5-2023

Survey of models for "computer music"

Gabriel Pell
University of Tennessee at Chattanooga, vyz797@mocs.utc.edu

Survey of Models for "Computer Music"

Gabriel S. Pell

Departmental Honors Thesis
The University of Tennessee at Chattanooga
Department of Computer Science

Examination Date: April 7th, 2023

Roland M. Howell
Professor of Computer Science
Thesis Director

David R. Schwab
Professor of Computer Science
Department Examiner

Jonathan B. McNair
Professor of Music
Department Examiner

Introduction:

Far before the age of digital computers, scientists, engineers, and composers have conducted exhaustive experimentation on the idea of music created without the traditional practices of human composition. Even as far back as ancient Greece, Pythagoras equated the works of musicians to the laws that bind nature while Ptolemy and Plato sought to replicate the supposed algorithmic nature, then called formalisms, of the many improvisational works of the time. Though their works were purely speculative and theoretical in nature, there is no question that their ideas of a piece composed entirely of mathematical statements with less emphasis on a creative element was indeed based in fact [1].

These ideas of a generated musical piece caught the attention of even the most famous of musicians later in history. Take, for instance, Wolfgang Amadeus Mozart, who saw the potential of a new factor in algorithmically creating works of auditory art: random chance. His contributions to the world of algorithmic music lie within his musical "games" that he titled *"Musikalisches Würfelspiel"*, or "Dice Music", in which small fragments of pre-composed musical ideas were chosen via throws of the dice. These stochastic methods of writing music never truly died out in the 18th century, but rather would carry over into later experiments of the more modern day, such the works of John Cage and his method of laying out a constellation chart over blank sheet music to simulate the process of notation and dynamics [1].

Because algorithmic composition does not inherently require either an analog or digital computer to be demonstrated, it also is not, by itself, an example of computer music. The composer Stephen Smoliar would note during his machine learning experiments with EUTERPE that there was a certain amount of available auditory material that was increased with the introduction of the electronic medium, as the machines used for musical research allowed the composer to synthesize most replicable sounds and timbres. It is this introduction of digital computers to this process that this paper hopes to focus on.

The manual methods of stochastic algorithmic composition are undoubtedly effective, as evidenced by the popularity of Mozart's Dice Music after his initial showings. However, the results of these experiments are either reliant on fully composed chunks of preexisting material or are so randomized in their structure that it becomes difficult to appreciate the final product. For these issues, the use of a computer would both ensure a more efficient process and allow for two more approaches to algorithmic composition besides those of only random chance. These methods are known as the "rule-based approach", by which software is used to interpret the musical validity of a generated piece, and the "artificial intelligence approach", by which those theorems are used by software to create something that isn't necessarily directly linked to one or two inspirational composers [1].

When discussing the various notable models for music composition from artificial intelligence, it becomes necessary to distinguish the different philosophies behind their respective designs. More than just the technical and algorithmic differences in these models, these differences in thought reflected on what the purpose of these creations were, whether it be as a demonstration of what a machine was capable of replicating, or as an experimental teaching tool. As we review the following models, it will become apparent that these philosophies correspond with the "eras" of AI Music Composition, whether because of technological capabilities having a direct impact on the designs, or more recently, what is deemed as popular applications of artificial intelligence.

It should be noted, however, that unlike traditional historical "eras", these periods of history in computer generation and artificial intelligence that this paper outlines don't have consistent dates or years to identify them. In fact, many instances of a ‘previous’ or ‘latter’ era can be found in the same point on a timeline. Rather, the differences in these eras lie primarily with substance and intention. For instance, the works of Rudolph Zaripov may have been published a few years after those of Hiller and Isaacson, but the philosophies of Zaripov are still very much consistent with that of the era before the Illiac Suite, an influential computer piece that will be explored in more depth with its appropriate era discussion [2].

“Pre-Composition / Augmented Stochastic” Era:

Before exploring the breakthroughs in computer generation led by the efforts of the more well-known composers and computer scientists, it is important to explore the efforts of those who gave those people the foundation to complete their models. Before the applications of software-based filters and rudimentary applications of machine learning, there were theories, both musical and scientific, which were not meant to construct entire scores in the blink of an eye, but rather to grasp the fundamentals of how such a composition could be achieved. Many of these engineers and composers will unfortunately never be granted the recognition they deserve.

