Musician and Mega-Machine: Compositions Driven by Real-Time Particle Collision Data from the ATLAS Detector

Juliana Cherston
Responsive Environments Group
MIT Media Lab
75 Amherst Street
Cambridge, MA 02139, USA
cherston@media.mit.edu

Ewan Hill
Dept. of Physics and Astronomy
University of Victoria
Victoria, BC, Canada
TRIUMF
Vancouver, BC, Canada
e4hill@uvic.ca

Steven Goldfarb
Department of Physics
University of Michigan
Ann Arbor, MI 49201, USA
steven.goldfarb@cern.ch

Joseph A. Paradiso
Responsive Environments Group
MIT Media Lab
75 Amherst Street
Cambridge, MA 02139, USA
joep@media.mit.edu

ABSTRACT

We present a sonification platform for generating audio driven by real-time particle collision data from the ATLAS experiment at CERN. This paper provides a description of the data-to-audio mapping interfaces supported by the project's composition tool as well as a preliminary evaluation of the platform's evolution to meet the aesthetic needs of vastly distinct musical styles and presentation venues. Our work has been conducted in collaboration with the ATLAS Outreach team and is part of a broad vision to better harness real-time sensor data as a canvas for artistic expression. Data-driven streaming audio can be treated as a reimagined form of live radio for which composers craft the instruments but real-time particle collisions pluck the strings.

Author Keywords

Sonification, Real-Time, Auditory Display

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing

1. INTRODUCTION AND RELATED WORK

In 2015, the Large Hadron Collider (LHC) at CERN began smashing protons at a centre of mass energy of 13 TeV, the highest energy ever achieved by a particle collider. The ATLAS experiment at the LHC records the particle collision data to probe our understanding of particle physics at these unexplored energies. We present a musical tool that enables composers to use real-time particle collision data from ATLAS to build aesthetically diverse audio streams, thereby allowing the public to listen to live collision events interpreted through different artistic lenses.

NIME'16, July 11-15, 2016, Griffith University, Brisbane, Australia.

This project builds upon a considerable number of sound-based works inspired by physics research that integrate data to varying degrees. Some of the pioneering work in scientific data emerged from large scientific research labs. In 1970, composers working with Bell Laboratories composed music using measurements of Earth's magnetic field [13], and in the 1990s, sound artists used planetary data collected by NASA to produce 'Symphonies of the Planets' [1]. Also in the early 1990s, a parody pop group of CERN employees known as Les Horribles Cernettes sang physics-inspired music. Their photo was the first to be featured on the newly formed World Wide Web [9]. More recently, Ryoji Ikeda has built audio-visual soundscapes based on physics concepts learned during a year-long artistic residency at CERN [14].

Some projects have made more direct use of particle collision data to drive artistic visions. Examples include a project using simulated data from the ALICE experiment at CERN to map particle trajectories through the detector's time projection chamber to spatialized audio [25], and a project sonifying a now famous plot showing evidence for the Higgs boson [6, 12]. One project used quantities drawn from ATLAS data to drive parametric variables in a custom Csound instrument [20]. Rather than use experimental data itself, Bill Fontana, winner of the Collide@CERN prize, produced sound art using audio directly recorded from components of the collider including magnets and cooling systems [5]. In a similar spirit, Jo Thomas has incorporated audio produced from The Diamond Light Source synchrotron into a sound art installation [7].

Real-time particle data has also 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. Real-time audio feeds from microphones placed in natural environments are also relatively common [18]. However, only recently have projects worked to continuously convert real-time, large-scale sensor data to sound. For example, the cosmic piano sonifies incoming cosmic rays in real-time for live performance settings [10], and 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 [19]. Broadly, aesthetic considerations have taken on growing importance in the field of data sonification [8].

