Proposing "Computer Bending" and "Circuit Vacuuming" as Techniques in Experimental Music

Edgar Berdahl
Louisiana State University
edgarberdahl@lsu.edu

Anthony T. Marasco
Louisiana State University
amarasco@lsu.edu

ABSTRACT

"Computer Bending" is proposed as an experimental technique in which a performer uses a computer to carry out computations of some social significance, while subverting the operation of the hardware or software and listening to the state of the computer as an audio signal at the audio sampling rate. This technique is illustrated using a musical example in which a Raspberry Pi is analyzing gravity wave data.

Also, in contrast with circuit sniffing, it is suggested that "Circuit Vacuuming" can be a fruitful enhancement. In Circuit Vacuuming, the state of a computer or circuit is measured or estimated in a large number of points, to be used for listening over a multichannel loudspeaker setup.

1. INTRODUCTION

1.1 Circuit Bending

Circuit bending can be conceptualized as an experimental technique in which the intended usage of electronic circuits is subverted in order to create strange sounds. It is quite possible that the original circuit designers probably had no idea that their circuits could produce such strange sounds, but with circuit bending, the circuits are led to operate in ways other than originally intended by the designers.

Circuit bending has a history beginning with musicians such as Reed Ghazala. Typically, toys or other battery-powered circuits with audio outputs are modified in order to facilitate the creation of new sounds. Procedurally, Ghazala recommends first listening to toys while probing them, and then adding new wires, buttons, potentiometers and/or other controls, in order to cause the circuits to work differently [1].

Indeed there is something poetic about a Speak & Spell toy being made to grind out weird rhythms, blend vowels into electronic textures, motorboat, growl, grunt, click, and make all sorts of more or less speech-sounding yet very electronic sounds. And there is something magical about the normally very vanilla-sounding Casio SA-2 being brought to generate aleatoric music [1].

Despite the beauty of circuit bending, there are some technical challenges. One of them is that, by using the circuits in ways other than originally intended, they can potentially be broken. Some tricks can be used to help avoid damaging the circuits being bent. For starters, both Ghazala and Collins recommend only hacking with battery-powered devices (this is also for safety) [1, 2]. Moreover, Ghazala recommends bending circuits that operate at only 6 or fewer volts — he reports that in his experience, this significantly reduces the risk of applying too much voltage to sensitive components. For similar reasons, he also recommends not connecting wires too closely to the highest potential of the battery pack [1]. Nonetheless, when engaging in circuit bending, it is advisable to have multiple (ideally identical) devices on hand, so that if one of them gets damaged, one can try moving on to the next device.

1.2 Research Variations on Circuit Bending

Some research is underway on an alternative to circuit bending in which circuit components are never damaged. In "Bit Bending," the circuit elements are simulated in order to produce the sound. In this case, if a virtual component becomes operated outside of its allowable parameters, it can simply be re-instantiated. However, Bit Bending is a research-level endeavor that is currently only being applied to specific circuits [3].

McPherson and Zappi propose another variation on circuit bending called hardware-software feedback loops. In this variation, a low-latency digital controller is connected in feedback with circuit elements in order to create instrument modules [4]. This technique has a lot of potential that is waiting to be explored.

2. APPLYING THESE CONCEPTS WITH COMPUTERS INSTEAD

2.1 Motivation

As the electronic circuits in toys and other audio products becomes more advanced, it becomes more challenging to circuit bend them. One reason for this is that more components are surface-mount, and surface-mount capacitors and resistors may tend to fall off if hobbyists are soldering onto them. Furthermore, the spacing between the pins on the integrated circuits is tending to become smaller, which also makes it harder to solder leads onto them using hobbyist tools.

Concurrently, computers are becoming more advanced and more ubiquitous, so it is gradually becoming more relevant to think about applying circuit bending concepts to computers.

2.2 Challenges

However, circuit bending cannot really be easily applied very well to computers. For one thing, most computers are too expensive to risk breaking them. Also, even if circuit bending a relatively inexpensive computer like a Raspberry Pi, when making a bend, it will tend for either nothing to happen, or the Raspberry Pi will crash, or it will turn off. It is challenging to bend a Raspberry Pi to produce intriguing results as comes more naturally with a Speak & Spell.

