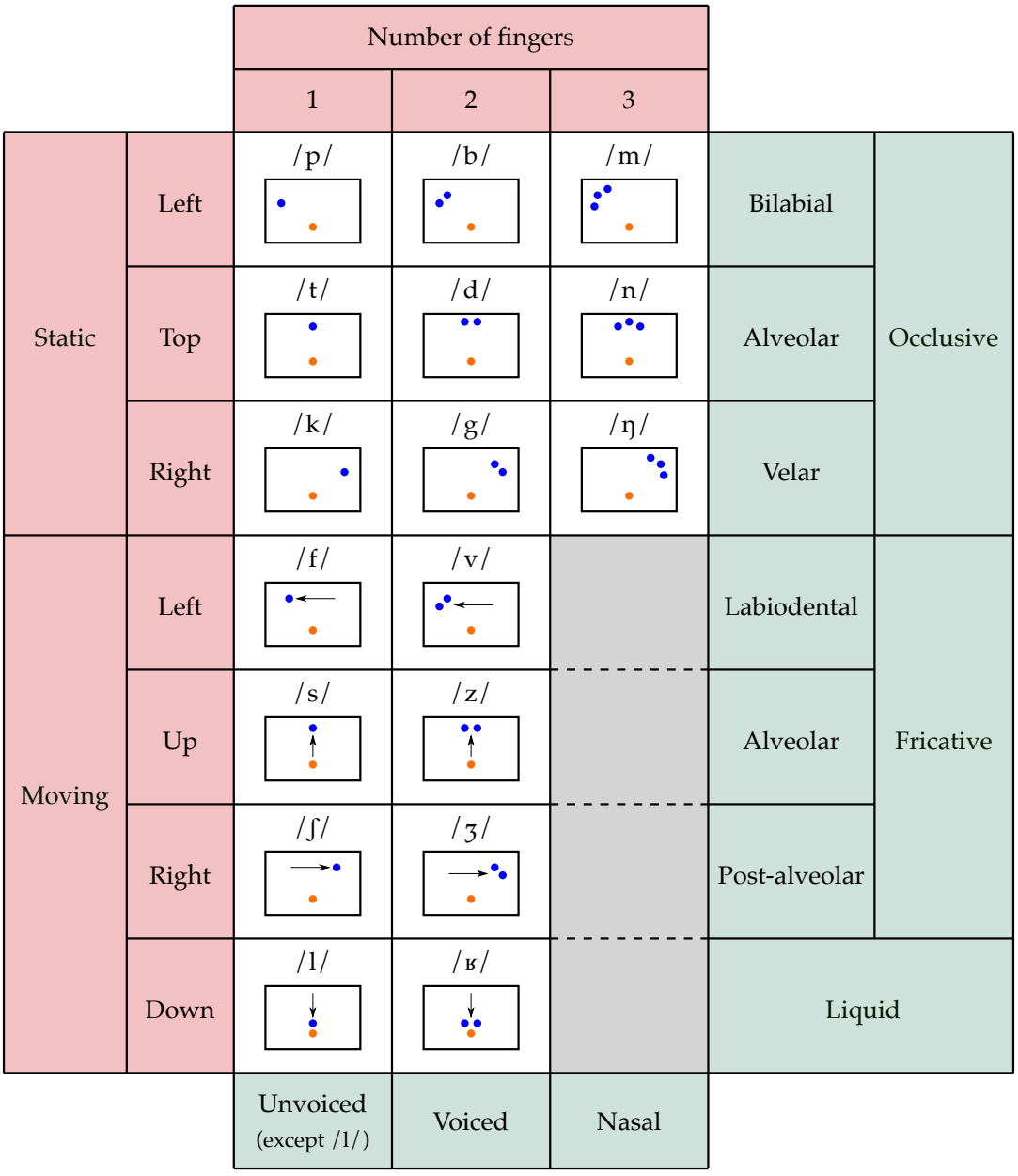


Figure 6.3: *The gestural grammar for consonants. Each entry features a phoneme, and a schematic representation of the corresponding gesture. Cells with a red background describe the gesture associated with each phoneme; cells with a green background describe the phoneme itself. Each rectangle represents the Sensel Morph tablet surface; orange dots represent position of the thumb, and blue dots represent other fingers.*

6.3.4 Gesture recognition

Just like the gestural grammar is conceptually separated into gestures for vowels and consonants, the recognition process comprises one part for vowel recognition and one for consonant recognition. The vowel recognition process is fairly straightforward; it simply consists in determining the vowel closest to the thumb on the space pictured in figure 6.2, which is affixed as a mask on the tablet.

Consonant recognition is more involved: in contrast with vowel recognition, where the position of one finger may be directly exploited, the simultaneous motion of several contacts has to be taken into account, and thresholds have to be set from which a consonant is to be considered detected. Given how simple the gestures are currently, such work may conceivably be performed manually, by manually specifying detection rules; the process, however, would be tedious. Besides, the exploratory nature of this work forces us to assume that the gestural grammar chosen here is only temporary, thus an approach that would be easily adapted to new gestures should be preferred.

We thus need a classification algorithm, one that takes as input gestures of fingers on a two-dimensional surface, and outputs symbols from some alphabet. We do this with the help of the MuBu software library.

MuBu

MuBu [114] is both the name of a data structure and of a software library for creating and manipulating this data structure in the Max/MSP language. The MuBu data structure (short for *MultiBuffer*) aims to represent streams of time-aligned data of arbitrary types in a unified format. This includes audio signals, contours of synthesizer parameters and audio descriptors, and more. Here we use multibuffers to store trajectories of one or several points in 2D space, representing the trajectories of fingers on a tablet.

The MuBu library is a set of Max/MSP objects that allow for creating and interacting with multibuffers. In particular, objects for training and using statistical models using multibuffers are included in the library [56]. The particular model to use depends on the temporality (synchronous/asynchronous) of the required task and the nature of the task itself (classification/estimation).

For recognition of gestures among our discrete set of consonants, the task is one of classification, and it is performed asyncronously, meaning that we want to only get one result at the end of the gesture, dependent on the whole history of the gesture, rather than a set of likelihoods at each instant. The statistical model included in MuBu that meets these requirements is that of Hierarchical Hidden Markov Models.

Hierarchical Hidden Markov Models

Hierarchical Hidden Markov Models (HHMM for short) are a variant of Hidden Markov Models (HMM) that was introduced in [49] in response to some shortcomings of HMMs. Like (first-order, discrete) HMMs systems, HHMMs model systems with a discrete set of states where the probability of transitioning to a certain state depends only on the current state. Like in those

HMMs, one does not directly have access to the sequence of visited states, but only to a sequence of *observables* related to the states with a certain probability distribution. In fact, as noted in [49], HHMMs can be seen as particular cases of HMMs; the specificity of HHMMs lies in the fact that "base" states are grouped into collections of states, themselves grouped in higher-level collections, and so on. Each collection q^d is itself a Markov model whose states are the elements (q_i^{d+1}) of q^d, with an associated transition matrix, and a distinguished state q_e^{d+1} that signals when to switch from the Markov model associated to q^d to another Markov model on the same level, based on the transition matrix of the higher-level Markov model of which q^d is a state.

It is possible to take advantage of the additional hierarchical structure of HHMMs as compared to HMMs to learn more efficiently on data featuring different hierarchical scales and long-term correlations. Two examples are given in [49], that of speech and that of cursive handwriting. That last application is of special interest to us, given how conceptually related finger trajectories on a flat surface are to handwriting.

We train two HHMMs to perform a classification task; one HHMM detects gestures consisting in simple contacts (corresponding to occlusives, see tablet 6.3), the second detects gestures where fingers move while in contact with the tablet surface (corresponding to fricatives and liquids).

Each time a consonant is detected, the vowel corresponding to the current position of the thumb is observed. The combination (in that order) of the detected consonant and vowel constitute the next syllable to play.

6.4 Disyllable synthesis

6.4.1 Determining the disyllable to play

Section 6.3.4 describes how gestures are converted into consonant-vowel syllables. As stated in the introduction, however (and as will be explained in more detail in section 6.4.2), the audio engine takes as input *disyllables*, not syllables. Before any digital signal processing is performed, the first step is thus to convert the input syllable to a disyllable. This is easily done by observing the last vowel of the previous syllable, and prepending the new input syllable with it. This ensures that contiguous disyllables share a vowel, which is necessary for sound continuity.

6.4.2 Disyllable synthesis

To turn disyllable input into audio, we use the SuperVP vocoder (via Max/MSP) again. Whereas in Voks, synthesis was based on moving around a virtual playhead in a single buffer, here we use two buffers so as to perform the transition between two successive disyllables. We use the following two objects from the Max SuperVP library:

- `supervp.scrub~` for controlling the playback position in each buffer, just like in the case of Voks. This object is assigned a buffer, takes as input a time stream $\tau(t)$, and plays, at each instant t, a sound perceptually similar to the audio in the buffer at time $\tau(t)$. Thus $\tau(t)$ can be seen as a "virtual playhead". Here we use two instances of this object, each attached to a distinct buffer. We will refer to the corresponding virtual playheads as "playhead 1" and

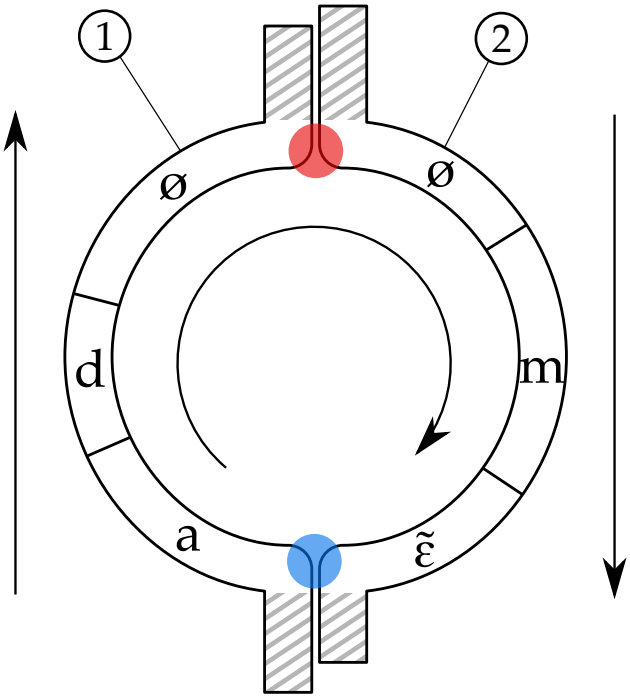


Figure 6.4: *Schematic representation of the two buffers used in disyllable synthesis. Buffer 1 contains the disyllable /adø/ and is read from bottom to top. Buffer 2 contains the disyllable /ømɛ̃/ and is read from top to bottom. The disyllables are "glued" at the red point, which corresponds to the same phoneme /ø/ in both buffers. Buffers are read clockwise on this representation. Once buffer 1 is no longer being read, it can be replaced with another disyllable whose first vowel will have to coincide with the last vowel of buffer 2 — here, /ɛ̃/. The buffers will then be glued at the blue point. Extremal, hashed zones of the buffers are not used.*

"playhead 2".

- `supervp.cross~` to perform cross-synthesis between the two buffers. This object takes two audio streams as input, as well as a coefficient $\lambda(t)$ between 0 and 1, and plays, at each time t, a mixture of the first audio stream, weighted by $(1 - \lambda(t))$, and the second audio stream, weighted by $\lambda(t)$. Here the inputs to this object are the output of each `supervp.scrub~` object.

The synthesis process is illustrated in figure 6.4, in a situation where the sequence of phonemes /adømɛ̃/. Let us describe the process step by step, in this situation. We will assume that the disyllable /adø/ has just been synthesized, and that the command for synthesizing the disyllable /ømɛ̃/ has just been received.

0. The synthesis method is based on the assumption that no new command is received before the synthesis process for the last one has been completed. Thus we assume that the disyllable /adø/ has been played. The cross-synthesis object plays only the audio corresponding

to buffer 1 (i.e. $\lambda(t) = 0$). Playhead 1 is positioned on the red dot on figure 6.4: the vowel /ø/ in buffer 1 is currently playing. The contents of buffer 2 do not yet matter.

1. Upon receiving the command for a new disyllable, the corresponding audio is loaded from the disk into buffer 2. The buffer contents are now as shown on figure 6.4.
2. Playhead 2 is positioned on the red dot, so both playheads are located on the same vowel (here /ø/).
3. The coefficient $\lambda(t)$ continuously goes from 0 to 1 in a fixed transition time, so that the cross synthesis object transitions from playing only audio coming from buffer 1 to only audio from buffer 2.
4. Playhead 2 starts moving from the red dot to the blue dot; listeners hear the phonemes /ømɛ̃/.
5. The situation is now symmetrical to the one at step 0; the system, now playing the vowel /ɛ̃/, is ready to receive a new disyllable command, swap the contents of buffer 1, and start the synthesis process over again.

6.4.3 Signal transformations

In chapter 2, section 2.6, we presented a number of expressive transformations that can be controlled with Voks. Those include pitch control as well as diverse timbre transformations. The same modifications may also be performed using F-Voks, which relies on the same underlying technology as Voks, SuperVP. To control them, we use the same graphic tablet, a Wacom Intuos Pro. Performers using F-Voks thus have two tablets to work with: the Sensel Morph, for text selection using the grammar described in section 6.3.3, and the Wacom Intuos Pro, for pitch and timbral control.

6.5 Perspectives

The two projects described in this chapter are both attempts to answer the question of textual freedom, but they are at very different stages. Voks-TTS is a mere extension of Voks, and as such, it is close to being usable in practice — although it has remained a prototype as of now. F-Voks, on the other hand, is both more ambitious and much less mature.

6.5.1 Voks-TTS

Voks-TTS gives satisfying auditory results, but is composed of separate parts — a Python server, a Max interface — which have to be launched together at once and made to communicate. As such, efforts should now be directed towards the development of a version packaged adequately so as to be easy to transmit and launch. Even at the algorithm level, however, there is still room for improvement, notably as concerns the automated labeling process, which for now remains very basic.

6.5.2 F-Voks

As for F-Voks, it is far from ready to be used in practice. The main issues are limited textual possibilities, and lack of expressivity.

Expanding the text space

The method presented here makes it impossible to synthesize utterances featuring consonant clusters. The restriction is related both to our grammar and to the synthesis method; on both fronts, the problem is related to the combinatorics of consonant clusters, of whom there are many more than single consonants: counting clusters both inside words and across words, [79] observes several hundredths of distinct consonant clusters in a French corpus. Making it possible to synthesis consonant clusters with F-Voks would require advancements both in the control method and the synthesis method.

Control In French speech, consonants occur fast enough (with durations in the order of magnitude of 100ms [94]) that concatenating several ones makes it impossible to specify them separately using the grammar presented here. To select consonant clusters, a new gestural paradigm thus has to be found, either by:

- Relying on gestures that can be performed and chained much faster than the ones presented here. In a sense, this is what happens during natural speech production: articulators move fast, playing one phoneme after the other.
- Or by specifying consonant clusters at once with a single gesture. This would require a grammar complex enough to be able to select a cluster among the hundreds possible, but simple enough that it can be learned and used by humans.
- A last option would be to combine those two approaches, by mapping single gestures to common and/or fast clusters, and leaving other clusters to be performed via a sequence of gestures.

Synthesis In terms of synthesis, the disyllable approach also faces a combinatoric problem when consonant clusters are introduced. Diphone synthesis is already reliant on a relatively high number of audio samples, given that a recording of each combination of two phones that might appear consecutively is needed. With disyllables, the combinatorics are even greater: possible consonant clusters are manifold, and recording them all would be impractical.

A possible solution would be to synthesize all possible disyllables beforehand, which would spare the need for recording them. They would still, however, take up a large amount of space on disk. An alternative would be to adapt the system presented here to perform diphone synthesis instead of disyllable synthesis.

6.5.3 Improving expressivity

As of now, attempts at synthesizing voice using F-Voks have yielded a sound that lacks expressivity. There are certainly several factors responsible for this, not least of which is the lack of

training on this new instrument. Increasing responsivity might also be an improvement worth exploring. The current version of F-Voks waits for the performer to have performed the whole disyllabic gesture before triggering advancement to the next disyllable, which results in a delay between gesture and sound. Designing a gestural grammar closer to the synthesized sound, and replacing disyllables with the finer notion of diphones might be a way to make the system more responsive.

6.5.4 Vowel continuum

Finally, an interesting improvement to F-Voks would be the integration of continuous vowel specification. Currently, vowels are chosen among a discrete set, but the interface for vowel selection is a Cantor-like, continuous tablet. As such, it is only natural to wish for the ability to play vowels in between those discrete categories. Such intermediary vowels might be approximated by performing a weighted average of the spectral envelopes of vowels from the discrete set. There is no obvious way, however, to combine such a continuous vowel synthesis method with disyllable or diphone synthesis.

Chapter 7

An Engineering Perspective

Contents

7.1 Introduction

The development of such a piece of software as Voks involves a number of technical considerations, especially as regards technological choices. This chapter is an attempt at allowing readers to benefit from the experience gained while developing vocal synthesis systems during this doctorate.

In the first section, we take a look at the requirements that constrain technological choices. We then list out the options available to the developer of a vocal synthesis system. Sections 7.4 and 7.5 give an account of the advantages and difficulties encountered with the options that we ended up going for in this doctorate in the case of two projects. The first project is the development of Voks, whose design this dissertation covers in detail. The second project is reimplementation of Cantor Digitalis, another vocal synthesizer. The principle behind Cantor Digitalis, the reasons for its reimplementation and the project itself are recounted.

7.2 Constraints

The very notion of software covers an exceedingly large range of objects. Admittedly, a characteristic shared by virtually all software is the ability to turn input of some sort into output of some sort; but the nature of that input (respectively, output), and the manner in which it gets fed into (resp. extracted from) an algorithm varies drastically from system to system. The developer of a digital music instrument certainly faces many challenges unknown to, say, a scientist developing some scientific software that performs mathematical computations on data contained in a file on disk and outputs some new data in the form of another file.

In this section, we list the constraints that an implementation of Voks has to respect. Those were instrumental in the technical choices made during development, which will be detailed further in the chapter.

Broadly speaking, constraints may be classified into two types: constraints directly related to the features of the software itself, and constraints related to the context in which the software is developed and used. We will call the former *internal constraints*, and the latter *external constraints*. Naturally, the boundary between those categories is not clearly defined, nor are they mutually exclusive.

7.2.1 Internal constraints

Internal compatibility

The first requirement for a software system is for its constituents to be compatible with one another. Admittedly, if there is any incompatibility between the technologies used, it will inevitably become obvious once building a functional system proves impossible. Nevertheless, to avoid wasting time, compatibility constraints should be pondered as early as possible in the development cycle.

The choice of a main programming language is a critical one when it comes to compatibility, as it acts as a glue which binds together other technologies such as libraries.

Audio-related constraints

A digital music instrument naturally needs to be able to process and output sound. While there are libraries for inputting and outputting sound files in most popular programming languages, the requirement that synthesis be performed in real time narrows down the technological options.

A specificity of Voks is the fact that it is sample-based, thus relying on pre-existing audio data to function. This implies the need for:

- Choosing a format for the source data to be saved in and stored on disk,
- Being able to load the data on disk in the working memory of the program.

Finally, depending on the underlying audio engine, spectral processing might have to be performed. Real-time spectral processing constitutes a paradigm distinct from processing in the time domain:

- When performing time-domain signal processing, one deals with a limited number of audio signals which are, at least conceptually, being dealt with at every sample.
- In the spectral domain [119], computations are only performed every R samples, but each computation involves N complex values, where R is the *hop size*, the number of samples between two consecutive spectral analysis windows, and N is the size of the analysis window, usually in the order of the hundredths or thousands. What is more, the developer of a synthesizer based on spectral-domain processing will have to include a way to convert data between the audio domain and the spectral domain — at least from audio to spectrum, and possibly the other way round.

Gestural control

The ability to take as input real-time data from outside elements, most notably control interfaces, is critical for a music instrument. Several communication protocols, some of which are listed in section 7.3, are available, some of which have become standard. Depending on the language, library and protocol used, communication might be either easy, difficult or downright unfeasible.

Graphical interface

Some applications require a graphical user interface (GUI) to be usable. Again, some languages and/or libraries make it easy to develop a GUI, while the task might require hours of learning and development with others.

7.2.2 External constraints

Existing technologies

The technologies already in use in the environment where a system is developed must be considered. Using a known technology, one might be able to integrate software bricks that have been developed for other projects into one's own, for instance; the experience of one's colleagues can also be taken advantage of in the form of technical discussions. On the contrary, reflecting about the technologies in use can help become conscious of their limitations and decide whether it is worthwhile to break with tradition and start with something new.

In our case, Voks was developed in a team whose members had made extensive use of the Max/MSP programming language: previous vocal synthesizers, including Voks' initial version, Vokinesis, were coded in it, and other lab members coded in Max as well. In particular, fellow

doctoral student Thomas Lucas, with whom we had many an opportunity to discuss scientific and technical matters, coded the MonoReplay instrument mostly in Max, and used the SuperVP vocoder as its audio engine.

With that said, help can usually be found with just about any technology on dedicated spaces on the internet. For subjects related to audio programming, in addition to language-specific forums, the Audio Programmer Discord server is a particularly helpful community, with active users competent in many areas related to digital audio.

Development speed

Development of the same piece of software can take a drastically different time to be developed depending on the programming language. The importance of development speed in a programming language is immediatly obvious; less obvious is the fact that development speed is not necessarily constant. Using some technologies might save time at first, only to result in systems difficult to extend and/or maintain; some languages have a steeper learning curve than others, but taking the time to learn them might make implementation of complex easier than with others.

Transmission

The fact that the software developed during this doctorate is, by definition, "research software", does not imply that there are no plans to disseminate it for others to use — starting with colleagues, but ultimately including the general public. It is thus essential that it be practical to transmit it to other potential users.

This implies several things: that the system is compatible with the users' hardware and software; that it is packaged in a way that it is feasible for them to acquire it. The legal aspect of transmission must also be taken into account.

Portability Many options exist to write software once that can be used on many hardware architectures and/or operating systems (OSs). Software written in most common programming languages runs on most common systems; many software libraries are available for several systems as well. It is often the case, though, that those solutions are imperfect: code written for one system frequently has to be slightly adapted to run on another, and the compilation process often differs. There are tools, called *build systems*, that allow developers to write code once for many systems, though those add a new layer of complexity. In any case, no code is guaranteed to run on a given OS or architecture, as long as it has not been tested on it. Even software that has been tested on some hardware with some OS is not guaranteed to run on another version of that OS on the same hardware.

The fact that a piece of software runs on a system does not guarantee that it runs properly; various compatibility issues may arise. Libraries linked dynamically might not be present, or not found, on the host system. In the case of a digital audio instrument, there might be obstacles to accessing to the graphics or audio system, or to getting the software to talk with control interfaces.

Packaging Getting users to run software on their machine (prefereably with ease) is also essential. Distributing software in the form of an executable has the merit of simplicity from the point of view of users, although it requires producing and distributing one distinct file for every OS and architecture. In the case where the executable is not totally self-contained, typically by linking to dynamic libraries, one might still need to check for the presence of dependencies on the host system, preferably providing a way to install them if required.

Distributing software in the form of source code is also possible. This saves the need for distributing one executable for every configuration, but it shifts the burden of compilation to the user, requiring from them a level of technical proficiency substantially higher than that generally needed to run an executable.

Lastly, software may depend on multimedia assets such as images and audio, which must then be distributed along with it — either bundled with it, or copied to the host filesystem during installation. This is particularly relevant for a sample synthesis system, which by definition relies on existing audio data.

Licensing Legally, no one is allowed to use software unless the developer (or their employer) has granted them a license to use it. When choosing the conditions in which a piece of software will be transmitted, the license of the technologies that it relies on must also be considered.

There are several types of licenses, from more to less permissive. On the most restrictive end are proprietary licenses. Such a license restricts the freedom of the user of a piece of software to use, tamper with, or share it. Some proprietary licenses are paid, some are not. Even when using non-paid software, a proprietary license might restrict the right to share the software, which might be problematic in the case of a library.

The counterpart to proprietary licenses, so-called *free*[1] licenses, come in different flavors, but they all allow users to inspect and modify the source code. Some are *restrictive*: while their users are allowed to run, modify and study the source code, sharing is subject to conditions such as reusing the same license. Others are *permissive* in that they do not impose any restrictions on sharing.

The most famous restrictive free software license is the GNU General Public License (GPL), which forbids diffusion of software based on GPL-licensed software under another license. The CeCILL license was developed as a French alternative to the GPL, compatible with it, but better-suited to the French legal system. Popular choices for permissive licenses include the MIT and BSD licenses.

Note that it is also possible to provide software under several licenses at once. The user can then choose under which terms they will use the software. A common such scheme is licensing a library under both the GPL and a proprietary, paid license: anything developed using the library will then either need to be licensed under a free license, or require the developer to pay the owner of the library.

[1]The word "free" here refers to freedom, not to price — as mentioned above, some proprietary licenses might not require payment.

Perennity

Any piece of software is dependent on many other technologies, starting from the hardware that it is running on. Other dependencies can include the operating system, drivers for control interfaces, various software libraries and many more. Even before digital instruments, perennity has been an issue for live electro-acoustic music, as pointed out by [121, 9]. A more recent, thorough discussion of those issues is given in [12]. In any case, dependence of a software synthesizer on other technologies, though inevitable to some extent, must be pondered insofar as it jeopardizes its perennity.

Sustainable development

Use of most of today's technology has some impact on the environment, and attention should be paid to the consequences of new developments, notably when it comes to energy and hardware consumption. The NIME (New Interfaces for Music Expression) community has been considering that impact, and has made available resources to aid the reflexion around related issues. [81]

7.3 Technology options

In this section, we draw up a list of technologies that one might consider when developing an application such as Voks. Those technologies include programming languages as well as vocoders, and protocols for communicating between different programs or parts of a same program.

First we turn to programming languages. Note that the choice of programming language is not exclusive, as it is perfectly possible for several languages to coexist in a single system. With that said, adding a new programming language — or any other technology, for that matter — to the equation increases the complexity of the system, and the decision must thus be carefully pondered.

7.3.1 Programming language

Max

Max is a programming language targeted specifically to implementers of interactive real-time applications. Its development started in the '80s [101] and it has been evolving since, gaining various features related to audio and real-time interaction, including digital signal processing, video, and communication using the MIDI and OSC protocols. It also includes a system for visually building graphical user interfaces. In addition to its built-in features, libraries have been developed by the community, allowing programmers to perform various tasks related notably to sound processing [114] and gestural interaction [56].

General mode of operation Max is a visual programming language based on the dataflow programming paradigm: programmers code not by typing text, but by modifying a graph of connected so-called objects — the equivalent of a traditional programming language's functions —

laid out visually in a dedicated interface. Programs are still represented as text files (more specifically, under the JSON format [65], a popular one for serializing data as text [117]), though the text form of Max source code is not designed to be read by humans, and a dedicated, proprietary editor is needed to do any non-trivial development work in Max.

Max programs can be either interpreted, using the same proprietary program that is used to edit Max source files, and compiled into a standalone application. The editor and interpreter being proprietary implies that Max programs are dependent on Max's mother company, Cycling '74, during their whole life cycle. Having only one option for the editor, in particular, makes it difficult to seamlessly integrate Max coding into a software development workflow that includes e.g. tests and version control.

Max programs can be tracked using version control systems such as Git, SVN or Darcs, though the above considerations make it challenging in some regards.

Features A number of extensions have been integrated to Max over the years, extending its domain of application. Those include MSP, a signal processing extension that enables audio programming, as well as Jitter, which provides operations on a custom data type, the matrix, enabling image and video processing.

In addition to such extensions, which are now officially part of the language, libraries have been developed by the community, allowing programmers to perform various tasks related notably to sound processing [114] and gestural interaction [56].

Using Max Max is targeted not only at programmers, but also artists who need to make ideas concrete without having to spend too much time learning new concepts and technologies. As such, simple programs are not excessively verbose, and the ecosystem makes running them a straightforward task. Max is thus well-suited to implementation of simple concepts and prototypes. As we will see in section 7.4.1, however, Max suffers from a number of issues which one should carefully consider before using it for more ambitious projects.

Pure Data

Pure Data was developed by the creator of Max, Miller Puckette, as an attempt to fix some of its shortcomings. [102] As such, its underlying logic is essentially the same as Max's. Pure Data is very different from Max in other regards, however:

- Pure Data is free and open-source; it is released under a permissive license.
- Pure Data is minimalistic when compared to Max. Development is still done in a graphical interface, but that interface is much simpler than that of Max, consisting in simple black lines, rectangles and characters on a white background. The number of available objects is also drastically reduced compared to that of Max, and it lacks features such as graphics manipulation, or advanced graphical user interface building.

A welcome feature is the ability to embed the audio processing that happens in a Pure Data program into a program written in one among the most common general-purpose programming

languages (namely, C, Java, Objective-C, C++, Python, and C#). The dataflow paradigm, which is arguably better suited than the imperative one to some forms of audio programming, can then be used for strictly audio-related tasks, while a traditional language is used for the rest.

FAUST

FAUST [93] is an audio programming language based on a combination between functional programming and a restricted form of dataflow programming.

FAUST programs are written in plain text, although they are textual representations of diagrams [92]. The choice of textual representation makes it possible to do things that would be very difficult in a graphical language, such as iterations (FAUST's restricted counterpart to "for" loops in traditional programming languages) or writing complicated equations. It also spares the need for a graphical editor, and makes it easier to work with version control. Besides, the FAUST compiler is able to output a graphical representation of the program. As a counterpart, new programmers have to learn how to convert their diagrammatic ideas into text using FAUST's syntax.

FAUST is well suited to synthesis algorithms that can be expressed as mathematical equations in the time domain. In fact, the compiler is able to output automatically-generated documentation which includes mathematical equations that describe the behavior of any valid FAUST program. On the other hand, FAUST cannot be used for sound manipulation in the spectral domain, although there have been, and still are, efforts to overcome this limitation. [68] Thus the adequacy of FAUST with regards to the algorithm to be implemented must be carefully pondered.

One of FAUST's major strengths is the ability of its compiler to compile programs into a wide variety of formats. Those include the executable format, to be used directly, but code in other programming languages, notably C, can also be generated. Formats to be used with other audio technologies are also available, including Max and Pure Data externals, as well as common audio plugin formats.

Faust is free and open-source. The compiler itself is licensed under the restrictive (but still free) GNU General Public License (GPL), but anything generated by it is not subject to such restrictions.

C++ and JUCE

Many general-purpose programming languages feature real-time audio libraries. A popular choice is C++, which has both low-level facilities, very valuable for audio programming, and higher-level ones, which makes it a more than viable candidate for development of complex applications. C++'s wide popularity is also an advantage, in that it is easy to find resources and progrrammers that use it.

JUCE is a "Swiss army knife" C++ application framework, which provides facilities for a wide range of tasks, most notably audio and graphical interface design. It is cross-platform: in principle, programs can be developed once and run on diverse platforms, including Linux, MacOS and Windows. In practice, some changes may be needed to adapt code that compiles on one system to another, but the process is generally fairly smooth.

The power and versatility of C++ and JUCE come at a price: learning to be proficient in C++ takes time, more so than many other languages, both general-purpose (such as Python) and audio (such as all the previously mentioned languages). JUCE adds another layer of complexity.

JUCE is a popular choice among developers of audio applications for end users. It is open source, and released under two separate licenses: the free, but restrictive GNU GPL, and a commercial license which is paid if the organization that uses JUCE makes more than a certain revenue.

7.3.2 Vocoder

For our purposes, it has also been necessary to choose a vocoder. Not only have two choices been considered, but a version of Voks has been developed with each of those.

World

World is a library compatible with C and C++. The code can be classified into two modules: an analysis module, and a synthesis module. The analysis module takes an array of audio data and turns it into a data structure representing the spectral representation described in section 1.4.3. The synthesis module takes that representation and produces sound as output.

By default, the functions from the World library do not allow synthesis based on real-time; however, the code can be adapted to make that possible. On the other hand, analyzing audio in real time is impossible.

World being a C/C++ library, it is only compatible with C/C++ applications by default. However, many other languages feature so-called foreign-function interfaces, which make it possible to use code from another language (usually, at least C). In some other cases, the library can be adapted to be used in other languages; that is the case with Max's and Pure Data's external mechanism. Note, however, that that task may not be trivial.

World is released under the permissive BSD license.

SuperVP

Like World, SuperVP is a C++ library, but it has also been packaged into an executable, a Max library, and is one of the audio engines of the graphical program AudioSculpt [11].

It is essentially a phase vocoder, with the addition of techniques for spectral envelope estimation and transformation [109, 110] as well as transient detection and processing. [111, 108] As such, it allows independent change in pitch and sound, but also spectral envelope conservation/distortion, as well as cross-synthesis, among other manipulations. It can be used in real time.

SuperVP is released under a proprietary license. It is paid, except for the Max library, which can be used for free.

7.3.3 Communication protocol(s)

There are several protocols for communicating data in real time between programs and control interfaces or other programs.

MIDI

The MIDI protocol was designed in the 80's for the purpose of communicating digital data between synthesizers and computers. MIDI messages can trigger the start or the end of a note, with a given pitch and velocity, the change of a control parameter or another event such as a tempo change.

The simplicity of MIDI has allowed it to become and remain very popular up until today: most musical interfaces and software synthesizers today are equipped to send or receive MIDI data, and audio programming languages or libraries are generally MIDI-capable.

MPE

Despite its versatility, the MIDI protocol influences the way to think about a performance. The MIDI paradigm of individual notes whose every parameter are determined once at their beginning is especially limiting.

MPE is a protocol which builds on MIDI to provide control over notes after their beginning, making it possible, among other things, to continuously bend the pitch of individual notes. Being based on MIDI, MPE data can be transmitted in the same way as MIDI data; nevertheless, the sending and receiving devices should be able to produce/understand MPE data. Some control interfaces output MPE data; they can often be told apart from plain MIDI interfaces, which are often piano-like, because they usually feature some kind of continuous sensors, such as a sensitive surface.

OSC

OSC is a protocol which can be seen as a newer alternative to MIDI. It is also based on discrete, digital messages. The main difference with MIDI is the amount of information that messages contain. Whereas MIDI messages are constrained by a somewhat restrictive format, OSC messages are labeled with URL addresses and can contain several types of data, among which strings of unbounded size and high-precision (64 bit) numbers.

OSC has not supplanted MIDI, even though it is more capable, but most programming language are able to send and receive OSC data.

Analog data

Although any form of programming deals with digital data, data needs not be transmitted in digital form, and there are many interfaces and synthesizers that communicate using analog data.

Sound is arguably the most important of data that can be transmitted analogically, since any device that measures or produces sound must deal with analog sound at some point. Of course,

analog audio data can be treated as actual sound intended to be heard, but it can also be diverted to transmit other kinds of data, leveraging the ability of a wide range of devices to deal with audio, and the availability of audio analog-to-digital and digital-to-analog converters.

Another kind of analog data is fairly common in some electronic audio circles: control voltage. Control voltages were used to control synthesizers before digital devices became widely available. The usage of the method has decreased since, even though its simplicity and versatility still attracts some users. Many control interfaces and synthesizers feature control voltage inputs and/or outputs, often together with MIDI.

Other protocols

The aforementioned formats are but a few among the infinity of possible formats for real-time data transmission. In fact, some hardware interfaces come with their own specific protocol, and one has no choice but to deal with it. Thus the protocol to use is often not directly a choice *per se*, but only one of the aspects to consider when choosing an interface.

7.4 Development of Voks

The general mode of operation of Voks has been described in 2. Here are the technological choices that had to be made during its design and development:

- That of a set of control interfaces, both continuous and discrete;
- That of a vocoder, featuring the ability to analyze audio, modify it, and resynthesize it;
- That of a format for storing syllabic labeling data;
- That of a programming language to glue all of these together, and perform tasks such as playback time management.

7.4.1 Language

Previous vocal synthesis systems developed in our team were generally coded in Max. We trusted, somewhat blindly, that our predecessor's choice was a good one, not only for their use case, but also for ours. This choice had a big impact on development.

Max's strength

Max's main strength lies in the ability to do audio and real-time programming with ease, without needing to dive into low-level details. The dataflow programming paradigm does require some getting used to, but the visual nature of the language, as well as the wide number of available functions, make it easy to start implementing ideas.

Scope issues in Max

From our experience, however, programs written in Max become very difficult to deal with — that is: debug, modify and expand — once they are composed of several interacting parts. This is

mainly due to limitations in Max's limited notion of *scope*.

The basic unit of data in Max is the *message*; the notion of message is analogous to that of notifications which arises when using the observer pattern in object-oriented programming [57]. Max messages, however, are different from observer-related notifications in a number of ways:

- Max messages are ubiquitous. There being no notion of variable, messages are the main way that information gets passed around.
- Mas messages are simple: they consist in an atomic piece of data such as a numeric value or a string, labeled by a string identifier. Max is not designed to pass around complex data.
- There is no subscription mechanism: messages are brodcast to the whole program, and the decision of whether or not to take them into account is based solely on their string identifier.

Thus Max messages are inherently global; to introduce a semblance of scope, one needs to make up a way to encode the starting point/destination of any message in its identifier string. In our experience, this approach has proven to be tedious and prone to confusion; the difficulty of tracking down the whereabouts of incoming and outgoing messages makes debugging Max programs difficult. Generally speaking, Max's message system has been a major hindrance in the development of Voks.

Coupling between coding and program execution

A notable characteristic of Max is a certain coupling between programming and executing a program, and between form and content. Programs are being executed live constantly as the editor is open, and the source, in return, can be modified to some extent by simply executing it; changing the way that the code is displayed, for instance by resizing the editor window, or by viewing bringing up the code of a subfunction, has repercussions in the saved textual representation of the code; if designing a graphical interface, it is being designed in the same editor as that which is used to write Max code.

This coupling has virtues, as it makes directly evident the relation between the programmer's actions and their result in the program being developed, but it also has a number of unfortunate consequences, such as:

- Inability to modify the source code of a running program without having an additional instance of the program running in parallel — and possibly interfering with the first instance;
- Trouble keeping track of the changes made deliberately when coding as opposed to those that are a result of viewing the source code and/or executing the program;
- Trouble reasoning on the behavior of the program.

Opacity of Max programs

Whereas in imperative languages and other textual languages, features such as variable names make programs self-documenting to an extent, the logic of Max programs is usually difficult to grasp. The general rule stating that code is easier to write than to read is exacerbated with Max. As with any other language, commenting code is crucial and improves on that situation

somewhat; still, it tends to be difficult to maintain, modify and extend Max programs.

