Resin: a Vocal Tract Resonances and Head Based Accessible Digital Musical Instrument

Nicola Davanzo
nicola.davanzo@unimi.it
Università degli Studi di Milano
Milan, Italy

Federico Avanzini
federico.avanzini@unimi.it
Università degli Studi di Milano
Milan, Italy

ABSTRACT

Recent developments in sensor technologies allowed the definition of new human-computer interaction channels, useful for people with very limiting motor disabilities such as quadriplegia. Some of these sensors are available pre-packaged on the mass market, complete with computer interaction softwares, while others are easily achievable at low costs through DIY approaches. In this article we present Resin, an Accessible Digital Musical Instrument dedicated to people with quadriplegic disability. Resin exploits two interaction channels, head movements and the shape of the vocal tract, detected through the corresponding acoustic resonances, to control musical performance parameters. The structure of the instrument is discussed, from both the hardware and software points of view. Feature extraction algorithms for both channels are explained, particularly focusing on the vocal tract resonances interaction paradigm.

CCS CONCEPTS

• **Applied computing** → **Arts and humanities**; **Sound and music computing**.

KEYWORDS

accessibility, musical instrument, vocal tract, head tracking

ACM Reference Format:
Nicola Davanzo and Federico Avanzini. 2021. Resin: a Vocal Tract Resonances and Head Based Accessible Digital Musical Instrument. In *Audio Mostly '21 Proceedings.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3478384.3478403

1 INTRODUCTION

Playing a musical instrument, or more in general being able to actively participate to musical activities, is recognized as an important human right by the World Health Organization [26]. People unable to control one or more limbs, with severe motor disabling conditions such as cerebral palsy, quadriplegia, lock-in syndrome, are often excluded from the possibility of playing an acoustic musical instrument. The development of accessible digital musical instruments (ADMIs) dedicated to those people has undergone a great development in recent decades due to the availability of affordable sensors such as eye trackers (e.g. Tobii[1]), head trackers and breath sensors. This research area has produced ADMIs such as EyeHarp [28] and Netytar [6], which both use the gaze point as an interaction channel, or such as Jamboxx [12] and Magic Flute [10] which exploit head movements, but also some experiments with the use of electroencephalographic sensors, such as MusEEGk [3] and P-300 Harmonies [27].

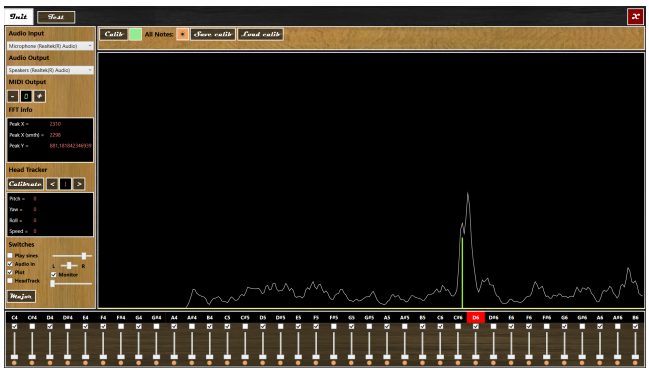

Figure 1: Resin's graphical user interface

In this article we propose Resin, a monophonic ADMI dedicated to quadriplegic musicians, which is a MIDI musical interface. It experiments with two interaction channels that are still little exploited and used in ADMIs design: head movement and vocal tract resonances. Head movement along the horizontal axis (yaw) is used to control notes onset and dynamics (MIDI *Pressure*, *Velocity*, *Note On* and *Note Off* parameters), while resonances within the vocal tract are used to control the note *Pitch*. Resin consists of a software and a hardware parts, the latter being a mixture of DIY and pre-built components. Resin's interface is depicted in Fig. 1. We will discuss Resin's interaction paradigm and implementation in sections 3 and 4, after discussing in the next section the current state-of-the-art in the relevant research fields.

