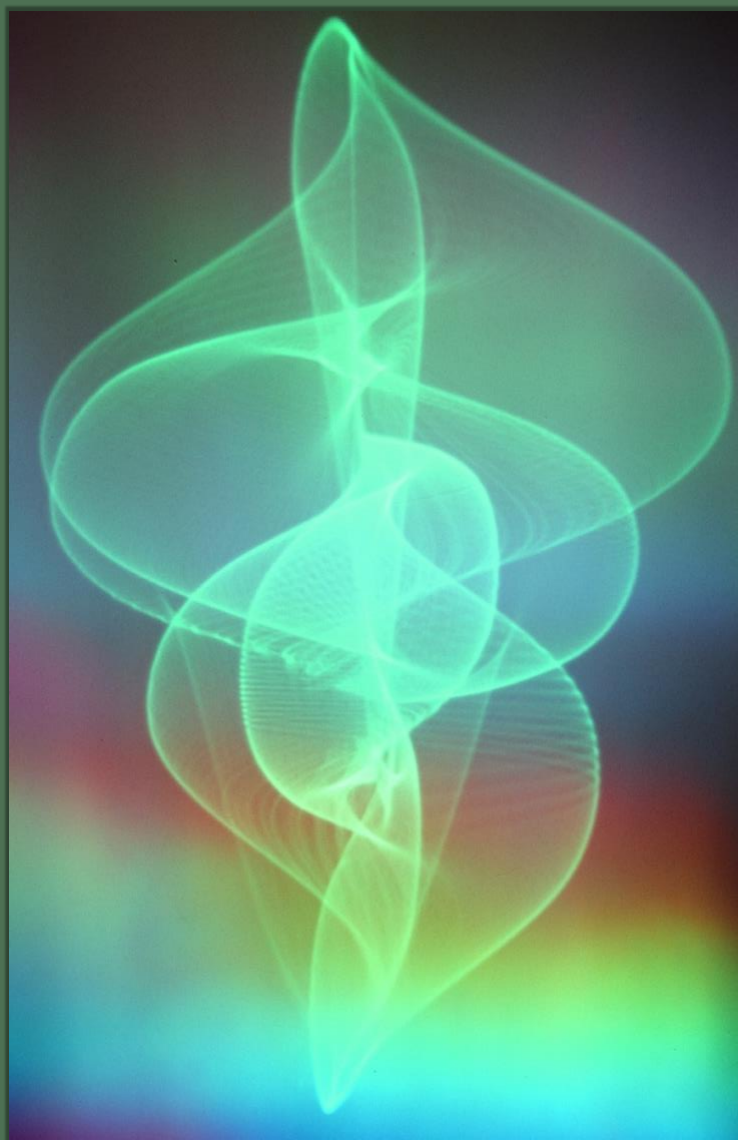


Journal SEAMUS

Spring/Fall 2019

The Society for Electro-Acoustic
Music in the United States

Volume 30, Number 1–2



ISSN 0897-6473

Editor-in-Chief

Drake Andersen

Assistant Editors

Nick Hwang

Michael Lukaszuk

Copy Editors

Nicholas Cline

Dave Mahony

Digitization Manager

Holli Wittman

Review Board

Bruce Bennett

Dániel Péter Biró

Ted Coffey

Paul Koonce

Colby Leider

Jennifer Logan

Loren Ludwig

Cover Art

Maggi Payne

Contact Information

Drake Andersen

Department of Music

Vassar College

124 Raymond Ave.

Poughkeepsie, NY 12604

journal@seamusonline.org

Journal SEAMUS

Volume 30, Number 1–2

Spring/Fall 2019

ISSN 0897-6473

Please visit us at www.seamusonline.org to find out more about SEAMUS

Information for Contributors

Interested contributors should submit manuscripts electronically. Microsoft Word is the preferred format. If another word processor is used, files should be saved in rich text format (RTF) with an accompanying PDF version. Main articles are generally 2,000 to 6,000 words.

Editing Guidelines Please use Times New Roman fonts with font size 12. Manuscripts should be formatted and prepared using *The Chicago Manual of Style*, 17th edition (2017) as a guide. References should follow Author-Date format. Specific citations should be provided in text in parentheses. Footnotes should be used sparingly and reserved for explanation beyond the text of the article. All references should be listed after the text of the article in a section

labeled “References.” Any computer code should be placed in fixed-width format to facilitate readability. Images, figures, musical examples, and other graphics should be sent as separate attachments for ease of layout. The approximate location of each graphic should be indicated in the text by a (sequentially numbered) label and a brief caption.

Graphics Any artwork, graphics, photos, and flowcharts should be sent as separate individual files. We recommend uncompressed graphic files such as TIFF at 300 dpi.

Submission All submissions, including articles, reviews, review proposals, and items for *Tips and Tricks* should be emailed to the Editor-in-Chief, Drake Andersen: journal@seamusonline.org.

About SEAMUS

Founded in 1984, The Society for Electro-Acoustic Music in the United States (SEAMUS) is a non-profit national organization of composers, performers, and teachers of electro-acoustic music representing every part of the country and virtually every musical style. Electro-Acoustic music is a term used to describe those musics, which are dependent on electronic technology for their creation and/or performance. Many members of SEAMUS, like Jon Appleton, the guiding light in the conception of the Synclavier, are recognized world leaders in their fields. All are dedicated to the use of the most advanced technology as the tools of their trade.

SEAMUS seeks to provide a broad forum for those involved or interested in electronic music. Through its journal, newsletter, national meetings, and its national archive at the University of Texas, SEAMUS seeks to increase communication among the diverse constituency of the relatively new music medium.

The Society's objectives include:

To encourage the composition and performance of electro-acoustic music

To develop a network for technical information and support

To promote concerts and radio broadcasts of electro-acoustic music both in the US and abroad

To create an exchange of information through newsletters and other means of communication

To establish and maintain a national archive and information center for electro-acoustic music

To attract a wide diversity of members and supporters

To advocate licensing and copyright concerns

SEAMUS strives to address not only relevant technology but also the non-technical issues pertinent to the electro-acoustic music community. In a field usually dominated by technical concerns, it is refreshing to hear paper sessions devoted to aesthetics, collaboration, education, and of the ethical and social issues facing electro-acoustic musicians. The provocative sessions provide fuel for lively discussions during the national meetings.

SEAMUS Board of Directors

President	Elizabeth Hoffman	president@seamusonline.org
Vice President of Programs	Leah Reid	vp_programs@seamusonline.org
Vice President for Membership	Abby Aresty	vp_membership@seamusonline.org
Member at Large	Eli Fieldsteel	eli.fieldsteel@gmail.com
Member at Large, Outreach	Ralph Lewis	ralphlewis.music@gmail.com
Treasurer	Heather Mease	treasurer@seamusonline.org
Secretary	John Gibson	johgibso@indiana.edu
Editor, SEAMUS Newsletter	Jiayue Cecilia Wu	newsletter_editor@seamusonline.org
Editor, SEAMUS Journal	Drake Andersen	journal@seamusonline.org
Diversity Officer	Elizabeth Hinkle-Turner	dir_conferences@seamusonline.org
Director, Recordings	Scott Miller	dir_recordings@seamusonline.org
Director, Tech. Development	Becky Brown	techdev@seamusonline.org
Director, Communications	Mark Vaughn	dir_communications@seamusonline.org
Immediate Past President	Ted Coffey	ejc3q@virginia.edu

Contents

Journal SEAMUS

Volume 30, Number 1–2, 2019

From the Editor	
	2
Articles	
SCAMP: A Suite for Computer-Assisted Music in Python <i>By Marc Evanstein</i>	3
Kinetic Relationships in Visual Music Composition: Interactive Changes over Time <i>By Michael Rhoades</i>	18
Hacking Pärt: A Robust, Compositionally-Based System for Generating Live Harmonies <i>By Christopher Tignor</i>	25
Studio Report	
Inventorying and Evaluating the University of Illinois Experimental Music Studios' Archive <i>By David M. Luftig</i>	33

From the Editor

I am pleased to announce the publication of Journal SEAMUS Volume 30 as a double issue. This volume represents a significant step forward in realizing the journal's potential as a digital-only publication. The included articles make extensive use of multimedia content, much of which can be accessed simply by clicking on the hypertext links within the text. We hope that this new approach to accessing supplementary content will encourage even greater engagement with the work in this issue—and in issues to come.

We are committed to continually explore new modes of presenting our members' work as we work through the Journal backlog and begin to look towards future issues. This is probably a good place to add that if you have an idea for an article that takes advantage of our new, digital format in an interesting way, we want to hear about it! Feel free to submit your manuscript at any time by following the instructions on this page: <https://seamusonline.org/journal-seamus/>. Or if you'd like to pitch something that's still in progress, that's fine, too—just drop us a line at: journal@seamusonline.org. We'll be happy to discuss any projects you have in mind!

I would like to express my appreciation for all of the SEAMUS members who contributed cover art for future journal issues in response to our call. For Volume 30, the Journal team has elected to feature a photograph by 2022 SEAMUS Award laureate composer Maggi Payne. Maggi's striking image, which dates back to the mid-1970s, reminds us of the rich legacy of technologies that generations of SEAMUS members have used to make art—both sonic and otherwise! We look forward to featuring other members' art in future issues.

I am proud to say that this issue showcases the diversity of thought, method, and specialization that characterizes all of SEAMUS's activities, and which Journal readers have long come to expect and appreciate. I hope you enjoy Journal SEAMUS's Volume 30, featuring an overview of Marc Evanstein's innovative framework for computer-assisted composition in Python, Michael Rhoades's creative examination of several classes of kinetic relationships between sonic and visual elements devised in his lustrous artistic language, Christopher Tignor's comprehensive account of his elegant Arvo Pärt-inspired harmonization system for live performance, and David M. Luftig's richly documented reporting on the archival holdings of the Experimental Music Studios at the University of Illinois.

As always, thanks are due to the Journal SEAMUS staff, as well as to the SEAMUS leadership and members who have offered their support to the Journal's operations in countless ways, both large and small. That includes readers like you!

Drake Andersen, Editor-in-Chief

SCAMP: A Suite for Computer-Assisted Music in Python

Marc Evanstein

University of California, Santa Barbara
marc@marcevanstein.com

Abstract

This paper introduces SCAMP, a computer-assisted composition framework in Python designed to bridge the gap between the continuous timing of synthesis-based frameworks and the discrete timing of notation-based frameworks. SCAMP allows the composer to quickly audition and iterate over musical ideas based on the sonic result, and then flexibly quantize and export the music in western notation. The framework offers varied and highly flexible options for playback, including built-in, soundfont-based playback via *fluidsynth*, as well as the ability to play notes using MIDI or OSC messages to external synthesizers. Although fundamentally note-based, SCAMP supports both microtonality and continuously varying curves in pitch, volume, and other arbitrary synthesis parameters. Timing is controlled via a flexible clock system capable of coordinating multiple streams of music following separate tempo curves, and notation can be exported in the form of either MusicXML or LilyPond (via the *abjad* library). The goal of SCAMP is to act as a hub, connecting the composer-programmer to the other tools in their compositional workflow, and allowing them to abandon the framework when it no longer serves the aims of a given composition.

Motivation

Consider a composer who wishes to compose a piece for string quartet based on climate data. Having downloaded the data in CSV format, they wish to process it and experiment with different mappings by ear. Finally, having crafted their preferred mappings into an overarching musical form, they wish to output some preliminary notation, and then reshape the result by hand in their preferred score-writing software. Or

consider a composer who wants to write a piece for instrument and electronics where a simple mass-spring simulation generates a stream of glissandi, which simultaneously drives a modular synthesizer and results in written notation for the instrumentalist. Or consider a composer who wishes to write an algorithmically generated piano concerto in which the piano and orchestra follow separate accelerating and decelerating tempo curves. The pianist requires a score notated from the point of view of the piano's tempo curve, while the conductor requires a score notated from the point of view of the orchestra's tempo curve, perhaps with a renoted piano part for reference. What these scenarios have in common is the translation of musical data between different domains. In particular, all three contend with the transition between the continuous domain of sounding music and the discrete (and idiosyncratic) domain of notated music.

These considerations were the driving forces behind the creation of SCAMP (Suite for Computer-Assisted Music in Python), a GPL3.0-licensed pure-Python framework for music composition, which is the subject of this paper.

Overview

Computer-assisted composition offerings can be broadly divided into two groups: those aimed at the direct creation of sound, which usually treat time, pitch, and other musical parameters as continuous (e.g. PureData [Puckette 1997], Max/MSP [Puckette 2002] and SuperCollider [McCartney 2002]); and those that aimed at the creation of a traditional western score (e.g. *abjad* [Bača et al. 2015], OpenMusic [Bresson, Agon, and Assayag 2011], or music21 [Cuthbert and Ariza 2010]) which, due to the constraints

Features	Limitations
<ul style="list-style-type: none"> • Compose in continuous time, quantize to notation in discrete time • Coordinate multiple time streams using nestable clocks with variable tempi • Continuous (non-quantized) timing and pitch space; glissandi, volume envelopes and other continuous parameter curves • Playback using SoundFonts, MIDI stream, OSC, and user-defined implementations. • Notation via Lilypond or MusicXML • Easily understood syntax, with significant flexibility hidden behind sensible defaults 	<ul style="list-style-type: none"> • Not designed for direct sound synthesis • Note-based (broadly defined) • Generative toolkits, models of scales, pitch class sets, etc. not included in the main library • Timing is not sample-accurate (does not run on the audio thread) • Not explicitly designed for live-coding • No graphical front-end

Table 1. Key features and limitations of SCAMP.

imposed by notation, generally work with time in a discrete way.

SCAMP is designed to bridge these two worlds. Instead of a score, the composer interacts with an ensemble of virtual instruments, auditioning musical ideas in continuous time, outside of notational constraints. Once this result is deemed satisfactory, the user can then flexibly quantize the result and export it as music notation, in the form of either MusicXML (Good 2001) or LilyPond (Nienhuys and Nieuwenhuizen 2003), via the *abjad* library.

SCAMP manages time with a sophisticated system of clocks, capable of remaining tightly coordinated while following independent tempo curves. Moreover, nested relationships are possible; processes can spawn accelerating or decelerating subprocesses. Although the concept of tempo suggests beats and fractional units of time, all durations are in fact floating-point, defaulting to seconds if no tempo is indicated. When notation is needed, it can be generated in reference to any of the clocks used.

SCAMP is fundamentally note-event-based; however, notes are very flexible objects, capable of incorporating arbitrary continuous parameter curves. Microtonality, glissandi, and continuous volume envelopes are simple to achieve. In fact, so long as the playback method is designed to respond to them, arbitrary parameters can also be included and shaped in a continuous fashion.

