Musical Applications of the Enigma Cipher

Akira Brown

MSc by Research

Reid School of Music

Edinburgh College of Art

School of Humanities and Social Sciences

2023

Abstract

The use of ciphers in music composition has traditionally been concerned with the concealment of information rather than its conception, however, the creation of pseudo-random sequences is a unique biproduct of the encipherment process. This research project examines how the Enigma cipher system can be employed as a pseudo-random generator (PRG) in the composition of electronic and acoustic music. Through the research and development of an Enigma cipher simulation in Pure Data (Pd), approaches to parameter mapping and audification of Enigma PRG sequences are explored and applied in the creation of a portfolio of musical works.

Adherence to determinist systems, post-tonality, and random processes, typifies the electronic and acoustic avant-garde music of the 20th century. The structural complexity of music from this era tends to exhibit randomness properties regardless of the level of determinacy in its composition. Boulez's view of serialism as "mechanical wheels-within-wheels tending almost towards the random" is a description that is equally applicable to mechanical cipher devices such as the Enigma, especially given its innate capabilities as a pseudo-random generator. This project calls upon several key works of this era, each of which serve as a conceptual framework to support the composition of music using the Enigma cipher and its pseudo-random sequences.

Declaration

I declare that this thesis and research project has been composed solely by myself and that the work has not been submitted for any other degree or professional qualification.

Akira Brown 30 August 2023

Acknowledgements

I would like to thank my supervisors Dr. Tom Mudd, Dr. Jules Rawlinson, and Dr. Martin Parker. Their insight and encouragement during my research project was invaluable and their overall support during my time at Edinburgh University made me feel welcomed.

I would like to thank my colleague and friend Harry for pushing me to pursue my research, and my friends Jen, Alex, Greg, and Jeremy for listening to my endless rambling about Enigma machines.

Finally, I must thank my partner Amy for her endless patience, sage advice, and for keeping me on track towards the end of my MA.

Contents

List of Figures

List of Files

Files a located on GitHub accessible via the following link:

https://github.com/akirabrown-gh/enigma

Folder `portfolio_music` contains the audio recordings that contribute to the portfolio of works for this submission. This folder includes:

```
1_Ringstellung.ogg
2_Durations_1a.ogg
3_Durations_1b.ogg
4_Study_13.ogg
5_Enigma_Krell_1.ogg
6_Enigma_Krell_2.ogg
7_ENIGMA-N.ogg
8_Ciphertext_for_Susan_merrick.ogg
9_W.ogg
10_ENIGMA-N_Live_at_Q121.ogg
```

Folder `portfolio_scores` contains the music scores that contribute to the portfolio of works for this submission. This folder includes:

```
durations_1a_score.pdf
durations_1b_score.pdf
ringstellung_doublebass_score.pdf
ringstellung_full_score.pdf
ringstellung_pre-score.xlsx
ringstellung_viola_score.pdf
ringstellung_violin1_score.pdf
ringstellung_violin2_score.pdf
ringstellung_violincello_score.pdf
```

Folder `pd_patches` contains the Pd objects and example patch files that contributed to this part of the submission:

A1.enigma.cipher.pd
A2.enigma.printing.pd
B1.enigma.random.pd
B2.enigma.monitoring.pd
B3.enigma.no-input.encryption.pd
B4.enigma.distribution.pd
B5.enigma.random.walk.pd
B6.enigma.interpolation.pd
bohlen.pd
bohlen~.pd
C1.musical.cipher.pd
C2.bohlen.pd
C3.easley.pd
D1.enigma.sonification.pd
D2.enigma.morphing.pd
D3.enigma.mixture.pd
D4.enigma.morphing.mixtures.pd
D5.enigma.duration.pd
D6.enigma.time-structures.pd
D7.enigma.trapezoids.pd
E1.enigma.audification.pd
E2.enigma.oscillator.pd
E3.enigma.noise.pd
E4.enigma.tenney.pd
E5.enigma.wavetable.pd
E6.enigma.drone.pd
E7.enigma.morphing.wavetable.pd
E8.enigma.waveshaping.pd
easley.pd
enigma_noise.wav
every.pd

```
F1.more.musical.cryptography.pd
F2.enigma.text-to-music.pd
list_enigma.pd
output~.pd
signal_enigma.pd
steep~.pd
```

Folder `no-input_encryption` contains the example no-input encryption sequences in .txt and .wav formats which contribute to this part of the submission:

```
no-input_16900-characters.txt
no-input_42250-characters.txt
no-input_16900-characters.wav
no-input_33800-characters.wav
no-input_118300-characters.wav
no-input_422500-characters.wav
```

Chapter 1 Introduction

Through practice-based research, this project contributes to the existing field of research into the use of ciphers systems in music composition and supports the creation of accessible and sustainable software in the field of computer music. Specifically, it documents the use of the Enigma cipher to generate pseudo-random sequences and their application in the creation of acoustic and electronic music. To this end, the project documents the research and development of an Enigma simulation in the Pure Data programming language, presents an investigation into the Enigma as a pseudo-random generator (PRG), explores the musical implications of alphabetical parameter mapping, and employs the Enigma PRG in the creation of a portfolio of musical works.

The process of encipherment is, by default, a process of creation, the characteristic of which is unique to each cipher system. What differentiates the use of ciphers in this research project from their application in musical cryptography, is use of ciphers to *create* rather than *conceal* information. It has long been acknowledged that cipher text by itself is perceivably random in its appearance. This unique biproduct of the cryptographic process, when utilised as a method of musical control, draws an interesting connection between the Enigma cipher and random processes such as chance operations and sources of uncertainty. Furthermore, the project's use of ciphers in music goes beyond the tuning restrictions of acoustic instrumentation in historic musical cryptography and explores the potential of microtonality in musical ciphers.

In terms of musical style, the works in this portfolio and the inspirations behind their creation are heavily informed by Western avant-garde music from between 1950 to the early 1970s. Specifically, this era is seen to encompass both acoustic and electronic music composed using serial composition methods, post-tonality, and random processes. This locus and era of music represents a strict adherence to determinist and non-determinist systems of composition with an emphasis on the ordering of musical events. Furthermore, the primigenial state of studio technology in early electronic music prescribes the use of simple sine-tones and noise bands, often under stringent algorithmic control. In consequence, the Enigma cipher system and algorithmic potential of Pure Data are well suited to the realisation of avant-garde music from this point in time.

Although this research project is not the first instance of the creation of an Enigma cipher simulation in a music-oriented graphical programming language like Pure Data, it is however a detailed academic account of the simulations research and development that is so far absent from academic literature. In addition, the realisation of this simulation in an accessible open source programming language encourages its future use in both a creative and research capacity.

This introductory chapter begins by providing some definitions of the terminology used throughout the research project, starting with those related to the cipher systems and Enigma, followed by an overview of the research field of musical cryptography, a definition of pseudo-random, a brief overview of Pure Data, and finally a definition of sonification and its related terms. The chapter then presents my own motivations for research, my rationale for using Pure Data as a programming language in which to develop an Enigma cipher simulation, and a summary of the aims and contributions of this research project. Finally, an overview of the structure of the thesis and the contents of each chapter are given.

1.1 Definitions

Cipher Systems and Their Usage

As this research project is centred on the use of the Enigma cipher in the context of music composition, definitions of cipher systems and the associated terminology are provided. Churchhouse (2009, p. 3) defines:

“A *cipher system*, or *cryptographic system*, is any system which can be used to change the text of a message with the aim of making it unintelligible to anyone other than the intended recipients. The process of applying a cipher system to a message is called *encipherment* or *encryption*. The original text of a message, before it has been enciphered, is referred to as *the plaintext*; after it has been enciphered it is referred to as the *cipher text.* The reverse process is *encipherment*, recovering the original text of a message from its enciphered version, is called *decipherment* or *decryption*." *Cryptography* then, "is the study of the design and use of *cipher systems*".

The terminology and definitions of cipher systems are applicable to understanding the working principle of the Enigma cipher in Chapter 2.

Enigma Cipher Machines

The term *Enigma machine* is commonly used to describe a whole family of electromechanical cipher machines that were produced under the brand name Enigma (Crypto Museum, 2023a). Prior to WWII, Enigma machines were marketed for use in a variety of governmental, industrial, commercial, and domestic environments as a means of securing information (Cryptologia, 2001). This project focuses on this commercial line of Enigma machines, specifically the Enigma D (A26) and Enigma K (A27). With this detail established – except for the detailed analysis of the cipher in Chapter 2 – the majority of the text will simply use the designator 'Enigma cipher' or 'Enigma PRG' (pseudo-random generator) where applicable.

Enigma cipher machines are electromechanical implementations of a cipher system. This cipher system may be explained both mathematically and by describing the electromechanical operation of the machine itself. Furthermore, this electromechanical cipher system may be simulated using mathematical operations that are typically found in computer programming languages. Therefore, the software simulation developed for this research project is considered a simulation of the cipher system in the Enigma D (A26) and Enigma K (A27).

Musical Ciphers and Musical Cryptography

This research project is closely associated with the fields of musical ciphers and musical cryptography due to its use of cipher systems in the process of music composition. Traditionally, a practitioner of musical cryptography will employ a musical cipher to conceal a message, often the name of a place or person, in the score of a composition (Miranda, 2016).

The use of ciphers systems–both inside and outside of music composition–to *create* rather than *conceal* information does not, however, appear to have a precursive field of research. As such, this research project contributes to the broader field of ciphers in music, in which the practice of musical cryptography is a part of.

Pseudo-Random

Definitions of pseudo-random and pseudo-random sequences largely concentrate on such sequences having the ‘appearance’ of randomness despite their being generated from a deterministic process, and the requirement that such sequences must “satisfy one of more standard tests for statistical randomness” (Merriam-Webster), such as the NIST and Diehard statistical test suites (RANDOM.ORG, 2023).

The term pseudo-random came into regular use in academic literature around the turn of the 1960s to describe “random numbers which are obtained on computers” (Hull and Dobell, 1962, p. 230). Titsworth and Welch state that “pseudo-random sequences are periodic sequences which have certain random-like properties” (Titsworth and Welch, 1959, p. 2).

Pure Data

According to the Pure Data homepage (puredata.info, 2023), “Pure Data (or just “Pd”) is an open source visual programming language for multimedia”. It is a “major branch of the family of patcher programming languages known as Max”.

The shorter term “Pd” will be used in reference to Pure Data for the majority of this thesis. Justification for the use of the Pd programming language for the simulation of the Enigma cipher and its use in the creation of a portfolio of works will be given later in this chapter.

Sonification

At its root, *sonification* is “the practice of generating sound, or music from data sets” (Doornbusch, 2009). Parkinson and Tanaka (2014, p. 151) clarify further, stating “sonification is, at its most basic, the conversion of something from a non-sound medium into sound”. A recent example of sonification can be found in Guy Birkin’s (2020) *SARS-CoV2_LR757995_2b* in which the coronavirus genome sequence is converted to MIDI data and mapped to musical parameters. Grond and Berger (2011, p. 363) define *parameter mapping sonification* as “the association of information with auditory parameters for the purpose of auditory display”. This project employs a similar parameter mapping process in the association of Enigma-derived pseudo-random sequences with musical parameters such as pitch, duration, dynamics, and timbre.

An alternative method of sonification to parameter mapping is that of *audification*. Dombois and Eckel (2011) state that audification is “interpreting any kind of one-dimensional signal (or of a two-dimensional signal-like data set) as amplitude over time and playing it back on a loudspeaker for the purpose of listening”.

Whereas the process of parameter mapping is used in this project to determine the ordering of music events in the compositions, audification has been employed a means to sound synthesis using data from the Enigma cipher.

1.2 Motivations for Research

The practice-based research that underpins this project represents a continuation of my personal creative practice as a sound artist working with analogue and computer music environments. The majority of this work manifests as recorded improvisation and live performance using modular synthesizers, however, the exploration of experimental sound techniques using Pd has been a passion for over a decade. However, the patches developed in Pd during this time have found comparatively little use in my creative output when compared to that made using modular systems. This reflection was a key motivation for using Pd in an academic research project, with the aim of applying my knowledge and experience in this programming language to the production of both a creative output and a significant contribution to music research.

The use of the Enigma cipher in this context was first explored a year prior to this project and encapsulates an interest coding non-musical algorithms in Pd for sonic experimentation. In preparation for exploring the musical applications of the Enigma cipher in an academic capacity, a gap in research discourse on the musical use of cipher systems soon became apparent. Whilst the Enigma machine has previously been simulated in a patcher language and used in a musical work, this idea has not been thoroughly explored at an academic level. Matmos' piece *Enigma Machine for Alan Turing*, first performed in 2006, is credited to feature an original Enigma machine and employs an "Enigma Encryption Max Patch" whereby "MIDI information within the song is being encrypted by a JAVA emulation of the Enigma machine" (Matmos, 2020). Beyond these credits no further information is provided on the role that the Enigma machine played in the work. Therefore, a further motivation for this project was the creation of a more concise record of the use of an Enigma cipher in music composition.

In addition to the above reflections, the accessibility and sustainability of music software environments is another important factor in the motivation to conduct this research project using Pure Data. Since Pd is an open source programming language, developing an Enigma cipher natively in this environment and making it publicly available further contributes to this previously unexplored area of research.

1.3 Why Pure Data?

The list_enigma object was created in the Pure Data open source visual programming language. Developing the object in this language allows for the Enigma cipher to be employed as a piece of music technology, where it can be used to manipulate and create musical data. list_enigma may be used as a cipher in Pd just like any other Enigma simulator, with the caveat being that it is restricted to simulating the commercial Enigma D (A26) and K (A27) models only. The main advantage for this object existing in software like Pd is that the inputs and outputs (and internal states) of an Enigma cipher may be automated and connected to a variety of sound generators and processors, in addition to sending/receiving MIDI information to external devices. This allows for a comprehensive integration of the Enigma cipher into a music technology environment.

Simulating the Enigma cipher in an open source program also increases accessibility of this research project, since the list_enigma object and the accompanying example patches can be explored without the need for a commercial software licence or dongle. Users can freely use the list_enigma object for their own creative works or future research into this field.

1.4 Aims and Contributions (Summary)

1. To produce a simulation of the commercial Enigma cipher in a computer music environment using an open source programming language and to make this accessible to the public for creative use and further research.
2. To document the research and development of an Enigma cipher simulation in the graphical programming language Pure Data.
3. To make accessible a thorough methodology and initial taxonomy on the use of the Enigma cipher as a pseudo-random generator, including a primary analysis of its pseudo-random sequences.
4. To further explore the approaches to alphabetical parameter mapping than those in musical cryptography through microtonal tunings.
5. To contribute to the current field of research on the use of ciphers in music composition and expand on the traditional uses of ciphers in this field.

1.5 Chapter Overview

Following this initial introduction, a further four chapters are included in this thesis. An overview of each chapter and their contents is provided below.

Chapter 2 The Enigma Cipher

The second chapter, *The Enigma Cipher*, presents a brief overview of early commercial Enigma machines and a detailed description of the working principle of the cipher system in the commercial Enigma models D (A28) and K (A27). The chapter begins with an overview of the early development of commercial Enigma machines. I then present my rational for focusing on the commercial Enigma models with regard to available research sources and ethical considerations. After this I present an analysis of the working principle of the Enigma cipher system, providing detail on the operation of the electromechanical scrambler, and understanding on wheel wirings, its cipher period, polyalphabetic foundation, and the double stepping anomaly.

Chapter 3 Methodology

The third chapter explores the methodological approaches taken in the research project and is divided into three parts. The first part introduces the use of the Enigma cipher as a pseudo-random generator (PRG), starting with a taxonomy of Enigma PRG approaches, an analysis of sequences generated using the *no-input encryption* method, and a brief overview of additional techniques.

The second part introduces the Pd Enigma simulation, list_enigma, presenting an outline of the object in use as both a cipher and pseudo-random generator, with an explanation of its working principle in regard to the simulation of Enigma wheel wiring, rotation, and double stepping.

The third part presents a taxonomy of alphabetical pitch maps and duration maps used in this project. Several systems of pitch mapping are presented and analysed for their suitability in different situations and with reference to the works in the portfolio. Similarly, duration maps employed in the compositions are explored with reference to similar systems used by Boulez and Cage and in response to Koenig's theory of time-structures. Finally, Enigma cipher audification and synthesis methods are introduced with reference to the accompanying Pd patch examples and the works in the portfolio employing these techniques.

Analysis of Works

The fourth chapter presents a contextual and technical analysis that covers each of the eight musical works in the portfolio. The contents of each part is largely individual to the works in question but are generally composed of an initial background to the piece's conceptual development, with reference to inspirational works where applicable, followed by an explanation of its technical realisation. Moreover, where existing works are referenced as a muse or framework, a brief outline of the technical and/or aesthetic aspects of the piece is given with reference to research sources.

Conclusion

The final chapter summarises the achievements of the research project in response to the aims and contributions outlined in this introduction and considers the potential further academic research.

Chapter 2 Enigma Machines

The purpose of this chapter is to provide a historical account of the commercial line of Enigma ciphers machines and to detail the working principle of the D (A26) and K (A27) models. In addition, I will also consider my ethical and research considerations for focussing on this line commercial Enigma machines.

2.1 Notes on Enigma Research

Enigma History

As one of the main aims of my research project is the development of a working Enigma simulation in Pd, correctly identifying a specific model of Enigma machine on which to base this simulation is something I considered an important aspect of the project. However, presenting a brief yet concise history of Enigma is challenging. Whilst electrical engineer Authur Scherbius is often credited as developing Enigma, the inventors and engineers Hugo Koch, Willi Korn, and E. Richard Ritter have also played an important part in Enigma's development over time (Crypto Museum, 2023b). Patents for various models of Enigma machine have been filed under a several different names, with Crypto Museum (ibid) acknowledging that the "ownership of certain patents is sometimes a bit clouded". As such, it is difficult to attribute Enigma as a family of cipher machines to any single inventor or entity.

The many variations and models of Enigma further complicate the process of providing a simple overview of Enigma's history. As Hamer, Sullivan, and Weierud (1998, p. 1) acknowledge, "Enigma is too often considered to be a single machine notwithstanding the existence of both commercial and military models". The earliest commercial models, such as the *schreibende Enigma-Chiffriermachine* or 'writing Enigma cipher machine' (Wik, 2018, p. 84) employ a printing mechanism and, according to Kruh and Deavours (2008, p. 5), are "cryptographically unlike" the later models. The subsequent *glühlampenmaschine* or 'glow lamp machine' is the most widely documented and well-known variation, notably the commercial D (A26) and K

(A27) models and the various military variations on which these are based (Crypto Museum, 2023c). Just as it is difficult attributing the Enigma band to any single inventor or company, there is also much ambiguity around the identification and labelling of Enigma machines in an official capacity.

Commercial Enigma

My research project concerns the above mentioned commercial glow Enigma machines due to the detailed and empirical research sources available for this model. Information around the physical construction and working principle for this family of Enigma machines is widely documented and the descriptions in the research sources I have used show few discrepancies. Apart from the use of glow lamps, the most significant attributes of this family are that they employ a “scrambler” (Welchman, 2016, p. 39) with five cipher drums; a settable reflecting *Umkehrwalze*, three automatically rotating non-reflecting wheels, and a non-rotating and non-reflecting *Eintrittswalze*[1] (Hamer, Sullivan and Weierud, 1998). The wheel wiring data for these machines is also well documented (Kruh and Deavours, 2002; Crypto Museum, 2023e). The culmination of detailed information and description of these machines make them an ideal candidate on which to base a project investigating the musical applications of the Enigma cipher.

Ethical Considerations

Further to this decision, there are additional ethical considerations that concern the context of Enigma and its use by war organisations. Simons and Reuvers of Crypto Museum (2023d) note that during the Crypto Museum’s genesis “we were not always certain about whether or not we were allowed to reveal certain information. Some devices might still be ‘classified’, but there is no way to verify that, as the ‘list of classified equipment’ is classified itself”. They also note that “security agencies and

[1] According to Hamer, Sullivan, and Weierud (1998), the terms *Eintrittswalze* (ETW), wheel, and *Umkehrwalze* (UKW) are the original terms used by Bletchley Park, whilst reflector, rotor, and stator are the “more common American terms”. Furthermore, all three components have also been referred as drums (Gaj and Orlowski, 2003), cylinders and rollers in the English language sales pamphlet (Cryptologia, 2001). For continuity, the original Bletchley terms are used here, with drum used to describe all three more generally.

defence organisations are often not amused when cipher machines and other cryptographic material appears on the surplus market. In many cases the items should have been destroyed, but have accidentally (or intentionally) escaped demolition”. Crypto Museum states that they have developed a positive reputation with the security agencies in question, but acknowledge the ethical issues around publishing information about this sensitive topic, not wanting to “endanger any person, organisation or mission; civil or military”.

With Crypto Museum’s experience and observations in mind, it can be concluded that focusing on the commercial Enigma machine is the most ethically responsible approach for this research project, since it appears to have been developed, manufactured, and marketed independently of any involvement from security agencies.

2.2 Working Principle

This section provides an explanation of the electromechanical operation of the later commercial Enigma machines, their double stepping anomaly, and their function as a polyalphabetic cipher. The information provides a summary of several reliable sources on the working principle of the Enigma machine, primarily from Hamer, Sullivan, and Weierud (1998), Kruh and Deavours (2002), Welchman (2016), and Crypto Museum (2023). As Simons and Reuvers of Crypto Museum (2023a) state, “many attempts have been made to describe the working principle of the Enigma machine on the internet. Some of these are correct, and some are not.” This information is relevant for understanding how the cipher has been simulated in Pure Data and the cipher system application as a pseudo-random generator. How the simulation is employed in this way and within a musical context will be discussed in Chapters 3 and 4 respectively.

As this text focuses primarily on the commercial Enigma D (A26) and Enigma K (A27) machines, the simple designator *Enigma* will be used throughout the rest of the text.

Figure 1 Top view of Enigma K cipher machine (Crypto Museum, 2023e)

2.2.1 The ‘Scambler’

Figure 1 shows a photo of an Enigma cipher machine model K (A27) from the Crypto Museum website (2023e). When the operator presses a letter-key at the front of the machine, an electrical path is formed, connecting the internal battery with one of several lamps that light up the letters in the above the keyboard via an electromechanical ‘scrambler’. The scrambler (detailed in figure 2) is made up of five circular metal drums (wheels) with internal wiring that cross-couple their electrical terminals in an arbitrary fashion. These 26 terminals correspond to the 26 letters of the Latin alphabet.

