Generación automática de notas musicales mediante autocodificadores variacionales condicionales
Automatic generation of music notes through conditional variational autoencoders

Trabajo de Fin de Grado

Curso 2024–2025

Autor

Pablo García López

Directores

Miguel Palomino Tarjuelo

Jaime Sánchez Hernández

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Generación automática de notas musicales mediante autocodificadores variacionales condicionales
Automatic generation of music notes through conditional variational autoencoders

Trabajo de Fin de Grado en Ingeniería Informática

Autor
Pablo García López

Directores
Miguel Palomino Tarjuelo
Jaime Sánchez Hernández

Convocatoria: *Junio 2025*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Resumen

Generación automática de notas musicales mediante autocodificadores variacionales condicionales

Un resumen en castellano de media página, incluyendo el título en castellano. A continuación, se escribirá una lista de no más de 10 palabras clave.

Palabras clave

Deep Learning, Conditional Variational Autoencoders

Abstract

Automatic generation of music notes through conditional variational autoencoders

An abstract in English, half a page long, including the title in English. Below, a list with no more than 10 keywords.

Keywords

Deep Learning, Conditional Variational Autoencoders

Contents

List of figures

List of tables

Chapter 1

Introduction

"Predicting the future isn't magic, it's Artificial Intelligence"
— Dave Waters

1.1. Motivation and Objectives

In recent years, deep learning has revolutionized generative tasks in fields like image synthesis, natural language processing, and audio production. Within music, research has generally split into *symbolic* approaches (focusing on note events, pitches, and durations in formats like MIDI) and *non-symbolic* approaches (focusing on raw audio waveforms or spectrograms).

Commercial digital audio workstations (DAWs) and synthesizers already allow users to generate audio with great precision. However, these are often not driven by *deep-learning*–based methods. Moreover, there is a compelling interest in exploring new audio possibilities achieved by *learned* latent representations, e.g., timbres or articulations that might not exist in standard synthesizer libraries.

This Bachelor's Thesis therefore focuses on the implementation of a **Conditional Variational Autoencoder (CVAE)** for directly generating non-symbolic musical notes. Through conditioning, guiding the network with desired musical parameters—such as approximate pitch, instrument type, or intensity—should be feasible. The aim is to combine the *flexibility* and *freshness* of learned audio representations with the *usability* of a straightforward interface that lets users "play" with different configurations to generate sounds. While the results may not surpass the polish or versatility of commercial synthesizers, such a model can reveal new pathways for interactive sound design and serve as a research-driven educational tool.

In any way, we would like this project to serve as an introduction and guide for students or anyone interested in the use of deep learning in music. While we do not assume extensive knowledge from the reader, we also will not go into excessively detailed explanations in order to keep the text accessible.

1.2. Work Plan

This section describes the work plan to follow in order to achieve the objectives outlined in the previous section.

(Pongo aquí esto mejor yo creo) We will need to define a metric for the loss function, in order to quantify how good the sample generation provided by the CVAE is.

Chapter 2

State of the Art

In this chapter, we aim to first provide a brief overview of the evolution of algorithmic composition and, secondly, explore non-symbolic (i.e., low-level) music generation more in depth.

2.1. Brief history of algorithmic composition

Algorithmic composition is the process of using some formal process to make music with minimal human intervention (Alpern, 1995) and can be divided into two main categories: *non-computer-aided* and *computer-aided* methods. The reader should note the following sections are nothing but a succint run-through of algorithmic composition and will necessarily be uncomplete in terms of its content.

2.1.1. Non-computer-aided methods

Algorithmic composition dates back thousands of years. In Ancient Greece, philosophers such as Pythagoras (500 B.C.) viewed music as fundamentally linked to mathematics, believing that musical harmony reflected universal order (Simoni, 2003). These ancient Greek "formalisms" however are rooted mostly in theory, and their strict application to musical performance itself is probably questionable (Grout and Palisca, 1996). Therefore, it can't really be said that Ancient Greek music composition was purely algorithmic in the sense we have defined it, but it undoubtably set the path towards important formal extra-human processes.

