Neuroscience Informatics 2 (2022) 100041

Contents lists available at ScienceDirect

Neuroscience Informatics

www.elsevier.com/locate/neuri

Original article

Electroglottography based real-time voice-to-MIDI controller

Eugenio Donati *,[1], Christos Chousidis

School of Computing and Engineering, University of West London, St. Mary's road, W55RF, London, UK

ARTICLE INFO

Article history:
Received 1 October 2021
Received in revised form 7 January 2022
Accepted 10 January 2022

Keywords:
Electroglottography
Bioimpedance measurements
EGG-to-MIDI
Voice-to-MIDI
Voice information retrieval
Real-time audio conversion

ABSTRACT

Voice-to-MIDI real-time conversion is a challenging problem that comes with a series of obstacles and complications. The main issue is the tracking of the human voice pitch. Extracting the voice fundamental frequency can be inaccurate and highly computationally exacting due to the spectral complexity of voice signals. In addition, on account of microphone usage, the presence of environmental noise can further affect voice processing. An analysis of the current research and status of the market shows a plethora of voice-to-MIDI implementations revolving around the processing of audio signals deriving from microphones. This paper addresses the above-mentioned issues by implementing a novel experimental method where electroglottography is employed instead of microphones as a source for pitch-tracking. In the proposed system, the signal is processed and converted through an embedded hardware device. The use of electroglottography improves both the accuracy of pitch evaluation and the ease of voice information processing; firstly, it provides a direct measurement of the vocal folds' activity and, secondly, it bypasses the interferences caused by external sound sources. This allows the extraction of a simpler and cleaner signal that yields a more effective evaluation of the fundamental frequency during phonation. The proposed method delivers a faster and less computationally demanding conversion thus in turn, allowing for an efficacious real-time voice-to-MIDI conversion.

1. Introduction

The human voice represents the most primordial musical instrument of all and the most common mean of musical expression throughout history. With the development of the Musical Instrument Digital Interface (MIDI) protocol in the mid 1980s, numerous devices were developed to deliver MIDI messages to electronic instruments whilst emulating the physical and mechanical characteristics of classical musical instruments. This development spawned a range of opportunities for musicians to apply their instrumental technique to the control of analogue and digital sound generators. However, throughout the years, the development of such tools based on the singing voice produced mainly unreliable and unsatisfying results. As in human voice the sound is generated within the body, the use of such "instrument" presents a lack of manual control compared to instruments such as keys or strings. This forces the process to be based on the real-time conversion of microphonic audio signals which represents a challenge even for advanced technologies and fast processors. A voice-to-MIDI conversion process, thus, is subject to many errors and imprecisions due to the complexity of the operation and the effect of external noise elements. Firstly, the fundamental frequency evaluation of an audio signal in real-time requires computationally expensive procedures. Secondly, the use of voice-to-MIDI converters in noisy environments would generate errors due to the sound contamination caused by external sources. Throughout the years many applications attempted to propose solutions to these issues by employing for instance statistical predictions systems [1] or by reducing the computational demands of the pitch tracking technique [2] and [3].

This paper proposes a solution for the development of a voice-to-MIDI real-time converter that, based on the previous research shown in [4], [5] and [6], lays its functioning principles in the use of Electroglottography (EGG). The advantage of this approach resides in the EGG delivering a much simpler signal as opposed to microphones. This can be observed in the comparison between an EGG and its audio counterpart as shown in Fig. 1. In the time domain, the EGG waveform presents itself much close to a sinusoidal wave whilst the audio results are more complex. In the frequency domain, the audio presents a lot more harmonic content when compared to EGG. As EGG evaluates the movement of vocal folds, it bypasses the vocal tract, omitting, in turn the added harmonics. Such simplicity makes the fundamental frequency extraction from EGG more computationally sustainable. Moreover, as the EGG

* Corresponding author.
E-mail addresses: Eugenio.donati@uwl.ac.uk, eugenio.donati@gmail.com (E. Donati), Christos.chousidis@uwl.ac.uk (C. Chousidis).
[1] The author conducted the research as part of a PhD at the University of West London under the Vice Chancellor Scholarship Scheme.

https://doi.org/10.1016/j.neuri.2022.100041

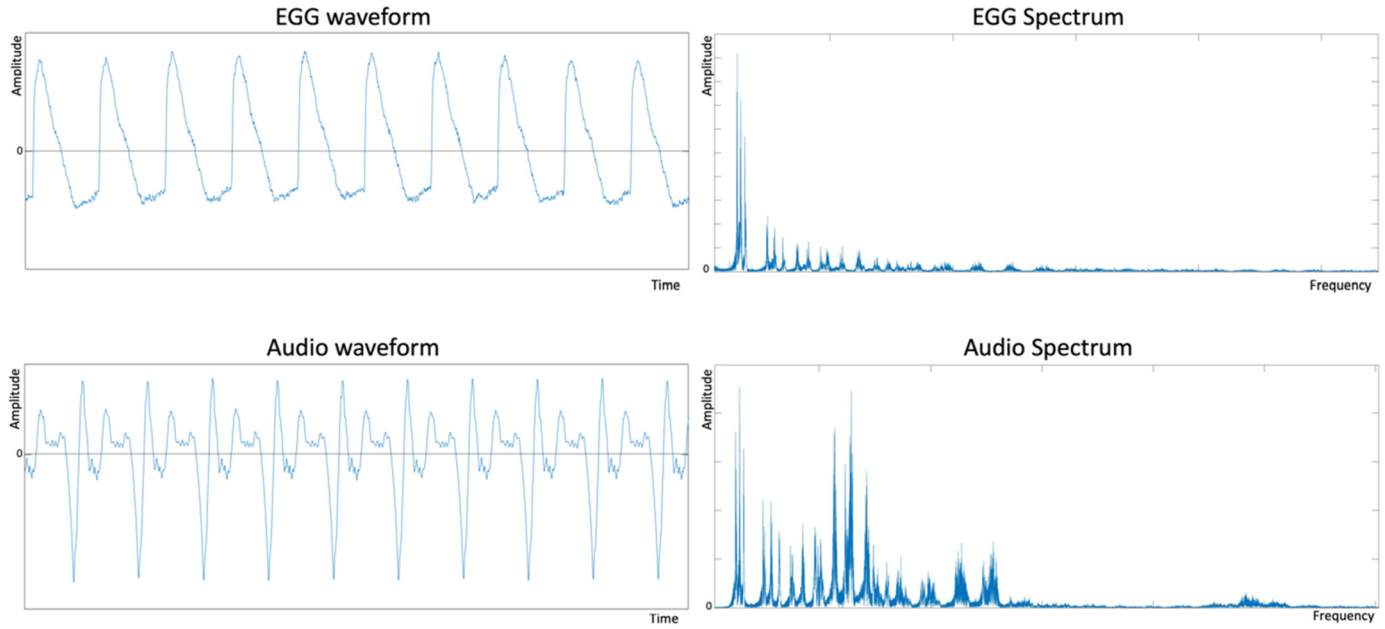

Fig. 1. Comparison between EGG and microphone in both time and frequency domains.

signal is originated directly from the vocal folds, it bypasses any interference created by external noises elements and sound sources making it non-susceptible to sound contamination.

2. State-of-the-art

The main challenge behind the conversion of audio into MIDI is the evaluation of the fundamental frequency, or f_0, that can result inaccurate and computationally expensive. This process, also known as pitch tracking, is even more challenging when applied in real-time as the computation of f_0 can take up to $\sim$40 milliseconds, depending on the computational platform, and can cause an audible delay between audio and the resulting MIDI.