Take for instance the works of Pierre Barbaud, a French composer (and interestingly enough, actor) who is credited as one of the inspirations of Lejarin Hiller, who would go on to generate the first completely computer-written score. Barbaud is from a time of experimentation where there was no blanket term for “Computer Music”. Instead, his work fell under the category of Generative Music, which was used to prove musical theorems and validity as opposed to Algorithmic Composition, which was purely to create a pleasing piece of music [2]. It seems none of Barbaud’s papers on generative music were ever translated into English, and what’s even more unfortunate is what was supposed to be his appearance at the First International Conference on Computer Music, Massachusetts Institute of Technology (ISCM). According to a review of the conference written a few months after its October 1976 debut, Barbaud's tapes were never shown during any of the panels, as they had apparently been "mislaid" the night before the ISCM dinner at the Statler Hilton Hotel in Boston [3].

Another of these generative music researchers is one from the then Soviet Union, and the first of the computer musicians to have been translated internationally, Rudolph Zaripov. His influential paper on computer music was translated 9 years after its publication in 1960, and Laejarin Hiller himself implies that Zaripov's work was underway before the completion of their Illiac Suite[3]. Where later computer scientists would be algorithmically constructing pieces that would take vague influences from baroque, classical, and contemporary composers, Zaripov would focus his research on Russian folk tunes; simplistic in structure but ideal for his studies into generating a theoretically valid tune. The Soviet-made "Ural" computer utilized a set of filters based on "rules" defined by Zaripov and his selection of folk songs. For instance, Zaripov notes that many of these tunes utilize a pattern found in the song "Young Maidens" in which later measures have the same rhythmic and melodic features repeated as previous measures, save for identical pitch, in the form of fragments or sequences. A "special apparatus", as he calls it, is designated with the ability to randomly generate number values in accordance with pitches, intervals, and durations. A filter is implemented so that only values that satisfy these "rules" can be transcribed, eliminating the potential discord that comes with a purely stochastic approach [4]. As you will see, this is a fairly similar approach for the algorithmic composition of models for the next era.

"Full Composition / Rule-Based" Era:

As mentioned previously, the first musical score to be completely written by a computer is credited, so to speak, to Lejarin Hiller, an assistant music professor, and Leonard Isaacson, a mathematician, both in association with the University of Illinois. So influential was this composition that when talking about computer music, it became impossible to dissociate their work with the field as a whole. A majority of Hiller's musical career had existed without the inclusion of any computers, and it wasn't until his collaboration with Isaacson in 1956 that he realized the potential presented by the University of Illinois's ILLIAC-1 (Illinois Automatic Computer) in modeling a fully-automated composition [5]. What resulted from the collaboration is a string quartet split into four movements. The 'Illiac Suite', as it was called, would serve as a foundation for the early research made into artificial intelligence and its applications into music composition.

However, the 'Illiac Suite' is not a continuous music product in the sense that a traditionally composed four-movement string quartet is. Rather, the piece is a result that samples from a series of four experimentations for how the ILLIAC-1 could use software to handle the composition process when accounting for a balance of absolute randomness and redundancy. To this end, Hiller and Issacson utilized the Monte Carlo method to randomly generate the tones that would be used in the four movements. The method was experimental at the time, and without the use of high-speed computers such as the ILLIAC-1, it was seen as slow for most problems. However, under the structured rules of a musical composition, the trials conducted with the method were a resounding success [6][7].

The first of these experiments was a way to construct the procedures that would govern the other three. It began with Hiller and Isaacson programming the ILLIAC-1 to generate a series of random integers ranging from 0 to 14 to represent the tones that would make up the first movement. As this was meant to be less complex, they decided to only use ‘white notes’, that is notes without sharps or flats, until the computer had eventually returned to ‘C’, the decided tonic note. As with Zaripov and many of the other computer composers during this time, there had to be guidelines based on preexisting styles to ensure that the end result was palatable. The rules that would be used to convert these tones from a random dissonant series to a serviceable melody was a musical idiom known as “strict first-species counterpoint”, which is a popular method of composition that emphasizes rules of dissonance and how to overcome it using multiple voices. The method was developed to imitate the unnamed rules used by Renaissance composers such as Giovanni Pierluigi da Palestrina, whose style was the primary one reflective in the first movement of the Illiac Suite. The second experiment and movement consisted of the same restrictions set about by the first, but with additional screening and an emphasis on Palestrina’s four-voice counterpoint [6][7].

