# Auditory Display For Maximizing Engagement and Attentive Capacity

by

Juliana Mae Cherston

BA in Physics, Harvard University (2013)



Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Masters in Media Arts and Sciences

[Master of Science] at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2016



Author ..... Signature redacted ....................
Program in Media Arts and Sciences

Certified by..... Signature redacted ...............
Joseph A. Paradiso
Alexander W. Dreyfoos (1954) Professor in Media Arts and Sciences
Program in Media Arts and Sciences
Thesis Supervisor

Accepted by ........ Signature redacted ....................
Pattie Maes
Academic Head
Program in Media Arts and Sciences

# Auditory Display For Maximizing Engagement and Attentive Capacity

by

Juliana Mae Cherston

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning
on August 24th, 2016, in partial fulfillment of the
requirements for the degree of
Masters in Media Arts and Sciences

## Abstract

Two projects in scientific data sonification are presented. 'Quantizer' is a platform that enables composers to develop artistic sonification schemes using real-time data from the ATLAS detector at CERN. Three sample audio streams are available for real-time consumption by the public and the public engagement potential for the project is studied. 'Rotator' uses sonification as a practical tool for analysis of high dimensional data. Users can swipe data between their auditory and visual channels in order to best perceive the structure of a dataset. A dual audio-visual presentation mode is found to be a promising alternative to use of a purely visual display mode. Thesis Supervisor: Joseph A. Paradiso Title: Alexander W. Dreyfoos (1954) Professor in Media Arts and Sciences

Thesis Supervisor: Joseph A. Paradiso
Title: Alexander W. Dreyfoos (1954) Professor in Media Arts and Sciences, Program in Media Arts and Sciences

**Auditory Display For Maximizing Engagement and Attentive Capacity**

by

Juliana Mae Cherston

The following people served as readers for this thesis:

Thesis Reader ....................................................
Joseph A. Paradiso
Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Thesis Reader ....................................................
Seth Lloyd
Professor of Mechanical Engineering
Department of Mechanical Engineering

Thesis Reader ....................................................
Sepandar Kamvar
Associate Professor
LG Career Development Professor of Media Arts and Sciences

**Auditory Display For Maximizing Engagement and Attentive Capacity**

by

Juliana Mae Cherston

The following people served as readers for this thesis:

Signature redacted

Thesis Reader ................................................................
Joseph A. Paradiso
Professor of Media Arts and Sciences
Program in Media Arts and Sciences

Signature redacted

Thesis Reader ................................................................
Seth Lloyd
Professor of Mechanical Engineering
Department of Mechanical Engineering

Signature redacted

Thesis Reader ................................................................
Sepandar Kamvar
Associate Professor
LG Career Development Professor of Media Arts and Sciences

# Acknowledgements

Thank you to my friends in the Responsive Environment group, and to my academic advisor, Joe Paradiso, for his mind-opening academic perspective throughout the last two years leading up to the submission of this thesis. Thank you to my readers Seth Lloyd and Sep Kamvar for helping to broaden application areas.

I extend sincere gratitude to Natasha Jaques for going out of her way to brainstorm concrete application areas for the Rotator project. Thank you also to both Natasha Jaques and to Sara Taylor for sharing an excerpt from the Snapshot dataset and for formatting the data appropriately on my behalf. Thank you to Prof. Paolo Zuccon for providing temperature data from the AMS control room and for meeting with me. Thank you to Evan Lynch and to Akito Von Troyer for contributing audio compositions to the Quantizer project when these contributions were most urgently needed, and thank you to Ewan Hill and to the rest of the ATLAS Outreach team for growing to play such a substantial role on the Quantizer project. Thank you to Gershon Dublon for helping me to think with more breadth and perspective about the purpose of a thesis.

A big hug to Gavin Lund for helping me to brainstorm, for patiently teaching me new concepts in audio synthesis, and for showing such genuine care and support throughout this whole process.

Thanks to Ina Jazic, Bianca Datta, Juliana Nazare, Cristian Ignacio Jara Figueroa, Martin Saveski, and Xavi Benavides for emotional support and for many cheer ups!

Thank you to my rock-star sister Caroline Cherston for sharing wise advice and for continuing to be such a partner-in-crime.

Finally, thank you to my parents for their love and above-and-beyond levels of dedication, as well as for their patience with me during the last 6 months.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Listening in on the animated chatter amongst physicists at the main restaurant of the European Organization for Nuclear Research (CERN) these days, one might guess that the 2015 boost of the Large Hadron Collider (LHC) from 7 TeV to 13 TeV has channeled energy not only into the protons subject to collision but also into the physicists themselves, most of whom are tasked with studying the properties of these collisions. Science, ultimately, is a human pursuit that - much like any other domain of research - is subject to the creative ingenuity of the theorist or the analytical capacity of the experimentalist. An algorithm does not generally write itself[1]; rarely do patterns in sometimes terabyte-sized datasets self-identify. New advents in machine learning and artificial intelligence (AI) may begin to challenge such bold claims about the inevitable role of the human in scientific research, but for the time being, even the most sophisticated work in data analysis is dependent on the human either extracting known features from data, or else methodically tuning an algorithm in the hopes of stumbling on a new and meaningful pattern. In the domain of particle physics, extracting a known feature may mean, for instance, that the researcher specifically

[1]If an algorithm is auto-generated, a human has still written the algorithm-generating algorithm

searches for two photons with particular resonance mass, transverse momentum, and polar angle collectively known with some degree of certainty to indicate the decay of a pion. On the other hand, a more exploratory particle physics search to stumble on new patterns could involve a researcher hunting through a minimum bias data stream to locate either wholly unexpected excesses in energy, or evidence for the existence of a more obscure particle whose existence is posited by a theorist. For example, perhaps a w-boson extracted from the data is a decay product of a theorized particle from an exploratory search. These two modes of analytical approach - one categorized by application of known patterns to a raw dataset, and one categorized by more freeform exploration, are prevalent in many fields of academic research, and both approaches remain subject to the human capabilities of the researcher (which, as the reader may well know, is in itself the basis for a wide field of research known as human factors in computing).

In the first case, a researcher is subject to human error, and thus will seek to verify and validate his results. Perhaps noise was improperly removed from the data, or perhaps an algorithm was incorrectly encoded. In the second case, a researcher is subject to his own often subjective choices about where to hunt for patterns. Visual representation, praised for its ability to rapidly convey information, is commonly used both for validation and for exploration. In addition to researchers receiving extensive training in scrutinizing plots, visual display technologies long preceded technologies for presenting data in alternative sensory modes, perhaps leading to the now prominent conception of visualization as a cornerstone of academic research.

There is undoubtedly a place for visualization. In particle physics, The Feynman diagram is a fine example of a graphical notation that has been constructed with a specific application area in mind: it rapidly and effectively describes complicated particle decay processes on the basis of the fundamental particles and force couplings involved. On the other hand, one begins to wonder whether alternative modes of data representation may better serve the researcher in specific contexts.

In the last few decades, a research community has formed to study sonification, the representation of data as sound both for aesthetic and practical purposes. Sonification has strong appeal: an audio signal is defined by a large number of parameters that are readily-perceived by the listener, making it well-suited for represention of high-dimensional data. One is also immediately convinced of the usefulness of sonification in cases where the user wishes to receive data-driven feedback while performing a task that occupies their visual attention, such as a neurosurgeon wishing to determine whether an electrode has been implanted into the proper brain region.

But, critical to much of this thesis is the notion that sound is also an art form; data-driven audio creates an opportunity for artists to engage with the same datasets as do scientists, forming a new artistic medium for outreach, education, and artistic expression. The increasing prevalence of real-time sensors everywhere suggests that the time is ripe for such an artistic movement to to fluorish. The Quantizer project, composing an extended chapter of this thesis (chapter 2), is a reaction to the growing opportunity for real-time sonic expression, in this case for the purpose of exposing high energy physics data as an artistic tool and for promoting physics to a broader audience.

While there is a flurry of excitement around sonification, the bulk of existing literature presents tools and methods focused heavily on auditory display. However, in practice, we ought to strive to place both our eyes and our ears where they are best suited when analyzing data in order to maximize our bandwidth when studying a high-dimensional dataset offline, and maximize our attentive capacity when monitoring data in real-time (such as in an experimental control room). The Rotator tool, composing the bulk of this thesis, enables researchers to strategically divide incoming data between their eyes and ears. Here we have studied the perceptual influences of varied display modes on the researchers ability to characterize data.

The motivation for an audio-visual analysis interface was born out of a significantly more abstract proposal for a rotational interface consisting of data encoded in a geometrically high-dimensional audio-visual entity. At any given moment, up to 3-dimensions of data would be projected visually, and the remaining dimensions would be projected into the auditory domain and used to control auditory features. The user would then rotate this geometrically high-dimensional audio-visual object in order to adjust the auditory and visual projections and thus perhaps begin to make sense of the high-dimensional data as a whole. This idea is still of great interest, and addressed in the concluding statements of this thesis, (chapter 8) but the perceptual challenges for the listener are sure to be fierce, and therefore the application spaces still somewhat vague. The Rotator application, in which users can drag audio and visual boxes around the screen in order to dynamically set data display modes, is seen as a more concrete iteration of the full-fledged abstraction that has more immediate potential. In fact, three application areas for the interface are suggested in chapter 6 and one application area is studied more carefully in chapter 7.

It is critical to recognize that while application areas are deeply considered, this thesis is ultimately aimed at presenting a tool used to study human perception of data under varying display modes and is not a study in any particular application space. Effective categorization of the influence of the tool on our perceptual capabilities is a critical step in subsequently determining suitable application areas. Bearing this framing in mind, the evaluation section of this thesis (chapter 7) studies the influence of the tool on a users ability to characterize high-dimensional data. In the future, the results of this evaluation could possibly guide sonification work in the previously described geometrically high-dimensional audio-visual space.

This thesis serves as a preliminary contribution in its own right to the study of human perception, both in the domain of scientific outreach and in the domain of scientific research. However, I can ultimately imagine deepening Rotator into a ready-to-use tool for data analysis or monitoring in the field of physics, for example, using the perceptual results of this thesis to guide the project.

# Chapter 2

# Quantizer

## 2.1 Resources

Two shorter papers on the Quantizer platform have been published and serve as useful, compact references for the project (see [42] and [41]). The Quantizer webiste is available at **quantizer.media.mit.edu**. Additionally, code is available on Github [2]. The project's Soundcloud account is available as well [3].

## 2.2 Overview

The Quantizer platform is an application that promotes the use of scientific data for aesthetic purposes by enabling users to drive musical compositions using real-time experimental particle physics data. From the earliest days of human artistic pursuit, the vision of the artist is constrained by the tools available in the relevant epoch. Early Egyptian painters only had certain earthly pigments available to them and until 2600 BC could not synthesize blue pigment; the advent of the photographic camera in the early 19th century triggered a never-before-explored mode of artistic expression; musical instrument quality and design throughout modern history is a direct function of fabrication modes available.

However, one element of the artistic process that has remained true until very recently is that the artist maintains complete reign over their artistic output. For instance, the musician composes each note of a musical piece, and the painter draws each brushstroke. Recent, significant advancements in real-time audio synthesis, combined with the growing prevalence of real-time sensor data from sensors everywhere, ushers in a growing opportunity to incorporate sensor data directly into a composer's artistic vision, enabling artists to design instruments that are then played by data through predefined data-to-audio mapping schemes. The musician's role in this case is adapted from possessing full control over the musical piece to possessing control only over the framework. The musician designs the instrument but data pluck the strings. The resulting music reflects characteristics of the data driving it, but it is ultimately an aesthetic experience.

The premise of the Quantizer project is to expose a stream of real-time particle collision data from the ATLAS detector at CERN through a set of tools that enable this data to drive unique musical experiences crafted by each composer. The audio from select compositions is streamed to a website for real-time consumption by the public, much like a radio with multiple available musical channels. One recalls that the radio is heralded as a unifying force in early 20th century America. Families once gathered around radios during times of political turmoil to listen to Roosevelt's famous Fireside Chats or to experience live streaming of musical performances. The radio enabled a performer or speaker, for the first time in history, to address the nation at once. Quantizer's website can be framed in a similar - though more mild - sense, encouraging a trend in which members of the public can simultaneously tune into experimental data through an auditory channel for entertainment purposes or otherwise. For composers, it is now possible to interact with real-time experimental physics data through tools that are founded on a new degree of playfulness.

## 2.3 Prior Art

This project builds upon a consider number of sound-based works inspired by physics research that integrate experimental data to varying degrees.

### 2.3.1 Audio Inspired by High Energy Physics

Beginning in the early 1990s, a parody pop group of CERN employees known as Les Horribles Cernettes performed physics-inspired music at high energy physics events, with lyrics that lamented marriage with a high-energy physicist constantly distracted by his experiments [37]. One of their most famous pieces, 'Collider', was performed periodically at CERN until 2012 [45].

More recently, Ryoji Ikeda, winner of the Collide@CERN prize, has built an audio-visual soundscape called Supersymmetry based on physics concepts learned during an artistic residency at CERN [76]. Ikeda is well known for several works artistic sonification of scientific data [88] [63]. The exhibit is staged in the upper level of a dark parking lot and consists of two parts: an 'experiment'- tables featuring moving axes and hypnotic sounds and light patterns, and an 'experience,' consisting of fast-changing sounds and patterns symbolizing complex data pouring into a control room.

On the one hand, Jonathan Jones, art critic of the Guardian newspaper, responded to Ikeda's exhibit in the following critical manner:

> Ikedas installation Supersymmetry, staged in the darkened uppermost level of a multistory car park, is apparently what you get when you introduce an artist to the worlds most advanced particle research institute and

> its renowned Large Hadron Collider. A lot of sound and light, signifying nothing. Why does CERN want artists to respond to it anyway?" [9].

On the other hand, Georgescu and Levi comment that the detachment between scientist and object of study is magnified in experiments at CERN and renders alternative modes of expression all the more compelling:

> A good portion of today's research in physics does not rely on direct interaction with the objects of study, simply because these are outside the reach of our senses and exist on extreme time and length scales. But in particle physics and, in particular, at CERN, the distance between us and the physical phenomena is pushed to another level of complexity. It is perhaps this very complexity that inspired the work of Ikeda [54].

In these reactions to Ikeda's work, one immediately recognizes a tension between the hyper-rational spectator who sees artistic reaction to scientific experimentation as devoid of meaning, and the connection-seeking spectator who observes that as experiments grow in complexity and scale, we ought to find new ways to create tangible and compelling associations between ourselves and the physical phenomena under scrutiny.

### 2.3.2 Direct, Offline Use of Physics Data in Audio

While the aforementioned audio-based worked were broadly inspired by CERN, a set of projects have made more direct use of scientific data to drive artistic vision.

Some of the pioneering work in scientific data sonification emerged from large scientific research labs. In 1970, composer Charles Dodge worked with Bell Laboratories to compose music using 2920 measurements of Earth's magnetic field over the course of the year. The data points were mapped to a four octave span, with interpolations between the data points added to create tempo and register [62]. In the 1990s, sound artists used planetary data collected by NASA's Voyager I and II to produce a series of recordings called 'Symphonies of the Planets', in which ambient space sounds coming from vibrations of the planets, electromagnetic fields of planets and moons, planetary magneto-spheres, radio waves bounding between the planets and their inner atmospheres, charged particle interactions of the planets, solar winds, and charged particle emissions from the rings of some planets were used as content in a musical record [17]. Fiorella Terenzi, an astrophysicist and composer, has similarly developed techniques for recording radio waves from distant galaxies for musical composition, coining the term 'acoustic astronomy' [89], and Stephen P. McGreevy has built RF receivers to listen directly to sounds from the Aura Borealis (from which he as produced several commercial recordings) and other types of what he calls 'electromagnetic smog [66].

In terms of more recent sonification projects affiliated with experiments at CERN, a project was completed in 2010 as a collaboration between the ALICE experiment and the University of Music and Dramatic Arts Graz, Austria in which simulated particle trajectories through the ALICE detector's time projection chamber are mapped to spatialized audio [94]. The simulated dataset used for this project contains current spike measurements caused by particles ionizing gas in the time projection chamber of the detector. The resulting audio is spatialized with respect to the center of the detector in order to enable perceptual grouping of the data into particle tracks, an effective means of using auditory pattern recognition to achieve a feeling for the structure of the raw data. In order to further enhance the perceptual separation of tracks, overtones are added to the base frequency and weighted based on a linear mapping of the data as a function of the $\phi$ angle of the data point in cylindrical coordinates. Furthermore, the charge deposit of each electron influences the level of impulse that excites a filter bank of resonators. In this way, we see that the physics goal of recognizing the presence of track formations guide sonification choices. One downside to the rendering methods used is that each event takes between 10 and 20 minutes to generate, thus eliminating the possibility for real-time processing.

One parameter-mapping sonification by Rhoades uses quantities drawn from ATLAS detector data to drive parametric variables in a custom Csound instrument. Rhoades was granted access to several datasets collected by ATLAS, and used a number of calculated physics parameters including transverse mass, electromagnetic fraction, and missing energy in the sonification. The sonification process used for this project is described in depth in [81]. The compositions include varying degrees of post-processing and editing of the audio, a benefit of composing sonifications using offline data that is not possible in real-time applications.

Rather than use experimental data itself, Bill Fontana, another winner of the Collide@CERN prize, produced sound art using compilations of audio recorded directly from components of the collider including magnets and cooling systems [28].

In a similar spirit, Jo Thomas visited the Diamond Light Source synchrotron in Oxfordshire, UK and discovered that audio output is already used by researchers to monitor the beam. Inspired by this, she has incorporated audio produced from the Diamond Light Source synchrotron into a sound art installation [30].

More recent audio projects making direct use of high energy physics data have sonified the now-famous plot that provides 5 sigma evidence for the existence of the Higgs Boson particle [29][46].

### 2.3.3 Historical Examples of Audio Use In Research at CERN

There exist additional sporadic examples of the use of audio for data monitoring in the context of particle physics. For example, since 1928, Geiger-Mueller tubes have enabled physicists to work on machinery in risky settings while using the auditory channel to monitor risk of radiation exposure (early versions of the Geiger-Mueller counter required such high voltage levels that a sparkover would result in an audible bang.)

In the 1960s, CERN developed a device called a sonic chamber for particle localization. Specifically, in 1962, B. Maglic of CERN used audio recordings from two microphones to determine the position of a spark to the accuracy and precision of the chamber resolution by measuring the time interval between initiation of the discharge and the arrival of the sound of the leading edge of the shockwave in the chamber gas. The two delay times were used to reconstruct the position of the spark in the plane. To perform this conversion, a gated oscillator was turned on by the chamber trigger and off by the probe signals, and dedicated circuitry kept track of the clock pulse count [51].

Critically, the simplicity of using a sonic approach in the spark chamber experiment meant that the chamber optical system (consisting of a relatively complicated array of lenses, prisms, and mirrors) was no longer necessary. That said, sonic chambers were better suited to detection of only single tracks due to the difficulty of controlling reflected wave fronts that arrive after the direct wave from the spark [51].

In limited settings, audio is used directly for detector calibration at CERN. For example, microphones have been mounted in the LHC tunnel in order to monitor the performance of collimators, devices that remove stray particles from the particle beam. If a collimator is hit, it may be damaged, and rather than examining each collimator for damage, a study on the auditory output of the collimator-beam collision was conducted. Beam impacts were successfully detected using acoustic sensors in collimator test procedures in 2004 and 2006 [39]. The audio enables listeners to identify which of the hundreds of collimators was hit as well as deduce whether damage was incurred. Other radiation-resistant sensors such as accelerometers attached to the collimators were also used in experiments. Sounds from collimator experiments in the LHC taken in 2011 are available here [10]. Additionally, a 2005 LHC Progress Report made some interesting observations regarding direct audification of beam oscillations [53]. Specifically, particles in an accelerator follow sinusoidal motion known

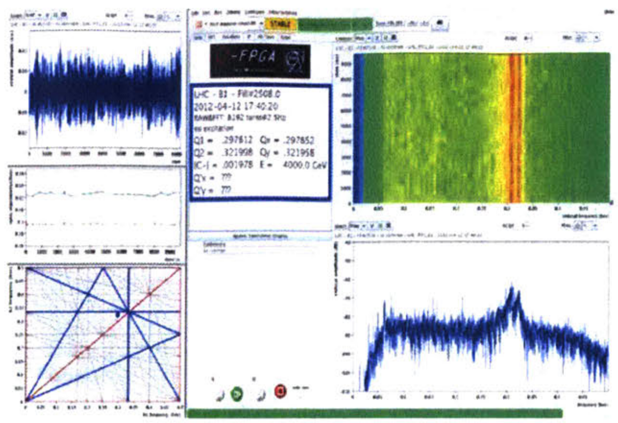


Figure 2-1: The LHC Tune Viewer interface used for auditory monitoring purposes in the LHC BBQ experiemnt

as betatron oscillations. If the number of periods of these oscillations per turn is an integer number then the amplitudes of these oscillations will increase due to the magnetic forces keeping the particle trajectories stable. Work to stabilize the beam is further complicated by the fact that each bunch of protons contains many billions of such particles, each with slightly different energies that will thus have slightly different oscillations due to variations in magnetic pull. Without any careful monitoring procedures in place, the beam size gets bigger, which lowers the collision rate.

Diode detectors similar to those used in older radio receivers were developed, which convert modulations of beam position into signals in the audio frequency range. As a result of this nanometer-accuracy diode-based oscillation detection system, a feedback system that auto-adjusts the magnetic field strengths was developed.

That said, prior to the development of an automated feedback system, researchers noted in published papers the effectiveness of monitoring the beam by directly listening:

> During Base Band Q (BBQ) operation on the Super Proton Synchrotron (SPS) it was found that a trained person using headphones can distinguish between the sound of betatron and synchrotron frequencies, as well as many important events during an acceleration start, transition, and

> ejection. Once the sound of a normal machine cycle is imprinted on the mind, it is very easy to catch any anomalies which can occur during machine tuning or due to various failures. This is mainly thanks to the logarithmic characteristic of human hearing, which makes it possible to capture small details which are nearly impossible to observe by eye when presented in a graphical way. Listening to the beam can easily be done in parallel to other activities, such as operating a computer and could give new dimension to the work of machine operators! [53]

One continues to wonder whether further work on audification would in this way allow control room operators to better monitor an accelerator experiment much like a car mechanic uses sounds to guide their subsequent repair work.

### 2.3.4 Audio generated from Real-Time Scientific Data Streams

Real-time particle data has been sonified to limited extents. It is common for experiment control rooms to support simple, real-time, auditory alerts, as in the case of an ATLAS control room alarm sounding when a particle beam is discarded, for example. Real-time audio feeds from microphones placed in natural environments are also relatively common [77]. Very limited examples exist incorporating real-time physics data into musical compositions. For instance, the cosmic piano sonifies incoming cosmic rays in real-time for live performance settings [40]. However, only recently have projects worked to continuously convert real-time, large-scale sensor data to sound. For example, the Tidmarsh project allows musicians to generate spatialized sonifications of real-time environmental data to enhance one's sense of presence while moving through a virtual representation of a natural environment [80]. Both Tidmarsh and Quantizer are sonification platforms that seek to involve many composers.

Broadly, it is apparent within the sonification literature that aesthetic considerations have taken on growing importance [59]. The project under discussion fits within the aforementioned, nascent domain of real-time scientific data sonification and is the first known platform to enable sonification of real-time experimental physics data for aesthetic purposes. Also, although as mentioned, many musical pieces have been built atop high energy and astrophysics data, to our knowledge, Quantizer is the first framework that supports multiple compositions to simultaneously map to incoming data.