Ars Nova marked a pivotal shift in musical thought, where composers such as Philippe de Vitry and Guillaume de Machaut began to disentangle rhythm from pitch and text. By systematically applying rhythmic patterns—known as the *talea*—to fixed melodic cells called the *chroma*, they developed a method of composition that can be seen as an early form of algorithmic music-making (Simoni, 2003). This approach can be better understood by looking at Figures 2.2, 2.1 and 2.3, which

respectively represent the talea, chroma and the mapping between them of *De bon espoir-Puisque la douce-Speravi* by Guillaume de Machaut.

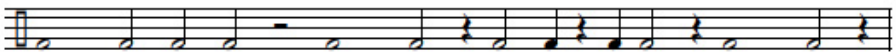

Figure 2.1: Talea of the isorhythmic motet *De bon espoir-Puisque la douce-Speravi* by Guillaume de Machaut

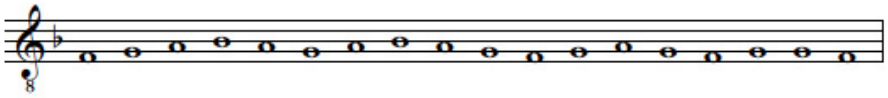

Figure 2.2: Color of the isorhythmic motet *De bon espoir-Puisque la douce-Speravi* by Guillaume de Machaut.

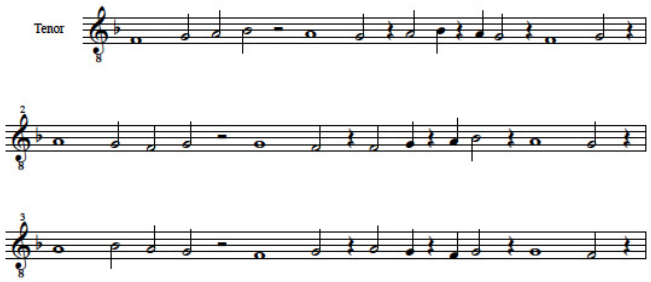

Figure 2.3: The tenor of *De bon espoir-Puisque la douce-Speravi* by Guillaume de Machaut

In the Renaissance and the Baroque periods, algorithmic methods became more explicit through forms like the canon, where composers, like Johann Sebastian Bach, created strict rules dictating how single melodies are to be imitated by multiple voices at different times.

A famous Classical-era example is Mozart's *Musikalisches Würfelspiel* ("Dice Music") in which musical phrases were randomly assembled by dice rolls to allow any composer to form a waltz, explicitly employing chance-based algorithmic composition (Maurer, 1999).

The 20th century introduced more complex algorithmic techniques through serialism, where composers like Arnold Schoenberg and Alban Berg employed systematic tone-row matrices (see Figure 2.4 to structure their compositions through fixed rules. Composers such as John Cage and Karlheinz Stockhausen later incorporated chance and probabilistic methods, further extending the tradition of algorithmic music before the advent of computers (Simoni, 2003).

2.1.2. Computer-Aided Methods

The advent of computers in the mid-20th century significantly advanced algorithmic composition, introducing computational techniques that expanded creative

	I_0	I_{10}	I_3	I_4	I_2	I_1	I_{11}	I_9	I_8	I_7	I_5	I_6	
P_0	E♭	D♭	G♭	G	F	E	D	C	B	B♭	A♭	A	R_0
P_2	F	E♭	A♭	A	G	G♭	E	D	D♭	C	B♭	B	R_2
P_9	C	B♭	E♭	E	D	D♭	B	A	A♭	G	F	G♭	R_9
P_8	B	A	D	E♭	D♭	C	B♭	A♭	G	G♭	E	F	R_8
P_{10}	D♭	B	E	F	E♭	D	C	B♭	A	A♭	G♭	G	R_{10}
P_{11}	D	C	F	G♭	E	E♭	D♭	B	B♭	A	G	A♭	R_{11}
P_1	E	D	G	A♭	G♭	F	E♭	D♭	C	B	A	B♭	R_1
P_3	G♭	E	A	B♭	A♭	G	F	E♭	D	D♭	B	C	R_3
P_4	G	F	B♭	B	A	A♭	G♭	E	E♭	D	C	D♭	R_4
P_5	A♭	G♭	B	C	B♭	A	G	F	E	E♭	D♭	D	R_5
P_7	B♭	A♭	D♭	D	C	B	A	G	G♭	F	E♭	E	R_7
P_6	A	G	C	D♭	B	B♭	A♭	G♭	F	E	D	E♭	R_6
	RI_0	RI_{10}	RI_3	RI_4	RI_2	RI_1	RI_{11}	RI_9	RI_8	RI_7	RI_5	RI_6	

