Converting the Voynich MS text to music

René Zandbergen

Version 2, 23 July 2024

Abstract

This paper describes a method to convert Voynich MS text to music. This is done just for entertainment purposes – it is not meant to suggest that this is what the text in the MS represents.

There are countless ways in which one can convert the Voynich MS text to music. In order to obtain a result that is at least somewhat pleasing, many properties of the Voynich MS text are taken into account into the selected conversion method. After a brief introduction, the various rules for this conversion are explained. This is based primarily on a new word model for the Voynich MS text by the author, which is being published in parallel.

Next, the paper presents a detailed mapping of Voynich MS characters to musical notes, and all additional rules that help to create music rather than a succession of notes. Finally, a link to several pieces of example music is provided. As this activity was undertaken for the purpose of the first annual international Voynich MS day (4 August 2024), this link will only be active from that date onward.

This paper is best fitted to those who already have some knowledge of Voynich MS text analysis and of music theory. It will be of no interest for readers who have neither.

1. Introduction

The Voynich MS is a late medieval hand-written book, which is famous because its text is in an unknown writing system, which cannot be deciphered. An example of this writing is shown in Figure 1. Much more information about the Voynich MS and its text can be found at the web site of the author [1].

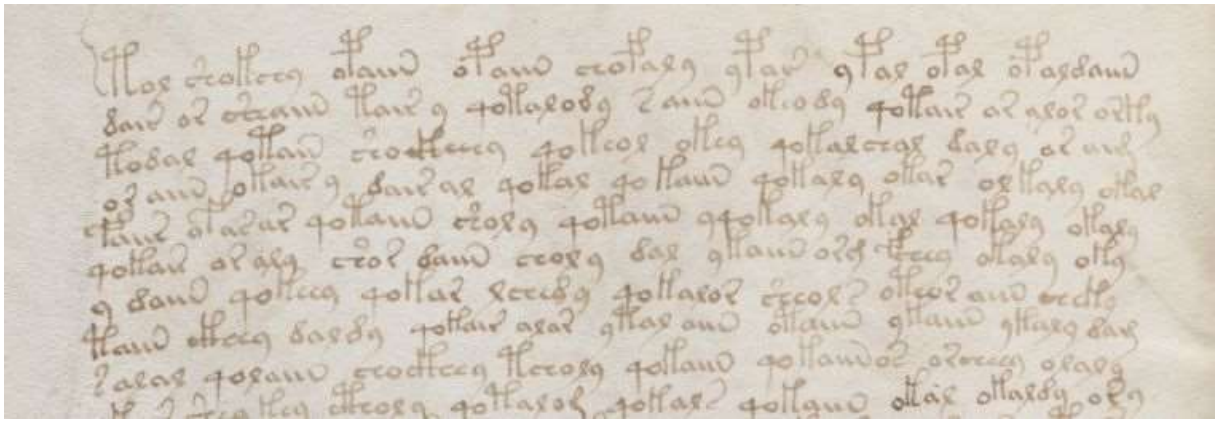

Figure 1: example of the writing in the Voynich MS

[1] See https://voynich.nu/

Many different proposals have been made for the meaning of the Voynich MS text. These range from an encrypted plain text to meaningless strings of characters, or even text written during a trance. Occasionally, people have also suggested that the symbols might encode music. While there have been several music compositions inspired by the Voynich MS, I am not aware of any concerted effort to create music directly from this text. The idea to do this was born during online discussions on how to organise the first international Voynich MS day, which will take place yearly on 4 August.

Clearly, there are innumerable ways in which one can convert the Voynich MS text to music.
If one isn't very careful, the result is likely to be a chaotic concatenation of dissonances. In the present activity, Voynich text analysis and some elementary music theory are combined in order to avoid that, and instead come up with something that one can listen to without too much discomfort.

2. Some initial decisions

At the start of this activity, a number of high-level decisions have been made. These are to some extent arbitrary. Certainly, many other approaches could have been chosen, and anybody inspired by the present activity is certainly welcome to explore that.

