*Proceedings of the 17th Linux Audio Conference (LAC-19), CCRMA, Stanford University, USA, March 23–26, 2019*

# ISOCHRONOUS CONTROL + AUDIO STREAMS FOR ACOUSTIC INTERFACES

*Max Neupert*

The Center for Haptic Audio Interaction Research
Weimar, Germany
max@chair.audio

*Clemens Wegener*

The Center for Haptic Audio Interaction Research
Weimar, Germany
clemens@chair.audio

## ABSTRACT

An acoustic interface (also: *hybrid controller*) is presented. By tapping, scratching, rubbing, bowing, etc. on the surface, excitation signals for digital resonators (waveguides, lumped models, modal synthesis and sample convolution) are created in synchronicity with augmenting control signals. It is described how a direct acoustic excitation delivers an intimate and intuitive interaction. Questions are raised about which protocols to use for isochronous audio and control transmission as well as file formats. Standardization of such protocols is desirable for future hybrid instruments with analog interfaces. A first step towards standardization is made with the publication of our implementation.

## 1. INTRODUCTION

Recent developments in the musical instrument controller market follow the demand for more expressive and continuous control. At the same time more computing power allows for expensive synthesis methods so that more parameters can be made use of as a continuous stream of control data in several degrees of freedom.

### 1.1. Keys or silicone?

A keyboard of the MIDI standard is generally sufficient to generate the parameters for a simple electronic representation of a piano. Mod-wheel and pitch-bend only extended this affordance mildly. For instruments with a continuous articulation like wind and string instruments the single parameter velocity is inadequate. When *Yamaha* came out with the CS-80 in 1977 it pioneered after-touch on every key and laid the foundations for a class of 'extended keyboards' such as the *Haken Continuum* [1], McPherson's *TouchKeys* [2] and the *Seaboard* [3] by *Roli*. All these instruments make multiple parameters per key available continuously. A standardization effort of these parameter streams lead to the MIDI Polyphonic Expression (MPE) specification. Jones' *Soundplane* [4] and Linn's *Linnstrument* likewise belong to this group of instruments but do away with the traditional (and some may say reactionary) piano key layout.

### 1.2. Exciting audio

A full audio signal is offering even more expression compared to just control-rate parameters. Therefore, contact microphones (piezoelectric sensors) have become a staple of electro-acoustic exploration. They have also found their way in commercial music instruments, but mostly as cheap threshold trigger pads delivering way below their potential. Only a handful of commercially available instruments, namely *Korg's Wavedrum* (1994), Zamborlin's *Mogees* [5] (2014) and the ATV *aframe* [6] (2017) have put them to much more adequate use by feeding the excitation signal into a digital resonator. In the context of research a variety of implementations for experimental and affordable instruments with acoustic interfaces have been proposed. From ceramic tiles as a source for percussive sounds [7], to acrylic sheets instead of guitar strings [8], [9] or intricate prototypes with vibration insulated pads for eight fingers [10].

### 1.3. Marrying control and exciter

Miller's tiles [7] and Momeni's *Caress* [10] consider the processing of the contact microphone as sufficiently expressive. Cook's *Nukelele* [11] combines two sensors, one at audio rate and one at control rate, to create the affordance of an Ukulele which is played with both hands on different positions of the instrument. As one would with a guitar, a hand controls the parameters while the other provides an excitation signal. Former is the control rate input and latter the audio rate input.

The *Kazumi* by Zayas is an instrument which combines capacitive sensing and piezoelectric microphones on the same surface [12]. It features seven separate faces in a prismatic heptagonal shape. Each of the faces has a copper capacitive sensing layer which divides it into six areas from bottom to tip, combined with a piezo mic underneath.

We want to augment the sound signal with additional parameters, so we simultaneously track the **position** of touch on the surface. This way we make a second hand for generating parameters obsolete. (Figure 1) Our implementation creates a percussive instrument which can be hit, but also can be melodic and played in continuous gestures by rubbing, scratching, or bowing on its edge.