Here we present a platform that not only exposes real-time particle collision data as a canvas for artists but also live-broadcasts the audio online. The online broadcast is much like a reimagined online radio in which composers build the instruments but sonified real-time sensor data streams drive the musical content. While composers working with the platform have thus far focused predominantly on audio aesthetics, there is an opportunity for both composers and listeners to learn more about predictable patterns in the data if these patterns are appropriately highlighted by the composer. The composition engine consists of two main interfaces. Physics enthusiasts can use a Python-based interface to integrate some basic scientific data analysis tools and equations, and composers can use an Open Sound Control (OSC) interface to sonify data that has already been extracted from the collision event file. This paper will describe the mapping schemes available to the composer as well as the architecture of the live audio streaming system. The platform's evolution to meet the needs of distinct composition styles and presentation venues is also addressed.

2. THE LHC AND ATLAS

The Large Hadron Collider at CERN is a machine used to accelerate particles (usually protons) up to nearly the speed of light. The particles collide at the centre of massive particle detectors that are responsible for measuring the properties of particles produced in the collisions.

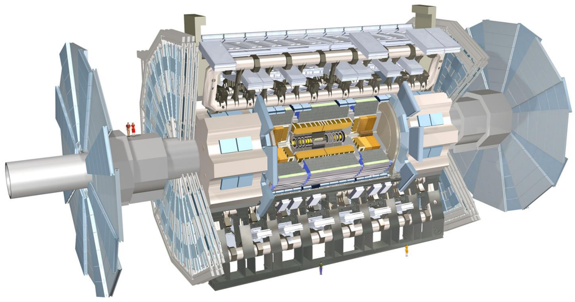

Figure 1: Illustration of ATLAS Detector [22].

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 deepest questions of nature: "What is the nature of dark matter?", "Are there extra physical dimensions?", "What is the origin of mass?", and "Are there any deeper symmetries that govern the laws of our universe?". The ATLAS collaboration is made up of over 3000 scientists from 38 countries.

The ATLAS 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 [22].

The inner detector measures the trajectories of charged particles. 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. 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 is used to measure the trajectories and momenta of muons with the aid of a system of toroidal magnets.

ATLAS 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, ϕ) are used in the transverse plane, ϕ being the azimuthal angle around the beam pipe [22]. The polar angle, θ, is defined from the +z axis.[1]

The LHC collides particles at a nominal rate of 40 MHz [22], which corresponds to a shortest time between collisions of 25 nanoseconds. A "triggering system" is used to filter out the common events from the rare events and only saves a fraction of the events to disk for later analysis at a rate of the order of kHz.

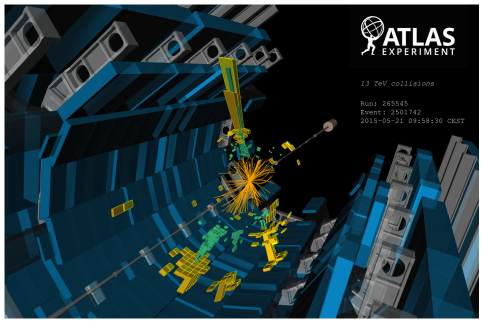

Figure 2: Sample particle collision event display produced by the ATLAS experiment [23].

3. SYSTEM ARCHITECTURE

3.1 Overview

Our sonification platform interfaces with partially reconstructed real-time data from the ATLAS detector to produce live audio streams. The project consists of two parts: a composition engine and a website featuring real-time audio. The composition engine is built using Python and the Open Sound Control networking protocol, and a flow chart showing the steps used to convert data to audio is provided in Figure 3. The project's associated website is built using Flask [21], Icecast [3], and Flask-SocketIO [15].

3.2 Event Stream

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 is accessed with permission from the ATLAS experiment. Use of the live data stream is restricted to compositions supported by the project's associated website, but a special set of simulated events and data for education purposes is available to all composers.

The event stream contains low-level event data (such as calorimeter energy deposits) and only partially reconstructed data (such as tracks) rather than full ATLAS reconstructions of particles (such as electrons and muons). While fully reconstructed particle information could also prove interesting to sonify, reconstructions are likely to introduce additional delays to a real-time sonification process.