1. The line as a functional unit.

The fundamental observation of Prescott Currier in the 1970's, that each line appears to be some kind of a unit of information, will be at the basis for the approach [2]. Every line of text will be processed separately, and will result in one line of music. Consecutive lines will be separated by a short break. More specific consequences of this principle will appear in the following.

2. Ignore the labels

This perhaps disappointing decision is a direct consequence of the previous one. Labels are almost all single words and would lead to just a few individual notes each.

3. The various known properties of the MS text should cause audible distinctions

At this point this is just one of the aims of this activity. It is considered desirable that all of the following properties of the Voynich MS text should be discernable in the music:

- The overall language (A, B or C) used on the page [3]
- Words with or without initial ⟙
- Unattached finals [4]

4. Characters will be translated to the height of the notes, not their duration

The duration of the notes will be kept very simple, in order for the music to fit into a 4/4 bar system, as explained further below.

5. Characters will be mapped primarily to the 'white keys' of the piano keyboard

[2] See Currier (1976)
[3] A, B and C are the recently defined 'RZ language', for which see:
https://www.voynich.nu/extra/rz_lang.html
[4] The meaning of this term will be clarified in the following

This decision means that the music will be in the keys C major or A minor [5]. It will strongly reduce the emergence of dissonances.

6. The word spaces in the MS text will not, by themselves, be translated into music

They may play a role in the parsing of the text, but they will not lead to breaks in the music.

7. Note assignments will be based on a word model (word paradigm) of the Voynich MS text

The word model used for this pupose was recently developed by the author of this paper and is being published separately [6]. It is only summarised below. For the conversion to music, a number of simplifications introduced in that publication will be used here.

8. A second processing of each line will create a chord progression

Without this, the resulting music would be very monotonous and repetitive.

3. More on the line as a functional unit

The use of this principle means that the conversion to music of every line of text depends entirely and exclusively on the contents of that line. A convenient way to reflect the Currier/RZ language is to decide for a major or a minor key on the basis of this. This decision is made independently for each line, using the most obvious property, which is the occurrence of the bigram  in that line.

If this bigram occurs, the line will be in A minor, otherwise it will be in C major. For very long lines, typically circular texts, the criterium is that it should appear at least once for every 60 characters in the line, in order for the key to be minor. As a result, pages of the MS in the A language will typically be entirely in the major key, while pages in the B language will be mostly in the minor key. Pages in the C language will show a mixture of lines in the C major and the A minor keys.

In addition, lines ending in  or will result in a feature affecting the end of a musical line [7], which is called the “Picardy third”. This means that the line is played in a minor key, but the ending is converted to the major variant of this minor key, i.e. A minor becomes A major [8].

Finally, the ‘line as a functional unit’ puts a constraint on the chord progression, which is discussed further below under that heading.

4. On Voynich word structure

The apparent words in the Voynich MS follow some relatively strict rules about which characters can appear in which positions, or follow which other characters. This word structure has been analysed by different people in different ways. These attempts are described in some detail at the author’s web site [9]. They range from the very simple split of words into roots and suffixes by John Tiltman, to the far more complex word grammar by Jorge Stolfi. We will use a word model that has been derived

[5] Or any of the lesser known parallel modes

[6] René Zandbergen: A New Look at the Structure of Words in the Voynich MS (2024) – to be published

[7] If or appear elsewhere within the line, they have no special effect.

[8] This is explained with many examples in the following youtube video:
https://www.youtube.com/watch?v=jGaNdKabvQ4 .

[9] See: https://voynich.nu/a3_para.html

taking into account these earlier efforts, and which is a compromise between these two extremes, in terms of complexity and completeness. Its first publication is still being finalised [10]. This model is based on 'word chunks', where every word in the MS is either one word chunk, or a concatenation of two (or in rare cases more) word chunks. Figure 2 shows a simplified version of the word chunk model, that has been taken from that publication.

