Modeling, Listening, Analysis, and Computer-Aided Composition

*Silvio Ferraz Mello Filho
Universidade de São Paulo / FAPESP / CNPq
silvioferraz@usp.br
Orcid: 0000-0001-8263-6640
DOI: 10.46926/musmat.2021v5n1.116-125

Abstract: *The main objective of this short paper is to present results and a computational modeling process as a tool to aid musical composition. These results are related to the composition of the piece "Intermezzi and Capriccio" from a Fantasia, an ensemble of piano solo pieces worked from a cross analysis of the first Intermezzo of Johannes Brahms' Four Pieces for Piano Op. 119. In this way, the musical analysis I propose aims to compositional results, which in turn guides, the analysis itself. In this sense, the objective is not that of an analysis and modeling in order to reconstruct an original, but of analysis and modeling that allows it to unfold into new objects.*

Keywords: *Composition. Brahms. Musical modeling. Computer-aided analysis. Max/MSP. zl objects*

I. From analysis to patch design

First, it is important to present here what I am considering as musical analysis. I consider that all musical analysis is a fable. It is the invention of connection operators, sometimes evident or not. It is also important to say that I consider the analysis to be something not restricted to the score, and that it involves what I would call a listening scene: a heterogeneous field where music, space and performance are intermodulated. If there is a primacy of the score, it is because the whole space will be modulated by its realization; it would be the structured and repeatable domain of this plan.

For me, as a composer, analyzing is something like finding not only connection operators but also the series of nonsense that emerge in the listening fable and, from there, finding variables that allow the analyzed work to unfold outside itself. In this sense, everything in the scene of listening, performance, reading, participates in what we call the object of analysis.

It was in this sense that I programmed a series of patches, in Max/MSP environment, based on the "analysis of the score reading" of the first Intermezzo of Brahms' Op. 119, considering

*The Brahms analysis presented in this paper was part of the examples showed at the "Porto International Symposium on the Analysis and Theory of Music: Musica Analítica 2019", under the title "Repeating Difference: Music analysis and rewriting". Here, in this paper, I try to present more detailed aspects of the computer modeling used as example at Porto Conference. The author thanks the funding agencies FAPESP and CNPq.

Received: May 19th, 2021
Approved: June 2nd, 2021

constants in figural level (intervals, sense of phrases, harmony) and gestural (accents, realization speeds etc.), where each constant opens to a variable.

The proposal is based on studies carried out in the 1970s by André Riotte (composer) and Marcel Mesnage (computer engineer) . Mesnage and Riotte understand that the analysis must be "independent of a particular theory or style, consisting of progressively leading the initial complexity to a simpler combination of entities". I therefore observe that the more defined the material, the more formed and closed in precision relations, the harder the material for its compositional unfolding. In view of a compositional decision, the objective is not to find an "analytical elegance", such as the search for out-of-time symmetries, but processes that have the most open potential for unfolding.

The starting point of this method of analysis is to define the relevance of parameters and processes, and to observe whether they are pertinent to the whole work or just a section and if such pertinence concerns the work or a singular listening to this work. The focus thus is not analysis with an end in itself, but analysis as a compositional objective, with a view to automating the realization and updating of results. Automation here constitutes a prototype of the composition from which some compositional solutions can be maintained or changed.

The modeling carried out that I performed from Brahms' work was implemented in Max/MSP, and is based on the language of lists as present in software from the 1990s such as Patchwork and later OpenMusic. The list language used in modeling that I propose part of Lobjects and ListOps, a series of objects capable of operating with lists in Max/MSP that James MacCartney (ListOps) and Peter Elsea (Lobjects) programmed in the early 1990s to work with lists in Max/MSP. The zl Objects then allowed actions similar to some functions present in Open Music and Patchwork (and PWGL), with the possibility of working in real time of performance.

For the analysis stage prior to the implementation, I made use of a reading of objects and traces of the perceptible surface of Brahms' work, leading to a simple analysis which could kept the main affective elements of the first Intermezzo of Op. 119 by Brahms (first 15 measures are shown in Figure 1), with a compositional rewriting purpose.

