MISTIFY: DYNAMICALLY RECREATING XENAKIS' *Mists*

Thor Kell[1]

[1] *McGill University*

Correspondence should be addressed to: thor.kell@mail.mcgill.ca

Abstract: **This paper presents an applied analysis of Xenakis' *Mists*. A late work for solo piano, *Mists* contains several stochastically generated sections. Squibbs has analyzed these in enough detail that it is possible to write Python code to dynamically recreate them. This method allows an examination of where Xenakis is using the stochastic methods discussed, and where he is deviating from them for his own compositional reasons.**

1. INTRODUCTION

Iannis Xenakis stands as one of the most important composers and technologists of the twentieth century. As the author of Formalized Music [1], he defined a highly personal musical language based on mathematics and probability theory. As an architect, he designed iconic buildings that are timelessly futuristic. As what we would now call an interface designer, he help create the UPIC system, one of the first graphical methods for composing music [2]. And all this to say nothing of the quality of his music.

Mists, written in 1980, is the third of the four piano works that Xenakis wrote. It is also the largest of his four piano pieces, and contains many of his key ideas: branching processes, stochastic composition, full-keyboard scales, and the spatial transformation of material.

Xenakis produced works for the piano at the rate of about one per decade - the piano is clearly not a typical instrument for him. Here, however, he uses it to create dramatic and dynamic textures that act as the basis of *Mists*. Indeed, Ronald Squibbs suggest that the form of the piece is based on changes of textures, between the opening runs, the so-called arborescences, and the large stochastic clouds of notes [3].

Mists was written after Xenakis' creation of the UPIC computational composition system in 1977, but was not written with this groundbreaking pieces of technology. To say it is not a 'technological' piece, however, does both the piece and the composer a disservice. The piece uses several advanced mathematical techniques to generate its material. These are discussed in detail by Squibbs in his Ph.D disseration [4] and in his 2002 article [3]. Squibbs' research has been absolutely core to this paper - I am deeply in his debt.

It was Berio who said that the best way to analyze a piece was to write another piece. Thus, the best way to analyze a Xenakis piece is to write code that will write the same Xenakis piece for you. Specifically, I will attempt to recreate the stochastic sections of the piece, using the Python[1] programming language. Rather than analyzing the stochastic processes in *Mists* using computers (along the lines of Burgoyne's thesis [5]), I will use a human analysis, by Squibbs, to inform the creation of automatic composition software. Ariza's work in recreating Xenakis' sieve methods in Python [6] is very similar to this, though Ariza focuses on a specific technique rather than on a specific piece. This method is perhaps closer to the analysis / synthesis techniques often used in electrical engineering: the quality of the analysis is defined by how close the resulting synthesis is to the original input. Appendix A contains some code samples - the entire script is available on the author's website[2]

I will focus on providing a detailed examination of when and where Xenakis is using the methods stated by Squibb's analysis, and where he appears to be modifying them to suit his compositional needs. This method allows us to make sophisticated guesses about where Xenakis is stepping away from his mathematics and making intuitive choices about composition.

[1]http://python.org

[2]http://tide-pool.ca/mists

2. *Mists* OVERVIEW

The key feature of *Mists* are the three main textures that it moves between. These are referred to by Squibbs as arborescences, continuous random walks, and discontinuous random walks. The arborescences are compound melodies, and are not stochastically generated. They thus cannot be programmatically recreated, and do not form part of this analysis.

The Continuous Random Walks (CRWs) occur in the opening three minutes and in the closing two minutes, and can be seen as notated in Figure 1. It is not made clear if these sections are mathematically generated or composed by Xenakis. This texture is discussed in detail in Section 3.

Figure 1: Continuous Random Walk texture, from the score.

The Discontinuous Random Walks (DCRWs) form the center of the work, from the third minute to the tenth minute. They can be seen in Figure 2. Xenakis even changes the notation (at mm. 41) to indicate the onset of this material. This texture is defined using stochastic methods. The recreation of these methods is discussed in detail in Section 3.