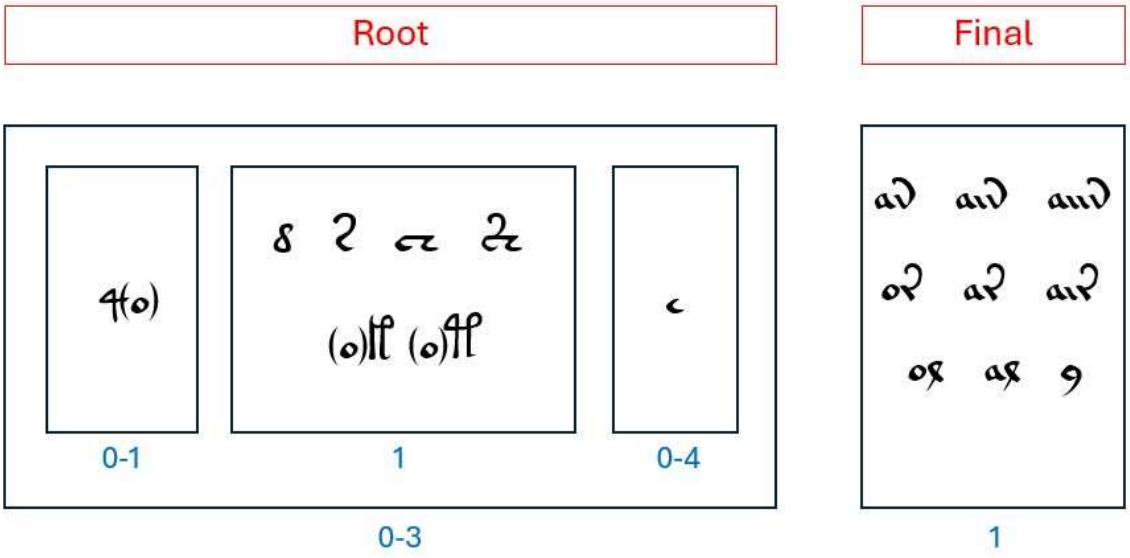

Figure 2: word chunk model from Zandbergen (2024)

This simplified model has been selected because it is more suitable for the conversion to music. The simplifications, discussed in more detail in the publication, may be summarised as follows:

- Some very rare characters have been replaced by more common ones: ꝛ → ſ , ꝝ → ꝛ , ꝭ → ꝺ , ꝯ → ꝰ , ꝫ → ꝛ , ꝟ → ꝟ [11]
  This only affects a total of 115 characters (out of approximately 195,000) , many of which appear in labels, which are not part of the music.
- Following a hypothesis by Lisa Fagin Davis, we substitute: ꝑ → ꝓ and ꝑ → ꝕ [12]
- In adition to this, we have simplifed all pedestalled gallows (the term commonly used for characters like ꝺ and ꝼ). While we do not know what these characters represent, we will treat them as ligatures as follows: ꝺ as a ligature of oꝕ , and ꝼ of oꝓ [13]
- The two characters ꝛ and ꝟ appear like line-ending variants or beautifications of some more simple variants. The character ꝛ and ꝟ have been converted to ꝛ and ꝺ respectively, except that they introduce the musical variant called 'Picardy third', as described before.

[10] Zandbergen (2024).
[11] Each of these substitutions has been chosen such that the character is similar, if possible, and the substitutions lead as much as possible to valid words
[12] See Fagin Davis (2022)
[13] With similar substitutions for other pedestalled gallows, see Zandbergen (2024)

- Words ending either with  or with  have received an additional trailing  . Replacing word-final or with essentially always leads to valid words, while words ending are relatively rare. For our musical purpose, there is no problem in changing words ending into the ending , instead of , in order to preserve some more variety.
- All and all non-word-final have been converted to . Word-final , which is not very common but also not rare, has been converted to . Furthermore, both and have been reduced to . With this, our set of finals is reduced to the list shown in Table 1, where each can be assigned a number as included in the third column in the same table.