[1]The code actually uses a quantity called "pseudorapidity" instead of the polar angle. The pseudorapidity is defined as $\eta = -\ln \tan(\theta/2)$ and is used by ATLAS because there is roughly an even distribution of activity per unit of pseudorapidity [17].

3.3 Data-to-Audio Conversion

Collision events are queued for sonification, and eventually a set of data parameters are extracted from the data files using composer-defined selection rules and streamed as a multichannel time series of Open Sound Control messages appropriate for controlling audio synthesizers.

The default data streamed as OSC messages are:

- **Liquid Argon (LAr) EM Calorimeter:** This detector measures the energy deposited by particles that interact primarily electromagnetically. The positions and magnitudes of the energy deposits are streamed as OSC messages.
- **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 as OSC messages.
- **Particle Tracks:** The inner detector is used to reconstruct charged particles as tracks. The direction of the track trajectory and the track momentum are streamed as OSC messages.
- **Resistive Plate Chamber (RPC):** 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 as OSC messages.

Energy, momentum, and direction are fundamental quantities used in particle physics. This fact motivated our choice of the quantities to sonify.

3.4 Web Stream

The generated audio can be routed to an Icecast streaming server using Jack Audio [4] and BUTT [16], and is embedded in a dedicated website using HTML5 audio. The website also provides some additional information about each collision and is updated in real-time using SocketIO. This information includes a new collision event notification, a graphical display of the collision produced by ATLAS, and some basic plots of the data.

4. MAPPING INTERFACES

4.1 Overview

In order to compose audio, users must interact with two interfaces: a Python interface that controls the set of selection requirements for the data, and an OSC interface that controls the audio that the resulting time series of OSC data will produce. Each interface comes with default settings and customization options that accommodate different levels of proficiency with the tools. Physics enthusiasts can choose to write their own custom code in the Python data parsing engine and composers can choose to integrate custom-built audio synthesis patches driven by incoming OSC messages. The four possible combinations of the Python and OSC interface usages are:

- **default Python + default OSC:** For users who know nothing about the data nor composing music.
- **default Python + customized OSC:** Ideal for users comfortable building compatible synthesizers but who do not know anything about the data.
- **custom Python + default OSC:** Can be used by physics enthusiasts (e.g. ATLAS physicists) looking to interact more closely with the data but who do not know how to build OSC-compatible software to compose music.
- **custom Python + custom OSC:** For the rare users who understand the data and how to program with OSC messages.

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 1: Sampling of Command line Arguments

4.2 Default Python Interface

The default Python interface filters data in accordance with basic approaches used by ATLAS physicists to isolate useful information for an analysis.

ATLAS searches for undiscovered particles often require hunting for rare events ("signal") in a larger collection of other events ("background"). When performing these 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 default 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.

Due to the complicated geometry of the detector, ATLAS physicists often place selections on the trajectories of the particles in the event to ensure that they were accurately measured. 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 $0.18094 < \theta < \pi - 0.18094$.

It should be noted that the default Python interface does not discard full events on the basis of their content because the rate of real-time input event files is very low. However, a physics analysis seeks to study specific processes and will select and discard whole events. For this reason amongst others, the default Python interface differs from a standard physics analysis.

The default OSC interface uses pre-determined data range cuts to linearly scale the measured data values into a MIDI-note range that is defined by the composer. Finally, transformations are applied in order to produce a time series of data driven by the detector's geometry and in accordance with the user's musical settings. Table 1 provides a summary of built-in command line settings.

In all sonifications of ATLAS data the event durations are stretched because the time for the particles to propagate through the detector is only a few tens of nanoseconds and the detector signals themselves may last only up to a few microseconds. For the purposes of streaming audio live on the project's associated website, the event durations are forced to be approximately 25 seconds, which matches the approximate time between incoming events into the sonification platform. A more in-depth evaluation of the real-time nature of the platform can be found in [11].

