PATRICIA: proof-of-concept implementation and validation of a real-time singing synthesizer

Leonardo A. Z. Brum[1*], **Eduardo A. L. Meneses**[2] , **Edward D. Moreno**[3]

[1]LaSiD - Laboratório de Sistemas Distribuídos – Universidade Federal da Bahia
Av. Adhemar de Barros, s/n - Campus de Ondina, Instituto de Matemática e Estatística – 40170-110 Salvador, BA

[2]Metalab – Société des arts technologiques
1201 St Laurent Blvd – H2X 2S6 Montréal, QC, Canada

[3]Programa de Pós-Graduação em Ciência da Computação – Universidade Federal de Sergipe
Av.Marcelo Deda Chagas, s/n – 49107-230 São Cristóvão, SE

leonardobrum@ufba.br, emeneses@sat.qc.ca, edward@dcomp.ufs.br

Abstract. *This paper describes the implementation and validation process of PATRICIA, a proof-of-concept prototype of a system that performs real-time singing voice synthesis (SVS) for the Brazilian Portuguese language. A technological mapping was conducted to study the latest industrial developments in real-time SVS and give directions for PATRICIA design and implementation. The implemented architecture is described, and for the validation process, a series of video recordings of the system working was made as a musical demonstration. This demonstration was subjected to an evaluation by musical educators. A performance analysis with two CPU usage indicators was also performed on two different devices. The results of the validation are discussed and future enhancements are pointed out to overcome the current limitations of the system.*

1. Introduction

The goal of singing voice synthesis (SVS) is to generate singing using computational methods. Regarding the input, SVS systems deal with two types of data: phonetic data, corresponding to the lyrics of the song to be synthesized, and musical data, which provide sound qualities, e.g., pitch and duration, to the synthesized voice. Singing synthesizers can be applied, for example, in the educational field [1, 2]. Digital files containing singing can be easily created, shared, and modified for learning purposes, dispensing with the human presence of a singer as a reference, or even recordings.

Another application of this type of synthesis is in the artistic field itself. Investments in virtual singers, such as Hatsune Miku, have been made in Japan, creating a new cultural phenomenon [3, 4]. In addition, singing voice synthesizers can assist in the process of musical composition, acting as simulators [5]. The game and entertainment industries have also demanded innovations in the field of SVS [6].

The possibilities of use of SVS have been extended by the development of real-time synthesizers. This kind of SVS allows for the use of the singing synthesizer as a musical instrument for live artistic performances. However, only a small number of real-time synthesizers have been developed compared to the universe of SVS, in general [7].

Furthermore, both in industry and academia the absence of tools that provide SVS for the Brazilian Portuguese language was observed. This paper describes the implementation and validation processes of a proof-of-concept prototype of a real-time singing synthesizer called PATRICIA, which was designed for the aforementioned idiom.

This work is organized as follows. Section 2 presents the related work according to the results of a technological mapping, which establishes the state of the technique of the real-time SVS area. Section 3 describes PATRICIA's implementation and architecture. Section 4 presents the experiments conducted to validate the proof-of-concept prototype. Section 5 discusses the results of the validation process. Finally, Section 6 presents a brief conclusion and points out future work.

2. Related work

This work carried out a technological mapping in November 2023, in order to update the results presented at SBCM 2019 [7]. When performing the mapping, we searched for patents related to real-time SVS, to obtain a better understanding of the state of the technique in this area. The research process is described below.

The search for patents related to real-time singing voice synthesizers was carried out in the WIPO (World Intellectual Property Organization) database, called Patentscope[1]. The following search string was used:

```
FP:(FP:(("SINGING SYNTHESIS" OR
"SINGING VOICE SYNTHESIS" OR
"SINGING SYNTHESIZING" ) AND
("REALTIME" OR "REAL-TIME"))).
```

The results were identical to those previously obtained. The search yielded eight patent deposits. All were owned by Yamaha Corporation of Japan and covered an apparatus, method, and storage medium for real-time vocal

*Supported by CAPES.

[1]https://patentscope.wipo.int/

synthesis. However, most of the patents were filed outside Japan, probably to ensure international legal protection.

The patented product was presented as a prototype in a 2012 scientific article [8]. This prototype was called Vocaloid Keyboard and consisted of a musical keyboard with a Vocaloid singing synthesizer [9] as an embedded system, which allowed its user to make an instrumental performance. The presence of Vocaloid as the instrument software indicates that the SVS technical approach used was that of concatenative synthesis. In terms of hardware, an Arduino board was used, at least in the prototyping phase.