## 2.4 ATLAS Experiment Overview

The ATLAS detector is one of the two general purpose detectors built along the LHC. The ATLAS collaboration is using the detector to probe some of the most profound questions one can ask about the nature of our universe by recreating extremely high energy conditions that do not occur naturally on Earth. What is the nature of dark matter? Are there extra physical dimensions? What is the origin of mass? Are there any deeper symmetries that govern the laws of our universe? The ATLAS collaboration is enormous; it is made up of over 3000 scientists from 38 countries.

The detector is approximately cylindrical in shape with a length of $\sim$ 46 metres and a diameter of $\sim$ 25 metres. It is made up of several different detector layers including an inner detector surrounded by a solenoid, an electromagnetic (EM) calorimeter, a hadronic calorimeter, and a muon spectrometer [90].

Protons and lead ions are made to collide at the center of the detector, and the energy in the collision generates rare particles with lifetimes that can be on the order of $10^{-22}$ seconds and that rapidly decay to more common and detectable particles like photons and electrons. Each layer of the ATLAS detector serves a purpose for understanding the byproducts of the initial collision. The inner detector measures the trajectories of charged particles generated in the collision. The solenoid is used to bend the trajectories of the charged particles for particle identification and momentum measurement purposes. The calorimeters are used to measure the energies of particles;

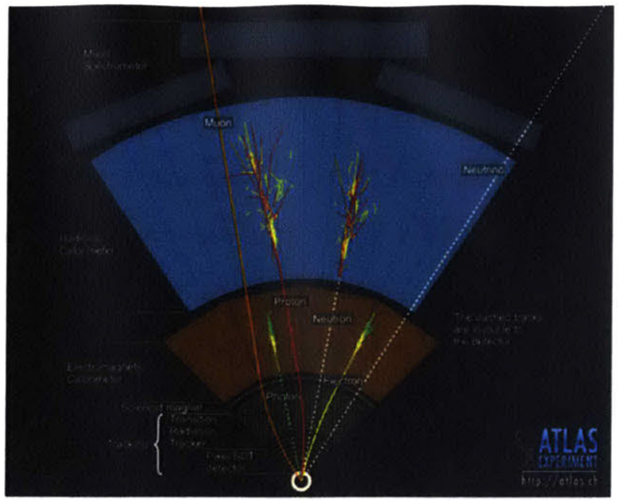



Figure 2-2: Layers of ATLAS detector showing trajectories of different particle types

the EM calorimeter measures the energies of particles that interact predominantly electromagnetically and the hadronic calorimeter measures the energies of particles that interact mostly via the strong interaction. The muon spectrometer measures the trajectories and momenta of muons with the aid of a system of toroidal magnets.

ATLAS physicists work to reconstruct rare particles from their lower energy decay products in order to develop evidence for the existence of new physics phenomena. To do so, physicists apply known theoretical models for particle decay that are commonly illustrated via Feynman diagrams [67] and branching ratios.

The ATLAS detector uses a right-handed coordinate system with its origin at the nominal interaction point in the centre of the detector and the $z$-axis coinciding with the axis of the LHC beam pipe. Cylindrical coordinates $(r, \phi)$ are used in the transverse plane, $\phi$ being the azimuthal angle around the beam pipe [90]. The polar angle, $\theta$, is defined from the +z axis.

## 2.5 Quantizer: Data Structure

During ATLAS data taking, a small subset of particle collision information is routed to a set of machines responsible for generating the real-time audio streams featured on the project's website. This outreach-designated data stream contains roughly 1 collision event every 25 seconds and Quantizer is the first known project external to ATLAS to be granted access to real-time data from the experiment.

The following data streams are extracted by the Quantizer system, chosen for their direct relevance to fundamental physics analyses:

- **Liquid Argon EM Calorimeter** This detector measures the energy deposited by particles that interact primarily electromagnetically, such as electrons and photons. The positions and magnitudes of energy deposits are streamed.
- **Hadronic Endcap Calorimeter** This detector measures the energy deposits of the particles that interact primarily via the strong interaction and are only used in the two ends of the cylindrically shaped ATLAS detector. The positions and magnitudes of the energy deposits are streamed.
- **Particle Tracks** The inner detector is used to reconstruct charged particles as tracks. The track trajectory and the track momentum are streamed.
- **Resistive Plate Chamber** These detectors are part of the muon spectrometer and are only used in the more central region of the cylindrically shaped ATLAS detector. The positions of the detector hits are streamed.

## 2.6 Quantizer: Architectural Overview

Figure 2-3 shows the overall architecture of the Quantizer system.

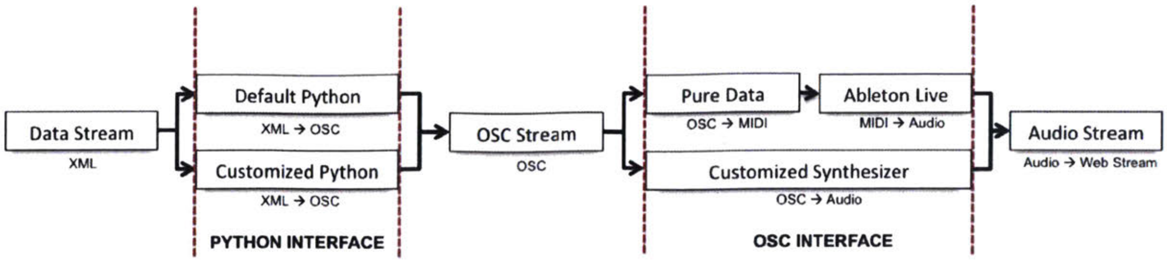



Figure 2-3: Flowchart showing flow of data through the Quantizer platform

Data is routed from the live XML event stream to a set of Python scripts responsible for parsing and processing the data and producing a time-series of messages sent over Open Sound Control (OSC), a common networking protocol used in audio synthesis. These scripts are known as the tool's 'Python interface'. In order to exert control over this data processing step, composers can either set a series of Python flags summarised in table 2.1, or can choose to include their own code in *analysis.py* to guide the set of data cuts applied. The latter approach may be of particular appeal to composers with more physics experience.

Once the OSC stream is generated (whether through the tool's default settings or through customizations), the composer can interact with the resulting OSC messages in what we call for convenience the 'OSC Interface' (see Figure 2-3.) Composers experienced with tools like Max MSP [21] and Pure Data [20] can build custom data-to-audio mappings that interface with this OSC stream. Alternatively, a series of default Pure Data patches have been built in the process of working with a set of early composers. One patch, for example, produces a MIDI stream that triggers Ableton Live [18] synthesizers in order to produce audio. Another patch allows the composer to receive the parsed and processed data stream at once and build custom musical timing in Pure Data. See section 2.8 for screenshots.

Three compositions are streamed to the project's associated website via Broadcast Using This Tool (BUTT) [75] and the Icecast streaming server [19]. A website screenshot is shown in Figure 2-4. Jack Audio [22] is used to route the audio to BUTT,

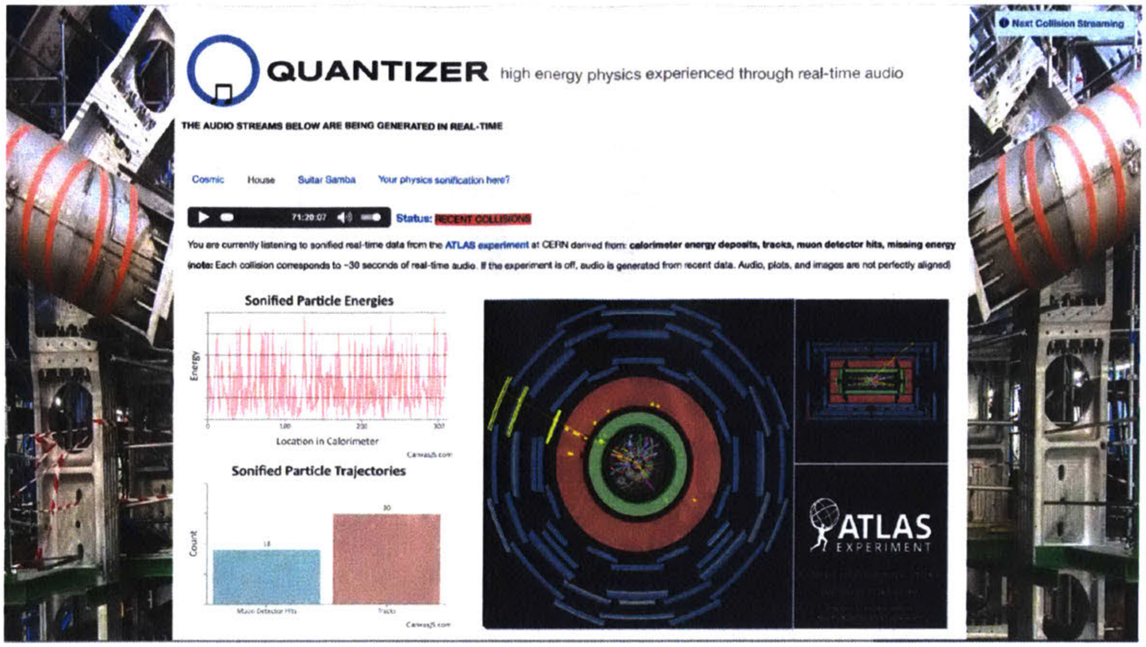



Figure 2-4: Screenshot of the Quantizer webpage. the four tabs contain 3 live audio streams and 1 offline audio-visual interface. The status indicator indicates whether real-time data is available (currently reads 'Recent Collisions' in red). Two plots to the left of the page show basic statistical information for the data produced after all parsing and processing steps have been applied for one of the compositions. The amount of real-time statistical data that can be made public is limited due to protections placed by ATLAS on recently taken data). Finally, the image to the right of the screen shows an event display generated by ATLAS for the collision event currently producing audio. Details on the sonification process are included at the base of the page (not pictured).

and the website itself is built atop Python Flask [83]. In order to update some basic web-based data plots and the web-based ATLAS detector status indicator, Gevent [8] and the WebSocket protocol [16] are used. (The web-based status indicator alerts viewers as to whether real-time data or recent data is currently streamed based on whether collisions are taking place in ATLAS). In this way, the system takes raw XML data and produces multiple web-based, real-time audio streams.

## 2.7 Details on Python Data Processing

As introduced in section 2.6, rule-based selections are applied in order to extract the most compelling data from the collision event file. Some of these data cuts are described below.

Due to the complicated geometry of the detector, ATLAS physicists often place selections on the geometric trajectories of the particles in the event to ensure that they passed through a region of the detector in which measurements are trusted or will produce relevant physics. Similar geometric selections are made by the sonification platform's default Python interface. For example, all the tracks and calorimeter energy deposits are required to have a polar angle direction/position of $0.18094 < \theta < \pi - 0.18094$. (see, for example, [24] for a reference on geometric selections commonly used in ATLAS physics analyses). Since the RPC data stream only contains geometric coordinate information, for events with excessively high RPC counts, a radial geometric cut is applied.

When performing particle searches, ATLAS physicists will often require a particle in an event to have a minimum energy or momentum in order to better isolate signal events from background events. In a similar way, the Python interface selects a subset of the data that meet minimum energy and momentum requirements. For example, each track is required to have a momentum in the transverse direction above 1 GeV/c. For the liquid argon calorimeter, only particles with energy between 0.05 and 0.1 GeV are preserved; for the hadronic endcap calorimeter, only particles with enery between 0.61 GeV and 1.2 GeV are preserved; for particle tracks, tracks with energy between 5 GeV and 40 GeV are preserved. In this way, the composer can work with a known energy range when composing with the system.

Once cuts based on detector geometry, particle energy, and particle transverse mo-

Figure 2-5: Data streaming order when the composer sets the Python 'spatialized' flag to 'true'. First, inner detector data is streamed, followed by calorimeter data. Lastly, RPC data is streamed. Each individual layer is streamed with respect to the layer's theta coordinate. Alternative streaming modes are supported as well.

| Argument | Description |
|---|---|
| –geo | Set scanning geometry used |
| –maxbeats | Set max data points per stream |
| –spb | Set seconds per beat |
| –uniform | Impose a beat by discretizing data |
| –spatialize | Stream with respect to detector layer |
| –layertimeratio | Ratio of relative time spent per layer |
| –overlap | Turn off event queuing |
| –sendall | Bypass timing |
| –preamble | Frontload event-level parameters |

Table 2.1: Sampling of Command line Arguments

mentum have been applied, the remaining data processing is driven by a set of musically-inspired preferences specified by the composer and shown in table 2.1.

Most notably, data is either discretized with respect to the detector geometry and streamed as a time series of OSC messages on a beat structure, else it is streamed purely with respect to a cylindrical geometric coordinate selected by the composer (eta, phi, r). Streaming with respect to the detector's radius is treated as a mode suitable for spatialized audio, since if selected, the data is streamed from the innermost layer to the outermost layer, as depicted in Figure 2-5. Some modes, including '–sendall', '–preamble', 'and '–uniform', were integrated as result of close partnerships with composers who expressed their musical desires throughout the development process and influenced the scope of the tool.

## 2.8 Sample OSC Mapping Interfaces

Musicians with composition experience often opt to design their own creative, experimental projects that interface with the OSC message streams produced by the Python code base. Their ability to successfully do so serves as a measure for the flexibility of the platform. We have worked closely with two composers to develop compositions that showcase the diversity of projects that can be built using data. These two case studies are highlighted in this section. Additionally, two default OSC interfaces built by the author are briefly described. All mapping interfaces are capable of reading in the OSC streams produced by the Python parsing scripts previously described.

### 2.8.1 Cosmic

An audio stream called "Cosmic" was produced as a custom Max/MSP [21] patch by Evan Lynch (see Figure 2-6). The audio is spatialized in order to approximate the sensation of the listener's head positioned at the center of the detector. Mixing parameters for software-defined audio synthesizers are determined by additional event-level parameters that were added to the system (an example is the 'effective sum', a measure of the sum of all track energies.) In order to support this composition style, a number of additional command line tools were also added to the Python interface including streaming event-level parameters a few seconds before the remaining data in order to appropriately tune the synthesizer, 'spatializing' the audio by streaming the data with respect to detector layer (beginning with the inner detector and moving outwards), and controlling the amount of time spent streaming information from each detector layer. The cosmic stream helped us to strategically broaden the default Python feature set available to composers and also serves as a first example of a composer successfully interfacing a custom synthesizer with the OSC streams.

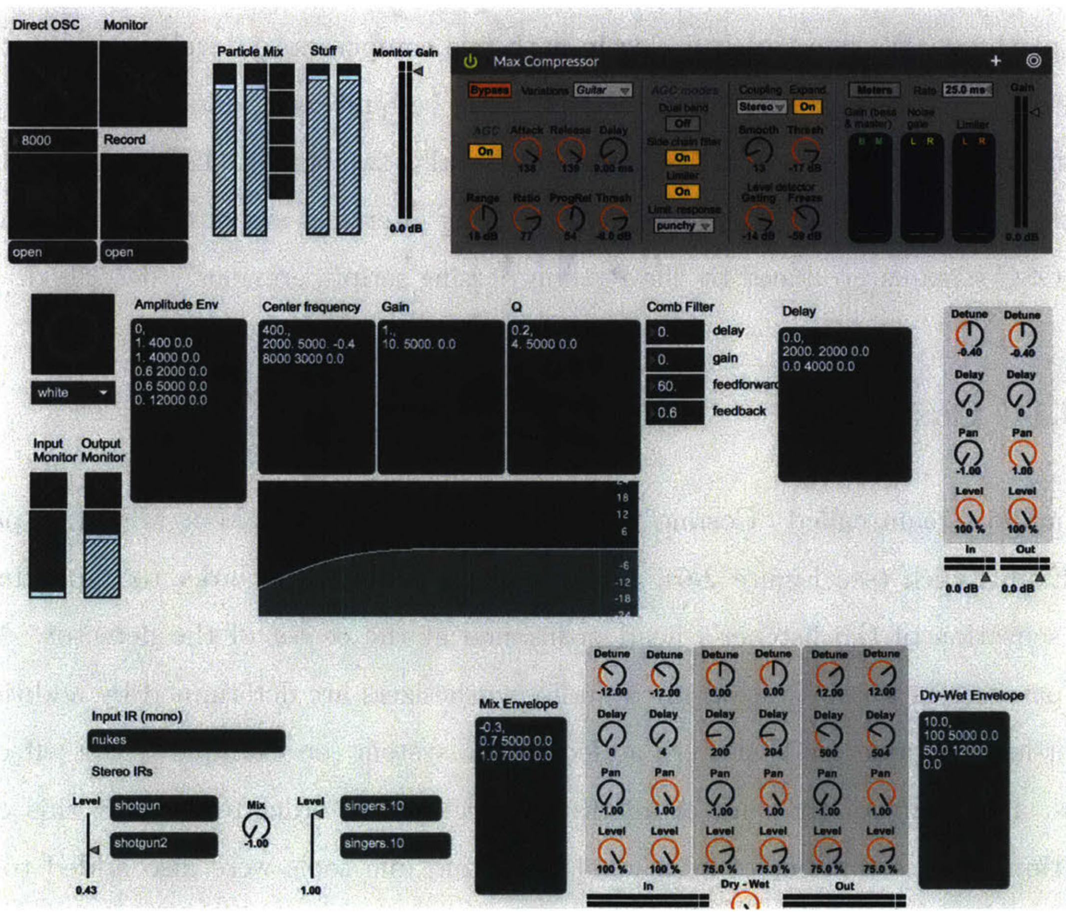



Figure 2-6: 'Cosmic' synthesizer produced in Max MSP by Evan Lynch as a Quantizer OSC interface. 'Cosmic' is one of the 3 real-time streams featured on the Quantizer website

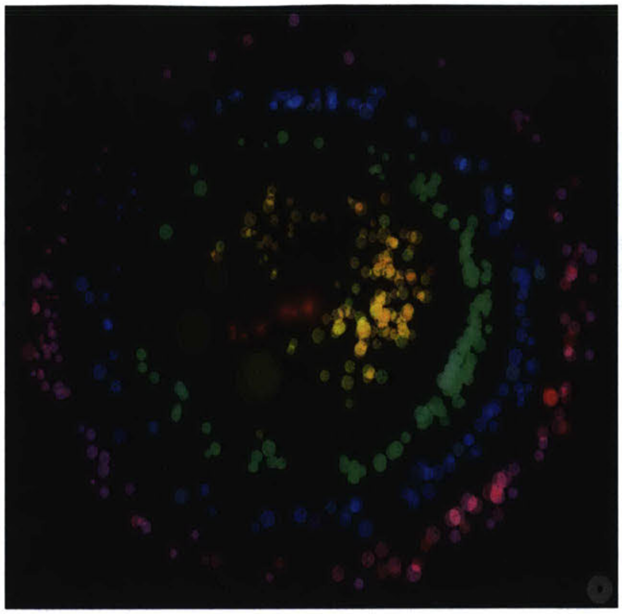


Figure 2-7: Audio-visual OSC interface produced by Akito van Troyer.

### 2.8.2 Audio-Visual

An animated audio-visual experience was produced by Akito van Troyer as an extension of a pre-existing project called Constellation [93]. This project makes further use of the layer-by-layer data streaming built for Cosmic, but in this case the data trigger audio clips in a soundscape of dots (see Figure 2-7). Each dot is associated both with a detector layer and with a particular sound clip. Sound clips are clustered according to their spectral properties, and the user can also explore the soundscape by clicking on dots.

The graphical interface is an artistic 2D interpretation of the detector where each ring of dots represents a different detector layer. The inner detector is represented by yellow dots, the calorimeters by green and blue dots, and the RPCs by pink dots.

When a new collision event file is received, particle tracks are first drawn as lines in the innermost layer. Next, calorimeter energy deposit magnitudes control the diameters of triggered dots. Finally geometric positions of RPC hits control dots fired in the outermost layer.

Additional command-line options were added in order to support this artistic vision, including the restriction of the total number of data points per OSC stream in order to keep the sound produced more manageable, as well as some additional functions to project 3-dimensional detector data onto a 2-dimensional surface.

### 2.8.3 Default Interfaces

Two default interfaces are shown here. The customized timing interface allows composers to bypass the Python data timing algorithms and develop custom triggers to summon the next data point. Banging the 'store_all' method (pictured in Figure 2-8) returns the next data point. For example, one could imagine developing a resonator with a decay parameter that is excited based on the incoming energy deposit value; sufficient decay of the excitation could trigger the subsequent data point, which then triggers the subsequent excitation.

Figure 2-9 Shows a screenshot of an interface that the Quantizer team traditionally refers to as 'the default interface'. It allows composers to adjust tempo, set data/MIDI ranges per stream, control a beat structure using the calorimeter data, etc. This tool was successfully used as a DJ'ing interface when the platform was presented at the Montreux Jazz Festival in July, 2015. An improv pianist played along to real-time audio generated from particle collisions, and a DJ adjusted the data-to-audio mapping throughout the performance [60].

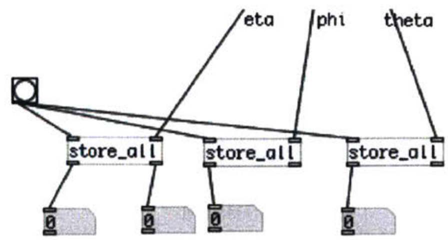



Figure 2-8: Excerpt from the 'Customized Timing' Interface

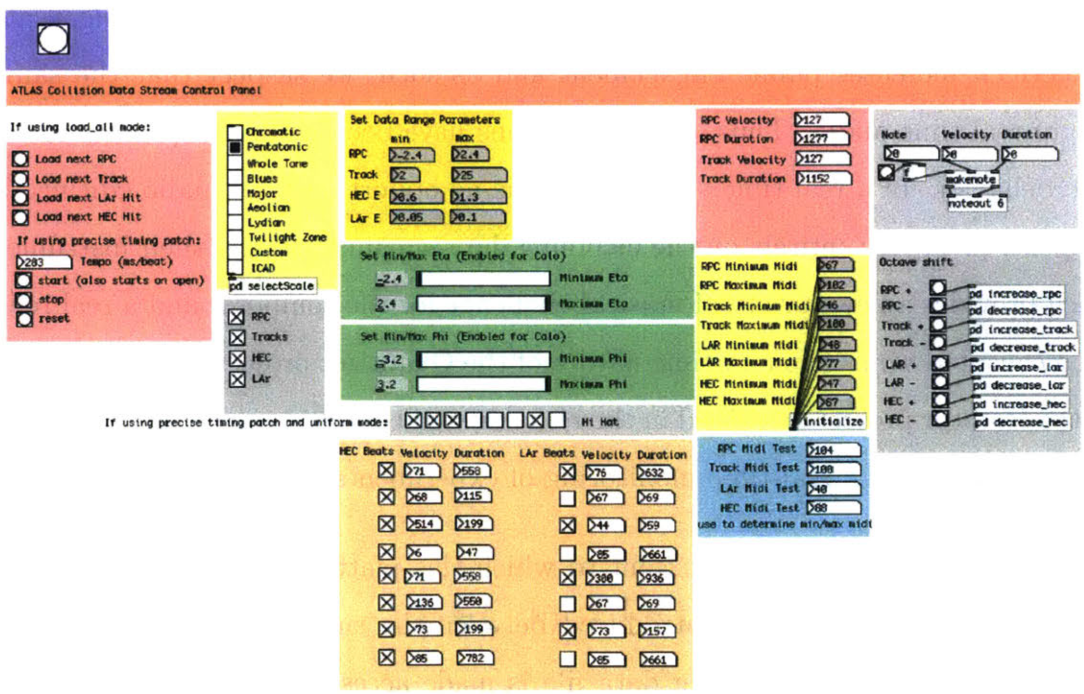



Figure 2-9: Pure Data Default Interface that exposes simple mapping controls to composers with less experience in Pure Data

## 2.9 Evaluation: Real-Time

| Stage | Time Delay |
|---|---|
| Collision→Data File | ~minutes |
| Data Transfer | ~seconds |
| Queuing | ~10's of seconds |
| Sonification | ~seconds |
| Stream to Web | ~10's of seconds |

A key feature of the project is the real-time nature of the data. It is previously established that the real-time nature of ATLAS data visualizations available at atlas-live.cern.ch increases public engagement and as such, we suspect that the same is true of real-time audio. Thus, design decisions have been made to optimise for real-time behavior. In a real-time setting, there is no opportunity for audio editing. The composer must rely entirely on the incoming data stream to trigger their instruments, which provides a new variety of musical challenge for the composer and a real reward when executed well. The real-time nature of the Quantizer project also sets the stage for the Rotator project described in the remainder of this thesis in which one imagined usage scenario is the real-time monitoring of experimental control room data.