2 STATE-OF-THE-ART

Alongside recent reviews of ADMIs, such as the works of Larsen [15] and Frid [9], a recent work by Davanzo and Avanzini [5] has proposed an analysis of interaction channels available to develop ADMIs specifically dedicated to quadriplegic users, framing and dividing those channels into four macro-groups (channels relating to eyes, mouth, head movement and brain activity). The use of vocal

[1]Tobii Eye Tracker 5 on Tobii's website: https://gaming.tobii.com/product/eye-tracker-5/

Audio Mostly '21, Sep. 1-3 2021, Trento, IT

ACM ISBN 978-1-4503-8569-5/21/09…$15.00
https://doi.org/10.1145/3478384.3478403

tract resonances is however an interaction modality not included in the above list.

Regarding Resin's vocal interaction paradigm, no similar systems have been found in the literature, to the best of the authors' knowledge. However some analogies can be drawn. It can be stated that the internal shape of the oral cavity, which is the terminal part of the vocal tract, varies in accordance with tongue movements. Various systems have been tested to detect the tongue position for interaction purposes. In Tongue Music [19], the detection happens through hall-effect sensors coupled with magnets. In Niikawa's Tongue Controlled Electro-Musical Instrument [20] the tongue presses some buttons positioned on the palate to play chords. Other detection methods, not used in musical contexts, include the use of textile pressure [2], ultrasound [13, 29], magnetoresistive [21] or optical sensors [11, 23].

Based on pitch detection techniques, Imitone [1] is a software interface able to detect the pitch of voice or whistle through a microphone, and convert the input in MIDI messages in real time. An analogy can be traced also with Silent Speech Interfaces (SSI). These are interfaces aimed at recognizing facial and buccal movements in order to reproduce speech without the need for the user to emit any sound. According to [7], such interfaces usually exploit electromagnetic articulography, electromyography, ultrasounds, microphones, electroencelography or neural cortex implants.

A comparison could also be drawn with interfaces that exploit mouth shape detection to control sound filters. The *Talkbox* is a common analog electric guitar effect that consists of a speaker channeling the guitar sound into a rubber tube, having the other end placed in the musician's mouth. The sound comes out of the mouth to be picked up by microphone. Several interfaces, such as Mouthesizer [17], use cameras and computer vision techniques to detect mouth movement and use it to control sound filters. Eye Conductor [22] is an ADMI that exploits the same paradigm.

Head tracking is a consolidated interaction channel. Davanzo and Avanzini's work [4] tested and evaluated it for both general human-computer and musical interaction. It has been shown that the movement of the head allows for stability, speed, and precision of movement. It has been exploited in ADMIs such as HiNote [18] to select notes on a virtual keyboard. It has also been used to control virtual reality instruments [8, 24], and augmented acoustic instruments [14].

3 INTERACTION IN RESIN

In this section we discuss the two interaction channels exploited by the instrument: vocal tract resonances and head movement.

3.1 Vocal tract resonances

The process responsible for the production of vocalized sounds in humans is often modeled by considering two main components. In the first one, vocal cords vibrate creating a "pulse train", whose frequency defines the voice pitch. In the second one, various components of the vocal tract, such as the mouth, act as a filter enhancing some bands in the spectrum of the vocalized signal. The corresponding resonance frequencies (formants) characterize different vowels [25]. Some singing styles, such as tuvan throat singing, exploit the resonances created by the vocal tract to combine them into complex melodies [16].

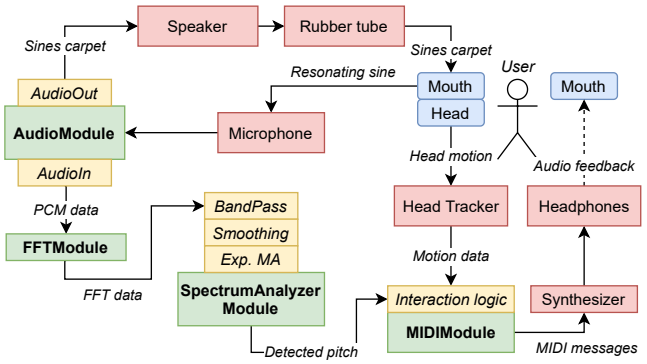

Figure 2: Functional diagram for Resin

Figure 3: Resin's mouthpiece, consisting of the sound tube and the Lavalier microphone