A method to improve the accuracy and speed of pitch tracking is proposed by Arvin and Doraisamy [7] using a peak detection approach to avoid complex calculations such as FFT. The authors apply a windowing approach to the peak detection pitch tracking method proposed in [8]. The employed window size is of 100 ms which results in noticeable latency for real-time application. The research also highlights how the accuracy of the system heavily relies on high-quality microphones and silent environment to obtain accurate readings. Another method is suggested in [9] where an autocorrelation approach is combined with probabilistic models. The research yields an accurate conversion with a latency of about 12 ms which is acceptable for real-time performances. However, despite the accurate and fast conversion, the system is applied only on pre-recorded guitar sound, reduced in sampling rate to reduce complexity. In addition, the system was not tested with a real-time audio input. The use of pre-recorded sounds highly affects the overall performance as it bypasses the real-time processing of live inputs. The specific conversion of voice into MIDI presents even more challenges because of the many ways in which voices can differ. In [1], where a voice-to-MIDI conversion is discussed, the f_0 evaluation is performed with an autocorrelation method combined with a musical probabilistic method for the likelihood of certain notes appearing. The system yielded good results on long sung note sequences with a total error below 9%. However, it was run on a database of recordings and not tested on real-time applications. Another novel method for voice-to-MIDI conversion was developed in [3]. The proposed approach employs a novel correlation function known as Correntropy. This system results showed accurate performances with low latency but similarly to the previous it was tested only on recorded data. The state of the research of both audio and voice-to-MIDI shows the process heavily depending on processing power and silent environments. Even where a fast, non-complex approach can be used for pitch tracking, the addition of external noise and sound sources can severely affect the performance delivering inaccurate results. Moreover, the need for high-quality microphones affects the latter problem even further.

Despite the lack of literature for audio-to-MIDI and voice-to-MIDI converter specifically for real-time applications, several platforms are available on the market. The *A2M* plugin produced by Beatbar offers a real-time audio-to-MIDI to any audio input, including microphones, offering a choice of either FTT or autocorrelation. *A2M* offers an option to balance between latency and accuracy by changing the input buffer size. This affects the outcome as a smaller buffer size delivers lower latency but poorer accuracy and vice versa. Moreover, the system does not support any pitch-bended input, and therefore any sung note will be quantised to the standard frequency of the chromatic musical scale. As musical notes in the chromatic scale are defined with specific frequency values, if a singer produces a note slightly off the standardised value the system is unable to deliver the exact frequency. Another example of voice-to-MIDI conversion on the market is *Doubler2* from Vochlea. This system is designed to lock the MIDI output in a specific musical key. As the distance between musical notes is of several hertz, pitch tracking becomes easier as the needed accuracy decreases, and the input buffer size can be lowered. The developers of *Doubler2*, moreover, specify that the use of dynamic microphones is necessary and suggests the use of an especially designed microphone. Such information suggests how a low sensitivity microphone and a silent environment are also needed for satisfactory performance.

3. Overview of phonation and EGG

As mentioned in chapter 1, the EGG delivers a signal by analysing the movement of vocal folds. This characteristic is the reason why EGG can deliver a simpler signal when compared to the audio from a microphone. The foundation of this difference resides in the very principles of voice production.

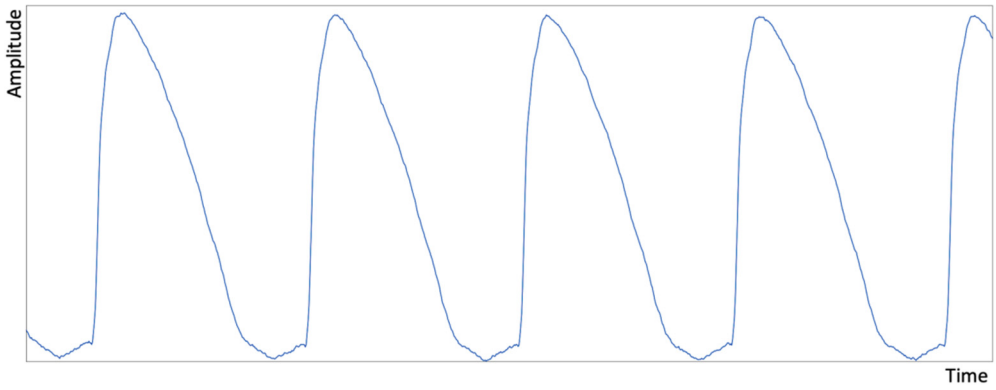

Fig. 2. Generic EGG signal.

3.1. Phonation

In human phonation, the vocal folds are stimulated by a flow of air coming from the lungs through the trachea and create a vibrational cycle. The folds move repetitively from a position of contact to one of non-contact and act as a conversion mechanism transforming the kinetic energy of the airflow into acoustic energy [10]. As the vibrational cycle is periodic in nature, the number of cycles per second effectively defines the frequency of oscillation. This represents the fundamental frequency of the produced voice [6]. This signal, then, reaches the vocal tract where the various physiological elements resonate and in turn add to the original sound multiple harmonics. This latest stage of phonation is what defines the perceived distinguishable voice sound.

3.2. EGG

EGG evaluates the behaviour of vocal folds by applying an electrical current across the larynx. By means of a pair of electrodes, a low voltage and high frequency AC signal is applied across the larynx cartilage. The oscillation of the folds causes a change in the distance between them that, in turn, causes a change in the level of bioelectrical conductance (and in turn impedance) across the larynx [11]. This effectively performs an Amplitude Modulation (AM) on the originally applied signal. By demodulating the resulting AM signal, the system can determine the modulating frequency [12], [13]. The result is a sinusoidal-like signal that represents the succession of the vibrational cycles and, thus, the frequency of oscillation [14].

Because of its functioning, EGG can be particularly efficient in the evaluation of the fundamental frequency. By being in direct association with the vocal folds, it performs the reading before any resonance is added by the vocal tract and, thus, bypasses the consequent added harmonics [15]. EGG, however, is not as effective for tracking the amplitude of the sound as this depends on the pressure level of the airflow and not on the vocal folds' speed of vibration. As shown in Fig. 2, the resultant signal presents itself as much simpler in comparison to that of a recorded voice sound.

4. A system for EGG-to-MIDI conversion

This paper proposes a new approach for the conversion of voice into MIDI by using EGG as a signal source. Due to the nature of voice, however, to achieve a device capable of successfully convert voice information into MIDI some specific issues must be addressed. The following sections will go through the details of said issues describing how these were addressed and combined to achieve a functioning prototype. The system developed in this project was implemented using the visual programming language *Pure Data* [16], [17].

4.1. Amplitude and frequency tracking

Human voice is a unique instrument and, as such, its parameters are controlled in a unique way. Whereas in musical instruments the control over these aspects is related to the mechanics of a physical object, in singing those are dictated "internally" through physiological elements. Due to the lack of external mechanics, both amplitude and frequency tracking become essential for the system to detect a note and its amplitude envelope stages.

Pitch and envelope tracking are both performed in *Pure Data* through the [sigmund~] object (Fig. 3). [sigmund~] performs an FFT based fundamental frequency evaluation delivering f_0 in the form of MIDI note numbers and an RMS estimation of the signal amplitude over time. The real-time RMS estimation of the amplitude allows the system to evaluate whether the sound is in its steady state or in an attack/release state. As EGG cannot deliver accurate amplitude readings by nature, a binary approach is used where velocity values of 0 and 100 are used for *note-off* and *note-on* messages respectively. A threshold value is then set to determine the steady state and the binary value of velocity is triggered accordingly (Fig. 3); the threshold value was set through a process of tests and empirical observations.

4.2. MIDI allocation

As seen in section 3.1, human voice deprives a system such as this, of the possibility of using physical controls. This peculiarity could cause the wrong allocation of MIDI messages if they were to be allocated only according to the steady and non-steady states of the envelope. When performing a *legato*, for instance, a singer is effectively producing multiple pitches over one continuous cycle of phonation. In other words, the pitch variation effectively happens across the steady state of the envelope. If the MIDI allocation is based exclusively on the latter, the program would produce a *note-off* message only for the latest note which would in turn result in the previously played notes still being active.