Below is a description of the extent to which this platform manages to operate in real-time. The first delay is introduced between the time of particle collision and the time that the corresponding data file is made accessible. Once the data file is available, a small delay is introduced for copying the file to the appropriate machines. Next, the data file is queued for conversion to audio. Currently, the queue has a maximum length of two files, meaning that the worst case queuing time is equal to 2x the event audio duration. Next the input file is read, the data extracted and filtered, and the audio generated. This file access and sonification process is relatively quick. A final audio buffering delay is introduced by the Icecast streaming server as well as browser caching behaviours.

Based on the data in table 2.9, the sum of all delays can be treated in the following ways:

1. Time of collision $\rightarrow$ time of web-stream: minutes
2. Time files are accessible $\rightarrow$ time of web-stream: tens of seconds
3. Time files are accessible $\rightarrow$ audio generated: seconds

The total delay (1) is a bit slow, but this is largely due to the necessary time to perform partial physics reconstructions, and generating and uploading the resulting data. These tasks are performed by ATLAS and are beyond our control. The audio buffering delay (2) of the order of 10's of seconds is common for existing Internet radio stations [23]. Finally, a few-second processing delay (3) is negligible for the current application since new data files are only made available to us roughly every 25 seconds. The queuing delay is easily minimised by tuning the event audio duration to closely match the time between events entering the queue.

## 2.10 Evaluation: Public Outreach

The web-component of the project was released on May 20th, 2016. It is important to reach a large audience in order to meet the outreach goals of the project. To help promote the project to its intended audience, articles were published by MIT [48], CERN [79], Nature Physics Books/Arts [74], Popular Science [73], Smithsonian Mag [72], Engadget [49], and Gizmag [50]. As of June 16th, 2016, the site has received 28,693 page views. We have also received ~40 requests via email to either use the project to compose audio, use pre-existing sound samples, or use the data stream. These messages are summarized around common themes below, and some email excerpts are included.

- **Theme: enthusiasm for the scientific outreach goals of the project** e.g. "Thanks for doing the things that keep science refreshing and fun"

  This genre of feedback suggests that we are meeting our outreach goal of promoting interest in physics among a public audience. Outreach is increasingly important for large-scale experiments heavily dependent on external funding.

- **Theme: Wishing to extend concept to other scientific projects** "I'm working on the the Belle2 experiment and I'm an electronic music producer - I love your idea. Is there a way I could fork your project? I would love to do something similar for Belle2"

  This genre of feedback suggests that the project inspires further work in the space of creating real-time radio channels for the public to listen to experiments.

- **Theme: wishing to either use prerecorded clips or generate custom compositions for other musical projects** e.g. "Definitely interested in producing music from real time data from ATLAS. A friend and myself are an electronic improvisational ensemble and we are always seeking new ways to introduce the element of chance into our live performance"

  This genre of feedback suggests that the prerecorded audio clips we have produced are useful for integration in other sound-based works.

- **Theme: Questions regarding how structure is obtained and proposals for alternative sonification approaches e.g.** "I like this idea but you just get neat musical stuff pretty much no matter what signal you throw into this, right? Stuff with structure sounds neater then just noise I suppose, but maybe you could set one transient to each type of particle? instead of modulating pitch/frequency only modulate the amplitude, or only modulate the pitch in octaves. I think an overall sound effect more like an ordered Geiger counter then the (albeit pretty) stuff would be good; if you want to hear the structure of the data."

This genre of feedback suggests that we could be placing more thought into the sorts of data-to-audio mappings that each interface promotes. In particular, though the principle goal of this project was to produce aesthetically satisfying compositions, listeners nevertheless crave an ability to better interpret the structure of the collision event from the audio stream. It is a principle challenge of this project to on the one hand provide composers with enough flexibility to express their artistic visions, while on the other hand guide the composer towards compelling mappings. Our ability to do so is somewhat limited by the real-time dataset in question: timing information for different elements of the collision is noticeably lacking from the data stream. That said, some possibilities already exist for composers to elucidate structure in the data. As an example, they may discover that the number of charged particles decreases with increasing momentum, or that there are clusters of energy deposits in individual collision events. These patterns, if highlighted effectively by the composer in the audio, can also teach listeners about the physics of the collisions. As future work, we ought to further develop the tool to help more directly motivate compositions that give listeners a sense of the structure in the data, elucidated by psycho-acoustic properties of the audio like perceptual clustering and stream segregation. A direct sonification of the data streams might similarly prove compelling to listeners.

## 2.11 Summary

The Quantizer platform has enabled the development of three real-time musical compositions driven by real-time data from the ATLAS detector, as well as one offline audio-visual interface and one live performance. The web-component of the project has extended the reach of the platform to tens of thousands of viewers, and we anticipate extending the website to include works by additional composers in the future. Reflecting back the divergent feedback written about Ikeda's project in section 2.3.1, one is motivated to consider what purpose a tool like this serves: it appeals to the listener who seeks to more directly experience femtometer-scale physics phenomena taking place within kilometer-scale experiments; it promotes a new genre of playful musical composition in which the musician is now seen as an instrument designer and no longer as an instrument player, or more specifically, as a composer who algorithmically specifies the overall contour of their composition, which is then fleshed out and explored by real-time data.

# Chapter 3

# Rotator: Overview

## 3.1 Background and Contextualization

While walking down the streets of Boston, we are unfailingly submerged in a three-dimensional, 360° sonic cacophony of city life that is commonly referenced in academic literature as an *auditory scene.* Unlike audio, however, visual input is directional, providing a field-of-view of only 130-135° vertically and 200° horizontally with respect to the center of one's gaze. When a car screetches, we impulsively jolt our heads to bring the source of the noise into our field-of-view. In this way, we are accustomed to using visual and sonic information in tandem, placing our eyes where they are most beneficial and tuning our attention to precise features embedded within the sound. Can we build scientific research and monitoring tools that similarly harness both modes of information transmission, enabling us to maximally benefit from our sonic and visual capabilities?

Recalling once again certain well-known auditory scientific tools like stethoscopes, Geiger counters, and experimental control room alarms, it is clear that the emerging field of sonification is ripe with opportunity to study practical uses of audio in scientific research. That said, the field of sonification is often met with skepticism by scientists heavily accustomed to visual information display techniques. Suppose that, rather than make a case for the inherent superiority or inferiority of audio as an information display mode, we instead strive to create a more natural and customizable harmony between our eyes and our ears when analyzing or monitoring data.

Sonification researchers often take note of shrinking device screen sizes, provocatively using this trend as evidence that we will need to find auditory alternatives for certain visualizations. I wish to further extend this observation by contextualizing data sonification within the framework of *Responsive Web Design*, an umbrella term referring to the construction of a webpage that can smartly customize its visual rendering on the basis of a device's screen real-estate (see Figure 3-1). An audio-visual data analysis tool like Rotator is suggestive of the role that sonification may grow to play as a natural extension to the Responsive Web Design Framework. At one extreme, visual data floods across enormous monitors. At the other extreme, there is no screen, and all content is relegated to an auditory periphery. An array of audio-visual intermediary states lay between these two extremes.

Suppose, for illustrative purposes, that a researcher is in a control room monitoring the stability of a sensor network deployed in a city. In the control room, the researcher users a set of large monitors to watch the data. A sensor node deployed in a park begins to exhibit some suspicious behavior, so the researcher opts to leave the control room to investigate the physical device. While en route, he takes out his phone to continue monitoring the network. Due to the phone's more limited screen real-estate, a large subset of the sensor data is automatically shifted to his auditory periphery, and the visual data is resized and rearranged. Furthermore, much like within the aforementioned physical scenario in which a screeching car causes a person's head to turn, so too might a researcher swipe a

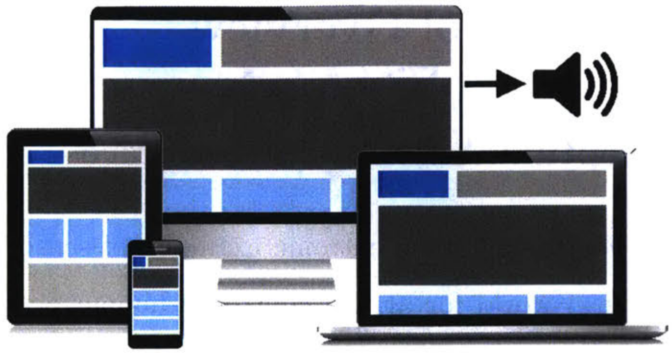


Figure 3-1: An illustration of the prominent concept of Responsive Web Design: the design of a webpage to intelligently scale to an arbitrary screen size. An interface like Rotator helps to frame sonification as a natural extension of this model

peculiar-sounding audio stream into his visual field for multimodal inspection.

Before turning to application areas within scientific research, we briefly ask: how might such a capacity for perceptual manipulation manifest in the physical world if sensory modes for data presentation could be more precisely selected? Perhaps you wish to place your visual gaze on a tractor so as to avoid intercepting its path but you wish to mute its fierce and ear-splitting sound. Perhaps you wish to increase the volume of the pleasant bird chirps up above but have no need or desire to actually see the birds. Suppose, in order to hone in on a task, you wish to render your auditory input directional as well, accepting sound only from source within your visual field of view. Here the auditory scene has been treated as a series of discrete sound sources whose presentation modes can be edited and adjusted at will on the basis of your personal preferences or on the basis of your information processing capabilities.

## 3.2 Rotator Tool Overview

### 3.2.1 Perceptual Adjustment Interface

While we have considered some extreme examples of sensory mode manipulation in the physical world for at-will control of human perceptual experience, for the purpose of this thesis we consider how the same principles in sensory control ought to influence scientific data analysis and monitoring, in particular for datasets with a large slew of parameters requiring our consideration. The Rotator project is chiefly concerned with how our perception of the structure of a dataset can be informed and influenced by the presentation modes of its constituent data streams.

Consider a simple analysis task consisting of $n$ data streams. How can these $n$ streams be perceived? If each stream has 3 possible presentation modes: auditory ('A'), visual ('V'), and both auditory and visual ('AV'), then the total number of presentation modes is $3^n$. Furthermore, there are an enormous number of approaches one can take for representing a data stream visually or sonically. Though the total number of perceptual modes is thus quick to skyrocket even for small numbers of data streams, for most datasets, certain presentation options are more immediately worth considering. As one example, all data can be presented sonically, and tiny subsets of data can simultaneously be scanned visually on an as-needed basis. Alternatively, $k$ data streams can be presented sonically and the remaining $n - k$ data streams can be presented visually. Are there particular presentation states that can increase our *attentive capacities*, namely, the number of streams that we can accurately perceive at once in an analysis task? Can the movement of some data streams into our sonic periphery free up our eyes to complete an alternative task? Far beyond the scope of this thesis, are there methods for predictively modeling how to best

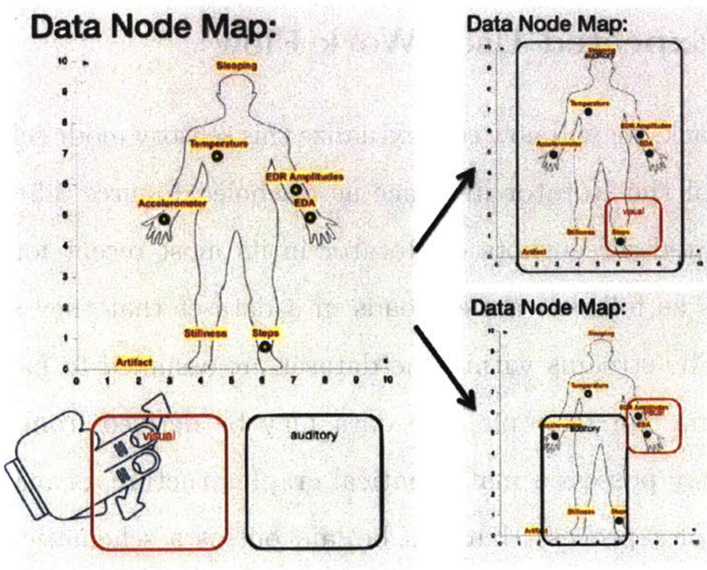



Figure 3-2: Excerpt from the Rotator interface applied in this case to a wearable sensing system: users resize and slide auditory and visual boxes around a graph of data stream nodes in order to dictate in which sensory mode each nodes will be presented. For example, in the upper right corner, the auditory window has been stretched to include all the data nodes, and the visual window has been shrunk to contain just a single node

present a dataset to a user on the basis of how it is changing over time, or is consistency of presentation mode most critical?

The Rotator tool provides users with control over how each stream in a dataset is presented through an interface consisting of auditory and visual sliding windows as pictured in Figure 3-2. The audio is spatialized with respect to the center of the sliding audio window, which creates a structured auditory scene around the user. The audio spatialization also aids the user in perceiving the data streams as segregated sound sources [82]. The concepts of auditory scene analysis and auditory stream segregation are addressed as part of the Related Work summary in chapter 4. Detail on data sonification approaches in literature are also discussed in chapter 4.

### 3.2.2 Expected User Work Flow

Stepping back, we will now contextualize this sensory mode selection map within the scope of the Rotator interface as a whole. Figures 3-5 and 3-6 show raw and annotated screenshots of Rotator in its most recent form. The expected workflow is as follows: a user loads in a dataset that they seek to analyze or monitor. The streams within the dataset are assumed to have a geometric interpretation. For example, the data may be derived from a physical sensor network, may possess a mathematical graph structure, or may represent different stages of a process that can be laid out as a schematic. The user begins to explore the dataset by moving and resizing the auditory and visual windows around the node network. For instance, the user may find that listening to a spatialized sonification of all of the data nodes while only looking at a visual representations of a few nodes at a time is most comfortable. Besides sensory mode adjustments, the user can explore the data by changing the window size currently visualized or sonified, making adjustments to the data-to-audio mapping settings in the audio control panel, choosing to view the Fourier transform of any individual data stream, enabling a 'play' mode that scans the data and simulates a real-time monitoring scenario, or cluster the data nodes, among other controls. When both audio and visual modes are enabled, a vertical red line slides across the visual plots to provide an approximate indication to users of the data region currently being sonified.

### 3.2.3 Clustering Tool

A user may wish to reduce the total number of streams sonified in a given moment, since current literature suggests we will begin to experience cognitive overloading with as few as 4 streams. To do so, a node clustering mode has been implemented. A user can click on a series of nodes which will visually expand

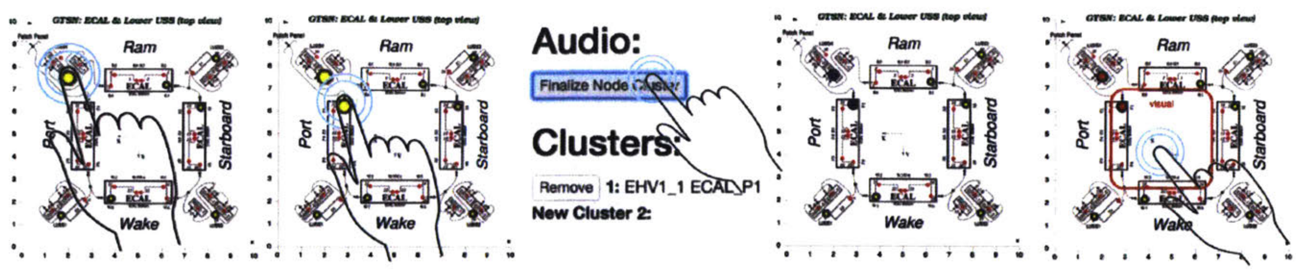



Figure 3-3: Clustering workflow, pictured for the Alpha Magnetic Spectrometer application area (see chapter 6.3). A user selects two data nodes and generates a cluster. The nodes are now treated as a single entity such that if any node in the cluster is present in the visual or auditory windows, then all the nodes will be highlighted

them in size. Then, a user can click 'finalize node cluster' in the cluster control, which will heretofore group the selected nodes and treat them as a single data stream. A clustering workflow is pictured in Figure 3-3.

The average value between the selected streams will be used both for visual and sonic display. After consideration, averaging was chosen over display of correlation functions in order to preserve the user's confidence that the x-axis represents time or frequency. In order to nevertheless understand correlations within a cluster, the Pearson product-moment correlation coefficients between each pair of nodes in the cluster will be displayed, giving the user a sense of the structure of their cluster. The Pearson correlation coefficient is a standard statistical measure of the linear correlation between two variables X and Y, giving a value between +1 and 1 inclusive, where 1 is total positive correlation, 0 is no correlation, and 1 is total negative correlation. It is defined as

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{3.1}$$

An example of the visual representation of two temperature data streams that have been clustered together is pictured in Figure 3-4. As a sample use case for this clustering feature, consider the Alpha Magnetic Spectrometer (AMS)

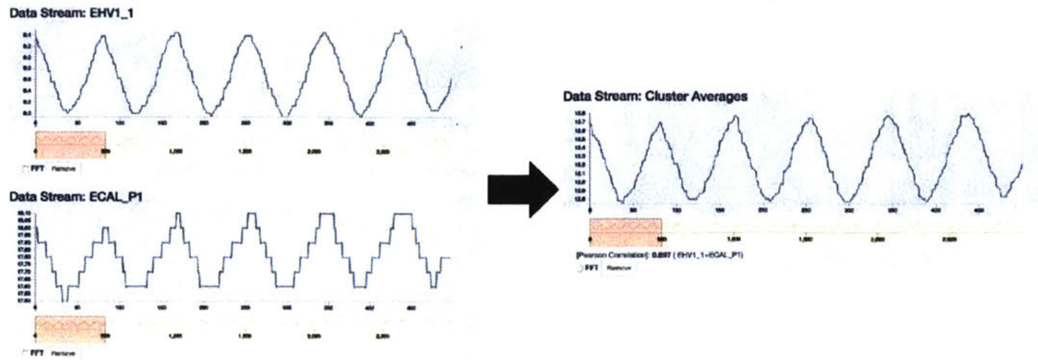



Figure 3-4: An arbitrary number of data streams can be clustered. In this image, two temperature data streams (left) are clustered into an average stream (right) with Pearson correlation coefficients listed.

temperature monitoring scenario that will be described in section 6.3. AMS is a high energy particle detection experiment aboard the International Space Station. It contains a high number of temperature nodes that are carefully monitored, and there may be regions of AMS that are close in proximity and therefore follow very similar temperature variations. On the one hand, this structure to the data may emerge on its own simply by listening to the high number of streams at once. On the other hand, the listener may find that monitoring tasks are more easily accomplished by clustering together regions of the detector for the purpose of reducing the total number of auditory or visual sources. If the Pearson correlation coefficients between cluster nodes were to change dramatically, the user can choose to delete the cluster and reexamine each node independently from one another.

### 3.2.4 Sonification Options

The premise of sonification is to develop data-to-audio mappings that make use of the high dimensionality of an audio signal. Audio synthesis approaches developed for particular Rotator application areas are described and justified in chapter 6, and sonification theory is developed in chapter 4. General sonification controls available in the interface include tempo adjustment, per-stream audio gain adjustment, a toggle for turning on and off audio spatialization, modulation depth and fundamental frequency controls for oscillator-based synthesizers, and the ability to listen to a direct audification of each stream (note: synchronization with graphics still not available for this mode). Here, a 'synthesizer' refers to a mechanism that can take data as input and produce audio as output.

## 3.3 Possible Interface Extensions

Currently, the synthesizer applied to each data stream is hard-coded in the dataset, and it is easy enough to modify these hard-coded synth selections. However, one can immediately imagine an interface extension that allows the user to define the data-to-audio mappings directly within the interface. It is important to be mindful of the number of controls exposed to the user due to an effect known in software development as 'feature creep', in which additional features bloat rather than simplify an interface. Therefore, interface enhancements focused on automating controls are more immediately desirable than features that provide additional user controls.

That said, another natural extension to the interface would be an ability to draw free-form auditory and visual regions atop the node map rather than remain restricted to rectangular regions, though for this our spatialization model (in which audio is spatialized with respect to the center of the rectangle) would need to be reconsidered.

## 3.4 Summary of Goals

We have taken some time in this chapter to hint at large-scale contextualization of an audio-visual tool within the fields of sonification and multimodal perception, as well as to summarize the layout and available functionality of the Rotator tool in its current from. Before proceeding, it is important to bear in mind the scoped goals for the current iteration of this project, which are summarized as follows:

- Build a readily installable tool that supports flexible, per-stream audio and visual display modes
- Customize the tool for three specific, illustrative application areas
- Conduct a preliminary study within one of the application areas regarding the impact of data presentation mode on the user's qualitative assessment of data structure as well as on cognitive load

The first goal, namely, the construction of the basic tool, is described in Rotator Architecture (chapter 5). The second goal, customization of the tool for specific application areas, is summarized in Prospective Application Areas (chapter 6). Finally, a preliminary investigation on the influence of display mode on one's perceptual experience and on one's experience of cognitive load is described in Evaluation (chapter 7).