Current passing between the letter-keys/lamps and the scrambler, first enters a fixed wheel called the *Eintrittswalze* (ETW) on the far right. The ETW is fixed at the right-hand side of the scrambler and distributes electrical current between the letter-keys/lamps and four remaining wheels via 26 electrical terminals. To the left of the ETW are four rotating wheels which slot onto a central spindle about which they rotate. The internal wiring of each of these central wheels differs from one another and

can be placed in any order as required by the operator, leading different electrical paths through the scrambler.

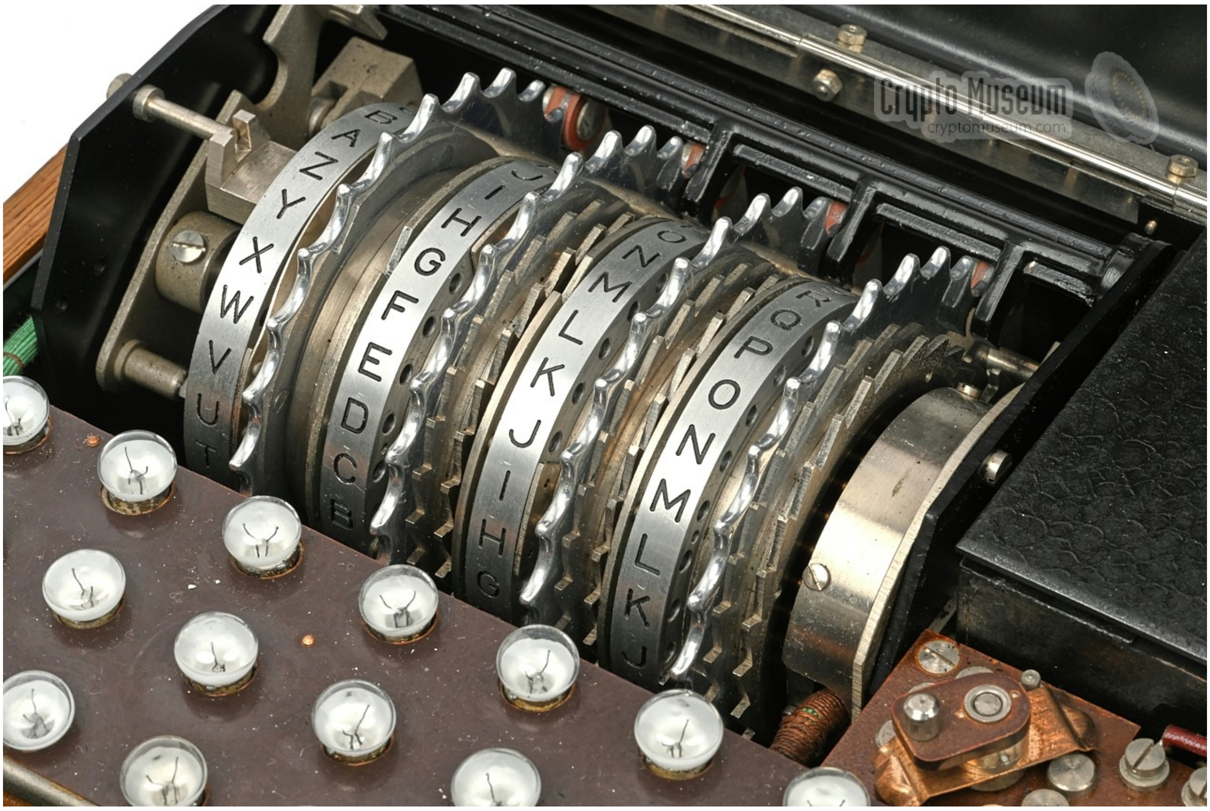

Figure 2 Enigma K wheels (Crypto Museum, 2023e)

The three middle wheels have 26 terminals on either side. The terminals of one wheel make physical contact with the terminals on the following wheel, forming an electrical path between them. The leftmost of the four wheels is the *Umkehrwalze* (UKW), which has terminals only on one side. The internal wiring of this wheel cross-couples its terminals in 13 pairs. Current flowing into one terminal of the UKW will appear at a different terminal on the same side and will therefore travel back through the previous four drums on a different route to which it came. Figure 3 shows the electrical signal path through the Enigma machine.

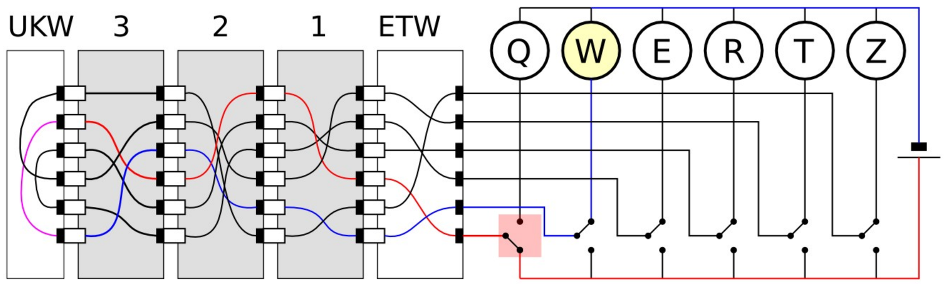

Figure 3 Flow diagram of Enigma cipher

Mechanical Behaviour

The cipher system of the Enigma is both a product of the flow of current through the scrambler and the mechanical movement of the wheels. This section explains the mechanical behaviour of the scrambler.

The four wheels to the right of the ETW are manually settable[2] by the operator and each have an alphabet ring[3] showing their current position. Moreover, the three wheels in-between the ETW and UKW automatically 'step' as plain text typed into the machine. The left-hand wheel (LHW) moves the quickest, stepping to its next position each time a letter-key is pressed. The middle wheel (MW) only steps when the LHW has stepped to specific *turnover* position[4]. The right-hand wheel (RHW) is the slowest of the three but works in much the same way as the MW, stepping in response to its neighbour. The mechanical algorithm of Enigma is therefore a product of the interaction of how each of the three wheels step in relation to the wheel's individual turnover positions, which is in turn affected by the wheel order on the spindle.

Double Stepping and Cipher Period

Since the RHW advances with each keypress, it therefore requires 26 keypresses for this wheel to move through all the letters on its alphabet ring. Additionally, since the

[2] The process of moving the wheels to a specific position prior to encipherment/decipherment is called the *Grundstellung* or 'initial setting' (Crypto Museum, 2023f).
[3] The alphabet ring is also settable and works in combination with the *turnover* position of the wheel.
[4] The *turnover* position is different for each of the three supplied wheels, so their order on the spindle also has an effect on how the cipher system works.

MW only moves once during the RHW's full revolution, one would expect this wheel to require 26 x 26 = 676 keypresses to do the same. In fact, a 'glitch' in the mechanics of the scrambler gives rise to an anomaly called *double stepping* (Crypto Museum, 2023f). Consequently, when the LHW moves in response to the MW stepping to its turnover position, the MW then makes an additional step on the next keypress. As a result, a whole revolution is skipped. In actuality, the MW only requires 25 x 26 = 650 keypresses to complete a full revolution. The LHW requires 26 x 25 x 26 = 16,900 keypresses to complete a full revolution. Therefore, the *cipher period* of the Enigma is 16,900 characters long.

Polyalphabetic Cipher

Without the rotation of the wheels, the electrical signal paths through the scrambler would remain through each successive keypress, resulting in a cipher system with just one cipher alphabet. In fact, the rotation of the wheels through the 16,900 positions mentioned above, leads to the same number of cipher alphabets, making the Enigma machine a *polyalphabetic cipher*. With three wheels to choose from and six different ways of ordering them on the spindle, the number of cipher alphabets available to an Enigma user are in the order of 16,900 x 6 = 101,400 alphabets.

Rotor Wirings

In order to consider how each wheel contributes to Enigma's cipher system, the wheel wiring given by Kruh and Deavours (2002) and Crypto Museum (2023e) may be consulted. The turnover and notch positions for each wheel, which is essential in the development of a simulation of an Enigma cipher:

Wheel	ABCDEFGHIJKLMNOPQRSTUVWXYZ	Notch	Turnover
ETW	QWERTZUIOASDFGHJKPYXCVBNML		
I	LPGSZMHAEOQKVXRFYBUTNICJDW	G	Y
II	SLVGBTFXJQOHEWIRZYAMKPCNDU	M	E
III	CJGDPSHKTURAWZXFMYNQOBVLIE	V	N
UKW	IMETCGFRAYSQBZXWLHKDVUPOJN		

The table demonstrates that each wheel is, in effect, a simple substitution cipher that 'shuffles' the order of the alphabet. The effect of wheel rotation, however, also results in an 'offset' of each of these substitution cipher in much the same way that a Caesar cipher 'shifts' the alphabet up or down a certain number of places. This aspect of the Enigma's cipher system and its implementation in the project's Enigma simulation will be discussed in more detail in Chapter 3.

2.3 Summary

The field of Enigma research spans a wealth of resources, from the origins of the commercial machines and their working principle to in-depth analyses of codebreaking in the context of WWII. Simons and Reuvers of Crypto Museum improved the accessibility of the technical information, presenting an up-to-date collation of disparate sources with corrections on previously misunderstood or unknown parts of Enigma history. In the context of music composition, this technical understanding is a requirement for the accurate simulation of an Enigma cipher in Pd. Moreover, this understanding also plays an important part in discovering ways in which the Enigma cipher may be used as a pseudo-random generator.

Chapter 3 Methodology

The purpose of this chapter is to describe the methodology of the research project and how the Enigma cipher can be used as a pseudo-random generator in music composition, as opposed to its intended use as a cipher device.

The methodology is delivered in three parts. Section 3.2 presents a brief taxonomy of approaches for using the Enigma cipher as a pseudo-random generator (PRG). Section 3.3 introduces the list_enigma Pd object and the accompanying patch examples. Finally, Section 3.4 presents a simple taxonomy of alphabetical parameter maps for use with the Enigma PRG.

3.1 Enigma Pseudo-Random Generator (PRG)

3.1.1 Taxonomy of Enigma PRG Approaches

This subchapter presents a taxonomy of three possible methods for employing the Enigma cipher as a pseudo-random generator (PRG). The first, *same letter stepping*, is the simplest of the three and is a way of generating random-like sequences identified by Gordon Welchman (2016, p. 38). The second, *no-input encryption*, extends the limited cipher period of the Enigma and therefore increases the possible length of sequences. The third, *scrambler monitoring*, involves the internal state of the cipher system to extract additional data.

Same Letter Stepping

This is the simplest method of generating pseudo-random patterns using the Enigma. As Welchman observes (ibid), if the “letter P is pressed repeatedly, the encoded letters lit up on the lampboard will appear to be in a random sequence”. This action causes the cipher to step sequentially through all of the possible cipher alphabets for a given wheel order and UKW position. The resulting sequence of ciphertext is 16900 characters long and contains all the encipherments of the plain text input in order of linear progression through the wheel positions.

A caveat of the *same letter stepping* method lies in an anomaly of the Enigma's cipher system. Since no letter can be enciphered as itself, the resolution of the PRG is reduced to 25 characters – every character of the alphabet except the one being used to step the cipher. We could also consider the sequence length a limitation of this method. Below is an example of a sequence generated in this way by repeating the letter A:

```
R K H L K U D T H S Y V I C W N Z W W D M W K G E O
```

No-Input Encryption

The name *no-input encryption* is derived from the musical practice of feeding the output of a mixing console to one of its own inputs to produce a sustained tone, commonly referred to as *no-input mixing*. This is slightly more complicated in operation to same letter stepping. The operator first presses an initial letter-key, which, along with the UKW position and wheel order, acts as a 'seed' for the pattern. The next letter in the sequence would be the enciphered character from the previous keypress. An advantage to this method over same letter stepping is that, since each enciphered character is different, all 26 letters may occur, increasing the resolution of the PRG. Consequently, this method overcomes the cryptographic anomaly previously noted. Below is an example of a no-input encryption sequence starting with the letter A:

```
R S N G W S P W L O V A I E D R S J U M A W F W M B
```

The second and most notable feature of this method is its ability to produce sequences that are longer than the cipher period of the Enigma. The length of a given pattern is dependent on whether, once all three wheels have returned to their original starting position, the following letter is the same as the first letter pressed. If paired, the pattern starts over from the beginning, leading to a pattern length matching the Enigma’s cipher period. If the next letter differs from the first letter pressed, then the sequence continues for another 16900 characters until finally the letters match. This phenomenon is discussed further in section 3.3.

Scrambler Monitoring

The name *scrambler monitoring* combines Gordon Welchman’s description of the Enigma machine’s “scrambler” (Welchman, 2016, p. 38) with Daniel Pallocks’ use of the label “monitor” on his online Enigma simulator to display the internal states of the cipher (Pallocks, 2016). This method monitors the virtual electrical signal path of the cipher and takes a reading from the output of each wheel. This produces nine cipher characters, each representing a more secure level of encipherment/decipherment in passage through the cipher path. This method also overcomes the first issue of the same-letter anomaly mentioned earlier. Observing the internal states of the cipher shows that the input character may appear at all wheels *except* when leaving the ETW as the output character.

Another immediate advantage of scrambler monitoring is that we now have nine parallel pseudo-random patterns available where previous there was only a single cipher. We can also read these nine internal states serially rather than in parallel. The potential in applying *scrambler monitoring* to *no-input encryption* will be analysed in the following section.

3.1.2 Analysis of No-Input Encryption Sequences

Sequence Length and Number

During the research project, the Enigma cipher simulation was employed to calculate the total number of pseudo-random sequences generated through *no-input encryption*. This revealed sequence lengths between one and 25 cipher periods, the largest being 422,500 characters long. A total of 614 *no-input encryption* sequences were obtained for all possible combinations of wheel orders, umkehrwalze positions, and parallel cipher paths.

Two text files have been included as examples of the shortest and longest *no-input encryption* sequences. These are located in the folder `/no-input_encryption`. The sequences held in the text files `no-input_16900-characters.txt` and `no-input_422500-characters.txt` may be recreated using the following Enigma cipher settings:

	16900 characters:	422,500 characters:
Wheel order:	`1-2-3`	`1-2-3`
Wheel positions:	`C-A-A-A`	`D-A-A-A`
Input character:	`Z`	`A`

Whilst a sequence of 16,900 characters is perfectly adequate and perceivably random when being used to order musical events, the sequence becomes perceivably periodic at audio rate. The use of *no-input encryption* sequences for this purpose is explore in more detail later in this chapter.

Identifying Patterns

When the internal states of the cipher are monitored during *no-input encryption*, short patterns are observable in the resulting sequence. These patterns occur as the middle and left-hand wheels step infrequently compared to the right-hand wheel. Whilst the right-hand wheel is stepping through its 26 positions, the other two wheels remain static. Therefore, the electric signal path through these wheels (and the static UKW) remains unchanging. Characters entering the same point in this unchanging path from the right-hand wheel will be enciphered as the same sequence of five characters. When the middle or left-hand wheels step, a new selection of 26 possible routes will be created, and a different set of patterns are likely to occur during the following 26 key-presses.

	Position in Scrambler									
	INPUT	ETW	W1	W2	W3	UKW	W3	W2	W1	OUTPUT
Wheel Positions	**Encrypted Character**									
AAZ	A	J	F	T	Q	L	X	H	C	R
AAA	R	D	S	A	C	E	Z	Q	K	S
AAB	S	K		Q	M	B	V	C	Y	N
AAC	N	X	W	K	R	H	G	D	P	G
AAD	G	Q	Y	P	F	G	C	W	E	W
AAE	W	B	M	J	U	V	W	N	O	S
AAF	S	K	F	S	N	Z	N	X	W	P
AAG	P	R	J	G	H	R	K	U	H	W
AAH	W	B	E	N	Z	N	S	A	G	L
AAI	L	Z	A	A	C	E	Z	Q	Q	O
AAJ	O	I	B	A	C	E	Z	Q	E	
AAK			M	V	B	M	Q	J	T	A
AAL	A	J	N	V	B	M	Q	J	S	I
AAM	I	H	T	X	L	Q	T	F	O	E
AAN	E	C	F	A	C	E	Z	Q	Y	D
AAO	D	L	W	J	U	V	W	N	R	R
AAP	R	D	U	T	Q	L	X	H	Z	S
AAQ	S	K	L	P	F	G	C	W	F	J
AAR	J	P	H	Z	E	C	A	S	X	U
AAS	U	G	D	H	K	S	F	G	Q	M
AAT	M	Y	B	J	U	V	W	N	C	A
AAU	A	J	S			T	I	O	V	W
AAV	W	B	C	X	L	Q	T	F	H	F
AAW	F	M	E	J	U	V	W	N	X	W
AAX	W	B	D	F	S	K	H	L	V	M
AAY	M	Y	C	B	J	Y	R	P	U	B

Identified Patterns:					Occurances:
A	C	E	Z	Q	4
J	U	V	W	N	3
P	F	G	C	W	2
T	Q	L	X	H	3
V	B	M	Q	J	2
X	L	Q	T	F	2

Figure 4 Spreadsheet of *no-input encryption* patterns

Figure 4 shows the spreadsheet containing patterns identified, colour-coded, and listed from a *no-input encryption* sequence. These patterns are the basis for the composition *Ringstellung* discussed in Chapter 4.

3.1.3 Additional Techniques

Probability Distributions

Pseudo-random sequences generated by the Enigma cipher through both *same letter stepping* and *no-input encryption* are of a uniform distribution, with all letters equally likely to occur (with the exception of the input letter in *same letter stepping*). Using *scrambler monitorin*g and summing the internal states of the cipher, a response closer to a normal distribution is produced. Patch example

`B4.enigma.distribution.pd` presents an analysis of the two distribution types. A screenshot the resulting graphs are given in figure 5. Note that summing the scrambler states in this way also increases the resolution of the pseudo-random sequence from 26 characters to 250 characters.

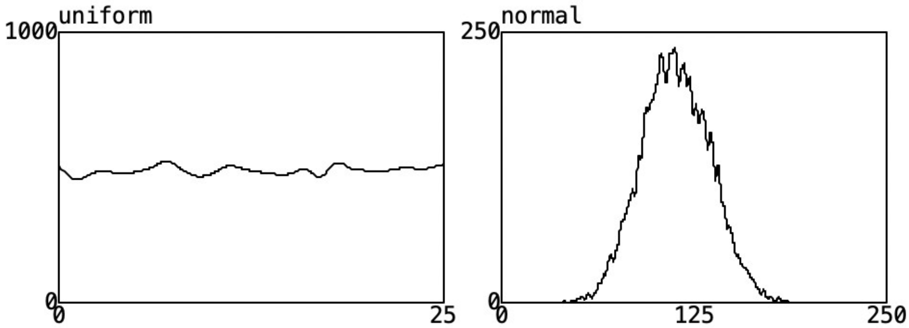

Figure 5 Enigma PRG probability distributions

Random Walks

Another way of increasing resolution is by implementing a random walk using the Enigma PRG, summing the current value with the previous value, and coercing the resulting sequence into a predetermined range. Example patch `B5.enigma.random.walk.pd` is a variation Pd's own random walk example but using the Enigma PRG.

Interpolation

Finally, we can interpolate between values generated by the Enigma PRG to create smooth transitions rather than discrete steps, a form of random generation that is more closely related to the *fluctuating random voltages* in the Buchla *Source of Uncertainty* than the discrete pseudo-random sequences described up till now (Strange, 2022, p. 84). Patch example `B6.enigma.interpolation.pd` demonstrates the effect of different approaches and combinations of linear interpolation and lowpass filtering as shown in fig. n

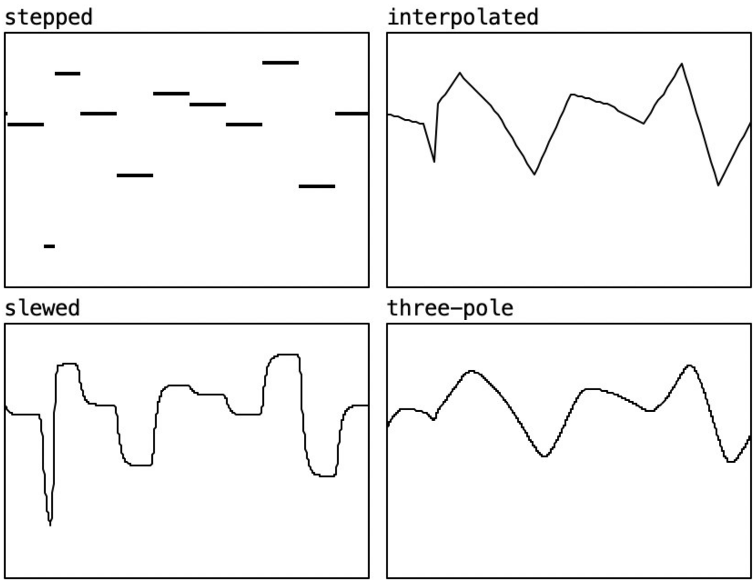

Figure 6 Enigma PRG interpolation

The use of the Enigma PRG in Pd will be explored Chapter 3 and its use in music composition in Chapter 4*.* In general, the pseudo-random techniques employed in the works in the portfolio primarily use *no-input encryption*, augmented by *scrambler monito*ring to identify patterns or control multiple parameters with at the same time.

3.2 Enigma Cipher Simulation: list_enigma

The purpose of this section is to introduce the list enigma Pd object and the accompanying patch examples, exploring various methods of parameter mapping and sound generation using list_enigma and its signal-rate cousin, signal_enigma~.

Included Files

The folder `/patch_examples` contains a collection of Pd files[5]. The list_enigma abstraction, a series of example patches built around list_enigma that demonstrate its use as both a cipher and a pseudo-random generator, and several additional abstraction objects developed during this research project to support the example patches and use of list_enigma in the creation of the works in the portfolio.

[5] The audio file 321a_a.wav is also included in this folder, which is an audification of a short Enigma PRG sequence, and is used in some of the example patches.

