Analysis of First-Counterpoint-Music Generator Based on Nyquist

Chencheng Liu
Data Science, University College London, London, U.K.

Keywords: Nyquist, Music Generator, First-Counterpoing-Music Generator.

Abstract: As a matter of fact, music composing based on software is widely analysed. This study introduces the development and implementation of the first species counterpoint music generator based on Nyquist programming. Reviewing early research on counterpoint music generators, including the pioneering work of Hiller and Isaacson, and the use of Variable Neighbourhood Search(VNS) algorithms to optimize generators in recent years, the author aims to create a music generator that can generate music according to the first species counterpoint rules, and in experiment, author introduces four rules that must be met to generate music, such as using only concertos and intervals and avoiding large leaps. By modifying the range and rules, the generator can quickly generate the music conforming to the counterpoint method, but the rules are too strict and the adjustment of the range is complicated. The summary points out that with the reduction of programming algorithms, more and more music creators can use computers to generate specific styles of music, Nyquist has great potential, especially in the composition of classical music.

1 INTRODUCTION

The first work on computer counterpoint dates back to 1958, when Hiller and Isaacson used the ILLIAC computer, a computer built at the University of Illinois, to compile a generator that could quickly generate music satisfying Fux counterpoint (Lejaren & Leonard, 1958). By refining the rules of counterpoint into regulations, and step by step, they make the music generated by the generator meet the corresponding requirements, proved that musical concepts can be interpreted into computer language to produce musical output. Next, considering about generate music less imitative and restrictive in scope, they wrote a program for the generation of dissonant chromatic music, to make music more like it's made by people and less like by machines. Finally, they used second-order Markov chains to make abstract musical rules mathematically embodied by computers. After this, Lewin also compiled the music generator, which generated music by adding "global rules" to the original first species counterpoint, making the generation more efficient, but less aesthetically friendly (Farbood & Schöner, 2001). He used his own understanding to change the rules, making the original rules simpler and easier to generate.

Although the pioneer research is half a century old, in the last 10 years, there have been studies that have changed the method of using programs to generate music that meets counterpoint, constantly trying to optimize the generator. Dorien and Kenneth generated the score satisfying the first species counterpoint through Variable Neighbourhood Search (VNS) algorithm, which has been implemented in Optimuse, a user-friendly software (Komosinski & Szachewicz, 2015). The VNS algorithm is also used in an Android software (FUX), which has been able to generate music that satisfies the fifth counterpoint according to the preset music length (Herremans & Sorensen, 2013).

At present, although there are many counterpoint-music generators, it is still necessary to study how to build that generator. The creation of modern music is not restricted only by the rules of counterpoint. Different styles of music also have to meet different rules, but it is less rigorous and convenient to evaluate than traditional counterpoint. Therefore, writing counterpoint music generator will be an important reference for computer composition.

Nyquist is a language for sound synthesis and music creation that combines the functions of score language and signal processing language, capable of simultaneously processing musical events and sound signals (Dannenberg, 2008). Based on an interactive

Liu and C.
Analysis of First-Counterpoint-Music Generator Based on Nyquist.
DOI: 10.5220/0013515800004619
In *Proceedings of the 2nd International Conference on Data Analysis and Machine Learning (DAML 2024)*, pages 285-288
ISBN: 978-989-758-754-2

Lisp interpreter, Nyquist is flexible and easy to use, allowing users to design instruments and generate sounds with simple expressions. It supports multiple platforms, including Linux, macOS, and Windows. Nyquist's core language is Lisp and has expanded its capabilities for sound synthesis and signal processing.

Because of Nyquist's ease of use and its support for mathematical models (e.g., Markov chains), the next generation of generators will be written entirely on the Nyquist platform.The attempt to use Nyquist to randomly generate music is not new (Li et al., 2024). Although the attempt of this paper is also random generation of music, it is very different from their research in that this attempt pays more attention to the generation of Nyquist sheet music.
Before jumping right into the topic, here is a brief introduction to the first counterpoint method. Counterpoint originated in ancient Europe, and the first known book on counterpoint was parallel organum (about 900 A.D), which was very different from what is known today (Jeppesen, 2013). All the theory of counterpoint introduced here are from Johann Joseph Fux (Fux & Wollenberg, 1992).

In classical composition, musicians classify note combinations (chords) by the relationship between two notes played at the same time (intervals), classifying them into perfect consonance, imperfect consonance and dissonance (Blom et al, 2016; Dannenberg, 2010). Perfect consonance: unison, fifth, octave; imperfect consonance: third, sixth and dissonance is any interval else. All chords that appear in the first species counterpoint are consonance. Also to be mentioned is the marching of music, where the implication of direct motion is that the interval of the previous chord coincides with the interval of the next chord, which is forbidden exists between perfect consonance in the first species counterpoint. With the above foreshadowing, the research method will be clearer.