In the proposed algorithm, the MIDI allocation is managed through a bespoke abstraction: [midi_manager]. This abstraction (Fig. 3) allows the triggering of a *note-off* message as a direct consequence of change in pitch that, combined with the binary approach to velocity, ensures the correct allocation of MIDI messages. Whenever a new MIDI note number is received from [sigmund~], it is passed through as a *note-on* message and stored into an object [f] which is acting as a buffer. To ensure that these actions are performed in succession, a [trigger] object is used as it provides signal sequencing with negligible delay. In between those actions, a non-numerical value used as a single activation signal (*bang*) is sent to the buffer forcing it to output whatever value was previously stored. By adding a velocity of zero the MIDI note is delivered as a

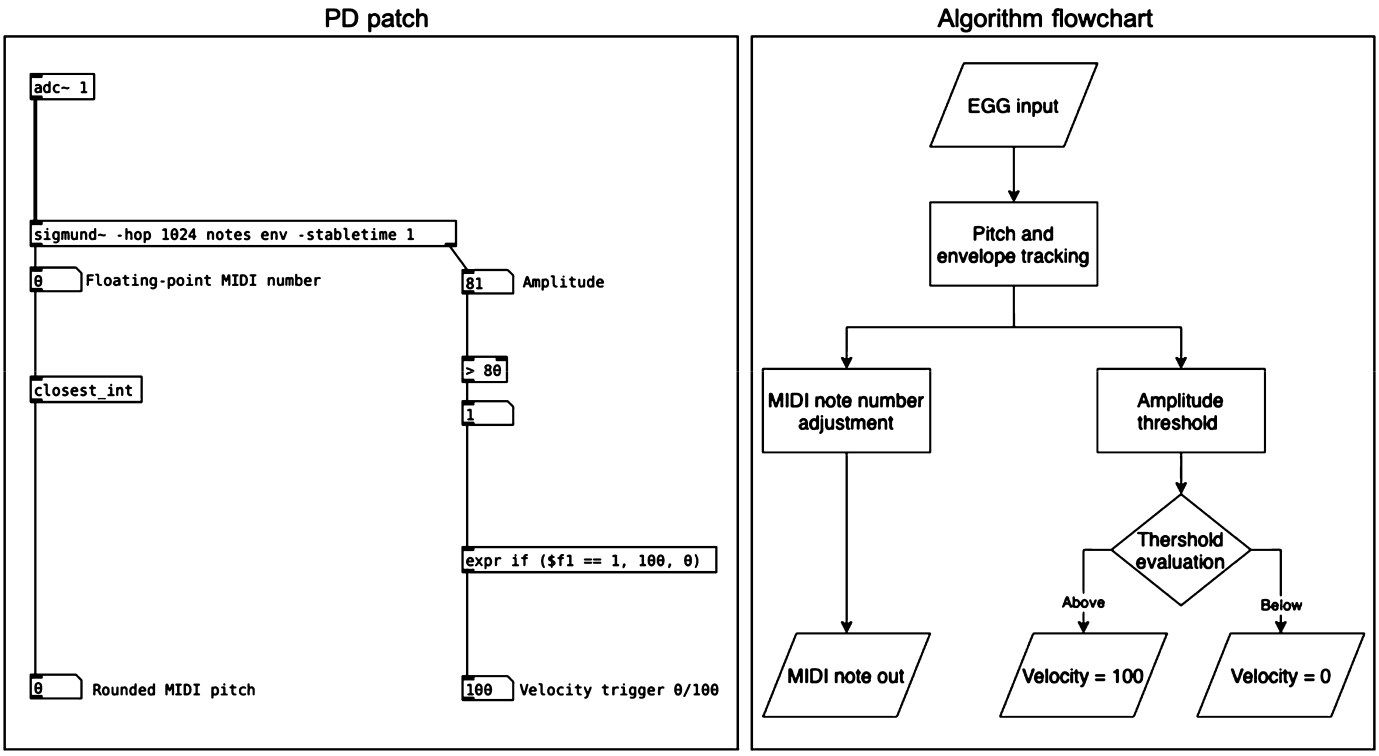

Fig. 3. Pitch and envelope tracking, PD patch and relative flowchart.

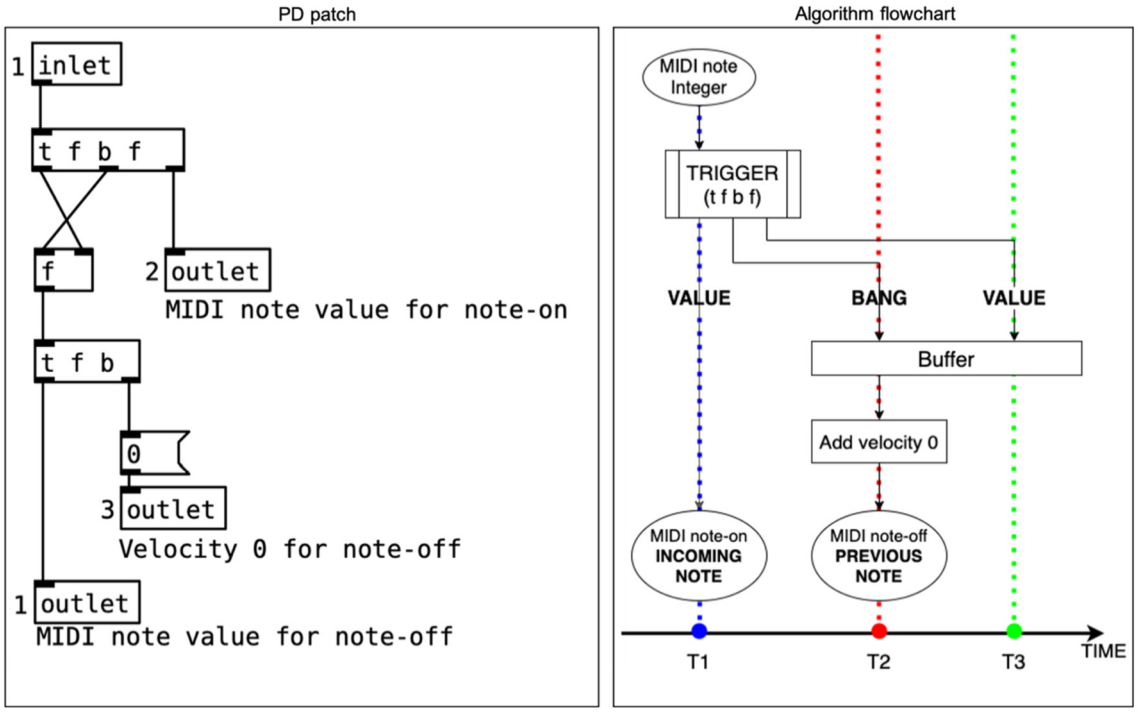

Fig. 4. MIDI note-on/off PD patch and relative flowchart.

note-off. As the *bang* is sent to the buffer before providing it with a new value, any time a new note is received the previous will be outputted as a *note-off*. Fig. 4 shows the Pure Data patch and the relative flowchart for the core of [midi_manager] abstraction.

As the system relies on a note taking over the previous, no note-off message is at this stage produced when phonation is completely stopped. To address such issue the abstraction applies a velocity of zero to the current note as soon as the phonation act exits its steady state. The evaluation of the state of the envelope is derived directly from the [sigmund~] object. Additionally, the described method requires an initialisation of the buffer. At the launch of the program, a MIDI note number (0) is stored inside [f] providing the buffer with a fictitious "previous value".