2.2.1 Additional Challenge of the Observer Effect

Even some buses on computers operate so fast that there is an additional challenge due to the Observer Effect (colloquially referred to as the Uncertainty Principle) in that on these buses, it is impossible to measure the voltage without changing it [5] so much that the computer can stop working.

Consider the following example. The RAM bus speeds in computers tend to be quite fast. The following was observed with a Dell Precision T1700, which has RAM running at 1600 MHz. At frequencies this high, the wires connecting the CPU to the RAM modules do not behave in the same way as wires do at low frequencies. Instead, they are subject to transmission delays as well as potentially causing and receiving significant electromagnetic interference [6]. A consequence of this is that even connecting a short wire to the pin of a RAM chip, even if this wire doesn't connect to anything, has the potential to crash the computer, because it changes the pattern of voltages appearing on the wire. Indeed, such a crash was observed with the Dell T1700. After such a wire was plugged into the RAM socket in parallel with the RAM, the computer did boot up once, but then it reported an error, it crashed, and then Windows 10 ceased to boot up anymore. Clearly this kind of setup is not so conducive to interactively creating music.

2.3 Circuit Sniffing

Circuit sniffing is a way of skirting the Observer Effect– this is because it usually interferes a lot less with a electrical circuit to hold an inductor near it than it does to connect a wire to it. (Technically, holding the inductor near the computer circuit still affects the circuit, but the authors have not yet managed to crash a computer this way.)

So in circuit sniffing, one holds an inductor (also known as a coil or an electromagnetic pickup) near a computer or other circuit and listens to the audio signal that develops across the inductor. In effect, one is listening to the low-frequency components of electromagnetic signals emanating from the device. This is a mixture of the electromagnetic signals due to a large number of different currents flowing through different parts of the device. For example, one can employ the inductor like a kind of electromagnetic stethoscope, moving it around to different parts of a computer listening to them until finding interesting sounds.

Christina Kubisch has used circuit sniffing to realize what she calls "Electrical Walks." Listeners wear headphones that sniff out electromagnetic signals in the air while they walk around a city, holding their heads near different electronic devices. A documentary video on this subject reveals an amazingly wide variety of sounds that could be sniffed. [1]

Circuit sniffing can also be performed using a radio [7]. Nicolas Collins discusses related issues at length in Chapter 3 "Circuit Sniffing: Using Radios and Coils to Eavesdrop on Hidden Electromagnetic Music" of his book [2]. For example, he writes that, some "of the earliest realizations of computer music were heard through radios placed on top of the central processing units of mainframes: engineers would run programs with instruction cycles whose lengths were calculated to emit a composed sequence of radio frequencies, which were duly demodulated by the radios" [2].

The practice of circuit sniffing of computers emphasizes the fact that "digital" computers are implemented using signals that are actually analog in their physical representation. Due to this property, computers unavoidably emit an orchestra of electromagnetic signals, which can be picked up using inductors or radios. Received as audio signals, these signals vary at the audio sampling rate, depending in detail on the operation of the computer. This procedure results in audio signals that can potentially have remarkably complex structure.

2.4 Databending / Glitch music

Other techniques for computer emphasize the *digital* aspect of the computer. For example, the term *databending* indicates the process of creating something artistic by opening a file of one format as if it had been stored in another format, and then potentially editing it as well [8]. This process induces a particular kind of structure on a media file. When viewing the file as an image or listening to it as music, a listener may not be able to understand all of the aspects of the imposed structure, but she or he can usually perceive some aspects of them. Usually, in this process, an artist applies a technique multiple times in different ways or to different files until arriving at a satisfying result.

Glitch music includes the process of applying databending to audio formats; however, glitch music also encompasses creating sounds using all sorts of failures or errors (whether intentional or unintentional), as well as the music created by using such sounds as samples [9]. For example, early works of glitch music were created by the German group Oval, who mutilated compact discs (CDs) and then sampled the sound of CD players trying to play them back.

Indeed, in today's society, with computers working correctly so often, digital media can sometimes become boring because it can seem so perfect and sanitized. In contrast, the failure of computers can potentially be more interesting, which is part of the databending/glitch aesthetic. By causing computers to fail in some way, new sounds and sound structures can be derived [9].