The patch examples are roughly divided into categories alphabetically. `A` examples explore ciphers, with `A1.enigma.cipher.pd` employing list_enigma as a typical Enigma machine simulator, which is extended to include printing of cipher text to the Pd window in example patch `A2.enigma.printing.pd`. `B` patch examples introduce the use of list_enigma as a pseudo-random generator, exploring the Enigma PRG taxonomies earlier in this section, whilst `C` patch examples explore alphabetical pitch maps for 12-TET, 13-TET, and Bohlen's 13-Step scale. Patch examples `D` combine these pitch maps with the Enigma PRG patches and incorporate the duration maps outlined later in this chapter. `E` example patches explore Enigma cipher audification and synthesis, first using the signal_enigma~ and then via wavetable synthesis of Enigma PRG sequences contained in audio files. Finally, the `F` patch examples present some non-PRG ideas with the list_enigma object intended for further exploration.

3.2.1 Overview and Working Principle

This part provides a basic description of how the list_enigma object appears in Pd, a brief explanation of how it is interacted with, and how the object simulates the Enigma cipher. This may be useful for those who are unfamiliar with Pd and during examples of parameter maps later in the chapter.

Figure 7 shows a basic Pd patch built around the list_enigma object. The object has a single inlet that accepts both the plain text character to be enciphered and the initialization settings for the cipher. The plain text character must be a lowercase symbol between `a` and `z`, and the initialisation messages must be formatted as above. Wheel order, ring setting, and wheel position may be initialised independently or at the same time (as pictured above) and in any order.

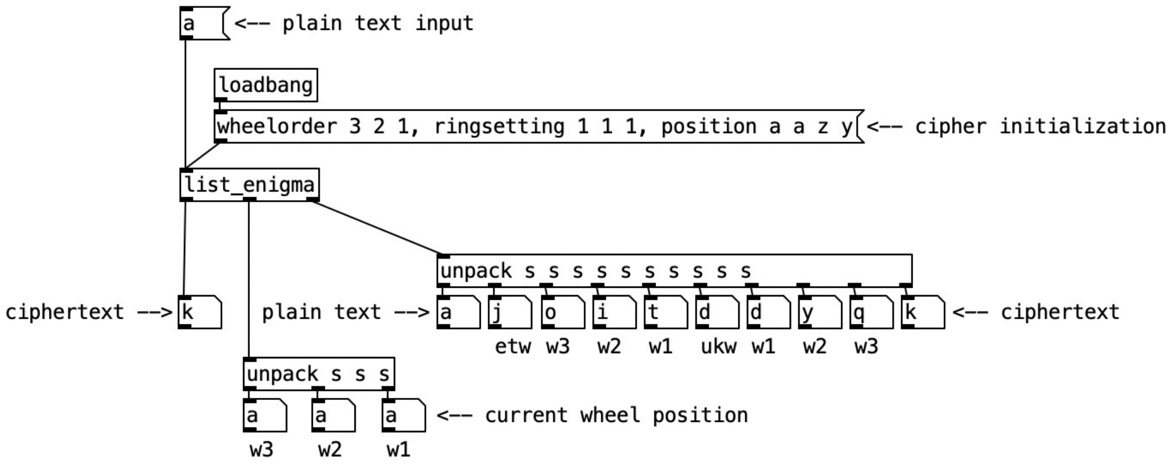

Figure 7 Basic Pd patch using list_enigma object

From left to right, the outlets of the list_enigma object correspond to: The ciphertext output; a *list* containing three symbols which correspond to the current wheel positions of the cipher; and another *list* containing 10 symbols which allow the user to monitor the internal states of all wheels in the cipher.

Working Principle

Figure 8 shows the internal patch of the list_enigma object. The patch contained in the parent window is responsible for unpacking the plain text and initialization arguments and routing these to the ciphers and counters sub-patches (which themselves call upon the turnover and arrays sub-patches), and receiving and packing cipher text, internal states, and stepped wheel positions into the object's outlet.

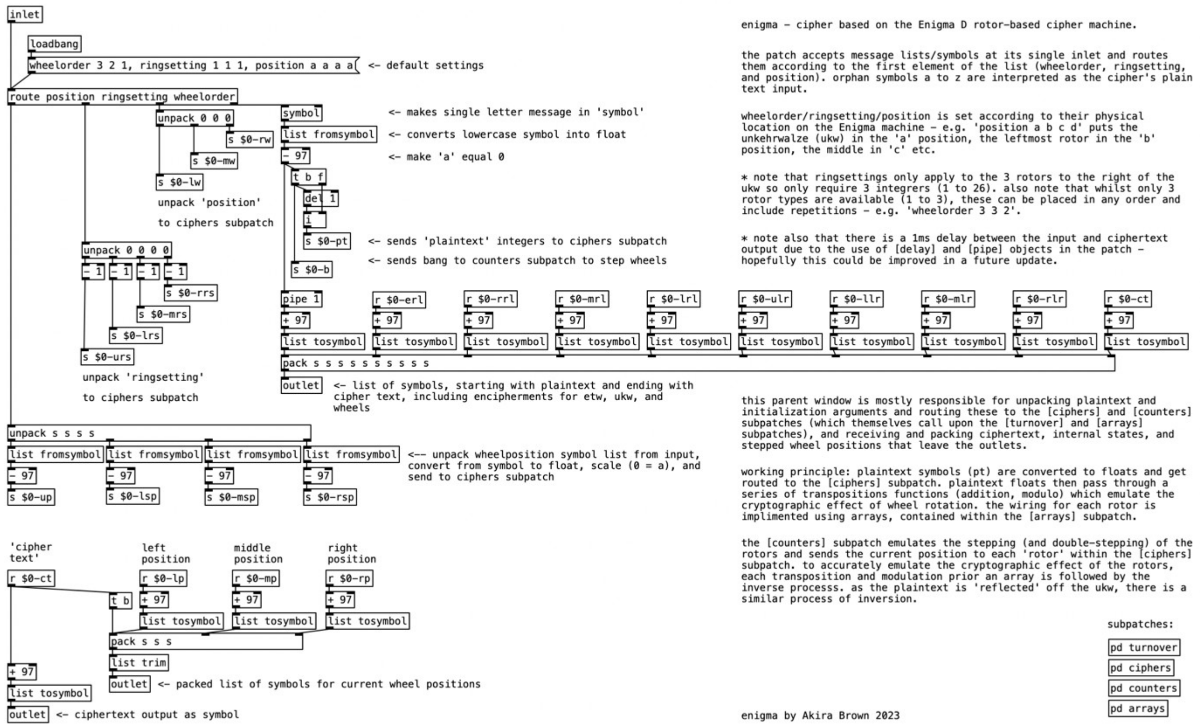

Figure 8 Parent window of list_enigma object

The plain text character input is converted from a symbol to a number and routed to the ciphers sub-patch. These numbers then pass through a series of mathematical operations (addition, subtraction, modulo) which simulate the cryptographic effect of wheel rotation on the electrical signal path. The arrays sub-patch holds several arrays, each containing the wheel wiring transfer functions for both directions of travel through the Enigma cipher. The counters sub-patch works in tandem with the ciphers sub-patch to enact the stepping and double stepping of the wheels.

Simulating Wheel Rotation

To accurately simulate the cryptographic effect of the wheels as they turn, the lookup table for each wheel effectively needs to 'rotate'. To achieve this, the value used to address the array is offset by the current position of that wheel – i.e., for a rotor in its C position, an offset of +3 would be added to the value addressing the lookup table. The wheel position value is then subtracted from the value produced by the lookup table.

Figure 9 shows an example of a simple four-character cipher that is rotated virtually by applying positive and negative offsets before and after the lookup table, identified by the unchanging transfer function in grey.

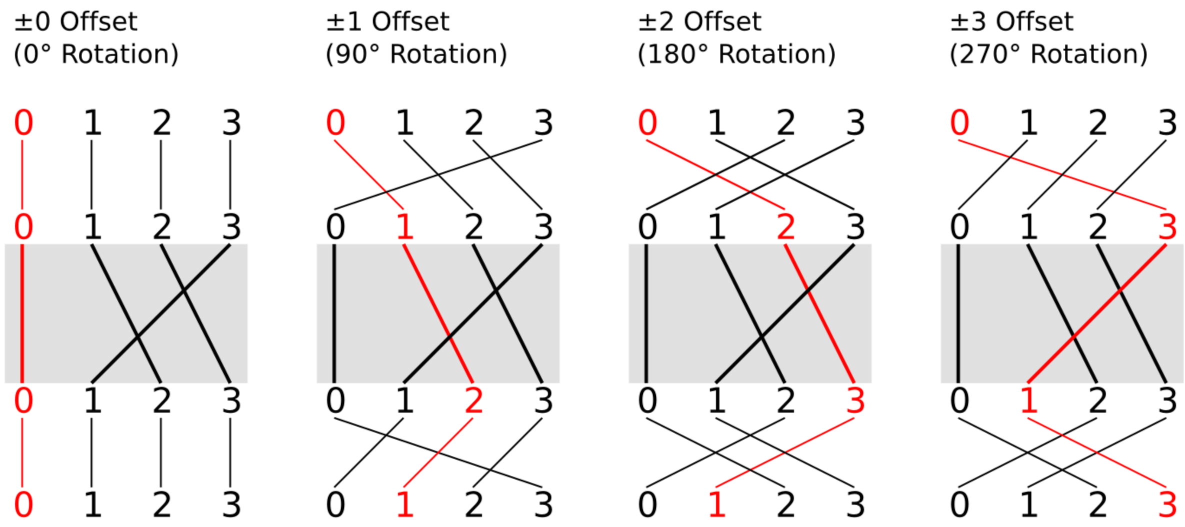

Figure 9 Cryptographic effect of wheel rotation

When simulating the return journey through the Enigma cipher, a second set of arrays are employed which contain the inverse transfer functions of the wheel wirings.

Rotation and Double Stepping

The following is a description of how the counters of the list_enigma object simulate the mechanical algorithm in the Enigma machine that causes the wheels to step in a specific way.

Three counters are connected in series. When the first counter reaches a specific value, the second counter is incremented. When the second counter reaches a specific value, the third counter is incremented. These specific values are the *turnover* positions and are unique to each wheel, as outlined in Chapter 2. Thus, the behaviour of the counters is in part determined by the wheel order.

The counters in the list_enigma object also implement the double-stepping anomaly of the Enigma cipher. When either the first or second counter have incremented to their respective turnover values, they each produce a logical `1`. These two binary values are compared using an AND gate. When both counters are at their turnover values, the logical `1` from this gate increments the third counter and remains high whilst the counters are left in this state. The output of the gate is also connected to a spigot upstream of the second counter, which will cause the second counter to increment upon the next bang along with the first counter. Once the first and second counters have incremented once more, neither the third nor the second counter are at their turnover values. The AND gate returns to a logical `0` and the spigot closes, completing the double stepping procedure.

3.2.2 Enigma PRG with list_enigma

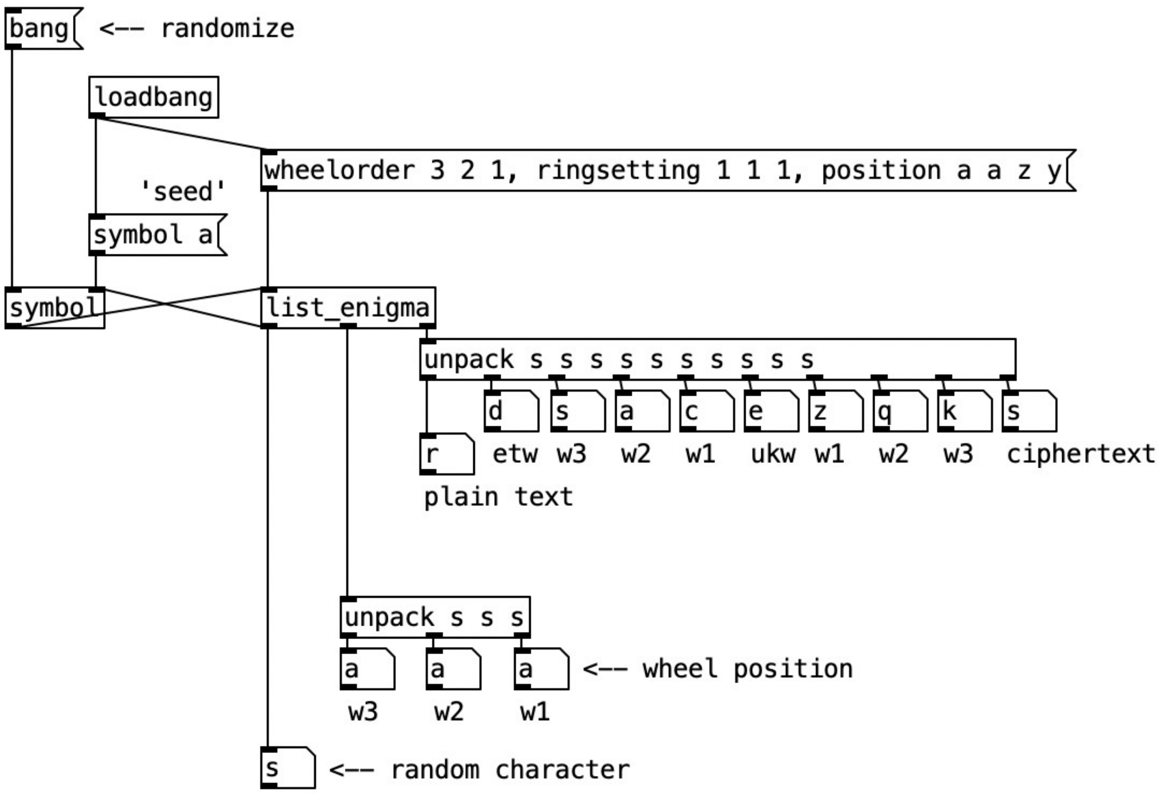

Figure 10 No-input encryption Pd patch

Figure 10 shows the basic patch configuration for implementing list_enigma as a pseudo-random generator using the *no-input encryption* method, as given in the patch example `B3.enigma.no-input.encryption.pd`. The symbol a is first stored in the symbol object on initialization. Upon receipt of a ‘bang’, this symbol is then enciphered by the list_enigma object, whose output is then returned to the symbol object and held, ready for the next encipherment.

This basic configuration serves as the foundation on which the majority of the works in the portfolio are based. Furthermore, the works in the portfolio make use of interpolation, as demonstrated in `B6.enigma.interpolation.pd`; sine-tone mixtures, as in `D3.enigma.mixture.pd`; audified noise bands, as in E4.enigma.tenney.pd; and drones, based on `E6.enigma.drone.pd`.

The following section examines the various methods of parameter mapping using the pseudo-random sequences originating from the plain text input, cipher text output, and internal states of the list_enigma object in this configuration.

3.3 Taxonomy of Parameter Maps

During the development of the musical works for this research project, several methods of parameter mapping became standard practice and acted as a starting point for each work. In general, parameter mapping of amplitude is implemented in a simple process of scaling values either linearly, exponentially, or logarithmically, in order to achieve the desired dynamic response of an envelope. Similarly, the mapping of timbre is often somewhat arbitrary and ‘to taste’, rather than requiring a specific parameter mapping system.

This section details the pitch and duration maps which are shared between the pieces in the portfolio. First, a brief taxonomy of these pitch maps will be given, followed by an outline of duration maps in relation to Boulez, Koenig, and Cage. During the project’s development, a number of additional abstractions were developed for use with the list_enigma object, so this section also details the *easley* and *bohlen* objects included in the example patches and used in the compositions in the portfolio.

3.3.1 Pitch Maps

There are many possible ways of mapping the alphabet to pitch, and besides the mostly incomplete systems traditionally used in the field of musical cryptography, there is little documentation of alphabetical pitch maps exploring non-standard tuning systems. This section presents several approaches. As I will be discussing both scales built from harmonic ratios and those created by equally dividing larger intervals, the terms *just intonation* and *equal temperament* will be used (as opposed to the *equal divisions of the octave* or 'EDO').

Chromatic Pitch Map

This method employs a 1:1 linear map of the 26 letters of the alphabet to notes of the 12-TET chromatic scale. This is particularly suitable in instances where Enigma PRG sequences are being mapped to traditional Western acoustic instruments or MIDI equipment. Pitches are mapped relatively rather than absolutely – i.e., a letter may address any pitch so long as the following letter addresses the pitch a semitone above. The total span of pitches is 2 octaves and 2 semitones, so the 26 letters to not fit neatly into octave ranges. This results in the first 2 pitches of the sequence occurring more frequently than the remaining 24.

My pieces *Ringstellung* and *Durations 1a* and *1b* make use chromatic pitch mapping.

13-TET

This method maps the alphabet to two octaves, each with a ratio equal to $\sqrt[13]{2}$. The 13-TET scale is not derived from harmonic intervals, like the justly intoned 12-tone and related 12-TET scale. This method is particularly suitable for exploration within Pd since the environment allows for fine control over the frequency of sound generators. Similarly, the 13-TET scale is a good match for the alphabet and map evenly over 2-octaves. Conversely, it is not as suitable to existing acoustic instrumentation, and support for this tuning system in external MIDI equipment is unknown. Instances of 13-TET scales used in music are limited; Easley Blackwood's *Twelve Microtonal Etudes for Electronic Music Media, Op. 28* from 1980 being a rare example (Cedille, 2023). In order to explore the 13-TET scale in Pd, I created the *easley* object in homage to Easley's work in this field. Like Pd's native *mtof* object for 12-TET scaling, *easley* is configured to respond to MIDI note numbers, and in accordance with a personal standard, MIDI note 69 is made equal to 440Hz.

Study 13 is a contemplation on the 13-TET pitch map and its application to parallel scrambler monitoring to produce mixtures of superimposed sine-tones.

Bohlen-Pierce

This method maps the alphabet over three octaves according to the Heinz Bohlen's *13-Step scale*. Unlike the chromatic 12-TET and 13-TET systems, Bohlen's scale is a *non-octave repeating scale*, built up from odd harmonics, and repeating a 3:1 'tritave' interval instead of the 2:1 octave. Bohlen's scale can also be built using just intonation or for equal temperament, resulting in 13 equal intervals with a ratio equal to $\sqrt[13]{3}$.

The *13-Step scale* was first outlined in the early 1970's by Heinz Bohlen as a "derivation of a scale from the combination tones, assuming only odd-number harmonics" (Huygens-Fokker Foundation, 2013). Bohlen's early manuscript describes a just-intoned scale made up of the ratios `1/1, 27/25, 25/21, 9/7, 7/5, 75/49, 5/3, 9/5, 49/25, 15/7, 7/3, 63/25, 25/9, and 3/1`. The scale was also discovered independently by Roberts and Mathews (1984), becoming more widely known as the Bohlen-Pierce scale after the publication *Theoretical and experimental explorations of the Bohlen-Pierce scale* (Mathews et al., 1988). This later paper outlined an equally-tempered version of Bohlen's scale, employing 13 intervals with a $3^{1/13}$ frequency ratio.

As with the 13-TET method above, this is most suitable to exploration within Pd due to ulack of support for microtonal tuning over MIDI, however, for devices with support for ¼-tone tuning I have identified that a Bohlen-Pierce scale may be constructed using minor thirds. Unlike the 13-TET scale, there is some support for the Bohlen-Pierce scale in acoustic instruments, such as the Bohlen-Pierce clarinet (International Clarinet Association, 2019). Interestingly, a notation system has been devised that distances Bohlen-Pierce notation from the standard 12-tone notation, using the letters N to Z as a pitch reference. Figure 11 shows the pitch names and tritave ranges in this form of Bohlen-Pierce.

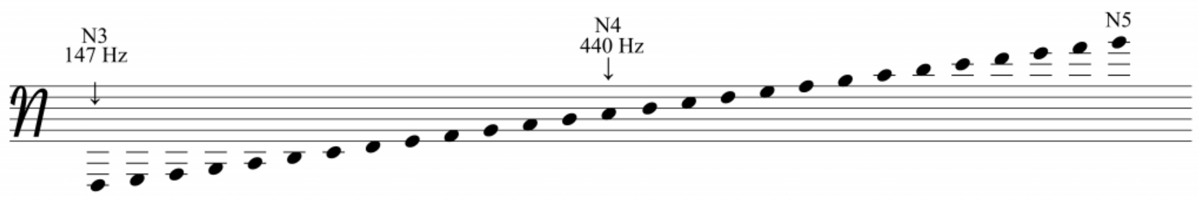

Figure 11 Bohlen-Pierce notation (International Clarinet Association, 2019)

In order to explore the Bohlen-Pierce scale for this project, I created the *bohlen* object, which pays homage to Heinz Bohlen's original *13-Step scale* by implementing it in just intonation, calculated from harmonic ratios. As with the *mtof* and *easley* objects, MIDI note number 69 = 440 Hz. This is also to align with the aforementioned Bohlen-Pierce notation, whereby the pitch N is "anchored with the common tuning note of A in the traditional system" (International Clarinet Association, 2019).

Enigma Krell 1 and *Enigma Krell 2* make prominent used of the Bohlen-Pierce scale.

Visual Comparison

Fig. n presents a table showing 26 pitches in 12-TET, 13-TET, and the equally-tempered Bohlen-Pierce scale. A 26-TET scale is included for reference, although this is not featured in the portfolio works. As with the previously mentioned Bohlen-Pierce notation, all scales are centred with their 13th interval located at 440 Hz / MIDI note 69.