The Vocaloid Keyboard prototype had, on its control panel, alphabetic buttons on the left, organized as follows: two horizontal rows with diacritics and consonants, and, below them, five buttons with vowels, organized in the shape of a cross. With the left hand, the user can press these buttons to generate syllables; meanwhile, the musical keyboard can be played with the right hand to provide musical notes. A display with Japanese *katakana* characters shows the generated syllables. A commercial version of this product was released in 2017 as a keytar called VKB-100.

3. Implementation of PATRICIA

To the best of our knowledge, the system presented in this paper is the first singing synthesizer specifically designed for singing in the Brazilian Portuguese language. An outline of the system architecture was proposed in [10], as a future work, and a proof-of-concept prototype was presented in [11]. The synthesizer is called PATRICIA, an acronym for *Programa que Articula em Tempo Real o Idioma Cantado Inscrito em Arquivo* (Portuguese for Program that Articulates in Real-Time the Singing Idiom Written in a File) and was implemented in SuperCollider [12], an environment and programming language for real-time audio synthesis. GitHub was used for version control.

The development of PATRICIA sought to follow the evolutionary software prototyping model, in which the best-understood functional requirements are implemented, as opposed to rapid or throwaway prototyping [13]. Hence, it is expected that the synthesizer, from a phonetic point of view, correctly selects the syllables that make up the lyrics of the song. Musically, what is desired is that the system adequately controls the qualities of sound, such as pitch and duration.

PATRICIA was initially conceived as an embedded system that performs vocal synthesis on a MIDI keyboard to provide real-time performance. This initial conception of PATRICIA found a greater correspondence with the Vocaloid Keyboard system, selected by the technological mapping presented in Section 2. Therefore, a concatenative SVS approach was chosen for the implementation of PATRICIA.

However, for the proof-of-concept prototype, the complexity of the concatenative synthesis technique and the construction of a voice database were transferred to an existing system, the MBROLA speech synthesizer [14], which performs concatenative synthesis. MBROLA was chosen because, in addition to being free software with a wide range of voice databases available, it works through command lines, responding satisfactorily to real-time calls made by the SuperCollider audio server.

The PATRICIA architecture counts on three modules: a musical processor, a phonetic processor, and an audio processor, which are described in detail in this section.

3.1. Musical processor

The MIDI *Note On* message has two data bytes: one of them indicates the MIDI note number, whereas the other provides the velocity, i.e., how hard the key was pressed on the musical instrument. The musical processor module of PATRICIA deals with these two data bytes as input to generate a fundamental frequency and a loudness value on its output.

Equation 1 shows the formula used to calculate the fundamental frequency f_0, in hertz (Hz), for each MIDI *Note On* message, given its corresponding MIDI note number n:

$$f_0 = \sqrt[12]{2^{n-81}} \cdot 440 \tag{1}$$

In *Note On* messages, velocity varies between 1 and 127. Since the loudness of the audio depends on the force used to play the instrument, the velocity parameter can be mapped to an intensity measured in decibels (dB), a logarithmic scale unit. The specifications of the MIDI protocol establish the formula shown in Equation 2 to map a velocity V, which varies between 1 and 127, into a loudness L, measured in decibels:

$$L = 40 \cdot log(V/127) \tag{2}$$

The formulas presented by Equations 1 and 2 were implemented in the musical processor module, and their results were attributed as inputs to PATRICIA's audio processor.

3.2. Phonetic processor

In the Vocaloid Keyboard prototype, both phonetic and musical inputs are given in real time by the user. This kind of performance was possible because this product was designed for the Japanese language, whose syllables structures are simpler than those of western languages. Most Japanese syllables consist of a consonant and a vowel, in that order. Thus, when the user presses, for example, the buttons "T" and "A" simultaneously on the Vocaloid Keyboard panel, the synthesis engine necessarily interprets this as a "TA" since an eventual "AT" syllable does not exist in Japanese [15]. In contrast, Brazilian Portuguese has syllables whose structure increases the challenges of using manual controllers to map a real-time phonetic input.

However, even the commercial version of Vocaloid Keyboard renounces this phonetic input method

to give song lyrics in advance [16]. Thus, giving a previously prepared phonetic input in a text file appeared to be the most suitable method for the implementation of a Brazilian Portuguese real-time SVS system.