Table 1: Summary of finals

Original finals	Reduced finals	Equivalent
		0
		1
		2
		3
		4

5. Complete word model of the Voynich MS text, for conversion to music

When parsing the Voynich MS text, by definition every word chunk ends with a final, but not every final is preceded by a root. We also find that a root may be followed by another root rather than a final. This may be summarised by noting that every word chunk consists of a final preceded by 0 or more roots. Cases where there are four roots appear to be very rare, so we may say that there are 0 to 3 roots. Each root may have its own trailing sequence of characters. There may be more than three characters in a row, but we will ignore any such additional characters, for reasons that will become clear later. Typically only the first root is likely to have a preceding , but this restriction is not necessary. This completes the description of the word chunk model shown in Figure *2*.

6. Bringing in the music

6.1 The basics

As indicated above, the Voynich music will be organised in bars of 4 beats each. Each beat is equivalent with a quarter note. As described above, word chunks may consist of 1 – 4 units, namely one final preceded by 0 to 3 roots. Each of these units will occupy one beat. In order to stay aligned to the bars, if the number of units is odd, the word chunk will be preceded by a rest of one beat.

In its most standard form, a word chunk consists of one root character and one final. The root character will be mapped to an absolute note (key on the piano keyboard) value, while the final is mapped to a key relative to the root. If there are more than one root character, each one will be mapped to its absolute key value, and the final is relative to the last root.

Any  characters trailing a root will be converted to notes stacked on the root note. How this is done is clarified further below. A preceding  will cause the root note to be split into two halves, i.e. two eighth notes, where the first is lowered by one step in the chromatic scale [14].

The similar characters  and  result in notes that are one octave apart. The characters  and  , which look like a pair of characters, result in two stacked notes each. The complete mapping of the root keys to a piano keyboard is shown in Figure 3, for a line of music that uses the major scale (C major).

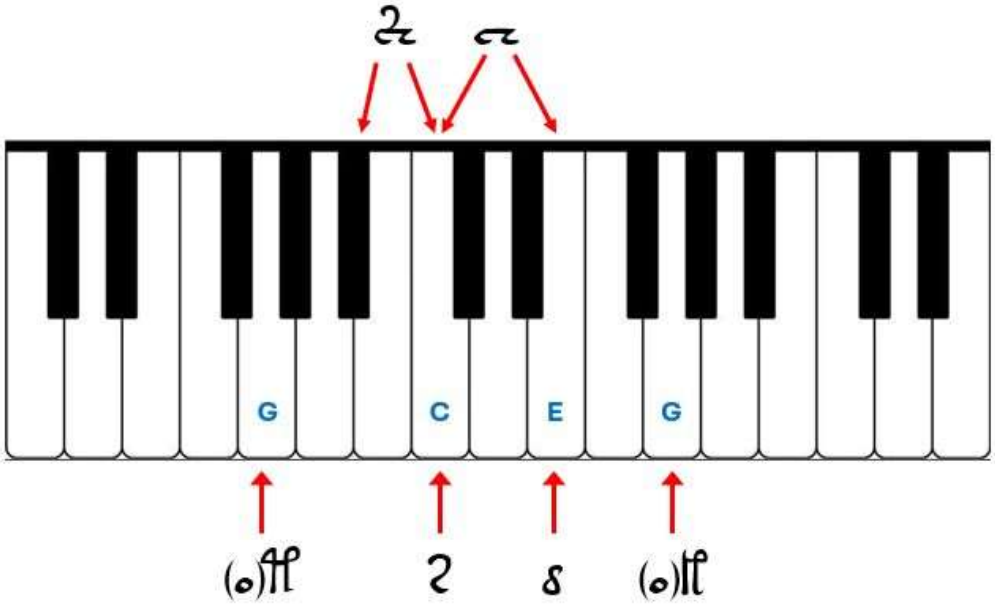

Figure 3: mapping of the root keys in a major scale

The character is the only character that results in a black key, which is the dominant 7th of the scale (B-flat in case of C major). For a line of music using the minor scale (A minor), all root characters are mapped to the white key that is two keys left of the one used for the major scale. In the case of , this means G + A .

