

Some anatomical insights into *Structure verticale*

Kai Yves Linden



Ateliers du Forum IRCAM 2021

Overview

Timeline

Concepts and technical set-up

Electronic «instruments»

Formal structure

Tempo and delay

Onset detection

Practical insights

Conclusion

Kai Yves Linden



born 1960 in Düsseldorf, Germany, studied composition with Wolfgang Hufschmidt at Folkwang University of Arts in Essen, Germany, from 1982 to 1986. Currently working as software engineer in a standardization institute he pursues compositional projects in his free time. His compositions comprise chamber works for solo instruments and ensembles and vocal music.

Vertical Structure · Structure verticale · Vertikale Struktur

for flute (with B-foot) and electronic extension in real time

pour flûte (avec patte de si) et extension électronique en temps réel

für Flöte (mit H-Fuß) und elektronische Erweiterung in Echtzeit

2017/2021

"Vertical Structure" is a mixed music for flute and electronics depending on IRCAM technologies Antescofo, Spat and MuBu/PiPo. In my presentation I will explain the conceptual principles of the piece and dissect some excerpts from its 19 minutes duration. I will also consider practical strategies in the compositional process to gain productivity and creativity when working with a complex software set-up.

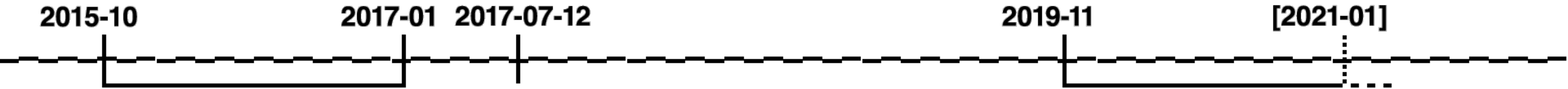
Timeline



- **2015-2017 first version**
 - Collaboration with flutist Anne Horstmann
 - First performance 12 July 2017 Margarethenkapelle, Dortmund
 - Notes on web page: <https://kymbala.de/musik/oeuvre/verticale/>
- **2019-2021 second version**
 - rewriting mostly the electronic part
 - adding ca. 7 minutes (giving a total duration of ca. 19 minutes)
 - final editorial work ongoing



Timeline



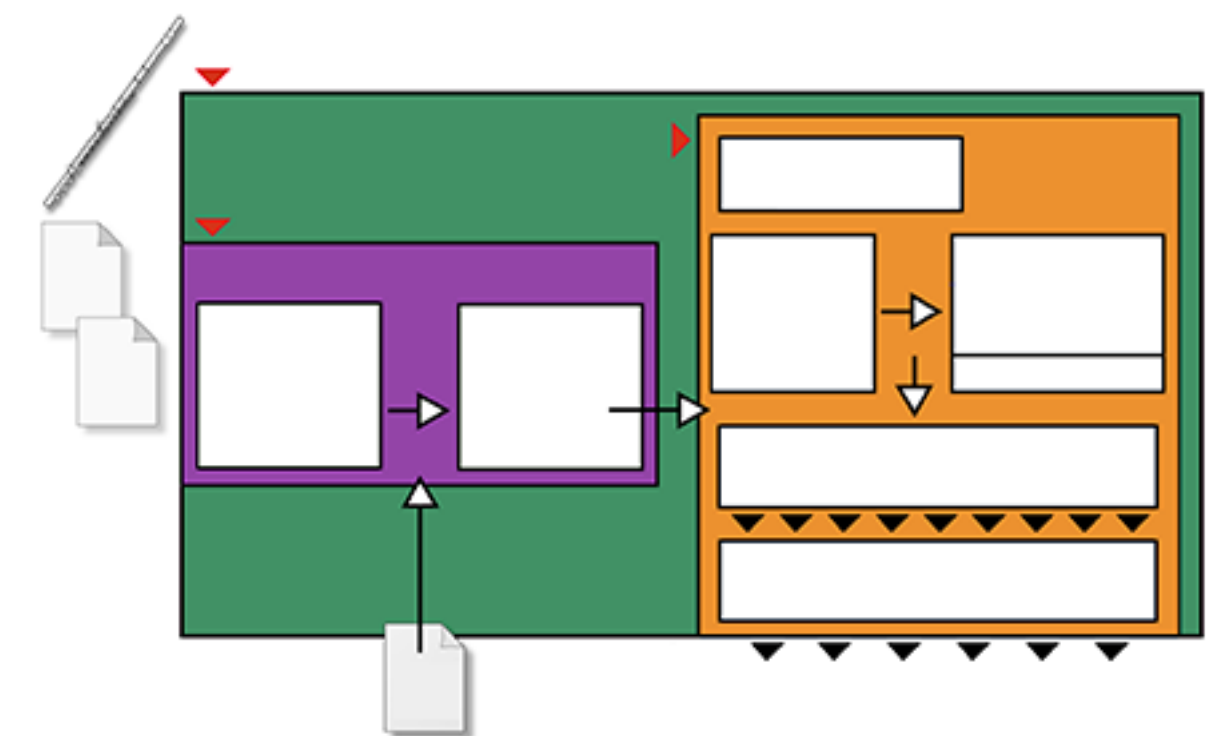
Écrire et réécrire...

(write and re-write) – favored working style

“Vertical Structure” is perhaps my most important work – at least in terms of hours I have invested into it. It was also a decisive experience in finding out what I actually want to do in music. Although the revision added more than a third of the final duration, I still consider it the same work, now in its definite form. (Yes, one has to finish and leave things sometime...)

Concepts and technical set-up

- The notes played by the **flute** are followed by the **score follower**.
- **Messages** from the score are sent to **receivers** in the sound processing environment.
- The entire performance is continuously **recorded** so parts of it can be **replayed** with arbitrary **delay**.
- Live and delayed signals are **routed** to and between **modulators** and **generators**.
- The gain (volume) of **internal signal lines** is modulated and pre-balanced.
- These are passed to the **spatialization** and **reverb** module for the final mix.



Microphone (attached to instrument)

Technical set-up

Signal processing environment

Sequencing

IRCAM Antescofo

Score following

Action management

Listening machine

Scheduling machine

Recording unit

buf-rec

Structure verticale
Sound processing

Signal path control

inroute

Modulators

VZ (delay/playback)

TP (transposers)

RM (ring modulators)

SIN (sine generator)

Internal output gain control

inmix

Spatial distribution

IRCAM Spat

spat

Performer's part

Result score

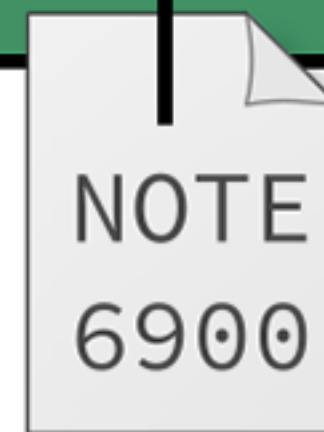
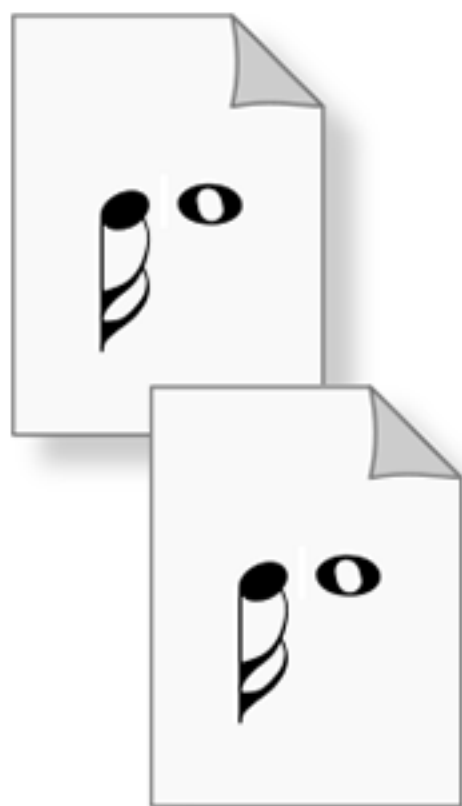
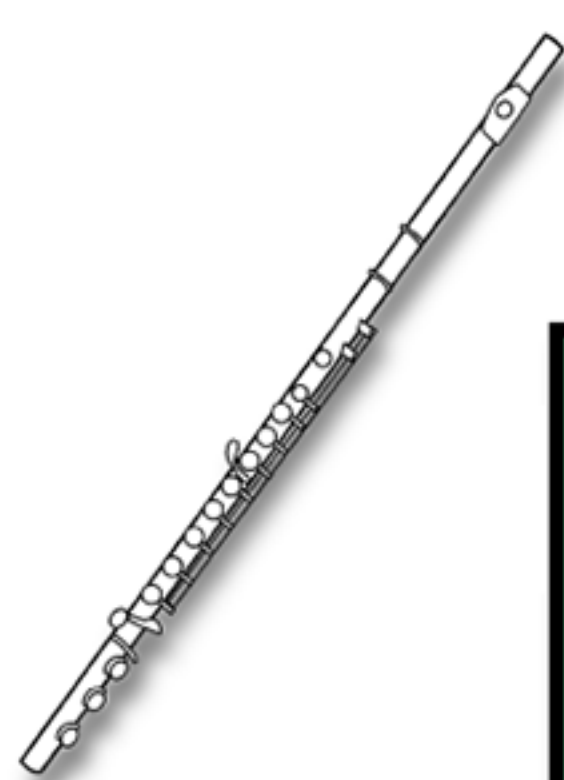
Max

Action score
(machine-readable)

NOTE
6900

1 2 3 4 5 6

(4, 6 or 8) Loudspeakers



Electronic «instruments»

- ▶ Bounded number of signal processing objects by disposition
 - ▶ based on fundamental electronic modulation methods
- ▶ «Personalities» – on the sound stage and “*under the surface*”
- ▶ Multiform «mirror» of the flute

«Instrument» types

Core

(Modulators)

VZ: buffer playbacks

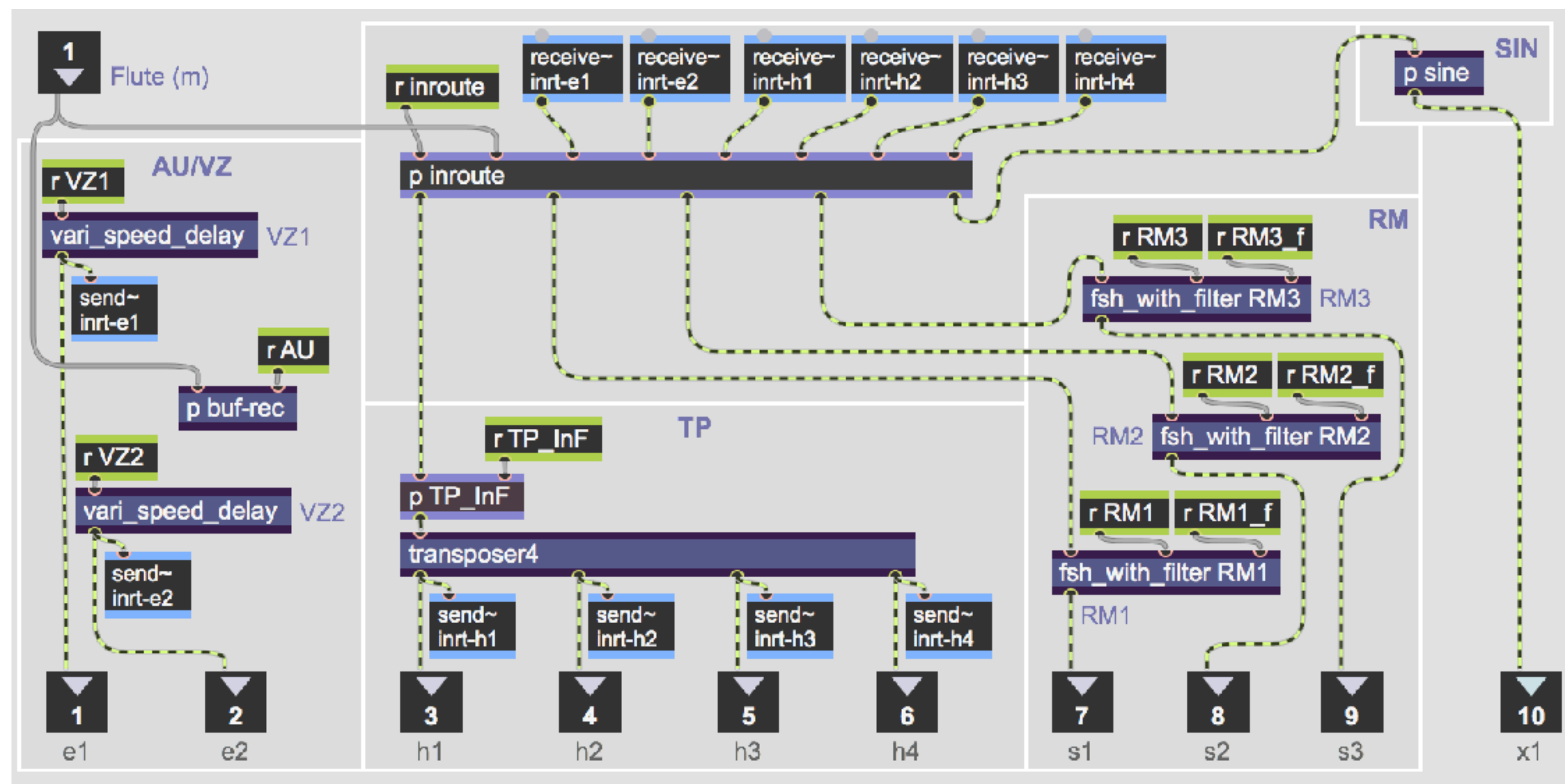
TP: transposers

RM: ring modulators

Sideline

(Generators)

SIN: sine wave oscillator



Max patcher *soundprocessing*

Electronic «instruments»

- ▶ (1.) **Polarity** between
 - ▶ the instrumentalist (the human actor) and
 - ▶ the electronic sounds (the machine)

- ▶ (2.) **Differentiation** within
 - ▶ the electronic sounds («instruments») and
 - ▶ in their relation to the sounds produced by the instrumentalist

«Instrument» types

Core

(Modulators)

VZ: buffer playbacks

TP: transposers

RM: ring modulators

Sideline

(Generators)

SIN: sine wave oscillator

The three main types of electronic «instruments» modulate the flute sounds by changing the pitch or frequency, each in a different way.

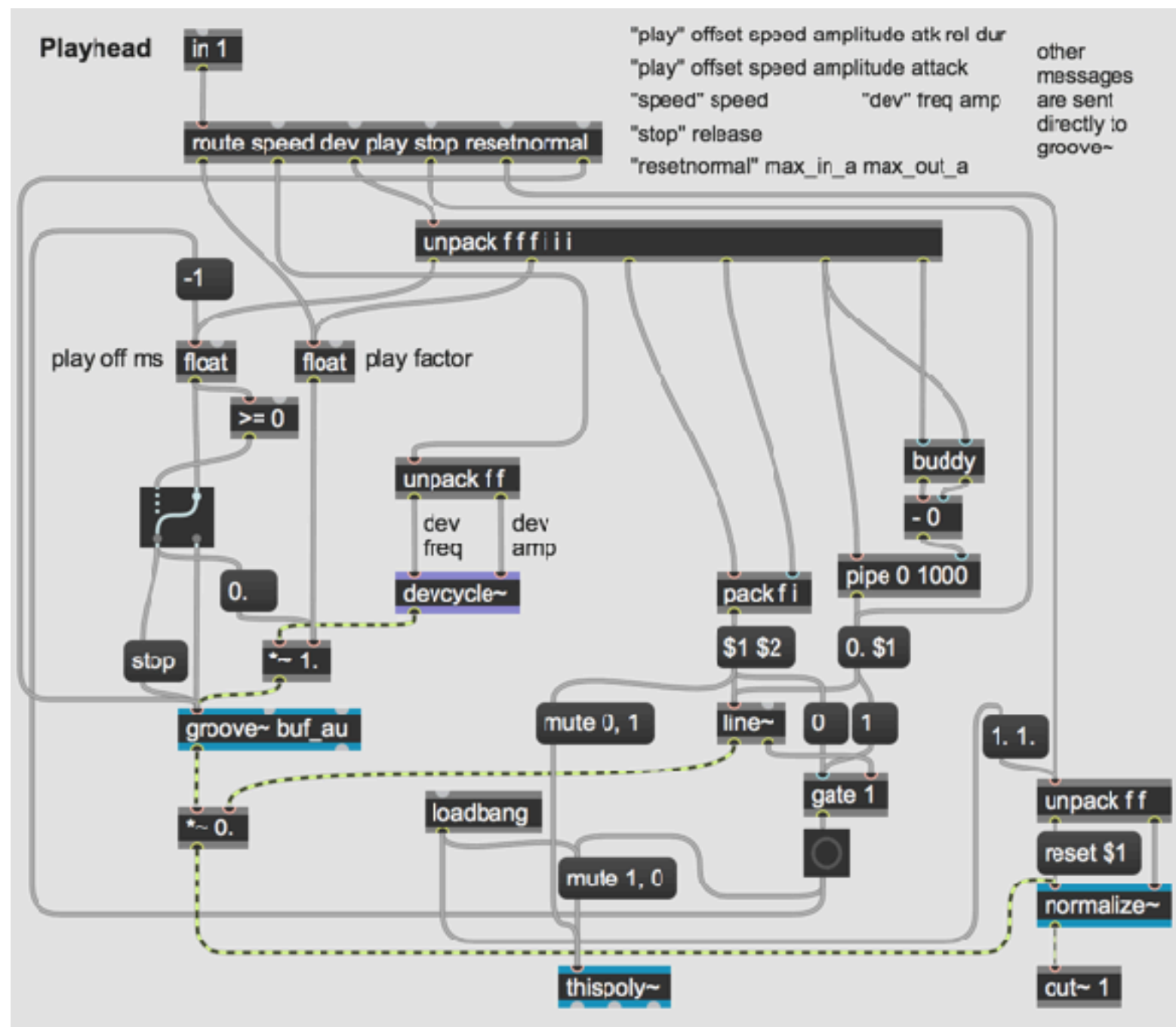
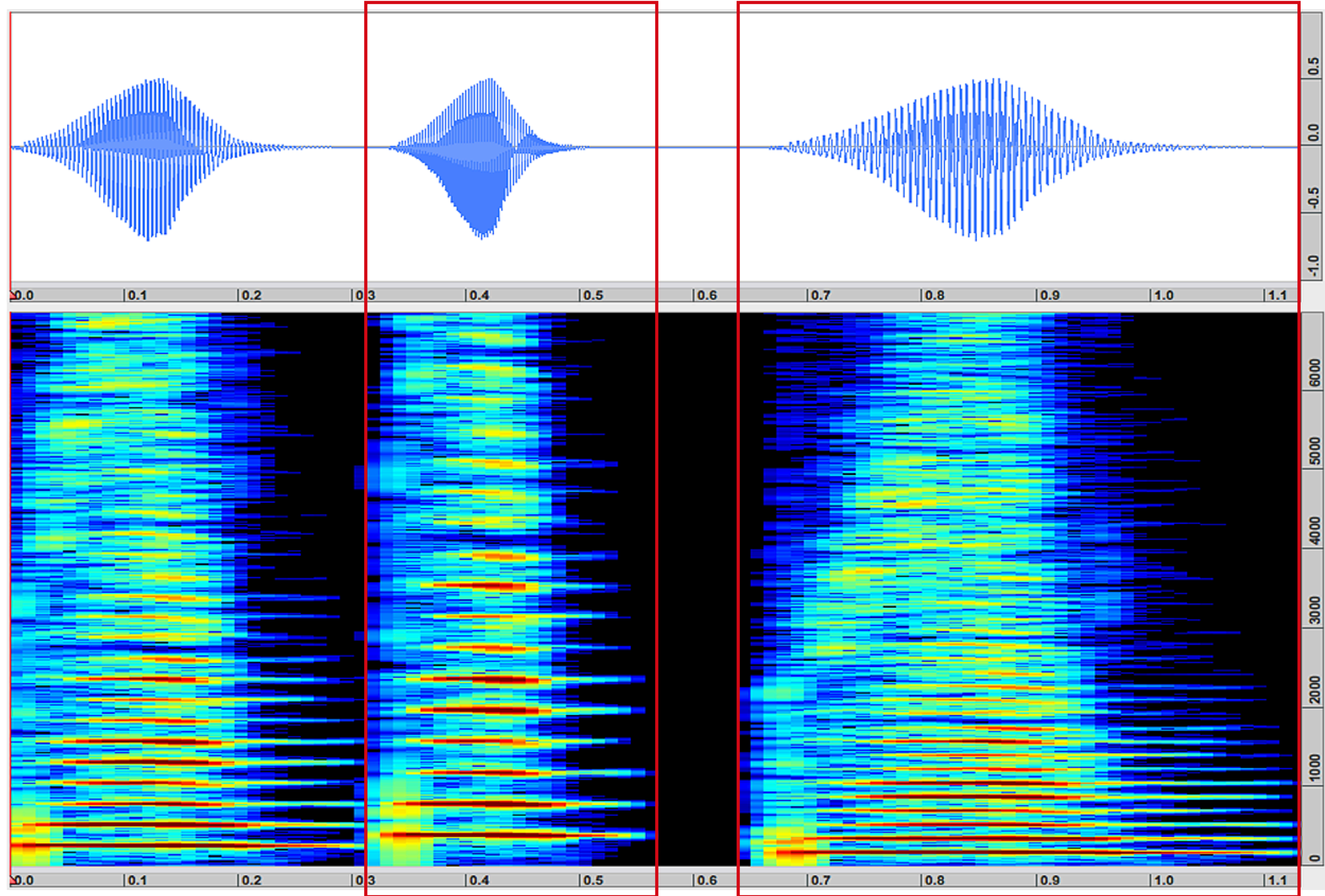
Three types of pitch/frequency change: 1. Playback speed – *Changing pitch and duration*

Ex.

