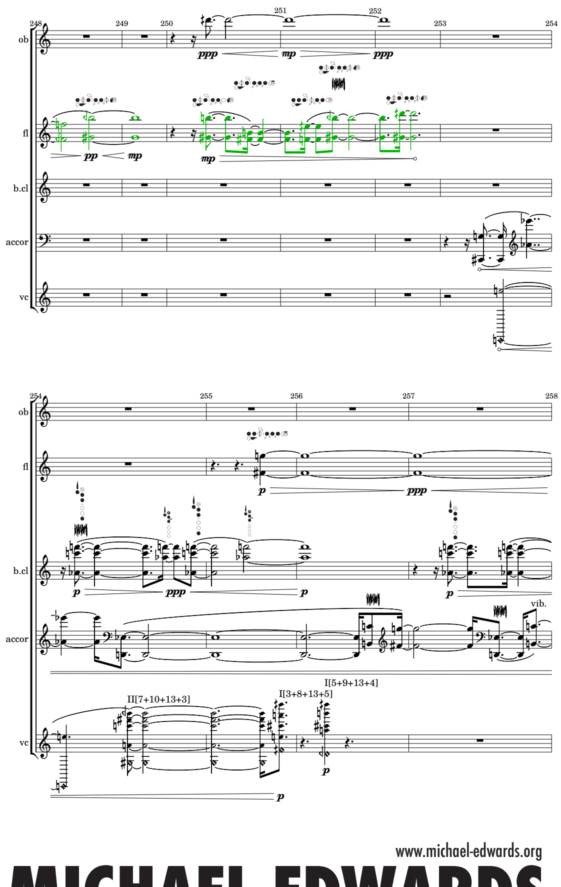
ME108

HYPERBOLES 6

“to prefer the destruction of the whole world to the scratching of my finger”

for oboe, flute, bass clarinet, accordion, cello, and computer



www.michael-edwards.org

MICHAEL EDWARDS

HYPERBOLES 6

"to prefer the destruction of the whole world to the scratching of my finger"

for oboe, flute, bass clarinet, accordion, cello, and computer

MICHAEL EDWARDS

michael edwards
hyperboles 6
“to prefer the destruction of the whole world to the scratching of my finger”
for oboe, flute, bass clarinet, accordion, cello, and computer

m@michael-edwards.org
www.michael-edwards.org

ME108

hyperboles 6
("to prefer the destruction of the whole world to the scratching of my finger")
for flute, oboe, bass clarinet, accordion, cello, and computer

Michael Edwards
m@smichael-edwards.org

October 18, 2023

Programme Note

Humans exaggerate on a regular basis. Typical hyperboles might be "this bag weighs a ton", "I'm so hungry I could eat a horse", or any of the invariably (!) hilarious "Yo' mama so fat..." jokes. Perhaps even worse than footballers ("I hit the post. I was gutted") are artists. Have you ever sat silently suffering ("dying", anyone?) at a contemporary poetry reading, as the reciter over-emotes their way through a litany of subtexts we can summarise by "me, me, me! I'm so deep and clever!"? Such occasions often merely reinforce the popular perception of artistic outputs as being expressive of the creator's emotions. But that is less interesting than artworks' invitation to be social, communal, and at the same time to introspect and inspect our personal, perhaps emotional reactions to intrinsically neutral objects. Hence this neutral, utterly calm aesthetic object for your perusal now.

The hyperbole relevant to *hyperboles 6* comes from David Hume's *A Treatise of Human Nature* (1739):

"Where a passion is neither founded on false suppositions, nor chuses means insufficient for the end, the understanding can neither justify nor condemn it. 'Tis not contrary to reason to prefer the destruction of the whole world to the scratching of my finger. 'Tis not contrary to reason for me to chuse my total ruin, to prevent the least uneasiness of an Indian or person wholly unknown to me."

hyperboles 6 was written expressly for *Ensemble S201*. Other works in my *hyperboles* series are, in order, for flute, violin, saxophone quartet, computer-controlled compressed air instrument orchestra, and cello. For more information see http://bit.ly/1Q2bgFr

Score Interpretation: General Points

1. Red notes: unstable sonorities. These are so quiet that for the notated duration we move between unstable, extremely quiet sounds and silence, as the performer(s) tries to make audible the most tiny sounds possible. These are used at the beginning of each section.
2. Green notes: These indicate parts of the piece where the performer may deviate from the score with improvisational interventions. The performer may stick to or deviate from the indicated rhythms/pitches/dynamics as they see fit, but materials inserted here should not interrupt the character of the piece too much and definitely should not rise too much above the prevailing dynamic. E.g. in the flute version, as several of the sound files contain amplified mouth noises, these may be a good addition to the textures in these improvised interventions.
3. Pauses may be inserted in the performance at any point; pressing the space bar halts the MIDI sequencing (see below).
4. Each section and subsection (i.e. 2 and 2.2 but not sub-subsections such as 2.2.2) is accompanied by an Indian Rasa and associated colour. The Rasa, or mental state, should be internalised by the performer(s) and used to set the mood of the music that follows. This should be a subtle effect however. At no point should histrionics or extreme movements or dynamics be used to portray the Rasa, even in the improvised interventions. Where possible, the indicated colours should be used

in the selection of light projected onto the stage and/or performer. A light mixer with RGB (red, green, blue) and white faders is useful here if no light automation is to be used. The colour black can be interpreted as no coloured light; grey then is a little white; yellowish is a little red and green in equal levels, with yellow being more of both. The iPad Mira software (see below) will show when these colours are to change (or timings/cues can be used instead; for timings see the colours text file for the version you are playing, e.g. data/hyperboles-flute-colours.txt). Once the colour changes, a slow fade should be set in motion: please do not change colours abruptly.

5. Versions for solo instrument and computer, or in the case of *hyperboles 6*, for the oboist:
    1. Following the example of (or perhaps unashamedly stolen from, depending on your point of view) Samuel Beckett's *Not I*, a narrowly-focussed spotlight should be aimed at the performer's mouth (no more than this) throughout the performance. However, the performer may move in and out of this light *ad lib*, i.e. they are not expected to keep their mouth in the light beam throughout the complete performance. The colour of this light may change along with the Rasa (see above) or remain white, according to the desires of the performer/set designer and the resources of the performance space. In the premiere (Splendor, Amsterdam, 14th June 2014; Anne La Berge, flute) we used a white halogen light bulb taped to the microphone stand, with a little tape blocking out light leakage to the audience and focussing the beam onto the performer's face when she stepped into it.
    2. At three points in the piece a mouth action instruction is given in a box ("bare teeth", "stick out tongue", "smile inanely"). In each case this should be performed with the mouth in the spotlight in a neutral, clear, but unexaggerated manner. These instructions have no intended meaning whatsoever, rather, they may or may not provoke a variety of responses in the listener, depending upon their sympathies towards the performance and their individual experiences and characteristics.
    3. At the end of the piece, after a significant pause, the player is invited to ask the audience "Is that enough, or do you want more?". There is a sound file of Anne La Berge asking this question; this can be triggered by the performer by pressing the 'i' key on the computer, if they prefer. What to do if the audience requests more is left to the discretion of the performer.

Oboe-Specific Notes

There are only three high notes in the whole piece. As such the oboist's role is curious and, given the role of the spotlight and mouth actions (see above), rather theatrical. It would be best to place the oboist outside the semi-circle of the rest of the ensemble, perhaps seated higher and further back from them. Like the other instruments though, the oboe should be amplified.

Flute-Specific Notes

Multiphonics

The pitch structure of this piece is formed by transitions between a user-selected permutation of the fundamental pitches F, F#, G#, and A, as multiphonics, followed by low C and high G. For each of the four fundamental pitches there are two possible multiphonics (or in the case of G#, six) which will be cycled through as each pitch is selected. This results in a total of 12 multiphonics, as shown below. Similar constructions apply to the bass clarinet and cello parts.

Score Interpretation

1. Purple notes: These notes are to be sung gently along with the indicated multiphonics/single notes. If a sung note is difficult to execute, for example in conjunction with the G6 harmonics, establish the note(s) first then add the voice tentatively and explore the resulting sonority.
2. Diamond note heads indicate the fundamental note from which the high G should be played from. This will be one of low C (C4), E flat, G, or the 2nd C (C5).
3. In the body of the multiphonics there are three spectral areas which the performer should try to bring out according to the indicated context:
    1. the lower part of the spectrum, i.e. the notated pitches (indicated by normal noteheads);

2. a higher noise band (indicated with an erratic scribbled line over the notes);
3. a very high whistle tone (indicated by a sine wave with an upward pointing arrow).

One spectral area is focussed upon in each of the three sections, the order of which is user-selectable. The other two spectral areas will be used to a lesser degree in each section, with more changes of focus as the elevation increases (see Terrain Data above). Please note that at no point should we hear an ascending/descending whistle tone harmonic series, for example, on low C when trying to achieve the high whistle tone or any other tone for that matter.

Bass Clarinet-specific notes

As with the flute and/or cello parts, in the body of the indicated multiphonics there are three spectral areas which the performer should try to bring out according to the indicated context:

1. two circled arrows means **alternate**: move the emphasis between the sounding pitches gently and delicately but also constantly;
2. an erratic scribbled line over the notes means introduce a higher noise band into the tone through increased breath sound but avoid any beating/dissonant sounds;
3. an absence of the previous two marks indicates **balance**: try to attain a static mix of the multiphonic pitches.

Accordion-specific notes

1. "vib." = vibrato. This should always be light but fast, though speed and depth may be varied ad lib.
2. an erratic scribbled line over the notes means focus on the **noise** parts of the spectrum, possibly by allowing air sounds from the bellows to dominate more than the pitches realised/notated normally.

Cello-Specific Notes

The cello part of this piece is developed from *hyperboles 5* (cello and computer), written for Ellen Fallowfield. In particular it concentrates on Ellen's work developing and notating cello multiphonics. This is fully documented online at www.cellomap.com. Ellen's overview of how these are created is given below:

> "Multiphonics seem to occur when harmonics with nodal points closely situated on a string are encouraged to sound together by making some compromises with left-hand position, bow speed/pressure and point of contact. In other words, the player finds a left-hand position, bow speed/pressure and point of contact that is 'acceptable' enough to several harmonics to enable them to sound simultaneously."

Most of the notes in the cello part are, as with the flute and bass clarinet, multiphonics. For the cello, these are indicated by a diamond-headed note at the nodal point, as with harmonics. The sounding partials of the multiphonic are indicated on the same note stem above the diamond head, as smaller normal note heads. In addition, following Ellen's notation, above the note and any other symbols (see below) there is a textual indication of the multiphonic. This includes the string and the resulting partial numbers, e.g. II[7+10+3].

At points in the score where several multiphonics are linked into phrases it's important not to rush position and fingering changes. A leisurely quasi-glissando is implict in all such changes and in no sense are we expected to hear crystal clear legato tone exchanges.

At very quiet points, e.g. in the red and sections marked "tranquillity", and particularly at the beginning, *col legno tratto* is highly recommended.

Score Interpretation

1. *pizz* is not cancelled by *arco* as it only ever applies to the single note it is placed over. Some pizz. multiphonics will of course be impossible to achieve in spectral terms. In this case proceed as if a bowed multiphonic were about to be played but execute a (possibly quite dull-sounding) pizz. instead, accepting the results without forcing particular pitches to be realised. (The placement of *pizz* is another process controlled by the terrain data: the higher the elevation, the more short notes are to be played pizzicato.)

2. Be careful with the dynamic of quiet pizzicati: using the flesh of the thumb might be better for producing very quiet notes.
3. flaut. = flautando; sp = sul ponticello
4. There are three essential ways of playing the multiphonics in the score. As with *pizz*, the signs used are not cancelled, rather, they apply only to the note they are directly above.
    1. an erratic scribbled line over the notes means focus on the **noise** parts of the spectrum at the expense of the sounding harmonics;
    2. two circled arrows means **alternate**: search through the audible harmonics, moving from one to the other;
    3. an absence of the previous two marks indicates **balance**: try to attain a static mix of the harmonics.

Essential Hardware for versions with computer

1. Macintosh running OSX 10.7.5 or above
2. High-quality external sound card with 8-channel output (quad or stereo also possible)
3. High-quality condensor mic(s) for each of the acoustic instruments

It is envisaged that solo performers will read the score from the computer screen and thus be able to follow the bar numbers and other interface elements as the performance proceeds. The software will turn pages automatically if desired (see below).

Optional Hardware

1. Additional microphones, clip-on or otherwise
2. iPad (retina display) running Mira software (from cycling74.com; see below) *or* faderfox FX3 or similar MIDI controller (mapped to device ‘a’ in the MIDI Setup of MaxMSP)

The piece may be performable without the help of another person but it would be useful to have someone balance levels during the performance with either a MIDI controller or an iPad.

iPad + Mira The iPad Mira interface is particularly useful for controlling the levels during performance as it allows someone in an audience position to do this wirelessly. The hyperboles.app (or MaxMSP) software sends two frames to Mira: one for the lighting control, the other for audio level balance (so two iPads might be useful if the lighting engineer is remote from the person mixing).

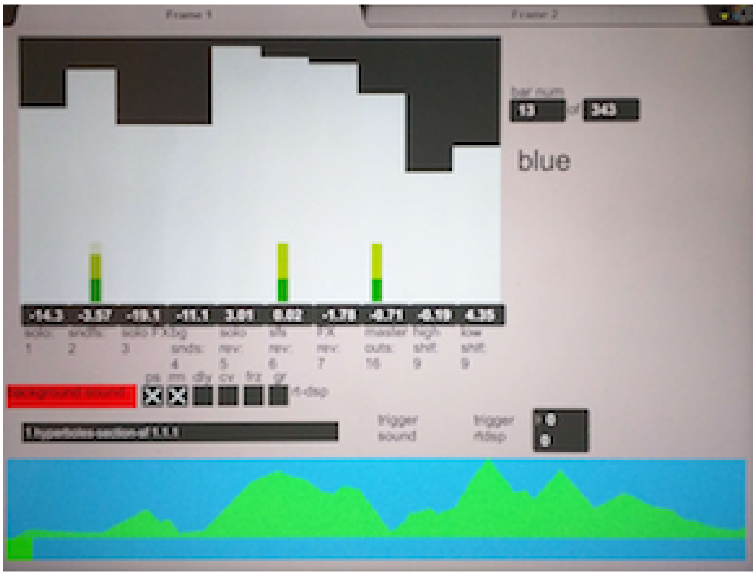


Figure 1: iPad Mira control

As with the MIDI faders, once the initial preset is loaded, the interface dial might be out of line with the MIDI controller and/or Mira fader. In order to avoid sudden parameter changes, the software dials will not reflect the MIDI controller or Mira fader levels until the latter reaches the stored value.

Particular attention will need to be paid to the sound files and the instrumental FX levels during performance as either of these might— depending upon the configuration and performance context—become too loud.

Diffusion

The 8-channel outputs are in the following order:
1 2
3 4
5 6
7 8

If a 4-channel performance is required, make sure you generate 4-channel sound files and then simply ignore output channels 3-6. If stereo is required, map (sum) 7 onto 1 and 8 onto 2 and optionally regenerate stereo sound files (but 4-channel files will work fine in this configuration).

The solo signal is sent to all 8 outputs. The *solo attenuation* menu sets the gain reduction in decibels as they are sent to outpus 3/4, 5/6, and 7/8. The *initial distance* sets the distance in meters from the soloist to the front speakers; the overall distance sets the distance in meters from the front speakers (1/2) to the back speakers (7/8). These distances are used to set delays.

As all the mixing is done from the iPAD/midi faders/computer interface, nothing more than overall level control should be necessary by the sound engineer.

However, if the musician will perform the piece without someone controlling levels via the iPAD/midi faders/computer interface, it may be beneficial to have a separate microphone amplifying the soloist independently of MaxMSP. In this case the sound engineer would balance that level with the four outputs from the computer (which should probably avoid sending the solo signal, due to the signal-delay interferences this might create).

For a better sonic image, you might feel the need to add delays to the main outputs 1-8. To set the delays edit the text file main/data/main-delays.txt. The eight milliseconds values after “1,” map to channels 1-8, in order.

Essential Software for versions with computer

This piece is dependent on a number of different software packages. It currently runs only Macintosh computers; it is optimised for a screen resolution of 1280x800 pixels. If you just want to play a pre-configured version of the piece all you will need is hyperboles.app or MaxMSP plus the patches and files which this documentation forms a part of.

zsa.descriptors

The MaxMSP software uses the zsa.descriptors externals for real-time audio analysis. This will need to be installed separately from http://www.e–j.com/index.php/download-zsa/ The free version is fine for our non-commercial purposes here. Essentially you need to unpack the zip archive and copy the whole zsa.descriptors folder to Documents/Max 7/Packages in your home folder, then restart Max or the app. If everything is OK you should see a zsa.descriptors splash screen when you load hyperboles.

Since Max 7 zsa.desciptors is available via Max’s Package Manager. If you’re using the hyperboles app, zsa.descriptors comes preinstalled.

slippery chicken

The essential software infrastructure is based on communication between MaxMSP and the composer’s algorithmic composition package *slippery chicken*, written in the Common Lisp programming language. If you wish to experiment with the underlying algorithms and generate your own version of the piece you will need hyperboles.app (or the Max patches plus *slippery chicken*). *slippery chicken* comes packaged with hyperboles.app.