[14] In layman's terms: it will be played by the black key just left of the root key.

The finals are played relative to the root notes. Referring to Table 1, if the final is represented by a 0, the root note is extended to a length of two beats. If not, the second note is one of: 2 down, 1 down, 1 up, 2 up, as shown in Figure 4. For and this jump is relative to the C. This method is valid for any root key (note), not just the C as used in Figure 4.

For unattached finals, there is no root key to offset from, and we will use the basis of the scale (C for major and A for minor).

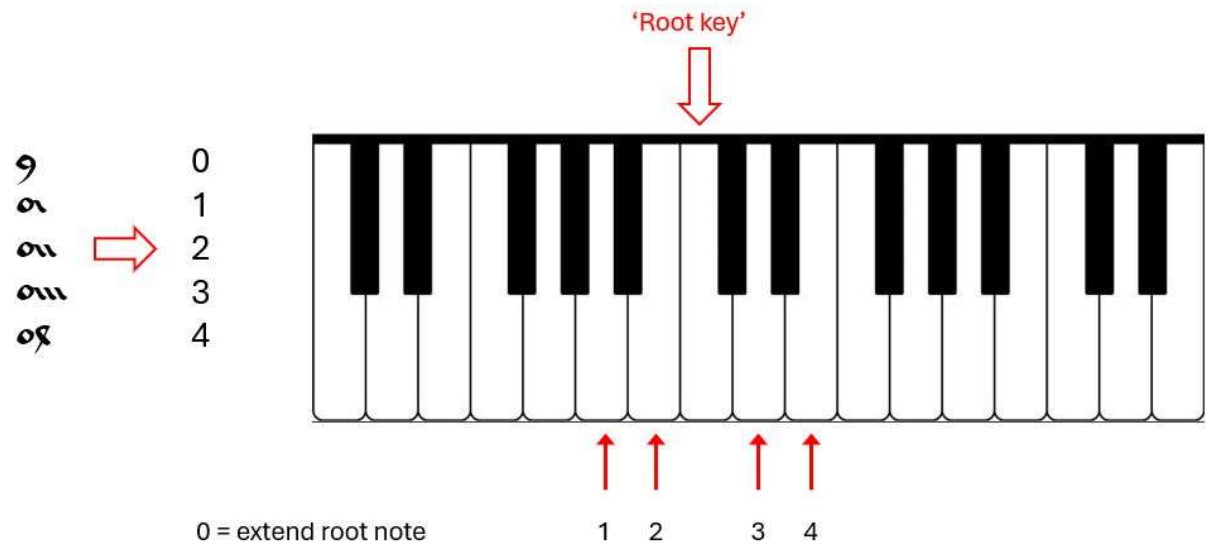

Figure 4: mapping of the final notes relative to the root

6.2 Stacking notes

Depending on the number of characters following each root character, 1-3 notes may be stacked on top of the root note [15]. In case the root character is or , the stacking is on top of the higher of the two notes. In initial experimentation, notes were stacked according to the standard chord formation practice of stacking thirds, meaning that the stacking would lead to 7th, 9th and 11th chords. It was found that a more pleasing result was obtained by stacking notes according to the following pattern:

- C → E → G → A → C (etc)
- In case the root note is and the scale is major, use B-flat (Bb) instead of A

6.3 Examples

All this is best clarified using some examples. To display music, we may use a method that is convenient both for musicians and non-musicians, namely the 'piano roll' that is commonly used by

[15] At most 3 trailing characters are used because stacking more notes would lead to less pleasant results.

sequencer programs. It has a piano keyboard on the left, and shows the height and duration of the notes to the right of it, in a matrix representation.