In Resin, the vocal tract is stimulated through a synthesized sound conveyed to an ad-hoc built hardware component, which we will refer as *sound tube*. This consists of a speaker, properly muffled on the sides, which emits a synthesized sound (which we will refer to as *sine pad*) into a rubber tube. The musician puts the tube end in their mouth and grasps it with their teeth as they play, keeping their mouth slightly open. The sound, produced by Resin's software, is a linear combination of different sinusoidal components, whose frequencies are tuned to successive semitones in the equal temperament (A5 tuned to 440Hz). The interface allows the musician to select which notes (which we will refer as *playable notes* are included. By varying the mouth shape, some of these sinusoidal components resonate louder. A small Lavalier microphone is placed in the mouth next to the sound tube, and picks up the filtered sound. Resin 's software therefore recognizes the resonating frequency/note. Resin's mouthpiece is depicted in Fig. 3.

3.2 Head tracking

The hardware required to operate Resin includes a low cost head tracker, built using an Arduino Uno (or Arduino Nano) microcontroller[2] and a MEMS gyroscope/accelerometer GY-521 MPU-6050.[3] Arduino runs a script which translates acceleration into absolute position data. Head movement is detected in a range of 40 degrees [-20°; +20°] in the horizontal plane. Head movements are used to perform attack and release actions on the instrument. A new note (attack) is triggered when an inversion of the head motion is detected, prior passing through the central position (0°). The strum

[2]Arduino Uno on Arduino website: https://store.arduino.cc/arduino-uno-rev3
[3]Invensense MPU-6050 datasheet: https://tinyurl.com/4p5cpmde

intensity can be chosen to be proportional to the distance from the center at the inversion point, or to the average movement speed in the previous instants, depending on the selected interaction mode. Movement speed determines also the *channel pressure*. The latter is calculated as the distance between head position in the current sample and in the previous one, filtered by an exponentially moving average filter.

3.3 Performance logic

Musical performance takes place in the following way. The system continuously detects the resonating note. An indicator on screen highlights the detected note n_r; as soon as a head strumming action is detected, a MIDI *note-on message* is sent for n_r, with a MIDI velocity determined by one of the two approaches for strum intensity mentioned above; the note remains on-set even if the detected resonating note changes, until a subsequent strumming action occurs. In this case, the old n_r is stopped by a *note-off* event, and is replaced by the new n_r; *channel pressure* varies continuously in proportion to head movement speed, contributing to note dynamics.

4 IMPLEMENTATION

We now discuss the actual implementation of Resin, at its current version. Interaction, sound processing and generation are summarized in Fig. 2.

4.1 Hardware

The employed microphone is a Røde smartLav+[4], with a frequency response of 20Hz-20kHz and a sensitivity of -32.0dB re 1 Volt/Pascal, equipped with a pop filter. The microphone is enclosed in a cellophane layer to prevent water infiltration. The system was tested using an Alesis iO2[5] sound card, at 48 KHz/24-bits.

4.2 Software, audio generation and processing

Resin software is coded in C#, using the Windows Presentation Foundation graphical framework, part of the .NET 4.8 framework. It is available for download from its GitHub Repository,[6] licensed under the Open Source GNU GPL V3 license.

Prior to playing, the user selects which are the *playable notes*, using the interface. Therefore the *AudioModule* class generates the corresponding sine pad (see Sec. 3.1) where sinusoidal frequencies are the fundamental frequencies of the selected notes, and all the components have the same amplitude. The resulting sound is conveyed to the performer's mouth through the sound tube.

After the musician's mouth has filtered the sound, the *AudioModule* class receives the PCM signal from the microphone. The *FftModule* class performs then a Fast Fourier Transform of an audio buffer. In accordance with the sound card specifications, the FFT is performed considering a 48 KHz/24 bit sampled signal and an audio buffer of 43 ms, resulting in an array of 2064 samples. After applying a Hamming Window, the audio buffer is zero-padded to 4096 values, in order to improve frequency resolution. The number of FFT bins obtained is therefore 2048.

[4]Rode Smartlav+: https://it.rode.com/microphones/smartlav-plus
[5]Alesis iO2 sound card: https://www.alesis.com/products/legacy/io2
[6]Resin on GitHub: https://github.com/LIMUNIMI/Resin

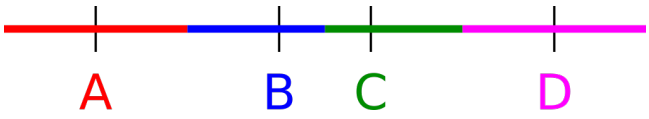

Figure 4: FFT spectrum division (horizontal axis) for notes A, B, C and D. Different bin groups are highlighted with different colors. Since B and C have a distance of a single semitone, their associated group is smaller.