Should you need it, you can download the *slippery chicken* app (free) at: http://www.michael-edwards.org/sc/source.html (Please read the _readme.txt file which comes with the app in order to download further software necessary for the installation.) Once the app is installed and running please update to the latest software release by typing (update-app-src) at the Lisp prompt (see http://michael-edwards.org/sc/robodoc/utilities_lsp.html#utilities2fupdate2dapp2dsrc for more details).

lilypond

If you are experimenting with the algorithms you will also need the free and open source music notation software Lilypond: http://lilypond.org/download.html

Running the Software

If you only want to perform the pre-configured version of the piece just start hyperboles.app or open the patches in MaxMSP.

Once hyperboles has started, click the open button at the bottom left next to DSP, and make sure the sampling-rate is to 48KHz. Select the instrument version from the big bold drop-down menu at the top. When you are ready to begin, press the 's' key to start the countdown to the beginning of the piece. (You might also want to check the Max window to make sure no errors have occurred. There should be no "sflist~: can't find file. . . " errors, though you might see these when configuring a new piece; if you do, just restart hyperboles and try again without regenerating the sound files.)

NB. If selecting the instrument from the menu causes the progress bar along the terrain to stop working, you've probably got Lisp running and listening via OSC: quit Lisp, restart the software, load the instrument and the progress bar should work.

If, on the other hand, you want to "roll your own" (i.e. make your own version of the piece) and you're not using the app, start the *slippery chicken* app. Once you get to the SC> prompt, type the following Lisp call:

```
(osc-call)
```

Now start hyperboles, and set the sampling-rate to 48KHz as detailed above. Press the 'algorithm tweaks' button. If you're using the app you can start *slippery chicken* by pressing the 'start lisp' button. Now select a preset, etc.: see the instructions below. Once you're happy press the **generate piece** button. The patch window will turn red whilst Lisp is doing the necessary calculations. Once the patch is back to its original colour, press the **generate score** button. Again, the patch will turn red while Lilypond renders the score PDF. Once it is ready, you will see the score in the window to the right of the main patch. You can turn pages by using the buttons on the patch or the left and right arrow keys. Before you can perform the piece you will also have to render the sound files by pressing the 'generate snds' button.

Each time you start hyperboles it will load the last generated version of the piece.

NB. Do make sure you save the main audio control patch after generating a new version of the piece, so that various states are saved.

As the electronics in the piece are 8-channel, you'll need to configure a stereo version if you're practising with headphones or two speakers. Click the open button near the DSP toggle (bottom right of audio control patch), then click the I/O mappings button at the buttom right of the Audio Status window. Set Output channels 3,5,7 to Output 1 and Output channels 4,6,8 to Output 2.

Algorithm Interface Details

The **global proportions** menu sets the proportions of the global structure (see the Structure section below).

The **starting tempo** sets not only the tempo with which the piece begins but the tempo from which all other tempi are proportionally derived.

The **starting rhythm/voice permutations** menu sets the rhythm sequence permutation which begins the piece. Changing this will result in different rhythmic material being allocated to the solo instrument and three computer parts throughout the piece. See the *Rhythm Sequences and Terrain Data* section for more details.

The **starting note order permutations** refers to the ordering of the basic pitches which form the background to the piece. For instance, in the flute version, the pitches in the drop-down menu are the lowest notes of the multiphonics we use. A4 refers to middle A, FS4 is the F sharp below, GS4 is G sharp, and F4 is F. So the first note will form the fundamental starting multiphonic of the piece. In all versions, a transition takes place over the duration of the piece through the ordering of these notes and then on through to some single pitches, low C and high G in the flute version. For each of the four given pitches there are several ways of playing them which will be cycled through as each pitch is selected in sequence (e.g. in the flute version, two to six possible multiphonics for each basic pitch).

The **terrain** menu sets the elevation curve data that will be used in several processes and as described in the Terrain Data section below. If the MaxMSP<–>Lisp connection is running, the newly selected terrain will be displayed on the main audio control patch. If the curve is then saved as part of a preset on the main audio control patch, it will be displayed next time the patch is opened, whether the MaxMSP<–>Lisp connection is running or not.

The **terrain exaggeration** value provides a way to process the terrain elevation curve. Positive values will result in elevations being pushed away from the middle point of the curve, thus exaggerating, or increasing steepness of ascent/descent. This will result in more dramatic changes of, e.g., note activity in all parts. Negative values will tend to level out changes and thus result in a more uniform piece.

The **focus transition** menu determines the order of the manner of playing the notes/multiphonics for the three sections; in the flute version this means the dominant multiphonic spectral areas. See the Terrain Data and Score Interpretation sections below for more details.

The **rests curve minimum** value will set the minimum value of the scaled terrain curve when it is used to turn notes into rests (as described in the Terrain Data section below). The lower this value is, the more rests will be inserted (in particular at the beginning).

The **maximum no. of repeats** and **minimum no. of repeats** values set the min./max. number of repeated bars 'stuck on' at the end of every sub-subsection. See the Terrain Data section below.

The **change repeat bar** tick box determines whether the bar to be repeated will change on a sub-subsection boundary or not. The effect of this will only be noticeable once the number of repeats is high enough (i.e. longer than a complete rhythm sequence, e.g. towards the end of the piece).

The **presets** section allows you to save interface configurations. Once you have set the parameters, you can hold the shift key and click on any of the little round buttons in order to save state in that place. To recall a state, you simply click on the associated button. You will be prompted to save the presets file if you try and quit without saving. You can also save your presets in a file of your choosing by pressing the **save presets as** button. Similarly, you can load a previously saved presets file by clicking the **load presets** button. The **save presets** button saves to the default file for the current instrument; this will be reloaded automatically upon starting the patch. To re-initialise the piece to the values determined by the composer and specified in the Lisp code file, click the **reinitialise** button.

N.B. Once you have configured and generated the piece with Lisp interaction, remember to save the preset via shift-click then "save presets"; this is also a good time to store (as preset 1, so it's loaded automatically next time) and save presets on the main audio control patch (by default this should be hyperboles-instrument.json e.g. hyperboles-flute.json). This way the terrain graphic and status bar in the main window and on the iPad will display the correct current data.

The **staff size** number controls how large the generated notation will be. The higher the number, the larger the notes.

Once you have configured the piece, you must click on the **generate piece** button to create the notes and other data, as well as the MIDI file. You can then choose to generate the score—this will be displayed in the score window once rendered by Lilypond.

Click on **generate snds** in order to generate the sound files for the piece. These may or may not have already been generated, depending on whether you're creating a new configuration or not, but only those which have not been generated will be created here. Note that this is a computationally intensive operation so may take a long time, perhaps as long as 20 minutes. Note that the sound files are all generated from samples kindly provided by the following musicians:

1. Flute: Anne La Berge, Amsterdam, August 18th 2013
2. Violin: Mieko Kanno, Edinburgh, June 30th 2014

3. Soprano saxophone: Gianpaolo Antongirolami, Macerata, Italy, June 22nd 2014
4. Cello: Ellen Fallowfield, Edinburgh, January 28th 2015

The samples are also used in the simulation of the solo part in the different versions of the piece. They are not to be repurposed without the express permission of the performers and the composer.

When the score is rendered as a MIDI file, the solo part will be on channels 1 (with microtones on channel 2) and the computer parts spread over channels 3-8.

The **sine amplitude** value will determine how much sine wave—at the frequencies of the pitches indicated in the computer parts of the score—will be mixed with the samples when we generated the sound files. 0.001 is a good starting value as the sines should only be subtly present.

The **load score** button allows you to display a previously generated score PDF file in the score window.

You can change pages in the score by clicking on the two "page +/-" buttons at the bottom of the algorithm interface or by using the left and right arrow keys on your computer keyboard. Alternatively, MIDI notes 0 and 1 on channel 13 (the first two green buttons—top left—on a faderfox FX3 MIDI controller) will perform page back/forward.

As the performance is driven by a MIDI sequencer (see below), we can have the patch automatically turn pages for us by clicking the **auto-page** toggle (which is on by default). Of course, each slight variation in algorithmic parameters will result in a different score. As it's not possible to know in advance how Lilypond will lay out the bars, in order for automatic page turns to happen, click the **enter page turn bars** and edit the bar number list to reflect the first bar number of each score page.

If **auto-page** is on, then pages turns can also be made on an iPAD running the *forScore* app. This is accomplished by the Max patch sending programme out messages 1 and 2 (previous, next) on channel 16 of MIDI device 'n' (Max options / MIDI setup). This will need to be enabled in *forScore*: go to Settings / Page turners & shortcuts, press Previous Page then change page in Max; do the same for Next Page. *forScore* should recognise the incoming MIDI messages (CF00 and CF01). This all presupposes that you have a MIDI over network set up via the Audio MIDI Setup application on the laptop (see Figure 2).

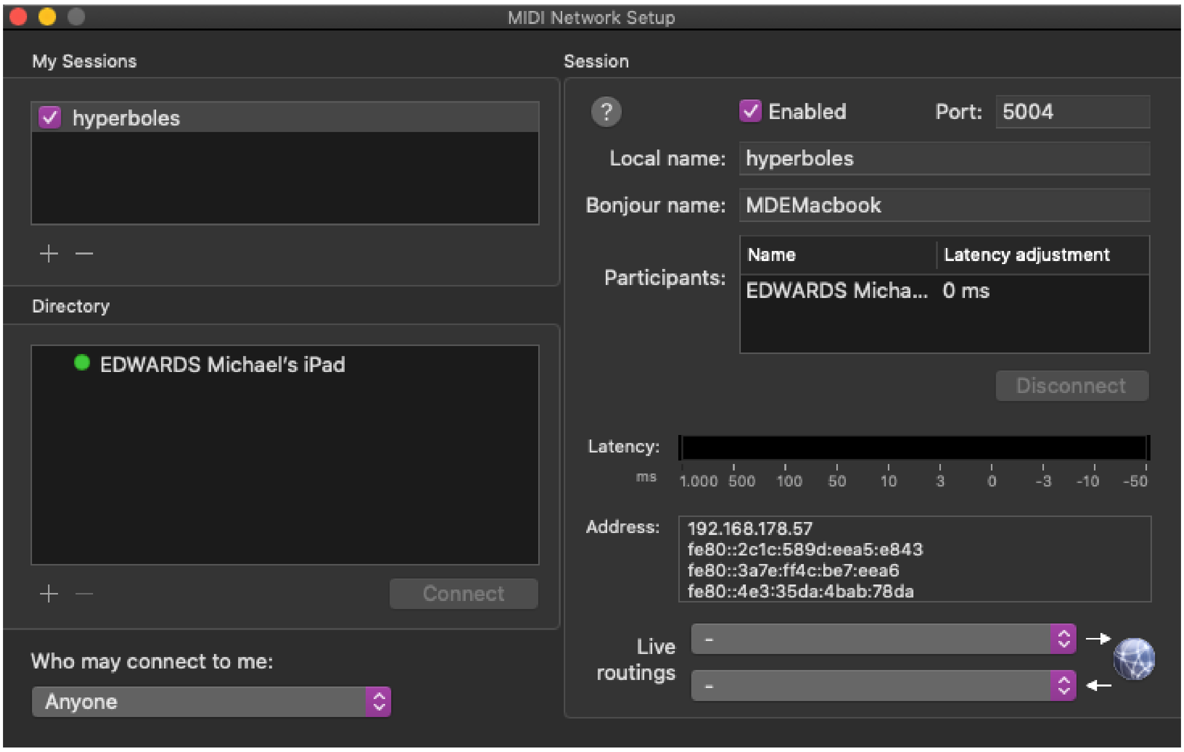


Figure 2: MIDI over network page in OSX's *Audio MIDI Setup*

The **start lisp** button starts a version of Lisp in which *slippery chicken* is preloaded/compiled. This allows you to generate new versions of the piece/score and sound files. It can be used for generating different configurations of the piece instead of the separately installed *slippery chicken* app. This will run in a terminal window. You can safely hide this and return to Max but do make sure to reselect the instrument(s) version (e.g. flute) you're playing from the dropdown menu back in the audio control window, before you proceed to render your own versions.

The **free lisp** button disconnects the MaxMSP<–>Lisp connection and returns you to the Lisp prompt in the Lisp interpreter (either the *slippery chicken* app or the terminal window opened when you press the "start lisp" button). If you wish to resume this connection, simply type (osc-call) in the Lisp interpreter.

In order to start the piece, click the 'start' button on the audio patch or press the 's' key; to stop it, click the 'stop' button (this will also reset levels and other parameters to preset 1). You can pause the performance at any point by pressing the space bar. **countdown before starting** sets the number of beats the software will wait before actually starting. You can see the countdown happen in the audio window, just below the start button.

The audio patch

The blue audio patch allows control over various aspects of the real-time audio processing in the piece. Sequencing of real-time effects and sound files is done via a MIDI file which begins once the 'start' button is pressed and the countdown is over. MaxMSP does not allow much control over MIDI file playback so for rehearsal purposes I would suggest playing the MIDI file (found in the output folder) via software such as Rondo (http://www.fracturedsoftware.com/rondo/) or a DAW like Reaper (http://reaper.fm/). This allows scrolling through the piece to start at any arbitrary bar (they are numbered in Rondo's interface) and if the output of Rondo is set to IAC Bus 1 (via the Sequence->Destination menu), then the MaxMSP patch will pick up the MIDI notes and trigger the sound files and effects presets in all the right places. Whether using Rondo or not, the solo instrument simulation can be turned on and off via the 'simulate solo' toggle; clearly it should be off (cross not visible in the toggle box) for live performances (its state is saved in the presets).

The dials along the top of the audio window reflect MIDI faders/dials on channel 13: controller numbers 0-3 (labelled 1-4 for convenience) control levels for the solo instrument, sndfiles, solo DSP (effects aka FX) and background sound files respectively; numbers 4-6 control the reverb levels for the solo instrument, sound files, and solo DSP: all reverb levels are post-fader e.g. if the solo signal is attenuated by 6db on controller 0 and the solo reverb by 3db on controller 4, then the solo level sent to the reverberator is attenuated overall by 9db.

Controller 15 (labelled 16) controls the master gain. Controller numbers 8 and 9 (labelled 9 and 10) control the gain on high and low shelf filters applied to the scored computer parts, i.e. the sound files under controller 1. In addition, gain levels for high and low sound files can be entered in the two number boxes below the filters; the division point is middle C in the computer parts of the score.

Two further number box controls allow the setting of:

1. The solo plus FX threshold ("solo note off") below which the combined signal must fall before the FX will be changed. Solo FX are changed automatically on-the-fly yet changing whilst the soloist plays a note, or the FX are still ringing, might cause clicks or other disturbances, hence we wait until the signal drops below the threshold before changing. In addition, when we change, there is first of all a ramp down, then a ramp up once the FX are changed. The length of this ramp is not something the user has control over but if it must be changed, edit the argument to the v-fade abstraction at the bottom of hyperboles-rtdsp.maxpat.

2. The ramp down time for the sound files triggered via the score. The ramp starts at the point in the score where the note stops, which means that in the computer performance the sound files hang on over the rests. Also worth mentioning is that the computer parts' pitches in the score will generally be reflected in the sound files: they are related to the soloist's nearest pitches via proportions from the harmonic series. However, shorter notes are generally played back as short percussive effects and these may deviate considerably from the scored pitch.

The two 'mic' buttons will open controls for two microphone inputs for the soloist (on inputs 1 and 2). The controls for these should be familiar and/or self-explanatory. The two signals can be panned and processed separately but are mixed to mono before being sent to the effects.

The toggle labelled "MIDI=1 asco=1" specifies whether the MIDI sequencer or antescofo object is used for playback. For all versions other than the *platforme* this should be set to MIDI, i.e. a cross should be visible on this toggle.

The tempo and bar numbers are indicated along with a metronome, which can be turned off via the toggle.

Solo FX The toggles here are read-only, i.e., they show the currently activated real-time FX being applied to the solo instrument signal and whose combined gain is set by controller 2. The 'open' button below the FX toggles will reveal the FX patch; changes can be made here and the individual effects levels can be viewed and changed but it is not envisaged that changes should be made just prior to or during a performance.

The "noise thresh" number box sets the decibel level below which the incoming instrument audio signal is turned off so as to avoid processing the noise floor.

For advanced users: There are six files in the data directory, each of which store seven presets for each of the six processing modules, e.g., flute-convrev2.json, flute-gran.json, flute-freezer.json, flute-modelaypattr.json, flute-freq-shift.json, and flute-ringmodm.json. Similarly named versions for each instrument should be present in the same directory; if not, simply copy another instrument's files and rename them.

When creating new instrument versions it is essential to step through and audition the presets which will be used during the piece. Usually this will be done by the composer during preliminary rehearsals. Opening e.g. data/hyperboles-flute-rt-dsp.txt will show how many FX combinations will be stepped through (18, 21, or 27, depending on the chosen *global proportions*). You can select and audition these individually in the FX patch by typing a number into the "tests presets" box then playing some suitable material into the microphone. You can adjust any of the numerous parameters for active FX modules and store them in the requisite preset: hold the shift key and click on the lighter grey (= currently active) small round preset button. Press the "save all fx presets" button when you're done and all FX parameters will be saved and used from then on when performing the piece.