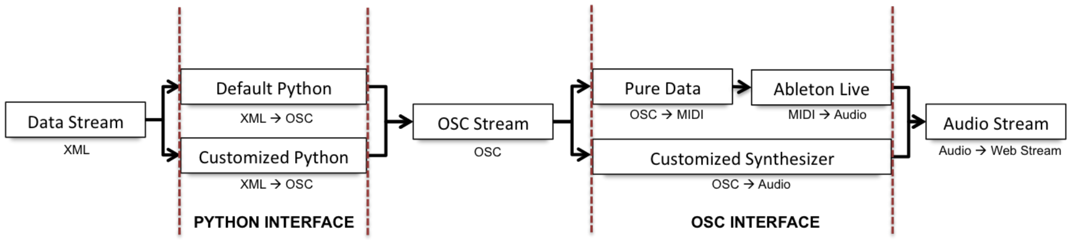

Figure 3: Chart depicting flow of data through platform including the three streams (data, OSC, audio) and two interfaces (Python, OSC) referenced throughout the paper. Software and data formats are defined for each stage.

4.3 Default OSC Interface

The default OSC interface is a simple Pure Data (PD) graphical interface (also referred to as the "PD GUI") that enables users with less experience in audio synthesis to quickly and easily compose audio that uses a musical beat structure. The GUI has also been built for real-time DJ'ing to data. Note that more experienced users who wish to produce free-form, experimental audio are expected to develop custom interfaces in place of the default interface described in this section. See section 5.2 for some examples.

The default GUI maps incoming OSC messages to MIDI notes and is compatible with commercially available sound synthesis tools like Ableton Live [2]. More advanced users might also choose to build automated DJ'ing patches to control this GUI.

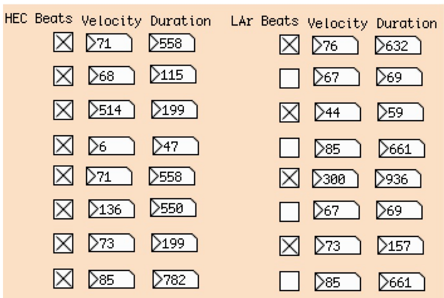

Figure 4: Extract from the default PD graphical interface for real-time DJ'ing and for users with less experience in audio synthesis. The eight toggle boxes per calorimeter allow the user to select on which beats the audio will be played. The settings depicted show one calorimeter playing audio every beat while the other only plays every other beat.

The interface exposes some simple controls to the user, including setting the MIDI ranges of each OSC stream, shifting the musical octaves, and toggling on and off individual OSC streams. The interface can also be used to map the data to a selected musical scale, as well as to adjust the musical tempo, note durations, and note velocities. Each of these features can be modified in real-time while DJ'ing.

To use the GUI, composers stream data at discretized time intervals using the Python "uniform" command line argument. For certain OSC streams where there are large amounts of data available, the user can create rhythms by defining which of the music beats are muted. Currently, 8 music notes or beats are played before the software will repeat the same rhythm. This control is operated by toggling on and off the 8 possible beats, as shown in Figure 4.

4.4 Default Interfaces and Target Audience

The ATLAS data are governed by underlying statistical distributions that are not obvious at first glance. Part of the beauty in this project is the challenge to the composer to learn about the data from various sources and then utilise known distributions in the data to form the patterns in the music. The default interfaces are designed for use by the general public and already incorporate a few basic underlying patterns in the data to make qualitatively perceptible patterns in the music. For example, the probability distribution of the track momenta can be made to produce the root note of a scale more frequently, which can be beneficial in traditional music styles where the music should often return to that root note. In the default interface, quantities with smooth, non-trivial distributions, like track momenta, are used to drive the MIDI notes. Other variables are predominantly used by the default interface for rejecting data.

5. SYSTEM USAGE AND STUDIES

A core goal for this project has been to create a flexible sonification platform that can be used to develop many different musical styles. This goal motivated a partial separation of data tools and music tools into our two interfaces to accommodate users with varying degrees of knowledge of music composition. Additionally, common software was chosen so that a larger number of composers could start using the platform with relative ease. Early on, we conceived of a wide array of possibilities for how and where the audio would be output (ranging from live DJ'ing to real-time web-streaming) and we designed the system accordingly. This section summarizes how we have built or modified the system to support diverse use cases.