Figure 2.4: Serialism matrix

possibilities. Early pioneers like Lejaren Hiller and Leonard Isaacson composed the *Illiac Suite* (1957), one of the first pieces generated entirely by computer algorithms (Hiller and Isaacson, 1959). They utilized a generator/modifier/selector framework, where musical materials were algorithmically created, modified, and selected based on predefined rules (Maurer, 1999).

Composer Iannis Xenakis introduced *stochastic music*, employing probabilistic methods to generate musical structures. For instance, in his work *Atrées* (1962), Xenakis used probability distributions and random number generators to determine musical elements (Xenakis, 1992).

Computer-aided algorithmic composition can be categorized into three main approaches:

1. Stochastic systems: they incorporate randomness, ranging from simple random note generation to complex applications of chaos theory and nonlinear dynamics (Nierhaus, 2009).

2. Rule-Based systems: these utilize explicitly defined compositional rules or grammars, similar to earlier non-computer methods like the Renaissance canons or serialist compositions we have talked about. Notable examples include William Schottstaedt's automatic species counterpoint program and Kemal Ebcioglu's CHORAL system, which generate music based on historical compositional rules (Cope, 1991).

3. Artificial Intelligence systems: these systems extend rule-based methods by allowing a computer to develop or evolve compositional rules autonomously. David Cope's Experiments in Musical Intelligence (EMI) exemplifies this approach, analyzing existing compositions to create new music emulating specific composers' styles (Maurer, 1999).

2.2. Non-symbolic music generation

In Section 2.1 we gave an overview of historical algorithmic composition along with its two main branches: non-computer-aided and computer-aided methods, which largely focus on *symbolic* or high-level approaches. In this section, however, we turn our attention to *non-symbolic* music generation, where the emphasis is on generating and shaping audio signals directly.

We begin with an overview of foundational digital synthesis systems, which provided the bedrock for modern audio generation. We then discuss recent AI-based approaches, including various deep-learning architectures capable of producing music at the waveform (or spectrogram) level. Although this thesis aims to ultimately employ a conditional variational autoencoder for generating musical notes, understanding the broader ecosystem of audio-focused methods places our work in context.

2.2.1. Traditional Synthesis Systems

2.2.1.1. Additive Synthesis

Additive synthesis is a sound creation method based on the Fourier Theorem, which states that any sound can be decomposed into a sum of sine waves, or partials (Fourier, 1822). By controlling the frequency, amplitude, and phase of each partial, one can construct complex timbres from these elementary components. Historically, this idea finds early expression in acoustic instruments such as the pipe organ (see Figure 2.5), where multiple pipes combine to produce rich harmonic textures, and in pioneering electronic devices like the Telharmonium—often considered one of the first additive synthesizers.

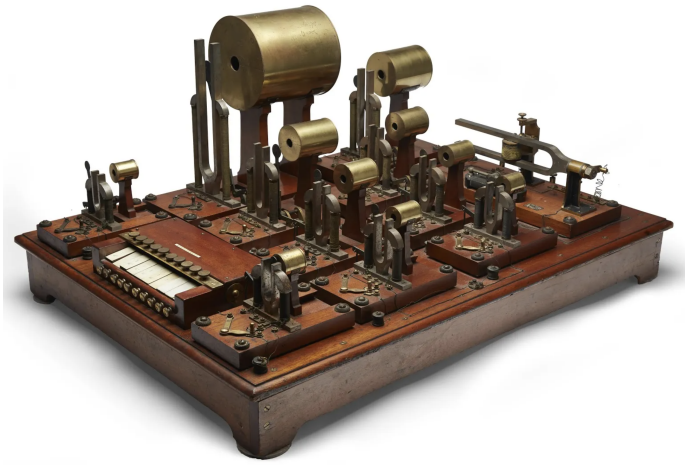

Figure 2.5: Pipe organ created by Hermann von Helmholtz around 1862

The method was further advanced in the mid-20th century through the work of Max Mathews at Bell Labs, who demonstrated the vast potential of digital addi-