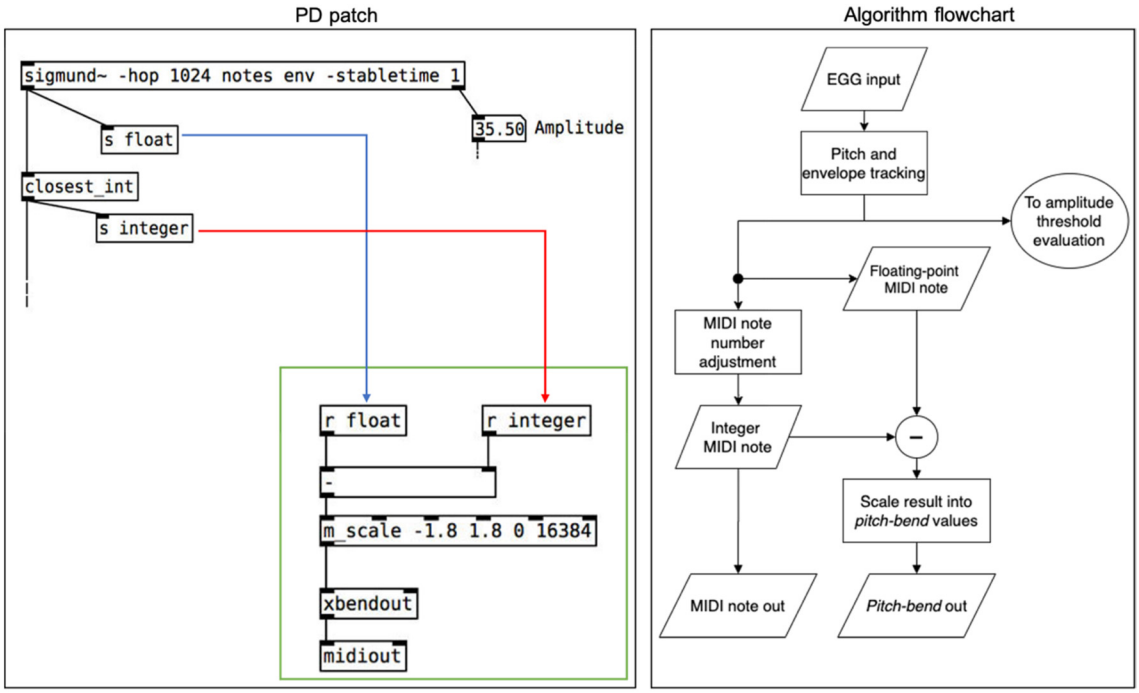

Fig. 5. Pitch-bend processor, PD patch and relative flowchart.

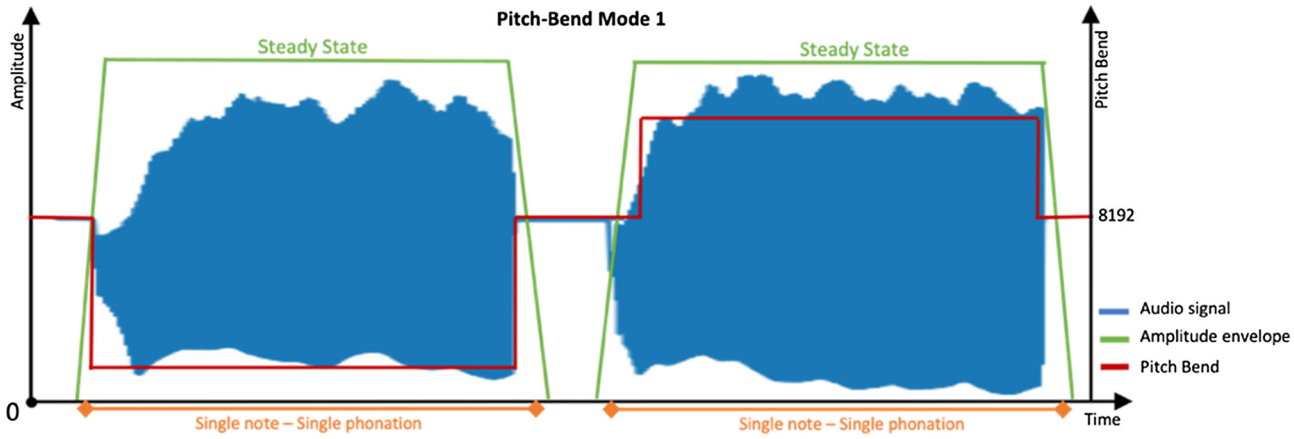

Fig. 6. Pitch-bend mode 1 functioning.

4.3. Microtonal variation and pitch-bend

In most cases, a singer is not able to tune the voice on the exact frequency defined by the chromatic scale and a microtonal shifting will often take place. As MIDI note numbers do not specify values between semitones this will cause a "undesirable correction" of the sung pitch and thus a difference between the sung note and the note produced by the receiving MIDI device. This frequency displacement is addressed using the *pitch-bend* functionality of the MIDI protocol.

The [sigmund~] object delivers a MIDI note number in floating-point form where the decimal part specifies the microtonal shift. Said decimal part is converted in *pitch-bend* values that are then outputted together with the *note-on* message; this is achieved by scaling the decimal part of the MIDI note into the 2bytes range of the *pitch-bend* message with 8192 (half range) being the no *pitch-bend* point (Fig. 5).

At this point, the system can tune an electronic instrument receiving MIDI to the exact frequency produced by the singer. However, the functioning of the [midi_manager] causes a new *note-off/on* succession to be produced for any new pitch received and thus for the *pitch-bend* to only affect the microtonal shift of single notes. Therefore, at this stage, the system is not able to produce a legato over a single phonation.

To allow the production of legato, an additional novel technique was developed for the system. This allows the user to select between the two approaches which are referred to as *Pitch-bend Mode 1* and *Pitch-bend Mode 2*. In the latter, a signal gate is used to stop the [midi_manager] from retriggering notes during the steady state of the envelope. At this point, the *note-on*|*note-off* messages are limited to the *attack*|*release* stages of the envelope. Using the *pitch-bend* only within the steady state of the envelope, the system can "navigate" through the changes in pitches performed over a single phonation cycle mimicking, in turn, the legato effect. For this to work the *pitch-bend* range of the system must be matched with that of the receiving sound generator. To improve flexibility, a physical switch and a dial were added to allow the user to select between the two *pitch-bend* modes and set the appropriate *pitch-bend* range. Figs. 6 and 7 show how the pitch bend messages are managed by the two modes. In Fig. 6 *pitch-bend* is set as a fixed

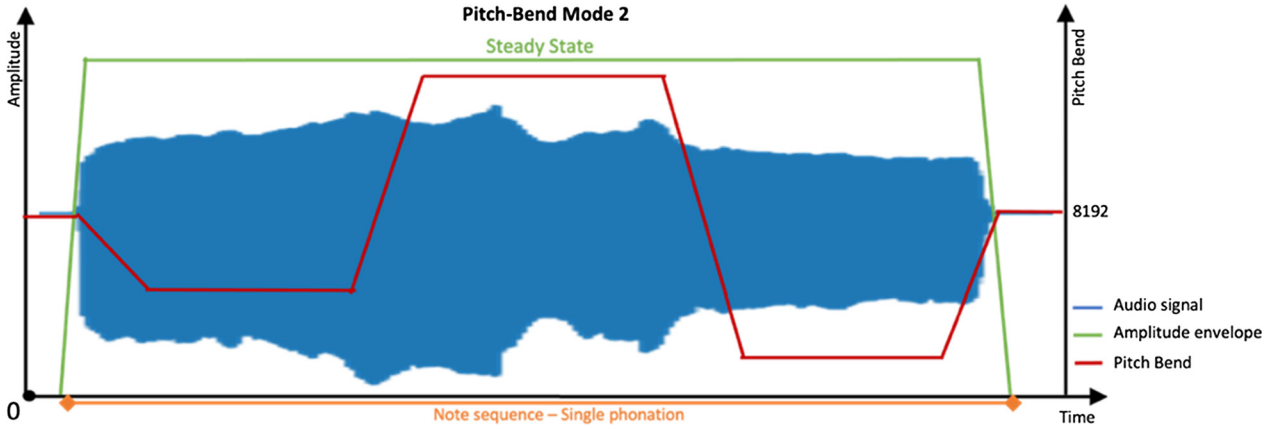

Fig. 7. Pitch-bend mode 2 functioning.

