Ensemble Feedback Instruments

Muhammad Hafiz Wan Rosli
hafiz@create.ucsb.edu

Karl Yerkes
yerkes@mat.ucsb.edu

Timothy Wood
fishuyo@mat.ucsb.edu

Hannah Wolfe
wolfe@umail.ucsb.edu

Charlie Roberts
charlie@charlie-roberts.com

Anis Haron
anisharon@umail.ucsb.edu

Fernando Rincón Estrada
fernandorinconestrada@gmail.com

Matthew Wright
matt@create.ucsb.edu

ABSTRACT

We document results from exploring ensemble feedback in structured electroacoustic improvisations. A conceptual justification for the explorations is provided, in addition to discussion of tools and methodologies. Physical configurations of intra-ensemble feedback networks are documented, along with qualitative analysis of their effectiveness.

Author Keywords

Feedback, Multiuser, Ensemble, Second-order Cybernetics, Stability theory, Attractors, Feedback Delay Network

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing—Methodologies and techniques, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing—Systems

1. INTRODUCTION

We present findings and theory exploring systems in which each (electronic) instrument accepts an audio input that it incorporates into its output. The infinite openness of this requirement defines a wide yet restricted scope of possible instruments (Section 4.2) that can be connected to create interesting architectures and configurations.

In our initial explorations, 2-8 players each control an arbitrary individual instrument modeled as outputting the signal $say(t)$ in response to receiving the input signal $hear(t)$ as follows:

$$say(t) = transform(hear(t - delay(t)), new(t)) \quad (1)$$

In other words, each instrument outputs some transformation of recent input along with an arbitrary new signal.[1] Ideally the sound of each output is spatially distinct (see Section 5) with individual $speaker_i$ gain control.

[1] Our $transform$ takes two arguments and combines the two signals: summation as the obvious default, but potentially convolution, cross-synthesis, turn-taking, etc., plus effects before and after such combination.

The new signal $new(t)$, the delay time $delay(t)$, and the causal transformation procedure $transform$ are all potentially time-varying and under direct user control. In general the delay is at least some nonzero hardware delay and bounded by a (potentially zero) delay memory limit. Likewise $new(t)$ consists at least of the nonzero hardware noise floor $hwNoise$ in addition to any intentional sound.

The connections among the nodes of such an ensemble are variable. In the most general case there is complete time-varying control of a matrix mixer M determining the input to each instrument as a weighted sum of the outputs from all n instruments:

$$\begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,n} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1} & M_{n,2} & \cdots & M_{n,n} \end{bmatrix} \cdot \begin{bmatrix} say_1 \\ say_2 \\ \vdots \\ say_n \end{bmatrix} = \begin{bmatrix} hear_1 \\ hear_2 \\ \vdots \\ hear_n \end{bmatrix} \quad (2)$$

$$hear_i = M_{i,1} \times say_1 + M_{i,2} \times say_2 + ... + M_{i,n} \times say_n \quad (3)$$

The matrix implementation (Section 4.1) adds additional delay d_M such that each node "hears" a mix of what everybody "said" in the past:[2]

$$hear_i(t) = M_{i,1} \times say_1(t - d_M) + ... + M_{i,n} \times say_n(t - d_M) \quad (4)$$

Additional performer(s) can modulate the matrix over time, and/or each individual instrument performer may control the matrix in addition to personal output level and gain or attenuation in echoing. In the most general case everybody modulates the mixer simultaneously with some sort of combining rules. Each performer is charged with continuously monitoring overall system behavior and modulating at least one gain value.

Terminology is dangerously confusing because the matrix mixer's outputs are the inputs to the individual instruments, and vice versa. Our experience suggests that best practice in discussing design and implementation of such systems is always either letting each instrument anthromorphically "hear" and "say" or explicitly stating "from *source* to *destination*" rather than just, e.g., "in" or "out."

This architecture accommodates a variable number of performers and a wide variety of possible individual instruments (Section 4.2). It creates an ensemble dynamic of interdependence with rich output sounds that cannot necessarily be attributed to individual players, a sort of collective instrument that can be more than the sum of its parts. Feedback loops (Section 2.1) create special musical possibilities including enabling "low floor high ceiling" situations [16] where a performer controlling a single gain knob has a vital and temporally articulate impact both on the perfor-

[2] A digital matrix mixer, in addition to greater minimum d_M, could also add variable extra delay per connection.

mance as a whole and potentially also on the behavior of every other player's instrument (Section 3.1).

