Multi-mapped Neural Networks for Control of High Dimensional Synthesis Systems

Sam Pluta

Peabody Institute of The Johns Hopkins University
spluta@gmail.com

Abstract. This paper outlines NN Synths [1], a software instrument that uses multi-mapped regression-based deep learning neural networks[2] to control multiple high dimensional synthesizers. The paper discusses the reasoning behind the use of high-dimensional synthesizer algorithms and then presents the designs of two individual software synthesizers at use in the NN Synths instrument. It then outlines the larger ecosystem of the multi-mapped performance space: showing why the archipelagic nature of these synthesizers requires the user to have rapid access to multiple different mappings for expressive performance, and how easy switching between multi-mapped synths facilitates expressive traversal of the larger multi-dimensional performance space.

Keywords: Sound Synthesis, Neural Networks, Supervised Learning, Archipelagic Thinking

1 Introduction: Neural Pathways through Analog Dreams

In Rebecca Fiebrink and Laetitia Sonami's 2020 paper, *Reflections on Eight Years of Instrument Creation with Machine Learning* (Fiebrink & Sonami, 2020), Sonami posits a fascinating challenge: "ML can now allow many input controls to be targeted to many synthesis parameters...But what synthesis currently allows for such dynamic control?" Having spent the past 15 years working with analog synthesizers, as well as creating the Live Modular Instrument, a software instrument whose design was highly influenced by the modularity of the Buchla (Pluta, 2009-2021), I believe that, due to the following design features, modular analog synthesis is a perfect model to provide *an* answer (there must be many!) to Sonami's challenge:

1) Modular analog synthesis provides a framework where individual modules have simple functions and any number of these modules can be plugged together to create complex systems.
2) A modular framework allows the user to construct emergent systems that, through cybernetic interaction[3], can quickly accumulate complexity, to the point where the user

[1] `https://github.com/spluta/LiveModularInstrument/tree/master/modules/NN_Synths`

[2] The software uses the FluidMLPRegressor (a regression-based multi-layered perceptron) from the Fluid Corpus Manipulation Project (Green, Roma, & Tremblay, 2021)

[3] The Buchla synthesizer is the perfect model for this type of interaction. In "Bay Area Experimentalism: Music and Technology in the Long 1960s", Ted Gordon describes the cybernetic nature of the Buchla in the following way: "control could flow in multiple directions, not only from human to module, but from module to module, and indeed back to the human"(Gordon, 2018). This interaction pushes the musician beyond a state of knowing and into an "emergent and unpredictable" flow state.

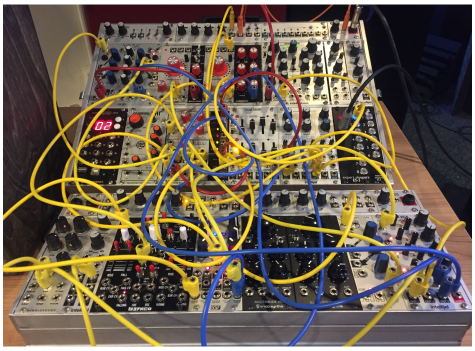

Fig. 1. Typical Analog Synth Complexity

can comprehend the system, but is unable to predict its exact outcome (I know turning this knob to the left will be interesting, even if I don't fully know what will happen).
3) The system creates a naturally intuitive way, through knobs and control voltage, to access manipulation points at any node in the setup, allowing the user to access internal functionality as easily as inputs and outputs.
4) Audio feedback is as intuitive as plugging one cable from a latter node in the audio/control circuit to a previous node, thus facilitating increased sonic complexity with little added routing (feedback in digital systems is notoriously unintuitive).

The goal in designing each of my NN Synths is to mine the features of the analog modular for my digital instrument design, to create complex synthesis systems: 1) by connecting large numbers of Unit Generators, 2) whose complexity emerges through cybernetic interaction, 3) and where the result is an instrument on which the user has easy access to and can freely manipulate parameters throughout the synthesis space.

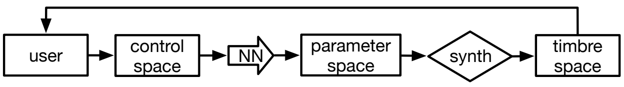

Fig. 2. The layout of the control/parameter/timbre space.