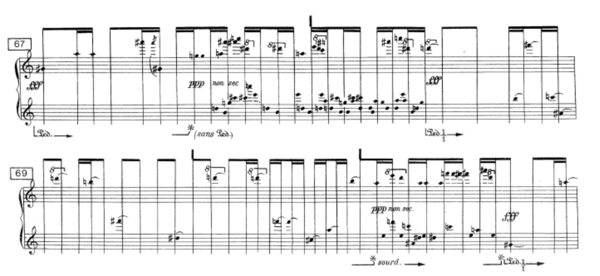

Figure 2: Discontinuous Random Walk texture, from the score.

Squibbs posits that the form of the piece is based on these texture changes, and further suggests an overall A (CRW) : B (DCRW) : A (CRW) form, punctuated by arborescences. The arborescences are not well served by this analysis: further investigation of them would be welcome.

The overall pitch content is based on a 30 note scale (Figure 3), and various transpositions of it. The placement of the transpositions furthers the idea of an ABA form: certain scales that appear in the opening appear again in the conclusion of the piece.

Figure 3: The initial scale used.

3. METHOD AND RESULTS

The Continuous Random Walks and the Discontinuous Random Walks were both recreated programmatically. The recreation of both the pitch and rhythmic content was attempted. Tempo, dynamics, and articulation are not discussed by Squibbs, and were not recreated.

3.1. Continuous Random Walks

Creating the the pitch content of the CRWs is relatively simple: select a note, select a transposition of the scale, select a direction, and go. The walk many change direction one or more times, or it may not.

The rhythmic durations are less simple. Squibbs shows an exponential ($y = e^{-0.163x}$) distribution (Figure 4) that is used to select rhythm, but does not make it clear if this distribution applies to the CRWs, or only the DCRWs.

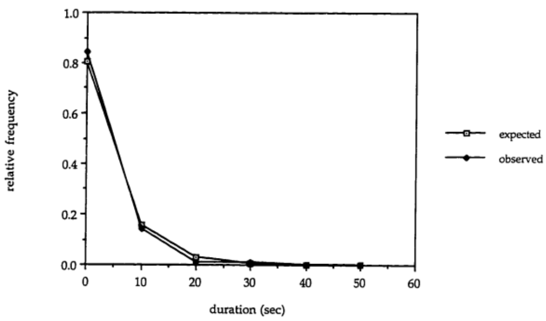

Figure 4: The exponential distribution used.

Comparing the graphical transcription (Figure 5) of the score with the generated CRW data (Figures 6 and 7) strongly suggests that the rhythms are not chosen using the above distribution. Looking at the notation bears this out: the rhythm accelerates and decelerates over time, suggesting perhaps a sine function.

(Due to issues with the MIDI generation code, all notes are held for the same amount of time - rhythm is represented by the time between note onsets, rather than note length.)

It thus seems that the CRWs are not stochastically generated - or if they are, the methods are not those used by the other sections of the piece. A discussion of the Discontinuous Random Walks follows.

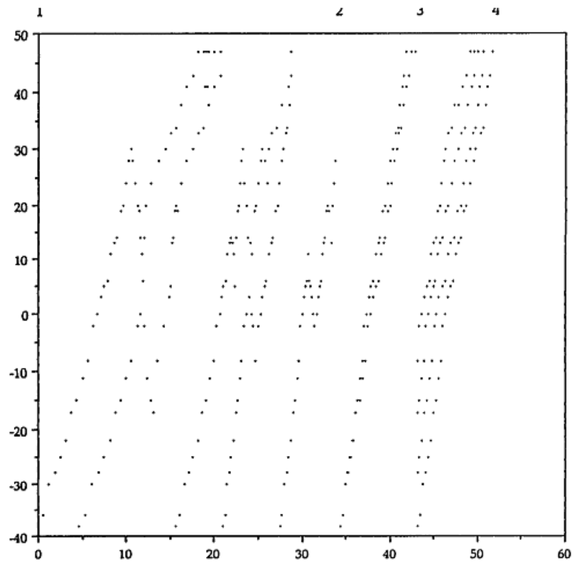

Figure 5: Continuous Random Walk texture, notated graphically by Squibbs.

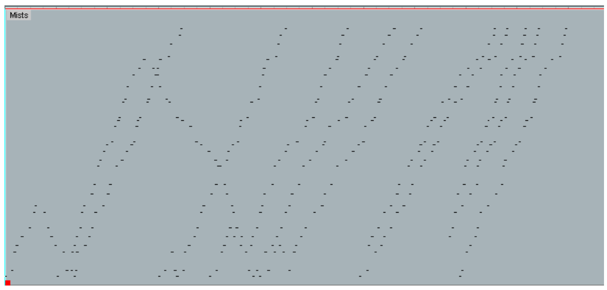

Figure 6: An attempt at generating the opening CRW.