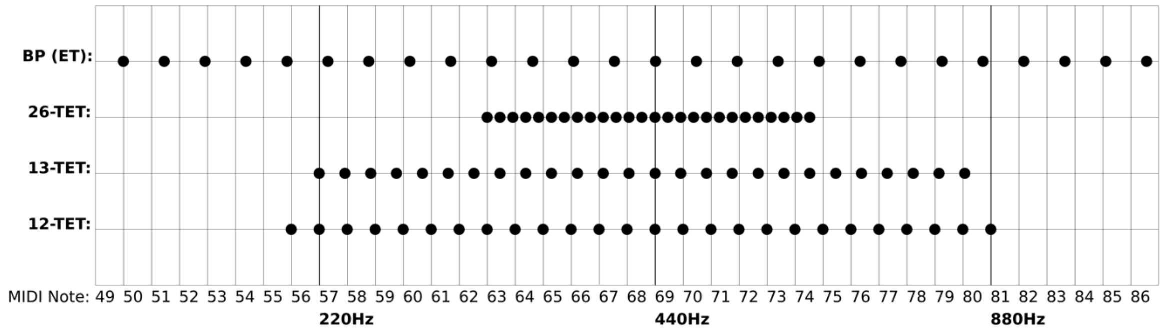

Figure 12 Comparison of pitch map ranges

3.3.2 Duration Maps

As with pitch maps, it is helpful to design a system of alphabetic mapping to act as a framework for determining duration. These can be mapped linearly in a 1:1 ordinal relationship with the alphabet, or arbitrarily in a non-linear fashion, such as in Messiaen's *communicable language* (Shenton, 2016, p. 79). For this project I have taken the first approach as this offered a more robust mathematical foundation. As a starting point, the alphabet is first replaced with an ordinal numbering sequence between 1 and 26:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

The most straightforward approach from here is then to multiply a minimum unit of duration by the letter's numeric substitute. This is the same approach taken by Pierre Boulez in his *Structures 1a* from 1953. Figure 13 shows the ordering of relative durations, in which "the chosen basic unit (demisemiquaver) is multiplied by from 1 to 12, and arranged in an increasing mathematical series" (Ligeti, 1957, p. 39).

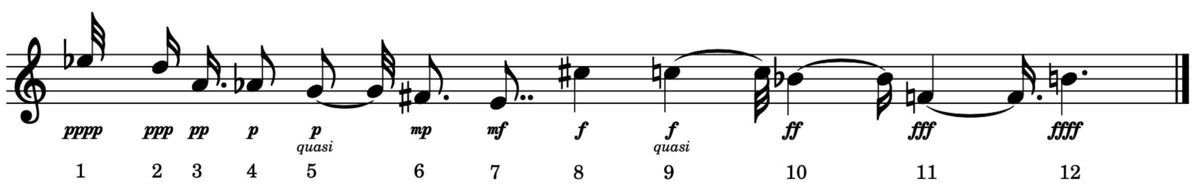

Figure 13 Note durations in Boulez's *Structures 1a* (Ligeti, 1957)

To make this duration map in Pd, I have created the *every* object – a type of counter designed to work in tandem with the Enigma PRG, shown in figure 14. It works by counting the number of ‘bangs’ received at its left inlet and counting up to the value given at its right inlet, at which point a bang is sent from its outlet. When configured to work alongside the *list_enigma* object, the cipher text output determines the number of bangs required till the next plain text value is received and enciphered. In Enigma terms, this is equivalent to the operator, upon seeing the letter J light up, being required to press the next key 10 times before the machine will step to the next wheel position.

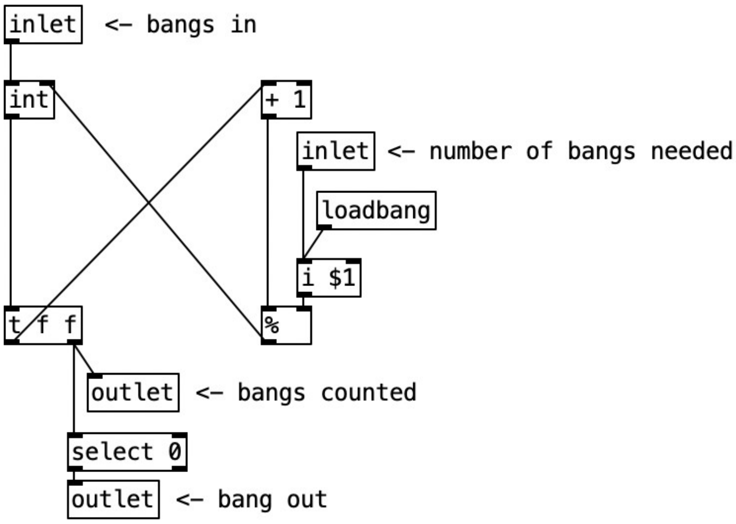

Figure 14 Pd *every* object

Notes & Rests

When compared to the largest duration value in Boulez’s tone row, the Enigma duration map described above clearly produces a duration that is considerably longer. For example, with a minimum duration of 50ms (letter A), the longest duration would

be 1300ms (letter B). It should also be noted that this method does not accommodate rests, so any moments of silence between notes would be the effect of amplitude envelopes rather than articulated by the sequency. Moreover, when played by instruments which produce sustained notes (e.g., organ, clarinet), these individual durations essentially combine into a single unity.

Koenig provides a framework for analysing the qualities of time-structures when systematised in this way:

“If however we do not regard the time-values as durations but as distances, we obtain a sequence of entry points at which sounds of any duration can commence. If the durations are shorter than the distances (in order to achieve pointillist effects), they result in duration + rest. Pointillist time sequences also result from durations filling the entire distance, if sounds and rests enjoy equal rights and alternate” (Koenig, 1963, p. 14).

Koenig views these time-structures as representable in three ways: *Pointillist,* where the “time-points are clearly marked” – such as through the use of staccato or percussive amplitude envelopes; *plane-like*, where “the distances between the time-points are occupied by “durations”” and “direct connections” – as in the unity of durations mentioned above; *articulated in planes*, whereby “some durations are replaced by corresponding rests”; and a *transition* state moving between these three (Koenig, 1963, p. 15).

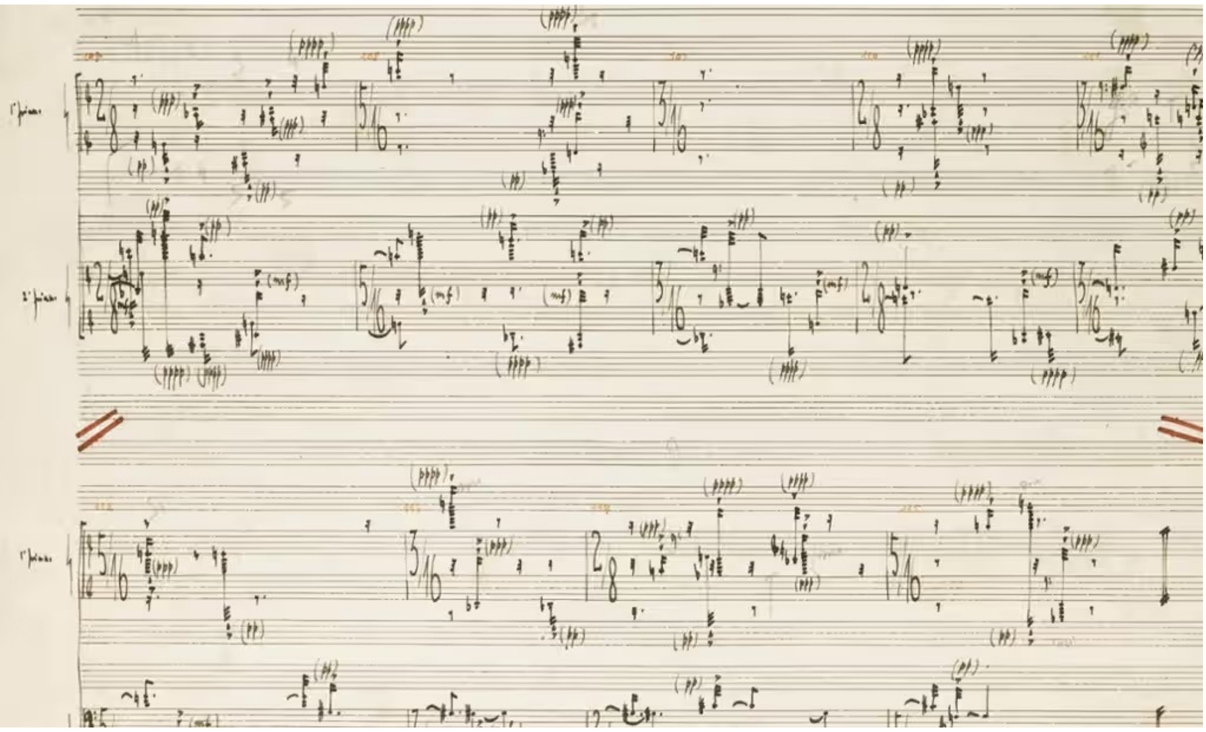

Figure 15 Boulez’s handwritten score for *Structures 1a* (Service, 2015)

There is clearly an important interplay between note durations and how accurately the amplitude envelope of an instrument can represent these durations. In the case of Boulez's *Structures 1a*, the percussive envelope of the piano innately imbue rest-like property that may be absent in the note duration as scored and may be considered *pointillistic* when the note's natural decay diminishes before the end of the notated duration. As Boulez's handwritten score shows (Service, 2015), duration is articulated in planes through the use of both notes and rests.

Koenig's idea of notes and rests "enjoying equal rights" is exemplified in Cage's composition process behind *Music of changes*. In his book *Silence*, Cage describes a process of selecting between two groups of 32 sounds and silences using coin oracle of the *I Ching* (Cage, 1968, p. 26). The arrangements of sounds and silences in these charts correspond to a chart identifying the 64 hexagrams of the *I Ching* in Wilhelm's 1951 translation (Jensen, 2009, p. 99) shown in figure 16. It is this 'splitting' of the source data into equal parts to define notes and rests that my duration map follows, with the first half of the alphabet (A to M) producing rests and the second half (N to Z) producing tones.

TRIGRAMS UPPER ▶ LOWER ▼	Ch'ien ☰	Chên ☳	K'an ☵	Kên ☶	K'un ☷	Sun ☴	Li ☲	Tui ☱
Ch'ien ☰	1	34	5	26	11	9	14	43
Chên ☳	25	51	3	27	24	42	21	17
K'an ☵	6	40	29	4	7	59	64	47
Kên ☶	33	62	39	52	15	53	56	31
K'un ☷	12	16	8	23	2	20	35	45
Sun ☴	44	32	48	18	46	57	50	28
Li ☲	13	55	63	22	36	37	30	49
Tui ☱	10	54	60	41	19	61	38	58

Figure 16 Wilhelm hexagram chart (Jensen, 2009, p. 99)

This duration map is used in the portfolio works *Durations 1a* and *1b*, *Study 13*, *Krell Patch 2, ENIGMA-N*, and *Ciphertext for Susan Merrick*.

3.4 Enigma Audification and Synthesis

Thus far, we have only discussed Enigma PRG sequences mapped to musical parameters. It is also possible to sonify text-based data through a process called *audification*[6]. For pseudo-random sequences, this also affects the apparent randomness of the data since the speed at which the data is read is now in the magnitude of 10x to 1000x faster. The sequence length therefore has a noticeable effect on whether or not the resulting waveform is perceivable as a periodic tone or as aperiodic noise.

The following section explores several methods of sound synthesis using Enigma PRG sequence as employed in the portfolio of works and with reference to implementations given in the Pd patch examples accompanying this research project.

Audification

The patch example `E1.enigma.audification.pd` presents a simulation of the Enigma cipher in Pd using signal-rate objects. This allows for the real-time audification of cipher data, however, in order to accurately simulate the scrambler rotation behaviour, a block size of `1` is required. A disadvantage of this approach is a higher CPU load. In practice it is far more economical to place an entire Enigma PRG sequence in an array and to produce noise and periodic waveforms using wavetable synthesis.

Three audio files are included as example of audification periodicity. These are located in the folder `/no-input_encryption`.

When played back at a sample-rate of 44.1kHz, the 16,900-character audification has a duration of ~0.38 seconds and exhibits audible periodicity when looped. The 33800-character audification has a duration of ~0.76 seconds and exhibits slightly less periodicity, although still detectable. At 118,300 characters (7 cipher periods),

[6] Definitions of *sonification* and *audification* are provided in Chapter 1 Introduction.

the audification duration is ~2.68 seconds, with periodicity becoming ambiguous and hard to detect. The longest audification – 422,500 characters (25 cipher periods) – lasts ~9.58 seconds and its periodicity is almost imperceivable.

Noise Generators

The example patch `E3.enigma.noise.pd` shown in figure 17 presents a wavetable oscillator configured to play back the pseudo-random sequence in the file `enigma_noise.wav`. In this instance, the sequence is played back at a rate of 16900 characters per second, resulting in a noise spectrum that has a subtle yet detectable repetition. Manipulating the playback rate towards slower speeds leads to a reduction of bandwidth whilst also extending the period of repetition.

This method of bandwidth manipulation is further extended in the example patch `E4.enigma.tenney.pd`, which realises the instrument described by James Tenney as used in his composition *Analog #1: Noise Study* in 1961. Tenney states that “the instrument is designed to produce noise-bands by random amplitude-modulation of a sinusoidal carrier” (Tenney, 1969, p. 6). This synthesis patch and Tenney’s ‘Noise Study’ forms the basis of the portfolio work *ENIGMA-N*, described in Chapter 4.

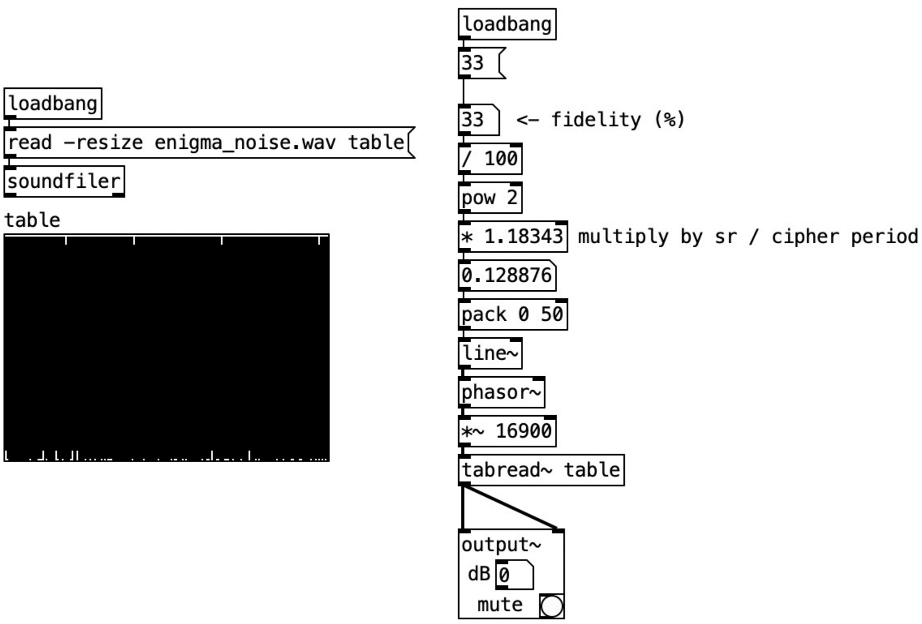

Figure 17 Enigma noise wavetable Pd patch

Periodic Oscillators

Whilst the Enigma noise generators described above contain complete pseudo-random sequences, we are also free to use only portions of these sequences, thereby increasing the perception of periodicity. The example patch `E2.enigma.oscillator.pd` explores the continuum between periodicity and aperiodicity by applying *same letter stepping* at audio-rate and monitoring the first few internal states of the scrambler. Reading the left-hand wheel immediately after the *Eintrittswalze* gives a cipher period of just 26 characters and produces a clear periodic waveform with few noise components except for aliasing errors. The middle wheel gives a cipher period of 650 characters and oscillates at a frequency around $1/25^{th}$ the rate of the left-hand wheel. This wheel exhibits clear periodicity but its spectral context has noise-like components. Right-hand wheel gives the full cipher period of 16900 characters and oscillates at $1/650^{th}$ the rates of the left-hand wheel, approaching the threshold of perceptible periodicity at fairly low frequencies.

Example patch `E7.enigma.morphing.wavetable.pd` offers an alternative approach to wavetable synthesis using Enigma-derived pseudo-random sequences, whereby the number of values and position in the wavetable can be manipulated. Furthermore, it is possible to ‘scan’ around the wavetable, crossfading between different sections of cipher text. This “morphing wavetable” approach is similar to the methods used in synthesizers like the PPG Wave and Synthesis Technology’s E350 Morphing Terrarium. (Sound on Sound, 1996)(Synthesis Technology, 2023).

Finally, rather than using a sawtooth oscillator to read through the contents of the wavetable sequentially, the pseudo-random sequence can instead be used as a nonlinear transfer function to process other waveforms. Patch example `E8.enigma.waveshaping.pd` demonstrates the processing of a sine wave in this manner, with larger cipher periods increasing the harmonic content of the waveform in a manner not dissimilar to oscillator sync or wavefolding.

This particular synthesis technique is employed in the study *Enigma Krell 2*.

3.5 Summary

The taxonomy given in section 3.1 for approaches to obtaining pseudo-random sequences from the Enigma cipher presents an in-depth investigation into the sequence characteristics and the effect of feedback on the cipher system. The overview and working principle of the list_enigma object given in section 3.2 and the approaches to Enigma synthesis given in section 3.4 present an overview and explanation of the Enigma cipher simulation – an exploration of the example patches provides further insight into the usage of the list_enigma object. The parameter maps given in section 3.3 outline several systems that work well with an alphabetical character set.

Chapter 4 Analysis of Works

4.1 Ringstellung

This work began its life during the development of Enigma simulation in Pd and relied on Daniel Pallocks' (2016) Universal Enigma simulator to generate the pseudo-random sequences. As with many of works in this portfolio, *Ringstellung* makes use of the no-input encryption method of generating pseudo-random patterns, however, this piece specifically examines the patterns that occur in the scrambler's signal path when using this feedback technique. The piece was written for a string quintet and was performed by a larger ensemble for this recording.

The piece was written in two stages: A precomposition stage whereby the feedback algorithm was manually iterated, logged, and analysed for patterns, and a scoring stage in which the text was mapped to pitch, and patterns were emphasised through orchestration and embellishment.

Precomposition

Figure 18 shows the monitor view of Daniel Pollacks' (2016) Enigma simulator used to derive the pseudo-random sequences for use in *Ringstellung.* The process of iteration began with the wheels set to `A-Z-Y` and the UKW set to A. The rationale for these initial wheel positions is that on the first key press of the letter A, the middle wheel turns over to the A position and remains in this position throughout the following 26 keypresses. This is required in order to maximise the probability of patterns occurring, as whenever this central wheel turns over a new set of potential patterns is generated.

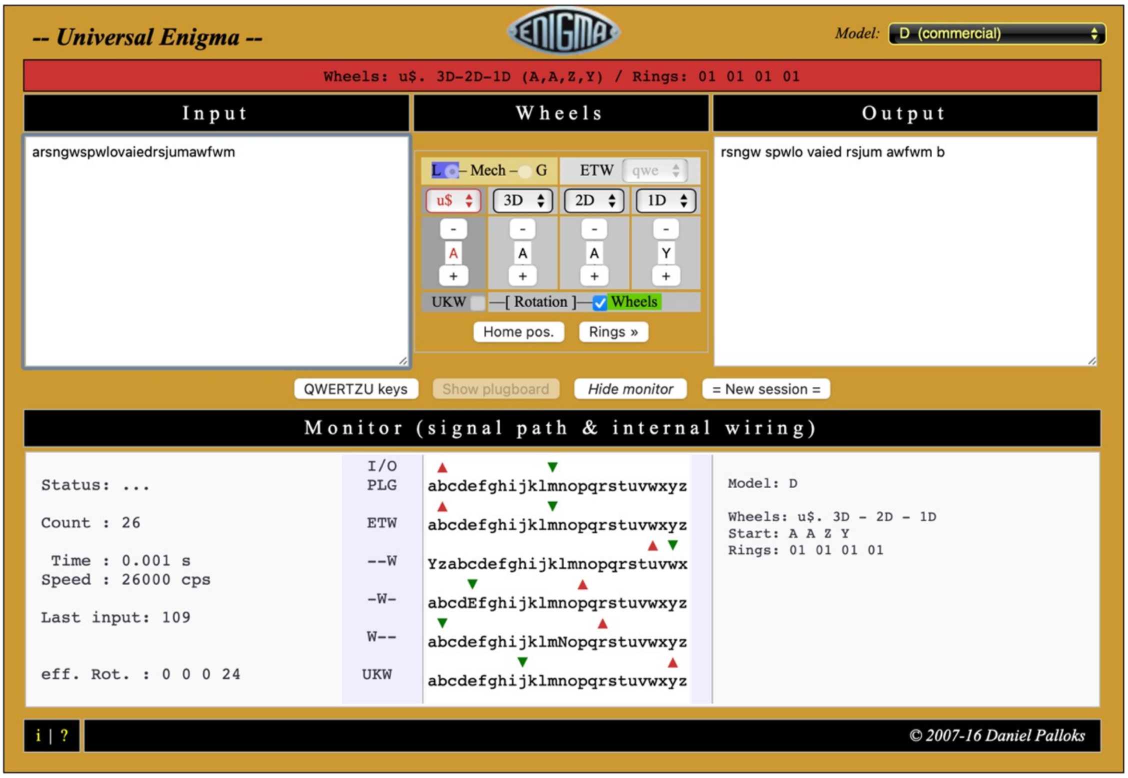

Figure 18 Daniel Pallocks' online Enigma simulator (Pallocks, 2016)

For each input character fed into the cipher, the readings monitored in Pollack's simulator were added to an Excel spreadsheet (see fig N) according to the position in the scrambler and with reference to the current wheel positions. Once the 26 iterations had been logged, the data in the table was analysed for patterns which were then colour coded. The frequency of each pattern is mapped to dynamics, whilst the characters themselves are mapped to pitch.

For the piece's pitch map, I opted to address each instrument according to their particular pitch range. The letter A is therefore mapped to the lowest pitch of each instrument – E2 for double bass, C2 for violincello, C3 for viola, and G3 for violin. An unexpected aspect of this approach is that when all instruments are responding to the ciphertext simultaneously they produce a chord that forms a major triad. As the pitch map is chromatic, this serves to inject a subtle sense of harmony that sits in satisfying contrast to the underlying atonality.

Notation

Once the pitch map for each instrument had been created, an expanded version of the first table was produced mapping the pitches for each instrumet. This table acts as a guide in translating the cipher text to music notation. Figure 19 shows the pitch map in detail to the left of part of the scoring guide used in the composition. Whilst each pattern that occurs within the sequence is mapped to a specific dynamic, this alone would not provide enough emphasis of the pattern. I therefore determined that patterns would be sounded by all players simultaneously, whilst characters that fall outside of a pattern should be played solo.