1.1 Motivation & Related Work

"[Get] 15 people. Take a bunch of garden hose and cut it up into 8 or 10 foot lengths. Tie it all together in a knot. Grab 2 pieces. Talk into one and listen to the other... So, you know who you're hearing, but you don't know who you're talking to." – Stewart Brand [5]

Inspired by this cybernetic experiment, we constructed an audio feedback system with an analogous randomized topology of a directed graph of audio streams. We continued to develop and play with this system because it seemed to offer every member of our ensemble an essential role while allowing flexibility in membership number. Moreover, it promised the possibility of exploring many systems of interdependency among players where control of the system is distributed, continuous, and extreme. These goals have been present in many CREATE Ensemble pieces over the years. Jordà [6] remarks "If all roles are essential, the result is the product of all the individual contributions... The more interplay and flexibility are allowed in a multi-user system, the more complex will the final expression."

The notion of feedback encompasses concepts of chaos, noise, non-linearity and indeterminism. One of the main explorations of this system is the concept of "order from noise", discussed by the cybernetician, Heinz Von Foerster [15] and exemplified by many audio feedback works by composers and artists [1, 2, 9, 14, 6]. By exploring feedback systems in the auditory domain, we seek to find not only compositionally aesthetic palettes, but also computable and intelligible methods in which order can be attained from chaos, and vice versa.

2. THEORY

We consider the general behavior of ensemble feedback instruments first by relating the mix matrix M to signal flow topologies, then by considering the behavior (particularly stability) of idealized simple instruments in such systems, and finally considering human performers dynamically shaping these behaviors.

2.1 Graphs / Network Topologies

Consider the weighted directed graph (digraph) G whose adjacency matrix is our mix matrix M (i.e., each nonzero element represents an edge). G has 2^{n^2} possible topologies where n is the number of players (because M has n^2 entries each of which can be zero or nonzero). Of these, we are primarily interested in those with one or more cycles that are not self-cycles.

So far we have explored graphs that are relatively simple, like those where the number of edges E follows $n \leq E \leq 2n$. Figure 1 shows several topologies we've tried.

Figure 1a shows our most basic arrangement, a ring in which each player processes the input from the neighbor to her left. Figure 1a shows a ring superimposed with its reverse. This arrangement appears when we transition from a ring to reverse ring. Figure 1b we call "duets" and Figure 1c we call "trios".

Any player may change the current topology by modifying the matrix mixer directly, but each player may change the topology in limited fashion with her gain control: turning all the way down removes edges from G.

2.2 Signals and Systems Dynamics

If G is acyclic (a DAG, i.e., non-feedback) then it corresponds to the usual effects-patching paradigm of sequential processing, mixes, and splits. If G contains loops then we must also consider each instrument's $delay_i$, perhaps only a very short $hardwareDelay$.

Consider a single-loop graph such as in Figure 1a where each ith instrument echoes its input at exactly the same amplitude after $delay_i$: $say_i(t) = 1.0 \times hear_i(t - delay_i)$. Let M be unity-gain, with every amplitude either 0 or 1. If instrument 1 outputs a unit impulse it will come out of instrument 2 after $delay_2$, and so on infinitely around the ring at the same volume. Each instrument is now outputting an impulse train (each with its own phase) whose period we call the "loop delay," simply the sum of all instruments' delay times:

$$delay_L = \sum_i^n delay_i \tag{5}$$

Now let any one instrument lower its echo gain by one dB: each time the impulse passes through this instrument it decays, so every instrument will output the same decaying impulse train. Specifically it decays by one dB per $delay_L$. If $say_i(t) = 10^{gain_i/20} \times hear_i(t - delay_i)$ then we can consider the gain (dB) of the entire loop as

$$gain_L = \sum_i^n gain_i \tag{6}$$

when M is unity gain, otherwise

$$gain_L = \sum_i^n (gain_i + 20\log_{10}(M_{j,i})) \tag{7}$$

where j is the successor of i in the loop.

When $gain_L < 0$dB the loop will decay. When it's nearly zero the decay is slow enough that if any instrument outputs a noise burst or attacklike transient then we have essentially the Karplus-Strong plucked string model [7].

