The Modular Explosion - Deja Vu or Something New?

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

ABSTRACT

Modular Synthesis was supposed to be over by the 80s. Already in the early 70s, the MiniMoog and its default-synthesis-path spawn sounded a gradual death knell for the great modular rigs that grew stronger with microprocessor-driven analog systems that had all their 'patching' managed digitally. As the 80s unfolded, MIDI and digital synthesis seemed to have sealed the modular coffin, and those hulking rigs grew silent, drifting into disrepair and being discarded or sold for a song at auction. Those of us who had working modular rigs would make sheepish excuses for why we still used them. Things have changed - over the last years, there's been an explosion in modular systems, with more companies making them now than in their heyday a half-century ago. What's going on here, and what's special about these things? This paper examines this question from several perspectives, exploring the allure of modular systems, what they're good at, and extrapolating where technology might be bringing them in the relatively near future. I also introduce the large one-of-a-kind modular system that I designed and built between 1975-1988 (composed of circa 125 custom modules) and illustrate some of the more unique modules and how I use them now, including the recent Patchwork system that allows people to interact with it via the web.

1. Analog vs. Modular

At the dawn of commodity commercial digital synthesizers, analog synthesis was thought to be over. The trade magazines were filled with sleek photos of very agile digital synthesizers sporting all manners of LCDs and buttons to interact with elaborate menus, and old analog synthesizers were disposed of as 'junk'. But it seems that we acted too quickly. Infatuated with the crispness and acrobatic novelty in digital sounds, musicians crowded into the digital pool. Listening now to music coming from that era of the 80s, the sounds made by real-time commercial digital synthesizers tend to be brittle, simplistic, and iconic, the former reflecting hardware limitations and the latter perhaps because it was too easy to rely on presets - musicians would just choose their sounds from a menu without any significant tweaking.

Things are different now. Even though embedded digital hardware has advanced enormously and is capable of wielding infinitely more sonic power, analog synthesis has regained respect. The old gear is now tremendously valuable and hybrid synthesizers that have analog and digital sections are common (the old analog synth ASICS, such as the Curtis Electronic Music [CEM] chips are again in production!). Advocates talk lovingly about analog sonic 'haze' and the aesthetic and interesting ways in which analog systems can intrinsically go into overdrive, etc. Analog hardware naturally and even gracefully produces such artifacts since beneficial 'unintended' features, like different manners of progressive distortion, come for free via circuitry operating outside of normal bounds or 'suboptimally' implemented. These artifacts must generally be explicitly built into a digital system. This is a world that effects pedal designers have exploited for many decades [1,2], where boutique manufacturers even scavenge particular vintage transistors for particular sonic characteristics. In principle, however, nearly everything about analog synthesis, even such symbiotic 'nits', can be simulated digitally now at good fidelity and with much more flexibility, soon even learned by example from the original hardware without a detailed pre-defined causal model. Hence the days of mainstream 'analog' may again be numbered.

But the recent resurgence of modular synthesis is different. Modular synthesizers, even dating back to their genesis in the late 60s through mid 70s, weren't purely analog devices. Most had digital functions as well – e.g., sequencers, clocks, and logic gates for trigger conditioning. Even the moniker 'Subtractive Synthesis' that modulars were tarred with for much of their tenure is a diminutive exaggeration – many modules, even in the early days of modulars could add harmonics, distortion, and otherwise thicken up a sound under voltage control, and not just filter complexity away. The fundamental idea behind modular synthesis is not how the noise is made, but rather the concept of patching, where the artist is surrounded by an array of compatible hardware to produce, sculpt, nuance, and control sound, and exploit his or her sensory-motor skills to produce a complex sonic or musical environment. The next sections will explore this in more detail.

2. Where did modular systems come from?

Patching and modularity have very deep roots in both electronics and audio. Patching is most commonly associated with the dawn of telephony well over a century ago, when operators would manually patch calls across lines using cords terminated in what we still commonly call 'phone jacks,' said to go back to 1878 [3]. Patchbays have long been fundamental to radio and recording studios [4] not long after they were enhanced by the vacuum tube, enabling audio to be flexibly routed. Electronic test equipment, from which early electronic music systems were derived, tends to be modular, with each piece of gear realizing a particular set of functions. Analog computers [5] were also established by the 1950s – these enabled engineers to 'program' solutions to differential equations by patching analog integrators, differentiators, and other linear components together via front panel jacks.

Although patching was common in audio and electronics, musical instruments prior to the mid-60s generally were conceived as discrete, stand-alone units. From the Martenot to the Ondioline, from the Novachord to the electronic organ [6], electric instruments tended to be standalone closed systems that made their sounds with their own control interface (rare distributed instruments like the Choralcelo [7] existed from 1910, but this was mainly a single control console that drove distributed electromechanical synthesizer/transducers, and, as extensions were wired in by the company, it wasn't open-ended and reconfigurable). The notion of a musical instrument as a 'kit of user-reconfigurable parts' was an alien one, perhaps because pre-existing acoustic musical instruments, which

Presented at the Voltage Connect Conference, March 10-11, 2017. Berklee College of Music, Boston MA

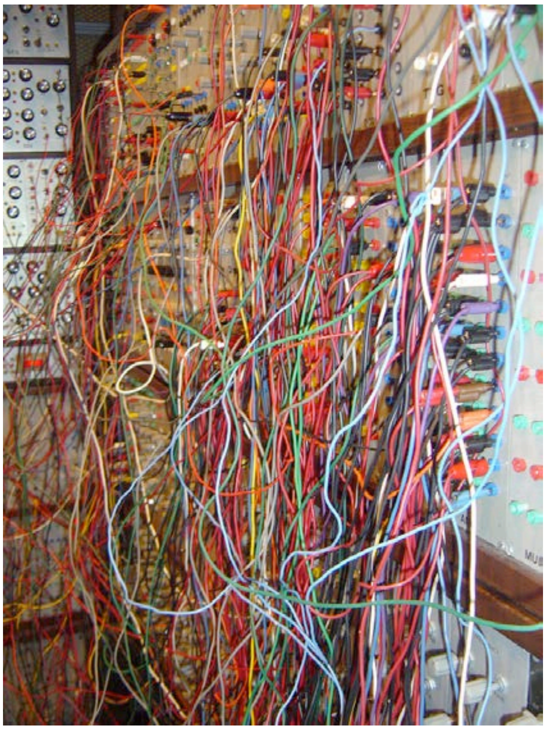

Figure 2: A large patch in my modular system

inspired many early electronic instruments, themselves tend to be standalone discrete devices.

Modular synthesizers changed that [8]. The notion of modularity, patchability, and voltage control that Bob Moog [9] and Don Buchla [10] brought to the audio forefront in the mid-60s opened up a new era in electronic sound. Flexible routing of audio already existed in studio patchbays, but now an artist could route control signals as well. Moog's 'One Volt per Octave' exponential control standard enabled oscillators, filters, and other audio processors to be collectively used or voiced in an optionally tonal, musically-relevant fashion.

Control and audio signals in a modular system are separated mainly by frequency (traditional note/timbre/amplitude controls are much slower than audio), but in principle, this is a false distinction. There is sufficient bandwidth in most control functions in even a vintage modular system to enable 'control' signals to also be in the audio band or even higher, enabling various kinds of modulation. This was a revelation for me, as I thought of audio and control as being separate until I got a tour of a Serge Modular by the designer's brother Ivan Tcherepnin at Harvard in 1974.

Such open patching produces extreme flexibility, as modules that may have a set of canonical purposes can be creatively pushed into different sonic territory by patching them differently or combining with other modules in new ways. And rather than defining an entire instrument all in one closed unit, a modular system in principle can absorb anything within reach of a patchcord – modules can be designed separately and with limited scope – their power comes when they patch together.