2.5 Sonification of Computer Activity

The process of sonification also emphasizes the digital and programmable aspect of computers. In sonification, one begins with some important data, and then one runs an algorithm to convert this data into sound for listening. Sonification can be used in products, such as in a Geiger counter, or it can be used for auditory display purposes, or it can be used for artistic purposes [10, 11].

[1] `https://vimeo.com/54846163`

The applications and usages are as wide-ranging as the possibilities enabled by computers. In artistic contexts, this can present a limitation. If there are too many possibilities, then it can be hard to decide how it should sound, and one can get distracted by technical aspects rather than being able to spend enough time on the sound design [12].

Some projects have aimed to sonify computer activity. For example, parallel computation algorithms and other program code for debugging have been sonified [13, 14, 15, 16]. Similarly, network activity has been sonified [17, 18]. Researchers have even sonified the behaviors of the operating system [7]. However, events generating these audio signals have, for the most part, not varied at the audio sampling rate. Accordingly, the actual sounds of the computers are not being listened to, but rather, the actions being taken by the computer are somehow being sonified.

Another variation along these lines is *lstn*, a program by Michael Chinen that uses a variety of tricks to access memory fragments that relate to a running process [19]. *lstn* scans through vast volumes of memory and passes them as audio samples directly to the sound interface.

lstn sounds quite striking, but it produces verbose sounds, which can be hard to relate back to what a running process is actually doing because the sound is not precisely synchronized with the running process. One of the reasons for this is that sonification by way of an operating system is limited by the speed at which the operating system processes interrupts. Most common operating systems do not process interrupts much faster than once per millisecond, so typically these approaches will not result in audio signals that are maximally representative of precisely what the computer is doing.

3. COMPUTER BENDING

3.1 Concept

In contrast, the authors are currently interested in an approach that emphasizes electronics and the aesthetic of listening directly to what the computer is doing at the audio sampling rate in an artistic context. Accordingly, **the authors would like to define "Computer Bending" as the following: when a performer runs some computations of social significance on a computer, while subverting the operation of the hardware or software and listening to the state of the computer at the audio sampling rate.**

Unorthodox methods of (mis)using a computer are concurrently employed in order to make the sound more interesting. This generally involves pushing the computer to its limits in some way. For example, running multiple computationally-taxing applications simultaneously, overwriting and then reading a disk, interrupting system actions improperly, reading corrupt files, etc. can be incorporated into the practice of Computer Bending. This results in the operation of the computer or software being subverted in an artistic way.

Computer Bending is therefore a form of glitch music that places an emphasis on the computations of social significance. However, in contrast with circuit bending, the computer's circuitry does not necessarily need to be altered, sparing the device from any potential damage.

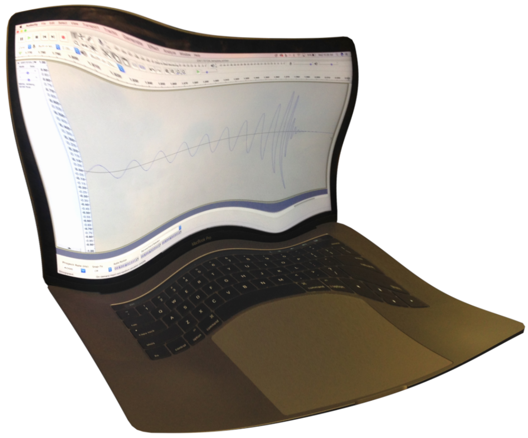

Figure 1. Metaphorical representation of Computer Bending.

3.2 Example

To test out some of these ideas, the first author created the work *Sound of Computing the Signal Analysis of Gravity Waves*. This work also featured the fact that small computers are getting so powerful that even a Raspberry Pi can analyze gravity wave signals. Indeed, the python software from the Gravitational Wave Open Science Center could be installed onto a Raspberry Pi [20]. Figure 6 shows demos from the Gravitational Wave Open Science Center running in the lower right and upper left windows. The bottom left window shows the messages from the playback of "empty" gravity wave sensor data as audio, and the top right window (see Figure 6) shows the video output from the Raspberry Pi board itself.