[original]

$f \times 3/2$

$f \times 2/3$



Max patcher *polyplayhead* in sound processing unit **AU/VZ**

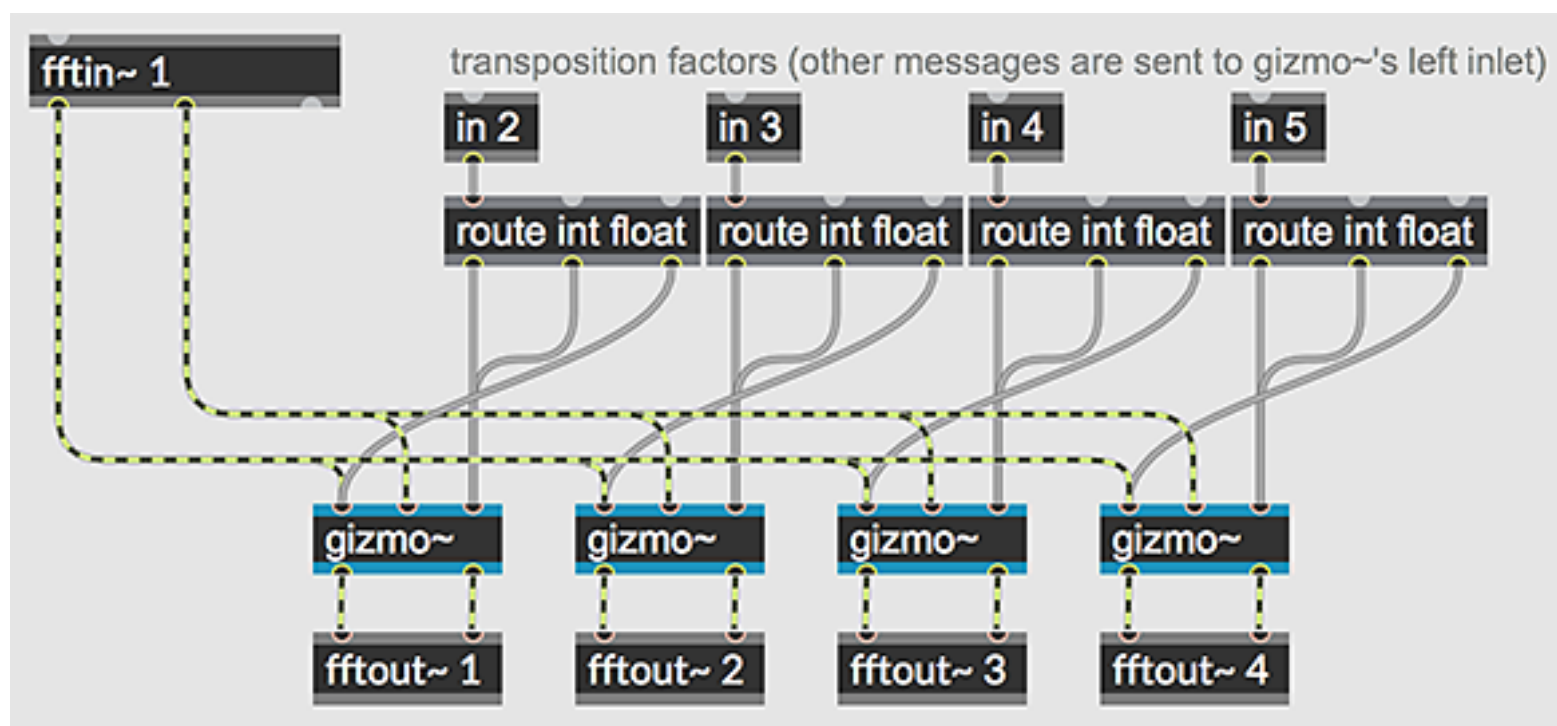
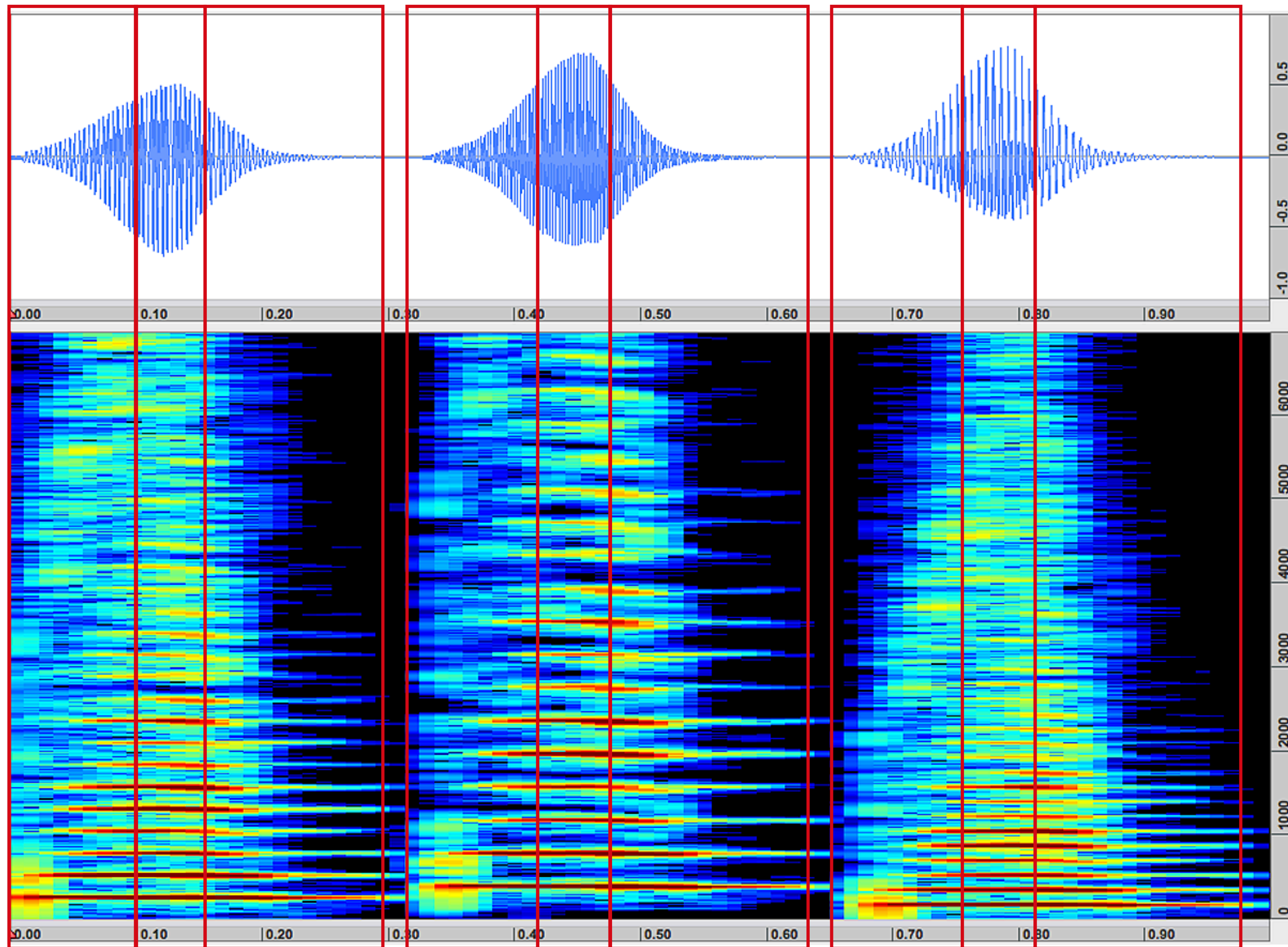
Three types of pitch/frequency change: 2. Pitch shift in the frequency domain – *Transposing the spectrum*



$f \times 1$

$f \times 3/2$

$f \times 2/3$

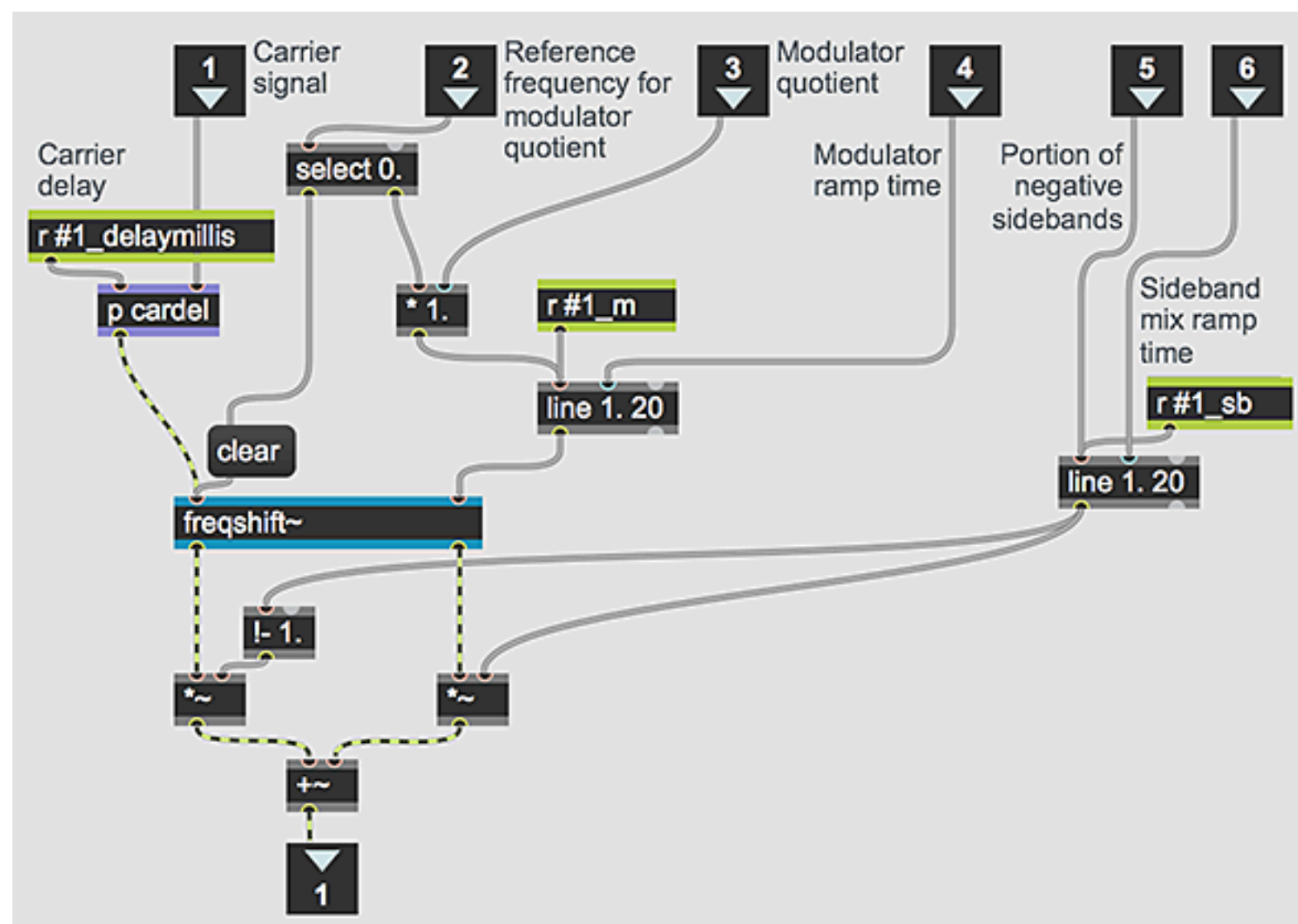
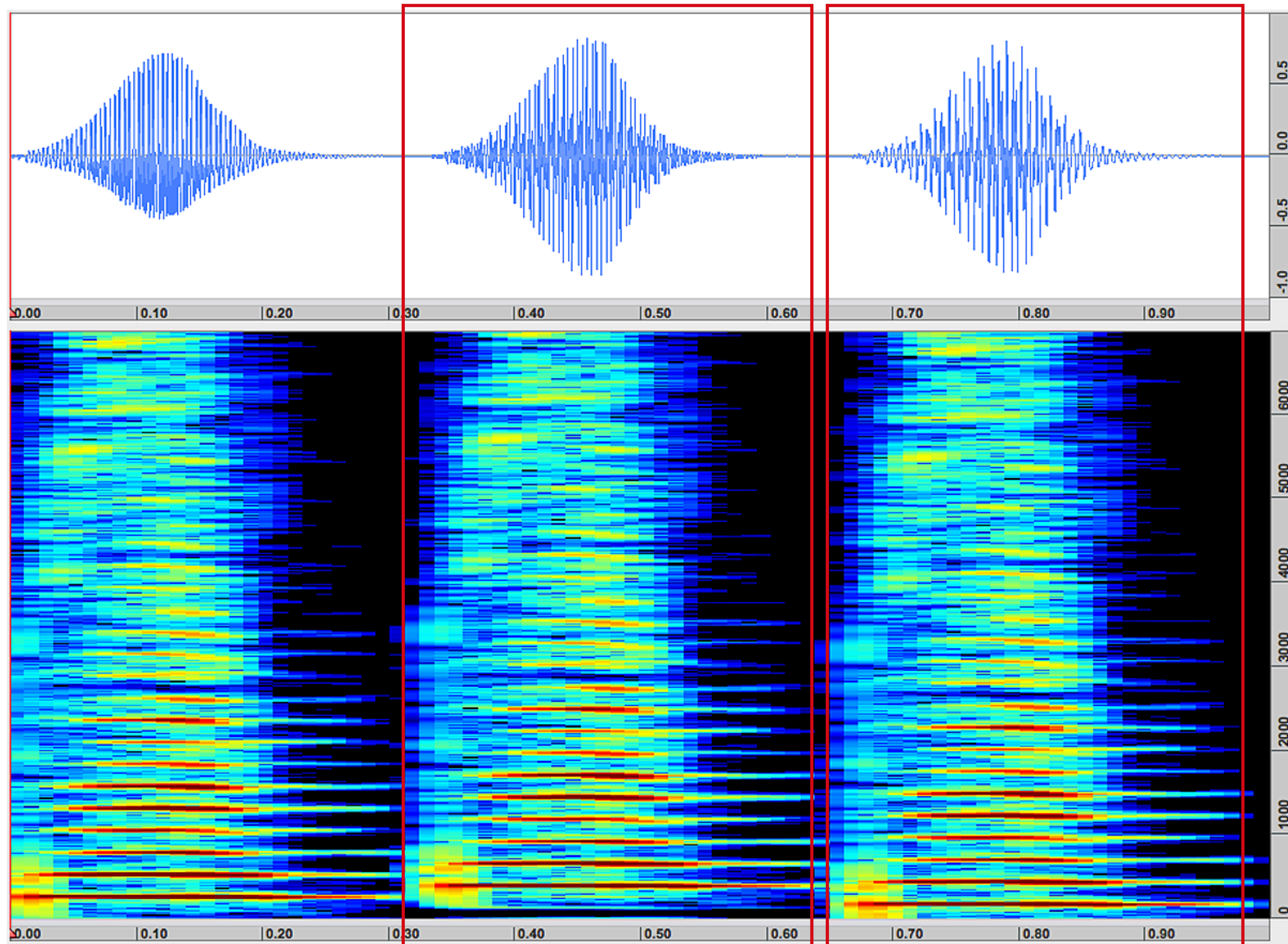


Max patcher *pfft_tp4* in sound processing unit **TP**

Three types of pitch/frequency change: 3. Frequency shift (ring modulation) – *Distorting the spectrum*