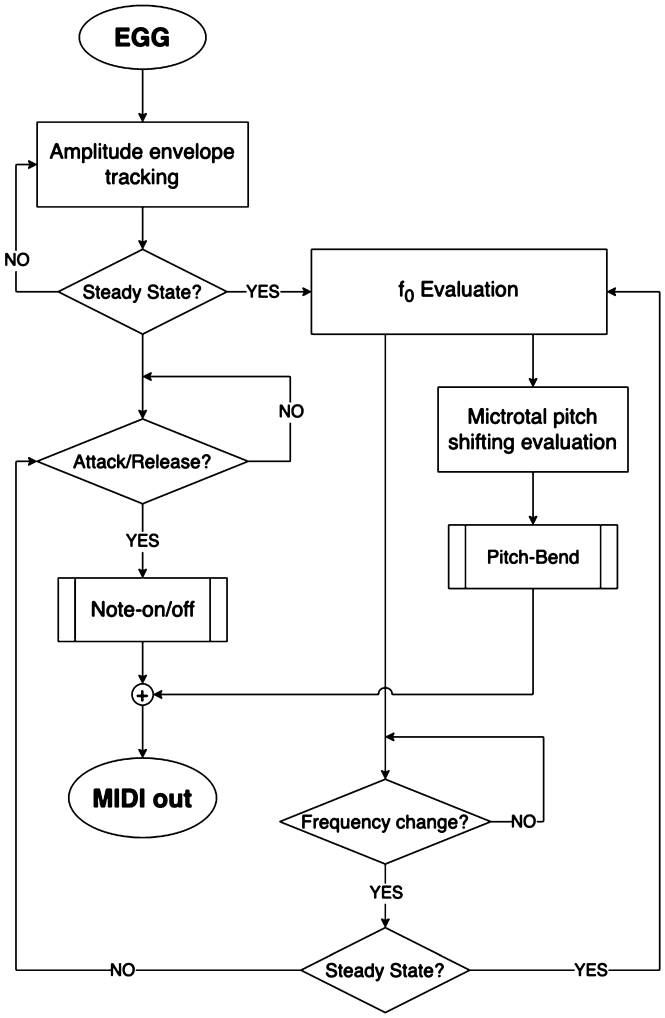

Fig. 8. General algorithm for EGG-to-voice converter.

value for the steady state of each phonation. In Fig. 7 the value of the *pitch-bend* message changes over time across a single phonation creating the legato effect.

4.4. The general algorithm

The three operations discussed in section 4.1, 4.2 and 4.3 were then combined in a single Pure Data (PD) patch. Fig. 8 shows the flowchart of the overall algorithm.

5. Performance assessment

The completed system was assessed in a real-time situation by embedding the PD patch onto a *Bela* board. *Bela* is a development platform specifically designed for audio DSP offering built-in converters and high performances for real-time signal processing with a latency as low as 100 µs [18]. The Ableton Live Digital Audio Workstation (DAW) was used, and Ableton Operator virtual synthesiser was employed as the receiving MIDI instrument. To evaluate both the performances of the system and the efficiency of EGG, four elements were recorded simultaneously. During every single use, the following elements were recorded in the DAW:

- Audio from the voice (microphone)
- EGG output
- Output of the receiving MIDI device
- MIDI note message

The subjects were required to perform a single note and two note sequences (staccato and legato) to evaluate the functioning of both MIDI note allocation and *pitch-bend* processing. A spectral analysis was conducted to evaluate whether both the MIDI notes and the synthesiser output matched the frequency of the recorded voice and EGG (Figs. 9 to 11).

Pitch-bend Mode 1 and *Pitch-bend Mode 2* were used respectively for the staccato and the legato. As explained in chapter 4, *Pitch-bend Mode 2* allows the system to generate a note sequence around a single MIDI note using *pitch-bend*. This functionality is further demonstrated by the results were in the legato a single MIDI note is observed whilst a sequence of frequency is shown in the spectrum.

The performance evaluation of the *pitch-bend* was further assessed by recording the incoming MIDI messages in Ableton as a MIDI automation (Figs. 12 to 14). The difference in the *pitch-bend* behaviour shows again the different implementation of the modes. When *Pitch-bend Mode 1* is used (Fig. 12), a single value of pitch bend is applied to every note to match the frequency produced by the user. In Fig. 13, on the other hand, the *Pitch-bend Mode 2* delivers a single MIDI note with a varying value of *pitch-bend* that effectively produces the changes in frequency. Fig. 14 shows the value of *pitch-bend* applied for a single note.

The recorded MIDI messages showed a match between the produced frequency and the generated MIDI note number. According to the equal temperament, an octave, from lower to higher notes, represents a ratio of 2:1, and contains 12 semitones. This gives between two successive semitones a ratio of approximately 1.0595. For example, for the successive notes A4 and A#4, A4 = 440 Hz

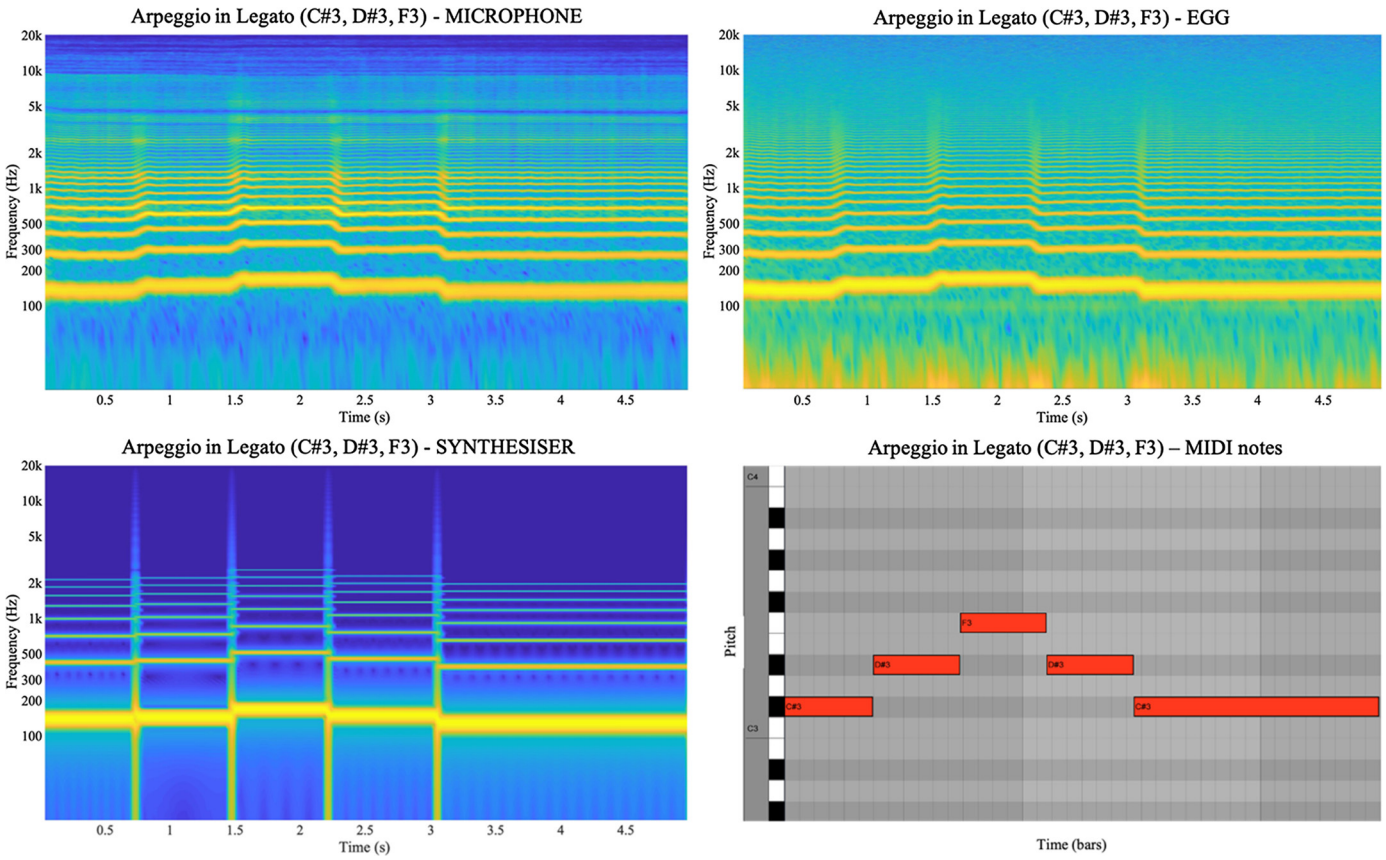

Fig. 9. Arpeggio in legato using Pitch-bend Mode 1. Microphone, EGG, and MIDI synthesiser spectrograms.