tive synthesis for generating evolving and intricate soundscapes (Mathews, 1963). Although the flexibility of additive synthesis allows a precise crafting of any sound, its complexity made it less practical compared to the more cost-effective subtractive synthesis during the analog era. With the rise of digital signal processing, however, additive synthesis experienced a revival. This influenced the appearance of modern hybrid synthesizers that incorporate both additive and substractive techniques (Roads, 1996; Tagi, 2023a).

2.2.1.2. Subtractive Synthesis

Subtractive synthesis is one of the most widely used methods in sound synthesis systems. Conceptually, this approach is not harder to understand than additive synthesis: starting with a complex waveform as the raw material, we want to shape it by filtering out unwanted frequencies, much like sculpting a figure from a block of marble. What do we shape this raw signal with? Well, a substractive synthesizer primarily uses these components:

- Oscillators: is the responsible for generating the initial complex waveforms rich in harmonics.
- Filters: which remove (or subtract) selected frequency components. This can be done with filters such as the so-called low-pass or high-pass, which respectively remove high and low frequencies.
- Amplifiers and Envelope Generators: amplifiers control the overall level of the sound over time while an envelope generator is a tool that shapes how a sound evolves when a note is played by controlling four different dimensions (see Figure 2.6):

  1. Attack: how quickly the sound reaches its peak.
  2. Decay: how fast it drops from the peak to a steady level.
  3. Sustain: the level at which the sound holds while the note is sustained.
  4. Release: how rapidly the sound fades after the note is released.

  These simple stages allow you to craft sounds that can be sharp and percussive or smooth and evolving (Hahn, 2022).

- LFOs (Low-Frequency Oscillators): LFOs operate at very low frequencies that are below the threshold of human hearing and can create effects like vibrato or tremolo, therefore bringing the possibility of adding movement and life to a sound (Tagi, 2023c).

Historically, subtractive synthesis dates as back as 1930 with instruments such as the Trautonium and continued to be used throughout the 20th century, for example, by Robert Moog's Minimoog (Réveillac, 2024).

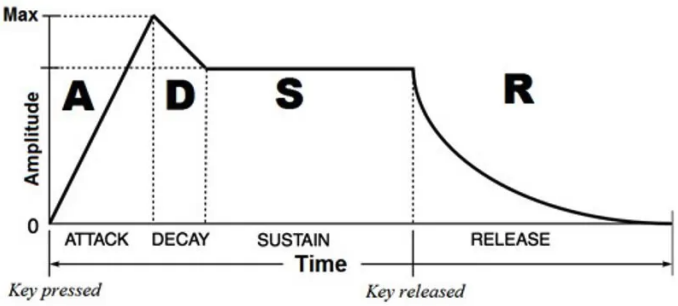

Figure 2.6: Illustration of Attack (A), Decay (D), Sustain (D) and Release (R)

2.2.1.3. Frequency Modulation (FM) Synthesis

Frequency modulation synthesis (FM synthesis) is a method of sound design in which one oscillator, known as the *modulator*, modulates the frequency of another oscillator, called the *carrier*, which allows to create new frequency components without filters (see Figure 2.7). In simple terms, rather than "sculpting" a sound by removing frequencies (as in subtractive synthesis), FM synthesis generates complex spectra by dynamically altering the pitch of a carrier with a modulating signal (Cymatics, 2025).