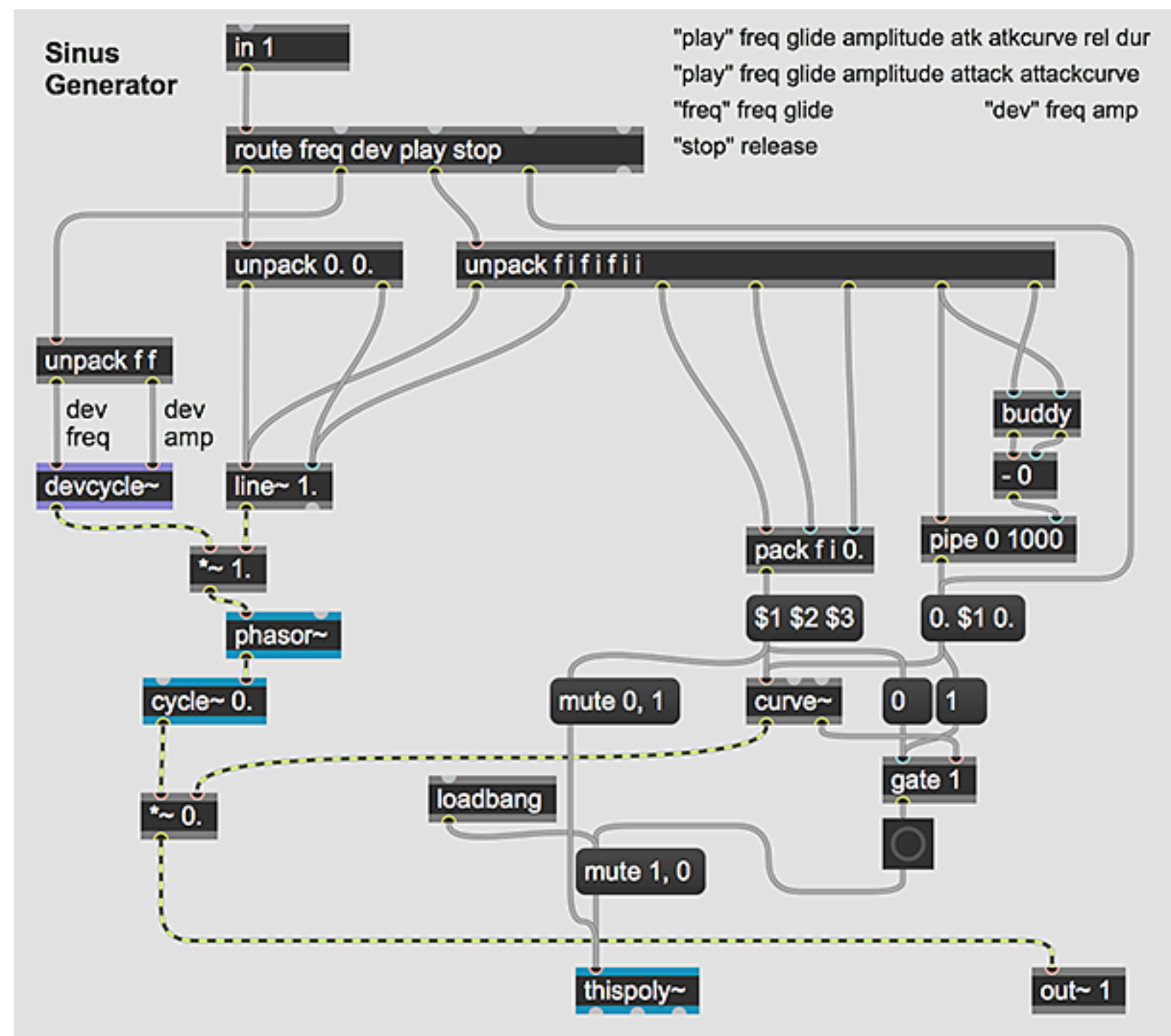
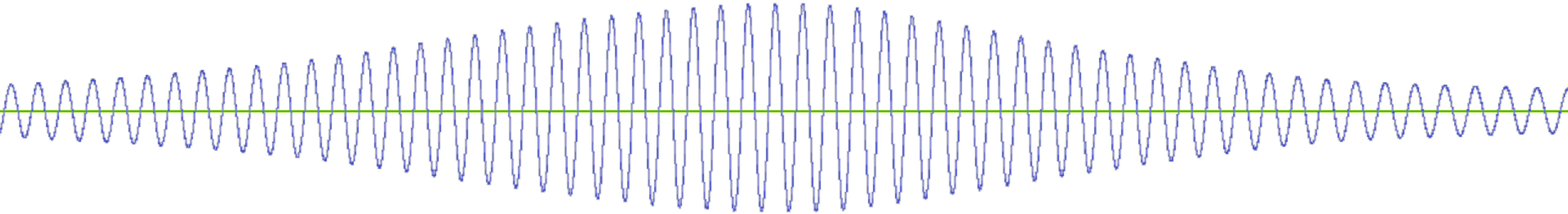
Ex.

$f + 0$ $f + f \times 1/2$ $f + f \times -1/3$



Max patcher *fsh* in sound processing unit **RM**

Additional unit: Sine wave oscillator



- ▶ **“Outside”** the disposition
- ▶ however, already present as modulator in RM
- ▶ Roles:
 - ▶ “sustaining pedal” (“sympathetic vibrations”)
 - ▶ “[pre-]echo”
 - ▶ “screen” (envelope co-modulated by flute or other sources)

Max patcher *polysin* in sound processing unit **SIN**

The first minute

All «instruments» are introduced, by and by, in the opening sixty and so seconds.

▶ **Excerpt m1-16**

Musical score for the first four measures. The top staff is for Flöte (Flute) and the bottom two staves are for Elektronische Klänge (Electronic Sounds), specifically Sinusgenerator (SIN) and Transponierer (TP). The tempo is marked as quarter note = 69. The time signature changes from 4/4 to 5/4 and back to 4/4. Dynamics include *mf*, *p*, *poco cresc.*, *p*, *mf*, *mp*, *p*, and *mp*. The score features various rhythmic patterns, including triplets and sixteenth notes.



Musical score for the next four measures, starting at measure 5. The staff is for Flöte (Fl). The tempo is marked as quarter note = 69. The time signature changes from 4/4 to 3/4 and back to 5/4. Dynamics include *p*, *f*, *mf*, *poco cresc.*, *p*, *ff*, *f*, *p*, *ff*, and *ppp*. The score features various rhythmic patterns, including triplets and sixteenth notes. A box labeled 'A' is present at the end of the excerpt.



A

Fl

Flute part with dynamics and articulation: *mp*, *ppp*, *mp*, *p*, *pp*, *p*, *ppp*, *p*, *pp*, *f*, *mf*, *ff*, *p*, *pp*, *f*, *pp*, *mf*, *p*, *mf*, *pp*

Articulation: *lip*, *tgp*, *æol*, *(æol) ord*

Tempo/Performance: *II*

Measure markings: 3, 4+8, 4, 5, 4

E

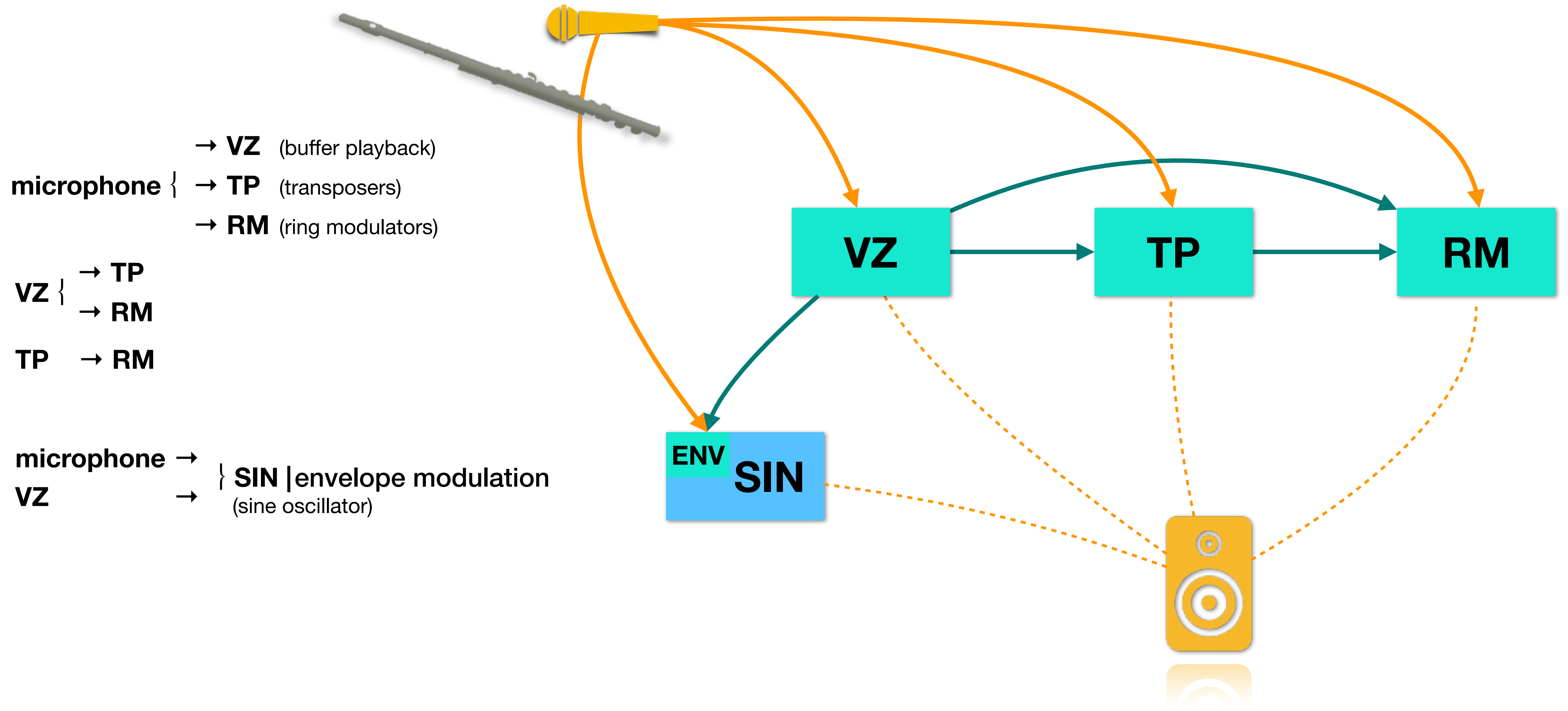
Ensemble part with dynamics and articulation: *pp*, *p*, *ppp*, *p*, *pp*, *f*, *mf*, *ff*, *p*, *pp*, *f*, *pp*, *mf*, *p*, *mf*, *pp*

Performance markings: Verzögerer (VZ), TP, VZ+RM

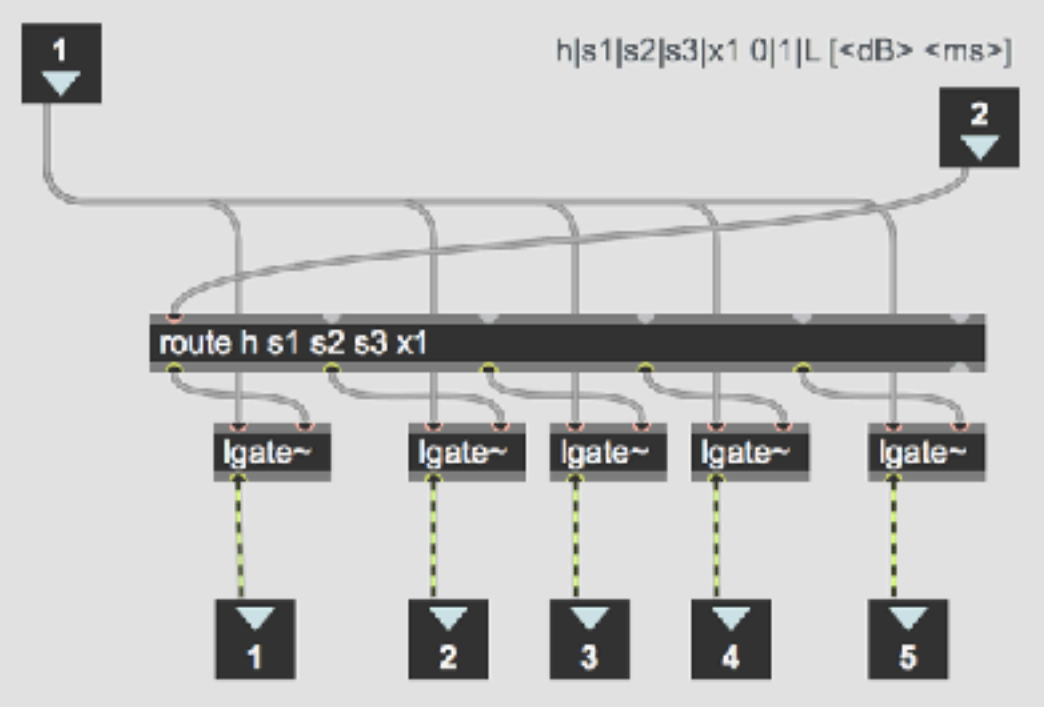
Tempo/Performance: *II*

Measure markings: 4+8, 4, 4, 3, 3, 3, 5, 4, 4

Input routing: Overview of signal feed lines

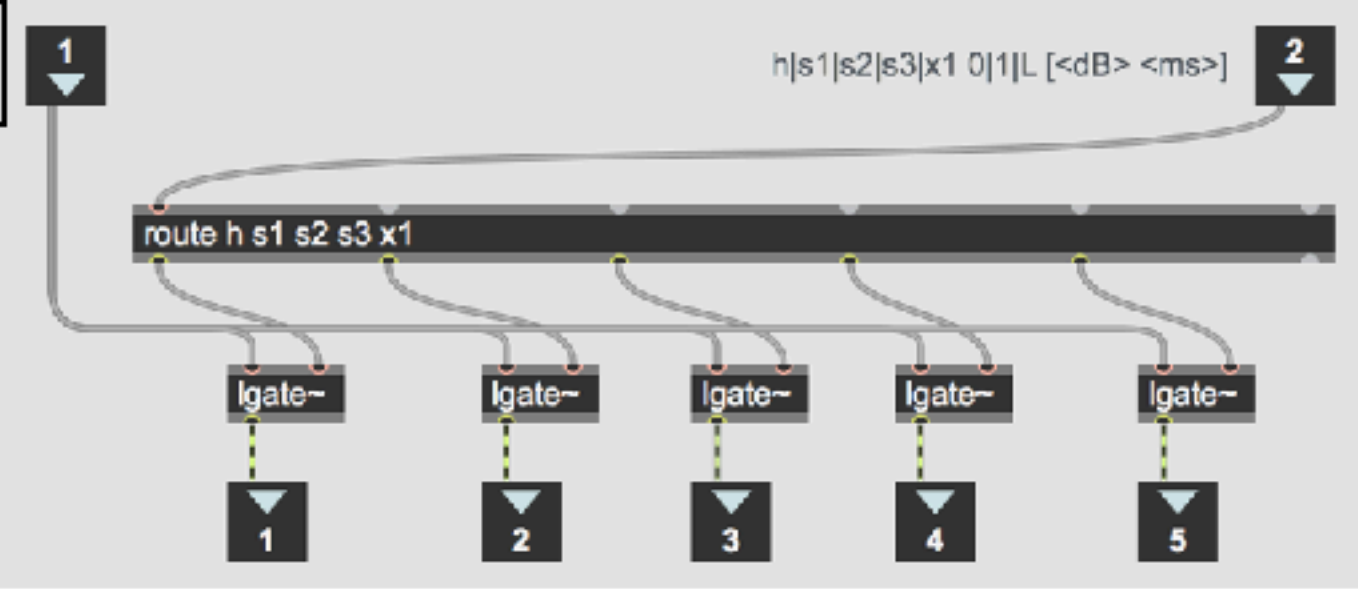


Input routing: Signal feed lines (Max patchers)