1) *The slow descending cascades*: Brahms works with two interval sets in order to draw arpeggios and short melodic structures. One first, formed by the descending sequence of thirds (intervals 3 and 4) and another, also descending formed by seconds (intervals 1 and 2).
2) *The interval universe*: There is a clear interval circularity, an interval constancy and a cyclic set of transpositions that is extremely evident at the first glance at the score: a simple third reiteration algorithm (intervals 4 and 3) within the key you chose, a reiteration that implies the idea of alternation of the two intervals 4 and 3, being able to incorporate in the set of chords the augmented $(4 - 4 - 4(-4))$ or diminished $(3 - 3 - 3 - 3)$ sequences, as well as greater or lesser.
3) *The flow continuity always with slightly oscillating progress*: With regard to the temporal structure, Brahms reiterates sixteenth notes, which apparently means a constant, but in a letter to Clara Schumann notes that: "Every measure and even every note must sound like a great rhythm."
4) *The separation between two layers with the presence of a melodic line latent to the acute inflection of the resumes of the arpeggios*: In the specific case, the inflection and accentuation of the higher note (or notes) of the arpeggio enhances latent melodies, which also contributes to the positioning of the note at precise inflection points of the measure or even by prolonging its duration. A compositional strategy, constantly present in this and other works by Brahms.

This implementation step sought not to restrict the interval cycles to the tonal field and to consider only what I am calling the interval color used by Brahms. Out of tonality restriction, it was then possible to find a total of 16 possible permutations of thirds. In order to program a generative algorithm, a generated interval lists were designed to trigger sequences of intervals, with cycles that exceed the 4 intervals of the original and, from them, the sequence of notes is generated, calculated from an inflection note (high or low).

Aiming at a new composition born from the analysis of Brahms, the lists (interval, notes, time) can be transformed resulting in a gradual anamorphosis of the initial gesture (the descending cascades), in order to maintain the initial musical affection in another musical result. The main procedures employed in this regard are: (1) reversal of the arpeggio sense (acute-severe, severe-acute), (2) anamorphosis by expanding the intervals, (3) changes in time with variations around an average time, (4) redistribution of dynamic accents allowing the appearance of latent melodies through highlights, (5) increase or decrease in the interval cycle, (6) inflection note control from an external signal obtained by a musical instrument. One of the most radical elements of anamorphosis is the substitution of the inflection note by a group of notes, in case an arpeggio in a quick appoggiatura from a fragment of the same interval structure employed at the time of automation[1].

II. Patch steps and algorithm design

Even employing list objects provided for Max/MSP, this language which is quite developed in software such as PatchWork, PatchWorkGL and Open-Music, is nevertheless quite incipient in Max/MSP. There are some list libraries, but more complex operations such as those found in other software are not native to the program and do not cover all needs for musical programming. In this sense, a large part of the operations are carried out through the packaging and unpacking of lists with intermediate calculations.

The intervallic compositional thought found in Brahms is quite simple and is inherent in the play of arpeggios, but for compositional and post-tonal objectives it was, as I already said, considered only as a regular sequence of intervals, which can be changed in half step.

In the series of images below, I expose the Max/Msp modules in detail together with the interval analysis carried out from the original by Brahms:

1) Arpeggio sequence analysis, transformation in cascades, Max-MSP modules with the possibility of working at different intervals. In this way, in addition to the pattern of thirds $(3-4)$ taken from Brahms, the patch opens an interval variable that allows through a panel of cursors (multislider), to configure sequences of thirds, as well as seconds (direct opening up to tritons) or open intervals from one tenth. This mechanism has the function of both partially redoing the Brahms model and expanding it with compositional intent. In view of a limitation of tessitura to that of a normal piano, the patch delimits bass and treble with which when using very open intervals the result generated is of small descending or ascending cycles, thus comprehending the breadth of possibilities of the model learned from Intermezzo of Brahms (Figure 2).
2) Since the purpose of designing the patch was not only to perform an analysis of Brahms' work, but to create a compositional device from Brahms, it caught my attention when observing the arpeggios the possibility of converting cascades of more than five or six notes. In the figure below I present the arpeggios and short melodies (joint degrees) present in

[1] The developed patch allows for a composition in interaction between live instrumentalist (captured instrument or any MIDI instrument) and an automatic composition performed by an object of uninterrupted generation in irregular time cycles of inflection notes based on a simple attractor. For this mode of operation, attractors were also implemented for gradual and close variations of time, number of notes per arpeggio, dynamic variation.