It is when Hiller and Isaacson began their third movement of experimentation that dynamic and rhythmic variety could be added; where the computer is given less “creative” restrictions and is able to utilize the redundancy patterns programmed into the software to create something contrary to the first two movements. While Hiller noted that the third movement had more complexities in the structure due to the reduced restrictions on the Monte Carlo method, especially the more *avant-garde* qualities of the first section, the second section was noted to be easily comparable to that of a string quartet devised by Béla Bartók [6]. It is at this point that I must emphasize how quickly the transitions between that of a 16th-century piece could be fabricated into that of a 20th-century piece simply by changing the degrees of freedom by which a program follows. If anything, this is an illustration of just how correct the great thinkers of the classical Greek era were about the algorithmic nature of music as a whole. The introduction of digital computers was a way to realize to what extent that nature could be replicated given enough data, a subject that would be expanded upon in the continuing era of Artificial Intelligence.

When Hiller and Isaacson began development on the fourth and final movement of the Illiac Suite, their goal had shifted tremendously. There would now be very little influence from the structures of the preexisting counterpoint or contemporary compositions. Instead, the focus was to synthesize the tones from purely mathematical formulae. To this end, the stochastic nature of the note production process was done via the use of Markov Chains, for which each interval in the score was given a value of probability based on the 'weight' of how often each note should appear. Where the 'Chain' aspect of the term comes into play is when weights are decided upon from the previous occurrences. For instance, when the interval of a fifth appears in the piece, its weight is drastically decreased while still keeping the same base ratio to eventually return to for all 14 tones programmed into the Monte Carlo method [6][7].

While the Illiac Suite is by far the most recognized contribution to algorithmic music from Hiller, it is certainly not his only. He, along with Robert Baker began researching a method of computer-assisted composition almost ten years before its official release in 1963, making it an extension of his research when he was partnered with Issacson [8]. In a way, MUSICOMP (MUsic SImulator-Interpreter for COMPositional Procedures) was also an extension of the methods produced by Hiller's research into the Illiac Suite. In fact, the ILLIAC-1's involvement in the initial testing phases of MUSICOMP before its decommission in 1963 is proof of this. Hiller himself recalls that MUSICOMP routines had already been programmed into the ILLIAC-

1 before it was removed from service. As a result, he had to rewrite the entire program on the new IBM 7090 [9].

Once again, this new musical work was divided into a set of experiments for their movements (five this time), with each experiment representing a different approach for generating music. This, however, is where the similarities end. Unlike Hiller's previous work, the 'Computer Cantata' as it was called was less bound to structural parameters from any particular musical era and was instead dictated by musical logic similar to the final movement from the Illiac Suite. Most importantly, this was not just an attempt to compose a fluid, aesthetically pleasing piece, but rather a way to synthesize the various instruments and English vocal timbres that were available to the secondary machine that was being used during these experiments: the CSX-1 computer also in association with the University of Illinois. These timbres included various onomatopoeic noises like 'boom' 'crack' and 'snap' as well as instrumentation that was meant to cover the widest variety of sound qualities possible, from trumpet to maracas [9]. The final result is a piece that certainly achieves its goal of stretching the limits of the sound able to be synthesized by both machines. However, in doing so, the structure and tonality have become far more complex and dissonant than the more limited and musically conservative Illiac Suite.