5.1 Summary of Events

Three events were held that enabled us to further study and improve our system in various usage scenarios.

- **Composition Workshop:** July, 2015. A workshop was conducted for roughly 20 composers at the International Conference on Auditory Display (ICAD). This workshop was held to test the usability of the composition engine. Most workshop attendees were experienced working in Pure Data with minimal physics knowledge and preferred to use the default Python interface with their own custom OSC interfaces [11].

- **Performance at Montreux Jazz Festival:** July, 2015. A Jazz pianist improvised alongside audio generated from ATLAS data using this platform. The default Python interface was used alongside the PD GUI to "DJ" the data by adjusting the parameter mapping, the tempo, and the subset of data streamed.

- **Website Alpha Launch:** November, 2015. Access to the website was granted to roughly 50 users. Three real-time audio streams generated from proton collisions were featured. Feedback was collected and incorporated in preparation for the website's public launch.

These three events helped to guide us in building additional flexibility into the system. In particular, we added a number of different streaming modes to the Python interface and added a number of new variables to the platform.

Many workshop participants wished to bypass the timing strategies in the Python interface but still make use of the data parsing and processing. As a result, a new streaming mode ("send all", see Table 1) and an associated PD patch that accepts arrays of data is now provided for composers who wish to program their own timing information.

The improv performance demonstrated that the system can be used in a live setting. The PD GUI responded quickly enough to be used for DJ'ing the events in real-time and allowed the user to interact with performers on stage: the pianist could react to the music from our system and the user of our system could react to the pianist.

The alpha launch of the project website demonstrated that the system could successfully stream live collision data to the web with sustained audio quality. The alpha launch also demonstrated that with sufficient computing power, the system can be parallelised such that the real-time data from the ATLAS detector can simultaneously drive three different musical compositions.

5.2 Composing with Custom OSC Interface

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.

Figure 5: Front end of Cosmic synthesizer. Zoom for detail

5.2.1 *Cosmic*

An audio stream called "Cosmic" was produced as a custom Max/MSP patch by Evan Lynch (see Figure 5). 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 audio synthesizers are determined by additional event-level parameters that were added to the system including missing transverse momentum. 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, 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.

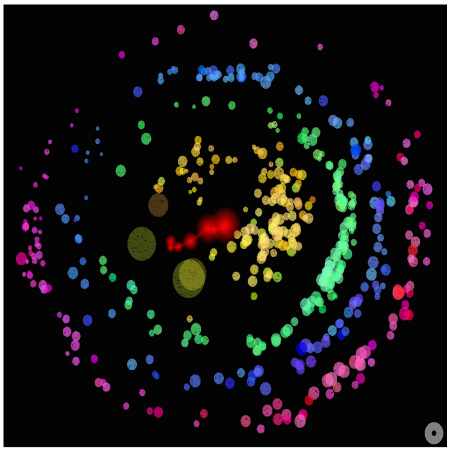

Figure 6: A screenshot of the audio-visual OSC interface

5.2.2 *Audio-Visual*

An animated audio-visual experience was produced by Akito van Troyer as an extension of a pre-existing project called Constellation [24]. 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 6). 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 interface is an artistic 2D interpretation of the detector where each ring 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.

6. EVALUATION OF THE PD GUI

While we expect many composers to develop experimental audio using custom-built synthesizers, we also seek to validate the system's default tools. As a preliminary validation, an experiment was run in which one composer attempted to create several compositions of distinct, traditional musical styles using only the GUI and the default Python interface. Any failure encountered would expose an inflexibility of the GUI as a music producing tool. The term "musical style" is subjective and was not defined prior to the experiment in a way that would allow for the precise quantification of the musical styles composed. As a result, this experiment is treated as a method for locating limitations in the system.

The musical pieces were composed by adjusting various properties in the PD GUI such as the tempo, MIDI mapping per stream, musical scale, and calorimeter beats, along with selecting different musical instruments from Ableton Live. Some of the musical styles that were attempted include: samba, classical, pop-rock, electronic, and tango.