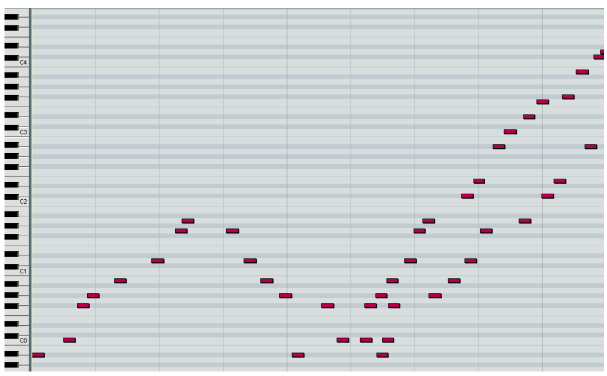

Figure 7: Detail of an attempt at generating the opening CRW.

3.2. Discontinuous Random Walks

The rhythmic choices here are created from the above exponential distribution (Figure 4). The shortest duration in the score is a 64th note, and the longest is a half note tied to a 32nd note. These two durations were used to bound the results of the exponential distribution. As will be seen, this gave excellent results.

The pitch choices are slightly more complex. Pitch is controlled by two distributions: a Cauchy distribution (Figure 8) and a hyperbolic cosine distribution (Figure 9).

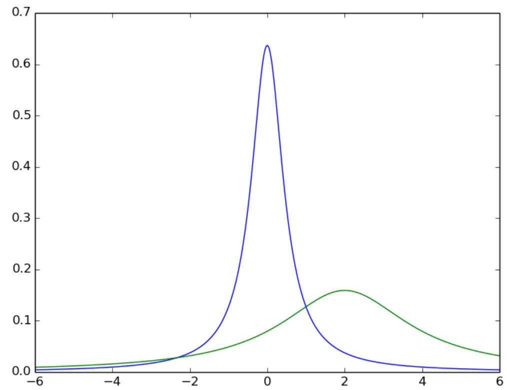

Figure 8: An example of Cauchy distributions.

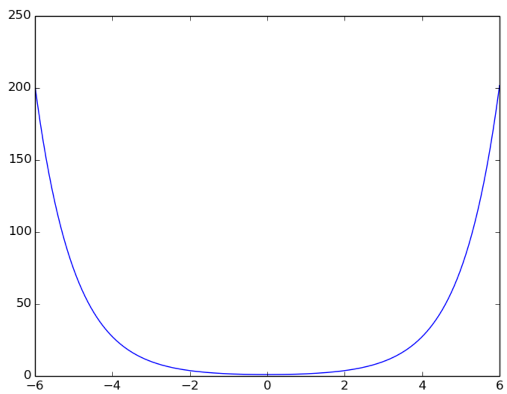

Figure 9: An example of a hyperbolic cosine distribution.

The result of these distributions is used to move up and down the scale relative to the current point. As can be seen, the Cauchy distribution will result in small steps, whereas the hyperbolic cosine will result in leaps. Squibbs does not note where Xenakis may have used each of these distributions, but he does note that in many places Xenakis is clearly applying some sort of registral control. Examples of the generated DCRW can be seen in Figure 11 and Figure 12 - compare to the graphical transcription in Figure 10.

The above results are quite good, but raise the key question of how Xenakis is controlling register in such a way. A trivial answer is that the result is simply down to the probabilities, and Xenakis made no other choices. A more satisfying answer is that he is sculpting them entirely intuitively, without recourse to any mathematics. A much more interesting answer is that Xenakis is varying the parameters of his distributions as he goes.