In many ways, modular systems were a bold statement illustrating that new possibilities in electronics enabled so much innovation in music that it wasn't clear what a default 'patch' would be. The MiniMoog and Arp Odyssey annealed this open complexity down to a default synthesis patch in 1970-72, although for a while then we had default-path systems that could be richly patch-overridden, like the ARP 2600, the Korg MS-20, and the ElectroComp EML-101. The Oberheim 4-

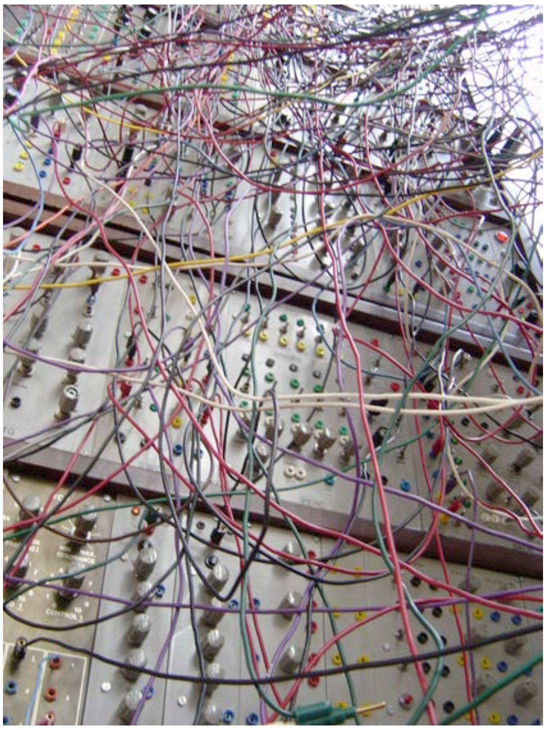

Figure 1: Another day, another patch

Voice and the other true polyphonic synthesizers that followed effectively ended patching, as cord-based patching is very cumbersome if not infeasible for polyphonic synthesizers, where all 'patches' were handled digitally and/or via internal analog switches (duophonic analog keyboards [11] existed, but that was pretty much the limit for patching).

Modular synthesis may well have been more influenced by embryonic digital synthesis than analog computers and audio patchbays. Miller Puckette, the pioneer of graphical dataflow frameworks for music (encompassing MAX, MAX/MSP, and PureData) [12], once told me that he had heard that modular synthesis was inspired by Bob Moog having seen Max Mathew's Unit Generator programming framework [13] in his youth. As the legendary pioneer of computer music at Bell Labs dating to the 1950s, Max conceptualized his non-real-time sound-generating programs as a flow chart of communicating software modules (oscillators, amplitude gates, filters, envelope generators, etc. – a.k.a. unit generators) that would then be coded up in one of his MUSIC languages (Unit Generators first appeared in Max's MUSIC 3 in 1960 [14]). In many ways, these anticipate the patch diagrams and modular architecture of a hardware synthesizer.

3. Just what's so special about modulars?

A decade or two ago, those of us who had large modular systems were embarrassed to admit it. Now they are sources of pride. Even more, there is an implicit gear boom here, as more manufactures are making and selling modules now than in their heyday of the 1970s. The surviving synthesizer designers from a half-century back are again happily and profitably designing Eurorack modules (the current physical module standard that allows hardware from different manufacturers to be housed in the same enclosure while exploiting standard connectors and voltage levels), although they share today's stage with a panoply of young designers who are creating a wild and baroque market of modules often very different from what were around in the beginning. At the expense of potentially

Figure 3: My full synthesizer rig as of 2012, set up at the MIT Museum.

enormous clutter and fragility (in addition to the sheer mass of gear on your back), musicians of all sorts are again embracing modular synthesis for sound design, installations, and even live performance. But what's special about these systems? Why are they back? I'll try to list a few of my favorite reasons below, in no particular order.

3.1 Modulars as highly tangible experiences

The Graphical User Interface (or GUI), inspired by Doug Englebart and first realized by Alan Kay and collaborators at Xerox PARC, has dominated modern computing ever since it was popularized by the Apple Macintosh in the mid 1980s. Despite its success, the GUI has long been challenged in Human-Computer Interface (HCI or CHI) research as being too limited an abstraction – two fingers on a mouse (or touch screen) interacting with planar pixels is a poor substitute for engagement in the real 3D world. The main response to the GUI began with 'Graspable User Interfaces' from Fitzmaurice, Ishii, and Buxton in the mid 90s [15] and now resides in the burgeoning field of 'Tangible Interfaces,' [16] led mainly by my colleague Hiroshi Ishii at the MIT Media Lab. Many examples of tangible interfaces have been created to illustrate how well adapted they are to natural interaction. Perhaps the ultimate expression of tangible interfaces lie with 'Ubiquitous Computing,' the widespread diffusion of computer interaction into everyday objects and environments as predicted by Mark Weiser at Xerox PARC in 1989 [17] and now better known as the Internet of Things. Although many attempts have been made to define generic families of Tangible Interfaces [18], they tend to lead to clutter – graphical pixels can appear, disappear, or change at will, but physical matter is still far from exhibiting such malleability.

Tangible interfaces, however, can lead to deeper levels of user engagement. And modular synthesizers could be considered to be large tangible interfaces. Modulars offer a very different way of interacting with a machine to design and produce sound in contrast with drawing or typing at computer screens or going through cascading menus to find particular parameters to adjust. Everything you can do is in front of you, not hidden behind a screen, tab, or menu item, and the artist has hands directly on the cables, knobs, buttons, and other artifacts of sound production, exploiting their sensorimotor skills and spatial memory (I can remember the details of a synthesizer patch involving several hundred patchcords and well over a hundred modules for at least a week after I make it). It's a rich, highly-engrossing, consuming, immersive, immediate, and serendipitous environment. Unpatched modules suggest ideas waiting to be realized and flickering LEDs suggest trigger sources or control variations. Switches and knobs on every parameter give immediate response. Even though graphical patching frameworks like Max/MSP, PureData, Reaktor, etc. promise similar functionality and are widely and enthusiastically used, they still live on a flat GUI with limited real-estate.

Modular systems suffer the bane of tangible interfaces, however, in that they don't generalize well (even though there may be a similar processor behind the panel of many modules these days, each module does something different). This leads to physical clutter and considerable expense, as each module must be manufactured or constructed, and many are needed to form a versatile system. Software 'modules' of course can be cloned, created, downloaded, stored, and deleted with little consequence, as the number of simultaneous real-time modules is limited only by the capacity of the host computer. Nonetheless, just as mixing boards haven't yet yielded their physical clutter of knobs and buttons effectively to a pure GUI, modular synthesizers provide a level of immersion and immediacy that GUIs can't attain.

3.2 Modulars present an awesome spectacle

Although as mentioned above, modulars can involve clutter, they can be amazing to behold. Big modular rigs are beautiful, with their multitude of knobs and switches, blinking LEDs everywhere, and (often multicolored) patchcords galore.

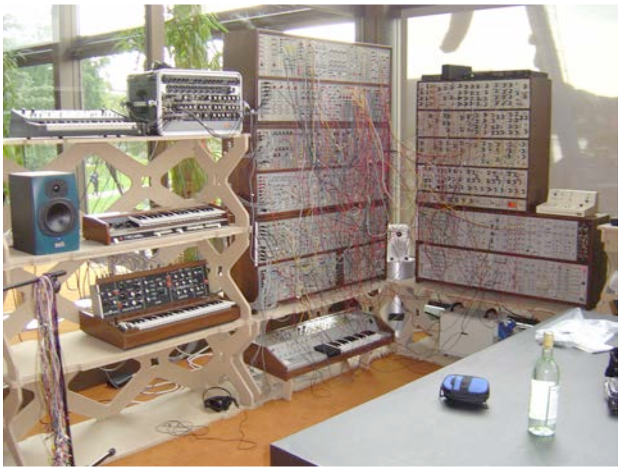

Figure 4: The Synth setup at Ars Electronica in 2004

Modulars dominate vintage synthesizer picturebooks [e.g., 19]. Especially in the current era of sleek stage rigs (where the typical couple of keyboard synthesizers on display are fairly minimal and look more or less the same), or worse-yet, laptop music, a modular rig's complexity is a source of fascination for the audience and even the performer. The public has regularly expressed amazement at installations I've done with my large modular system, excited by the appearance of extreme-scale hardware making real-time music. This phenomenon dates to concerts from the 1970s, when well-known progressive rock musicians such as (most notoriously) Keith Emerson [20] or popular electronic artists like Klaus Schulze or Tangerine Dream would famously surround themselves with hulking synthesizer rigs onstage (we've even seen this in academic music, with custom behemoths like Sal Matirano's SalMar Construction [21]). Audiences would thrill not only to the then unusual sounds that were produced, but also to the sight of the musician interacting with such a large and complicated machine, a technological shaman coaxing audio magic out of their mysterious 'bonfire' of LEDs and cables [22]. The tendency of famous legacy synthesists to surround themselves with mountains of gear has certainly been fodder for clever satire (e.g., [23,24]), but has resurged as musicians again bring modulars onstage in popular genres ranging from improvisational electronic music to neopsychedelic/space rock and acid jazz.