Once the phonetic input method has been determined, it is necessary to discuss the format of such input. Syllables could be written according to the official Portuguese orthography, but a phonetic notation was preferred to give the user two types of control over the voice synthesis:

1. The lyrics of the song are not a text in prose, but in verse, so certain phenomena that occur in versification, such as elision, where two syllables merge into one, must be taken into account. The syllables of the verses do not necessarily coincide with the grammatical ones [17] and are better controlled at the phonetic level.
2. Brazilian Portuguese has several regional variants. For example, the first syllable of the word 'tia' (Portuguese for 'aunt') can be pronounced as [ti] or [tʃi][2], depending on the region of the Brazilian speaker [18], so that the synthesis mechanism would transform an orthographic syllable 'ti' on the system input into a specific phonetic variant chosen arbitrarily, excluding other possibilities.

In PATRICIA, each line of the lyrics text file must have a syllable written in phonetic notation with its phonemes separated by hyphens. The phonetic notation used is similar to that of SAMPA [19], which uses only ASCII characters. It is possible to alternate among distinct text files to synthesize several songs. The selection of phonetic files is made using the *Program Change* MIDI message, allowing the system to handle up to 128 different song lyrics. Once a lyrics text file is selected, for each time a *Note On* message is received, the phonetic processor reads a line from this file and retrieves a syllable. The correspondence between a syllable and a musical note is the basis of singing synthesis.

The phonetic processor also indicates which MBROLA voice database must be used. The selection of the voice databases is made by the MIDI channel number, which can assume values between 1 and 16. MIDI channel changes can be made in real time, allowing the performer to use up to three voice timbres while playing a single song. The feasibility of this change depends on how MIDI controls are disposed on each device or musical instrument. Although this means that PATRICIA supports up to sixteen different MBROLA databases, only four of them are available for the Brazilian Portuguese language. They are called *br1*, *br2*, *br3*, and *br4* and consist of voice samples controlled by two different sets of phonetic symbols. The set used by the databases *br1*, *br2* and *br3* is different from the *br4* set.

[2]The International Phonetic Alphabet (IPA) notation was used in these given examples.

3.3. Audio processor

As mentioned, PATRICIA uses the synthesis engine of an external program, a speech synthesis system called MBROLA. This synthesizer performs a concatenative synthesis by diphones, that is, a conjunction of two phonemes. A set of diphones of a given language is recorded and stored in a file that serves as a voice database for the system. For its turn, the speech to be synthesized can be stored in a text file with the *.PHO* extension. Each line of this file contains a phoneme written according to the set of phonetic symbols used, a duration in milliseconds, and a series of pitch targets composed of two numbers: the position of the target within the phoneme, which is a percentage of its total duration, and the frequency value in hertz at this position.

For each MIDI note and its corresponding syllable, it is generated a *.PHO* file. Thus, each line of this file contains these four parameters:

1. A phoneme from the syllable retrieved by the phonetic processor (the *.PHO* file will have as many lines as phonemes in this syllable).
2. A phoneme duration, set as 50 ms for semivowels and consonants, and 200 ms for vowels.
3. A pitch target position that is always 100%, which means that the pitch of the phonemes will be the same throughout their duration.
4. A fundamental frequency to provide the pitch, calculated by the musical processor according to Equation 1.

Once the *.PHO* file was created, PATRICIA uses the SuperCollider engine to send an MBROLA command line to the operating system of the host computer. This instruction references both the *.PHO* file and the Brazilian Portuguese MBROLA voice database selected by the phonetic processor. This creates the corresponding audio file in *AU* format.

Since each audio file generated by MBROLA has a constant duration, PATRICIA needs to extend it as long as the respective key is pressed in the MIDI instrument. To achieve this result, for each active MIDI note, the corresponding *.AU* file is played in a loop. The start position of the loop is in the middle of the file in the time domain, where the vowel is. The ending position is calculated by adding the period in seconds, i.e., the inverse of the fundamental frequency, to the starting position. The interval between the start and ending positions corresponds to one cycle or pitch period of the waveform. This cycle is repeated while the corresponding MIDI note is kept active, extending the duration of the syllable. When a MIDI Note off message is received, the loop that corresponds to this note is released, and the rest of the *.AU* file is played.

Figure 1 shows a partial view of a waveform generated by MBROLA for the first syllable of the word 'quando' singing a G_4 note. The region selected in (a) corresponds to a period of the waveform calculated by the audio processor, starting from the center of the sample. Repetition of such a segment, which appears seven times in (b), extends the duration of the audio.