Some filters are then applied to the resulting FFT magniture (short-time spectrum). A bandpass filter clears the frequency bins placed outside the range of the playable notes selected by the user; A smoothing filter performs spectral smoothing. Each element of the output spectrum is calculated as in Eq. (1):

$$O_i = \frac{I_{i-1} + I_i + I_{i+1}}{3}, \tag{1}$$

where O and I are respectively the output and input short-time spectra, while i denotes the i-th bin. An exponentially moving average filter is applied to successive short-time spectra, causing the energy of each bin to vary more smoothly over time, in order to prevent sudden oscillations due to noise. The filter is in the form described by Eq. (2):

$$O_i(t) = \alpha \cdot I_i(t) + (1 - \alpha) \cdot O_i(t - 1), \tag{2}$$

where t is discrete time, while α is an arbitrary constant set at 0.9. The *SpectrumAnalyzerModule* class then determines the pitch of the resonant note. Each playable note n is associated to the bin $B(n)$ where its fundamental frequency falls. The spectrum is divided into different groups of bins $G(n)$, each centered around the corresponding $B(n)$. Specifically, the upper and lower boundaries of $G(n)$, defined as $B_U[G(n)]$ and $B_L[G(n)]$, respectively, are defined by Eqs. (3) and (4):

$$B_U[G(n)] = \frac{B(n) + B(n+1)}{2}, \tag{3}$$

$$B_L[G(n)] = \frac{B(n) + B(n-1)}{2}. \tag{4}$$

Fig. 4 graphically summarizes this groups division approach. *SpectrumAnalyzerModule* then proceeds by determining the mean quadratic spectral energy $E(n)$ for each $G(n)$, defined as:

$$E(n) = \frac{1}{N_{bin}(G(n))} \sum_{b \in G(n)} E(b)^2. \tag{5}$$

The estimated resonant note is the one corresponding to the group with the greatest energy. Finally, the *MIDIModule* generates the corresponding MIDI messages.

4.3 Graphical User Interface

The *FFTPlot* class draws the short-time spectrum together with two indicators Fig. 1. A bar denotes the single bin with the highest energy, while a second bar indicates the $G(n)$ with the highest energy. The lower part of the interface highlights the detected pitch. Some buttons and sliders allow to select the playable notes, as well as to perform a manual calibration of the volume of each note.

5 DISCUSSION AND CONCLUSIONS

We have discussed the design, structure, and implementation of Resin, an ADMI dedicated to quadriplegic musicians, easily assembled using low cost components.

Accuracy in note recognition is generally sufficient for playing but could still be improved. Sinusoidal energy spreads through multiple FFT bins, due to the algorithm's nature, resulting in some imperfections. In the current version, each sine in the pad has the same frequency as its associated MIDI note. This results in a more natural interaction: the mouth shape associated with each note is similar to the one required to whistle that note. In an attempt to improve detection precision, we tested the use of larger distances between sine frequencies (thus mapped to notes having different frequencies). However this solution resulted in the interaction being less natural, so it was abandoned. Instead, we opted for the possibility of choosing which notes are "playable" (e.g. only those belonging to the C major scale, or pentatonic scale). With a smaller set of notes, the $G(n)$'s become larger, thus improving recognition. The speaker/tube/microphone system does not have a flat frequency response. To (at least partially) mitigate this problem, an automatic calibration system was implemented, which proceeds to "flatten" the detected spectrum by adjusting the volume of each sinusoidal component while not performing. The sine pad could be audible from the outside and cause nuisance. While it is possible to greatly reduce its volume while preserving the instrument functionality, below a certain value the system may be sensitive to ambient noises, as well as to the MIDI synthesizer feedback. This could be improved by using a more directional microphone, or headphones. The head-based "strumming" metaphor implemented in Resin may be suitable for simulating some types of instruments such as strings, while being less suitable for others. The entry barrier in learning Resin could be very high. A user able to whistle in tune is probably facilitated in learning the instrument. However, the initial learning curve may affect objective evaluation. Planned future developments include an evaluation of the instrument through case studies and execution of pre-established exercises, as well as refinement and tuning of the detection algorithms for both vocal resonances and head movements.