While Neural Networks (NNs) are equally adept at controlling parameter spaces of low to high dimensionality and systems ranging from linear to chaotic; for my own aesthetic purposes, highly dimensional, non-linear and chaotic systems are where I have found NNs to be most expressive. I have designed different NN Synths with parameter spaces ranging from 12 dimensions to 50, many of which use elements of feedback to invoke chaotic behavior in the timbre space[4]. The chaotic, non-linear nature of these instrument designs, combined with a large number of parameters (in these cases 16 or 50), make

[4] Though they can be correlated, the parameter space is not to be confused with the timbre space. While the parameter space is the one we can quantify and the one I will quantify below, the end-goal is to create synths where the user is able to traverse, with low-dimensional controllers, a multidimensional (and in these cases nonlinear) timbre space(Lerdahl, 1987): the one that the performer feels and the one that we hear.

them a challenge to map in a traditional manner. Yet, "artificial neural networks possess several properties that make them particularly attractive for applications to modelling and control of complex non-linear systems" (Suykens, Vandewalle, & de Moor, 1995), and they seem to be the perfect control structure for these chaotic synthesis systems.

2 Synthesis Models

2.1 CrossFeedback0 Synthesis Model

Cross-Feedback Synthesis, a technique where two oscillators frequency modulate each other, creates a chaotic output from two linear systems, thus providing a fertile playground for neural net-based control of audio. My CrossFeedback0 module[5], a circuit that one would easily design using an analog synthesizer, represents one such system. The sound source for the circuit is a single triangle wave. This triangle wave is modulated by a sine wave, which is modulated by 1) itself (internal feedback), 2) a resonant lowpass filter, and 3) the triangle wave that *it* modulates. This output is sent to an inline set of user controllable filters: a distortion, a Dust-based amplitude modulator, and a resonant LPF, whose Q is user controlled and whose frequency is controlled by the user and a frequency modulating triangle wave. All in all, there are 16 independent parameter inputs spread across every node of the the synthesis matrix, which without neural networks would be far too many for one user to control at the same time. Due to the cross-feedback design, even the slightest changes in the parameter space can result in large changes in the timbre space, while other regions of the timbre space are quite linear.

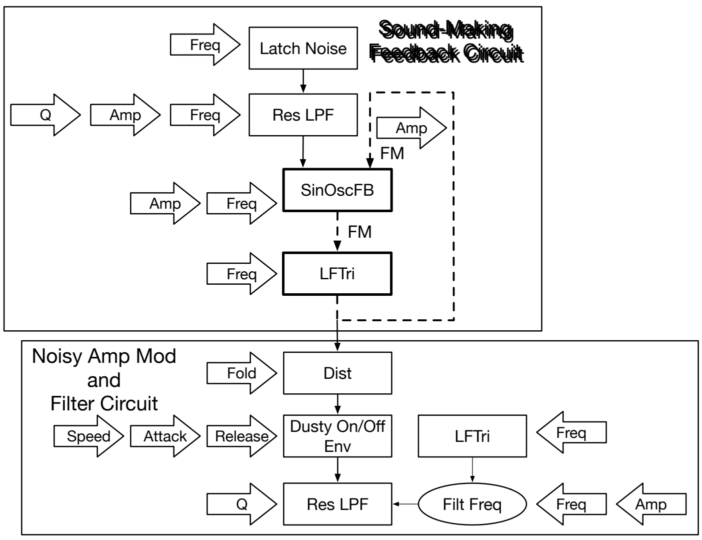

Fig. 3. Structure of CrossFeedback0 SynthDef. Boxed items show the Unit Generators used, and the arrows show the user-controllable inputs to the system.

5 https://github.com/spluta/LiveModularInstrument/blob/master/modules/NN_Synths/01_CrossFeedback0/CrossFeedback0.sc

2.2 FM7 Synthesis Model

While the CrossFeedback0 synth achieves a high level of expressiveness with only 16 parameters, I have also attempted systems with much higher dimensionality. One such setup uses Stefan Kersten's FM7 UGen (Kersten, 2008) from the SCPlugins package and uses a design based on Fredrik Olofsson's work with this UGen (Olofsson, 2008). The FM7 is a model of the popular 1980's digital synthesizer, the Yamaha DX7, a user rout-able circuit of six intertwined oscillators. For each of these oscillators, the user has direct control over frequency, phase, and amplitude, and each oscillator can also be phase modulated by the five other oscillators in the circuit in addition to itself. The user has control over the modulation level of each oscillator through a six by six matrix of adjustable values.