Presets The 'store' and 'recall' numbers allow various preset settings to be stored in a file and recalled in a later session. If you store any presets you will be prompted to save the file once the patch is closed. Note that preset 1 is loaded from the file hyperboles-flute.json or similar file reflecting the instrumental configuration for the version you are playing (e.g. hyperboles-violin.json). This is found in the data directory, and is loaded by default at startup. This file should therefore be used for local performance settings. Feel free to overwrite the file with new settings (though you might want to back up the original). Also note that interface dial values will override fader levels (which are controlled by the dials) and should therefore be set properly before storing a preset.

N.B. Once a preset is loaded, the interface dial might be out of line with the MIDI controller/Mira. In order to avoid sudden parameter changes, the software dials will not reflect the MIDI faders/dials levels until the latter reaches the stored value.

Background sound files The three computer parts are combined onto the single 'sndfiles' fader along with the glissandi sound files which start the piece and subsequent sections. (The latter are notated in the score by the words [gliss. sound] above the top computer part; they begin on every section and subsection, i.e. 1, 1.2, 1.3, 1.4, 2, 2.1 etc.)

In addition, on the second sub-subsection in every subsection (i.e. 1.1.2, 1.2.2, 1.3.2 etc.) a background sound is triggered; this is indicated in the score by the words [background sound]. This is a generally quiet and/or distant sound which may be mixed in at more or less low levels in order to provide some textural variation. This is desirable but not essential. When a background sound is playing, it will be indicated in green on the audio interface; when not playing, it will be red.

Algorithm Specification

This is a flexible, musician-tweaked algorithmic piece originally written for flute and computer in 2013/14 but since developed and extended for other solo and group instrumental combinations with or without the computer. As should be clear by now, there is no permanently fixed score as such; rather, a score is generated by the underlying software via configuration of the parameter space presented by a separate software interface. There is no requirement to do this on the part of the musician(s) however as a pre-configured version can be delivered by the composer.

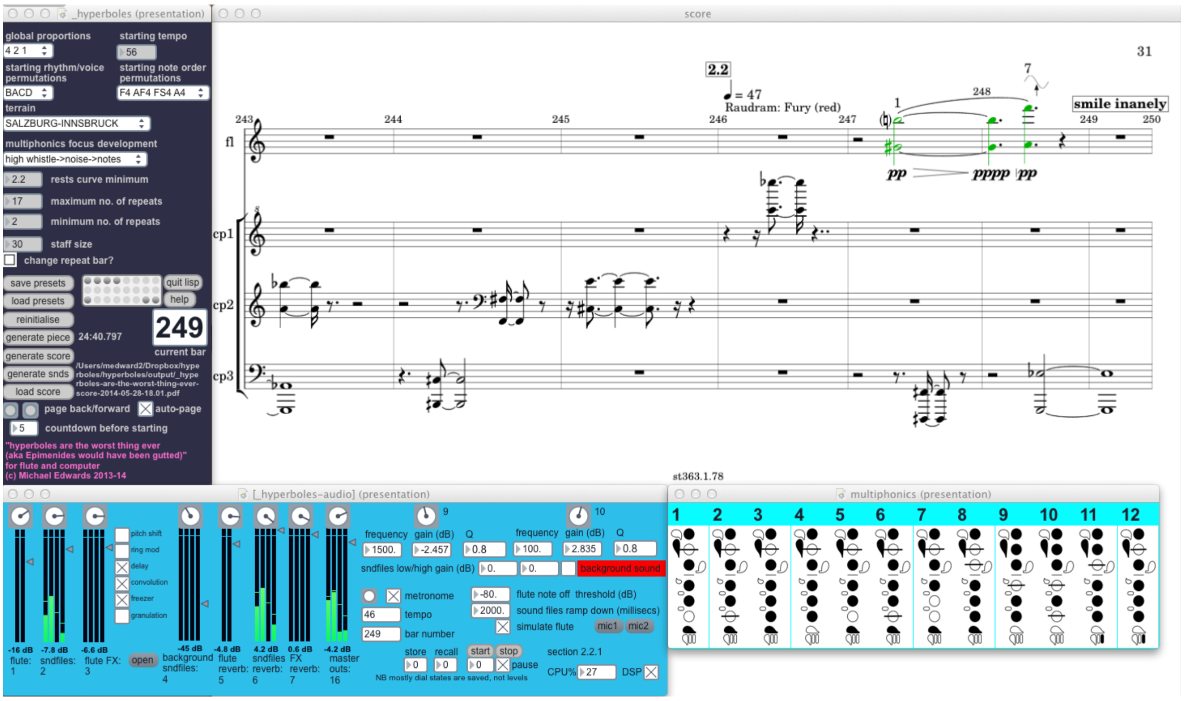


Figure 3: The algorithm interface

The generative ideas then are the crux of the piece, rather than the details of any one score generated by them. Or to put it another way, the fixed score is not the essence of the music but a tool towards reaching one possible realisation of the ideas as expressed in software—I'm just swapping one code for another, really.

The objective of this approach is the algorithmic[1] specification of global structure, parameters, tendencies, data, and data processing techniques which are then used to lead to different versions of the piece. These, though all different, share identifiable common characteristics.

Structure

The formal structure of the piece is generated by an iterative process expanding outwards from three simple proportions, either 4:3:2, 3:2:1, or 4:2:1 (others could be possible but these produce suitable durations of c. 52, 15, or 25 minutes respectively, accepting the other default parameter values such as tempo). The depth of iteration in this process is theoretically unlimited but we use two generations for this piece so that durations are manageable. Using the 4:3:2 proportions we can see how the section lengths unfold accordingly:

```
((((1) 324)
  (((1 1) 81)
   (((1 1 1) 36) ((1 1 1 1) 9) ((1 1 1 2) 9) ((1 1 1 3) 9) ((1 1 1 4) 9))
   (((1 1 2) 27) ((1 1 2 1) 9) ((1 1 2 2) 9) ((1 1 2 3) 9))
   (((1 1 3) 18) ((1 1 3 1) 9) ((1 1 3 2) 9)))
  (((1 2) 81)
   (((1 2 1) 36) ((1 2 1 1) 9) ((1 2 1 2) 9) ((1 2 1 3) 9) ((1 2 1 4) 9))
```

[1]The algorithms are deterministic, i.e. there is no randomness here.

```
    (((1 2 2) 27) ((1 2 2 1) 9) ((1 2 2 2) 9) ((1 2 2 3) 9))
    (((1 2 3) 18) ((1 2 3 1) 9) ((1 2 3 2) 9)))
   (((1 3) 81)
    (((1 3 1) 36) ((1 3 1 1) 9) ((1 3 1 2) 9) ((1 3 1 3) 9) ((1 3 1 4) 9))
    (((1 3 2) 27) ((1 3 2 1) 9) ((1 3 2 2) 9) ((1 3 2 3) 9))
    (((1 3 3) 18) ((1 3 3 1) 9) ((1 3 3 2) 9)))
   (((1 4) 81)
    (((1 4 1) 36) ((1 4 1 1) 9) ((1 4 1 2) 9) ((1 4 1 3) 9) ((1 4 1 4) 9))
    (((1 4 2) 27) ((1 4 2 1) 9) ((1 4 2 2) 9) ((1 4 2 3) 9))
    (((1 4 3) 18) ((1 4 3 1) 9) ((1 4 3 2) 9))))
  (((2) 243)
   (((2 1) 81)
    (((2 1 1) 36) ((2 1 1 1) 9) ((2 1 1 2) 9) ((2 1 1 3) 9) ((2 1 1 4) 9))
    (((2 1 2) 27) ((2 1 2 1) 9) ((2 1 2 2) 9) ((2 1 2 3) 9))
    (((2 1 3) 18) ((2 1 3 1) 9) ((2 1 3 2) 9)))
   (((2 2) 81)
    (((2 2 1) 36) ((2 2 1 1) 9) ((2 2 1 2) 9) ((2 2 1 3) 9) ((2 2 1 4) 9))
    (((2 2 2) 27) ((2 2 2 1) 9) ((2 2 2 2) 9) ((2 2 2 3) 9))
    (((2 2 3) 18) ((2 2 3 1) 9) ((2 2 3 2) 9)))
   (((2 3) 81)
    (((2 3 1) 36) ((2 3 1 1) 9) ((2 3 1 2) 9) ((2 3 1 3) 9) ((2 3 1 4) 9))
    (((2 3 2) 27) ((2 3 2 1) 9) ((2 3 2 2) 9) ((2 3 2 3) 9))
    (((2 3 3) 18) ((2 3 3 1) 9) ((2 3 3 2) 9))))
  (((3) 162)
   (((3 1) 81)
    (((3 1 1) 36) ((3 1 1 1) 9) ((3 1 1 2) 9) ((3 1 1 3) 9) ((3 1 1 4) 9))
    (((3 1 2) 27) ((3 1 2 1) 9) ((3 1 2 2) 9) ((3 1 2 3) 9))
    (((3 1 3) 18) ((3 1 3 1) 9) ((3 1 3 2) 9)))
   (((3 2) 81)
    (((3 2 1) 36) ((3 2 1 1) 9) ((3 2 1 2) 9) ((3 2 1 3) 9) ((3 2 1 4) 9))
    (((3 2 2) 27) ((3 2 2 1) 9) ((3 2 2 2) 9) ((3 2 2 3) 9))
    (((3 2 3) 18) ((3 2 3 1) 9) ((3 2 3 2) 9)))))
```

The sum of the proportions defines the basic length of the rhythmic sequences (or phrases: see below) used in the piece. So 4:3:2 produces 9-bar sequences. These sequences are then organised in groups of four, three, and two 9-bar sequences, as we see above, with this combination forming a single higher-level grouping of a subsection which is then repeated, e.g. the subsections labelled (1 1) (section 1, subsection 1), (1 2) . . . (3 2) in the above. Each subsection is of exactly the same 81-bar duration (9x4 + 9x3 + 9x2 bars). The number of subsections in each section is then determined by the same basic proportions: 4:3:2. A similar process is undertaken with different proportions, e.g. 3:2:1 would involve 6-bar sequences organised into 36-bar subsections, hence the selected proportions determine the overall duration quite directly.

Rhythm Sequences and Terrain Data

The rhythm sequences used are pre-defined as three single-sequence groups of four-part counterpoint (generally one for the solo instrument and three for the computer). These are mapped onto the instruments over the duration of the piece via a simple permutation routine, using only a single group of four-part counterpoint per section.

Each of the contrapuntal voices is assigned a letter: A, B, C, and D. These are the letters in the **starting rhythm/voice permutations** menu in the algorithm interface: e.g. if you choose BACD, then the solo will start with contrapuntal voice B, the next voice will play A, the voice below that C, and the lowest voice D. However, each of the instruments will play each contrapuntal voice at some point during the permutations, i.e., B is only the starting voice for the solo in the example just given: it will also play A, C, and D. The exact rhythms of the four voices through each of the three sections can be seen below.

However, the choice of whether a computer voice will play or not depends on a curve provided by the elevation data of a road (or train/walking) journey as input from Google Earth. Currently seven journeys are available: Salzburg to Mittersill and Salzburg to Innsbruck (Austria); Basel (Switzerland) to Graz (Austria) on the train; Nagano to Yamanashi (road) and Akita to Morioka by foot (Japan); Banff to Lake

Figure 4: Four-voice rhythm sequences

Louise (Alberta, Canada); and Dingle to Tralee (SW Ireland).

This terrain data is stretched over the duration of the piece and used for a variety of further processes in the piece:

1. To add rests to the four parts: Though the same rhythm sequences are used and re-used in several permutations throughout, the notes in these are processed by a method which turns notes to rests according to an activity-level algorithm driven by the current elevation on the journey.
2. Certain bars in the rhythm sequences are 'stuck on' (i.e. repeated) at the end of every sub-subsection (see structure extract below). The number of repeats at a given point in the piece is determined by the terrain data so that the higher the elevation the more repeats there are (we could think of this as pausing to admire the view).

```
(((1) 324)
 (((1 1) 81)
  (((1 1 1) 36) ((1 1 1 1) 9) ((1 1 1 2) 9)  ((1 1 1 3) 9) ((1 1 1 4) 9))
                                                       repeats in here ^
```

3. From section two, the first rhythm sequence allows improvisational intervention (see above). Whether this extends into the second or third written sequence depends on the terrain data curve. A higher elevation implies more improvisation.
4. Many of the notes in the instrumental parts will include a symbol indicating a manner of playing. For example, the flute version includes different multiphonic spectral foci (see above). The focus—or another method of playing in a different version of the piece—changes more quickly when at a higher elevation in the terrain data.
5. Though dynamics are quiet throughout, higher elevations will also result in slightly higher overall dynamics.

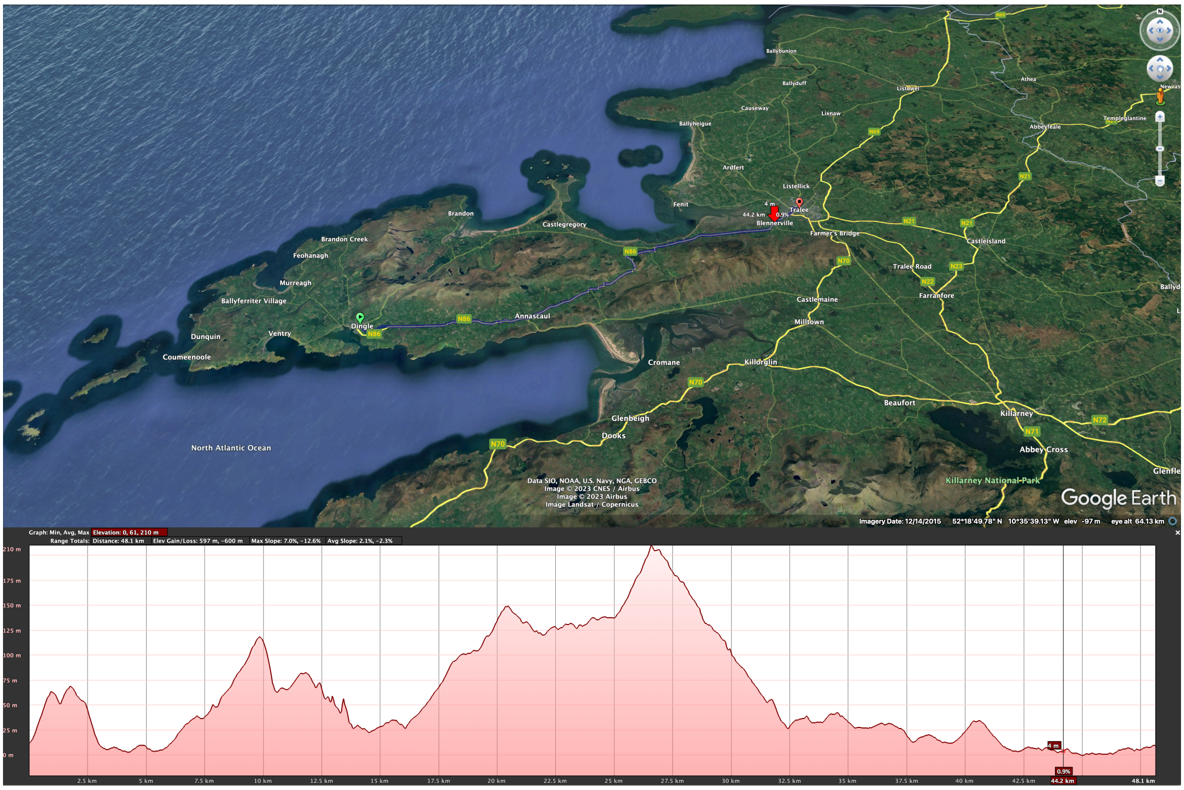


Figure 5: Dingle to Tralee (foot)