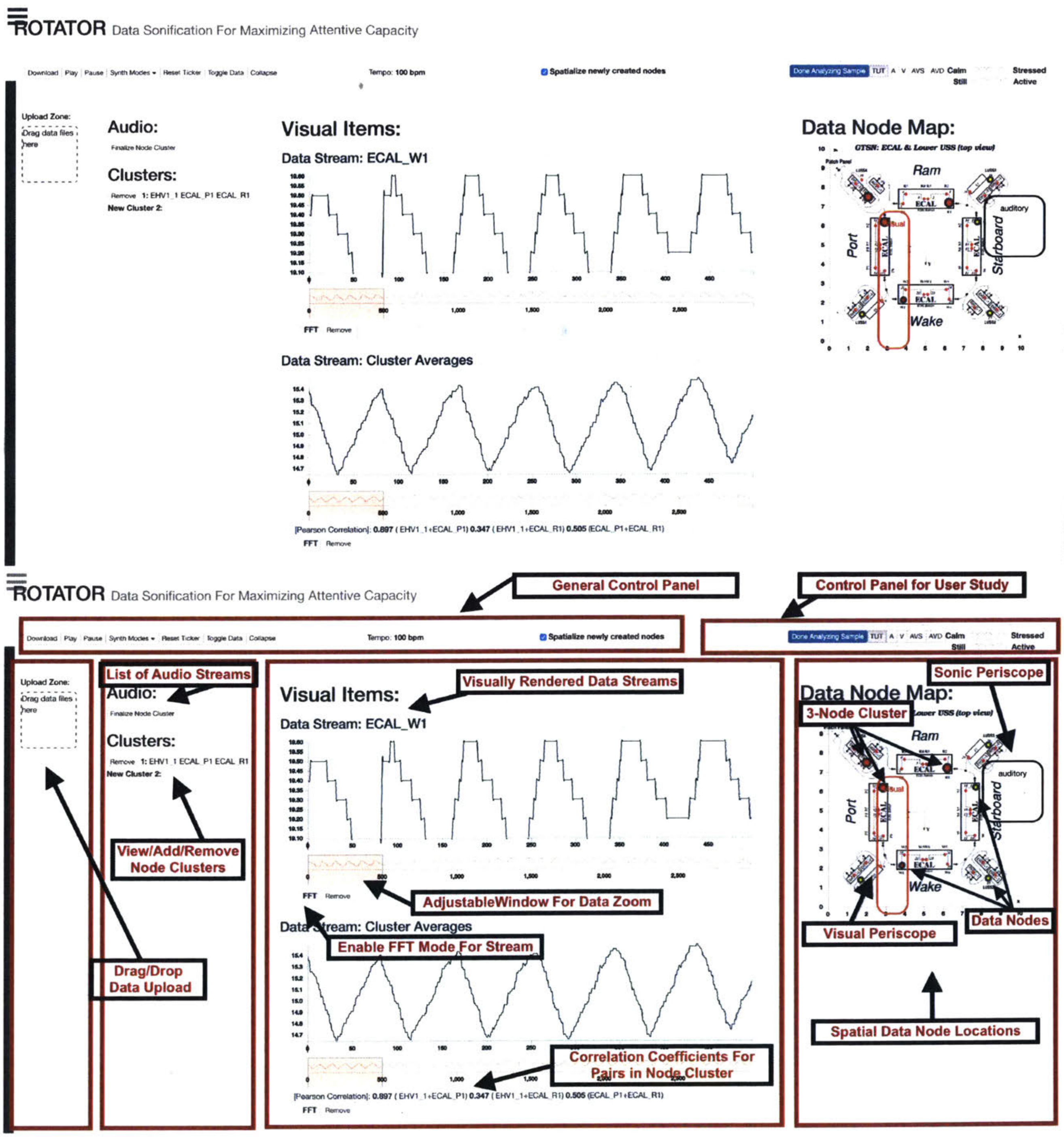



Figure 3-5: Rotator Interface (pictured for AMS application), with and without annotation

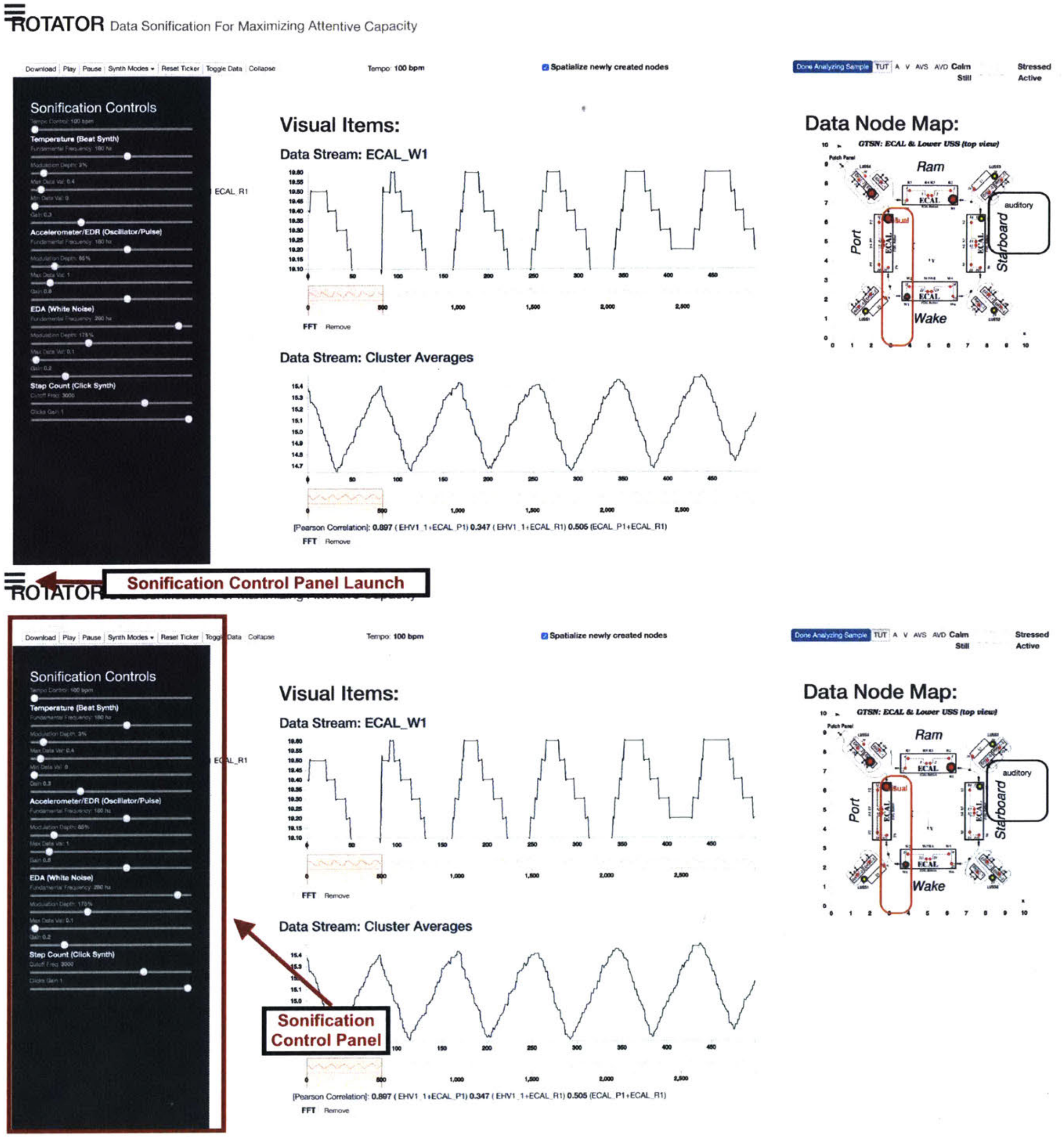



Figure 3-6: Rotator Synth Control Panel, with and without annotation. Note that as pictured, synth controls are currently labeled for the wearable biosensor application, whereas graphics are for the AMS application. In practice, one would align synth labels appropriately

# Chapter 4

# Prior Art

*"When we enter a room, we seem to see it all at once; we are not permitted this illusion when listening to a symphony. "Of course," one might declare, for hearing has to thread a serial path through time, while sight embraces a space all at once. Actually, it takes time to see new scenes, though we are not usually aware of this. That totally compelling sense that we are conscious of seeing everything in the room instantly and immediately is certainly the strangest of our "optical" illusions. Music, too, immerses us in seemingly stable worlds! How can this be, when there is so little of it present at each moment?"*
— Marvin Minsky, *Music, Mind, and Meaning*

We have long puzzled over the interplay between our visual and auditory senses. As Minsky eloquently states, perception of a visual scene does not in fact occur instantaneously. Instead, human eyes rapidly dart around a scene in an effect known as 'saccades' in order to enable the brain to build up a corresponding mental map. Why can audio feel stable to the listener despite its temporal nature? Why can a series of discrete audio streams blend together into a cohesive whole? This chapter addresses theory and prior work in the study of auditory perception and attentive capacity.

## 4.1 Theory: Auditory and Visual Interplay

Studies have repeatedly shown that humans can only monitor four to five items simultaneously before experiencing visual overload [92][43]. Generally, research addressing this perceptual limitation has focused on methods for pre-processing, filtering, and auto-labeling data in order to render only the most critical information to the user. However, it is also important to recognize that the visual system and the auditory system can serve complimentary roles in data display. While visual displays work well for providing detailed renderings of local areas of a dataset, aural scanning may be better suited for finding regions of interest with short temporal durations in a large dataset. In audio, Cariani et al. write that the human benefits from expectancy violation effects in the auditory cortex: 'On all timescales, repeating patterns of sound and their evoked auditory events build up strong representational expectancies of their continuation. This effect is created even with arbitrary and highly artificial repeating sound patterns' [38]. Changes in pitch, intensity, and duration create distinct neural signatures, like mismatch negativity (MMN), that are similar to neural effects seen in linguistic and musical violations and suggest the existence of a broad neural mechanism for the buildup of temporal pattern expectancies. MMN is an effect through which a fronto-central negative potential arises in the brain with a latency of 150-250 ms and can be triggered even if the subject is not consciously paying attention to the auditory stimulus [38].

Expectancy-violating effects can also be used in visualizations to highlight content in one's visual periphery. The Perifoveal display built at the Media Lab (figure 4-1) renders data differently based on whether it is in the forefront of the users vision or on the outskirts of their gaze. Any noteworthy peripheral data can summon attention through changes in brightness, detail, and size [61].

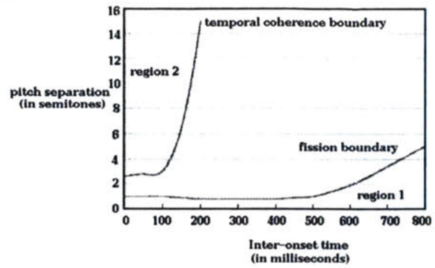
While the Perifoveal display attempts to make more explicit use of the visual

Figure 4-1: The Perifoveal display encodes information into the user's visual periphery that can summon an onlooker's attention

periphery, it can be argued that audio is uniquely well-suited to monitoring data in one's attentive periphery since perception of audio emanating from a source does not required directional fix. One immediately recalls the famous 'cocktail party effect' through which it is possible to deliberately fix one's attention on a particular sound source even when, for example, many conversations are taking place at once. Relatedly, expectancy-violating stimuli such as hearing one's name voiced can involuntarily summon attention. An attention-grabbing auditory effect has been described in the literature as an 'auditory event', which for non-speech audio is defined as 'a mental construct typically produced by temporal acoustic contrasts that distinguish subsequent from preceding sound patterns' [38].

## 4.2 Auditory Scene Analysis

Auditory Scene Analysis (ASA) describes a set of heuristics that model the organization of incoming auditory data into a scene around the listener. Within an auditory scene of discrete and sufficiently segregated sound sources, the listener's attention can shift from source to source [32]. Pioneering work on audi-



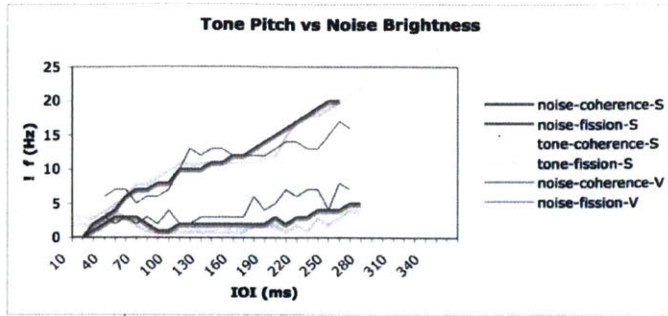




Figure 4-2: Both images drawn from work by [32]. Left: van Noorden Diagram phases for pitch. Tones falling in region 2 are perceived as segregated, tones falling in region 1 are perceived as unified, and tones within the fission and temporal coherence boundaries are ambiguous to the listener and dependent on attention. Right: Comparison of von Noorden diagrams for pitch and noise brightness

tory scene analysis was conducted by Bregman in 1994 [36]. Cariani's biological framing of 'expectancy violation' in the previous section is qualitatively similar to Bregman's 'old-plus-new' heuristic': when new components are abruptly added to a spectrum of partials, the ASA system is skilled at deducing which partials are a continuation of the previous signal and which are newly added. The newly added partials are perceived as a separate sound. Both authors conclude that the onset of sound is a crucial moment in auditory scene analysis [36].

Efforts have been made to model perceptual segregation of streams in terms of common auditory parameters. For example, the van Noorden Diagram maps the perception of two tones with respect to their pitch difference and intertone onset interval into regions where one or two streams are perceived, as well as an ambiguous region in which perception is dependent on attention [26]. While the van Noorden diagram is typically used to map stream segregation with respect to pitch differences, Barass et al. have made an effort to broaden the palette of such diagrams to other audio properties including brightness of a noise grain, and then to amplitude and inter-level difference panning of the noise grain [32]. Their experiment was conducted by asking subjects to select an intertone onset interval and then to adjust a variable parameter until segregation between two streams can no longer be discerned. This data is collected over many trials. The resulting van Noorden diagram for brightness of noise grain is very similar to the original pitch diagram (see figure 4-2), suggesting that similar perceptual processes may be involved. Some differences exist in the resulting van Noorden plot when loudness is varied, and stereo panning demonstrated that sources greater than 60 degrees apart more readily segregate into separate streams.

Von Noorden diagrams can be useful in guiding sonification mapping algorithms away from regions of perceptual ambiguity. In addition, perceptual equivalence classes that have been identified can be useful in motivating mapping choices. For example, it is difficult to distinguish low-frequency harmonics even if they have different phase spectra. It is also difficult to distinguish harmonic, low frequency sounds with the same fundamental frequency since they usually produce the same pitch despite differences in spectral content [38].

## 4.3 Spatialization for Enhanced Segregation

The auditory cortex creates an auditory scene based on the rich content of incoming audio into each ear. When two sound sources have different internal representations in the auditory cortex, they will also have separate sets of perceptual attributes. When the sounds are fused together by the auditory cortex into a single representational object, their respective attributes will blend together as well. Similar harmonic structure and similar onset time are two of the strongest factors that cause the fusion of incoming audio to objects and events. Together, these generate a unique timbre for the event [38].

When two frequency components lack common harmonic structure and onset time, the listener may hear the distinct properties of individual instruments such as pitch, timbre, loudness, duration, and location [38]. However, interference and masking between multiple audio streams can impede the listener from fully segregating the streams. An example provided by [35][34] is binaural interference, a phenomenon in which the presence of a low-frequency noise interferes with perceived interaural time differences for a simultaneously presented high frequency narrow-band noise.

Spatialization can be used to minimize interference and masking effects. The use of spatialization to regulate stream segregation has a number of specific advantages. Firstly, it is an independent parameter in the sense that the spatialization of a particular stream can be modified without risk of the change interfering with other parameters. (Conversely, if pitch and loudness are modified independently, interaction between these two dimensions may cause ambiguity in the resulting stream) [32]. Secondly, it provides the user with a physical map for the sound sources which can aid in the sonic learning process.

The capacity for spatialized audio to aid in the perception of segregated streams has been evaluated only a handful of times in literature. In particular, a study of discrete signal identification in audio showed that when audio streams were separated at greater than 60 degree angles, participants were able to identify significantly more independent sources than when no spatialization was imposed [86]. While there is an intuitive argument for the potential for spatialization to aid in stream segregation, there are also some risks to consider. For example, in visual search experiments, the introduction of spatialized search items has been shown to increase response time but decreases accuracy in users. Furthermore, when two streams are spatially segregated, the division of attention required to monitor each stream is at risk of increasing cognitive load (due to the increased demands on ones spatial reasoning). A further limitation is that our capacity for temporal reasoning between spatially segregated streams has been demonstrated to be inferior as compared to cases where multiple data parameters are encoded into a single auditory stream. Therefore, there are a considerable number of

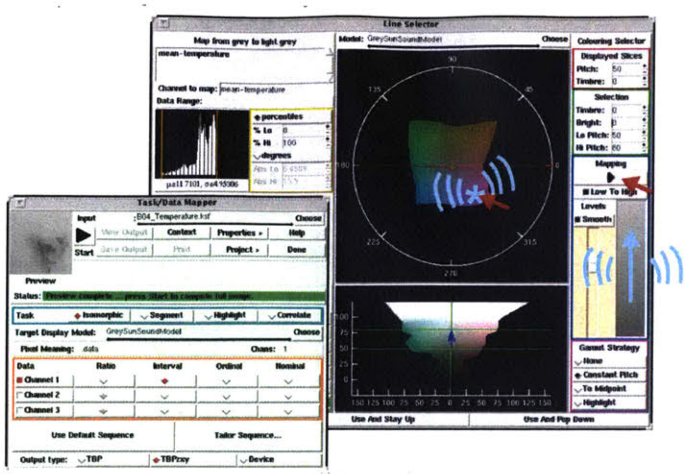


Figure 4-3: Screenshot from the Personify Sonification Platform built at CSIRO [31]

drawbacks in using spatialized audio [86].

A study on spatialized audio identification that is most closely related to the current discussion compared user identification of two pitch segregated signals that are spatially segregated from two pitch segregated signals emanating from a single source [86]. Researchers found improved signal classification accuracy when audio streams are spatially segregated. Trials were also performed where the timbral properties of the set of two pitch contours were modified, (though within each trial the two streams had the same timbral properties). Fourteen timbres were tested altogether by altering modulation index, the ratio of frequency of the carrier to the frequency of the modulator and oscillator envelope. Certain timbral properties were perceived more clearly than others when audio was spatialized, and the timbres that resulted in best and worst results were analyzed further. When spectral centroids were extracted, it appeared that longer attack periods correlated with improved perception of the signals.

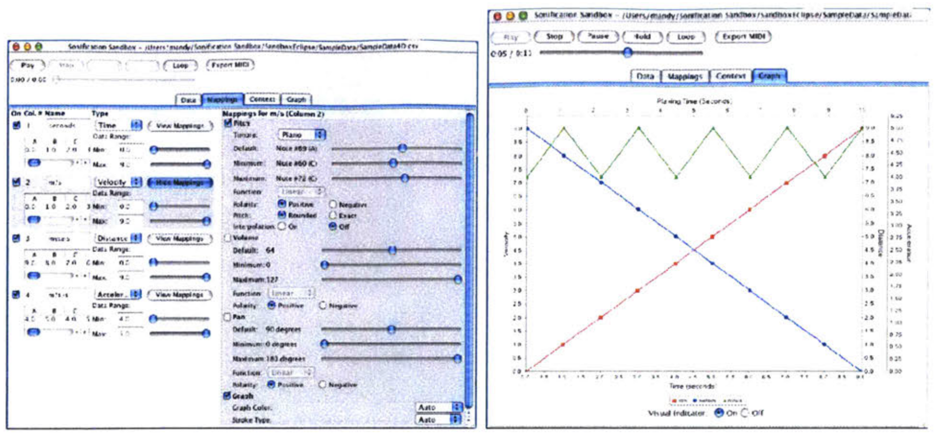


Figure 4-4: Screenshots from Sandbox Sonification Platform built at Georgia Institute Of Technology [95]

## 4.4 Data Sonification Platforms

Up until now our discussion has focused predominantly on prior work in the study of auditory perception. Rotator, in more practical terms, is intended as a sonification platform.

A number of data sonification platforms have been developed in the past. Four examples are the Sonification Sandbox (a Java application developed by the Georgia Institute of Technology, figure 4-4) [95], SonArt (a platform developed at Stanford University, figure 4-5 [33], Personify (a scientific data sonification platform built at CSIRO, figure 4-3 [31], and MUSE (A musically-driven data sonification platform created by UC Santa Cruz) [78]. Each tool makes use of visual information to aid in the user experience. For example, Personify asks users to customize a visual representation of a data-to-audio mapping space where axes correspond to musical properties, and the Sonification Sandbox allows users to view line plots of data streams as they are being sonified. However, unlike the sonification platforms mentioned above, the Rotator platform is specifically aimed at diversifying the way that users *distribute* data across their senses, which we have yet to see as the focus area of a sonification tool.

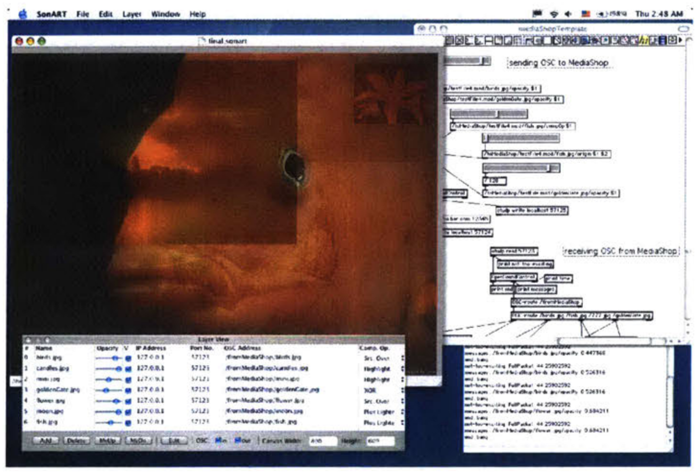


Figure 4-5: Screenshot from Sonart, a sonification platform built at Stanford [33]

# Chapter 5

# Architecture

## 5.1 Architectural Overview

Rotator is a client-side application built in Javascript and architected in React [12] and Flux [7]. The most heavily used low-level APIs are D3 and Web Audio. The decision to build Rotator as a web-based application was reached for a few principle reasons. Firstly, real-time, web-based audio technology has improved tremendously over the last decade and is rapidly becoming a standard. Secondly, previous experience integrating together desktop synthesis tools like Pure Data, Max MSP, and Ableton Live suggest that the installation process for a new user is arduous; a web-based application requires only a browser. Finally, extending a niche visualization tool like ROOT[1] was entertained as a possibility and may still be considered in the future. Web Audio does have some drawbacks, however. First of all, it is still under active development and therefore lacks certain basic audio streaming features that are commonplace in desktop audio processing software. Secondly, it is designed for browser integration, and therefore more

[1]ROOT is a C++ framework used for visualization in particle physics [14].

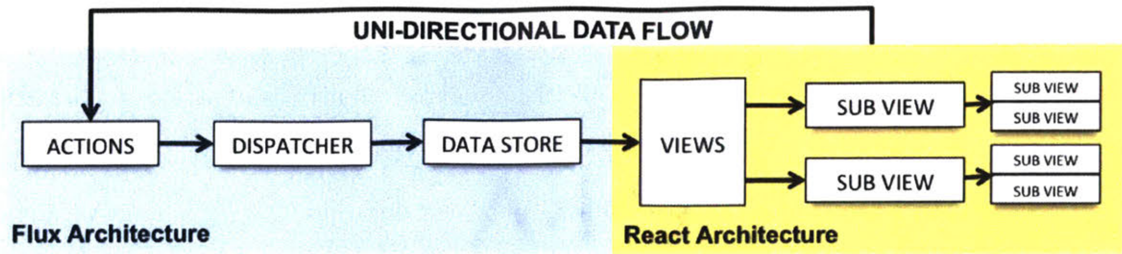



Figure 5-1: Generalized Flux/React Architectural Paradigm (Note that the above diagram is not specific to the Rotator application)

work is required for integration into apps.

### 5.1.1 React

React is a Javascript framework released in 2013 to aid developers in create large, data-driven client-side applications. A React user interface is built up from a hierarchy of modular, reusable view components. Each component contains both view logic as well as a render function that generates markup code, a break from traditional web design paradigms in which strict separation between Javascript and HTML code is expected.

To the extent possible, developers pass data to the root of the React component hierarchy and each component passes down data to its children as required. This architecture can be particularly efficient when the data used to render the application changes over time; rather than directly update the Document Object Model (DOM) as in traditional Javascript applications, React uses a virtual DOM, a lightweight abstraction of the DOM in which a diff operation (an operation that detects changes) can more rapidly reveal updates to the application state in between sequential component re-renderings. React takes full control over DOM updating for this reason. Note that in the context of the Rotator project, the lack of DOM access proved repeatedly troublesome since any tool or library that requires DOM access must either be integrated outside the scope of the React framework and managed separately, else hacked to cooperate with React's rendering protocols. Even certain features of standard libraries used for this project like d3 require deliberate steps to integrate. Increasingly, React-compatible libraries are being released, though often with only partial feature support. For example, [57] is a first pass at Web Audio/React integration that proved too minimal for the purposes of the Rotator project but is still reflective of a growing trend in the developer community towards building React-compatible libraries. For the time being, the onus is often on the developer to choose alternative integration strategies in order to benefit from React. Nevertheless, it is an immensely popular tool choice.