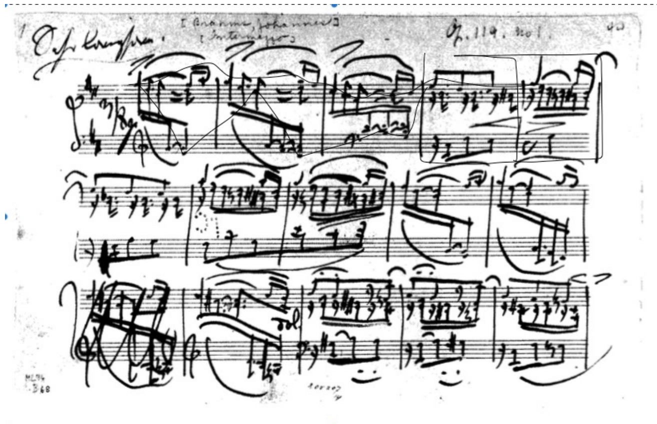

Figure 1: *Manuscript draft of the original from Brahms Op. 119.*

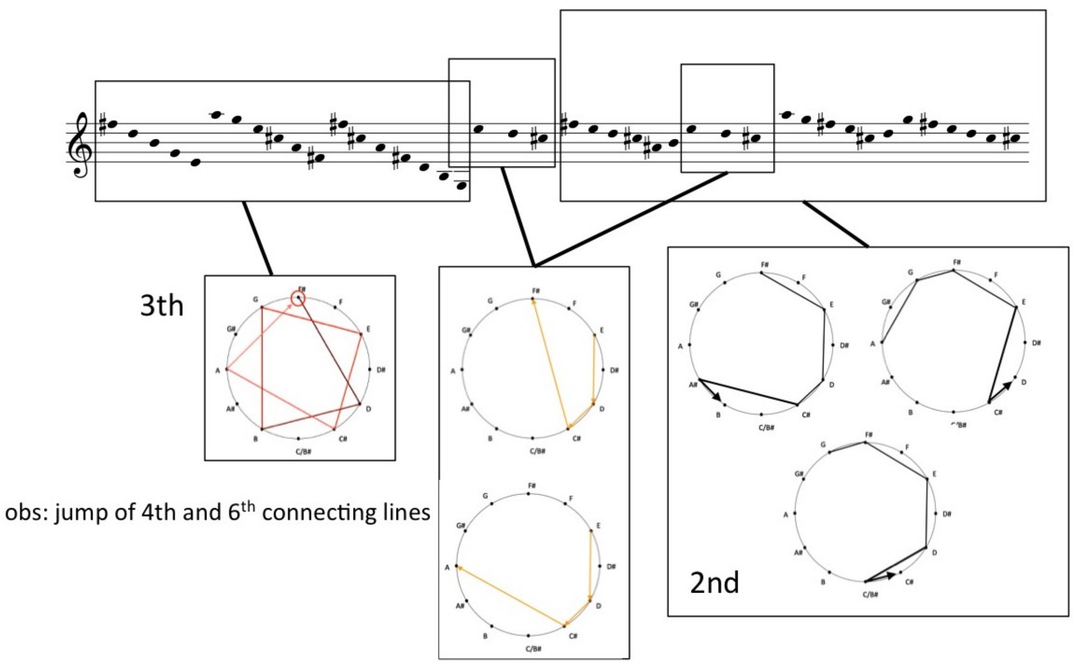

Figure 2: *The arpeggio sequences with chromatic clock analysis.*

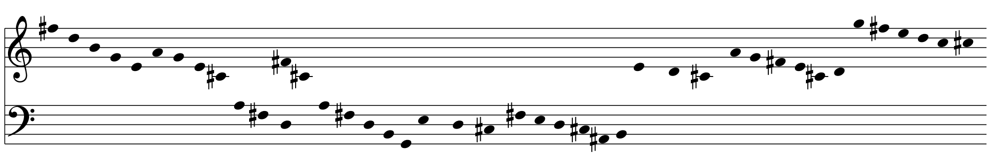

Figure 3: *The conversion of the arpeggios and melodic lines into descending cascades.*