	Position in Scrambler									
	INPUT	ETW	W1	W2	W3	UKW	W3	W2	W1	OUTPUT
Wheel Positions	**Encrypted Character**									
AAZ	A	J	F	T	Q	L	X	H	C	R
AAA	R	D	S	A	C	E	Z	Q	K	S
AAB	S	K	K	Q	M	B	V	C	Y	N
AAC	N	X	W	K	R	H	G	D	P	G
AAD	G	Q	Y	P	F	G	C	W	E	W
AAE	W	B	M	J	U	V	W	N	O	S
AAF	S	K	F	S	N	Z	N	X	W	P
AAG	P	R	J	G	H	R	K	U	H	W
AAH	W	B	E	N	Z	N	S	A	G	L
AAI	L	Z	A	A	C	E	Z	Q	Q	O
AAJ	O	I	B	A	C	E	Z	Q	E	V
AAK	V	V	M	V	B	M	Q	J	T	A
AAL	A	J	N	V	B	M	Q	J	S	I
AAM	I	H	T	X	L	Q	T	F	O	E
AAN	E	C	F	A	C	E	Z	Q	Y	D
AAO	D	L	W	J	U	V	W	N	R	R
AAP	R	D	U	T	Q	L	X	H	Z	S
AAQ	S	K	L	P	F	G	C	W	F	J
AAR	J	P	H	Z	E	C	A	S	X	U
AAS	U	G	D	H	K	S	F	G	Q	M
AAT	M	Y	B	J	U	V	W	N	C	A
AAU	A	J	S	D	D	T	I	O	V	W
AAV	W	B	C	X	L	Q	T	F	H	F
AAW	F	M	E	J	U	V	W	N	X	W
AAX	W	B	D	F	S	K	H	L	V	M
AAY	M	Y	C	B	J	Y	R	P	U	B

Central Patterns:						Dynamics:
A	C	E	Z	Q	4	*fff*
J	U	V	W	N	3	ff
P	F	G	C	W	2	f
T	Q	L	X	H	3	p
V	B	M	Q	J	2	pp
X	L	Q	T	F	2	ppp

Figure 19 *No-input encryption* patterns with dynamics assigned

Similarly, when the output/input character is repeated every two bars, this is reflected by all players sounding their pitches simultaneously. As mentioned in Chapter 2, the word *Ringstellung* is translated as ring setting – whilst this piece is not a study of different ring settings on the Enigma cipher, the ring concept brings to mind the use of feedback in the algorithm and the circular interaction of encryption through the machine's signal path. This theme of circular movement is incorporated into the orchestration of the piece, whereby solo notes are passed between each instrument in the quintet. This action can be seen in the lower section of the right table in Figure 20. The resulting effect is subtle with individual notes panning back and forth in the stereo field.

	Instrument			
	BASS	CELLO	VIOLA	VIOLIN
Character	**Pitch**			
A	E2	C2	C3	G3
B	F2	C#/Db2	C#/Db3	G#/Ab3
C	F#/Gb2	D2	D3	A3
D	G2	D#/Eb2	D#/Eb3	A#/Bb3
E	G#/Ab2	E2	E3	B3
F	A2	F2	F3	C4
G	A#/Bb2	F#/Gb2	F#/Gb3	C#/Db4
H	B2	G2	G3	D4
I	C3	G#/Ab2	G#/Ab3	D#/Eb4
J	C3#/Db3	A2	A3	E4
K	D3	A#/Bb2	A#/Bb3	F4
L	D#/Eb3	B2	B3	F#/Gb4
M	E3	C3	C4	G4
N	F3	C#/Db3	C#/Db4	G#/Ab4
O	F#/Gb3	D3	D4	A4
P	G3	D#/Eb3	D#/Eb4	A#/Bb4
Q	G#/Ab3	E3	E4	B4
R	A3	F3	F4	C5
S	A#/Bb3	F#/Gb3	F#/Gb4	C#/Db5
T	B3	G3	G4	D5
U	C4	G#/Ab3	G#/Ab4	D#/Eb5
V	C#/Db4	A3	A4	E5
W	D4	A#/Bb3	A#/Bb4	F5
X	D#/Eb4	B3	B4	F#/Gb5
Y	E4	C4	C5	G5
Z	F4	C#/Db4	C#/Db5	G#/Ab5

Wheel Position, Bar, Articulation, Instrument	**Position in Scrambler**									
	INPUT	ETW	W1	W2	W3	UKW	W3	W2	W1	OUTPUT
	Encrypted Character									
AAZ	**A**	**J**	**F**	**T**	**Q**	**L**	**X**	**H**	**C**	**R**
1, 2	ALL	ALL	VIOLIN1	ALL	ALL	ALL	ALL	ALL	VIOLIN2	ALL
ARTICULATION	STACCATO	STACCATO	STACCATO	LEGATO	LEGATO	LEGATO	LEGATO	LEGATO	STACCATO	STACCATO
VIOLIN1	G3	E4	C4	D5	B4	F#/Gb4	F#/Gb5	D4		C5
VIOLIN2	G3	$4		D5	B4	F#/Gb4	F#/Gb5	D4	A3	C5
VIOLA	C3	A3		G4	E4	B3	B4	G3		F4
CELLO	C2	A2		G3	E3	B2	B3	G2		F3
BASS	E2	C#/Db3		B3	G#/Ab3	D#/Eb3	D#/Eb4	B2		A3
AAA	**R**	**D**	**S**	**A**	**C**	**E**	**Z**	**Q**	**K**	**S**
3, 4	ALL	ALL	VIOLA	ALL	ALL	ALL	ALL	ALL	CELLO	ALL
ARTICULATION	STACCATO	STACCATO	STACCATO	LEGATO	LEGATO	LEGATO	LEGATO	LEGATO	STACCATO	STACCATO
VIOLIN1	C5	A#/BB3		G3	A3	B3	G#/Ab5	B4		C#/Db5
VIOLIN2	C5	A#/BB3		G3	A3	B3	G#/Ab5	B4		C#/Db5
VIOLA	F4	D#/Eb3	F#/Gb4	C3	D3	E3	C#/Db5	E4		F#/Gb4
CELLO	F3	D#/Eb2		C2	D2	E2	C#/Db4	E3	A#/Bb2	F#/Gb3
BASS	A3	G2		E2	F#/Gb2	G#/Ab2	F4	G#/Ab3		A#/Bb3
AAB	**S**	**K**	**K**	**Q**	**M**	**B**	**V**	**C**	**Y**	**N**
5, 6	ALL	ALL	BASS	CELLO	VIOLA	VIOLIN2	VIOLIN1	VIOLIN2	VIOLA	ALL
ARTICULATION	STACCATO	STACCATO	STACCATO	STACCATO	STACCATO	STACCATO	STACCATO	STACCATO	STACCATO	STACCATO
VIOLIN1	C#/Db5	F4					E5			G#/Ab4
VIOLIN2	C#/Db5	F4				G#/Ab3		A3		G#/Ab4
VIOLA	F#/Gb4	A#/Bb3			C4				C5	C#/Db4
CELLO	F#/Gb3	A#/Bb2		E3						C#/Db3
BASS	A#/Bb3	D3	D3							F3
AAC	**N**	**X**	**W**	**K**	**R**	**H**	**G**	**D**	**P**	**G**

Figure 20 Pitch maps and arrangement spreadsheet

During the process of scoring *Ringstellung* using music notation I adapted how the cipher text should be sonified. It was originally planned that differences between pattern and chaos would be even further emphasised through legato and staccato articulation respectively, however, it became clear that this would be largely unplayable with any degree of accuracy. Instead, a glide was placed between the first two notes of each pattern. I also chose to emphasise the double notes that occur when plaintext becomes cipher text with an accent. Figure 21 shows the first seven bars of music notation for *Ringstellung*.

Figure 21 Section of *Ringstellung* full score

Summary

Overall, the piece was a successful study in sonification of Enigma-derived random sequences to show the incidence of pattern. Whilst the underlying data relating to specific characters is somewhat obscured, it's easy to identify the points at which plain text is fed back and becomes cipher text. Moreover, pattern and chaos is clearly differentiated and after repeated listening, the patterns themselves become individually recognisable. The movement of encryption through the Enigma machine signal path is subtly reflected in the passing of cipher characters between instrument groups, however, the way in which the patterns interrupt this process diminishes its overall effect. The reliance on quarter note durations throughout the piece is satisfactory given its short duration, however, a longer work would clearly require more rhythmic variation.

4.2 Durations 1a and 1b

This composition in two parts was inspired by Pierre Boulez's *Structures 1a* and John Cage's *Music of changes* for their structural and post-atonal complexity. Given my experience so far employing the Enigma cipher as a pseudo-random generator with the above parameter maps, it would appear a worthwhile experiment to see if the Enigma could produce music with a similar quality.

Precomposition

For this composition, eight Enigma PRGs were employed–two per voice for four voices in total; this appeared to be an appropriate number of voices to produce a score that would be more likely to be playable by a human pianist.

The duration map follows the systems outlined in Chapter 3, with the alphabet split in half, producing 13 durations and 13 rests. Mapping amplitude was a simple process of converting the alphabet characters to integers and scaling between 0 - 127. The pitch maps follow the chromatic method as this is most suitable for piano instrumentation, however, the span of each Enigma voice differs between the two pieces. The pitch maps for the four Enigma ciphers for the compositions *Durations 1a* and *Durations 1b* are given in figures 22 and 23 respectively.

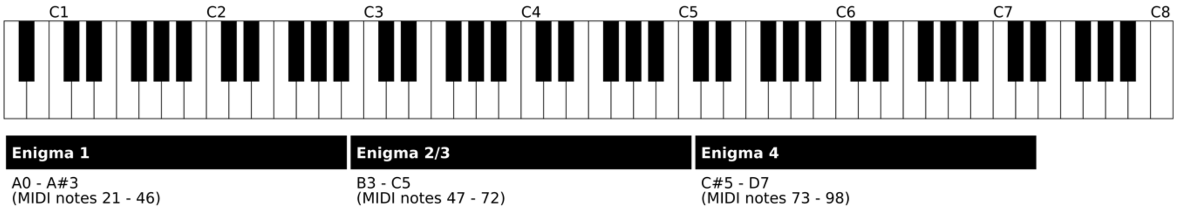

Figure 22 Keyboard spans in *Durations 1a*

For *Durations 1a*, two Enigma ciphers are dedicated to the pitches between approximately C3 and C5. As a result, notes in this range occur twice as frequently as the single Enigma ciphers mapped to the lowest and highest ranges of the piano.

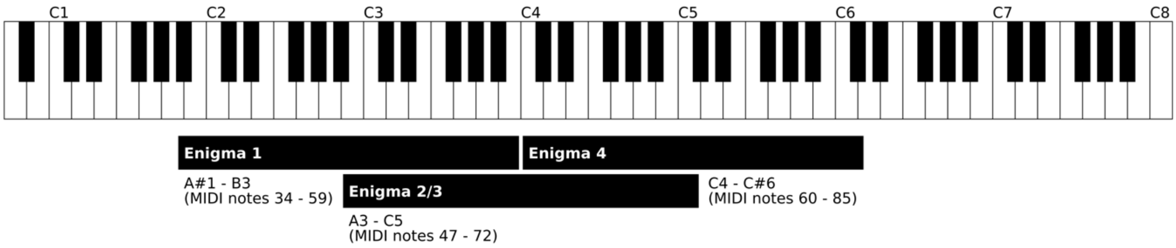

Figure 23 Keyboard spans in *Durations 1b*

By comparison, fig. n shows the more limited overall pitch range employed in *Durations 1b*. With the half of the alphabets produced by Enigma 1 and 4 mapped to the same pitches as Enigma 2 and 3, there is even more emphasis on the middle of the keyboard compared to *Durations 1a*. Moreover, the overall frequency range of this piece is reduced. This overlapping also results in three and four-note chords that are not possible in the previous composition.

Notation

Initially I attempted to notate the score by hand in MuseScore, but with little pre-existing experience scoring with music notation, particularly rhythmic notation, MIDI data was instead recorded into Logic from Pd so that a basic score could be produced. Figure 24 shows a section of the score for *Durations 1a*. As Logic does not generate dynamics markings automatically in relation to the varying velocities generated by the Pd maps, the score was configured so that dynamics could be represented as different colours.

Figure 24 Section of *Durations 1a* score

Summary

The scores for *Durations 1a* and *1b* are included in the project and can be located in the folder `/portfolio/scores`. I should be noted that scores are not considered final in a performable sense, as there are instances where the pitch range is far too wide to be played simultaneously or in quick succession. Colour-coded dynamics could be changed out for correct dynamics notation. The composition of these two pieces was a useful experiment in the process of creating written notation using algorithmic sequences in Pd. To conclude, the Enigma PRG sequences are clearly well suited to compositions with a similar structure and post-tonal complexity as those these works are inspired by.

4.3 Study 13

This composition was inspired by Karlheinz Stockhausen's *Studie II*. Completed in 1954 at the West Deutche Rundfunk (Williams, 2016), *Studie II* is a composition for tape studying the organisation of sine-tones in groups, both vertically to achieve synthetic timbres, and horizontally for ordering sine-tone groups into a larger structure. Whilst the organization of the piece follows a serial method, what is particularly significant for my composition is how Stockhausen abandoned the usual 12-tone system in favour of a larger scale of 25 tones. *Study 13* was born from my first experiments with parameter mapping in Pd and the organisation of single sine-tones with Enigma PRG sequences. These early experiments were crude and unsophisticated, however, further investigation into Stockhausen's creation of *Studie II* gave rise to a conceptual framework on which to develop a piece using the Enigma PRG in Pd. Most influential were the superposition of single sine-tones to produce complex "synthetic sounds" (Toop, 1979, 380), and the articulation of a note's dynamics and duration as trapezoid-shaped amplitude envelopes (shown in figure 25). In addition, the reverberation employed in *Study 13* led to some surprising artefacts that contributed positively to the piece's sonic character.

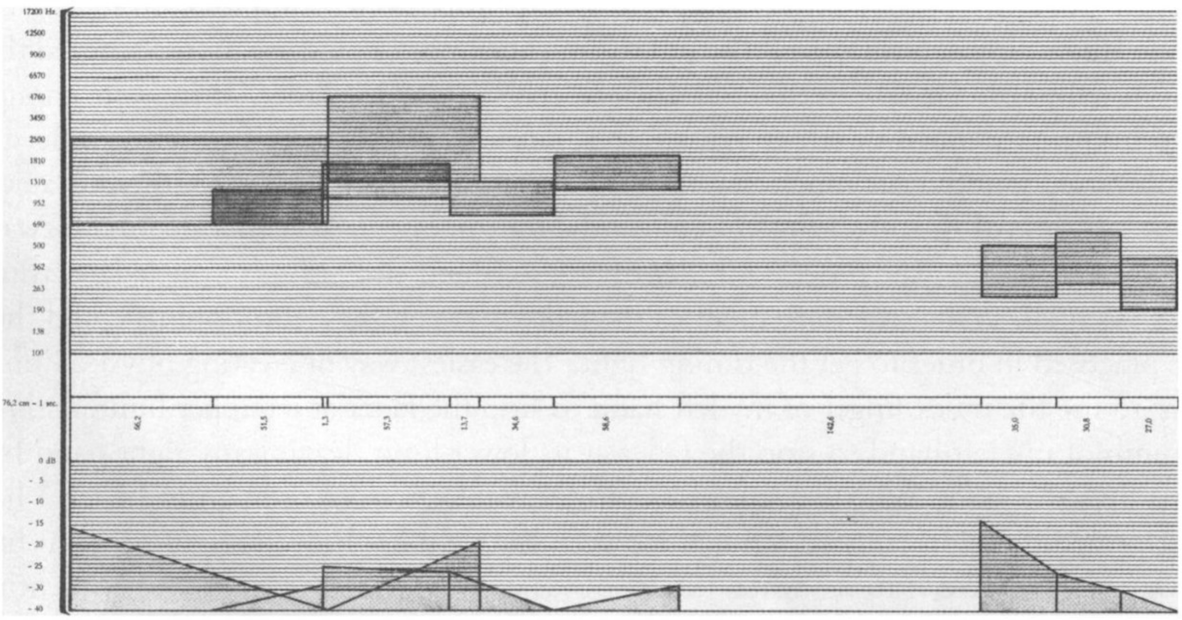

Figure 25 Part of realisation score for *Studie II* (Williams, 2016, p. 465)

The following section will provide a comparison between these elements in my piece (*Study 13*) and in *Studie II*, reporting on the technical development and considerations for the application of the Enigma PRG in the ordering of sine-tones and control of dynamics and duration.

Pitch Mapping

Creating a piece in the style of *Studie II* provides a good opportunity to explore the possible approaches to pitch mapping, since I felt the superimposed sine-tone ‘groups’ emphasise the interval relationships more than through successive pitches. As suggested by the name, *Study 13* is as much a tip-of-the-hat to Stockhausen’s piece as it is a study in of the 13-TET chromatic scale that maps so well alphabetically. Also, rather than superimpose five tones, *Study 13* uses the nine internal states of the Enigma cipher through scrambler monitoring. Combined with the 13-TET scale, this results in a far denser sine-tone group compared to that of *Study II*.

As Williams notes, the frequencies of the sine-tones in *Studie II* are each based on an interval with the ratio of $\sqrt[25]{5}$ and are employed in groups of five superimposed tones (Williams, 451). These intervals are “slightly larger than the equal tempered semitone (12√2)” (Clarkson, 1960, 646). When we compare this to the $\sqrt[13]{2}$ ratio of the 13-TET scale and the $\sqrt[13]{3}$ ratio of the Bohlen-Pierce scale, it would seem that perhaps the latter would be more suitable due to its ratio similarity with Stockhausen’s $\sqrt[25]{5}$ scale. The decision of which scale to use also had implications on the overall frequency range throughout the piece. When mapping the nine internal states of the cipher directly to nine sine-tones, the frequencies of these sine groups remain within a constant range: two octaves in 13-TET and three octaves (or two tritaves) in Bohlen-Pierce. This leads to a sequence of sine groups which is quite monotonous, since the nine sine-tones generally occupy the same frequency ranges on successive notes. By employing a further Enigma PRG output to transpose these sine groups, this second sequence essentially serves as the main pitch sequence, with the former instead describing the timbral makeup of each note. By doing so, the overall frequency range is extended to just under four octaves for 13-TET or just under six octaves in Bohlen-Pierce.

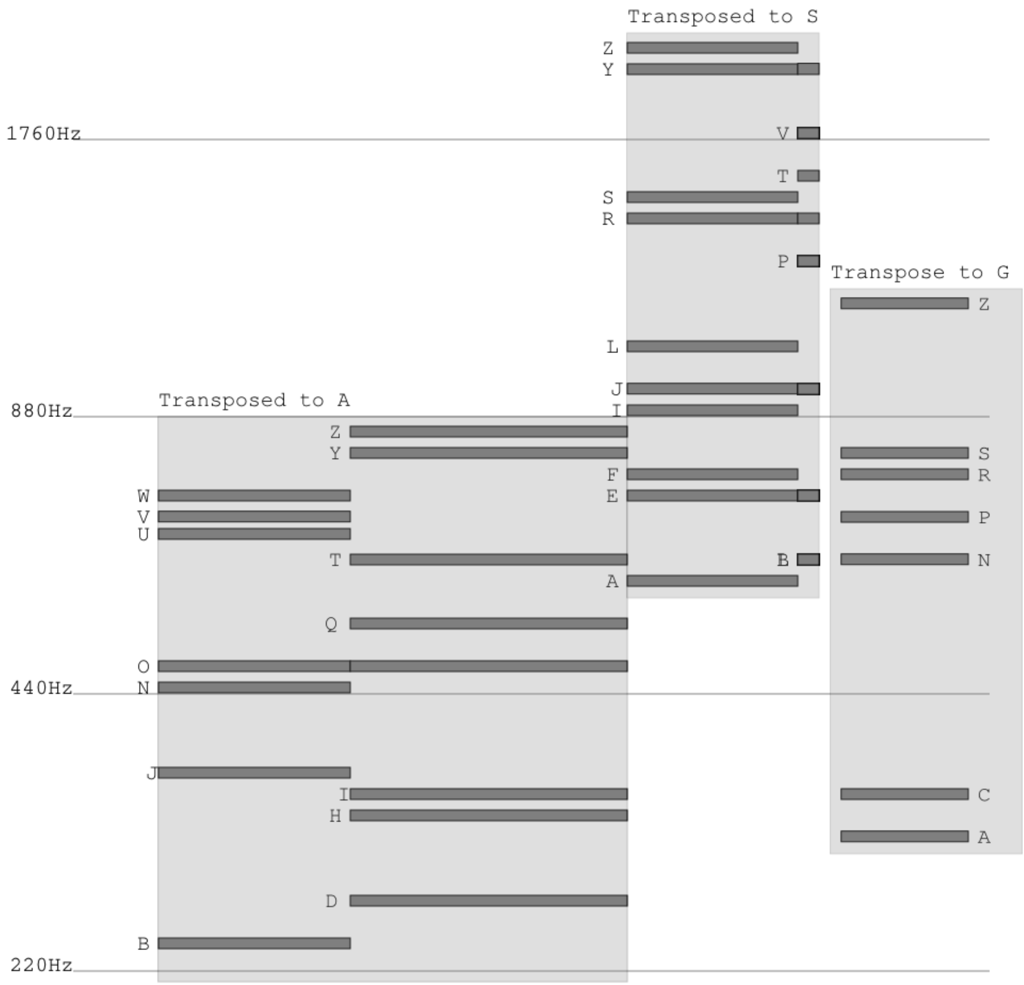

Figure 26 Diagram showing pitch ranges in *Study 13*

Upon audition of the two scales for this piece, the reduced transposed range of 13-TET groups sounded more pleasing than in Bohlen-Pierce, with the increased density of sine-tones leading to a more cohesive and singular timbre. The diagram shown in figure 26 gives a visual example of the effect of transposing the mapped sine groups.