Figure 5 shows the translation of some standard words (root and final only) in case the key is major. This represents two and a half bars, and the grid size is a 16th note. One beat is one quarter note, so it spans four grid cells. The character 𐨈 maps to E, according to Figure 3. The second row of words give the Eva transliteration, and in the third row the finals have been mapped to a number.

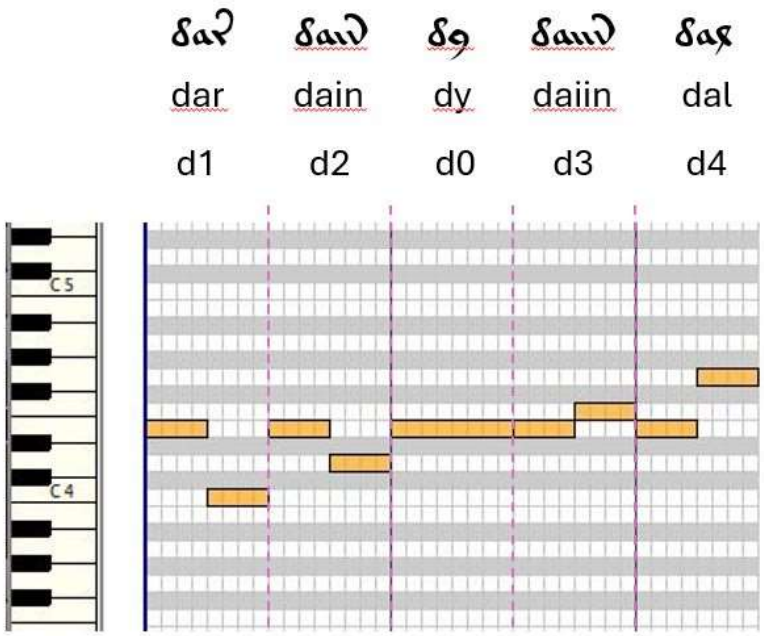

Figure 5: translation of some standard words to music

We may now do the same for some words including strings of 𐨌 and a leading 𐨟. In this case, it is assumed that the key is minor.

In Figure 6, the words 𐨀𐨟𐨙 and 𐨀𐨟𐨌𐨙 are still standard 2-note words, but 𐨀𐨟𐨌𐨌𐨈𐨙 and 𐨟𐨀𐨟𐨌𐨌𐨈𐨙 both have two roots, so 3 notes. They are therefore both preceded by a rest. These examples demonstrate the effects of stacking notes, and of the word-initial 𐨟.

Unattached finals would be represented by a rest of one beat followed by a single note of one beat, for the relative note of the final. This turned out to be quite undistinctive, even uninteresting, so the approach was adapted such that the second beat is split into two halves, where the first half gives the reference note, and the second half the offset note for the unattached final.

A special case is the unattached final 𐨙 (which in some cases originally was 𐨀), because this final is interpreted as prolonging the root note, but there is no root note. For the time being, this means that it is considered a null note. This may be reconsidered should there ever be an updated implementation.