It should be noted that SCAMP is not a synthesis engine, instead offloading playback to one of the following:

- SoundFont-based (Rossum 1995) playback via *pyFluidSynth* (Whitehead [2015] 2023). SCAMP automatically handles the pitch bends and channel management required for microtonality and glissandi.
- Playback via a MIDI stream to an external synthesizer. (SCAMP likewise handles pitchbends and microtonality.)
- Playback via OSC messages (Freed and Schmeder 2009) to an external synthesizer.

- User-defined playback implementations via the `PlaybackImplementation` interface.

The goal of the SCAMP framework is to act as a hub that connects the composer to other resources, while offering as open-ended a mental model as possible. As such, the core of the framework does not aim to offer utilities, such as generative toolkits or scalar/harmonic models, which align with particular aesthetic approaches. Such utilities may, however, be imported from third-party libraries, including the package *scamp_extensions*, maintained by the author.

The basic features and limitations of SCAMP are summarized in Table 1.

Designed for Flexibility

Figure 1 illustrates both the internal and external dependency structure of SCAMP. The packages in yellow represent external dependencies. Most of these dependencies exist to connect SCAMP to various forms of musical input and output: LilyPond notation via *abjad* (Bača et al. 2015), SoundFont interpretation and playback via *sf2utils* (Jolly 2023) and *pyFluidSynth*

(Whitehead [2015] 2023), MIDI input/output via *python-rtmidi* (Arndt [2015] 2023) and *midiutil* (Wirt [2016] 2022), and OSC input/output via *pythonosc* (Timothé [2013] 2023). In addition, SCAMP depends on the PEG parsing library *arpeggio* for the parsing of note properties expressions (“TextX/Arpeggio” [2012] 2023), and *pynput* for handling mouse and keyboard input (moses-palmer [2015] 2023).

The packages in blue comprise the SCAMP suite itself. The *scamp* package contains the main functionality of the suite, with time management (*clockblocks*), parameter shaping and control (*expenvelope*), and MusicXML export (*pymusicxml*) separated out into self-contained packages. Note that *clockblocks* relies on *expenvelope*, since the tempo curve of a clock derives from the *Envelope* class.

This modular structure was chosen for two key reasons:

- Easier maintenance: keeping different aspects of functionality separate keeps the scope of each component more manageable.
- Discardability and reusability: not all composers will need all of the

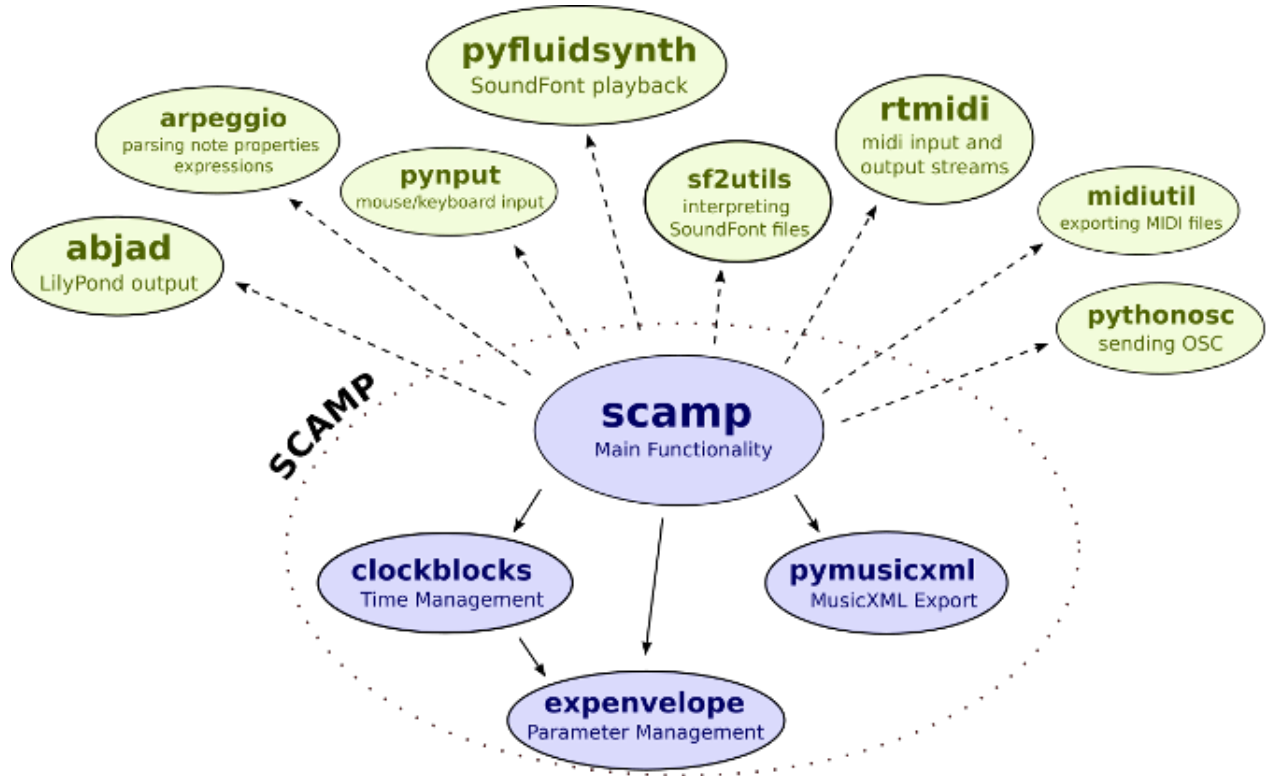


Figure 1. Internal and external dependencies of SCAMP.

functionality that SCAMP has to offer. For instance, some may be interested only in polyphonic tempo control or direct MusicXML export. By isolating these components from the rest of SCAMP, composers with different goals can potentially incorporate them into other frameworks.

This second point is an important one: every composer develops a unique workflow, reflecting their unique set of aesthetic concerns. For most, this ultimately means patching together a variety of tools. SCAMP’s modular structure allows composers to use (and perhaps repurpose) only that which is relevant to them.

One final aspect of SCAMP that provides enormous flexibility for the composer is that it is situated within the broader Python ecosystem. This affords:

- Easy access to a wide variety of libraries that can be adapted for musical purposes, such as data processing (e.g. *numpy*, *scipy*, *matplotlib*) and machine-learning (e.g. *tensorflow*, *scikit-learn*) toolkits.
- In particular, access to the many forms of data input and output available through both standard Python and third-party libraries.

Taken together, the goal of these design choices is to create a framework that is as adaptable as possible to the needs of different composers and compositions.

“Hello World”

As an introduction to the SCAMP API, we begin with a program that plays a short arpeggio:

```
# import the scamp namespace
from scamp import *
# construct a session object
s = Session()
# add a new violin part to the session
violin = s.new_part("Violin")
# looping through the MIDI pitches
# of a C major arpeggio...
for pitch in [60, 64, 67, 72]:
    # play each pitch sequentially
```

```
# with volume of 1 (full volume)
# and duration of half a beat
violin.play_note(pitch, 1, 0.5)
```

The result of this program will be the sound of a violin playing a C major arpeggio, at max volume, over the course of two seconds. The first step, after importing the SCAMP namespace, is to create a Session object. Most SCAMP programs will start in this way, as the Session object is the central hub through which most of the functionality of SCAMP flows. In the same way that a session in a DAW encompasses tracks, transport, and recording functionality, the Session object combines the functionality of an Ensemble, a master Clock, and a Transcriber (see Figure 2).

The Ensemble functionality of the Session object — i.e. its role as a host of the instruments being used — is employed in the third line, where a new violin part is created and stored in the violin variable. By default, new parts created in this way will play back via *pyFluidSynth* using a General MIDI Soundfont, with fuzzy string matching being used to find an appropriate preset based on the name of the part. Finally, the call to `play_note` takes three arguments: MIDI pitch (where 60 = Middle C), volume (on a scale from 0 to 1), and length in beats.

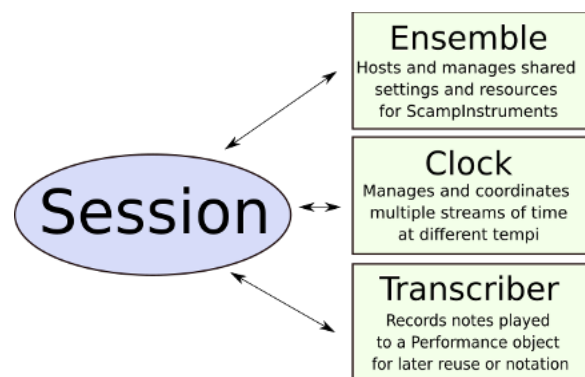


Figure 2. The functionality of a Session object.

Duration

One might reasonably ask at this point how long a beat is. As mentioned above, Session inherits from the Clock class, which is defined in *clockblocks*, a subpackage of SCAMP detailed in a separate paper (Evanstein 2019).

When a Session object (or Clock object more generally) is constructed, it becomes the default clock used on the current thread and is also assigned a default tempo of 60 BPM, or one beat per second. By way of these defaults, when play_note is called in above example, it looks to see which clock is operating on the current thread, finds the session (s) we created, and plays the note for 0.5 beats on that clock (which corresponds to 0.5 seconds at the default tempo of 60). In this way, the whole four-note arpeggio lasts two seconds.

We can play back the arpeggio faster or slower simply by setting the session's tempo attribute. For instance, if we add the following line directly after the second line in the above example, the arpeggio will last twice as long:

```
s.tempo = 30
```

Calls to play_note are blocking by default, not moving on to the next line until the note has finished. This conforms to musical expectations (playing a note takes time) and is a model of musical time similar to that used by the Euterpe language (Smoliar 1971). However, SCAMP also allows the user to start and end notes manually. In fact, internally, the function call violin.play_note(60, 0.7, 1.5) is essentially equivalent to:

```
# start playing a middle C with volume 0.7
# returns handle for further manipulation
note = violin.start_note(60, 0.7)
wait(1.5)
note.end()
```

Note that, like the play_note function, the wait function captures the current clock from context and takes the number of beats as an argument.

Finally, in addition to the options above, it is also possible to call play_note in a non-blocking manner by setting the “blocking” keyword argument to False. For instance, the following would play two notes, each lasting two seconds, that overlap by one second:

```
violin.play_note(60, 1, 2, blocking=False)
wait(1)
violin.play_note(64, 1, 2)
```

It is important to note that durations need not be of rational lengths. For example, after constructing the session and adding a violin part, the following would repeatedly loop through our arpeggio using random (floating point) durations between 0.5 and 1.5 seconds:

```
# import the random module from
# the python standard library
import random

while True: # loop forever
    for pitch in [60, 64, 67, 72]:
        # pick a duration between 0.5 to 1.5
        dur = random.uniform(0.5, 1.5)
        # play a note of that duration
        violin.play_note(pitch, 1, dur)
```

The default tempo of 60 BPM was chosen because it allows the composer to think in terms of durations in seconds if so desired. This exemplifies a guiding philosophy of the SCAMP framework: musical parameters are treated as continuous during the compositional process unless explicitly quantized. It is only later on, when converting the music to a score, that quantization becomes a necessity.

Generating Notation

To generate notation from the above examples, we use the third capacity of the Session object: its role as a Transcriber. The purpose of a Transcriber is to keep track of any notes played by the instruments that have registered with it, and to save the result as a Performance, which is essentially a note-event list, similar in nature to (and savable as) a MIDI file. This performance can then be quantized and converted into a Score, which is capable of saving either to LilyPond or MusicXML.¹

¹ The advantage of first transcribing the music as a Performance and then converting it to a Score is that the performance is free of notational constraints

and can be replayed, either in part or in whole, with different instruments or tempo settings.



Figure 3. Examples of notation generated from introductory examples. Snippets a) and b) result from steady note lengths. Snippets c) and d) result from random note lengths, with d) using a simpler QuantizationScheme.

To transcribe the first example above and convert it to music notation, we would do the following:

```
from scamp import *

s = Session()
violin = s.new_part("Violin")

# begin transcribing (defaults to using
# all instruments within the session)
s.start_transcribing()

for pitch in [60, 64, 67, 72]:
    violin.play_note(pitch, 1, 0.5)

# stop transcribing and save the
# note events as a performance
performance = s.stop_transcribing()

# quantize and convert the performance to
# a Score object and open it as a PDF
# (by default this is done via abjad)
performance.to_score().show()
```

This results in the notation shown in Figure 3a. When show is called, the score representation within SCAMP is converted to an *abjad* score, which then outputs and compiles LilyPond code and displays the result as a PDF. It is also possible to instead call show_xml, which uses *pymusicxml* to export a MusicXML document and opens the

result in a score editor (e.g. MuseScore, Sibelius, Finale).

By default, the to_score function quantizes the music to 4/4 time. If a different time signature is desired, one merely has to provide it as a keyword argument:

```
performance.to_score(
    time_signature="3/8"
).show()
```

This results in Figure 3b. A looping or non-looping list of time signatures can also be provided, as can a list of beats on which to place bar lines.

Quantization

Things get more interesting when we use random floating-point durations, since some quantization is required:

```
# ---- setup code omitted for brevity ----
s.start_transcribing()

for _ in range(2): # loop twice
    for pitch in [60, 64, 67, 72]:
        dur = random.uniform(0.5, 1.5)
        violin.play_note(pitch, 1, dur)

performance = s.stop_transcribing()
performance.to_score(
```

```
time_signature="3/4"
).show()
```

This results in Figure 3c. By default, divisors of the beat up to 8 are allowed, which may be more complex than desired. Simpler results can be achieved by using a custom QuantizationScheme that limits the max divisor to 4:

```
performance.to_score(
    QuantizationScheme.from_time_signature(
        "3/4", max_divisor=4
    )
).show()
```

This results in Figure 3d. Considerable customizability of QuantizationScheme is possible: not only is it possible to set the max divisor, but it is also possible to specify the degree of preference for simpler ratios, or even to provide a list of all allowed divisors.

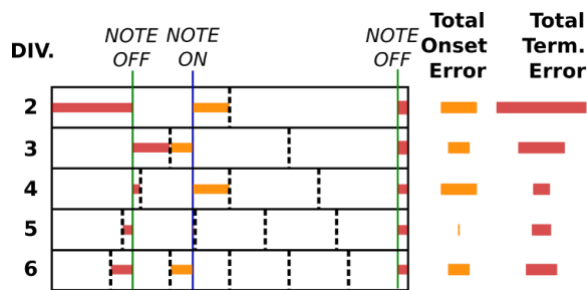


Figure 4. Illustration of the quantization process.

Moreover, while quantization is always done at the beat level, the beat length and divisor preferences can vary from beat to beat, according to the scheme imposed by the user.