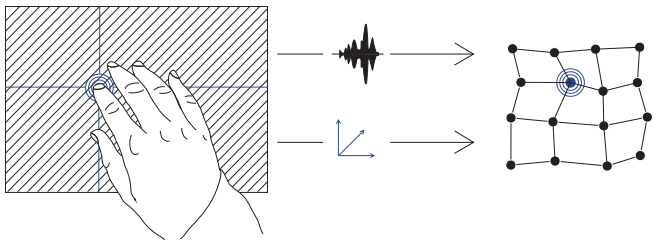

Figure 1: *Hybridity of audio and control data*

### 1.4. Instrument versus controller

Great effort has been put into abstracting controller hardware to become universal input devices for software instruments. The generic controller is an interface to change parameters on the synthesizer in which the actual sound is generated. In our instrument it's not so easy to define where the controller ends, and the instrument starts. Cook writes that "...many of the striking lessons from our history of intimate expressive musical instruments lie in the blurred boundaries between player, controller, and sound producing object." [11].

In our instrument we are using the actual audio signal from the surface which then is fed into a digital filter on the computer. In effect, a significant component of the final sound is defined by the spectrum and gesture of the excitation signal. While in the literature the term 'hybrid controller' is found [9] we prefer to describe the *Tickle* as an 'acoustic interface'. In our opinion 'hybridity' is too generic and there is no declaration of its components, while 'acoustic interface' adds clarity to its nature.

## 2. THE TICKLE

The following section describes the components of the instrument.

### 2.1. Hardware

The case is made of bent steel with wooden side panels. Its top surface is a printed circuit board and has a capacitive touchpad, three endless rotary encoders with associated RGB LED and up/down buttons (for transposition or other parameters). On the back are six ports:

1. External in (if plugged-in it mutes the built-in sensor)
2. CV out Y axis (0-4 V)
3. CV out X axis (0-4 V) or note
4. Host (micro-USB port)
5. Gate or envelope (0-5 V)
6. Excitation (audio signal)

### 2.2. Surface

After a brief evaluation of piano key layouts and variations thereof [13] it was concluded, that a piano key layout is contradictory to the intended interaction with the instrument. A hexagon pattern was chosen to have equal distanced and sized[1] segmentation without empty spaces on the surface. It is also found in other electronic instruments and controllers, for example, the *Synderphonics Manta* [14]. From the 8-Bit resolution in X and Y axis we can calculate in which of the 14 hexagons printed on the surface a touch occurred. The capacitive touch sensing is single-touch, so polyphony cannot be achieved by simultaneous touches. A two or more point gesture will produce erroneous ghosting touch points and thus needs to be avoided while playing. However, with voice allocation we can let one touch resonate while a new touch gets its own resonator, so subsequent touch events may have overlapping resonances.

### 2.3. Material and Texture

To create an acoustic excitation signal we rely on a hard material that captures the spectra of different gestures. In addition to the rigidity of the material, a textured surface is essential to create enough noise when rubbed and wiped. Silicone surfaces are not suitable for our application since they absorb too much of the subtle interaction.

### 2.4. Residual and Resonance

Generally, we want the physical surface of the instrument to resonate as little as possible, so that we can feed the dry residual signal of the touch gesture (rub, scratch, hit, flick, bow etc.) as excitation signal into a digital resonator (See also [7]). This way the full power of physical modeling synthesis algorithms may be accessed. The practice of sending generated noise-bursts or clicks into digital resonators which can be found in literature for physical modeling and which is still the standard in many soft- and hardware implementations is crippling the true potential of such algorithms.

[1] except for the hexagons at the edges

### 2.5. Synthesis

For the sound synthesis we employ techniques of digital reverbrators which at their heart are delay lines, feedback and filters. They can be understood as modeled simulations (waveguides and mass-spring models) of the physics happening in real instruments as described by Smith [15]. These models can be generated with Berdahl and Smith's *Synth-A-Modeler* compiler [16] which has received a graphical interface with Vasil's *SaM-Designer* [17]. *Synth-A-Modeler* generates *Faust* code which can be compiled in a variety of other formats such as a *Pure Data* external. With the *Pure Data* object `pmpd~` from Henry's PMPD [18] library which creates static mass and spring models, we achieved nice sounding string, plate, and gong topologies. However, we are not aiming for perfect recreations of classic instruments, our interest lies in the exploration of synthetic sounds with an acoustic and intimate level of control. Algorithms such as the nested comb filter delay as described by Ahn and Dudas [19] prove interesting and fun to interpret with our instrument while being surprisingly cheap to compute. We can employ our acoustic interface to excite *extended, hybrid* and *abstract cyberinstruments* as described by Kojs et al. [20]. Convolution methods with samples can be useful to digital Foley artists to articulate a sample in a plenitude of variations.