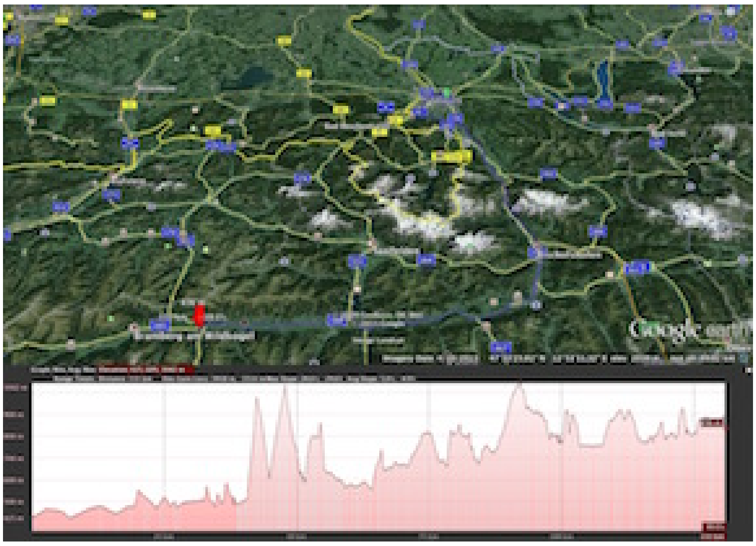


Figure 6: Salzburg to Mittersill

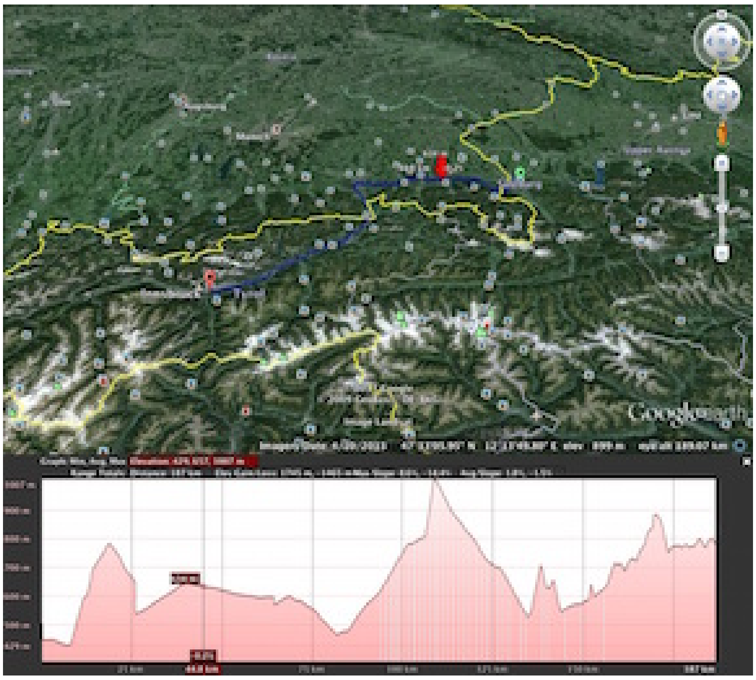


Figure 7: Salzburg to Innsbruck

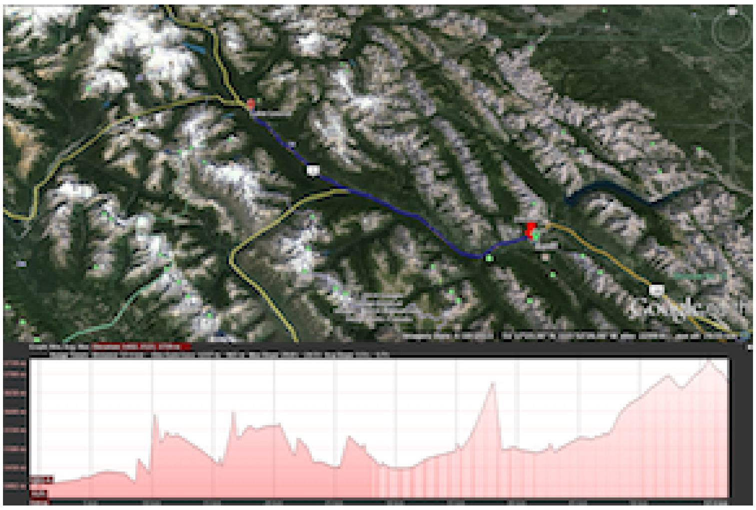


Figure 8: Banff to Lake Louise

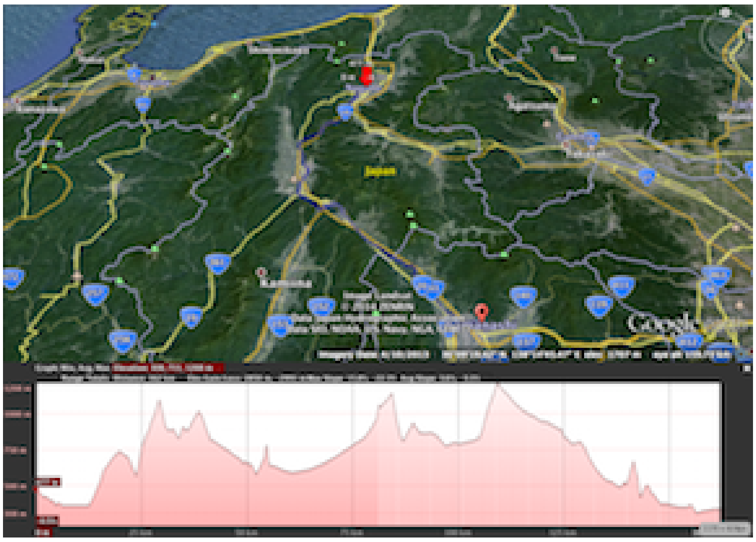


Figure 9: Nagano to Yamanashi

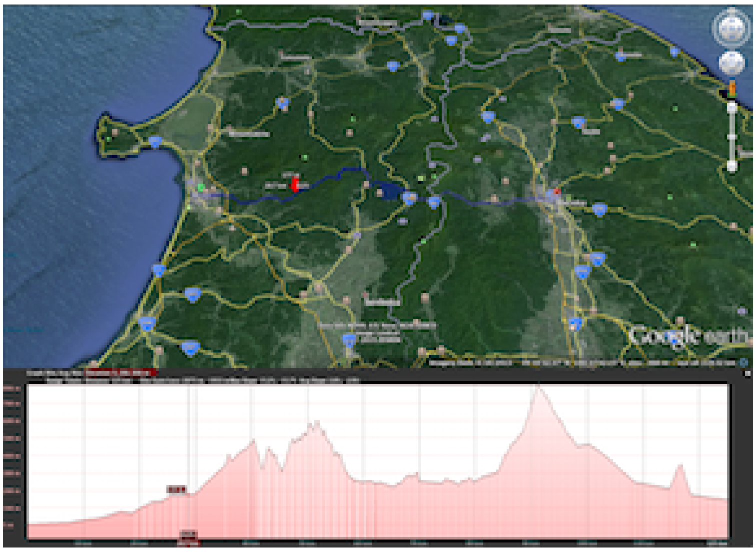


Figure 10: Akita to Morioka

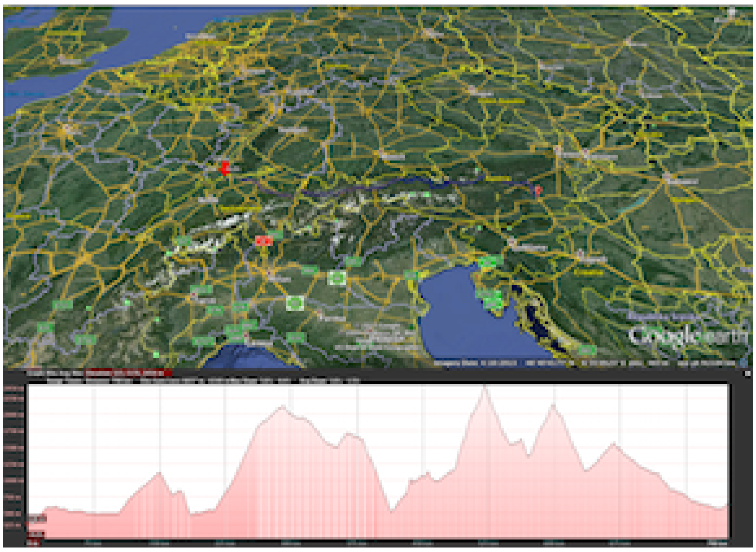


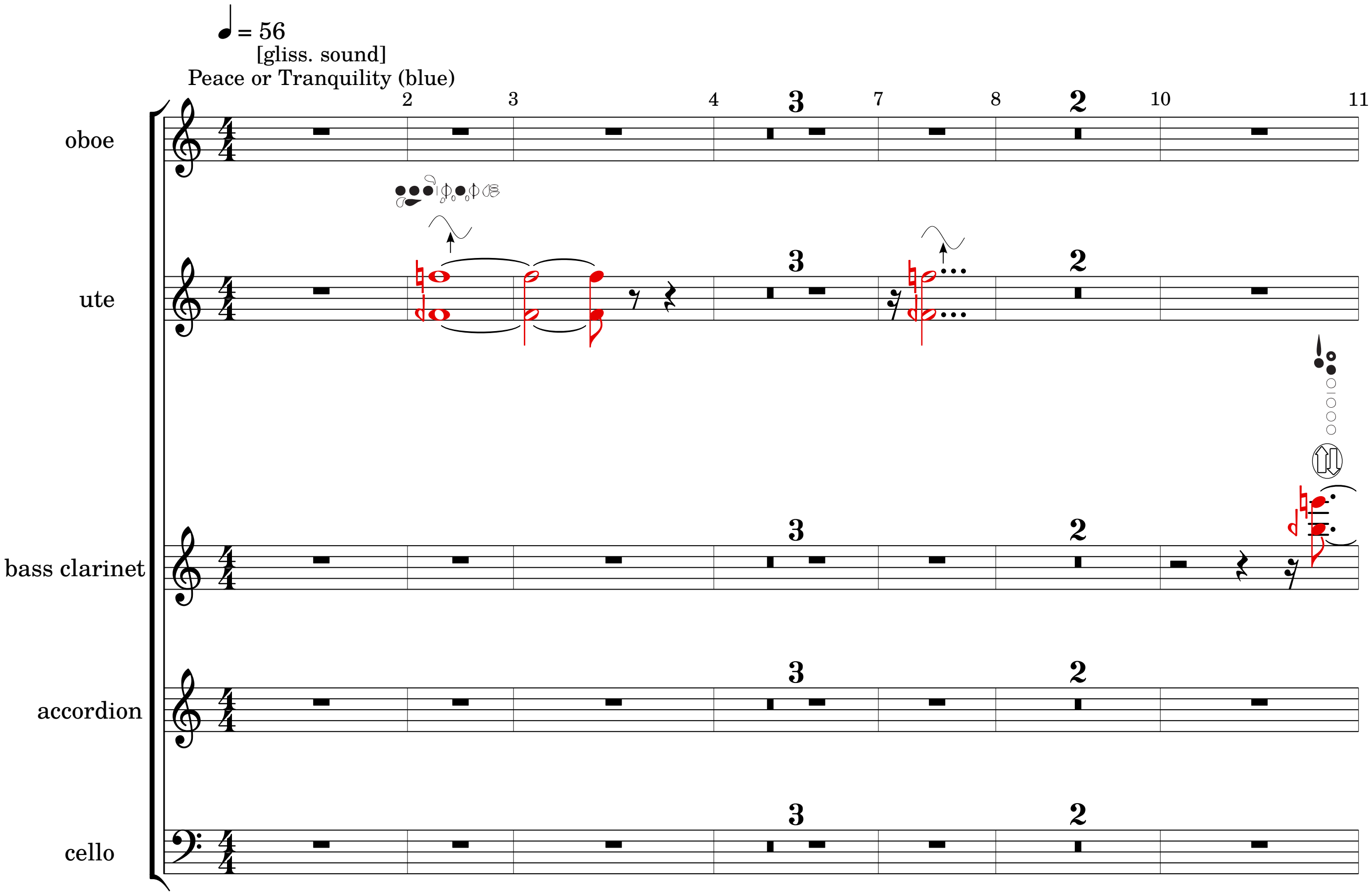
Figure 11: Basel to Graz (train)