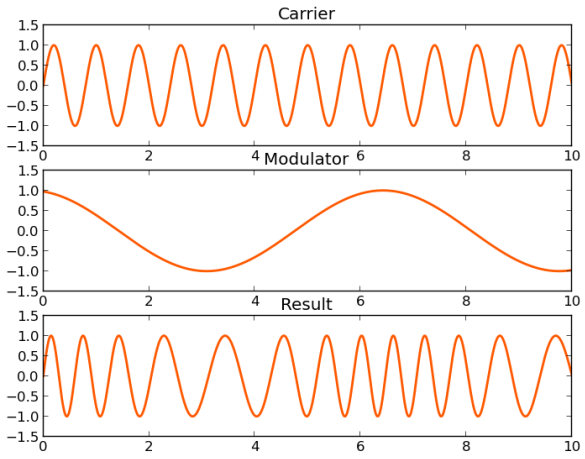

Figure 2.7: Illustration of a carrier, a modulator and the output

FM synthesis is the result of John Chowning's experiments in 1967 at Stanford University: by using sine waves (using one to modulate the frequency of another) Chowning discovered that a variety of new timbres could be generated. (Cymatics, 2025).

FM synthesis revolves around the building block of an *operator*, that typically includes an oscillator, an amplifier, and an envelope generator (recall we have talked about these in substractive synthesis). Operators can serve as carriers, modulators, or both, and they are arranged in various configurations or algorithms to produce different sound textures (Tagi, 2023b).

2.2.1.4. Other Approaches: Granular, Physical and Spectral Modeling

Beyond the traditional methods of additive, subtractive, and FM synthesis, there do exist other important and more modern sound generation techniques. We will very briefly talk about three of them.

Granular Synthesis works by breaking a sound into tiny segments called grains. These grains can then be individually rearranged to create a rich variety of sound textures, from subtle ambiences to complex, glitch-like effects (Roads, 1996).

Physical Modeling Synthesis takes a different route by simulating the behavior of real-world systems, such as vibrating strings, which allows for realistic emulations of acoustic instruments (Allen and McCreary, 1997).

Spectral Modeling involves analyzing a sound's frequency content (often with Fourier techniques) and then resynthesizing it by manipulating its spectral components. This way, interpolation and morphism between sounds can be achieved in a simpler way than with other traditional synthesis methods.(Serra, 1998).

2.2.2. Modern AI-Driven Non-Symbolic Music Generation

Unlike the traditional systems based on handcrafted signal-processing algorithms, deep learning methods for non-symbolic music generation learn representations directly from data. They typically produce raw audio waveforms or time-frequency representations, such as spectrograms. In recent years, several influential neural architectures have emerged, capable of generating musical audio directly at the waveform level. Our model will also follow this paradigm.

2.2.2.1. Waveform Modeling Approaches

WaveNet is a neural network initially designed for generating realistic speech audio directly from waveform samples. WaveNet operates by predicting each audio sample based on previously generated samples, using dilated causal convolutional layers. These dilations expand the receptive field, allowing the network to capture both fine-grained details and wider temporal context, which seems essential for modeling realistic audio textures. Although initially designed for text-to-speech synthesis, WaveNet was quickly adapted for music and demonstrated its effectiveness in capturing musical features at the waveform level (van den Oord et al., 2016).

Another significant development was the introduction of **SampleRNN** (Mehri et al., 2017), a hierarchical recurrent neural network (RNN) architecture specifically created to handle the complexity of raw audio generation. SampleRNN models waveforms at multiple temporal scales by stacking RNN layers hierarchically, allowing each layer to focus on different aspects of musical structure. Higher layers manage broader temporal dependencies, capturing long-term patterns, while lower layers handle local audio details. (Maurer, 1999).

Another significant breakthrough in non-symbolic music generation was achieved with Generative Adversarial Networks (GANs). An important example is *GANSynth*, developed by *Google Magenta*, which synthesizes audio notes using generative adversarial networks operating in the frequency domain (Engel et al., 2019). Unlike WaveNet and SampleRNN, which sequentially generate each sample, GANSynth produces entire audio clips simultaneously by generating spectrograms and instantaneous frequency components. This approach results in more realistic and coherent musical timbres. Additionally, GANSynth allows for audio synthesis control, which enables independent manipulation of pitch and timbre, making musical creativity and composition easy.

Chapter 3

Basic musical concepts

[Explain basic musical concepts and put émphasis on those that the dataset uses]

Chapter 4

Introduction to Deep Learning and Conditional Variational Autoencoders

[Here I want to give a basic introduction: what deep learning is, what we understand by learning, training a model, the importance of the dataset (and its partitions), loss functions, etc. I do not intend to go into much detail but rather provide the main intuitions.]