### 5.1.2 Flux

Flux is a design pattern that allows React applications to benefit from centralized and tightly-managed data storage. It is composed of a data store, developer-defined actions called by React's view components to modify the data store, and finally a dispatcher that regulates communication with the data store. See Figure 5-1 for the general dataflow.

### 5.1.3 React + Flux in Rotator

Together, Flux and React provide a scalable and portable framework for web development. Objects remain highly decoupled from one another, and data flows approximately uni-directionally through the view component hierarchy, easing the process of tracking down bugs and optimizing the application's re-rendering speed.[2]

Figure 5-2 shows a simplified flowchart of the React view components designed

[2] Note that in the Rotator application, some breaks from this uni-directional paradigm were required, e.g. for the purpose of supporting audio synchronization across multiple React components.

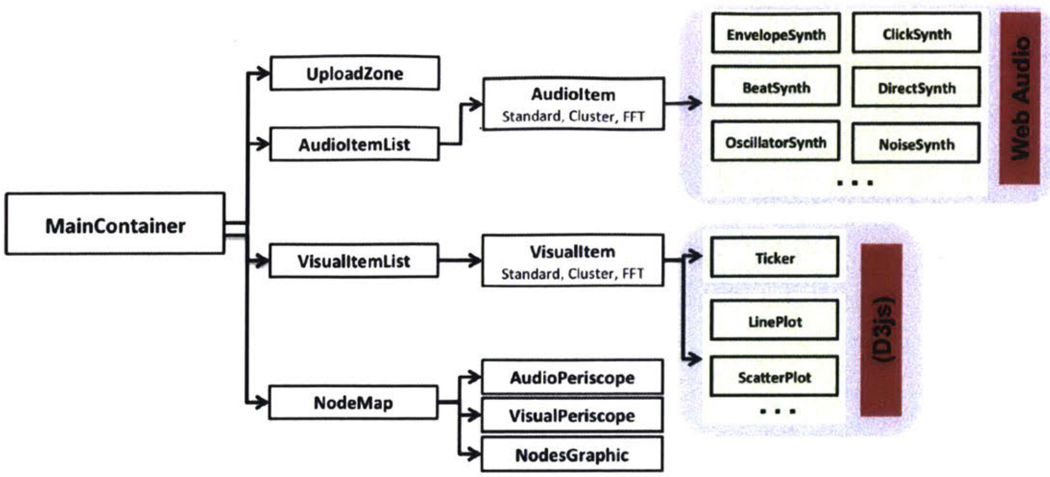



Figure 5-2: Simplified React view component hierarchy for the Rotator application. Note that additional communication loops exist that are not pictured

for the Rotator application. There are three principle data flow chains: one controlling sound synthesis, one controlling visualizations, and one controlling the node map used to navigate within a dataset. A combination of callback functions and store updates enable communication between sibling components.

## 5.2 Data Stream Format

Each data stream's header is formatted with a title, a set of (x,y) coordinates representing the node's position, and a synth type, followed by an array of data points. Figure 5-3 shows a schema for the expected CSV format as well as an example of a stream called "Body Temperature" drawn from a sensor at geometric location $(4.55, 6.9)$ to be synthesized using the beat synthesizer. Possible extensions to the interface could include drag-and-drop node placement and a user-adjustable or smartly automated synthesizer mapping control, each of which would serve as an alternative to hardcoding corresponding parameters in the data file. However, each application area considered in this thesis included a geometrically fixed node map and strong reason to keep the

```
<title>
<x_coord> <y_coord>
<synth_type>
<data_array>
```

```
Body Temperature
4.55 6.9
beat
0.0666326160454
0.0666326160454
0.0666326160454
```

Figure 5-3: Data file schema (left) and example implementation of schema (right)

synth:data mapping fixed. Therefore, the current implementation sufficed.

## 5.3 Graphics Architecture

Graphics are generated using d3: a common, low-level Javascript library for building data-driven visualizations from bottom up [6]. Certain d3 features including the brush component require DOM access, impossible within React. The React implementation of the d3 Brush component [13] was broken at the time of engineering and has a number of limitations. Therefore, a trick was used to exclude d3-dependent components from the React hierarchy while still benefiting from data passed through the component chain. React's standard *shouldComponentUpdate* method is set to always return false, whereas the *componentWillReceiveProps* method calls the appropriate component rerendering functions. In this way, React no longer controls the components, and the component is therefore free to mutate the DOM while still receiving updated props passed through the chain of React components. See listing 5.1.

```
shouldComponentUpdate(nextProps, nextState){
  return false
};

componentWillReceiveProps(nextProps) {
  if(nextProps !== this.props){
    this._updateChart(nextProps)
    this._updateTicker(nextProps);
  }
}
```

Listing 5.1: A technique for excluding d3 components from the React component hierarchy so that the d3 component can mutate the DOM

## 5.4 Synthesizer Architecture

Sound synthesis is performed using Web Audio. In the current software architecture, each data stream's associated audio synthesizer component is mounted and activated when the user wishes to sonify the stream. Six synthesizer types are currently available: **noise**, **envelope**, **clicks**, **oscillator**, and **beats**, and **direct**. Each synth's gain is independently adjustable.

### 5.4.1 Noise Synth

The white noise synth populates an audio buffer of size 100,000 with random values and then channels the buffer through a low-pass filter with a cutoff frequency driven by the data. Rapidly changing data results in a windy sound.

### 5.4.2 Envelope Synth and Click Synth

The envelope synthesizer defines an envelope's attack, decay, sustain, and release times, ramping up an oscillator to a frequency driven by the incoming data. The click synth is another iteration of the envelope synth but with parameters shorter in duration, and applied to a buffer filled with white noise. The click synth envelope parameters are defined in listing 5.2.

```
envelopeModulator.attack = .005 // seconds
envelopeModulator.decay = .005 // seconds
envelopeModulator.sustain = 0.1 // multiply gain.gain.value
envelopeModulator.release = 0.01 // seconds
```

Listing 5.2: Click synth envelope parameters

```
scale_factor = (.01 * modDepth * fund_freq) / maxVal
scaled_data_pt =  2 * maxVal (cur_data_pt - maxVal) / (maxVal - minVal)} + maxVal //scales the data between
    minVal and maxVal
output_freq = fund_freq + scaled_data_pt * scale_factor //varies output frequency from the fundamental frequency
    based on modulation depth
```

Listing 5.3: Calculation of output frequency for oscillator synth based on a maximum value, fundamental frequency, and modulation depth set by the user (note that minVal is hardcoded)

```
osc_1_output_freq = fund_freq
osc_2_output_freq = fund_freq + (1/10) * (Math.exp(Math.abs(6 * (scaled_data_pt + 0.6))) - 1) * scale_factor
```

Listing 5.4: Calculation of output frequencies for the two oscillators composing beat synth based on fundamental frequency and scale factor

### 5.4.3 Oscillator Synth and Beat Synth

The oscillator synth has user-defined fundamental frequency and modulation depth and uses incoming data to modulate the pitch of the oscillator. The beat synth creates a beat effect between two oscillators using data to modulate the small frequency gap between each oscillator. The further that the incoming data is from a preset threshold value, the higher the frequency of the beats. Furthermore, basic wave shaping distinguishes data values above and below the threshold; values below the threshold trigger two sinusoidal oscillators, and values above the threshold trigger two triangle oscillators.

The methodology used to incorporate user-defined parameters is summarized by listing 5.3 for the oscillator synth. The data-driven beat modulation for the beat synth is controlled by listing 5.4. Note that the modulation depth chosen by the user often exceeds 100% since within a particular data window, the data may only contain a fraction of the stream's total range.

### 5.4.4 Direct Synth

The direct synth populates an audio buffer that is played back at a rate determined by the user-defined tempo. This synth is not yet compatible with the moving graphical ticker and is also technically an audification rather than a sonification. Therefore direct synthesis is treated as alternative mode for the software accessible in the upper control panel.[3]

### 5.4.5 Custom Synthesizer

To properly integrate an entirely new audio synthesizer component into Rotator, a set of data parameters must be specified as props. Both the required and optional parameters are summarized in listing 5.5. Information about the data region contained in the d3 brush element, the location of the audio periscope necessary for adjusting spatialization, and all user-defined synthesizer tunings are passed into each synth component. Callback functions are used by the first initialized synth to help maintain app synchronization. Most importantly, each synth must be passed the *audioContext* object that is defined higher up in the component hierarchy. The *audioContext* is Web Audio's principle structure for storing a complete chain of audio effects built up by all active synthesizers.

```
class GenericSynth extends Component{
    propTypes{
        //ID and AudioContext
        ID: React.PropTypes.number.isRequired,
        context:React.PropTypes.object.isRequired,

        //range of values in the user-defined sliding window
        inWindow: React.PropTypes.array.isRequired,

        //spatialization parameters for controlling panner
        spatialcoords:React.propTypes.object.isRequired,
        center:React.propTypes.object.isRequired,
        spatialize:React.propTypes.bool.isRequired

        //User-defined synth settings
        scale_factor:React.PropTypes.number,
        ff:React.PropTypes.number,
        gain:React.PropTypes.number,

        //synchronization callback functions used by first synth
        setbeat:React.propTypes.func,
```

[3]Note: Web Audio streams at a sampling rate of 44,100hz and does not allow adjustments to this sampling rate. An attempt was made at resampling the audio using the *audio − resampler* React library [5], but this approach was ultimately abandoned in favor of simple playback rate changes

```
        setNextTime:React.propTypes.func,

        //synchronization accessors for all synths
        spb:React.PropTypes.number.isRequired,
        nextTime:React.propTypes.number.isRequired,
        approxbeat:React.propTypes.number.isRequired,
    }
}
```

Listing 5.5: PropTypes for Synthesizer Component Compatible With Rotator

## 5.5 Custom Audio Scheduler/Synchronizer

A principle goal of this application is to enable the user to listen to multiple data streams synchronously in order to, for example, hear correlations between streams. A sample-accurate scheduler was therefore deemed important. However, building a sample-accurate audio synchronizer/scheduler within the hierarchy of React components designed for this application poses an engineering challenge, in particular due to the order-of-magnitude 10's of milliseconds delay in sibling component rerendering. In terms of design goals, the user should be able to take the following actions while the audio is playing and expect the audio to remain synchronized across streams:

- Add one or more data streams
- Remove one or more data streams
- Change tempo
- Change window size

### 5.5.1 Standard Practices Used

As mentioned in the prior section, all Web Audio nodes are defined within an audioContext, and each audioContext has an associated high-precision clock that can normally be used to schedule sample-accurate audio. While there are no formal standards for how to design a reliable audio scheduler, a common practice is to use Javascript's setTimeout function to periodically poll whether the time of the next note to be played is within a preset schedule-ahead time. If true, then the note is scheduled using web audio's high precision clock, and the nextNoteTime and beatNumbers are advanced. An evident trade-off exists: a high-schedule ahead time results in fewer dropouts from browser lag but increased latency when a user tries, for example, to change tempo. Functionally, as long as the schedule-ahead time is large enough to accomodate Javascript's clock imprecision, this approach will work. However, a great deal of customization that will be described in this section is required in order to make this type of scheduler keep a set of synthesizer components built in React in synchrony (or, for that matter, a set of modular synthesizer components built in any framework)[4].

### 5.5.2 Customizations

A set of oscillator synths were devise a plan for the additional engineering needed to create a reliable metronome across React components. Firstly, since the Browser restricts the number of audioContexts that can be active at any time, and since we wish for all synthesizers to access the same clock, it makes sense to define a single audioContext in the app MainContainer and to pass a reference to the audioContext to the set of active audio synthesizers lower in the component chain (as previously shown in 5.5).

However, trouble arises since there are small but critical delays in rendering time between each node's synthesizer component when React re-renders the chain of components, and also because the synth components cannot directly

[4]see Figure 5-2 for a reminder of the Rotator view component hierarchy and the set of sibling synth components that must be kept in synchrony

communicate with one another but must communicate through callback function contained in parent components. Let's consider the design goals listed above and the decisions made to accommodate each design goal.

### 5.5.3 Additional Info on Add a Stream

Two cases exist: either the audio stream is the first to be added, or there already exist other streams. For the first audio stream, the beatNumber of the metronome is assumed to be 0, and next beat time is calculated based on the current time. The first audio stream will begin triggering a callback function living in the MainContainer that stores information about the subsequent beat's number and time.

When other streams are added, the next beat's number and time are instead retrieved as props sent up to the MainContainer from the first synthesizer and back down from the MainContainer to the remaining synthesizers in order to initialize the new stream, allowing the streams to remain synchronized. This is a hard task: the new component must accurately know which beat to play and at what time to play it despite the lack of access to any single accurate clock due to the order-of-magnitude tens-of-milliseconds delay in component rerendering. Roughly, the solution used is to test conditions with respect to a heavily rounded timestamp (thus accurate across all components) but still use very precise mathematically calculated timing information for actually scheduling audio events. In sum this design deviates somewhat from the unidirectional dataflow ideal that one strives to achieve when designing in React, but the use of callback functions for communicating between sibling components is still considered to be within standard practice.

An alternative design option would have been to extract the scheduler code in its entirety to the MainContainer and to periodically rerender each synth with the time the next sonified data point should be scheduled and the value of the data point. This may have simplified the rerendering flow and eliminated the need for MainContainer callback functions, but has the downside of making the audio synthesis code itself less flexible and more confined to the beat structure set in the mainContainer. For example, what if a sonification tactic for one synthesizer involved going off beat, or changing the rate of data playback? At the time of design, the perceived benefit of flexibly allowing the synthesizers to deviate by adjusting their own scheduling information merited the additional work to allow the scheduling code to live in the synthesizer component. Given the synthesis techniques ultimately selected, it is now clear that extracting the scheduler code into its own component would be a worthwhile step. This change may also improve audio synchrony with the graphical ticker added late in the engineering process. In its current iteration, the ticker refresh rate is limited and can cause app freezing when larger many visual plots are displayed at once.

### 5.5.4 Additional Info on Tempo Setting + Window Size

If the tempo is changed, one runs the risk of React synth components hearing the change milliseconds before or after one another and thus rapidly growing out of sync. Without an explicit design change, this problem was likely to set in within 5-10 user-driven tempo changes. In order to preserve synchrony, code in the MainContainer and synthesizer ensures that tempo changes only take effect 1 beat after the most recently played beat. This design decision constrains how high the tempo can be set while preserving synchrony, but has sufficed.

A user may shrink the current window size while the synchronizer is scheduling notes in the previously contained data region. To accommodate this, we set the beatNumber to 0 any time it either equals or exceeds the currently reported window size: if(this.beatNumber >= this.props.inWindow.size).

# Chapter 6

# Application Areas

In this section, two prospective application areas for the Rotator tool are summarized as case studies, and a third application area is presented in the form of a brief summary. A biosensor dataset [15] serves an example of offline analysis of a high-dimensional dataset for validating extracted features and for understanding the interplay between parameters. A simulation of Shor's algorithm [11] serves as an example of sonification of a quantum system as a more natural representation of quantum superposition. A temperature dataset from the AMS control room illustrates the possibility of using a tool like Rotator to assign one's auditory and visual senses to separate frequency scales [4]. The node map used for each application area is pictured in Figure 6-1.

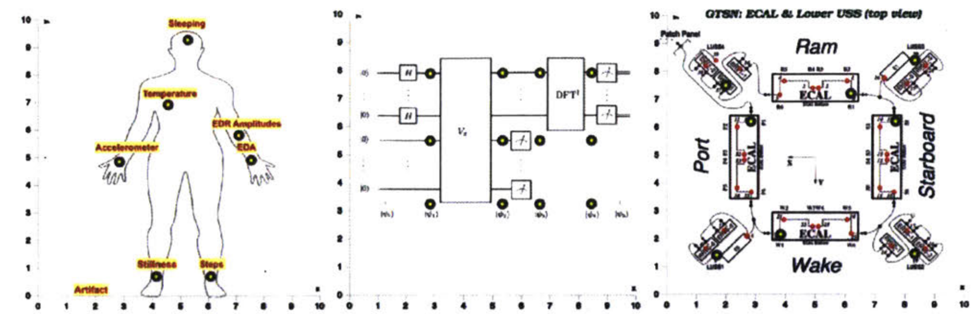



Figure 6-1: Node maps from each of the three application areas considered

## 6.1 Biosensor Data Analysis

### 6.1.1 Motivation

Despite their increased prevalence in academic research, machine learning algorithms often obscure the significance of features extracted from data and leave the researcher with minimal understanding of why an algorithm succeeds or fails. Furthermore, data analysis in machine learning is often extremely difficult and error prone, since in order to extract meaningful features the researcher must be able to see interactions between many of the raw signals recorded, and the potentially hundreds of features that can be extracted from them.

For this reason, and despite its limitations, data visualization is still regularly relied upon for validating the truthfulness and accuracy of extracted features. As an example, in the specific task plotted in Figure 6-2, the researchers involved originally computed the steps feature incorrectly and only corrected the error after plotting the raw data alongside the computed features.

However, it is also clear from Figure 6-2 that attempting to examine a large number of signals visually is in itself an arduous task. Despite plotting only two raw data signals and only four of the approximately 400 features related to the problem at hand, the figure is cluttered and confusing, and the researchers

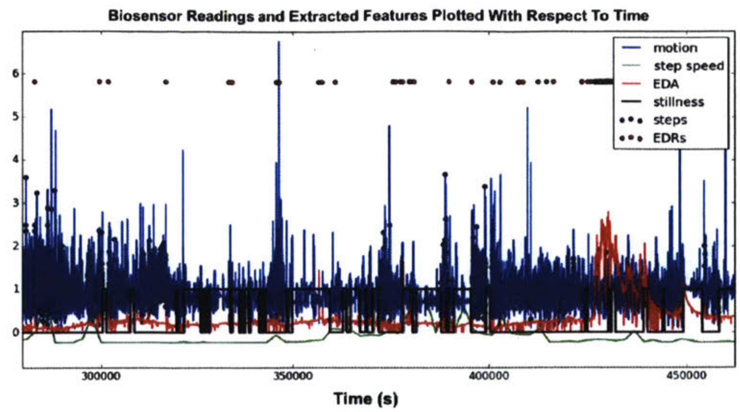



Figure 6-2: Example of raw biosensor data plotted alongside a set of extracted features, all with respect to elapsed time. The user can understand the interrelations of only a limited number of streams when plotted visually

working with it are likely to miss important information. While alternative visualization approaches may exist, any plot of e.g. six parameters at once is likely to be visually overwhelming. Dimensionality reduction schemes can help, but in this case the raw data streams and features are obscured. Our task is to integrate a dataset combining raw data and computed features into the Rotator platform to enable the researcher to distribute data between their senses.

### 6.1.2 Dataset

A dataset was provided by the Affective Computing group. This data was collected from a user study participant equipped with the Affectiva Q wrist-worn physiological sensor that records accelerometer, skin temperature, and Electrodermal Activity (EDA) data (a standard measure of variations in skin conductance in the skin of the wrist). The principle task of the researchers who own this dataset is to develop models that predict a person's stress, happiness, and ultimately, depression [64].

Because physical activity (steps), sedentary activity (stillness), and movement speed are all relevant to depression, the researcher must correctly extract these from the data. Similarly, a peak with a particular waveshape in the EDA signal represents an Electrodermal Response (EDR), which occurs due to increased Sympathetic Nervous System (SNS) activity and can be indicative of increased emotion and stress; therefore EDRs must be correctly detected as well. However, EDRs that occur during movement or when the persons body temperature is high are less likely to relate to stress. This sort of understanding of the interrelation of multiple features and signals at once plays a central role in deducing how to best analyze the data.

### 6.1.3 Integration of Biosensor Data Into Interface

The Rotator interface was originally intended for applications in which data derived from a network of sensors is analyzed. However, the dataset under discussion is derived exclusively from a single wrist-worn sensor. Therefore, a geometric layout was imposed in which each data node is spread across a human body contour line, as shown in Figure 6-1. The advantage of a scattered geometric layout is to allow for spatialization of audio which as described in chapter 4 should enhance the listeners capacity to segregate data streams. A physical map associated with a physical body is also thought to expedite the training process for learning the per-stream sonifications (though this advantage has not been explicitly tested). The disadvantage of this approach is that it may lead users to believe that the dataset was derived from a multitude of sensors situated around the body. For the purpose of this thesis, the geometric layout choice is simply explained to participants of an associated user study (see chapter 7 for additional discussion).

Custom sonifications were developed for each stream in order to maximize the listener's capacity for stream segregation based on theory addressed in chapter 4. The sonification scheme is summarized in table 6.1, sample audio files will be available at URL: **http://resenv.media.mit.edu/#Projects#rotator** and an example of sound waves juxtaposed with their corresponding input data streams is shown in Figure 6-3. Each synth's architecture will be familiar from chapter 5. This data lends itself very well sonification in that many data types are present: transients (steps, EDR), continuous rapidly varying signal (accelerometer, EDA) continuous, slow-changing signal (body temperature), and binary signals (artifact detection). Clicks and pulses are used for the transient data, which are particularly readily perceived when spatialized with respect to the listener.