As illustrated in Figure 4, the choice of quantization for each beat is made by comparing the (floating-point) note onset and termination times that fall within that beat to the grid formed by each divisor. The winning divisor is that which minimizes the (weighted) square error. To the degree that a preference for simple divisors is specified by the user, this square error is weighted by the *indigestibility* of the divisor as described by composer Clarence Barlow (2012). The user can also specify a relative weighting of onset vs. termination error.

One weakness of this approach is that it does not allow nested tuplet structures. A possible future development, therefore, would be the incorporation of the Q-Grid approach proposed by Nauert (1994), which allows for such structures, taking into account their degree of complexity.

Multi-Part Music

The above examples, for simplicity, were all single-part; however, writing multi-part music is straightforward:

```
from scamp import *
import random

# one way of setting an initial tempo
s = Session(tempo=100)
oboe = s.new_part("oboe")
bassoon = s.new_part("bassoon")

# define a function for the oboe part
def oboe_part():
    # play random notes until we have
    # passed beat 7 in the session
    while s.beat() < 7:
        pitch = int(random.uniform(67, 79))
        volume = random.uniform(0.5, 1)
        length = random.uniform(0.25, 1)
        oboe.play_note(pitch, volume, length)
    # end with a note of exactly the right
    # length to take us to the end of beat 8
    oboe.play_note(80, 1.0, 8 - s.beat())

# define a function for the bassoon part
def bassoon_part():
    # simply play quarter notes on random
    # pitches for 8 beats
    while s.beat() < 8:
        bassoon.play_note(
            random.randint(52, 59), 1, 1
        )

s.start_transcribing()
# start the oboe and bassoon parts as
# two parallel child processes
s.fork(oboe_part)
s.fork(bassoon_part)

# wait for forked processes to finish
```



Figure 5. Example of notated multi-part music in SCAMP, having been exported as MusicXML and imported into MuseScore.

```
s.wait_for_children_to_finish()
perf = s.stop_transcribing()
# here we call show_xml, which will open
# the result in, for instance, MuseScore
perf.to_score().show_xml()
```

This results in the notation seen in Figure 5. The fork method of the Session object, inherited from the *clockblocks* Clock class, takes a function as a parameter and runs it as a parallel child process. There is considerable flexibility here: child processes can spawn their own child processes, and so on. Unlike a naive approach to parallelism using Python’s built-in threading module, all processes forked in this way will remain tightly coordinated, with the clock also compensating for calculation time.

Multi-Tempo Music

One of the most powerful aspects of SCAMP’s approach to time-management is that each forked processes is associated with its own clock, and can have its own smoothly-varying tempo, as illustrated by this example:

```
from scamp import *

s = Session()
trumpet = s.new_part("trumpet")
trombone = s.new_part("trombone")

# Have the session as a whole speed up to
# 100 BPM over the first nine beats
s.set_tempo_target(100, 9)
```

```
# here we define a trumpet part to run as
# a child process.
def trumpet_part():
    # play eighth notes for three beats
    while s.beat() < 3:
        trumpet.play_note(67, 1, 0.5)
    # tell the clock for this child process
    # to slow down to 1/2 speed over six
    # beats in the parent process
    current_clock().set_rate_target(
        0.5, 6, duration_units="time"
    )
    # keep playing eighth notes until 12
    # beats pass in the parent process
    while s.beat() < 12:
        trumpet.play_note(67, 1, 0.5)

# Fork the trumpet part as a child process
# (It will be influenced both by its own
# tempo and that of the session as a whole)
s.fork(trumpet_part)
s.start_transcribing()

# Play quarter notes for 12 beats
while s.beat() < 12:
    trombone.play_note(60, 1, 1)

# Stop transcribing and show the result
perf = s.stop_transcribing()
perf.to_score(time_signature="3/4").show_xml()
```

This results in the notation shown in Figure 6a. Within the context of a session-wide acceleration from 60 BPM to 100 BPM, the clock for the trumpet part slows down to a rate of 0.5.²

² In SCAMP, “rate” and “tempo” are different methods for describing the same underlying property:

a tempo of 60 BPM corresponds to a rate of 1, and a tempo of 120 BPM corresponds to a rate of 2, etc.

a.)



b.)

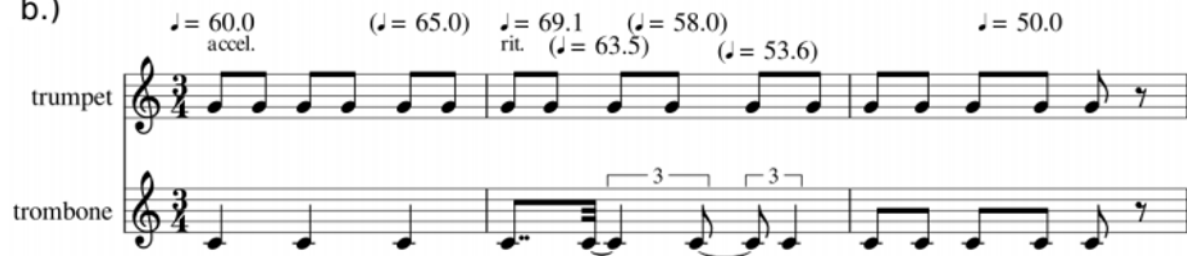


Figure 6. The same music quantized to two different clocks, having been exported as MusicXML and imported into MuseScore.

An important aspect of capability regarding polytempo music in SCAMP is that performances can be transcribed in reference to whichever clock is desired. For instance, in the example above, we can generate a score relative to the tempo of the trumpet part by replacing the corresponding lines with:

```
# fork returns the clock associated
# with the newly forked process
trumpet_clock = s.fork(trumpet_part)
s.start_transcribing(clock=trumpet_clock)
```

This results in the notation shown in Figure 6b.

Skipping Forward in Time

It may be that a composer, having devised an algorithmic process lasting 20 minutes, would like quickly test what this process sounds like at the 15-minute mark. In SCAMP, this is possible by simply calling a “fast-forward” method on the master clock (generally the Session object):

```
s.fast_forward_to_time(900)
```

Similar to the process of “nesting” in Euterpe (Smoliar 1971), this causes the program to execute as rapidly as possible up to the appointed

time, while skipping note playback calls. In this way, the program is in exactly the same state as it would have been had it run normally.

Envelopes and Continuous Parameters

The accelerandi and decelerandi in the section on “Multi-Tempo Music” are an example of the continuous shaping of a parameter over time, in this case tempo. Each clock manages its tempo using a TempoEnvelope, a subclass of Envelope, the piecewise-exponential envelope class defined in the separate package *expenvelope*.

Instances of Envelope are used broadly in SCAMP when any musical parameter is changing over time, such as in glissandi or dynamic playback. They can also be employed by the composer for the larger-scale shaping of algorithmic processes.

A simple, evenly-spaced Envelope can be constructed and plotted as follows:

```
e = Envelope.from_levels([60, 72, 66, 70])
e.show_plot("Evenly Spaced")
```

This results in the plot shown in Figure 7 below (plotting is done using matplotlib).

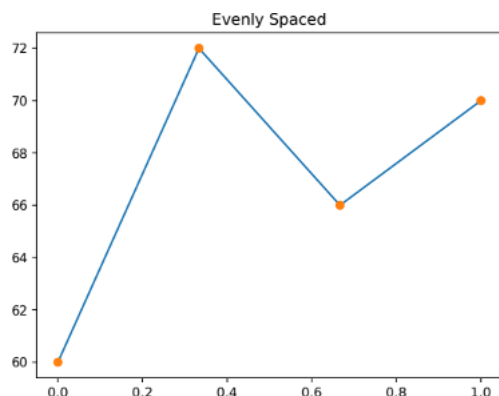


Figure 7. A simple Envelope.

Each segment of the envelope can furthermore be assigned a duration and a shaping attribute:

```
e = Envelope.from_levels_and_durations(
    [60, 72, 66, 70], [3, 1, 1],
    curve_shapes=[2, -2, -2])
e.show_plot("Uneven with Shaping")
```

This results in the plot shown in Figure 8. The values in the curve shapes attribute range from $-\infty$ to ∞ , with 0 being linear, negative values corresponding to early change, and positive values corresponding to late change.

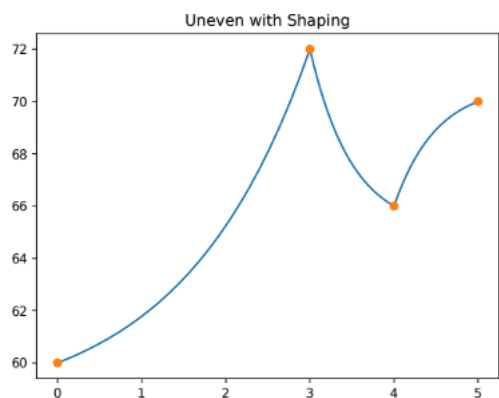


Figure 8. An Envelope with varied segment duration and shape.

The Envelope class is equipped with a range of utilities, including integration (needed by TempoEnvelope to determine the length in time corresponding to a certain number of beats at a changing tempo), approximation of arbitrary functions, and mathematical operations, such as addition, multiplication, division, and concatenation.

Glissandi

The Envelope in the example above can be interpreted as a glissando by simply passing it to the pitch argument of the play note function:

```
instrument.play_note(e, 1.0, 4)
```

This results in the notation shown in Figure 9. Note that, since the note is shorter than the duration of the envelope, we only use the first four beats of the envelope.



Figure 9. Glissando resulting from passing an Envelope as the pitch argument to play_note.

Since directly creating an Envelope object each time a glissando is desired would be cumbersome, as a shortcut, any list given as the pitch argument to play_note is interpreted as an envelope:

```
# results in evenly spaced glissando
instrument.play_note([60, 70, 55], 1.0, 4)
```

If segment durations and curve shapes are desired, a list consisting of [values, durations, curve shapes] may be used:

```
# same as envelope "e" above
instrument.play_note(
    [[60, 72, 66, 70], [3, 1, 1], [2, -2, -2]], 1.0, 4
)
```

When envelopes are creating using list shorthand like this, by default SCAMP *does* resize the envelope to the length of the note. This example therefore produces the notation found in Figure 10 below, in which quintuplets result from normalizing a 5-beat envelope to a 4-beat note duration.



Figure 10. Glissando resulting from passing a list of [values, durations, curve shapes] to the pitch argument of `play_note`.

Microtonality

The reader may have noticed quarter-tone accidentals in Figure 10. Microtonality in SCAMP is as simple as using floating-point values for pitch. When using MIDI or SoundFont-based playback (which also utilizes the MIDI protocol), SCAMP internally handles all pitchbend messages. Since these messages are channel-wide, notes that change pitch (or might change pitch) are placed on separate channels, with channels being recycled automatically.

If more specificity of pitch is desired in the notation, the user may turn on microtonal pitch annotations in the engraving_settings:

```
engraving_settings.\
  show_microtonal_annotations = True
piano.play_chord([62.7, 71.3], 1.0, 1)
piano.play_chord([65.2, 70.9], 1.0, 1)
piano.play_chord([71.5, 74.3], 1.0, 1)
```

This results in the notation shown in Figure 11.

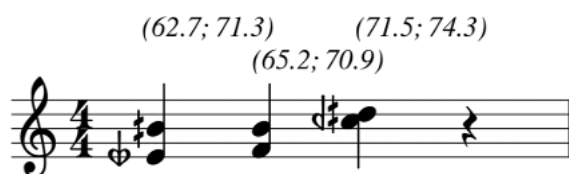


Figure 11. Microtonal pitch annotations.

Dynamics

An Envelope object (or its corresponding list shorthand) may also be used to apply a continuous volume curve to a note. For instance, a forte-piano-crescendo can be achieved in the following way:

```
fpcresc = Envelope.from_levels_and_durations(
  [1.0, 0.3, 1.0], [0.3, 3.7])
```

```
)
piano.play_note(60, fpcresc, 4)
```

Since dynamic notation is more subjective than pitch notation, arbitrary parameter curves like this are not currently notated. However, dynamics may be added to the score via the note properties argument (see below).

Anatomy of a `play_note` Call

As mentioned above, `play_note` internally breaks down into separate calls to `start_note`, `wait`, and `end`. With continuous changes of parameter, further calls are involved. For instance, consider the following:

```
# glissando and crescendo over 4 beats
instrument.play_note([60, 70, 55], [0.3, 1.0], 4)
```

This is equivalent to the following separate calls:

```
note = instrument.start_note(60, 0.3)
# start a 2 beat pitch-change ramp
note.change_pitch(70, 2)
# start a 4 beat volume-change ramp
note.change_volume(1.0, 4)
wait(2)
# start the second pitch ramp
note.change_pitch(55, 2)
wait(2)
note.end()
```

While a call to `play_note` is clearly more succinct, there are situations where the length and course of the note are not known in advance, and which are therefore better suited to the second approach. This is especially true with the incorporation of live input.

Playback Implementations

In the preceding examples, all playback is done via `pyFluidSynth` using a default General MIDI soundfont. However, one of the major flexibilities built into SCAMP is its variety of available playback implementations, as well as the ability to define custom implementations.

Consider the following setup:

```
s = Session()
```



```

# Calling "new_part" results in a default
# SoundfontPlaybackImplementation, and
# add_streaming_midi_playback gives this a
# MIDISreamPlaybackImplementation as well
# (here, port 2 is used for output)
piano = s.new_part("piano").\
    add_streaming_midi_playback(2)

# "new_osc_part" initializes the instrument
# with an OSCPlaybackImplementation
synth = s.new_osc_part("synth",
    ip_address="127.0.0.1", port=57120)

# Calling "new_silent_part" results in an
# instrument with no PlaybackImplementation
silent = s.new_silent_part("silent")

```

As illustrated by Figure 12, session `s` will contain an ensemble of three `ScampInstruments` named “piano”, “synth”, and “silent”. The “piano” instrument is created with the default `SoundfontPlaybackImplementation` to which we add a `MIDISreamPlaybackImplementation`. By calling `new_osc_part`, the “synth” instrument is created with an `OSCPlaybackImplementation` instead of the default. Finally, the “silent” instrument is created with no `PlaybackImplementation`.

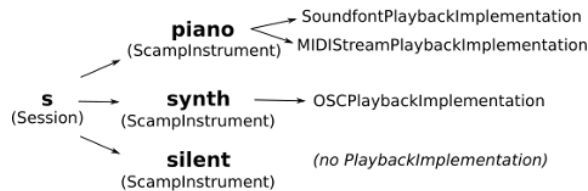


Figure 12. Playback implementations.