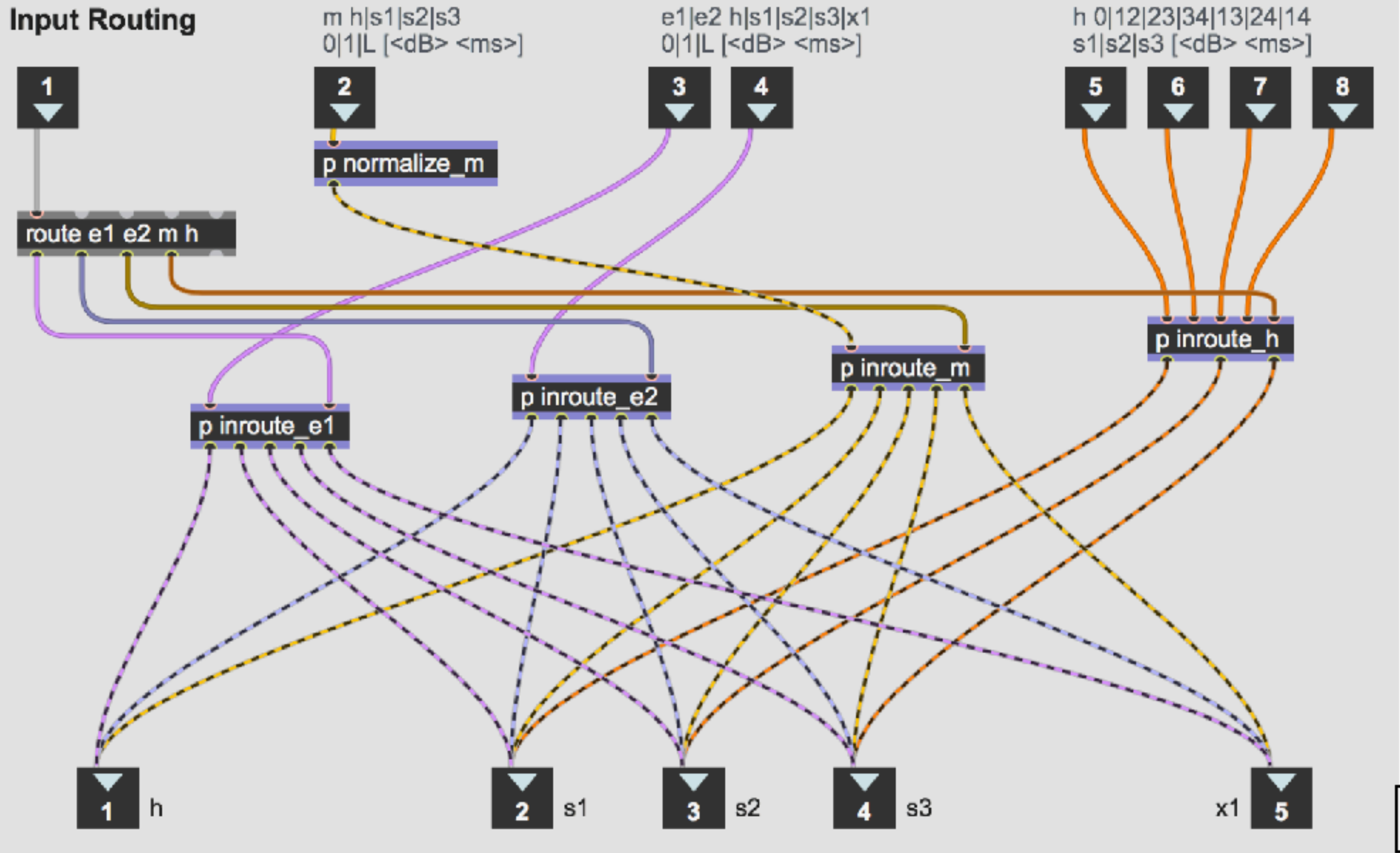


inroute_e1
inroute_e2

inroute_m

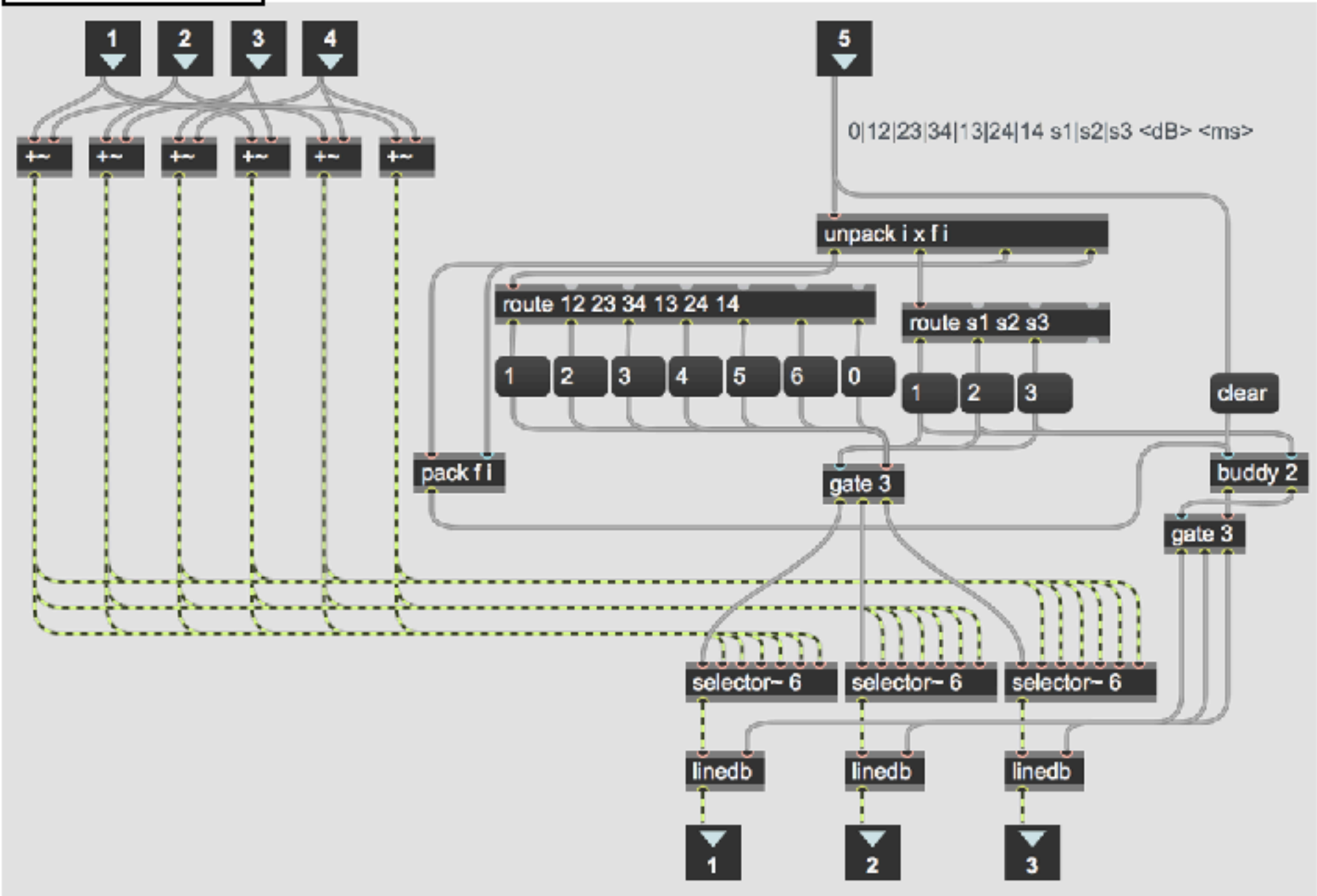


Input Routing

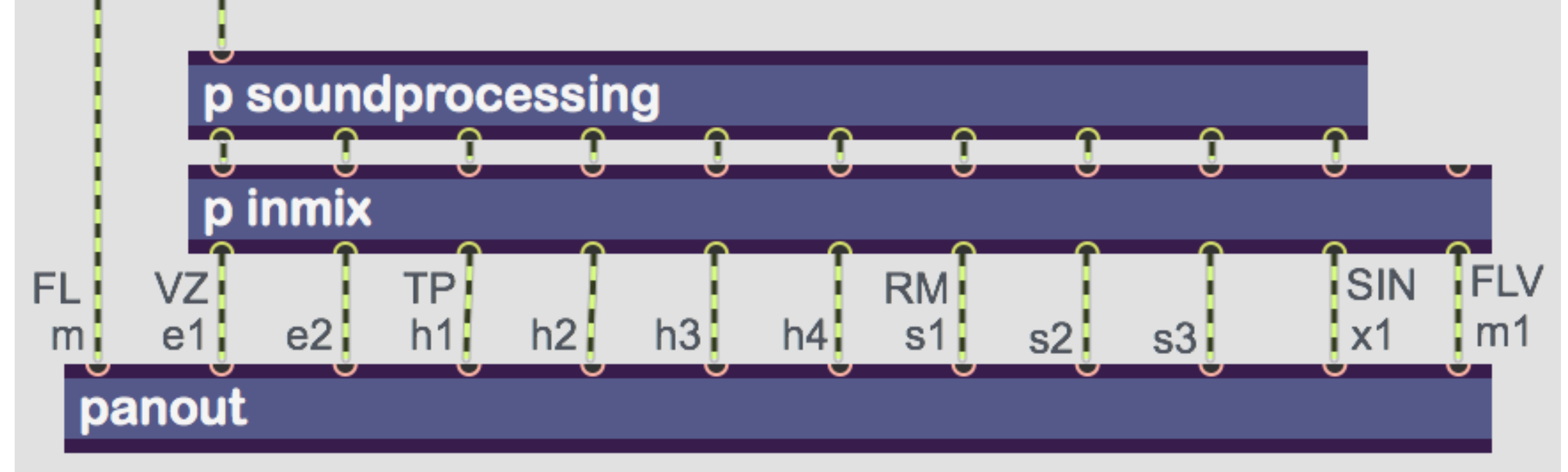


Max patcher *inroute*

inroute_h



Internal output mixer



incoming levels

faders for manual adjustment

gains controlled by action score

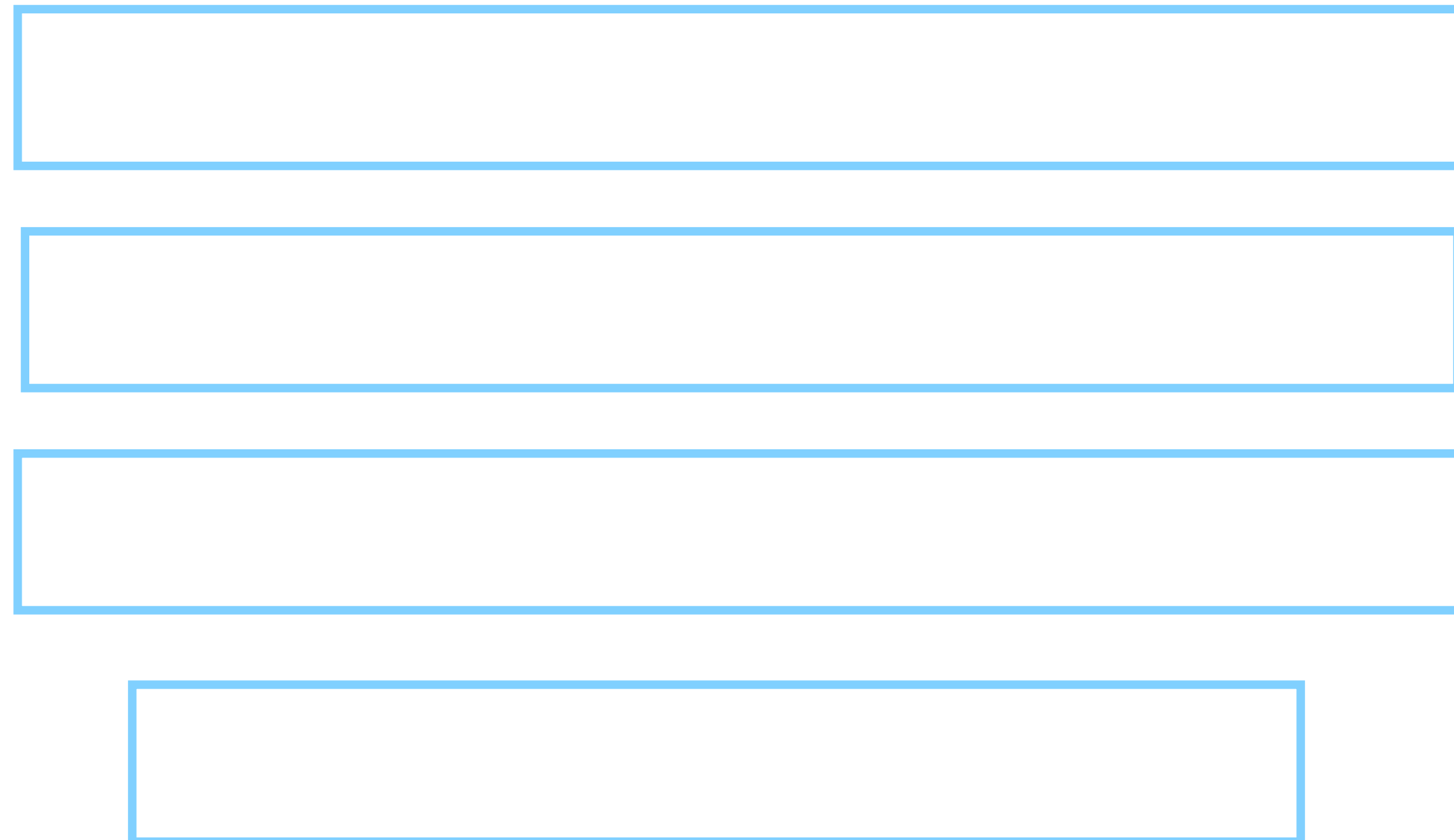
outgoing levels

Internal Output Mixer

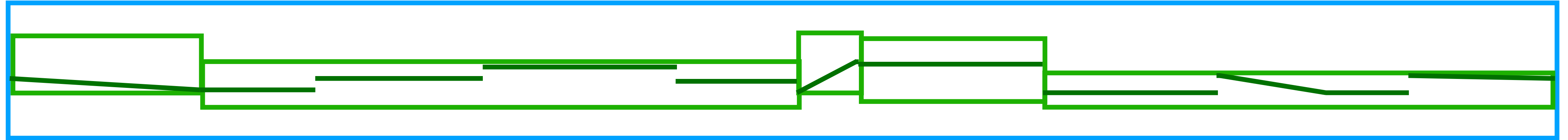
Max patcher *inmix*

Formal structure and tonal space

- “Verticale Structure” is not built on a blueprint.
- The composition was the adventure of an expedition without map.
- I’m fascinated by the ideas of temporality and contingency.
- The diagram shown here is the result of an retrospective breakdown.

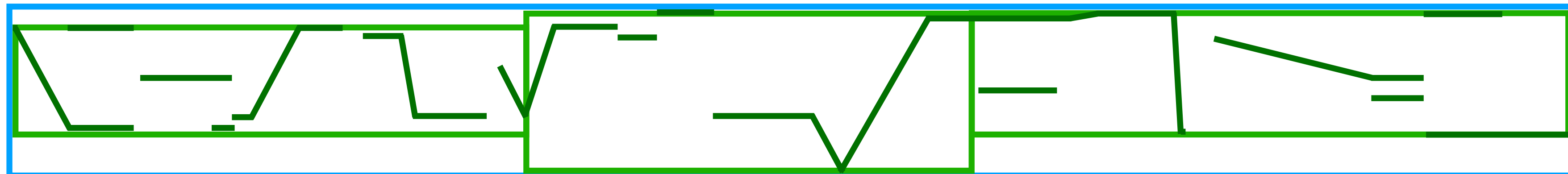
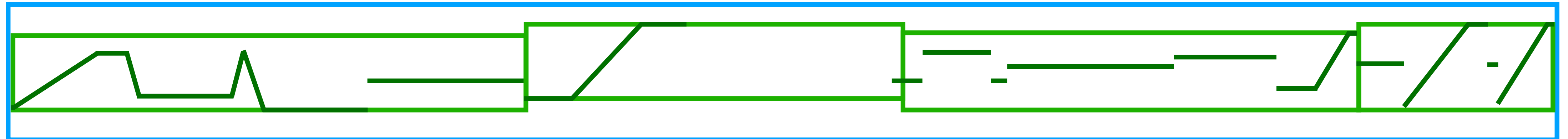
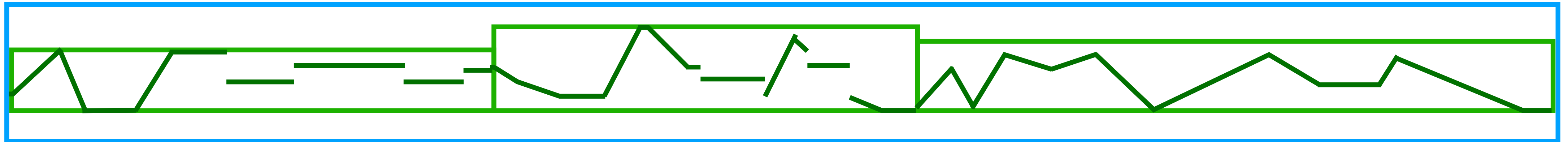


Formal structure and tonal space: Movement in the tonal space (flute part)

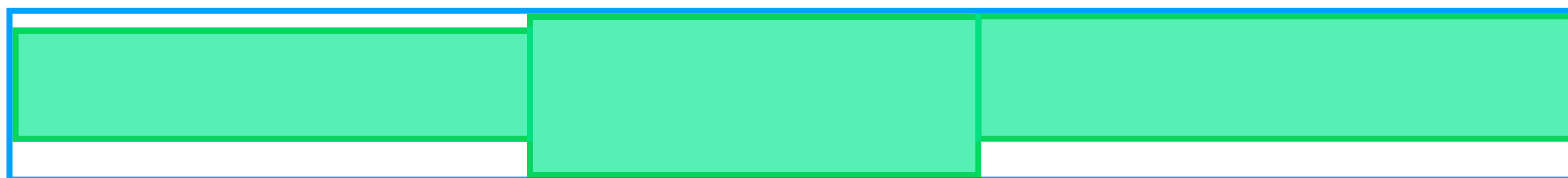
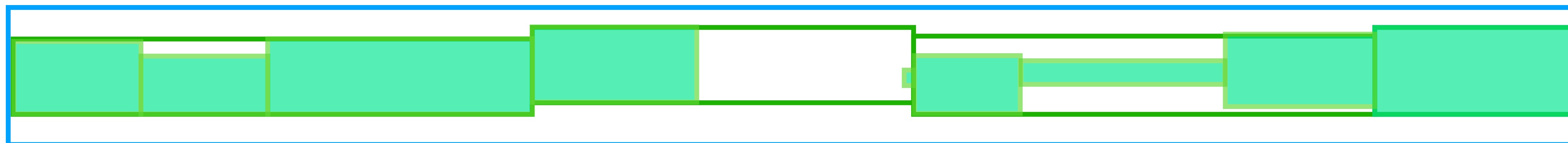
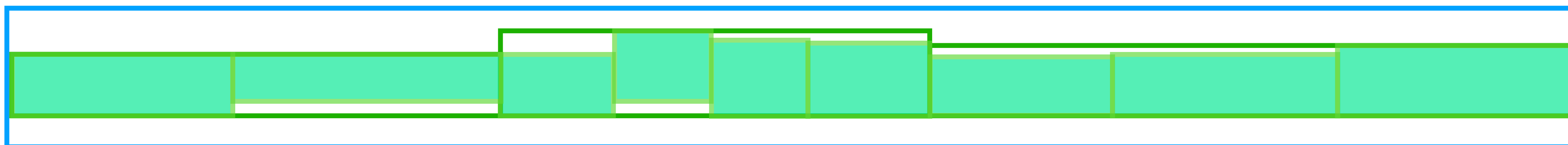
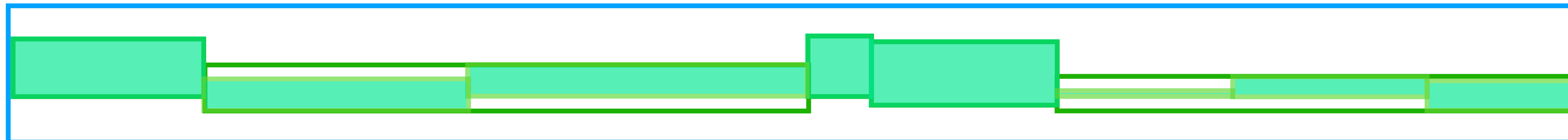


green boxes: local tonal compass

dark-green line: center of attraction

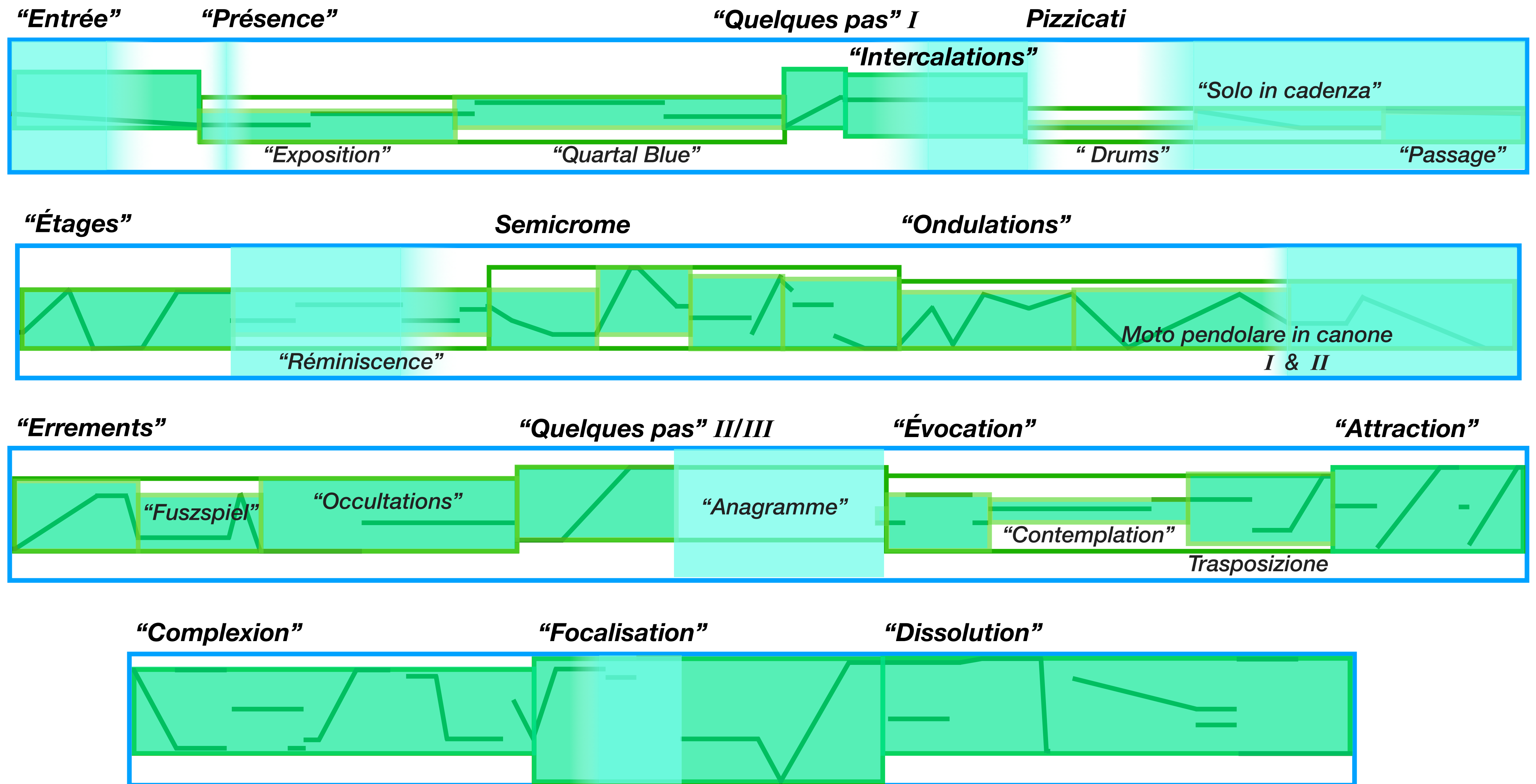


Formal structure and tonal space: Structure formants* (defining episodes and variants)



* property-sets regarding pitches, durational patterns, texture, generative tendencies, stylistic models, ...