Figure 2: *The Tickle instrument*

### 2.6. Software Architecture and Code

Our hardware is based on a Cypress PSoC 5 microprocessor and runs a firmware which is digitizing the capacitive sensing surface and the signal from the piezoelectric sensor. It communicates to a custom kernel driver which is then communicating to user-space software like our *Pure Data* external or a VST-plugin. Our kernel driver for

Linux as well as the *Pure Data* external are published under a free license. A repository of the source[2] is available (mirrored on github[3]).

### 2.7. Drivers and Communication

A great challenge was to transmit control rate signals married to a stream of audio with a stable latency and reliable offset to each other. The capacitive sensing reports every 4 ms a position while the audio streams with a sample rate of 48 kHz and a block size of 64 samples. Currently the user-space software is expected to match these settings to work reliably. We wrote our own Linux kernel driver receiving this isochronous stream of control and audio rate signals via USB from the device.

## 3. STANDARDS FOR TRANSMISSION AND STORAGE

We believe that acoustic interfaces will soon become a category of their own and manufacturers will introduce hybrid controllers to the market. To make these new devices work with synthesis software there will have to be a standardization effort for interoperability. McMillen and Thew published a proposal on how to send sound spectrum information over MIDI and OSC [21]. However many questions are yet to be answered about which format and standard should be used for audio and data. A plethora of further questions arise when thinking about a possible integration of a track with control and audio-as-synthesis-source into a DAW. With this publication and the open source driver we wish to start a discussion about possible open standards for transmission, storage, and integration of analogue interfaces into the creative workflow of musicians.

### 3.1. Specifications for the Driver

Our aim is an isochronous transfer of data and audio rate signals with minimal latency, and more importantly, with little jitter [22]. The touch position data needs to be present before the audio arrives to be able to tune the synthesis. There can't be any variation to the offset between signal and data. The audio stream doesn't need to be continuous; it could start on the touch event and end with it. In a future polyphonic version, several audio streams could exist in parallel. The implementation could be a data protocol with (multichannel-) audio streaming segments on demand, as well as an continuous audio stream with additional data interwoven. The touch events should refer to a specific sample in the audio, possibly with a timestamp. Other interface data like extra knobs, faders, potentiometers or rotary encoders don't need this precision in timing.

### 3.2. Surveyed Communication Protocols

We've considered different established and experimental protocols. Each was evaluated against the aforementioned goals.

1. A **kernel module** driver was our choice, as it gives us the maximum amount of control to make sure it meets our criteria. However, it needs an installation procedure. On *Windows* and *Mac OS* the operating system vendor restricts who can distribute kernel modules, in fact we have paid *Apple* and applied for kernel signing and are still waiting for any response after 5 months. On Linux, Secure Boot needs to be deactivated or the kernel extension manually signed. A custom kernel driver means additional development overhead and for the customer the fear that the device will be rendered useless if support ends.
2. **Audio spectrum data** (via MIDI or OSC). Another approach would be to break down the audio into metadata and then send this over established protocols like MIDI or OSC which would allow for a partial reconstruction. This was proposed in the aforementioned draft by McMillen [21]. We dismissed this approach because we see it as necessary to include a full audio stream to reduce the latency required for the analysis of such descriptive meta information. It also creates a computational overhead on both, the sending and receiving device.
3. **Audio and MIDI Class Compliant** drivers are a viable alternative. It's possible to use one USB connection providing two virtual devices, an audio interface, and a HID or a MIDI device. Using standards means compatibility, no driver installs and continuous support. However, it's not guaranteed that latency and offset will be consistent. Another problem lies in limitations of popular proprietary DAWs like *Ableton Live*, which will only allow the use of one sound card at a time. Assuming that the sound synthesis happens in a plugin of the DAW, this restriction would block the plugin to access the audio device. Clock drift issues caused by multiple ADC devices may be corrected with adaptive resampling. For Linux with a *Jack* environment it is provided by the `zita-a2j` tool [23].
4. **Control Data as Audio Signal.** Control data may be sent as signals at audio rate, not unlike control voltage in synthesizers or upsampled sensor output in Wessel's *Slabs*, which features 96 channels of audio [24]. It could also be encoded as frequencies and later be decoded with a Fourier transformation like the *Nuance* as described in Michon et al. [25].
5. **MIDI 2.0** There is no indication that MIDI 2.0, which is currently in prototyping stage at the *MIDI Manufacturers Association*, will include the feature to send audio streams for acoustic interfaces.

This list claims no completeness, for example we have not surveyed protocols like *Ultranet* or AVB. It's likely we have overlooked something and there may be a sensible solution to our problem already available.