One final compositional model that fits within the era of ruled-based algorithmic music is the program devised by the Greek composer Iannis Xenakis, which had no official name but was synonymous with his book Formalized Music [1]. This program was compiled in the FORTRAN IV language and utilized a similar form of Markov Chains that were demonstrated in the works of Hiller and his various partners. One distinction from these chains is expanded upon when

Xenakis explains the factors of each Matrix of Transition Probability (MTP), which are linked to one another through the characteristics of frequency, intensity, and density. With this distinction, the piece is less likely to be in the same perpetual state for a significant amount of time. Rather than encouraging a certain level of redundancy as previous compositional models had, Xenakis's MTP model increased the likelihood that the musical piece changes states, allowing for a higher degree of movement and variety within the composition [10].

Xenakis did not base his matrices of composition on a particular composer or musical era. Instead, he programmed his model to reflect his own unique style of avant-garde structure when creating works such as *Metastasis*, a work that sought to express his musical theories involving calculus and complex music theory [10]. In this way, his model was not an experiment to prove that a computer was capable of creating a piece of music, as that had already been proven with Hiller's works and even through the efforts of Zaripov a decade previously. Rather, his model's purpose was to aid its user in the process of composition, a purpose that would be explored more in the age of artificial intelligence.

Artificial Intelligence / Machine Learning Era:

Following the success of experimental computer music from composers such as Hiller and Xenakis, computer generated music became a subject not just reserved for the scientists and composers within the field. It garnered an influx of mainstream attention that would only grow as the years progressed. Just after Xenakis's publication of his book *Formalized Music* in 1963, Ray Kurzweil composed a piano piece created entirely by a computer and had a chance to display it on national television in 1965 with the popular game show "I've Got a Secret". In the show, one of the contestants correctly guessed that Kurzweil's performance of a previously unheard piano sonata was in fact, programmed by Kurzweil and entirely composed by a computer [11].

In the turbulence of this mainstream popularity, composers and engineers began to test the full capabilities of music-composing programs and dabble into the concepts of machine learning. An early example of this comes from Stephen Smoliar and his studies into the then recently developed EUTERPE system in 1967. While the original system was created by the computer scientist Marvin Minsky, who would later be the co-founder for the AI division at the Massachusetts Institute of Technology (M.I.T.), Smoliar was responsible for extending its usage to the MAC PDP-6 computer. EUTERPE acts as both an interpreter and compiler and consists of six "voice programs" or strings of words. These words can be of two types based in Smoliar's version of the system: 'Note words', which represent notes on the score of music, and 'Macro instructions' which give the program functions that are equivalent to the words labeling them. Even though EUTERPE was designed as a fully-packaged interpreter, it is still versatile enough to be embedded in higher-level languages such as Lisp, through which the composer could design their own functions to use when writing [12]. Smoliar's custom version of Minsky's design was created, not for a computer to compose a piece of music, but rather to aid the human composer using a high-level interpretation of their instructions. While the process today could be considered rudimentary, EUTERPE is a fine example of the breakthroughs made in technology motivated by the wishes of the composer to augment the creation process.

It was the following year of 1968 when what could be considered more modern research into the applications of artificial intelligence into music began. It was inevitable that the subject would be breached, as popular publications began to see the potential application of experimental artificial intelligence to the process of music creation. Take the writings of A.W. Slawson, a writer for *Journal of Music Theory*, who would claim something to the same effect, stating that questions in music analysis would increasingly become questions in AI. He would go on to explain that the future of algorithms dictating the musical structures in computer music would shift from the traditional Markov Chains to that of synthetic pattern recognition. The majority of the research at the time was led by two major papers which surprisingly have very little recognition or documentation today [2].

The first of these is “Pattern in Music” written by Herbert Simon and Richard Sumner in association with Carnegie-Mellon University, who explained the functionalities of two programs they authored. Sumner and Simon had first reported on a program they had written that translates language patterns from users into common musical notations, likely for the purposes of aiding the composer in notating delicate articulations and dynamics. Their second program had a function that was a bit of the inverse of most other models at the time. Its purpose was to take a section of encoded musical score and decipher an expressible structure found within the piece. In doing so, the program could compare unique musical styles from eras and artists and simulate the thought process a listener would experience while observing a new composition. Both of these models utilized two types of data structures to format their inputs and results: “declarative statements”, which consist of simple musical statements such as ‘the first note is C’ and procedural statements, which were included to characterize each feature of a piece such as ‘each note after the first is a fifth above its predecessor on the diatonic scale’ [2]. What resulted was a demonstration of how a machine’s understanding of musical structure could be used in future projects involving assisted notation and aesthetic evaluation.