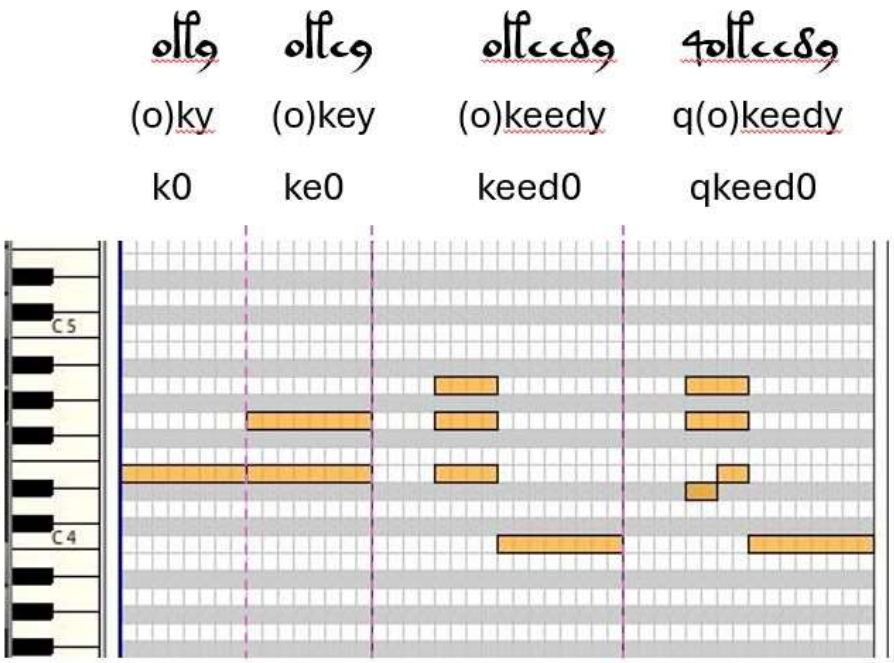

Figure 6: translation of some additional words to music

6.4 Chord progression

Each line of text in the MS results in a sequence of notes either in the C major or A minor scales. This would certainly lead to unattractive music, so it is of interest to add some variation by using additional chords. To preserve the strictly major or minor nature of each line, the chord progressions are limited to major and minor chords, and after some experimentation it was found that keeping it simple produced the best result. Chord changes are allowed along fifths, allowing to go up at most two fifths, and down at most one, both in absolute and relative sense [16]. This results in the possibilities shown in Figure 7.

Chord changes will be allowed at the start of each bar. Whenever there is a chord change, all notes in the corresponding bar(s) will be transposed by the same fifth as the chord change, in order to avoid dissonances.

The 'line as a functional unit' consideration places a constraint on the chord progression. Every line is forced both to begin and end with the basic chord (C major or A minor/major), while respecting the maximum relative change of going down one fifth.

[16] There are numerous youtube videos explaining the 'circle of fifths' but I found none that is sufficiently simple to recommend.

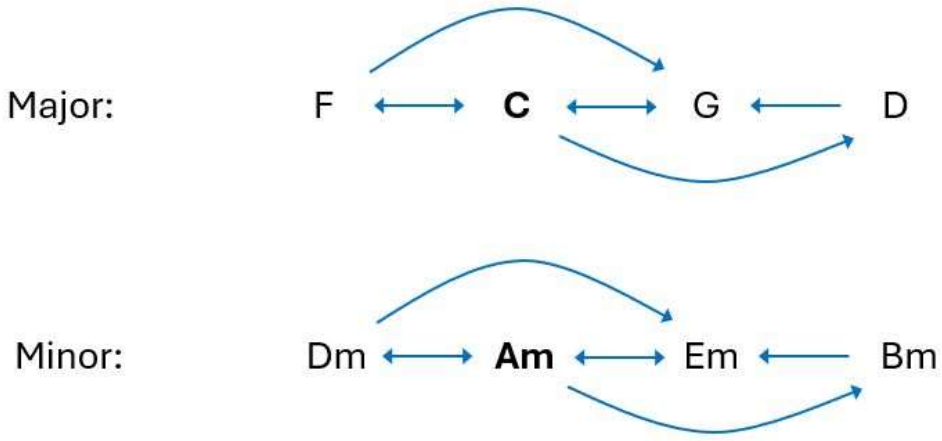

Figure 7: permitted chord changes

Chord changes are triggered by the last 'final' note in each bar. They are based on the absolute height of this final, which is the combination of the root note and the offset caused by the final. Table 2 shows the logic, where note nr. 1 represents the chord's base note (C or A, or other when already offset to another chord). Changes are only made when they don't exceed the maximum offset of +2 or -1 as mentioned above.

Table 2: chord change logic

Note:	1	2	3	4	5	6	7	8
Fifth change:	Down 1	Up 1	Up 2	-	Up 2	Down 1	Up 1	Down 1

Furthermore, changes are not made when the final is represented by a 0 (see Figure 4). This is overruled when there has not been a change in the last 4 bars.

