

Towards Open 3D Sound Diffusion Systems

Fernando Lopez-Lezcano

CCRMA, Stanford University

nando@ccrma.stanford.edu

ABSTRACT

This paper describes the rationale, design and implementation of general purpose sound diffusion systems running on commodity PC hardware, and using open source free software components with minimal additional programming. These systems are highly configurable and powerful, can be extended as needed, and are a good fit for research environments as they can adapt to changing needs and experimentation in new technologies. This paper describes two examples: the system we have been using and extending for the past few years for sound diffusion in concert, and the system running permanently in our Listening Room at CCRMA, Stanford University.

1. A BIT OF HISTORY

1.0.1 The Quad Mainframe Years

Diffusion of electroacoustic music using multiple speakers and surround sound has been happening at CCRMA since its very beginning. The computing environment at the time was a mainframe (a PDP10 and later the Foonly's F2 and F4, both of which were PDP10 clones) and a custom designed realtime hardware synthesizer (the Samson Box) which featured five channel audio output. A lot of the music created and the research that made it possible dealt with spatial sound. As examples we can cite John Chowning's seminal paper on the simulation of moving sound sources [1] and the pieces he created, such as Stria or Turenas [2].

Concerts were organized to diffuse music using four channels with an analog mixer feeding a hodge-podge of speakers, including Altec A3's, Community Light and Sound HF horns and rented W bins. The setup included four 3-way crossovers from Yamaha and enough amplifiers to go around. Speakers coalesced later into four Meyer MSL3 speakers and its matching subwoofers, and amplifier and crossover racks. Concerts happened at Stanford's old AI Lab, Frost amphitheater and other spaces, and even around Lake Lagunita (with numerous speakers and quad pieces around the lake, and even musicians in boats).

Copyright: ©2014 Fernando Lopez-Lezcano et al. This is an open-access article distributed under the terms of the [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.0.2 The NeXT/SGI world

The mainframe, the SamsonBox and its four channel sound world ended in the early 90's as CCRMA moved its computing environment to a network of NeXT workstations. A central server and distributed workstations replaced the mainframe, and this also brought an interregnum of stereo CD quality sound.

In 1993 quad returned to the NeXTs through the development of the "QuadBox", a joint project between the author (working at the time at the SFC Campus of Keio University, Japan) and Atsu Tanaka (at the time a student at CCRMA). The experimental four channel D/A converter boxes hooked up to the DSP port of the NeXT and could play four channel soundfiles again.

We also bought a few SGI workstations and in 1998 we equipped one of them with an 8 channel soundcard hooked up to a cube of speakers in Studio D, and went from quad to experimental full 3D surround.

1.0.3 Gnu/Linux

After NeXT's demise, we started a slow migration to a network of GNU-Linux powered computers. Early sound support under Linux was limited and playing more than stereo was hard. At the beginning we used two independent stereo sound cards (ENS1370) with no clock synchronization to play four channel pieces. With identical hardware, the drift between front and back channels in a medium length piece was not a problem. Bill Schottstaedt added support for this in Snd and we kept surround going.

The Sonorus STUD/IO 8 channel soundcard was the first multichannel card we experimented with in Linux. In 1999 an OSS commercial closed source sound driver was released and we used one in Studio D.

Eventually affordable multichannel sound cards appeared in the market and ALSA provided a better driver infrastructure. The Midiman 1010 8 channel audio interface was introduced in 2000 and as soon as ALSA drivers became available it became the core of our Linux multichannel workstations.

We eventually moved our outdoor concerts from Frost to our backyard in The Knoll, and from the 4 Meyer system we transitioned into an array of 8 RAMSA speakers with big and heavy Yamaha power amplifiers which we used for both indoors and outdoors concerts.

1.0.4 New Studios, New Sound System

With the complete remodel of The Knoll in 2004 we gained new studios and a small concert hall. The Listening Room is a full 3D Studio with an acoustically transparent grid

floor and was initially equipped with 16 speakers in a 4 + 8 + 4 configuration. Over time the conventional DM-3200 digital mixer proved to be difficult to use and impossible to expand, and we started designing a software based custom control system.

In 2009 we bought a brand new set of 8 QSC HPR122i speakers and matching subwoofers to get better sound quality from our outdoor diffusion system. This is when development and testing of the new system started to take place, inspired by the work we had already done for the Listening Room.

The initial 8 speaker system was controlled by a digital mixer, and was expanded first to 16 speakers and 4 subwoofers in 2011, and then to 24 speakers and 6 subwoofers in 2012. This enabled us to start diffusing concerts with 3D sound. At the same time we slowly moved from using the digital mixer to a software-only solution running in one of our workstations.