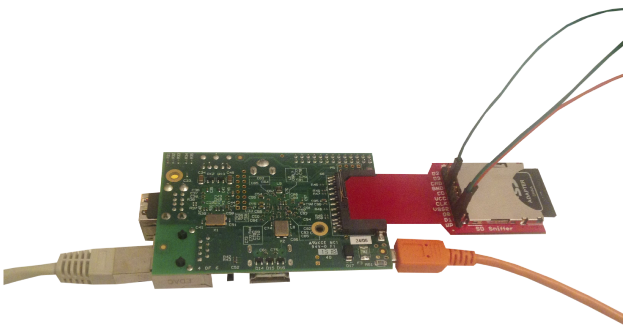

Figure 2. Raspberry Pi shown with "SD Sniffer" from `sparkfun.com` used for sampling the D0 and D1 data lines of the SD memory card.

For the creation of this work, various tasks such as these were run on the Raspberry Pi. During this time, the audio output from the Raspberry Pi could be heard. Also, the sound of the typing itself could be heard. In addition, electrical voltage signals from the SD memory were listened to. This was accomplished by way of an "SD Sniffer" as shown in Figure 2.

However, the data lines of the SD cards are sensitive to the input impedance of audio amplifiers. It is therefore best to use high-impedance buffer circuits to buffer these signals before connecting them to sound interfaces. In this context, the issue of the Observer Effect is accordingly mitigated.

A recorded performance of this work can be viewed at the link down below. [2] The different voltage signals can be heard oscillating in related ways but nonetheless being somewhat independent. Indeed most of the time in the composition, these signals can all be heard as being distinct signals.

Yet *Sound of Computing the Signal Analysis of Gravity Waves* is more than the sum of all of its signals alone. This is because the signals oscillate according to intriguing and complex structures, and the computer activity is also relating to a socially relevant task, the analysis of gravity wave data.

4. CIRCUIT VACUUMING

4.1 Concept

The prior work resulted in the realization that listening to multiple signals at the audio sampling rate could expose even more structure in the signals. Therefore, the authors are suggesting another concept. **Circuit Vacuuming is proposed as being the practice of measuring or estimating the state of a computer or circuit at a large number of points, to be used for listening over a multichannel loudspeaker setup.** Circuit Vacuuming can be carried out by directly sampling circuit signals such as using buffers, or it can be carried out using inductors like the telephone coil recommended by Nicolas Collins [2].

Figure 3 shows a metaphorical representation of Circuit Vacuuming, in which a metaphorical *circuit vacuum* is used to measure the state of a circuit over a large number of points. Then the circuit vacuum passes these circuit signals on to a multichannel loudspeaker setup, so that the signals can be heard in a spatial setting (see Figure 3). It needs to be a multichannel loudspeaker setup in order for it to truly be Circuit Vacuuming.

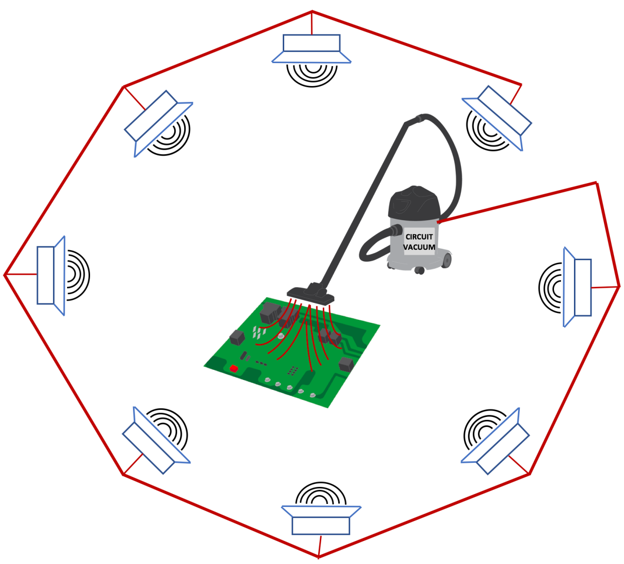

Figure 3. Metaphorical representation of Circuit Vacuuming with eight channels of sound for listening .

[2] http://edgarberdahl.com/%20listen/2018/10/01/SoundComputingGravityWaves.html

4.2 Independent Components Analysis

With multichannel data at one's disposal, it is intriguing to think about applying source separation techniques to reveal more structure in the data. The authors have investigated how to do this using the FastICA (Fast Independent Components Analysis) algorithm [21].