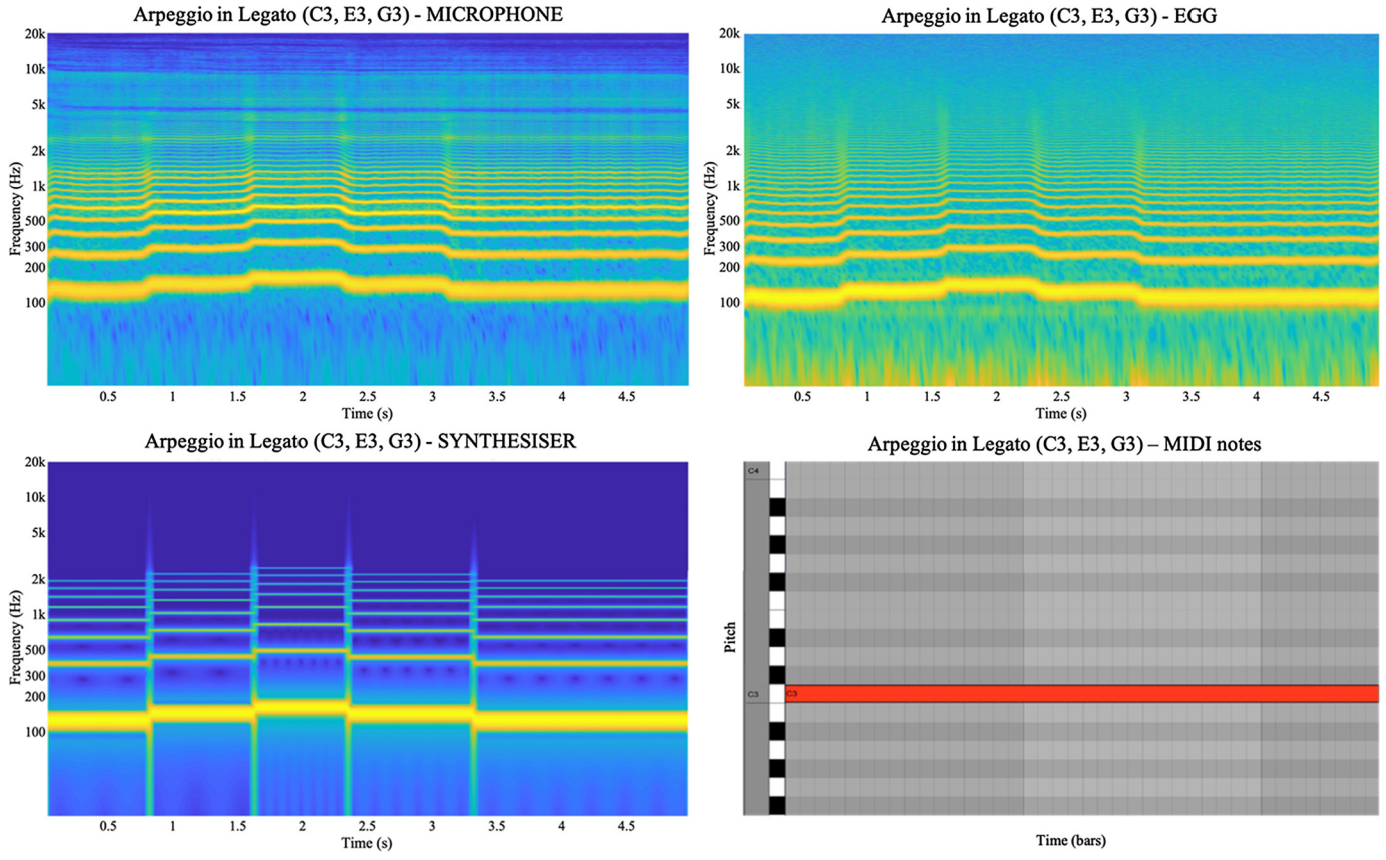

Fig. 10. Arpeggio in legato using Pitch-bend Mode 2. Microphone, EGG, and MIDI synthesiser spectrograms.

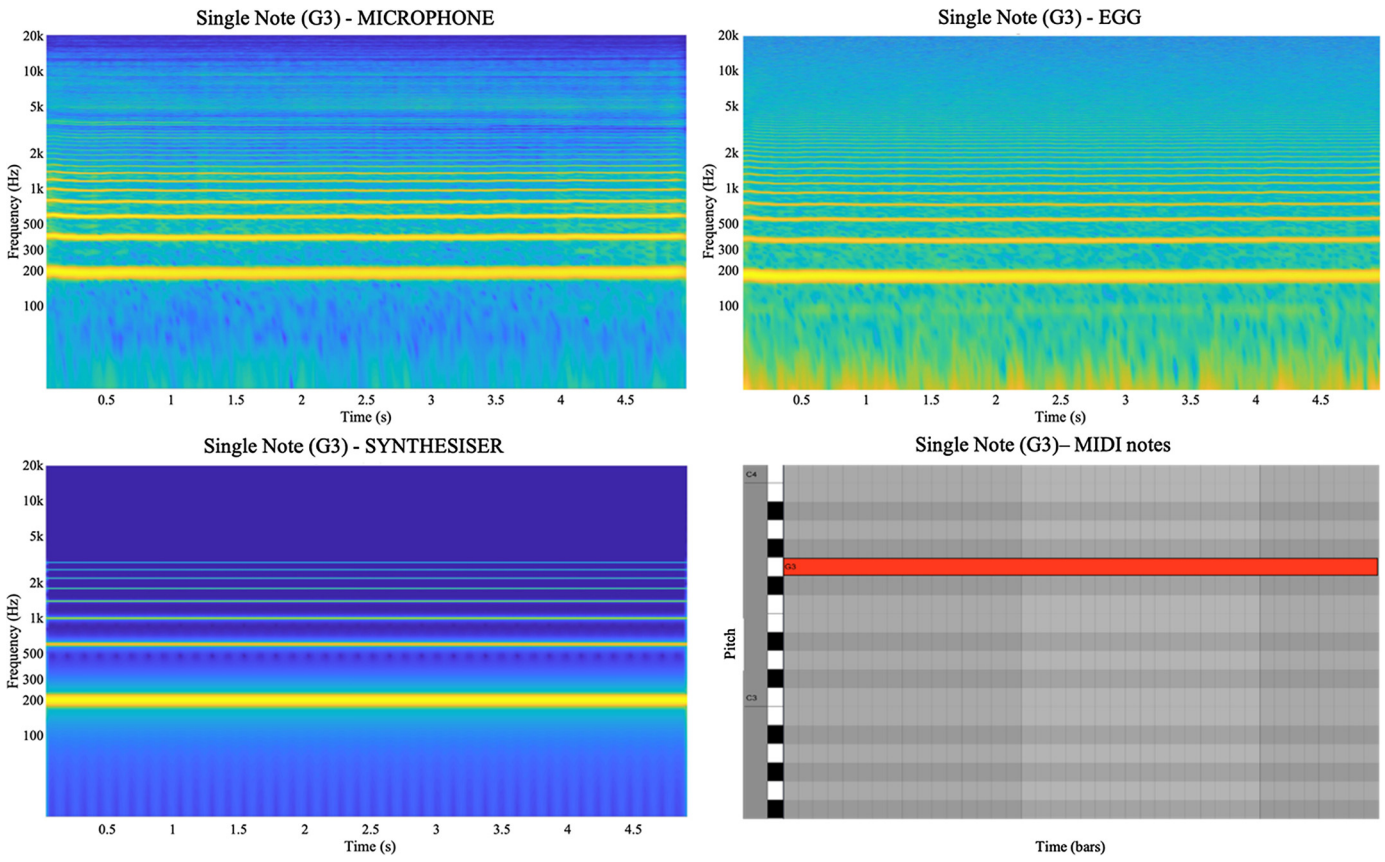

Fig. 11. Single note over continuous single phonation. Microphone, EGG, and MIDI synthesiser spectrograms.