2. GOALS OF THE SYSTEM

Our outdoor and large scale concerts have always had an eclectic mix of pieces with different aesthetics and diffusion needs. Some are stereo to be diffused in realtime, some are multichannel fixed media pieces, some are multichannel with realtime performance, some require Ambisonics decoding. Requirements are different from concert to concert and from piece to piece in a given concert so we needed a flexible system. These are some of the goals that surfaced over the years as we developed the software:

2.1 Simplicity

The main goal of the system is the diffusion of pieces that have already been composed for multichannel diffusion either as fixed media pieces or for realtime performance through multiple speakers. We are not creating a spatial compositional environment, or a spatial movement description language.

2.2 Transparency

The system has to provide the most accurate sound possible with the equipment available. Loudness and delay differences between speakers should be compensated for, and once calibrated the system should provide a flat frequency response and a linear phase response. It should not contribute audible noise of its own to the venue where the concert takes place.

2.3 Versatility

The system should be able to diffuse pieces using multiple spatialization paradigms. It should also be adaptable so that pieces composed with different diffusion requirements can be performed over a single set of loudspeakers without changing the physical layout. It should also be easy to reconfigure for different concert venues with different requirements and number of speakers.

2.4 Commodity Hardware

The system should try to use commodity hardware as much as possible, and avoid designing hardware unless there are no commercial alternatives. Off the shelf hardware lowers the total system cost and allows the system to grow in channel count and capabilities as new, cheaper technologies become available.

2.5 Free Software

The system should be built from free software open source components. Having access to the source code of all applications (and even the operating system itself) is invaluable in a research environment such as ours.

2.6 Footprint

In conventional venues the mixer is located at the front of the house, far from where most of the audience is seated. Diffusion of multichannel pieces usually requires the mixing position to be located at or near the center of the speaker array. It is therefore important that the footprint of the mixing equipment be kept to a minimum.

3. SOUND DIFFUSION SYSTEMS

3.1 Hardware Based and Hybrid Systems

The simplest sound diffusion system is composed of a digital mixer feeding an array of loudspeakers. This solution is simple, fairly inexpensive, and uses components that are readily available. However, it does not scale well for large loudspeaker arrays. Most digital mixers are designed for mixdown to stereo or 5.1 or 7.1 formats, and have only 8 or 16 general purpose mixing buses.

While low cost systems can be built for arrays of up to 16 or less speakers, going beyond that involves very expensive and physically big digital mixers. The flexibility of digital mixers is limited, specially when diffusing high channel count electroacoustic music, they are complex to operate, and have many features that are not normally used for our purposes and increase their price.

In addition to simple digital mixers, there are companies that offer packaged solutions for multichannel diffusion, usually a hybrid of hardware and proprietary software. We avoid the cost of designing and building these systems, but their closed nature makes their use in an open research environment such as ours problematic.

In some cases diffusion systems are designed and built for specific purposes and embedded into concert halls or sound diffusion venues. They can be built from off the shelf hardware, custom designs or a mix of both. Examples include the diffusion system in the MUMUTH [3] concert hall in Graz, Austria, the WFS (Wave Field Synthesis) system that is part of the 104 Lecture Hall in TU Berlin [4], Germany or the Klangdome 43 speaker system at ZKM, Karlsruhe, Germany.

3.2 Software Based Systems

These involve software specifically designed to act as an audio router, diffusion and mixing engine running on stan-

standard computers. Examples of systems that use this approach: Jonty Harrison's BEAST [5] (Birmingham Electroacoustic Sound Theatre) system and its BEASTMulch SuperCollider based control software [6] running on OSX, James R. Mooney's M2 system [7], the ICAST [8] system by Stephen David Beck, Joseph Patrick, Brian Willkie, and Kenley Malveaux, André Bartetzki's software-based Mixing Desk for Acousmatic Sound Diffusion [9] written in SuperCollider and others. The flexibility of this approach is what we needed.

4. BUILDING A SYSTEM WITH FREE SOFTWARE

The emergence of Open Source and Free Software (free as in freedom) systems that span from the operating system itself to all user programs allows the creation of reliable and configurable diffusion systems. The following sections describe how commodity hardware and free software components can be connected together to first replace the basic functionality of a digital mixer, and then go beyond it.

4.1 Computer Hardware and Interfaces

General purpose personal computers have become powerful and inexpensive and have become the core of our system. Our workstations use 4 or 6 core processors, large amounts of RAM, an SSD (Solid State Disk) for the operating system and programs, and high capacity conventional hard disks or SSDs in a redundant RAID array configuration to store sound and video materials.