When a note is played on an instrument, each of its `PlaybackImplementations` is told to start a note, do any pitch/volume/parameter changes, and then end the note at the appropriate time. Thus, when a note is played using “piano” above, MIDI messages will be sent to `fluidsynth` to generate sound, as well as sent out of port 2. When a note is played using “synth”, appropriate OSC messages (with address patterns like “synth/start note” and “synth/change pitch”) will be sent to port 57120. Finally, when a note is played using “silent”, nothing will happen, since it has no `PlaybackImplementations`.

The Note Properties Argument

The `play_note` function in SCAMP can take a fourth, optional `properties` argument. This argument acts as a wildcard, accepting a properly formatted string or a dictionary which specifies a variety of playback and notation details:

```

# passing comma-separated key value pairs
piano.play_note(60, 1, 1,
    "notehead: x, articulation: staccato")
# just a value; SCAMP infers that you
# are referring to articulation and dynamics
piano.play_note(60, 1, 1, "staccato, f")
# "play_chord" can take multiple noteheads
piano.play_chord([60, 65], 1, 1,
    "noteheads: x/circle-x")
# passing a dictionary also possible
piano.play_note(60, 1, 1,
    {"articulations": ["tenuto", "accent"]})

```

This produces the notation shown in Figure 13. In addition to modifying the notation, articulations like `staccato` and `tenuto` will affect playback duration, and accents will affect volume. Exactly how (if at all) these notations affect playback is adjustable via SCAMP’s playback settings.



Figure 13. Illustration of notation resulting from various properties.

The `properties` argument may also be used to determine note spelling, either directly or by suggesting a key:

```

# spell the note with a sharp
piano.play_note(63, 1.0, 1.0, "#")
# play some notes in F minor
pitches = [65, 63, 61, 60]
durations = [1/3] * 3 + [1]
for pitch, dur in note_info:
    piano.play_note(pitch, 1.0, dur,

```

"key: F minor")

This results in the notation shown in Figure 14.



Figure 14. Notes spelled both directly and by specifying a key context.

The properties argument can also be used to define aspects of playback other than pitch and volume, if implemented by the PlaybackImplementation. For instance, with a properly configured OSC receiver, the following will produce a 3 beat glissando with increasing vibrato frequency and depth:

```
# on the other end, an OSC receiver is
# setup to play notes that take vibrato
# OSC messages as well as the usual
vib = s.new_osc_part("vibrato_inst", 57120)

# any property entries starting or ending
# with "param" will be treated as extra
# playback parameters
vib.play_note(
  [72, 64], 0.5, 3,
  {"param_vib_depth": [0.5, 2],
   "param_vib_freq": [3, 8]}
)
```

Any properties starting with the prefix "param_" are interpreted as additional playback parameters. In the case of an OSCPlaybackImplementation this results in outgoing OSC messages with address patterns "vibrato_inst/change_parameter/vib_depth" and "vibrato_inst/change_parameter/vib_freq."

Directions for Further Development

One area for future development is in the relationship between playback and notation of dynamics and other user-defined playback parameters. Currently, SCAMP allows for the notation of both tiered dynamics (e.g. "p", "f") and dynamic spans (hairpins, crescendi);

however, these dynamics do not affect playback. If and how best to do this is an open question.

In addition, the possibilities of graphical notation for representing continuous pitch, volume and other parameter curves is being explored. Since the Performance class retains continuous-time musical data, alternate mappings to different kinds of score are possible. Promising experiments have been conducted with converting Performance objects to graphical notations in the form of SVG files.

This points to the value of expanding the options for translating musical data. Currently, a Performance can be converted to a Score, but not vice-versa. Likewise, a Score can be translated to an abjad/LilyPond or MusicXML representation, but not vice-versa. One development goal is to enable two-way translation between all of these representations.

In terms of playback, a medium-term goal is to offer SFZ-file-based playback using LinuxSampler ("SFZ File Format" 2017). As a modern, open standard for sampled sound playback, SFZ files offer exactly the kind of flexibility that SCAMP is designed to afford. Another playback goal is to enable certain note properties, such as "pizzicato", to trigger a preset switch, much as they do in professional score-writing software, such as Sibelius.

One last area of development is in the domain of live input. Although SCAMP was not originally designed for live coding, the ability to receive and respond to real-time MIDI and OSC input has since been added. However, for this capability to be truly stable, the system of clocks will need to be rewritten, a task which is currently underway.

Conclusions

Although these planned developments are worthy goals, the true value of SCAMP is in the music it makes possible. One such effort in my own compositional practice has been a collaboration with Stanford University Biologists Elizabeth Hadly and Allison Stegner, in which I used SCAMP to create a composition based on data from a sediment core taken at Searsville Reservoir, a site of global significance for defining the Anthropocene (Evanstein 2022).

Beyond my own work, I have taken steps to grow awareness of SCAMP and develop a broad and varied user-base. This has included the creation of an online forum, regular workshops and seminars, and detailed online documentation and tutorials. Through these efforts, users around the world have begun to use SCAMP and create compositions with it.

SCAMP was developed and re-designed over a period of many years, based on my own compositional practice. Its true potential, however, can only be discovered through interaction with other composers, approaching it with potentially radically different compositional aims.

References

Arndt, Christopher. (2015) 2023. “SpotlightKid/Python-Rtmidi.” Python. <https://github.com/SpotlightKid/python-rtmidi>.

Bača, Trevor, Josiah Oberholtzer, Jeffrey Treviño, and Víctor Adán. 2015. “Abjad: An Open-Source Software System for Formalized Score Control.” In *Proceedings of the First International Conference on Technologies for Music Notation and Representation -- TENOR’15*. <https://doi.org/10.4135/9781412974615.n18>.

Barlow, Clarence. 2012. *On Musiquantics: Von Der Musiquantenlehre Translated*. Royal Conservatory The Hague.

Bresson, Jean, Carlos Agon, and Gérard Assayag. 2011. “OpenMusic: Visual Programming Environment for Music Composition, Analysis and Research.” In *Proceedings of the 19th ACM International Conference on Multimedia - MM ’11*, 743. Scottsdale, Arizona, USA: ACM Press. <https://doi.org/10.1145/2072298.2072434>.

Cuthbert, Michael Scott, and Christopher Ariza. 2010. “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.” In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*. International Society for Music Information Retrieval. <https://dspac.mit.edu/handle/1721.1/84963>.

Evanstein, Marc. 2019. “Clockblocks: A Pure-Python Library for Controlling Musical Time.” In *Proceedings of the 2019 International Computer Music Conference*. New York, New York, USA.

Evanstein, Marc. 2022. “Searsville 1891-2015.” 2022. <http://marcevanstein.com/Compositions/Searsville/Searsville.html>.

Freed, Adrian, and Andy Schmeder. 2009. “Features and Future of Open Sound Control Version 1.1 for NIME,” 5.

Good, Michael. 2001. “MusicXML: An Internet-Friendly Format for Sheet Music.” In *Proceedings of XML*. Boston, Massachusetts, USA.

Jolly, Olivier. 2023. “Sf2utils · GitLab.” GitLab. 2023. <https://gitlab.com/zeograd/sf2utils>.

McCartney, James. 2002. “Rethinking the Computer Music Language: SuperCollider.” *Computer Music Journal* 26 (4): 61–68. <https://doi.org/10.1162/014892602320991383>.

moses-palmer. (2015) 2023. “Pynput.” Python. <https://github.com/moses-palmer/pynput>.

Nauert, Paul. 1994. “A Theory of Complexity to Constrain the Approximation of Arbitrary Sequences of Timepoints.” *Perspectives of New Music* 32 (2): 226–63. <https://doi.org/10.2307/833610>.

Nienhuys, Han-Wen, and Jan Nieuwenhuizen. 2003. “LilyPond, a System for Automated Music Engraving.” *Proceedings of the XIV Colloquium on Musical Informatics*, 6.

Puckette, Miller. 1997. “Pure Data: Another Integrated Computer Music Environment.” 1997. <https://puredata.info/docs/articles/puredata1997>.

Puckette, Miller. 2002. “Max at Seventeen.” *Computer Music Journal* 26 (4): 31–43.

Rossum, Dave. 1995. “The SoundFont® 2.0 File Format,” 9.

“SFZ File Format.” 2017. May 5, 2017. <http://doc.linuxsampler.org/sfz/>.

Smoliar, Stephen W. 1971. "A Parallel Processing Model of Musical Structures," September. <https://dspace.mit.edu/handle/1721.1/6915>.

"TextX/Arpeggio." (2012) 2023. Python. textX. <https://github.com/textX/Arpeggio>.

Timoth  . (2013) 2023. "Python-Osc." Python. <https://github.com/attwad/python-osc>.

Whitehead, Nathan. (2015) 2023. "PyFluidSynth." Python. <https://github.com/nwhitehead/pyfluidsynth>.

Wirt, Mark Conway. (2016) 2022. "MIDIUtil." Python. <https://github.com/MarkCWirt/MIDIUtil>.

Kinetic Relationships in Visual Music Composition: Interactive Changes Over Time

Michael Rhoades

Ball State University

mrhoades@perceptionfactory.com

Introduction

Today, visual music is generally a creative expression poised at the confluence of temporally based visual art and music. Essentially, it is comprised of oscillating amalgams of light and air. Here the blending of two mediums forms a third, which includes aspects of each and yet is unique in comparison to either in isolation. When occupying a mutual space/time without the intention of direct interrelationship on the part of the composer, these atomic constituents nonetheless create higher-level formations. Human tendencies toward apophenia, instinctively perceiving and creating relationships where none might exist, act as cognitive influences. However, for composers intending to exert creative influence toward such molecular formations, a wealth of potential exists. Here we will investigate several possibilities in an attempt to apprehend them knowing that, ultimately, they will dissolve into a singular expression in which they are indistinguishably inherent.

Two key and elemental structural factors in the composition of visual music reside in areas of kinetic and transitional relationships. It is posited here that mastery of these components engenders formal progressions that establish and maintain compositional cohesion. Further, their judicious and deliberate implementation provides opportunities toward the unification of these otherwise unrelated compositional elements. It is within this framework that further investigation into these relationships is exposed as fundamental to the composition of visual music. In this writing we will delve into the kinetic relationships leaving the study of transitions to another time.

The term *kinesis* refers to movement. Movement equals change over time, which is intrinsic in temporally-based art. Thomas Wilfred and Oscar Fischinger, generally considered the fathers of visual music, approached kinetic

relationships in varied manners. Wilfred was primarily interested in relational movement between visual objects. His work normally did not include an audio element (Orgeman et al. 2018). Oscar Fischinger primarily explored kinetic relationships between visual and musical objects with an emphasis on the former (Keefer and Guldemon 2013). Fischinger's approach is the subject of this writing, however perspectives described are equally pertinent considering Wilfred's.

Kinesis, as witnessed in visual objects, is relatively straightforward. Objects seen as traversing the field of view from coordinates (x, y, z) to coordinates (x₁, y₁, z₁) are naturally ascribed characteristics of movement. However, in reference to musical expression, the notion of kinesis may not be readily apparent. To begin, sounds move in terms of waves of air emanating from a source outward, in varying degrees, in multifarious directions. Sound is the vibration (movement) of air. Further, considering that movement equals change over time, it is obvious that a musical composition changes over the course of its duration. Finally, and perhaps more directly, working with two or more channels of audio diffusion, the perception of the location of sound events can be established and may traverse the listening space to other locations via varying paths (Rhoades 2021).

In this study, eleven forms of kinetic relationships between audio and visual elements are explored. Though they may often overlap or may be intermingled, here they are teased apart and investigated in isolation in order to apprehend individuated approaches. Below is a list of the relationships that will be considered. It is not intended as an exhaustive list but instead as a point of departure into the reader's own investigation.

- Disparity
- Unity

- Rhythmic
- Timbre
- Harmonic
- Contrapuntal
- Spatial
- Energetic
- Momentum
- Speed
- Opposition

In addition to a written description of each example cited in the text, included in each section is a link to relevant audio/visual clips. This approach is derived from the practice-based perspective owing that direct experience, beyond often being an effective learning aid, is a powerful communicator of ideas. The examples utilized are derived from my own compositions. This is in part due to an intimate familiarity with the work and further to ensure there are no intellectual property issues involved. It can be realized that when experiencing the examples, they might be considered subjectively derived. Certainly, as human beings, we often experience quite different phenomena even when subjected to the same stimuli. It is therefore requested that if you do not see that which is being described immediately, take time to consider it. There will be a gain of apprehension if the extra effort is extended as necessary.

One last consideration before proceeding is that precursors to the advent of visual music, such as the work of the painter Wassily Kandinsky, were attempts to make static visual art attain musical and temporal characteristics. This approach to visual art translated through into the beginnings of experimental cinema, in which static visual art could be temporally focused (Abbado 2017). Therefore, it is not surprising that here we will consider and compare several attributes of the visual movement from a theoretically based musical perspective.

Disparity – Disconnection

A sense of disconnection between audio and visual elements occurs when they appear to act independently in terms of movement. This relationship is prominent in visual music due, at least in part, to the ease of implementation. It is safe and easy to implement from an editing

perspective since there is no concern with connecting audio and visual movements. Often overused, it lacks the characteristics of an approach intended to provide formal qualities that guide the audience through the ebb and flow of the composition. In this manner, it can quickly become monotonous lacking interest fostered by variety and surprise. From another perspective, when the audio and visual unfold in the same spatiotemporal dimension, there exists a relational connection formed by the propensity of human beings to organize phenomena into comprehensible, yet perhaps limited, patterns. As mentioned in the introduction, apophenia can be a major source of sense-making in this paradigm.

As opposed to implementing kinetic disparity throughout an entire composition, it can be perhaps beneficially employed effectively in specific instances. For instance, it can provide a sense of release from tensions during which it allows the audience to gain a sense of emotional equilibrium... a relaxing if you will. Disparity between audio and visual elements can be inserted as a background movement to provide additional interest and dimension to a scene without detracting from the main material, which might be interacting in other manners. It can additionally provide contrast to a section during which the audio and visual are tightly aligned. Finally, it can provide a segue between themes, phrases, or movements. The suggestion here is, as with herbs in cooking, it be used sparingly and wisely.

To experience an example of kinetic disparity, watch and listen to the audio/visual file linked below (A/V Example 1). Please keep in mind that though a section might exhibit disparate characteristics, other characteristics might also be exhibited.

[Audiovisual Example 1. Disparity \(1 Disparity.mp4\)](#)

1:1 – Unity

1:1 kinetic relationships between audio and visual elements in a composition could be termed one of unity. Here, the audio and the visual work in tight unison. Every movement of the visual is accompanied by directly related, and recognizable, movements in the audio. Again,

this relationship can quickly become quite monotonous if left unchecked. Oscar Fischinger's work often employed this approach. Though often a technological feat to achieve, it becomes easy for an audience to anticipate. This eliminates the element of surprise that characterizes interest in nearly any creative endeavor.