The second of these papers is “Linguistics and the Computer Analysis of Tonal Harmony” written by Terry Winograd in association with M.I.T. His program was originally designed for the use of harmonic analysis, specifically chord labeling. His theories backing up the methods used in his model were based on the principles of *systemic grammar* which was developed in the early 1960s by M. A K. Halliday, in which there are a set number of options with an entry condition. This structure can be interpreted by a computer as a way to impart to the inputs given to it a sort of 'meaning' in a rudimentary sense. By using these semantic pathways, Winograd's Lisp (short for List Processing) program was able to avoid parsing through chord pressions and musical pathways that it deemed as ungrammatical or too ambiguous [2]. These two influential projects brought about a new baseline for how an AI model could interact with musical concepts and be tasked with creating something of its own. Interpretation of human motive was the new paradigm for creating machine-learning algorithms to interact with composers for decades until the revolutionary works of David Cope and his quintessential model EMI.

David Cope had composed with digital computers almost a decade before he had begun work on EMI, and had been manually transcribing for even longer. He noted during his musical experimentation with IBM computers that the works that would be composed would take an exorbitant amount of time and would often clash with his preferred composition structures. The inspiration for his ideal model truly unfolded in the early 1980's when he was commissioned for an opera, during which he was suffering from a deadly case of writer's block. He would ultimately finish a working version of this new project in the five years he was given to complete his commission, but documentation for a final product would not see light until 1991. In his words, his solution in fixing his predicament was to procrastinate and begin work on a separate project using his previous knowledge of AI systems to aid him in his compositions. This would eventually evolve into what is now known as Experiments in Musical Intelligence (EMI, or Emmy for short). Cope had noted while working on his earlier builds that he had trouble defining what his musical style was when he composed. Because of this, his first priority when creating EMI was to build a program that could take the data from his previous works and analyze them in order to find the 'style' of composition that ultimately defined them [13].

Much like the previous endeavors into musical AI, EMI is primarily an analytical tool for how a composer or listener would think through the process of a composition and is also written in the most popular AI language at the time: Lisp. The EMI system can perform analysis by utilizing a more inclusive and refined version of the finite Markov Chain series known as Augmented Transition Networks (ATN), a process that was first utilized by the researchers who developed Natural Language Processing (NLP). NLP is used to generate transformative yet equivalently meaningful natural language, or to reduce the limitations of machine understanding when 'conversing' with a human user. The ATNs take the natural language input and give the syntax flexibility to allow for sentence structures to be interchangeable but still valid in the eyes of the program. This same principle is applied to musical structure when used in EMI; phrases can have an organic balance of both repetition and variation without the need for constant redundancy [14][15].

Besides the breakthroughs in human language comprehension, by far the most vital element to EMI's composition capabilities is the use of reliable databases. Cope himself noted in his official documentation of EMI that the majority of failures that occurred during test runs of EMI were caused by poorly created databases, not faulty code. The viability of databases were determined not only by how free of formatting and notation errors they were, but also by how well they balanced their data between broadening different structures of composition and ensuring the composer's style was perceivable. Most often, databases store musical data for use in EMI via structures that Cope calls *events*. Each event describes the various attributes that pertain to an individual note, which are listed as five elements: on-time, pitch, duration, MIDI channel, and dynamics, all of which are represented numerically, or with the occurrence of rests, represented by simply not existing.

These two components, combined with a data file for global variables for the pattern matching program, allow for EMI to compare large sets of musical data, replicate their styles, and produce scores that have an uncanny resemblance to the source material without having to factor in stochastic dissonance. Once prototypes of the model were complete, Cope first set out to replicate pieces by well-known composers Bach, Mozart, Chopin, and Brahms to get a feel for how 'style' could be represented by EMI as a whole, an endeavor he achieved rather easily. Then, after using his new program to find his own definitive 'style', Cope was able to finish his opera commission in less than two days out of the five years he was given to finish it [13][14].