When $gain_L = 0$dB there is "marginal stability": every sound will echo at the same volume forever; since the $hwNoise$ is continuous it will accumulate, gradually adding energy that never leaves the system.

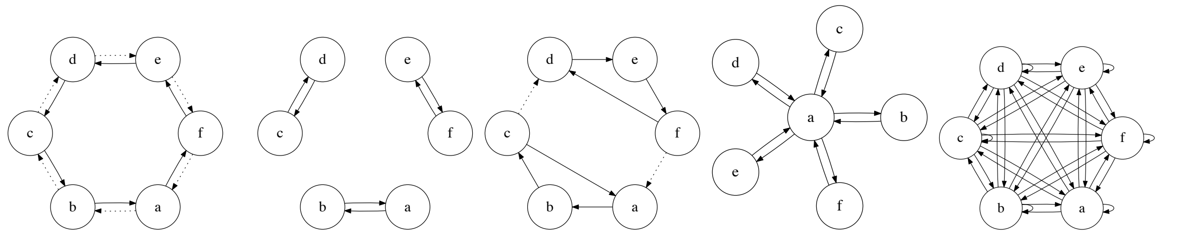
When $gain_L > 0$dB then the echoes themselves will grow, so not only will accumulated $hwNoise$ grow faster, but any $new(t)$ will generate increasingly loud echoes. Eventually such exponential growth exhausts a finite resource such as digital conversion limits or available analog current, voltage, or power and therefore cause some kind of clipping, limiting, or system failure.

Now let M be more densely connected than a single ring, such as in Figure 1d or 1e. Now the echoes can grow in density to increase overall system energy even if every $gain_i < 0$, so $gain_L$ becomes meaningless. Instead, system stability depends on M's eigenvectors: a lossless FDN design (corresponding to $gain_L = 0$ aka "marginal stability") requires M's eigenvalues to lie on the unit circle and for the eigenvectors to be linearly independent [13]. The absolute value of the largest eigenvalue is $\leq$ the spectral norm. The *spectral norm* corresponds to $gain_L$, again stable/decaying when < 1 and unstable/growing when > 1:

$$SpectralNorm(M) = max \frac{\|M\vec{v}\|}{\|\vec{v}\|} \tag{8}$$

We can guarantee marginal stability when M is orthogonal, because by definition all the eigenvalues will be on the unit circle and are linearly independent. An orthogonal matrix is a matrix A where $A^T A = I$ and I is an identity matrix. The basic version of loop graphs that we tested, shown in figure 1a,1b, and 1c, have orthogonal M matrices. These matrices are permutations of coordinate axes.

In traditional IIR filter or FDN design, either the im-



(a) Single Ring (b) Three "Duets" (c) Two "Trios" (d) Star (e) Fully connected

Figure 1: Several topologies we tried. (a) and (c) have variations shown with dotted lines: (a) can be a forward loop (solid arrows only), backwards loop (dotted lines only), or double loop (both), and (c) can be two "trios" (solid lines only) or connected trios (solid and dotted lines).

pulse response tends to decay and the design is therefore good ("stable" etc.), or the impulse response tends to grow (poles outside the unit circle) and the design is therefore bad ("blows up" etc.). In contrast, we welcome and embrace $gain_L > 0$dB regimes in which the overall energy in the loop tends to grow, *as long as it doesn't grow too fast.*

The net gain (dB) per second of the system determines the overall rate at which sound will explode or decay; for loops we have

$$d\text{dB}/dt = gain_L/delay_L \tag{9}$$

This is an average behavior over time and over the ensemble; the system dynamics may have a rich inner structure of asymmetries throughout the ensemble. In particular if $gain_i$ is high then any circulating sound jumps up discretely by $gain_i$ at each moment it comes out of i. Lowering a loop's $delay_L$ and $gain_L$ by the same factor keeps the same $d\text{dB}/dt$ but with smaller and more frequent steps that eventually perceptually blur to smooth.

Our model instrument so far is

$$say_i(t) = 10^{gain_i(t)/20} \times hear_i(t - delay_i) + hwNoise_i(t) \tag{10}$$

The system becomes much richer when one or more instruments perform additional processing such as modulating delay time and gain, multitap, internal feedback, time-varying filtering, granulation, analysis/synthesis, compression, gating, etc. Although the theoretical stability analysis is valid only when every instrument is linear and time-invariant (LTI), in practice the qualitative dynamics also hold with non-LTI instruments, particularly the effect of any performer changing output level or delay time.