Duration and Amplitude Mapping

As with *Durations 1a* and *1b*, *Study 13* also employs the method of dividing the pseudo-random sequence equally into 13 notes and rests, although this piece recreates the method of interpolating between two amplitude values over the duration of the note as done in *Studie II*. The amplitude mapping process is simple, with the letters of the pseudo-random sequences replaced with integers, which are in turn scaled to the required amplitude range in Pd. The amplitude scale is linear, so the letter ‘N’ delivers an amplitude of 50%, letter ‘O’ delivers an amplitude of 53.84%, 1/26th louder than ‘N’. Figure 27 shows a diagram of the trapezoidal envelopes with the note’s initial and final amplitude labelled alphabetically. The diagram also shows the resulting durations obtained by multiplying 100ms by the numeric substitute of the alphabetic sequence.

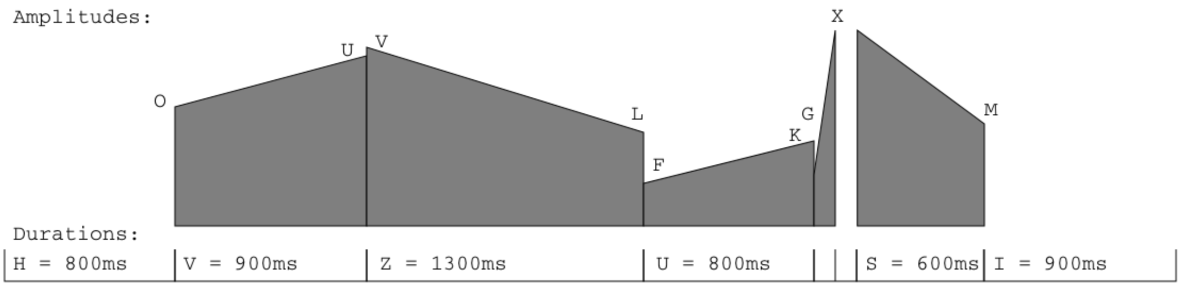

Figure 27 Diagram showing trapezoid envelopes in *Study 13*

The first version of *Study 13* used a single bank of nine sine-tones summed to a single amplitude envelope, however, this caused issues with ‘pops’ and ‘clicks’ when sine-tones were cut off at a non-zero crossing. Applying a subtle slew of ~100ms between envelope events reduced the issue, however, through my research into the development of *Studie II* and my own brief experience working with analogue tape, it became apparent that crossfading between sounds would produce transitions more akin to those in *Studie II*. To achieve this, two parallel voices were coded, whereby notes alternate between the two voices. The subtle slewing of the amplitude envelopes was left in place, both to avoid the aforementioned ‘pops’ and ‘clicks’ and so as to smoothly fade between the two sine groups, allowing for satisfying overlaps of tones. Figure 28 shows a simplified diagram of the Pd patch used for *Study 13*, with the elements in black highlighting the two alternating voices.

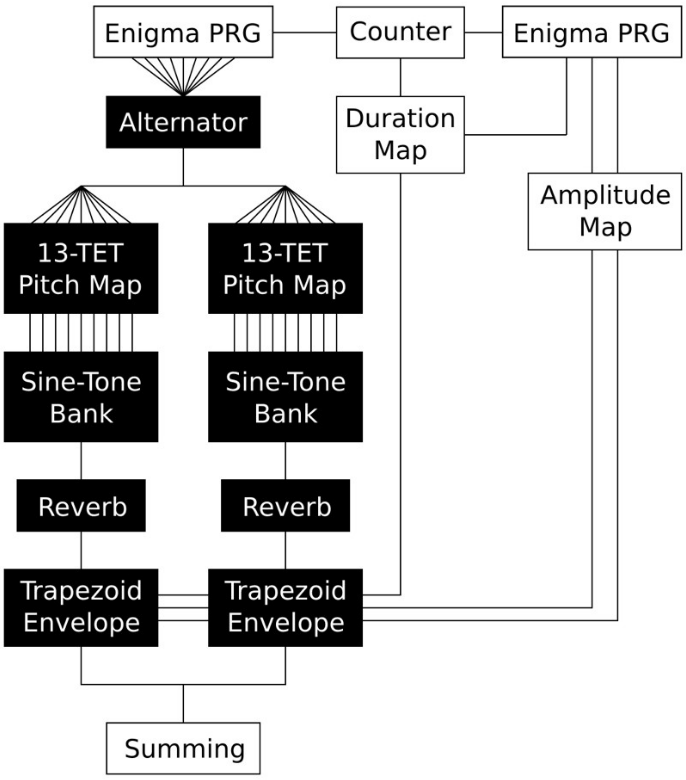

Figure 28 Flow diagram of *Study 13* Pd patch

Reverberation

Stockhausen's notes on *Studie II* describe an "aleatory modulation of the sequence of sinusoidal notes after reverberation", with Williams interpreting *aleatory* here as "the non-linear way in which sine waves are reflected around the reverb chamber and mixed together with multiple echoes of themselves in different time and, more importantly, different phase relationships" (Williams, 2016, p. 458). Koenig suggests that "reverberation was used primarily as a convenient way of mixing several sine waves together without build-up of tape hiss caused by overdubbing" (ibid). This aligns with Maconie, who describes a process of "injecting descending arpeggio strings of five sine tones at a time into an echo chamber at high speed and recording the reverberation, in an apparent attempt to make the partial tones combine in advance" (Maconie, 2016, p. 33).

In *Studie II*, this process occurs *before* to the application of amplitude envelopes to the acoustically mixed sine groups, and in order to emulate this effect, Pd's basic mono *rev1~* object is placed after the summed sine-tones but before the amplitude envelope. As two parallel voices are being used to simulate crossfades, a *rev1~* object is employed on each voice. The particular arrangement of these alternating voices means that the reverb tails also alternate. Figure 29 identifies the appearance of sine-tones from previous frequency groups overlapping with the sine-tones of the new group as part of the reverb tail.

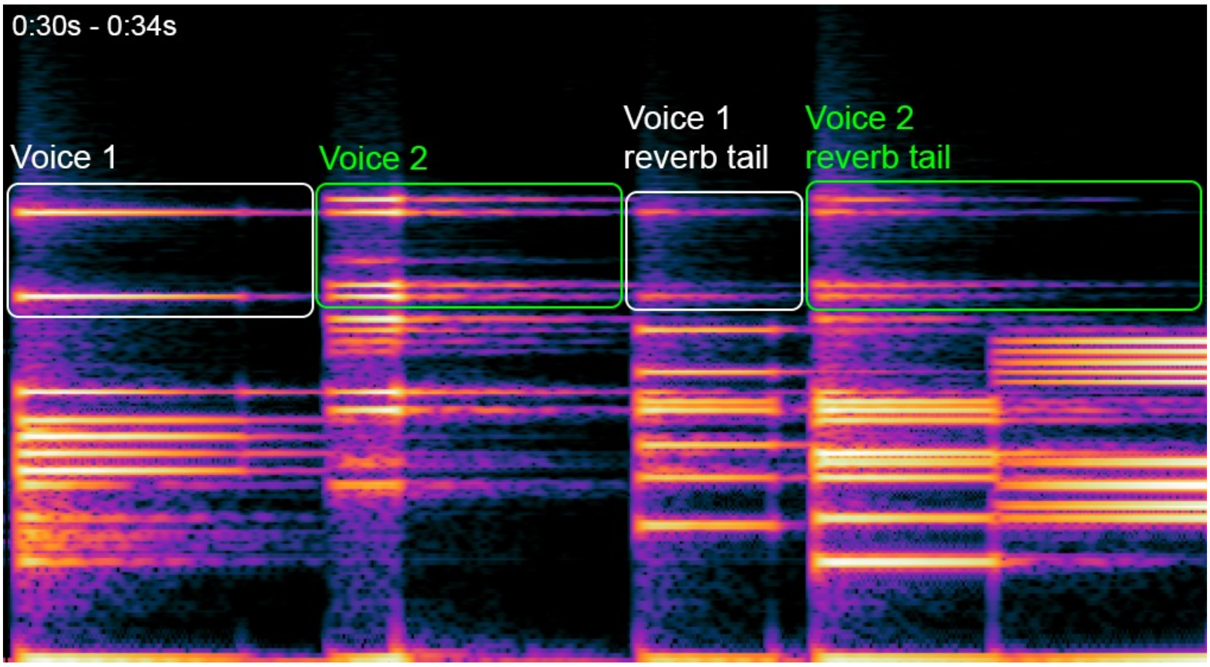

Figure 29 Spectrograph showing reverb tails

The overall effect of the interaction of the *rev1~* object on the alternating voices is sonically pleasing and adds a degree of unpredictable character to the sine-tones in what Stockhausen might refer to as "aleatory modulation" (Williams, 2016, p. 458).

Summary

Pure Data and is well suited to experimentation with Enigma-derived pseudo-random sequences using the list_enigma object and presents a customisable environment for creating *Elektronische Musik* in real-time. Pd's mathematical operations offer fine control over frequency, allowing for an in-depth exploration of alphabetical pitch maps with microtonal scales. The application of *scrambler monitoring* leads to parallel streams of rhythmically-related pseudo-random sequences, which works well in the creation of superimposed sine-tone groups. Finally, an advantage of the determinist nature of Enigma PRG sequences is the repeatability of its pseudo-random sequences and compositions themselves.

4.4 Enigma Krell 1 and 2

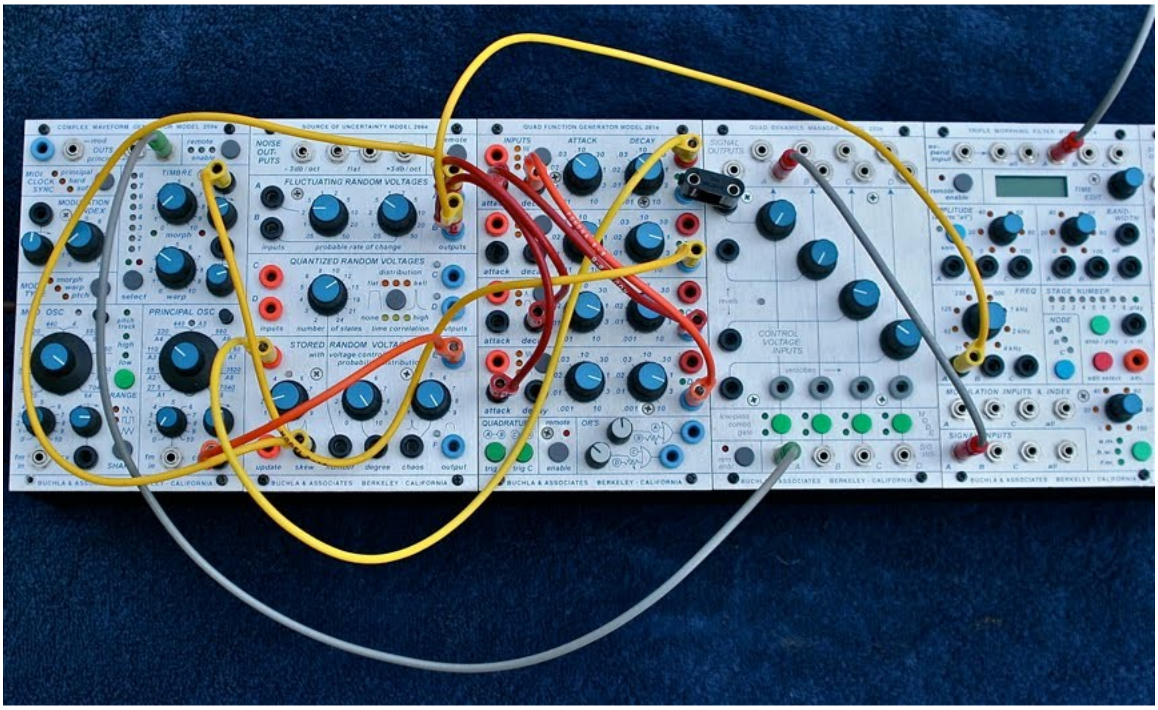

Figure 30 Todd Barton's Buchla patch (MATRIXSYNTH, 2012)

Perhaps the most iconic use of random sequences in recent music made with modular synthesizers is Todd Barton’s self-generating *Krell Muzak* pieces, created on the Buchla 200e (Figure 30). The name references Bebe and Louis Barron’s *Ancient Krell Music* from the 1956 film Forbidden Planet, and though the processes of their realisation and the end results are quite different, both exhibit a similarly enigmatic sci-fi aesthetic. The pieces, titled *Krell Muzak 1* and *Krell Muzak 2* first appeared online on Barton’s Soundcloud (bartonmusic, 2012) linked to a post on MATRIXSYNTH showing a photo of the patch setup on a Buchla used in the piece’s creation (MATRIXSYNTH, 2012).

Included in the portfolio of works are two studies and variations on the Krell patch using the Enigma PRG. The first is realised by means of a hybrid setup that combines Pd with a Buchla 200 modular synthesizer, and the second is a realisation of the Krell patch entirely within Pd. Both of these pieces also act as studies of alphabetic pitch mapping using the Bohlen-Pierce scale.

Barton’s Krell Patch: Working Principle

Figure 31 shows a diagram of the routing of Barton’s Krell patch. The central function generator produces an asymmetrical rise/fall cycle that modulates the amplitude of the oscillator via a lopass gate. Whenever the function has completed its cycle, an ‘end of cycle’ trigger causes the stored random section to send a new voltage to the pitch of the oscillator. The effect is of an oscillator tone fading in and out, each time at a new pitch. The patch is further augmented by three more cycling function generators and two fluctuating random voltages that modulate the overall rate, pitch transposition, and timbre of the patch.

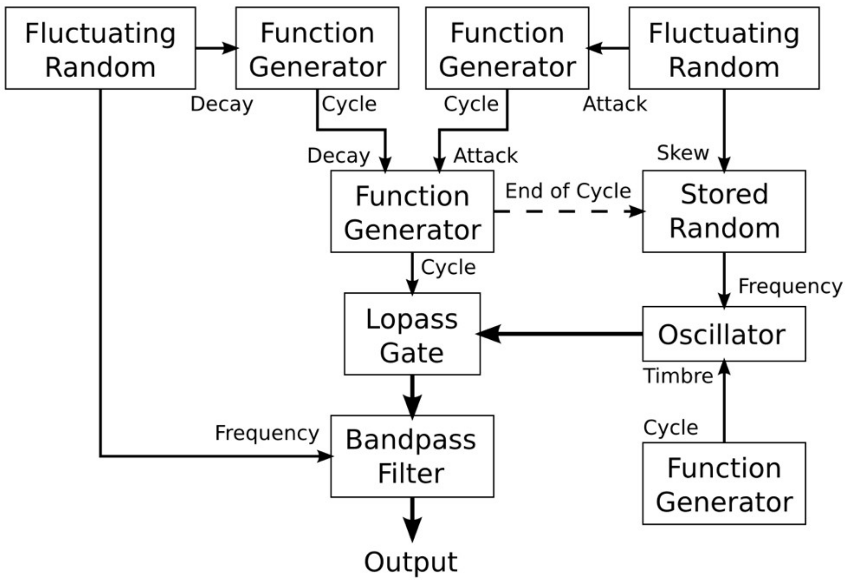

Figure 31 Krell patch flow diagram

Buchla/Pd Implementation

Figure 32 shows a flow diagram of the hybrid realisation, whereby the *stored random voltages* section and complex oscillator in the Buchla have been replaced with the Enigma PRG and a Buchla-style oscillator built in Pd. Communication between the two domains is achieved via an audio interface. Pd’s *threshold~* object is used to detect triggers produced by the function generator's 'end of cycle' trigger. Whilst the Pd-based Enigma oscillator allows accurate pitch mapping in the Bohlen-Pierce scale, this is the only aspect of the Krell patch that is mapped to the Enigma PRG. Duration and timbre are modulated by basic cycling functions/LFOs as per Barton's original Krell patch. The piece as recorded is two overlayed and synchronised recordings of the patch, both featuring an additional static drone oscillator, and each spaced one tritave apart. Whilst this patch could be considered 'authentically Krell', the influence of the Enigma is quite minimal except for providing a unique framework on which to define the pitch sequences.

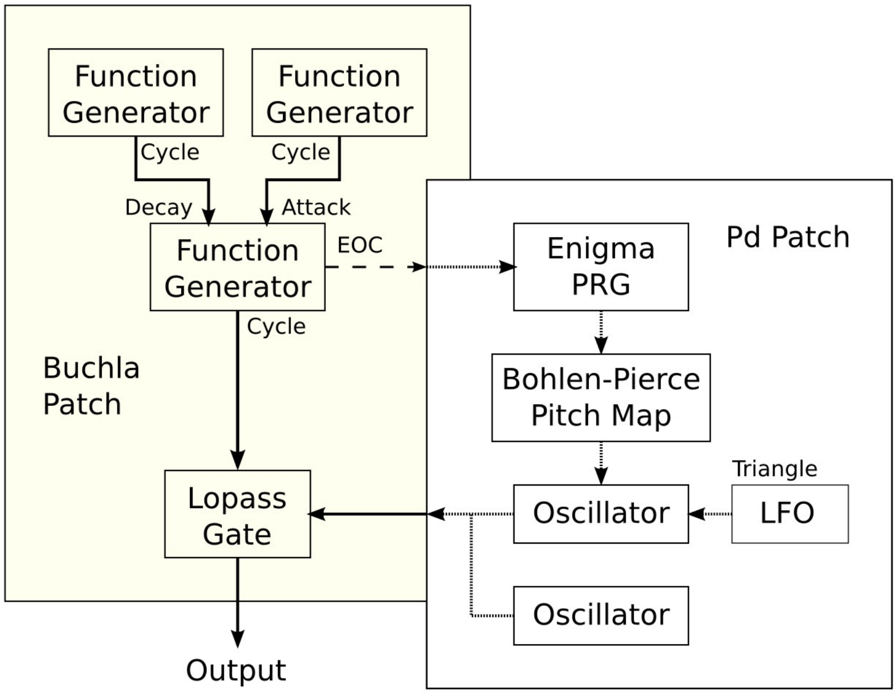

Figure 32 Flow diagram of processes in *Enigma Krell 1*

Full Pd Implementation

When fully realised in Pd there are more opportunities to engage the Enigma PRG in a Krell-style generative patch. Figure 33 shows a flow diagram of the Pd patch used for the piece. As with the *Study 13* piece, an initial Enigma PRG is used to define the duration of notes, but instead alternates between rising and falling durations of the amplitude envelope. Two further Enigma PRGs are employed: One with two readings taken from the internal states which are mapped to both pitch and timbre, and a second that operates as a slower pace and provides transpositions of the main pitch sequence – this is equivalent to the *fluctuating random voltages* employed in Barton's Krell patch to skew the probability of higher or lower pitches being generated.

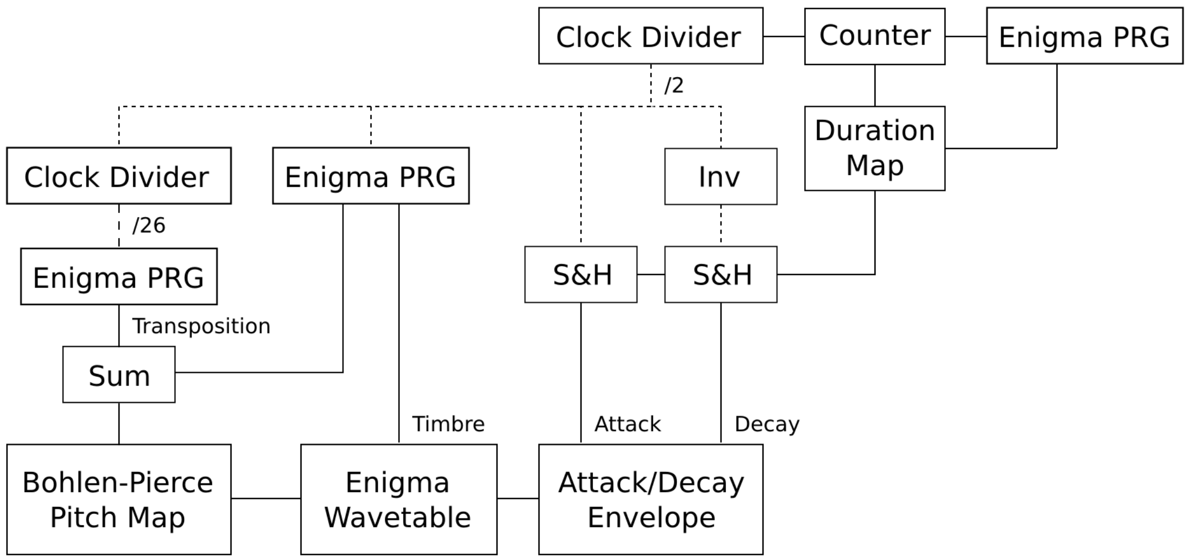

Figure 33 Flow diagram of Pd patch for *Enigma Krell 2*

The native Pd implementation also makes use of the Enigma cipher as a sound source in place of the Buchla-style oscillator patch used in the hybrid method above. This follows the wave-shaping process described in Chapter 3. Continuous variation of timbre is achieved by mapping the output of an Enigma PRG to the index modulation so that each new note has a different timbre.

Summary

The list_enigma object allows Enigma PRGs to be used in place of random and pseudo-random generators in classic synthesizer patches. In instances where sequences with discrete steps between values are required, the Enigma PRG provides an effective and repeatable pseudo-random sequence with a uniform distribution. Though the resolution of values is limited, this does not seem to be a noticeable disadvantage in practical use. Furthermore, the ability for Enigma PRG sequences to be employed for the purpose of waveshaping further explores the flexibility of the Enigma cipher in a musical context.

4.5 ENIGMA-N

This composition uses James Tenney's *Analog #1: Noise Study* as a conceptual framework on which to systematise Enigma PRG parameter mapping and noise synthesis. Completed in December 1961, Tenney's noise study was created at Bell Labs using Max Mathew's MUSICn software, inspired by the sounds of road traffic and tunnels from his regular commute along Route 22 to and from Bell Labs (Tenney, 1969)(Manning, p. 193). The piece is an exploration of noise tones with varying amplitude envelopes, bandwidths, and frequencies, and attemps to emulate the sonic environment Tenney had experienced on his commute. After discovering Tenney's *Computer Music Experiences, 1961 – 1964*, which details the noise study's instrument design and use of randomization, this seemed like an excellent piece on which to base a similar investigation into the mapping and audification of Enigma-derived pseudo-random sequences.