4.1.1 Noise control

Most spaces in which we deploy our system have a very low noise floor, so a no-noise design is a must for us. The computer case and associated components should have passive cooling or fanless water cooling. We use passively cooled TNN500 and TNN300 cases from Zalman (no longer in production) [10]. These cases use heat pipes to transfer heat from the processor, the graphics card and the chipset to passive heat sinks and have fanless power supplies. The system has no moving parts if it is fitted with SSD hard disks.

Other companies currently market similar designs (for example the cases made by A-Tech Fabrication), in most cases for Home Theater PCs which require completely silent operation.

4.1.2 Soundcards, D/A and A/D Converters

For 8 or less channels there are many PCI or PCIe soundcards with Linux support and the best low latency behavior. Examples of low cost 8 channel soundcards we have used include the Midiman 1010 and the Layla 3G. Other alternatives for 8 or less channels include Firewire or USB2 soundcards but, in general, the latency is not as good as in PCI or PCIe. One example with good Linux support is the Presonus 1818VS USB2 soundcard.

Most soundcards include an SP/DIF digital output that can be used to drive a small stereo D/A converter. The end result is an inexpensive system with a total of 10 analog

outputs, suitable for driving up to 8 main speakers and 2 subwoofers.

For higher channel counts we need to use outboard D/A and A/D converters with ADAT or MADI interfaces. In this case the RME family of PCI or PCI express soundcards offers very good sound quality, high channel counts and good Linux support.

A solution for high channel count applications is ethernet connected D/A A/D systems. We have used 1/2 of a digital snake as a high quality, high channel count and low cost per channel "soundcard" (the Network Sound Audiostreamer 32 channel product). As the ethernet protocol used by this product is very simple we were able to write a custom program (jack-mamba [11]) that acts as a Jack client (see below) and transforms the digital snake into an "ethernet soundcard".

4.1.3 Peripherals

USB MIDI control surfaces can be used to directly control the software programs. We have used the BCF2000 and BCR2000 Behringer products, which are very cheap but so far reliable. Products with motorized fader control are particularly interesting as the hardware always reflects the status of the system. It is easy to connect the individual controls to the software that runs the diffusion system.

4.2 Software

GNU-Linux can be installed from one of several distributions that supply ready to install packages for most audio applications. Examples include Fedora with the Planet CCRMA repository (which the author created in 2001 and still maintains), Ubuntu Studio, KXStudio, ArchLinux, Gentoo with the Pro-Audio overlay and many others.

As our goal is to replace hardware based solutions such as digital mixers, we need to run the audio interfaces with very low latencies. The stock Linux kernel is suitable for most audio applications, but the lowest latencies can only be achieved by using a kernel that has the RT (realtime)[12] patches applied. With a properly tuned system it is easy to achieve latencies of a few milliseconds, even under full load.

We can also choose the desktop environment that the user experiences after logging into the computer (Gnome3, KDE and others). For concert diffusion applications we usually select one that places minimum CPU and memory load on the system. Suitable choices include Xfce or LXDE.

4.3 A Basic System

By installing the following software packages we can piece together a simple diffusion system with no additional programming:

- Jack [13] is a low latency sound server which can be used to drive a soundcard and interconnect multiple audio applications with no additional latency. It has a very simple API and all Linux audio software can interface with it. It is the glue that connects many different audio

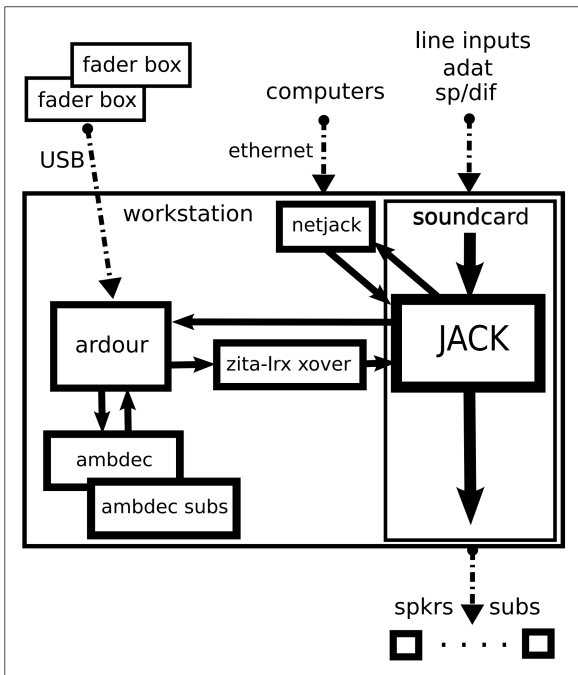


Figure 1. A basic system.

applications together seamlessly. Using Jack our diffusion system can also receive audio through an ethernet connection from other computers that run Jack.