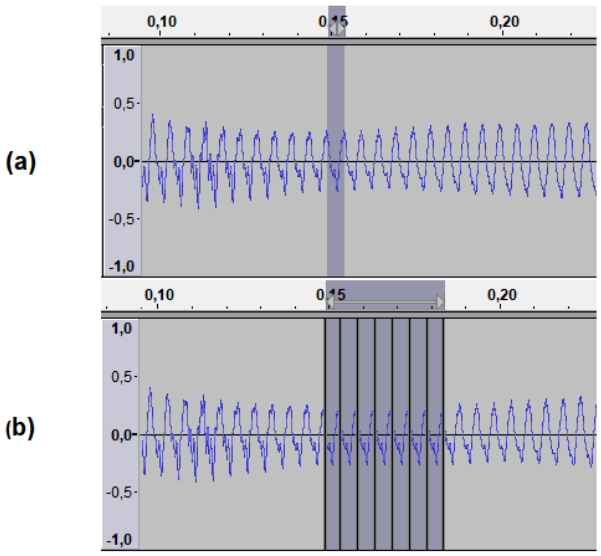


Figure 1: (a) Loop segment calculated by PATRICIA. (b) Repetition of the segment, extending the duration of the sample.

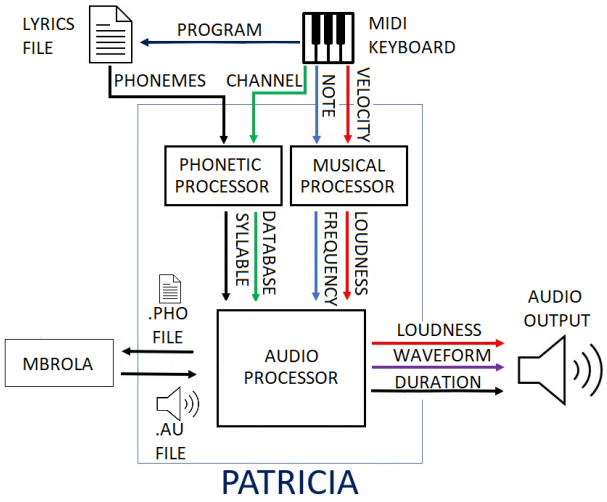


Figure 2: PATRICIA system architecture diagram.

The differences between the sampling rates of the MBROLA and SuperCollider databases, that is, 16000 Hz and 44100 Hz, respectively, had to be considered for the audio to be reproduced correctly. If the MBROLA audio, whose sampling rate is lower, were reproduced at the standard SuperCollider rate, the result would be an accelerated reproduction and an increase in the fundamental frequency of the audio signal itself, which would sound higher pitched. In order to maintain the proportion between both sampling rates, the audio processor defined a reproduction rate corresponding to the division between them, that is, 16000/44100 Hz.

The successive repetition of the process described above results in the generation of a synthesized singing voice in the audio output of the host computer. An amplification is applied to this output according to the loudness value calculated by the musical processor as shown in Equation 2.

The PATRICIA system architecture diagram is shown in Figure 2, which highlights the interaction between the internal modules of PATRICIA and its external components, i.e., the input sources, the MBROLA synthesizer, and the audio output.

4. System validation

As a proof of concept, the singing synthesizer described in this work has only a minimum set of functional requirements implemented. These requirements are validated by the experiments described in the following, which sought to evaluate the musical capabilities of the system and to perform a comparative analysis between its performance on a personal computer and on a single-board computer.

4.1. Musical demonstration

A brief demonstration of PATRICIA's musical capabilities was recorded, and the resulting videos are available on the system's YouTube channel [3].

The videos show PATRICIA running in the SuperCollider IDE. At the same time, a picture-in-picture image shows the MIDI keyboard being played at a mirrored angle, while the Audacity audio editor records the output signal. A different voice database was used in each of the videos.

In another recording, it was possible to demonstrate the potential educational use of the PATRICIA system. An arrangement for a three-voice choir of the first two verses of the song "Asa Branca", by Luiz Gonzaga and Humberto Teixeira, was made. PATRICIA was used to generate the audio for each of the voices, using the *br1* database for the tenor, who performs the main melody, *br2* for the contralto voice, and *br3* for the bass. The audios, in the Audacity editor, are played separately and then together in another video on the project channel. When teaching choral singing, it is common for the teacher or conductor to present the melodic lines of each voice to the choir members using a musical instrument, such as a piano or keyboard. Using the PATRICIA synthesizer, it will be possible to use a musical instrument to generate not only the melodic line but also the audio of the voice itself to be sung. Three of the songs chosen for the video recordings were taken from Brazilian children's folklore. The fourth song is the aforementioned "Asa Branca", which is part of Brazilian northeastern culture. A summary of the recordings is presented in Table 1.