Consider the case of a Dell Precision T1700. As mentioned in Section 2.2.1, the Observer Effect made it too difficult to sample the DDR memory lines directly. However, six telephone coils could be placed along the integrated circuits on the DDR memory modules. A sample recording over the course of a few minutes is displayed in Figure 4, which shows that the amplitude envelopes of the signals all tend to behave rather similarly while the computer is working.

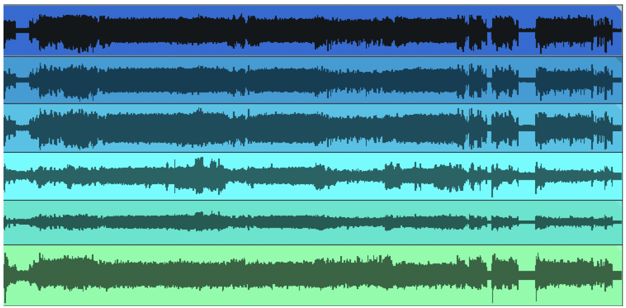

Figure 4. Unseparated channels. Here the input signals are displayed from six telephone coils that were placed on the DDR RAM chips in a Dell Precision T1700.

In contrast with the work in Section 3.2, in which independent voltage signals were measured directly, the signals measured here by the telephone coils are each due to a large number of source currents in the DDR memory modules (and elsewhere nearby) in the computer. Therefore, these signals can be considered to be a memoryless mixture of some more fundamental signals.

This is where the FastICA algorithm comes in. The algorithm can take signal mixtures and separate them into component signals that it considers to be more independent. Specifically, the FastICA algorithm looks for signals that are statistically independent and non-Gaussian, and it tries to isolate these signals to reveal more structure [21].

When the FastICA algorithm was applied to the signals shown in Figure 4, it was able to produce "unmixed" signals that seemed to be more independent. This can be observed in Figure 5, whose amplitude envelopes can be seen to be varying with somewhat more structure than in Figure 4.

In informal listening tests with this data and some other data, the FastICA algorithm seemed to be able to reveal some structures that were otherwise harder to hear in the data. In any case, in the view of the authors as composers, the FastICA seemed, at least so far, to be a useful step for preparing more independent signals with Circuit Vacuuming.

4.3 Example

The second author created the work *Sifting through Digital Sands*, an eight-channel fixed-media etude composed through the Circuit Vacuuming of a Dell Precision T1700

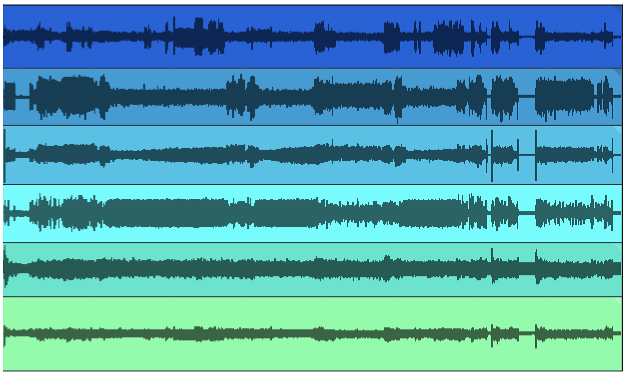

Figure 5. Channels separated to some extent using the FastICA algorithm. More structure seems to be revealed in the signals.

desktop PC. The source audio was recorded by affixing twelve telephone coil inductors around the interior of the computer, namely upon the DDR memory modules (channels 1-6), an interior fan (channels 7), the graphics card module (channel 8), the PSU (channel 9), the hard drive (channel 10), and two flatpack IC's on the motherboard (channels 11-12). During the recording process, various tasks were run on the PC, including working within Maya Motion Graphics and Rhino 3D CAD, running computationally intensive projects through BOINC, and putting the PC into and out of sleep/wake modes.

The recorded data was then source-separated using the FastICA algorithm and loaded into a digital audio workstation. The sound clips were re-arranged there in order to create *Sifting through Digital Sands*.