2 DATA AND METHOD

The main goal of the project is to use Nyquist to create a program that can randomly generate music satisfying the first species counterpoint. At the beginning, this study will refer to the work of Hiller and Isaacson, and this study lists the rules that need to be met to generate music as rules:

- Only consonances are included in the music
- Start and end are perfect intervals
- Avoid leaps of major 6th or greater to keep the melody smoothly
- No direct motion to perfect interval

As a preliminary experiment, this study limited the generated music to 2 parts and the tonal range to 8 white keys (The mode used for the counterpoint part must be the same).

This research gave priority to programming the basic parts necessary to generate music, such as score generation and instrument functions. Since the listed rules do not include the regulation of rhythm, this paper naturally focused on the relationship of pitch between notes. Taking advantage of the feature of nyquist music score that can accept multiple notes to generate chords at the same time, this study listed the possible consonances and asked the music score to obtain tones only from these chords to satisfy the first condition. After making the generated music meet the second rule through the score editing function in Nyquist, it is found that the generated music was too monotonous and bland. The more restrictions was subjected to in the creation of music, the less diversity of works it could create. On the contrary, more freedom in the creation of music would make the generated music more possible. Considering that the music to be composed by the program would be constrained by the first pair of positions (which would be an extremely liberating constraint), this study increased the range of tones from the original 8 white keys to 10.

This study began to try to make the parallel notes in the music conform to the corresponding rules. First, this study classified chords with different intervals. By using the Markov chain function in Nyquist and the control of the first-order Markov chain, the perfect chord could not reach the next perfect chord directly, thus satisfying rule 4 (last rules). Meanwhile, in order to satisfy the third rule, this research further grouped chords with the same interval and modified the previous Markov chain. Groups with too large a pitch span cannot communicate with each other, so that the music is always smooth and stable. At this point, the program has satisfied all the rules listed above. In terms of generating the timbre of music, in order to restore the classical style as much as possible and identify the tone better, this research chose the piano sound effect brought by Nyquist as the playing instrument for generating the score.

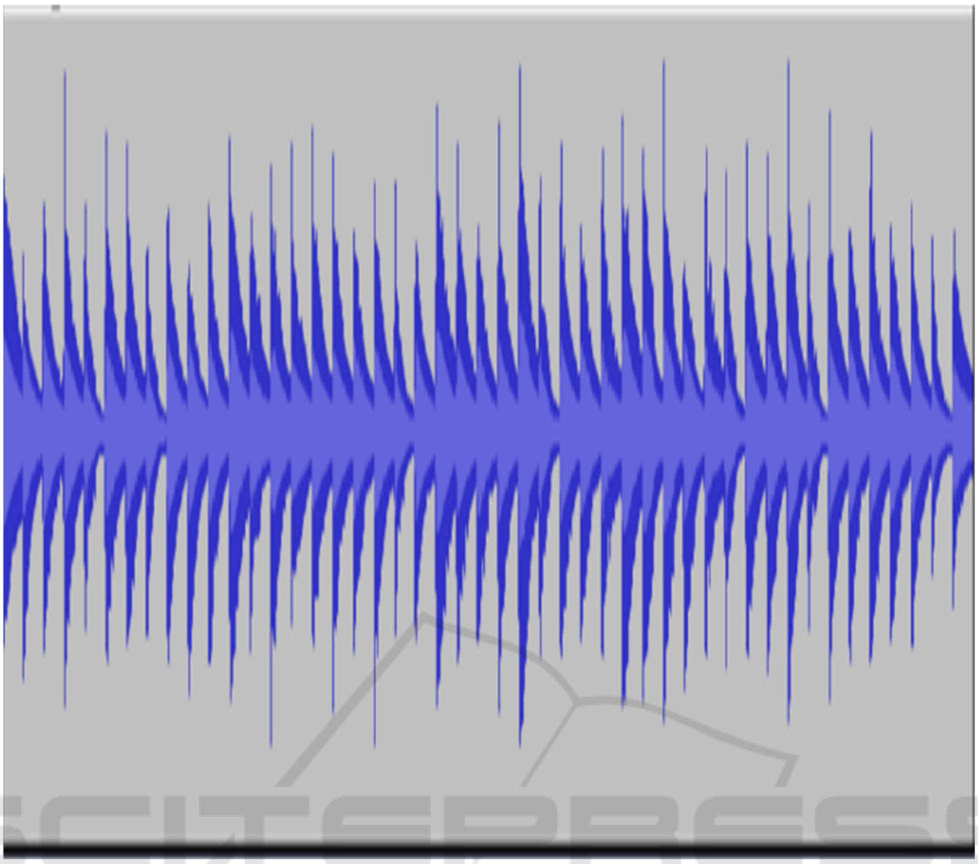

Figure 1: Generated music (Photo/Picture credit: Original).

3 RESULTS AND DISCUSSION

In addition to auditory recognition, Nyquist's program can further display the generated score, which is a randomly generated score with a total duration of 24 seconds and 0.5 seconds per syllable as shown in 'Figure 1'.