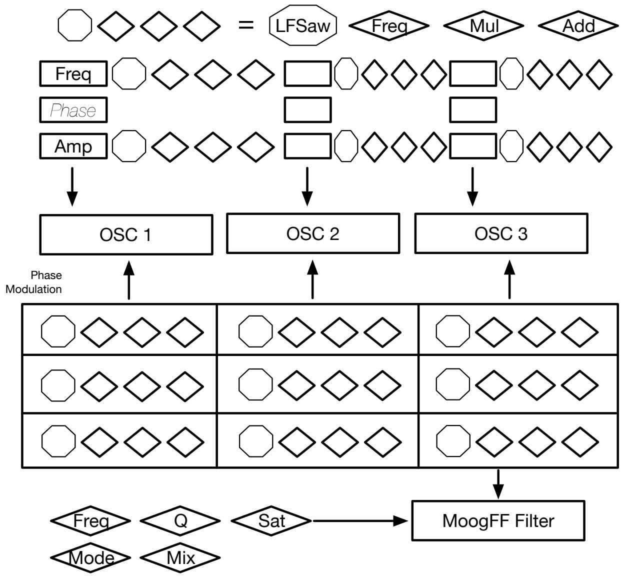

Fig. 4. Structure of my NN Synth using the FM7. Boxed and octagonal items show the Unit Generators used, and diamonds show the user-controllable inputs to the system.

To keep things from being too chaotic, my circuit uses just 3 of the 6 oscillators of the FM7[6], and we only hear the output of the first two oscillators (hard panned left and right). Riffing on Olofsson's design, my synth uses 15 Sawtooth waves to control two different sets of parameters of the FM7: 1) a Sawtooth Wave is used as input to the frequency and amplitude of the sounding (sin wave) oscillators and 2) a Sawtooth Wave is used to control the modulation arguments in the phase modulation index matrix. The user has control over the frequency, multiply, and add arguments of each sawtooth wave. In addition to controlling the three arguments of each sawtooth wave, the user also has direct control over the frequency, Q, saturation, mode, and mix arguments of a

[6] https://github.com/spluta/LiveModularInstrument/blob/master/modules/NN_Synths/04_FM7Saw/FM7Saw.sc

MoogFF filter. All told, there are 50 adjustable parameters in the circuit, and like with CrossFeedback0, the timbre space has the potential to change drastically with slightest change of the parameter space. This circuit is well beyond the complexity that a single human can control (or even comprehend) directly, but the perfect grounds for using a neural network to intuitively control the complex system.

3 The Joy of Multidimensionality: A Modular Approach

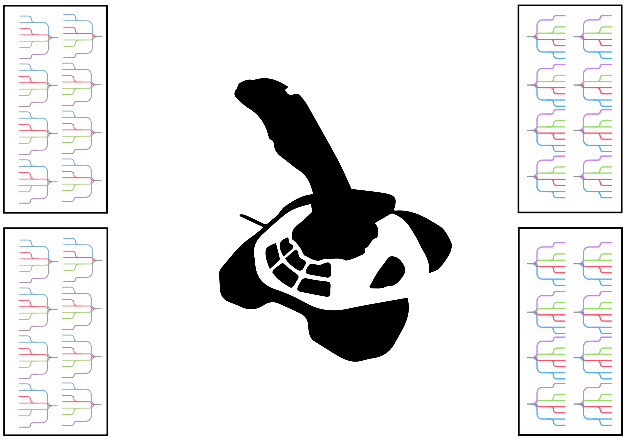

Fig. 5. 8 times 4 NN mappings per controller.

In *Music at Hand* (De Souza, 2017), Jonathan De Souza lays out a theory of phenomenology which emphasizes the lived experience of the performer/composer/ improviser when physically interacting with their instrument, and theorizes how the physical layout of particular instruments influence the music that is made with them. Neural network-controlled instruments provide a particular challenge in this arena, since the inputs to the NN can be just about anything that creates values between 0 and 1. Thus, my approach for the NN Synths instrument is to use the neural network to expand my curated control space into the desired parameter space, leaving a control space with an intuitive set of controls, while still allowing maximum expressivity in the timbre space.