7.4.2 Vocoder

The choice of a vocoder was a crucial one. Two vocoders were tried, World and SuperVP. They have a comparable, satisfying sound quality, but work very differently.

World

World being under the form of a library left us with the responsibility to choose in which way the analysis and synthesis would be performed, and how the data would be exchanged. Since the synthesis takes a significant amount of time, the choice was made to allow the result of the analysis to be saved on disk for future reuse, hence the packaging of the analysis code into an executable.

Storing the result of the analysis on disk spares us the complexity of interfacing the analysis module with Max in real time. However, a file format for representing that data still had to be chosen and implemented. Since the World analysis results in vectors that evolve in time (the aperiodicity ratio and the spectral envelope), the file format associated to the Jitter extension was chosen. The Jitter format allows the storing of matrices of arbitrary dimension on disk, and the loading of such matrices in Max, where a large collection of dedicated objects allow the Max programmer to manipulate them. Thus the Jitter extension was also used for manipulating the World representation once loaded from files on disk. It was only natural, then, to package the synthesis module into a Max object that takes Jitter matrices as input and outputs audio.

In practice, those natural-seeming choices ended up bringing their fair share of complexity. Implementing a Max external in C, in particular, involves using unusual idioms with only the Max application programming interface to get by. The use of the Jitter extension, for matrix manipulation, and of the MSP extension, for audio output, constitute two simultaneous, additional sources of complexity. The compilation process for Max externals itself is, from our experience, quite obscure, with only example projects, but no written documentation, to rely on.

Finally, in terms of usability, the process of performing an analysis whose result is stored on disk ended up being a major hassle which, together with the need to manually label syllables, significantly limited artistic spontaneity in practice.

SuperVP

SuperVP was already wrapped into Max objects, which removed the need to make the choices mentioned in the previous paragraph. In terms of general workflow, the objects are much simpler than the ones we developed around World, since they directly read from audio data in real time.

The counterpart to this simplicity is a less direct access to the vocoded representation: whereas with World, it could be directly modified using Jitter matrix manipulation objects, the only transformations available with SuperVP for Max are those that are made accessible via input parameters.

7.4.3 Control interfaces

The ease with which control interfaces were connected varied greatly. Any interface that output MIDI can be connected to a Max program without trouble.

The graphic tablet that we mainly used, the Wacom Intuos, was a more delicate matter. The communication relied on a Max external whose last version was released in 2013; the external itself relied on a Wacom driver itself no longer officially supported, which in turn required the operating system version to not be too recent. Possible solutions to that problematic state of affairs are still being investigated; a good candidate is the Sensel tablet, but it remains to be tested in real contexts.

For rhythm control, the several interfaces that were considered all had their advantages and drawbacks. Buttons attached to MIDI interfaces were the most convenient solution implementation-wise, but no MIDI button was found small enough to fit in a thereminist's hand while performing. The different solutions that were tried, including buttons and pressure sensors, all involved some additional hardware to connect to the computer, but the devices were able to transmit MIDI data.

The theremins that we used interestingly featured two kinds of output: control voltage and audio output. There were two control voltage outputs, one for pitch and one for intensity. Since the theremin was used as a control interface and not as a sound generator, control voltages again seemed like the natural choice for receiving data. Again, the natural choice turned out not to be the most convenient one: some additional hardware was needed to convert the voltage into digital data. Furthermore, one of the voltage output ranges was non-conventional, enough so that even more hardware was needed to rescale it. After that first experience, a new version was developed which used the theremin's audio output, converted to digital form by a standard audio interface. An existing Max object was then used to detect pitch and intensity. The second solution, though conceptually more elaborate, was more manageable in practice.

7.5 Reimplemention of Cantor Digitalis

In parallel to the work on Voks, this doctorate was also an occasion to work on another vocal synthesizer: Cantor Digitalis (see section 1.3.1) was developed prior to this doctorate, and was reimplemented during it. The reason for that reimplementation is the dependence on proprietary technologies at risk of becoming obsolete, notably Max/MSP, and the will to integrate the synthesizer with other software and hardware.

The mode of operation of Cantor Digitalis is fundamentally different from that of Voks; as a consequence, the reimplementation was a good opportunity to experience other kinds of technological choices. In this section, we describe Cantor's mode of operation, we go over the issues related to the pre-existing version, and the technological choices made during its reimplementation.

Cantor Digitalis allows for synthesis of non-articulated voice; in other words, in enables the synthesis of vowels, but not consonants. Musical parameters such as pitch and intensity, as well as phonetic parameters, are respectively input either via the use of a graphic tablet, or via a graphical

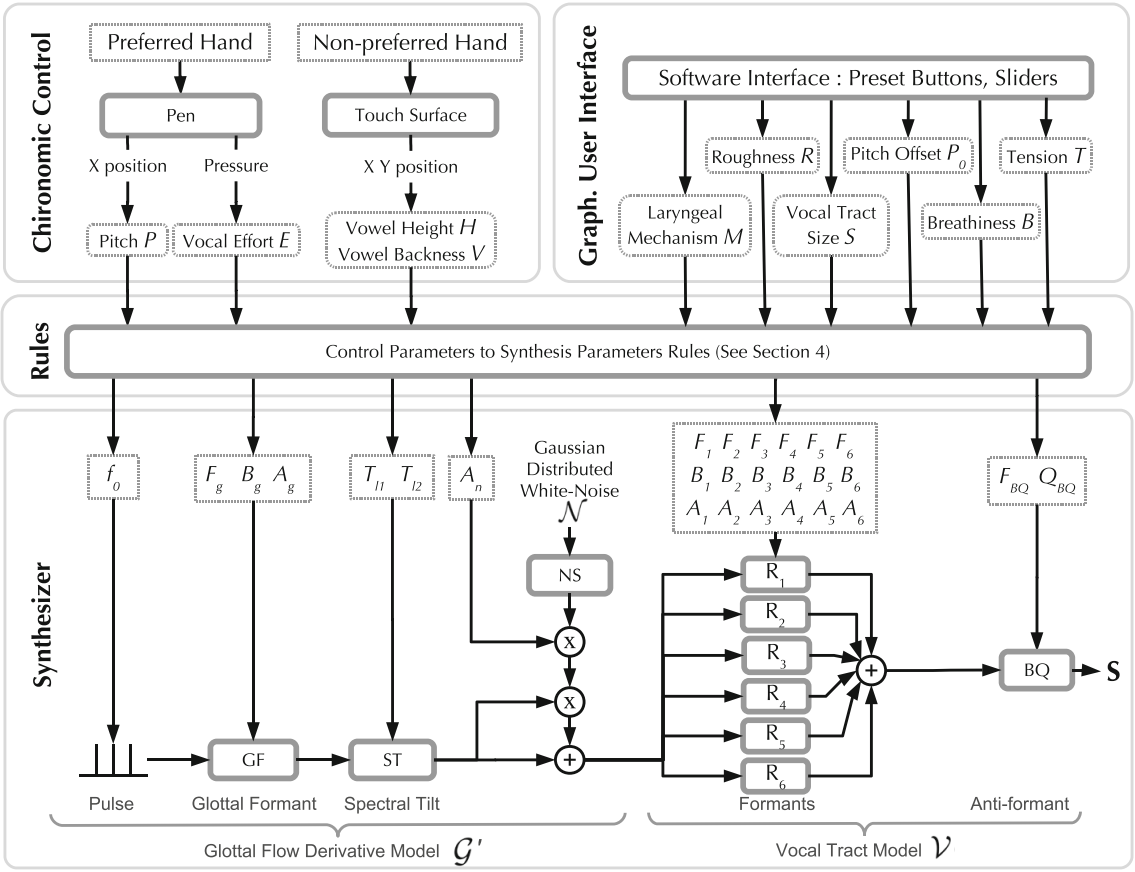


Figure 7.1: *The architecture of Cantor Digitalis.* Figure reproduced from [48]

interface. The general architecture of the system is summarized on figure 7.1.

7.5.1 Cantor Digitalis' synthesis algorithm

Cantor Digitalis is a purely parametric synthesizer: it produces sound based on a finite number of parameters specified in real time. The synthesis engine is based on the source-filter model.

In Cantor, the source (*glottal flow derivative model* on figure 7.1) is an impulse train of a certain fundamental frequency, filtered by a set of filters whose characteristics are determined based on a spectral model of the glottal flow [37]. This signal is then mixed with filtered noise, to add some breathing to the sound.

In line with the notion of formants discussed in section 1.2.2, the filter is modeled as a sum of resonant filters (R_1 to R_6 on figure 7.1), characterized by a center frequency, an amplitude and a bandwidth.

Low-level and high-level parameters

The synthesis algorithm depends on many parameters: the fundamental frequency and amount of noise in the source, the center frequency F_1 to F_6 of each formant filter, are but a few examples of those.

Most of those parameters are rather low-level, and consequently are not suited for direct control by a performer. For that reason, higher-level parameters, with musically meaningful names, such as "Tension" and "Vocal effort", are defined; those are the parameters that will be made directly accessible to the performer. Values of the lower-level parameters are then defined as functions of those higher-level parameters ("Rules" block on figure 7.1).

Interface

Some of the high-level parameters may be controlled using a graphic tablet, using a stylus. The graphic tablet used is a Wacom Intuos Pro, which is also able to detect finger contact besides stylus position and pressure.

The tablet controls synthesis as follows:

- The tablet's horizontal dimension is mapped to pitch,
- The vertical dimension may be mapped to another parameter such as tension or breathiness,
- Stylus pressure is mapped to vocal effort,
- Finger position in a rectangle is used to continuously select the vowel to be digitally sung.

To accomodate the needs of users without a graphic tablet, a so-called "demo mode" allows users to control some parameters using a mouse and a keyboard.

As versatile as the model of graphic tablet used may be, it is not enough to control all of Cantor's high-level parameters at once. Thus all the parameters are accessible through a graphical interface.

Presets

A system of presets allows users to save the values of all parameters and recall them all at once. Some pre-existing presets are available and allow users to directly work with specific voices ("Tenor", "Soprano", ...), or sound effects obtained by pushing certain parameters to the extreme ("Foghorn", "Lion"). In particular, some presets make the range of accessible pitches so low that individual glottal impulses are heard in place of a continuous sound with an audible pitch.

Limitations of Cantor Digitalis

This paragraph describe issues pertaining to Cantor Digitalis. In short, those have to do with compatibility, software obsolescence and proprietary software. Those issues are very general in software development as a whole. Various standardization efforts exist to remedy them. The Faust programming language (see section 7.3.1) adopts another, interesting approach; section 7.5.2 describes how it was used in the specific case of Cantor Digitalis.

Cantor Digitalis was developed in Max (formerly known as Max/MSP), a proprietary, graphical programming language. To run a Max program, one needs to have an operating system compatible with the version of Max used. Although Max programs are stored as plain text, the format used is not designed to be modified directly. Instead, to modify a Max program, one needs to own a license for the Max software, an integrated development environment for the language with the same name. This means that, even though the code of Cantor Digitalis is technically open source, in practice, there are still proprietary barriers to inspecting and modifying its code.

Cantor Digitalis may not be used with any model of graphic tablet other than the one it was developed for. It relies on a Max "external", that is, a piece of software that can be accessed in Max, but is not part of Max itself. This external, `s2m.wacom`, can be downloaded at `https://metason.prism.cnrs.fr/Resultats/MaxMSP/`; it takes care of communication between the graphic tablet and Cantor Digitalis.

At the time of writing, `s2m.wacom` was last updated in 2016, and does not work if the latest drivers for the Wacom tablet are installed. One needs to install older drivers, and even then, the tablet is not guaranteed to be recognized in Max. This is a serious issue, as the `s2m.wacom` external is an integral part of Cantor Digitalis.

7.5.2 Reimplementation

This doctorate is part of a project whose goal is to develop an autonomous module integrating sound synthesis algorithms for use in both educational and artistic contexts. Cantor Digitalis could not be directly used on the platform selected for the project, a Raspberry Pi computer running Linux. Thus a choice of technology had to be made for reimplementation.

In such an application, with a well-identified hardware platform running a specific operating system, the problems described in section 7.5.1 are not too critical. However, all other things being equal, implementing the synthesis algorithm using a technology that makes it easy to transfer the algorithm on other technologies can only prove beneficial. As it turns out, the programming language Faust is such a technology.

Faust is particularly well-suited to algorithms expressed in the time domain. The description of Cantor Digitalis in [48] is composed mostly of description of digital filters with an infinite impulse response, in the form of transfer functions that are readily converted into temporal-domain recursive equations, and expressions of the values of low-level parameters in terms of high-level ones.

Cantor Digitalis has thus been reimplemented in Faust. The functional (as opposed to procedural) nature of the language makes it possible to follow the structure of [48], with many definitions in the code corresponding directly to equations in the paper.

Whatever format the algorithm is compiled into, it does not take care of the mapping between parameters and physical control interfaces, or preset management; those part have to be dealt with elsewhere. The benefit of this approach is modularity; more precisely, separation between the signal processing algorithm itself and parameter management. This guarantees that the algorithm will remain usable even in the event of deprecation of a particular platform or interface.

Sections 7.5.3 and 7.5.4 describe how the core algorithm was integrated into usable systems.

7.5.3 Mono

Mono is a software suite developed by the Puce Muse association. It provides a common architecture for designing software instruments, with an elaborate preset system that allows control parameters to not only be stored and recalled but also interpolated (when possible). "Mono"s are used in artistic performances as well as in workshops in elementary, secondary, and music schools. Depending on the level, those workshops aim to teach students cooperation, familiarize them with sound synthesis, or discover existing musical material in a new light.

Cantor Digitalis has been integrated to Mono, yielding *MonoCantor*. As Monos are developed in Max, and Faust allows direct creation of a Max external, integration of the audio engine was relatively easy. After integrating the engine, the audio parameters it exposed had to be connected to Mono's complex parameter and preset system. Finally, a visualizer was added. It integrates a guide for the performer to see what pitch they are playing, as well as a visualization of the synthesized waveform.

7.5.4 Module

The SMAC project, which this doctorate is a part of, centers around a platform integrating a range of sound synthesis algorithms. Said platform aims to cater to the needs of both experienced musicians and educators. It will function as a modular synthesis module and embark a screen for selection of the audio algorithm. Cantor Digitalis was also integrated to this platform.

The platform is based on a Raspberry Pi computer running Linux. This makes it impossible to run Max programs; on the other hand, it features an instance of Pure Data, a free, open source of Max. The Faust Cantor code was thus compiled into a Pure Data external and ported on the platform.

7.6 Recommendations for future work

Among the constraints listed in section 7.2, some are tangible right from the start of the project: initial development speed, basic compatibility issues. Some others only become evident later on in the process, if ever at all. With that in mind, and given the amount of constraints to take into account, a common strategy is to develop a prototype, that is, a first version of the system that disregards some of the more long-term constraints, with the intention to later develop another version that will take them into account.

Starting with a prototype is often a perfectly reasonable strategy, but associated with it is one risk: that of blurring the boundary between prototype and final vision. It is easy, for instance, to keep working on something that has become too complex to be a prototype, with resources initially intended for a prototype. On the contrary, one may rush the transition from prototype to final product, or skip the prototype phase altogether when one would have been necessary. Those risks are particularly high when working on one's own, and/or with lack of long-term planning.

Thus when developing a system such as a vocal synthesizer, the distinction between phases — prototype, final product, and possibly more intermediary prototypes —, as well as the technical choices and allocated resources, should be made as clear as possible from the beginning of the project, and frequently reevaluated.

We believe that Voks lacked a clear separation between a prototyping phase and a final product phase. As a consequence, systematic use of the Max language, which in our opinion should be reserved to prototyping, for reasons stated in this chapter, was a major hindrance during the development of Voks. In the future, for such projects, we would recommend not dismissing technologies that need time to be mastered, and instead taking that time investiment into account when planning and/or starting with a clearly-defined prototyping phase.

Chapter 8

Musical uses of Voks

Contents

8.1 Introduction

Voks has been first and foremost conceived as a music instrument, designed to be used on stage. Although the COVID-19 pandemic, which started around the middle of this doctorate, severely limited the possibility of assembling a vocal synthesis ensemble, Voks has been used in a number of musical performances. Nevertheless, it has been demonstrated in performances, both recorded and live.

8.2 NIME

8.2.1 Presentation

The first public performance featuring the latter version of Voks is associated with the NIME article [131], which exposes the general principles of the instrument when controlled with a theremin.

Figure 8.1: *Still image from a video of the NIME'19 performance. From left to right: the T-Voks soloist Xiao Xiao; and the C-Voks players: Boris Doval, Christophe d'Alessandro, Grégoire Locqueville.*

The NIME conference is special in that its participants can apply to give not only a talk, but also a musical performance, which was the case here. The article [26], reproduced in the appendix of this document, features programme notes for the performance, after a short description of the instrument itself. Figure 8.1 is a picture from the performance. Each part of the performance has been filmed, and is available as a separate video, publicly available online; hyperlinks to those are given in the corresponding sections.

This was a good opportunity to exhibit multiple facets of the instrument. To put the spotlight on T-Voks, the object of the related article, while showcasing the potential of the Voks family in general, the ensemble that played was structured around a soloist playing T-Voks, Xiao Xiao, accompanied by three C-Voks players, namely Christophe d'Alessandro, Boris Doval and Grégoire Locqueville. Still, the soloist only stands out on some occasions; in the first piece *All Shall Be Well* (see below), notably, the ensemble is treated more as a quartet than as an accompanied soloist.

8.2.2 Demonstration video

A video was realized as part of the submission for the conference[1]. The video was the first public demonstration of Voks, and its aim was to showcase the potential of the instrument, as well as to give a general idea of its mode of operation. It thus features a number of short clips each showing a different facet of the instrument. In each of the clips, T-Voks was played by Xiao Xiao.

[1] `https://www.youtube.com/watch?v=jJdVsv_-WIo`

La Vie en Rose

The first clip is the viewers' first contact with the instrument. To put the focus on the instrument rather than on artistic choices, it shows a famous song, played in fairly traditional way — namely La Vie en Rose, by Édith Piaf and Louiguy, accompanied by Boris Doval on the piano. The choice of song, which is largely associated with France around the world, is a nod to the French origin of the instrument.

My Funny Valentine

The second clip gives both an overview of Voks' mode of operation, and a demonstration of its timbre-changing capabilities: Xiao Xiao is shown recording her voice speaking the words of the jazz standard My Funny Valentine, by Richard Rodgers and Lorenz Hart. Then screenshots show the labeling process and Voks' graphical interface, to give a succinct idea of the process to go from an audio recording to a live performance. Finally, the song is shown played on T-Voks, again with a live piano accompaniment by Boris Doval, but this time with a simulated vocal tract elongation and an increase in the relative amount of unvoicing. This makes it evident to viewers that Xiao Xiao's original, female voice can be changed to a male-sounding one which is comparable to the voice of Chet Baker, of whom My Funny Valentine is the signature song.

Pierrot Lunaire

The next clip demonstrates Voks' ability to adopt a vocal style that departs somewhat from traditional sung voice: the Sprechstimme technique, pioneered by Engelbert Humperdinck but most notably associated with composers of the Second Viennese School. [118] The specifics of the technique depend on the composer and even on the work being considered, but it can generally be described as drawing both from speaking and singing without being either. Schönberg's *Pierrot Lunaire* uses the technique all throughout the piece.

The video clip is an extract from *Der kranke Mond*, a movement where the *Sprechstimme* voice, played by T-Voks, is accompanied by Benoît Fabre on the flute. The source sample for the vocal synthesis is a recording by Grégoire Locqueville's male voice, with a simulated shrinking of the vocal tract so as to turn it into a female-sounding voice, further demonstrating Voks' timbre-changing capabilities.

Chun Xiao

The last video clip shows an example of spoken voice, albeit one with distinctive features: the language is Mandarin Chinese, a tonal language, and the text is a traditional poem, Chun Xiao, by Meng Haoran.

To demonstrate Voks' prosodic capabilities, the recording of the source sample by Xiao Xiao, a native Chinese speaker, is again shown, with the specificity that the tonal dimension of the text is obliterated: the text is spoken with a monotone voice, and only during synthesis are tones added through the performer's gestures.

NIME concert

Although the video was realized before the NIME conference, it was later enriched with an extract from the concert that took place at the conference (see the next section 8.2.3, demonstrating its relevance not only as a proof of concept, but as an actual live instrument. The video was also later used as part of the submission for the Guthman instrument competition (see section 8.4).

8.2.3 Performance

The programme of the performance was as follows:

All Shall Be Well

A creation of a work composed by Christophe d'Alessandro composed for the occasion, on a text by Julian of Norwich[2].

All Shall Be Well is a motet, that is, a polyphonic vocal composition; the form dates back to the 13th century. It makes use of the eight voices available on stage, including the four synthetic voices afforded by Voks and the four natural voices of the players. This allows for interesting interactions between both types of voice.

The text itself is a short excerpt from *Revelations of Divine Love* [91] by 14^{th}-15^{th} century anchoress Julian of Norwich. It can be decomposed into four variations on a verse, followed by a longer verse which echoes those. Here is the excerpt, divided into verses to make its structure apparent (the original text written as prose):

> *I may make all thing well,*
> *I can make all thing well,*
> *I will make all thing well,*
> *and I shall make all thing well;*
> *and thou shalt see thyself that all manner of thing shall be well.*

The form of the motet is ABA′CA″B′A‴, where

- A, A′, A″ and A‴ are each composed of a repeated recitation of the first four verses, alternating both between the four musicians and between natural and synthetic voices. Repetitions feature several different intonations, and each A section goes faster than the previous one. The resulting dialogue allows listeners to compare, almost systematically, natural and synthetic voice, under a variety of conditions, but based on the same phonetic content.
- B and B′ are polyphonic sections, here again featuring both natural and synthetic voices. Starting with individual voices at the beginning of B, the number of simultaneous voices increases to culminate with all eight voices at the end of B′. Since the A sections featured spoken voices heard horizontally (that is, one after the other), the B sections can be seen as their musical and vertical counterpart.
- The central section C is the only one featuring no natural voice, and to not seek to imitate natural voice. It features simultaneous pitch rises and descents, accelerations and slow-

[2]Video available at `https://www.youtube.com/watch?v=bfuob7yqKXY`

downs. All musicians repeatedly play the fifth verse, with one of them starting from a slow, low-pitched voice and gradually moving to a fast, high pitched one, another starts from a slow, high-pitched voice and moves to the opposite, and so on, with all possible combinations combined. The result is an intense vocal texture, a chaotic-sounding choir featuring unfamiliar voices.

La Vie en Rose

A video of the performance is available at `https://www.youtube.com/watch?v=I_hYHj5o82s`.

Constrained by the ensemble, the arrangement is an *a capella* — with the obvious peculiarity being that all the voices are synthetic. Contrary to the previous piece, no natural voice is used here.

Here the T-Voks soloist is clearly highlighted; as a counterpart, they have no opportunity to use novel playing modes, being limited to imitating a human singing the main melody. The three accompanists, on the other hand, while always providing a traditional, tonal harmonic foundation, go through various playing modes:

Babbling Playing in scrub mode (see section 2.3.2) using the same textual material as the soloist, the accompanists move their stylus back and forth on the surface of the tablet. This produces a shower of syllables on the verge of intelligibility. The words being the same as those of the main melody, the accompaniment echoes it in a comical way.

Sustained vowel The time index is kept constant at a judiciously chosen value, so that only the syllable /u/ can be heard. From the perspective of the player, C-Voks turns into a limited version of Cantor Digitalis [48], that is, an instrument playing a vowel, whose pitch and vocal effort are controlled with a graphic tablet, but whose phonetic output, a single vowel, cannot be altered. This results in realistic-sounding *a capella* harmonies.

Beatbox In what we shall here call the "beatbox section", each accompanist plays a different role. One C-Voks player plays using the same standard, syllabic playing mode as the T-Voks player. The T-Voks player thus ceases to be a soloist, and a T-Voks/C-Voks duo is played instead. Listeners can then appreciate the similarities and dissimilarities between both instances of Voks.

The two other C-Voks players provide a beatboxed rhythmic section, in the following way. The scrub mode is selected; the textual material is composed of vocal imitations of percussion or bass notes — /pf/ to imitate a kick, /ts/ for a high hat, /k/ for a snare, and /pum/ for the bass. Players scrub their stylus on the zone of the tablet corresponding to the percussive or bass sound they wish to play. One player plays the percussion sounds (which all happen to correspond to unvoiced syllables). The other player plays the bass sound (corresponding to the voiced syllable /pum/), repeatedly, arpeggiating chords. Thus precise rhythms are played with vocal, non-semantically meaningful sounds. Note that in principle, samples of actual percussive sounds and bass notes could have been used, given that those are also well approximated as the sum of harmonic sound and filtered white noise (see section 1.4.3). To stay within the general spirit of the performance, vocal sounds are

however preferred.

The beatbox featured in that section remains very slow compared to even a beginner beatbox performance using one's mouth. This is to be expected for two reasons. First, this playing mode is dissimilar to all others, requiring musicians to play accurate rhythms using the continuous scrub mode. Thus the experience gained by playing other modes is hardly of any benefit for this one. This parallels the difficulty experienced by newcomers to beatbox, who have to use their vocal apparatus in ways unlike those that they were used to. Second, Voks players are disadvantaged compared to beatbox players in that they only have one "virtual articulator" at their disposal, in the form of the stylus. In contrast, beatbox players take advantage of the diversity of articulators in their mouth, usually refraining from consecutively producing sounds that share a place of articulation. This spares them the delay needed to make an articulator ready between a sound and the next one.

Chun Xiao/Improvization

A video of the performance is available at `https://www.youtube.com/watch?v=okX5BpcTWFk`.

This piece showcases Voks' ability to speak in the specific case of tonal languages. The form is a hybrid one, starting like a skit and progressively transitioning into an improvized section.

The skit's theme is language education, an allusion to the educational applications of Voks envisioned (see chapter 4). Again, all voices are synthetic. The T-Voks soloist, plays the role of a Mandarin teacher trying to get her students to say the poem correctly. The postures and even the interfaces of both the teacher and the students reflect their role: the teacher's theremin has her perform large, demonstrative gestures, whereas the student are bent over their tablet, scribbling on it with their stylus as they would on a notebook with a pen.

Verse by verse, the teacher gives a demonstration, moving her hands in front of her theremin and synthesizing the verse with the adequate tones. The C-Voks players then attempt to reproduce her prosody, but do not manage to do as well. A transcription in pinyin of the poem, a romanization system that indicates tones with diacritics, is given in figure 8.2 alongside the original text and a translation in English.

As an aside, in this performance, the C-Voks players played at the best of their ability, even though their role required them to play incorrectly. The fact that Xiao Xiao, the musician playing the role of the teacher, is a native Mandarin speaker, while none of the students are, was simply taken advantage of.

After the students' unsuccessful attempt to reproduce their teacher's prosody, the teacher manifests her dissatisfaction, at which point the exercise starts again. This time, however, instead of imitating the poem phonetically, the students use the playing modes at their disposal (see section 3.3) to conjure up the images conveyed by the poem.

春眠不觉晓,	Chūn mián bù jué xiǎo,
处处闻啼鸟。	Chù chù wén tí niǎo.
夜来风雨声,	Yè lái fēng yǔ shēng,
花落知多少。	Huā luò zhī duō shǎo.

Sleeping in the spring, one hardly knows it's daylight,
Birds are heard everywhere trilling.
There've been sounds of wind and rain in the night,
How many blossoms have been falling.

Figure 8.2: *Original text of the poem (top left), pinyin transcription (top right), and English translation as quoted in [63] (bottom)*

8.3 The OCEN ensemble

OCEN was founded at Sorbonne Université in 2019 as a space to both explore the artistic possibilities made possible by research on interfaces for musical expression, and use musical practice to find new perspectives and problematics for research.

It was not conceived as an ensemble with fixed structure, and aims to include students (mostly from Sorbonne Université). Sadly, the COVID-19 pandemic has brought the ensemble's activites to a halt; no students have actually taken part in the project, and the ensemble has only performed one concert, which we talk about in the following section.

The concert took place on the Jussieu campus of Sorbonne Université in Paris. It involved eight musicians, each playing their own electroacoustic instrument: Serge de Laubier, Pierre Couprie, Hugues Genevois, Michel Risse, Christophe d'Alessandro, Laurence Bouckaert, Vincent Goudard and Grégoire Locqueville. The only vocal instrument apart from Voks was Cantor Digitalis, played by Christophe d'Alessandro.

The concert was a structured improvisation driven by John [59], a program that generates and displays guides for improvising. The "improvisation score" from John consisted in a timeline with sections during which subsets of performers were to improvize based on a textual indication. Given that no other instrument was able to articulate text, we treated this concert as an opportunity to explore non-articulated playing modes some more.

8.4 Guthman musical instrument competition

8.4.1 The competition

T-Voks was submitted to, and was awarded second prize at the 2022 Guthman musical instrument competition, an event aiming to bring together music technologists from all around the world, for them to showcase the most innovative music instruments. The event is annual and takes place at the Georgia Institute of Technology in Atlanta. Despite being based in the United States, it

attracts participants from other countries, with six out of the nine 2022 finalists from outside the US, although it remains very Western-centered (only one non-North American, non-European finalist — from Venezuela). It also attracted the attention of musician and music technology Youtuber Benn Jordan, who realized a video involving some of the finalists.

As one would expect, the competition was an good way to make our work known to a relatively large and remote audience, but it was also a new opportunity to test Voks in the context of a performance with a new perspective, with the notable inclusion of a short theatrical segment. Finally, the friendly atmosphere made it easy to get acquainted with and discover the work of fellow music technologists.

Participants in the 2022 edition of the competition went through several selection stages:

- Selection to participate in an online semifinal; the selection was based on a video demonstration and some text that were submitted by the participants. Each of the 26 semifinalists has their video demonstration featured on a webpage[3] where web users could vote for instruments until the end of the competition.
- Selection to participate in the final. Nine finalists were invited to Georgia Tech to present their instrument and participate in a concert.
- Final votes by a jury composed of three experts in music technology and design. The judges' votes selected a first, second and third prize, as well as a "Special Award" for the instrument deemed most mature. In addition, the instrument with the most votes from web users received the "People's Choice" award.

The instruments that were distinguished illustrate the diversity of the instruments submitted: alongside T-Voks, those instruments that were distinguished with a prize included a fully acoustic instrument, an installation based on unstability and chance, and an electro-mechanical sound modulation device.

The competition consisted in two main events: a presentation of about 20 minutes, where each contestant was free to talk about and demonstrate their instrument, and a 5-minute-long musical performance. Each contestant was paired with a local musician, with whom they rehearsed for one day, for the musical performance. It should be noted that the judges' deliberation took place before the musical performance; only the presentation had an influence.

8.4.2 Presentation in front of the jury

The presentation consisted in a combination of a traditional talk with slides, short musical demonstrations, and a humorous skit involving a personnification of T-Voks. As one would expect, the voice of the T-Voks character was synthesized using T-Voks, controlled by Xiao Xiao — the borrowed voice was also that of Xiao Xiao. The combination of the presenter's natural voices, T-Voks' singing voice and T-Voks' talking voice was a way to demonstrate the capabilities of the instrument not only for singing, but also for talking.

The skit consisted in T-Voks obnoxiously interrupting the presentation and taking control of

[3] https://guthman.gatech.edu/semifinalist-gallery-2022

Xiao Xiao's gestures, in an effort to make spectators and presenters alike reflect upon the philosophical implications of controlling someone else's or one's own voice. The skit also featured a few brief demonstrations of some capabilities of T-Voks, including drastic timbre changes, imitation of animal sounds, and ability to speak in a tonal language, with the recitation of two lines from the Chun Xiao poem.

8.4.3 Musical performance

The performance[4] was treated as a fairly traditional vocal concert: Xiao Xiao played two songs, accompanied by local pianist Zachary Shah. The songs were:

- *La Vie en Rose* by Édith Piaf and Louiguy, the same song as in the previous NIME concert;
- *Non lo dirò col labbro*, an aria from the Baroque, Italian-language opera *Tolomeo*, by George Frideric Handel. The title of the aria means "I won't say it with my lips"; the choice of aria was an allusion to the principle of performative vocal synthesis, with which performers can pronounce words without using their lips.

The songs had been practiced thoroughly in the preceding weeks, both by Xiao Xiao and by the pianist: the pianist was paired with us by the organizers of the event, and we had the opportunity to get in touch with him remotely a few weeks before the concert. This allowed us to agree on a programme so that both musicians could practice independently. The day of the performance was also dedicated to rehearsing.

The traditional, somewhat rigid nature of a non-improvized concert featuring a (synthetic) voice accompanied by a piano was a way to contrast with the exuberance of part of the presentation of the previous day. It was also a very practical choice related to Voks' workflow: to sing any new textual content, one needs to go through a cumbersome data-preparing process; even once textual content is prepared, the fact that the performer needs to go through the text linearly does not encourage improvization or deviations from the initial plan.

8.4.4 Interview with Benn Jordan

Benn Jordan is a musician that has been making music involving a great deal of electronic and digital technology for more than two decades. On his YouTube channel, he shares his experience and discusses new developments in music technology.

Upon hearing about the Guthman competition, Benn Jordan contacted the finalists of the 2022 with the aim of realizing a video featuring them and their instruments[5]. The video consists in a series of demonstrations of each featured instrument, followed by an interview with the participant defending it.

[4]Video available at `https://www.youtube.com/watch?v=7BW7mErQXMY&t=519s`
[5]Video available at `https://www.youtube.com/watch?v=zag9Cu1Z6A8`

8.4.5 Comments on the competition

Preparation of the competition

Preparations for the Guthman competition differed significantly from the NIME concert. The NIME concert was the first public live demonstration of T-Voks, featured an ensemble composed of both T-Voks and C-Voks, and was longer than the Guthman concert, three characteristics which we saw as incentives for exploring a broad range of techniques and playing modes. Playing modes that do not aim to imitate natural voice, in particular, were highlighted.

The program of the Guthman competition was deliberately less exuberant, with T-Voks almost exclusively imitating natural voice; in fact, only during the skit did Xiao Xiao aim for anything else than imitating an existing singing style at the best of her ability. As such, a special focus was paid to details that contribute to vocal style:

- Synchronization between rhythm, vocal effort and pitch. The interaction between rhythm control and vocal effort, in particular, led to a number of considerations that can be found in section 3.1.2.
- Characteristics of source samples. Audio samples were sent back and forth between a female professional opera singer and us, with the aim of obtaining the best possible source sample. In spite of those efforts, the samples recorded by the professional singer turned out to feature too high a vocal effort, due to our incorrect initial belief that a greater vocal effort would always be desirable (see section 3.2.1). We then turned to Boris Doval, a male singer less experienced in opera, but closer to us (being involved as a researcher in the design of Voks). Through trial and error, Boris Doval was able to provide source samples the resynthesized version of which was deemed satisfying for a live performance.

Lessons regarding general theremin playing

Incidentally, from the admission of Xiao Xiao, training for a performance adhering to a set of specific vocal styles has resulted in some realizations related to non-augmented theremin technique.

As a general remark, she felt that playing with vocal sounds instead of the traditional theremin sound made her listen better to details of her theremin technique. An explanation for this phenomenon remains to be found, though it is not surprising: replacing the theremin sound that Xiao Xiao has gotten accustomed to by a new one, she is able to effectively listen to her output with new ears, so to speak. As a first example, the addition of linguistic articulation led her to develop a wrist gesture for note-level articulation management through intensity (in addition to that which she controls with a button/pressure sensor); but she later noticed, watching experienced thereminists play, that they used the same gesture. Thus T-Voks triggered her discovering and understanding of that gesture, as well as her noticing it in other thereminists' playing.

Another lesson of the addition of linguistic articulation control is more concerned with beginners. As was noted in section 3.1.2, the presence of linguistic articulation in T-Voks forces thereminists to change the way they think about articulation. This could be beneficial to beginner thereminists, who often have trouble focusing on phrase-level intensity shaping, dedicating most of their attention to note-level shaping.

Conclusion

This dissertation has developed the design, development and use of the family of performative vocal synthesizers, Voks. Voks makes it possible to articulate a text with both vowels and consonants, enabling synthesis of arbitrary text. The synthesizer's modular design implies that it is not restricted to a certain set of control gestures; rather, it has been controlled with a number of control interfaces, and new ones can be added relatively easily.

To perform the synthesis, two vocoders have been used: WORLD is a free and open-source software library which allows access to its internal representation, making it possible to apply a wide variety of expressive transformations. SuperVP, in contrast, is proprietary and closed-source, and can only be used through the interface provided by its developers. This limits the possibility for audio manipulations, although pitch, rhythm, and some timbre manipulations remain accessible. SuperVP has the benefit of usability, with its ease of use and ability to process sound instantly.

The possibility of synthesizing a text with consonants introduces two related scientific questions. The first question is that of control of rhythm. In line with previous research on Vokinesis — Voks' first iteration —, Voks is primarily based on a control method whereby rhythm is dealt with on a syllabic basis, with a cyclic hand movement composed of alternating presses or taps, and releases. This approach is inspired by the frame-content theory of speech production. [78] The choice of an algorithm for converting such discrete data into continuous time is not a trivial one. To make talking about algorithms for rhythm control clearer, we have formalized the notion of *time index*, which relates the temporality of an audio recording and that of a live performance. Using that formalism, a modified version of Vokinesis' control algorithm, the so-called *parabolic method*, has been devised; it avoids, to an extent, the synthesized audio lagging behind on control gestures.

The second question is related to the richness of linguistic content in an articulated voice, and the resultant difficulty of controlling phonetic content. In view of the diversity and speed of possible vocal sounds, the strategy adopted in Voks is to specify the text in advance with an audio recording; this approach, however, subjects performers to a rather drastic constraint. We propose avenues for improving that state of affairs; the first such improvement is a simplified method for specifying text by typing rather than by recording audio. The second improvement, developed with intern Paul Chable under the supervision of Christophe d'Alessandro, is much more ambitious, and also less mature. It consists in specifying the textual content of an utterance

as it is being played using a gestural grammar that associates gestures to syllables.

Voks' potential for new control methods has been harnessed with the testing of a number of new interfaces. For pitch control, in particular, several have been tested, with a special focus on the theremin, whose relevance has been verified informally through its use in performance contexts.

Initially conceived as a music instrument, Voks has been considered for other uses, including educational ones, with the hope that using a modality different from the vocal apparatus to produce speech might help foreign language learners understand and assimilate prosodic patterns better. A preliminary test has been conducted, focusing on intonation: it tests how well subjects can reproduce intonation patterns using Voks.

Perspectives

This work has raised a number of scientific and engineering questions which remain to be settled.