3.3 Modulars stoke OCD collection behavior

I posit that the incidence of obsessive-compulsive behavior is probably higher in prog-rock musicians/fans than in the general populace. Although I'm already over-reaching here, I can certainly generalize this to modular synthesizer owners (alas, prog fans by no means dominate this genre any longer). Together with pedal-obsessed guitarists, modular synth devotees are driven to expand their rig by adding new units that do different things, as conveyed nicely in recent documentaries [25,26]. Or perhaps they just like adding new units for the sake of adding new units. When I was building my large modular system, ideas for new modules kept coming – after only a year of 'spare time' had passed at the peak of this, I had built over 80 different modules. There is an implicit gear race here, as more manufactures are making modules now than ever before, offering plenty of fodder for obsessive collection behavior. Modules are cheaper in normalized dollars now due to Chinese (etc.) manufacturing, economies of scale, advances in PCB fabrication, widespread/free electronics CAD, offloaded distribution, online sales, etc. And in a modular system, more hardware means more sound and complexity, hence you never run out of compositional reasons to add (physical, temporal, and financial constraints otherwise provide wakeup calls).

Figure 5: The innards of a handmade module

3.4 Modulars encourage innovation

This happens at a few levels. First, the encapsulation of discrete concepts into a module encourages proliferation – when designing a module, you don't need to make an entire synthesizer, but just realize a single concept that can be arguably wild and unconventional. Module designers are by default 'lead users' [27] – usually musicians themselves who push original ideas or innovatively tweak old ones to create new sonic capabilities that they would want to have. Online forums give immediate exposure and enable an ecosystem for cross-fertilization of ideas. This may have commonality with software plug-ins, but, as mentioned earlier, the physical sight of the modules arrayed around you strongly suggests ideas of how to use them. I believe that seeing many real objects is more serendipitous than finding them on a flat screen or pulling them down from a menu.

A huge differentiation also exists in the inability of modular systems to effectively store their state. In the early days of modular systems, people would make 'patching diagrams' to log the dispensation of each patch cable and knob setting used so that the patch could later be reconstructed. Except for very simple patches, this is a very futile exercise. My routine patches involve hundreds of cords, and small changes in adjustments can produce very different effects, hence thorough logging is generally not effective. Modular systems just don't lend themselves to preserving their setup, a capability that's endemic to digital synthesis. This can be seen as a downside of modulars (some musicians tape foam blocks over their 'optimum' patches so they can bring their rig from gig to gig with some degree of repeatability), but I see it as a mandate to create. Modular systems have no presets except for patching habits you've accumulated, and there are always easy ways of sidestepping these to mix things up. When you're faced with an unpatched device, you are presented with enormous opportunity to do something new. You have no choice really – a complex patch will never be the same, hence you're always voyaging into new territory.

3.5 Modules don't go obsolete

In principle, a module is always useful. The 1 V/Octave, more-or-less 'line-level' audio, and 5-volt logic standards that modulars have pretty much held to for over a half-century still live on in today's EuroRack rigs. There is no complex digital interconnection or host operating system that will become obsolete. Even though MIDI cables are still around, for example, their days are numbered as USB takes over and new

Figure 7: FeelIO - Wearable robot arm with a knob [46]

protocols emerge. The module does what it does, and barring physical hardware degradation, it will always do that, and can still be used with both old and new modules. There has been evolution in connector standards – modulars of yesteryear started out using ¼” phone jacks, but also incorporated banana plugs (Buchla and Serge devices) and pin jacks (PAIA and my own system). These have all yielded to today’s dominant $^1/_8$” mini phone jack. As connectors can be easily swapped out or converted with simple adaptors, this is not a significant interoperation issue.

When I built the main sections of my modular system in the early 80s, I considered making a microprocessor-controlled digital synthesizer. ‘Homebrew’ computer-controlled, keyboard-interfaced digital wave engines were then starting to appear (e.g., made by people like Bill Buxton [28] and Roger Powell), producing devices along the lines of the PPG Wave Computer. Although I built a computer-controlled analog synthesizer into a CAMAC module hidden in the midst of a high-energy physics experiment at CERN in 1980 [29], I decided to forego that complex route and concentrate on modules instead. Many interesting chips were coming out then that could be encapsulated into a functional module within a day or two of work (see Sec. 5) and used immediately. If I had built the digital synthesizer, it would involve a long development effort and would have rapidly become obsolete. My modules, however, are still viable, current, and very much in use.

3.6 Modulars engage ‘both parts of the brain’

When I’m engaged in building a modular patch, I’m working both as a hardware/computer engineer and a musician/composer. My technical side is essentially designing a unique hardware state machine and sound processor (what I often think of as a musical ‘Rube Goldberg’ machine) to realize the sounds that my artistic side wants to hear. Because of this, composing complex synth patches is the most engaging activity that I know of. Hours pass with little consequence, and I become totally focused on forging, then refining the patch to become what I want it to be. This is not engineering as done by evolving diagrams/math and running simulations/CAD, nor is it composition done note-by-note on pen and paper or computer. This is sheer in-the-moment improvisation at both technical and aesthetic ends, using long-honed intuitive skills developed across the board. I see it as audio/technical ‘sculpture-on-demand’, where you quickly chisel out a rough version of what the piece could be, then finely subtract and add detail that anneal it to perfection. Hours and days can pass in a haze as all you really see, hear, and think about is what the patch is becoming. For musical/technical folk, modulars offer an extreme state of flow [30].

Figure 6: SoundStage - Synth Patching in VR [47]

4. What comes next?

The physicality of modular systems is at the heart of their appeal, as outlined above. Over the last decade or so, a variety of research projects and products have looked at incorporating aspects of modular synthesizers into other musical devices. The graphical dataflow music software environments were already discussed above. Although these exist for the most part only on a computer, hardware has been designed to bridge the gap, where software patches composed in PureData or other frameworks can be downloaded into embedded audio processors that just run the patch in real time. There are many of these now, including the Nord Modular System [31], Bela [32], the Organelle [33], the OWL [34], the Axoloti [35], TeensyAudio [36], and even a legacy project in my own group aimed at running PureData patches on early smartphones [37]. These constructs, however, are mostly aimed at performing musicians or practicing artists who want the convenience of not having to use a laptop or computer in their concert or installation, but rather bring a ‘closed box’ that just runs their patch and features the knobs, buttons, or controller/interface that they want. The complexity of patching stays in dataflow software running under a GUI on an offline computer – the ‘box’ is mainly a way to present a patch as a closed audio ‘product’. Accordingly, although these projects serve good and useful purposes, they do not bring patching into the physical world, hence don’t offer the affordances listed in the last section and don’t really bear on modular synthesizers.

Over the last two decades, compact modular musical concepts have appeared as blocks that can be snapped or clicked together in different geometries. An early digital version of this was the BlockJam from Sony/CSL [38], where all synthesis was done in a host computer and the blocks were identical pieces of hardware with displays and a tactile interface that were programmed for different control functions and could be arbitrarily assembled to realize complex sequencers. Another version that more directly embodied synthesis was the SoundBlocks project by Harrison [39] and McPherson [40]. Perhaps the best known recent manifestation of this idea is the LittleBits – a snap-together modular synthesizer toy launched by Media Lab alumna Adiar Bdyer and her associates in collaboration with Korg [41]. Recent variations on this theme include the Patchblocks [42] – these are a set of common digital modules that attain individual function through code downloaded into them and can be used standalone or physically ganged together.

Projects and products like these focus more on topologies of adjacent blocks as opposed to patch cords that are free to connect anywhere. A very recent project that addresses that in a new yet still compact way has been recently completed by Jie Qi in my research group. Her ‘Code Collage’ project [43] realizes modules as a library of flat electronic stickers that can be put on paper, with ‘patches’ made with copper tape between

Figure 8: The Synth Rig in my Living Room circa 1986

sticker patch points (if patches cross, they can be insulated from one another by a piece of standard sticky tape) – in principle, your synthesizer can become a uniquely crafted book, with common signals spanning several functional pages bussed across the spine.

As interesting and diverse as these projects are, none push into the dense forest of patches and maze of modules that make working with a modular uniquely engaging. I argue that modular systems more-or-less as we know them now will persist until we can push Virtual Reality to the point where virtual environments have a similar level of tactile complexity as we see in the real world – only then will we have such detailed emulated physical diversity without the clutter.