The EDA signal is sonified as white noise with a cutoff frequency modulated by the data. The user may intuitively draw a connection between the rushing-water sound of white noise and the activity level of the skin. Furthermore, it has been demonstrated in prior literature, and also is rather self-evident, that transients and noise are most readily distinguished by the listener. Recalling discussions from chapter 4, this phenomenon is largely due to the fact that transients are most likely to have distinct onset times. Next, the accelerometer data is sonified as a high-frequency-band oscillator with modulated frequency. The continuous and rapidly changing nature of the raw accelerometer data lends itself well to to this simple oscillator audification. The body temperature was sonified using the low-frequency-band beat effect synth described in section 5.4. In this way, the beat frequency increases based on how far the temperature deviates from its average value. Furthermore, the wave shape change distinguishes positive and negative deviations from the average. Thus, the user can readily determine how far body temperature has drifted, and in what direction, effectively creating sonic axes, since perception of deviation is of particularly paramount importance for evaluating this particular stream. In other cases, it is expected that users will be able to learn to identify deviations from the expected sound with sufficient accuracy even without sonic axes imposed. The frequency band

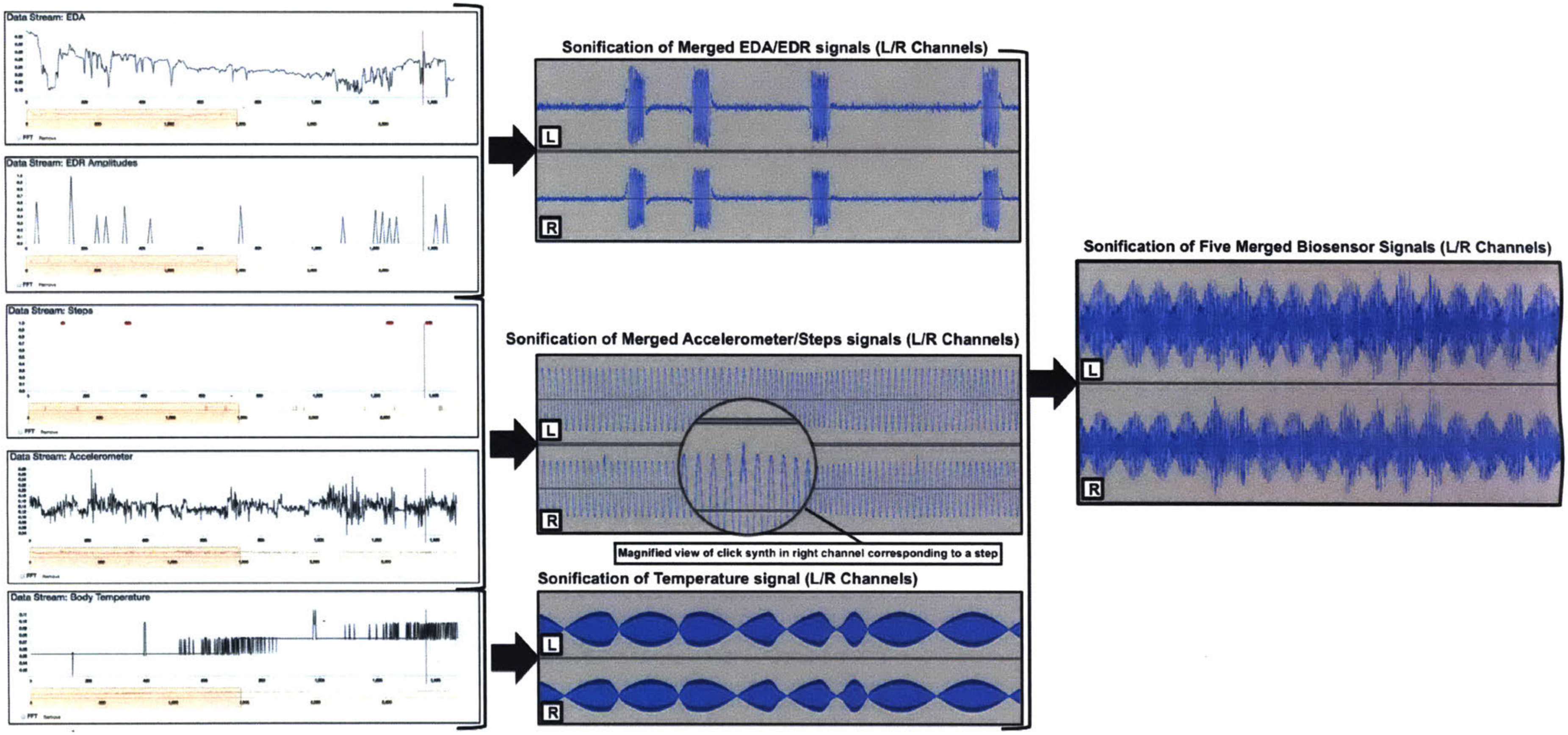



Figure 6-3: Sample sonification of five biosensor data parameters. The left-most five plots contain excerpts from each data stream. Three audio streams depict audio from EDA and extracted EDRS (top), accelerometer and extracted steps (middle) and temperature (bottom). The rightmost plot shows the resulting audio stream when all five parameters are sonified at once. Audio files are available at [1]

| | | |
|---|---|---|
| Accelerometer | continuous | Oscillator with modulated frequency |
| EDA | continuous | Filtered white noise with modulated cutoff |
| Temperature | slow-changing | Beat effect + wave shaping |
| Steps | transient | Clicks generated using envelope function |
| EDR | transient | Oscillator-driven pulse |
| Artifact Detector | binary | TBD |

Table 6.1: Biosensor Sonification Scheme Summary

of the temperature synth is kept sufficiently separated from the high frequency accelerometer synth, bearing in mind the Von Noorden diagram for pitch segregation (section 4.2). Taken together, this application area allows for the user to both verify the accuracy of extracted features (in many cases the feature extraction algorithms are far from perfect), as well as derive a feeling for the stress and activity levels of the person under study by growing acquainted with the auditory interplay between variables as well as by distributing data between their visual and auditory senses. This application area is the subject of a user study conducted to evaluate the Rotator interface. Therefore, more detail is provided in chapter 7.

## 6.2 Shor's Algorithm

Audio may be a very suitable data display mode for quantum systems. This presumption is largely due to the nature of quantum superposition, a phenomenon in which a quantum particle exists in many states at once with some probability of a measurement yielding each state. As we have just seen in section 6.1, audio can be layered in a way that enables the simultaneous experience of many streams at once. Here we use Rotator to sonify the quantum states that arise in a famous quantum factoring algorithm called Shor's algorithm.

### 6.2.1 Background and Related Work

Mathematically, a quantum system is comprised of a set of state vectors and corresponding probability amplitudes. In time-dependent systems, a time-dependent term evolves the quantum superposition such that the probability amplitudes for each term change predictably. Equation 6.2.1 shows a general representation of a time-dependent quantum state with $c_n^{(0)} e^{\frac{-iE_n t}{\hbar}}$ corresponding to the evolving coefficients of each state, and $\psi_n(x)$ corresponding to the states.

$$\psi(x,t) = \sum_{n=1}^{\infty} c_n^{(0)} e^{\frac{-iE_n t}{\hbar}} \psi_n(x) \tag{6.1}$$

It is notoriously difficult to convey quantum superpositions in a visually convincing manner. Usually, visualizations of the system's corresponding probability distribution functions (PDFs) or density matrices suffice (for example, see the QuTiP tool Python Quantum Toolbox [65]). For a two-level quantum system, the Bloch sphere can be used to encode the amplitudes of the $|0\rangle$ and $|1\rangle$ states as well as the state's overall phase. Finally, a nontrivial number of attempts at more complex visualizations for higher-dimensional quantum systems have been described as well, including but not limited to [52][47][85][55]. However, only a small number of quantum system sonifications have been attempted ([44][70][84][68][96]).

### 6.2.2 Overview of Shor's Algorithm

The most fundamental building block of a quantum algorithm is a qubit- a quantum bit that, when measured, will yield one of two states, frequently written in Dirac notation as $|0\rangle$ and $|1\rangle$ or as vectors:

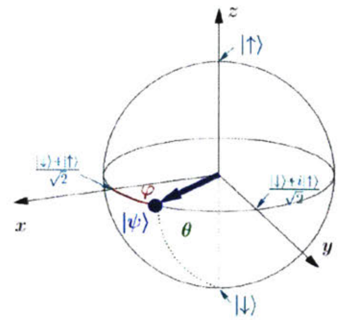



Figure 6-4: Example of Bloch sphere visualization that can convey the state of a single qubit

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

However, prior to measurement, the qubit can exist in any vector state spanned by this two-dimensional vector space.[1] In order to perform more complex computations, one can construct a high-dimensional system of multiple qubits by taking the tensor product of a set of qubits. Corresponding high-dimensional linear operators can also be generated by taking the tensor product of low-dimensional operators. The high-dimensional operator can act on all states in a quantum superposition simultaneously through an effect known as quantum parallelism, and quantum parallelism is at the heart of the speedups experienced in quantum algorithms relative to their classical counterparts.

Shor's algorithm is a famous algorithm that factors numbers in polynomial time, faster than the classical algorithms which run, at best, in sub-exponential time. A brief summary of the algorithm sufficient for readers of this thesis is provided in this section, but consult e.g. [71] for a very commendable treatment.

Suppose that we wish to find the prime factors of N. Firstly, we make an important observation regarding the series $\sum_x a^x \pmod{N}$:

[1] these vector states correspond to points on the Bloch sphere

$$
\begin{aligned}
\sum_x a^x \pmod N)|_{n=15,a=2} &= 2^0 \pmod{15} + 2^0 \pmod{15} + 2^1 \pmod{15} + 2^2 \pmod{15} \\
&+ 2^3 \pmod{15} + 2^4 \pmod{15} + 2^5 \pmod{15} + 2^6 \pmod{15} + \ldots \\
&= 1 + 2 + 4 + 8 + 1 + 2 + 4 + 8 + \ldots \\
\sum_x a^x \pmod N)|_{n=21,a=2} &= 2^0 \pmod{21} + 2^0 \pmod{21} + 2^1 \pmod{21} + 2^2 \pmod{21} \\
&+ 2^3 \pmod{21} + 2^4 \pmod{21} + 2^5 \pmod{21} + 2^6 \pmod{1} + \ldots \\
&= 1 + 2 + 4 + 8 + 16 + 11 + 1 + 2 \ldots.
\end{aligned}
\tag{6.2}
$$

As illustrated in the above two examples for N=21 and N=15, there is a periodicity $r$ that arises in the series $\sum_x a^x \pmod N$ which we call the 'order'. Critically, the order of N is proven to evenly divide $(p-1)(q-1)$, where $p$ and $q$ are the prime factors of N. For example, for N = 21, we see in equation 6.2.2 that the order $r$ is 4, the prime factors of $N = 15$ are 5 and 3, and indeed $(p-1)(q-1) = 8$ which is divisible by 4. Similarly, for N = 21, the order $r$ is 6, the prime factors of $N = 21$ are 7 and 3, and indeed $(p-1)(q-1) = 12$ which is divisible by 6. To make use of this series to compute prime factors, we take the following steps[2]:

1. Choose an integer $x < N$
2. If x, N have common factors, then GCD(x,N) give factors of N
3. Else, x is coprime to N, so we compute the order $r$ of $x^j \pmod N$
4. Once a suitable r is known, then taking $GCD(x^{r/2}, N)$ yields factors of N

All would be well if there were a fast classical order-finding algorithm, but it turns out that for large N, the order may be nearly as large as N itself.[3] For a fast order-finding algorithm, we turn to quantum mechanics. Figure 6.2.2 shows the circuit diagram for the quantum order finding algorithm. Two registers of qubits are prepared in the 0 state (let's say that $t$ qubits compose the first register and $n$ qubits compose the second register). Next, a $H^{\otimes(8)}$ operator[4] is

[2]GCD is 'greatest common denominator' operation for which a fast classical algorithm exists

[3]famously, the difficulty in factoring large numbers is the principle behind RSA encryption

[4]Notation: $\otimes$ indicates a tensor product. $H^{\otimes(n)}$ corresponds to taking the tensor product of n Hadamard gates, yielding a higher-dimensional operator

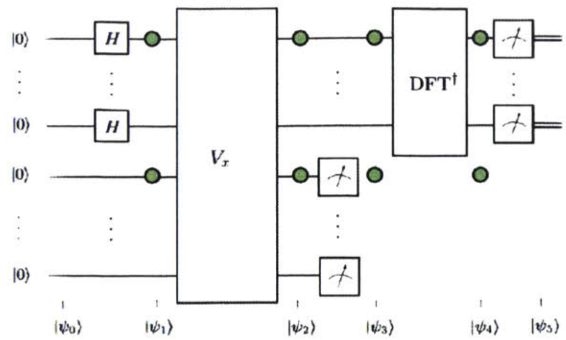



Figure 6-5: Quantum circuit diagram for Shor's algorithm. Algorithm stages that are sonified are marked with green circles (one green circle per quantum register at each stage $\psi$ that is sonified)

applied to the first register. The Hadamard operator is a simple matrix operator that expands the set of qubits into a superposition of all basis states with equal amplitude given by $\frac{1}{\sqrt{2^t}}$ such that the state remains normalized. This stage of the algorithm corresponds to state $\psi_1$ in Figure 6.2.2.

Next, a unitary linear operator $V_x$ is applied, where:

$$V_x = (|j\rangle\,|k\rangle) = |j\rangle\,|k + x^j\rangle$$

and $|j\rangle$ and $|k\rangle$ are the states of each of the registers of the quantum system. Application of $V_x$ yields, generally:

$$|\psi_2\rangle = \frac{1}{2^t}\sum_{j=0}^{2^t-1} |j\rangle\,|2^j modN\rangle$$

The quantum nature of the system allows for all powers to be computed simultaneously and stored in the superposition. Furthermore, there is now a periodicity in the second register. Regrouping terms with respect to the second register, a linear sequence in the first register also becomes apparent. This is most clearly seen by example for e.g. N = 21:

$$
\begin{aligned}
\psi_2 = \frac{1}{512}[ & (|0\rangle + |6\rangle + |12\rangle + \ldots + |504\rangle + |510\rangle)\,|1\rangle \\
& (|1\rangle + |7\rangle + |13\rangle + \ldots + |505\rangle + |511\rangle)\,|2\rangle \\
& (|2\rangle + |8\rangle + |14\rangle + \ldots + |506\rangle)\,|4\rangle \\
& \ldots \\
& (|5\rangle + |11\rangle + |17\rangle + \ldots + |509\rangle)\,|11\rangle]
\end{aligned}
\tag{6.3}
$$

Performing a measurement on the second register yields any of the above states, each of which has the same periodicity. The quantum state after the second register measurement is $\psi_3$ in Figure 6.2.2.

Finally, in order to unveil the periodicity that we seek, we apply a quantum Fourier transform (QFT) operator, which results in a probability distribution containing very sharp peaks a distance $\frac{2^t}{r}$ apart ($\psi_4$). Measurement of the state after the QFT operator is applied ($\psi_5$) is extremely likely to yield one of these peaks, which is subsequently used to calculate r (sometimes with additional iterations of the algorithm required).

### 6.2.3 Integration With Rotator

We now bring the discussion back to the Rotator tool. Shor's algorithm is an interesting choice for sonification. On the one hand, it is a quintessential quantum algorithm with a periodic structure that can be elucidated sonically. On the other hand, the high bit numbers per register required even for small N result in very large superpositions of states to work with. The states have equal amplitude at many stages of the algorithm (as compared to e.g. the quantum harmonic oscillator where a time-dependent term modulates the probability amplitudes as the system evolves), and the periodic structure in the algorithm does not lie in the probability amplitudes of the states, but instead in the series of states themselves. To make this periodicity apparent in audio, we must iterate through the states in superposition rather than play them simultaneously,

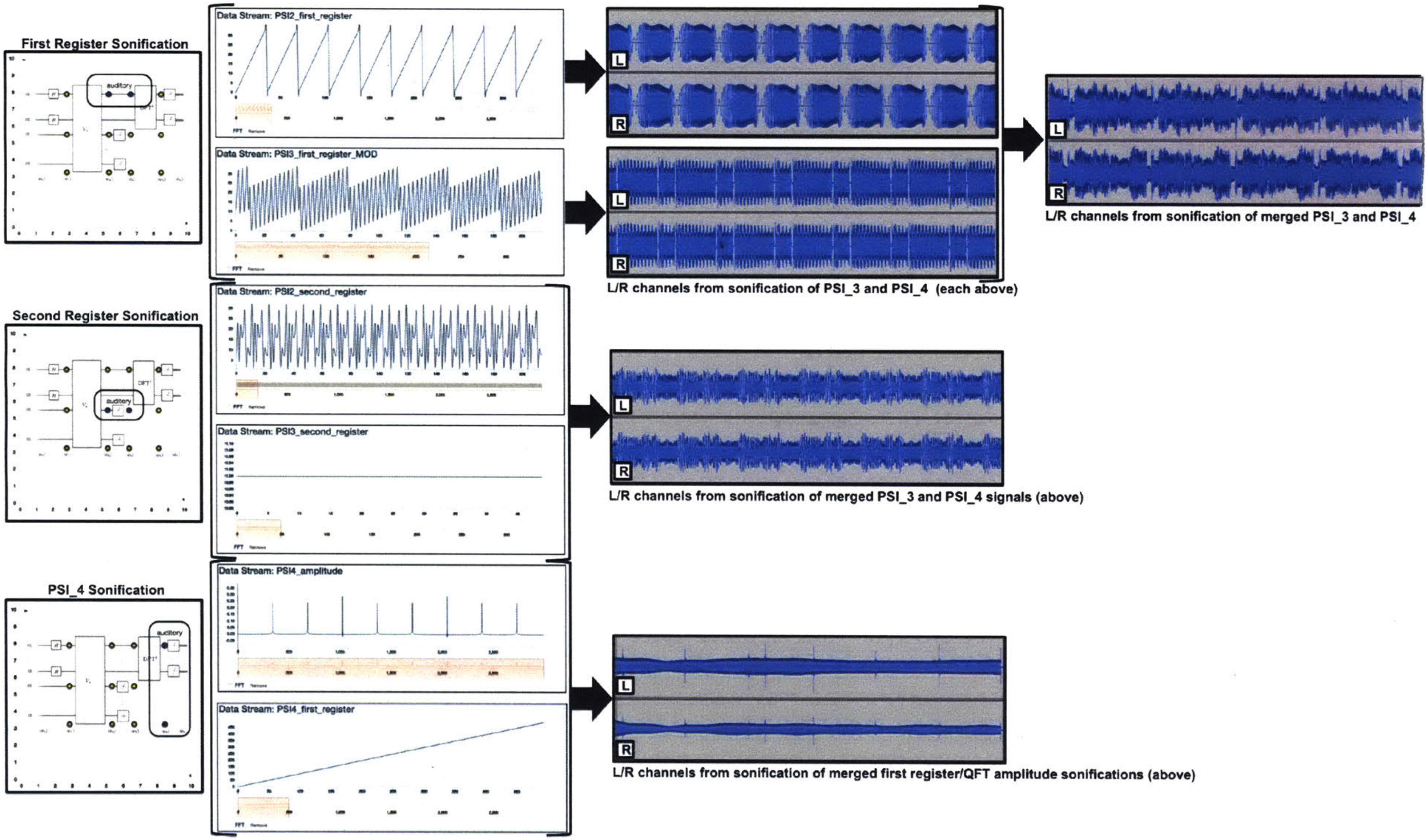



Figure 6-6: Sample audio and visual captures of qubit states from Shor's algorithm integrated into the Rotator platform

which breaks from our initial motivation to sonify quantum systems in the first place! In the literature, only one other example of a sonification of Shor's algorithm exists [96] and in this case the composer chose to sonify each and every quantum gate contributing to the algorithm rather than sonify the stages of quantum states, a wholly alternative sonification approach that arguably misses an opportunity to elucidate the notion of quantum superposition.

Still, a quantum algorithm circuit diagram is ideal for integration with Rotator, since it serves as a node map that can guide users in exploring different stages of the quantum algorithm. The following initial approach was taken:

First, a Javascript-based simulation of Shor's algorithm was located within a quantum computation tool called jsqubits [11]. The simulation was modified to output data at each of four stages labeled $\psi$ in the included circuit diagram. Sample files were read out as three 1D-arrays corresponding to first register states, second register states, and amplitudes for $N = 35$.[5] First register and second register states were placed in appropriate locations in the circuit diagram node map for $\psi_1$, $\psi_2$, $\psi_3$, and $\psi_4$, and amplitudes were placed as nodes below the map. A mod N operator was applied to $\psi_2$ and $\psi_3$ in order to make a periodicity more apparent in audio.

$\psi_4$'s amplitude stream corresponding to the output of the QFT was sonified using the 'clicks' synth. Since there is predictable variation in the spread of each peak in the QFT output, a single click indicates a narrow peak and 2-3 clicks indicate a peak with more spread, with click count corresponding to the sampling rate of the signal. The remaining streams were sonified using oscillator synths, where the oscillator's pitch is modulated based on the data stream being sonified. The pitch band separation between oscillators, the periodicity in many of the streams, and the audio spatialization all contribute towards

[5] Since jsqubits is implemented in Javascript, full integration of the platform into Rotator would be relatively straightforward in the future.

keeping the streams segregated, in particular given that a user is likely to only require listening to a maximum of 2-3 streams at a time in this application. Figure 6-6 shows excerpts of visual and audio captures from some example use cases. These include listening to the first register before and after the second register is measured as well as listening to the second register before and after it is measured. In each of these cases, one can see that the placement of the Rotator audio window will generate spatialized audio in which the user hears the state of the quantum system at $\psi_2$ to their left, and the state of the quantum system at $\psi_3$ to their right, almost as if the sequential states existed in superposition (which is not a viable interpretation of the physics of the system, but is nevertheless an interesting method for creating a feeling of presence within the algorithm as one seeks to understand its progression). The previous quantum state is heard to the listener's left, and the next state is heard to the listener's right. As a reminder, the first two examples in figure 6-6 correspond to sonification of the actual states of each register of the quantum system. One can also separately sonify the amplitudes of each state in either of these sonifications; although the amplitudes within each term of state $\psi_2$ and $\psi_3$ are equal, the change in amplitudes between $\psi_2$ and $\psi_3$ cues the listener into a net reduction of possible states after register two has been measured, given that the states are normalized. The final example in Figure 6-6 shows a user listening to the amplitude stream and first register stream of $\psi_4$ after the QFT has been applied, hearing the sequential amplitude peaks that are known to be multiples of $\frac{2^t}{r}$. In sum, one can use Rotator to explore the sequential quantum states of an algorithm. It may be more fitting to call this application area a sonification of the quantum order-finding algorithm since iterations of the order-finding algorithm that are often needed to determine the factors, as well as the final steps used in the algorithm to compute the prime factors of N are omitted[6].

Although the integration approach we have just described was suitable for the Rotator platform, it left more to be desired in terms of our initial motivation to use auditory superposition to convey quantum superposition. As articulated earlier, the high bit counts required in many quantum algorithms at first made it hard to imagine the usefulness of auditory superposition in an application area like this one. However, looking back at equation 6.2.2, we see that at the $\psi_3$ stage of Shor's algorithm, the total number of bits in register 2 for e.g. $N = 35$ is reduced to 12. For smaller N, an opportunity is created to hear the 12 corresponding first registers in superposition, where the value of the second register is used as a control bit to set the fundamental frequency of a oscillator with pitch modulated by the sequential states of each term's first register. Modifications are required in the Rotator application to support this approach. The output consists of 12 oscillators that have a small pitch shifts with respect to each other, due to the shifts between each first register term apparent in equation 6.2.2. While not yet fully developed, this approach is closer to [84] in the relevant literature and is a more versatile implementation approach for any future quantum system sonifications in Rotator.

## 6.3 AMS Temperature Data (Brief Summary)

A brief description of a third application area is given in this section

### 6.3.1 Introduction

The alpha-magnetic spectrometer (AMS) is a particle physics detector aboard the International Space Station responsible for detecting particles that may con-

[6]these final steps are purely classical, which motivated the omission

tribute to our understanding of dark matter, anti matter, and cosmic rays [91]. Much like any experiment demanding constant monitoring, an AMS control room located at CERN is constantly manned. AMS is installed on the ISS main truss. Among other parameters, a set of temperature nodes located throughout AMS are carefully monitored for any signs of overheating or malfunction. A dataset consisting of 8 temperature nodes located throughout the detector was provided. The AMS RAM and WAKE sides of the detector, which contain radiating panels, are facing away from the truss. AMS PORT and STARBOARD sides are instead facing along the truss length. On the STARBOARD side there is another payload that causes thermal conditions to be different. In summary, different sides receive different sunlight depending on their relative angle with respect to the Sun and on the actual geometry of the payloads installed in the station, and this changes across an orbit and as a function of spacecraft attitude.

The position of each node on the detector impacts the temperature of the node, and furthermore there are oscillations in the data at different timescales corresponding to 92-minute orbitals (high frequency) and lower frequency drift in orbit of the International Space Station that progressively changes the average ISS angle with respect to the sun direction (known as the beta angle).

### 6.3.2 Motivation and Sonification

Rather than aim for maximal stream segregation, we suspect that an array of temperature nodes lends itself to a sound that blends together, creating the feeling of being surrounded by a temperature field. As users grow accustomed to this ambient hum, they will be able to identify warmer and cooler regions of the detector, and can hear the temperature changes move around the detector as its orbit changes. Sudden change of the ISS attitude (referring to orientation in space) may result in sudden changes to AMS temperature readings. ISS attitude changes may, for instance, be in response to a request for facilitating the docking of an incoming vehicle.