Pitch interface study

Work with a variety of interfaces has shown that the graphic tablet, the interface historically used for controlling predecessors to Voks, is not the only viable choice. The theremin, in particular, has proven an interesting and successful alternative to the tablet for controlling pitch. No formal study of the theremin as a pitch control interface, however, has been conducted, nor has any comparison of the theremin with other interfaces been attempted. Such a study could be partially modeled on [24] and [21], which evaluate the quality of control of intonation using a graphic tablet.

Rhythm control improvement and evaluation

Voks' rhythm control algorithm suffers from a significant delay between the performer's gestures and the synthesized sound. Preliminary tests have highlighted the complexity of syllabic rhythm control, and as such, improvements to that algorithm could come from several areas: discrete alternatives to the buttons and keys currently used could be found, or the algorithm which turns that discrete input into continuous time could be redesigned. Continuous interface options could also be explored, which would mandate redesign of the algorithm as well. Improvements on the labeling which the algorithm relies on is another possibility.

Dissemination

Although it has been shown by way of example that Voks can be used in practice, the process for using it is still cumbersome. The synthesizer still needs to be improved so as to rely on a combination of technologies that could be transmitted and used with relative ease, by users with a reasonably low degree of technical proficiency. Efforts in this direction are under way.

Several members of the theremin community have notably expressed interest in augmenting

their instrument with vocal synthesis; most of them are likely not to be audio programmers, and catering to them would undoubtedly be a fruitful endeavour.

Appendix A

Publications

The following documents, published during this doctorate, are reproduced in the next few pages:

- **T-Voks: The singing and speaking theremin** [131]
- **Borrowed Voices** [26]
- **Voks: Digital instruments for chironomic control of voice samples** [76]
- **Prosodic disambiguation using chironomic stylization of intonation for native and non-native speakers** [129]
- **Évaluation de la stylisation chironomique pour l'apprentissage de l'intonation du français L2** [130]

T-Voks: the Singing and Speaking Theremin

Xiao Xiao
Grégoire Locqueville
Christophe d'Alessandro
Boris Doval
LAM - Institut Jean le Rond d'Alembert
xiaosquared@gmail.com
{gregoire.locqueville,christophe.dalessandro,boris.doval}
@sorbonne-universite.fr

ABSTRACT

T-Voks is an augmented theremin that controls Voks, a performative singing synthesizer. Originally developed for control with a graphic tablet interface, Voks allows for real-time pitch and time scaling, vocal effort modification and syllable sequencing for pre-recorded voice utterances.

For T-Voks the theremin's frequency antenna modifies the output pitch of the target utterance while the amplitude antenna controls not only volume as usual but also voice quality and vocal effort. Syllabic sequencing is handled by an additional pressure sensor attached to the player's volume-control hand.

This paper presents the system architecture of T-Voks, the preparation procedure for a song, playing gestures, and practice techniques, along with musical and poetic examples across four different languages and styles.

Author Keywords

theremin, vocal synthesis, gesture

CCS Concepts

•**Applied computing** → **Sound and music computing;** Performing arts; •**Human-centered computing** → *Gestural input;*

1. INTRODUCTION

Invented in 1920, the theremin is one of the world's first electronic musical instruments [19]. It is particularly visually attractive as there is no contact between the player and the instrument, only the dance of both hands around the two antennas. Proximity to one antenna controls the frequency while the other controls the amplitude of the output sound, which is traditionally generated by an analogue oscillating circuit using the heterodyne principle.

Despite its relatively simple waveforms, the theremin's expressivity has often been compared to the human voice due to the sensitivity of its control interface. Like the voice the theremin allows for subtle melodic variations (vibrato, portamento, etc.) and refined volume control. The present research aims to further the analogy between the theremin and the singing voice, taking advantage of recent work in performative singing synthesis.

NIME'19, June 3-6, 2019, Federal University of Rio Grande do Sul, Porto Alegre, Brazil.

One strategy for high-quality performative synthesis is based on vocoding and modifying pre-recorded voice samples, as demonstrated by Vokinesis [13, 12]. In this system, intonation and vocal effort are controlled using a graphic tablet. Articulation timing and rhythm are defined by syllabic control points, which can be triggered by the press and release of a control button or through the continuous variation of a fader pedal.

The present work is built upon Voks, a new performative synthesizer based on the WORLD vocoder [28, 26] that improves upon the capabilities of Vokinesis. While Voks can be used with the same graphic tablet interface as Vokinesis, we have mapped the control of intonation and vocal effort to the pitch and volume antennas of the theremin. The addition of a pressure sensor attached to the player's volume modulating hand triggers the advancement of syllabic control points. This combination of Voks, the theremin and pressure sensor is called T-Voks[1].

This paper first summarizes prior work in vocal synthesis and theremin augmentation. Next, it presents the technical side of Voks, describing underlying principles of voice analysis, synthesis and control, as well as the software and hardware implementation. Considerations from the performer's side are then discussed, including playing gestures, practice techniques, and a set of musical and poetic examples across languages and styles.

2. RELATED WORK

2.1 Speech and Singing Synthesis

While speech synthesis manages to achieve convincing results, the synthesis of expressive voices remains hard. *Performative vocal synthesis* approaches this problem by using human gestures in real time as an input to add expressivity to an artificial voice. Several such speech or singing synthesis systems have been proposed in the past. These systems can be differentiated by their synthesis algorithms, the level of control for the performer, and the control interfaces used.

Some performative voice synthesis systems generate sound from the ground up, usually using a formant synthesis algorithm [16, 17]. This approach allows for free speech, but the numerous parameters involved can be difficult to control. For that reason, other systems make use of a pre-recorded voice [22, 13]. Intermediary approaches have also been devised, such as Glove-Talk I [15], where specific gestures are mapped to specific short words, whose characteristics are used to determine the speaking rate and the stress of each word. Another intermediary approach is diphone-based concatenative synthesis, which offers both the flexibility of pure synthesis and the realism of re-synthesis. In this method, speech is synthesized by the concatenation of short

[1]T-Voks demo video: `http://youtu.be/jJdVsv_-WIo`

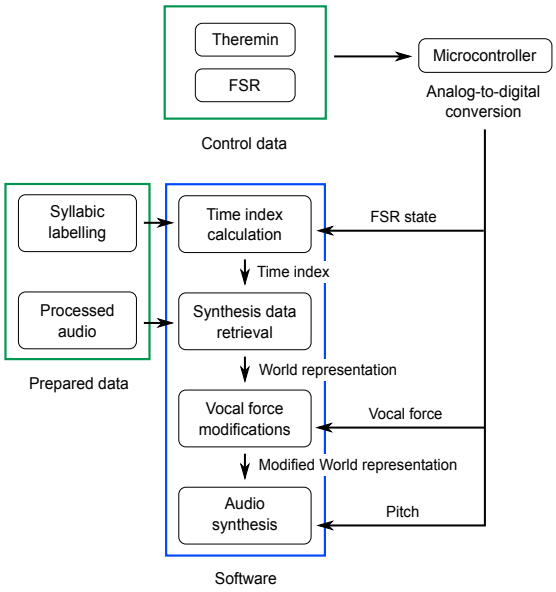


Figure 1: *T-Voks principles and architecture. Control gestures by the player are in shown the top green frame (FSR stands for force sensitive resistor). The left green frame represents the linguistic data prepared for performance (recorded voice and syllabic control points). The blue frame represents real-time processing of gestural and linguistic data for performative singing synthesis.*

sound pieces, from a dictionary of all possible phoneme to phoneme transitions in a language (e.g. about 1200 units in French). Such algorithms are used in offline synthesizers such as Vocaloid [21].

Speech control involves numerous parameters. For real-time use with input from human gestures, systems must make a tradeoff between degree of control and ease of use. On one end of the spectrum, Glove-Talk II [16] offers direct control over several vocal articulation mechanisms. However, even a well trained performer (accomplished pianist, over 100 hours of training) "finds it difficult to speak quickly, pronounce polysyllabic words and speak spontaneously" [16]. The resulting sound quality is similar to early formant-based text-to-speech systems. On the other end, Calliphony [22] lets a performer reproduce a pre-recorded signal with a speed and fundamental frequency chosen in real-time. The sound quality and intelligibility are excellent, but the linguistic material is fixed for a given performance, and timing control is unnatural.

Vokinesis [13] offers more direct control over rhythm while still being based on existing audio. The timing and rhythm of consonants are controlled through sequencing syllabic sized chunks using various methods like tapping or continuous expression pedal motions. Vokinesis enable the control of continuous voice parameters (e.g. pitch and vocal effort) and can be used with continuous control surfaces, such as the Seaboard, the LinnStrument, the Soundplane, the Hakken Continuum, or any other interface that outputs Multidimentional Polyphonic Expression (MPE) data. The interface of choice for Vokinesis is the graphic tablet, which offer a two-dimensional continuous surface, as well as stylus-pressure detection. It enables the reuse of familiar gestures from handwriting, which most people have learned at a young age [8, 11, 17].

In summary, it does not seem possible to control all the aspects of voice production through only hand (or feet) gestures. Re-sequencing a pre-recorded voice is a good compromise between modification capabilities and sound quality. Diphone based concatenative singing synthesis sounds very natural, but it can hardly be applied to real-time performative synthesis because it does not seem possible to select any arbitrary sequence of diphones on the fly (the task would be to select one diphone among about 1200 in less than about 150-200 ms). The syllable seems a better candidate as the basic unit for real time speech or singing control, where syllables more or less correspond to the musical notes in a score. This is the solution chosen for the augmented singing theremin.

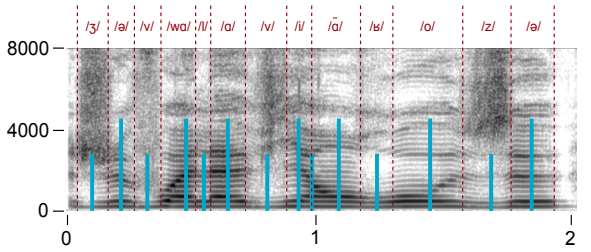


Figure 2: Syllabic control points
Spectrogram of a voice recording of the French sentence "Je vois la vie en rose" used as input for T-Voks (x-axis labelled in seconds, y-axis in Hertz). The phoneme boundaries are marked in red, and the location of control points (see section 3.2.3), in turquoise. When opening her hand, the performer triggers a gradual increase of the time index with a control point associated with a tall mark; when closing it, the target is set to a control point associated with a short one.

2.2 Augmented Theremin & Gesture Singing

To increase the sonic possibilities of their instrument, contemporary thereminists such as Dorit Chrysler and Carolina Eyck employ guitar pedals such as delay and reverb, as well as loopers to create additional layers with their voices [3, 1]. A vocal formant filter pedal can be used to make the theremin sound even more like the human voice, as exemplified by Rob Schwimmer [5]. The theremin has also been used as a controller for modular synthesizers, most notably by thereminist Coralie Ehinger [2]. Other players, such as Lyn Goeringer and Rachel Gibson, have used the theremin's control voltage (CV) output to manipulate textures generated on the computer [4, 18].

Recent research has begun to explore the intersection between singing and freehand gestural control. The Theremin Orchestra combines live vocal performance with capacitive sensors used in traditional theremins, some of which apply effects to the singers' voices while others translate the singers' hand gestures into traditional theremin sounds [9]. The Gestural Envelopes project employs wearable inertial motion sensors to track a performer's hand gestures, which control the syllable timing of pre-recorded singing, as well as some effects [23].

3. DESIGN AND IMPLEMENTATION

3.1 Control Principles

T-Voks is based on syllabic control and modification of pre-recorded voice utterances. There are three main parameters:

1. **Intonation** is altered by the vertical antenna of the theremin. Pitch scaling requires modification of the signal fundamental frequency (f_0) using a vocoder.

2. **Vocal effort** is adjusted by the loop antenna, or volume control of the theremin. Realistic vocal effort modification is a complex process: it involves joint sound intensity, spectral slope, voice/unvoiced ratio modification. This requires a specially designed vocoder and modification rules [29].

3. **Syllabic Sequencing** is managed by an additional interface, managed by the volume-control hand. This requires syllabic labels associated to the signal and time-scaling of the synthesized signal using a vocoder.

Syllabic control is the main difference between playing T-Voks and the usual theremin. It is rooted in the frame-content theory of speech production [24]. This theory postulates that speech utterances can be decomposed into syllabic "frames," associated with the opening/closing motions of the jaws, and "content," associated with smaller units like consonants. The speech rhythm is given mainly by the frames, although the content represents the details in articulation (i.e. the phoneme articulation).

For performative synthesis, controlling the frame timing is necessary and sufficient. Each frame is defined by two time control points: one for the open phase, and one for the closed phase. Figure 2 displays the placement of control points on a speech utterance. Note that a control point alone for each syllable is not sufficient: a biphasic control is needed [12].

3.2 Software

Written in C++ and Max, Voks is the performative singing synthesizer software behind T-Voks. Like its predecessors Calliphony [22, 10] and Vokinesis [13, 12], it is based on the time and frequency scaling of recorded voice utterances, with the possibility of some voice quality modifications. Rhythm is controlled by resequencing using the same syllabic control points as Vokinesis.

Voks is implemented using the WORLD vocoder, which features better sound quality and much improved robustness. It is designed with a modular architecture, with a control shell and an external procedure for syllable labeling. This modularity enables Voks to be incorporated into new programs. T-Voks is one such example. Voks can also be used with the same interfaces as used with Vokinesis, such as a graphic tablet or a fader pedal.

The following sections describe the functionality of the WORLD vocoder, how a new song is prepared, and how a song is synthesized during performance.

3.2.1 The WORLD Vocoder

Designed for speech synthesis, the WORLD vocoder [28, 26] is free software that allows real-time signal-level modification of a voice recording.

WORLD consists of two independent units, an analysis module and a synthesis module. Those units respectively allow for conversion from a monophonic audio file into a specific spectral representation (analysis), and conversion from that representation back into playable audio (synthesis). While analysis is only possible on an existing audio file on disk, the synthesis module can be used in real-time; the input representation to the synthesizer is updated just as sound is output from it.

The WORLD representation is based on a modified version of the source-filter model of speech production [14] . In WORLD's version of that model, a voice signal is modeled as the sum of a filtered pulse train (the periodic part) and filtered white noise (the aperiodic part). To describe the input sound, the representation needs to include at each time

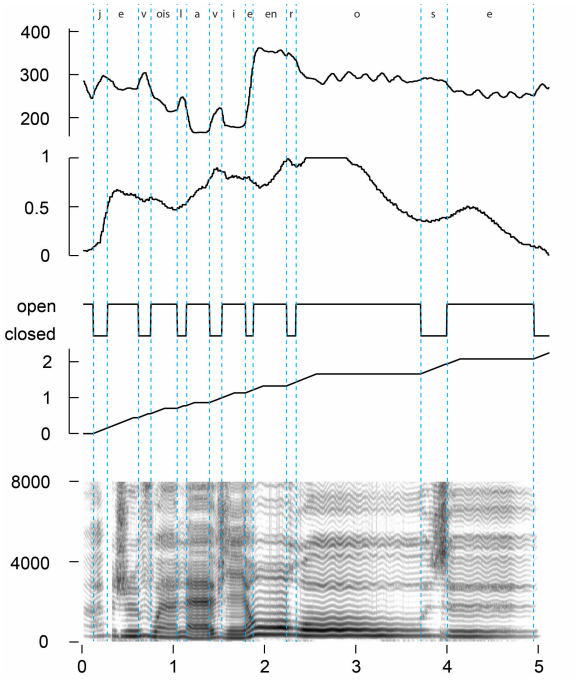


Figure 3: T-Voks performance
Time evolution (in second) of control parameters during a T-Voks performance of the sentence "Je vois la vie en rose." From top to bottom: syllable labels, fundamental frequency (in Hertz), vocal effort, FSR state and the resulting analysis time (in second), and a spectrogram of the output sound (ordinate labelled in Hertz). The opening/closing instants, marked in turquoise, do not correspond to control points, but trigger an increase in the time index until it reaches the next control point.

both the frequency of the pulse train and enough information to be able to recreate the two filters. In practice, those data are specified every 5ms, and the filter information is stored as two floating-point number arrays. One array contains a power spectrum for the whole signal (both the periodic and aperiodic parts). The other array specifies at each frequency bin the ratio of the power from the aperiodic part of the signal with the total power of the signal.

The f_0 is estimated at each frame by the Harvest algorithm [27]. Harvest acknowledges that the input audio might be entirely aperiodic at some frames, and does not specify a f_0 for those frames. The f_0 computed by Harvest is not directly used by T-Voks but is used by two subsequent algorithms, CheapTrick [25] and D4C [26]. CheapTrick computes an estimate of the power spectrum of the signal at each frame. D4C then estimates, at each frame and each frequency bin, how much of that power spectrum comes from the periodic part of the signal and how much comes from the aperiodic part.

To generate audio, the synthesis module takes a power spectrum and an aperiodicity ratio arrays as input, as well as a numeric value for the f_0. If fed with that input at a sufficiently high frequency, it can generate a realistic voice in real-time. The synthesis module has been ported into a Max object, allowing for its use in the T-Voks Max patch.

3.2.2 Voice Signal Modifications

In the WORLD representation, the input voice signal is split into three sets of parameters at the analysis stage: the f_0, the periodic spectrum and the aperiodic spectrum. At

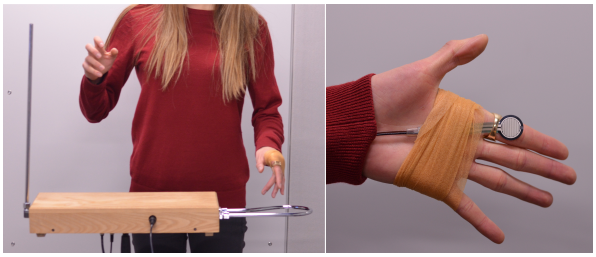


Figure 4: *Augmented theremin: a Theremin, a FSR, and an analog to digital interface. Analog control signals by the performer can then be processed using MAX/MSP. The whole interface being played is shown on the left. The right hand controls pitch, by subtle variations of distance between the hand to the upright antenna and hand shape. The Left hand controls voice quality, by variation of distance between the hand and the flat antenna. The left hand also controls biphasic syllabic sequencing, by pressure of the thumb on a FSR. The location of the FSR in the performer's left hand is shown on the right.*

the synthesis stage, these three sets of parameters are input into the synthesis engine. Modifications take place between analysis and synthesis.

Pitch scale modification is straightforward: the synthesis f0 contour replaces the analysis f0 contour. Time scale modification is obtained by duplication or suppression of time frames for the three sets of parameters, without any parameter modification.

Voice quality modification is more intricate. It is obtained by spectral tilt modification (on the periodic and aperiodic spectra), periodic-aperiodic ratio modification, and signal amplitude modification. Note that the spectral representation in WORLD allows for other types of modification, that are not used in the present demonstration (e.g. formant modifications, vocal tract apparent length modification, etc.).

3.2.3 *Preparing a Piece*

Playing a piece on T-Voks requires three files to be prepared in advance: an audio sample, an analysis file, and a labeling data file.

The audio sample is the data that will be re-sequenced. It must be strictly monophonic, with a clearly defined pitch. Best results are obtained by ensuring that the target text is clearly spoken or sung in monotone, with a pitch close to the range that will be used during performance. Modifications of the recorded signal prior to the performance are possible, as long as their result complies with the conditions mentioned above. Resampling with a factor of about 1.2, for instance, allows for a "gender swap" effect while leaving the audio valid for use in T-Voks. Prior to use in T-Voks, the chosen audio file must be analyzed using the WORLD analysis module and stored for T-Voks to read during performance.

The labeling data file is a text file specifying the locations of all control points mentioned in section 3.1 As of now, the labelling process is done manually though a semi-automated procedure based on automatic detection of the phonemes in audio could be envisioned.

3.2.4 *Synthesis during Performance*

At the beginning of a performance, the analysis file, as well as the file containing a syllabic labelling of the audio, are loaded. Along with the control data received from the augmented theremin, their contents will drive the synthesis, which comprises three consecutive operations. Figure 3 shows the evolution of the control parameters when playing the verse "*Je vois la vie en rose*," as well as a spectrogram of the resulting audio.

First, a time index is received. That index, which we call the *analysis time*, specifies the temporal position in the original audio on which the synthesis should be based. An analysis time is associated with each instant in the time of the performance, or *synthesis time*. Once the current value of the analysis time is known, the WORLD representation at the corresponding time frame is queried in the loaded analysis file. That representation, while already a valid input to the WORLD synthesis module, then undergoes a few modifications to account for the changing values of vocal effort that are being received.

The vocal effort is varied by modifying the intermediate WORLD representation right before it is used in the synthesis. Among those modifications, only two affect the periodic part of the voice. One is a variation of the ratio between the first and higher harmonics, to simulate a change in the frequency of the glottal formant. The other is the use of a spectral slope. One last modification involves both the periodic and the aperiodic part, since it consists of varying the relative prominence of those parts: increasing the prominence of aperiodicity results in a breathier voice, thus reducing the perceived vocal effort.

3.3 Hardware and Computer Interface

The hardware of T-Voks is based on a Moog Etherwave Plus theremin, which features control voltage (CV) outputs for pitch and volume. The pitch CV ranges from -2.5v to 4.5v, with a change of 1 volt per octave of the theremin's pitch. The volume CV from our theremin measured from 0 to 5.5v. Each CV output is fed into an analog input pin of an Arduino Uno. For optimal function, the CV ranges should be regulated to the 0 to 5 volt range of the Arduino. A force sensitive resistor (FSR) is used to control the sequencing of syllables. Its data uses another analog input of the Arduino. Our FSR is attached to a signal conditioning circuit from Interface-Z[2], whose output vary from 0 to 5 volts between maximum pressure and no pressure.

The Arduino is connected to a computer running the T-Voks Max patch, and is programmed to convert the pitch, volume, and FSR analogue data to digital values ranging from 0 to 1024. The Max patch receives data from Arduino using the Arduino2Max object, communicating over Serial with a baud rate of 115200. In the T-Voks patch, prepared pieces are loaded by setting the proper audio, analysis, and syllable labeling files, with shortcut buttons to load files for several example pieces. Pressing the button also serves to restart a piece.

Pitch data from the theremin modulates the fundamental frequency of a loaded piece at its current time index while volume data modulates the vocal effort. Although the output of the FSR belongs to a continuous range of values, it is only used as a binary switch, allowing the player to choose between two states, pressed and released. Each time the FSR changes from one state to another, a time index starts increasing until it reaches the next syllabic control point. Pressing the FSR triggers a transition from a syllabic nucleus of the original recording to a syllabic liaison, whereas releasing it triggers the inverse transition, loosely mimicking the movements of a jaw during speech, as described in the frame/content model of speech production.[24]

[2]https://www.interface-z.fr/pronfiture/contact/148-pression-force-fsr.html

4. MAKING MUSIC WITH T-VOKS

4.1 Theremin Playing Techniques

Theremin performance is a highly individual art, with no single, standardized technique, though some educational resources do exist [6]. Despite widely held views on the theremin's difficulty, many players (including our thereminist, the first author, who is self-taught) manage to accurately play melodies within several months of practice.

For most right-handed players, the right hand controls pitch and the left hand controls volume. While only the nearest point between the body and each antenna directly modifies the output tone, the rest of the player's body also influences the response of the antennas' capacitive fields. To reliably find notes and intervals, some thereminists have developed hand and finger gestures, such as opening and closing the hand toward the antenna for the span of an octave. Shaking the pitch-control hand produces a vibrato.

Unlike most traditional instruments, which must be actuated to produce a sound, the theremin outputs a tone by default and must be explicitly silenced. Raising and lowering the volume-control hand is used to cleanly delineate notes by removing unwanted glissandos in between. The quality of these movements sculpts the attack, duration and decay of each note, enabling a wide range of articulations, though legato across large intervals and sharp staccatos are difficult to achieve. Larger movements of hand, wrist and arm defines the dynamics across phrases.

4.1.1 Controlling Syllables

Syllable sequencing is controlled by the hand responsible for volume modulations (for our thereminist, the left hand). The movement to press the sensor must be comfortable enough to perform repeatedly, reliable enough for the system to detect, and fast enough to articulate several syllables in rapid succession. Moreover, it should not interfere with other gestures of the same hand, wrist, and arm for articulation and dynamics.

After experimenting with several different sensor placements, consistently satisfactory results were found with the FSR positioned against the first knuckle of the index finger, held in place by a ring and pressed by the thumb. Its cable runs down the palm, along the arm, around the shoulder and out behind the player. It is secured by an elastic band around the palm and easily hidden by a long-sleeved shirt. Only the thumb and forefinger of the volume hand are involved in syllabic control, leaving free range of motion for the rest of the hands and fingers, as well as the wrist, forearm, and elbow.

The addition of syllable control alters the volume hand's techniques. A syllable change at the same time as a note change hides the usual glissando to the new note, and removes the need for the volume hand to dip between the notes. Syllables also add more attack and textural variation, liberating the volume hand to focus on phrase-level dynamics rather than note-level articulation. While the pitch hand is not involved in syllabic control, the addition of articulated syllables has inspired new pitch-control gestures, as described in the next section.

4.2 Musical and Poetic examples

To showcase the versatility of T-Voks, four demonstration pieces were created, practiced and recorded. Each features a different language and musical or poetic style. Here we describe how each example was created as well as playing techniques specific to each style. These examples are demonstrated in the accompanying video (see footnote 1).

La Vie en Rose. To recreate the famous chorus of the 1940s French song by Edith Piaf, we recorded a vocal sample from a female speaker. This speaker has no musical background, showing the capability of T-Voks to work with untrained voices.

As French is a syllable-timed language [7], articulation of each new syllable is done with a quick tap to the FSR. The tap release begins the syllable rhyme, which is held by Voks until the next syllable begins. During sustained vowels, playing technique is no different from the unmodified theremin, which can replicate key features of Piaf's signature vocal style, including dramatic vibratos and small glissandos at the start and end of phrases.

My Funny Valentine For our version of the jazz standard made popular by Chet Baker in the 1950s, a vocal sample from a female American English speaker was resampled by a factor of 1.22 to yield a male voice. The modulation of vocal effort along with volume change lends a "breathiness" to the synthesized voice, inspired by Chet Baker's singing.

Unlike French, English is a stress-timed language [7]. Syllable control requires paying more attention to stress timing and inter-syllable transitions. One example is when to repress the FSR after a vowel to trigger a consonant word ending. For phrases ending in a consonant, volume fades must be carefully controlled to not reach silence in order to hear the final articulation (e.g. "...favorite work of *art*").

Pierrot Lunaire - Der Kranke Mond *Sprechstimme* is a vocal technique where singing imitates the continuous pitch contours of speech. A classic example is Arnold Schoenberg's *Pierrot Lunaire* suite, where a narrator, typically a soprano, recites poetry in German accompanied by a small instrumental ensemble. Schoenberg indicated that notated rhythms should be closely followed while notated pitches, once found, should be quickly altered by rising or falling sweeps.

An excerpt from one movement of *Pierrot Lunaire* was recorded by a male speaker, which was transformed into a female voice by a resampling factor of 0.8. As a stress-timed language, German shares the same considerations for syllable advancement as English. For a convincing *sprechstimme*, pitch slides and their volume curve must also correspond to the correct stress pattern. Pitch slides are achieved by small displacements of the fingers or by pivoting around the wrist.

Chun Xiao Mandarin Chinese is a tonal language, where the same syllable pronounced with different frequency contours changes in meaning. Each syllable can be pronounced with one of four tones, which is carried by the syllabic rhyme [20]. Classical poetry is typically recited with exaggerated tone enunciations.

A well-known Tang dynasty short poem was recorded by a native speaker pronouncing each syllable in monotone. The poem was then "recited" using T-Voks, with each tone shaped entirely by the theremin. Each syllable was triggered by a quick tap and release of the FSR.

Tones were created mostly with the pitch-hand, with the volume hand creating a gradual fade in and fade out. The pitch hand rests in place for tone 1, whose pitch stays steady. Other tones, whose pitch change in different ways, were produced using fluid wave-like gestures of the entire hand, pivoting at the wrist. These hand sweeps are larger than those required by *Pierrot Lunaire*, with the forearm remaining largely stationary.

4.3 Practice Methods

The addition of syllable advancement introduces a significant cognitive load to an instrument that already requires full concentration for playing. In early stages of using T-Voks, the thereminist invented exercises for syllable advancement to get used to the new task. These exercises

involved triggering new syllables at different rhythms while performing simple pitch changes with the other hand (e.g. scales and arpeggios).

For each musical example, the thereminist would isolate the difficulties of syllabic control and pitch control, practicing only the correct rhythm of syllable advancement, or only the notes and their phrasing on any sustained vowel.

From the early stages of learning a melody, the thereminist played along with example recordings to help stay in tune. Playing along with recordings and attempting to imitate the singer as closely as possible also helps to inform interpretations. At later stages of learning a song, the thereminist would alternate between playing along with a singer and playing with only an instrumental accompaniment track in order to find her own expression. To practice the Chinese poem, the thereminist (a Chinese speaker) alternated between vocal pronunciation and T-Voks replication in order to find gestures that replicate the tonal contours of her actual voice.

5. CONCLUSIONS

The intrinsic vocal quality of the theremin is particularly well suited to singing synthesis control. This work presents, to the best of our knowledge, the first singing theremin, i.e. the first encounter between performative singing synthesis and the (augmented) theremin. Expressive, accurate, and precise singing is obtained, when the instrument is played by a well-trained theremin performer.

It is interesting to compare singing synthesis using the augmented Theremin (T-Voks) and singing synthesis using a graphic tablet and a stylus (C-Voks). The synthesis engine and control principles are similar, but control interfaces are different, resulting in different types of expressive gestures and different musical styles.

Natural sounding singing is obtained by re-sequencing of recorded utterances, and therefore to the expense of freedom in terms of linguistic content. The performer is able to play any score (i.e. any rhythm and tones, but only with the fixed text loaded in the synthesis engine). Future work will address the question of free text singing.

6. ACKNOWLEDGMENTS

Part of this work has been done in the framework of the SMAC (FEDER IF0011085) project. Xiao Xiao was partially supported by the MIT-France program during her stay at Sorbonne Université.

7. REFERENCES

[1] Carolina Eyck. `https://www.carolinaeyck.com/`.

[2] Coralie Ehinger. `https://coralieehinger.ch`.

[3] Dorit Chrysler. `http://www.doritchrysler.com/`.

[4] Lyn Goeringer. `http://www.lyngoeringer.com/portfolio/?page_id=57`.

[5] Rob Schwimmer. `http://www.robschwimmer.com/`.

[6] Theremin world: Learn to play the theremin.

[7] D. Abercrombie. *Elements of General Phonetics*. Edinburgh University Press, 1984.

[8] M. Astrinaki. *Performative statistical parametric speech synthesis applied to interactive designs*. PhD thesis, University of Mons, 2014.

[9] M. Blasco. The theremin orchestra. `http://half-half.es/the-theremin-orchestra`.

[10] C. d'Alessandro, A. Rilliard, and S. Le Beux. Chironomic stylization of intonation. *JASA*, 129(3):1594–1604, 2011.

[11] N. d'Alessandro and T. Dutoit. Handsketch bi-manual controller: Investigation on expressive control issues of an augmented tablet. In *Proc. NIME07*, pages 78–81. NIME, 2007.

[12] S. Delalez and C. d'Alessandro. Adjusting the frame: Biphasic performative control of speech rhythm. In *Proc. INTERSPEECH 2017*, pages 864–868, 2017.

[13] S. Delalez and C. d'Alessandro. Vokinesis : syllabic control points for performative singing synthesis. In *Proc. NIME 2017*, pages 198–203, 2017.

[14] G. Fant. *Acoustic theory of speech production*. Mouton, 1970.

[15] S. S. Fels and G. E. Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *Neural Networks, IEEE Trans. on*, 4(1):2–8, 1993.

[16] S. S. Fels and G. E. Hinton. Glove-talkii-a neural-network interface which maps gestures to parallel formant speech synthesizer controls. *IEEE Trans.on Neural Networks*, 9(1):205–212, Jan 1998.

[17] L. Feugère, C. d'Alessandro, B. Doval, and O. Perrotin. Cantor digitalis: chironomic parametric synthesis of singing. *J. Audi. Speech Mus. Proc.*, 2017.

[18] R. Gibson. The theremin textural expander. In *Proc. NIME18*, pages 51–52. ACM, 2018.

[19] A. Glinsky. *Theremin: Ether Music and Espionage*. University of Illinois Press, 2005.

[20] P. Hallé. Evidence for tone-specific activity of the sternohyoid muscle in modern standard chinese. *Language and Speech*, 73:103–124–1043, 1994.

[21] H. Kenmochi and H. Ohshita. Vocaloid-commercial singing synthesizer based on sample concatenation. In *INTERSPEECH07*, pages 4009–4010.

[22] S. Le Beux, C. d'Alessandro, A. Rilliard, and B. Doval. Calliphony: A system for real-time gestural modification of intonation and rhythm. In *Speech Prosody*, 2010.

[23] A. Lough, M. Micchelli, and M. Kimura. Gestural envelopes: Aesthetic considerations for mapping physical gestures using wireless motion sensors. In *Proc. ICMC07*, pages 60–64. ICMC, 2018.

[24] P. F. MacNeilage. The frame/content theory of evolution of speech production. *Behavioral and Brain Sciences*, 21(4):499–511, 1998.

[25] M. Morise. Cheaptrick, a spectral envelope estimator for high-quality speech synthesis. *Speech Communication*, 67:1 – 7, 2015.

[26] M. Morise. D4c, a band-aperiodicity estimator for high-quality speech synthesis. *Speech Communication*, 84:57 – 65, 2016.

[27] M. Morise. Harvest: A high-performance fundamental frequency estimator from speech signals. In *Proc. Interspeech 2017*, pages 2321–2325, 2017.

[28] M. Morise, F. Yokomori, and K. Ozawa. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Trans. on Information and Systems*, E99.D(7):1877–1884, 2016.

[29] O. Perrotin and C. d'Alessandro. Vocal effort modification for singing synthesis. In *Proc. INTERSPEECH 2016*, pages 1235–1239, 2016.

Borrowed Voices

CHRISTOPHE D'ALESSANDRO, LAM, Institut Jean le Rond d'Alembert, Paris
XIAO XIAO MIT Media Lab, LAM, Institut Jean le Rond d'Alembert, Paris
GRÉGOIRE LOCQUEVILLE, LAM, Institut Jean le Rond d'Alembert, Paris
BORIS DOVAL, LAM, Institut Jean le Rond d'Alembert, Paris

1. PROJECT DESCRIPTION

Borrowed voices is a performance featuring performative voice synthesis, with two types of instruments: C-Voks and T-Voks. The voices are played *a cappella* in a double choir of natural and synthetic voices.

Performative singing synthesis is a new paradigm in the already long history of artificial voices. The singing voice is played like an instrument, allowing singing with the *borrowed voice* of another. The relationship of embodiment between the singer's gestures and the vocal sound produced is broken. A voice is singing, with realism, expressivity and musicality, but it is not the musician's own voice, and a vocal apparatus does not control it.

The project focuses on ***control gestures***: the music explores vocal sounds produced by the vocal apparatus (the basic sound material), and "*played*" by the natural voice, by free-hand Theremin-controlled gestures, and by writing gestures on a graphic tablet. The same (types of) sounds but different gestures give different musical "*instruments*" and expressive possibilities.

Another interesting aspect is the distance between synthetic voices and the player, the voice being at the same time embodied (by the player gestures playing the instrument with her/his body) and externalized (because the instrument is not her/his own voice): ***two different voices sung/played by the same person***.

Fig. 1. T-Voks (Theremin + button, right) & C-Voks (stylus/tablet + button/surface, left)

2. TECHNICAL NOTES

Borrowed voices is part of a long-term research project on performative voice synthesis initiated at the *Speech Conductor workshop [1]*. Previous systems *Ramcess [2], Cantor Digitalis [3],* and *Vokinesis [4][5]*, explored various synthesis engines (formant synthesis, concatenative synthesis) and control

interfaces (graphic tablet, MIDI and MPE keyboards, pedals, Puce-Muse MétaTouche, Touché, Expressive-E Touché, Theremin …).
This research has been accompanied by several musical performances and concerts featuring the ***Chorus Digitalis***, an experimental musical choir of synthetic voices. Note that performative voice synthesis systems have been developed in other research groups: the *Voicer [6]* or *Handsketch [7],* or for musical projects like *Luna Park* [8].
T-Voks (presented in an accompanying paper [9] is a Theremin-controlled voice synthesizer. This will be the first public appearance of this new instrument, with its impressive visual and sound presence. For T-Voks the Theremin's frequency antenna modifies the output pitch of the target utterance while the amplitude antenna controls not only volume as usual but also voice quality and vocal effort. An additional pressure sensor attached to the player's volume-control hand handles syllabic sequencing. Metrical control is needed for accurate syllabic timing control. A pair of control points defines each syllable, considered as the basic rhythmic frame. The *"arsis"* corresponds to the constriction (weak beat of the syllable) and the *"thesis"* corresponds to the vocalic nucleus (strong beat of the syllable). The left hand uses a force sensitive resistor (FSR) button for biphasic sequencing of rhythmic units (see Figure 1, left).
C-Voks is a voice synthesizer controlled by a pen on a graphic tablet, using drawing gestures and a new version of the Vokinesis system [6][7]. Different modes of timing control are available: speech rate, scrubbing and metrical control. The speech rate mode corresponds to direct control of the signal playback speed. Scrubbing corresponds to direct control of the playback time position. Metrical control is the same mode as used in T-Voks. The non-preferred hand or feet using a button or pedals perform biphasic sequencing of rhythmic units. Vocal effort or voice quality is controlled by pressure on the tablet and settings.
The sound engines for C-Voks and T-Voks are similar: pitch, time scales, vocal effort, voice quality are modified according to these gestural controls with the help of a real-time high-quality vocoder (WORLD).
This gives an interesting confrontation of free hand gestures of the Theremin and calligraphic gestures for singing or speaking.