## 4. FUTURE WORK

Future research may be conducted to implement the following features to the instrument: 1. **Multi-touch** to relieve from ghosting issues when two fingers touch the surface simultaneously. It also allows for **polyphony** later on. 2. **Pressure sensing** [26] either for every point or at least globally for the whole surface. 3. **Haptic feedback** is challenging to implement due to the feedback into the sensor, but can give the user a much more intense sense of reality. The *Lofelt Basslet*[27] is a good demonstration of such a device. 4. Integrated sound synthesis either implemented by a) **analog circuitry** or b) an **embedded computing** platform, for example the *Bela* board [28]. 5. **Playful interfaces** to manipulate mass-spring models in real-time as seen in Allen's *Ruratae* [29].

[2]Source code: https://gitlab.chair.audio/explore/projects
[3]Github mirror: https://github.com/chairaudio

Figure 3: *One of the more unconventional and unintended ways to play the Tickle (Picture from Synthposium Moscow)*

## 5. CONCLUSIONS

Our instrument *Tickle* combines several well-known techniques and technologies which on their own are not new. Touch pad, contact microphone, and physical modeling synthesis have been around for decades. However, in their combination they synergize to a powerful intuitive instrument which allows for a natural and intimate [30] interaction with precise and reproducible control over sound. Feeding an analogue excitation signal into a (digital) resonator can create familiar as well as alien sounds. Sounds which either behave like instruments we know: Violin, guitar, snare drum, cymbal, gong, marimba, etc., or sounds which are distinctly synthetic but have an analogue touch to it.[4]

With this paper we hope to have shown the necessity of sample accurate, low latency and jitter free communications for acoustic interfaces and started a discussion on how to achieve it.

## 6. ACKNOWLEDGMENTS

Our gratitude belongs to Philipp Schmalfuß for coming up with new synthesis techniques as well as Björn Kessler, Joachim Goßmann, Richard Dudas, Sven König for valuable input and feedback. Extra thanks to Brian Bixby for the copy editing and the Sündermann-Oeft family for hosting us during LAC19. We also would like to thank the following institutions: The ESF of the European Union which had funded us for one year through the EXIST! program, as well as Startup incubator *neudeli* and *Bauhaus-Universität Weimar* for access to workshops and network.

[4] A playlist of video demonstrations with the instrument can be found on our website https://chair.audio

## 7. REFERENCES


[1] Lippold Haken, Ed Tellman, and Patrick Wolfe, "An Indiscrete Music Keyboard," *Computer Music Journal*, vol. 22, no. 1, pp. 30, 1998.

[2] Andrew McPherson, "TouchKeys: Capacitive Multi-touch Sensing on a Physical Keyboard," in *Proceedings of the 2012 International Conference on New Interfaces for Musical Expression*, Michigan, 2012.

[3] Roland O. Lamb, *The Seaboard: discreteness and continuity in musical interface design*, Dissertation, Royal College of Art, London, June 2014.

[4] Randall Evan Jones, "Intimate Control for Physical Modeling Synthesis," M.S. thesis, University of Victoria, Victoria, 1993.

[5] Bruno Zamborlin, *Studies on customisation-driven digital music instruments*, Dissertation, Goldsmiths, University of London and Université Pierre et Marie Curie - Paris 6, London, Oct. 2014.

[6] "Product Development Story: ATV aFrame – A Next-Generation Electric Instrument Created through a Fusion of an Acoustic Instrument, DSP Technologies, and Traditional Japanese Craftsmanship," *ICON*, Sept. 2016.

[7] Miller Puckette, "Infuriating Nonlinear Reverberator," in *Proceedings of the 2011 International Computer Music Conference*, Huddersfield, 2011, p. 4, International Computer Music Association.

[8] Daniel Schlessinger and Julius O. Smith III, "The Kalichord: A Physically Modeled Electro-Acoustic Plucked String Instrument," in *Proceedings of the 2009 International Conference on New Interfaces for Musical Expression*, Pittsburgh, June 2009, p. 4.

[9] Romain Michon and Julius O. Smith III, "A Hybrid Guitar Physical Model Controller: The BladeAxe," in *Proceedings of the 2014 International Computer Music Conference*, A. Georgaki and G. Kouroupetroglou, Eds., Athens, 2014, p. 7, International Computer Music Association.

[10] Ali Momeni, "Caress: An Enactive Electro-acoustic Percussive Instrumentfor Caressing Sound," in *Proceedings of the 2015 International Conference on New Interfaces for Musical Expression*, Baton Rouge, May 2015, NIME '15, pp. 245–250.