I use two different gestural controllers in my NN setup: 1) a Logitech Extreme 3D Pro flight simulator joystick which provides a 3D control space via X, Y, and yaw and 2) a pair of 2D Sliders on my iPad Lemur interface, which result in a 4D control space[7]. I could use a simple 2D slider or a set of 1D sliders as my control space. However, for the particular sounds and type of control that I am pursuing, I have found the following criteria to be important aspects of audio control: 1) like most musical instruments, I want the control system to fit expressively under the hand and 2) it is better for me to use systems of control which are intuitive, but simultaneously at the limit of or slightly

[7] My system allows up to a 10 item vector as input and with this I have experimented with other controllers, like the Leap Motion and Roli Blocks.

beyond my spacial comprehension. The joystick and the multi-2D layouts both fulfill these criteria.

High dimensional synths like CrossFeedback0 and FM7 have an enormous number of possible settings for the musician to contend with. For example, if we were to limit each dimension of CrossFeedback0's 16 parameters to just 100 values, the space would have 10^{32} different settings, or $one - trillion^3$ (for FM7, there are a googol!). Within that space, much of the material is dark matter and is essentially uninteresting. The key is to find "pockets of magic" within the system: the places where all of the parameters line up, and where subtle changes between parameters result in compelling sounds, then to map a set of input vectors to a set of output vectors (just like Wekinator (Fiebrink, 2016)), and train the regression neural network.

Because the parameter space of each synth in my system is very large, and I have found the pockets of material to be generally small, any single mapping of input vectors to output vectors will traverse a minute number of points within a synth's parameter space. Furthermore, each parameter space contains infinite possible magic pockets. Thus, my approach does not try to make a "single perfect" mapping of my controller space to my parameter space. Instead, it allows eight unique mappings per synth.

In addition, each controller manipulates four different synths. Thus, there are 32 mappings available to each controller in the system (4 synths with 8 mappings each). Since I currently use two controllers, this results in 64 simultaneously available mappings in 8 different synths (and I could easily add more with more controllers). The software uses a switching system within each synth so that one NN is available per synth at any one time. A second switching system over a controller's set of 4 synths facilitates monophonic or polyphonic control, forcing only one to synth play at a time, or allowing multiple. Like in the rest of my Live Modular Instrument, the NN Synths instrument allows the users to easily and rapidly access multiple maximally unique timbre spaces, and expressivity comes from its ability to present the performer with multiple paths into and out of musical situations.

4 Affordances of the Multi-mapped Model

> *"The archipelago exists at the intersection of geomaterial reality and imaginative metaphoricity"* (Baker, 2020).

The model proposed above acknowledges the *archipelagic* (Glissant, 1997) nature of vast multidimensional spaces, especially those designed around chaotic generators. These spaces, far too large and complicated to fully consciously comprehend, contain many islands of unique materials in vast oceans of emptiness. The islands chosen to be included in a set of training vectors "are not assembled through equivalence, but through difference and proximity" (Vickers, Allado-McDowell, & GPT-3, 2020). A single set of training vectors (8 in this case) mapped to a synthesis system represents a world of possible interactions and relationships. Viewed in relief of the multiverse of alternate mappings for the system, this multi-mapped approach gives us at least one curated set of perspectives on the multi-dimensional space, through which we can musically travel.

Neural network-mapped synthesis systems intertwine the already cybernetic natures of sound synthesis and supervised learning. Mayer states that "it's typical that algorithms

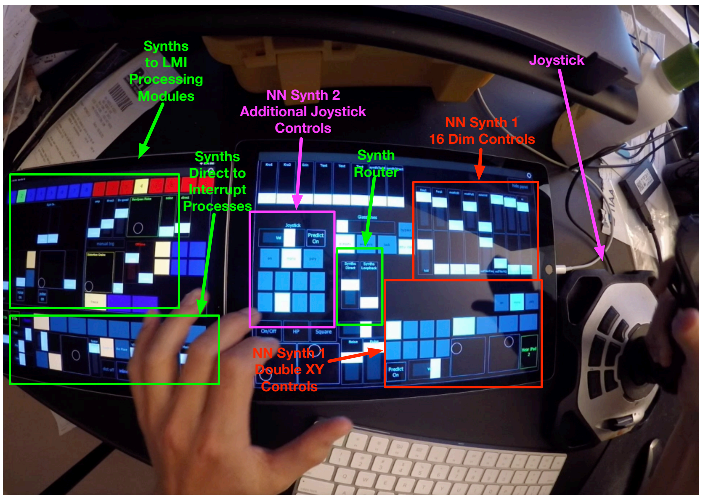

Fig. 6. Live Modular Instrument and NN Synths control layout.