A few recent projects already hint at what this could be. Starting with UPF's Reactable [44] in the mid 2000s, GUI-driven modular architectures emerged from computer screens and moved to tabletop projections controlled by touch and hand gestures. The recent ubiquity of affordable and high-fidelity VR glasses with full head and hand tracking have ushered in the beginnings of virtual modular music architectures. Some examples include FractOSC (not really a full modular environment but a game that progressively unlocks different sequencers and synths that build out a virtual studio) [45] and SoundStage [46], which more closely approaches a real modular synth realized in VR.

These incarnations will improve as new ideas are introduced, VR experiences are refined, and technology advances. But modular synth patching in the real world is also a very haptic experience – sensory-motor capability is critical for efficient and effective interaction as you feel you fingers on the real knobs and patchcords. The above VR experiences all lack this – you are gesturing in the air, and experiencing only visual and audio indicators as you 'grasp' a cord or knob or hit a button. To cure this, one of my students suggested a generic 'patch panel' with a knob, switch, button, jack depression, etc. mounted on a robot arm that moves in advance so that the right 'example' of a controller is present at the right place when your hand approaches. For a decent-sized rig, this would require a large arm capable of rapid and wide translation – a complex and potentially dangerous device. The 'FeelIO' demo by Xavier Benavides Palos in the Fluid Interfaces Group at the MIT Media Lab hints at an intriguing solution here [47]. Instead of mounting the exemplar controllers on a standalone robot, they could be mounted on lightweight robot arms attached to the user's hand. When the tracked hand approaches the location of a controller, the arm with the proper device (knob, switch, button, 'jack') swings into position so it can be grasped and adjusted as if you're using the virtual item. This way the amount of needed physical translation is limited, and the tactile experience of the controllers follows you everywhere the VR experience persists.

5. The Paradiso Synthesizer

By the time I turned eighteen, I was determined to build a modular synthesizer. And indeed I did, in several spurts over about fifteen years, from 1974 to 1988. It grew to encompass over 125 modules and absorbed over six commercial synthesizers, filling a good-sized room and gaining some notoriety as a 'monster synth' in the musical press [48]. The story of its development has been told [49,50], and high-level information with sound examples is posted online [51,52] with a subset of early schematics [53]. Although I occasionally put together exotic keyboard patches for live, expressive playing, I mostly build huge autonomous musical state machines with my rig, where it produces a very complex musical or sound-art

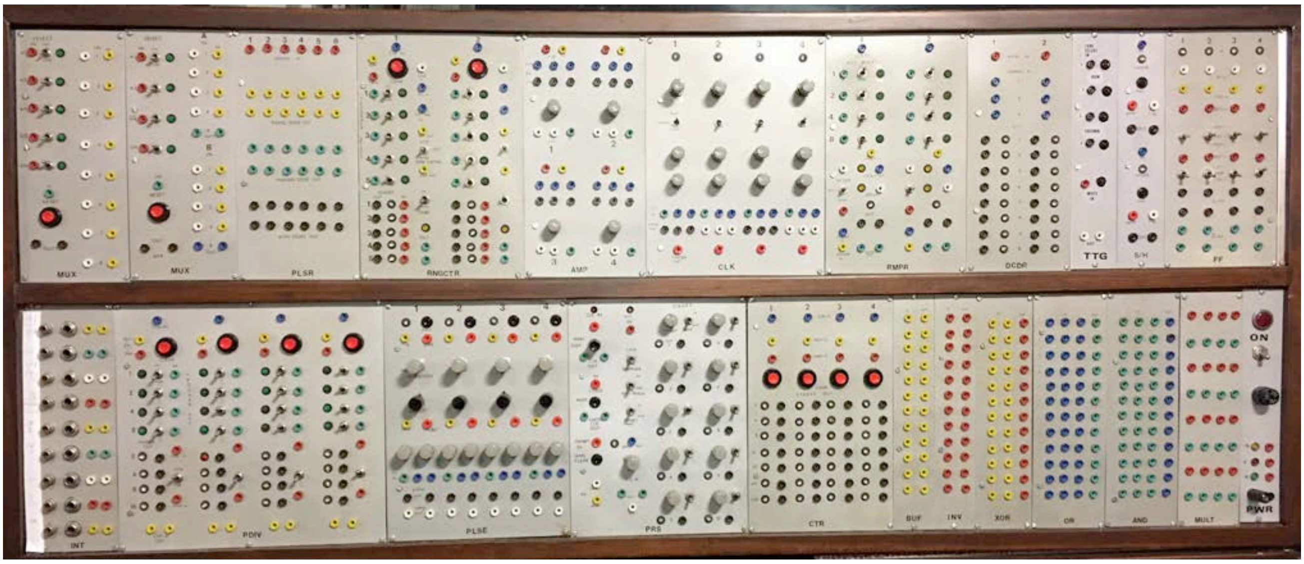

Figure 9: The Logic Cabinet - most of these modules perform mainly digital functions

stream that goes on forever. Each patch has a particular flavor, but its sounds and motifs are always changing and never repeat. In the following sections, I'll overview a few of my synthesizer's particulars, getting into a bit of technical detail and highlighting some of the more unusual modules and 'circuit-bending' implementations to illustrate what 'hobbyist' synth building was like at the time, along with insights accumulated along the way.

5.1 Inspirations, Inputs, and Outputs

As there was no internet in those days, I started learning about the innards of synthesizers from electronics hobbyist magazines and a few papers in the Journal of the Audio Engineering Society. A few of these were pivotal for me – e.g., Don Lancaster's seminal article 'ICs for Electronic Music' [54] provided inspiration for many modules, and the series of Radio-Electronics articles detailing the PAIA synthesizer by John Simonton [55,59] started me off (the early PAIA units tended to have compromised quality, but I improved their design somewhat before formalizing them into modules). I didn't know about Bernie Hutchins' Electronotes [56] until years later.

As the output impedance from all modules was low, I used unshielded pin jacks (a legacy of its early PAIA origins) for everything without much fear of crosstalk, incorporating the lesson I got on the fluid interchange of control and audio signals from Ivan Tcherepnin during my 1974 Serge demo. I use cables with stacking plugs, allowing outputs to be easily fanned across as many inputs as desired. Pin jacks are nicely compact, but friction-fit however, and some have become prone to spontaneously sliding out – a defect avoided by the mini phone jacks dominant in today's modular systems. The voltage levels are standard (logic levels are 5V CMOS and analog outputs range circa +/- 10 V), but all logic inputs are conditioned with small resistors and Zener diode clamps to prevent damage from applied overvoltage. This way, any output, analog or digital, can be connected to any kind of input without risk. Many modules exclusive-OR their digital inputs with a logic 0/1 switch – this way, without an input applied, the switch allows the bit to be manually set or reset, and when an input is provided, the same switch provides a normal-or-invert function (these modules also display input state via LEDs). Similarly, several of the modules have analog control inputs conditioned by a potentiometer that is 'null' at 12 O'Clock, and increases positive gain with clockwise rotation and inverted gain with counterclockwise rotation.

5.2 Canonical Modules

As usual, my system contains many precision VCOs, VCFs, VCAs, etc. My initial VCO designs from the mid-70s used homemade exponentiators [57] with 8038 function generator chips (influenced by the Lancaster article [54]), getting me at best a few usable octaves of musical range. These were all traded out for reverse-engineered Aries modules some years later, which in turn were replaced with custom designed boards using CEM 3340 chips in the late 80s, which remain to this day. The VCFs are a potpourri of designs, with several derived from ARP's multimodal state-variable filter designed by Dennis Colin [58]. VCAs are incarnated in a variety of modes – a couple have the PAIA discrete Gilbert Cell design [59] many are based on CA3080 OTAs, one comes from a voltage-controlled tone/volume/balance chip (LM1035) made for automotive audio systems, others can be made from modules based around embedded NE572 compander chips or MC1496 balanced modulators. One module sports two 6-input VCAs made from biased A-series CMOS buffers, as suggested by Lancaster [54]. LFOs can be realized by switching precision VCOs into low range – there are also many dedicated LFOs scattered throughout the system and realized in different ways. All are voltage-controlled. One module sports quad LFOs based around the LM566 with triangle and pulse outputs (these are often used as clock sources). Dedicated envelope generators come from the PAIA Attack/Decay [59] and ADSR units (these are modified to have voltage-controlled attack and decay or release, and the former add a variety of trigger outputs and reset inputs), and additional envelope generators are embedded into other modules. There are several multi-channel analog mixers in my system, with each input selectable as normal or inverted (8 of the mixer channels also have ¼" outputs to easily interface to external equipment). There are 2 stereo output units that feature integral multichannel mixers, and equalization or switchable stereo enhancement. The synthesizer sports two channels of analog noise coming from transistor breakdown [59] filtered into white/pink/and random DC voltage as well as a fixed and an adjustable digital noise source. The synth has 4 sample/holds, 2 exponential lags with

Figure 11: The Phoneme Generator module, with XOR'ed switches and input bits at left

adjustable attack/decay, and 2 channels of linear slew with voltage-controllable rate. For interfacing external audio, the synth sports a dual microphone preamp, 3 envelope followers, and a frequency-to-voltage converter with 1V/octave output.