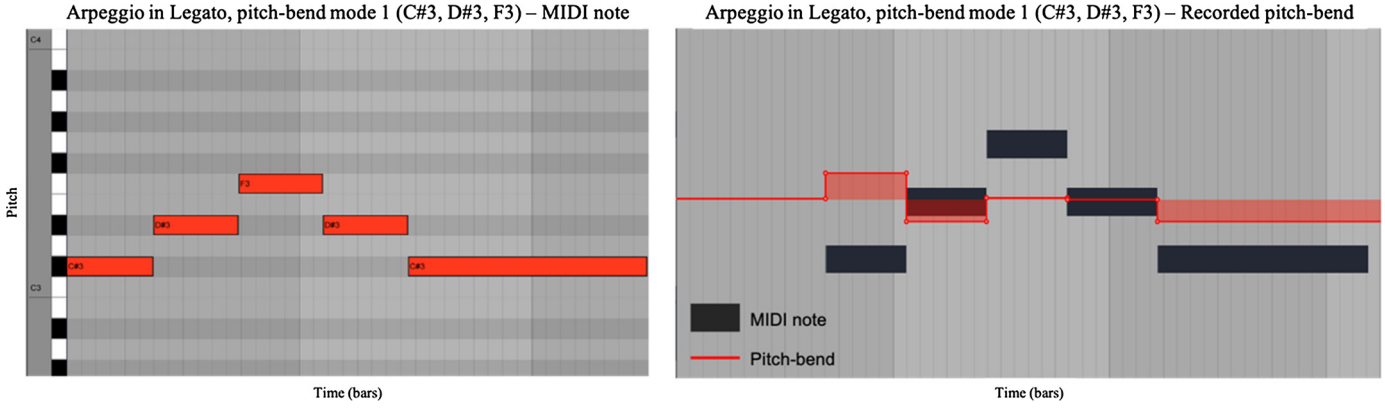

Fig. 12. Arpeggio in legato using Pitch-bend Mode 1. Recorded pitch-bend variation.

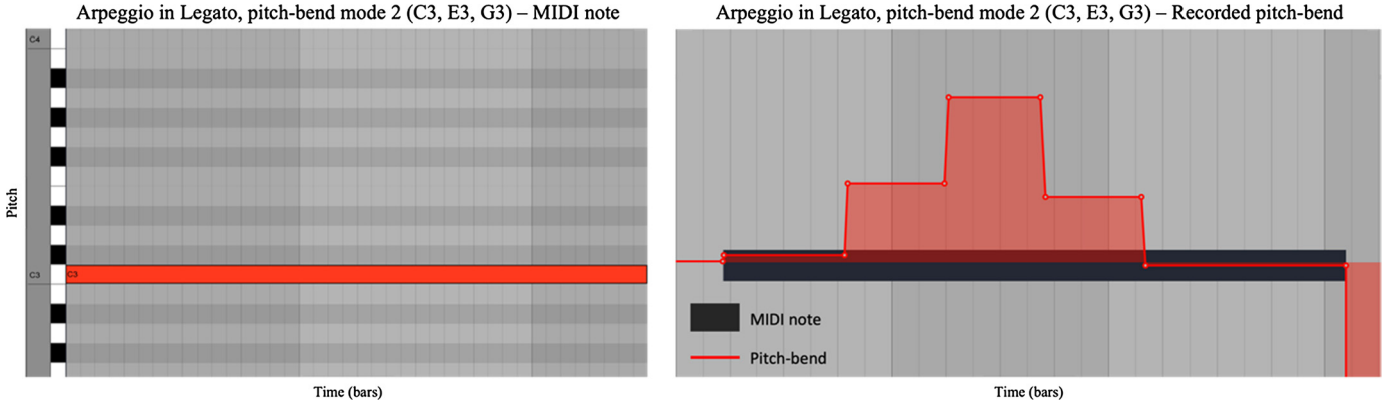

Fig. 13. Arpeggio in legato using Pitch-bend Mode 2. Recorded pitch-bend variation.

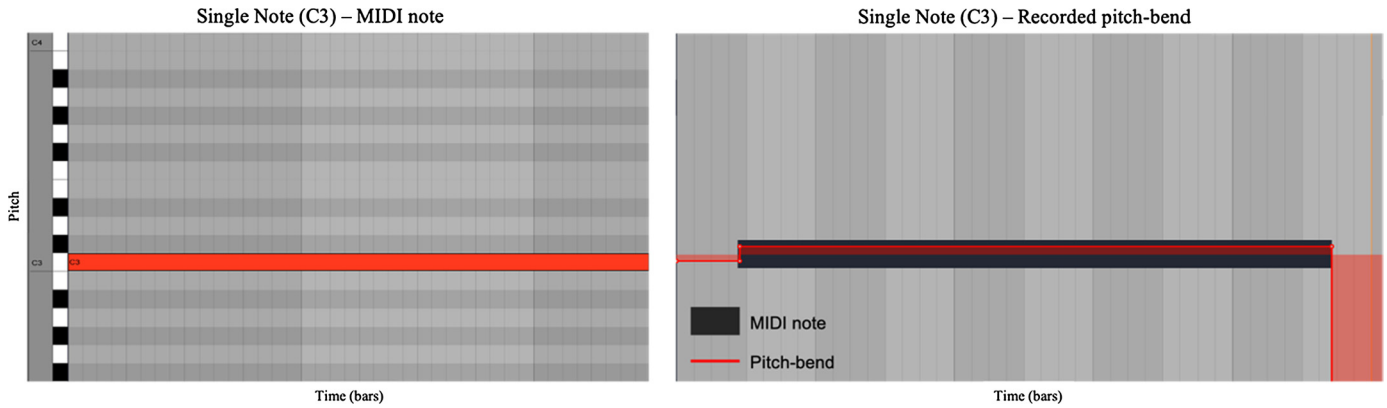

Fig. 14. Single note using Pitch-bend Mode 1. Recorded pitch-bend variation.

Table 1
MIDI to frequency calculation based on equation (1).

	Sung	MIDI note number	Frequency (Hz)
	C#3	49	138.59 Hz
Arpeggio in legato (Fig. 6)	D#3	51	155.56 Hz
	F3	53	174.61 Hz
	C3	48	130.81 Hz
Arpeggio in legato (Fig. 7)	E3	52	164.81 Hz
	G3	55	195.99 Hz
Single note (Fig. 8)	C3	48	130.81 Hz

and A#4 = 466.16 Hz, the ratio is calculated as shown in equation (1):

$$\frac{A4\#}{A4} = \frac{466.16}{4440} \approx 1.0595 = 2^{1/12} \tag{1}$$

Using as reference point A4 = 440 Hz, the frequency f_n of a note that is "n" times higher than A4 can be calculated as follows:

$$f_n = (2^{n/12}) \cdot 440\ \text{Hz} \tag{2}$$

For notes lower than A4 "$-n$" is used instead.

According to the MIDI standards, A4 corresponds to the note number 69. Therefore, parameter n can be expressed as:

$$n = M - 69 \quad \text{where: } M = \textit{Note Number}.$$

Equation (2) can be then modified to equation (3) to link frequencies and "note numbers", when notes are associated with MIDI.

$$f_n = 2^{\left(\frac{M-69}{12}\right)} \cdot 440\ \text{Hz} \tag{3}$$

Therefore, for a given f_n the relevant MIDI note number M will be given by equation (4).

$$M = \left[39.86314 \cdot \left(\log \frac{f_n}{440}\right)\right] + 69 \tag{4}$$

Table 1 shows how the calculation presented in equation (3) and (4) were implemented to evaluate the correct functioning of the system.