- Zita-lrx [14] is a high quality LinkwitzRiley [15] 4th order speaker crossover. It connects any other sound sources in the Jack graph to the speakers and subwoofers, and can compensate for delay and level differences between speakers. It is controlled through a simple text configuration file.
- Aj-snapshot [16] is a Jack client that can record the connection graph of all Jack applications, and reconnect them on demand. With it we can reload our connection graph for different setups (it is a memory for the patchbay embodied by Jack).
- Ardour [17] is a DAW (Digital Audio Workstation) with basic capabilities similar to that of ProTools or other commercial alternatives. It has proven useful for generic multichannel diffusion as its design does not impose a limit on the number of buses or tracks available. It can become the front end of our diffusion system, used either as a multichannel playback engine for fixed media pieces, or as a mixer for realtime diffusion. It can be controlled through MIDI or OSC, and it is easy to bind controls from external control surfaces to any parameter. Loading different sessions during a concert is a simple way to redefine our interface to the requirements of individual pieces. Ardour can use plugins (LV2, LADSPA) to provide extra features like equalization, limiting and compression, etc.
- Ambdec [18] is a dual band Ambisonics decoder that can decode up to full 3rd order Ambisonics. Presets are

available for the most common regular speaker configurations.

We can interconnect all these programs through Jack to replace the basic functions of a digital mixer. A soundcard provides analog and/or digital I/O, and the netjack Jack backend allows remote computers to send and receive audio through ethernet. Ardour is the front end for the system and Zita-lrx provides crossovers and level and delay equalization. Ambdec can decode pieces realized in Ambisonics.

We can run a second instance of Ardour to play multichannel pieces or act as a mixer for external sources. This makes it easy to have a diffusion Ardour session running permanently while being able to load several pieces in succession during a concert.

4.4 A Basic System with DRC

With a few additional programs we can calibrate the frequency and phase response of the speakers:

- Aliko [19]: impulse response measuring software
- Digital Room Correction (DRC) [20]: uses impulse responses to generate speaker calibration filters
- Jconvolver [21]: low latency convolution engine.

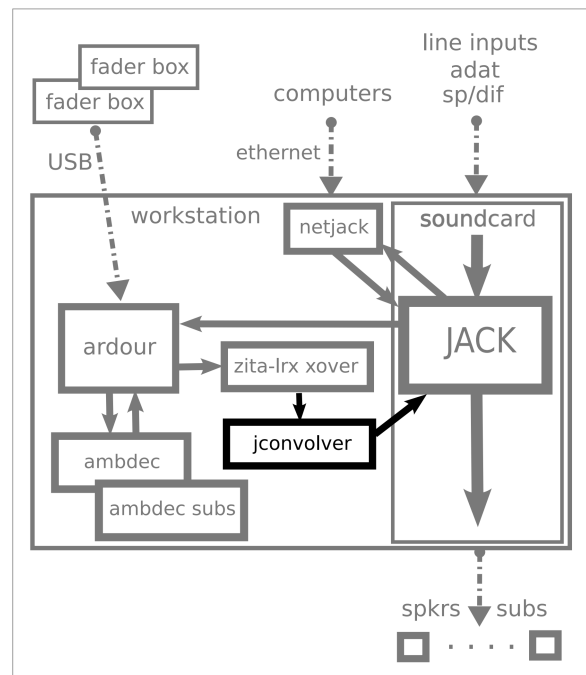


Figure 2. A basic system with DRC.

Aliko uses sine sweeps to measure impulse response accurately. We record the impulse responses of all speakers once installed in the concert venue and derive correction filters using the DRC (Digital Room Correction) software package.

In this example system the Zita-lrx crossover connects to Jconvolver, a low latency convolution engine which runs

the filters created by DRC and connects them to the speakers. In our experience DRC creates filters that make an audible difference in the frequency response and impulse response of the speakers, with a noticeable improvement in clarity specially in the mid frequency range.

The tradeoff for this improvement in sound quality is the latency added by the convolution process (10 to 15 mSecs). That has not proven to be a problem so far, but can be made optional by rerouting the signal path around Jconvolver.

4.5 Using SuperCollider

Even though Ardour is a very flexible DAW, it has a fixed track based paradigm. We have coded most of the core functionality of the diffusion system in SuperCollider [22] (it is also the language used for coding the BEASTMulch control software and other similar systems). Running the Supernova [23] synthesis server instead of the older scsynth provides load balancing of the DSP load among multiple cores and fully utilizes the power of contemporary processors.

Although the basic functions of the system are now written in SuperCollider we still use Ardour as the graphical front end to the system.