3. PROGRAM NOTES

The piece played at NIME'19, entitled ***"Borrowed Voices"*** is especially composed to explore the various possibilities of vocal instruments played in double a choir of natural and artificial voices. It is a collective creation of the ***Chorus Digitalis,*** featuring Xiao Xiao (T-Voks, voice), Grégoire Locqueville (C-Voks, voice), Christophe d'Alessandro (C-Voks, voice), Boris Doval (C-Voks, voice). The players' natural voice join occasionally the synthetic vocal quartet in a double choir formation (8 voices). New musical possibilities opened by performative singing synthesis are explored in prepared improvisation and compositions. The sound and gestural material offered by this musical formation allow for :

- ***Voice deconstruction*** sounding like "computer music" or "electroacoustic voice." Parametric representation and modeling of the voice allows for extreme variations. Specific features of the voice can be emphasized (formants, pitch, voice quality, vocal tract size, roughness), and a rich sonic material based on the voice can be worked out in real time.
- ***Voice imitation*** on the contrary privileges the proximity between natural and synthetic voice. How close to a natural voice can a synthetic voice be? In some situation, a realistic voice is desirable. It is at the (possibly interesting) risk of an "uncanny valley" effect.
- ***Voice extension*** in between deconstruction and imitation, the augmented voice is a realistic-sounding voice with augmented (naturally impossible) features: for instance, a voice with a very large register, a male/female voice, a very slow, very rapid pronunciation, small and large vocal tracts. Another aspect of voice augmentation is the specific vocal gestures allowed by the control interfaces: here the Theremin and graphic tablet.

Borrowed Voices is composed of four clearly contrasted movements rolled into a single unbroken piece. Like previous works of the ***Chorus Digitalis*** the music is polystylistic: it makes use of multiple styles and techniques, and various languages, i.e. pieces of *Borrowed Styles*. The four movements of ***Borrowed Voices*** are braid together as follows. The introduction, or **Movement 1,** is an 8-voice motet written by Christophe d'Alessandro. This movements are based on English textual material, a 15th century poem by Julian of Norwich. The motet is designed as a contrapuntal vocal game, based on the strong rhythmic and verse structure of the text, and on the synthetic/natural dialectics both at the choir level (4 + 4) and at the individual level (double voice). After a short transition comes **Movement 2.** It is a solo French song: La vie en rose (Edith Piaf, Louis Guglielmi, arranged by Boris Doval), sung by T-Voks, with a 3-voices accompaniment. The solo voice is sung in an impersonation paradigm, with the typical vocal expressiveness and musicality of the French "chanson réaliste". The vocal arrangement (3 C-Voks) by Boris Doval for the accompaniment explores freely the sounds and gestures of the synthetic choir. **Movement 3** is a Chinese poem recitation, performed by Xiao Xiao on T-Voks. It shows the musicality of Chinese speech recited using Theremin gestures. The hand motions are drawing in space a gestural equivalent of Chinese tones and vocal expression. **Movement 4** concludes the piece in a prepared improvisation, featuring T-Voks, 3 C-Voks, and natural voices of the 4 musicians. The improvisation is prepared in the sense that the texts (and moods) of Movements 1, 2 and 3 are reused (English, French and Chinese) by the 4 singers in this conclusive part.

4. MEDIA LINK(S)

- Video: https://youtu.be/XxIz6MnT9HM
- Video: https://www.dropbox.com/s/4xyvyv0e5jyy2b3/TVoks%20NIME19%20submission.mov?dl=0

ACKNOWLEDGMENTS

Part of this work has been done in the framework of the SMAC (FEDER IF0011085) project. Xiao Xiao was partially supported by the MIT-France program during her stay at Sorbonne Université.

REFERENCES

[1] C. d'Alessandro, N. D'Alessandro, S. Le Beux, J. Simko, F. Çetin et Hannes Pirker, (2005), « The Speech Conductor: Gestural Control of Speech Synthesis » proc. eNTERFACE 2005, Mons, Belgium, pp. 52-61

[2] N. D'Alessandro, B. Doval, C. d'Alessandro, S. Le Beux, P. Woodruff, Y. Fabre, T. Dutoit « RAMCESS: Realtime and Accurate Musical Control of Expression in Singing Synthesis », Journal on Multimodal User Interfaces, Vol. 1, No. 1, March 2007, p 31-39.

[3] L. Feugère, C. d'Alessandro, B. Doval, and O. Perrotin. Cantor digitalis: chironomic parametric synthesis of singing. EURASIP Journal on Audio, Speech, and Music, 2017.

[4] S. Delalez and C. d'Alessandro. Adjusting the frame: Biphasic performative control of speech rhythm. In Proc. INTERSPEECH 2017, Stockholm, Sweden, pages 864-868, 2017.

[5] S. Delalez and C. d'Alessandro. Vokinesis : syllabic control points for performative singing synthesis. In Proc. NIME 2017 Aalborg University Copenhagen, Denmark, pages 198{203, 2017.

[6] L. Kessous. Gestural control of singing voice, a musical instrument. Proceedings of Sound and Music Computing, 2004.B.A. R. Jensenius and M. J. Lyons, *A NIME Reader*. Cham: Springer International Publishing, 2017.

[7] N. D'Alessandro and T. Dutoit. Handsketch bi-manual controller: investigation on expressive control issues of an augmented tablet. In Proceedings of the 7th international conference on New interfaces for musical expression, pages 78{81. ACM, 2007.

[8] G. Beller, « Gestural Control of Real-Time Speech Synthesis in Luna Park », Proc. Sound and Music Computing, Padova, 2011

[9] X. Xiao, G. Locqueville, C. d'Alessandro, B. Doval. T-Voks: Singing and Speaking with the Theremin. In Proc. NIME 2019 (this issue)

Speech Communication 125 (2020) 97–113

Contents lists available at ScienceDirect

Speech Communication

journal homepage: www.elsevier.com/locate/specom

Voks: Digital instruments for chironomic control of voice samples

Grégoire Locqueville [a], Christophe d'Alessandro [*,a], Samuel Delalez [b], Boris Doval [a], Xiao Xiao [a]

[a] Institut Jean le Rond d'Alembert, Sorbonne Université, CNRS, UMR 7190, 4 place Jussieu, Paris, 75015, France
[b] Lunii,18 rue Dubrunfaut, Paris 75012, France

ARTICLE INFO

Keywords:
Voice synthesis
Singing synthesis
New interfaces for musical expression
Performative synthesis
Real-time vocoder

ABSTRACT

This paper presents Voks, a new family of digital instruments that allow for real-time control and modification of pre-recorded voice signal samples. An instrument based on Voks is made of Voks itself, the synthesis software and a given set of chironomic (hand-driven) interfaces. Rhythm can be accurately controlled thanks to a new methodology, based on syllabic control points. Timing can also be controlled with other methods, including scrubbing and playback speed variation. Pitch, vocal effort, voice tension, apparent vocal tract size, voicing ratio, aperiodicity ratio of the voice samples can be modified thanks to a real-time high-quality vocoder. Different forms of chironomic control of the vocal parameters are proposed. Pitch is controlled by continuous hand motions using a stylus on a surface (C-Voks) or a theremin (T-Voks). Other interfaces can be used as well. Syllabic rhythm is controlled using a biphasic button. Scrubbing, playback speed and timbre related parameters can be controlled using the theremin, control surfaces or continuous controllers like faders. In addition to realistic imitation of speaking or singing voices, other playing modes yield new interesting sounds. Voks participated in comparative perceptual evaluation of singing synthesis systems. It has been demonstrated in a live musical settings, using different control interfaces. In addition to musical or poetic performances, applications of performative vocal synthesis to language learning and speech reeducation are foreseen.

1. Introduction

1.1. Performative vocal synthesis

Performative vocal synthesis, or voice instruments, are the meeting point of voice synthesis and new interfaces for musical expression[1]. While the synthesis of acoustic musical instruments such as the pipe organ, the piano, strings and winds have reached high levels of realism, the speaking and singing voices remain a challenge for digital sound synthesis and new musical instruments research. Performative vocal synthesis is an important issue for voice research, with applications in the fields of music (new instruments, studio), language education, speech therapy (control of voice source substitution).

A fundamental difference between vocal synthesis and musical instrument synthesis is the additional linguistic content. The speaker or singer must in real time perform a musical task (pitch, rhythm and timing, voice force and quality) and a linguistic task (a stream of phonemes, syllables, words, sentences). Performative vocal synthesis requires two main types of processes: 1. selection and planning of the linguistic and musical material to be sung and 2. use of an external control device for sound synthesis to mimic the motions of the internal voice apparatus.

Voks is a new paradigm in the already long history of artificial voices. The voice is played like an instrument, allowing for singing or speaking with the borrowed voice of another (d'Alessandro et al., 2019) with realism, expressivity and musicality. In the Voks system, linguistic material is prepared in advance, by speech recording and labelling. Any utterance is composed of voice signal samples (ranging from a single syllable to entire sentences) enriched with syllabic marks that allow for accurate rhythmic control. A Voks set, for a given performance, is a set of linguistic utterances, that can be selected and played on the fly. The synthesis is achieved by real-time control of a high-quality vocoder. The vocoder allows for real-time fundamental frequency (F0) scaling, time scaling, vocal effort and voice quality modifications. These modifications are driven by the player's gestures, using various chironomic (hand controlled) interfaces. The first performative speech synthesis system,

* Corresponding author.
E-mail address: christophe.dalessandro@sorbonne-universite.fr (C. d'Alessandro).

[1] Part of this work has been presented at NIME 2017 (Delalez and d'Alessandro, 2017), Interspeech 2017 (Delalez and d'Alessandro, 2017) and NIME 2019 (d'Alessandro et al., 2019; Xiao et al., 2019).

https://doi.org/10.1016/j.specom.2020.10.002
Received 14 February 2020; Received in revised form 10 June 2020; Accepted 9 October 2020
Available online 2 November 2020

allowing for synthesis of any text, was Glove Talk (Fels and Hinton, 1993, 1998). It converts hand gestures to speech, based on a neural network gesture-to-formant model. The gesture vocabulary is based on a correspondence between hand shapes and articulators positions. Synthesis is based on a formant synthesizer. However, even a well trained performer (accomplished pianist, over 100 h of training) "finds it difficult to speak quickly, pronounce polysyllabic words and speak spontaneously" (Fels and Hinton, 1998). Though it enables the real-time performative synthesis of speech, neither version of Glove Talk can be considered a singing synthesis system, as melodic and rhymes control was highly limited. The formant model has been used in several singing synthesis systems. This approach has the advantage of granting the user total control over the generated sound. A relatively small number of parameters drives a synthesis algorithm (Berndtsson, 1996; Cook, 1993; d'Alessandro et al., 2006, 2007; d'Alessandro and Dutoit, 2007; Feugère et al., 2017).

Another approach is diphone-based concatenative synthesis, which offers both the flexibility of pure synthesis and the realism of resynthesis. In this method, speech is synthesized by the concatenation of short sound pieces, from a dictionary of all possible phoneme-to-phoneme transitions in a language (e.g. about 1200 units in French). Such algorithms are used in offline singing synthesizers (Bonada et al., 2016; Feugère et al., 2016; Kenmochi and Ohshita, 2007; Umbert et al., 2015). A performative text-to-speech synthesis environment based on Hidden Markov Models synthesis has been presented (Astrinaki, 2014; Astrinaki et al., 2012). In this case, the linguistic material is entered in text form on a computer keyboard or from a text file. More recently, several neural network based synthesis systems appeared, with application to singing synthesis (Blaauw and Bonada, 2017).

Several control interfaces for melodic control have been proposed. The discrete nature of the traditional piano-like, MIDI keyboard, makes it ill-suited to the control of voice (d'Alessandro et al., 2005). The graphic tablet has been used in singing synthesis systems (d'Alessandro et al., 2006, 2007; d'Alessandro and Dutoit, 2007; Feugère et al., 2017). It has the advantage of reusing the expert gestures of writing, allowing for similar or even higher precision for singing pitch control (d'Alessandro et al., 2011). Other interfaces sending continuous data streams can be used, particularly those using the MIDI (or Multidimentional) Polyphonic Expression protocol (The MIDI Manufacturers Association, 2018), including the Continuum (Haken et al., 1992) and the Seaboard (Lamb and Robertson, 2011).

1.2. Chironomic control of voice samples

As it does not seem possible to control all the aspects of voice production through only hand (or feet) gestures, chironomic control of voice samples is a good compromise between modification capabilities and sound quality (Le Beux et al., 2010). The present work focuses on accurate melodic and rhythmic control. Melodic control is based on earlier work on chironomic control (d'Alessandro et al., 2011; Feugère et al., 2017). The paradigm of syllabic control points introduced in the Vokinesis system (Delalez and d'Alessandro, 2017a, 2017b) offers accurate control over voice rhythm and timing. Voice quality and vocal effort are processed using a high quality vocoder (Morise et al., 2016). Voks architecture is presented in Fig. 1.

This architecture contains three main blocks: data preparation (left, green frame), chironomic control (top, green frame) and real-time processing (center, blue frame). Voks is based on WORLD (Morise et al., 2016), a powerful vocoder, whose underlying signal model is described in Section 3.1. WORLD allows for analysis of a speech signal in a spectral-domain representation for further processing and synthesis (see Section 3.3). Prior to the performance, an audio sample, represented in the left green panel in Fig. 1, is recorded. This linguistic material is then labelled, resulting in a text file containing a representation of the syllable locations. Details about the labelling method and the resulting *label file* can be found in Section 2.3.

The top, green panel in Fig. 1 represents the user off-line and real-time controls. As a musical instrument, Voks takes real-time data as input. Some of those data are acquired via gestural interfaces; additionally, some other values can be set using a graphical interface. The acquisition of real-time data, and their interplay with the rest of the system, are discussed in Section 5.2.

The center, blue panel in Fig. 1 represents data processing and audio synthesis. At each instant of the performance, the real-time data and the data contained in the labelling file are converted, using *time rules*, to a *time index* that indicates the position in the original audio upon which to base the synthesis. This method for temporal control, detailed in Section 4.1, allows the extraction of a *WORLD frame*, the WORLD representation of the original audio at one specific instant. This frame then undergoes some or all of the modifications described in Sections 4.2–4.4. Those modifications are driven by parameters that are functions of the real-time inputs; the rules used to turn input data into parameters for modification are described in Section 4. The resulting representation is then fed to the synthesis module mentioned in Section 3.3.

Note that Voks is based on the same principles as the earlier

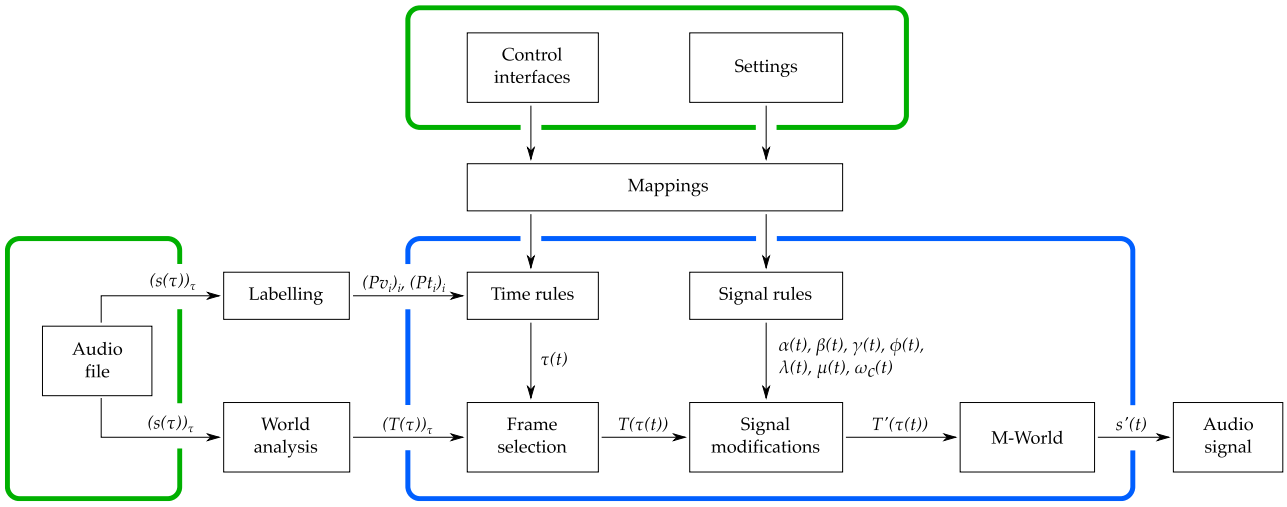


Fig. 1. General architecture of Voks. The green boxes enclose input data; the blue box encloses the real-time software processing that takes place at the time of the performance. The meaning of the time index τ is explained in Section 2; that of other Greek letters is explained in Section 4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Calliphony and Vokinesis systems (Delalez and d'Alessandro, 2017a, 2017b; Feugère et al., 2016). The architecture in Fig. 1 represents Calliphony and Vokinesis as well. The main difference between these systems is the vocoder used, RT-PSOLA (Le Beux et al., 2010b) for Calliphony and Vokinesis, vs WORLD for Voks. As WORLD offers an explicit spectral decomposition of the speech signal, it has been preferred to RT-PSOLA: more flexibility in signal processing and transformations is allowed. For pitch and rhythm control the quality and functions of Voks and Vokinesis are equivalent.

2. Sample preparation and labelling

Voks is based on rhythmic sequencing, pitch and voice quality modifications of pre-recorded speech samples. The principles for vocal rhythm control are presented, and a new method using syllabic control points is proposed in this section.

2.1. Speech material

To use Voks, the first step is to record a suitable speech sample. The suitable speech samples are monophonic recordings. Note that the vocoder may show some limitation for processing non-standard vocal techniques, such as growling, that would disrupt the harmonicity of the recording too much. Excessive reverberation may also jeopardize the results. Other monophonic, harmonic sounds (such as many auto-oscillating instruments, bird songs, etc.) may give satisfactory results. Clearly articulated syllables are necessary for accurate syllabic rhythm control, but are not required for non-syllabic playing modes (e.g. *scrub* and *speed*).

The recorded samples need to be *prepared* before being of any use to the performer, this involves two different steps:

- *syllabic labelling* for marking of the signal with rhythmic anchors or control points.
- *signal analysis* for transformation of the time-domain samples to a source-filter representation suitable for signal modification in the spectral domain and pitch scaling.

The Voks system is capable of using samples provided by a Text-to-Speech system, which can deliver both the signal samples and syllabic labels at the same time. The sound quality is often poorer than natural speech. Therefore recording natural voice is generally preferred for high quality musical performances.

2.2. Rhythmic model

Speech and singing rhythm is defined by the position in time of consonants and vowels. Following MacNeilage (1998), the syllable can be regarded as the minimal suprasegmental unit for rhythm control. According to (Wagner, 2008, p. 41–53), the syllable can consistently be associated to rhythmic beats in speech and music. However, while some languages (e.g. French, Chinese) have syllable-timed isochrony, others have different ones. Mora-timed isochrony (e.g. Japanese) is based on units smaller than the syllable, and stress-timed isochrony (e.g. English) is based on units larger than the syllable. In the present work, an utterance is organized into a syllable stream. Uttering a syllable creates a musical note, a rhythmic unit. Of course one syllable can carry several musical notes (this is called a melismatic syllable, or melism), or a same note can extend over several syllables. Accurate control of syllable positions and timing is required for controlling the vocal rhythm. A simple idea would be to use only one control point for each syllable, for instance the perceptual centers (P-centers) of the syllable (Wagner, 2008; Barbosa and Bailly, 1994). P-centers are generally defined by tapping experiments, i.e. by synchronisation of a manual gesture with the perceived position of the syllable. This would correspond to hand-clapping or foot tapping, which are common ways of marking or following the perceived rhythm of music. P-centers are a *perception* concept, but for rhythm *production* it seems that only one point per syllable is not sufficient.

Another approach draws from the frame-content theory of speech production (MacNeilage, 1998), where syllables are related to oscillations of the mandible. In this theory, speech production, i.e. planning and realization of the articulatory motion for a given stream of phonemes and prosody, is considered to be the superposition of two coordinated processes. The relatively fast succession of segmental articulatory events defines the phonemic *content*, that is carried by the syllabic *frame*. The frame corresponds to the stream of syllables, each syllable being a biphasic cycle of opening/closing of the jaw. For each syllable, two points are needed, one corresponding to the opening, and the other to the closure of the mandible. Note that this is somewhat analogous to keyboard playing. When playing the pipe-organ for instance, a first motion is to depress the key, and a second motion, as important as the first, is to release the key. The duration of a note is a result of both "control points".

Syllabic rhythm depends on the syllable structure. The syllable is composed of three parts: the attack, the vocalic nucleus, and the coda. The attack and the coda correspond to zero, one or more consonants, and the nucleus generally corresponds to a vowel. A syllable always contains a vocalic nucleus, but not necessarily an attack and a coda. In an actual voice utterance, syllables are chained; attacks and codas of successive syllables correspond to consonants, i.e. the opening and closure motions of the vocal apparatus, while vowels correspond to the open positions. These opening and closing cycles can be exploited for rhythmic control. The concepts of *arsis* and *thesis* (derived from Greek prosody) are then very useful for our purpose. *Thesis* represents the stable part of the segment, in our case the vowel, or nucleus, and *arsis* represents the transient part between nuclei. The coda of one syllable and the attack of the next one (when they exist) are grouped to form the arsis. If there is neither a coda nor an attack, the arsis still exists, and corresponds to a short transition between two vowels. Controlling syllabic rhythm implies controlling those time points.

We define the *Syllabic Control Points* (SCP) as temporal marking points for rhythm production. *Vocalic Points* (P_v) are the SCP that correspond to the vocalic nuclei or thesis, and *Transient Points*(P_t), those that correspond to the transient phases, or arsis. These points define a target temporal location for each phase: when a vocalic phase is triggered (see Section 4.1.1), the target timestamp is at the corresponding P_v until the next transient phase is triggered. Once this transient phase is triggered, the target timestamp evolves from the current P_v to the next P_t, and the synthesis signal stops at that P_t until the next vocalic phase is triggered, and so on. Controlling the timing of these points allows for accurate rhythmic control while preserving the correct articulation. Although the syllable is the main unit discussed herein, the SCP may in principle be used for mora-timed or stress-timed isochrony as well. In this case P_v and P_t would be associated to time spans smaller of larger than the syllable, corresponding to morae, feet of stress-groups.

2.3. Syllabic control points

The SCP allows for accurate rhythmic control over the unwinding of syllables. Such control requires a *labelling* representing the temporal arrangement of syllables in the sample, in the form of a text file containing a series of timestamps corresponding to the *syllabic control points*. Placement and positions of the two SCP per syllable P_v and P_t are displayed in Fig. 2. Each syllable begins with a P_t, followed by a P_v in the syllabic nucleus. P_v are placed near the center of the vowel corresponding to the vocalic nucleus to ensure a correct pronunciation. The P_t are placed at a stable or silent zone in the cluster of consonants between two vowels. This guarantees an accurate control of the instant of occurrence of the next vocalic phase. When the consonant cluster ends with an unvoiced plosive, the corresponding P_t must be placed during the silence prior the explosion. Special treatments for the phrase initial

and final P_t are needed, in order to ensure the pronunciation of every phoneme entirely. The phrase initial P_t have to be placed at the end of the silence prior to the first phoneme, and the phrase final P_t must be placed at the beginning of the silence following the last phoneme.

SCPs have to be set beforehand. It can be done manually, identifying the locations of the SCP with the help of an audio editor and reporting them in a text file. This task can be automated, or semi-automated, as has been done in Delalez (2017), with the help of automatic speech recognition and segmentation algorithms, and simple rules for locating control points based on that segmentation.

3. Signal representation and vocoder

Labelled speech samples need to be processed in real-time. This is achieved by a real-time voice coder (vocoder) system based on the source-filter voice signal decomposition. In Delalez and d'Alessandro, 2017a, 2017b a real-time PSOLA vocoder was used (Le Beux et al., 2010b). Other real-time high-quality vocoder systems can be used as well. In Voks the WORLD vocoder has been chosen and modified, because of its high sound quality and ability to perform various spectral modification in real-time. The following subsections focus on the WORLD vocoder.

3.1. Source-filter model

WORLD is based on the linear source-filter model of speech production (Fant, 1970).

$$s(t) = \left(III_{T_0} g(t) + n(t)\right) * v(t) * l(t) \tag{1}$$

$$= \left(III_{T_0} g(t) + n(t)\right) * c(t) \tag{2}$$

where $*$ is the convolution operator, and $III_{T_0} = \sum_n \delta(t - nT_0)$. The source filter model is made of a source component, corresponding to phonation, and a filter component corresponding to the vocal tract and lip radiation. The source component is modelled as the sum of $III_{T_0} g(t)$ a periodic (harmonic) component and n an aperiodic (noise) component. g represents a glottal pulse. The aperiodic component accounts for the sound corresponding to the fast motion of articulatory organs during consonants (transient noise), but also for the turbulent airflow (breath noise, aspiration or frication noise) that accompanies periodic vibrations of the vocal folds during voiced sounds or fricative consonants. The evolution of both the periodic and aperiodic components are assumed to be slow compared to the fundamental periods. $v(t)$ is a time-varying linear filter accounting for the vocal tract action on the source signal. $l(t)$ represents the effect of lip radiation. v and l can be associated in a same filter c.

WORLD is based on a slightly different formulation of the source filter model. The glottal source g is regarded as a filter, and associated to c to form the periodic component filter $h_p(t)$. A second filter is defined for the aperiodic component h_{ap}. With this formulation, the harmonic component is written as a simple Dirac comb III_{T_0}, with period $T0$, filtered by a filter with response $h_p(t)$; similarly, the noise component is seen as white noise $n(t)$ filtered by a filter with response $h_{ap}(t)$:

$$s(t) = III_{T_0}(t) * h_p(t) + n(t) * h_{ap}(t) \tag{3}$$

In the spectral domain, that equation becomes

$$S(\omega_i) = III_{F_0}(\omega_i) H_p(\omega_i) + N(\omega_i) H_{ap}(\omega_i) \tag{4}$$

where H_p and H_{ap} are the Fourier transforms of h_p and h_{ap}. WORLD does not actually deal with H_p and H_{ap}; it uses parameters closely related to those. Those parameters are a total spectral envelope $(E(\omega_i))$ and another filter, representing the "aperiodicity ratio" $R(\omega_i)$. They are related to H_p and H_{ap} by the following relations:

$$\begin{aligned} E(\omega_i) &= |H_p(\omega_i)|^2 + |H_{ap}(\omega_i)|^2 \\ R(\omega_i) &= \frac{\left|H_{ap}(\omega_i)\right|^2}{E(\omega_i)} \end{aligned} \tag{5}$$

The aperiodicity ratio $R(\omega_i)$ indicates, in each frequency band, what fraction of the total spectral power comes from the unvoiced part of the signal. Together, those three parameters allow for resynthesis of a high-quality signal (Morise and Watanabe, 2018).

3.2. WORLD analysis

WORLD is a collection of C language source programs, distributed under the permissive 3-clause BSD license. The original WORLD software (Morise et al., 2016) is organized in two main parts. The analysis part can be used to estimate WORLD parameters for a given audio file. The synthesis part computes the voice signal corresponding to the WORLD parameters. It is possible to modify the voice parameters between analysis and synthesis, by manipulation of the intermediate WORLD parameters.

WORLD follows and improves the principles of STRAIGHT/TANDEM, a successful high-quality vocoder based on short-term Fourier analysis/synthesis of speech (Kawahara and Morise, 2011). The main features of these vocoders, in Eq. (3), is that all the spectral information is incorporated in the filter component (spectral envelope) of the model. The excitation component spectral envelope is flat, either a white noise or a flat pulse train. The filter spectral envelope is as smooth as possible.

The WORLD analysis module takes as input audio sample $s(\tau)_{\tau\in[0.T]}$, where T is the duration of the audio, and $s(\tau)$ is the audio sample at time τ. Assuming a source-filter model described in Section 3.1, parameters of the model $F(\tau_i)$ are estimated at evenly spaced time points or *frame* $(\tau_i)_{i\in[0.N-1]}$ (where N is the total number of analysis points).

The entire set of parameters $(F(\tau_i))_{i\in[0\ldots N-1]}$ for a given recording is called the *WORLD parameters*, whereas for a given time τ,$F(\tau)$ will be called the *WORLD frame* corresponding to τ. For each frame, three types of parameters are computed, the pitch component, the spectral envelope and the periodicity ratio. At time τ, a "WORLD frame" F_τ is then the triplet $(f, E(\omega_i), R(\omega_i))_{i\in[0\ldots N_{\text{fft}}/2]}$, where N_{fft} is the size of the discrete Fourier transform used.

The pitch component $f(\tau)$ is obtained by the HARVEST method (Morise, 2017). It represents the fundamental frequency in Hertz (0 for unvoiced speech) for each frame. The spectral envelope E is obtained using short-term Fourier analysis and the CHEAPTRICK method (Morise, 2015). For each frame a vector of $N_{\text{fft}}/2 + 1$ points representing the spectral amplitudes only is computed. The aperiodicity ratio R is computed using the D4C method (Morise, 2016). For each frame a vector of $N_{\text{fft}}/2 + 1$ points representing aperiodicity (between 0 and 1) is computed.

With a frame rate of 5 ms, a sampling rate of 48 kHz, a FFT size N_{fft} of 2048 points, and double precision real numbers, Voks parameter rate is 3.28 Mbyte/s (compared to 96 Kbyte/s for the 16 bits monophonic audio signal). WORLD analysis is an automatic and robust process that is performed off-line prior to synthesis.

3.3. Real-time synthesis: M-WORLD

The synthesis process performs the reverse operation of the analysis process: it takes a stream of WORLD frames $F(\tau)$ as input, and outputs a synthesized audio signal $s(t)$. The re-synthesized audio from WORLD parameters is not mathematically equal to the input audio, but it is almost perceptually identical to it (Morise and Watanabe, 2018). For Voks, the WORLD analysis parameters are exported as a Jitter/Max matrix. Using the Max SDK, new software was developed to package the original WORLD C functions as a real-time Max (Puckette, 2002) external called M-WORLD. M-WORLD takes modified WORLD frames as

input. The original pitch component is replaced by real-time pitch data coming from an input device. The spectral envelope and aperiodicity ratio vectors are also modified in real time based on input data.

The synthesis process is based on overlap-adding a train of filtered glottal pulses and filtered noise, based on Eq. (). The synthetic pitch contour is converted into a point process corresponding to $III_{T_0}(t)$. Each time point represents the position of a filtered glottal pulse, that is computed as the impulse response $h_p(t)$. Filtered noise $n * h_{ap}(t)$ is computed and added to the periodic component.

4. Real-time control and signal modifications

4.1. Time, tempo, and rhythm control

This section details the implementation of rhythmic control based on principles described in Section 2.

Rhythm control in Voks amounts to specifying, at a given instant t during performance time, a numeric value $\tau(t)$ (in seconds) and the *time index*, corresponding to a temporal position in the original sample. Once $\tau(t)$ has been computed, the corresponding WORLD frame $F_{\tau(t)}$ is selected, undergoes some modifications (described in Section 3), and is synthesized to deliver the audio signal.

Three different rhythm control modes are available in Voks: syllabic, scrub and speed. The syllabic mode is actually a rhythmic control mode: it is akin to tapping or hand clapping synchronously with syllables to create the rhythmic patterns. The Scrub mode corresponds to the direct control of the time index, and the speed mode to controlled variation of reading tempo of the samples. These three modes of control are likely to produce very different types of musical gestures.

4.1.1. Syllabic control

The syllabic rhythm control mode is close to natural rhythm control in voice production. Rhythmic beats correspond to syllables. The opening and closing motions of the mandible for the natural voice corresponds to a two states button in performative control, as explained in Section 2. When using syllabic rhythm control, the control parameter of the rhythm control device $\rho(t)$ can take on two values, 1 and 0. In addition to those two control states, the system itself can be in two states internally: *frozen state* and *running state*. The two internal states describe the ratio between the reading speed of input samples and synthesized samples. These states are represented in Fig. 3: the $\rho(t)$ parameter is the bottom green line, and the time index is the blue curve just above. The corresponding oscillogram and spectrogram, together with phoneme labels are above the blue line. The circled *2* indicates a zone where the system is in the frozen state, and the *3*, a zone where it is in the running state.

The time index $\tau(t)$ represents the actual position in time of the analysis frame used for synthesis. For this reason, $\tau(t)$ takes its values between time 0 (beginning of the utterance) and the utterance duration. When the system is in the frozen state, the time index $\tau(t)$ is constant, waiting for a change in $\rho(t)$. The synthesis parameters are "frozen," resulting in the repetition of the same spectral pattern. Note that intensity and pitch can still vary, giving life to the sound. As soon as a change in $\rho(t)$ from 0 to 1 (or from 1 to 0) is registered, the system switches to the running state. The time index $\rho(t)$ begins increasing with a constant, positive slope, called *articulation speed*, until the next vocalic (resp. transient) control point is reached. The system then switches back to the frozen state, and the time index again becomes constant; its value is now that of the control point in question. In other words, the articulation rate controls the reading speed of input sample. It is equivalent to the speed of articulation in natural speech.

Three possible relationships between the recorded samples and the performer's gestures timing are possible, as illustrated in Fig. 4. When the performer's gesture is faster than the recorded samples, a change in $\rho(t)$ happens before the time index $\tau(t)$ has reached its target value. In this situation $\tau(t)$ jumps to the control point it was aiming for and immediately sets the next control point as its new target value. This discontinuity manifests by a vertical slope in the graph of $\tau(t)$, as can be seen in the portion of Fig. 4 marked with a circled 1. This discontinuity ensures that the delay of the time index relative to the performer's gesture is bounded. When the performer's gesture is slower than the recorded samples, the system enters the frozen state, in which $\tau(t)$ is constant. This manifests by a horizontal slope as seen on the portion of the signal marked with a circled 2. The third situation never happens in practice, when the recorded samples and performer's gestures are exactly with the same timing.

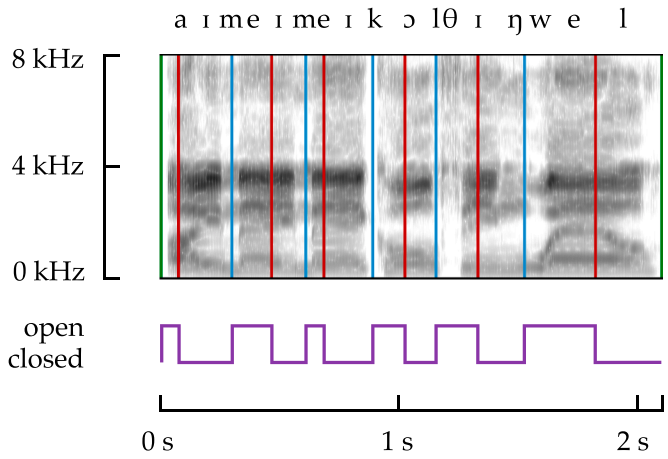


Fig. 2. Syllabic control points for the sentence *"I may make all thing well"* (/aɪmeɪmeɪkɔlθɪŋwel/). Nucleic control points are marked with red lines, transient control points with cyan lines, superimposed on the spectrogram of the audio sample. Green lines indicate the starting and ending points. The purple graph indicates which portions of the signal will be played when the controller is respectively open and closed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The articulation speed, i.e. the slope of the portions of the graph similar to the one marked with a circled 3 in Fig. 4, can be set using the graphical interface, or mapped to a control device. Increasing its value makes discontinuities in the time index less likely. The farther away from 1 it is, the more temporal distortion of some phonemes is noticeable, which can reduce the realism and intelligibility of the synthesized voice.

The syllabic control mode is needed when one wants to play a precise rhythm, as when singing a song or saying a text, where syllable nuclei coincide with note onsets or beats. This mode allows for accurate syllabic placement in time.

4.1.2. Scrub control mode

Scrub mode allows for replay by directly controlling the time index. Contrary to the syllabic rhythm control method, the rhythm produced has no precise anchoring in the phonetic content. The scrub mode relies on a continuous control parameter $\eta(t)$. We assume $\eta(t)$ only takes on values in the interval $[0, 1]$, O representing the beginning and 1 the end of the voice utterance ($\eta(t)$ is normalized by T, where T is the length of the pre-recorded audio). Scrub mode then consists in linearly mapping each value of the continuous parameter $\eta(t)$ to a value for the analysis time τ:

$$\tau(t) = T\eta(t) \tag{6}$$

By increasing η with a given speed, the performer can play the utterance faster or slower; by decreasing η, they can also play backwards, and by making it constant, play a steady sound. The effect of playing Voks in the scrub mode, and variation of η are displayed in the left half of Fig. 5.

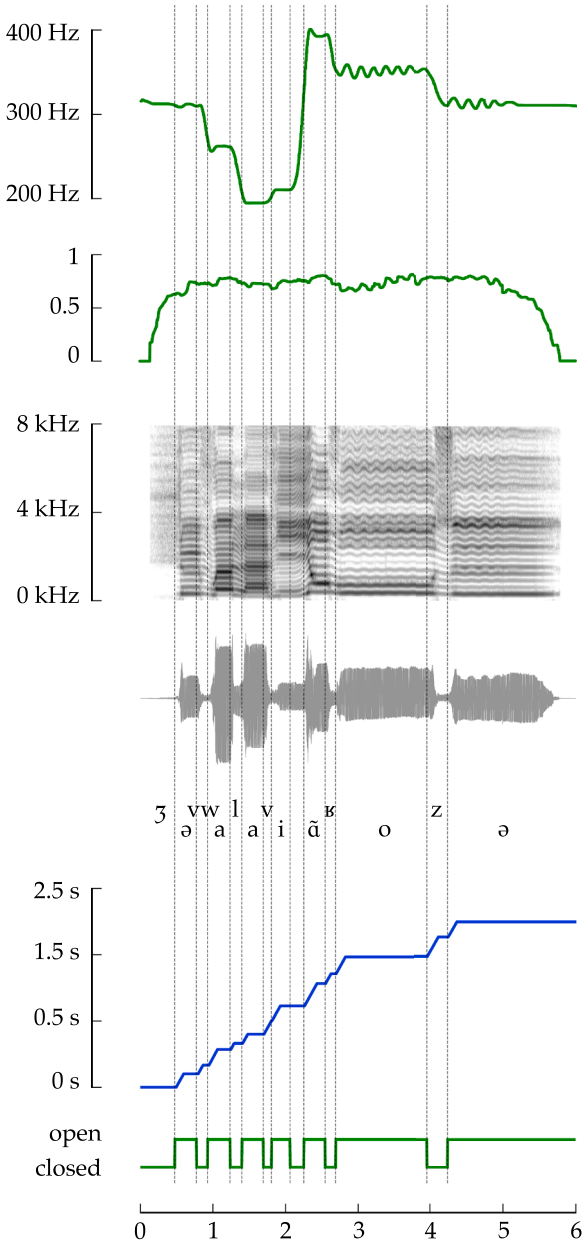


Fig. 3. Performance using Voks in the syllabic control mode. French sentence "Je vois la vie en rose" (/Z@*vwalavi*ĀK*oz*@/, duration 2.5 s). Synthesis duration: 6s. Input control data in green, internal data in blue, output audio in greyscale. From top to bottom: pitch, normalized vocal effort, spectrogram, oscillogram, phonemic labels, internal time index τ (blue), binary rhythm control (green). Times at which the rhythm controller is pressed or released are marked with a vertical dotted line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The scrub mode is a direct *time* control mode. It is not well suited for accurate syllabic placement in time. It is rather suited for DJing-like effects and gestures, like the direct manipulation of a play-head on a turntable.