However, similar to the disparate relationships, employed judiciously, it creates strong moments of connectedness providing "landing spots" for an audience thus allowing for cohesion in the composition. Such relationships, paired with others, can be utilized to set up and fulfill or defy audience expectations. It can provide moments of stability in moments of uncertainty. It can also be used very subtly. As opposed to the rigid unity of an army marching, it can be used in a manner that maintains a loose unity. This subtlety perhaps requires a maturity on the part of artists skilled with their tools and with their sense of creativity. With some work it can be accomplished and is enjoyable to experience.

A/V Example 2 contains four clips that span the gaps between tight unity and unity of a more subtle nature.

[Audiovisual Example 2. Unity \(2_Unity.mp4\).](#)

Rhythmic Qualities

The rhythmic qualities of a visual music composition are achieved in numerous ways. There are the obvious punctuations, accents, and patterns presented in the music via typical rhythm instruments such as drums. Beyond those the visual music medium presents opportunities for rhythmic relationships such as successions of cuts in the visual flow or accented events in musical progressions. Interrelating rhythmical patterns affords audience interest on several strata. These patterns may be perceivable in the audio or visual separately, together, or in an exchange wherein each provides aspects of an overall rhythmic structure.

As with the unity type relationships, the rhythmic patterns can occur in a 1:1, tightly synchronized, manner, which is fairly straightforward. This approach can be used to accentuate major rhythmic passages in the composition. Somewhat isolated punctuations

provide points of unison that allow an audience to easily experience a connective link between the audio and visual. When used sporadically, they provide landing points in a composition. Let's experience an example of a tight synchronization approach (A/V Example 3a).

[Audiovisual Example 3a. Rhythmic \(3_Rhythmic_00.mp4\).](#)

Going deeper, variations to this approach and deviations from it can elevate a composition to a higher level of complexity and can provide a timeless quality to the work. In the next example (A/V Example 3b), the visual provides for rhythmic punctuations that the music only infers. Thus, the visual rhythms become an aspect of the auditory experience and vice versa. This is just one example of myriad possible rhythmic relationships in visual music. Rhythmic syncopations, simplifications, and exchanges provide for imaginative experiences.

[Audiovisual Example 3b. Rhythmic \(3_Rhythmic_01.mp4\).](#)

Timbral – Cross Associations

Timbre is a mysterious quality that is difficult to define. Webster defines it as "the character of a musical sound or voice as distinct from its pitch and intensity. Here we might simplify this by calling it the tone of one or more sounds. Both of these descriptions lack clarity, yet we understand differences in timbre when we hear them. Typically, the term "timbre" is used as a descriptor in relation to sound and music. However, when considering visual music compositional relationships, timbral qualities might be generalized to include visual material as well.

The union of the audio and the visual establishes the overarching timbre of a visual music composition. It could be considered that the composer establishes the timbre of a composition rather than it being inherent to it. For instance, watch the following example that does not include sound (A/V Example 4a).

[Audiovisual Example 4a. Timbral \(4_Timbral_00a.mp4\).](#)

Now, imagine what the sound that accompanies this segment might consist of. Of course, the audio decided upon by the composer greatly influences the perception of the visual and vice versa. It is in establishing these types of cross associations that creates the timbre of the composition. Once established by the composer, this relationship becomes a formal element, a thematic relationship that is varied and extended throughout a composition. Let's watch the previous example, this time, with audio (A/V Example 4b).

[Audiovisual Example 4b. *Timbral*](#)
[\(4 *Timbral* 00b.mp4\).](#)

Here we are interested in how the tone of the sound and the tone of the visual are in sync as they change over time. The first section of the next example exposes the original thematic material (A/V Example 4c). There the composer establishes the tone of the composition through the connection of the audio with the visual. The second section presents a variation to the original theme in both the audio and visual (A/V Example 4d). It utilizes the elements of each yet presents them in a varied yet related manner. The third and fourth sections take the variation further from the original theme and, yet again, are perceptually related (A/V Example 4e). Notice how the tones of each section change and how this occurs in changes to both the audio and the visual.

[Audiovisual Example 4c. *Timbral*](#)
[\(4 *Timbral* 001.mp4\).](#)

[Audiovisual Example 4d. *Timbral*](#)
[\(4 *Timbral* 002.mp4\).](#)

[Audiovisual Example 4e. *Timbral*](#)
[\(4 *Timbral* 003.mp4\).](#)

Other similar implementations of timbral relationships might include subjective emotional responses such as somberness, heaviness, excitement, happiness... These perhaps describe an emotional timbre and are presented as a combination of the audio and visual material.

Harmonic / Complimentary

Harmony, whether in music or myriad other forms, occurs when there is an interaction in a complimentary manner. For example, two people dancing or perhaps walking hand in hand are usually acting in harmony. One might consider geese flying in a V formation a form of harmonic interaction. In traditional music notation, notes that occur simultaneously... that are stacked in numerical relationship on the musical staff, are usually considered harmonically related.

In the following example, the white hazy objects in the foreground seem to be acting independently of the music. However, the colored shapes in the background could be considered to be acting in a loose harmony with the music. This is an example of harmony and disparity interacting within the same scene (A/V Example 5).

[Audiovisual Example 5. *Harmony*](#)
[\(5 *Harmony*.mp4\).](#)

Contrapuntal / Interactive

Counterpoint is a very interesting interaction that provides for excellent formal material. Take a moment and think about it. How would you describe counterpoint? Musically speaking, counterpoint could be described as two or more, perhaps related and yet different, melodies occurring in the same time frame. Generally, two people living in the same house, living two separate yet congruent lives, could be considered to be living in counterpoint with one another.

J.S. Bach was, in many ways, a quintessential master of counterpoint. For example, his "Continuo" bass line in his fugues often acted in counterpoint to the rest of the improvised compositional elements. Often in his work, two or more melodic lines were played simultaneously achieving a tight contrapuntal interaction.

In the following example, we are interested in observing that the audio and visual act in a contrapuntal manner: there are separate lines of interaction, yet each occurs in relatively the same time frame (A/V Example 6). They are complementary and yet they do not occur in sync with regard to timing. Nonetheless, the timing of

each is clearly taking place in relation to the overall phrasing.

[Audiovisual Example 6. *Contrapuntal* \(6 *Contrapuntal.mp4*\).](#)

Spatialization / Perceived Location

Musical spatialization refers to the perceived location of a sound event. This is achieved by several methods, each beyond the scope of this text. Suffice it to say that with few notable exceptions, the perception of the location of a sound is accomplished through amplitude panning or wavefield interference patterns (Rhoades 2021). I commonly work with 32 audio channels, which facilitates the spatialization of sound to a high degree of granularity. However, since this is possible to a very limited degree with stereo, the example below will provide a limited idea of the intention involved. Here we generalize the term “spatialization” to include the perceived location of visual events as well.

The perceived locations of audio and visual events are particularly interesting in a visual music composition. When operating in synchronization the effect is obvious and powerful. Imagine a visual object, existing in a 2D space. It moves from right to left and back again across the field of view. As it does a musical object is perceived to move in synchronization with it. The viewer’s eyes and ears now expect this as it becomes more prominent through repetition... the two events become linked in the viewer’s perspective. When it is altered or done away with altogether the surprise of the sudden contrast is dynamic.

In the first section of this example, the tendency of the audio and the visual is to move from left to right (A/V Example 7). Near the end, the correlation is much looser except at the very end where it once again syncs up. Experimenting with the spatial relationships between the audio and visual can pull a composition together into a singular expression and it can act as a guide for the viewer’s perception.

[Audiovisual Example 7. *Spatial* \(7 *Spatial.mp4*\).](#)

Energy / Intensity

How might we describe the attributes of energy or intensity of a visual music phrase or set of phrases? We might describe them as brightness verses darkness, force verses release, power verses weakness, or strength verses gentleness. It is difficult to describe beyond allusion yet we know it when we experience it.

In the following example, notice how the intensity shifts as the scene changes at about halfway through (A/V Example 8). Both the auditory and the visual reflect the intensity of each section. The first may be interpreted as “powerful driving” the second as “gentle undulations”.

Let’s see if you agree.

[Audiovisual Example 8. *Energy* \(8 *Energy.mp4*\).](#)

Does it work here? Is it obvious? This type of audio/visual unification can be used to amplify the effects of intensity in a scene. As seen here it can emphasize a shift in intensity. As can be witnessed in this example, it causes a feeling of tension and release.

Momentum / Temporal Energetic Contrast

Momentum is one of my favorite subjects. Consider for a moment, what is momentum? For this study, it could be described as the changes in the speed of object movement, which occur in an exponential manner. Perhaps the change occurs very slowly at first and then suddenly increases very quickly as the knee bend of the curve is reached in time. Look at the diagram below for an example of such an exponential function over time (Figure 1). Notice that the Y-axis values increase very slowly from 0 to 2 seconds. The knee bend occurs at approximately 4 seconds at which time the increase in the Y values begin increasing at a very rapid pace. This exponential increase in speed of an object’s movement takes on the characteristic of momentum.

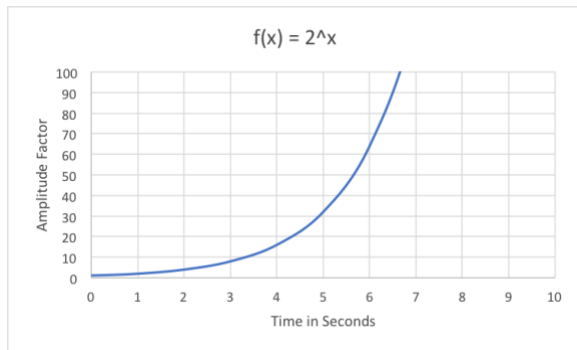


Figure 1. *A graph depicting an exponential increase in value over time.*

Reverse exponential motion is also a possibility. This would represent a shift in speed that is very fast at first and then, when the knee bend of the curve is reached, begins to rapidly slow.

When this effect is experienced in a composition, it takes on a very visceral quality... one can feel it physically. It is a very powerful effect and used as a compositional element it can be profoundly captivating. Whether, linked with objects in contrast with one another, in conjunction with other kinetic attributes, or implemented on an entire scene, shifts in energetic actions such as this are extremely desirable compositional and formal elements. See if you can notice the exponential shift in the following examples (A/V Example 9). It is unmistakable.

[Audiovisual Example 9. Momentum \(9_Momentum.mp4\).](#)

Speed – Rate of Compositional Progression

What is intended by the rate of compositional change? One example comes from traditional musical terminology, which is that of *accelerando* and *decelerando*. These terms designate a predominately linear increase or decrease, generally in phrasing, over time. Similarly, the terms *crescendo* and *decrescendo* may designate a linear increase or decrease in perceived loudness over time. Another feature of rate of compositional progression is in the realm of static verses dynamic (audio or visual objects moving or sitting still). Figure 2 depicts a linear increase in value over time.

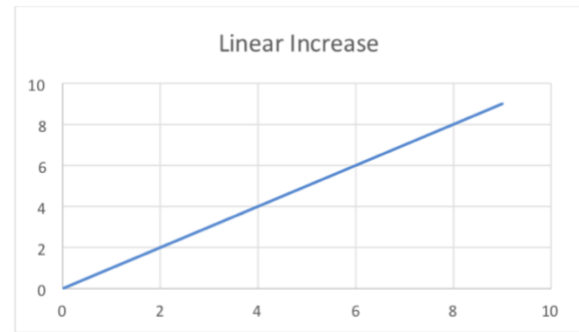


Figure 2. *A graph depicting a linear increase in value over time.*

Kinetic relationships between these types of rates of change between audio and visual elements can occur in sync or in inverse relationships. It can be quite subtle or overt. In the following example, notice the end of the clip where the speed of the visual decreases and the audio increases (A/V Example 10). It is perhaps subtle but watch and listen carefully.

[Audiovisual Example 10. Speed \(10_Speed.mp4\).](#)

Opposition / Push Pull

This last example is when the audio and the visual move in perceptible opposition. The feeling derived from this type of relationship is that the audio and visual seem to push or pull one another. The cause of this perception is often related to one aspect occurring slightly out of sync with the other. When the audio and visual phrasings seem related in some manner and yet one occurs sooner than the other, the perception of this pushing or pulling takes place.

It is quite a dramatic effect.

[Audiovisual Example 11. Opposition \(11_Opposition.mp4\).](#)

Conclusion

In this paper we have briefly considered the notion of kinetic relationships as they refer to the manners in which elements in a visual music composition interact in terms of movement. Similar to more common cinematic elements of transition, such as wipes, cuts, fades, washes, dissolves, and etc., kinetic elements contribute

toward a composition that is formally interesting as opposed to one that is dull, tedious, or monotonous. They provide for instantiations of tensions and release, temporal flow, and overall formal development of compositional themes and elements. This is especially important when working with abstract or non-objective compositions.

These kinetic relationships rarely exist in isolation: they usually overlap and/or interact within the same event and within the same scene. Take the afore mentioned example of a crescendo. If it is a very linear increase in amplitude, it is a crescendo moving in isolation, which we defined as an aspect of speed. However, if it occurs in an exponential manner, besides being strictly an aspect of speed, it will portray the properties of momentum as well. Such is the nature of these interactions when implemented in conjunction. Further, these interactions may occur between musical and visual elements or simply between one or the other. For instance, as the musical elements are acting in opposition to one another, the visual elements may be acting in unison or chorus. The combination of these two disparate actions forms a third that is nearly impossible to describe. The complexity and diversity of such interactions is readily recognizable to the audience, if unconsciously, and maintains a recognizable intentionality that is a core of compositional integrity.

Ultimately, it is the composer, whether a human being, generative algorithm, or an artificial intelligence, whose challenge it is to use these, and the myriad other, types of interactions in manners that achieve desired and/or

unexpected effects of the composition as a whole. Certainly, by developing an awareness, and subsequent deliberate implementation, of the kinetic interactions that occur throughout a composition and embracing their potential as compositional elements, we expand our pallet of expression into unforeseen modes of creativity. For it is in reaching for such expansions of ourselves that we are active catalysts of momentum toward an ever-evolving multiverse of consciousness.

Note on A/V Examples

Duplicate versions of all of the audiovisual examples are available on the following page: <http://www.perceptionfactory.com/journal-seamus/>

References

- Abbado, Adriano. (2017). *Visual Music Masters: Abstract Explorations: History and Contemporary Research*. Skira.
- Keefer, Cindy, and Jaap Guldemon. (2013). *Oskar Fischinger 1900-1967: Experiments in Cinematic Abstraction*. EYE Filmmuseum.
- Orgeman, Keely, et al. (2018). *Lumia: Thomas Wilfred and the Art of Light*. New Haven: Yale University Art Gallery.
- Rhoades, Michael. (2021). "Composing Holochoric Visual Music: Interdisciplinary Matrices." VTechWorks Home, Virginia Tech, 1 February 2021, vttechworks.lib.vt.edu/handle/10919/102159.