4.2. Evaluation by music educators

The five musical demonstrations described in the previous subsection were made available for a subjective evaluation by five music educators who work in the State of Sergipe, Brazil.

The evaluation was carried out using a form that contained 20 questions, four for each demonstration,

[3] Available at https://www.youtube.com/@ProjetoPATRICIA

Table 1: Video recordings for demonstration.

Video number	Song title	MBROLA database	Duration
1	O cravo brigou com a rosa	*br1*	56”
2	Atirei o pau no gato	*br2*	35”
3	Terezinha de Jesus	*br3*	55”
4	Asa branca	*br4*	1’35”
5	Asa branca (three voices)	*br1*, *br2*, and *br3*	1’18”

which the educator had to watch before responding. For the demonstrations corresponding to videos 1 to 4, according to Table 1, the four questions were as follows:

1. Do you consider the sung syllables to be understandable?
2. Do you consider the generated song to be natural?
3. Do you consider that the musical pitches generated correspond to the musician’s performance?
4. Do you consider that the duration of the generated notes corresponds to the musician’s performance?

These four questions aim to evaluate, respectively, the following psychoacoustic characteristics of the generated song: intelligibility [20], that is, not only the correct selection of syllables from a technical point of view, but their effective understanding; naturalness [21], which would be the evaluation of how close the synthesized voice would be to a natural human voice; pitch and duration control according to the keyboard performance watched on the videos. The possible answers to the questions, corresponding to a Likert scale [22] with four options, giving up a neutral position, are shown in Table 2.

Table 2: Likert scale score for each answer

Answer	Score
I totally agree	4
I partially agree	3
I partially disagree	2
I totally disagree	1

Regarding the demonstration of video 5, the last two questions were modified. The penultimate of them asked, “Do you consider that the singing generated corresponds to the arrangement shown in the above score?”, considering the score used for the three-voice arrangement. Finally, the last question aimed to evaluate the educational usefulness of the synthesizer, posing the following question to educators: “Do you think that the PATRICIA synthesizer would be a useful tool for teaching choral singing?”.

The average of the scores given by each of the five users in each question corresponds to a Mean Opinion Score (MOS) [23] for each requirement. Tables 3 to 7, below, present the results obtained for each of the demonstrations. In them, educators are identified as E1... E5 and the questions as Q1...Q4.

Table 3: Evaluation of video 1 demonstration.

	E1	E2	E3	E4	E5	MOS
Q1(intelligibility)	2	4	3	3	3	3
Q2(naturalness)	2	3	2	1	1	1.8
Q3(pitch control)	3	4	4	4	3	3.6
Q4(duration control)	3	4	4	3	3	3.4

Table 4: Evaluation of video 2 demonstration.

	E1	E2	E3	E4	E5	MOS
Q1(intelligibility)	3	4	3	3	3	3.2
Q2(naturalness)	2	3	3	2	1	2.2
Q3(pitch control)	3	4	4	4	4	3.8
Q4(duration control)	3	4	4	4	4	3.8

Table 5: Evaluation of video 3 demonstration.

	E1	E2	E3	E4	E5	MOS
Q1(intelligibility)	2	3	2	3	1	2.2
Q2(naturalness)	2	3	2	1	1	1.8
Q3(pitch control)	2	4	4	4	4	3.6
Q4(duration control)	3	4	4	4	4	3.8

Table 6: Evaluation of video 4 demonstration.

	E1	E2	E3	E4	E5	MOS
Q1(intelligibility)	3	4	3	3	3	3.2
Q2(naturalness)	3	3	2	1	1	2
Q3(pitch control)	3	4	4	4	4	3.8
Q4(duration control)	2	4	4	3	3	3.2

Table 7: Evaluation of video 5 demonstration.

	E1	E2	E3	E4	E5	MOS
Q1(intelligibility)	3	3	3	3	3	3
Q2(naturalness)	3	3	3	2	1	2.4
Q3(conformity to the arrangement)	3	4	4	4	4	3.8
Q4(educational usefulness)	4	4	4	3	2	3.4

4.3. Performance analysis

The experiments aimed at analyzing the performance of the system were conducted on two devices: a personal computer and a Raspberry Pi single-board computer. Table 8 presents, in greater detail, the specifications of these computers. The musical keyboard used in these experiments was the M-AUDIO Keystation Mini32 MK3 MIDI controller.

Table 8: Specifications of the computers used in experiments.