for synthesis come into being in feedback loops, where aesthetic judgments are influencing further refinements" (Mayer, 2020). This is especially true the more complex the system becomes. The instrument designer creates an initial patch, listens, makes a change or addition to the patch, listens again, and repeats the process until the final result is satisfactory. Supervised training of deep learning neural networks sets up another cybernetic relationship. According to Fiebrink, "when the process of exploration and engagement is physical, rather than abstracted into mathematical functions and programming code, composers are able to engage in tight, *enactive* (Wessel, 2006) action-feedback loops which further inform their embodied understanding of the instrument and their own musical aims" (Fiebrink, 2017). If training is a fluid process, as it is in Fiebrink's Wekinator or in my NN Synths instrument, the designer/performer can quickly develop a mapping (by listening to the system and creating mapping points) and then immediately perform with the system to see if the mapping is acceptable. If it is, keep it. If it is not, either add another point or two, take point or two away, or simply throw the training away and do it again.

Independently, synthesis design and model training are both cybernetic systems. However, if the instrument design interface allows for synthesis design and model training to happen concurrently, we can amplify Human-Instrument feedback loops, even further strengthening cybernetic feedback in the design system.

Zooming outward, multi-mapping adds a further cybernetic layer. In my system, trainings do not exist in isolation, but next to each other, in groups of eight. The set of eight trainings constitute a quarter of the archipelagic system. After performing with and traversing the eight vectors of a training set, the performer/designer can asses the value of each vector within the system. A training that might have been expressive on its own, may be too similar to another in the set. Or perhaps one training in a set sits awkwardly in the hand when compared to the others. There are many reasons why a

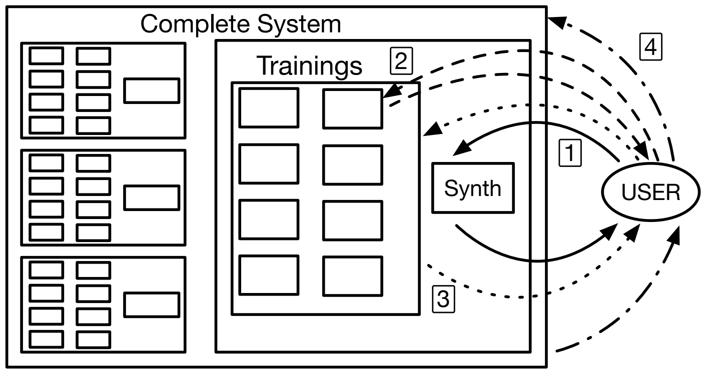

Fig. 7. Cybernetic feedback loops of the NN Synths system: 1) synth design, 2) training, 3) mult-mapped set, 4) 4 synth system.

training may not feel correct, but in this system it is easy enough to replace any training with a new one in less than one minute.

The final layer in this Russian doll of cybernetics is the four layered synthesis design. As I say in my dissertation, *Laptop Improvisation in a Multi-Dimensional Space*, "by creating a setup where the performer is able to easily switch between combinations of sound modules, we create a system where multiple ways of interacting with sound are available to the performer at all times" (Pluta, 2012). The four-layered system of NN Synths, the complete archipelago, provides an ecosystem with multiple options, thus facilitating expression. But the ecosystem must be worth more than the sum of its parts. If one synth does not complement the others, it does not add to the expressive power of the collection. By letting the user easily and instantly swap out synthesis modules and training sets, cybernetic interaction extends to all aspect of instrument design.