[11] Perry R. Cook, "Remutualizing the Musical Instrument: Co-Design of Synthesis Algorithms and Controllers," in *Proceedings of the SMAC 2003*, Stockholm, Aug. 2003, p. 4.

[12] Charles Gantt, "Design Challenge Project Summary: Kazumi," Apr. 2016.

[13] Max Neupert, "Optimizing chromatic Keyboards for small, non-tactile Surfaces," in *Proceedings of Korean Electro-Acoustic Music Society's 2017 Annual Conference*, Gwangju, Oct. 2017, pp. 29–31.

[14] Jeff Snyder, "Snyderphonics Manta Controller, a Novel USB Touch-Controller.," in *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression*, Oslo, 2011, pp. 413–416.

[15] Julius O. Smith III, "Physical Modeling Synthesis Update," *Computer Music Journal*, vol. 20, no. 2, pp. 44, 1996.

[16] Edgar Berdahl and Julius O. Smith III, "An Introduction to the Synth-A-Modeler Compiler: Modular and Open-Source Sound Synthesis using Physical Models," in *Proceedings of the 2012 Linux Audio Conference*, Stanford, Apr. 2012, pp. 93–99, CCRMA, Stanford University.

[17] Edgar Berdahl, Peter Vasil, and Andrew Pfalz, "Automatic Visualization and Graphical Editing of Virtual Modeling Networks for the Open-Source Synth-A-Modeler Compiler," in *Haptics: Perception, Devices, Control, and Applications*, Fernando Bello, Hiroyuki Kajimoto, and Yon Visell, Eds., Cham, 2016, pp. 490–500, Springer International Publishing.

[18] Cyrille Henry, "Physical modeling for pure data (PMPD) and real time interaction with an audio synthesis," in *Proceedings of Sound and Music Computing 2004*, Paris, Oct. 2004.

[19] Jae-Hyun Ahn and Richard Dudas, "Musical Applications of Nested Comb Filters for Inharmonic Resonator Effects," in *Proceedings of the 2013 International Computer Music Conference*, Perth, 2013, vol. 2013, pp. 226–231, International Computer Music Association.

[20] Juraj Kojs, Stefania Serafin, and Chris Chafe, "Cyberinstruments via Physical Modeling Synthesis: Compositional Applications," *Leonardo Music Journal*, vol. 17, pp. 61–66, Dec. 2007.

[21] Keith McMillen and Barry Threw, "Acoustic Instrument Message Specification," June 2014.

[22] Robert H. Jack, Tony Stockman, and Andrew McPherson, "Effect of latency on performer interaction and subjective quality assessment of a digital musical instrument," in *Proceedings of the Audio Mostly 2016 on - AM '16*, Norrkoping, Sweden, 2016, pp. 116–123, ACM Press.

[23] Fons Adriaensen, "Controlling adaptive resampling," in *Proceedings of the Linux Audio Conference*, Stanford, Apr. 2012, p. 7.

[24] Adrian Freed, "David Wessel's Slabs: a case study in Preventative Digital Musical Instrument Conservation," in *Proceedings of Sound and Music Computing 2016*, Hamburg, Aug. 2016, p. 5.

[25] Romain Michon, Julius O. Smith III, Chris Chafe, Matthew Wright, and Ge Wang, "Nuance: Adding Multi-Touch Force Detection to the iPad," in *Proceedings of Sound and Music Computing 2016*, Hamburg, Aug. 2016, p. 5.

[26] Adam R Tindale, Ajay Kapur, George Tzanetakis, Peter Driessen, and Andrew Schloss, "A Comparison of Sensor Strategies for Capturing Percussive Gestures," in *Proceedings of the 2005 International Conference on New Interfaces for Musical Expression*, Vancouver, 2005, vol. 2005, p. 4.

[27] Amir Berrezag, "US Patent 2017/0019008 A1: Vibrating Actuator," Jan. 2017.

[28] Andrew McPherson and Victor Zappi, "An Environment for Submillisecond-Latency Audio and Sensor Processing on BeagleBone Black," in *Audio Engineering Society Convention 138*, Warsaw, May 2015.

[29] Andrew S. Allen, *Ruratae: A physics-based audio engine*, Dissertation, University of California, San Diego, San Diego, 2014.

[30] David Wessel and Matthew Wright, "Problems and Prospects for Intimate Musical Control of Computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, 2002.