5.3 Logic, sequencing, and routing modules

Logic circuitry is critical in generating the complex state machines that drive my most elaborate patches. This connectionist architecture is evident when looking at a complicated patch – cables tend to cluster heavily around the logic cabinet, where a multitude of dependent triggers are combined, processed, and split, forming the patch's 'brain'. Logic modules include standard AND, OR, XOR, NOT gates, as well as flip-flops (which can be switched into RS, D, or T modes), binary dividers (ripple and programmable), binary rate multipliers, binary ring counters, binary encoder/decoders, etc.

Triggers and gates can be generated by a quad voltage-controlled monostable module based around the LM322 (producing both pulse and ramp outputs) [60], as well as another module based around an array of five adjustable 555 timers. Probabilistic triggers can come from a wide variety of sources (e.g., by putting clocked counter outputs into coincidence with other signals to produce pulses that stochastically fire at a desired average rate, or from a voltage-controlled random trigger device made from a PAIA electronic windchimes unit [61], which also features a set of adjustable ringdown filters to emulate the chimes).

There are three dedicated ring-counter-style sequencers in my system. One is based around the PAIA 12-stage analog sequencer, while the other two are custom-designed shift-register-based devices (one 9 stage and the other 10-stage) with selectable XOR feedback at each stage for pseudorandom capability. The pseudorandom devices can be run at very high frequency to generate complex digital audio waveforms at their analog summing output (inspired by [62]), which can be dynamically modulated by XOR'ing external bits into the feedback train or routing different shift register output bits back to its input. The logic cabinet also houses a dual 5-stage ring counter module that can be used for arbitrary purposes.

The synthesizer also contains a set of bi-directional analog multiplexers and analog switches, enabling signals of any sort

Figure 10: The Voltage-Controlled Chaos and Alarms Modules (left) and a Phase-Locked Loop (Right)

to be dynamically routed to different sources or destinations as well as 2 voltage-controlled stereo panners.

Very complex behavior can be generated by feeding signals output from such simple logic chips through combinatorial logic networks back to their inputs. When clocked at audio rates, this produces complex 'musical' sequences, and I have put together lab assignments for students in my 1999-2000-era electronics class around this idea, where they are asked to produce such a musical state machine on their own using a set of CMOS 4000-series logic chips on a breadboard [63]. In this case, the chips act as the synthesizer's logic 'modules', and patching happens by plugging stripped wire into breadboard holes (a patching technique exploited now by so-called 'micro/nano modular' or breadboard synthesizers, such as the Casper Electronics NovaDrone [64], the SoundMachine NanoSynth [65], etc.).

5.4 Boutique modules

There are many modules on this synthesizer that are somewhat unusual. The original cabinet contains two phase-locked loops (PLLs), again inspired by the classic Lancaster article [54]. These are based around the 4046, and produce a square wave output that 'chases' the dominant frequency of an input signal. The tracking lag and damping can be widely adjusted (soon these will be voltage and gate controlled, as I implement a few long-desired mods). I find these modules to be extremely useful in 'dirtying' a lead voice, for example – e.g., the tracking can be adjusted to be barely stable, so different notes lock on different harmonics and/or the PLL's oscillator can be very jittery or even 'gritty' in an interesting way. The PLLs can be cascaded, or their control loop can be externally closed, enabling elements like programmable dividers or other functions to be introduced into their feedback network and causing the PLL to track at higher harmonics. As also suggested by Lancaster [54], I have module based around a top-octave generator driven by a PLL that can follow an input signal with an entire chord, assembled from any of the 12 notes coming from the octave divider's taps as mixed in an integral analog summer.

Another interesting module produces voltage-controlled chaotic sequences. Based around a clocked, sequentially-sampled circuit [66] that iterates the logistic map, this device produces an output waveform that can be driven through bifurcations and into classical chaos. And of course, this 'chaos factor' can be voltage controlled. When clocked at sequencer rates and used as a VCO control source, the produced pitch train can sound 'riffy' at low chaos levels and 'random' when deep into chaos. When driven directly at audio

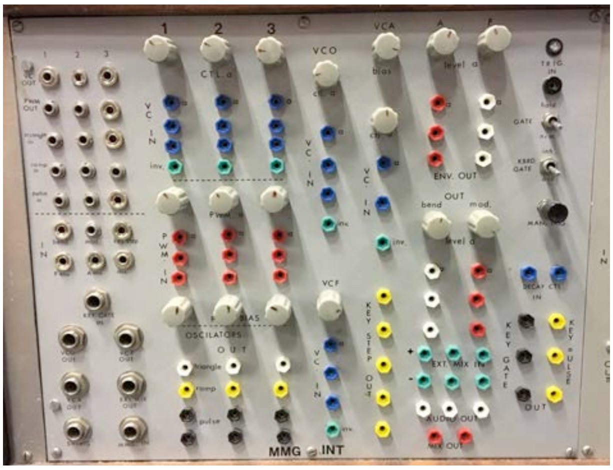

Figure 14: The MiniMoog Interface module - the umbilical cables running to the Minimoog attach at left

Figure 13: Umbilical cables connecting to customized patchpoints at the rear of the MG-1. The other end of this cable runs to an Interface module analogous to that above

rates, the output produces a very complex and natural-sounding gurgling 'fuzz' that's delightfully intermittent at low chaos, smoothly degenerating into digital noise at higher levels.

Many modules leverage the 'personality' of particular integrated circuits that were coming available at the time and had great implication for sound generation. One of the most versatile devices in my synthesizer is based around a Votrax SC01 phoneme synthesizer chip [67]. Before computers could talk directly via software, they fed bit sequences to discrete speech synthesizers that generated and gracefully transitioned between sequential phonemes to form words. The SC01 takes a 6-bit phoneme input with 2 bits of inflection – when hit with a gate, the bits are latched and the phoneme is produced. Of course, these bits can come from a sequence or other processes running on the synth, causing it to babble in accordance. My speech module is much more versatile, however - the pitch of the phoneme synthesizer can be voltage controlled or it can be driven by a PLL, so the phoneme being produced tries to follow an input audio frequency – also, the output can be arbitrarily discriminated, forming a richly harmonic and distorted speechlike sound that can be effectively filtered and processed.

Other modules are based around analog delay lines (again inspired by that classic Lancaster article [54]) that can be controlled and modulated in many ways, and the synthesizer also contains a DTMF dialer – yes, you can feed it bits from a sequencer and it is able to dial a phone. One module is based around a MM5871 rhythm generator chip as was used in cheap drum machines, together with a variety of ringdown filters and

Figure 15: The Assimilated VL-Tone (the original device is embedded at lower left)

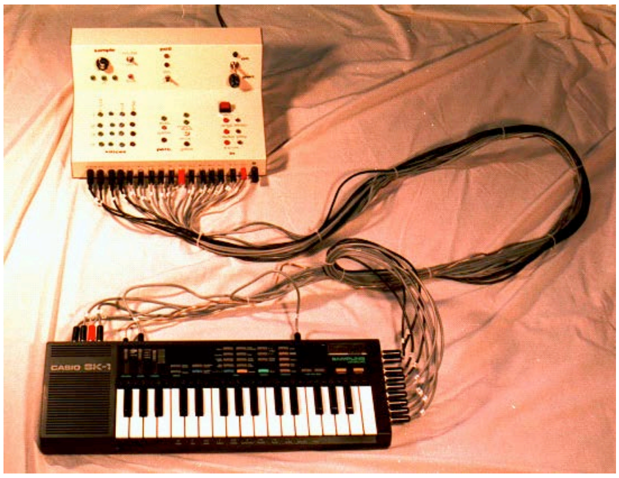

Figure 12: The SKI and its standalone Interface module

a noise source for the percussive voices, while another incorporates a SN76477 sound-generator – essentially a simple synthesizer-on-a chip that was used for audio generation in early personal computers. There are also modules based around multilevel window comparators (which trigger at particular voltages) and analog/digital converters (ADCs). The bits output from these devices are analog-summed with normal/inverted adjustable weights, producing a complex output mapping to a monotonically increasing input. This can turn, for example, a triangle wave into a very harmonically rich pulse train.