A recorded version of this work can be heard at the link below.[3] The most energetic moments of audio were produced by the DDR memory modules, especially during moments of sleep/wake state changes and the resuming of a BOINC data processing session. The motherboard IC's sporadically generated short bursts of pitched activity, mainly during the initial moments of waking or putting the PC to sleep. Including a stronger preamp coupled with a tighter fit around these ICs could reveal more interesting audio results during normal use cases in the future.

Circuit Vacuuming appears to be a fruitful way of presenting data derived from computer hardware in a large number of channels. The FastICA algorithm could potentially be used to reveal more structure in the data. It would be interesting to consider whether there are any other applications in which the FastICA algorithm can reveal compositional structure in audio recordings.

5. CONCLUSIONS AND FUTURE WORK

Following a review of related prior concepts, the authors have proposed the techniques of "Computer Bending" and "Circuit Vacuuming." These have been illustrated by way of two compositions. These techniques seem widely applicable and could be used to create a large number of future compositions.

This topic might eventually be explored within the context of cyber-security. The authors are planning to build an array of 24 inductors into a tabletop on which a laptop can be placed. It will be interesting to listen to these audio signals in 24 channels while running programs of cyber-security interest and testing whether people can hear the sound of a password being checked, of encryption keys being used, or other security tasks.

[3] `https://drive.google.com/open?id=1xsNnN7WxgYo5rO0isG4cBRSth-wypfJn`

6. REFERENCES

[1] R. Ghazala, *Circuit-Bending: Build your own alien instruments*. Indianapolis, IN, USA: John Wiley and Sons, 2005.

[2] N. Collins, *Handmade Electronic Music: The Art of Hardware Hacking*. New York, NY, USA: Routledge, 2014.

[3] K. J. Werner and M. Sanganeria, "Bit Bending: An Introduction," in *Proceedings of the International Conference on Digital Audio Effects (DAFx-16), Maynooth, Ireland*, 2013.

[4] A. McPherson and V. Zappi, "Exposing the scaffolding of digital instruments with hardware-software feedback loops." in *NIME*, 2015, pp. 162–167.

[5] U. S. Inan and A. S. Inan, *Engineering Electromagnetics*. Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[6] T. Granberg, *Handbook of digital techniques for high-speed design*. Upper Saddle River, NJ, USA: Prentice Hall, 2004.

[7] A. M. Sarroff, P. Hermans, and S. Bratus, "SOS: Sonify Your Operating System," in *Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research*, Marseille, France, 2013.

[8] D. Geere, "Glitch art created by databending," *Wired Magazine*, August, 17, 2010.

[9] K. Cascone, "The aesthetics of failure: 'Post-digital' tendencies in contemporary computer music," *Computer Music Journal*, vol. 24, no. 4, pp. 12–18, 2000.

[10] G. Kramer, *Auditory display: Sonification, audification and auditory interfaces*. Addison-Wesley, 2000.

[11] S. Barrass, "The aesthetic turn in sonification towards a social and cultural medium," *AI & society*, vol. 27, no. 2, pp. 177–181, 2012.

[12] P. R. Cook, "Re-Designing Principles for Computer Music Controllers: A Case Study of SqueezeVox Maggie," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Pittsburgh, PA, United States, 2009, pp. 218–221.

[13] J. M. Francioni and J. A. Jackson, "Breaking the silence: Auralization of parallel program behavior," *Journal of Parallel and Distributed Computing*, vol. 18, no. 2, pp. 181–194, 1993.