2.3 Human Factors and Cybernetics

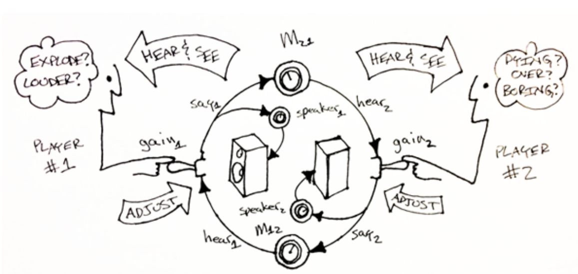


Figure 2: Feedback "duet"

The usable limits of $d\text{dB}/dt$ in practice depend on human reaction time in the second-order cybernetic loop of Figure 2. Suppose each player needs $reaction_i \approx 0.2$ seconds to perceive sound growth and correspondingly adjust $gain_i$ (including instrument latency). The venue and sound system determine the usable dynamic range between the maximum non-painful sound $sound_{Max}$ and the threshold of ensemble inaudibility $sound_{Min}$. At each moment the ensemble has a headroom h dB between the current volume and $sound_{Max}$; perhaps $h = 10$ when the ensemble is at a comfortable *forte*. As long as

$$d\text{dB}/dt < \frac{h}{\min_i reaction_i} \tag{11}$$

then whoever in the ensemble reacts fastest has just enough time to turn down. In other words, when the energy grows it grows slowly enough to be controlled by the performers within the limits of human reaction time. Of course there is no problem with decreasing sound: complete silence can be a welcomed aspect of a performance. Practicing lowers $reaction_i$ as members learn to recognize each others' *new* sounds and also to recognize and predict feedback network dynamics.

What happens when players attempt to push $gain_L$ in opposite directions? On average each instrument contributes $gain_L/n$; consider the effect of the distribution around this mean. Suppose $gain_a = 1$ and $gain_b = -1$ with the rest zero. Each hypothetical impulse maintains 0 dB as it completes each cycle, but now all ring nodes from a to just before b are 1 dB louder than the rest. In a different kind of volume war, now let $gain_a = 10$ and $gain_b = -10$, so still $gain_L = 0$ but now the two portions of the ring differ by 10dB. In general, $gain_L$ determines overall growth vs. decay, with the relative disparities in $gain_i$ giving volume differences along the loop: As player i increases $gain_i$ then not only does $gain_L$ move from decay towards growth but also the local region of the ring from i downstream becomes louder. Large volume imbalances also emphasize the $hwNoise$; as $gain_b$ decreases the signal-to-noise ratios for all nodes from b to a also decrease, and so a's amplification of this noisy signal will result in more overall noise than if b's output had been louder.

Suppose the individual loudspeakers are turned down, so all sound is quieter. Performers would naturally increase $gain_i$ to achieve the same comfortable volume level, but since this also increases $gain_L$ it will move the entire system towards growth. Likewise if individual speakers are too loud performers will naturally turn down, pushing $gain_L$ towards decay. So although in theory, feedback network dynamics are independent of presentation volume, in practice they interact and require careful calibration.

3. MUSICAL IDEAS

Connecting individual instruments into a feedback network essentially creates one larger instrument. It is possible that each instrument could be controlled by a set of predetermined instructions and this would lead to a certain class of compositions. Our practice focuses on the use of this instrument in the context of improvisational compositions. A group of performers becomes an integral part of the instrument, relying on their own perception/action feedback loops to keep the system within the group's aesthetic bounds. The explicit signal feedback network becomes entwined with the perception/action feedback loops of the performers as described in [14]. Due to the sometimes unpredictable behavior of feedback networks, our compositions have very loose structures based upon particular repeatable gestures that affect the class of sounds that the instrument generates as opposed to specific musical textures.

3.1 Musical Maneuvers

Open channel With a loop topology, set each $gain_i \approx 0$dB and $delay_i$ so $delay_L$ is above the regime of squealing higher pitch but within the perceptual present, perhaps $10ms < delay_L < 4000ms$. Any ring of instruments in open channel creates a feedback loop, opening musical potential.