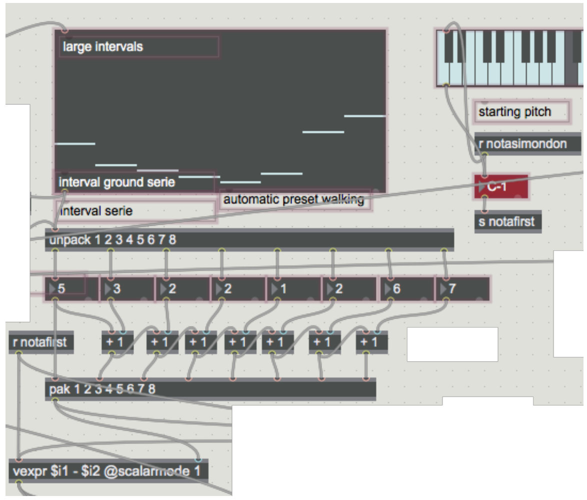

Figure 4: *The central interval module, multislider/intervals/plusminus module/sum from a first note (which can come from a MIDI keyboard or a pitch-tracker, allowing the patch to be associated with a signal input audio).*

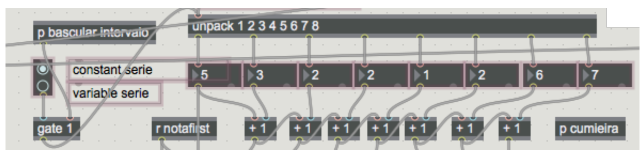

Figure 5: *Plusminus modules, which randomly add or subtract halftone to the intervals according to the arpeggios model that alternate intervals 4 and 3 and the diatonic melodic model that alternates intervals 1 and 2). In this module it is possible to work with a constant series of intervals or with a variable series via plusminus.*

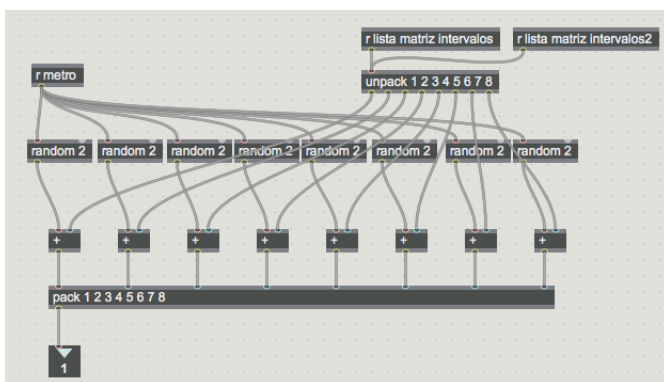

Figure 6: *Continuation of plusminus modules.*

Brahms and their conversion into cascades where some notes can be taken as pivots, like the G at the end of the first arpeggio, resumed as the beginning of the next and the F♯ at the end the second arpeggio also resumed to the octave as the beginning of the next (Figure 3).
Another important element in this analysis is the use of latent melody that Brahms performs in the upper voice. In this sequence I also observe an interval constancy and even a regularity and repetition mechanism. This data was used to draw the subpatch <cumeeira> (Figure 9),

which updates each new cascade sequence with a new high note according to interval regularity. From a cascade cycle calculated based on the first note, comes a new note that has a melodic interval relationship with that first.
For the design of this patch it was necessary to design a calculation subpatch x → dx and dx → x that allows calculating a series of notes from a series of intervals and vice versa, native operator in software such as OpenMusic and Patchwork (Figures 7 and 8).