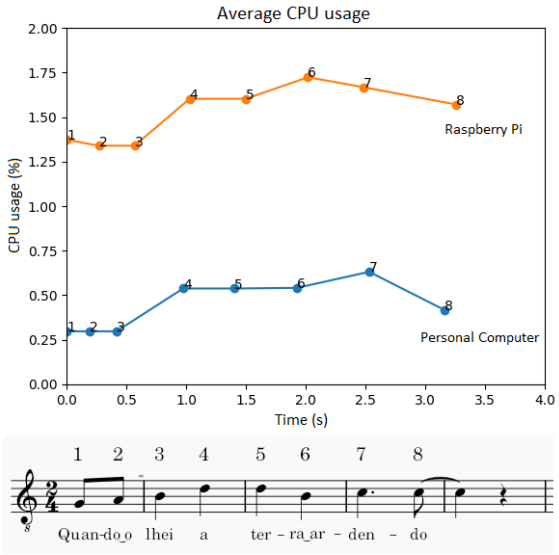
Computer	Operating system	CPU	RAM size
Positivo Master N8140 Blackstone Laptop	Windows 11 Pro v.22H2 64 bits	Intel Core i5-7300U CPU @ 2.60GHz	8 GB
Raspberry Pi 4 Model B Rev 1.5	Raspberry Pi OS Debian 11 (bullseye) 64 bits	Broadcom BCM2711 SoC @ 1.8GHz	8 GB

The proposed comparative analysis is based on the measurement, on each device, of the following parameters related to the SuperCollider audio server, the scsynth process:

- Average CPU usage: represents the average CPU load over time while the audio server is running. The average CPU usage by the audio server is calculated over a given time interval. This is a useful metric for evaluating the server's overall performance.
- CPU usage peaks: indicates the highest CPU usage peak while the audio server is running, recording the maximum value reached by the CPU load while performing audio processing. This measurement can be used to identify moments when the CPU load reaches critical levels or when peaks in processing demand occur.

Figure 3 shows the average CPU usage measurements by the audio server on each of the devices, indicated in the graph by their respective operating systems. The performance of the first verse of the song "Asa Branca", in 120 bpm, was chosen to complete the experiment. The various points represent the moments in which each note was played and are numbered from 1 to 8 so that they correspond to the score placed at the bottom of the graph. A larger number of points could be generated to cover a longer duration of the musical performance, but the note-to-note correspondence between the two performances would be less clear, and the score would tend to either become illegible or take up too much space.

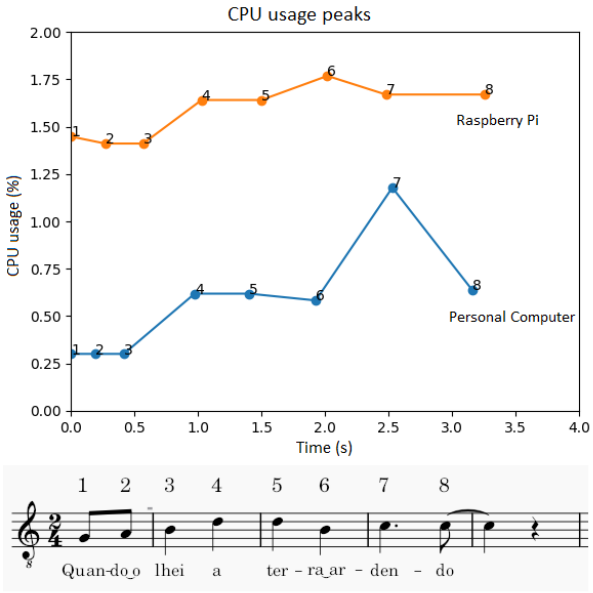
Figure 3: Average CPU usage on each device.



It was observed that, although the average CPU usage was higher on the Raspberry Pi than on the personal computer, on both devices the variation of the measurement over time behaved similarly. On the Raspberry Pi, the average CPU usage of the SuperCollider audio server varied within the range of 1.25 to 1.75%, while the variation on the personal computer occurred within the range of 0.25 to 0.75%.

In turn, Figure 4 presents measurements of CPU usage peaks by the scsynth process on both computers, relating them to the musical notes present in the score displayed at the bottom, similar to the pattern adopted in Figure 3.

Figure 4: CPU usage peaks on each device.



Here, the behavior of the graph was similar to that of the previous measurement. The CPU usage peaks were more intense on the Raspberry Pi, but the variation in measurement within the level presented by each device was similar, except for a higher CPU usage value observed when the seventh note was activated on the personal computer. The CPU usage peaks varied between 1.25 and 1.75% on the Raspberry Pi and between 0.25 and 1.25% on the personal computer. It is believed that exploring the parallelism features of SuperCollider can improve the synthesizer's performance on the Raspberry Pi.