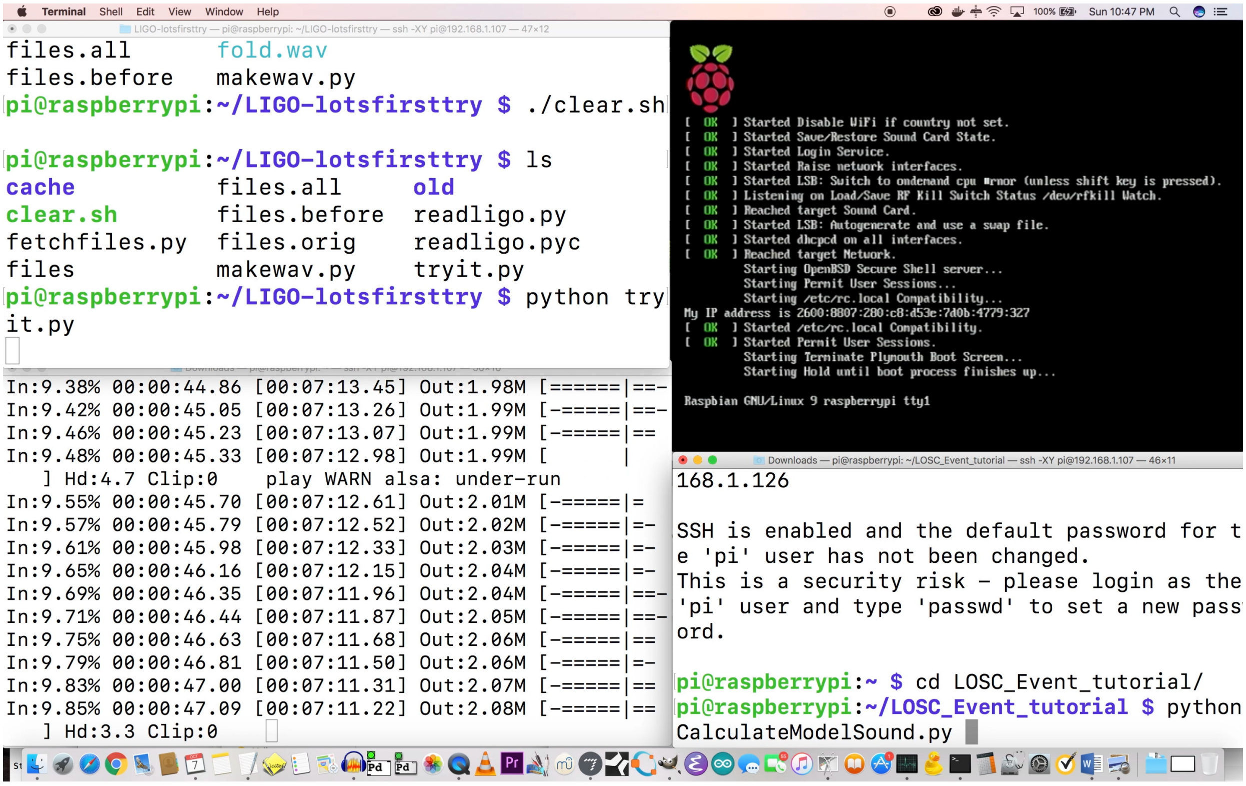

Figure 6. Screenshot of terminal windows running remotely on a Raspberry Pi during a performance of *Sound of Computing the Signal Analysis of Gravity Waves*.

[14] C. J. DiGiano and R. M. Baecker, "Program auralization: Sound enhancements to the programming environment," in *Proceedings of the conference on Graphics interface'92*. Morgan Kaufmann Publishers Inc., 1992, pp. 44–52.

[15] D. B. Boardman, G. Greene, V. Khandelwal, and A. P. Mathur, "LISTEN: A tool to investigate the use of sound for the analysis of program behavior," in *Computer Software and Applications Conference, 1995. COMPSAC 95. Proceedings., Nineteenth Annual International*. IEEE, 1995, pp. 184–189.

[16] P. Vickers and J. L. Alty, "Siren songs and swan songs debugging with music," *Communications of the ACM*, vol. 46, no. 7, pp. 86–93, 2003.

[17] M. Ballora, N. A. Giacobe, and D. L. Hall, "Songs of cyberspace: an update on sonifications of network traffic to support situational awareness," in *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications 2011*, vol. 8064. International Society for Optics and Photonics, 2011, p. 80640P.

[18] M. Gilfix and A. L. Couch, "Peep (The Network Auralizer): Monitoring Your Network with Sound." in *LISA*, 2000, pp. 109–117.

[19] M. Chinen, "lstn," Prezi presentation, 2010, https://prezi.com/qvjy3_tyxtbm/lstn-slow/.

[20] M. Vallisneri, J. Kanner, R. Williams, A. Weinstein, and B. Stephens, "The LIGO open science center," in *Journal of Physics: Conference Series*, vol. 610, no. 1. IOP Publishing, 2015, p. 012021.

[21] A. Hyvärinen, J. Karhunen, and E. Oja, "Independent Component Analysis. Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control," 2001.