4.1.3. Speed control mode

Speed mode gives control on the sample replay speed, i.e. the *tempo*. This corresponds to the first derivative of the position of the time index τ in the recorded utterance:

$$\begin{aligned} \frac{d\tau}{dt}(t) &= 1_{v(t)\neq 0} g(\kappa(t)), \text{ or equivalently} \\ \tau(t) &= \int_0^t 1_{v(t)\neq 0} g(\kappa(u))du \bmod T \end{aligned} \tag{7}$$

where:

- $\kappa(t)$ is the value of the speed control parameter.
- g is a function that maps values of κ (between 0 and 1) to speeds both negative and positive. It is continuous and increasing, antisymmetric around 0.5 (that is, $g(0.5-\kappa) = -g(0.5-\kappa)$, and features a strip around 0.5 where its value is zero.
- the $1_{v(t)\neq 0}$ factor denotes the function that equals 0 when the vocal effort v is zero, and 1 otherwise.
- $x \bmod T$ refers to the remainder in the euclidean division by T, the length of the utterance.

The $1_{v(t)\neq 0}$ factor appears in the equation to keep the utterance from silently progressing when the vocal effort is zero. Using the remainder in the division by T makes the sample loop back to the beginning when its end is reached.

The right half of Fig. 5 illustrates various behaviors of speed mode: it begins with looping slower and slower with a positive speed; at around 5s, the effort drops to 0, stopping progression of the time index until it becomes strictly positive again. Around 8s, the control value $g(\kappa)$ becomes 0, stopping the progression again, before taking lower and lower negative values, which has the effect of playing the audio (reversed in time) and accelerating.

4.2. Melodic control

Pitch (melody) is controlled by the player's gesture. For synthesis, the fundamental frequency $\phi(t)$ is obtained by simple re-scaling of the gestural contour. This contour can be produced by various interfaces as detailed below. As the pitch is often expressed in the MIDI format, $\phi(t)$ in Hertz is obtained by:

$$\phi_{Hz}(t) = 440 \cdot exp\left(\frac{\phi_P(t) - 69}{12}\right)$$

with $\phi_P(t)$ the pitch in MIDI (assuming an equal temperament). The frequency $\phi_{Hz}(t)$ is then fed to the M- WORLD synthesis module.

4.3. Glottal source control

The spectral representation of WORLD allows for spectral modification of voice quality. Voice quality parameters are related to the time and spectral features of the voice source (d'Alessandro, 2006). The main voice quality parameters are vocal effort, vocal tension and noise in the source. These parameters can be modified in the spectral domain, according to spectral modeling of the glottal source (d'Alessandro and Doval, 1998; Doval et al., 2006). Using this theory, WORLD spectral representation is well suited to voice quality transformations.

4.3.1. Glottal formant

Voice tension is an important voice quality parameter. In the spectral domain, the glottal pulse corresponds to a peak of the spectral envelope in the region of the first harmonics. Changing the voice tension results in a shift of this peak, called the *glottal formant* (Doval et al., 2006). This makes the voice sound more tense, when the center frequency of glottal formant is raised, and more relaxed when the center frequency of glottal formant is lowered.

As the glottal formant center frequency is situated near the fundamental frequency, change in the relative level of the first and higher harmonics emulates the effects of a glottal formant shift. To manipulate

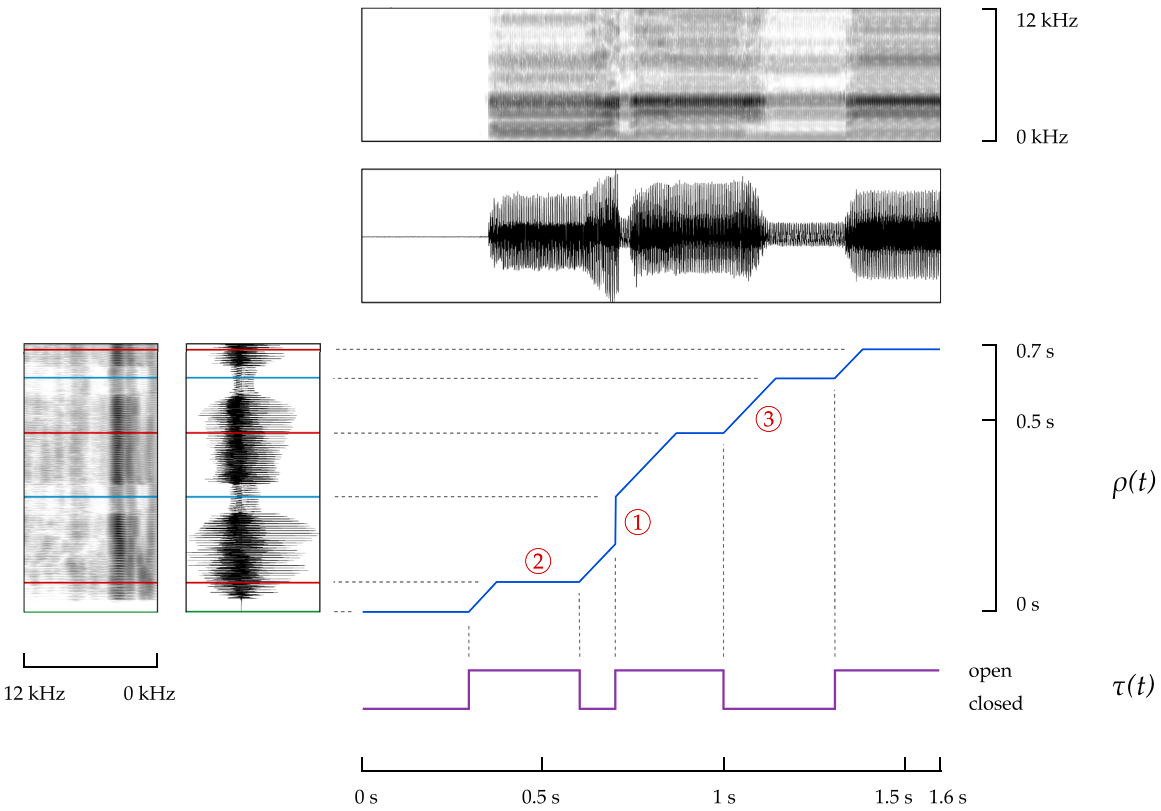


Fig. 4. State of the control parameter (open or closed, in purple) and the associated temporal evolution of the time index (in blue). On the X-axis, performance time; on the Y-axis, recording time. On the left, spectrogram and waveform of the original sample, with the control points marked as on Fig. 2. On top, spectrogram and waveform of the generated sound. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the harmonics, a comb filter is applied to the spectral envelope. Separation of the harmonics is done by multiplying the whole power spectrum of the periodic part $|H_p(\omega)|$ by a function that is approximately equal to 1 for $\omega = 2\pi f$ and to 0 for $\omega \geq 2{\cdot}2\pi f$. A sigmoid function is used because a smooth function is required. The harmonics are weighted, according to the desired glottal formant modification using a γ parameter. This is equivalent to weighting the entire power spectrum with a function $g_\gamma(\omega)$. The modified power spectrum is:

$$\begin{aligned} |H_p^{'}(\omega)|^2 &= \Big((1-\gamma)E_1(\omega) + \gamma E_{\text{sup}}(\omega)\Big)\big|H_p(\omega)\big|^2 \\ &= g_\gamma(\omega)\big|H_p(\omega)\big|^2 \end{aligned} \tag{8}$$

where E_1 and E_{sup}, the envelopes corresponding respectively to the first and higher harmonics, are sigmoids defined by

$$\begin{aligned} E_1 &= 1 - \tanh\left(8\left(\frac{\omega}{2\pi F_0} - \frac{3}{2}\right)\right) \\ E_{\text{sup}} &= 1 + \tanh\left(8\left(\frac{\omega}{2\pi F_0} - \frac{3}{2}\right)\right) \end{aligned} \tag{9}$$

with F_0 the fundamental frequency. In terms of the total power spectrum and the aperiodicity ratio, Eq. (8) becomes

$$\begin{aligned} E^{'}(\omega) &= [((1-R(\omega))g_\gamma(\omega) + R(\omega))]E(\omega) \\ R^{'}(\omega) &= \frac{R(\omega)E(\omega)}{E^{'}(\omega)} \end{aligned} \tag{10}$$

The weighting of the first harmonic can be seen in the second graph in Fig. 6 : here the first harmonic exceeds the envelope of the harmonics of the original sound.

4.3.2. Spectral slope

A higher vocal effort corresponds to higher intensity and higher spectral richness (lower spectral tilt). Conversely, a softer voice corresponds to lower intensity and the attenuation of higher harmonics (higher spectral tilt). In the spectral domain, an additional spectral slope, that is, a low-pass filter of order 1, with controllable cutoff frequency, is applied to the harmonic part:

$$H_p^{'}(\omega) = \frac{H_p(\omega)}{\sqrt{1 + \left(\frac{\omega}{\omega_c}\right)}} \tag{11}$$

where $\omega_c$is the cutoff frequency, used as a control parameter.

The fourth graph in Fig. 6 shows the difference between the original harmonic envelope and the harmonic envelope of the same vowel with a spectral slope applied.

Eq. (11) can only decrease vocal effort (produce a softer voice). Convincingly increasing vocal effort is more difficult, because higher order harmonics are often masked by noise. More sophisticated methods are needed (e.g. distortion Perrotin and d'Alessandro, 2016). In the present study, only lowering vocal effort is implemented.

4.3.3. Periodic-aperiodic ratio

A welcome feature of WORLD's parametric representation is its built-in separation of the periodic and aperiodic components of the signal. By varying the respective amounts of noise and voicing in the signal, one can achieve various interesting effects. The variation is achieved by simple amplitude scaling:

$$\begin{aligned} H_{ap}^{'}(\omega) &= \alpha H_{ap}(\omega) \\ H_p^{'}(\omega) &= \beta H_p(\omega) \end{aligned} \tag{12}$$

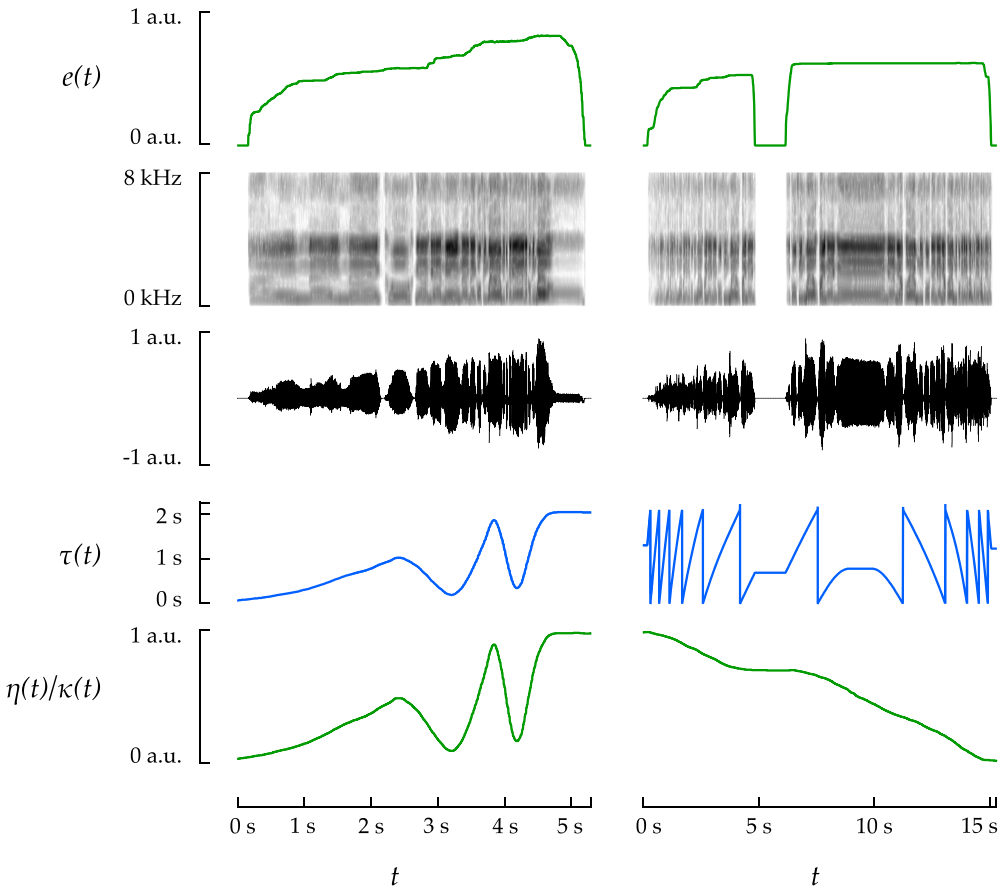


Fig. 5. Short performance using Voks in scrub mode (left) and speed mode (right). Sentence "I may make all thing well" (duration 2 s). Synthesis duration: 10 s (left) and 20 s (right). From top to bottom: vocal effort, in arbitrary units, in green, spectrogram and waveform of the output sound, internal time index τ, in blue (see Section 2), scrub/speed control parameter η (left) or κ(right), in arbitrary units, in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where α and β are control parameters. Significantly decreasing β with respect to α produces a lax voice (joined to a lowering of vocal effort and/or tension). Setting it equal to zero produces a very convincing whispered voice.

The weighting of the aperiodic part independently of the periodic one can be seen in the third graph of Fig. 6: while the harmonic peaks are unchanged, the noise in between them is increased.

4.4. Vocal tract control

The size of the vocal tract has a dramatic effect on the spectrum. This effect is generally noticeable; the perceived size of the vocal tract is an important parameter, that allows listeners to instinctively estimate e.g. the perceived age and gender of a voice. Linear frequency warping is easily implemented thanks to the source/filter decomposition in WORLD:

$$\begin{aligned} H'_{ap}(\omega) &= H_{ap}(\lambda\omega) \\ H'_{p}(\omega) &= H_{p}(\lambda\omega) \end{aligned} \tag{13}$$

where λ is a warping factor.

For a uniform vocal tract (corresponding to the neutral vowel /ə/) the λ is a warping factor corresponding to a *vocal tract size factor*: the scaling factor of a fictitious "vocal tract" whose characteristics are described by H_{ap} and H_p. For other vowels, the situation is more complex and the linear warping factor λ does not correspond exactly to a vocal tract scaling factor. However, the perceptual effect of linear warping corresponds well to an apparent vocal tract size change, at least for voices. According to Fant (1966): "The scaling of children's data from female data comes closer to a simple factor independent of vowel."

The (e) graph in Fig. 6 shows the power spectrum of a vowel played with a vocal tract size factor of 0.8. While the space between harmonics remains the same as in the original signal, their envelope is a stretched version of the original envelope, with a stretching factor of 1.25, the inverse of the vocal tract scaling factor.

Values of λ close to 1 ($0.85 \lesssim \lambda \lesssim 1.2$) allow for turning a perceived male voice into a perceived female ($\lambda < 1$) or the other way around ($\lambda > 1$). More extreme values of λ lead to more extreme effects, such as the "chipmunk voice" for values of λ significantly lower than 1.

Note that the transformation $H(\omega) \mapsto H(\lambda\omega)$ is applied both to the harmonic part of the signal and the noise. Much of the noise comes from the motion of articulators (e.g. tongue, lips, etc.), which should not, in principle, be affected by a vocal tract change. However, some of the noise does directly come from the vibration of vocal folds. Decoupling that noise from the harmonic vibration by applying the transformation only to H_p and not to H_{ap} gives rise to a less natural voice, and in some cases even gives the impression of two voices being heard at the same time.

5. T-Voks: theremin-controlled Voks

5.1. Instrument design

T-Voks is a theremin-controlled performative voice synthesizer based on Voks (Xiao et al., 2019). The theremin is particularly spectacular because it is played by free-hand motion, in the "ether" without any contact between hands and the instrument. Pitch is controlled using the right hand[2] and the theremin's vertical antenna, while volume is controlled with the other hand using the looped antenna (see Fig. 7).

[2] Right-hand (resp. left-hand) means here the preferred hand (resp. non preferred). It was actually the right hand in our theremin experiments, but it could be the left as well for another player.

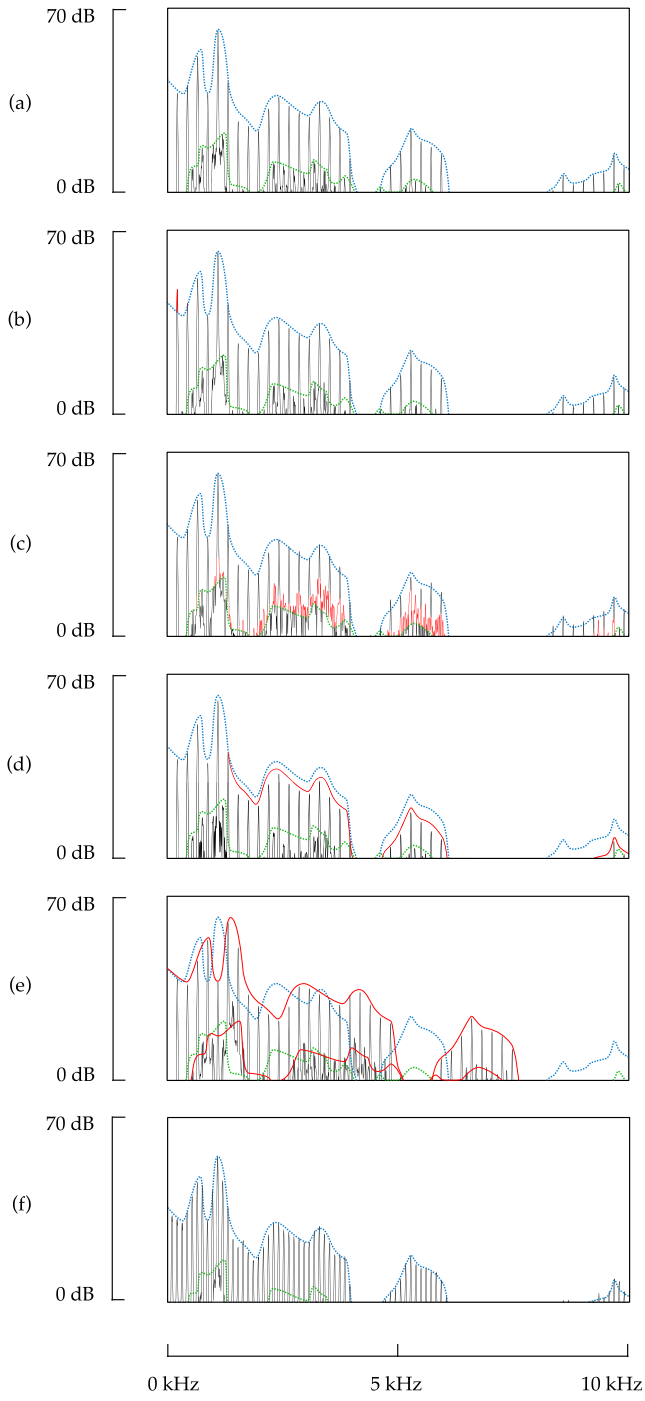


Fig. 6. Power spectrum of the vowel /a/ played (a) with no timbre modifications (b) with a shifting of the glottal formant simulated by applying a gain of 10 on the first harmonic (c) with a gain of 10 applied on the aperiodic part (d) with the application of a spectral slope with a cutoff frequency of 1000 Hz (e) with a simulated vocal tract elongation with a factor of 0.8 (f) with fundamental frequency one octave higher. In (a) an envelope has been manually drawn around the peaks corresponding to the periodic (blue) and aperiodic (green) parts of the signal. These envelopes appear in (b), (c), (d), (e) and (f); in (b) and (c), deviations of those envelopes from the original ones are also marked in red. In (d) an updated envelope has been drawn as a red dotted line, also manually. In (e), a stretched version of the envelopes with a stretching factor of $\frac{1}{0.8}$appear as a red dotted line. In (f), the original envelopes remain relevant. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

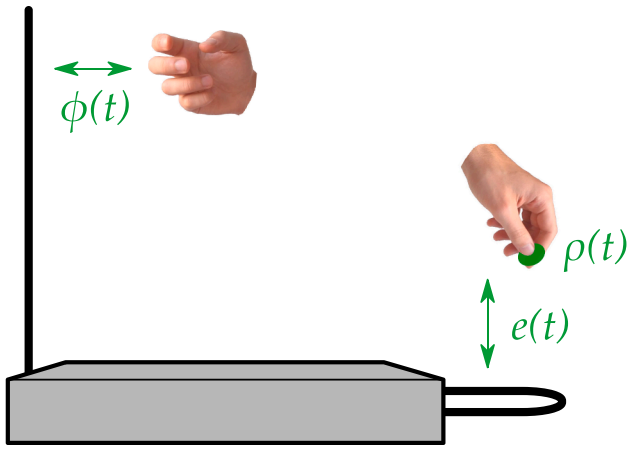


Fig. 7. The T-Voks control interface. The antennae of a theremin respectively control pitch ($\phi(t)$) and vocal effort ($e(t)$), and a pressure sensor located in the hand controls vocal effort. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.1.1. *Pitch control*

Pitch control using the theremin is challenging. Like in fretless string instruments the player must find the pitch in a continuum, without predefined visual or tactile steps or marks. An additional difficulty comes from the free hand motion, without haptic feedback, such as the neck of a violin. Despite widely held views on the theremin's difficulty, many players manage to accurately play melodies within several months of practice. Theremin performance is a highly individual art, with no single standardized technique, though some educational resources do exist (Theremin world: Learn to play the theremin, 2019). To reliably find notes and intervals, thereminists have developed specific hand and finger gestures, such as opening and closing the hand toward the antenna for the span of an octave and shaking the right hand to produce a vibrato. In addition, while only the nearest point between the body and each antenna directly modifies the output tone, the rest of the player's body also influences modifies the electromagnetic fields detected by the antennas.

5.1.2. *Vocal effort and syllabic control*

Unlike most traditional instruments, which must be actuated to produce a sound, the theremin outputs a tone by default and must be explicitly silenced. Raising and lowering the volume-control (left or non-preferred) hand in relation to the horizontal antenna is used to cleanly delineate notes by removing unwanted glissandos in between. The volume control is associated with vocal effort in T-Voks. The quality of these movements sculpts the attack, duration and decay of each note, enabling a wide range of articulations, though legato across large intervals and sharp staccatos are difficult to achieve. Larger movements of hand, wrist and arm defines the dynamics across phrases.

An additional pressure sensor for the volume-control hand allows for syllabic rhythm control. When the pressure value is lower than a threshold ("open hand"), the control parameter ρ (Section 4.1) is equal to 1; when ρ is higher than the threshold, ρ is equal to 0.

The movement to press the sensor must be comfortable enough to perform repeatedly, reliable enough for the system to detect, and fast enough to articulate several syllables in rapid succession. Moreover, it should not interfere with other gestures of the same hand, wrist, and arm for articulation and dynamics. After experimenting with several different sensor placements, consistently satisfactory results were found with the syllabic control button positioned against the first knuckle of the index finger, held in place by a ring and pressed by the thumb. Only the thumb and forefinger of the volume hand are involved in syllabic control, leaving free range of motion for the rest of the hands and fingers, as well as the wrist, forearm, and elbow.

The addition of syllable control alters the volume hand's techniques. A syllable change at the same time as a note change hides the usual glissando to the new note, and removes the need for the volume hand to dip between the notes. Syllables also add more attack and textural variation, liberating the volume hand to focus on phrase-level dynamics rather than note-level articulation. While the pitch hand is not involved in syllabic control, the addition of articulated syllables has inspired new pitch-control gestures. Note that the addition of syllable advancement introduces a significant cognitive load to an instrument that already requires full concentration for playing.

5.2. Hardware and computer interfaces

5.2.1. Theremin

The theremin used in T-Voks is the Etherwave Plus from Moog, which features control voltage (CV) analog outputs for pitch and volume. Pitch is associated to the vertical right-hand controlled antenna. The pitch CV ranges from 2.5v to 4.5v (with a change of 1 V per octave of the theremin's pitch). Vocal effort is associated to the left-hand horizontal antenna, or volume antenna. The volume CV from our theremin measured from 0 to 5.5v. Voltage must be digitized before computer processing. This is achieved using an Arduino Uno. Each CV output is fed into an analog input pin of Arduino, with the CV ranges regulated to the 0 to 5 V range of the Arduino.

5.2.2. Syllabic control interface

In a first experiment, a force sensitive resistor (FSR) is used to control the sequencing of syllables. Its data uses another analog input of the Arduino. Our FSR is attached to a signal conditioning circuit from Interface-Z[3], whose output vary from 0 to 5 V between maximum pressure and no pressure. Although the output of the FSR belongs to a continuous range of values, it is only used as a binary switch, allowing the player to choose between two states, pressed and released. Its cable runs down the palm, along the arm, around the shoulder and out behind the player. It is secured by an elastic band around the palm and easily hidden by a long-sleeved shirt. In a second experiment, a wireless interface (Bluetooth mouse controller) is used, giving more freedom to the performer, and direct input to the computer.

6. C-Voks: tablet-controlled Voks

C-Voks (standing for *C*alligraphic *Voks*) is a voice synthesizer based on Voks controlled by a graphic tablet. The graphic tablet has been used as a control interface for singing instruments before (Feugère et al., 2017). In addition to the tablet, a MIDI controller is used for control of rhythm and other parameters, as well as preset management. C-Voks is a new implementation of the Vokinesis and Calliphony systems. The main differences between these systems are the vocoder used (RT-PSOLA for Vokinesis and Calliphony, WORLD for C-Voks) and the graphical user inferface. They are otherwise very close as far as sound quality, playing modes and musical functionality are concerned.

6.1. Instrument design

C-Voks is a bimanual voice synthesizer controlled by a pen on a graphic tablet (preferred hand) and an additional interface, which allows for the control of syllable sequencing, voice quality and vocal tract scaling. The instrument design relies on a different approach toward chironomic control than for the theremin. In the case of C-Voks, very accurate motions of the pen on a surface allow for multimodal reinforcements (Perrotin, 2015). Taking advantage of the cooperation among visual, audio and kineaesthetic modalities, C-Voks is easily accessible to beginners, who can perform simple melodies almost at the first training session.

6.1.1. Pitch control

C-Voks pursues the lineage of a series of instruments based on the graphic tablet and writting/drawing gestures: Cantor Digitalis (Feugère et al., 2017), Calliphony (Le Beux et al., 2010) and Vokinesis (Delalez and d'Alessandro, 2017). Playing C-Voks is reminiscent of playing Cantor Digitalis and other tablet-based melodic instruments.

The graphic tablet offers a three dimensional control: the two position coordinates of a stylus on the rectangular surface of the tablet, as well as a value for the vertical pressure of the stylus on the surface.

The graphic tablet as a musical instrument already has some precedent (Zbyszynski et al., 2007). It is particularly well-suited to intonation pitch control in speech (d'Alessandro et al., 2011) and singing (d'Alessandro et al., 2014). This is because on the one hand the gestures on the tablet re-use the manual skills acquired for hand writing and drawing, and on the other hand because the visual, kineaesthesic and auditory modalities collaborate in the task (Perrotin and d'Alessandro, 2016). In C-Voks, horizontal position of the stylus on the surface of the tablet is mapped to pitch. A mask with lines locating the notes of a chromatic scale on the surface, shown in Fig. 8, is affixed to the tablet for visual assistance of the player.

6.1.2. Time and rhythmic controls

C-Voks offers three different modes for rhythm and timing control: syllabic rhythm control, speech rate (speed) control and signal scrubbing. Depending on the timing control mode, different kinds of control parameters are needed: a continuous control for the speech rate and scrub modes, or a discrete, binary parameter for the syllabic control mode.

When playing in scrub or speed mode, the time parameter is continuous, and although any continuous control dimension can be used, one of the axes of a graphic tablet is especially well-suited to the task. In this case the stylus is used for two simultaneous tasks: melodic control and timing control. Gestures on the surface can create new sounds and new ways to play with the voice. Moving the time index τ with fast motions, using either scrub or speed mode, results in a rapid succession of syllables, either forward or backwards, too fast and chaotic to have been produced by a human, but still retaining most of the attributes of voice. When using speed mode, each time the time index τ reaches its maximum value, it loops back to 0. For sentences that last several seconds, this just results in the same text being repeated over. However, for shorter utterances, typically single syllables, fast repetition makes the audio less voice-like, granting it an acousmatic quality. Depending on speed, pitch, source sample used, and whether it is being played forward or backward, the resulting sound can evoke different textures, like e.g. babbling, bubbles, or a motor.

When playing in syllabic mode, the time parameter is discrete, and a two state button or keyboard needed, like for T-Voks. In this case the bimanual control is divided in melodic and vocal effort control (preferred hand) and syllabic rhythm control (non preferred hand).

6.1.3. Vocal effort and voice quality controls

The voice parametric representation in M-WORLD allows for many kinds of voice quality control in C-Voks : vocal effort, vocal tension, periodic-aperiodic ratio, apparent vocal tract size.

The most important voice quality parameter is vocal effort. This parameter combines volume and spectral variation of the vocal source. The stylus pressure is used to control vocal effort in C-Voks, whatever the timing control mode. Other parameters are controlled using a MIDI controller with buttons, knobs and sliders:

- One of its buttons is used as the temporal control when playing in syllabic mode.

[3] https://www.interface-z.fr/pronfiture/contact/148-pression-force-fsr.html.

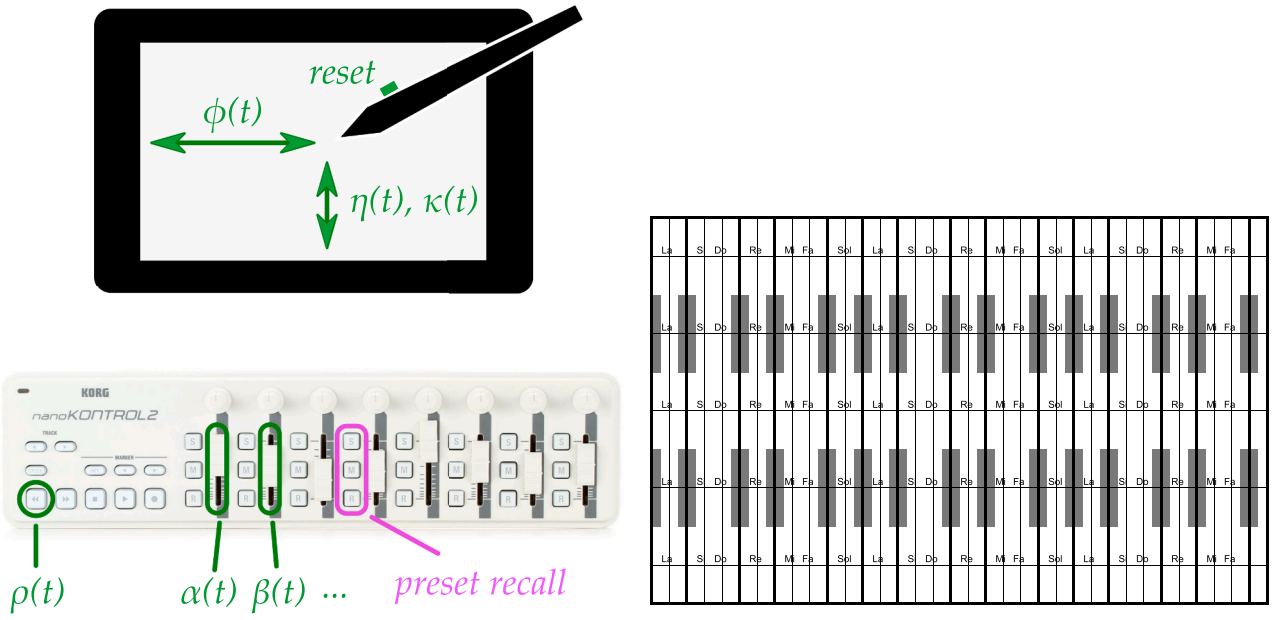


Fig. 8. The C-Voks control interface, and mask affixed to graphic tablets for pitch accuracy. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 9. A performance (d'Alessandro et al., 2019) using one instance of T-Voks (left) and three instances of C-Voks. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- Sliders and knobs control various additional parameters such as vocal tract scaling factor, articulation speed, overall sound level and many more.
- Numerous buttons allow for easy selection of presets.

Vocal effort is varied by the combination of three signal modification:

- Glottal formant shift, as described in Section 4.3, to increase/decrease tension.
- Application of a different gain to the periodic and the aperiodic part of the signal, as described in Section 4.3.3: for a low vocal tension, the periodic part is attenuated and the aperiodic part enhanced.
- Variation of the cutoff frequency of the spectral slope, as described in Section 4.3.2: when the vocal effort parameter is high, the cutoff frequency is set to a high value, allowing more higher harmonics to make it through than in the converse case.

In addition to the periodic/aperiodic ratio modification related to vocal effort changes, the respective gains of the periodic and aperiodic parts of the signal are also control parameters that can be set directly. Muting the periodic part then results in whispered voice, which the performer can then modulate to produce breathing, blowing, hissing sounds.

The vocal tract size can be changed by spectral morphing (Section 4.4). Vocal tract size parameter expands or contracts the voice spectral envelope by a given factor. Shortening the vocal tract gives a female-sounding voice using a sample recorded by a male speaker, and vice versa (with a vocal tract scaling factor equal to about 0.8 and 1.22 respectively). More extreme changes give "child" or "giant" voices.

6.2. Hardware and computer interfaces

6.2.1. Graphic tablet

The graphic tablet used is a Wacom Intuos Pro. It is composed of a flat, rectangular surface, and a stylus. When the tip of the stylus is in contact with the surface, its position along both the X- and the Y-axis, as well as its pressure on the surface, are sent to Voks. The position along the X-axis is mapped to pitch by a linear relationship, the pressure is mapped to vocal effort, and the position along the Y-axis is mapped to

the time control parameter, respectively $\eta(t)$ in scrub mode and $\kappa(t)$ in speed mode (see Sections 4.1.2 and 4.1.3). In syllabic mode, position along the Y-axis is not used.

A push button on the stylus is also used for resetting the time index value τ at 0 when in syllabic and speed mode.

6.2.2. MIDI controller

In addition to the graphic tablet, a MIDI controller Korg Nanokontrol 2 has been used for concerts. It is a MIDI controller that features 8 groups of controls — each composed of one slider, one knob, and three push buttons — as well as 11 more push buttons on the left. The sliders and knobs are mapped to the continuous parameters that are not controlled by the tablet : vocal tract scaling factor λ, voiced and unvoiced factors α and β, etc. One of the 11 push buttons on the left is mapped to the syllabic parameter ρ when in syllabic control mode (Section 4.1.1).

The push buttons from the 8 groups, originally intended for the soloing, muting, and recording of tracks, form an 8x3 matrix that is used in C-Voks for preset recalling. During rehearsals, the performer can define *presets*, that is, predefined values of:

- a sample to be recalled
- specific values for the parameters,
- a playing mode

which will be attached to one of the 24 buttons, and recalled at the time of the performance by simply pressing the corresponding button.

7. Comparative perceptual evaluation

7.1. Singing synthesis challenge fill-in the gap

Singing synthesis evaluation is an important but difficult task. This section reports on the Singing Synthesis Challenge held at SCA Interspeech 2016, in San Francisco, USA (Online, cited 2020-06-12, 2020), where Voks has been evaluated in a comparative perceptual paradigm. This international challenge aimed at perceptual comparison of singing systems. This challenge was the third international singing synthesis evaluation, using common material shared by different research groups, following evaluation at Stockholm Musical Acoustic Conference in 1993 (Session synthesis of singing, 1993) and at ISCA Interspeech Conference in Antwerpen (Synthesis of singing challenge, 2007), in 2007. The Singing Synthesis Challenge "Fill-in the Gap (FiG)", was organized by one of the authors as a special session at the Interspeech 2016 conference in San Francisco, on september 10th 2016 (see a presentation of the challenge at Online, cited 2020-06-12, 2020), in the framework of the ChaNTeR project. The following results have been presented at the closing ceremony of Interspeech 2016 in San Francisco, but have never been published. New statistical analyses of the results are given in the following sections.

7.1.1. Presentation of the challenge

The task was to synthesize well known jazz standards with new lyrics, written for the occasion. Two popular song where chosen: "Summertime" (song S) music by George Gershwin (1934), and "Autumn Leaves" (song A) music by Joseph Kosma, originally in French "les feuilles mortes" (1946). These songs were selected because they are international jazz standards, innumerable versions of these songs in many languages and various vocal styles (opera, pop, jazz ...) have been recorded. Original lyrics have been written for both songs in English and French for the FiG challenge. The scores are displayed in Fig. 10. Participants were free to select their preferred version, or translate the lyrics into any language. A huge number of recordings and instrumental playback was available (e.g. on the web) and could be used for reference, acoustic analysis, machine learning, or comparison. The listening test was only performed for a capella (unaccompanied) versions of the songs, although the participants were also encouraged to produce accompanied versions for playing during the InterSpeech conference.

All aspects of singing synthesis and all methodologies were welcome, including both off-line (studio) singing synthesis systems, with no limits on time for producing the result, and performative (real-time) singing instruments. The Special Session FiG has been announced in December 2015. The musical material, score and lyrics, and the FiG Challenge rules and instructions were issued on January 21th, 2016, about two months before the Interspeech 2016 paper submission deadline, March 23th, 2016.