5. Discussion of the results

The experiments carried out were able to validate the current version of the synthesizer as a proof-of-concept. Real-time control of the basic sound qualities – pitch, intensity, duration, and timbre – as well as the correct selection of diphones proved to be feasible, as they could be verified both aurally and visually by analyzing the behavior of the audio signal generated in the Audacity editor as the songs were played. It is possible to verify that even the imprecisions in the performance were reflected in the audio output, which is a desired characteristic when dealing with a real-time synthesizer.

Regarding subjective evaluation by music educators, the requirement that received the least favorable evaluation was naturalness. The main cause of this is probably

PATRICIA's dependence on MBROLA, which was not designed to synthesize the singing voice. Other causes are the absence of voice expression control and a duration extension algorithm that generates a very short loop segment, covering only one period of the synthesized voice waveform. In addition, click noise was noticeable in some syllabic transitions.

Regarding the intelligibility of the generated syllables, most of the evaluations were positive, but this may be because, most likely, the music educators already knew the lyrics of the evaluated songs beforehand. Furthermore, the expected control at the phonetic level of the synthesized singing was not at all possible, given the configurations of the voice databases used, since, at least in the cases of *br1* and *br3*, some diphones, such as [ti] and [di], had already been programmed to correspond to a specific regional variant of the Brazilian Portuguese language, resulting in the palatalized syllables [tʃi] and [dʒi], respectively.

The control of pitch and duration was also well evaluated by the educators, including in relation to the demonstration of three-voice singing, in which these parameters were evaluated together under the name of "conformity to the arrangement". The educational usefulness, in turn, divided opinions, although three of the five educators stated that they fully agreed that the PATRICIA synthesizer would be a useful tool for teaching choral singing.

Finally, both measurements of system performance described in Section 4 resulted in very low values of CPU usage and, consequently, quite satisfactory performance. This result indicates that the choices, in terms of software, were correct: SuperCollider proved to be quite efficient in real-time audio processing, and MBROLA responded well to calls made by the server. However, in terms of hardware, the experiments lead us to believe that the synthesizer could be implemented in a device with much lower computing power, making the solution cheaper, especially when its final version is developed as an embedded system.

6. Conclusion and future work

The system proposed here, PATRICIA, is a real-time vocal synthesizer and the first to be specifically designed to generate vocals in Brazilian Portuguese. This is a work in progress, and the synthesizer described is a proof of concept with a limited set of functional requirements implemented in the SuperCollider language and environment.

Concatenative synthesis and lyrics provided in advance in a text file proved to be, respectively, the most suitable technical approach and phonetic input method for the system, as discussed. Musical input is given in real time via MIDI, and synthesis is performed by MBROLA, a speech synthesizer.

For future versions of the system, the implementation of a series of modifications is planned.

- The creation of PATRICIA's own voice database with better audio quality. The samples will be recorded by a female professional singer instead of using speech samples. This will improve the naturalness and intelligibility of the synthesized singing voice, overcoming the dependency on the MBROLA databases.
- The implementation of an SVS algorithm in PATRICIA's code is another factor contributing to the independence of this system from MBROLA. The use of deep learning methods and techniques can bring the system closer to the state of the art in SVS [24].
- More comprehensive treatment of MIDI messages in order to provide the system with artistic expression controls.
- Optimization of the synthesizer performance by using the supernova process instead of scsynth, which will allow to exploit the parallelism capabilities of the device's multicore processor.
- The integration between PATRICIA and a MIDI keyboard will allow the synthesizer to work as an embedded system for the instrument. In the current version, the keyboard acts as an external peripheral input to the system.
- Transformation of PATRICIA into an augmented instrument, that is, one equipped with controls that go beyond the conventional technique by which it would be played. In such a context, gestural controls that allow phonetic input to also occur in real time could be implemented in PATRICIA, which would be beneficial for its use in more artistic environments.

Among the challenges that will be faced with future implementations, the following can be mentioned:

- The discussion of the patents of the Vocaloid Keyboard, which is a similar product and property of Yamaha Corporation.
- The construction of a voice database from recordings of a singer will raise ethical questions about human research.
- The systematization of tests that allow an evaluation of the product by artists who, when using it, will validate the functional requirements considered to be more subjective.
- The establishment of a hardware platform that is more suitable, in terms of cost/benefit, for the implementation of the synthesizer.

Finally, it can be asserted that PATRICIA is a system classified in a research area with few developments, real-time SVS, and boasts a pioneering feature: it is designed for the Brazilian Portuguese language. Consequently, its limitations do not diminish its significance as a valuable contribution to Computer Music. Moreover, it opens up a series of possibilities and challenges for future research.

7. Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] Anastasia Georgaki. Virtual voices on hands: Prominent applications on the synthesis and control of the singing voice. In *Journées d'informatique musicale*, Paris, France, 2004.

[2] Rong Li. Intelligent analysis of music education singing skills based on music waveform feature extraction. *Mobile Information Systems*, 2022(1):9747342, 2022.

[3] Kimi Kärki. Vocaloid liveness? hatsune miku and the live production of japanese virtual idol concerts. In *Researching Live Music*, pages 127–140. Focal Press, 2021.

[4] Jessica Tsun Lem Hui. Reconfiguring voice in the end: Virtuosity, technological affordance and the reversibility of hatsune miku in the intermundane. *Cambridge Opera Journal*, 34(3):364–379, 2022.

[5] Hideki Kenmochi. Singing synthesis as a new musical instrument. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5385–5388, 2012.

[6] Ji-Sang Hwang, Sang-Hoon Lee, and Seong-Whan Lee. Hiddensinger: High-quality singing voice synthesis via neural audio codec and latent diffusion models. *Neural Networks*, 181:106762, 2025.

[7] Leonardo A. Z. Brum and Edward D. Moreno. State of art of real-time singing voice synthesis. In *Anais do XVII Simpósio Brasileiro de Computação Musical*, pages 50–57, Porto Alegre, RS, Brasil, 2019. SBC.

[8] Shota Kagami, Keizo Hamano, Kazuki Kashiwaze, and Kazuhiko Yamamoto. Development of realtime japanese vocal keyboard. *Information Processing Society of Japan INTERACTION*, pages 837–842, 2012.

[9] Hideki Kenmochi and Hayato Ohshita. VOCALOID - commercial singing synthesizer based on sample concatenation. In *Proc. Interspeech 2007*, pages 4009–4010, 2007.

[10] Leonardo A. Z. Brum and Edward D. Moreno. Challenges and perspectives on real-time singing voice synthesis. *Revista de Informática Teórica e Aplicada*, 27(4).

[11] Leonardo A. Z. Brum, Eduardo A. L. Meneses, and Edward D. Moreno. Patricia: a real-time singing synthesizer prototype for the brazilian portuguese language. In *Proceedings of the International Computer Music Conference 2023*, page 176 – 180, 2023.

[12] James McCartney. Supercollider: a new real time synthesis language. In *Proc. International Computer Music Conference (ICMC'96)*, pages 257–258, 1996.

[13] Linda Sherrell. Evolutionary prototyping. *Encyclopedia of Sciences and Religions*, pages 803–803, 2013.

[14] 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, 1996.

[15] Haruo Kubozono. The mora and syllable structure in japanese: Evidence from speech errors. *Language and Speech*, 32(3):249–278, 1989.

[16] Kazuki Kashiwase. An over-the-shoulder keyboard that extends the potential for vocaloid performance. *Yamaha Corporation*, 2017. Accessed: 2023-01-29.

[17] Manuel Bandeira. A versificação em língua portuguêsa. *Enciclopédia Delta Larousse*, pages 3239–3249, 1960.

[18] Thaïs Cristófaro Silva and Camila Leite. Padrões sonoros emergentes:(oclusiva alveolar+ sibilante) no português brasileiro. *Caderno de Letras*, (24):15–36, 2015.

[19] John C Wells et al. Sampa computer readable phonetic alphabet. *Handbook of standards and resources for spoken language systems*, 4:684–732, 1997.

[20] Christian Benoît, Martine Grice, and Valérie Hazan. The sus test: A method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences. *Speech Communication*, 18(4):381–392, 1996.

[21] Katherine Morton. Naturalness in synthetic speech. *Proceedings of the Institute of Acoustics*, 12(Part 10):125–132, 1990.

[22] Ankur Joshi, Saket Kale, Satish Chandel, and D Kumar Pal. Likert scale: Explored and explained. *British journal of applied science & technology*, 7(4):396, 2015.

[23] Robert C Streijl, Stefan Winkler, and David S Hands. Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. *Multimedia Systems*, 22(2):213–227, 2016.

[24] Peng Bai, Meizhen Zheng, and Xiaodong Shi. A survey of singing voice synthesis. In Kun Qian, Xin Wang, Qinglin Meng, and Mingzhi Chen, editors, *Proceedings of the 10th Conference on Sound and Music Technology*, pages 19–30, Singapore, 2025. Springer Nature Singapore.