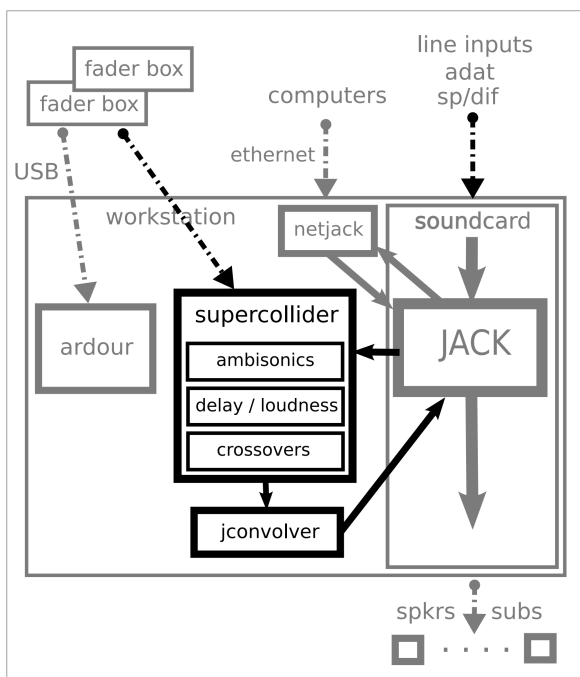


Figure 3. A SuperCollider based system.

4.5.1 Ambisonics

High Order Ambisonics is a flexible paradigm for sound diffusion. As many of our pieces use it, we incorporate Ambisonics decoders directly into the diffusion system. Anything realized natively in Ambisonics is very easy to diffuse. Ambisonics also helps us play pieces created for speaker arrays that do not match the deployed configuration — we create virtual sources using Ambisonics panners in Ardour, and diffuse the piece through those virtual speakers.

An additional software package (ADT [24][25] - Ambisonics Decoder Toolkit) makes it easy to design Ambisonics decoders for irregular or dome configurations, and can generate Ambdec configuration files or Faust code that can be compiled into SuperCollider UGens. ADT takes as inputs the speaker locations and the type of decoder to generate, runs in Octave/Matlab and outputs configuration files or Faust code.

4.6 The Current System

Our sound diffusion computer is one of our Linux workstations, currently an Intel 6 core machine with 64G of RAM and 1Tbyte of redundant fast disk storage, and housed in a TNN500 silent case. It has a RayDAT RME PCIe soundcard used mostly for connecting with external digital systems if needed, and for providing the master clock to Jack. The analog audio interface to external sources and the speakers is an ethernet connected 32 channel AudioStreamer box. A custom program written in C acts as a Jack client (jack-mamba [11]) and interfaces through a dedicated ethernet port to the AudioStreamer box, transforming it into a high quality low latency 32 channel I/O subsystem.

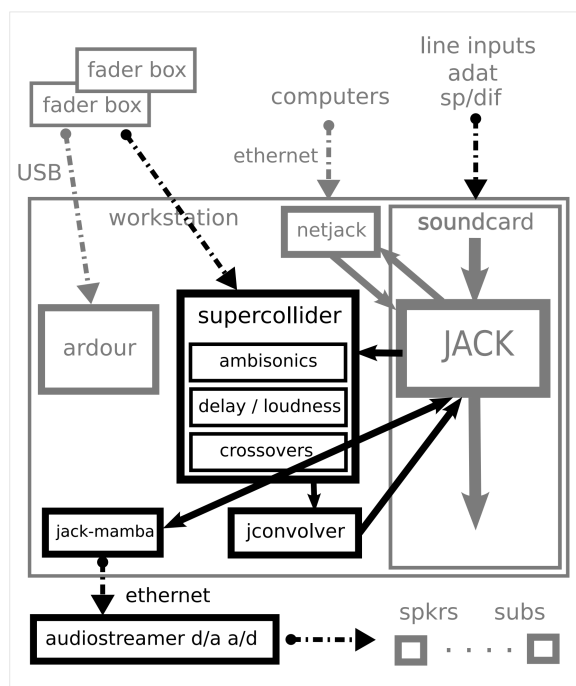


Figure 4. The current system.

We have been running our concerts with this system since 2010, first as the sidekick of a digital mixer and lately replacing it completely.

To further minimize the footprint of the system at the mixing position we can also remote the monitor (HDMI) and USB2 interfaces of the computer through ethernet connected extender boxes. As the audio I/O is also handled by a network connected box, the computer itself can be located anywhere in the venue, as long as we can connect to the peripherals with CAT5 ethernet cable runs.