Exciting the system Adding any synthetic (perhaps sampled) $new(t)$ to *open channel* inserts energy into the feedback loop that will spread over time growing and/or decaying. This tends to make a musical statement, dependent on choice of material and masking by other sounds.

Passing a sound *Exciting the system* with a sound event whose duration is on par with the typical $delay_i$ and with mild enough transformation that the excitation sound maintains some perceptual identity as it echoes and transforms, creating a theme with variations. This became the standard end goal of the setup procedure (Section 4.1) as well as the opening musical gesture of several performances.

Shaping the tail refers to all modification after *exciting the system* and makes up the majority of a composition's subtleties. Infinite varieties of shaping can occur, including all of the following maneuvers.

Decay $SpectralNorm(M) < 1$ (e.g., $gain_L < 0$dB), so *new* sounds briefly sustain then die away.

Growth $SpectralNorm(M) > 1$ (e.g., $gain_L > 0$dB), so *new* sounds echo louder and louder, possibly leading to *self noise* or *blowing up.*

Self noise *Growth* reamplifying $hwNoise$ to audible levels as a texture or tone, prevented from *blowing up.*

Blowing up *Growth* leading eventually (perhaps quickly) to exhausting a finite resource such as digital conversion limits or available analog current, voltage, or power and therefore some kind of clipping, limiting, or system failure. Sufficiently low $d\text{dB}/dt$ allows performers to control *growth* safely (Section 2.3).

Killing One or more $say_i(t) = 0$ for some duration, perhaps thereby disconnecting a loop, inserting local silence that can have global effects. The extreme antidote to *growth* and *blowing up.*

Clearing a delay line The collective instrument has a distributed memory residing in all the delay lines (plus the inaccessible and negligible hardware delays). Immediately zeroing all of an instrument's delay memory guarantees that the delay line will output silence for a time into the future, literally erasing the instrument's memory, which then immediately begins to refill with $hear_i$. During performance this offers a way to disrupt drones and other continuous sounds; also as a countermeasure against *blowing up.* During rehearsal, particularly practicing beginnings of pieces, it can be necessary to clear all delay lines.

Tempo shift the loop While *passing around a sound* at rhythmic $delay_L$, players may speed up the rhythmic pulse by decreasing $delay_i$ or slow down by increasing. Various methods for changing delay time may impose glitches, transposition, etc. (Section 4.2).

Spreading *Growth* plus nonlinear effects such as waveshaping or rapid delay length modulation that spread energy to other frequencies.

Thinning may refer to reducing temporal or spectral density. Amplitude modulation (e.g., *clearing a delay line*, slow tremolo, stuttering) can thin temporal density. Filters (e.g., notch, lowpass) can thin frequency density.

Pinching means bandpass filtering $hear_i$ to emphasize certain frequencies. Increasing Q and the gain eventually lead to ringing, so these must be continually (often gradually) monitored in the same way as $gain_i$. With practice one can "pull tones out of" sufficiently broadband feedback signals.

Topology changes An arbitrary matrix mixer allows dynamic topology changes; AD/DA implementations can crossfade these. Typically we start with the ring and transition through the Bi-ring to reverse ring (see Figure 1a). Another move starts with a ring and slowly adds additional edges, increasing complexity.

3.2 Composition

The ensemble feedback premise and the desire to begin performances using the technology transparently (so that the audience can understand what's happening) suggest a particular compositional structure that we have used extensively. Start by *passing around a sound.* Initially each instrument performs minimal transformation so that perceptually "the same sound" is going around. Of course any excitation can be chosen, in which case the specific selection can make a strong musical statement. Parsimony suggests the excitation be the loop's "intrinsic" sound, e.g., a burst of heavily amplified $hwNoise$. Another parsimonious choice is the ideal digital impulse, a distinctive click that evokes theoretical discussions of systems' impulse responses.

Thinking of the initial excitation and the network's response as the "first phrase" of a composition suggests starting with *growth* (but avoiding *self noise*) then shifting to *decay* so the continually transforming echoes eventually die away. The "second phrase" can develop in the direction of introducing additional excitation material and/or greater sound transformation. After a few episodes the audience will grasp the relationship among the performers' instruments and the piece can explore other musical maneuvers.