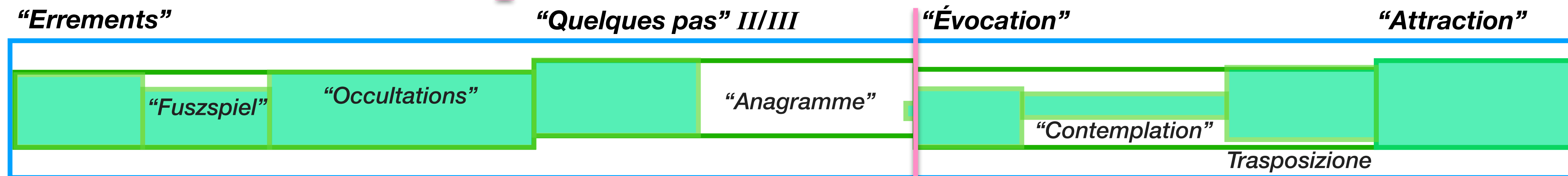
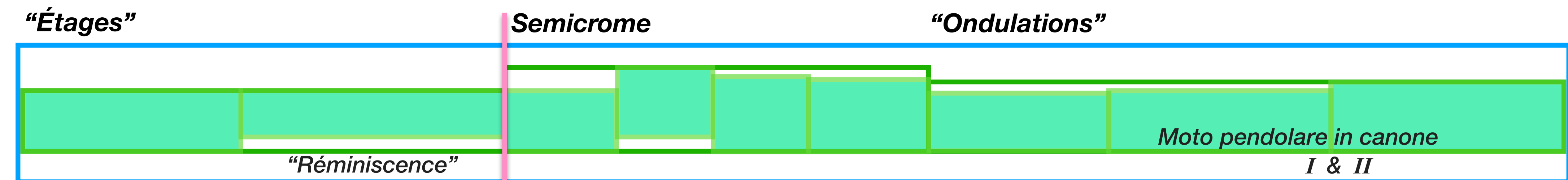
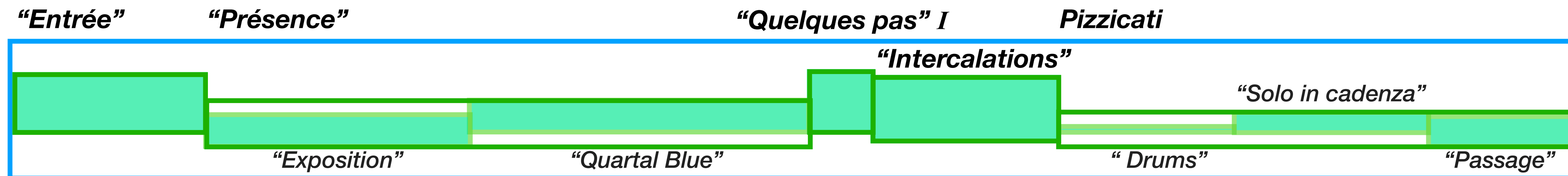
Formal structure and tonal space: Sections and sub-sections



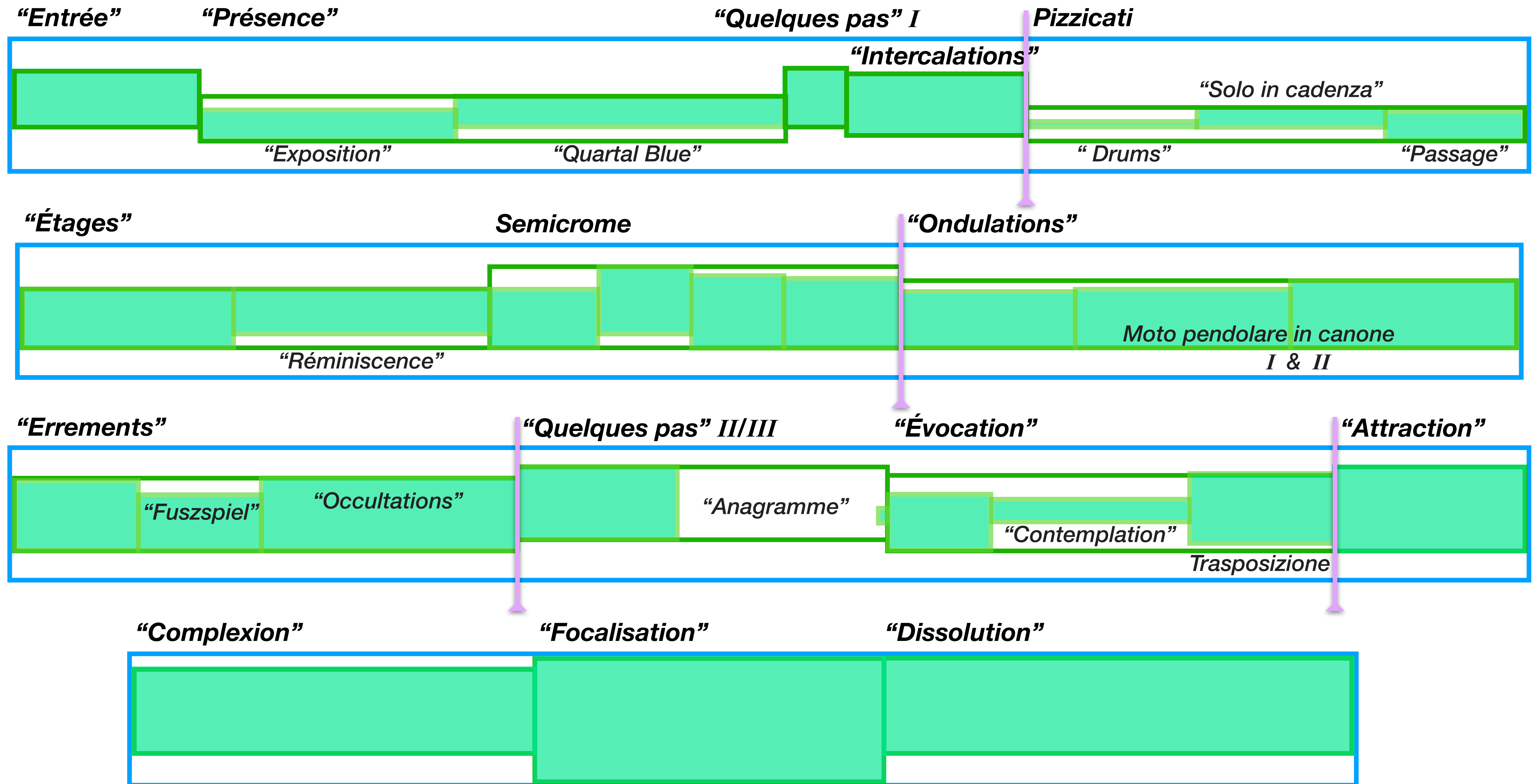
(The section names are working titles for easier location.)

indigo overlay: sections added or extended in the revision

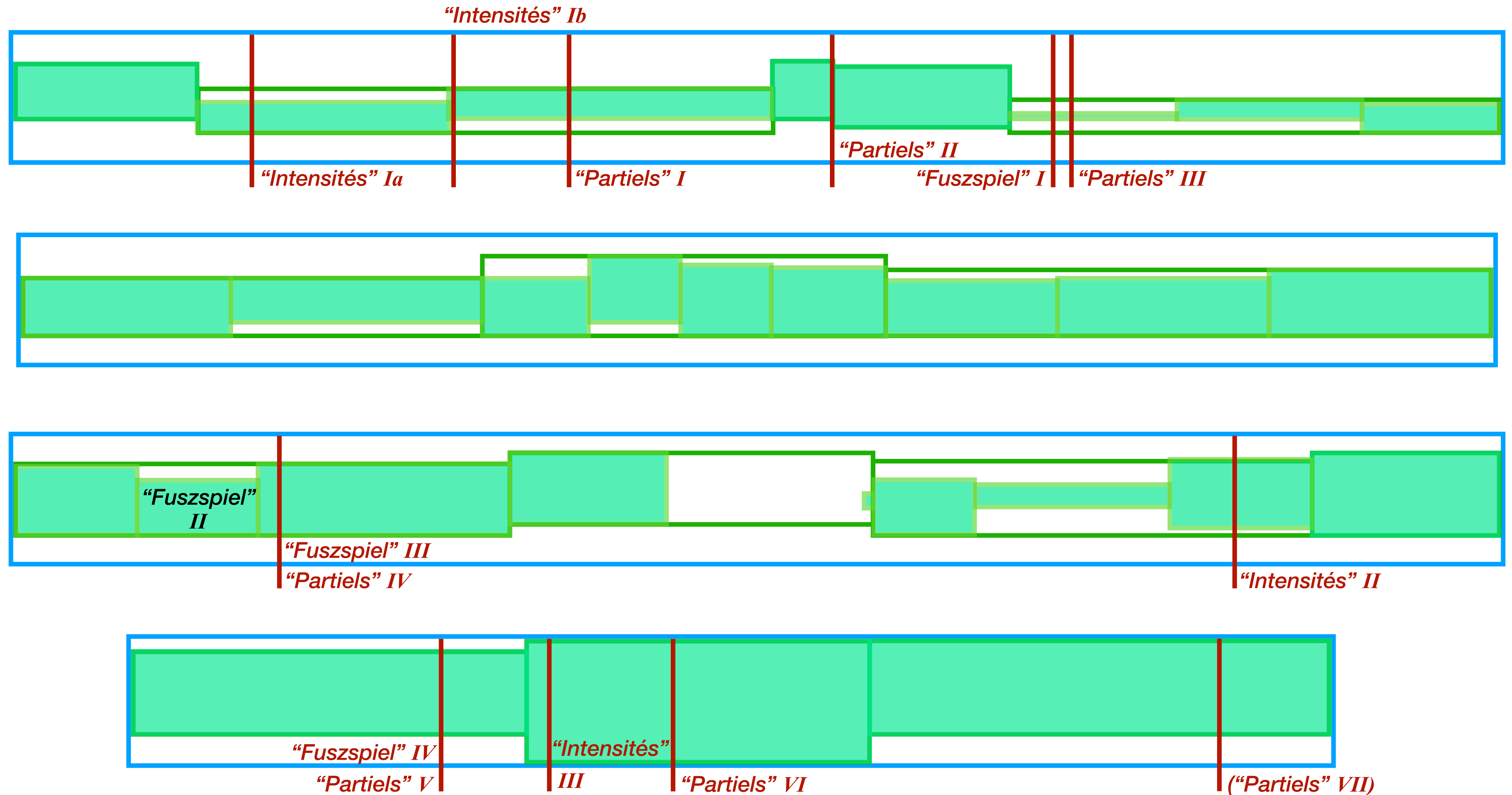
Formal structure and tonal space: Grouping into three divisions pink markers



Formal structure and tonal space: **Alternative grouping into five divisions** violet markers

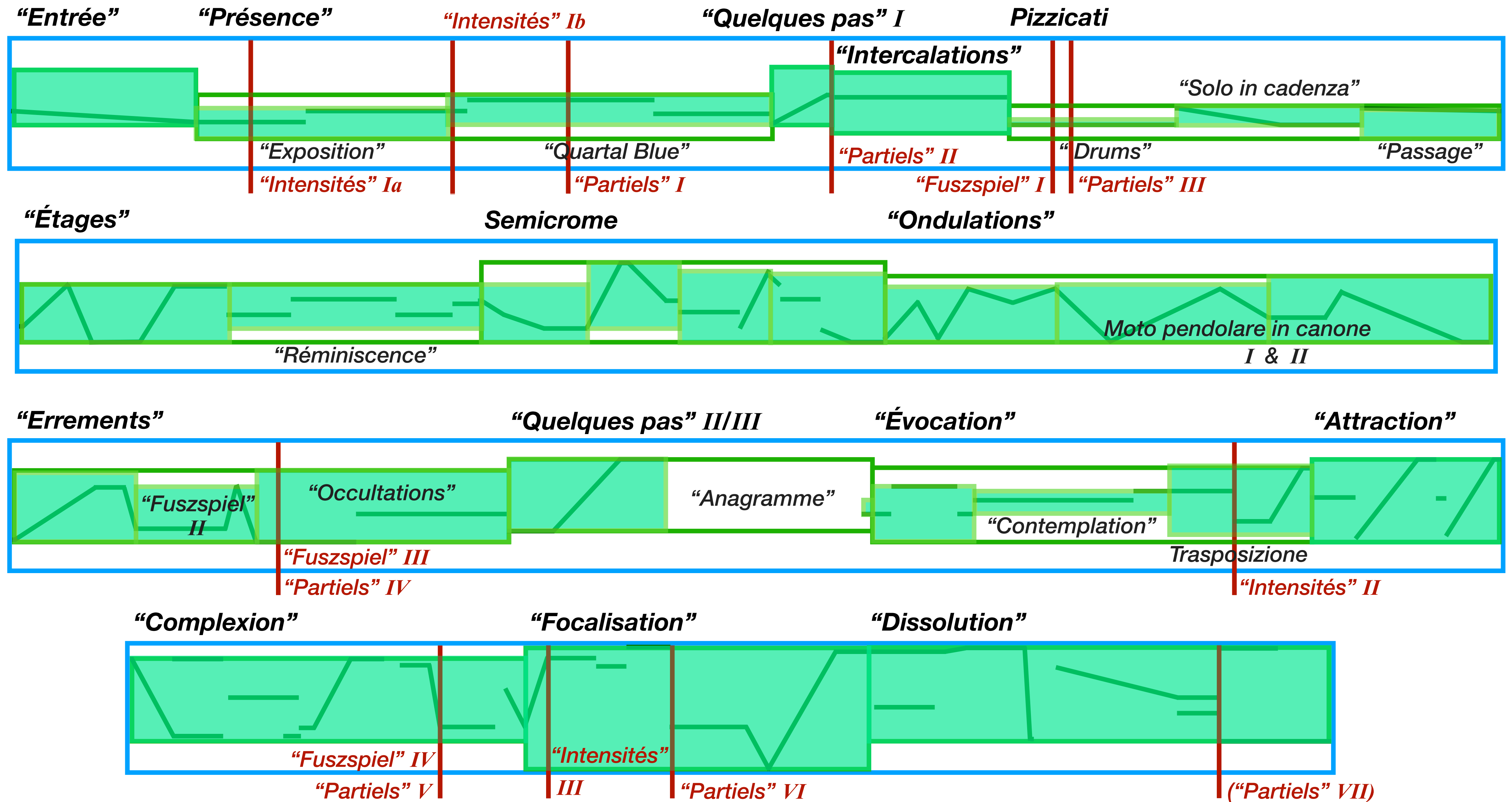


Formal structure and tonal space: Recurring phenomenal types which add a further layer

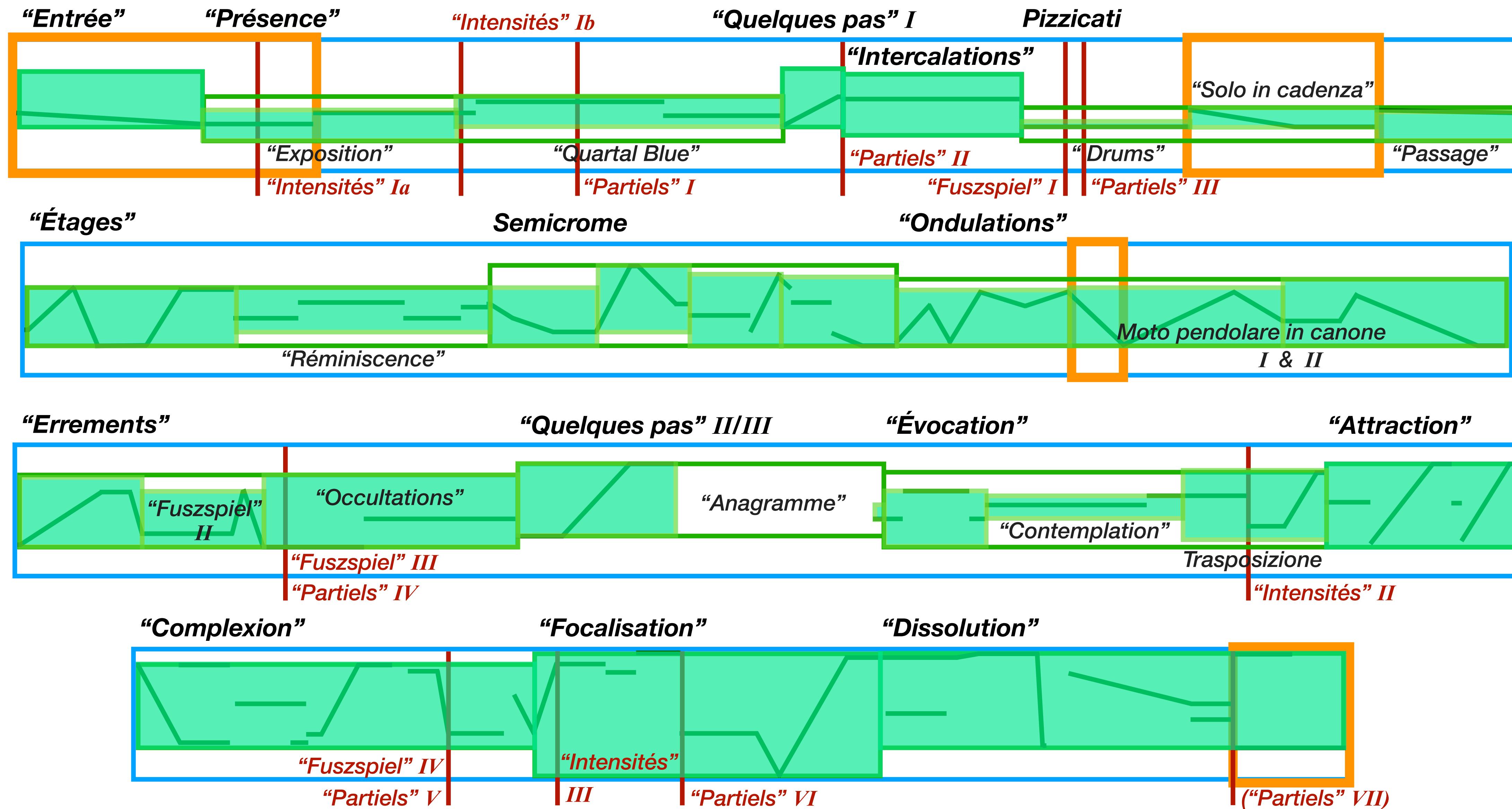


red lines: start points of phenotype occurrences

Formal structure and tonal space: A “dream narrative”