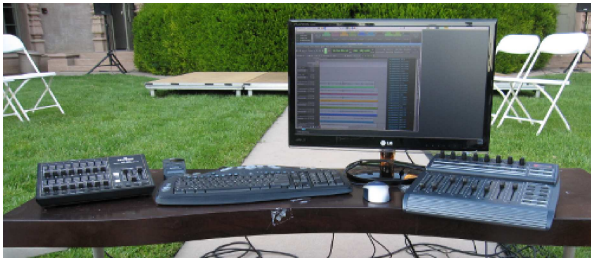


Figure 5. Mixing Station at the Transitions 2011 Concert.

4.7 Concerts

This is a short list of recent concerts that have shaped the evolution of the system (detailed information available in CCRMA’s web site):

- 2011: Transitions: (8 + 8).4: first deployment of a Linux only system in the first night of the concert using the Audiostreamer box as a D/A, first CCRMA outdoor concert with 3D sound
- 2012: Transitions: 24.6: upgrade to a full 3D surround 24.6 system
- 2013: Bing Concert Hall opening season:
 - Opening Night Fanfare (10 + 5 PA + .4)
 - Bada Boom Bada Bing Festival (2 nights, 24.6 in 12+8+4 dome, Bing’s Main Hall)
 - From Constantinople to California, virtual Hagia Sophia acoustics, 24.6 in 12+8+4 dome, Bing Main Hall
 - Theotokia and The War Reporter, chamber operas by Jonathan Berger, 24.6 in 12+8+4 dome, Ambisonics, ADT decoders, Bing Main Hall
- 2013: Transitions, 20.6, ADT Ambisonics decoders and first use of DRC for all speakers
- 2013: CCRMA Fall Concert, Bing Main Hall, 16.6: 5th order Ambisonics decoding through ADT (as SuperCollider Ugen) and DRC for all speakers
- 2014: CCRMA Winter Concert and Teleconcert, Bing Studio, 24.6 in 12+8+4 dome: first 3D deployment in the Bing Studio with multiple Ambisonics decoders in SuperCollider and DRC for all speakers

5. EXAMPLE: HAGIA SOPHIA AT BING

During the 2013 inaugural season of the Bing Concert Hall at Stanford, CCRMA supplied the diffusion system and virtual acoustics technology for the Stanford Live’s “From Constantinople to California” concert. This was the culmination of a joint research project between CCRMA and the Art and Art History Department that had been in the making for several years [26] [27].

We created virtual acoustics for the Hagia Sophia Dome in Istanbul, Turkey, for a live performance of the Cappella

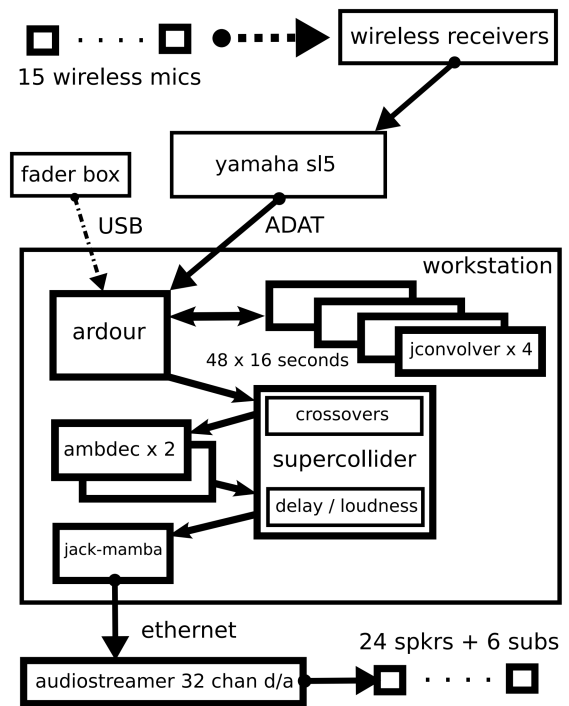


Figure 6. Hagia Sophia routing.

Romana byzantine chanting group, and deployed it inside the Bing Concert Hall for the second half of the concert.

From the diagram you should recognize most of the software components described above. The 15 wireless microphones were connected to the house mixer, equalized, and sent to our computer through a pair of ADAT links. All audio outputs were sent through the Audiostreamer box to 24 speakers and 6 subwoofers rigged in a dome configuration around the audience. The Ardour session provided mix-down and routing of all signals, sending the microphones to four Jconvolver convolution engines which were running a total of 48 16-second low latency convolutions, and mixing and spatializing their outputs in 3D 3rd order Ambisonics.

During the concert performance our diffusion computer was running all cores at between 40% and 50% of full load with no glitches. This concert demonstrated the versatility of the system as we were able to easily reconfigure it to our needs.