We eventually open up most performances to pure improvisation. We let our group instrument become a group mind, a collective instrument where each member's aesthetic decisions are always weighed against everybody else. Controlling the instrument seems to enter one into a meditative and intuitive practice where the performers' perception/action loops are sometimes enacting very small changes in their instrument control parameters. It is common for an end to the composition to come naturally as a particularly rich phrase decays away.

4. IMPLEMENTATION

We present specifics and practical details of three strategies for implementing the mixing matrix M, followed by general attributes and several specific examples of instruments that work in this framework.

4.1 Setup, Test, and Calibration Procedures

We explored three strategies for implementing the matrix mixer (including routing each instrument's *say* to the loudspeaker system as described in Section 5): direct wiring, a hardware mixing board, and a computer with a multichannel digital audio interface. These represent a continuum of increasing price, flexibility, complication, mix delay d_M, and ability to display input/output signals both for debugging and for tracking performance.

One common issue is that various audio equipment can require different signal levels, but any boost or attenuation to optimize this also affects $gain_L$.

4.1.1 Direct Wiring as Matrix Mixer

Direct wiring is a physical manifestation of M, e.g., wiring each *say* to one particular *hear*. A wire from a to b sets $M(a, b) = 1 - \epsilon$: no gain, miniscule hardware loss; likewise d_M is negligible. Instruments that cannot achieve $gain_i > 0$ may require external amplification or for instrument $i + 1$ to have adjustable input gain. Splitting each *say* also to a loudspeaker may require an external splitter (perhaps a small mixer), but it is often easiest to use left and right outputs if both can be set to the same *say* signal. Direct wiring decentralizes signal metering and works best when instruments with I/O level meters (e.g., laptop-based) alternate with instruments without. As future work we envision a hardware patchbay matrix allowing direct wiring topology changes during performance [1].

Allowing an instrument to *hear* more than one other instrument typically requires a mixer. Otherwise with in degree ≤ 1 there can only be separated graphs, each with a single cycle.

4.1.2 Mixing Board as Matrix Mixer

The aux sends usually provide the most general M on a hardware mixing desk, so say_i goes into mixer input i and $hear_i$ comes from mixer aux output i and the canonical loop has each channel i sending to aux i+1 and channel n sending to aux 1. If more channels are needed then assignment to the stereo and group outputs plus stereo panning provides some flexibility. Each connection through a mixer changes amplitude; usually each knob contributes to one or more entries of M, so even after carefully setting each knob to 0 dB additional manual empirical adjustments are required. To view each *say*, many mixers have a "solo" feature that can assign a given channel to a shared meter. Channel direct outputs are good for splitting each say_i to a personal speaker. Analog mixing boards have $d_M \approx 0$ and can create loops with very low $delay_L$ and very large $d\text{dB}/dt$.

4.1.3 AD/DA and Computer as Matrix Mixer

If external hardware (e.g., a mixer) splits each *say* both into M and into a loudspeaker then with an n-channel audio interface each AD input is a *say* and each DA output is a *hear*.[3] Otherwise if there are $2n$ DA outputs then there can also be one carrying each *say* to a speaker.

Routing everything through a computer opens the possibility of visualizing system parameters for the players and the audience, e.g., graph topolgies, instrument I/O levels or spectra, delay times, transformation parameters, etc.

Our Max/MSP implementation offers real-time control of the network's topology, saving and recalling topological presets and morphing among them in performance.

4.2 Instruments and Technologies

The context of an ensemble feedback network suggests a logical chain of parsimonious instrument design. Section 2 highlights the importance of controllable gain and delay time.[4] Direct low-latency gain control (e.g., a physical gain knob or even a laptop's quieter/louder buttons) maximizes the instrument's responsiveness and the ensemble's ability to handle rapid $d\text{dB}/dt$. Since $delay_i$ is necessary but the exact value is arbitrary, it is most expressive to allow each performer to modulate $delay_i(t)$ in real time (thus requiring interpolated delay reads). Instantaneous discontinuities in $delay_i(t)$ produce a particular glitchy discontinuity in $say_i(t)$, suggesting some kind of slewing mechanism that allows $delay_i(t)$ to be smoother than the performer's sensed control functions. A simple linear ramp that brings $delay_i(t)$ from the current value to each new value over *slew* seconds provides control of this glitchiness and provides a satisfying effect of playing back the delay line memory slower or faster than realtime for the duration of the slew each time a performer exercises control over the delay time. Another obvious choice is for *say* to crossfade between two delay taps, avoiding pitch discontinuity by omitting or repeating a section of delay memory.