Figure 7: *Converting arpeggios to cascades with transpositions based on supposed pivot notes.*

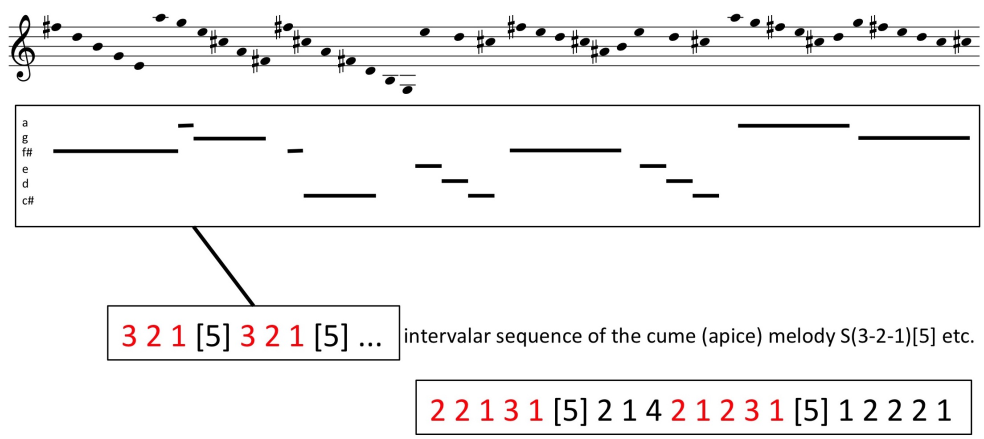

Figure 8: *Ridge (which has a correspondence with the <cumeeira> as commented above and in Fig. 5) notes that make up latent melody and its interval sequence.*

3) Still with respect to a compositional project resulting from the analysis and algorithmic implementation in Max-Msp, the patch comprises two important extensions of the original model. The first refers to extremely fast appoggiature performed either with an ascending model or with a descending one coming from the interval system towards the ridge note. In view of this direction, this change of trend of the cascade of notes, it was possible to determine an inflection point in relation to the note derived from the analysis of audio data or sending via MIDI. In general, this direction reversal module only reverses the reading order of the note sequence (Figures 4, 5, and 6).
4) When performing Brahms' modeling, it was of great importance to read Brahms' letter to the composer Clara Schumann, in which he specifically comments on this Intermezzo