6. EXAMPLE: THE LISTENING ROOM SYSTEM

Systems created with open source free software can also be used for controlling fixed studio systems. Our Listening Room is a medium sized studio (7m x 7.4m x 2.1m) that was built in 2004. It is acoustically very dead, has a very low noise floor (below 24 dB) and part of the floor is an acoustically transparent metal grid that allows for loudspeakers to be positioned below it.

In its current configuration it houses 22 speakers in a 1 + 6 + 8 + 6 + 1 configuration which allows for full 3D high quality sound diffusion. The speaker configuration



Figure 7. Hagia Sophia in Bing, background: the Cappella Romana group, foreground: the Ardour session.

can also decode 3rd order full periphonic Ambisonics audio streams.

The control computer boots Linux and automatically starts SuperCollider which in turn runs a custom program called OpenMixer [28]. This program controls the rest of the system, automatically starting and monitoring the rest of the software building blocks and connecting them together.

The user interface is simple and easy to use, and uses two low cost Behringer USB control surfaces (BCF2000 and BCR2000). It can connect several analog and digital sources to all speakers and can also automatically decode Ambisonics input signals.

The system has been running since 2009, and the commodity PC hardware plus software approach to solving the problem of controlling the diffusion of sound in the Listening Room was the inspiration for the current concert system.

7. FUTURE WORK

The project is evolving with each new concert and performance. A few of the long term goals of the project:

- Use an AVB (Audio Video Bridging) stack under Linux for more standard compliant and modular network based A/D D/A.
- OSC (Open Sound Control) control of all components to enable wireless control of the system (using tablets or smart phones).

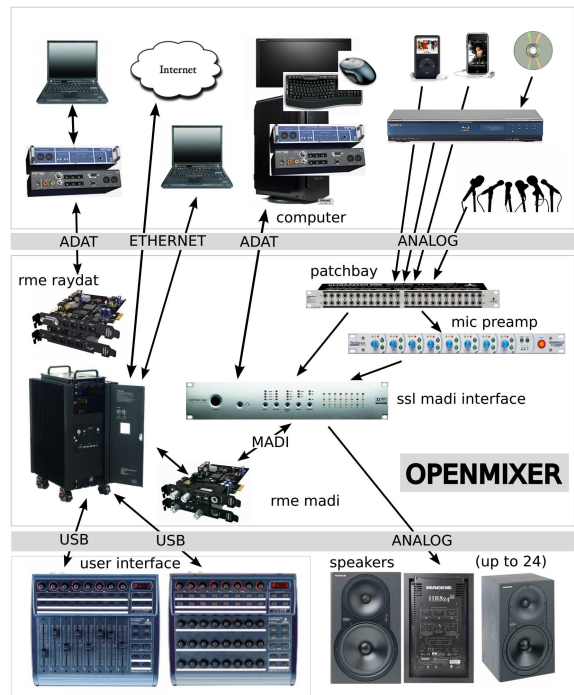


Figure 8. The Listening Room hardware.

- Include DRC (Digital Room Correction) in the system, and write software to automate the complete calibration process.
- Use native SuperCollider Ugens for creating the partitioned convolution used by DRC (it is currently done externally through Jconvolver for efficiency reasons).
- Create a GUI in SuperCollider, including banks of VU meters for monitoring, routing and fader control.
- Create a control layer over the rest of the system for acousmatiqce diffusion that goes beyond the “one fader, one speaker” paradigm.

8. CONCLUSIONS

A software based diffusion system was described, it runs on commodity PC hardware and uses open source free software components where possible, and SuperCollider programming where advantageous. The software and hardware have been used successfully on many concerts and events over the past few years, it can replace a digital mixer and go beyond its capabilities for sound diffusion projects. It has also proved to be adaptable when new requirements are needed and was modified for special events like the virtual Hagia Sophia virtual acoustics recreation in the Bing Concert Hall.

Acknowledgments

This work would not have been possible without the help of many individuals: Chris Chafe, Jonathan Berger and the community at CCRMA, including the composers and sound artists that have been guinea pigs for testing and deploying this system in many concerts. Infinite thanks to the many individuals that contribute their time to creating and maintaining professional audio software, and releasing it under free licenses. I would like to specially thank Fons Adriansen, Paul Davis and Stephane Letz for their many contributions that made this project possible, and Aaron Heller and Eric Benjamin for their ADT toolkit and never ending help and advice with Ambisonics decoding challenges.