5 Conclusion and Future Work

We seem to be standing on the shores of a creative world of synthesis opened up by machine learning. Using analog synthesis as inspiration is but one approach, and now that this pandora's box is opened, the feedback loop between synthesis and mapping will likely accelerate development in many directions. Chaotic differential equations (Sanfilippo, 2021), with multiple arguments controlling their shape, seem to be fertile ground for further exploration, and I have already designed synths using this model[8]. Physical models and wave terrain synthesis are other highly dimensional techniques, as is the more recent technique of direct neural synthesis(Engel et al., 2017). How else might we construct layers of neural networks to traverse an archipelagic parameter-space? How will new infrastructures of control spaces spawn new synthesis models? How will new synthesis models further influence the shapes of our control systems?

[8] https://github.com/spluta/LiveModularInstrument/blob/master/modules/NN_Synths/08_LotkaA/LotkaA.sc

References

Baker, J. S. (2020). Small islands, large radio: Archipelagic listening in the caribbean. In Y. Miguel & M. Stephens (Eds.), *Contemporary archipelagic thinking: Towards new comparative methodologies and disciplinary formations.* Rowman & Littlefield Publishers.

De Souza, J. (2017). *Music at hand: Instruments, bodies, and cognition.* Oxford, UK: Oxford University Press.

Engel, J. H., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., & Norouzi, M. (2017). Neural audio synthesis of musical notes with wavenet autoencoders. *CoRR*, *abs/1704.01279*. Retrieved from `http://arxiv.org/abs/1704.01279`

Fiebrink, R. (2016). *Quick walkthrough of wekinator - youtube.* Retrieved 2021-03-26, from `https://www.youtube.com/watch?v=dPV-gCqy9j4&t=2s&ab_channel=Wekinator`

Fiebrink, R. (2017, 12). Machine learning as meta-instrument: Human-machine partnerships shaping expressive instrumental creation. In E. T. Bovermann et al. (Ed.), *Musical instruments in the 21st century* (p. 137-151).

Fiebrink, R., & Sonami, L. (2020, July). Reflections on eight years of instrument creation with machine learning. In R. Michon & F. Schroeder (Eds.), *Proceedings of the international conference on new interfaces for musical expression* (pp. 237–242). Birmingham, UK: Birmingham City University.

Glissant, E. (1997). *Traité du tout-monde.* Paris: Gallimard.

Gordon, T. (2018). *Bay area experimentalism: Music and technology in the long 1960s.* ProQuest Dissertations & Theses.

Green, O., Roma, G., & Tremblay, P. (2021). *Fluid corpus manipulation.* Retrieved 2021-04-02, from `https://www.flucoma.org/`

Kersten, S. (2008). *Fm7 app on github.* Retrieved 2021-03-25, from `https://github.com/supercollider/sc3-plugins/blob/dd092a20cb66fc976d47ad402be601985cb8bf84/source/SkUGens/FM7.cpp`

Lerdahl, F. (1987, 01). Timbral hierarchies. *Contemporary Music Review*, *2*, 135-160. doi: 10.1080/07494468708567056

Mayer, D. (2020). *Algorithms in sound synthesis, processing, and composition: a dialectic game.* Almat 2020 - symposium on algorithmic agency in artistic practice. Retrieved 2021-03-28, from `https://www.researchcatalogue.net/view/921059/922503`

Olofsson, F. (2008). *fm7 patches.* Retrieved 2021-03-25, from `https://fredrikolofsson.com/f0blog/n-fm7-patches/`

Pluta, S. (2009-2021). *Live modular instrument.* `https://github.com/spluta/LiveModularInstrument`. GitHub.

Pluta, S. (2012). *Laptop improvisation in a multi-dimensional space* (Unpublished doctoral dissertation). Columbia University.

Sanfilippo, D. (2021). *modified lotka-volterra b.* Retrieved 2021-04-01, from `https://github.com/dariosanfilippo/modified_lotka-volterra_B/blob/main/modified_LV_B.dsp`

Suykens, J., Vandewalle, J., & de Moor, B. (1995). *Artificial neural networks for modelling and control of non-linear systems.* Springer US. Retrieved from `https://books.google.fr/books?id=cV7nkW9gMMgC`

Vickers, B., Allado-McDowell, K., & GPT-3. (2020). Atlas of anomalous ai. In (p. 9-27). Ignota.

Wessel, D. (2006). An enactive approach to computer music performance. *Le Feedback dans la Creation Musical*, 93–98.