[Then, I want to focus on VAEs, starting by explaining what Autoencoders are, what Variational Autoencoders are, and Conditional Variational Autoencoders (architecture, uses, problems, etc.). I also want to succinctly explain the mathematics behind them.]

Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter **??**.

Bibliography

Allen, J. and McCreary, G. *Physical Modeling of Musical Instruments*. Wiley, 1997.

Alpern, A. Techniques for algorithmic composition of music. `http://alum.hampshire.edu/~adaF92/algocomp/algocomp95.html`, 1995. Accessed: March 2024.

Cope, D. *Computers and Musical Style*. A-R Editions, 1991.

Cymatics. Sound design basics: Fm synthesis. `https://cymatics.fm/blogs/production/fm-synthesis?srsltid=AfmBOoq29DsSPaqoN8ozE9GTad-5rQlqAV6igbftb_BQZyQ4mJmBgtFk`, 2025. Accessed: 2025-03-13.

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A. Gansynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations (ICLR)*. 2019. Accessed: March 2024.

Fourier, J. B. J. *The Analytical Theory of Heat*. C. G. G. and J. A. G. Balbi, 1822. Foundational work on Fourier analysis, which underpins additive synthesis.

Grout, D. J. and Palisca, C. V. *A History of Western Music*. W. W. Norton & Company, New York, 5th edn., 1996.

Hahn, M. Subtractive synthesis: Learn synthesizer sound design. `https://blog.landr.com/subtractive-synthesis/`, 2022. Accessed: 2025-03-13.

Hiller, L. and Isaacson, L. *Experimental Music: Composition with an Electronic Computer*. McGraw-Hill, 1959.

Mathews, M. Computer music. *Computer Music Journal*, Vol. 7(4), 18–37, 1963. Pioneering work in computer music demonstrating the potential of additive synthesis.

Maurer, J. A. A brief history of algorithmic composition. 1999. Retrieved from `https://ccrma.stanford.edu/~blackrse/algorithm.html`.

MEHRI, S., KUMAR, K., GULRAJANI, S., KUMAR, R., JAIN, S., SOTELO, J., COURVILLE, A. and BENGIO, Y. SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. In *5th International Conference on Learning Representations (ICLR)*. 2017. Accessed: March 2024.

NIERHAUS, G. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer, 2009.

VAN DEN OORD, A., DIELEMAN, S., ZEN, H., SIMONYAN, K., VINYALS, O., GRAVES, A., KALCHBRENNER, N., SENIOR, A. and KAVUKCUOGLU, K. Wavenet: A generative model for raw audio. `https://arxiv.org/abs/1609.03499`, 2016. Accessed: March 2024.

ROADS, C. *The Computer Music Tutorial*. MIT Press, 1996. A comprehensive overview of computer music techniques, including additive synthesis.

RÉVEILLAC, J.-M. Synthesizers and subtractive synthesis 1: Theory and overview. 2024.

SERRA, X. *Spectral Modeling Synthesis: Theory and Applications*. Oxford University Press, 1998.

SIMONI, M. *Algorithmic Composition: A Gentle Introduction to Music Composition Using Common LISP and Common Music*. Michigan Publishing, University of Michigan Library, Ann Arbor, MI, 2003.

TAGI, E. Synthesis methods explained: What is additive synthesis? `https://www.perfectcircuit.com/signal/what-is-additive-synthesis`, 2023a. Accessed: 2025-03-13.

TAGI, E. Synthesis methods explained: What is fm synthesis? `https://www.perfectcircuit.com/signal/what-is-fm-synthesis`, 2023b. Accessed: 2025-03-13.

TAGI, E. Synthesis methods explained: What is subtractive synthesis? `https://www.perfectcircuit.com/signal/what-is-subtractive-synthesis`, 2023c. Accessed: 2025-03-13.

XENAKIS, I. *Formalized Music: Thought and Mathematics in Composition*. Pendragon Press, 1992.

Appendix A

Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Appendix B

Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.