Hacking Pärt: A Robust, Compositionally-Based System for Generating Live Harmonies

Christopher Tignor

ct@wiresundertension.com

Abstract

This article describes a system to generate live, three-voice polyphonic harmonies derived from the audio input of a violin, creating a “chorale” effect. Any other monophonic instrument capable of supporting a pickup transducer might also be used. The violin plays the top (soprano) of these three voices. The lower two voices (tenor, bass) are generated synthetically using precise MIDI control over Antares Auto-Tune plugins. As the synthesized voices are audio processed replicas of the input, they perfectly follow the performer’s nuanced inflections of pitch, volume, rhythm, and otherwise.

Inspired by an analysis of the prescriptive pitch rules from Arvo Pärt’s “Tintinnabuli” method, composers can configure their own coincident pitch relations, designing the resulting harmonies. This allows for loosely polyphonic harmonies featuring similar and oblique motion, and contrary motion when performing with a MIDI pedalboard. The synthesized voices add an electronic timbre rich in overtones and capable of deep, sustained bass frequencies. The system sets up within minutes

and is hands-free and autonomous, allowing the performer to remain focused on playing with no need to touch the computer post setup.

Compositional Technology

A useful definition of technology reaches beyond the electromechanical tools that saturate our world to include knowledge and methodologies that help shape culture. Compositional “rules” that composers create for themselves are also technologies that help define these artists’ voices. A notable example comes from the work of composer Arvo Pärt.

“Spiegel im Spiegel,” excerpted in Figure 1, clearly demonstrates Pärt’s “Tintinnabuli” style, a prescriptive method for choosing pitches for a melodic voice, known as the “M-voice” and its siblings, the Tintinnabuli voices, or “T voices”. The M-voice viola part typically moves by step. The piano part contains three voices: upper, lower and a middle ostinato. The upper piano voice can be derived from the current viola pitch by transposing it up 12 F major scale degrees and “rounding” the result to the nearest member of the F major triad.



Figure 1. Arvo Pärt’s “Spiegel im Spiegel” mm. 92-99 (above) and “Fratres” chord progressions (below)

For pitch class $F = 0$:

$$t = \text{round}(m + 12, \{0, 4, 7\})$$

Equation 1. *Deriving the pitch of the upper piano T-voice in Arvo Pärt's Tintinnabuli system.*

where t is the upper piano T-voice pitch, m is the M-voice viola pitch, and $\text{round}(x, y)$ returns the pitch with the minimum chromatic distance from x matching a pitch class from the ordered pitch class set y . Here, x is set to the M-voice transposed up by 12 scale steps and y is set to the F major triad. (Pärt rounds G, at the midpoint between F and A, up to A.)

The lowest piano voice is derived identically, but by transposing down by 11 scale degrees instead of up by 12 scale degrees. The ostinato is also derived from the M-voice. The first note is two scale steps above the viola. The third note is an octave above the viola. The middle note is the nearest member of the F major triad *below* the viola.

A similar relationship between M-voice and T-voice can be seen in Pärt's "Fratres." "Fratres" consists of a set of nine variations on the same additive sequence of chords, adjoined by ten reprises using percussive string gestures. Prior analysis by Akesson (2003) described these chord sequences, included in Figure 1. In these three-voice chords, the lower voice is simply the upper voice transposed down by 10 scale degrees along the A Phrygian dominant scale. These outer M-voices always begin and end on E and move step wise toward the middle of the sequence followed by a leap that ensures the sequences begin and end on a first inversion A major chord. Each variation consists of three descending sequences followed by three ascending sequences. The middle T-voice in these chords is always selected from the A minor triad, creating a luminous, liturgical quality derived from simple, constant linear motion grounded in the harmonic stasis of its inner Tintinnabuli A-minor sonority.

As per "Spiegel," we can describe the pitch selection of the inner T-voice in relation to the upper M-voice using transposition—here chromatic—and pitch rounding. In "Fratres," the unevenness of Pärt's M-voice to T-voice mapping requires a conditional design.

For p.c. $A = 0$:

$$t = \begin{cases} \text{round}(m - 8, \{0\}) & \text{if } m \% 12 = 7 \\ \text{round}(m - 8, \{3, 7\}) & \text{otherwise} \end{cases}$$

Equation 2. *Deriving the pitch of the inner T-voice in Arvo Pärt's Tintinnabuli system.*

indicating different pitch class set arguments to $\text{round}()$ for when the M-voice plays E versus other pitch classes. The first argument to $\text{round}()$ is always set to the M-voice pitch transposed down by 8 semitones.

As seen in "Spiegel," if the pitch class set argument to $\text{round}()$ contains even intervals (e.g. major thirds), a pitch lies at their midpoint. Here, a B/ M-voice produces such a midpoint at D following transposition. As soon discussed, in practice, avoiding such midpoints is critical. It is not uncommon for the $\hat{2}$ degree in the Phrygian mode to be expressed quite flat, creating a narrow interval between itself and the tonic (see Strange and Strange 2001, 143). Here this transposes to a flattened D, avoiding the midpoint and $\text{round}()$ returns Pärt's T-voice at C instead of E. Alternatively, we might simply transpose M-voices down by an added microtonal *epsilon*.

Pitch-Mapping using Auto-Tune

It is possible to realize pitch transformations of live acoustic input such as those described by Equation 2 through an iconoclastic application of Antares Auto-Tune software, pitch-shifting the original acoustic sound to generate the additional harmonic voices. This allows a solo acoustic instrumentalist to perform chord progressions such as those in "Fratres," containing musically satisfying harmonic voicings capable of parallel and oblique motion. Though not present in the "Fratres" sequences, contrary motion is also possible.

Antares Auto-Tune is state-of-the-art, professional pitch recognition and transposition software typically used in the recording studio to correct vocal performances. It supports a wide range of features to correctly identify the incoming pitch and produce natural sounding results, including sensitivity to incoming vibrato. In real-time, the software pitch-shifts an incoming monophonic signal to the nearest acceptable pitch from a set of pitches configured

by the user. MIDI control can also be used to define acceptable pitch targets, allowing producers to play a MIDI keyboard along with the vocal track to correct pitches in real-time.

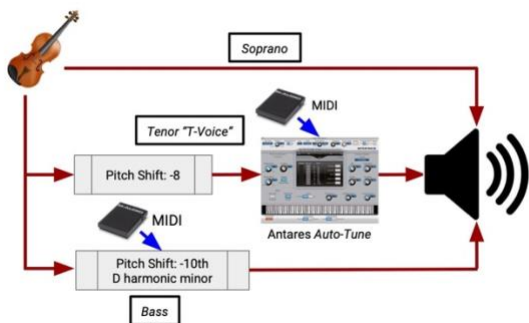


Figure 2. Chordal harmonization system for “Fratres”

In effect, for an incoming pitch, Auto-Tune implements the *round()* function from Equation 2, quantizing the audio to the nearest pitch (in cents) from the currently defined set of acceptable pitch targets. Tymoczko (2013) previously

discussed quantization theory in the context of transposing “continuous chords” (chords not tempered to the chromatic scale) into scalar (equal-tempered) chords. Here, Auto Tune reifies these theoretical insights, quantizing its pitch continuous input toward a “scale” defined by the user, here an arbitrary set of equal-tempered pitches. As Auto-Tune simply rounds continuous pitch input toward the nearest pitch target, mid-points between targets cannot be resolved “by convention”. Playing at these mid-points causes “pitch thrashing” as tiny changes in the input move it closer to either pitch target. In moving from the pitch-theoretical into practice, we thus strategically avoid these singularities, e.g. the Phrygian flattened $\hat{2}$ in “Fratres.”

The corresponding system design, configured for “Fratres” is presented in Figure 2. A MIDI toggle pedal triggers the custom *SuperConductor* software to send note-on and note-off messages to Auto-Tune, switching between these pitch target sets.



Figure 3. “Fratres” tenor voice Auto-Tune {A} (above) and {C, E} (below) pitch configurations. Highlighted keys indicate pitch targets enabled for transposition.



Figure 4. “Arrow in the Dark” tenor voice Auto-Tune F major (above) and D minor (below) pitch configurations. Highlighted keys indicate pitch targets enabled for transposition.

As per Figure 2, the direct sound of the violin performs the chord's soprano voice. The inner tenor voice is generated by first pitch-shifting the violin down 8 semitones and then routing this signal to Auto-Tune. The bass voice is generated by pitch shifting the violin down 10 scale steps in D harmonic minor using the Antares Harmony Engine plugin. Using a MIDI pedal and the *SuperConductor* software, the performer can instantly toggle Auto-Tune between either pitch class set $\{A\}$ or pitch class set $\{C, E\}$. When playing E at the beginning and end of the “Fratres” sequences, the player chooses pitch class set $\{A\}$. For all other notes the player chooses pitch class set $\{C, E\}$. *SuperConductor* emits the corresponding MIDI notes for pitches in all desired octaves that are then routed to Auto-

Tune. The two Auto-Tune pitch configurations are shown in Figure 3.

By configuring Auto-Tune with custom, varying pitch targets one can incorporate their own harmonic practice into original electroacoustic works. To demonstrate, violin excerpts from several of my own works are below. Recordings of performances are linked as each work is introduced.

The first work, [“Arrow in the Dark,”](#) maintains the same system design as Figure 2, but switches the tenor Auto-Tune pitch targets between the F major and D minor triads as depicted in Figure 4 (Tignor 2015a). For both these triads, their even interval (major third) produces a pitch, G, with an ambiguous target halfway between pitch targets F and A. The pitch

Figure 5. “Arrow in the Dark” mm. 1-21. Instruments h1 and h2 indicate the synthesized harmonic voices below the violin.

D#, easily avoided here, would be needed prior to the Tp-8 transposition to produce G. Again, a MIDI pedal registered with *SuperConductor* is used to toggle Auto-Tune between these pitch sets. A second MIDI pedal is used to toggle the scale mode of the Harmony Engine plugin, transposing the bass voice between the D natural and D harmonic minor scales in order to produce dominant chords. Figure 5 presents an excerpted score where these pitch set and scale adjustments are indicated using text where they occur.

As illustrated in the score excerpt in Figure 5, the system naturally supports several musical features not present in “Spiegel” or “Fratres,” including functional chromaticism, for example in measure 13 where I apply a secondary dominant. Additionally, by holding a note on the violin and switching pitch sets in the last measure I cause the inner voice to shift to its new nearest pitch (C₃), moving obliquely to the other sustaining voices. The glissandi rate of pitch changes can be configured to taste in Auto-Tune.

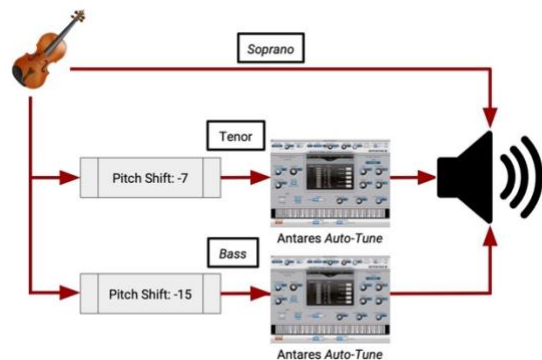


Figure 6. Chordal harmonization system for “Known by Heart.”

A yet more chromatic example can be seen in the opening to my piece [“Known by Heart”](#) (Tignor 2019). As Figure 6 shows, for this work the tenor voice is first transposed down 7 semitones before Auto-Tune pitch rounding. The bass voice is transposed down 15 semitones before processing. The Harmony Engine plugin previously used in this chain for fixed, scalar transposition is replaced with a second Auto-Tune plugin.

As illustrated in Figure 7, for “Known by Heart” the tenor voice Auto-Tune is configured with the pitches D and A in all octaves. The bass voice Auto-Tune is configured with the pitches from the E-flat major triad. The score excerpt in

Figure 8 depicts how this combination allows for some interesting mode mixture and dissonance as well as deep, rich E-flat bass tones at the violin’s open G in measures 5 and 7. Though the major third in this triad produces an ambiguous pitch target for F, the input pitch prior to the Tp-15 transposition would need to be G#, a pitch easily avoided in this harmonic layout.



Figure 7. “Known by Heart” tenor and bass voice Auto-Tune {D, A} and E-flat major pitch configurations, respectively. Highlighted keys indicate pitch targets enabled for transposition.

For the greatest flexibility over harmonization, the system can be configured to allow the performer to directly specify the pitch of the Auto-Tuned voice using any commercially available octave-ranged MIDI pedalboard. This likens the instrumentalist to an organist, playing the upper voice on their instrument while playing the bass voice with their feet. I demonstrate this at the beginning of the third movement of my work [“Artifacts of Longing,”](#) which also uses the system design from Figure 6 (Tignor 2015b). Here I fix the tenor voice’s Auto-Tune pitch at

Figure 8. “Known by Heart” mm. 1-8. Instruments h1 and h2 indicate the synthesized harmonic voices below the violin.

C#4 while playing the bass voice manually on a Roland FC-200 MIDI Foot Controller.

As seen in the score excerpt in Figure 9, in this manual configuration full control over the synthesized pitch is possible. Because the *round()* function that describes the Auto-Tune effect is monotonically increasing with respect to the input pitch (see Figure 10), contrary motion is not possible using a fixed set of target pitches but easily achieved by manually changing the target pitch while playing. The tapered and slightly detached bowing in this section assists in changing the bass pitch fractions of a second before voicing the source note on violin.

As evident from the cited recordings, the timbre of the harmonized voices sounds characteristically “electronic” with a rich synthetic timbre still audibly derived from the

violin. This is largely due to the extent of pitch-shifting. Though the synthesized voices naturally blend with their acoustic source, it is a non-goal of this project to create a one-man proxy for a string section. While expanding the vocabulary of the soloist, this new electroacoustic instrument wields unique electronic timbres with the extraordinary range of nuance available to the acoustic violin. Derived from live processing, the synthesized voices blend well with their acoustic source. Further, the inherent amplification potential allows for performances in the ubiquitous, louder commercial venues native to popular music.

System Implementation

Many live electroacoustic systems are not sensitive to the rigid performance constraints

Figure 9. “Artifacts of Longing, mvmt. 3” excerpt. Instruments h1 and h2 indicate the synthesized harmonic voices below the violin.

imposed by touring, multi-artist bills in commercial venues outside academic contexts. In these environments, several pragmatic concerns are paramount including portability, speed of set-up, reliability in the face of little-to-no sound-check, readily available software and hardware components, and the ability to run the system without the assistance of auxiliary technicians.