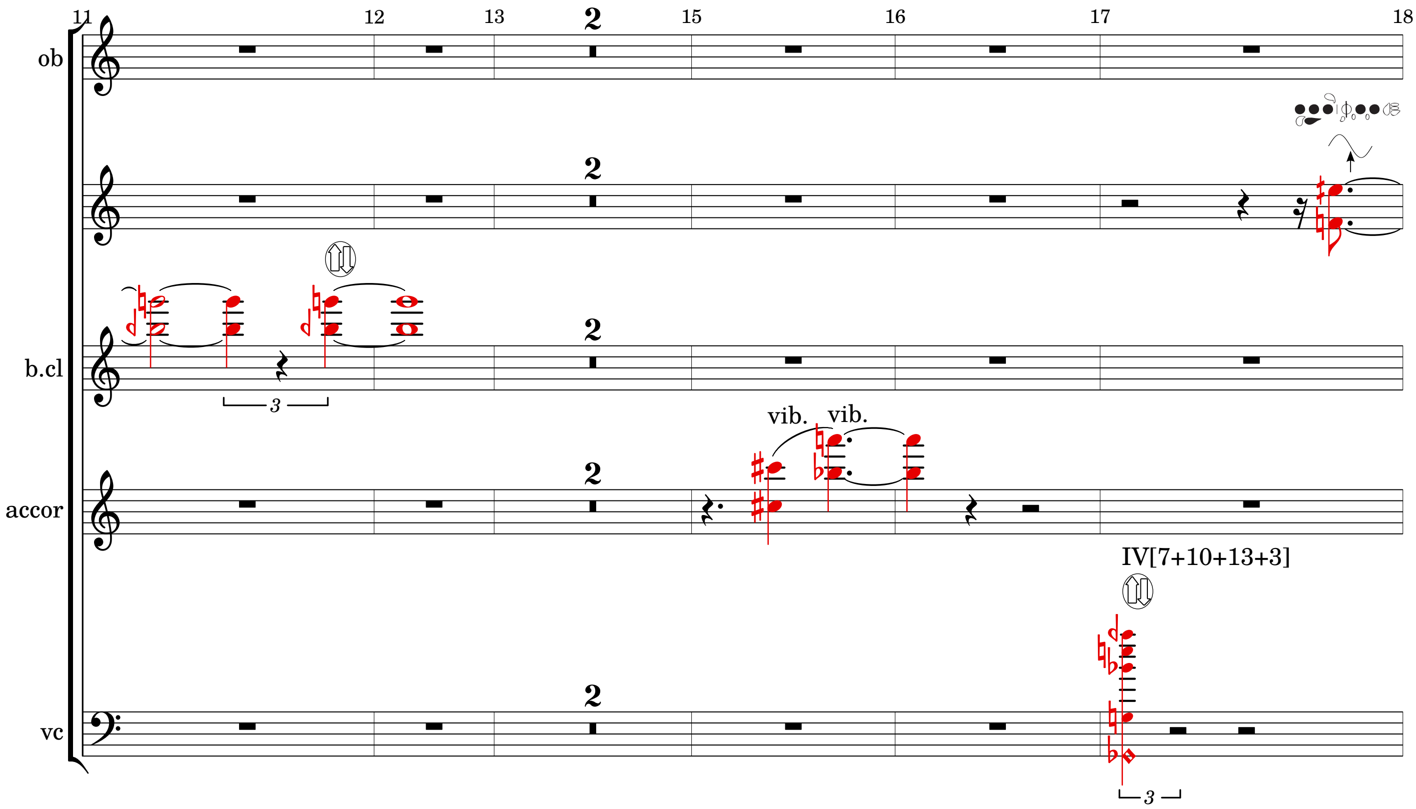
Commissioned by Ensemble S201

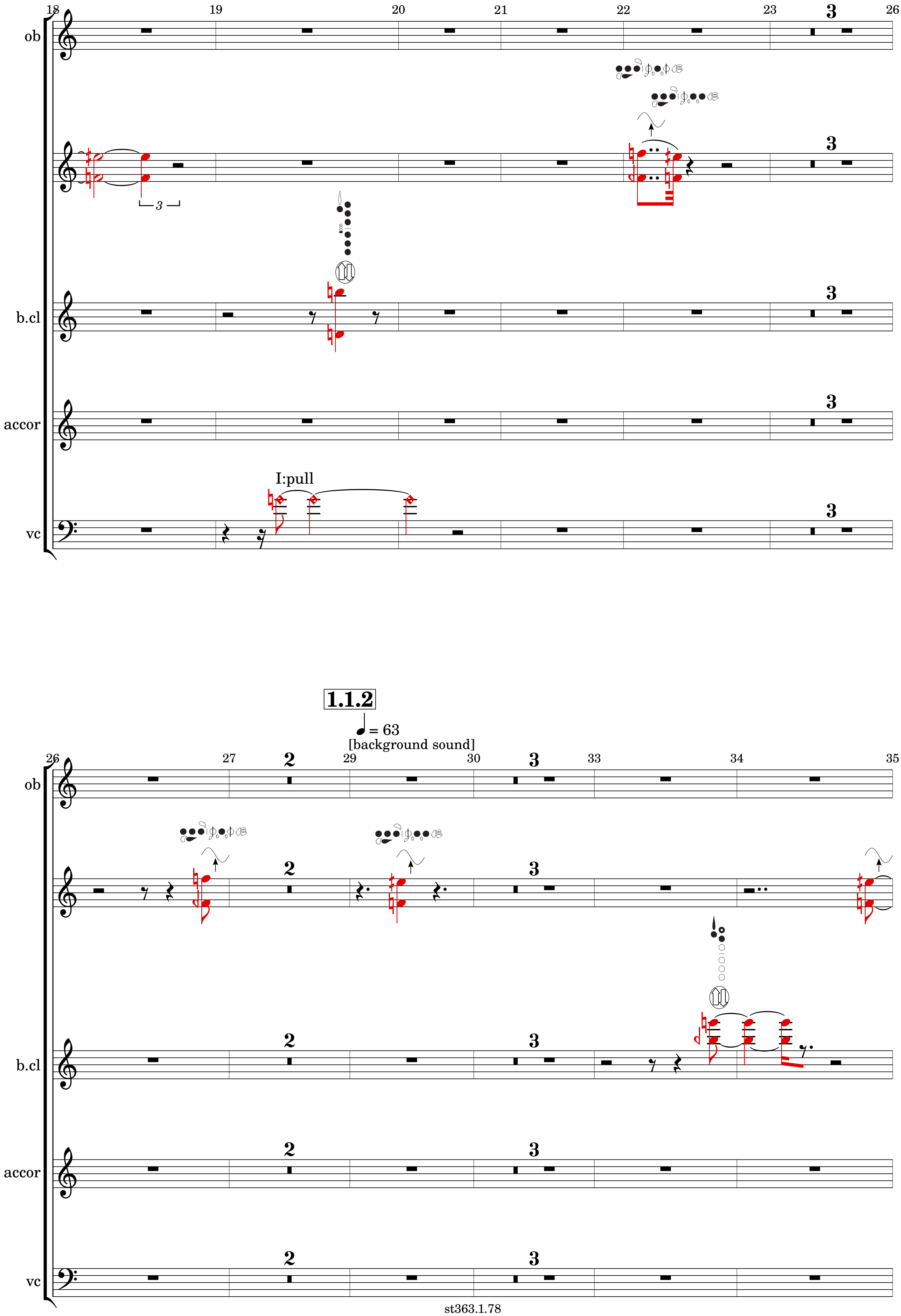
hyperboles 6

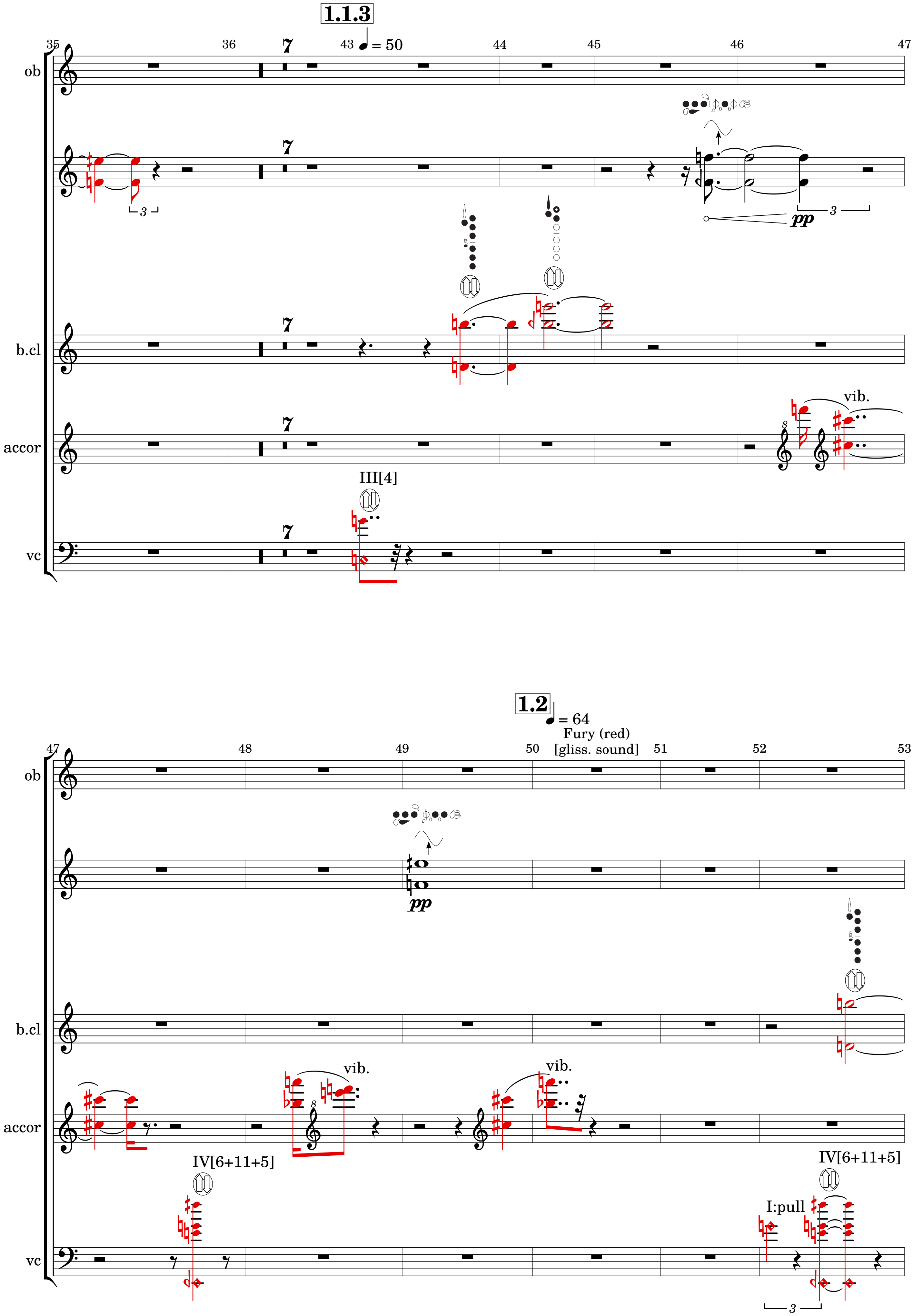
("to prefer the destruction of the whole world to the scratching of my nger")

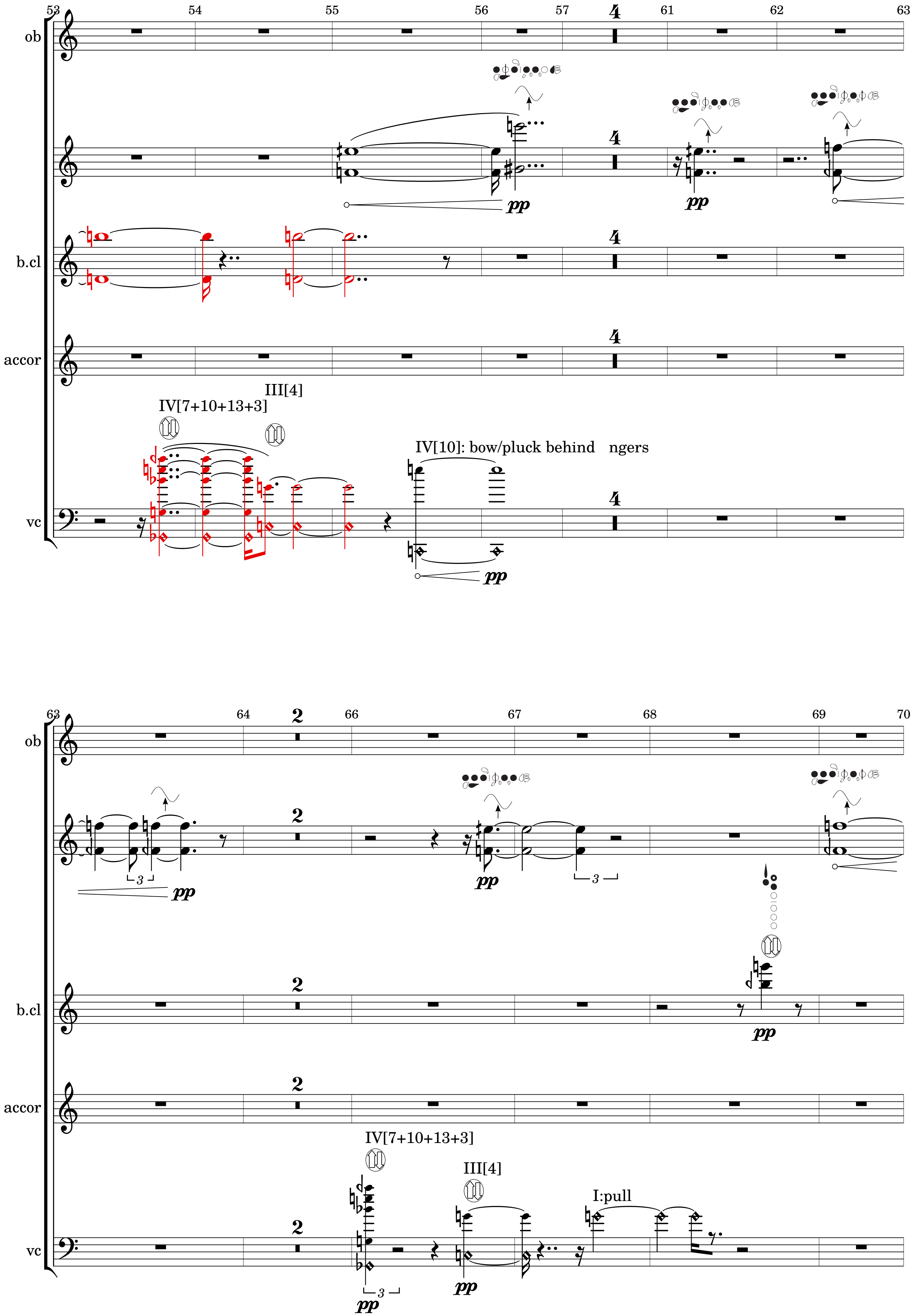
michael edwards 2020



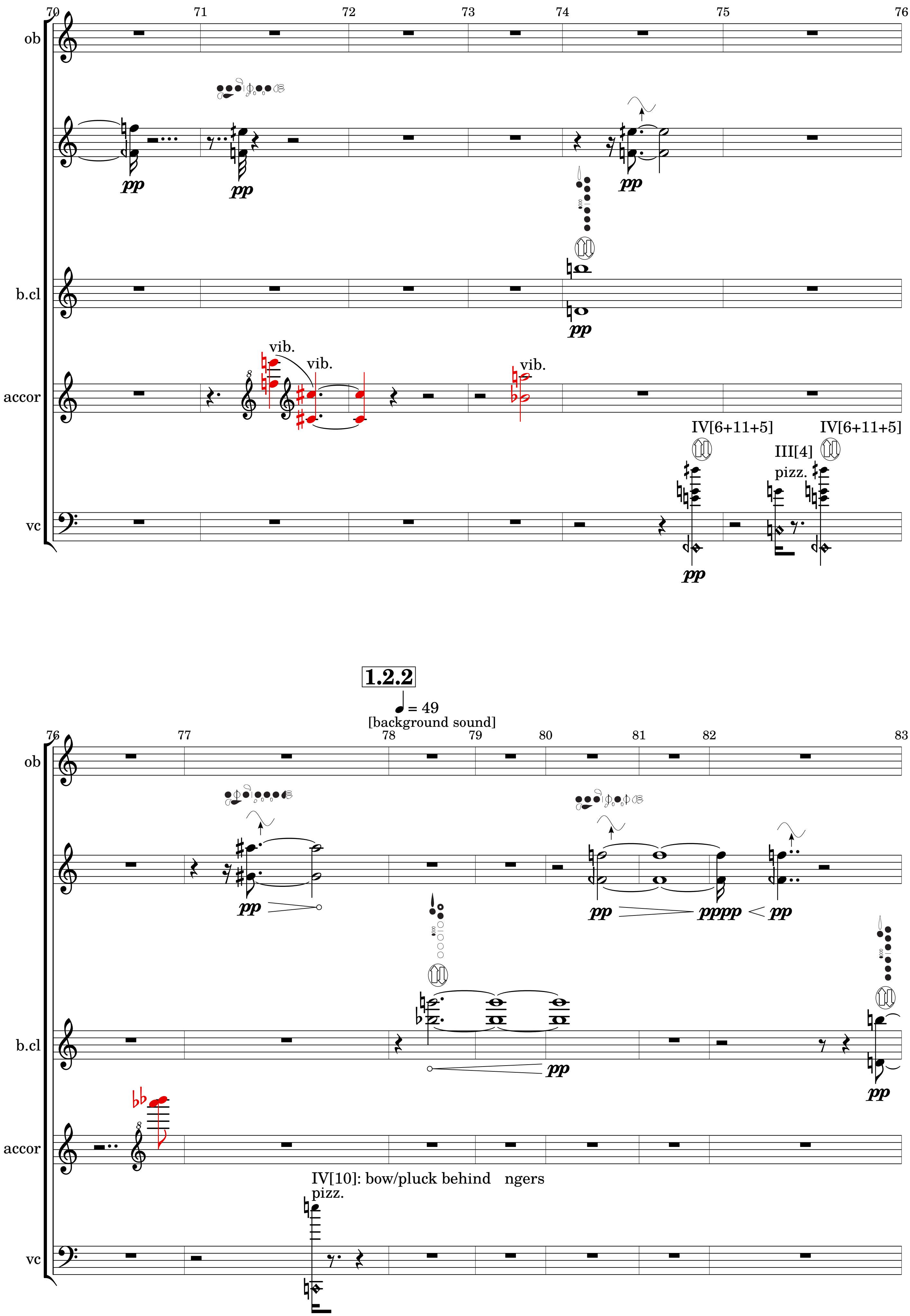


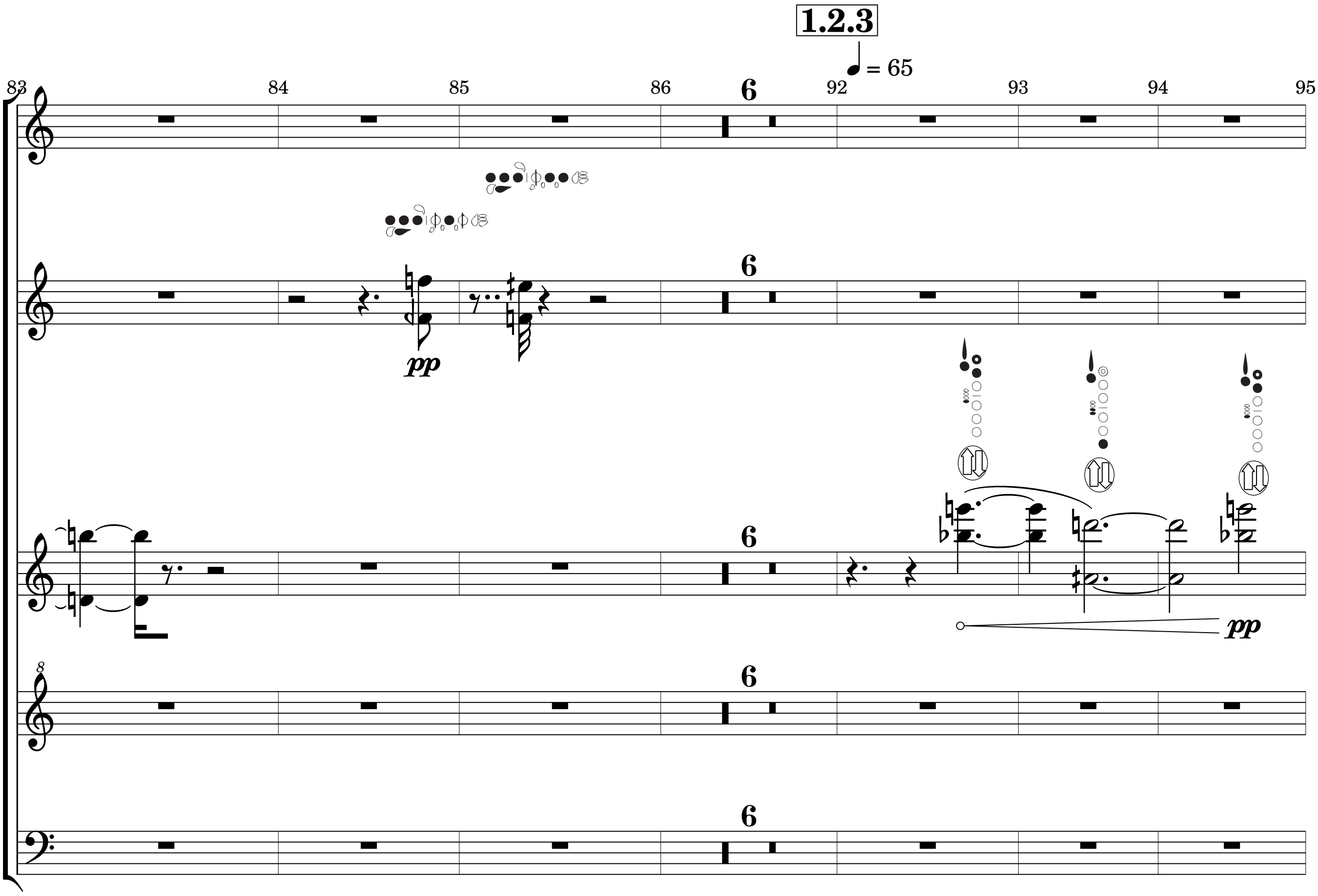

ob
b.cl
accor
vc
I:pull
1.1.2
= 63
[background sound]