7. Practical implementation

So far we have explained how the melody is generated from the Voynich MS text. In addition, we have a method to generate a chord progression. In order to create a pleasing piece of music, the chords can be played legato, arpeggiated, or both, and a bass line can be added. The bass line simply consists of the chord note on the first beat of each bar, and one fifth up, on the third beat. The accompaniment chords can be inverted in order to stay within user-defined limits [17].

Some very simple percussion is added as well. This consists of individual beats coinciding with the bass notes, with an emphasis when the melody has a rest. It also includes a distinctive sound to indicate the end of each line, as this tends to be quite abrupt. After all, we can be quite sure that the Voynich MS text was not really meant to represent music.

[17] This means that individual notes can be moved up or down an octave

The music is generated by a combination of dedicated and more generic software tools developed by the author:

1. A script to pre-process the Voynich text, as described in Sections 4 and 5. This script invokes the generic Voynich MS processing tools `ivtt` and `bitrans` as explained at the author's web site [18].
2. A dedicated tool to convert this text into notes, as described in Section 6 above. This tool has numerous options which were exercised during the development and experimentation. The result of running this tool is a text file that captures all note-on and note-off events for the melody, the accompaniment and the percussion.
3. A more generic tool to convert this text file with events into a standard MIDI file [19]. This allows selection of the instruments, volumes and a global transposition, among other things.

The MIDI files are ingested into the Ubuntu sequencer software 'Rosegarden' [20]. They are played on a Korg O3R/W expander in General Midi mode [21]. MIDI files are typically generated for one page of the Voynich MS.

Some example music has been generated. In this music, the melody is played on a harp and the accompaniment is played on legato slow strings and arpeggiated on another harp. The bass instrument is a jazz bass.

All music has been transposed down from C / Am to Bb / Gm.

The resulting music has been recorded and saved as MP3 files using the open source software 'Audacity' for Windows. A selection of MP3 files for different pages of the Voynich MS is available as from 4 August 2024, on the following web page:

https://voynich.nu/extra/voy_music.html

8. Acknowledgments

The Voynich MS is preserved as MS 408 at the Beinecke Rare Book and Manuscript Library of Yale University, New Haven (CT).

I am grateful to Marco Ponzi for his helpful review of an earlier version of this paper.

9. References and bibliography

Bennett, David: *Songs that use the Picardy third*. Youtube video: https://www.youtube.com/watch?v=jGaNdKabvQ4 (2020)

Currier, Prescott in: Mary E. D'Imperio (ed.): *New Research on the Voynich Manuscript*: Proceedings of a seminar, 30 November 1976. Privately printed pamphlet, Washington D.C., 1976. Available online: https://voynich.nu/extra/curr_main.html

Fagin Davis, Lisa: *Voynich Paleography*, Keynote paper presented at the 2022 Voynich MS Conference, Malta, 2022. Available online: https://ceur-ws.org/Vol-3313/keynote2.pdf

[18] See: https://voynich.nu/extra/sp_transcr.html#software
[19] MIDI = Musical Instrument Digital Interface. See References and Bibliography for the format specification.
[20] Using Ubuntu 22.04 and Rosegarden version 21.12
[21] Coincidentally acquired around the time that I first heard of the Voynich MS, over 20 years ago.

MIDI file format specification, version 1.1. Available online:
https://www.cs.cmu.edu/~music/cmsip/readings/Standard-MIDI-file-format-updated.pdf

Stolfi, Jorge: *A Grammar for Voynichese Words*, (2000). Available online:
https://www.ic.unicamp.br/~stolfi/voynich/00-06-07-word-grammar/

Tiltman, John H.: *The Voynich Manuscript, The Most Mysterious Manuscript in the World*. NSA Technical Journal 12 (July 1967), pp.41-85.

Zandbergen, René: https://voynich.nu (2004-2024)

Zandbergen, René: *A New Look at the Structure of Words in the Voynich MS* (2024) – to be published