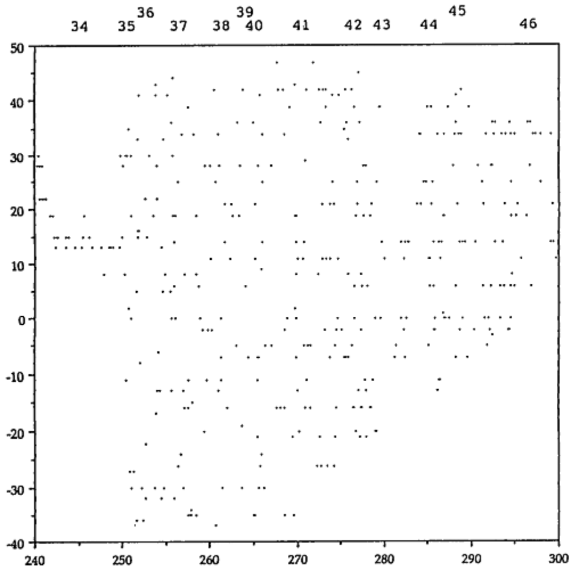

Figure 10: Discontinuous Random Walk texture, notated graphically by Squibbs.

Figure 11: An attempt at generating the middle DCRW.

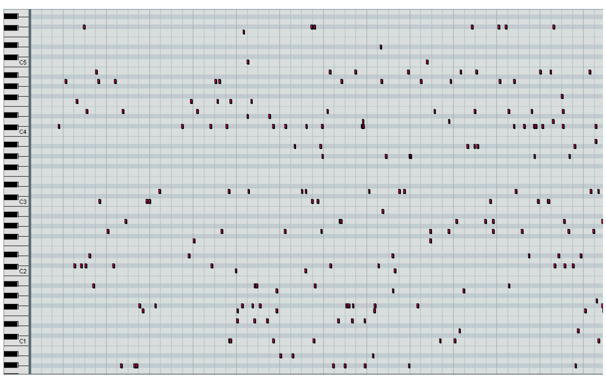

Figure 12: Detail of an attempt at generating the middle DCRW.

Consider the repeated high notes at mm. 117 (Figure 13). This could be achieved by setting up a very, very slim Cauchy distribution, such that it will always jump between the given two notes. Or, it may be two Cauchy distributions, both set so that they always continue to drone on their respective notes. Likewise, the descent at mm. 76 (Figure 14) could be done by offseting a Cauchy distribution so that it always returns negative step of some sort.

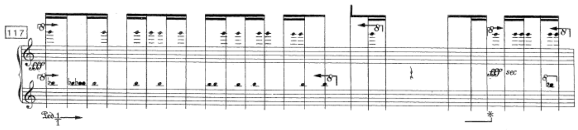

Figure 13: Measure 117, from the score

Figure 14: Measure 76, from the score

It must be noted that Squibbs provides no discussion of what Xenakis might be doing in these sections. It is *very* unlikely, however, that the results came out of the math without some sort of influence by Xenakis. The next section will continue this discussion, focussing on other areas and parameters where the results of attempting to generate the material is at odds with the material as written.

4. Other Areas of Interest

The transitional section, from mm. 31 to mm. 39, is defined by Squibbs as being part of the CRW section. This is somewhat belied by the number of changes of direction that the CRWs display. Attempts at recreating this were not ideal, as seen by comparing Figures 15 and 16

This may well be a lack of sophistication in my code, but it also points towards Xenakis using either more detailed mathematics, or using the mathematics as a basis for intuitive composition. Another example from this section is at mm. 39 (Figure 17).

Here, the two measures rising eight notes that lead into the DCRW section are not only not random rhythmically, but the notes played are the common tones between two of the transpositions of Xenakis' scale. Clearly, this moment was not generated stochastically. Or, if it was, Xenakis applied such strong control over the mathematics that he may as well have been picking the pitches and rhythms by hand.

Along these lines, other musical parameters are worth mentioning. The dynamics, articulations, and pedaling do not appear to be stochastically generated - or at the very least, Squibbs is not forthcoming about them. Given the textural nature of the piece, could this indicate that these parameters are actually the more important ones, rather than pitch and rhythm? Could Xenakis simply have generated relatively unimportant clouds of pitches in which to compose his articulations and dynamics? Or do all parameter share equal value, regardless of source? These questions are beyond the scope of this paper, but are worth mentioning to both contextualize the above discussion and comment on the limitations of the methods used herein.

Figure 15: An attempt at generating the transition CRWs.