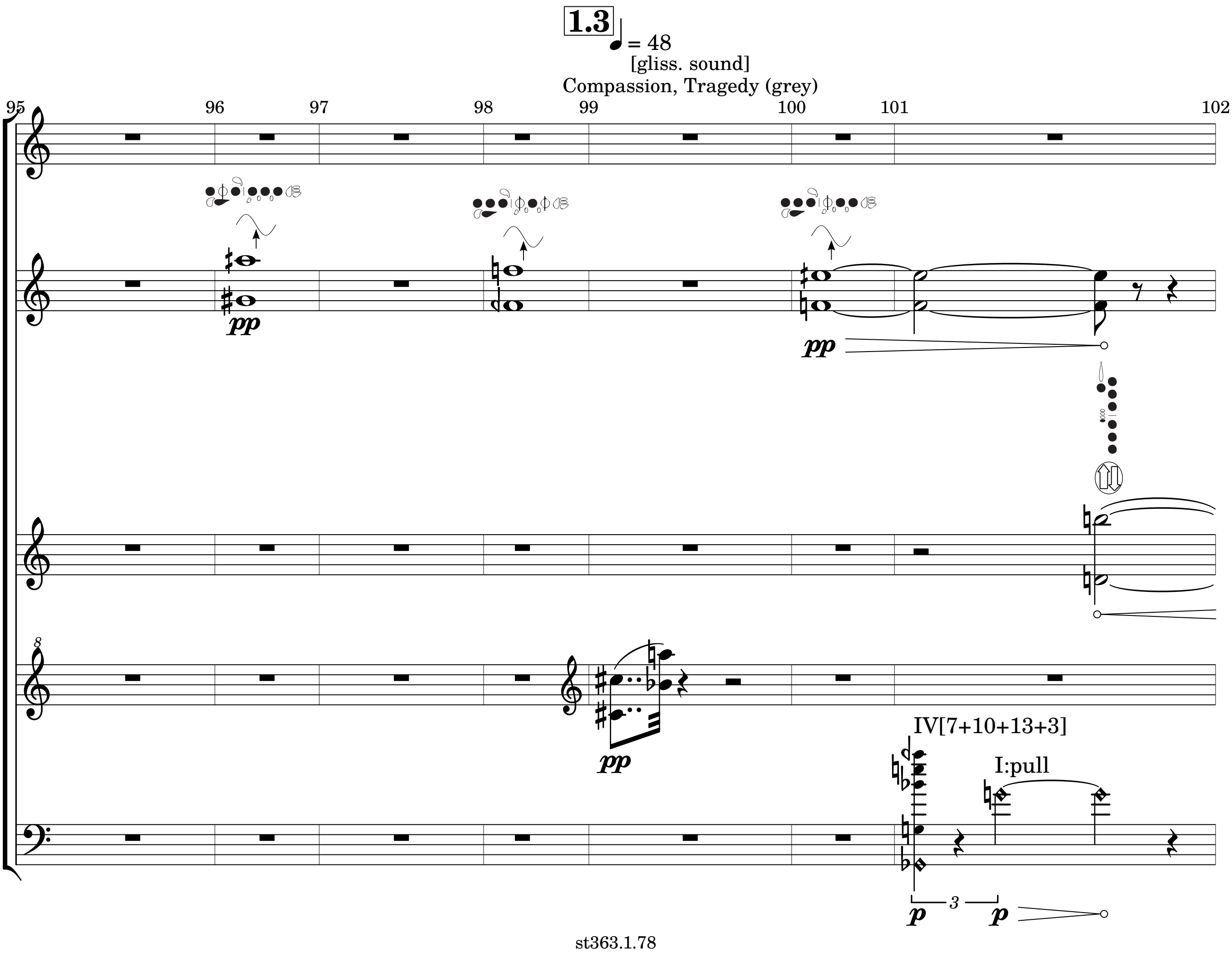

1.1.3
= 50
ob
b.cl
accor
vc
pp
vib.
III[4]
1.2
= 64
Fury (red)
[gliss. sound]
IV[6+11+5]
I:pull

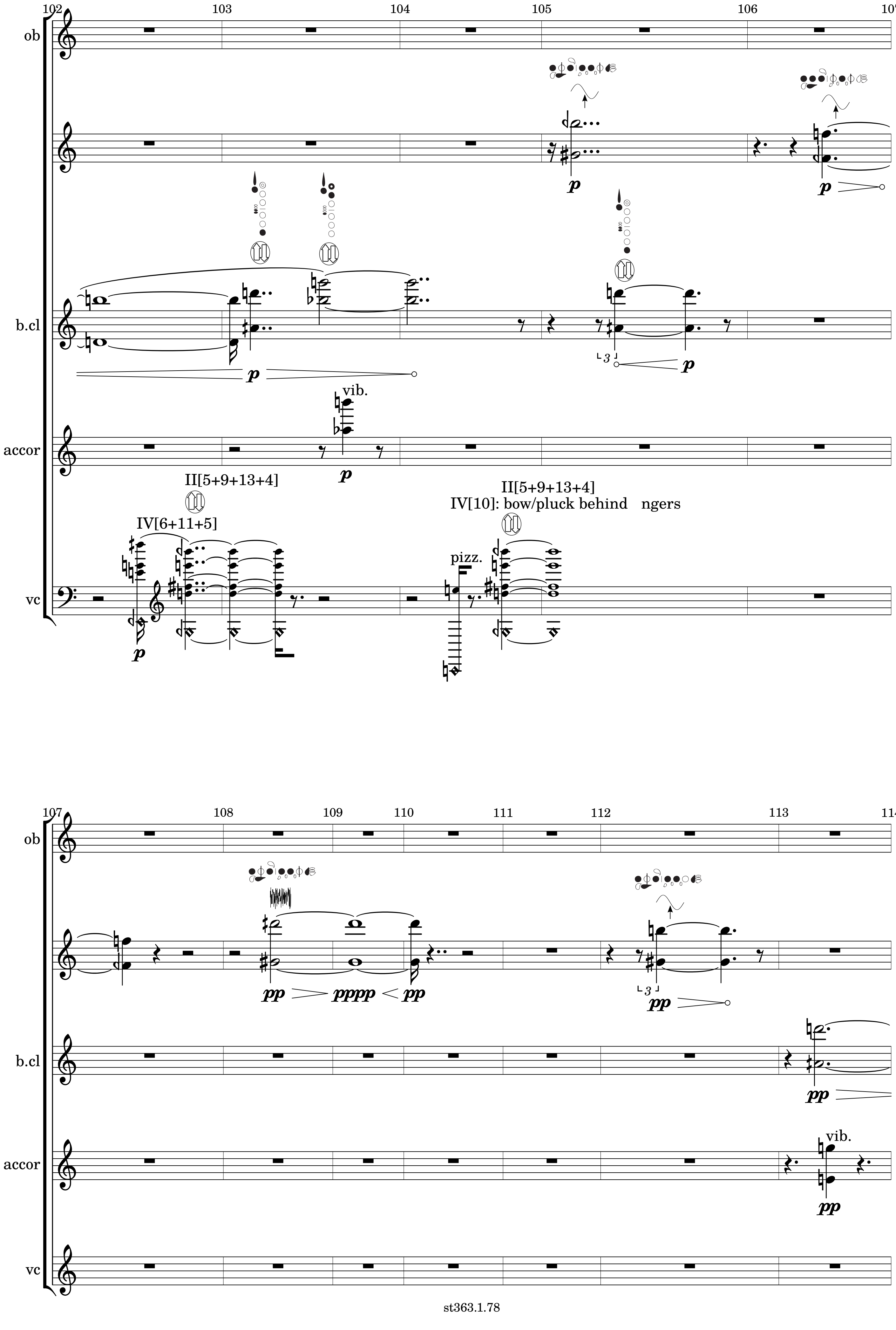


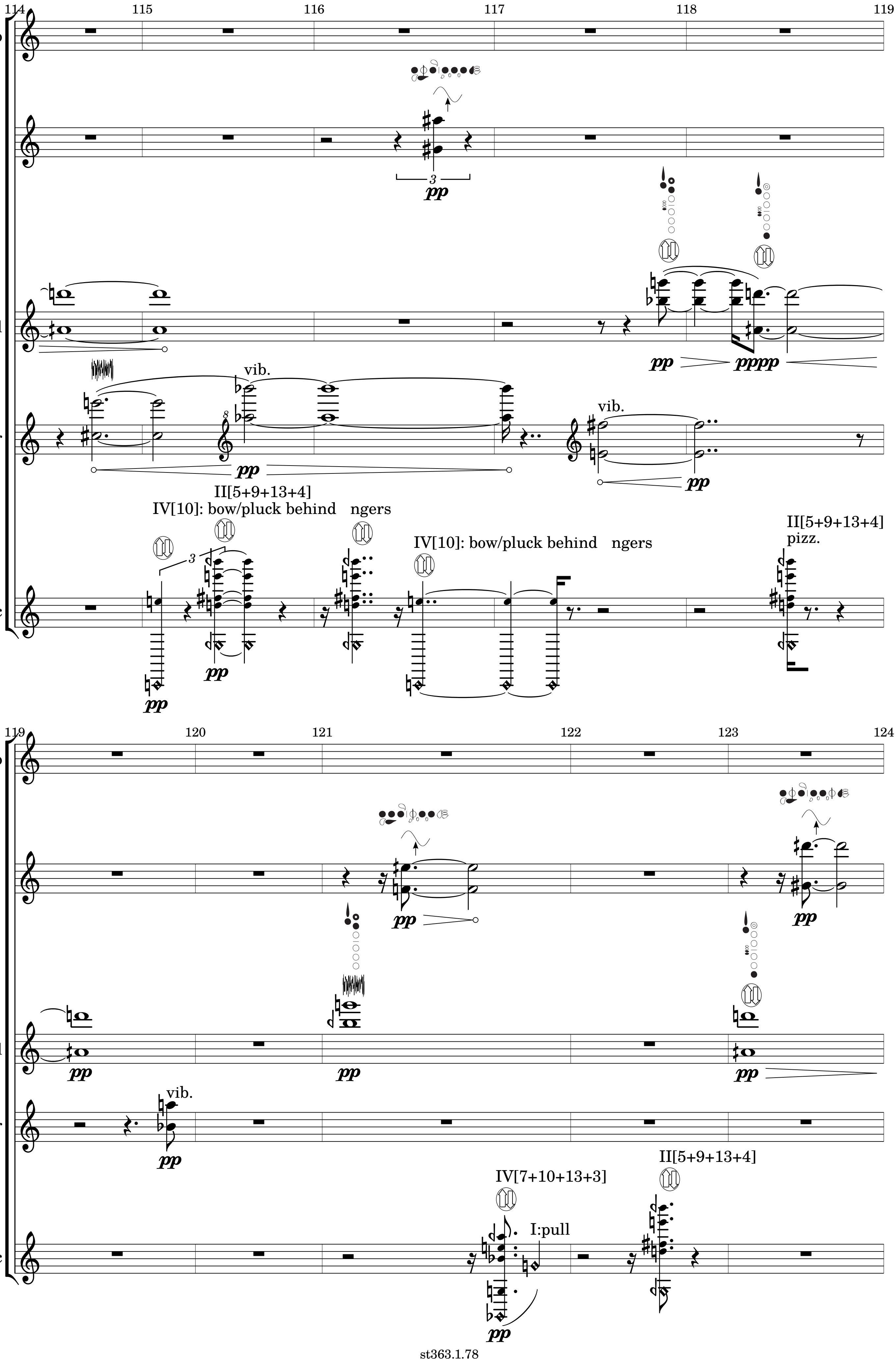
53
54
55
56
57
4
61
62
63
ob
b.cl
accor
vc
IV[7+10+13+3]
III[4]
IV[10]: bow/pluck behind ngers
pp
63
64
2
66
67
68
69
70
IV[7+10+13+3]
III[4]
I:pull
3
pp

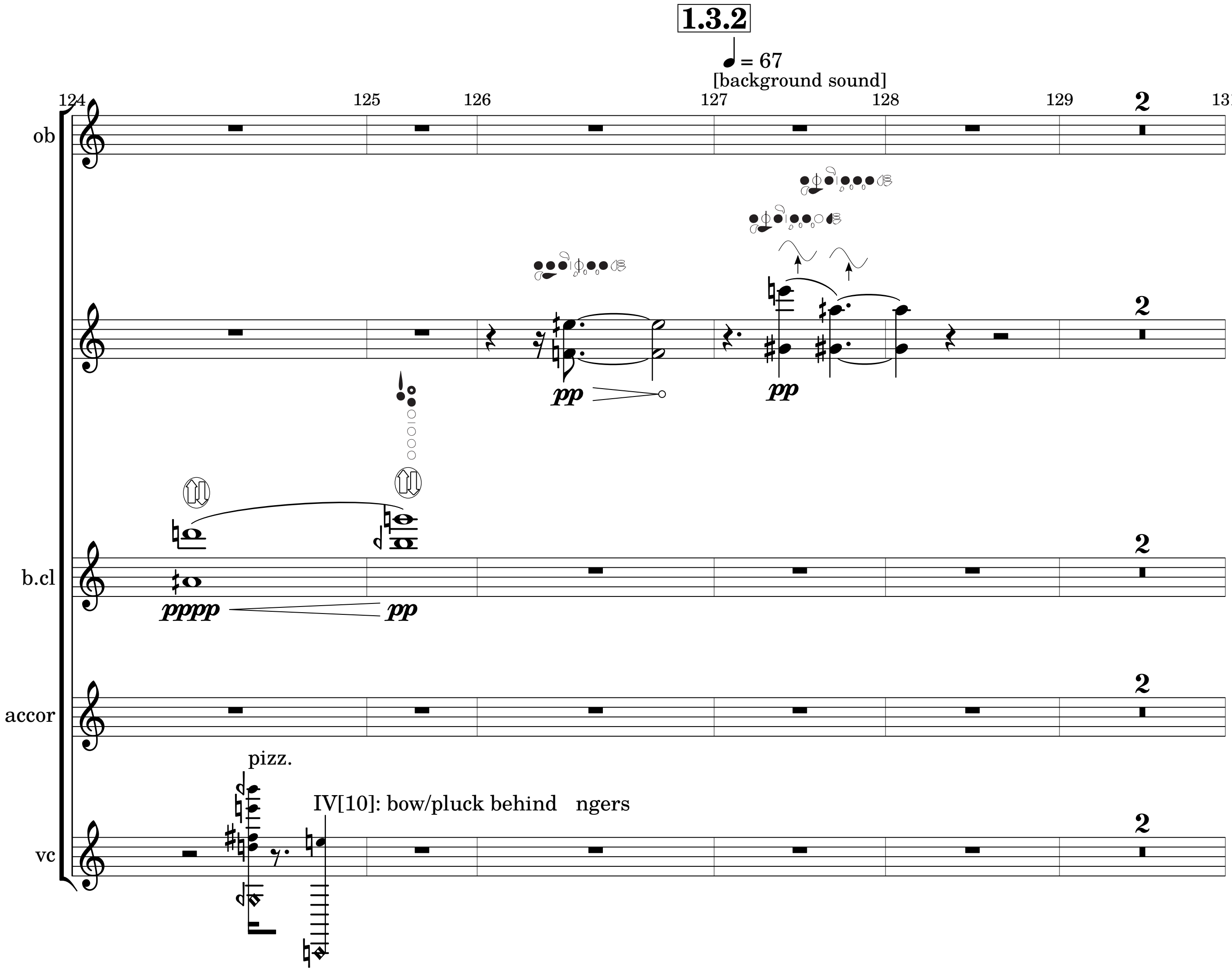

ob
b.cl
accor
vc
vib.
vib.
vib.
pp
IV[6+11+5]
III[4]
pizz.
1.2.2
= 49
[background sound]
pppp
IV[10]: bow/pluck behind ngers
pizz.