Several modules incorporate entire devices from that era – e.g., the synth contains a vintage Wurlizer Fuzzbox, a Lafayette spring reverb, and MXR phase shifter. These have all been modified to be voltage-controlled (the former two with a simple NPN transistor acting as an adjustable wet/dry bypass). The synth contains two analog pitch shifters derived from circuitry used in variable-speed tape players [68] that clock analog delays at linearly ramping frequencies, shifting audio out at progressively lower frequencies (with a bit of clever blanking at the reset). A unique module contains a set of 8 alarm oscillator cards, each of which makes a different kind of alert sound (e.g., sounding like various emergency vehicles). These can be gated on or off into each of two channels by logic signals and combined as a linear sum or via an XOR, which produces a strong ring-modulation effect. The oscillator in each sound card can also be independently voltage controlled.

5.5 Circuit bending and synthesizer assimilation

If I bought a standalone synthesizer of any sort while I was building my system, I designed a module to gracefully and seamlessly assimilate it into my system. During this period, my rig incorporated a MiniMoog, a Moog Satellite, a Radio Shack/Moog MG1, a Casio VL-Tone (the first 'toy' digital synthesizer), a Casio SK1 (the first 'toy' sampler), and to some

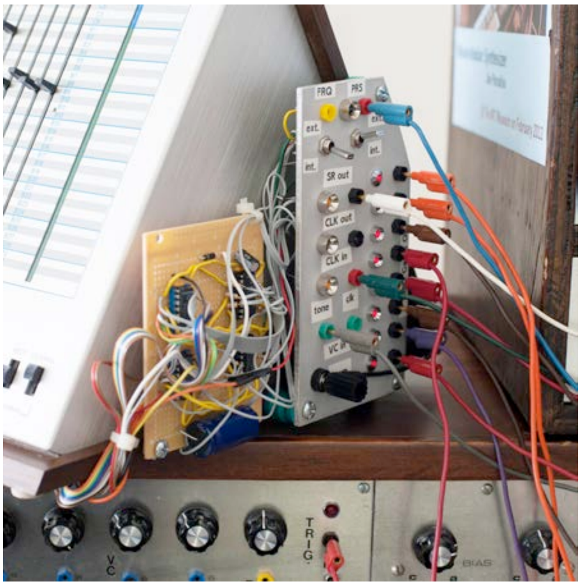

Figure 16: The Triadex Muse's Interface Panel

extent a Casio CS-101 (the early digital ‘home’ keyboard, not the later famous MIDI-capable CZ-101), an Optigan (a consumer sampler using optical soundtrack disks instead of tape racks), and a Wurlitzer (formerly Kalamazoo) organ. In recent years, the synth has also absorbed a Minsky/Fredkin Triadex Muse (the first standalone sequencer instrument).

My philosophy on synthesizer assimilation differs considerably from the standard practice of what is now called ‘circuit bending’ [69]. Although they can produce fun constructs yielding complex sounds, much of the circuit bending profession is empirically driven with limited engineering insight, randomly shorting pins together or pulling out signals without much knowledge of what they are, in order to cause the device to ‘glitch’ in an interesting way or produce an entirely different kind of sound. Accordingly, most circuit-benders produce unique stand-alone instruments that usually aren’t meant to interconnect with others in any significant way (as an aside, I find the sounds these instruments make to be often too consistently edgy and jarring because of the dominance of pulse waveforms – they ‘scream’ to me for further processing to dynamically smooth and nuance their timbres).

In my assimilation practice, I don’t see these synthesizers as being separate – I rather see them as interconnecting deeply with my modular – making them in effect compatible ‘modules’ that can be freely patched with the rest of my system. If I can’t get a comprehensive manual, I probe the circuitry thoroughly, and develop significant knowledge of what different elements do and what different signals are. I then select particular signals to extract from the device, and locate places where external signals can be injected, designing interface circuity to do this in a way that brings the voltage levels of the device and my system into compatibility. This results in a module panel that’s either mounted on the external device itself, or bolted into one of my synthesizer cabinets and connected to the external device through an umbilical (these connecting cables support 28 patch points I’ve extracted from the MiniMoog, 19 from the Satellite, and 21 from the MG-1, for example). When the umbilical isn’t attached, or the panel on the external device isn’t patched, the external synthesizer is unaffected, and works like it always did before. When the umbilical is connected and/or patches are introduced, the device

Figure 17: A Synth Patch from 2013 that incorporated the assimilated Muse

becomes an intricate part of my modular, giving me access to many parts of the external system.

For example, with the assimilated analog synthesizers, I can control the frequency and pulse width of each oscillator, access each oscillator waveform, control the frequency and Q of the filters, inject external audio, access the envelope generators and VCA, tap into the keyboard and mod/pitch wheels, etc. For the digital synthesizers, I provide internal signals that produce interesting, harmonically-rich voices (or separate voices that were mixed into common outputs in the original synthesizer), and throw CMOS switches across the most important buttons, allowing them to be controlled from the modular (e.g., step through notes stored in sequencer memory or change timbre – the SK-1 mod outlined in [70] is based on my techniques). For the VL-tone, I replaced the toy keyboard with a full-sized manual and can also play particular notes electronically by applying bits to a priority encoder. For the CS-101, I can sum the bits produced by its wave engine (which uses an external DAC) with arbitrary weights, which turns the most mellow voices into monster-fuzzed hard Canterbury/prog leads, or turns a smooth decay into a fascinating trill as low order bits that are now summed with more weight toggle on and off. I can also voltage-control the CS-101’s master pitch and extract gates and triggers from the keyboard – more fun hacks on this synthesizer are suggested in [71].

With the Triadex Muse (the world’s first autonomous sequencer product) [72], I can, for example, voltage-control the pitch and clock rate, inject an external clock, have access to all bits coming from the sequencer into the note-generating priority encoder, and inject external bits into the sequencer. As the Muse was probably inspired by Marvin’s hobby of improvising fugues at the piano [73], the Muse is all about melody and does nothing with timbre, as it just produces a square wave. When married to my modular, however, the Muse becomes a timbre and melody monster, producing thick waves of dynamically changing sound in accordance with state determined by the current patch.

I have also identified and located the row/column scanning signals that run the user interface for the (pre-MIDI) digital polyphonic keyboards that are integrated with my modular (the SK1 [74] and CS101 [71]), and provided them at a connector added to the rear of each device. I plan to build a module based around a simple panel-mounted breadboard and set of analog switches for each of these synthesizers – the breadboard provides the row/column signals that can be bridged via the switches – gating a switch on via a logic signal from the modular will either play a note or hit a control button on the panel, depending which row and column are bridged by the

Figure 18: The Patchwork Module

switch. The user can select the row and column by plugging wires into the breadboard (hence determine the note to be triggered or button to be pushed), and chords can be struck by connecting the switched column to the appropriate rows through diodes, again all on the breadboard to allow maximum flexibility with a very simple module.

5.6 Patchwork and control over the web

My students proposed and built a standalone module to connect my synthesizer to the web, allowing users to remotely stream the patch's audio and control it live while it was fully installed and running at the MIT Museum during January-April of 2012 [75]. Termed 'Patchwork' [76] (or sometimes 'Patchwerk'), this device was based around an embedded Intel Atom development board running Lynux and communicating with remote clients via the HTML5 Web-Socket standard. Patchwork supports 4 bipolar analog outputs and 8 logic outputs. The logic outputs can be configured as momentary 'pushbuttons' or latching toggle switches (it's typically set up to support 4 of each type). Patchwork has a virtual online front panel that mirrors the state of the physical front panel. Both physical and virtual panel controls can be adjusted by the users. The physical panel has 4 motor-driven knobs that allow a local user to adjust the corresponding analog output – the mirrored 'online' knob turns in correspondence, and when the online knob is adjusted, the physical knob turns accordingly. Ditto for the logic signals – their state on the physical panel is indicated by LEDs and via the animated switch or pushbutton state on the virtual panel. Up to a dozen online users can control a patch simultaneously. When they grab a knob or push a button, the displayed image of that item changes accordingly, and other users are locked out from touching it until it is released. If there are more users who want to interact, they are held in a queue until an interaction slot is available. Although there could be a delay of a few seconds between when a virtual controller is hit and when the result is heard (mainly due to audio stream latency), this didn't seem to present a significant deterrent to the online audience. Users are allowed to interact for a preset amount of time (we've varied it from 10 minutes to a half hour) and are knocked off afterwards to allow another user to have a turn. We also implemented a maximum limit on the amount of time a user can hold onto a particular controller to inhibit 'Bogarting' behavior. Active users have their chosen username visible in the online panel, and a scrolling Twitter window is also shown to display current online chat about the system (users have also been seen to pick their username in order to message one another).