9. REFERENCES

- [1] J. Chowning, "The simulation of moving sound sources," in *Journal of the Audio Engineering Society*, 1971.
- [2] —, "Turenas: the realization of a dream," in *Journes d'Informatique Musicale, Universit de Saint-Etienne*, May 25 2011.
- [3] P. Plessas and J. Zmölign, "The iem demosuite, a large-scale jukebox for the mumuth concert venue," in *Proceedings of the Linux Audio Conference 2012 (Linux Audio Conference 2012)*, S. Ccrma, Ed., Stanford (Vereinigte Staaten (USA)), 2012.
- [4] M. Baalman, S. Schampijer, T. Hohn, T. Koch, D. Plewe, and E. Mond, "Renewed architecture of the swonder software for wave field synthesis on large scale systems," in *Proceedings of the Linux Audio Conference 2007 (Linux Audio Conference 2007)*, 2007.
- [5] J. Harrison, *Diffusion: theories and practices, with particular reference to the BEAST system*, 1999 (accessed June 16 2014). [Online]. Available: <http://cec.sonus.ca/econtact/Diffusion/Beast.htm>
- [6] S. Wilson and J. Harrison, "Rethinking the beast: Recent developments in multichannel composition at birmingham electroacoustic sound theatre," in *Organised Sound*, 2010, pp. 239–250.
- [7] J. R. Mooney, "Sound diffusion systems for the live performance of electroacoustic music," Ph.D. dissertation, University of Sheefield, 2005.
- [8] S. D. Beck, J. Patrick, B. Willkie, and K. Malveaux, "The immersive computer-controlled audio sound theater: Experiments in multi-mode sound diffusion systems for electroacoustic music performance," in *Proceedings of the ICMC 2006 (International Computer Music Conference 2006)*, 2006.
- [9] A. Bartetzki, *A Software-based Mixing Desk for Acoustic Sound Diffusion*, 2007.
- [10] F. Lopez-Lezcano, "The quest for noiseless computers," in *Proceedings of the Linux Audio Conference 2009*, 2009.
- [11] —, "From jack to udp packets to sound and back," in *Proceedings of the Linux Audio Conference 2012*, 2012.
- [12] "Real-time linux wiki." [Online]. Available: https://rt.wiki.kernel.org/index.php/Main_Page
- [13] P. Davis and S. Letz, "Jack audio connection kit." [Online]. Available: <http://jackaudio.org/>
- [14] F. Adriansen, "zita-lrx, a 4th order 1r crossover." [Online]. Available: <http://kokkinizita.linuxaudio.org/linuxaudio/>
- [15] S. Linkwitz, "Active crossover networks for noncoincident drivers," in *Journal of the Audio Engineering Society*, Volume 24 Issue 1, 1976, pp. 2–8.
- [16] L. Moors, "aj-snapshot, a utility to store/restore alsa and/or jack connections." [Online]. Available: <http://sourceforge.net/projects/aj-snapshot/>
- [17] P. Davis, "Ardour." [Online]. Available: <http://ardour.org/>
- [18] F. Adriansen, "Ambdec: an ambisonics decoder." [Online]. Available: <http://kokkinizita.linuxaudio.org/linuxaudio/>
- [19] —, "Alike, an impulse response measurement tool." [Online]. Available: <http://kokkinizita.linuxaudio.org/linuxaudio/>
- [20] D. Sbragion, "Drc (digital room correction)." [Online]. Available: <http://drc-fir.sourceforge.net/>
- [21] F. Adriansen, "Jconvolver, a low latency convolution engine." [Online]. Available: <http://kokkinizita.linuxaudio.org/linuxaudio/>
- [22] J. McCartney, "Supercollider." [Online]. Available: <http://supercollider.sourceforge.net>
- [23] T. Blechmann, "Supernova - a multiprocessor aware real-time audio synthesis engine for supercollider," Ph.D. dissertation, TU Wien, Faculty of Informatics, 2011.
- [24] A. Heller, E. Benjamin, and R. Lee, "A toolkit for the design of ambisonic decoders," in *Proceedings of the Linux Audio Conference 2012*, 2012.
- [25] A. Heller and E. Benjamin, "The ambisonics decoder toolbox: Extensions for partial-coverage loudspeaker arrays," in *Proceedings of the Linux Audio Conference 2014*, 2014.
- [26] F. Lopez-Lezcano, T. Skare, M. Wilson, and J. Abel, "Byzantium in bing: Live virtual acoustics employing free software," in *Proceedings of the Linux Audio Conference 2013*, 2013.
- [27] J. V. Serinus, "Cappella romana: Time travel to constantinople." [Online]. Available: <http://www.sfcv.org/reviews/stanford-live/cappella-romana-time-travel-to-constantinople>
- [28] F. Lopez-Lezcano and J. Sadural, "Openmixer: a routing mixer for multichannel studios," in *Proceedings of the Linux Audio Conference 2010*, 2010.