We have experimented successfully with a variety of instruments (Figure 3) including off-the-shelf hardware devices (such as a Korg Monotron, Korg Kaoss Pad, and Dave Smith Instruments Mono Evolver) and custom designed hardware and software. Our "reference" embedded instrument is a Pure Data [10] implementation of the above parsimonious instrument, running on a Raspberry Pi with Satellite CCRMA [4] and potentiometers connected to an Arduino controlling input gain, output gain, delay, highpass filter, lowpass filter and slew time for changing delay. Our "reference" browser-based instrument is a simple Gibber [12] program with delay, filter, and gain level indicators alongside a live-coding interface for rapid iteration and experimentation.

Our *Electro Mechanical Chaotic Feedback Oscillator* generates $new(t)$ with an internal feedback loop between a handheld piezoelectric sensor and a mini speaker. Where loop gain is low enough to avoid audio feedback but high enough that the performer excites the system, the sensor will "bounce" chaotically, creating a series of sharp impulses to be filtered.

Custom software instruments in SuperCollider [8] and Max/MSP [17] include internal feedback delays, a bandpass filter bank, a granulator, $delay(t)$ controllable with slewed keypresses plus a controllable LFO, and a first-order cybernetic gain stage that continuously smoothly adjusts a signal's gain in the direction of a (controllable) target RMS value for the sum of $hear_i$ and the signal coming from the delay tap. Another implements a crude single-sinewave analysis/synthesis system using the `fiddle~` pitch detector.

5. SPATIAL SOUND

As in many "laptop orchestras", projecting each instrument's sound from a spatially distinct location greatly assists both performers and audience localize sound [3]. The various network topologies can naturally give rise to various spatialization patterns, e.g., the *passing around a sound* gesture is more dramatic when the sound discernibly travels a spatial path in parallel with the signal processing path.

For smaller venues we recommend having each player's monitor speaker be the only sound source for that person to be heard throughout the venue. In this case all sound would come from the stage, e.g., a semicircle of speakers behind a semicircle of performers.

For larger venues we recommend routing each player's output sound both to a relatively quiet on-stage monitor and also to a full-surround speaker for the audience. The sound's encircling the audience can give a powerful impression of putting them "inside the feedback loop" spatially,

[3]It's comforting to have a bank of analog faders feeding the loudspeakers in case things blow up.

[4]Instruments with very low max delay can work within loops with sufficient total $delay_L$, but obfuscate *passing a sound* because "the sound" comes out of two instruments at perceptually the same time.

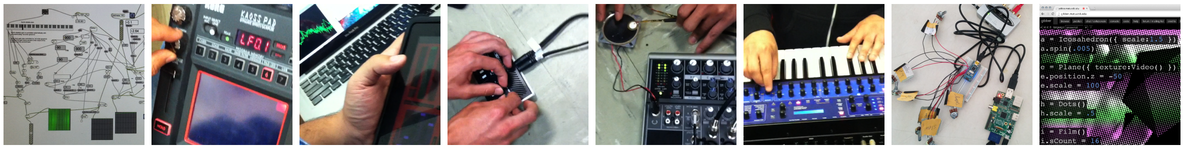


Figure 3: Instruments used (L to R): Max/MSP, Kaoss Pad, SuperCollider controlled by iPad, Monotron, *Electro Mechanical Chaotic Feedback Oscillator*, Mono Evolver, reference embedded instrument, Gibber

with *passing around a sound* possibly corresponding to a sort of circular panning.

6. CONCLUSIONS AND FUTURE WORK

The CREATE Ensemble at UC Santa Barbara used the research described here to give a trio of successful performances, which in turn suggested many interesting future research directions.

The software for the matrix mixer and several individual instruments (Max patch, PD patch, Gibber, and Raspberry Pi) has been released with an open-source license and are available on Github [5] where we invite collaboration on this project. We also envision canonical instruments for the iPad and Teensy3, a fully analog instrument, and an internal multitap delay feedback example instrument. We would like to experiment with homogenous ensembles (everybody playing the same instrument), which may give rise to more "predictable" soundscapes. We plan to try this with completely analog instruments and mix matrix. We will characterize various instruments' noise floors, e.g., as spectra or as the spectra from putting one in a growth loop.