Formal structure and tonal space: Examples (excerpts) in this presentation **orange outlines**



Moto pendolare in canone I

Tempo and delay



Canon with five comes (speed transpositions)

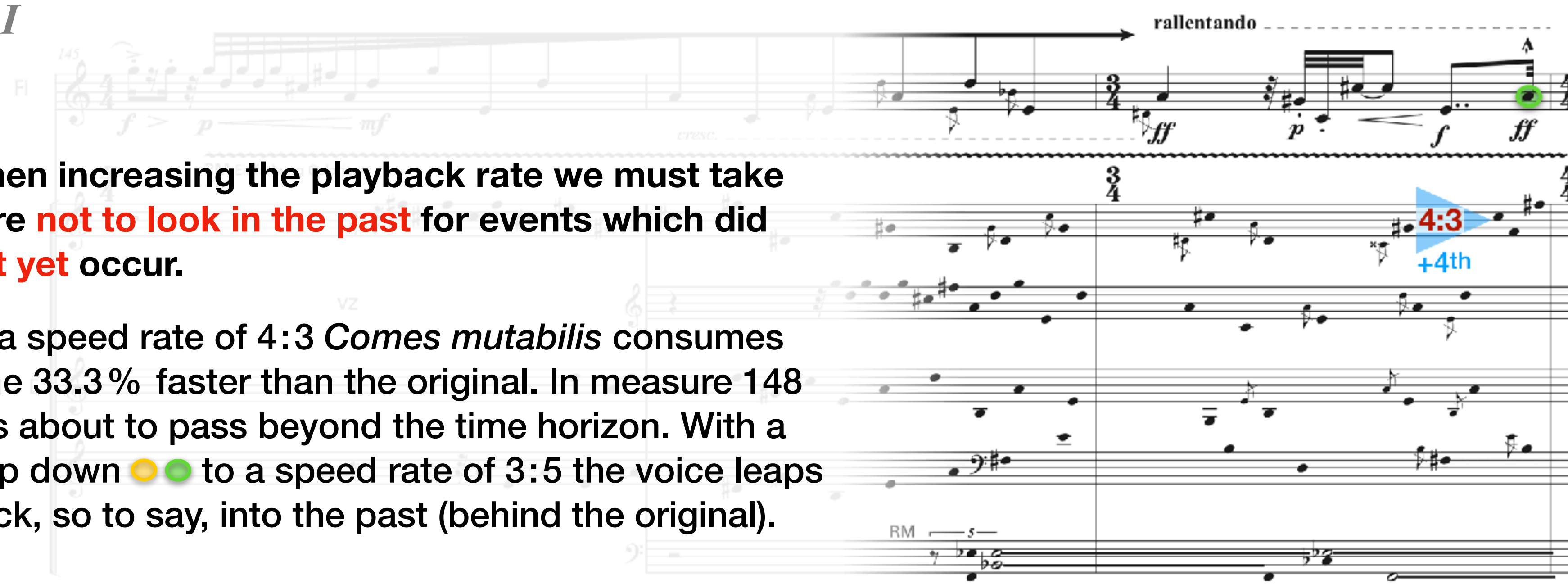
- ▶ Comes *mutabilis*: changes transposition, temporarily in diminution (accelerated);
- ▶ Comes in unison;
- ▶ A fourth slower (and lower);
- ▶ A minor seventh below;
- ▶ Comes *bassus duobus*: almost four times slower and deeper, middle portion of specimen in interlaced doubling, enriched by ring modulation.

The image displays a musical score for 'Moto pendolare in canone I' with five staves. The top staff is for Flute I (FI) starting at measure 145, with a tempo of $\text{♩} = 69$. It features dynamics $f > p$, mf , $cresc.$, ff , p , f , and ff . A 'rallentando' marking is present. The score includes time signatures $\frac{4}{4}$, $\frac{3}{4}$, and $\frac{4}{4}$. Annotations include 'RM Einfärbung Coloring' and 'RM s-'. The second staff has a blue bar with '15:16 -min.2nd' and a blue arrow pointing right. The third staff has a blue bar with '1:1 unison VZ' and a blue arrow pointing right. The fourth staff has a blue bar with '3:4 E -4th' and a blue arrow pointing right. The fifth staff has a blue bar with '9:16 -min.7th' and a blue arrow pointing right. The sixth staff has a blue bar with '4:15 -maj.7th -8ve' and a blue arrow pointing right. The bottom section of the score starts at measure 148 with a tempo of $\text{♩} = 46$. It includes dynamics p , p poss., and mf . Annotations include 'poco n', '3', '5', '16:17', '3:5 -maj.6th', '5', '3', '9:12', and '18'. The bottom two staves have blue bars with '3:5 -maj.6th' and a blue arrow pointing right.

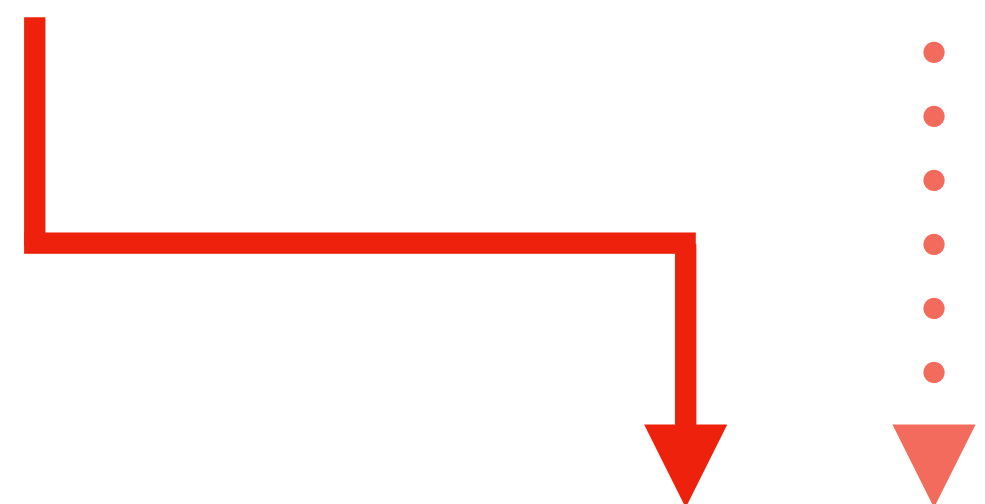
Tempo and delay

When increasing the playback rate we must take care **not to look in the past** for events which did **not yet** occur.

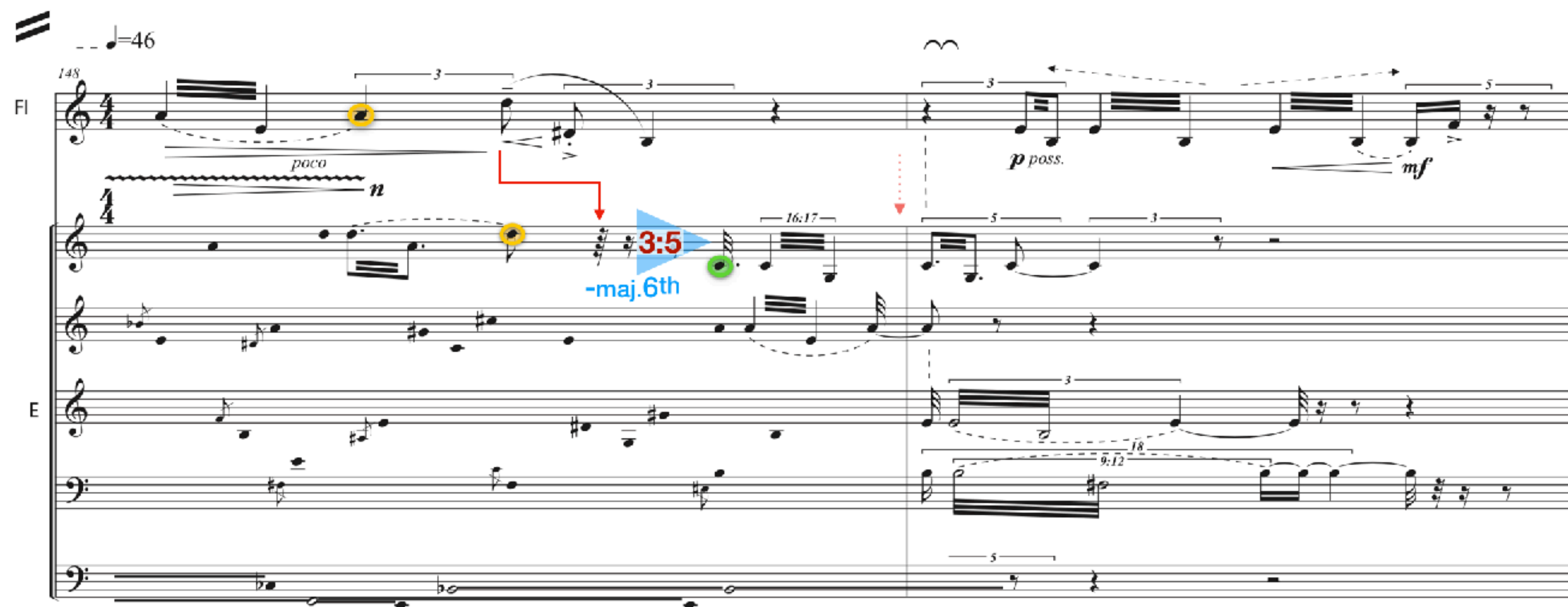
At a speed rate of 4:3 *Comes mutabilis* consumes time 33.3% faster than the original. In measure 148 it is about to pass beyond the time horizon. With a skip down   to a speed rate of 3:5 the voice leaps back, so to say, into the past (behind the original).



current offset



time horizon



Moto pendolare in canone I

$$t_r = t_e + (t_k - t_a) \times (1/s - 1)$$

where

- t_r the remaining time
- t_e the entry distance
- t_k the current time
- t_a the start time
- $1/s$ the inverse of the playback speed

Formula to compute the distance of equivalent time points.

current offset

next offset

Musical score for Flute I (FI) and Violin (VZ). The score is in 4/4 time and includes dynamics such as *f*, *p*, *mf*, *cresc.*, *ff*, *p*, *f*, and *ff*. A *rallentando* marking is present. Time markers t_a , t_k , and t_r are indicated with green brackets. The text "RM Einfärbung Coloring" is visible below the Flute I staff.

Musical score for Flute I (FI) and Violin (VZ). The score is in 4/4 time and includes markings such as *poco*, *p poss.*, and *mf*. It features triplets and other rhythmic patterns. A tempo marking of $\text{♩} = 46$ is shown at the beginning. The text "RM s" is visible below the Violin staff.

Moto pendolare in canone I

Excerpt m144-150

Although a polyphonic structure, it's not about distinct lines. This mirroring of the flute, with its crossing of wide intervals, aims at densification, entanglement and confusion, like a cloud of fog revealing the flute again when it passes by.

The image shows a musical score for an excerpt from 'Moto pendolare in canone I', measures 144-150. The score is written for Flute (Fl) and Violin/Viola (VZ). The Flute part is in the upper staff, and the Violin/Viola part is in the lower staff. The Flute part starts at measure 145 with a tempo marking of [♩=69]. The key signature is one sharp (F#). The time signature is 4/4, which changes to 3/4 in measure 149 and back to 4/4 in measure 150. The Flute part features a melodic line with dynamic markings: *f* > *p* (with a hairpin), *mf*, *cresc.* (with a dashed line), *ff*, *p*, *f*, and *ff*. A *rallentando* marking is indicated by a dashed line with an arrow above the staff. The Violin/Viola part is in the lower staff, with a 'VZ' marking. The score includes a wavy line labeled 'RM Einfärbung Coloring' between the staves. The bass line is also present, with a 'RM' marking and a '5' above it in measure 149.

♩ = 46

148

Fl

poco

n

p poss.

mf

4/4

3

3

3

5

3

5

3

16:17

9:12

18

5

Synchronisation and...

Baseline

- ▶ Antescofo follows pitch changes (considering energy).
- ▶ Latency is inherent in pitch detection.

Reminder

- ▶ Antescofo = *anticipatory score follower*
- ▶ If possible, align actions towards events in the future.

synchronisation

- ▶ perform something at the same time or rate
- ▶ align actions to events

anticipation

- ▶ imagine or experience an event in advance

latency

- ▶ time delay between cause and effect
- ▶ in FFT the duration of a frame (FFT size)
- ▶ the more precise (larger FFT size), the later

onset

- ▶ start time of the initial transient of a sound

...onset detection

To get information about onsets, an additional tool is needed.

IRCAM's MuBu/PiPo provides it.

Caveat

- ▶ Onset detection parameters must be adapted to characteristics of the sound to be analyzed.

- ▶ fast or slow attack?
- ▶ stationary sustain or fluctuations?
- ▶ how strong do events stand out from the background?
- ▶ ...

Onset detection

64 *mp* *sff* *f* *mf~ff*

lip tgp il veloce possibile (con licenza)

"PizzCadenza-1" "PizzCadenza-2" "PizzCadenza-3" "PizzCadenza-4"

In the Antescofo score of “Vertical Structure” the detection times of selected events are stored as offsets into the recording buffer, for later reference.

The four points shown here, for example, indicated by purple arrowheads, are named *PizzCadenza-1*, *PizzCadenza-2*, *PizzCadenza-3*, *PizzCadenza-4*.

66 *fpp* *p*

> lip tgp x kcl a tempo

"PizzCadenza-3" "PizzCadenza-4"

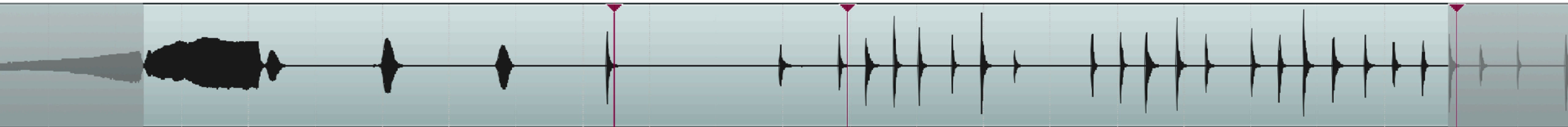
Due to the percussive character of the flute pizzicati, Antescofo’s detection lags a bit.

Onset detection

il veloce possibile (con licenza)

lip tgp

As our intention is to replay the air pizzicati in different order and timing...

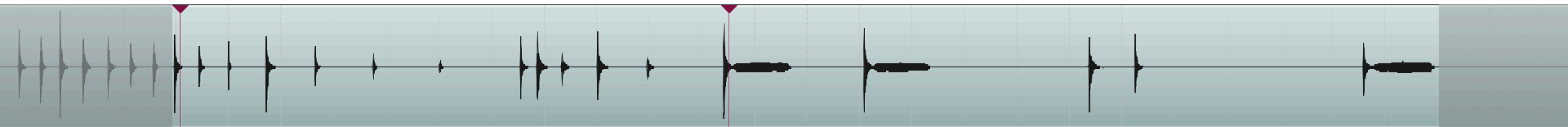


a tempo

> lip tgp X kcl

....more precise offsets into the recording buffer are needed, including all attacks in between the cued events.

That's where MuBu/PiPo comes into play.

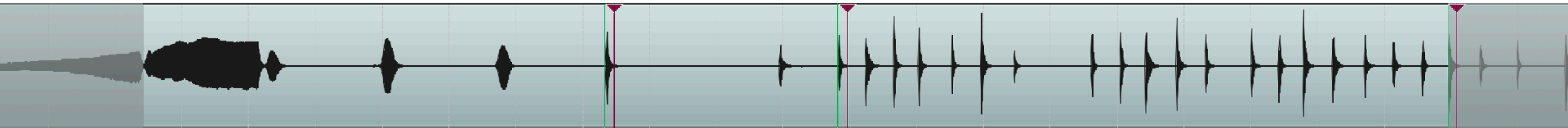


Onset detection

il veloce possibile (con licenza)

lip tgp

64 *mp* *sff* *f* *mf~ff* *mf* *f*



MuBu/PiPo 60ms

Antescofo 60ms

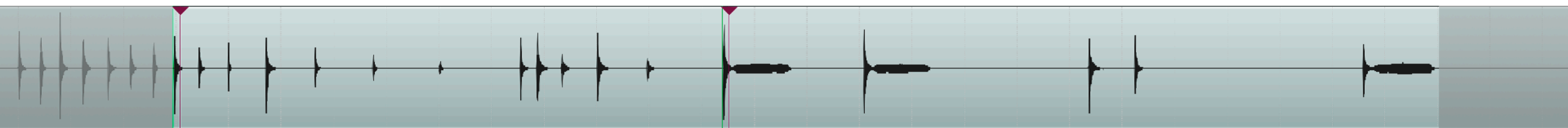
“Vertical Structure” interacts with MuBu/PiPo...

a tempo

> lip tgp × kcl

66 *fpp* *fpp* *p* lip lip

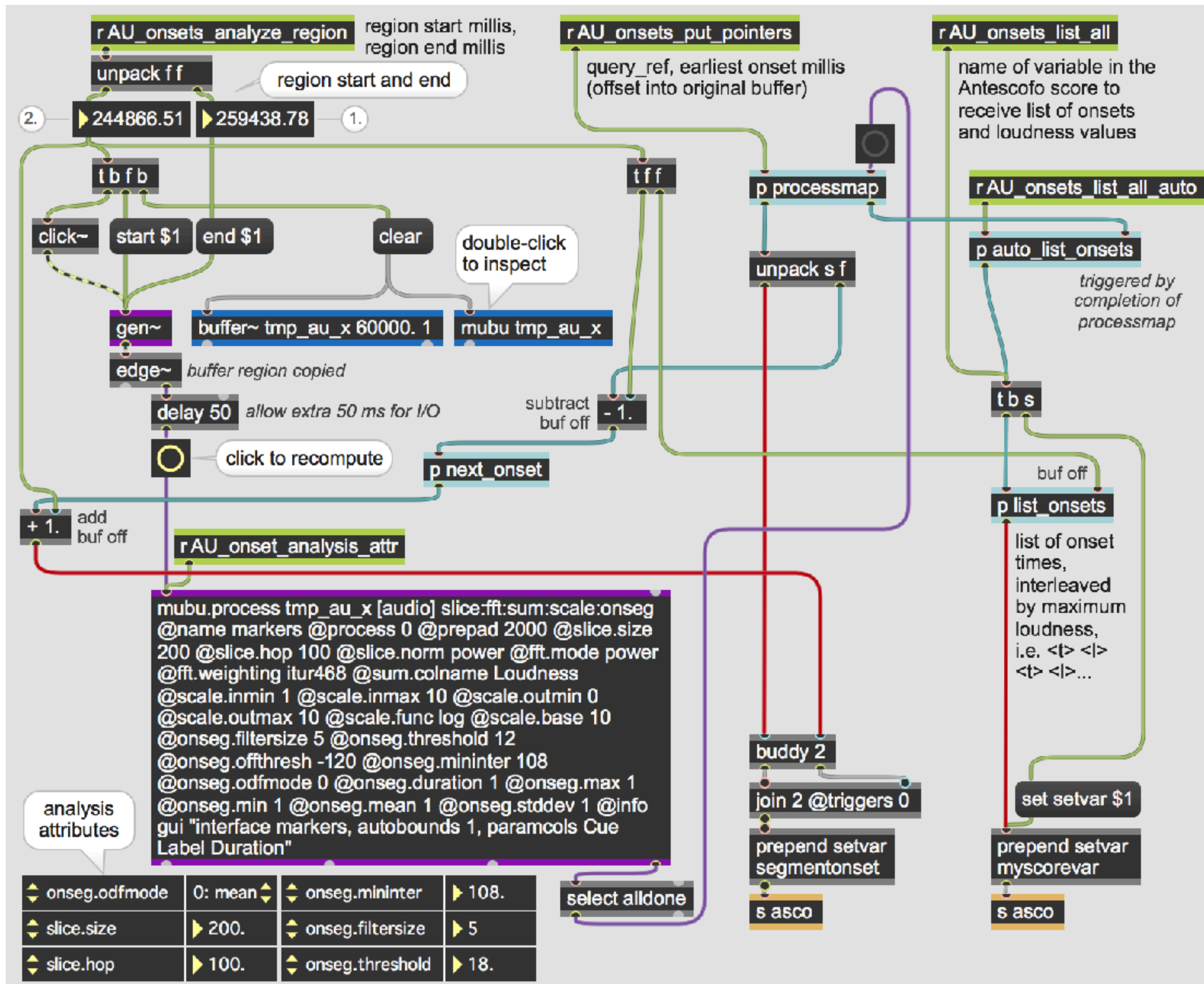
through a patcher which is a kind of query interface...