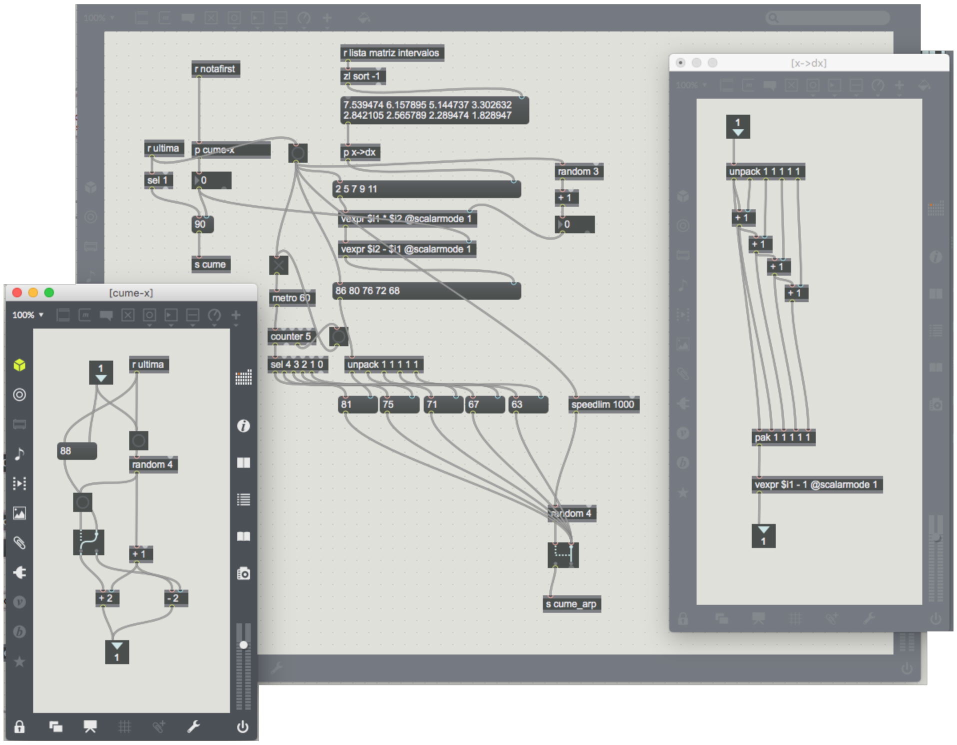

Figure 9: *Module (subpatch) [cumeeira] and its subpatches [cume-x] and [x → dx]): receiving and scaling (sort list) the list of intervals in the cascade, which will be applied to the new peak note from a calculation, performed on [cume-x].*

from Op. 119: "Every measure and even all notes must sound like a great rhythm". In this sense, I proposed to write the patch with two interchanging meter structures: stable meter and changeable meter. Through a checkbox it changes to a stable meter or to a changeable one, according to the result of the addition and statistical subtraction of values that make it possible to feel the change in tempo between notes or groups of notes. What is observed is the greater interest in the musical result. For the design of this and other timing devices in the patch I used the idea of pulse, which is expressed by the object <metro>, measure expressed by cycles counted via object <counter>, and which can be changed at any time, and rhythm, adding to the <counter> object a selector that defines which pulses should or should not pass and trigger the final note (Figures 10 and 11).

III. Final remarks

It is important to note as a conclusion that other elements were considered in the design of the patch, always keeping in mind that each data inserted from the score analysis behaves as a variable according all the time with the opening points that the Brahms system allows. Increase or decrease the number of notes of each downward current (Brahms arpeggio now 5 now 7 notes), modify the tempo until the point where the arpeggios become abrupt or even chords, and even the most unusual version of inversion of the direction (descending to ascending), expanding intervals to the point of working the edges of the piano tessiture (with great jumps) and dismantling the figures