Most of the matrices we have explored are permutation matrices (i.e., looped topologies). We would like to create an algorithm to rotate between different permutation matrices within the required spectral norm. We will also explore more connected matrices where one instrument's *hear* is a sum of multiple instruments' *say*. In particular, circulant matrices have a defined way to calculate the eigenvalues so they can be synthesized with controlled spectral norms.

We plan to implement a multitouch GUI for real-time control of the mix matrix whose behaviour is informed by the FDN stability conditions. The GUI would theoretically only allow transformations that keep the system within some margin of stability.

We would like to explore the politics and dynamics of new graph topologies. We would like to look at intelligent systems like a system that "listens" to the performers, and react accordingly, changing the topology based on "sections" of the composition or a system that changes based on the performers' input signal, self correcting itself, and transforming its output so as not to appear to have a mundane texture.

We are also interested in implementing feedback control and/or intelligence to the system, allowing it to automatically change topologies based on "sections" of a performance, or make aesthetic decisions by transforming its output [11].

We would like to visualize the topology of the feedback networks and how musical information is transformed and transferred. Visual feedback might include a live pole/zero plot and an estimate of the near future system output. Such a visualization would provide useful information to performers in addition to sparking audience insight into the processes at work. One example would be a top projection where the ensemble sits in a circle with audience surrounding the performers. The visualization would show the topology, waveform of input-output/ spectrum, delay time, and innternal feedback parameters projected directly in between the performers.

We look forward to continuing progress on what has been, thus far, a very satisfying musical endeavor.

[5]https://github.com/create-ensemble/feedback

7. REFERENCES

[1] Resonance (magazine). volume 9. London Musician's Collective Ltd., 2002.

[2] K. Aufermann. Feedback and music: you provide the noise, the order comes by itself. *Kybernetes*, 34(3/4):490–496, 2005.

[3] C. Bahn and D. Trueman. Interface: electronic chamber ensemble. In *Proc. NIME*, pages 1–5. National University of Singapore, 2001.

[4] E. Berdahl and W. Ju. Satellite CCRMA: A musical interaction and sound synthesis platform. In *Proc. NIME*, pages 173–178, 2011.

[5] L. Dammbeck. *Das Netz*. b.film Verleih, 2003. Film.

[6] S. Jordà. Multi-user instruments: models, examples and promises. In *Proc. NIME*, pages 23–26. National University of Singapore, 2005.

[7] K. Karplus and A. Strong. Digital synthesis of plucked string and drum timbres. *Comp. Mus. Jrnl.*, 7(2):43–55, 1983.

[8] J. McCartney. Rethinking the computer music language: SuperCollider. *Comp. Mus. Jrnl.*, 26(4):61–68, 2002.

[9] D. Novak. *Japanoise: music at the edge of circulation*. Duke University Press, 2013.

[10] M. Puckette et al. Pure Data: another integrated computer music environment. *Proceedings of the Second Intercollege Computer Music Concerts*, pages 37–41, 1996.

[11] C. Roads. *Composing Electronic Music: A New Aesthetic*. New York, Oxford University Press, 2015 (in press).

[12] C. Roberts, M. Wright, J. Kuchera-Morin, and T. Höllerer. Rapid creation and publication of digital musical instruments. In *Proc. NIME*, 2014.

[13] D. Rocchesso and J. O. Smith. Circulant and elliptic feedback delay networks for artificial reverberation. *IEEE Trans. Speech and Audio Processing*, 5(1):51–63, 1997.

[14] D. Sanfilippo and A. Valle. Feedback systems: An analytical framework. *Comp. Mus. Jrnl.*, 37(2):12–27, 2013.

[15] H. von Foerster. On self-organizing systems and their environments. In *Understanding Understanding: Essays on Cybernetics and Cognition*, pages 1–19. Springer New York, 2003.

[16] D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Comp. Mus. Jrnl.*, 26(3):11–22, 2002.

[17] D. Zicarelli. How I learned to love a program that does nothing. *Comp. Mus. Jrnl.*, 26(4):44–51, 2002.