ACKNOWLEDGMENTS

We thank Audio Modeling SRL for providing us with free access to the SWAM plugin suite for research purposes.

REFERENCES

[1] Evan Balster. n.d.. Imitone: mind to melody. https://imitone.com/.
[2] Jingyuan Cheng, Ayano Okoso, Kai Kunze, Niels Henze, Albrecht Schmidt, Paul Lukowicz, and Koichi Kise. 2014. On the Tip of My Tongue: A Non-Invasive Pressure-Based Tongue Interface. In *Proc. 5th Augmented Human Int. Conf. (AH '14)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/2582051.2582063
[3] Yee Chieh (Denise) Chew and Eric Caspary. 2011. MusEEGk: A Brain Computer Musical Interface. In *Proc. '11 Ann. Conf. Ext. Abs. on Human Factors in Computing Systems*. ACM Press, Vancouver, BC, Canada, 1417. https://doi.org/10.1145/1979742.1979784
[4] Nicola Davanzo and Federico Avanzini. 2020. Experimental Evaluation of Three Interaction Channels for Accessible Digital Musical Instruments. In *Proc. '20 Int. Conf. on Computers Helping People With Special Needs*. Springer, Cham, Online Conf., 437–445. https://doi.org/10.1007/978-3-030-58805-2_52
[5] Nicola Davanzo and Federico Avanzini. 2020. Hands-Free Accessible Digital Musical Instruments: Conceptual Framework, Challenges, and Perspectives. *IEEE Access* 8 (2020), 163975–163995. https://doi.org/10.1109/ACCESS.2020.3019978
[6] Nicola Davanzo, Piercarlo Dondi, Mauro Mosconi, and Marco Porta. 2018. Playing Music with the Eyes through an Isomorphic Interface. In *Proc. of the Workshop on Communication by Gaze Interaction*. ACM Press, Warsaw, Poland, 1–5. https://doi.org/10.1145/3206343.3206350
[7] Bruce Denby, T. Schultz, K. Honda, T. Hueber, J. M. Gilbert, and J. S. Brumberg. 2010. Silent Speech Interfaces. *Speech Communication (Special Issues)* 52, 4 (April 2010), 270–287. https://doi.org/10.1016/j.specom.2009.08.002
[8] John Fillwalk. 2015. ChromaChord: A Virtual Musical Instrument. In *Proc. 2015 IEEE Symposium on 3D User Interfaces*. IEEE, Arles, France, 201–202. https://doi.org/10.1109/3DUI.2015.7131770
[9] Emma Frid. 2019. Accessible Digital Musical Instruments—A Review of Musical Interfaces in Inclusive Music Practice. *Multimodal Technologies and Interaction* 3, 3 (July 2019), 57. https://doi.org/10.3390/mti3030057
[10] Housemate. n.d.. Magic Flute. http://housemate.ie/magic-flute/.
[11] Thomas Hueber, Elie-Laurent Benaroya, Gérard Chollet, Bruce Denby, Gérard Dreyfus, and Maureen Stone. 2010. Development of a Silent Speech Interface Driven by Ultrasound and Optical Images of the Tongue and Lips. *Speech Communication* 52, 4 (April 2010), 288–300. https://doi.org/10.1016/j.specom.2009.11.004
[12] Jamboxx. n.d.. Jamboxx. https://www.jamboxx.com/.
[13] Tokihiko Kaburagi and Masaaki Honda. 1994. An Ultrasonic Method for Monitoring Tongue Shape and the Position of a Fixed Point on the Tongue Surface. *J. of the Acoustical Society of America* 95, 4 (April 1994), 2268–2270. https://doi.org/10.1121/1.408637
[14] Ajay Kapur, Ariel J. Lazier, Philip Davidson, Wilson R. Scott, and Perry R. Cook. 2017. The Electronic Sitar Controller. In *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression* (first ed.), Alexander Refsum Jensenius and Michael J. Lyons (Eds.). Current Research in Systematic Musicology, Vol. 1. Springer International Publishing, Cham, 147–163. https://doi.org/10.1007/978-3-319-47214-0_10