57ms

56ms

Onset analysis queries



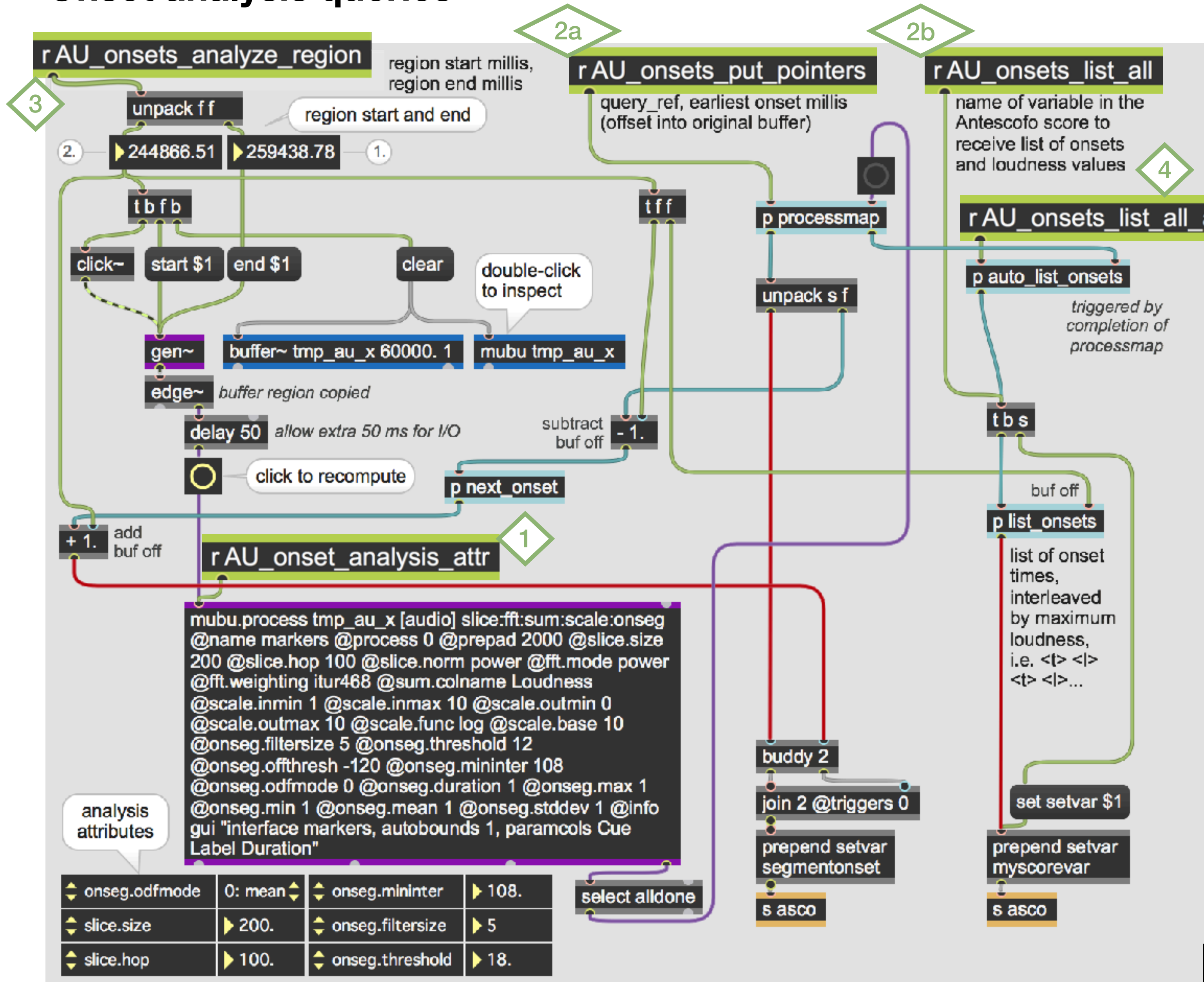
We send an analysis request to the patcher and receive the result in an asynchronous callback.

It's likely that analysis parameters (attributes) have to be adjusted. The values listed at the bottom left here are those used for the flute pizzicati.

References to be used in the callbacks may be registered before triggering the analysis. These may be key/value pairs which associate a symbol with an earliest onset time or the name of a variable to hold a complete list of all onsets found in the region. The latter may also be requested after processing the region.

The region is copied to a local temporary buffer and then processed by MuBu. On completion of the analysis all open callbacks are executed.

Onset analysis queries



1 eventually fine-tune analysis attributes (of pipo.slice and pipo.onseg)

2 a put a list of named pointers (results will be sent to variable segmentonset)

and/or

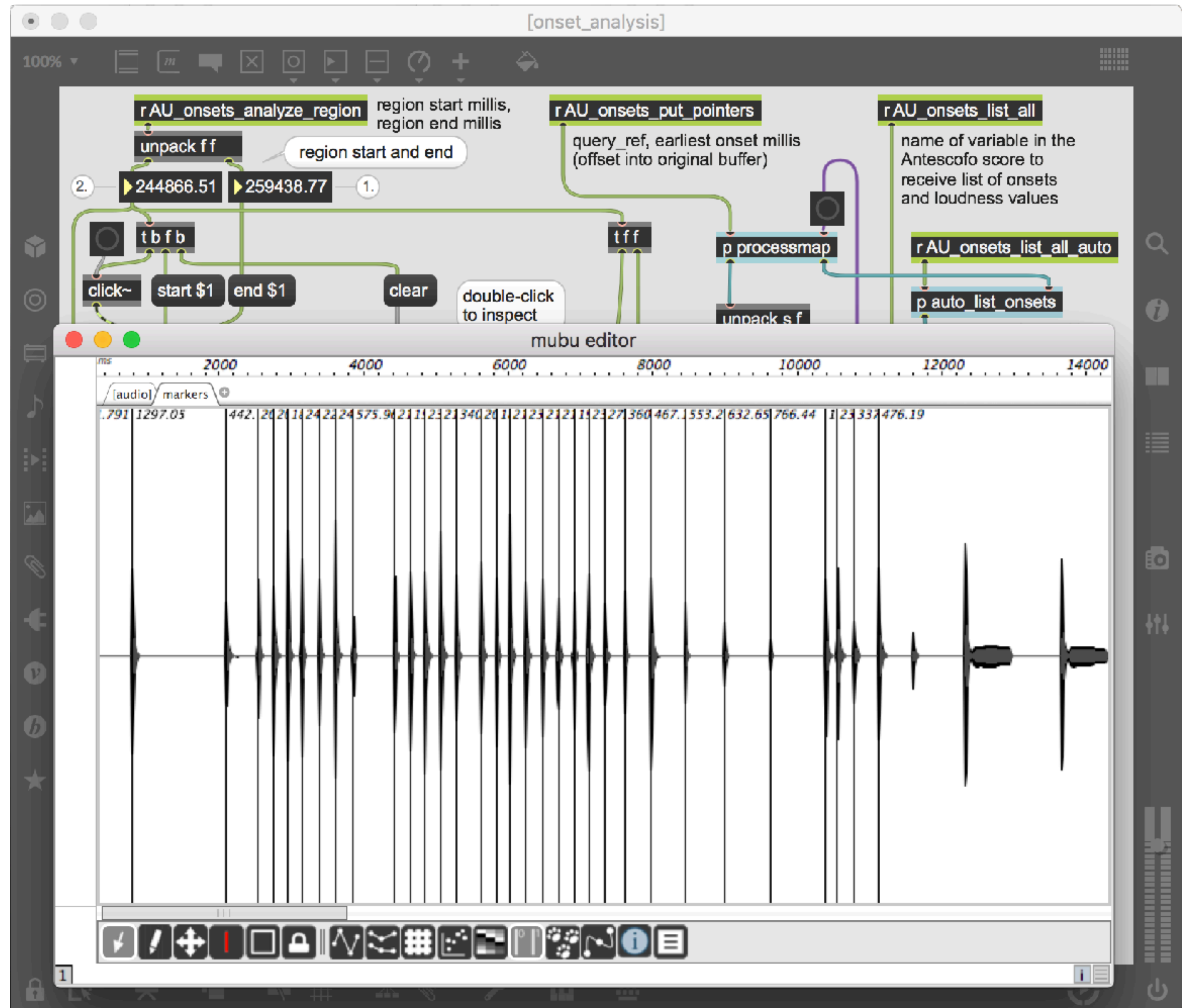
b to get *all* onsets in the region specify the name of a variable to hold the list

3 trigger analysis of a given region in the recording buffer

4 request immediate callback with the list of all onsets

Max patcher *onset_analysis*

Onset analysis queries



Max patcher *onset_analysis* with mubu window

Onset analysis queries

The mechanism in Antescofo which enables us to react to the variable being filled with the result of the query is the powerful *whenever* construct.

A *whenever* watches a variable. Execution of its body is triggered on every change to the variable.

Here, when the value of the variable `$pizzcad_onsets_raw` is updated with the list of onsets, the list is reshaped and reordered to create an accelerated and randomized recombination of the flute's pizzicati.

Antescofo score

```
}
NOTE 0 1/2
NOTE 0 2/3
NOTE E4 1/3 @pizz @staccato ; key-click
group OnsetRegionAnalyze_PizzCadenza {
  @local $rgn_start, $rgn_end, $pointers, $flattened
  let $pointers := tab ["PizzCadenza-1", "PizzCadenza-2",
                      "PizzCadenza-3", "PizzCadenza-4"]
  let $rgn_start := @au0ptr_pos("PizzCadenza-1") - @b2ms_r(1)
  let $rgn_end := @au0ptr_pos("PizzCadenza-4") + @b2ms_r(5/2)
  AU_onset_analysis_attr onseg.threshold 18 // not too many onsets
  let $flattened := @lace([ $pointers, @map(@au0ptr_pos_for_onsetsearch,
                                           $pointers) ],
                        (@size($pointers) * 2))
  AU_onsets_put_pointers $flattened
  AU_onsets_analyze_region $rgn_start $rgn_end
}
NOTE E4 1/2 @pizz @staccato ; <pizzicato> (lip)
NOTE 0 1
NOTE E4 1/2 ; <pizzicato> (lip) + ord.
@global $pizzcad_onsets_raw
group PizzCadenzaReplay {
  AU_onsets_list_all pizzcad_onsets_raw
  whenever PizzCadenzaReplay ($pizzcad_onsets_raw) {
    @local $earliest, $earliest_i, $i, $onsets
    let $earliest := @offms("PizzCadenza-1")
    let $i := 0
    let $earliest_i := -1
    let $onsets := @reshape($pizzcad_onsets_raw,
                          [@size($pizzcad_onsets_raw) / 2, 2])
    forall $x in $onsets {
      if ($earliest_i < 0 && $x[0] >= $earliest) {
        let $earliest_i := $i
      }
    }
    let $i := $i + 1
  }
}
```

Replay with onset data

In this passage the computer disregards the physiological conditions of a flute pizzicato...

▶ Excerpt m.63-70

[♩=46] lip tgp **il veloce possibile (con licenza)**

FI *mp* *sff* *f* *mf~ff*

E

* Bei den Pizzicati mit Vorschlagsnote den Tonhöhenwechsel so rasch wie nach dem Anblasen ausführen, sodass die beiden Tonhöhen wie ein Zweiklang erscheinen.

* The pitch change at the pizzicati with grace note is executed as fast as possible after the blown attack, so that the two pitches appear like a dyad.

a tempo > lip tgp × kcl

FI *fpp* *fpp* *p* lip

E

Replay with onset data

...to create an exaggerative parody of an occasional surge of fury expressed by the flute.

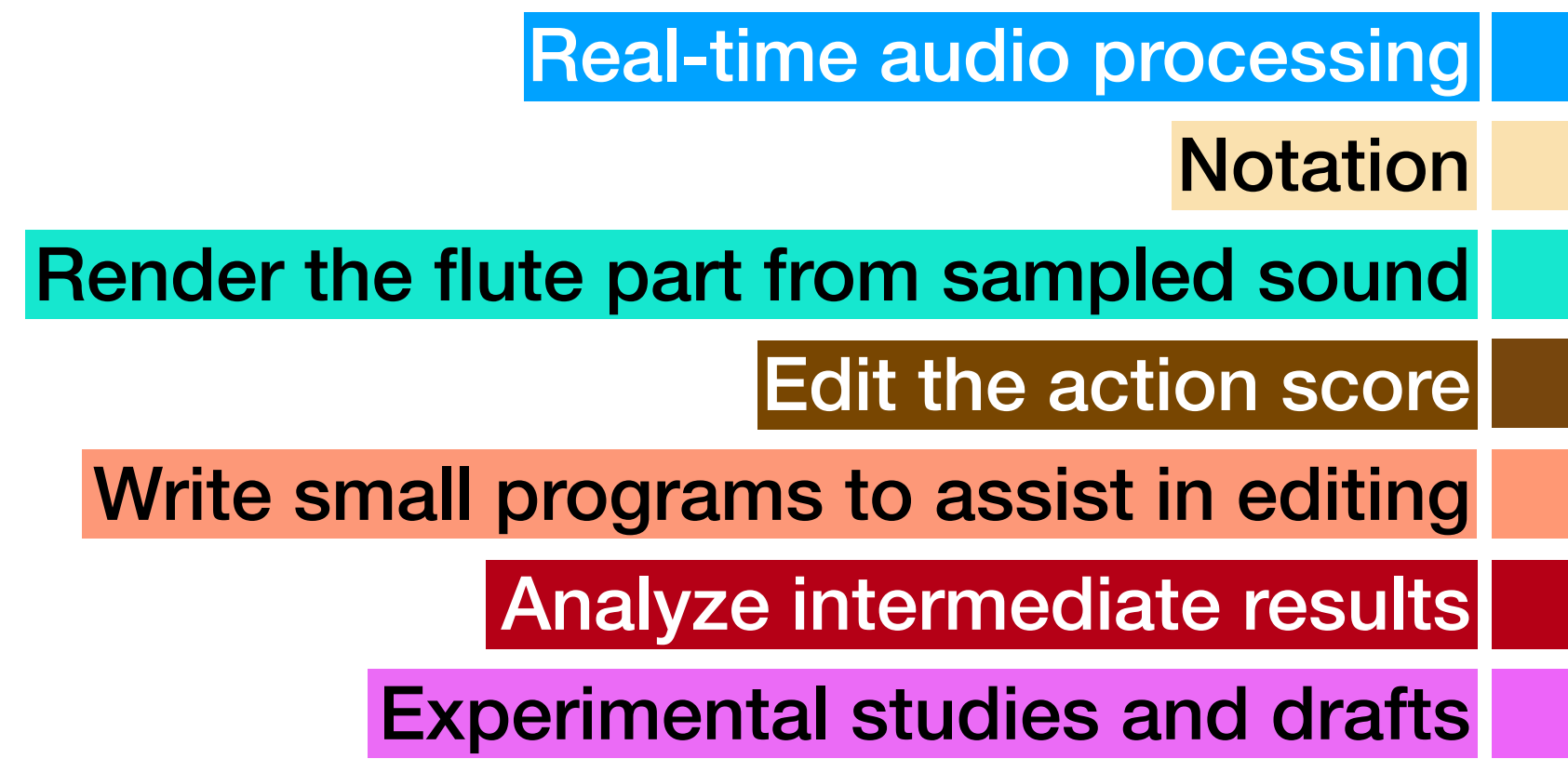
The image displays a musical score for Flute (Fl) and Violin (VZ) in 4/4 time. The Flute part begins at measure 68 with the instruction "lip" and a "2-3''" marking. The Violin part starts with "VZ" and "velocissimo". The score includes dynamic markings: *p poss.* for the flute and *poco* for the violin. A box containing "> x" is positioned below the violin staff, with a line connecting it to the onset data. The onset data is represented by a series of boxes and arrows: a box labeled "29" with an arrow pointing to a box labeled "48", which in turn points to a box labeled "5". A dashed vertical line is drawn at the end of the "5" box, corresponding to a measure in the flute staff. The score concludes with a double bar line and a 4/4 time signature.

Practical insights



With the exception of scribbled notes, *kritzeltettel* in German, and first sketches on paper, all my work takes place on a mid-2015 Mac Book...