To sonify the data, a simple mapping was created using eight spatialized oscillator synths, with temperature controlling the pitch of each synth.

### 6.3.3 Multi-Scale Perception

This dataset was particularly appealing due to its behavior at multiple time and frequency scales. In order to perceive both scales at once using visual tools, it is necessary to either zoom in and out, else look back and forth between two plots at different time scales. A tool like Rotator can be used to give the user a zoomed view of the low frequency oscillations of the AMS temperature data, while an accompanying visualization simultaneously provides the user with larger timescale, low frequency behavior. This behavior is most readily accomplished by increasing the modulation depth of the stream's sonification in order to amplify smaller-scale changes in the sound. Meanwhile, a lowpass-filtered signal can be visualized in order to remove higher frequency behavior from the visualization. In this way, the user is able to simultaneously track temperature variations of a temperature node at two different timescales, while only monitoring a single auditory and single visual track.

In the context of this thesis, it was decided to hone in on the biosensor application area due to the diversity of data available and to save any further exploration of multi-scale perception for future work.

# Chapter 7

# Evaluation

## 7.1 Introduction

The biosensor application area was selected in order to test Rotator because this application area is the most developed and thus has the widest diversity of sonification schemes built for it. Specifically, a small-scale user study was conducted in order to gauge the influence of adjustable perceptual modes on a user's ability to draw conclusions about the structure of a dataset.

## 7.2 Methodology

The biosensor dataset used in this study consisted of one day of activity from a person who self-reported as very stressed, and one day of activity from a person who self-reported as very calm in an effort to maximize the diversity of data samples. Broadly, the task given to users was to classify approximately six-minute-long excerpts from the biosensor time series dataset on a 5-point

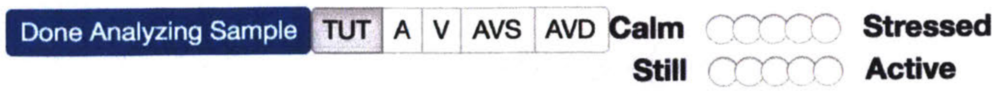



Figure 7-1: Control panel for user study. Under different perceptual conditions, users rank biosensor data samples based on perceived stress and activity levels in the sample

ranking of stressed/relaxed and active/still[1]. The first ~45 minutes served as a training session. The training session consisted of the following steps:

- Introduction to audio samples drawn from each of the five streams one-by-one
- Introduction to visual samples drawn from each of the five streams one-by-one
- Discussion of table 7-1 used as a qualitative guide for ranking stress/activity
- Introduction to audio and visual samples from a high-stress, low-stress, high-activity, and low-activity participants
- Opportunity to listen to samples of all audio streams playing simultaneously

Table 7-2 shows a reference given to all users for evaluating trials on the basis of stress and activity. The table is based on a heuristic commonly used for evaluating stress conditions on the basis of a user's EDA signal. Note that in addition to the stressed and active heuristics, a heuristic for a more charged emotional state was given (high temperature/EDA, low accelerometer motion). Users were instructed to consider a participant with a charged emotional state as having some increased likelihood of being stressed.

It is important to note that there are many biological caveats that are omitted from this study. For example, during sleep, people are known to experience 'EDA storms', huge surges in EDA reading that, if body temperature is low, could easily be confounded with a stress response when analyzed by a naïve researcher. For the purposes of a perceptual study, confounding factors like sleep status were not important (and in fact an 'isSleeping' stream was subsequently added to the interface). It is only

[1] Note that the user could set arbitrary rates for scanning the data

**Summary of States**

| Temperature | EDA/EDR | Acceleration/Steps | State |
|---|---|---|---|
| Low | High | Low | Stressed |
| High | High | Low | Sweating, some other emotional state |
| High | Either | High | Active, exercising |

Figure 7-2: Excerpt from an information sheet provided to users

important that the provided identification instructions be consistent across all user study participants.

After the training session, users were asked to rank between five and ten data samples under four perceptual conditions: (1) all streams presented as audio ('A'), (2) all streams presented as plots ('V') (3) all streams presented as audio and up to one stream at a time simultaneously presented visually ('AVS') (4) A subset of streams presented as audio and up to one stream at a time presented visually; the visual stream *cannot* also be sonified ('AVD'). Conditions (3) and (4) distinguish between a scenario where the user is observing a plot that they simultaneously hear and a scenario where the user is seeing a plot that they do not simultaneously hear. As long as the conditions of each trial were met, the users were free to move around the audio and visual boxes (see figure 3-2 as a reminder) as well as adjust the synthesizer mapping parameters available in the control bar. Figure 7-2 shows the input panel that was integrated into the Rotator interface for the purpose of the user study. The 'Done Analyzing Sample' button loads in an arbitrary 6-minute-window of data for the next test. Browser localStorage was used to collect responses.

After each of the four trials, the users were asked to fill out a NASA task load index (NASA-TLX) survey, a widely used assessment tool with over 6000 citations used for ranking perceived cognitive workload to complete a task [56]. At the end of the study, users filled out a final survey regarding their subjective experience performing the required task under each condition.

## 7.3 Results

Due to the small sample size of this user study, it is important to recognize that results are suggestive of trends, but should not be taken as conclusive.

### 7.3.1 NASA TLX Results and Discussion

To evaluate the subjective cognitive workload experienced by users after each task, each participant filled out the NASA TLX four times: once after each trial. The rounded and weighted NASA TLX scores for each participant are provided in Figure 7.1. We make a few key observations from the results in Figure 7.1: firstly, the participants' self-reported audio experience correlates in all 5 cases with their perceived task load ranking of the audio-only trial - that is, participants with more audio experience endured less cognitive workload when presented data sonically, and participants with less audio experience endured more cognitive workload when presented data sonically. This result confirms a widely verified conclusion in sonification literature that extended training periods and comfort working with audio contribute to one's ability to make use of sonification tools. Note that while in all cases, NASA-TLX cognitive load measurement for the auditory-only scenario were highest, 4 out of 5 users also stated that of all the conditions, their performance in the all-audio condition seemed to improve the most (this result, a survey question, is not pictured).

Secondly, and very critically, we observe in Figure 7.1 that for 4 out of 5 participants, the 'AVS' task was ranked as requiring nearly equivalent cognitive load as the all-visual task (the AVS task, as a reminder to the reader, is the task in which all audio was sonified but one stream at a time could also be visualized, almost like a 'peeking' mode). For the one participant for whom this was not the case, the 'AVS' task was still ranked the easiest to complete among the three out of four trials involving audio (NASA TLX of 69 compared to NASA TLX 77 for 'AVD' and NASA TLX of 83 for 'A'). Furthermore, in addition to self-reporting no prior audio experience, this subject verbalized extreme lack of familiarity working with audio throughout the

| Participant | TLX ('A') | TLX ('V') | TLX ('AVS') | TLX ('AVD') | Audio Experience? |
|---|---|---|---|---|---|
| 1 | 61 | 51 | 43 | n/a | A lot |
| 2 | 69 | 42 | 46 | 52 | A lot |
| 3 | 73 | 33 | 33 | 48 | None |
| 4 | 58 | 51 | 50 | 54 | Some |
| 5 | 83 | 38 | 69 | 77 | None |

Table 7.1: Summary of NASA TLX weighted scores for each of 5 user study participants under four different perceptual modes, which self-described audio experience marked for each participant. TLX scores are out of 100, with higher TLX scores indicate greater perceived cognitive workload, and lower scores indicate less perceived cognitive workload. See body of text for descriptions of perceptual modes

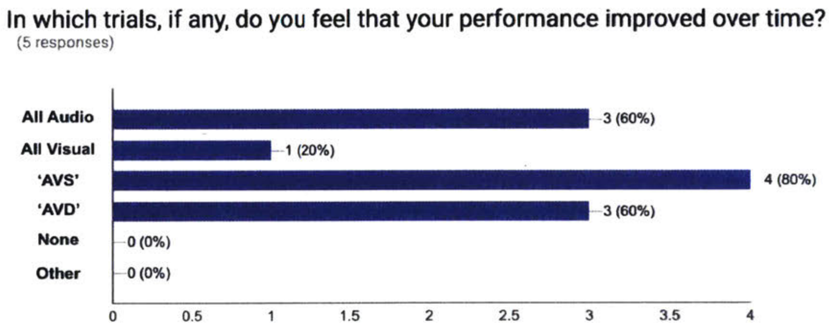



Figure 7-3: Perceptual conditions in which participants felt they improved through the duration of the study, as reported in a post-study survey

2 hour testing period. Further validating the NASA-TLX results indicating relative comfort under the 'AVS' condition, in the post-study survey, 4 out of 5 users included the 'AVS' state among trials in which they felt performance improved over time (see Figure 7-3).

This result is a promising first indication that there is merit to a tool like Rotator. One may have expected, given our familiarity with interpreting visual data, that participants would have strongly preferred the all-visual case. Instead, as summarized in Figure 7-3, most expended nearly equal subjective effort when only one plot was visualized at a time (compare the 'AVS' mode and 'V' mode in Figure 7-3). Furthermore, the all-audio trial was self-reported by all participants as requiring the most effort.

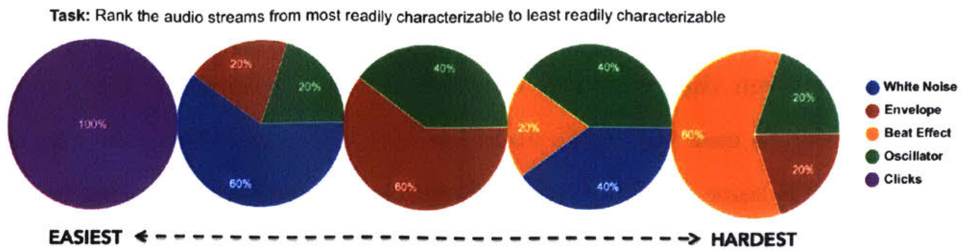



Figure 7-4: Ranking of the five synthesizers on the basis of how readily identifiable they are when played in tandem

There are a number of confounding factors to consider. Firstly, participants are only exposed to one possible visualization methodology, so it is possible that a different visualization approach would impact results. Secondly, the 'A' task was performed prior to the 'AVS' and 'AVD' tasks for all participants, meaning that they may have grown more accustomed to the audio during the latter tasks. However, based on textual user reports, the visual peeking feature in the 'AVS' mode was particularly useful as a means of validating any auditory cues. One participant writes 'I definitely felt the most confident in the 'AVS' scenario, because I could get a quick sense from the audio and then allay any questions/concerns with a few targeted visual queries.' another writes: 'The visual information was especially helpful in verifying what I was hearing.'

## 7.4 Evaluation of Synthesizer Choices

Users were asked to rank each of the 5 synthesizer types on the basis of how easily identifiable they are when played in synchrony. Rankings are shown in Figure 7-4. Interestingly, there was great consistency among the ratings, despite the discrepancies in prior audio experience and perceived task workload. All users rated the steps most easy to identify (sonified using the clicks synth) and 3/5 users rated the EDA second easiest to identify (sonified using the noise synth). Thus the windiness of white noise with modulated cutoff frequency, and the abrupt and spatialized transients, are most easily distinguished by participants. On the other hand, temperature data, sonified using the beat synthesizer, was least readily identifiable by participants. This is a curious result given that we recall the decision to impose sonic axes on the beat stream in order to ease the user's ability to determine whether temperature is low or high. There is some possibility that the accelerometer (sonified using an oscillator synth) interfered with user perception of the beat synth. Given the wide distribution in reported identifiability of the accelerometer, this possibility is inconclusive. A far more likely explanation involves the additional steps required to learn the features of the beat synthesizer e.g. remembering what high and low beat frequencies indicate, as well as remembering and identifying the two wave shapes corresponding to high and low. Perhaps with additional training the stream would become more readily identifiable.

## 7.5 Evaluation of Task Completion Times

Many users requested additional training time beyond the alotted 45 minutes for any future studies, and in particular would appreciate immediate performance feedback during the training. It is interesting to look at performance time across trials as a metric for effectiveness of training. Figures 7-5 and 7-6 show measures of performance time across trials and modes. Interestingly, participants with more experience working in audio (participants 03 and 07 in figure 7-5) actually took longer to complete the audio-based trials despite rating the task as requiring less cognitive effort according to the NASA-TLX. This may be due to an increased capacity for discerning auditory differences and therefore a desire to listen more carefully. Or, as one participant with some audio processing experience stated: "sorry I took so long, but the audio mode was more fun."

Overall, we again see that for most participants, the time required to complete the 'AVS' mode was very comparable to the time for completing the task all visually (see Figure 7-5). However, 7-6 shows only hints of improved performance speed over time, at least in the small number of trials users were exposed to in the study.

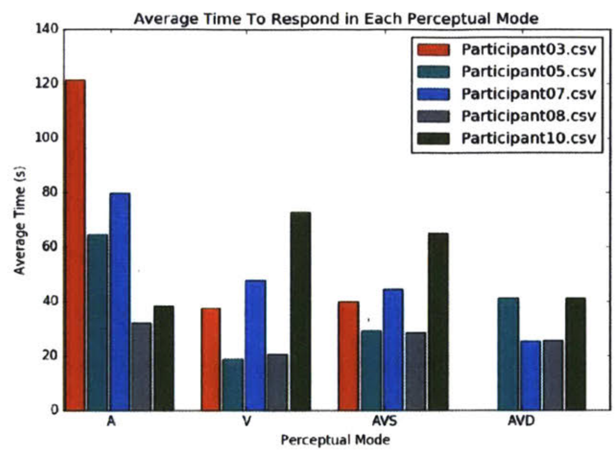



Figure 7-5: Ranking of average time spent in each mode for each participant Note that participant 03 did not complete the 'AVD' trials. Also note that participants 03 and 07 self-reported as having more audio experience

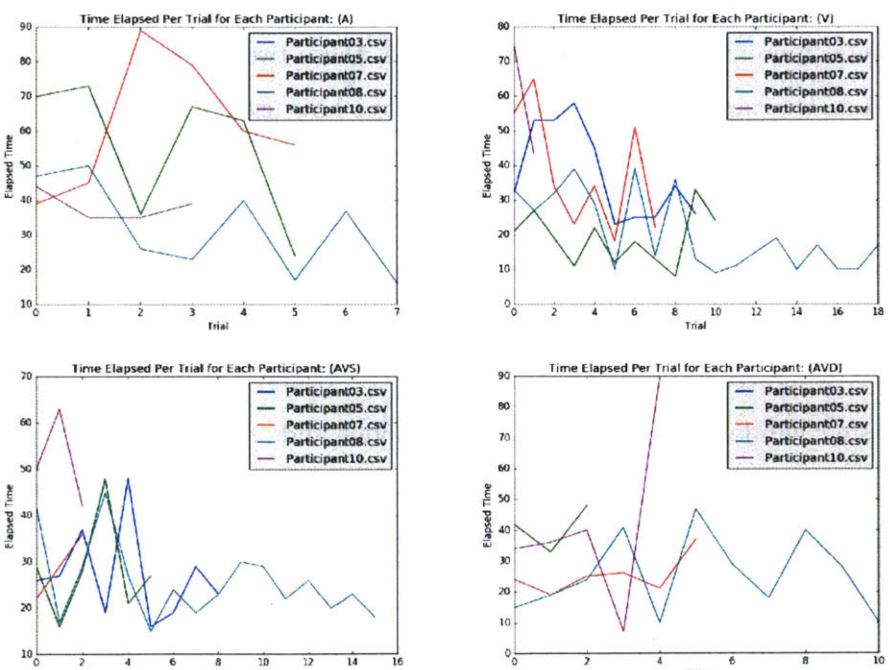



Figure 7-6: Per-Participant Times Across Trials

## 7.6 Evaluation of Task Accuracy

It is difficult to verify the accuracy with which users completed each task since there is no source of ground truth for the stress or activity level represented by the data. However, we can treat certain measures such as step count, EDR count, and average temperature as quantitative measures of each data stream and compare how readily users perceive each parameter under different perceptual conditions. Recall Figure 7-2 showing the relationships between parameters that users were seeking to identify. When all 3 parameters (step count, EDR count, and average temperateure) were plotted with respect to the user study participant's rating of activity and stress, no patterns were immediately discernable (see figure A-1, included as an appendix, for an example). Thus we immediately turn instead to single-parameter plots.

The numbers of detected steps that a person takes serves as the closest coarse 1-dimensional measure of their activity level. However, if users purely ranked according to step count, they would be misperforming, and failing to take into account the additional biosensor parameters. Therefore, these two dimensional plots should be treated as very approximate. Figure 7-7 shows how users ranked activity level based on step count for each sensory mode. Audio rankings show the most spread in ratings, suggesting the highest degree of perceptual ambiguity. Curiously, both 'V' and 'AVD' rankings (plots 2 and 4) show the most linear relationship between step count and activity rating, despite the fact that we saw above that users rated 'V' and and 'AVS' as areas where they predicted most comfort. Users were free to adjust the audio and visual windows during both 'AVD' modes and 'AVS' modes so it is unknown exactly how users were choosing to perceive the steps synth within any given trial. Still, the linearity of the 'AVD' plot lends credence to the merit of data presentation modes with auditory and visual separation. However, looking at equivalent plots comparing EDR count to stress rating (Figure 7-8), the most linearity occurs in the purely auditory and purely visual modes, with worst performance in the 'AVD' mode, suggesting that the synthesizer type guides which perceptual mode is most effective.

With more data from an expanded user study, we could perform a principle component analysis to better understand how all data parameters contributes to the user's likelihood of ascribing a high rating to a data sample's stress or activity measures.

## 7.7 Summary of Results

The most promising conclusion from this user study is the noteworthy jump in perceived task ease, NASA-TLX workload, and in some cases actual performance for the 'AVS' presentation mode. In particular, the ability to peek at individual data streams dramatically raised a user's comfort and performance time when compared with the audio-only mode. Rotator is expressedly designed to promote dual perceptual modes just like the 'AVS' mode, in which the large majority of data streams are relegated to one's auditory periphery and only a few data streams are visualized. Furthermore, there is a lot of evidence to suggest that performance would improve both in speed and accuracy if the total experimentation time were to extend beyond the two-hour sessions allocated per participant of this study.

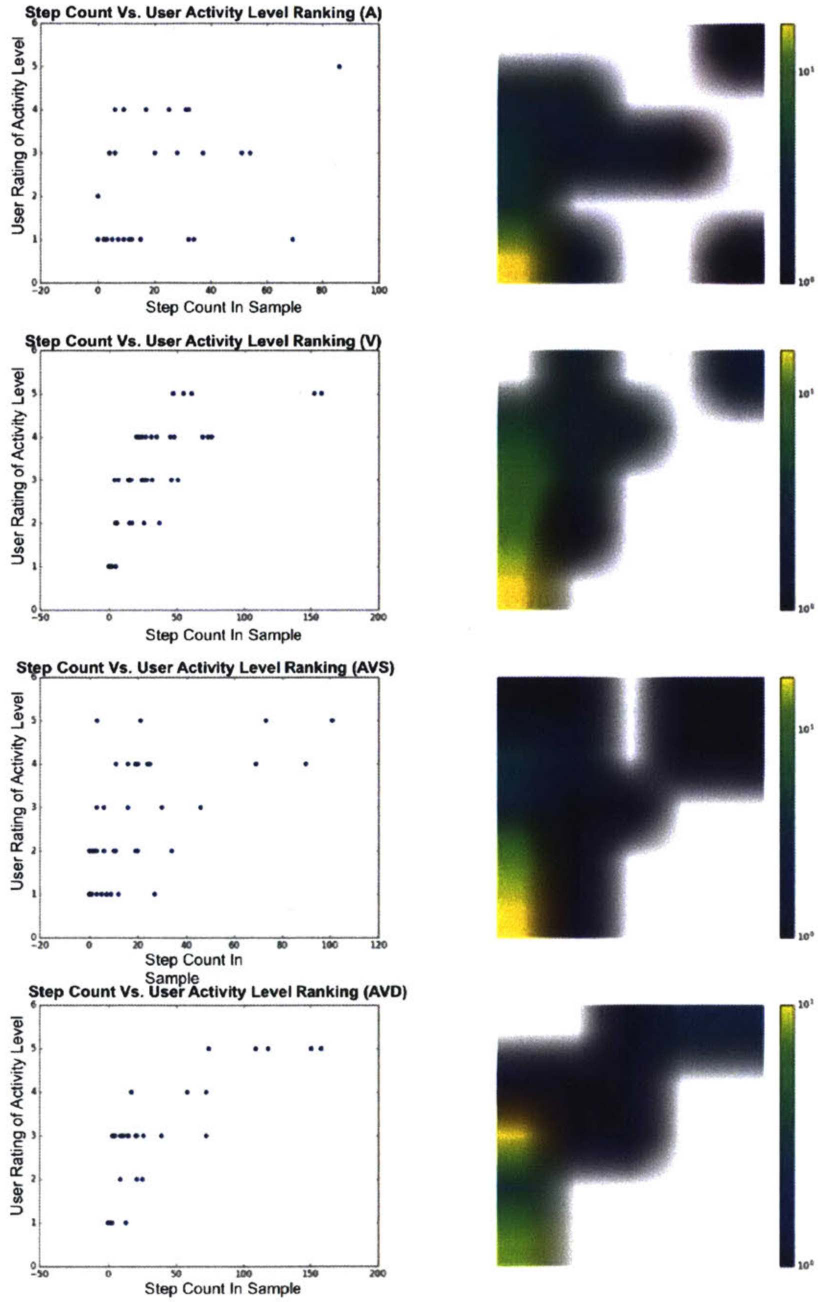



Figure 7-7: Scatter plot and heat map showing step count (x-axis) plotted with respect to user ranking of activity level(y-axis) for modes 'A' (top) 'V' (second) 'AVS' (third) and 'AVD' (bottom)

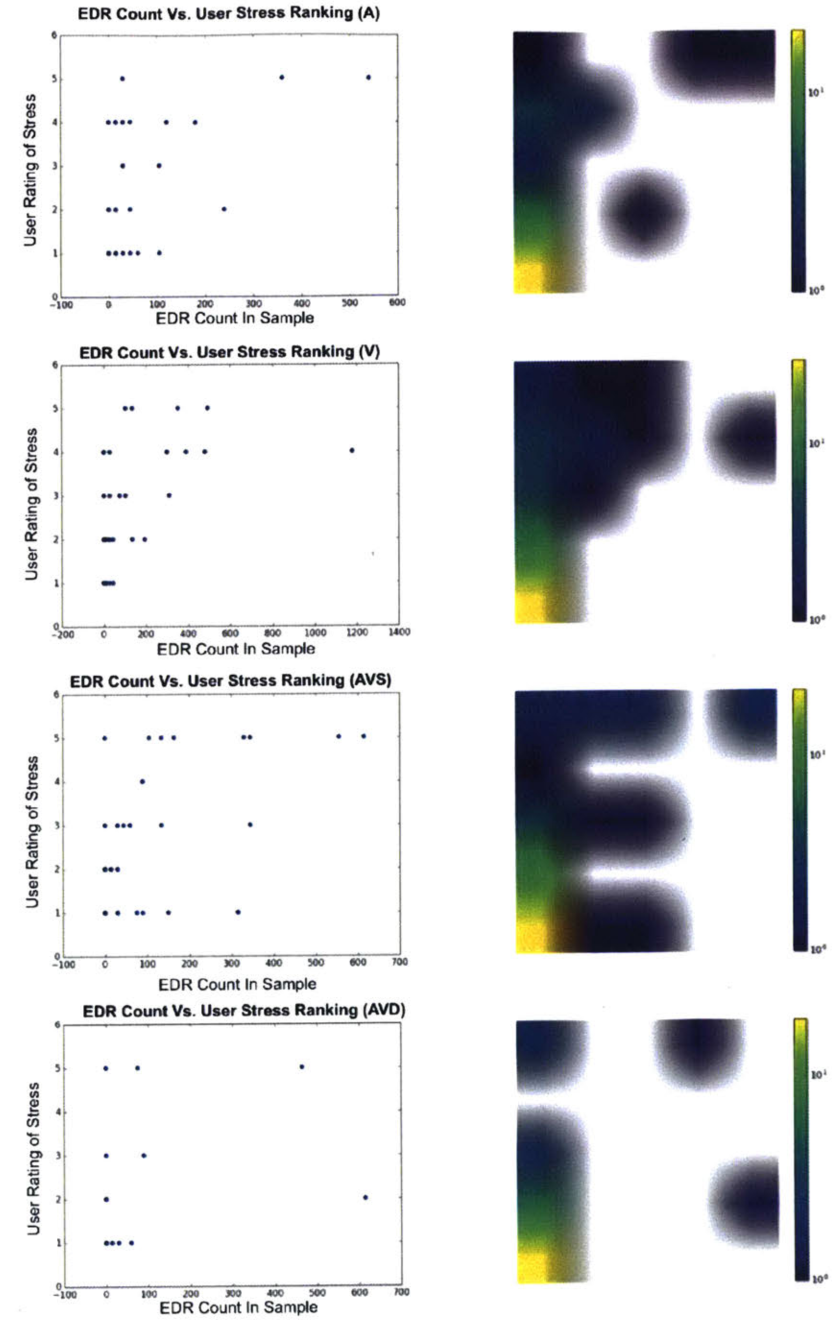



Figure 7-8: Scatter plots and heat maps showing EDR count (x-axis) plotted with respect to user ranking of stress level(y-axis) for modes 'A' (top) 'V' (second) 'AVS' (third) and 'AVD' (bottom)

# Chapter 8

# Concluding Thoughts

## 8.1 Future Thoughts: Rotator4D

I end this thesis by stepping very far back and thinking as abstractly as possible about how the Rotator tool can grow.