[15] Jeppe Veirum Larsen, Dan Overholt, and Thomas B. Moeslund. 2016. The Prospects of Musical Instruments For People with Physical Disabilities. In *Proc. 16th Int. Conf. on New Interfaces for Musical Expression (NIME '16)*. NIME, Griffith University, Brisbane, Australia, 327–331.
[16] Theodore C. Levin and Michael E. Edgerton. 1999. The Throat Singers of Tuva. *Scientific American* 281, 3 (1999), 80–87.
[17] Michael J. Lyons, Michael Haehnel, and Nobuji Tetsutani. 2003. Designing, Playing, and Performing with a Vision-Based Mouth Interface. In *Proc. 3rd Conf. on New Interfaces for Musical Expression (NIME '03)*. NIME, McGill University, Montreal, Canada, 116–121.
[18] Vahakn Matossian and Rolf Gehlhaar. 2015. Human Instruments: Accessible Musical Instruments for People with Varied Physical Ability. *Annual Review of Cybertherapy and Telemedicine* 219 (2015), 202–207.
[19] Hye Yeon Nam and Carl DiSalvo. 2010. Tongue Music: The Sound of a Kiss. In *Proc. 28th Int. Conf. Ext. Abs. on Human Factors in Computing Systems*. ACM Press, Atlanta, Georgia, USA, 4805. https://doi.org/10.1145/1753846.1754235
[20] Takuya Niikawa. 2004. Tongue-Controlled Electro-Musical Instrument. In *Proc. 18th Int. Congr. on Acoustics*, Vol. 3. Acoustical Society, International Conference Hall, Kyoto, Japan, 1905–1908.
[21] Hangue Park, Mehdi Kiani, Hyung-Min Lee, Jeonghee Kim, Jacob Block, Benoit Gosselin, and Maysam Ghovanloo. 2012. A Wireless Magnetoresistive Sensing System for an Intraoral Tongue-Computer Interface. *IEEE Trans. on Biomedical Circuits and Systems* 6, 6 (Dec. 2012), 571–585. https://doi.org/10.1109/TBCAS.2012.2227962
[22] Andreas Refsgaard. n.d.. Eye Conductor. https://andreasrefsgaard.dk/project/eye-conductor/.
[23] T. Scott Saponas, Daniel Kelly, Babak A. Parviz, and Desney S. Tan. 2009. Optically Sensing Tongue Gestures for Computer Input. In *Proc. 22nd ACM Symp. on User Interface Software and Technology (UIST '09)*. Association for Computing Machinery, New York, NY, USA, 177–180. https://doi.org/10.1145/1622176.1622209
[24] Stefania Serafin, Cumhur Erkut, Juraj Kojs, Niels C. Nilsson, and Rolf Nordahl. 2016. Virtual Reality Musical Instruments: State of the Art, Design Principles, and Future Directions. *Computer Music J.* 40, 3 (Sept. 2016), 22–40. https://doi.org/10.1162/COMJ_a_00372
[25] Ingo R. Titze and Daniel W. Martin. 1998. Principles of Voice Production. *J. of the Acoustical Society of America* 104, 3 (Sept. 1998), 1148–1148. https://doi.org/10.1121/1.424266
[26] United Nations. 2015. Universal Declaration of Human Rights. https://www.un.org/en/universal-declaration-human-rights/index.html.
[27] Zacharias Vamvakousis and Rafael Ramirez. 2014. P300 Harmonies: A Brain-Computer Musical Interface. In *Proc. 2014 Int. Computer Music Conf./Sound and Music Computing Conf.* Michigan Publishing, Athens, Greece, 725–729.
[28] Zacharias Vamvakousis and Rafael Ramirez. 2016. The EyeHarp: A Gaze-Controlled Digital Musical Instrument. *Frontiers in Psychology* 7 (2016), article 906. https://doi.org/10.3389/fpsyg.2016.00906
[29] Florian Vogt, Graeme McCaig, Mir Adnan Ali, and Sidney S. Fels. 2002. Tongue'n'Groove: An Ultrasound Based Music Controller. In *Proc. 2nd Int. Conf. on New Interfaces for Musical Expression (NIME 2002)*. NIME, Dublin, Ireland, 60–64.