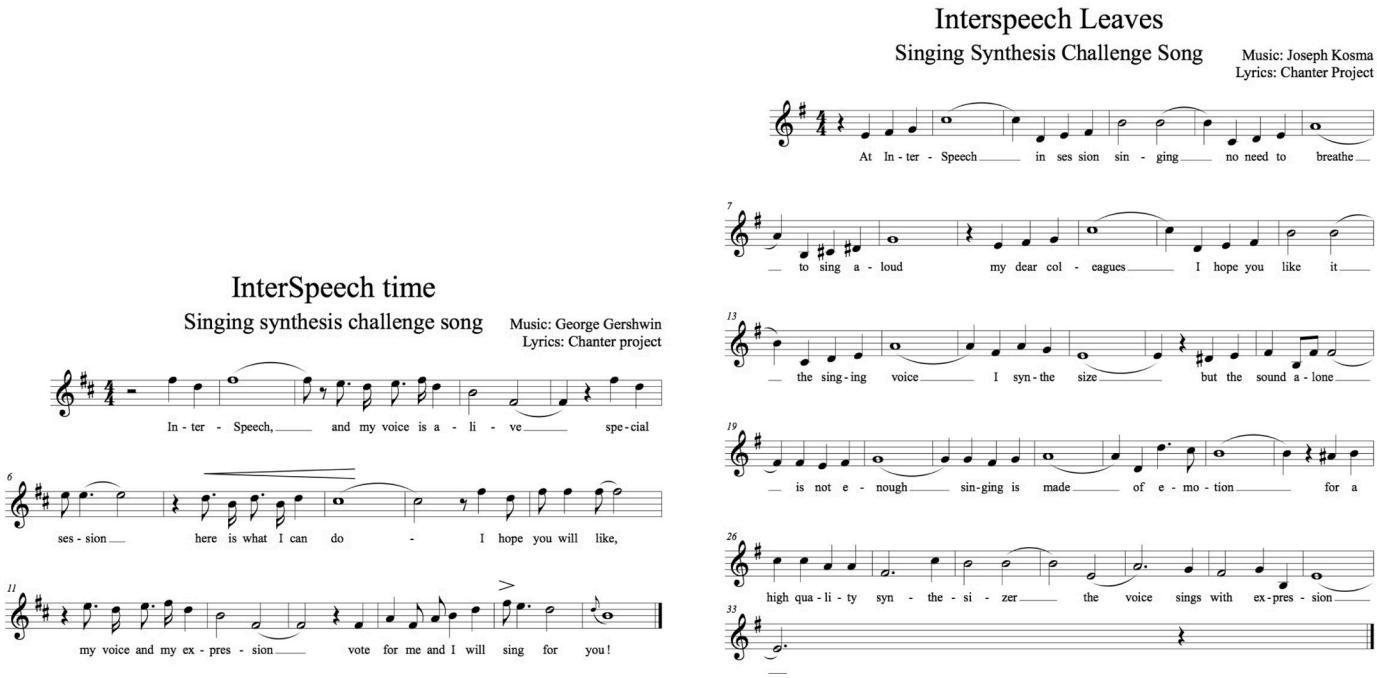


Fig. 10. Songs for the Singing Synthesis Challenge Fill-in the Gap at Interspeech 2016. The lyrics have been especially written for the challenge.

7.1.2. Participant to the challenge and test methodology

A number of papers were submitted to the Special Session, and among them 6 research groups were selected and participated in the singing synthesis challenge. Table 1 summarizes the languages, numbers of submitted songs, voice genders and participating labs. For a detailed description of each system, the reader is referred to (Online, cited 2020-06-12, 2020): the WBHSM concatenative synthesizer (UPF, Barcelona) (Bonada et al., 2016), ISIS, the Ircam Singing Synthesizer (Paris)) (Ardaillon et al., 2016), the Seraphim system (A*STAR, Singapore) (Chan et al., 2016), the Bertsokantari system (UPV, Bilbao) (del Blanco et al., 2016), the ACAPELA singing synthesis system (Mons) (Cotescu, 2016), and Calliphony, an earlier implementation of C-Voks. For the sake of simplicity, the system is coined C-Voks. Overall, 14 samples of synthetic songs were used for subjective evaluation. The total duration of the 14 samples was 11 mn 30 s. Therefore performing the test was relatively fast and easy. The samples were long enough, on average 49.3 s, to elicit a true musical appreciation, encompassing sound quality, musical quality, singing style, musical interpretation, and so on.

All the participants except C-Voks developed an off-line singing synthesis system (or text-to-chant) systems. In such systems, the score and lyrics are written in a text file, and the sound is computed off-line (and not in real time) according to this input data. C-Voks was the only performative singing synthesis system. For this system, the song is played in real-time, and after a number of trials, the best version is selected. More or less knowledge is used depending on the system, but all systems are based, like C-Voks, on recorded voice samples. The synthesis paradigm used for all the text-to-chant systems is concatenation of diphones or other voice segments.

An Absolute Category Rating paradigm measuring the Mean Opinion Score seemed appropriate for the evaluation task. The subjects were asked to rate the quality of the synthesized songs on a 5-point quality scale, with 1 being the lowest perceived quality and 5 the highest perceived quality. It was an Absolute Category Rating test measuring the Mean Opinion Score. The subjects were advised to listen over headphones and to use the full judgement scale for reporting their appreciation of the songs. They could listen to the 14 samples sounds as many time as they wished, and in the order they wanted. All the samples were presented on the screen. The order of presentation of the samples on the screen was randomized and different for each presentation, in order to avoid a possible visual presentation effect. An internet-based international listening test was advertised on relevant speech, singing and music mailing lists, and launched for 12 days between August 29th and September 9th 2016.

7.1.3. Instrument used in the challenge

The instrument used in the challenge was an early version of C-Voks called Calliphony. Like C-Voks, the Calliphony system was controlled with a stylus on a Wacom graphic tablet, using the preferred hand. Rhythm was controlled by pressing / releasing the control button using the non-preferred hand. The main difference between the current version of C-Voks and Calliphony is the vocoder used. Sound processing in Calliphony was performed with the help of a real-time PSOLA vocoder. Sound quality of the RT-PSOLA and WORLD vocoders are equivalent, but as WORLD offers additional spectral controls, it is preferred in the current version of C-Voks.

7.2. Results

7.2.1. Subjects

The listening test was launched worldwide and a grand total of 198 responses were received during the 12 days of test opening. Responses came from 18 different countries (France, Germany, Switzerland, United Kingdom, USA, Japan, Denmark, New Zealand, Spain, Austria, Belgium, Sweden, Canada, Poland, Australia, Brazil, Ireland, Marocco), with a noticeable bias towards Europa (with about 3/4 of responses) and France (with about 1/3 of responses). Among these 198 responses, only 80 complete responses, with scores for the 14 songs, were retained for further analysis. The other 119 responses were incomplete, probably just for curiosity, but they were not considered for analysis.

The total duration of the 14 sounds was 11 mn 30 s. On the 80 retained full test, only 22 responses took longer than the full stimuli duration. 56 responses took longer than 6 mn (half of the full sound stimuli duration). This means that subjects felt comfortable to take a decision on song quality before listening to the whole samples. The Number of subjects (Y-axis) having performed the test in less than a given time (X-axis, in mn) are plotted in Fig. 11. As a basis for analyses, we selected the 56 subjects that took 6 min or more. This is because the results obtained in ranking scores are the same for this group and the smaller set of 22 subjects that took more than 11 mn 30, although the scores are slightly different.

7.2.2. MOS, ranks and groups

The Mean Opinion Scores obtained for the 56 subjects are displayed in Fig. 12. The highest score is 4.21 MOS and the lowest 1.68. This indicates that the listeners used the whole scale for their judjements.

For further analysis of the Lab factor, a post-hoc Tukey's honestly significant difference (HSD) test was run on this factor. The analysis are reported in Table 2. The post-hoc test gives four statistically different groups A, B, C, D. The three Lab (L1, L2, L3, they are anonymized) in group D received statistically comparable scores. Both the MOS for all the songs for a same Lab and the best song for a given Lab are reported. Note that the same grouping result is obtained when all the 80 subjects are considered. The same ranking is obtained when taking the best Song in each Lab.

7.2.3. ANOVA

An ANOVA was run on the quality scores as a dependent variable, with the factors Song (song A or S), Voice (female or male voice) and Lab (the laboratory having produced the synthesis, 6 levels) as main factors, and the two way interactions between Lab * Voice and Lab * Song. Results of the analysis of variance are reported in Table 3. All main factors have a significant effect on the result. The interaction between Lab and Song is significant: this means that for some systems the difference in appreciation between the two songs is statistically different. Male voices received on average a significantly higher score (3.1) than female voices (2.4). The A song received significantly higher scores (3.0) than the S song (2.7). As expected, the factor Lab has the strongest effect size (cf. the η^2 column in Table 3). Most of the variance in the result is explained by the system that produced the song.

Table 1
Participants to the singing synthesis challenge fill-in the gap at Interspeech 2016.

# items	Lab	Lang	Songs	Voice	Style	Method
2	WBHSM	English	S A	Male	jazz	concatenative
4	ISIS	French	S A	Male, Female	jazz	concatenative
2	C-Voks	French	S A	Male	jazz	performative
4	ACAPELA	French	S A	Male, Female	jazz	concatenative
1	Seraphim	Mandarin	A	Female/male	pop	concatenative
1	Bersokantari	Basque	A	Male	traditional	concatenative

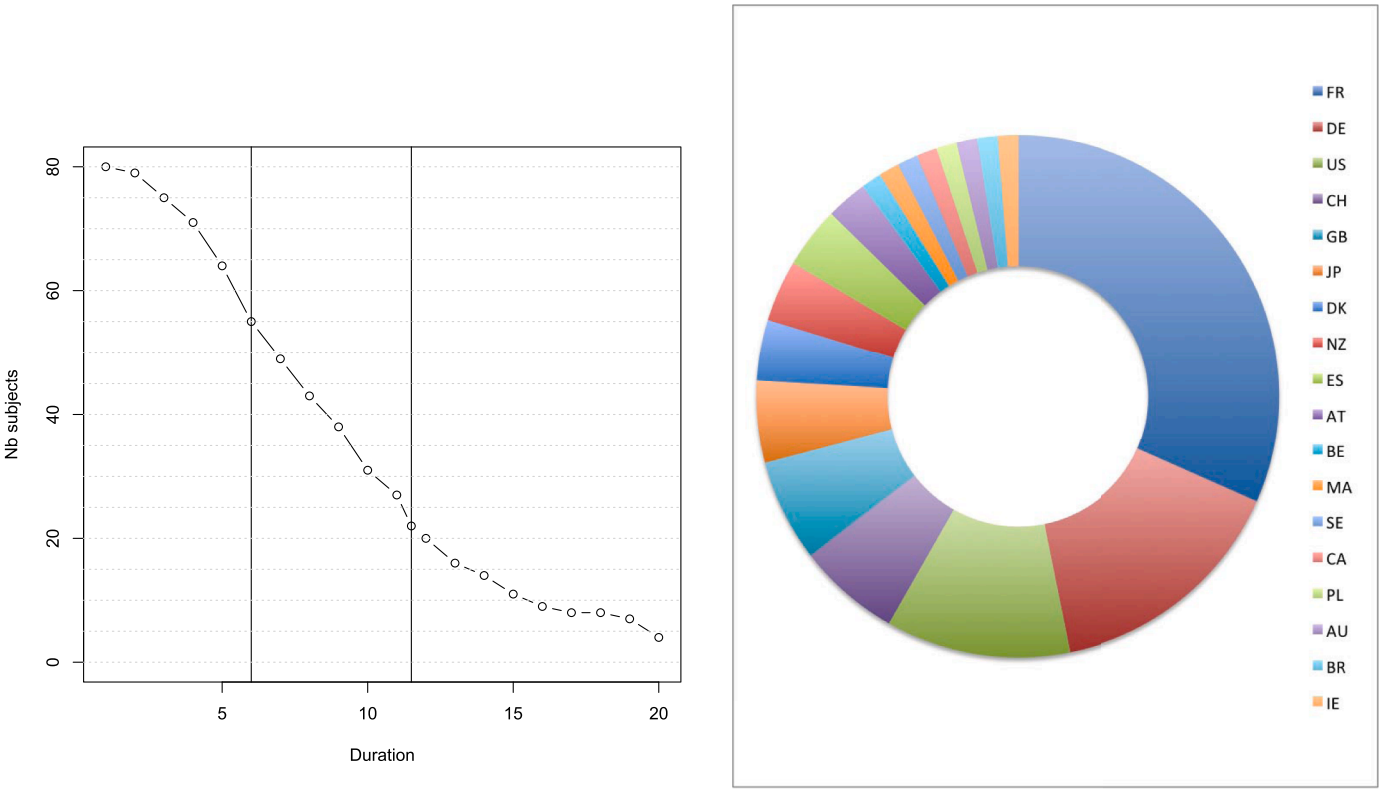


Fig. 11. Left panel: number of subjects (*y* axis) having performed the test in less than a given time (*x*-axis, in mn). The two vertical lines indicates the full song duration (11 mn 30 s) and more thanf halp song duration (6 mn). Right panel: countries of participant.

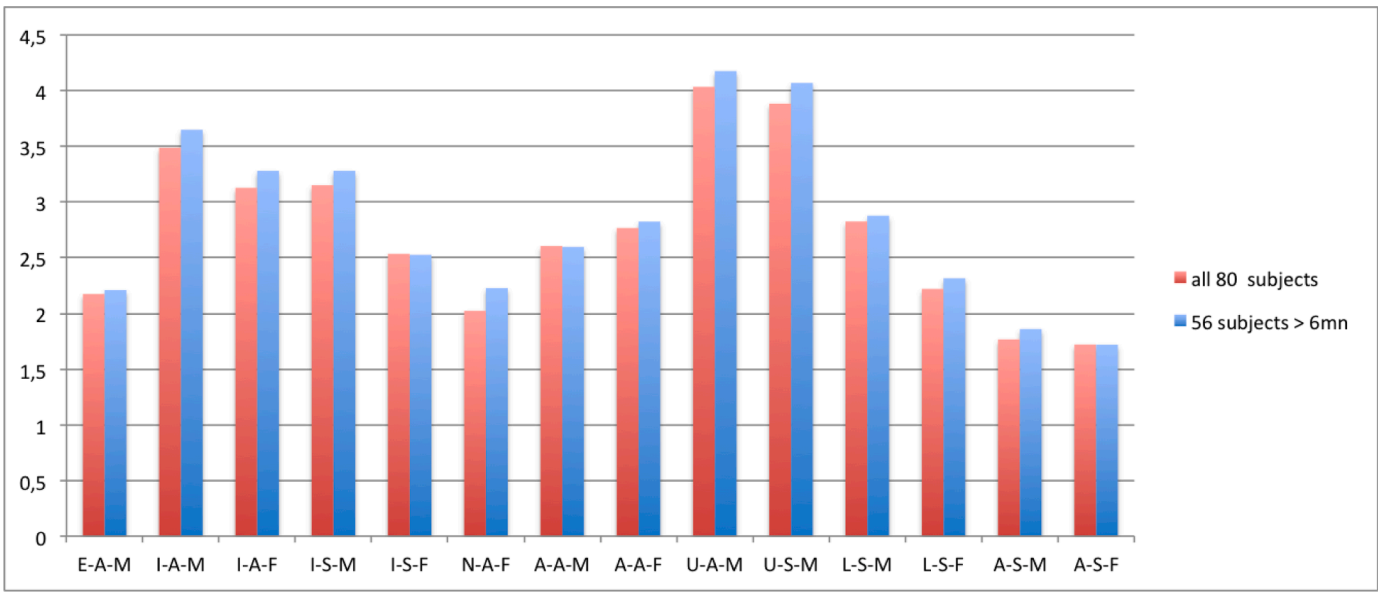


Fig. 12. Lab: I = ISIS E = L1 N = L2 L = C-Voks U = WBHSM A = L3, Song: A and S. Voice: M = Male F = Female.

7.3. Discussion

The top three systems of the Singing Synthesis Challenge Fill-in the Gap at Interspeech in 2016 were:

1. WBHSM Online, cited 2020-06-12 (2020).
2. ISIS Online, cited 2020-06-12 (2020)
3. C-Voks (audio examples 9 et 10, songs A and S with accompaniment)

C-Voks ranked third in the challenge, after two high-quality text-to-chant system. This demonstrates that the sound quality of performative (real-time) systems is lower than that of the best off-line concatenative systems. It obtained a MOS of 1.38 less than the best system. But it is higher than the three others off-line concatenative text-to-chant systems. Another comparative perceptual study, involving C-Voks (Calliphony) and ISIS, is reported in Feugère et al. (2016). Comparable results are obtained with the same ranking and a similar difference of about 1 point in the scale MOS between off-line concatenative

Table 2
Ranking and groups obtained by a post-hoc Tukey's HSD test.

Rank	Lab	Score	Group	Best song
1	WBHSM	4.15	A	4.21
2	ISIS	3.20	B	3.67
3	C-Voks	2.57	C	2.83
4	L1	2.25	D	2.70
5	L2	2.22	D	2.22
6	L3	2.20	D	2.20

Table 3
Effect of each factor on the dependent variable, as measured by an ANOVA.

	Sum Sq.	df	df error	F	p	η^2
Lab	341.1	5	758	73.3	<0.001	0.33
Gender	22.7	1	758	24.4	<0.001	0.03
Song	35.6	1	758	38.3	<0.001	0.05
Lab*Gender	3.1	2	758	1.7	0.19	0.00
Lab*Song	7.9	2	758	4.3	<0.001	0.01

text-to-chant and performative singing synthesis systems. As the most advanced research groups worldwide working on Singing Synthesis at this time took part in the challenge, it can be regarded as an accurate picture of current achievements.

8. Discussion and conclusion

8.1. New vocal expressions

Performative voice synthesis is a new paradigm in the already long history of artificial voices. The voice is played like an instrument, allowing for speaking or singing with the borrowed voice of another. In Voks, voice samples, produced by a true vocal apparatus, are played by free-hand theremin-controlled gestures, and by writing gestures on a graphic tablet. The same types of sounds, controlled with other gestures, give rise to yet other musical instruments and expressive possibilities. The relationship of embodiment between the singer's gestures and the vocal sound produced is broken. A voice is speaking or singing, with realism, expressivity and musicality, but it is not the musician's own voice, and a vocal apparatus does not control it. This introduces a special relationship between the synthetic voice and the player, the voice being at the same time embodied (by the player gestures playing the instrument with her/his body) and externalized (because the instrument is not her/his own voice). In some performances, two different voices can be sung and played by the same person (Video example 2).

Performative voice synthesis opens new expressive possibilities:

- Voice deconstruction sounding like computer music or electro-acoustic voice. Parametric representation and modeling of the voice allows for extreme variations. Specific features of the voice can be emphasized (formants, pitch, voice quality, vocal tract size, roughness), and a rich sonic material based on the voice can be worked out in real time.
- Voice imitation, on the contrary, favors proximity between natural and synthetic voice. How close to natural voice can a synthetic voice be ? In some situations, a realistic voice is desirable. It is at the (possibly interesting) risk of an "uncanny valley" effect.
- Voice extension in between deconstruction and imitation, the augmented voice is a realistic-sounding voice with augmented (naturally impossible) features: for instance, a voice with a very large register, a male/female voice, a very slow, very fast pronunciation, small and large vocal tracts. Another aspect of voice augmentation is the specific vocal gestures allowed by the control interfaces: here the theremin and graphic tablet.

These expressive features are useful for musical and poetic purposes, as well as for speech and voice communication applications (discussed below). Examples of performances are given in the next section (Table 4).

8.2. Performance examples in various languages

T-Voks and C-Voks have been used on stage for several musical and poetic performances (d'Alessandro et al., 2019; D'Alessandro et al., 2018). Four languages have been used so far: French, English, Mandarin and German (see Fig. 9 for a picture of a T-Voks and C-Voks performance).

8.2.1. Examples in French

Video example 1 is a musical example featuring the French song "La vie en rose" (Edith Piaf, Louiguy), played by T-Voks. Samples are from spoken utterances of a French female speaker with no musical background or voice training. Syllabic rhythm control works well for this example, as French is a syllable-timed language (Abercrombie, 1984). The theremin is able to replicate key features of Piaf's signature vocal style, including dramatic vibratos and small glissandi at the start and end of phrases.

Video example 2 is a poetic example featuring the fairy tale "Histoire de Brirouch", played by C-Voks (Vokinesis). In this example, the speed timing control mode is demonstrated. The stylus is only used for melodic, rhythmic and dynamic control. Samples are from spoken utterances of a professional French male singer.

8.2.2. Examples in English

Video example 3 is a musical example featuring the jazz standard "My Funny Valentine", popularized by Chet Baker. Samples are from spoken utterances of a non-professional female American English speaker, resampled by a factor of 1.22 to yield a male voice. A specific setting of vocal effort lends "breathiness" to the synthesized voice, inspired by Chet Baker's singing style.

Unlike French, English is a stress-timed language (Abercrombie, 1984). Syllable control requires paying more attention to stress timing and inter-syllable transitions. Controlling English, a language where diphtongs, i.e. vocalic changes inside a syllabic nucleus, are common, using the biphasic syllable control, was found challenging by performers.

8.2.3. Examples in German

Between speech and singing, video example 4 demonstrate an example of *Sprechstimme* in German. *Sprechstimme* is a vocal technique where singing imitates the continuous pitch contours of speech. In the score of his *Pierrot Lunaire* (Schoenberg, 1912), Arnold Schoenberg gives the following direction for achieving *Sprechstimme*: to "be well aware of the difference between speaking tone and singing tone", by singing the written pitches, but altering them right after, all the while singing the rhythm as written.

Samples are from an excerpt of *Pierrot Lunaire* recorded by a native

Table 4
Accompanying audio and video examples.

Video example 1	excerpt from La Vie en rose, Edith Piaf & Louiguy (T-Voks)
Video example 2	excerpt from Brirouch, video performance (C-Voks)
Video example 3	excerpt from My Funny Valentine, Richard Rodgers and Lorenz Hart (T-Voks)
Video example 4	excerpt from Pierrot Lunaire, Arnold Schönberg (T-Voks)
Video example 5	Chun Xiao, Meng Haoran (T-Voks)
Video example 6	excerpt from All Shall Be Well, Christophe d'Alessandro & Julian of Norwich (T-Voks and C-Voks)
Audio example 1 (still video)	Singing Synthesis Challenge Song A (C-Voks)
Audio example 2 (still video)	Singing Synthesis Challenge Song S (C-Voks)

French male speaker, resampled by a factor of 0.8 to yield a female voice. As a stress-timed language, but with a strong syllabic structure, German shares the same considerations for syllable advancement as English, but the smaller distinction between stressed and unstressed syllables (Wagner, 2008) makes it closer to French. For a convincing *Sprechstimme*, pitch slides and their volume curves must also correspond to the correct stress pattern. Pitch slides are achieved with T-Voks by small displacements of the fingers or by pivoting around the wrist.

8.2.4. Examples in Mandarin

Video example 5 is a poetic example in Mandarin. Mandarin Chinese is a tonal language, where the same syllable pronounced with different frequency contours changes in meaning. Each syllable can be pronounced with one of four tones, which is carried by the syllabic rhyme (Hallé, 1994). Classical poetry is typically recited with exaggerated tone enunciation.

A well-known Tang dynasty short poem was recorded by a Mandarin speaker pronouncing each syllable in monotone. The poem was then "recited" using T-Voks, with each tone shaped entirely by the theremin. Syllabic rhythm control is used.

Tones were created mostly with the preferred hand, with the non-preferred hand creating a gradual fade in and fade out. The pitch hand rests in place for tone 1, whose pitch stays steady. Other tones, whose pitch changes in different ways, were produced using fluid wave-like gestures of the entire hand, pivoting at the wrist. These hand sweeps are larger than those required by *Pierrot Lunaire*, with the forearm remaining largely stationary.

8.3. Future work

Three points must be mentioned to conclude this article. The first point is a practical one. Data preparation is a time-consuming preliminary task for playing with Voks. Possible solutions are discussed for faster preparation of the sound and linguistic data. The second point concerns forthcoming application of Voks for education and reeducation. Finally, the general question of performative voice synthesis is discussed.

8.3.1. Sample production and text-to-speech synthesis

The task of labelling a speech sample is currently a manual and tedious one: one must input the location of each one of the control points by hand. An automated procedure for labelling samples, based on an automated phoneme segmentation and rules to convert such a segmentation into a labelling, is a possible solution, although it is prone to errors. Another option would be to make the recorded speaker generate the labelling during the time of the recording, by using a gesture similar to the one used by the performer, such as the pressing of a button simultaneous with the uttered syllables.

Input samples can be generated by text-to-speech systems. Those systems aim to emulate ordinary speaking voices, which differs from samples recorded by humans specifically for Voks, in which speakers make an effort to articulate and detach syllables. Thus coarticulation in synthesis based on text-to-speech-generated samples features altered phonemes. Generating samples with the help of text-to-speech systems also has some benefits: it eliminates the need to record a sample prior to the performance, and it makes automatic segmentation easier, which is an important step of automatic biphasic labelling.

8.3.2. Applications to education and reeducation

In a forthcoming project, the use of manual gestures, mediated by new Human Machine Interfaces like Voks, will be investigated for designing innovative tools and methods for intonation education (training) and re-education (re-training). The control of a synthesized voice through hand gestures is a new research paradigm in the field of human-machine interaction. Previous studies have demonstrated that chironomic intonation using handwriting gestures on a graphical tablet can be even more precise and accurate than the natural voice in imitation tasks (d'Alessandro et al., 2011, 2014). The high performance in chironomy for performative voice synthesis can be attributed to its intrinsic multimodal integration (vision, kineaesthesis and audition (Perrotin and d'Alessandro, 2016), as well as to the existing dexterity of the handwriting movements (as used for writing and drawing purposes), which were repurposed for a new task.

It appears that performative voice synthesis could also foster new important applications in language acquisition and vocal substitution. A first foreseen application is to develop an educational program based on chironomy and to test it in language classes. The second foreseen application is to develop tools based on chironomy for vocal impairment assistance. In the case of phonatory function impairment, gestural control can improve expressive intonation in an augmented reality paradigm: phonation is controlled or enhanced by chironomy and articulation is controlled by the true vocal tract. An extreme case is that of vocal substitution. In the case of laryngectomy inducing a voice loss, the gestural control of intonation must enable the restoration of both linguistic and expressive intonation.

8.3.3. Next steps in performative voice synthesis

Voice instruments, or performative voice synthesizers, are still facing a compromise between sound quality and free selection of sound material. Some systems allow (Fels and Hinton, 1998) for free sound material (i.e. any sequence of speech can be produced, like in text-to-speech systems), but with poor quality. Other systems, like Voks, deliver high quality sound, but are limited to pre-recorded sound material (or pre-synthesized sound material).

Voks only allows for linear resequencing of prepared samples. The main difficulty for free, real-time speech control is the large combinatorial complexity of the possible syllables and the difficulty to specify or select them in real-time. In text-to-speech systems, the linguistic material is presented as a text: either typed on a keyboard or copied from a file. In Cantor Digitalis, the speech material is free, but limited to vowels. Selecting full text (sequences of vowels and consonants) on the fly, i.e. free text selection, for a performative voice synthesizer, has no straightforward solution to date. A possible strategy would be to present the perfomer with a number of possible subsequent syllables, computed in real-time based on the previous ones and the statistic distribution of syllables in the considered language.

CRediT authorship contribution statement

Grégoire Locqueville: Investigation, Software, Validation, Writing - original draft. **Christophe d'Alessandro:** Investigation, Supervision, Validation, Writing - original draft. **Samuel Delalez:** Investigation, Software, Validation. **Boris Doval:** Investigation, Validation. **Xiao Xiao:** Investigation, Validation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Part of this work has been done in the framework of the SMAC (FEDER IF0011085) project funded by the European Union and the Conseil Génér Région Île de France, and the Agence Nationale de le Recherche ChaNTeR and GEPETO Projects (ANR-13-CORD-0011, 2014–2017, ANR-19-CE28-0018, 2019-2023). The authors are indebted to Dr. Albert Rilliard for statistical analyses of the evaluation test.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.specom.2020.10.002

References

Abercrombie, D., 1984. Elements of General Phonetics. Edinburgh University Press.
Ardaillon, L., Chabot-Canet, C., Roebel, A., 2016. Expressive control of singing voice synthesis using musical contexts and a parametric f0 model. Proceedings of Interspeech 2016. ISCA, pp. 1250–1254.
Astrinaki, M., 2014. Performative statistical parametric speech synthesis applied to interactive designs. PhD dissertation. University of Mons.
Astrinaki, M., d'Alessandro, N., Dutoit, T., 2012. MAGE-A platform for tangible speech synthesis. Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2012. University of Michigan, Ann Arbor, USA, pp. 353–356.
Barbosa, P., Bailly, G., 1994. Characterisation of rhythmic patterns for text-to-speech synthesis. Speech Commun. 15 (1), 127–137.
Berndtsson, G., 1996. The KTH rule system for singing synthesis. Comput. Music J. 20 (1), 76–91.
Blaauw, M., Bonada, J., 2017. A neural parametric singing synthesizer. Proceedings of Interspeech 2017. ISCA, pp. 4001–4005.
Bonada, J., Umbert, M., Blaauw, M., 2016. Expressive singing synthesis based on unit selection for the singing synthesis challenge 2016. Proceedings of Interspeech 2016. ISCA, pp. 1230–1234.
Chan, P.Y., Dong, M., Ho, G.X.H., Li, H., 2016. Seraphim: a wavetable synthesis system with 3D lip animation for real-time speech and singing applications on mobile platforms. Proceedings of Interspeech 2016. ISCA, pp. 1225–1229.
Cook, P.R., 1993. Spasm, a real-time vocal tract physical model controller; and singer, the companion software synthesis system. Comput. Music J. 17 (1), 30–44.
Cotescu, M., 2016. Optimal unit stitching in a unit selection singing synthesis system. Proceedings of Interspeech 2016. ISCA, pp. 1255–1259.
d'Alessandro, N., d'Alessandro, C., Le Beux, S., Doval, B., 2006. Real-time calm synthesizer: new approaches in hands-controlled voice synthesis. Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2006. IRCAM, Paris, France, pp. 266–271.
d'Alessandro, C., d'Alessandro, N., Le Beux, S., Simko, J., Çetin, F., Pirker, H., 2005. The Speech Conductor: Gestural Control of Speech Synthesis. Technical Report. Final Project Report #6. eNTERFACE'05. Mons, Belgium.
d'Alessandro, C., Doval, B., 1998. Voice quality modification using periodic-aperiodic decomposition and spectral processing of the voice source signal. Proceedings of the 3rd ESCA International Workshop on Speech Synthesis. Jenolan Caves, Australia, pp. 277–282.
d'Alessandro, C., Xiao, X., Locqueville, G., Doval, B., 2019. Borrowed voices. Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2019. UFRGS, Porto Alegre, Brazil, pp. 2.2–2.4.
d'Alessandro, C., Doval, B., Delalez, S., Victor, W., Expert, R., 2018. Jouer avec les doubles artificiels de la voix: Cantor digitalis et Vokinesis. Conférence-concert. La voix à double tranchant. Solipsy, pp. 185–203.URL https://hal.archives-ouvertes.fr/hal-02009009
d'Alessandro, C., 2006. Voice source parameters and prosodic analysis. In: Sudhoff, S., et al. (Eds.), Method in Empirical Prosody Research. Walter de Gruyter, Berlin, New York, pp. 63–87.
d'Alessandro, N., Dutoit, T., 2007. Handsketch bi-manual controller: investigation on expressive control issues of an augmented tablet. Proceedings of the International Conference on New Interfaces for Musical Expression. NYU, New York, USA, pp. 78–81.
d'Alessandro, C., Feugere, L., Le Beux, S., Perrotin, O., Rilliard, A., 2014. Drawing melodies: evaluation of chironomic singing synthesis. J. Acoust. Soc. Am. 135 (6), 3601–3612.
d'Alessandro, C., Rilliard, A., Le Beux, S., 2011. Chironomic stylization of intonation. J. Acoust. Soc. Am. 129 (3), 1594–1604.
d'Alessandro, N., Woodruff, P., Fabre, Y., Dutoit, T., Le Beux, S., Doval, B., d'Alessandro, C., 2007. Real time and accurate musical control of expression in singing synthesis. J. Multimodal User Interfaces 1 (1), 31–39.
del Blanco, E., Hernaez, I., Navas, E., Sarasola, X., Erro, D., 2016. Bertsokantari: a TTS based singing synthesis system. Proceedings of Interspeech 2016. ISCA, pp. 1240–1244.
Delalez, S., 2017. Vokinesis: an instrument for suprasegmental control of voice synthesis. PhD dissertation. Université Paris-Saclay, Orsay, France.
Delalez, S., d'Alessandro, C., 2017. Adjusting the frame: biphasic performative control of speech rhythm. Proceedings of Interspeech 2017. Stockholm, Sweden. ISCA, pp. 864–868.
Delalez, S., d'Alessandro, C., 2017. Vokinesis: syllabic control points for performative singing synthesis. Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2017. Aalborg University, Copenhagen, Denmark, pp. 198–203.
Doval, B., d'Alessandro, C., Henrich Bernardoni, N., 2006. The spectrum of glottal flow models. Acta Acust. United Acust. 92, 1026–1046.
Fant, G., 1966. A note on vocal tract size factors and non-uniform f-pattern scalings. STL-QPSR 7 (4), 22–30.
Fant, G., 1970. Acoustic Theory of Speech Production. Mouton, The Hague, Nederlands.
Fels, S.S., Hinton, G.E., 1993. Glove-talk: a neural network interface between a data-glove and a speech synthesizer. EEE Trans. Neural Netw. I 4 (1), 2–8.
Fels, S.S., Hinton, G.E., 1998. Glove-talk II: a neural-network interface which maps gestures to parallel formant speech synthesizer controls. IEEE Trans. Neural Netw. 9 (1), 205–212. https://doi.org/10.1109/72.655042.
Feugère, L., d'Alessandro, C., Delalez, S., Ardaillon, L., Roebel, A., 2016. Evaluation of singing synthesis: methodology and case study with concatenative and performative systems. Proceedings of Interspeech 2016. ISCA, pp. 1245–1249.
Feugère, L., d'Alessandro, C., Doval, B., Perrotin, O., 2017. Cantor digitalis: chironomic parametric synthesis of singing. EURASIP J. Audio Speech Music Process. 2017 (1), 2. https://doi.org/10.1186/s13636-016-0098-5.
Haken, L., Abdullah, R., Smart, M., 1992. The continuum: a continuous music keyboard. Proceedings of the International Computer Music Conference. International Computer Music Association, p. 81.
Hallé, P., 1994. Evidence for tone-specific activity of the sternohyoid muscle in modern standard chinese. Lang. Speech 73, 103–124–1043. https://doi.org/10.1121/1.1531176.
Kawahara, H., Morise, M., 2011. Technical foundations of tandem-straight, a speech analysis, modification and synthesis framework. Sadhana 36 (5), 713–727. https://doi.org/10.1007/s12046-011-0043-3.
Kenmochi, H., Ohshita, H., 2007. Vocaloid-commercial singing synthesizer based on sample concatenation. Proceedings of Interspeech 2007. ISCA, pp. 4009–4010.
Lamb, R., Robertson, A., 2011. Seaboard: a new piano keyboard-related interface combining discrete and continuous control. Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2011. University of Oslo, Oslo, Norway, pp. 503–506.
Le Beux, S., d'Alessandro, C., Rilliard, A., 2010. Calliphony: a tool for real-time gestural modification and analysis of intonation and Rythm. Proceedings of the International Conference on Speech Prosody, SP 2010. ISCA, Chicago, USA, 101.
Le Beux, S., Doval, B., d'Alessandro, C., 2010. Issues and solutions related to real-time TD-PSOLA implementation. Audio Engineering Society Convention 128. Audio Engineering Society, pp. 1–6.
MacNeilage, P.F., 1998. The frame/content theory of evolution of speech production. Behav. Brain Sci. 21 (4), 499–511. https://doi.org/10.1017/S0140525X98001265.
Morise, M., 2015. Cheaptrick, a spectral envelope estimator for high-quality speech synthesis. Speech Commun. 67, 1–7. https://doi.org/10.1016/j.specom.2014.09.003.
Morise, M., 2016. D4c, a band-aperiodicity estimator for high-quality speech synthesis. Speech Commun. 84, 57–65. https://doi.org/10.1016/j.specom.2016.09.001.
Morise, M., 2017. Harvest: a high-performance fundamental frequency estimator from speech signals. Proceedings of Interspeech 2017. ISCA, pp. 2321–2325.
Morise, M., Watanabe, Y., 2018. Sound quality comparison among high-quality vocoders by using re-synthesized speech. Acoust. Sci. Technol. 39 (3), 263–265. https://doi.org/10.1250/ast.39.263.
Morise, M., Yokomori, F., Ozawa, K., 2016. World: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans. 99-D, 1877–1884.
Zbyszynski, M., Wright, M., Momeni, A., Cullen, D., 2007. Ten years of tablet musical interfaces at CNMAT. Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2007. NYU, New York, USA, pp. 100–105.
online, cited 2020-06-12,2020, https://www.isca-speech.org/archive/Interspeech_2016/booklet.pdf.
online, cited 2020-06-12, 2020https://chanter.lam.jussieu.fr/doku.php?id=winner_of_the_singing_synthesis_challenge_2016:start.
online, cited 2020-06-12, 2020https://chanter.lam.jussieu.fr/doku.php?id=evaluations:start.
Perrotin, O., 2015. Singing with hands: chironomic interfaces for digital musical instruments. PhD dissertation. Université Paris Sud - Paris XI, Orsay, France.
Perrotin, O., d'Alessandro, C., 2016. Vocal effort modification for singing synthesis. Proceedings of Interspeech 2016. ISCA, pp. 1235–1239.
Perrotin, O., d'Alessandro, C., 2016. Target acquisition vs. expressive motion: dynamic pitch warping for intonation correction. ACM Trans. Comput. Hum. Interact. 23 (3), 17.
Puckette, M., 2002. Max at seventeen. Comput. Music J. 26 (4), 31–43.
Schoenberg, A., 1912. Pierrot Lunaire for speaker, piano, flute (piccolo),clarinet (bass clarinet), violin (viola). and cello - op 21.
Synthesis of singing challenge, 2007. Synthesis of singing challenge, special session at interspeech. Proceedings of Interspeech 2007. ISCA.
Session synthesis of singing, 1993. Proceedings of the Stockholm Music Acoustics Conference (SMAC 1993), pp. 279–294.
Theremin world: Learn to play the theremin. 2019http://www.thereminworld.com/Learn-to-Play. Accessed: 2019-01-25.
The MIDI Manufacturers Association MIDI Polyphonic Expression, Los Angeles, CA 1.0 edition], 2018.
Umbert, M., Bonada, J., Goto, M., Nakano, T., Sundberg, J., 2015. Expression control in singing voice synthesis: features, approaches, evaluation, and challenges. IEEE Signal Process. Mag. 32 (6), 55–73.
Wagner, P., 2008. The rhythm of language and speech: Constraining factors, models, metrics and applications. Habilitationsschrift, University of Bonn, Bonn, Germany.
Xiao, X., Locqueville, G., d'Alessandro, C., Doval, B., 2019. T-Voks: the singing and speaking theremin. In: Queiroz, M., Sedó, A.X. (Eds.), Proceedings of the International Conference on New Interfaces for Musical Expression, NIME 2019. UFRGS, Porto Alegre, Brazil, pp. 110–115.