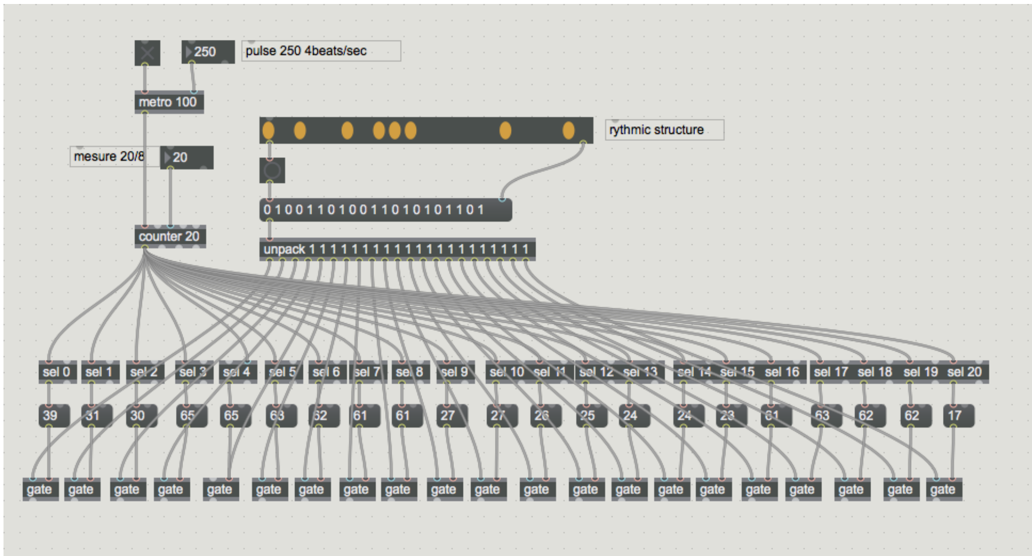

Figure 10: *Patch to change tempo (metro) and rhythmic structures.*

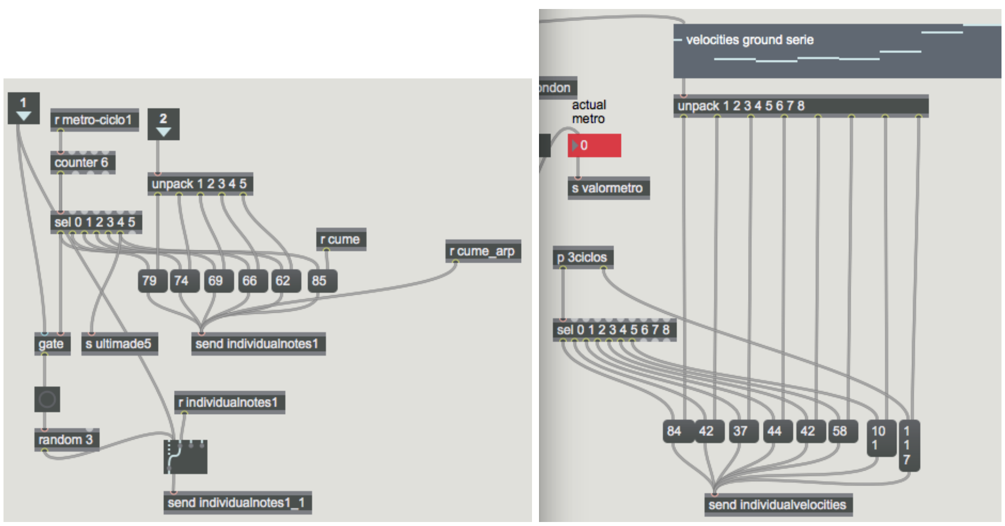

Figure 11: *Two modules from the patch created for modeling Brahms Intermezzo. The first example, related to one of the cycles (here with 6 notes) and the second related to the dynamic parameter that will be assigned to each new note in the arpeggio set.*

of descending cascades.

By having each note programmed as an individual element, it was also possible to control accents for each one of those notes, thus causing cycles of a certain size to be sectioned by ways of accentuating peak notes or even notes in meaningless positions. Such modifications allowed changes in the sonority of the piece, giving rise to the appearance of quick arpeggio gestures, chord structures, overlapping lines, amplification of tessitura, design of complex rhythmic structures.

In a compositional point of view, the patch follows a former idea of cycle superimposition. Each cycle have a different compass, and each one operates with different gesture elements: notes and intervals to compose the circular cascades, the introduction of accents assigning cascades apices or points of inflexion, the incrustation of fast apoggiattura gestures, large intervals changing allowing the born of new gestures, etc.

Finally, it is important to say that these open variable implementation of the patch, served as an experimentation field to develop compositional strategies I implemented when writing a composition of two "Intermezzi and Capriccio" for solo piano, which are part of a cycle entitled Fantasia for piano, I wrote in 2013 (Figure 12). In this piece I tried to start as close as possible to Brahms model and expand it with incrustation of new gestures and changing of basic (but pertinent) parameters leading to gesture changing. In this last figure I show tow moments of Intermezzo I, where a simple changing of intervals could transform radically the texture, leading to the birth of a new gesture in the entire compositional flow.