Human perception is limited. In particular, our Newtonian physics engines are genetically designed, or perhaps trained through real-world exposure at a young age, to operate in three dimensions, with little intuition for higher dimensional spaces. Evidence over the last century that the universe in which we live is itself a higher dimensional structure suggests that our three-dimensional perception may be a matter of biological convenience. Rotator4D would probe whether we can train and manipulate the human senses to perceive higher dimensional structures by strategically encoding overflow spatial dimensions into sound. For instance, a 4-dimensional dataset could be encoded as an object in a 4-dimensional audio-visual space. User-defined rotations would adjust both the sonic and visual mappings. Rotator4D would be an evolution of the original Rotator application. In both cases, the goal of the tool is to extend our attentive capacities. In the original Rotator application, we aimed to do so by encoding independent linear data streams into the auditory and visual spaces, whereas Rotator4D would focus on geometrically high-dimensional mappings. The original Rotator tool may be immediately useful for data monitoring and analysis scenarios previously addressed. Rotator4D would be more experimental and less tied to particular applications.

A tool like Rotator4D would raise questions about whether we can extend our capacity for spatial reasoning. Could our neural machinery learn to interpret a fourth spatial dimension encoded in an auditory space, or is our neural machinery inherently optimized to encode the 3-dimensional reality that we experience? Biologically speaking, this question amounts to whether our spatial encoding mechanisms are based on underlying geometric rules and operations, like distance measurement and rotations, or whether they are specifically based on development of a 3-dimensional topological map. Anflalo and Graziano tested the theory that we are capable of four-dimensional spatial intuition by developing a visual maze interface with a fourth direction [25]. At any stage, the user can perform rotations with respect to any axis in the four-dimensional space to reorient their view in the maze. Four dimensional topology is represented using a carefully described color-coding and visualization technique, and the map's topology is restricted to straight lines and orthonormal segments. Using the widely accepted premise that path integration (the ability to mentally sum the lengths and turns of a path through space), is a reliable marker of spatial acuity, users were asked to complete this four-dimensional maze and then, to the best of their abilities, point a vector back to the starting point of the maze. They were given feedback after each trial on the error in their approximated angle. Their ability to perform this path integration task accurately over many trials fit a two-phase transition: after 20 attempts, users performed as well as a bot that partially calculates the appropriate angle using 3D spatial reasoning.

After many more trials, however, users achieved a performance accuracy that exceeds optimal performance in a 4D maze completed using only 3-D ability. The results provide evidence that our underlying spatial processing machinery is capable of learning path integration in four dimensions. A handful of other attempts have been made to measure human ability to intuit four-dimensional space, sometimes by allowing the user to move along one axis and experience sequential 3-dimensional cross sections of the 4-dimensional shape [27].

Using the knowledge that our neural system may be equipped to develop 4-D spatial intuition, could an auditory signal replace one of the spatial dimensions? How could this extra dimension be rendered? Approaches to consider include spatialization, mappings to the timbre/frequency domain, different instruments/notes, sequential patterns, etc. How can this auditory mapping be coupled into visual perception so additional auditory dimensionality is intrinsically derived from what is seen? Does the human brain have sufficient integration between visual and auditory processing to even make this possible? Evidence of vision-impaired individuals gaining some level of learned 'visualization' from the ambiance heard from a series of vocalized clicks points to intriguing possibilities here [69], such as realized in other mammals like bats [87] and dolphins [58].

## 8.2 Concluding Statements

This thesis has summarized two sonification tools. Firstly, the Quantizer project was summarized, which is a tool aimed at real-time, data-driven artistic expression. The Quantizer project's associated website reached a large audience and is the first tool to expose real-time sonification of high-energy-physics data to the public. Secondly, the Rotator project was introduced, which is aimed at studying how moving some data into ones auditory periphery impacts perception of a dataset. Initial studies using the Rotator tool suggest that users may be comfortable analyzing a high-dimensional dataset even if most of the data is presented auditorily and they are only allowed to look at a single dimension of a data at a time. Finally, a few possible application areas for the Rotator tool were considered.

A large variety of methods can be used for presenting and exploring data sonically and visually, and optimal presentation modes are probably task/goal dependent. We have considered only a few possible areas here. It is also possible that faster, real-time integration of data into the tool could expand its potential influence. Audio samples described in this thesis will be available at: `http://resenv.media.mit.edu/#Projects#rotator`

# Appendix A

# Unused Plots from User Study

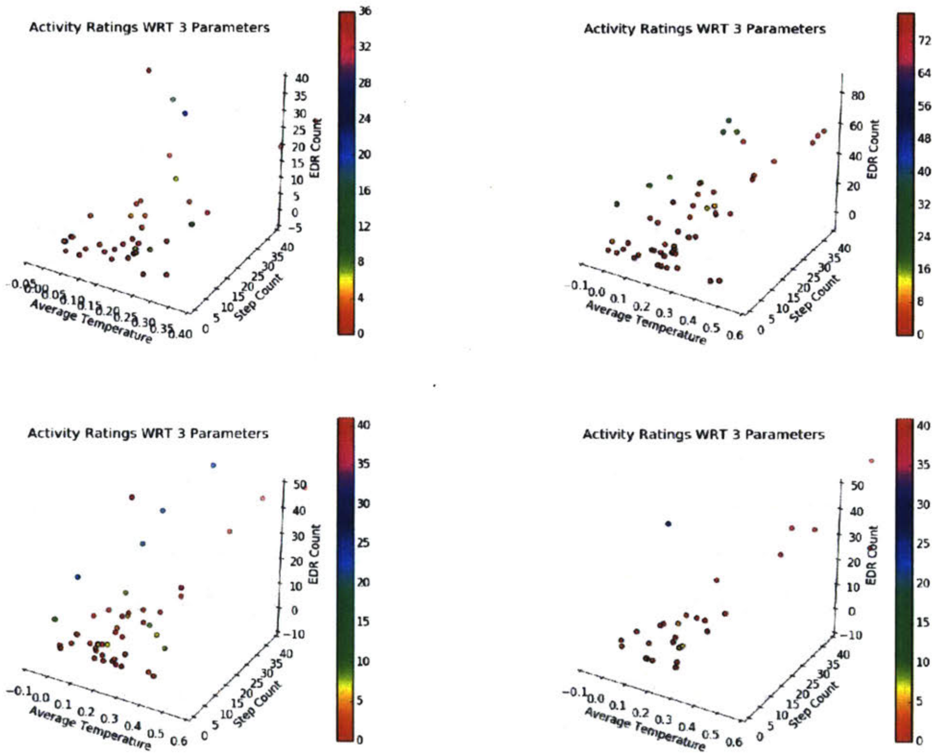



Figure A-1: User rating of activity level WRT average temperature, step count, and EDR count for modes 'A' (upper left) 'V' (upper right) 'AVS' (lower left) and 'AVD' (lower right). Higher values on the color bar represent a higher activity rating. Although this plot most accurately reflects how relevant parameters correlate to user ranking, no conclusions were drawn since there are no clear trends

# Bibliography


[4] **Granted temperature data samples by Prof. Paolo Zuccon (data taken between May 2014 - September 2014).**

[5] Audio-resampler: Simple web audio resampling library. Retrieved August 19th, 2016 from https://www.npmjs.com/package/audio-resampler.

[6] d3. Retrieved August 19th, 2016 from https://d3js.org/.

[7] Flux. Retrieved August 19th, 2016 from https://facebook.github.io/flux/.

[8] Gevent. Retrieved August 18th, 2016 from `http://www.gevent.org/`.

[9] http://conversations.e-flux.com/t/whats-art-got-to-do-with-science/1489.

[10] http://lhc-collimation-project.web.cern.ch/lhc-collimation-project/sounds2011/.

[11] jsqubits: Javascript quantum computer simulator. Retrieved August 19th, 2016 from http://davidbkemp.github.io/jsqubits/.

[12] React. Retrieved August 19th, 2016 from https://facebook.github.io/react/.

[13] react-d3 brush implementation. Retrieved August 19th, 2016 from https://github.com/react-d3/react-d3-brush.

[14] Root data analysis framework. Retrieved August 19th, 2016 from https://root.cern.ch/.

[15] Snapshot study. Granted data sample by Natasha Jaques and Sara Taylor. Study summary is available at http://snapshot.media.mit.edu/.

[16] Websocket protocol. Retrieved August 18th, 2016 from `http://www.websocket.org/`.

[17] NASA - Voyager Space Sounds, 1992. Retrieved August 24th, 2016 from `https://www.discogs.com/label/952354-NASA-Voyager-Space-Sounds`.

[18] Ableton live, 2014. Retrieved January 11th, 2016 from http://ableton.com/.

[19] Icecast, 2015. Retrieved January 11th, 2016 from icecast.org.

[20] Pure data, 2015. Retrieved January 11th, 2016 from http://puredata.info.

[21] Cycling '74 Max MSP, 2016. Retrieved January 11th, 2016 from https://cycling74.com/products/max/.

[22] Jack audio connection kit, 2016. Retrieved January 11th, 2016 from http://www.jackaudio.org/.

[23] The zone, 2016. Retrieved January 11th, 2016 from http://cjzn.streamon.fm/.

[24] Georges Aad, Brad Abbott, Jalal Abdallah, S Abdel Khalek, O Abdinov, Rosemarie Aben, Babak Abi, Maris Abolins, OS AbouZeid, Halina Abramowicz, et al. Measurements of the w production cross sections in association with jets with the atlas detector. *The European Physical Journal C*, 75(2):1–46, 2015.

[25] TN Aflalo and MSA Graziano. Four-dimensional spatial reasoning in humans. *Journal of Experimental Psychology: Human Perception and Performance*, 34(5):1066, 2008.

[26] Felix Almonte, Viktor K Jirsa, Edward W Large, and Betty Tuller. Integration and segregation in auditory streaming. *Physica D: Nonlinear Phenomena*, 212(1):137–159, 2005.

[27] Michael S Ambinder, Ranxiao Frances Wang, James A Crowell, George K Francis, and Peter Brinkmann. Human four-dimensional spatial intuition in virtual reality. *Psychonomic bulletin & review*, 16(5):818–823, 2009.

[28] Arts@CERN. Bill Fontana. 2013. Retrieved January 19th, 2016 from http://arts.web.cern.ch/bill-fontana.

[29] Lily Asquith. LHCSound: Sonification of the ATLAS Detector Data, 2011. Retrieved January 7, 2016 from http://lhcsound.wordpress.com.

[30] Kat Austen. Capturing the sound of light. 2012. Retrieved January 19th, 2016 from https://www.newscientist.com/blogs/culturelab/2012/09/capturing-the-sound-of-light.html.

[31] Stephen Barrass. Personify: a toolkit for perceptually meaningful sonification. In *ACMA*. Citeseer, 1995.

[32] Stephen Barrass and Virginia Best. Stream-based sonification diagrams. 2008.

[33] Oded Ben-Tal, Jonathan Berger, Bryan Cook, Michelle Daniels, and Gary Scavone. Sonart: The sonification application research toolbox. 2002.

[34] Virginia Best, Frederick J Gallun, Simon Carlile, and Barbara G Shinn-Cunningham. Binaural interference and auditory grouping. *The Journal of the Acoustical Society of America*, 121(2):1070–1076, 2007.

[35] Virginia Best, Andre van Schaik, and Simon Carlile. Two-point discrimination in auditory displays. 2003.

[36] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound.* MIT press, 1994.

[37] MALCOLM W Browne. Physicists discover another unifying force: Doo-wop. *The New York Times*, pages 4–6, 2008.

[38] Peter Cariani and Christophe Micheyl. Toward a theory of information processing in auditory cortex. In *The Human Auditory Cortex*, pages 351–390. Springer, 2012.

[39] Marija Cauchi, O Aberle, RW Assmann, A Bertarelli, F Carra, K Cornelis, A Dallocchio, D Deboy, L Lari, S Redaelli, et al. High energy beam impact tests on a lhc tertiary collimator at the cern high-radiation to materials facility. *Physical Review Special Topics-Accelerators and Beams*, 17(2):021004, 2014.

[40] Panos Charitos. A Cosmic Piano for ALICE, 2013. `http://alicematters.web.cern.ch/?q=ALICEcosmicpiano`.

[41] Juliana Cherston, Ewan Hill, Steven Goldfarb, and Joseph A. Paradiso. Musician and mega-machine: Compositions driven by real-time particle collision data from the atlas detector. *Proceedings of the international conference on new interfaces for musical expression, Brisbane, Australia*, 2016.

[42] Juliana Cherston, Ewan Hill, Steven Goldfarb, and Joseph A Paradiso. Sonification platform for interaction with real-time particle collision data from the atlas detector. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1647–1653. ACM, 2016.

[43] Mary Czerwinski, Nancy Lightfoot, and Richard M Shiffrin. Automatization and training in visual search. *The American journal of psychology*, pages 271–315, 1992.

[44] Alberto de Campo, Christopher Frauenberger, Robert Höldrich, Thomas Melde, Willibald Plessas, and Bianka Sengl. Sonification of quantum spectra. In *Proceedings of ICAD*, pages 6–9, 2005.

[45] Silvano de Gennaro. Les horribles cernettes-" goodbye cern" concert, 2012.

[46] Antonella Del Rosso. Higgs at 3.5 seconds into the melody, 2012. `http://cds.cern.ch/journal/CERNBulletin/2012/28/News%20Articles/1460881`.

[47] Rob Dimeo. Simple quantum visualizations using idl. *Journal of Physics*, 35:177, 1967.

[48] Margaret Evans. Music of the (data) sphere. 2016. Retrieved August 18th, 2016 from http://news.mit.edu/2016/media-lab-quantizer-particle-collisions-science-art-0628.

[49] Jon Fingas. Make music with the large hadron collider through a web app. 2016. Retrieved August 18th, 2016 from https://www.engadget.com/2016/05/30/large-hadron-collider-music-making.

[50] Michael Franco. This is what it sounds like when protons collide. 2016. Retrieved August 18th, 2016 from http://newatlas.com/quantizer/43568/.

[51] Morris Rubinoff Franz L. Alt. *Advances in computers*, volume 6. Academic Press, 1966.

[52] Ariane Garon et al. *On a new visualization tool for quantum systems and on a time-optimal control problem for quantum gates.* PhD thesis, München, Technische Universität München, Diss., 2014, 2014.

[53] Marek Gasior and Rhodri Jones. The principle and first results of betatron tune measurement by direct diode detection. Technical report, 2005.

[54] Iulia Georgescu and Federico Levi. Exhibition: Patterns in the dark. *Nature Physics*, 11(7):522–522, 2015.

[55] Evans M Harrell and Bernd Thaller. Visual quantum mechanics, selected topics with computer-generated animations of quantum-mechanical phenomena, 2001.

[56] Sandra G Hart and Lowell E Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Advances in psychology*, 52:139–183, 1988.

[57] Gary Haussmann. react-webaudio. Retrieved August 19th, 2016 from https://github.com/Izzimach/react-webaudio.

[58] David A Helweg, Whitlow WL Au, Herbert L Roitblat, and Paul E Nachtigall. Acoustic basis for recognition of aspect-dependent three-dimensional targets by an echolocating bottlenose dolphin. *The Journal of the Acoustical Society of America*, 99(4):2409–2420, 1996.

[59] Thomas Hermann, Andy Hunt, and John G Neuhoff. *The Sonification Handbook.* Logos Verlag, Berlin,GE, 2011.

[60] Matilda Heron. Physics and music collide at the Montreux Jazz Festival, 2015. Retrieved January 7, 2016 from `http://home.cern/about/updates/2015/07/physics-and-music-collide-montreux-jazz-festival`.

[61] Valentin Heun, Anette von Kapri, and Pattie Maes. Perifoveal display: combining foveal and peripheral vision in one visualization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 1150–1155. ACM, 2012.

[62] Eli Hougland. Charles dodge- earth's magnetic field and bell laboratories, 2014.

[63] Ryoji Ikeda. datamatics, 2006.

[64] Natasha Jaques, Sara Taylor, Akane Sano, and Rosalind Picard. Multi-task, multi-kernel learning for estimating individual wellbeing. In *Proc. NIPS Workshop on Multimodal Machine Learning, Montreal, Quebec*, 2015.

[65] JR Johansson, PD Nation, and Franco Nori. Qutip 2: A python framework for the dynamics of open quantum systems. *Computer Physics Communications*, 184(4):1234–1240, 2013.

[66] Douglas Kahn. *Earth sound Earth signal: Energies and Earth magnitude in the arts.* Univ of California Press, 2013. also see `http://www.auroralchorus.com/`.

[67] David Kaiser. Physics and feynman's diagrams. 93, 2005.

[68] Alexis Kirke, Peter Shadbolt, Alex Neville, Aurelien Antoine, and Eduardo R Miranda. Qmuse: A quantum computer music system designed for a performance for orchestra, electronics and live internet-connected photonic quantum computer.

[69] Daniel Kish. How i use sonar to navigate the world, March 2015. Retrieved August 24th, 2016 from `https://www.ted.com/talks/daniel_kish_how_i_use_sonar_to_navigate_the_world?language=en`.

[70] Alexandros Kontogeorgakopoulos and Daniel Burgarth. Sonification of controlled quantum dynamics. International Computer Music Conference 2014, 2014.

[71] C Lavor, LRU Manssur, and R Portugal. Shor's algorithm for factoring large integers. *arXiv preprint quant-ph/0303175*, 2003.

[72] Danny Lewis. Tune into the smashing sounds of large hadron collider data in real time. 2016. Retrieved August 18th, 2016 from http://www.smithsonianmag.com/smart-news/tune-smashing-sound-large-hadron-collider-data-real-time-180959240/.

[73] Ryan F. Mandelbaum. This is the sound of particles smashing together. 2016. Retrieved August 18th, 2016 from http://www.popsci.com/you-can-now-live-stream-lhc-data-live-as-music.

[74] Eduardo Reck Miranda. Music: The music of particle collisions. *Nature Physics*, 12(8):721–721, 2016.

[75] Daniel Nothen. BUTT - broadcast using this tool, 2014. Retrieved January 11th, 2016 from `http://butt.sourceforge.net/`.

[76] Cian O'Luanaigh. Japanese artist ryoji ikeda wins residency at cern, 2014.

[77] OSA Conservation. OSA Ears, 2014. Retrieved January 19th, 2016 from `http://www.osa-ears.org/`.

[78] Iulius AT Popa, Jeffrey E Boyd, and David Eagle. Muse: a music-making sandbox environment for real-time collaborative play. *perspective*, 3:4, 2015.

[79] Nicola Quadri. Make music with atlas data. 2016. Retrieved August 18th, 2016 from https://home.cern/about/updates/2016/05/make-music-atlas-data.

[80] Responsive Environments Group. Tidmarsh living observatory, 2015. `http://tidmarsh.media.mit.edu`.

[81] Michael Rhoades. Hadronized Spectra: The LHC Sonifications. 2013. Retrieved January 24th 2016 from `http://econtact.ca/16_3/rhoades_LHCsonifications.html`.

[82] Agnieszka Roginska, Edward Childs, and Micah K Johnson. Monitoring real-time data: A sonification approach. 2006.

[83] Armin Ronacher. Flask, 2014. Retrieved January 19th, 2016 from `http://flask.pocoo.org/`.

[84] Anna Saranti, Gerhard Eckel, and David Pirró. Quantum harmonic oscillator sonification. In *Auditory Display*, pages 184–201. Springer, 2010.

[85] Johannes Schmidt-Ehrenberg and Hans-Christian Hege. *Visualizing quantum mechanical phenomena*. ZIB, 1999.

[86] Hong Jun Song and Kirsty Beilharz. Concurrent auditory stream discrimination in auditory graphing. *Journal of Computers*, 3:79–87, 2007.

[87] Nobuo Suga. Biosonar and neural computation in bats. *Scientific American*, 262(6):60–68, 1990.

[88] Niccolò Tempini and Sabina Leonelli. Exhibition: Live by data, die by data. *Nature Physics*, 12(2):109–110, 2016.

[89] Fiorella Terenzi. *Music From the Galaxies*. Island, 1991.

[90] The ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider. *J. Instrum.*, 3:S08003. 437 p, 2008. Also published by CERN Geneva in 2010.

[91] Samuel Ting. Alpha magnetic spectrometer. In *39th COSPAR Scientific Assembly*, volume 39, page 1975, 2012.

[92] J Jay Todd, Daryl Fougnie, and René Marois. Visual short-term memory load suppresses temporo-parietal junction activity and induces inattentional blindness. *Psychological Science*, 16(12):965–972, 2005.

[93] Akito van Troyer. Constellation: A tool for creative dialog between audience and composer. Retrieved January 19th, 2016 from `http://vantroyer.com/lib/pdf/Constellation_Akito_van_Troyer2.pdf`.

[94] Katharina Vogt, Robert Höldrich, David Pirro, Martin Rumori, Stefan Rossegger, Werner Riegler, and Matevz Tadel. A sonic time projection chamber: Sonified particle detection at cern. *Proceedings of the International Conference on Auditory Display*, 2010.

[95] Bruce N Walker and Joshua T Cothran. Sonification sandbox: A graphical toolkit for auditory graphs. 2003.

[96] Hendrik Weimer. Listen to quantum computer music, 2010. Retrieved August 21st, 2016 at http://www.quantenblog.net/physics/quantum-computer-music.