Prosodic disambiguation using chironomic stylization of intonation for native and non-native speakers

Xiao Xiao[1], *Nicolas Audibert*[1], *Grégoire Locqueville*[2], *Christophe d'Alessandro*[2], *Barbara Kuhnert*[1], *Claire Pillot-Loiseau*[1]

[1]LPP, Sorbonne Nouvelle
[2]LAM, Sorbonne Université

[xiao.xiao, nicolas.audibert, barbara.kuhnert, claire.pillot]@sorbonne-nouvelle.fr,
[gregoire.locqueville, christophe.dalessandro]@sorbonne-universite.fr

Abstract

This paper introduces an interface that enables the real-time gestural control of intonation in phrases produced by a vocal synthesizer. Melodies and timing of a target phrase can be modified by tracing melodic contours on the touch-screen of a mobile tablet. Envisioning this interface as a means for non-native speakers to practice the intonation of a foreign language, we present an experiment where native and non-native speakers imitated the pronunciation of French phrases using their voice and the interface, with a visual guide and without. Comparison of resulting F0 curves against the reference contour and perceptual assessment of synthesized utterances suggest that for both non-native and native speakers, imitation with the help of a visual guide is comparable in accuracy to vocal imitation, and that timing control was a source of difficulty.

Index Terms: human-computer interaction, intonation, second-language acquisition

1. Introduction

Our research explores how chironomic stylization can be used by non-native speakers for intonation practice of a foreign language. "Chironomic stylization" means here vocal synthesis using real-time hand gesture control of stylized intonation patterns. Such multimodal practice addresses three sources of difficulty for intonation learning. First, it can train the ear to perceive unfamiliar features in speech by presenting them through visual and kinesthetic modalities. Second, control of pronunciation with hand gestures (chironomy) bypasses ingrained patterns in the natural voice that are difficult to correct [1]. Finally, vocal synthesis enables a learner to focus on the suprasegmental level without being preoccupied by fine-phonetic detail on the segmental level. The main hypothesis is that the multimodal approach provided by chironomy (kinestetic, visual and auditory) can reinforce the sensory experience of the learner and help in the learning process (i.e. grasping and memorizing intonation features).

An imitation paradigm for prosodic disambiguation has been chosen for assessing the ability of both native and non-native speakers to perceive, control and modify linguistically meaningful intonation patterns. This paradigm proved useful for studying intonation in language acquisition tasks [2, 3, 4]. Prior work on chironomic intonation of French phrases with native speakers and yielded similar results for vocal and gestural imitation [1]. In other words, chironomy can be a substitute for the human voice. Our work seeks to assess the feasibility of chironomy as vocal substitution for non-native speakers. The previous study, using a graphic tablet and stylus, allowed only the modulation of melody, with no change of rhythmic parameters.

Our work examines the simultaneous control of speech rhythm and melody. To this end, a mobile interface was developed that enables the control of both melody and timing of the synthesized pronunciation (Section 2). A performance and and perception study (with native and non-native speakers of French) was conducted using a corpus of ambiguous French phrases (with identical phonemic content that change meaning according to intonation) (Section 3). The study addresses the following questions: 1/ How does chironomic intonation compare with vocal intonation for phrase disambiguation ? 2/ How much do visually guided and non-guided chironomic intonations differ? 3/ To what extent does the additional control of timing add to the difficulty of the task? 4/ Do non-native and native speakers differ in their performance in different modalities?

2. Chironomic Control Interface

2.1. Performative Synthesis Architecture

The architecture is based on Voks [5], a high-quality performative vocal synthesizer that enables the real-time melodic and rhythmic control of previously recorded or Text-to-Speech speech samples, through the use of hand gestures. It is a MaxMSP application based on the WORLD vocoder [6, 7] initially developed with singing synthesis applications in mind, using a stylus on a graphic tablet or a Theremin [8, 9]. Given the widespread availability and popularity of mobile devices and their apps, we built a custom mobile interface for Voks, controlled by finger motions (instead of a stylus). It runs on a Mac OS X computer, which also runs a Node.js server that allows external devices to control Voks wirelessly. This server-client architecture was created to simulate the user experience of a mobile app, enabling proof-of-concept testing without the need to implement the synthesis software on a new platform.

When connected to the same WiFi network as the host computer, a mobile tablet or phone (herein a 9.7in Samsung Galaxy S2 tablet) can open the Gepeto interface in a Chrome browser. Communication between the interface and server uses the Websockets protocol. Messages for Voks are first sent to the server, where they can be saved, and then routed to Max via Open Sound Control (OSC). One way latency between mobile device and computer averages around 10ms. Subjects can control intonation for recorded sentences using the tip of their fingers on the touch-screen of a mobile device.

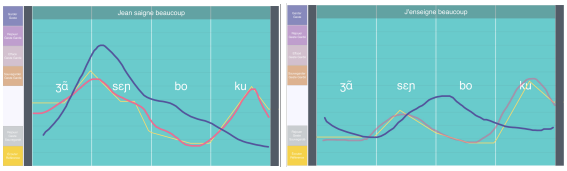


Figure 1: *Screenshots of the Gepeto interface showing a pair of phrases from the corpus The hot pink and dark purple lines are gesture traces from the user. The pink fades away after 1.5 second while the purple remains until it is erased and can be played back. The yellow line is the visual guide, a stylized version of the reference phrase's F0 curve.*

Id #	Phrase #	Phrase #bis
2	Tu parais très soucieux.	Tu paraîtrais soucieux.
	You seem very worried.	*You would seem worried.*
7	Jean lève son verre.	J'enlève son verre.
	Jean lifts his glass.	*I lift his glass.*
8	Jean saigne beaucoup.	J'enseigne beaucoup.
	Jean is bleeding a lot.	*I teach a lot.*
9	Jean porte un journal.	J'emporte un journal.
	Jean carries a newspaper.	*I carry a newspaper.*
10	Jean cadre la photo.	J'encadre la photo.
	Jean frames a photo.	*I frame a photo.*
21	C'est la morsure.	C'est la mort sûre.
	It's the bite.	*It's death for sure.*

Table 1: *Corpus of phrases*

2.2. Finger Intonation Control

The phrase to be controlled is shown at the top of the Gepeto interface. Below the target phrase is the control region, where tracing one's finger outputs a resynthesis of the phrase from Voks. The horizontal axis determines the temporal position in the original sample to resynthesize. It is divided based on syllable segmentation specified by a Praat TextGrid file [10]. Syllables are indicated with the International Phonetic Alphabet (IPA). For the present study, all syllables appeared with equal width in the interface, as French is often said to be "isochronic" [11]. Different rhythms can be realized by changing the speed of the finger's movement across the control surface (the "scrub" mode in Voks [5]). The vertical axis determines the output frequency regularly spaced on a semitones (ST) scale, with a range of 24ST (2 octaves) calibrated around the study corpus (116-466Hz). A higher vertical position results in a higher output sound.

A visual guide was made for each target phrase, which shows the stylized intonation curve of the reference recording generated by Prosogram [12]. Based on a perceptual model by [13], Prosogram simplifies a recording's pitch curve into straight line segments. Speech resynthesized with these stylized pitch curves are perceptually identical to the original stimuli. A stylized visual guide was chosen for ease of tracing and to avoid distraction from micro-prosodic vocal artifacts.

By default, a gesture in the control region leaves a hot pink trace that fades away after 1.5 seconds. The top button in the buttons panel toggles between the default "fade mode" and "held mode" where gestures stay on the screen until erased. In held mode, gesture traces appear in dark purple. When in held mode, the next three buttons are activated. One replays the gesture, with each point highlighted as it is resent to Voks. The next erases the gesture, and the last saves the gesture to the server. The button on the bottom left corner triggers the reference audio to play. When the visual guide is on, a cursor moves along the curve to indicate playback position.

3. Prosodic Disambiguation Experiment

3.1. Corpus, Subjects and Task

Six pairs of lexically ambiguous French phrases were selected from a larger corpus. Each pair shares the same sequence of phonemes and syllabic segmentation, but the two phrases have different meanings depending on their rhythm and intonation, with roughly equal plausibility of both meanings. Reference recordings featured a female native francophone speaker reading each as a declarative utterance.

Ten subjects took part in the experiment (2 male, 8 female, aged 20-48, mean age 32.7). Five were non-native speakers with different L1 backgrounds: Cantonese (S1), Portuguese (S2, S3), Mandarin (S6), Slovenian (S10). All have completed a semester-long course on French pronunciation and can be considered advanced. The 5 native speakers were undergraduate students in speech therapy. One non-native and three native subjects have musical experience (6-16 years).

Subjects first recorded themselves reading the phrases based on their own interpretation of the phrase. Next, subjects used the Gepeto interface to imitate the reference recording of each phrase to the best of their ability. Initially, subjects were asked to find a gesture for the phrase without any visual guidance. After the first gesture is submitted, the stylized pitch curve for the reference phrase appeared, and subjects were given another chance to find a gesture. Two familiarization trials were given for the gestural imitation task. Finally, subjects recorded vocal imitations of the reference phrases.

Phrases appeared in random order in all parts of the study, with paired phrases next to each other. For the imitation sections, no limits were imposed on the number of times references are played and the amount of time subjects spent finding the pronunciation of each phrase. The entire study, including verbal instructions and the subject information survey took between 1 and 1.5 hours. The study took place in a sound isolated studio. All audio was heard through monitor headphones, and voice recordings were made with an AKG C414 XLS microphone connected via an audio interface to a Macbook Air laptop computer. An external monitor displayed the current phrase and an interface for the audio recording sections

3.2. Intonation Contours comparison

The study collected 4 pronunciations per phrase with different modalities: vocal reading, non-guided gestural imitation, guided gestural imitation, and vocal imitation. Intonation contour distances were computed for each utterance.

3.2.1. F0 analysis and Intonation Contour Determination

To extract the F0 of subjects' vocal recordings, pitch analysis was performed with Praat and then manually verified. The starting and ending timestamp of the utterance within each recording were labeled in a Praat TextGrid. To compare with the results of [1], vocal imitations were fist aligned with the reference recordings using Dynamic Time Warping, which removes differences in timing. F0 values are sampled and compared at 10-millisecond intervals in the reference.

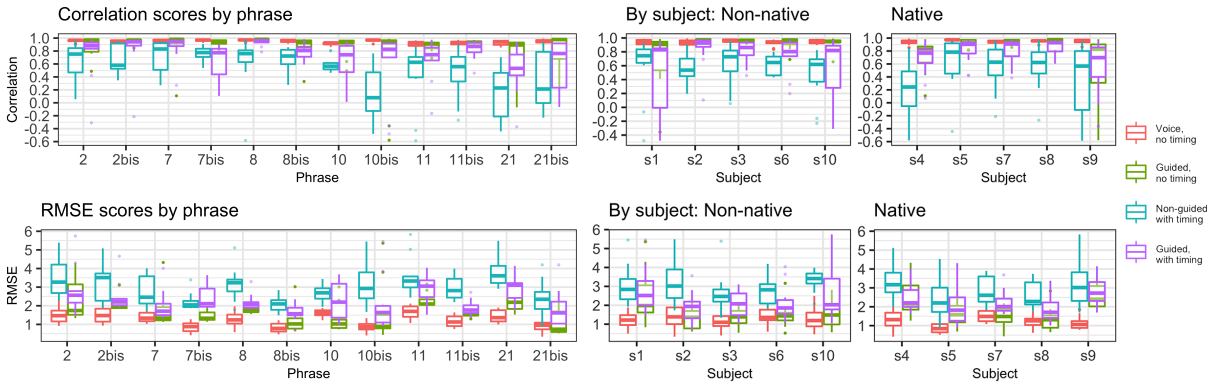


Figure 2: *Boxplot of scores used in statistical analysis. Horizontal line indicates median, hinges indicate 25th and 75th percentiles.*

Gestural imitation utterances were encoded as a series of data points. Each point is based on a 2D position touched by the subject in the control region of the Gepeto interface and includes the following information:

- f: the frequency in semitones relatives to the lowest frequency in the Gepeto interface
- *scrub*: the point in the original recording where phonemic information is taken, specified from 0 to 1
- *t-start*, *t-end*: the start and end time of the current point, relative to the start time of the gesture

3.2.2. *Intonation Contours Distances*

F0 contours were compared using two distance measures [14, 1]: correlation between contours and weighted Root Mean Square Error (RMSE), i.e. differences between contours. For both measures, the mean of each curve was subtracted from the F0 contours to normalize global register differences between individual voices.

Using the wCorr R package, Pearson's correlation was calculated between the reference and an imitation F0 contour, weighed by the intensity of the original contour to give more importance to phonemes with a higher sound level [15, 16]. Correlation measures similarity between two curves, and is 1 for identical curves. The mltools package [17] was used to compute the RMSE, which is 0 for identical contours and increases for divergent curves. Vocal contours were compared with the original F0 of the reference while gestural contours were compared with Prosogram stylizations of the reference curves. Between the stylized and original contours, mean correlation is 0.94 (SD: 0.04) and mean RMSE is 1.11 semitones (SD: 0.32 ST). For stylized contours across phrase pairs, mean correlation is 0.51 (SD: 0.38) and mean RMSE is 4.47 ST (SD: 0.91 ST).

Two sets of correlation and RMSE scores were computed for each gestural imitation, with F0 values sampled and compared at 10-millisecond intervals in the reference. For both, the region of interest in the reference file is determined by the first and last scrub values in the gesture. One set retains the original timing of the gesture and linearly scales the t-start values to match the length of the reference, interpolating when necessary. A second set of scores aligns the gesture by linearly scaling the gesture's scrub values in the region of interest, reflecting only how closely a subject followed the stylized reference F0 curve and do not take into account distortions in timing (e.g. if a subject moved too slowly when tracing one section of the gesture).

As neither correlation nor RMSE follow a Gaussian distribution, the Fisher Z and log transforms are used respectively for correlation and RMSE scores to obtain Gaussian distributions for statistical analysis.

3.3. Statistical Modeling of Comparison Scores

Multilevel models were fitted for further analysis, focusing on two types of comparisons. A first set of models were fitted for vocal imitation and guided gestural imitation scores without timing. Only guided scores were used because they represent the "best attempts" of gestural imitation. A second set of models were fitted for guided and non-guided gestural imitation scores with subjects' timing included, to assess the effect of the visual guide. Difference in intercepts for guided imitation scores between the no-timing and with-timing models represents the effect of timing control on the gestural imitation task. Figure 2 shows aggregate plots of scores used in the models.

All models were given random intercepts for phrases and random slopes for subjects based on condition. Frequentist multilevel models failed to converge when random slopes were included [18]. To prevent type 1 errors from removing the random slopes [19], we turned to Bayesian multilevel models using BRMS and Stan [20, 21, 22]. Weakly informative, "regularizing," priors based on [23, 24, 25] were used (intercepts (α) & slopes (β)- Normal(0, 10); σ_e & σ_{group}- HalfCauchy(0,10), Correlation Parameter - LKJ(2)) All models used two chains with 10000 iterations (2000 warm-up).

For each data subset and score type, four models were made with different fixed effects: on condition only, on subject nativeness only, on both without correlation, and on both with correlation. Condition and nativeness were contrast coded in each model (0.5 and -0.5). Models with the same dataset and score type were compared with LOO (Leave One Out).

3.4. Analysis of Results

All LOO comparisons yielded standard error (SE) differences of less than 3, indicating that no best model stands out. Table 2 shows the results of the full model, given by the following formula in lme4 notation [18]: $Score \sim Condition * Native + (1 + Condition|Subject) + (1|Phrase)$, applied to each data subset and score type.

For the models using scores without timing, the posterior mean correlation of vocal and guided gestural imitation is 0.95 (95% CrI=[0.92, 0.97]), and the mean RMSE score is 1.35 st

	Condition: 0.5 vocal, -0.5 gestural						Condition: 0.5 guided, -0.5 non-guided					
	$f = z(correlation))$			$f = log(RMSE)$			$f = z(correlation)$			$f = log(RMSE)$		
	Mean	Lower	Upper	Mean	Lower	Upper	Mean	Lower	Upper	Mean	Lower	Upper
Condition, β_1	0.18	-0.16	0.52	**-0.29**	**-0.57**	**-0.02**	**0.45**	**0.24**	**0.65**	**-0.28**	**-0.39**	**-0.17**
Native, β_2	-0.09	-0.44	0.25	0.06	-0.17	0.29	-0.03	-0.38	0.32	-0.03	-0.24	0.19
Cond.:Native, β_3	0.33	-0.35	1.01	-0.21	-0.78	0.36	0.10	-0.30	0.50	0.06	-0.16	0.28
$f^{-1}(\alpha)$	0.95	0.92	0.97	1.35	1.09	1.68	0.73	0.60	0.83	2.43	2.07	2.85

Table 2: *Posterior estimates for Fisher Z transformed correlation (*z(r)*) and log transformed RMSE (*log(r)*) values. The mean, lower 95%Credible Interval and upper 95%CrI estimates are shown for four Bayesian multi-level models. The left two models were fitted on vocal and guided imitation scores with neutralized timing. The right two models were fitted on guided and non-guided imitation scores with subjects' timing input. Rhat was 1.00 for all parameters. Parameters with entirely positive or entirely negative 95% credible intervals are in bold. The bottom row gives intercept values reverse transformed (f^{-1}) into correlation and RMSE scores.*

(95% CrI=[1.09, 1.68]). For RMSE, the slope for condition is negative for the entire credible interval, indicating slightly better imitation results for the vocal modality. In other words, when timing is not taken into account, vocal and gestural imitation perform similarly, with vocal imitation slightly better. For correlation, the effect of condition is uncertain because its credible interval spans both positive and negative values.

When timing was taken into account, gestures had lower correlation and higher RMSE, with a reverse transformed posterior means of 0.73 (95%CrI=[0.60,0.83]) for correlation and 2.43ST (95%CrI=[2.07, 2.85]) for RMSE. The slope for condition is positive for correlation and negative for RMSE across the entire credible interval, indicating the effect of the guide in improving gestural imitations. Nativeness and the interaction parameter had credible intervals that span both positive and negative values in all four models, so their effects are uncertain.

4. Perceptual Assessment

Given that the goal of learning intonation is to correctly convey intended meaning, a preliminary assessment of correctness was conducted as an online perceptual test using stimuli from one pair of phrases (10 and 10bis, 80 total stimuli) using JSPsych and JATOS [26, 27]. Gestural data with subjects' timing was resynthesized using WORLD [7]. 37 francophone natives listened to all stimuli presented in random order and selected between the two meanings in a forced choice. Figure 3 shows aggregate scores of "correctness", computed for each stimuli from the percentage of listeners that selected the intended meaning.

Z-tests ($p<0.05$) were used to compare scores against chance. For guided chironomy, 14 out of 20 stimuli scored significantly above chance, 6 from non-native subjects. For non-guided chironomy, 11 out of 20 scored significantly above chance, with only 3 from non-natives. Vocal imitation scored significantly above chance for all subject-phrase pairings. Fisher's exact test ($p<0.05$) was used to compare subject-phrase pairings across conditions. Guided chironomy performed significantly better than non-guided for 4 subject-phrases, where 3 were from non-native subjects. It also performed significantly better than reading for the 2 subject-phrase combinations with the lowest reading scores (0.19 and 0.43).

5. Discussion and Conclusions

This paper introduced an interface that enables the real-time gestural control of intonation in phrases produced by a vocal synthesizer. A study using an imitation paradigm for prosodic disambiguation has been conducted as a mean to explore chironomic stylisation of intonation for native and non-native speakers. The 4 questions raised in the introduction are revisited here.

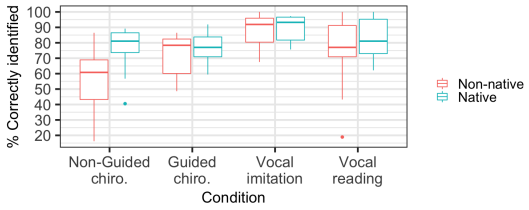


Figure 3: *Perceptive scores for stimuli of phrases 10 and 10bis, by condition and nativeness. Horizontal line indicates median, hinges indicate 25th and 75th percentiles.*

For quantitative measures, vocal and guided gestural imitation performed comparably when timing was not taken into account, a finding which corroborates prior results. Guided chironomy performed significantly better than non-guided chironomy in our results. Another finding is the additional difficulty from the added dimension of timing control, confirmed by lower comparison scores when timing was taken into account. Surprisingly, no statistically significant difference in quantitatives scores was found between native and non-native subjects, but the results of the perceptual experiment suggests further examination of this question. The perceptual experiment indicates that chironomy can be used to produce intonation that is correctly identifiable by native speakers. Vocal imitation performed the best overall, but it is well-known that production quality in mimicry can exceed what non-native subjects produce on their own. The non-native subjects available for this study are already fairly advanced, and errors in reading only occurred for a small percentage of stimuli. Nevertheless, the fact that subject-phrase combinations with mistakes in reading had high correctness scores in guided gestural imitation suggests an opportunity for chironomic practice. In conclusion native and non-native speakers succeeded in prosodic disambiguation using chironomic stylization of intonation. This allows us to envision this interface as a means for non-native speakers to practice the intonation of a foreign language.

6. Acknowledgements

This work has been supported by ANR GEPETO (ANR-19-CE28-0018) and ANR-10-LABX-0083. It contributes to the IdEx Université de Paris (ANR-18-IDEX-0001).

7. References

[1] C. d'Alessandro, A. Rillard, and S. LeBeux, "Chironomic stylization of intonation," *Journal of the Acoustical Society of America*, vol. 3, no. 129, pp. 1594–1604, Mar. 2011.

[2] P. J. Price, M. Ostendorf, and M. Park, "The use of prosody in syntactic disambiguation," *Journal of the Acoustical Society of America*, no. 90, pp. 2956–2970, Dec. 1991.

[3] Y. Zhang, H. Ding, P. Zelchenko, X. Cui, Y. Lin, Y. Zhan, and H. Zhang, "Prosodic disambiguation by chinese efl learners in a cooperative game task," *Proceedings of Speech Prosody 2018*, pp. 979–983, 06 2018.

[4] A. Fultz, "The use of prosody for disambiguation in english-french interlanguage," *Proc. 9th Gener. Approaches to Second Lang. Acquis. Conf.*, pp. 130–139, 2007.

[5] G. Locqueville, C. d'Alessandro, S. Delalez, B. Doval, and X. Xiao, "Voks: Digital instruments for chironomic control of voice samples," *Speech Communication*, vol. 125, pp. 97–113, 12 2020.

[6] Cycling74, "Max 6," https://cycling74.com/, 2011, accessed: 2019-05-05.

[7] M. Morise, F. YOKOMORI, and K. Ozawa, "World: A vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. E99.D, pp. 1877–1884, 07 2016.

[8] X. Xiao, G. Locqueville, C. D'Alessandro, and B. Doval, "T-Voks: the singing and speaking theremin," in *NIME 2019 International Conference on New Interfaces for Musical Expression*, Porto Alegre, Brazil, Jun. 2019, pp. 110–115.

[9] C. D'Alessandro, X. Xiao, G. Locqueville, and B. Doval, "Borrowed voices," in *International Conference on New Interfaces for Musical Expression NIME'19*, Porto Alegre, Brazil, Jun. 2019, pp. 2.2–2.4.

[10] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]," http://www.praat.org/ Version 6.1.08, 2019, accessed: 2019-12-05.

[11] F. Ramus, "Acoustic correlates of linguistic rhythm: Perspectives," *Proceedings of Speech Prosody 2002*, 04 2002.

[12] P. Mertens, "The prosogram: Semi-automatic transcription of prosody based on a tonal perception model," in *Proceedings of Speech Prosody 2004*, Nara, Japan, 2004.

[13] C. d'Alessandro and P. Mertens, "Automatic pitch contour stylization using a model of tonal perception," *Computer Speech and Language*, vol. 9, no. 3, pp. 257–288, 1995.

[14] D. Hermes, "Measuring the perceptual similarity of pitch contours," *Journal of speech, language, and hearing research : JSLHR*, vol. 41, pp. 73–82, 03 1998.

[15] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2020. [Online]. Available: https://www.R-project.org/

[16] A. E. . P. Bailey, *wCorr: Weighted Correlations*, 2017, r package version 1.9.1. [Online]. Available: https://CRAN.R-project.org/package=wCorr

[17] B. Gorman, *mltools: Machine Learning Tools*, 2018, r package version 0.3.5. [Online]. Available: https://CRAN.R-project.org/package=mltools

[18] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *Journal of Statistical Software*, vol. 67, no. 1, pp. 1–48, 2015.

[19] D. Barr, R. Levy, C. Scheepers, and H. Tily, "Random effects structure for confirmatory hypothesis testing: Keep it maximal," *Journal of memory and language*, vol. 68, pp. 255–278, 01 2013.

[20] P.-C. Bürkner, "brms: An R package for Bayesian multilevel models using Stan," *Journal of Statistical Software*, vol. 80, no. 1, pp. 1–28, 2017.

[21] ——, "Advanced Bayesian multilevel modeling with the R package brms," *The R Journal*, vol. 10, no. 1, pp. 395–411, 2018.

[22] Stan Development Team, "Stan modeling language users guide and reference manual," https://mc-stan.org, 2021, accessed: 2021-03-21.

[23] R. McElreath, *Statistical Rethinking: A Bayesian Course with Examples in R and Stan, 2nd Edition*, 2nd ed. CRC Press, 2020. [Online]. Available: http://xcelab.net/rm/statistical-rethinking/

[24] S. Vasishth, B. Nicenboim, M. Beckman, F. Li, and E. J. Kong, "Bayesian data analysis in the phonetic sciences: A tutorial introduction," *Journal of Phonetics*, vol. 71, pp. 147–161, 08 2018.

[25] L. Nalborczyk, C. Batailler, H. Loevenbruck, A. Vilain, and P.-C. Bürkner, "An introduction to bayesian multilevel models using brms: A case study of gender effects on vowel variability in standard indonesian," *Journal of Speech, Language, and Hearing Research*, vol. 62, 05 2019.

[26] J. de Leeuw, "jspsych: A javascript library for creating behavioral experiments in a web browser," *Behavior Research Methods*, vol. 1, no. 41, pp. 1–12.

[27] K. Lange, S. Kühn, and E. Filevich, "'just another tool for online studies' (jatos): An easy solution for setup and management of web servers supporting online studies," *PLOS ONE*, vol. 10, no. 6, pp. 1–14, 06 2015. [Online]. Available: https://doi.org/10.1371/journal.pone.0130834

Évaluation de la stylisation chironomique pour l'apprentissage de l'intonation du français L2

Xiao Xiao[1, 3] Nicolas Audibert[1] Grégoire Locqueville[2] Christophe d'Alessandro[2] Barbara Kuhnert[1] Rébecca Kleinberger[3] Claire Pillot-Loiseau[1]

(1) Laboratoire de Phonologie et Phonétique, Sorbonne Nouvelle, UMR7018, Paris, France

(2) Institut Jean le Rond d'Alembert, Sorbonne Université, UMR7190 CNRS, Paris, France

(3) MIT Media Lab, Massachusetts Institute of Technology, Cambridge, USA

[xiao.xiao, nicolas.audibert, barbara.kuhnert, claire.pillot]@sorbonne-nouvelle.fr, [gregoire.locqueville, christophe.dalessandro]@sorbonne-universite.fr, rebklein@mit.edu

RÉSUMÉ

Cet article présente la nouvelle analyse d'une étude pilote sur l'imitation de la prononciation de phrases françaises dans laquelle des locuteurs natifs et non-natifs sont aidés d'une interface gestuelle de modulation de synthèse vocale. L'interface est envisagée pour aider l'apprentissage de l'intonation d'une langue étrangère et permet le contrôle en temps-réel de la prosodie d'une phrase cible par le tracé de contours mélodiques sur une tablette tactile. Le présent travail propose une analyse des données de l'étude pilote en utilisant des modèles mixtes additifs généraux (GAMM), afin de modéliser et comparer directement les trajectoires chironomiques et les courbes de f0. Cette approche confirme les résultats antérieurs : la chironomie guidée serait comparable à l'imitation vocale outre le timing dont le contrôle est difficile pour tous les sujets. Elle suggère en plus des différences notables entre locuteurs natifs et non-natifs pour certaines phrases sous certaines conditions.

ABSTRACT

Comparing Chironomic Stylization and Vocal Pronunciation of French Intonation with General Additive Mixed Models

This paper presents a follow-up analysis of a pilot study where native and non-native speakers imitated the pronunciation of French phrases using their natural voice and a gesture-controlled interface for the real-time modulation of vocal synthesis. A pilot study was conducted to better understand how this interface might be used for non-native speakers to practice the intonation of a foreign language. Previous analysis of the pilot data used Bayesian multilevel linear models on similarity scores of subjects' f0 curves compared to the reference curves. The present work reanalyzes the same data using general additive mixed models (GAMMs). This new analysis confirms the prior results—that guided chironomy without considering timing is comparable to vocal imitation and that timing control is difficult for all subjects. It also reveals previously undiscovered differences between conditions and between native and non-native speakers.

MOTS-CLÉS : intonation, acquisition de langue étrangère, geste, GAMM, synthèse vocale, interaction homme-machine.

KEYWORDS: intonation, second-language acquisition, gesture, GAMM, vocal synthesis, human-computer interaction..

1 Introduction

Notre recherche explore l'utilisation de la stylisation chironomique pour l'apprentissage de l'intonation d'une langue étrangère. "Stylisation chironomique" désigne ici la synthèse vocale avec l'intonation stylisée et contrôlée en temps réel par la gestuelle de la main. Une telle approche multimodale procure trois avantages pour l'apprentissage de l'intonation. Tout d'abord, elle peut permettre d'entraîner l'oreille à percevoir des caractéristiques vocales peu familières en les présentant à travers des modalités visuelles et kinesthésiques. Deuxièmement, le contrôle de la prononciation par des gestes de la main (chironomie) contourne les schémas enracinés dans la voix naturelle qui peuvent être difficiles à corriger (d'Alessandro *et al.*, 2011). Enfin, la synthèse vocale permet à un utilisateur de se concentrer sur le niveau suprasegmental sans se préoccuper des détails phonétiques fins au niveau segmental. L'hypothèse principale est que l'approche multimodale apportée par la chironomie (kinesthésique, visuelle et auditive) pourrait renforcer l'expérience sensorielle du sujet et l'aider dans le processus d'apprentissage (c'est-à-dire saisir et mémoriser les caractéristiques d'intonation).

Des travaux antérieurs sur l'intonation chironomique de phrases françaises avec des locuteurs natifs ont conclu à des résultats comparables entre imitation vocale et gestuelle (d'Alessandro *et al.*, 2011). L'acquisition de l'intonation d'une langue étrangère pourrait ainsi être facilitée par le recours à la chironomie. Dans une étude précédente, les participants ont utilisé une tablette graphique et un stylet pour contrôler la mélodie de phrases synthétisées, mais sans permettre la modification des paramètres rythmiques. Le présent travail vise à évaluer l'utilisation de la chironomie comme substitution vocale par les locuteurs non natifs, et nous nous intéressons au contrôle simultané du rythme et de la mélodie de la parole. Pour cela, une interface mobile, Gepeto, a été développée pour permettre la modulation en temps réel de la mélodie et la synchronisation de la prononciation synthétisée.

Une étude pilote, utilisant un paradigme d'imitation pour la désambiguïsation prosodique, a été menée pour évaluer la capacité des locuteurs natifs et non natifs à percevoir, contrôler et modifier des modèles d'intonation correspondant à différentes réalisations possibles de phrases ambiguës, affichées dans le tableau 1 (Xiao *et al.*, 2021).

L'étude pilote visait également à recueillir des informations pour de futures itérations de l'interface Gepeto et pour la conception d'études additionnelles. Notre première analyse des données pilotes consistait à quantifier l'écart entre les courbes de F0 des sujets et le contour de référence, les scores étant analysés à l'aide de modèles linéaires bayésiens à plusieurs niveaux. Les résultats ont suggéré que pour les locuteurs natifs et non natifs, l'imitation chironomique à l'aide d'un guide visuel est comparable en précision à l'imitation vocale, et que le contrôle de la synchronisation était une source de difficulté. Étonnamment, aucune différence significative n'a été trouvée entre apprenants étrangers et natifs avec l'analyse précédente.

Cet article réexamine les données pilotes à l'aide de modèles mixtes additifs généraux (GAMM), qui permettent l'analyse statistique de données dynamiques non-linéaires (Wood, 2017; Wieling, 2018; Sóskuthy, 2017). Au lieu d'être réduites à un seul score de similarité par rapport à la référence, les courbes F0 sont directement modélisées indépendamment de leur courbe de référence.

Dans cet article, nous présentons tout d'abord l'interface de contrôle gestuel (Section 2) et un bref résumé de l'étude pilote (Section 3). Nous décrivons ensuite les modèles construits pour déterminer (1) si les courbes de chaque paire de phrases présentent une différence significative et (2) s'il y a des différences significatives entre les deux groupes de sujets.

2 Interface de contrôle

L'architecture est basée sur Voks (Locqueville *et al.*, 2020), un synthétiseur vocal performatif de haute qualité qui permet, en temps réel, le contrôle mélodique et rythmique d'échantillons de parole précédemment enregistrés ou de synthèse texte-parole, grâce à l'utilisation de gestes de la main. Voks est une application Max/MSP basée sur le vocodeur WORLD (Cycling74, 2011; Morise *et al.*, 2016), initialement développée pour des applications de synthèse chantée, par l'utilisation d'une tablette graphique contrôlée par un stylet ou d'un thérémine (Xiao *et al.*, 2019; D'Alessandro *et al.*, 2019).

Compte tenu de la diffusion massive des appareils mobiles, nous avons développé une interface mobile personnalisée pour Voks, contrôlée par les mouvements du doigt (au lieu d'un stylet). Le logiciel fonctionne sur un ordinateur Mac OS X, connecté via la bibliothèque logicielle Websockets à un appareil mobile (ici une tablette Samsung Galaxy S2 de 9,7 pouces) exécutant l'interface Gepeto, avec une latence moyenne d'environ 10 ms dans chaque direction. Voir (Xiao *et al.*, 2021) pour plus de détails sur l'architecture et l'interface.

La phrase à contrôler apparaît graphiquement en haut de l'interface Gepeto. Sous la phrase cible, la zone de contrôle utilise le tracé du doigt pour contrôler la resynthèse de la phrase de Voks. L'axe horizontal détermine la position temporelle dans l'échantillon d'origine à resynthétiser, qui est segmenté en fonction de la subdivision des syllabes spécifiée par un fichier Praat TextGrid (Boersma & Weenink, 2019). Les syllabes sont indiquées sur l'écran en alphabet phonétique international (API). La langue française étant souvent considérée comme "isochronique" (Ramus, 2002), toutes les syllabes sont affichées avec une largeur égale pour cette étude. Différents rythmes peuvent être réalisés en changeant la vitesse du mouvement du doigt sur la surface de contrôle. L'axe vertical détermine la hauteur mélodique du signal de sortie sur une échelle en demi-tons (DT) espacée de manière régulière, avec une plage de 24DT (2 octaves) calibrée autour du corpus d'étude (116-466Hz). Une position verticale plus élevée donne un son de sortie plus aigu.

Chaque phrase cible est accompagnée d'un guide visuel montrant la courbe d'intonation stylisée de l'enregistrement de référence généré par Prosogram (Mertens, 2004). Basé sur un modèle perceptif de (d'Alessandro & Mertens, 1995), Prosogram simplifie la courbe de hauteur d'un enregistrement en segments de droite, la parole resynthétisée avec ces courbes de hauteur stylisées ayant été évaluée perceptuellement identique aux stimuli d'origine. Un guide visuel stylisé a été choisi pour faciliter le traçage et éviter que les artefacts vocaux micro-prosodiques ne perturbent les utilisateurs.

Par défaut, un geste dans la zone de contrôle laisse une trace colorée dont la persistance dépend du mode choisi. L'utilisateur a le choix entre le "mode en fondu" où les traces s'estompent après 1,5 secondes et le "mode maintenu" où les gestes restent à l'écran jusqu'à ce qu'ils soient effacés (Figure 1). En mode maintenu, les trois boutons suivants sont activés. Un bouton rejoue le geste. Le bouton suivant efface le geste et le dernier enregistre le geste sur le serveur. Le bouton dans le coin inférieur gauche déclenche la lecture de l'audio de référence.

3 Corpus, Sujets, Tâche

Six paires de phrases lexicalement ambiguës ont été sélectionnées à partir d'un corpus plus large (Table 1). Chaque paire est composée de la même séquence de phonèmes, mais les deux phrases ont des sens différents dus à leur intonation qui induit un découpage prosodique différent dans les deux

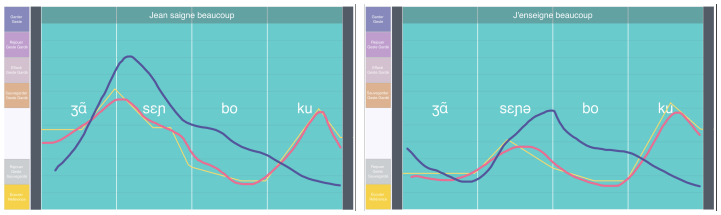


FIGURE 1 – Captures d'écran de l'interface Gepeto montrant une paire de phrases du corpus. Les lignes rose vif et violet foncé sont des traces gestuelles de l'utilisateur. Le rose s'estompe tandis que le violet reste affiché jusqu'à ce qu'il soit effacé et puisse être réécouté. La ligne jaune est le guide visuel.

cas. Les deux significations ont des probabilités d'occurrence à peu près égales. Les enregistrements de référence sont des énoncés déclaratifs lus par une locutrice francophone native.

Dix sujets ont participé à l'étude (2 hommes, 8 femmes, âgés de 20 à 48 ans, âge moyen 32,7 ans) [1]. Cinq d'entre eux sont des locuteurs non natifs issus de différentes langues maternelles : cantonais, portugais (deux sujets), mandarin, slovène. Tous ont suivi un cours de prononciation française d'un semestre et ont un niveau officiel avancé. Les 5 participants natifs sont des étudiants de premier cycle en orthophonie. Un non-natif et trois natifs ont déclaré avoir une expérience musicale (6-16 ans).

Un premier enregistrement est effectué, durant lequel il est demandé aux participants de lire les phrases en fonction de leur propre interprétation. Deuxièmement, les sujets sont invités à utiliser l'interface Gepeto pour imiter de leur mieux l'enregistrement de référence de chaque phrase. Lors de cette phase, les sujets doivent d'abord trouver un geste pour la phrase sans aide visuelle. Une fois le premier geste soumis, la courbe de hauteur stylisée de la phrase de référence apparaît et les sujets ont alors une autre chance de tracer un geste. Deux essais de familiarisation sont donnés pour la tâche d'imitation gestuelle. Enfin, les sujets enregistrent leur imitation vocale des phrases de référence.