Software used for “Vertical Structure”



Max/MSP

Externals and abstractions:

IRCAM Antescofo

IRCAM MuBu/PiPo

IRCAM Spat

Finale

Sample library:

IRCAM Solo Instruments

Sublime Text 3

with Antescofo package (syntax highlighting)

Eclipse (Java)

IRCAM AudioSculpt, Sonic Visualizer

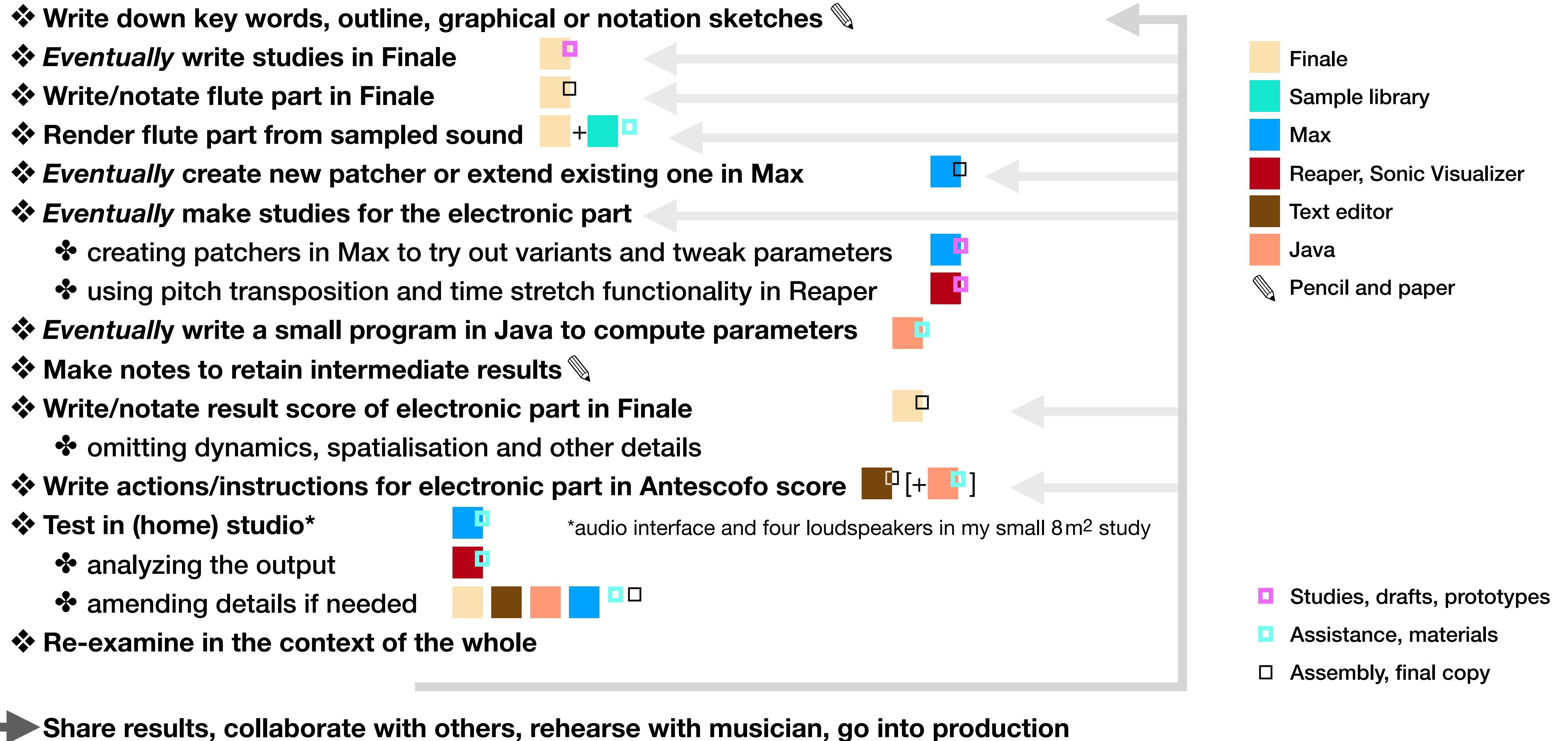
Reaper

A composer’s work today, especially when she employs sound processing and/or analytical or generative algorithms on the computer, not only takes place at a digital workstation but in fact resembles that of a software developer.

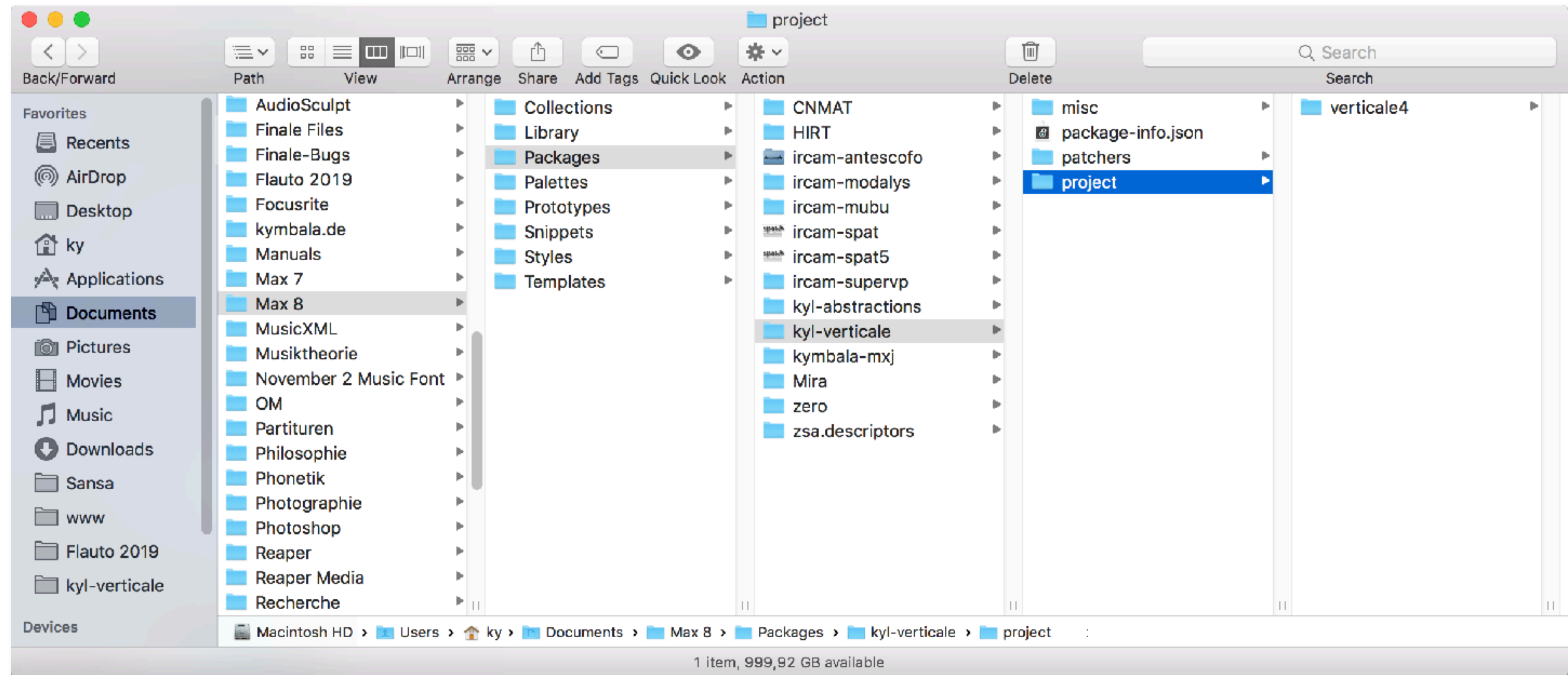
Typically, she uses – at different, partly iterative stages of the compositional process – a couple of different applications and, a lot of files are being produced.

It seems a good idea to adopt tools and best practices used in software development.

Typical work stages for a section/passage of “Vertical Structure”



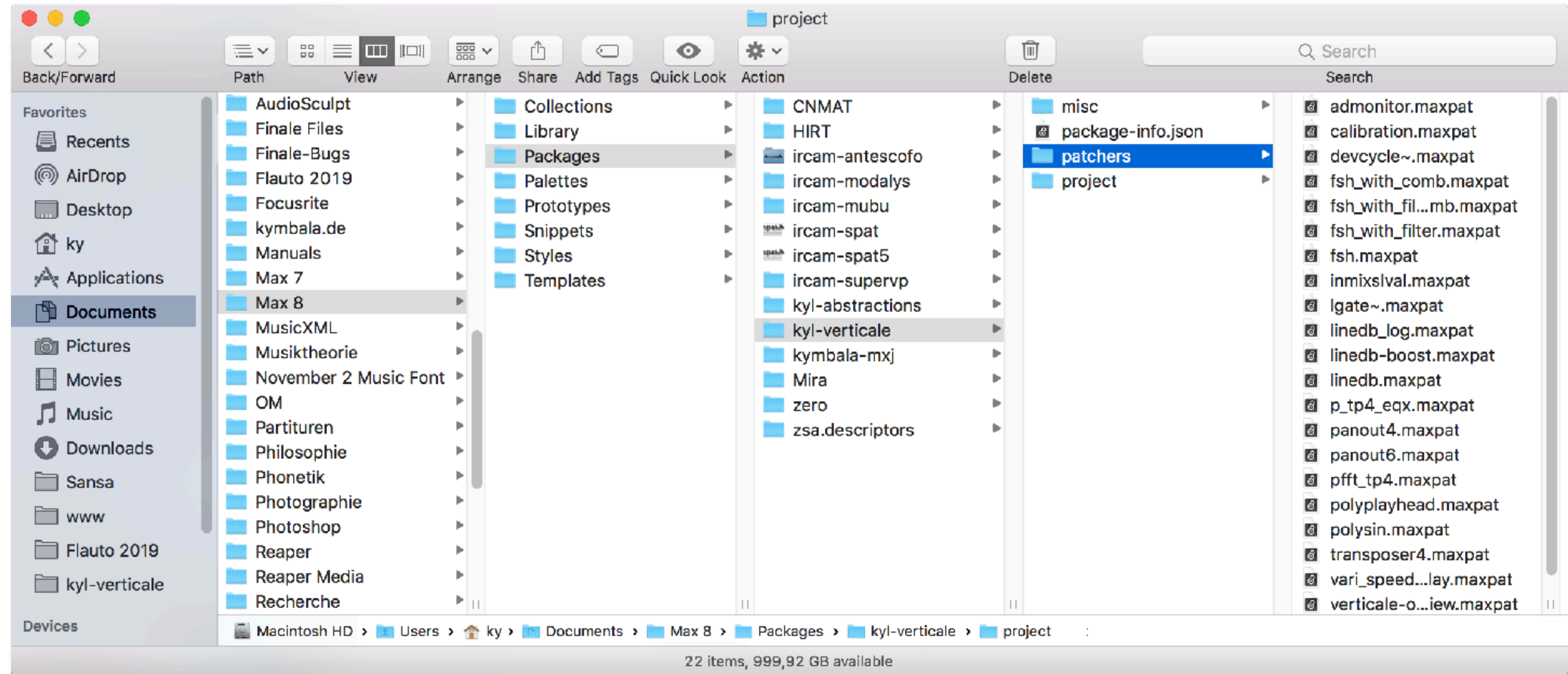
File organisation: All in one place



- ▶ Package kyl-verticale
- ▶ Project verticale4 (4-channel version)
 - ▶ Project inside package: all in one folder

I like the package and project concepts of Max and for “Vertical Structure” ended up combining both by placing the project folder inside a dedicated package for the project.

File organisation: Max patchers

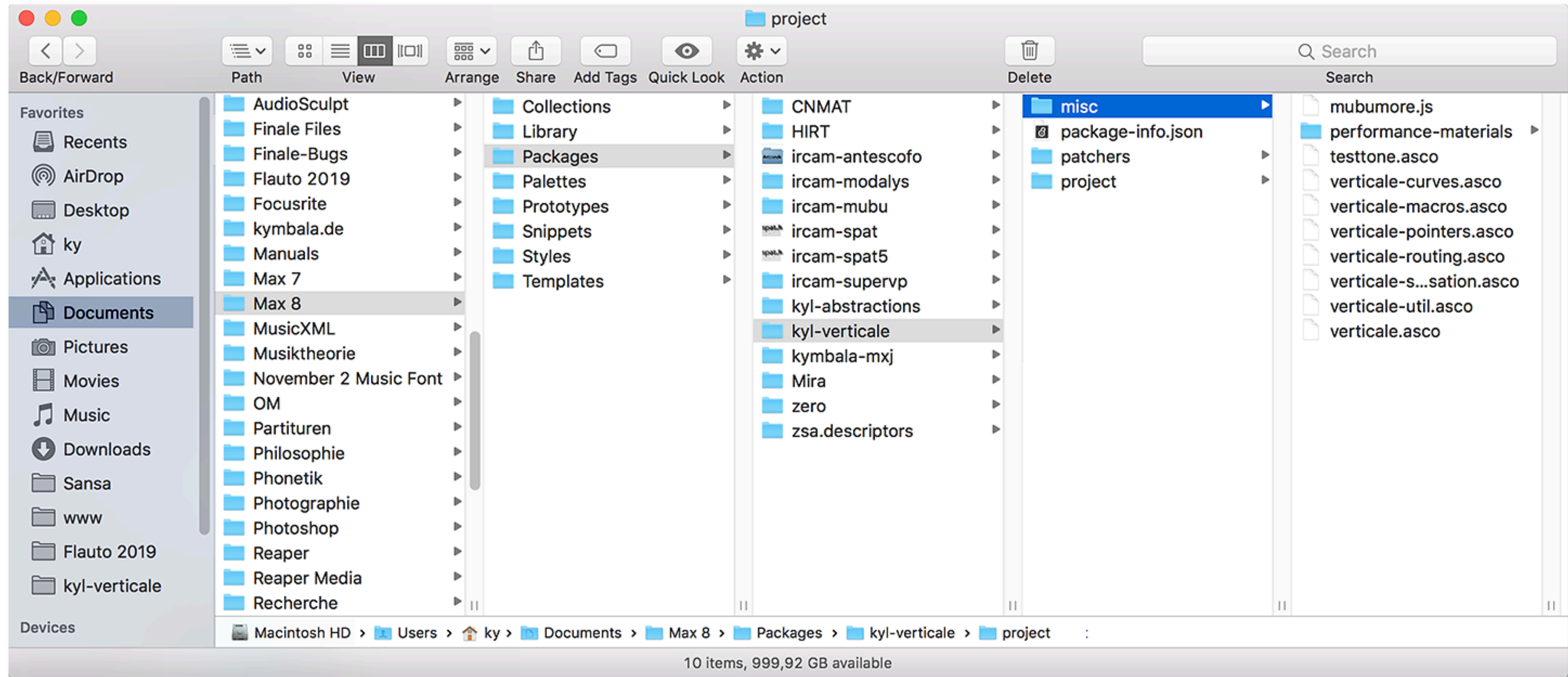


► Patchers

- Generic patchers in package `kyl-verticale`
- Main patcher in `project` (`verticale4.maxpat`)

Max patcher files are text-only files. Because of this, not only they can be shared easily but searched and compared with standard text-based tools, e.g. version control management.

File organisation: Miscellaneous



► misc

- Files used within but not edited with Max: Antescofo score
- Files not handled by Max: e.g. Finale files, documentation

You can put any type of file in a Max package. Max regards them as additional resources or ignores them.

Tracking changes with Git

Once you have all source files – Max patchers, Antescofo score etc. – in one folder, it's easy to set up version control management. The standard tool in the open source world today for tracking changes is Git.

Git tracks your changes and can show differences between revisions and, in the event, let's you go back to a revision. You can try out things on a branch and, if they work, merge the changes to the master tree.

If you're not into the console, there are graphical clients, and furthermore, the web interfaces of repository hosting services.

```
kyl-verticale — -bash — 93x42
Last login: Wed Jan 20 21:57:08 on ttys001
[Merlot:~$ cd Documents/Max\ 8/Packages/kyl-verticale ]
[Merlot:kyl-verticale$ git status ]
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   misc/verticale-macros.asco
        modified:   misc/verticale-pointers.asco
        modified:   misc/verticale-spatialisation.asco
        modified:   misc/verticale.asco
        modified:   patchers/fsh.maxpat
        modified:   patchers/fsh_with_filter.maxpat
        modified:   patchers/lgate~.maxpat
        modified:   patchers/panout4.maxpat
        modified:   patchers/pfft_tp4.maxpat
        modified:   patchers/polyplayhead.maxpat
        modified:   patchers/polysin.maxpat
        modified:   patchers/transposer4.maxpat
        modified:   patchers/vari_speed_delay.maxpat
        modified:   project/verticale4/patchers/verticale4.maxpat

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        patchers/calibration.maxpat

no changes added to commit (use "git add" and/or "git commit -a")
Merlot:kyl-verticale$ _ ]
```


Tracking changes with Git

- ▶ List changed files
- ▶ Commit (or discard) changes
- ▶ Track revisions
- ▶ Compare differences
- ▶ “Undo” on diverse levels
- ▶ Work on a copy of the file tree (branch)
- ▶ Merge (integrate) changes
- ▶ Save on a remote system (push)
 - ▶ Back up, implicitly
- ▶ Get from the remote system (clone, pull)
 - ▶ Work on different machines
- ▶ ...
- ▶ Clients on all major operating systems
 - CLI (command-line interface) or
 - GUI (graphical user interface)
- ▶ Repository hosting services
 - (public or private repositories)
 - GitHub, GitLab, Bitbucket a.o.
- ▶ Git official web-site ➡ <https://git-scm.com/>

```
kyl-verticale — -bash — 93x42
Last login: Wed Jan 20 21:57:08 on ttys001
[Merlot:~$ cd Documents/Max\ 8/Packages/kyl-verticale ]
[Merlot:kyl-verticale$ git status ]
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

       modified:   misc/verticale-macros.asco
       modified:   misc/verticale-pointers.asco
       modified:   misc/verticale-spatialisation.asco
       modified:   misc/verticale.asco
       modified:   patchers/fsh.maxpat
       modified:   patchers/fsh_with_filter.maxpat
       modified:   patchers/lgate~.maxpat
       modified:   patchers/panout4.maxpat
       modified:   patchers/pfft_tp4.maxpat
       modified:   patchers/polyplayhead.maxpat
       modified:   patchers/polysin.maxpat
       modified:   patchers/transposer4.maxpat
       modified:   patchers/vari_speed_delay.maxpat
       modified:   project/verticale4/patchers/verticale4.maxpat

Untracked files:
  (use "git add <file>..." to include in what will be committed)

       patchers/calibration.maxpat

no changes added to commit (use "git add" and/or "git commit -a")
[Merlot:kyl-verticale$ git add patchers/calibration.maxpat ]
[Merlot:kyl-verticale$ git commit -a -m "minor changes" ]
[master 0b3f0fe] minor changes
15 files changed, 7415 insertions(+), 4125 deletions(-)
create mode 100644 patchers/calibration.maxpat
Merlot:kyl-verticale$ _
```


Conclusion

The image shows a musical score for flute and timpani. The flute part is on the top staff, and the timpani part is on the bottom staff. The flute part includes a whistle sweep (wsl) at the beginning, followed by a time-stretched re-play with coloring ring modulators. The flute progressively emphasizes the air noise. The piece ends with a transition from the æolian sound to an ordinary low C. The timpani part includes a time-stretched re-play with coloring ring modulators. The piece ends with a transition from the æolian sound to an ordinary low C. The score is annotated with various musical notations and dynamics.

Annotations in the score include: *wsl*, *ppp*, *mp*, *ppp*, *poco*, *æol*, *ord*, *RM(← TP(← VZ))*, and *TP(← VZ)*.

The text box contains the following text:

The piece ends with a whistle sweep of the flute which is immediately echoed by a time-stretched re-play with coloring ring modulators. The flute progressively emphasizes the air noise. “Vertical Structure” is then closed by the transition from the æolian sound to an ordinary low C, a gesture which appears to me as if all sounds projected by the flute and mirrored by modulations returned into its body.

Conclusion

The image displays a musical score for a piece titled "Conclusion". The score is written for a flute and strings. The flute part is on the top staff, and the string parts are on the bottom staves. The score is divided into two systems. The first system is in 7/4 time and features a flute melody with a *ppp* dynamic and a *poco* marking. The second system is in 5/4 time and features a flute melody with a *pp* dynamic and a *poco* marking. The score includes various musical notations such as slurs, ties, and dynamic markings. Annotations include "wsl" (wind slurs) and "æol" (aerophone) markings. The score concludes with a fermata on the final note.

Flute part annotations:
- *ppp*
- *poco*
- *pp*
- *ppp*
- *poco*

String part annotations:
- RM(← TP(← VZ))
- TP(← VZ)

Other annotations:
- wsl
- æol → ord

Thank you for listening (or reading).

Supplementary material at

<https://kymbala.de/presentation/verticale/>

Kai Yves Linden

E-mail: kylinden@kymbala.de · Forum IRCAM User ID: [kyl](#)