After this system was covered in the popular online press (e.g., [77]), the user queues exploded whenever a new patch was developed, lasting for days as users all over the world

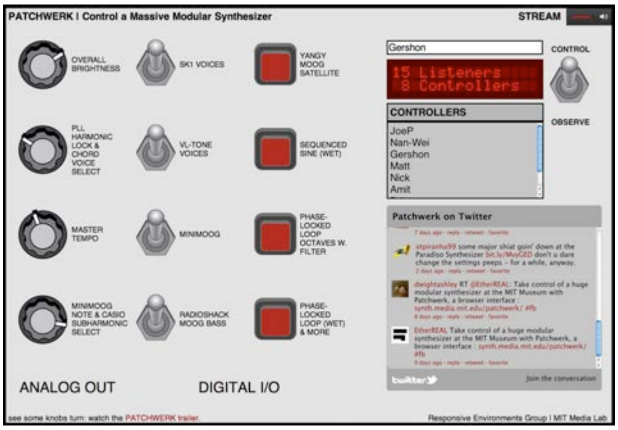

Figure 19: The virtual Patchwork window as seen by online users in their web browser

collaboratively controlled the synth patch. We encountered, for example, over 40,000 users who tried the system the first week it was introduced, with holding queues extending to about 250 people.

I found that composing a patch for the Patchwork system was very different than putting together a standalone patch. In order to avoid sonic boredom, standalone patches need to explore their sonic subspaces fully, hence I have to build in a lot of variation. In a Patchwork setup, this variation comes from user activity. If users stop interacting with Patchwork, it stays in a simpler sonic space – perhaps still playing sequences and producing varying sound, but the changes are much simpler until users again hit the controls to jerk or nudge it into different sonic territory. I found that users could both collaborate and compete here – some would find a sonic space where the audio would meet their aesthetic, but when another user turned a knob elsewhere, the original user would try to nudge it back.

As the dimensionality of control offered by Patchwork is so limited, especially for a massive patch, Patchwork's outputs were fanned out and expanded using a number of tricks. The simplest would be to have each knob or gate control several modules or parameters at once. Another strategy was to input the momentary pushbutton gates into binary counters where they would be expanded into many bits. Pushing the buttons advanced the counters, and the counter's bits could mode parts of the patch quite differently. Hence the buttons usually didn't do the same thing each time you pushed them. Another trick was to put one of Patchwork's analog outputs into the ADC or window comparator array to produce a series of bits toggling with voltage that could switch analog multiplexers and otherwise re-mode digital processes that were running the patch as the knobs were twisted. Accordingly, it could sometimes be hard to deliberately restore a patch to its prior setting with the Patchwork controls, although with continued user interaction, it broadly would cycle back to revisit sonic spaces.

I built two types of patches using the Patchwork. The first were pure atonal sound-art, where users could navigate a very rich space of electronic sound. The other patches were more conventionally musical, living in a defined tonal space, with coherent and compatible melody fragments, harmonies, etc. Users heavily engaged with both environments, although I suspect they represented correspondingly different sets of taste.

6. Conclusions

In this article, I've tried to trace why modular synthesizers are still with us, based around my own 45-year personal experience in working with and building them. The immersive physicality of modular rigs is hard to replace with generalizable software environments. For this reason, I suspect that modulars will stay with us for a while, eventually ceding to virtual incarnations when the haptic fidelity and graphic realizations of VR more closely approach that of physical interaction.

Modular synths are many things, but first and foremost, they are a passion, and one that demands attention, energy, space, and resources. Too many good things in life can get and have gotten in the way of synth hacking for many of us. Yet, my modular still calls to me like a siren – there's just a little more fun engineering left to realize a few new and lingering technical ideas, but a lot more music to explore and make with it. Every time I engage with my modular, I discover something new that it can do. This sonic 'toolkit' that I built so many decades ago still has tremendous life left. And yes, it's fully compatible with current Eurorack units (needing only a set of 1/8" to pin jack converters). Stay tuned – there's lots more to come! Modulars will still be around for a while, and their music will play on.

More information and sound examples from my system are linked from http://synth.media.mit.edu .

7. Acknowledgements

I am indebted to my dad for encouraging my love of music and technology as a child, introducing me to audio and soldering before I could even ride a bike. The technicians at Draper Lab, where I worked as an undergrad, are earnestly thanked for giving me electronics components and letting me use their PCB fab to make my first boards. And without Solid State Sales (later Eli Heffron and Sons), our local surplus electronics store in Cambridge back then, I couldn't have found the components that I needed to build my system. Finally, I thank the late Ivan Tcherepnin for spending that hour showing the Serge Modular to an aspiring 18-year-old synth enthusiast – this was an hour that resonated deeply through the years and decades that followed. In more recent times, I also acknowledge the members and alums of my current research team at the MIT Media Lab, especially Gershon Dublon, Brian Mayton, Nick Joliat, Mark Feldmeier, Spencer Russell, Don Derek Haddad, Bo Morgan, Matt Aldrich, and Nan-Wei Gong for working with me on Patchwork and many other music projects – even more importantly though, these guys got me playing keyboard again via our group jam sessions and reminded me how much fun making music can be.

8. REFERENCES

[1] Thompson, A., *Stompbox: A History of Guitar Fuzzes, Flangers, Phasers, Echoes & Wahs*, Miller Freeman Books, 1997.

[2] Hughes, T., *Analog Man's Guide To Vintage Effects*, For Musicians Only Publishing, 2004.

[3] Stokel-Walker, C., "The 19th Century plug that's still being used," BBC News Magazine, January 11, 2016. http://www.bbc.com/news/magazine-35253398

[4] Schmidt Horning, S., *Chasing Sound: Technology, Culture, and the Art of Studio Recording from Edison to the LP*, Johns Hopkins University Press, 2015.

[5] Ulmann, B., *Analog Computing*, Oldenbourg Wissenschaftsverlag, 2013.

[6] Rhea, T., "The History of Electronic Musical Instruments," in Darter, T. and Armbruster, G., *The Art of Electronic Music*, GPI Books, pp. 4-63.

[7] Jenkins, W., "The Choralcelo (Celestial Choir) History: 1888-1942," in The AMICA Bulletin, Vol. 45, No. 4, August/September 2008, pp. 201-233.

[8] Pinch, T. and Trocco, F., *Analog Days: The Invention and Impact of the Moog Synthesizer*, Harvard University Press, 2002.

[9] Moog, R.A., "Voltage-Controlled Electronic Music Modules," in the Journal of the Audio Engineering Society Volume 13, Number 3, July 1965, pp. 200-206.

[10] Aiken, J., "Don Buchla," in Darter, T. and Armbruster, G., *The Art of Electronic Music*, GPI Books, pp.75-83.

[11] Klein B., *Electronic Music Circuits,* Howard W. Sams & Co., Indianapolis IN, 1982.

[12] Puckette, M., "MAX at Seventeen," Computer Music Journal, 26:4, Winter 2002, pp. 31-34.

[13] Mathews, M. *The Technology of Computer Music*, MIT Press, 1969.

[14] Roads, C., "Interview With Max Mathews," Computer Music Journal, Vol. 4, No. 4, Winter 1980, pp. 15-22.

[15] Fitzmaurice, G.W., Ishii, H., and Buxton, W.A.S., "Bricks: laying the foundations for graspable user interfaces," in Proc. of CHI '95, ACM Press, 1995, pp. 442-449.

[16] Ishii, H. and Ullmer, B., "Tangible Bits: Towards seamless interfaces between people, bits and atoms," in CHI'97 Proceedings, pages 234–241, 1997.

[17] Weiser, M., "The Computer for the 21st Century," Vol. 265, No. 3, September 1991, pp. 66-75.