Les paires de phrases appariées sont présentées en ordre aléatoire dans toutes les parties de l'étude. Pour les tâches d'imitation, aucune limite n'est imposée sur le nombre de fois où les références sont réécoutées, ni sur le temps pris pour imiter chaque phrase. L'ensemble de l'étude, y compris les instructions verbales et l'enquête d'information sur le sujet, dure entre 1h à 1h30. L'étude se déroule dans un studio insonorisé. Le son est transmis via un casque filaire et les enregistrements vocaux sont effectués à l'aide d'un microphone AKG C414 XLS connecté via une interface audio à un ordinateur portable Macbook Air. Un moniteur externe affiche la phrase en cours et offre une interface pour les sections d'enregistrement audio.

№de phrase	A	B (bis)
2	Tu parais très soucieux.	Tu paraîtrais soucieux.
7	Jean lève son verre.	J'enlève son verre.
8	Jean porte un journal.	J'emporte un journal.
10	Jean saigne beaucoup.	J'enseigne beaucoup.
11	Jean cadre la photo.	J'encadre la photo.
21	C'est la morsure.	C'est la mort sûre.

TABLE 1 – Corpus des phrases

1. L'expérience a été réalisée pendant la pandémie de COVID 19 et la disponibilité des sujets était limitée

4 Analyse

4.1 Préparation des données

Pour chaque participant, 4 prononciations par phrase avec différentes modalités sont recueillies : lecture vocale, imitation gestuelle non guidée, imitation gestuelle guidée et imitation vocale. Une extraction automatique de la fréquence fondamentale (F0) des enregistrements vocaux des sujets est effectuée à l'aide de Praat puis validée manuellement. La segmentation des énoncés dans chaque enregistrement est étiquetée au format TextGrid. Chaque enregistrement est normalisé pour égaliser la F0 moyenne et normalisé dans le temps en rééchantillonnant les valeurs de F0 à des intervalles de 10 millisecondes dans l'enregistrement de référence.

Les énoncés d'imitation gestuelle sont encodés sous la forme d'une série de points dans un espace tridimensionnel. Chaque point est basé sur une position 2D touchée par le sujet dans la région de contrôle de l'interface Gepeto et comprend les informations suivantes :

- `f` : la fréquence en demi-tons relative à la fréquence la plus basse dans l'interface Gepeto
- `scrub` : l'instant de référence dans l'enregistrement original, normalisé sur une échelle de 0 à 1
- `t` : le temps d'apparition du point courant mis à l'échelle de 0 à 1 en fonction du temps de l'ensemble du geste.

Dans le cas d'une parfaite synchronisation temporelle entre le geste effectué par le sujet et le modèle de phrase à imiter, les valeurs de `scrub` et `t` sont donc égales.

Deux types de données gestuelles ont été analysés : l'un avec les données t pour la dimension temporelle, qui représente les courbes de fréquence des sujets avec leur contrôle de synchronisation, un autre avec les données de "scrub" pour la dimension temporelle, qui représente à quel point les tracés F0 des sujets ressemblaient au guide courbes indépendamment du temps. Une modélisation statistique a ainsi été réalisée sur 4 conditions : lecture, imitation vocale, ainsi que deux versions de données chironomiques, les unes non guidées et les autres guidées.

4.2 Modélisation statistique

À l'aide de la commande `bam` du package `mgcv` du language de programmation R, deux types de modèles mixtes additifs généraux (GAMM) ont été construits pour répondre à chacune des deux questions de recherche (Wood, 2017, 2011). Le premier type examine la différence de F0 entre les versions A et B de chaque phrase pour chaque condition et chaque type de sujet. Le deuxième type examine la différence entre les sujets natifs et non natifs pour chaque phrase et chaque condition distincte. Des modèles distincts ont été créés pour chaque sous-ensembles de données pour faciliter leur interprétation. Le type de spline de lissage par défaut ("tp") est utilisé pour tous les modèles, et une correction pour l'autocorrélation est incluse pour tous les modèles.

Question 1 : Différence significative entre les types de phrases

Le premier type de modèles s'exprime par la formule suivante :

```
f ~ typePhrase + s(t) + s(t, by=typePhrase) # effets fixes
    + s(t, subject, by="fs", m=1)   # effets aléatoires
    + s(t, subject, by=typePhrase, bs="fs", m=1)
```

`typePhrase` est une variable binaire définie sur `TRUE` pour les versions B de la phrase et `FALSE` sinon. Elle est incluse dans la spécification du modèle en tant qu'effet fixe, pour prendre en compte les différences constantes, et en tant que spline de lissage, pour tenir compte des différences non linéaires. Pour tenir compte de la variabilité entre sujets, deux lissages aléatoires sont ajoutés à la spécification du modèle, représentant les différences globales non linéaires pour chaque sujet et les différences liées aux versions A et B de la phrase.

Étant donnée la localisation variable de la première séparation prosodique entre les paires de phrases, des instances distinctes de ce modèle ont été créées pour chaque paire de phrases.

Question 2 : Différence significative entre les sujets natifs et non natifs

Le second type de modèles s'exprime par la formule :

```
f ~ isNative + s(t) + s(t, by=isNative)   # effets fixes
    + s(t, subject, bs="fs", m=1)   # effet aléatoire
```

La variable binaire `isNative` indique pour chaque sujet s'il est francophone natif ou non. Elle est incluse dans le modèle à la fois en tant qu'effet fixe et en tant que spline de lissage. Un seul lissage aléatoire basé sur le sujet capture les différences individuelles entre les locuteurs. Un modèle séparé a été construit pour chaque phrase individuelle, dans chaque condition.

4.3 Test de signification

Sur la base des recommandations de (Sóskuthy, 2017), deux méthodes sont utilisées pour tester la significativité de chaque modèle. Pour la première méthode, chaque modèle est comparé à une version simplifiée excluant les termes paramétriques et courbes lissées à effet fixe tout en gardant la même courbe lissée de base et les mêmes effets aléatoires. La deuxième méthode repose sur l'affichage des différences entre conditions intégrées dans le package `itsadug` via la fonction `plotdiff`, qui indique visuellement les régions significativement différentes (van Rij *et al.*, 2020). Une différence est considérée comme significative si le modèle à effets fixes est évalué significativement meilleur que le modèle de base ($p < 0{,}05$) et si l'inspection visuelle de la courbe de différence indique que la zone dans laquelle les conditions comparées diffèrent significativement correspond bien à celle attendue, en l'occurrence en excluant les parties initiales et finales des énoncés comme illustré par la figure 2. Un résumé de la significativité pour tous les modèles construits est présenté dans les tableaux 2 et 3.

5 Résultats et discussion

Les résultats des deux analyses sont résumés dans les tableaux 2 et 3. Pour les sujets natifs et non natifs, les versions A et B de toutes les phrases présentent des différences significatives dans l'imitation vocale. Pour l'imitation chironomique avec les données temporelles des sujets, seulement la moitié des paires de phrases ont des courbes lissées significativement différentes pour les deux versions. Les phrases dont les courbes lissées sont significativement différentes varient selon les sujets natifs et non natifs. Lorsque le temps n'est pas pris en compte, les deux groupes de sujets distinguent les phrases A et B pour 5 des 6 paires de phrases à l'aide du guide. Lorsqu'ils ne sont pas guidés, les locuteurs non natifs tracent des courbes significativement différentes pour 4 paires de phrases sur 6,

tandis que les locuteurs natifs n'ont des courbes significativement différentes que pour 2 paires de phrases. Ces résultats sont cohérents avec l'analyse précédente, la chironomie guidée sans timing est comparable à l'imitation vocale et le timing apparaît comme une source de difficulté.

L'analyse révèle un résultat nouveau et inattendu : les courbes des locuteurs natifs et non natifs ne sont pas significativement différentes pour la plupart des paires lors de la lecture initiale en fonction de leur propre interprétation des phrases. Cela suggère que malgré le contrôle inhérent à la tâche de lecture proposée, la distinction entre les paires de phrases est facilement cachée par la variabilité de prononciation dans chaque version de la phrase.

Des différences significatives entre les sujets non-natifs et natifs n'ont pu être trouvées que dans certaines des phrases. Il est intéressant de noter qu'aucune différence significative n'a été trouvée pour aucune des phrases en condition d'imitation vocale et de chironomie guidée, ni avec le timing propre aux sujets, ni en considérant le timing relatif (scrub). Ce résultat suggère que la chironomie guidée permet aux non-natifs d'atteindre le même degré de précision que les natifs.

	Condition					
№ phrase	**Lecture**	**Imitation Vocale**	**Chironomie Non Guidée**	**Chironomie Guidée**	**Chiro. Non Guidée Scrub**	**Chiro. Guidée Scrub**
	Sujets Natifs					
2	Non	Oui	Non	Non	Non	Non
7	Non	Oui	Oui	Oui	Non	Oui
8	Non	Oui	Non	Non	Oui	Oui
10	Non	Oui	Oui	Oui	Oui	Oui
11	Non	Oui	Oui	Oui	Non	Oui
21	Oui	Oui	Non	Non	Non	Oui
	Sujets Non Natifs					
2	Oui	Oui	Oui	Non	Oui	Oui
7	Non	Oui	Oui	Oui	Non	Oui
8	Non	Oui	Non	Oui	Oui	Oui
10	Non	Oui	Non	Non	Non	Oui
11	Non	Oui	Non	Non	Oui	Non
21	Oui	Oui	Oui	Oui	Oui	Oui

TABLE 2 – Résumé des situation où une différence significative peut être trouvée entre les courbes A et B pour chaque phrase

	Condition		
№ phrase	**Lecture**	**Chironomie Non Guidée**	**Chiro. Non Guidée Scrub**
2A Tu parais très soucieux	Non	Oui	Oui
2B Tu paraîtrais soucieux	Oui	Oui	Oui
11A Jean cadre la photo	Non	Oui	Non
11B J'encadre la photo	Non	Oui	Non
21A C'est la mort sûre	Non	Non	Oui

TABLE 3 – Phrases et conditions dans lesquelles les courbes des locuteurs natifs diffèrent significativement de celles des non natifs. Aucune différence significative n'a été trouvée pour toutes les combinaisons d'expressions et de conditions ne figurant pas dans ce tableau.

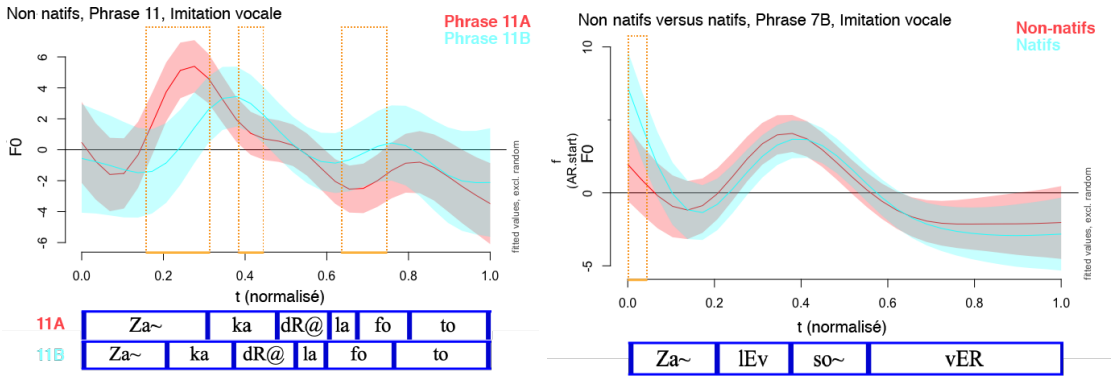


FIGURE 2 – Trajectoires modélisées et zones dans lesquelles la différence entre conditions est significative. Les différences au début des énoncés (graphique à droite) sont éxclues.

6 Conclusions

Les résultats de cette analyse montrent que les modèles de type GAMM apparaissent comme un outil approprié pour l'analyse des trajectoires gestuelles et leur comparaison avec les courbes de fréquence fondamentale. A ce titre, l'identification visuelle des zones dans lesquelles les divergences entre conditions sont les plus importantes pourrait se révéler particulièrement utile dans une optique didactique. Confirmant les résultats de (Xiao *et al.*, 2021), le contrôle du temps s'est avéré plus complexe tant pour les sujets natifs que non-natifs et sera donc un objectif prioritaire des travaux visant à l'amélioration de l'interface Gepeto. Il serait également utile de collecter des données sur chaque tentative du sujet plutôt que sur l'essai final uniquement.

Plus de soin peut également être apporté au choix des sujets, d'abord en effectuant à nouveau l'expérience auprès de sujets non-natifs de niveau moins avancé que ceux de cette expérimentation. En outre, au lieu de traiter les sujets non natifs comme un groupe uniforme, nous pouvons imaginer un diagnostic pour identifier des problèmes spécifiques dans l'intonation d'un sujet, comme détecter la différence entre montée et descente d'intonation ou l'exécution d'un schéma rythmique particulier. L'interface Gepeto pourrait être adaptée en un outil pour déterminer si le problème est enraciné dans la capacité à écouter, la capacité à contrôler la voix naturelle ou une erreur dans la représentation interne d'un son.

Enfin, le fait que les sujets natifs aient eu des difficultés à différencier les paires de phrases dans la condition de lecture suggère que les études futures devraient accorder plus de soin à l'étude de la variabilité de production de ce type de phrase par plusieurs francophones natifs. Néanmoins, cette analyse renforce la constatation que la chironomie a un potentiel en tant qu'outil de pratique de l'intonation.

Remerciements

Ce travail est soutenu par le projet ANR GEPETO (ANR-19-CE28-0018), ainsi que par le programme « Investissements d'Avenir » Labex EFL (ANR-10-LABX-0083). Il contribue à l'IdEx Université de Paris (ANR-18-IDEX-0001).

Références

BOERSMA P. & WEENINK D. (2019). Praat : doing phonetics by computer [computer program]. http://www.praat.org/ Version 6.1.08. Accessed : 2021-03-21.

CYCLING74 (2011). Max 6. https://cycling74.com/. Accessed : 2021-03-21.

D'ALESSANDRO C. & MERTENS P. (1995). Automatic pitch contour stylization using a model of tonal perception. *Computer Speech and Language*, **9**(3), 257–288.

D'ALESSANDRO C., RILLARD A. & LEBEUX S. (2011). Chironomic stylization of intonation. *Journal of the Acoustical Society of America*, **3**(129), 1594–1604.

D'ALESSANDRO C., XIAO X., LOCQUEVILLE G. & DOVAL B. (2019). Borrowed voices. In *International Conference on New Interfaces for Musical Expression NIME'19*, p. 2.2–2.4, Porto Alegre, Brazil.

LOCQUEVILLE G., D'ALESSANDRO C., DELALEZ S., DOVAL B. & XIAO X. (2020). Voks : Digital instruments for chironomic control of voice samples. *Speech Communication*, **125**, 97–113.

MERTENS P. (2004). The prosogram : Semi-automatic transcription of prosody based on a tonal perception model. In *Proceedings of Speech Prosody 2004*, Nara, Japan.

MORISE M., YOKOMORI F. & OZAWA K. (2016). World : A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, **E99.D**, 1877–1884.

RAMUS F. (2002). Acoustic correlates of linguistic rhythm : Perspectives. *Proceedings of Speech Prosody 2002*.

SÓSKUTHY M. (2017). Generalised additive mixed models for dynamic analysis in linguistics : a practical introduction.

VAN RIJ J., WIELING M., BAAYEN R. H. & VAN RIJN H. (2020). itsadug : Interpreting time series and autocorrelated data using gamms. R package version 2.4.

WIELING M. (2018). Analyzing dynamic phonetic data using generalized additive mixed modeling : A tutorial focusing on articulatory differences between l1 and l2 speakers of english. *Journal of Phonetics*, **70**, 86–116.

WOOD S. (2017). *Generalized Additive Models : An Introduction with R*. Chapman and Hall/CRC, 2 edition.

WOOD S. N. (2011). Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society (B)*, **73**(1), 3–36.

XIAO X., AUDIBERT N., LOCQUEVILLE G., D'ALESSANDRO C., KUHNERT B. & PILLOT-LOISEAU C. (2021). Prosodic Disambiguation Using Chironomic Stylization of Intonation with Native and Non-Native Speakers. In *Proc. Interspeech 2021*, p. 516–520.

XIAO X., LOCQUEVILLE G., D'ALESSANDRO C. & DOVAL B. (2019). T-Voks : the singing and speaking theremin. In *NIME 2019 International Conference on New Interfaces for Musical Expression*, p. 110–115, Porto Alegre, Brazil.

Bibliography

[1] George D. Allen. The location of rhythmic stress beats in english : an experimental study ii. *Language and Speech*, 15(2):179–195, 1972. PMID: 4653684.

[2] George D. Allen. The location of rhythmic stress beats in english: an experimental study i. *Language and Speech*, 15(1):72–100, 1972. PMID: 5073939.

[3] Georges Aperghis and Grégory Beller. Contrôle gestuel de la synthèse concaténative en temps réel dans luna park. *Rapport de recherche et développement*, 2011.

[4] Luc Ardaillon. *Synthesis and expressive transformation of singing voice*. Theses, Université Pierre et Marie Curie - Paris VI, November 2017.

[5] Gisa Aschersleben. Temporal control of movements in sensorimotor synchronization. *Brain and cognition*, 48(1):66–79, 2002.

[6] Maria Astrinaki, Nicolas d'Alessandro, Benjamin Picart, Thomas Drugman, and Thierry Dutoit. Reactive and continuous control of hmm-based speech synthesis. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 252–257, 2012.

[7] Lucie Bailly, Nathalie Henrich Bernardoni, Frank Müller, Anna-Katharina Rohlfs, and Markus Hess. Ventricular-fold dynamics in human phonation. *Journal of Speech, Language, and Hearing Research*, 57(4):1219–1242, 2014.

[8] Dan Barry, David Dorran, and Eugene Coyle. Time and pitch scale modification: A real-time framework and tutorial. In *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, volume 9, 2008.

[9] Nicola Bernardini and Alvise Vidolin. Sustainable live electro-acoustic music. In *Proceedings of the International Sound and Music Computing Conference*, 2005.

[10] Mark Beutnagel, Alistair Conkie, and Ann K. Syrdal. Diphone synthesis using unit selection. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[11] Niels Bogaards, Axel Roebel, and Xavier Rodet. Sound Analysis and Processing with AudioSculpt 2. In *International Computer Music Conference (ICMC)*, pages 1–1, Miami, United States, November 2004. cote interne IRCAM: Bogaards04a.

[12] Alain Bonardi and Jérome Barthélemy. The preservation, emulation, migration, and virtualization of live electronics for performing arts: An overview of musical and technical

issues. *J. Comput. Cult. Herit.*, 1(1), jun 2008.

[13] Fabian Brackhane, Richard Sproat, and Jürgen Trouvain. Wolfgang von Kempelen: Mechanismus der menschlichen Sprache/The mechanism of human speech. Commented transliteration and translation into English. *Studientexte zur Sprachkommunikation*, 87/88, 2017. `http://www.coli.uni-saarland.de/~trouvain/kempelen.html`.

[14] Didier Cadic, Cédric Boidin, and Christophe d'Alessandro. Vocalic sandwich, a unit designed for unit selection tts. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[15] Katy Carlson. How prosody influences sentence comprehension. *Language and Linguistics Compass*, 3(5):1188–1200, 2009.

[16] Francis Charpentier and Michel Stella. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In *ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 11, pages 2015–2018. IEEE, 1986.

[17] Donald G Childers and Chih K Lee. Vocal quality factors: Analysis, synthesis, and perception. *the Journal of the Acoustical Society of America*, 90(5):2394–2410, 1991.

[18] S.D. Christman. Handedness. In V.S. Ramachandran, editor, *Encyclopedia of Human Behavior (Second Edition)*, pages 290–296. Academic Press, San Diego, second edition edition, 2012.

[19] Christophe Coupé, Yoon Mi Oh, Dan Dediu, and François Pellegrino. Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science Advances*, 5(9), 2019.

[20] Christophe d'Alessandro, Samuel Delalez, Boris Doval, Lionel Feugère, and Olivier Perrotin. Les instruments chanteurs. *Acoustique et Techniques*, 89:36–43, 2018.

[21] Christophe d'Alessandro, Lionel Feugère, Sylvain Le Beux, Olivier Perrotin, and Albert Rilliard. Drawing melodies: Evaluation of chironomic singing synthesis. *The Journal of the Acoustical Society of America*, 135(6):3601–3612, 2014.

[22] Christophe d'Alessandro, Lionel Feugère, Olivier Perrotin, Samuel Delalez, and Boris Doval. Le contrôle des instruments chanteurs. In *14e Congrès Français d'Acoustique*. Société Française d'Acoustique, 2018.

[23] Christophe d'Alessandro and Piet Mertens. Automatic pitch contour stylization using a model of tonal perception. *Computer Speech and Language*, 9(3):257–288, 1995.

[24] Christophe d'Alessandro, Albert Rilliard, and Sylvain Le Beux. Computerized chironomy: evaluation of hand-controlled intonation reiteration. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[25] Christophe d'Alessandro, Albert Rilliard, and Sylvain Le Beux. Chironomic stylization of intonation. *Journal of the Acoustical Society of America*, 129(3):1594–1604, 2011.

[26] Christophe d'Alessandro, Xiao Xiao, Grégoire Locqueville, and Boris Doval. Borrowed voices. In *International Conference on New Interfaces for Musical Expression NIME'19*, pages 2–2, 2019.

[27] Nicolas D'Alessandro, Christophe d'Alessandro, Sylvain Le Beux, and Boris Doval. Real-time calm synthesizer: New approaches in hands-controlled voice synthesis. In *NIME*, volume 6, pages 266–271. Citeseer, 2006.

[28] Nicolas d'Alessandro and Thierry Dutoit. Handsketch bi-manual controller: investigation on expressive control issues of an augmented tablet. In *Proceedings of the 7th international conference on New interfaces for musical expression*, pages 78–81, New York, USA, 2007. ACM.

[29] Nicolas D'Alessandro and Thierry Dutoit. Advanced techniques for vertical tablet playing a overview of two years of practicing the handsketch 1. x. In *NIME*, pages 173–174, 2009.

[30] Samuel Delalez. *Vokinesis : instrument de contrôle suprasegmental de la synthèse vocale*. PhD thesis, Université Paris Saclay, 2017.

[31] Samuel Delalez and Christophe d'Alessandro. Vokinesis: syllabic control points for performative singing synthesis. In *NIME*, pages 198–203, 2017.

[32] Philippe Depalle and Gilles Poirot. SVP: A modular system for analysis, processing and synthesis of sound signals. In *Proceedings of the 1991 International Computer Music Conference, ICMC 1991, Montreal, Quebec, Canada, October 16-20, 1991*. Michigan Publishing, 1991.

[33] Kay Dickinson. 'believe'? vocoders, digitalised female identity and camp. *Popular Music*, 20(3):333–347, 2001.

[34] Randy L. Diehl. Acoustic and auditory phonetics: the adaptive design of speech sound systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 363(1493):965–978, 2008.

[35] Mark Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14–27, 1986.

[36] Boris Doval, Christophe d'Alessandro, and Nathalie Henrich Bernardoni. The voice source as a causal/anticausal linear filter. In *VOQUAL'03*, page 1, 2003.

[37] Boris Doval, Christophe d'Alessandro, and Nathalie Henrich Bernardoni. The spectrum of glottal flow models. *Acta Acustica united with Acustica*, 92(6):1026–1046, 11 2006.

[38] Homer Dudley. The vocoder—electrical re-creation of speech. *Journal of the Society of Motion Picture Engineers*, 34(3):272–278, 1940.

[39] Homer Dudley, R.R. Riesz, and S.S.A. Watkins. A synthetic speaker. *Journal of the Franklin Institute*, 227(6):739–764, 1939.

[40] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. van der Vrecken. The mbrola project: towards a set of high quality speech synthesizers free of use for non commercial purposes. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96*, volume 3, pages 1393–1396 vol.3, Oct 1996.

[41] Thierry Dutoit and Baris Bozkurt. Speech synthesis. In *Handbook of Signal Processing in Acoustics*, pages 557–585. Springer, 2008.

[42] Christophe d'Alessandro. Voice source parameters and prosodic analysis. *Methods in empirical prosody research*, 3, 2006.

[43] Gunnar Fant. *Acoustic theory of speech production*. Walter de Gruyter, 1970.

[44] Sidney S. Fels and Geoffrey E. Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE transactions on Neural Networks*, 4(1):2–8, 1993.

[45] Sidney S. Fels and Geoffrey E. Hinton. Glove-talk ii-a neural-network interface which maps gestures to parallel formant speech synthesizer controls. *IEEE Transactions on Neural Networks*, 8(5):977–984, 1997.

[46] Eva M. Fernández and Helen Smith Cairns. *Fundamentals of psycholinguistics*. John Wiley & Sons, 2010.

[47] Lionel Feugère. *Synthèse par règles de la voix chantée contrôlée par le geste et applications musicales*. Theses, Université Paris-Saclay, September 2013.

[48] Lionel Feugère, Christophe d'Alessandro, Boris Doval, and Olivier Perrotin. Cantor digitalis: chironomic parametric synthesis of singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2017(1):2, 1 2017.

[49] Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden markov model: Analysis and applications. *Machine learning*, 32(1):41–62, 1998.

[50] Ronald A. Fisher. Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.

[51] W. Tecumseh Fitch. The biology and evolution of music: A comparative perspective. *Cognition*, 100(1):173–215, 2006. The Nature of Music.

[52] James L. Flanagan and Roger M. Golden. Phase vocoder. *Bell system technical Journal*, 45(9):1493–1509, 1966.

[53] Kenneth Flowers. Handedness and controlled movement. *British Journal of Psychology*, 66(1):39–52, 1975.

[54] Jerry A Fodor and Thomas G Bever. The psychological reality of linguistic segments. *Journal of verbal learning and verbal behavior*, 4(5):414–420, 1965.

[55] Carol A Fowler. "perceptual centers" in speech production and perception. *Perception & Psychophysics*, 25(5):375–388, 1979.

[56] Jules Françoise, Norbert Schnell, Riccardo Borghesi, and Frédéric Bevilacqua. Probabilistic models for designing motion and sound relationships. In *Proceedings of the 2014 international conference on new interfaces for musical expression*, pages 287–292, 2014.

[57] Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, and Design Patterns. Elements of reusable object-oriented software. *Design Patterns. massachusetts: Addison-Wesley Publishing Company*, 1995.

[58] Albert Glinsky. *Theremin: ether music and espionage*. University of Illinois Press, 2000.

[59] Vincent Goudard. John, the semi-conductor: A tool for comprovisation. In Sandeep Bhagwati and Jean Bresson, editors, *Proceedings of the International Conference on Technologies for Music Notation and Representation – TENOR'18*, pages 43–49, Montreal, Canada, 2018. Con-

cordia University.

[60] Dik J. Hermes. Measuring the perceptual similarity of pitch contours. *Journal of Speech, Language, and Hearing Research*, 41(1):73–82, 1998.

[61] Charles E. Hoequist, jr. The perceptual center and rhythm categories. *Language and Speech*, 26(4):367–376, 1983.

[62] Peter Howell. An acoustic determinant of perceived and produced anisochrony. In *Proceedings of the 10th international Congress of Phonetic Sciences*, pages 429–433. Foris Dordrecht, Holland, 1984.

[63] Hui Huang. An analysis of classical chinese four-lined and five-charactered poem translation a comparative study of the english versions of meng haoran's chun xiao. In *Proceedings of the 2013 International Conference on the Modern Development of Humanities and Social Science*, pages 315–317. Atlantis Press, 2013.

[64] N.R. Ibbotson and John Morton. Rhythm and dominance. *Cognition*, 9(2):125–138, 1981.

[65] Colin J. Ihrig. *JavaScript Object Notation*, pages 263–270. Apress, Berkeley, CA, 2013.

[66] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[67] Peter M Janker. Evidence for the p-center syllable-nucleus-onset correspondence hypothesis. *Elements of general phonetics*, 7:94–124, 1996.

[68] Pierre Jouvelot and Yann Orlarey. Dependent vector types for data structuring in multirate faust. *Computer Languages, Systems & Structures*, 37(3):113–131, 2011.

[69] Lydia Kavina and Elizabeth Parcells. My experience with the theremin. *Leonardo Music Journal*, 6:51–55, 1996.

[70] Thomas Kisler, Uwe Reichel, and Florian Schiel. Multilingual processing of speech via web services. *Computer Speech & Language*, 45:326 – 347, 2017.

[71] Anssi P Klapuri, Antti J Eronen, and Jaakko T Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2005.

[72] Barbara Kühnert and Francis Nolan. The origin of coarticulation. *Coarticulation: Theory, data and techniques*, pages 7–30, 1999.

[73] Sylvain Le Beux, Boris Doval, and Christophe d'Alessandro. Issues and solutions related to real-time td-psola implementation. In *Audio Engineering Society Convention 128*. Audio Engineering Society, 2010.

[74] Sylvain Le Beux, Christophe d'Alessandro, Albert Rilliard, and Boris Doval. Calliphony: A system for real-time gestural modification of intonation and rhythm. In *Speech Prosody*, 2010.

[75] Sylvain Le Beux, Albert Rilliard, and Christophe d'Alessandro. Calliphony: a real-time intonation controller for expressive speech synthesis. In *SSW*, pages 345–350. Citeseer, 2007.

[76] Grégoire Locqueville, Christophe d'Alessandro, Samuel Delalez, Boris Doval, and Xiao Xiao. Voks: Digital instruments for chironomic control of voice samples. *Speech Communication*, 125:97–113, 2020.

[77] Thomas Lucas, Christophe d'Alessandro, and Serge de Laubier. Mono-replay : a software tool for digitized sound animation. In *NIME 2021 (International Conference on New Interfaces for Musical Expression)*, 4 2021. https://nime.pubpub.org/pub/8lqitvvq.

[78] Peter F. MacNeilage. The frame/content theory of evolution of speech production. *Behavioral and Brain Sciences*, 21(4):499–511, 1998.

[79] Andre Malécot. Frequency of occurrence of french phonemes and consonant clusters. *Phonetica*, 29(3):158–170, 1974.

[80] Stephen Michael Marcus. Acoustic determinants of perceptual center (p-center) location. *Perception & psychophysics*, 30(3):247–256, 1981.

[81] Raul Masu, Adam Pultz Melbye, John Sullivan, and Alexander Refsum Jensenius. Nime and the environment: toward a more sustainable nime practice. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. The International Conference on New Interfaces for Musical Expression, 2021.

[82] Piet Mertens. The prosogram: semi-automatic transcription of prosody based on a tonal perception model. In *Proc. Speech Prosody 2004*, pages 549–552, 2004.

[83] Masanori Morise. Cheaptrick, a spectral envelope estimator for high-quality speech synthesis. *Speech Communication*, 67:1 – 7, 2015.

[84] Masanori Morise. D4c, a band-aperiodicity estimator for high-quality speech synthesis. *Speech Communication*, 84:57 – 65, 2016.

[85] Masanori Morise. Harvest: A high-performance fundamental frequency estimator from speech signals. In *Proc. Interspeech 2017*, pages 2321–2325, 2017.

[86] Masanori Morise and Yusuke Watanabe. Sound quality comparison among high-quality vocoders by using re-synthesized speech. *Acoustical Science and Technology*, 39(3):263–265, 2018.

[87] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. World: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions*, 99-D:1877–1884, 2016.

[88] John Morton, Steve Marcus, and Clive Frankish. Perceptual centers (p-centers). *Psychological review*, 83(5):405, 1976.

[89] Eric Moulines and Francis Charpentier. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication*, 9(5-6):453–467, 1990.

[90] Meinard Müller. *Dynamic Time Warping*, pages 69–84. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[91] Julian of Norwich. *Revelations of Divine Love*, chapter XXXI. Grace H. Warrack, 1901. Avail-

able online at Project Gutenberg: `https://www.gutenberg.org/ebooks/52958`.

[92] Yann Orlarey, Dominique Fober, and Stephane Letz. Syntactical and semantical aspects of faust. *Soft Computing*, 8(9):623–632, 2004.

[93] Yann Orlarey, Dominique Fober, and Stéphane Letz. FAUST : an Efficient Functional Approach to DSP Programming. In *NEW COMPUTATIONAL PARADIGMS FOR COMPUTER MUSIC*, pages 65–96. Éditions DELATOUR FRANCE, 2009.

[94] Douglas O'Shaughnessy. A study of french vowel and consonant durations. *Journal of Phonetics*, 9(4):385–406, 1981.

[95] Laurel S. Pardue, Christopher Harte, and Andrew P. McPherson. A low-cost real-time tracking system for violin. *Journal of New Music Research*, 44(4):305–323, 2015.

[96] Gordon E. Peterson, William S-Y. Wang, and Eva Sivertsen. Segmentation techniques in speech synthesis. *The Journal of the Acoustical Society of America*, 30(8):739–742, 1958.

[97] Kenneth L Pike. *The intonation of American English*. University of Michigan Press, 1945.

[98] Bernd Pompino-Marschall. On the psychoacoustic nature of the p-center phenomenon. *Journal of Phonetics*, 17(3):175–192, 1989.

[99] Patti J. Price, Mari Ostendorf, Stefanie Shattuck-Hufnagel, and Cynthia Fong. The use of prosody in syntactic disambiguation. *the Journal of the Acoustical Society of America*, 90(6):2956–2970, 1991.

[100] Metason project. s2m wacom objects, version 1.1. Available at `https://metason.prism.cnrs.fr/Resultats/MaxMSP/`, 2016.

[101] Miller Puckette. Max at seventeen. *Computer Music Journal*, 26(4):31–43, 2002.

[102] Miller S Puckette et al. Pure data. In *ICMC*, 1997.

[103] Franck Ramus, Marina Nespor, and Jacques Mehler. Correlates of linguistic rhythm in the speech signal. *Cognition*, 73(3):265–292, 1999.

[104] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.

[105] Bruno H Repp. Automaticity and voluntary control of phase correction following event onset shifts in sensorimotor synchronization. *Journal of Experimental Psychology: Human Perception and Performance*, 28(2):410, 2002.

[106] Bruno H. Repp. Sensorimotor synchronization: A review of the tapping literature. *Psychonomic Bulletin & Review*, 12(6):969–992, Dec 2005.

[107] Bruno H Repp and Yi-Huang Su. Sensorimotor synchronization: a review of recent research (2006–2012). *Psychonomic bulletin & review*, 20(3):403–452, 2013.

[108] Axel Röbel. A new approach to transient processing in the phase vocoder. In *6th International Conference on Digital Audio Effects (DAFx)*, pages 344–349, 2003.

[109] Axel Röbel and Xavier Rodet. Efficient spectral envelope estimation and its application to pitch shifting and envelope preservation. In *International Conference on Digital Audio Effects*,

pages 30–35, 2005.

[110] Axel Röbel, Fernando Villavicencio, and Xavier Rodet. On cepstral and all-pole based spectral envelope modeling with unknown model order. *Pattern Recognition Letters*, 28(11):1343–1350, 2007.

[111] Axel Roebel. Transient detection and preservation in the phase vocoder. In *International Computer Music Conference (ICMC)*, pages 247–250, 2003.

[112] Axel Roebel. *ISiS: Ircam Singing Synthesis*. IRCAM, 2018. Available at `https://isis-documentation.readthedocs.io/en/latest/`.

[113] Bernard Roubeau, Nathalie Henrich, and Michèle Castellengo. Laryngeal vibratory mechanisms: the notion of vocal register revisited. *Journal of voice*, 23(4):425–438, 2009.

[114] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, Riccardo Borghesi, et al. Mubu and friends–assembling tools for content based real-time interactive audio processing in max/msp. In *ICMC*, 2009.

[115] Hans Schütte. Ein funktionsschema für die wahrnehmung eines gleichmäßigen rhythmus in schallimpulsfolgen. *Biological cybernetics*, 29(1):49–55, 1978.

[116] Stefania Serafin, Richard Dudas, Marcelo M Wanderley, and Xavier Rodet. Gestural control of a real-time physical model of a bowed string instrument. In *ICMC*, 1999.

[117] Charles Severance. Discovering javascript object notation. *Computer*, 45(4):6–8, 2012.

[118] Joseph Smith. Sprechstimme. *Journal of Singing*, 72(5):547–562, May 2016. Copyright - Copyright National Association of Teachers of Singing May/Jun 2016; Dernière mise à jour - 2016-05-17.

[119] Julius O. Smith. *Spectral Audio Signal Processing*. `http://ccrma.stanford.edu/~jos/sasp/`, accessed February 3, 2022. online book, 2011 edition.

[120] Will Styler. Using praat for linguistic research. *University of Colorado at Boulder Phonetics Lab*, 2013.

[121] Daniel Teruggi. Preserving and diffusing. *Journal of New Music Research*, 30(4):403–405, 2001.

[122] Leon S Theremin and Oleg Petrishev. The design of a musical instrument based on cathode relays. *Leonardo Music Journal*, 6(1):49–50, 1996.

[123] Ingo R Titze and Daniel W Martin. Principles of voice production, 1998.

[124] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 125:2, 2016.

[125] Rudi C. Villing, Bruno H. Repp, Tomas E. Ward, and Joseph M. Timoney. Measuring perceptual centers using the phase correction response. *Attention, Perception, & Psychophysics*, 73(5):1614–1629, 2011.

[126] Petra Wagner. *The rhythm of language and speech: Constraints, models, metrics and applications.*

online publication, 2008.

[127] Marcelo M Wanderley, Jean-Philippe Viollet, Fabrice Isart, and Xavier Rodet. On the choice of transducer technologies for specific musical functions. In *ICMC*, 2000.

[128] M.M. Wanderley and P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 92(4):632–644, 2004.

[129] Xiao Xiao, Nicolas Audibert, Grégoire Locqueville, Christophe d'Alessandro, Barbara Kuhnert, and Claire Pillot-Loiseau. Prosodic disambiguation using chironomic stylization of intonation with native and non-native speakers. In *Interspeech 2021*, pages 516–520. ISCA, 2021.

[130] Xiao Xiao, Nicolas Audibert, Grégoire Locqueville, Christophe d'Alessandro, Barbara Kuhnert, Rebecca Kleinberger, and Claire Pillot-Loiseau. Évaluation de la stylisation chironomique pour l'apprentissage de l'intonation du français l2. In *34e Journées d'Études sur la Parole (JEP2022)*. AFCP, June 2022.

[131] Xiao Xiao, Grégoire Locqueville, Christophe d'Alessandro, and Boris Doval. T-voks: the singing and speaking theremin. In *NIME 2019 International Conference on New Interfaces for Musical Expression*, pages 110–115, 2019.