The first species counterpoint generator written by Nyquist in above way can indeed generate music that satisfies the first species counterpoint quickly and strictly, and can adjust the length of the music as needed. However, the current code still has many shortcomings and improvements. First of all, the rules in the generator cannot be freely deleted, and there are too many restrictions between different rules, which makes it difficult to change the rules later. Second, the range of the current generator cannot be changed quickly, and each change requires regrouping the chords, which greatly increases the workload, and later it may be possible to add some preconditions to meet the fast grouping of different ranges.

As for the prospect of this generator, because this attempt shows that it is feasible to use Nyquist to achieve random music generation satisfying stricter rules, it has certain guiding significance for the second counterpoint and even the fifth counterpoint music generation that may appear later. In addition, if combined with the powerful timbre processing function of Nyquist itself, in the case of meeting certain rules of classical music, breaking the rules of some traditional music generation, it may produce more creative works different from the music in the market today, and these works may not only be novel but also beautiful.

Artificial intelligence is expected to play a bigger role in the future. Automatic creation refers to the computer through a certain algorithm to compose music, assisted creation refers to the computer to simplify the creation process. Through algorithmic music creation, and DAW and related tools, researchers can realize that some commonly used music arrangement, mixing software can be included in this list. It is worth mentioning that Band in a Box, this software can automatically arrange and orchestrate music in a variety of styles, reducing the threshold of creation. In addition, some software (or apps) with game entertainment can also be called creative aid tools. For example, the Garage Band on ipad has the so-called Smart Drum. Percussion parts can be made by simply dragging the corresponding instrument to a certain position on an 8 × 8 panel (the vertical coordinate of the panel is loudness, the horizontal coordinate is rhythm complexity) instrument, electronic timbre and vocal synthesis: Synthetic instrument timbre provides the convenience of writing and listening for the creator, and also reduces the cost of playing (much cheaper than having the player record). Electronic timbre is an area that is still being explored, but it has been widely used and is also very important for electronic music, which has many well-known synthesizers from hardware to software. Music storage, retrieval and dissemination storage. Storage is mainly related to the technology in the field of signal processing, involving lossy compression coding and lossless compression coding. Lossless compression coding generally makes use of the information redundancy of music itself. In terms of retrieval, it mainly uses retrieval methods other than meta information (song name, singer name, etc.). For example, search by humming melody, search by beating rhythm, search by song style (mood), search by song, and so on. Music appreciation and analysis Appreciation is the reproduction of music through playback equipment, modern electronic music, the reproduction of concert recording (sound field reproduction) is a very important topic, there are many efforts in this area, such as stereo, such as head recording, automated evaluation. One can use computers to identify good and bad versions of works; use the computer to identify the level of the performer (can be used for the automation of the instrumental music test). Computers analyze the structure, orchestration, mood, and emotion of music.

4 CONCLUSIONS

Overall, this experiment does show that with the lowering of the threshold of modern computer programming, more and more music creators can use computers to create their own music faster, and through certain rules to let the computer generate the corresponding style of music. In addition, although the use of Nyquist to generate music that satisfies certain conditions is not new, the generation of music that satisfies the first pair method through specific Markov chains also demonstrates the greater potential of Nyquist, whose strict score generation function is also well adapted to classical arrangements that attach great importance to rules.

REFERENCES

Blom, D. M., Bennett, D., Stevenson, I., 2016. *The composer's program note for newly written classical music: content and intentions*. Frontiers in Psychology, 7, 1707.

Dannenberg, R., 2008. *Nyquist reference manual*. Springer.

Dannenberg, R. B., 2010. *Style in music.* The Structure of Style: Algorithmic Approaches to Understanding Manner and Meaning, 45-57.

Farbood, M., Schöner, B., 2001. *Analysis and synthesis of Palestrina-style counterpoint using Markov chains*. ICMC, 18.

Fux, J. J., Wollenberg, S., 1992. *Gradus ad Parnassum, Concluding Chapters*. Music Analysis, 11(2), 209-243.

Herremans, D., Sorensen, K., 2013. *FuX, an Android app that generates counterpoint*. IEEE Symposium on Computational Intelligence for Creativity and Affective Computing (CICAC), Computational Intelligence for Creativity and Affective Computing (CICAC), 48–55.

Jeppesen, K., 2013. *Counterpoint: the polyphonic vocal style of the sixteenth century.* Courier Corporation.

Komosinski, M, Szachewicz, P 2015. *Automatic species counterpoint composition by means of the dominance relation*. Journal of Mathematics and Music. Mathematical and Computational Approaches to Music Theory, Analysis, Composition and Performances, 9(1), 45.

Lejaren. H., Leonard, I., 1958. *Musical composition with a high-speed digital computer*. Journal of the Audio Engineering Society, 17.

Li, W., Ma, Z., Zou, Z., 2024. *Synthetic Music Random Generation Based on Nyquist.* Highlights in Science, Engineering and Technology, 85, 591-596.