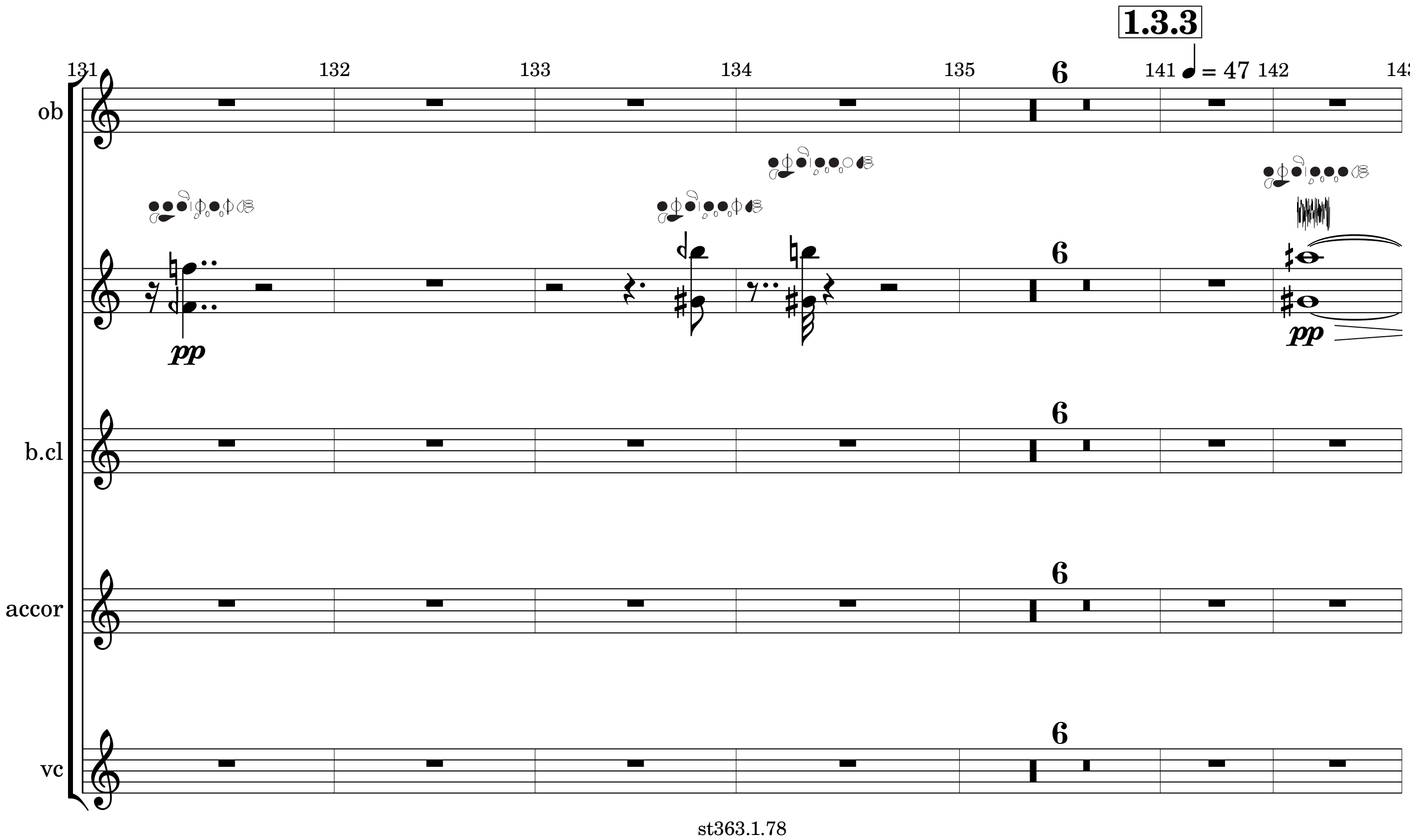

1.2.3
= 65
83
84
85
86
6
92
93
94
95
ob
pp
b.cl
pp
accor
vc

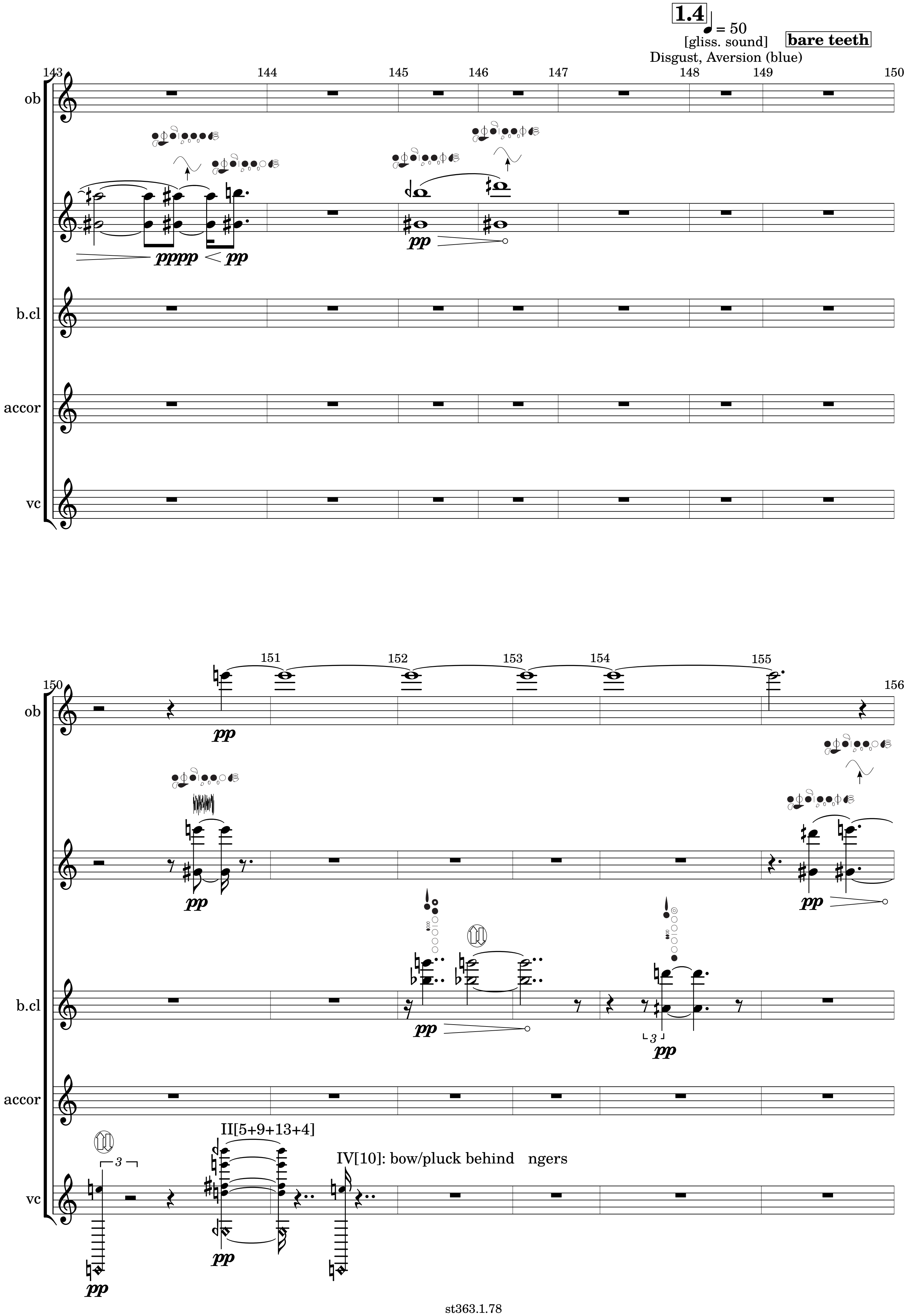

1.3
= 48
[gliss. sound]
Compassion, Tragedy (grey)
95
96
97
98
99
100
101
102
ob
pp
pp
b.cl
accor
pp
IV[7+10+13+3]
I:pull
vc
p
3
p

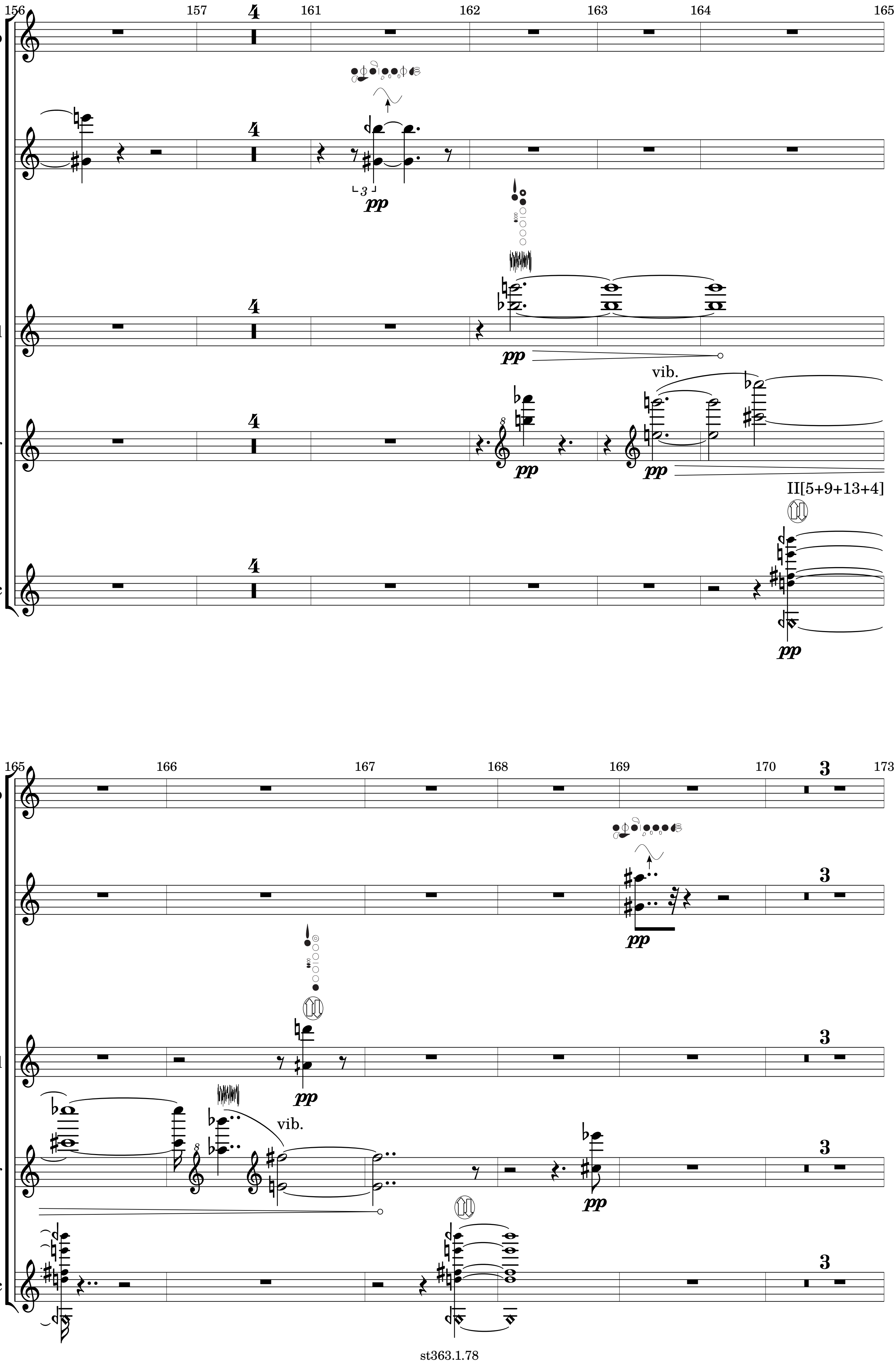

ob
b.cl
accor
vc
vib.
II[5+9+13+4]
IV[6+11+5]
pizz.
II[5+9+13+4]
IV[10]: bow/pluck behind ngers
3
p
pp
pppp

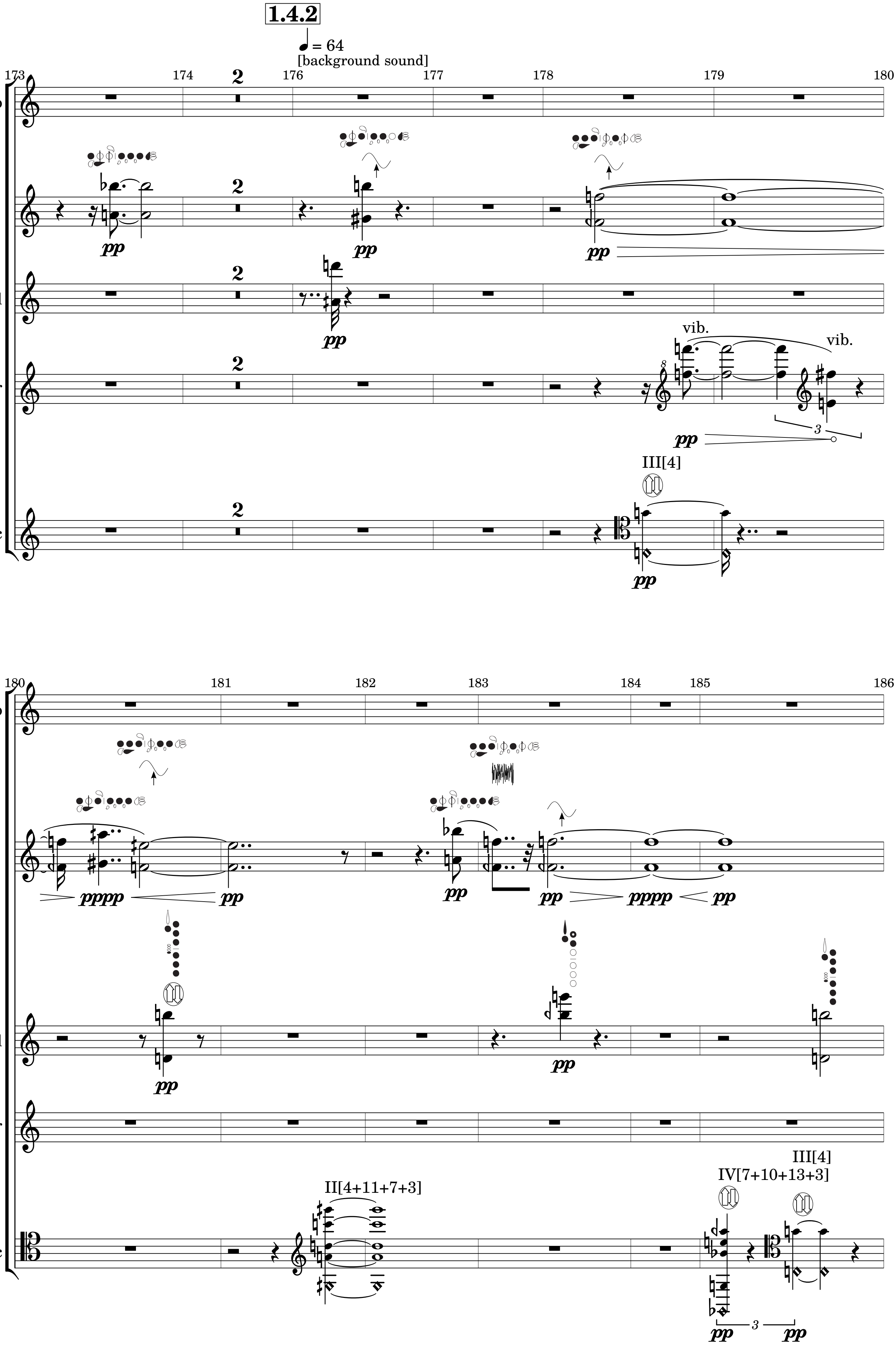

114
115
116
117
118
119
ob
b.cl
accor
vc
3
pp
vib.
8
pp
vib.
pp
pp
pppp
II[5+9+13+4]
IV[10]: bow/pluck behind ngers
IV[10]: bow/pluck behind ngers
II[5+9+13+4]
pizz.
pp
pp
120
121
122
123
124
pp
pp
pp
pp
pp
vib.
pp
IV[7+10+13+3]
I:pull
II[5+9+13+4]
pp

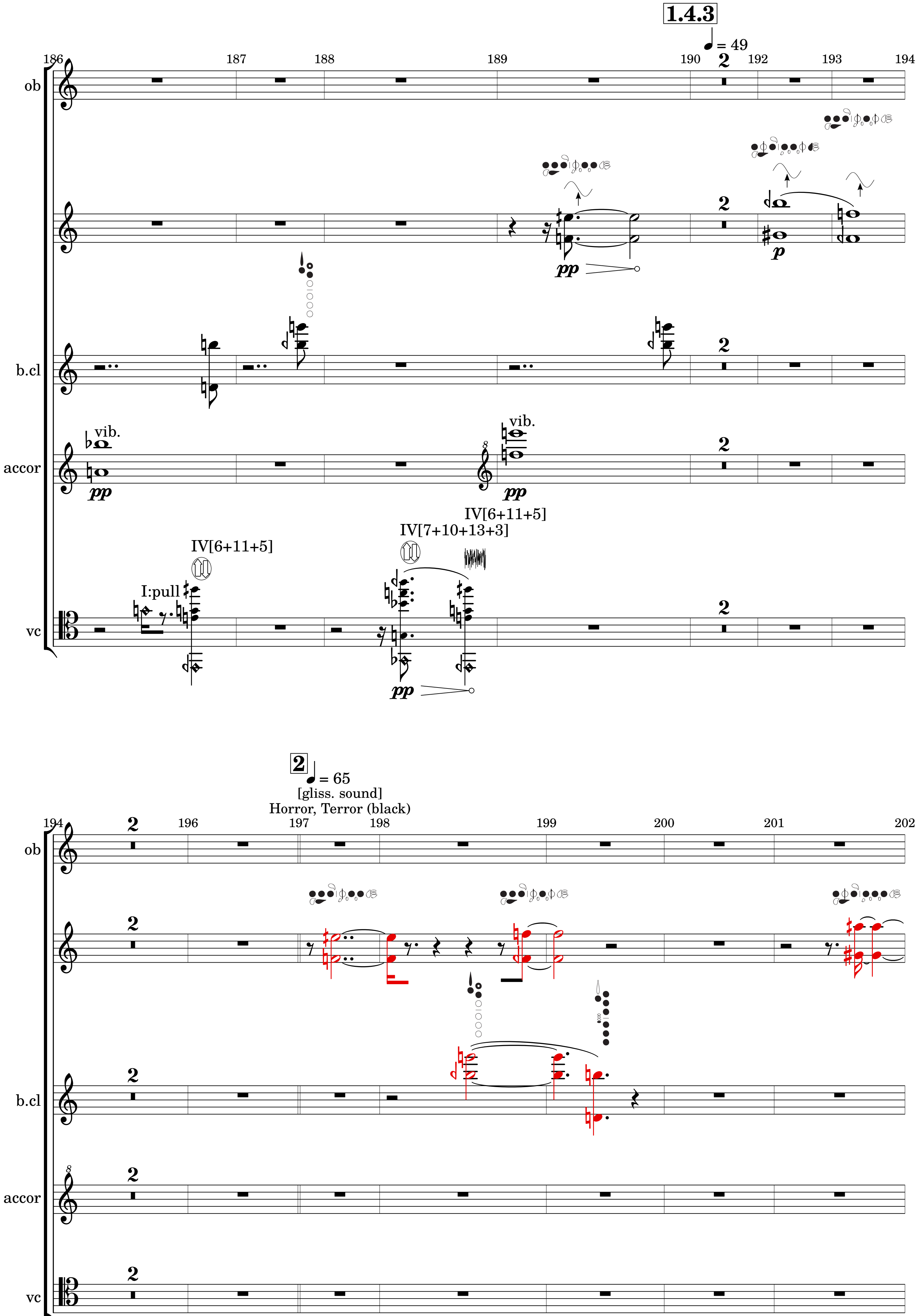

1.3.2
= 67
[background sound]
124
125
126
127
128
129
2
131
ob
b.cl
accor
vc
pppp
pp
pp
pp
pizz.
IV[10]: bow/pluck behind ngers