[18] Fishkin, K.P., "A taxonomy for and analysis of tangible interfaces," in the Springer Journal of Personal and Ubiquitous Computing, Volume 8 Issue 5, September 2004, pp. 347-358.

[19] Newcomb, M.B., *The Museum of Synthesizer Technology*, self-published, 1994.

[20] Vail, M., "The World's Most Dangerous Synth," Keyboard Magazine, June 1992, pp. 51, 70-71,75,79,130.

[21] Matirano, S., *Progress Report #1: An Electronic Music Instrument Which Combines The Composing Process With Performance in Real Time*, UIUC Internal Report (circa 1972).

[22] Frith, S. *Performing Rites: On the Value of Popular Music*, Harvard University Press, 1998.

[23] Live at the Necropolis: Lords of Synth | Adult Swim. https://www.youtube.com/watch?v=WXgNo5Smino

[24] Big Train - Keith Emerson. http://www.youtube.com/watch?feature=player_embedded&v=JwsflF5Fyho

[25] Taylor, C. (director), *Fuzz, The Sound That Revolutionized The World*, Brinkfilm, 2007.

[26] Fantinatto, R. (director), *I Dream of Wires*, 2014.

[27] Von Hippel, E., *The Sources of Innovation*, Oxford University Press, 1988.

[28] Buxton, W., Fogels, A., Fedorkow, G., Sasaki, L., & Smith, K. C.. "An Introduction to the SSSP Digital Synthesizer," Computer Music Journal (4), 28-38, 1978.

[29] http://web.media.mit.edu/~joep/CAMAC-Synth/CamacSynth.html

[30] Csikszentmihalyi, C. (1996). *Creativity: Flow and the Psychology of Discovery and Invention,* HarperCollins Publishers.

[31] http://www.nordkeyboards.com/products/nord-modular-g2

[32] Moro, G, Bin, A, Jack, RH, Heinrichs, C, and McPherson, AP, "Making High-Performance Embedded Instruments with Bela and Pure Data", in the Proc. Of the International Conference on Live Interfaces, June 29th - July 3rd, 2016.

[33] https://www.critterandguitari.com/products/organelle

[34] https://hoxtonowl.com/

[35] http://www.axoloti.com/

[36] https://www.pjrc.com/teensy/gui/

[37] Jacobs, R., Feldmeier, and M., Paradiso, J.A., "A Mobile Music Environment Using a PD Compiler and Wireless Sensors," in Proceedings of NIME 2008, Genoa, Italy, September 10, June 5-7, 2008, pp. 193-196.
[38] Newton-Dunn, H., Nagano, H., and Gibson, J., "Block Jam: A Tangible Interface for Interactive Music," in Journal of New Music Research. 32(4):170-177, December 2003.
[39] Harrison, J., "SoundBlocks and SoundScratch : tangible and virtual digital sound programming and manipulation for children," MS Thesis, MIT Media Lab, 2005.
[40] McPherson, A., "Interconnectable blocks for music and audio processing," M.Eng Thesis, MIT EECS Dept and MIT Media Lab, 2005.
[41] Little Bits Synth Kit, https://littlebits.cc/kits/synth-kit
[42] http://www.patchblocks.com/
[43] Qi, J., Demir, A., and Paradiso, J.A., "Code Collage: Tangible Programming on Paper with Circuit Stickers," to appear in the Proc. of CHI2017.
[44] Alber, N., Alonso, M., Geiger, G. and Jordà, S., "The Reactable: A Collaborative Musical Instrument for Playing and Understanding Music," Heritage & Museography (Her&Mus) 4 [Vol. ii, No. 2], 2010, pp. 36-43.
[45] http://fractgame.com/
[46] http://www.soundstagevr.com/
[47] Benavides Palos, X., "A platform for reaching into the environment of a remote collaborator," MS Thesis, MIT Media Lab, June 2016.
[48] Rule, G., "Keyboard of the Month," Keyboard Magazine, Vol. 22, No. 12, December 1996, p. 13. (see also Rule, G., "25 Years of Weird Gear" Keyboard Magazine, Vol. 26, No. 1, January 2000, pp. 60-64).
[49] Paradiso, J., "Diary of a Teenage Synth Hacker," Forward to Kettlewell, B., *Electronic Music Pioneers*, ProMusic Press, Vallejo, CA, 2002, pp. 4-9.
[50] Paradiso, J., "Modular Synthesizer," in Stocker, G., and Schopf, C. (eds), *Timeshift* (Proc. of Ars Electronica 2004), Hatje Cantz Verlag, Ostfildern-Ruit, Germany, 2004, pp. 364-370.
[51] http://synth.media.mit.edu
[52] http://www.media.mit.edu/~joep/synth.html
[53] Paradiso, J., *The Design, Construction, and Operation of an Electronic Music Synthesizer*, Tufts University Undergraduate Special Project Report, May 1977. Available at: http://web.media.mit.edu/~joep/papers/EarlySynthesizerReport.pdf
[54] Lancaster, D., "IC's for Electronic Music," Radio-Electronics, February 1974, page 49-52.
[55] Simonton, J.S. Jr., "Build a Modular Electronic Music Synthesizer," Radio-Electronics, May 1973, pp. 38-47.
[56] Electronotes currently lives at: http://electronotes.netfirms.com/
[57] Jung, W., *IC OpAmp Cookbook*, Sams Publishing, Indianapolis IN, 1974.
[58] Colin, D.P., "Electrical design and Musical Applications of an Unconditionally Stable Combination Voltage controlled Filter/Resonator," in the Journal of the Audio Engineering Society, Vol. 19, No. 11, December, 1971, pp. 923-927.
[59] Simonton, J.S. Jr., "More Synthesizer Modules," Radio-Electronics, September 1973, pp. 53-63, 102.
[60] Jung, W., *IC Timer Cookbook*, Sams Publishing, Indianapolis IN, 1977.
[61] Simonton, J.S. Jr., "Electronic Wind Chimes - Natural sounds achieved artificially," Popular Electronics, Vol. 3, No. 4, April 1973, pp. 51-55.
[62] Burhans, R., "Pseudo-noise Timbre Generators," in the Journal of the Audio Engineering Society, Vol. 20, No. 3, April 1972, pp. 174-184.
[63] http://www.media.mit.edu/~joep/MusicalStateMachine
[64] http://casperelectronics.com/finished-pieces/nova-drone/
[65] http://www.sound-machines.it/product/ns1nanosynth/
[66] Briggs, K., "Simple experiments in chaotic dynamics," American Journal of Physics, 55(12), December 1987, pp. 1083-1089.
[67] Ciarcia, S., "Build an Unlimited-Vocabulary Speech Synthesizer," Byte Magazine, Sept. 1981, pp. 38-50.
[68] Paradiso, J.A., "The WMFO Pitch Shifter," in *The Complete Technical Guide to WMFO,* pp. 86-109, 1987, available at: http://web.media.mit.edu/~joep/papers/WMFO-tech-guide.pdf
[69] Ghazala, R., *Circuit-Bending: Build Your Own Alien Instruments*, Wiley, 2005.
[70] Fluharty, T., "Sync and Sequence with the Casio SK-1 Sampler," Electronic Musician, Vol. 3, No. 4, April 1987, pp. 62-68.
[71] Styles, M., "Modifying The Casio 202," Polyphony, Vol. 9, No. 3, April 1984, pp. 16-18.
[72] https://en.wikipedia.org/wiki/Triadex_Muse
[73] Minsky, M., "Music, Mind, and Meaning," Computer Music Journal, Vol. 5, Number 3, Fall 1981, pp. 28-44.
[74] Messick, P. and Battle, J., "Build a MIDI Input for Your Casio SK-1," Keyboard Magazine, Vol. 13, No. 8, August 1987, pp. 34-40.
[75] Lindsey, L., "Video: The Paradiso Synthesizer," MIT News Office, March 14, 2012. http://news.mit.edu/2012/video-paradiso-synthesizer-0314
[76] Mayton, B., Dublon, G., Joliat, N. and Paradiso, J.A., "Patchwork: Multi-User Network Control of a Massive Modular Synthesizer," in Proceedings of NIME 2012, Ann Arbor, Michigan, May 2012, pp. 453-456.
[77] Dayal, G., "Control a Giant Modular Synthesizer From the Comfort of Your Home," Wired Online, March 15, 2012. https://www.wired.com/2012/03/giant-modular-synthesizer/