Final Thoughts:

Since the mainstream recognition of 'creative' artificial intelligence stemming from models such as EMI, computer composition has left the independent development cycle and has become a profitable asset for technology companies such as NVIDIA and IBM. As such, the models have become more proprietary and even for nonprofit organizations such as OpenAI, with their many repositories of free-to-use source code, there are not as many primary articles detailing the creative process from the developers of these models. By contrast, there is certainly no shortage of accessibility to the programs themselves. A simple web browser search for something like "AI Music" will result in countless builds of online software for music generation or articles detailing where to find these builds, almost all of it free-to-use and ranging from widely different genres and purposes.

The introduction of digital computers has allowed composers to experiment with the more algorithmic aspects of music creation in a way that still emphasizes the creative process, the process in this instance being the creation of algorithms to simulate the composition method. Since the successful endeavors of composers such as Zaripov and Hiller, the functionality of music-oriented computer programs to users even with little musical background became apparent, and the composition process was no longer delegated to the classically trained. Moreover, the alarmingly rapid evolution of modern artificial intelligence since the introduction of EMI such as the works of the OpenAI company has brought forth online software like MuseNet for auditory art aficionados to experiment with computer generated musical ideas without inherent knowledge of ATNs or neural networks. It is an unfortunate reality, then, that the most recent mainstream applications of artificial intelligence have been relegated to controversial discussions of visual art or the potential hazards when working with advanced chatbots. AI has so much more potential in composition and education yet musical AI tools are constantly sidelined by these more ambitious projects and controversies crowding public interest.

Works Cited:

[1]: J. Maurer A. IV. “A Brief History of Algorithmic Composition”. *Center for Computer Research in music and Acoustics.* March 1999. Available: https://ccrma.stanford.edu/~blackrse/algorithm.html (Accessed Feb 2023).

[2]: C. Roads, “Artificial Intelligence and music,” *Computer Music Journal*, vol. 4, no. 2, p. 13, 1980.

[3]: S. Arnold. “Review of First International Conference on Computer Music, Massachusetts Institute of Technology”. *Contact: a Journal for Contemporary Music.* 1977.

[4]: R. K. Zaripov and J. G. Russell, “Cybernetics and music,” *Perspectives of New Music*, vol. 7, no. 2, p. 115, 1969.

[5]: T. Patteson, “Lejaren Hiller – Musica Informatica”, 26 Feb. 2017. [Online]. Available: https://www.musicainformatica.org/topics/lejaren-hiller.php. (Accessed Feb 2023).

[6]: L. Hiller, 'COMPUTER MUSIC', Scientific American, vol. 201, no. 6th ed., p. 109-121, 1959.

[7]: L. Hiller, and L. Issacson, *Experimental Music: Composition with an Electronic Computer*. New York: McGraw-Hill, 1959.

[8]: “Musicomp – Musica Informatica”, [Online]. Available: https://www.musicainformatica.org/topics/musicomp.php. (Accessed March 2023).

[9]: L. Hiller, and R. Baker, 'Computer Cantata: A study in Compositional Method', Perspectives of New Music, vol. 3, no. 1st ed., p. 62-90, 1964.

[10]: M. Zaplitny, 'Conversation with Iannis Xenakis', Perspectives of New Music, vol. 14, no. 1st ed., p. 83-103, 1975.

[11]: “Ray Kurzweil on ‘I've Got a Secret’” YouTube, April 18, 2007. Available: https://www.youtube.com/watch?v=X4Neivqp2K4.

[12]: S. Smoliar, 'EUTERPE: A Computer Language for the Expression of Musical Ideas', Project MAC (Massachusetts Institute of Technology), vol. 129, no. 1st ed., 1967.

[13]: C. Garcia, “Algorithmic Music – David Cope and EMI”, 29-Apr.-2015. [Online]. Available: https://computerhistory.org/blog/algorithmic-music-david-cope-and-emi/. (Accessed March 2023).

[14]: D. Cope, 'Computer Modelling of Musical Intelligence in EMI', Computer Music Journal, vol. 16, no. 2nd ed., p. 69-83, 1992.

[15]: D. Cope, *Experiments in Musical Intelligence*. Madison, Wisconsin: A-R Editions.