The experiments presented in this chapter showed that EGG can be successfully used for an efficient and accurate conversion of voice-to-MIDI in a near real-time environment. The main limitation showed by the system is represented by the latency between a phonation act and the delivery of the MIDI messages. The performance assessment shows a latency of ∼20 ms that, despite acceptable in perceptual terms, would require for optimal real-time operation to be reduced to 10-15 ms [19].

6. Conclusions

6.1. Overview

The research presented in this paper sought to tackle the problems of voice-to-MIDI conversion by using electroglottography (EGG) as a signal source. The nature of EGG allowed a much simpler and computationally inexpensive analysis of a given phonation, in particular, for what concerns the evaluation of the fundamental frequency. The EGG signals were fed into an embedded system running a Pure Data algorithm on the development platform Bela. The proposed algorithm runs a pitch and envelope tracking on the incoming signal deriving, in doing so, both the fundamental frequency of the phonation and its amplitude envelope. While the pitch tracking allows a straight conversion of the voice frequency into MIDI notes, the amplitude tracking provided the threshold levels for the triggering of note-on and note-off messages. On top of the basic conversion, furthermore, *pitch-bend* messages were used to emulate the voice frequency offset; this allowed to track and replicate any microtonal shift that a singing act might have in comparison to the frequencies defined by the chromatic scale. The performances of the system prototype were assessed through simultaneous recording of its MIDI output and the EGG source signal. The constructed prototype resulted in a functioning device able to deliver the expected result. Despite the successful implementation of most of its parts, however, the system still presents limitations. Mainly, the performance assessment shows a latency causing a delay of ∼20 milliseconds between the starting of the EGG and the triggering of the note-on/note-off messages. These issues could derive from different factors: firstly, the computational deficiencies of the Bela board in running a Pure Data algorithm and, secondly, an excess of stages in the testing setup. Even though a latency of about 20 ms is acceptable in terms of real-time perception, to achieve optimal performances the system would need to run with a maximum of 10-15 ms total latency [19].

Although the use of EGG in voice-to-MIDI conversion offers a much-improved efficiency and effectiveness over audio signals, this technique would cause the performer to adjust the singing technique to favour the functioning of the program. In modern singing techniques, the formants and resonances of the vocal tract are employed to support tuning and add harmonics; singers, thus, tend to use this tool in the search and shaping of the desired sound. When using an EGG-to-MIDI converter, the pitch is derived exclusively from the vibration of the vocal folds; this peculiarity requires the singer to adjust his technique in a way in which the attention to the biomechanics is focused on reaching the correct tuning without the use of formants. User interaction, thus, effectively requires the development of a dedicated instrumental technique.

6.2. Future directions

The experiments conducted in this research project suggested a plethora of possible future directions for a further development of the system. The old-fashioned design of the EGG itself causes susceptibility to external interferences and requires expertise for the correct placement of the electrodes. The construction of a modern EGG with a multi-pair electrode system could tackle both limitations. Furthermore, a machine learning implementation could automate the placement evaluation for the multi-pair system by identifying the optimal combination for any given user. Another limitation is represented by EGG functioning itself; as an EGG measures the movement of vocal folds, any non-phonetic act that causes a displacement of the folds could result in an output signal. The employment of a second machine learning model could be used to discard unwanted signals and limit the conversion to singing acts. The abovementioned implementations could create an independent self-deployable system capable of identifying singing phonemes and delivering near real-time conversion of voice into MIDI messages with low computational needs and unaffected by the acoustic environment. Our current research focuses on addressing such issues through the development of a multi-sensor EGG and the implementation of two separate neural networks for both optimal electrode placement and isolation of singing acts.

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.

References

[1] T. Viitaniemi, A. Klapuri, A. Eronen, A probabilistic model for the transcription of single-voice melodies, in: Proceedings of the 2003 Finnish Signal Processing Symposium, FINSIG'03, 2003, pp. 59–63.
[2] N. Itou, K. Nishimoto, A voice-to-MIDI system for singing melodies with lyrics, in: Proceedings of the International Conference on Advances in Computer Entertainment Technology, 2007, pp. 183–189.
[3] M. Antonelli, A. Rizzi, A correntropy-based voice to MIDI transcription algorithm, in: Multimedia Signal Processing, 2008 IEEE 10th Workshop, IEEE, 2008, pp. 978–983.
[4] K. Kehrakos, C. Chousidis, S. Kouzoupis, A Reliable Singing Voice-Driven MIDI Controller Using Electroglottographic Signal, Audio Engineering Society Convention, vol. 140, Audio Engineering Society, 2016.
[5] K. Kehrakos, S. Kouzoupis, C. Chousidis, An efficient method of extracting singing voice information using electroglottographic signal, in: 23rd International Cinfress on Sound and Vibration, 2016.
[6] C. Chousidis, L. Lipan, The application of a novel voice-driven MIDI controller in music education and training, in: ICICTE 2016 Proceedings, 2016.
[7] F. Arvin, S. Doraisamy, A real time signal processing technique for MIDI generation, in: WSEAS International Conference. Proceedings, in: Mathematics and Computers in Science and Engineering, vol. 2008, 2008.
[8] T. Modegi, S. Iisaku, Proposals of MIDI coding and its application for audio authoring, in: Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No. 98TB100241), IEEE, 1998, pp. 305–314.
[9] O. Derrien, A very low latency pitch tracker for audio to MIDI conversion, in: 17th International Conference on Digital Audio Effects (DAFx-14) 2014, 2014.
[10] M. Garcia, Observations on the human voice, Proc. R. Soc. Lond. 7 (1854) 399–410.
[11] P. Fabre, La glottographie electrique en haute frequence, particularites de lappareillage, C. R. Séances Soc. Biol. Fil. 153 (8-9) (1959) 1361–1364 (in French).
[12] A.J. Fourcin, Laryngographic examination of vocal fold vibration, in: Ventilatory and Phonatory Control Systems, 1974, pp. 315–333.
[13] I. Titze, Interpretation of the electroglottographic signal, J. Voice (1990).
[14] T. Drugman, P. Alku, A. Alwan, B. Yegnanarayana, Glottal source processing: from analysis to applications, Comput. Speech Lang. (2014) 1117–1138.
[15] T. Drugman, B. Bozkurt, T. Dutoit, A comparative study of glottal source estimation techniques, Comput. Speech Lang. 20 (2012).
[16] Puredata.info. n.d. Pure Data - Pd Community Site [online], available at https://puredata.info/.
[17] M.S. Puckette, Pure data, in: ICMC, 1997, September.
[18] A. McPherson, Bela: an embedded platform for low-latency feedback control of sound, J. Acoust. Soc. Am. 141 (5) (2017) 3618.
[19] D. Stowell, M.D. Plumbley, Delayed decision-making in real-time beatbox percussion classification, J. New Music Res. 39 (3) (2010) 203–213.

Eugenio Donati received his BA (2013) from the UNINT University of Rome in Interpreting and Translation. He obtained his diploma in Sonic Arts (2013) from the Saint Louis College of Music of Rome. He received his BSc (Hons) (2017) in Applied Sound Engineering at the University of West London (UWL) where he also received the MSc (2018) in Digital Audio Engineering. Eugenio started his PhD in 2019 at UWL; his research focuses on Audio Electronics, Biomedical Acoustics and Machine Learning. Eugenio is part of the Biomedical Acoustics special interest group within the UK-Acoustics Network (UKAN) and its Early Careers coordinator.

Christos Chousidis received his B.Eng. from the Technological Institute of Crete in 1995. In 2006 he received his MPhil and in 2014 his PhD both from Brunel University. His research focusses on Biomedical Acoustics and Wireless Audio Networks. He is a member of IEEE and Audio Engineering Society (AES). Christos is also a member of the Technical Committees on Network Audio Systems (TC-NAS) and Machine Learning and Artificial Intelligence (TC-MLAI) within the AES. He is also a founding member of the UKAN's special Interest Group on Biomedical Acoustics.