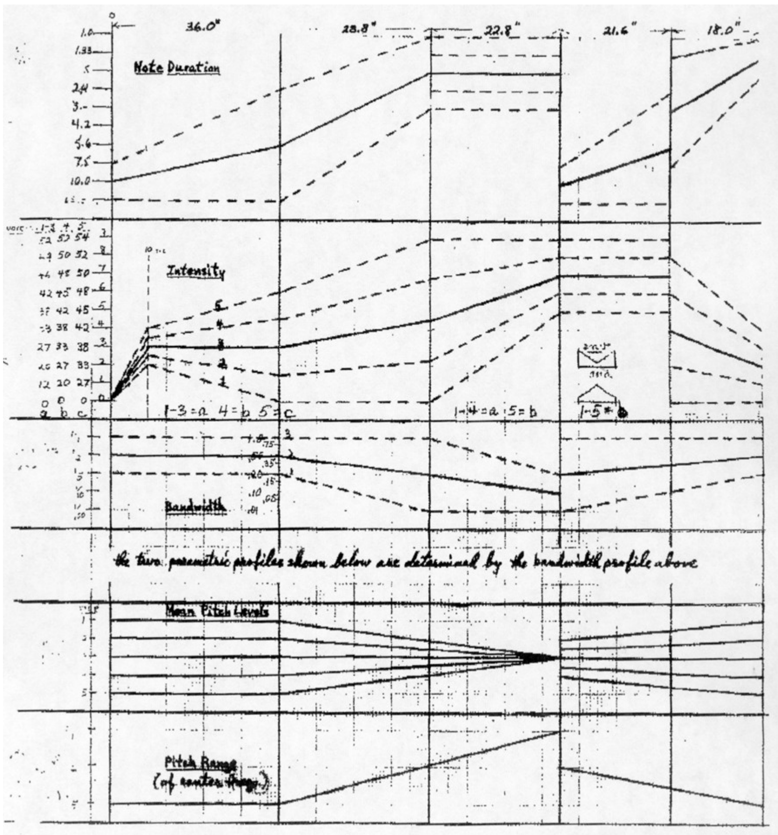

Figure 34 Formal outline of Tenney's noise study (Tenney, 1969, p. 9)

Furthermore, the programming of *ENIGMA-N* corresponded with my involvement with MODCAF's *Test Chamber* project at Farnborough Wind Tunnel, in which I would be performing live electronic music to be recorded for release. During the initial phase of experimentation, two particular occasions allowed me to form a connection between Tenney's piece and the wind tunnel. First, planes taking off from the airport next to the wind tunnel caused a distinctive rumble throughout the tunnel not dissimilar to lowpass-filtered noise and the opening of the noise study. Secondly, the sine sweeps being generated by the technical team to measure the frequency response of the tunnel recalled moments in the middle of Tenney's noise study where the noise tone is at its narrowest bandwidth.

The *ENIGMA-N* was programmed with real-time manipulation in mind so that it could be performed live in the wind tunnel environment, rather than following a strict pre-programmed structure.

Two recordings of this piece are included in the portfolio: A *studio* version accessed via the file `8_ENIGMA-N.aif`, and a *live* recording of the Test Chamber project recording, accessed via the file `10_Study_13_ENIGMA-N_Live_at_Q121.wav`. This performance features an initial section from *Study 13*, followed by a performance of *ENIGMA-N*.

Noise Instrument

The instrument used in this piece is the same as demonstrated in the `E4.enigma.tenney.pd` example patch, and employs the "continuous, linear interpolation between an initial and a final value (for each "note") in amplitude, bandwidth, and frequency" that Tenney (1961, p. 6) specifies in his noise study. This is not dissimilar to the interpolation between amplitudes used by Stockhausen in *Studie II* and in my *Study 13*. The *ENIGMA-N* patch uses the same technique of interpolating between values generated by the Enigma PRG but maps the various internal states to frequency and bandwidth in addition to amplitude and duration.

The patch is built up of four “voices”, each containing an Enigma noise instrument [7]and an Enigma PRG mapped to duration, initial and final centre frequency, initial and final bandwidth, and initial and final amplitude. The centre frequency is determined by the frequency of the carrier sine-tone, whilst the bandwidth is determined by the sample-rate of playback of the Enigma PRG data stored in an array. This data is read with interpolation using the *tabread4~* object to match the *randi* [8] object in Csound (Vercoe, 2023).

Mapping and Real-Time Control

The Enigma PRGs used for each voice are set up in a no-input encryption configuration and with a different letter-seed for each voice. Each voice follows the same global ‘clock’ with a range from 1ms to 1000ms per event. Altering the clock rate causes the various durations for each voice to fit within the same global time-structure. The duration and amplitude maps are similar to those employed in *Study 13*, except that all 26 duration values are used rather than splitting the alphabet evenly into notes and rests.

The frequency maps employed for both the sine-tone frequency and noise sample-rate follow a variable scale that is determined by an initial offset (lowest frequency produced) and the frequency range of the mapping. At the lowest and highest settings, the frequency modulation range of both are ~8Hz and ~12kHz respectively. Scaling is exponential so that pitches are evenly distributed across the frequency spectrum rather. To make the patch performable it is assembled to respond to MIDI CC messages for rate, centre frequency, bandwidth, volume, and the range of the parameter maps for centre frequency and bandwidth.

[7] Tenney’s noise instrument is recreated in the example patch E4.enigma.tenney.pd
[8] The *randi* object is observable in Tenney’s instrument diagram for *Analog #1* and is detailed in the Csound manual as generating “a controlled random number series with interpolation between each new number” (Vercoe, 2008). Given Csound’s close relationship to MUSICn, as mentioned in the manual, it is assumed that these objects are functionally identical.

Arrangement

Whilst the noise study follows a predetermined formal outline, the structure of *ENIGMA-N* was conceived through improvisation using the performance patch, which resulted in a rudimentary arrangement that allows for variation between performances. Figure 35 shows a graphical score that identifies the division of the piece into five sections, with notation for rate and ranges of frequency, bandwidth, and amplitude. Lines correspond to settings whereby the Enigma parameter maps have no effect on the noise instrument, with the greyed-out 2D shapes representing the range of modulation when applied. The sections go through several combinations of parameters that make up five distinct ‘sounds’. The arrangement itself explores the interpolation of the scaling of parameter maps between sections.

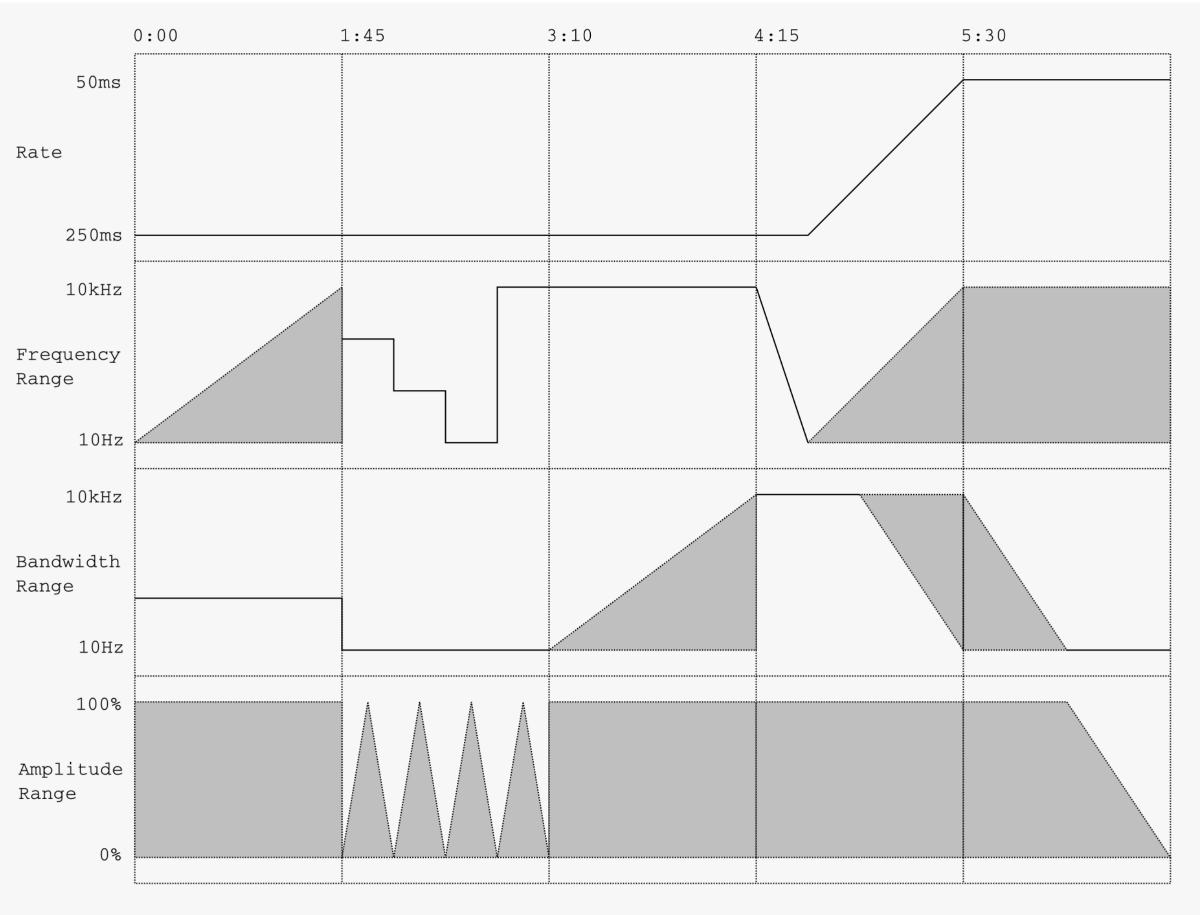

Figure 35 Structure of *ENIGMA-N*

Summary

Audification of Enigma PRG sequences is an effective approach to producing broadband noise comparable with typical noise generators. The periodicity of even the shortest sequences appears imperceivable when applied in the context of James Tenney's noise instrument. Interpolating between *no-input encryption* values further extends the creative range of the list_enigma object and the Enigma cipher in a musical context.

4.6 Ciphertext for Susan Merrick

This composition is a homage to Herbert Eimert's *Epitaph für Aikichi Kuboyama*. Produced between 1957-1962, Eimert's "composition for speaker and language sounds" is an electronic treatment of a spoken poem recording using the technologies and techniques of the WDR Studio in Cologne (Soundohm, 2012). Research on this work for tape appears sparse but the fundamental processes involved are recognisable upon listening, namely the use of tape editing and manipulation, reverberation, ring modulation, and filtering. Knowledge of the processes and equipment used at the WDR may supplement this aural analysis, with works and writing by Koenig, Manning, and Williams providing useful insight into the typical techniques used to realise *Elektronische Musik* in the studio.

Whereas the majority of the works in the portfolio apply parameter mapping to synthesized sounds, the use of speech sound in *Epitaph für Aikichi Kuboyama* inspired me to try a comparable approach with the Enigma cipher. By mapping the Enigma pseudo-random sequences to audio recordings of spoken letters, a stream of language sounds provides the sonic material on which to apply different audio manipulation techniques in a manner similar to Eimert's *Epitaph*. I collaborated with artist Susan Merrick to create the spoken material of the piece, which was edited into individual letter-sounds and implemented as a sample-playback patch in Pd automated by the Enigma cipher.

This was a valuable opportunity to experiment with Enigma control of external MIDI instruments, specifically the Make Noise *Shared System* which includes several devices roughly modelled on studio techniques from the era of the WDR.

Transformation Processes

Epitaph für Aikichi Kuboyama begins with a short spoken introduction, followed by an unprocessed recording of the poem used as the raw material, followed by a far longer arrangement of various combinations of audio processes and manipulations of the speech sounds. In my aural analysis I detected several recognisable audio processes: ring modulation, pitch-shifting/time-stretching, reverberation, filtering, and manipulation of amplitude envelopes. Furthermore, the processed sounds themselves appear to be cut up and arranged in a specific order, similar to Stockhausen's *Studie II* mentioned earlier. This follows Koenig's observations of the practices in electronic music studios at this time, whereby "the initial sounds are subjected to various transformations; for instance transposition, filtering, chopping, frequency modulation, amplitude modulation, ring modulation, reverberation and so on. Finally, the completed sounds are assembled to form the piece, whereby each sound, according to the rhythmic structure of the score, must occur as a certain point in time" (Koenig, 1965), p. 3). Manning (2004, p. 39-67) confirms the implementation of various forms of filtering, amplitude modulation, and reverberation at the WDR Studio and Eimert's use of a special device used to achieve independent manipulation of the pitch and speed of recorded sounds.

The following section details the processes used in *Ciphertext for Susan Merrick* in relation to the same processes used in Eimert's *Epitaph* and how these techniques are manifested in the Make Noise *Shared System*. Fig. n shows a diagram outlining the control and audio manipulation processes used in the work, divided into the Pd patch, the modular synthesizer elements (*Phonogene*, *Echophon* etc), and mixing console (summing).

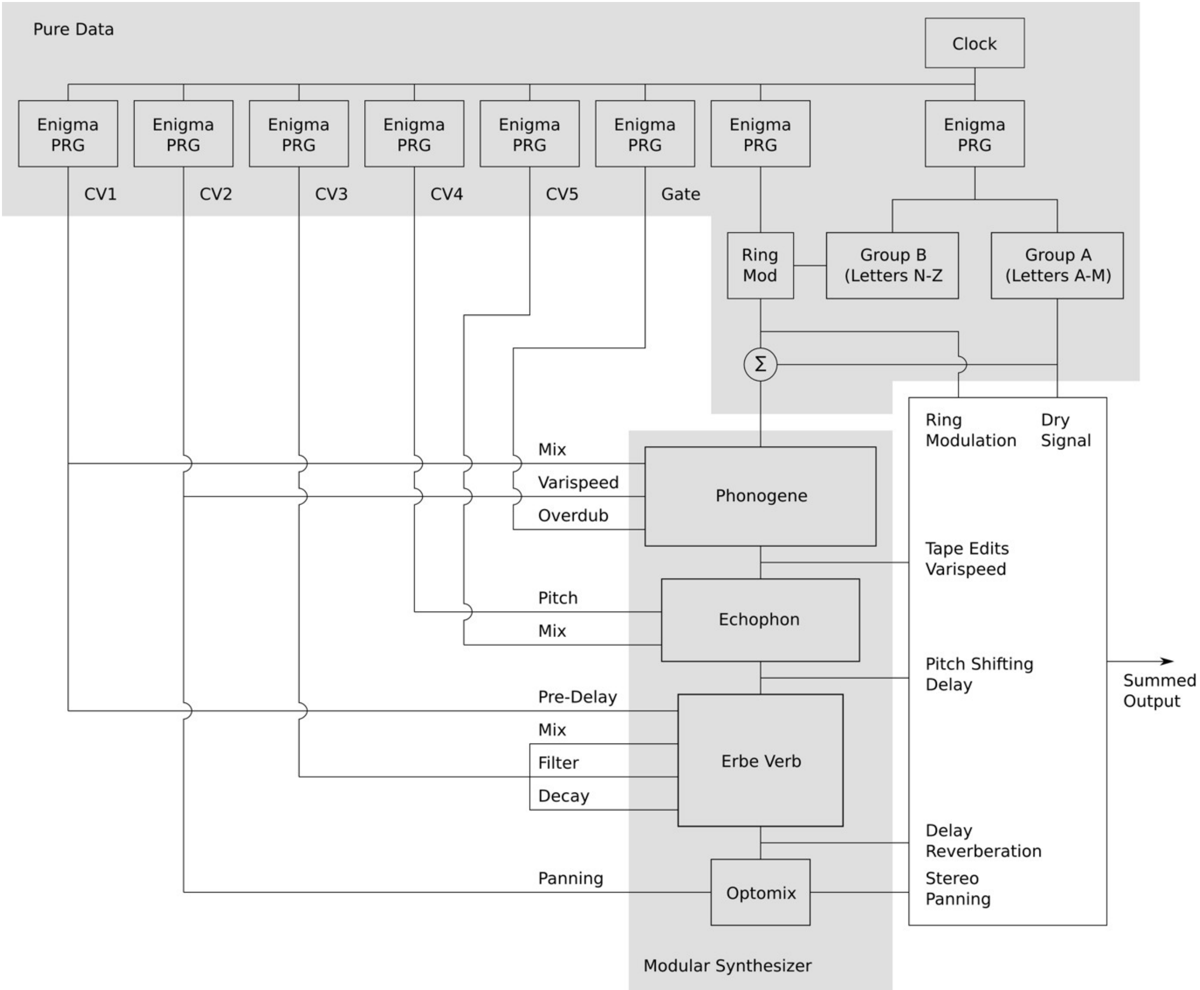

Figure 36 Flow diagram of processes in *Ciphertext for Susan Merrick*

Pd Patch Control

The Pd patch employed several Enigma PRGs configured so that they each operate with independent time-structures, synchronised by a global clock. Whilst a single Enigma cipher is able to produce ten pseudo-random sequences from its internal states, these advance simultaneously. By employing several ciphers to step independently, the number of combinations of parameter values and the complexity of the arrangement is increased. Furthermore, the aural effect has a “choppiness” that is similar to that of Eimert’s *Epitaph*.

A total of eight of the above Enigma PRG configurations are utilised; six for external manipulation of control voltage parameters in the modular synthesizer, one for control of the sine wave oscillator feeding the ring modulator, and one to sequence the spoken letters. A flow diagram of the processes used is given in figure 36.

The spoken alphabet is divided into two groups (letters A to M and letters N to Z) which are processed independently; the first half is left untreated, whilst the second half has ring modulation applied. Both the unprocessed and ring-modulated versions are sent to the mixing console and a summed version is sent to the modular synthesizer for further processing.

Modular Synthesizer Transformations

The modular synthesizer contains three modules that are useful to produce works inspired by the electronic music WDR era – The *Phonogene*, *Echophon*, and *Erbe-Verb*. This section will provide an overview of the three modules and explain how they were employed in the piece.

As stated by Tony Rolando (no date), “the PHONOGENE is a digital re-visioning and elaboration of the tape recorder as musical instrument … It is informed by the worlds of Musique Concrète (where speed and direction variation were combined with creative tape splicing to pioneer new sounds”. The primary function of the *Phonogene* in the context of *Ciphertext for Susan Merrick* is to simulate sound of different segments of audio spliced together on magnetic tape and played back at varying speeds and directions. This action is performed in real-time, with the overdubbing parameter sequenced via an Enigma PRG, in addition to control of *varispeed* and balance between the input and ‘tape’ sound.

According to Manning (2004, p. 71), among the tools available at the WDR Studio were “two variable-speed tape recorders … fitted with special devices that permitted not only normal pitch/duration transpositions but also, within certain limits, the variation of either of these basic characteristics independently”. This device “later became commercially available as the Spring Machine or Tempophon” (ibid) and is the inspiration behind Make Noise’s *Echophon* module (Rolando, no date). The *Echophon* is employed here to incorporate a pitch-shifting effect, which, when used in combination with the *varispeed* function of the *Phonogene*, allows for various combinations of independent manipulation of pitch and duration to take place. This follows the working principle of the “special devices” mentioned by Manning (ibid) as used to process speech in Eimert’s *Epitaph*. Two parameters of the *Echophon* are mapped to the Enigma PRGs: pitch, and blend between the input and pitch-shifted signals. In this instance, the latter is delayed by around one second to create a distinction between the two and increase the structural complexity of the piece.

As can be seen by Koenig (1965), Manning (2004) and Williams (2016), reverberation plays a significant role in the composition of *Elektronische Musik*. In *Ciphertext for Susan Merrick*, the Erbe Verb is employed in a similar fashion, with Enigma PRG sequences mapped to affect the pre-delay, tone, decay, and blend between wet and dry signals. The sequencing of these parameters gives a similar effect to that produced by the *Phonogene* – parameters jumping to new values giving the impression of tape splices. As with the *Echophon*, the use of pre-delay also creates further separation between the input and processed signals.

Performance and Arrangement

Whilst the sequence of spoken letters and manipulation of parameters is entirely determined by pseudo-random sequences, the balance between unaffected speech, ring modulation, chopped ‘tape’ sounds, pitch-shifted and time-stretch sounds, and reverberation, follows a structure whereby intensity and complexity build throughout the piece. This arrangement is realised through real-time performance of the strength of each of these effects on the original signal in addition to the balance between them in the summing stage. During the course of the piece, the rate at which the Enigma PRGs ‘step’ also increased substantially. Similar to *ENIGMA-N*, the arrangement follows a simple score that guides the piece, with room a degree of improvisation in response to the results:

```
0:00-0:30        Unprocessed + ring-modulated speech
0:30-1:25        Add reverberation
1:25-2:00        Add Echophon pitch-shifting
2:00-3:00        Add Phonogene “chopping”
```

3:00-3:25	Removed unprocessed + ring-modulated speech
3:25-4:25	Add automated panning
4:24-5:10	Crossfade to unprocessed + ring-modulated speech
5:10-5:30	Increase rate to noise spectrum
5:30-6:00	Manipulate filtering
	Crossfade to panned + processed sounds

Summary

As with *Study 13* and *ENIGMA-N*, the parallel sequences obtainable using *scrambler monitoring* allow for complex Pd patches to be created quickly without the need for multiple *random* objects. Combining multiple instances of list_enigma is an efficient way of achieving a significant number of related and unrelated pseudo-random sequences. The list_enigma object can easily be used to control external MIDI and analogue synthesizer equipment, offering the ability for Enigma PRG sequences to control music technology outside of the computer.

4.7 W

This was the last piece to be produced for the portfolio and represents a departure from the use of pseudo-random sequences and parameter mapping. Inspired by the minimalist drone works of French composer Eliane Radigue, *W* is a harmonic study on the internal wiring of Enigma machine wheels using digital wavetables and analogue filters. The piece is not based on any particular Radigue composition, but instead takes inspiration from her catalogue of works made using the ARP 2500 synthesizer. Furthermore, the compositional framework and realisation of *W* is informed by Radigue's interview with Tara Robert in *Pink Noises* (2010), which provides insight into her philosophical and technical approaches to electronic composition.