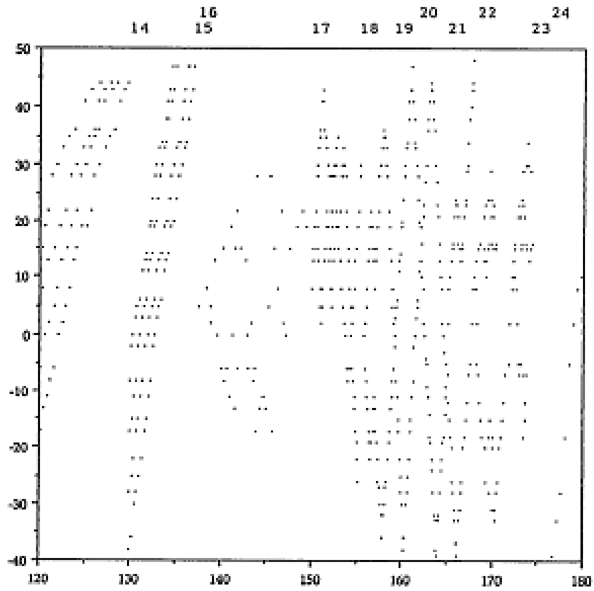

Figure 16: Transition CRWs, notated graphically by Squibbs (150 to 180).

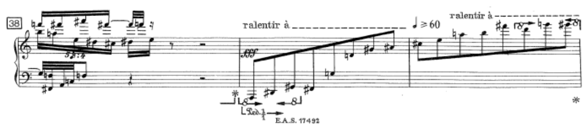

Figure 17: Measure 39, from the score

5. CONCLUSIONS

This paper has attempted to programmatically recreate several key passages from Xenakis's *Mists*, and use the reconstructions to highlight areas in *Mists* where Xenakis is meddling with his mathematics. These areas include sections of the piece that are explicable by Xenakis modifying the functions that generate his pitch material (mm. 76, for example), and sections where Xenakis is making such strong choices that his use of math or not becomes moot (mm. 39). If these bars were indeed stochastically generated, they were generated using such strict constraints that the composer may as well have written the notes down explicitly.

It must be noted that this technique is very reliant on the excellent analysis of Squibbs, and some of Xenakis' own writings about the piece. It stands as an adjunct method once a first analysis has been completed. Likewise, in this paper, many parameters are not discussed, as their potential mathematics have not been discussed in the source material.

What this method and paper do suggest, however, is where Xenakis was working outside of his formulae in *Mists*. In several sections of the piece, including the central "cloud" of discontinuous random walks, Xenakis is clearly applying more control to his parameters than can be explained by his own words or by Squibbs' analysis. This gives us a view into the intuitive workings of one of the most formal composers of the 20th century: where Xenakis the musical artist, rather than the musical mathematician, resides.

REFERENCES

[1] I. Xenakis: *Formalized music*. Indiana University Press, 1971.

[2] G. Marino, M.-H. Serra, and J.-M. Raczinski: *The UPIC system: Origins and innovations*. In *Perspectives of New Music*, pages 258–269, 1993.

[3] R. Squibbs: *Some observations on pitch, texture, and form in Xenakis' Mists*. In *Contemporary Music Review*, volume 21(2-3):91–108, 2002.

[4] R. J. Squibbs: *An analytical approach to the music of Iannis Xenakis: studies of recent works*. Ph.D. thesis, Yale University, 1996.

[5] J. A. Burgoyne: *Stochastic processes and database-driven musicology*. McGill University Libraries, 2011.

[6] C. Ariza: *The Xenakis sieve as object: A new model and a complete implementation*. In *Computer Music Journal*, volume 29(2):40–60, 2005.

6. APPEDIX A: SELECTED CODE

Mistify was written in Python, and requires the NumPy[3] library. For complete code, please see the author's website at tide-pool.ca/mists

```
def get_cauchy():
    result = 29 * numpy.random.standard_cauchy()
    return int(result)

def get_cosh():
    while True:
        x = (numpy.random.random() * 2) - 1
        y = numpy.random.random() * numpy.cosh(1)
        if numpy.cosh(x) > y:
            result = 29 * x
            break
    return int(result)

def get_exp(low, high):
    param = 0.163
    beta = 1.0 / param
    result = param * numpy.random.exponential(beta)
    while low > result or result > high:
        result = numpy.random.exponential(beta)
    return result
```

[3]http://www.numpy.org/