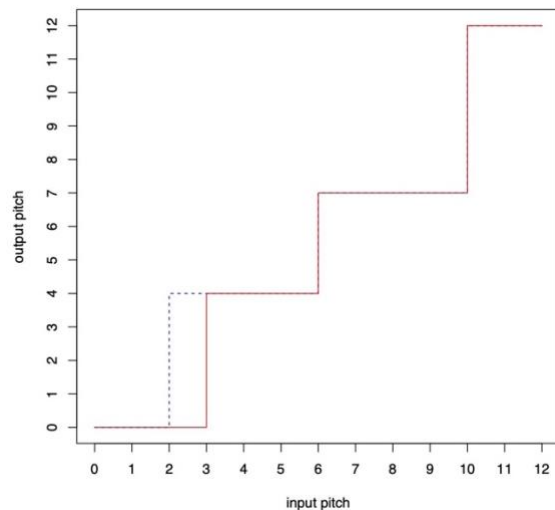


Figure 10. *round() function describing the Auto-Tune transposition effect using a major triad for target pitches. The ambiguous pitch target for input pitch 2 will be transposed according to the dashed line if played slightly sharp and the solid line if played slightly flat.*

The system described here can be unpacked, set-up, and ready to play in roughly 10 minutes. It has been vetted at over 200 performances internationally in venues ranging from meticulous concert halls to (less-meticulous) rock clubs without abnormal system behavior. Critically, when excessive ambient noise or a performance error occurs the system gracefully fails by simply selecting one of the other target pitches, still within the music’s harmonic vocabulary. Often these errors are due to excess overtones (e.g. the bow slips too close to the bridge) and the selected p.c. remains the same but the note is briefly higher than intended by one octave.

The stereo system is implemented using Ableton Live, a widely used Digital Audio Workstation (DAW), and has been run on a

modest 2011 MacBook Pro 2.66 GHz Intel Core i7 with 8 GB of 1067 MHz DDR3 RAM. As Live is only used as a host for the Antares plugins, the *SuperConductor* Max4Live software, and audio/MIDI routing, the DAW is never set to “play”. An RME Fireface UCX audio/MIDI interface was selected for its low-latency performance, critical considering all audio passes through the computer before being delivered to the house and needs to blend with the acoustic violin. The original violin signal is also routed through the system as ample reverb is applied to blend all voices into the desired effect.

In practice, two violin inputs are often used. A DPA 4099V supercardioid condenser instrument-mounted microphone is preferred for the unprocessed “soprano” violin path. (In the linked performances, an overhead microphone was used.) The signals for processing derive from an L.R. Baggs pickup transducer built into the bridge of the violin, then routed to an L.R. Baggs Para DI before arriving at the audio interface. The signal isolation provided by the pickup is critical in a performance environment for the Auto-Tune pitch recognition to work properly.

A Roland FC-200 MIDI Foot Controller is used as the MIDI controller. The *SuperConductor* software used to send MIDI note messages to configure Auto-Tune on the fly was implemented in Max4Live, largely in Javascript. The software is freely available from the author at: <http://wiresundertension.com/soft>.

System Limitations

Certain system limitations are inherent to the unorthodox application of Antares Auto-Tune for pitch-shifting. Notably, the rhythm of the synthesized voices is identical to the acoustic input except when manually changing pitch targets during sustained pitches (as seen in the last measure in Figure 8). This rhythmic coupling could be mitigated by delaying the synthesized voices, perhaps using MIDI foot pedals to control the delay times.

Auto-Tune’s pitch rounding, a quantization strategy seen in Pärt’s *Tintinnabuli* method, clearly imposes harmonic and contrapuntal restrictions. Some, like oblique motion being restricted to movement of the input voice, can be

mitigated by switching pitch sets during sustained notes. Others, like a lack of contrary motion, can only be remedied by manually playing the pitch-shifted voice with a MIDI pedalboard. As with organ pedalboards, this challenging technique can be done with few voices at once; likely one.

Without care, quickly changing pitches in a live setting can be problematic (note that the original music presented is slow, even liturgical in quality). This is largely due to the reverberance of prior notes corrupting the assumed monophonic input to Auto-Tune's pitch recognition. Care must likewise be taken with ringing open strings during string crossings; at times muting these with another finger works best. Thus, chordal techniques like bariolage cause Auto Tune to flail as it struggles to identify a single, stable pitch. While Auto-Tune supports vibrato, exceedingly wide vibratos will also throw off pitch recognition.

As seen in the included works, these "limitations" have been taken as creative boundaries in support of a specific style of composition and performance.

Conclusions

Composers have now long enjoyed an expanded electronic palette of sound. Yet, within today's vast landscape of electronic music performance (commercial and otherwise), the use of pre-recorded material remains common. This can pose meaningful limitations even for through-composed works, as fixed-time media is inherently incapable of critical interpretive choices including rubato, vibrato, adjustments to dynamic, and other inflections. For more open scores this problem magnifies. In returning interpretive choices to performers, we are faced with the problem of how to create instruments equipped with electronic forces that gracefully align with the performer's expressive instrumental training. Historically, live

processing of acoustic instruments has often leaned toward the aleatoric (and thus improvisation) or effects outside the domain of "notes" (e.g. timbre), and rarely toward techniques that preserve those specific pitch-relations acoustic composers slave over to achieve their work's specific musical affect.

The system described herein provides a practical means by which monophonic acoustic instrumentalists might extend their performance practice harmonically and contrapuntally. It both reifies and transcends Pärt's harmonic "method," building a new instrument from the essence of his techniques. The process of designing the coincident pitch-relations used to configure Auto-Tune offers a new compositional challenge, making the material dependency between performance input and electroacoustic processing compositionally explicit.

References

- Akesson, Linus. (2003). "Fratres," Personal web site. <http://www.linusakesson.net/music/fratres>.
- Strange, Patricia and Allen Strange. (2001). *The Contemporary Violin: Extended Performance Techniques*. Scarecrow Press.
- Tignor, Christopher. (2015a). "Arrow in the Dark." Musical performance. <https://vimeo.com/135898141>
- Tignor, Christopher. (2015b). "Artifacts of Longing, movement 3." Musical performance. <https://vimeo.com/154465915#t=716s>
- Tignor, Christopher. (2019). "Known by Heart." Bandcamp. <https://christophertignor.bandcamp.com/tracks/known-by-heart>
- Tymoczko, Dmitri. (2013). "Geometry and the Quest for Theoretical Generality." *Journal of Mathematics and Music* 7, no. 2: 127–44.

Inventorying and Evaluating the University of Illinois Experimental Music Studios' Archive

David M. Luftig

Revised in 2023

Washington State University
david.luftig@wsu.edu

Abstract

This study seeks to understand the content, organization, and context of the University of Illinois Experimental Music Studios' (EMS) archive. This project was carried out in 2016, while I was a graduate student of information science at the University of Illinois at Urbana Champaign. For this project, I spent seventeen weeks documenting the original EMS archive with a particular focus on the magnetic tape collection. The original EMS archive was housed within the School of Music before being moved into the University of Illinois Archives. The EMS was founded in 1958, and its audio collection started soon after. Although once a heavily utilized collection, these tapes sat unused for many years upon me encountering them.

The purpose of this project was to document what a historic institutional tape music archive physically looked like, what sort of materials were located within in it, and how those materials were cataloged and used. I believe that such observations will allow future scholars to more accurately contextualize a historic tape music program like the EMS. It was also my goal to recognize and document those historic materials that presented an immediate preservation risk.

Background

During the 1950s and throughout much of the 1960s, electroacoustic studios located within the United States were almost exclusively associated with universities. These studios and universities eventually became repositories for magnetic tape recordings. Unfortunately, as time passed, age-induced physical deterioration of these magnetic materials endangered the integrity of those

recordings. As the Association for Recorded Sound Collections (2015) notes, magnetic tape is highly susceptible to various deterioration issues even if stored in the most ideal conditions (54-56). Haus and Pelegrin Pajuelo (2001) argue, based on their own work in electroacoustic tape archives, that many old electroacoustic recordings are now at risk due to issues of age-based deterioration. Furthermore, these risks are often unrecognized or unaddressed by archivists due to the large volume of recordings that are held in many electroacoustic archives (see also Teruggi 2001; Zattra, Poli and Vidolin 2001; Polferman, Sheppard and Dearden 2006). The risk associated with these materials is due to the deterioration of the chemical composition of the magnetizable coating on the tape and the weakening integrity of the plastic film that the coating is placed upon. These deterioration issues are exacerbated in non-ideal environmental conditions. Unfortunately, all magnetic tape-based recordings will eventually fail (see Association for Recorded Sound Collections, 2015). Therefore, the only real preservation strategy is to try to prolong the life of the master tape through proper care while also creating copies of that recording for future use. Issues of preservation in electroacoustic studios may be compounded when a repository also stores original tape loops or spliced performance tape segments. Spliced tape can lose its ability to bind the tape segments together requiring extra time and specialized labor to repair.

The University of Illinois Experimental Music Studios

The EMS was created in 1958 by composer Lejaren Hiller who served as its director until 1968. Along with the Columbia-Princeton Music Center, the EMS was the first studio focused on electroacoustic and tape music within the Western Hemisphere. The period from 1968-1975 saw several different directors of the EMS. In 1976, Scott Wyatt was hired and would serve as Director of the EMS for forty years. Scott Wyatt was succeeded by Dr. Eli Fieldsteel in 2016.

In the ensuing decades following the inception of the EMS, many notable composers participated within the program, either as students or faculty. Such composers included: Herbert Brun, Kenneth Gaburo, Ben Johnston, Lejaren Hiller, Salvatore Martirano, David Rosenboom, Carla Scaletti, and many more. Over the years, the EMS generated commercial records both through individuals associated with the department or as a university department as a whole. Some of those early titles included: *Electronic Music from The University of Illinois* (Heliodor Records, 1967), *Computer Music from The University of Illinois* (Heliodor/MGM Records, 1967), *Music by Computers* (which

included compositions from Lejaren Hiller and James Beauchamp) (John Wiley & Sons, Inc., 1969). Early albums from individual composers affiliated with the EMS included Kenneth Gaburo's *Music for Voices, Instruments & Electronic Sounds* (Nonesuch Records, 1968), Salvatore Martirano's *L's GA* (Polydor Records, 1968), and Lejaren Hiller's *Avalanche, Nightmare Music, Suite For Two Pianos and Tape, Computer Music for Tape and Percussion* (Heliodor, 1970). By the 1970s, the EMS had created its own record label which lasted through the 1990s.

Aside from musical compositions, the EMS was also active in creating and utilizing unique electronic instruments and technology, often in association with University of Illinois engineers and computer scientists. For example, Lejaren Hiller's Illiac Suite, written in 1956, utilized a newly developed computer at the University of Illinois, the ILLIAC. The Illiac Suite is often credited as being the first piece of computer music (Chadabe 1997, 273-74). Additionally, James Beauchamp's Harmonic Tone Generator, developed in 1964, was one of the earliest voltage-controlled synthesizers. In 1969, Salvatore Martirano began constructing his own instrument, the Sal-Mar Construction. Built

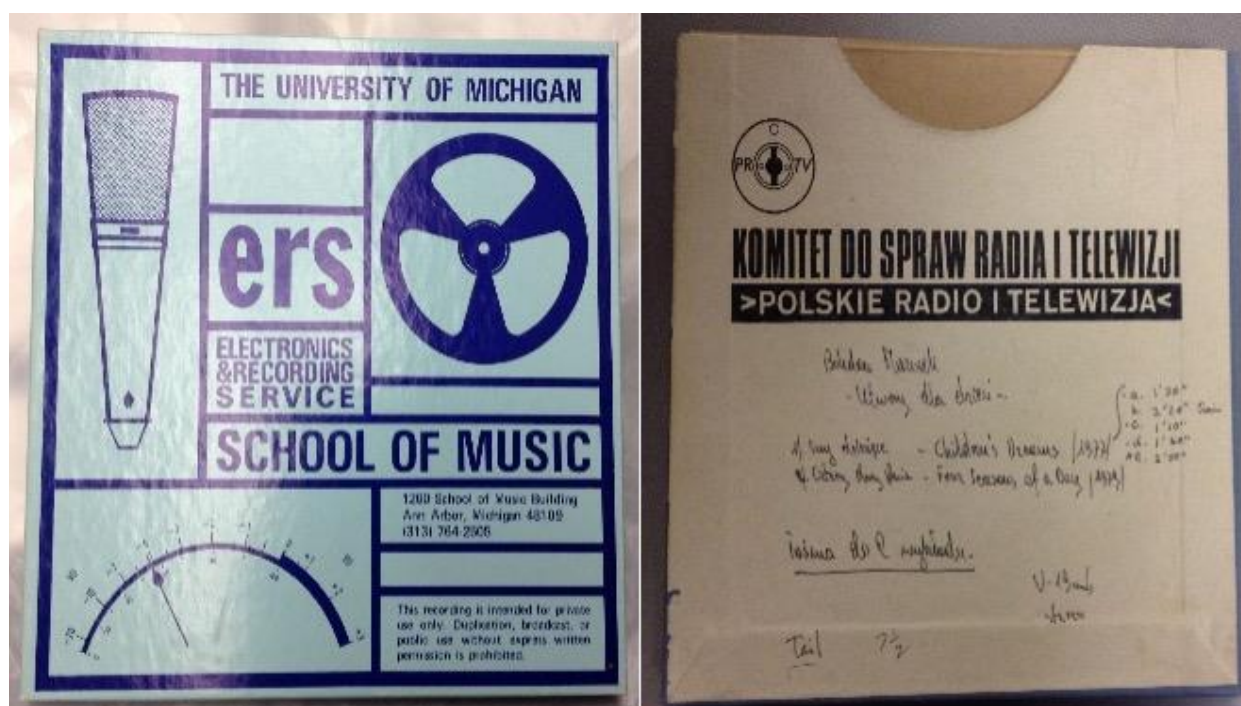


Figure 1. Two uncatalogued tape boxes. University of Michigan (left) and Polskie RI Telewizja (right).

partially from the circuit boards of the ILLIAC II, the Sal-Mar Construction was one of the first musical instruments to use modern transistor-transistor logic (TTL) digital circuitry (see Chadabe 1997, 289–90). Currently, the Sal-Mar Construction resides within the University of Illinois’ Sousa Archives and Center for American Music.

The EMS Audio Collection

The EMS’ audio collection began to be built shortly after the inception of the program although the majority of the collection was accumulated during the tenure of Scott Wyatt. According to my conversations with Scott, the collection materials were meant to serve several purposes. For example, it was to be utilized for pedagogical purposes, to be part of specific live performances, as an archive of materials gifted to the EMS, and as a general repository for materials recorded within the EMS. Scott continued to grow the tape collection until digital recording technology began to replace analog tape around the mid-1990s.