One system limitation came about during attempts to compose a tango. The composer felt unable to approximate this musical style since the 8-beat control could not produce complicated structures over the span of multiple musical bars. The current graphical interface would also prevent the composer from producing an eight-bar blues or any musical style requiring a 3/4 time signature (since a pattern of 8 beats is not divisible by 3).

More granular control of rhythmic structure would overcome this limitation and is easily implemented. More complicated features will also be added to allow the user to approximate things like time signatures and larger structures involving multiple bars.

With the exception of the tango, the composer felt that their attempt to create the musical styles listed above were successful using the settings offered by the PD GUI. The data distributions translated to musical note distributions that proved amenable to the composer's needs.

7. FUTURE WORK

The system is currently optimised for the partially reconstructed, real-time ATLAS data stream with only some support for fully reconstructed, publicly available sample event data files. We prioritized developing a real-time system, but updates are planned to make the system compatible with both data types and to make the Python interface more easily extensible by the user. The user will be able to read in new data parameters and sonify more interesting quantities by appending their own Python code. These enhancements will enable us to work with additional composers.

8. CONCLUSIONS

We have built a platform that can produce sonified streams of real-time particle collision data from the ATLAS experiment and route these audio streams to a corresponding website that will be launched to the public in 2016. The platform's associated composition interfaces are demonstrated to be flexible enough to support diverse audio styles ranging from traditional, rhythmic pieces to experimental, audiovisual experiences. Future work will involve refining the interfaces to make them more flexible, performing a more detailed evaluation of the Python interface, and working with additional composers. When the website launches publicly, it will be available at `http://quantizer.media.mit.edu`.

9. ACKNOWLEDGMENTS

We wish to thank the ATLAS experiment for letting us use the data and for the continued support. Thank you to Media Lab composers Evan Lynch and Akito van Troyer for developing creative custom synthesizers and mappings discussed in this paper. Thank you to CERN for the invitation to perform at the Montreux Jazz Festival, and to Al Blatter for his performance collaboration. Thank you to Domenico Vicinanza for providing design feedback in the project's early stages as well as helping to present the project at different venues, including organising the ICAD workshop. Finally, Thank you to Felix Socher for sharing his ATLAS data parsing code.

10. REFERENCES

[1] Symphonies of the Planets, 1992. NASA Voyager Recordings. LaserLight Digital.

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

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

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

[5] Arts@CERN. Bill Fontana. 2013.

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

[7] K. 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`.

[8] S. Barrass and P. Vickers. Sonification design and aesthetics. 2011.

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

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

[11] J. Cherston, E. Hill, S. Goldfarb, and J. Paradiso. Sonification platform for interaction with real-time particle collision data from the atlas detector. In *CHI '16 Proceedings (To Appear)*, 2016.

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

[13] C. Dodge. Earth's magnetic field: Realizations in computed electronic sound, 1970. Nonesuch Records.

[14] I. Georgescu and F. Levi. Exhibition: Patterns in the dark. *Nature Physics*, 11(7):522–522, 2015.

[15] M. Grinberg. Flask-SocketIO, 2016. Retrieved January 19th, 2016 from `https://github.com/miguelgrinberg/Flask-SocketIO`.

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

[17] K. A. Olive et al. Review of Particle Physics. *Chin. Phys.*, C38:090001, 2014.

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

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

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

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

[22] 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.

[23] The ATLAS Collaboration. ATLAS 4-jet event at 13 TeV - 21 May 2015 - Run 265545 Event 2501742, 2015. Retrieved January 19th, 2016 from `https://cds.cern.ch/record/2017847`.

[24] A. van Troyer. Constellation: A tool for creative dialog between audience and composer.

[25] K. Vogt, R. Holdrich, D. Pirro, M. Rumori, S. Rossegger, W. Riegler, and M. Tadel. A Sonic Time Projection Chamber: Sonified Particle Detection at CERN. 2010.