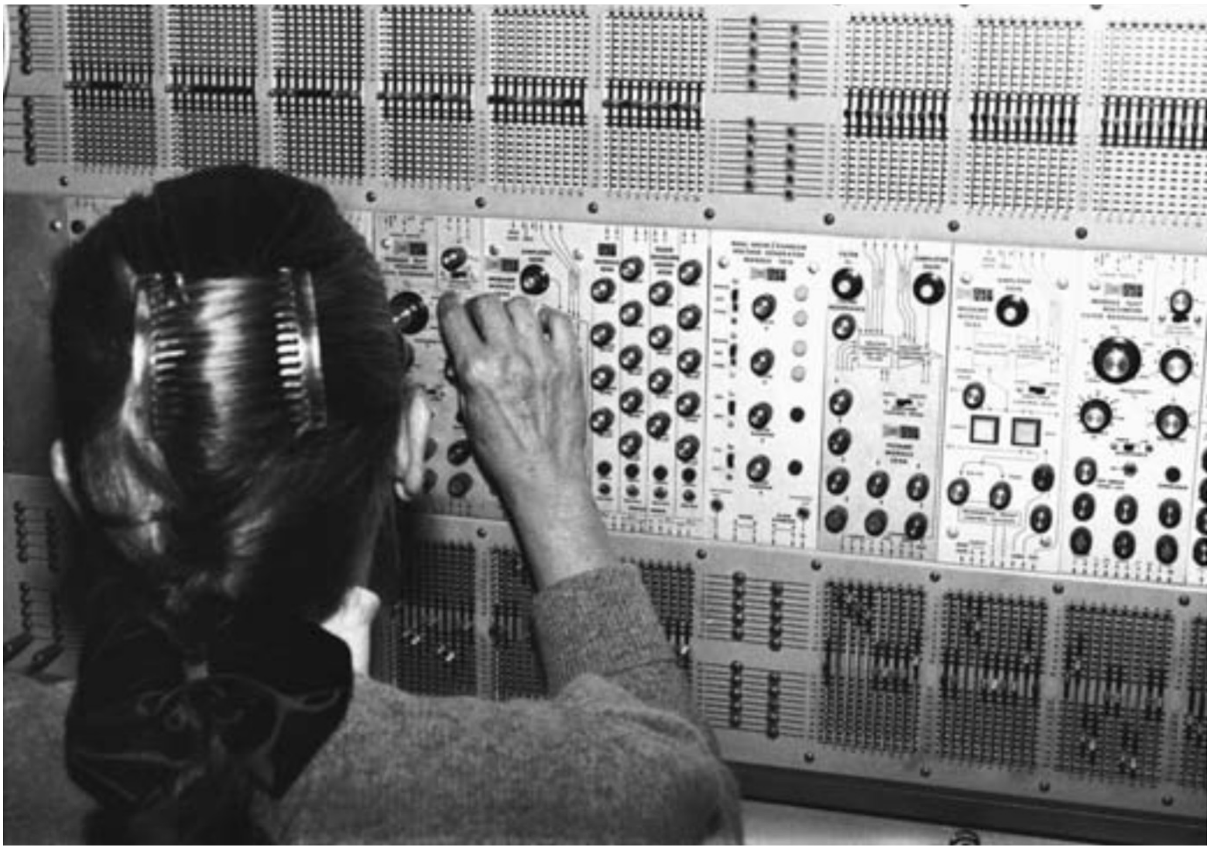

Figure 37 Eliane Radigue with the ARP 2500 (Rogers, 2010, p. 56)

Radigue's Approaches

Radigue's electronic works can be characterised by their use of continuous overlayed synthesized tones, spaced apart in harmonic and enharmonic relationships, and whose spectral contents and volumes are subjected to a slow evolution throughout the duration of the piece. Composition lengths are often in the vicinity of an hour in duration, such as her *Adnos* series and *Trologie de la Mort* (Radigue, 2021), compromised of several shorter sections using the same basic composition material as a starting point. In *Pink Noises* (Rogers, 2010, p. 57), Radigue outlines her method of composition with the ARP synthesizer:

"In analogical processes you have, of course, your oscillators. The Basic frequency on the five oscillators is the same for the whole piece. When I decide that, I work the whole piece on that. And then come all the ring modulations, frequency modulations, amplitude modulations–and you have potentiometers where you can change the proportion, the ratio. And it is only on that, that I work. I don't work on the oscillators;

when I have made my adjustment for one piece, it says there. I work after that on all the partials. And of course, to end it up, the two beautiful filters on the ARP really make it. Everything is worked out with the potentiometers on the ARP. There are about thirty parameters that you can change easily, that you can change without having a complete disappearance–you know what I mean, with these sounds, if you go too far it just becomes anything else, just like lots of effects."

Radigue (ibid, p. 57) identifies the mixing stage as a simple "fading in, fading out, crossfading" between the shorter sections within the work, however, the exact proportions and timings of these processes are considered crucial in the piece's execution. The sections themselves each present a slow exploration of the oscillator tones through careful adjustments of the ARP's parameters, "with great insistence on the game of the partial, of the subharmonic and overtones" (ibid, p. 58). It is this gradual process of subtle auditory investigation that is reproduced with the Enigma wavetables in *W*.

Development and Workflow

Initial development of the piece led to the patch example `E6.enigma.drone.pd`, in which the Enigma wheel wirings given by Kruh and Deavours (2002) are stored in an array and employed as droning wavetable oscillators. Since the wheel wirings change depending on which direction the signal path takes (as outlined in Chapter 3), the patch includes the capability to crossfade between the two wiring tables for the middle wheels and the ETW.

Radigue's admiration for the filters of the ARP 2500 and primary interest in harmonic exploration, demonstrate how the particular characteristic and capability of these filters is an intrinsic aspect of the work's creation. As Radigue exclaims, "the ARP filter is really the best I have ever heard. It's a very beautiful module" (Gluck, 2010, p. 48). During the development of the example patch, several filter topologies were investigated using both Pd's pre-packaged and 'raw' filters. I eventually settled on a non-resonant steep bandpass configuration using Pd's *rpole~* and *rzero~* objects, finding this to be suitably effective at sculpting the wheel wavetables and emphasising their harmonic complexities.

In practice, the experience of interacting with the Pd patch by slowly increasing or decreasing the value in Pd's number boxes felt cumbersome and lacked a sense of tactility. Similarly, the limited resolution of MIDI controllers did not offer enough fine control of filter settings whilst still offering the full range of frequencies. This tactile experience and particular response are further aspects of Radigue's work with the ARP that plays an important role:

"On the ARP, I found very slight changes, such as moving a knob very slightly, just a little touch here or there, could result in almost unnoticeably changes in a sound" (Gluck, 2010, p. 48).

Figure 38 Studio equipment used in the creation of *W*

As a result of my experiences improvising with the Pd patch, I began exploring a hybrid workflow, with Pd providing the wavetables oscillators and a combination of the Buchla *Dual Voltage Controlled Filter Model 291* and the stereo 31-band Yamaha *Q2031B Graphic Equalizer* for filtering the wavetables by hand. As these were the only filters available to me at the time, this meant that only four of the five Enigma wheels could be used in the piece. The four filtered wavetables were then mixed on an analogue console, where some further basic tonal shaping was carried out on its 3-band shelving/parametric EQ.

Summary

Whereas the previous compositions in the portfolio may be considered sonic embodiments or sonifications of the Enigma cipher algorithm and the random-like quality of its encrypted ciphertext, *W* is representative of something more ambiguous. Although the Enigma wheel wiring tables are made audible through the process of audification, the tables themselves are merely technical specifications rather than encapsulating the Enigma machine as a whole. When considering the complexity of the cipher algorithm, the perpetual cycling of these wavetables is perhaps more symbolic of the continuous rotation of Enigma wheels with machine-like relentlessness. The numerous approaches that these compositions take to interpreting the Enigma cipher algorithm can be seen as a parallel to the harmonic inspection of its wavetables for the purpose of hearing and experiencing the machine from a small selection of a wide variety of sonic perspectives.

Chapter 5 Conclusion

To conclude, this research project shows that the Enigma cipher is a reliable and flexible pseudo-random generator on par with those typically employed in the composition of electronic and acoustic music. In its simplest form, an Enigma PRG may be bought into existence simply by employing the *same letter stepping* technique. When employed in the ordering of musical events, the Enigma PRG produces sequences that are perceivably random but with the advantage of being repeatable. In practice, the alphabetical output and resulting resolution does not appear to be limited when employed in a typical musical contexts.

As with other Enigma simulations, the list_enigma object allows the user to monitor the internal states of the cipher system. In addition to the cipher text output of the cipher, these states may also be mapped to musical parameters, greatly expanding the potential of the Enigma cipher as a pseudo-random generator. Using the *scrambler monitoring* technique, it is possible to extract 10 rhythmically related pseudo-random sequences from a single cipher. By reading these states in series, we start to identify patterns in the sequence.

Whilst the periodicity of Enigma PRG sequences is identifiable through audification, the technique of *no-input encryption* may be used to extract pseudo-random sequences that are far longer than the Enigma machine’s cipher period. At typical sample-rates it is difficult to detect periodicity in the longer *no-input encryption sequences*. As a result, the Enigma PRG is just as capable in the role of noise generator as it is ordering musical events. Pd’s flexibility also extends access to the Enigma PRG to music technology outside of computer music environments, such as with external MIDI and audio devices.

Simulating the Enigma cipher natively in an open source programming language increases access to the list_enigma object negating the need for commercial software. Whilst the project presents a functional and accessible Enigma cipher simulation as a piece of music technology, the documentation of its research and development contributes to several research fields in an academic capacity. First and foremost, it offers a taxonomy of approaches to obtaining pseudo-random sequences utilising the Enigma cipher. This is presented alongside a detailed account of the cipher's working principle and references to prominent academic sources on the Enigma family of cipher machines. The exploration of alphabetical parameter mapping gives the future researcher a primer for using the Enigma cipher in a musical context. The application of non-standard tuning systems and the use of alphabetical pitch mapping indicates further exploration in the practice of musical cryptography within electronic music. Lastly, the list_enigma object demonstrates a potential for use in musical cryptography for Enigma's intended purpose for concealing information.

In summary, the research project is successful in achieving the aims and contributions as set out in the introduction, whilst presenting a number of possible avenues of future academic research.

Reference List

bartonmusic (2012) *Krell Musak 1.* Available at: https://soundcloud.com/user7621213/krell-muzak-1 (Accessed 30 August 2023).

Birkin, G. (2020) *SARS-CoV-2_LR757995_2b* | *Guy Birkin*. Available at: https://guybirkin.bandcamp.com/album/sars-cov-2-lr757995-2b (Accessed: 30 August 2023)

Cage, John. (2006) *Silence*. London: Marion Boyars Publishers Ltd.

Cedille (2023) *Easley Blackwood* | *Microtonial* | *Cedille Records.* Available at: https://www.cedillerecords.org/albums/easley-blackwood-microtonal/ (Accessed 30 August 2023).

Chiffriermaschinen Aktiengesselschaft (1924) Letter from ChiMaAG to the Swedish Military Attaché in Berlin, 13 November. Available at: https://www.cryptomuseum.com/crypto/enigma/b/files/ChiMaAg_19241114.pdf (Accessed: 30 August 2023).

Churchhouse, R. F. (2002) *Codes and Ciphers: Julius Caesar, the Enigma, and the Internet*. Cambridge: Cambridge University Press.

Clarkson, A. (1960) (review of Nr. 3, Elektronische Studien, Studie II by Karlheinz Stockhausen), *Notes*, 17(4), pp. 644–646. Available at: https://www.jstor.org/stable/892405 (Accessed: 30 August 2023).

Crypto Museum (2023a) *Enigma Cipher Machine*. Available at: https://www.cryptomuseum.com/crypto/enigma/index.htm (Accessed: 30 August 2023).

Crypto Museum (2023b) *Enigma History*. Available at: https://www.cryptomuseum.com/crypto/enigma/hist.htm (Accessed: 30 August 2023).

Crypto Museum (2023c) *Enigma D*. Available at: https://www.cryptomuseum.com/crypto/enigma/d/index.htm (Accessed: 30 August 2023).

Crypto Museum (2023d) *About*. Available at: https://www.cryptomuseum.com/about.htm (Accessed: 30 August 2023).

Crypto Museum (2023e) *Enigma K.* Available at: ttps://www.cryptomuseum.com/crypto/enigma/k/index.htm (Accessed: 30 August 2023).

Crypto Museum (2023f) *Enigma*. Available at: https://www.cryptomuseum.com/crypto/enigma/working.htm (Accessed: 30 August 2023).

Cryptologia (2001) 'THE GLOW-LAMP CIPHERING AND DECIPHERING MACHINE: ENIGMA', *Cryptologia*, 25(3), pp. 168–173. Available at: https://doi.org/10.1080/0161-110191889860 (Accessed: 30 August 2023).

Dombois, F. and Eckel, G. (2011) 'Audification', *The Sonification Handbook*, Hermann, T., Hunt, A. and Neuhoff, J. G. (eds). Berlin: Logos Verlag. Available at: https://sonification.de/handbook/download/TheSonificationHandbook-HermannHuntNeuhoff-2011.pdf (Accessed: 30 August 2023).

Doornbusch, P. (2009) 'Early Hardware and Early Ideas in Computer Music: Their Development and Their Current Forms', *The Oxford Handbook of Computer Music,* Dean, R. T. (ed). Oxford: Oxford University Press.

Gaj, K. and Orłowski, A. (2003) 'Facts and Myths of Enigma: Breaking Stereotypes'. *Advances in Cryptology – EUROCRYPT 2003. International Conference of the*

Theory and Applications of Cryptographic Techniques. Warsaw (Poland), 4–8 May. Available at: http://dx.doi.org/10.1007/3-540-39200-9_7 (Accessed: 30 August 2033).

Grond, F. and Berger, J. (2011) ‘Parameter Mapping Sonification’, *The Sonification Handbook*, Hermann, T., Hunt, A. and Neuhoff, J. G. (eds). Berlin: Logos Verlag. Available at: https://sonification.de/handbook/download/TheSonificationHandbook-HermannHuntNeuhoff-2011.pdf (Accessed: 30 August 2023).

Gluk, B. (2010) ‘An Interview with Eliane Radigue’, *array: the journal of the ICMA 2009 – 2010*. Available at: http://www.computermusic.org/media/documents/array/Array09-10.pdf (Accessed: 30 August 2023).

Hamer, D. (1997) ‘Actions involved in the ‘double stepping’ of the middle rotor’, *Cryptologia*, 20(1). Available at: https://www.cryptomuseum.com/people/hamer/files/double_stepping.pdf (Accessed: 30 August 2023).

Hamer, D., Sullivan, G. and Weierud, F. (1998) ‘ENIGMA VARIATIONS: AN EXTENDED FAMILY OF MACHINES’, *Cryptologia,* 22(3), pp. 211–229. Available at: https://doi.org/10.1080/0161-119891886885 (Accessed: 30 August 2023).

Hull, T. E. and Dobell, A. R. (1962) ‘Random Number Generators’, *SIAM Review*, 4(3), pp. 230–254. Society for Industrial and Applied Mathematics. Available at: https://www.jstor.org/stable/2027716 (Accessed: 30 August 23).

Hutchins, B. (1977) ‘The ENS-76 Home-Built Synthesizer System – Part 8, Random Sources’, *ELECTRONOTES*, 9 (76), pp. 3–16 Available at: http://electronotes.netfirms.com/free.htm (Accessed: 30 August 2023).

Huygens-Fokker Foundation (2023) *The Bohlen-Pierce Site*. Available at: https://www.huygens-fokker.org/bpsite/ (Accessed: 30 August 2023).

International Clarinet Association (2019) ‘The Bohlen-Pierce Clarinet: Exploring a New Tonality’, *The Clarinet*, 46(3). Available at: https://clarinet.org/the-bohlen-pierce-clarinet-exploring-a-new-tonality/ (Accessed: 30 August 2023).

Jensen, M. G. (2009) ‘John Cage, Chance Operations, and the Chaos Game: Cage and the “I Ching”’, *The Musical Times,* 150(1907), pp. 97-102. Available at: http://www.jstor.com/stable/25597623 (Accessed: 30 August 2023).

Koenig, G. M. (1963) *The Construction of Sound*. Available at: https://koenigproject.nl/wp-content/uploads/2019/08/The_Construction_of_Sound.pdf (Accessed: 30 August 2023).

Koenig, G. M. (1965) *The Second Phase of Electronic Music*. Available at: https://koenigproject.nl/wp-content/uploads/2019/08/The_Second_Phase_of_Electronic_Music.pdf (Accessed: 30 August 2023).

Kruh, L. and Deavours, C. (2002) ‘THE COMMERCIAL ENIGMA: BEGINNINGS OF MACHINE CRYPTOGRAPHY’, *Cryptologia*, 26(1). Available at: https://doi.org/10.1080/0161-110291890731 (Accessed: 30 August 2023).

Ligeti, G. (1975) ‘Decision and Automatism in Structure 1a’, *DIE Reihe*, 4, pp. 36-62.

Maconie, R. (2016) ‘‘Stockhausen’s “Elektronische Studien I and II”’, *The Musical Times*, 157(1936), pp. 23–35. Available at: https://www.jstor.org/stable/44862534 (Accessed: 30 August 2023).

Manning, P. (2004) *Electronic and Computer Music*. Oxford: Oxford University Press.

Mathews, M. V. *et al*. (1988) ‘Theoretical and experimental explorations of the Bohlen Pierce scale’, *The Journal of the Acoustical Society of America.* 84(3).

MATRIXSYNTH (2012) *Krell Muzak by Todd Barton on Buchla*. Available at: https://www.matrixsynth.com/2012/08/krell-muzak-by-todd-barton-on-buchla.html (Accessed: 30 August 2023).

Miranda, C. (2016) ‘Decoding the Schumann Cello Concerto’, *The Musical Times,* 157(1934), pp. 45–66. Available at: https://www.jstor.org/stable/44862487 (Accessed: 30 August 2023).

MODCAF (2023) *TEST CHAMBER* **|** *MODCAF*. Available at: https://www.modcaf.org.uk/testchamber/ (Accessed: 30 August 2023).

Pallocks, D. (2013) *Universal Enigma – Simulator by dp*. Available at: https://people.physik.hu-berlin.de/~palloks/js/enigma/enigma-u_v20_en.html (Accessed: 30 August 2023).

puredata.info (2023) *Pure Data – Pd Community Site*. Available at: https://puredata.info (Accessed: 30 August 2023).

Radigue, E. (2021) *Adnos* **|** *Eliane Radigue*. Available at: https://elianeradigue.bandcamp.com/album/adnos (Accessed: 30 August 2023).

Radigue, E. (2021) *Trilogie de la Mort*. Available at: https://elianeradigue.bandcamp.com/album/trilogie-de-la-mort (Accessed: 30 August 2023).

RANDOM.ORG (2023) *RANDOM.ORG – Statistical* Analysis. Available at: https://www.random.org/analysis/ (Accessed: 30 August 2023).

Roberts, L. A. and Mathews, M. V. (1984) ‘Intonational sensitivity for traditional and nontraditional chords’, *The Journal of the Acoustical Society of America*, 75(3) pp. 952–959.

Rogers, T. (2010) *Pink Noises: Women on Electronic Music and Sound.* Durham: Duke University Press.

Rolando, T. (no date) *(soundhack Erbe-Verb manual)*. Available at: https://www.makenoisemusic.com/content/manuals/Erbe-VerbManual.pdf (Accessed: 30 August 2023).

Rolando, T. (not date) *(PHONOGENE manual)*. Available at: https://www.makenoisemusic.com/content/manuals/Phonogenemanual.pdf (Accessed: 30 August 2023).

Rolando, T. (no date) *(soundhack ECHOPHON manual)*. Available at: https://www.makenoisemusic.com/content/manuals/EchophonManual.pdf (Accessed: 30 August 2023).

Service, T. (2025) ‘How to get your hands on Total Serialism’s ne plus ultra’, *The* Guardian, 14 May. Available at: https://www.theguardian.com/music/tomserviceblog/2015/may/14/pierre-boulez-structure-1a-serialism-manuscript (Accessed: 30 August 2023).

Sound on Sound (1996) *PPG Wave 2.3 & Waveterm B*. Available at: https://www.soundonsound.com/reviews/ppg-wave-23-waveterm-b (Accessed: 30 August 2023).

Soundohm (2023) *Herbert Eimert – Epitaph für Aikichi Kuboyama / Sech Studien – Soundohm*. Available at: https://www.soundohm.com/product/epitaph-fur-aikichi-kuboy-1 (Accessed: 30 August 2023).

Strange, A. (2022) *Electronic Music: Systems, Techniques, and Controls.* 2nd edn. Toronto: Responsive Ecologies Lab.

Synthesis Technology (2023) *Synthesis Technology – The Eurorack E350*. Available at: https://synthtech.com/eurorack/E350/ (Accessed: 30 August 2023).

Tenney, J. (1969) ‘Computer Music Experiences, 1961 – 1964’, *Electronic Music Reports #1*. Utrecht: Institute of Sonology. Available at: https://www.plainsound.org/pdfs/ComputerMusic.pdf (Accessed: 30 August 2023).

Titsworth, R. C. and Welch, L. R. (1959) *Modulation by Random and Pseudo-Random Sequences*. Jet Propulsion Laboratory, Pasadena, California report 20-387.

Toop, R. (1979) ‘Stockhausen and the Sine-Wave: The Story of an Ambiguous Relationship’, *The Musical Quaterly*, 65(3), pp. 379–391. Oxford: Oxford University Press. Available at: https://www.jstor.org/stable/741492 (Accessed: 30 August 2023).

Vercoe, B. (2008) *Csound Reference Manual*. At: http://www.csounds.com/manual/ (Accessed: 30 August 2023).

Wik, A. (2018) ‘The First Classical Enigmas. Swedish Views on Enigma Development 1924-1930’. *HistoCrypt 2018*. Uppsala (Sweden), 18–20 June. Linköping: NEALT Proceedings Series 34, Linköping University Electronic Press, pp. 83–88. Available at: https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=149&Article_No=16 (Accessed: 30 August 2023).

Williams, S. (2016) ‘Interpretation and Performance Practice in Realizing Stockhausen’s “Studie II”, *Journal of the Royal Music Association*, 141(2). Taylor & Francis Ltd. Available at: https://www.jstor.org/stable/44862999 (Accessed: 30 August 2023).