Regarding the tapes that were gifted to the EMS collection, they either were given to Scott in person when that composer came to campus, were exchanged at off-campus performances and symposiums, or arrived via correspondences in the mail. The physical tape cases in the collection reveals many different studio sources (See Figure 1).

Transition in the EMS Department and Archive

In 2016, Scott Wyatt retired and the new Director of the EMS, Dr. Eli Fieldsteel, came into the program. Although Scott had done an extraordinary job of growing and caring for the collection, the tape collection was no longer being utilized and the tapes were taking up a large amount of the EMS’ limited physical space. Additionally, many tapes were not immediately playable, as the paper tape leader needed to be replaced before playback. This could be especially time consuming as many of the tapes had leader separating all the individual compositions on that single reel of tape. Of greater concern though, were the issues regarding the deterioration of the actual magnetized tapes.

Because all the tapes in this collection were decades old, there was concern regarding how much longer those tapes would remain playable.

With these factors in mind, a decision was made between the EMS and the University of Illinois Sousa Archives and Center for American Music (SACAM) that the collection would be accessioned into the University Archives. SACAM already had a collection of EMS materials including electronic instruments and personal papers. Therefore, accessioning the tape collection to them made sense.

Method

I originally conceived of this project out of happenstance. I was having a conversation with a music student who told me of the EMS’ audio collection that he had heard was stored in a closet in the Music Building. I immediately contacted the School of Music Director, Scott Wyatt, regarding preserving and documenting this collection. It turned out that Scott was retiring but gave me the name of the incoming director. I contacted and arranged a meeting with the incoming director, Eli Fieldsteel. After meeting several times, Dr. Fieldsteel granted me full access to the collection and the ability to serve as an advisor to him regarding what to do with the collection moving forward.

For this project, I spent seventeen weeks working with the collection. I also traded emails and met with Scott Wyatt repeatedly. The overall goal of this project was to understand and document the EMS collection. Other goals included the implementation of a long-term preservation plan for these materials and helping to preserve the institutional memory of the EMS.

To assist in processing this collection, I created a spreadsheet to record all the data from the tape cases as well as data regarding the physical condition of each item. This spreadsheet is available in the “supplemental materials” links below. Additionally, I documented information regarding the ephemera found in the tape boxes, the location of materials, and other various notes. After this collection was moved to SACAM, I further assisted in the documentation of materials and the creation of the finding aid for this collection.

Although this study is primarily interested in

the tape collection, I also evaluated other materials including ephemera, musical scores, digital audio tapes, LPs, videocassettes, and optical discs. That said, most items in the collection were magnetic tape recordings.

Arrangement of Collection

The EMS tape collection was located in a ten foot by six-foot temperature-controlled storage closet that also shared space with other EMS materials (such as cables, boxes, cleaning supplies, posters, and ephemera, and other assorted odd and ends).



Figure 2. *The three metal shelves which tapes are stored.*

Within that closet space were four metal shelves that were six feet high by about thirty-four inches long. Each shelf contained six compartments (see Figure 2). Three of those shelves were located along the left side of the closet space (if looking toward the far wall). One shelf was on the far end of the closet. All tapes were situated on three of those shelves. The tape collection took up about twenty-four linear feet of space once they were boxed up.

Written on the spine of about 60% of the 457 overall tapes is a catalog number and a tape name (see Figure 3). For the cataloging classification system, all numbers start with “5” and are followed by three additional numbers (e.g., 5001, 5002, etc.). All of the cataloged materials were in 10” tape boxes (mostly Scotch 206 or

Scotch 207 tape boxes). The tape brand listed on the box was not necessarily representative of the tape brand that was used, as Scott Wyatt verified.



Figure 3. *Tapes sitting on shelf with red catalog number and tape name in black.*

Written on the back of the tape boxes is information about the recording (see Figure 4). This descriptive content was often typed but occasionally written by hand. The tapes with catalog numbers were easily accessible on the shelf and were organized from left to right sequentially.

An index-card catalog system was created and utilized by Scott Wyatt. Those index cards contain composition, composer, and playback information. These catalog cards were stored in a wooden box which I returned to Scott Wyatt (see Figure 5). Three manila folders within the archive contained a typed list of the recordings in the EMS archive with the corresponding cataloging schema. These lists were made in the late 1970s according to Scott. The cards contained both typed and handwritten entries for the collection. These folders, as well as the scans of them that I created, were given to SACAM upon accession.

Although there were a few tapes with no cataloging schema in 10” boxes filed along with the cataloged collection, most of the uncatalogued tapes were in 7” boxes and were less physically accessible (due to those tapes often being stacked on top of each other and on the lower shelves) (see Figure 6). Some of these tapes were grouped by author but often, the tapes were unorganized.

EXPERIMENTAL MUSIC STUDIO UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN TAPE ARCHIVE

EMS 5173 J. Randall—computer Pieces 7.5 & 15 2 TRACKS

DATE REC'D: TAP. IN: MONO: STEREO: QUAD:

DATE	TITLE	YEAR	STUDIO	TIME	COMPOSER, PERFORMER, PRODUCER
	1. Mulgott: Monologues by a Mass Murderer (13 ips)	1963	Princeton & Bell Labs, Murray Hill, N.J.	3:23	Vocalist: Melinda Kesler
	a. Electronic Prelude				
	b. Toronto (I leave my recently adopted children Alice and Nellie at 14 St. Vincent Street.				
	2. Lyric Variations for Violin and Computer (7.5 ips)	1968			Lakofsky—violinist

COPIES: DATE RECORDED: DATE REC'D: DATE REC'D: DATE REC'D: DATE REC'D: DATE REC'D:

ORIGINAL: DATE RECORDED: DATE REC'D: DATE REC'D: DATE REC'D: DATE REC'D:

SOURCE OF RECORDING: DATE RECORDED: DATE REC'D: DATE REC'D: DATE REC'D: DATE REC'D:

NOTES: Realized with Princeton (but not Columbia-Princeton) IBM 7094 Computer
D/A conversion accomplished at Bell Telephone Laboratories, Murray Hill, N.J.

Figure 4. The back of a typical 10" tape case that had been cataloged.



Figure 5. The wooden box in the EMS that held the catalog cards.



Figure 6. Uncatalogued tapes.

The uncatalogued collection represented a wide range of materials from different studios and composers. Some of these materials were gifts from external composers while many were created within the EMS (often as a student's class project). Additionally, some of these uncatalogued materials contained well-known compositions that were probably used for pedagogical purposes. Most of these tapes were in 7" boxes and had handwritten notes on the back. It stands to reason that the cataloged tapes were more integral to the EMS collection as they were the recordings that were much more easily accessible and had a cataloging schema assigned to them.

Findings

There were 457 magnetic tapes within the EMS collection. Of that these tapes, 274 were part of the original EMS catalog schema. Within the EMS catalogued collection, 133 tapes were acetate, 291 tapes were polyester, and 33 tapes had both polyester and acetate on the same reel. Of these acetate tapes, at least fourteen were marked as being master or submaster recordings. Although many of the tapes were marked as being copies or master recordings on their boxes, the majority of tapes had no provenance information. I observed that the tapes without provenance information often seemed to be copies of previously recorded compositions. Of the collection that was not catalogued, 38 tapes were clearly marked as being master recordings. Of these master recordings, fourteen tapes were acetate, one tape was a mix of acetate and polyester, and the rest were polyester based.

Overall, some notable master recordings found in this collection included:

- Selections written for by Lajaren Hiller and James Beauchamp for the 1969 album *Music by Computers* (on an acetate tape).

- Herbert Brun's "*U-Turn-To 3*," "*i told you so!*" (labeled as "new master"), and "*Snow '84*" (all of these recordings are on polyester tape)
- Lejaren Hiller's "*Seven Electronic Studies*," "*Algorithms I*," "*Machine Music*" (labeled "submaster"), and "*A Triptych for Hieronymus*" (all these recordings are on acetate tape).
- Ben Johnston "*Compositions*" and Ben Johnston and Jaap Spek (unknown title) (on acetate tape)
- John Melby's "*Chor der Waisen*" (marked as "dubbing master" and on polyester tape)
- Joel Chadabe's "*Mod 3*" (on acetate tape)
- Carla Scaletti's "*Lysogeny*" (on polyester tape)

The provenance of the materials from this collection is given in Table 1 below.

Upon an item level inspection of each tape, ninety-six tapes were determined to have an immediate preservation concern. These tapes exhibited signs of uneven tape winding, wrinkled or damaged tape, sticky shed, and other possible signs of deterioration. Of these tapes that were deemed the most in need of preservation consideration, eleven tapes were marked as being a master recording and one was tape indicated that it was a backup master recording.

The tapes represent various playback speeds, track types, and channel configurations. Within this collection, 443 tapes are ¼" tapes and fourteen are ½" tapes. Eighteen tapes are indicated as being quadraphonic sound recordings. The rest of the tapes were either mono or stereo recordings. The tapes utilized various playback speeds (mostly seven and a half inches per second or fifteen inches per second). Forty-

Provenance of Recording	Number of Tapes in Collection
Unknown/Unspecified Recording Source	346
Copies	65
Master Recordings	38
Sub Master, Backup Master, Dubbing Master, etc. Recordings	8

Table 1. *Provenance of Recordings in EMS Archive*

three of the tapes changed speed based on the individual song selection on that tape.

Although most of the cataloged tapes were in Scotch 206 or Scotch 207 tape boxes, those labels were not necessarily representative of the actual manufacturer of the tapes inside those boxes. In fact, most of those tapes, according to Scott Wyatt, are Scotch 201 or 202. Such tapes are made of acetate and therefore a particular preservation concern based on their inability to be baked, which is a common tape preservation strategy.

Forty-seven different international recording studios were mentioned on the tape boxes with many more studios being represented (but not explicitly mentioned on the tape case).

Of the tapes in the collection, sixty-one tapes had some ephemera enclosed within the box. These materials were often information that had been written by the composer or from someone inside the EMS. Ephemera included customs information for international tapes, live performance notes, or correspondences (see Figure 7). This ephemera presents another preservation challenge as some of these paper materials were very fragile and also deteriorating.

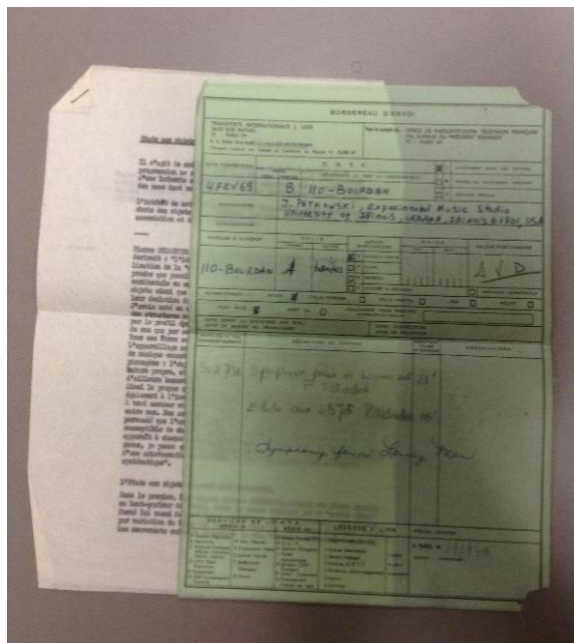


Figure 7. This example of ephemera was found in a Pierre Schaeffer tape case. The document is written in French and appears to be a customs form.

I was unable to determine the number of tapes that were meant to accompany live performances, but I suspect there was at least four or five recordings (based on the content written on the back of the tape cases). There is no actual spliced tape or original tape loops within the collection. According to Scott Wyatt, it was the policy of the EMS for the composer to retain those materials.

Although the bulk of the materials found in this space were tape recordings, some other materials were found. Such materials included a binder of original program guides for both internal and external performances, posters advertising EMS events or recordings, blueprints for proposed building renovations, a technical report regarding EMS instruments, sheet-music, a catalog of tape recordings in the Dartmouth Electronic Music Center Tape Library (1970), various SEAMUS CDs, and a few printed articles from journals and magazines. These materials were accessioned by the SACAM.

Conclusion

The EMS collection represents decades of the accumulation of materials. These materials included a large multi-use tape collection, pedagogical resources, documentation of the program, and various forms of ephemera. I believe that by understanding this collection as a whole, one can better understand both the day-to-day operations and the overall legacy of the EMS. Because of the significance of the EMS, care must be taken in the preservation of its archive. The most at-risk items in the collection are the magnetic tapes. The master recordings from within the tape collection should be a top preservation priority. This is especially true for master recordings on acetate tape from notable composers (some of which have already been observed to be suffering from tape deterioration issues).

Supplemental Materials

A list of materials found within the EMS archive can be found in within a Microsoft Excel spreadsheet here:

<https://www.ideals.illinois.edu/handle/2142/100315>

The completed finding aid for this collection, post-move to the University of Illinois archives, can be found here:

<https://www.library.illinois.edu/sousa/archon/index.php?p=collections/findingaid&id=11278&q=>

Acknowledgements

Lauren Bratslavsky (Illinois State University), Eli Fieldsteel (University of Illinois at Urbana-Champaign), Jason Harvey (University of Rochester), and Scott Wyatt (University of Illinois at Urbana-Champaign).

References

Association for Recorded Sound Collections. 2015. *ARSC Guide to Audio Preservation*. Washington, D.C.: Council on Library and Information Resources.

Chadabe, Joel. 1997. *Electric Sound: The Past and Promise of Electronic Music*. Upper Saddle River, N.J: Prentice Hall.

Haus, Goffredo, and Maria Luisa Pelegrin Pajuelo. 2001. "Music Processing Technologies for Rescuing Music Archives at Teatro alla Scala and Bolshoi Theatre." *Journal of New Music Research* 30, no. 4: 381–88.

Polferman, Richard, David Sheppard, and Ian Dearden. 2006. "Time to re-wire? Problems and Strategies for the Maintenance of Live Electronics." *Organised Sound* 11, no. 3: 229–42.

Teruggi, Daniel. 2001. "Preserving and Diffusing." *Journal of New Music Research* 30, no. 4: 403–5.

Zattra, Laura, Giovanni De Poli, and Alvis Vidolin. 2001. "Yesterday Sounds Tomorrow: Preservation at CSC." *Journal of New Music Research* 30, no. 4: 407–12.

Drake Andersen
Editor-in-Chief, Journal SEAMUS
journal@seamusonline.org

Department of Music
Vassar College
124 Raymond Ave.
Poughkeepsie, NY 12604

www.seamusonline.org