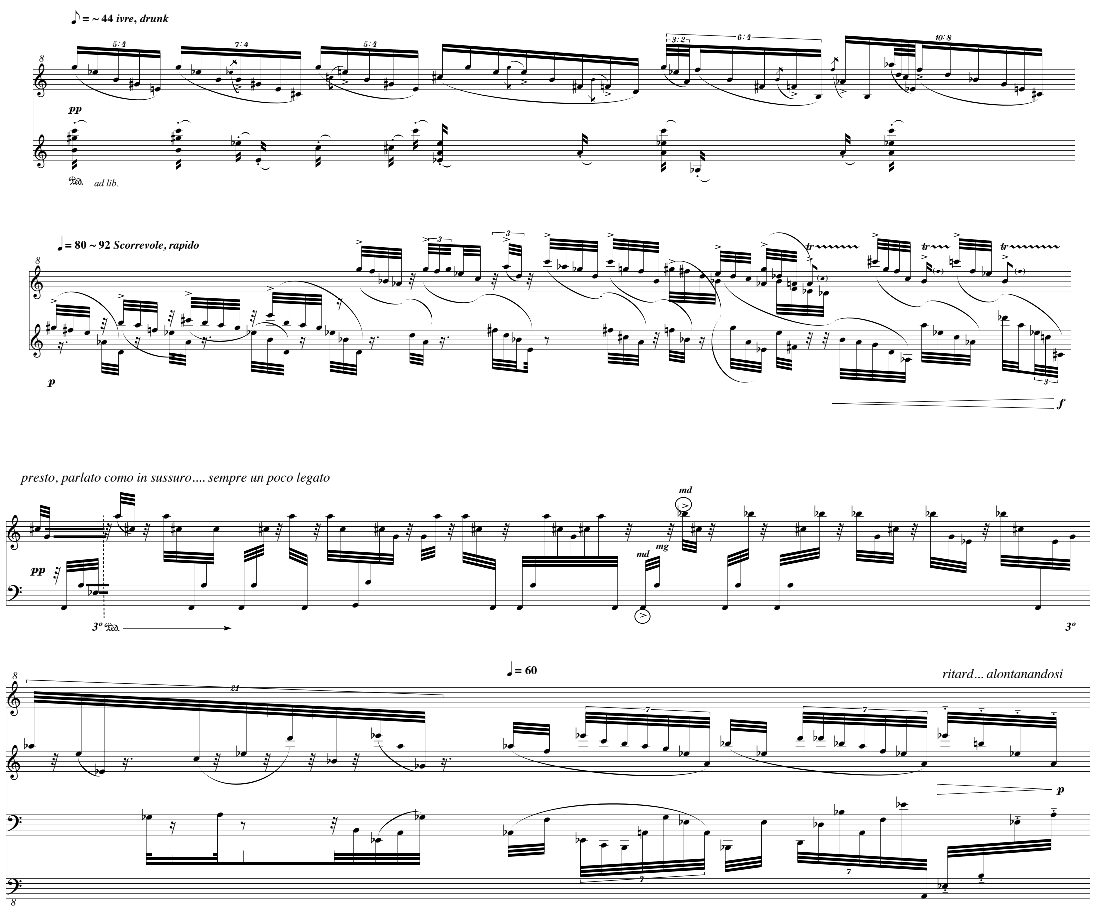

Figure 12: *Four excerpts from the score of Fantasia for solo piano (Intermezzo I and II). It shows different results from the same patch for a modeling from Brahms Intermezzo Op. 119.*

References

[1] Litzmann, B. (org). 1909. *Clara Schumann: Ein Künstlerleben nach Tagebüchern und Briefen*, (3 vols.). Leipzig: Breitkopf & Härtel.

[2] Dubnov, S.; Assayad, G.; Lartillot, O.; Bejerano, G. 2003. *Using Machine-Learning Methods for Musical Style Modeling*. *Computer*, 36/10, pp. 73–80. .

[3] Elsea, P. 2001. *Max tutorial: Using List*. Available at: http://peterelsea.com/Maxtuts_basic//Using_Lists.pdf.

[4] Ferraz, S.; Teixeira, W. 2018. Heterogenesis in Musical Rewriting and (Re)Performance. *Musica Theorica*, 3/2, pp. 189–202.

[5] Riotte, A. e Mesnage, M. 1993. Modélisation informatique de partitions, analyse et composition assisteées. In: *Cahiers de l'Ircam: la composition assistée par ordinateur. Paris: IRCAM.*

[6] Riotte, A., Mesnage, M. 2006. *Formalismes et modèles musicaux*, 1–2. Paris: IRCAM/Delatour.

[7] Roads, C. 1996. *The Computer Music Tutorial*. Massachusetts: MIT Press.