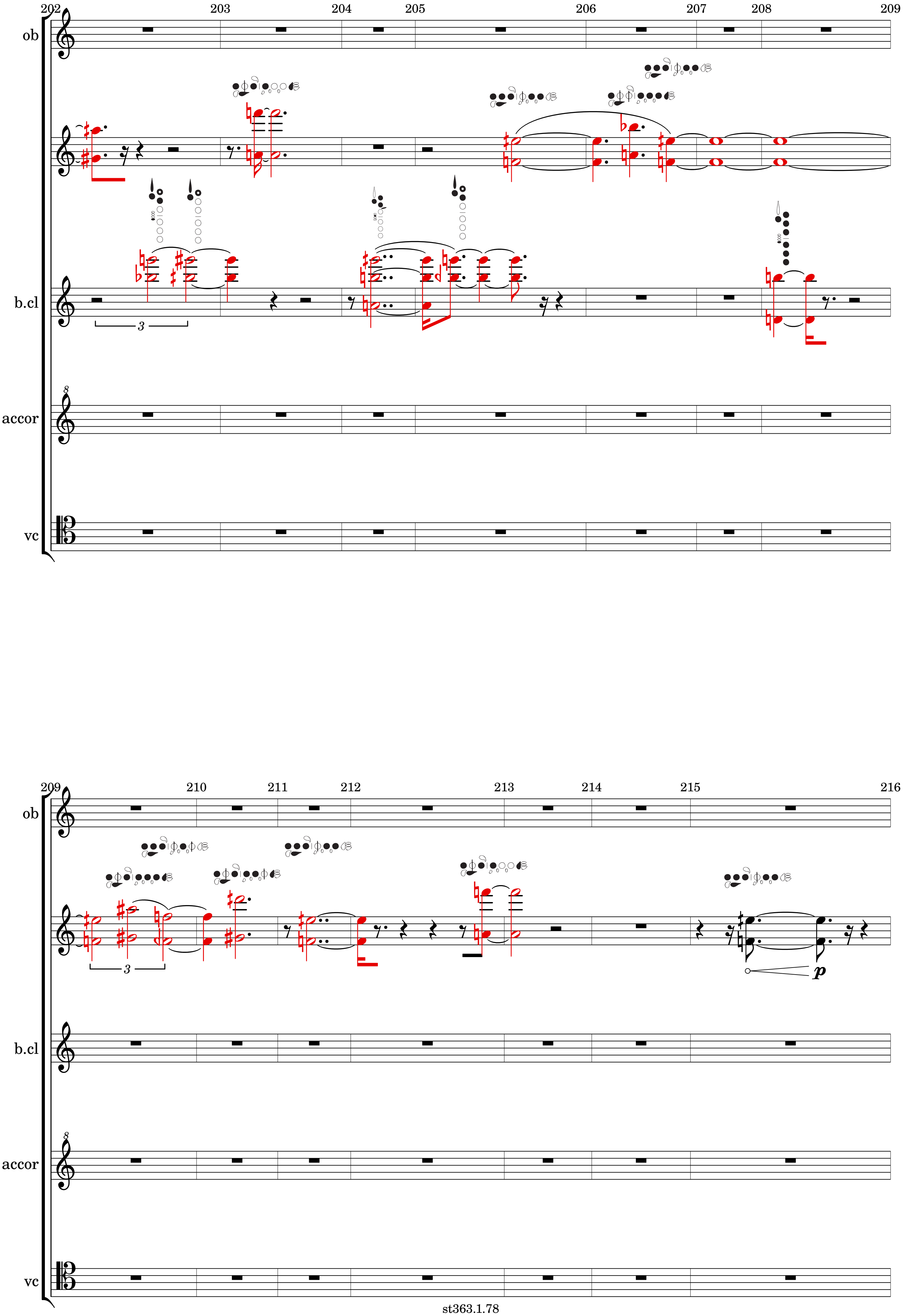

1.3.3
131
132
133
134
135
6
141
= 47
142
143
ob
b.cl
accor
vc
pp
pp

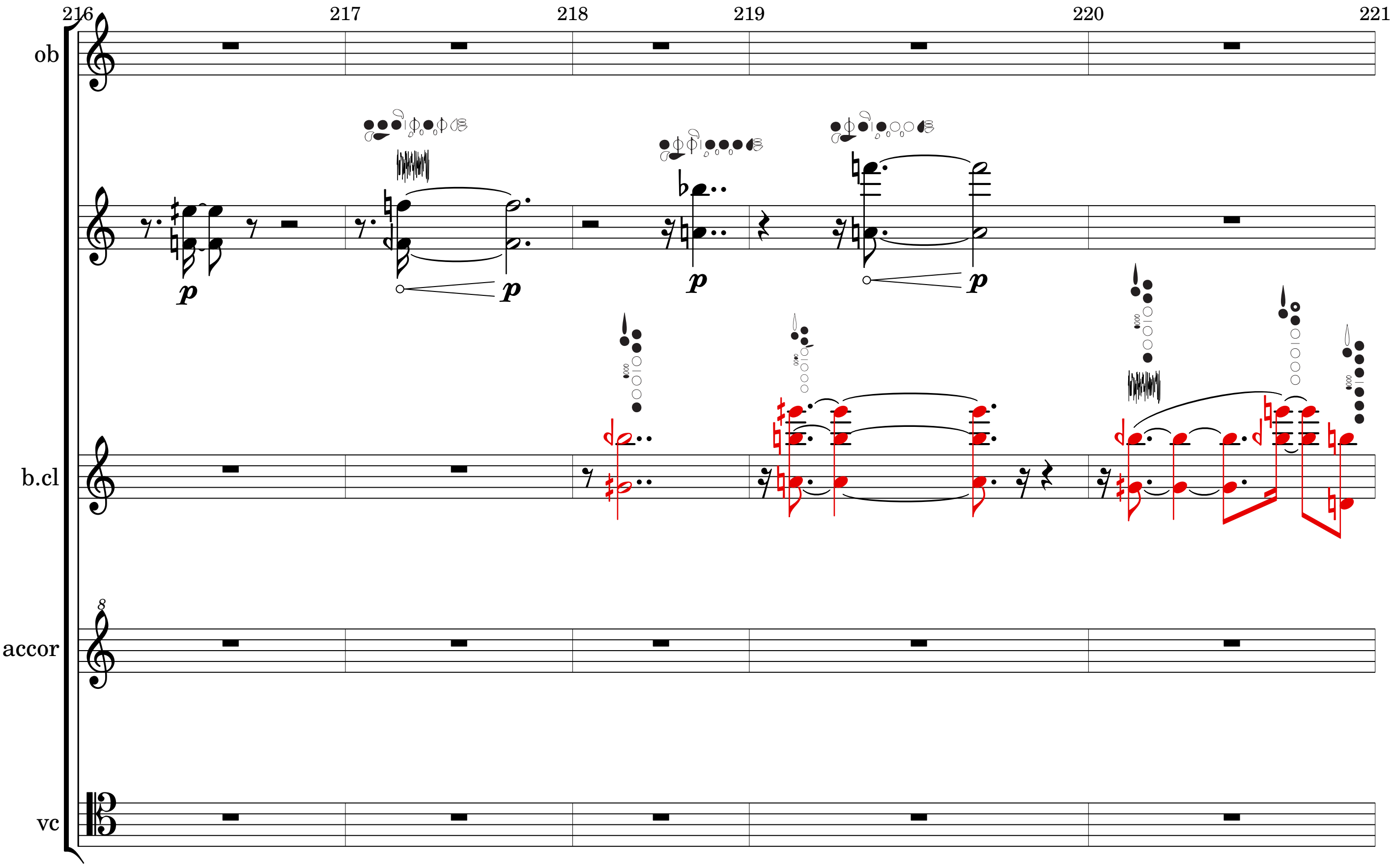

1.4
= 50
[gliss. sound]
bare teeth
Disgust, Aversion (blue)
143
144
145
146
147
148
149
150
ob
pppp
pp
b.cl
accor
vc
151
152
153
154
155
156
3
II[5+9+13+4]
IV[10]: bow/pluck behind ngers


156
157
161
162
163
164
165
ob
b.cl
accor
vc
pp
vib.
II[5+9+13+4]
165
166
167
168
169
170
173
ob
b.cl
accor
vc
pp
vib.

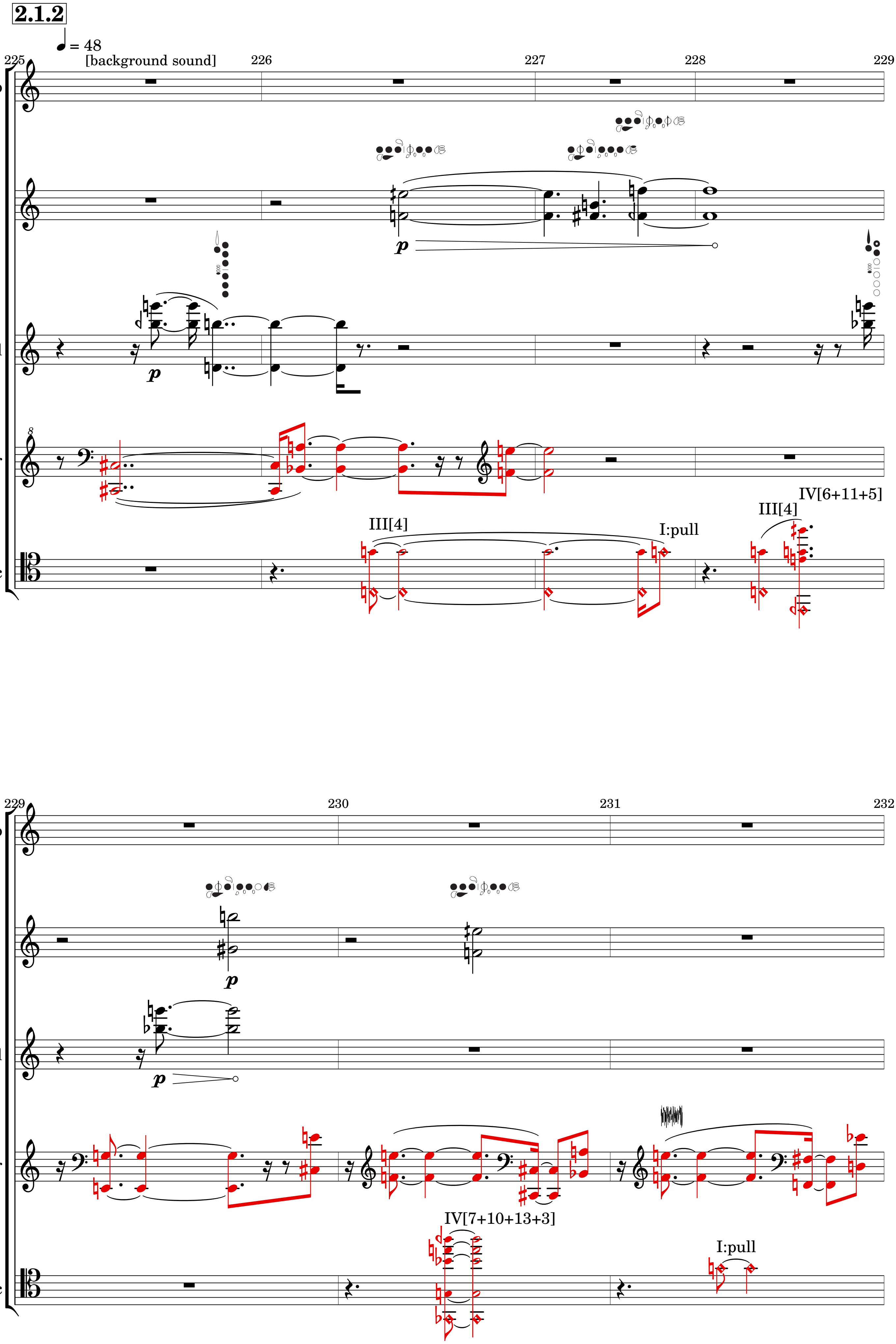

1.4.2
= 64
[background sound]
173
174
176
177
178
179
180
ob
b.cl
accor
vc
2
pp
vib.
3
III[4]
181
182
183
184
185
186
pppp
II[4+11+7+3]
IV[7+10+13+3]


1.4.3
= 49
186
187
188
189
190
192
193
194
ob
b.cl
accor
vc
vib.
pp
p
I:pull
IV[6+11+5]
IV[7+10+13+3]
IV[6+11+5]
2
= 65
[gliss. sound]
Horror, Terror (black)
194
196
197
198
199
200
201
202

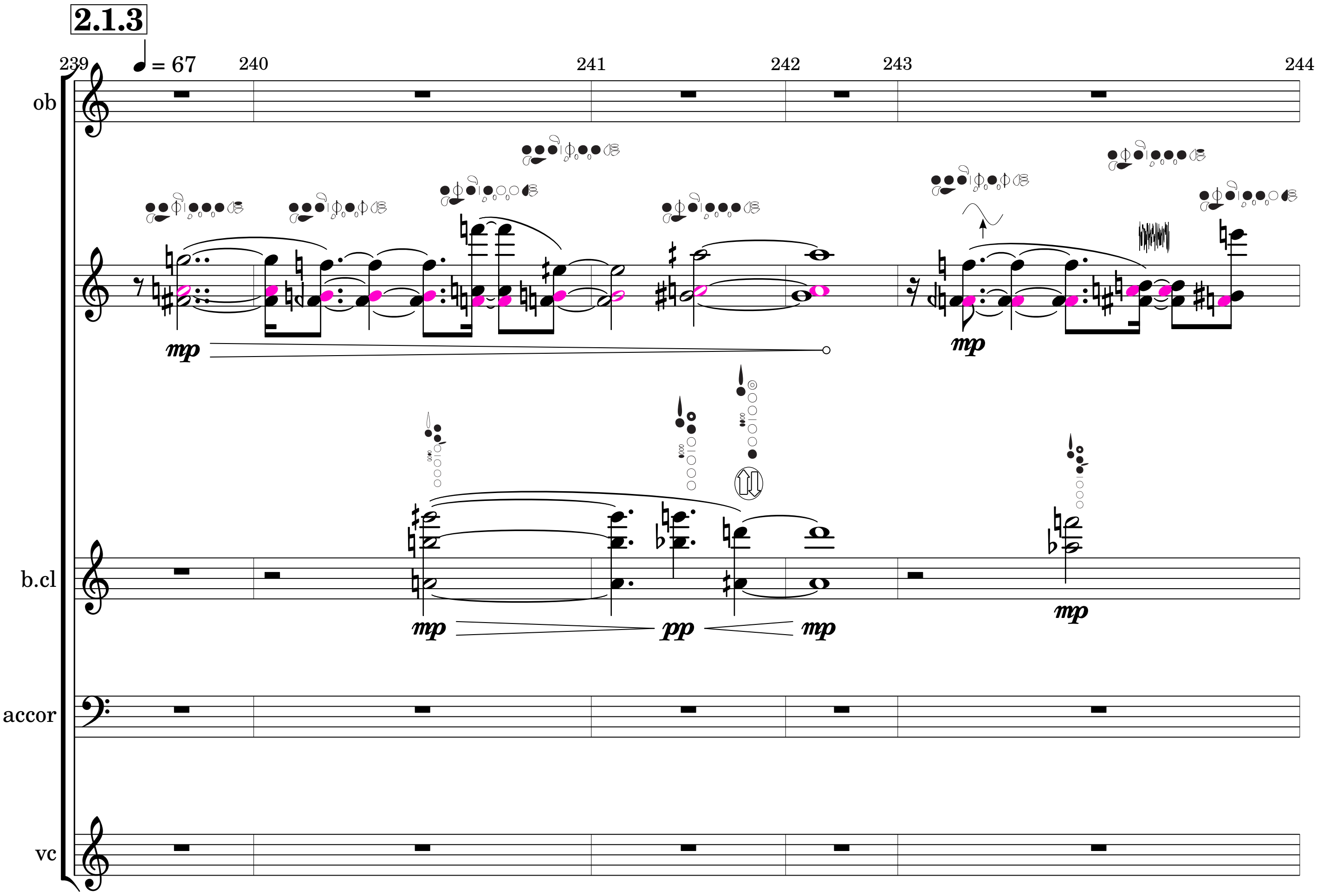

202
203
204
205
206
207
208
209
ob
b.cl
accor
vc
3
209
210
211
212
213
214
215
216
ob
b.cl
accor
vc
3
p

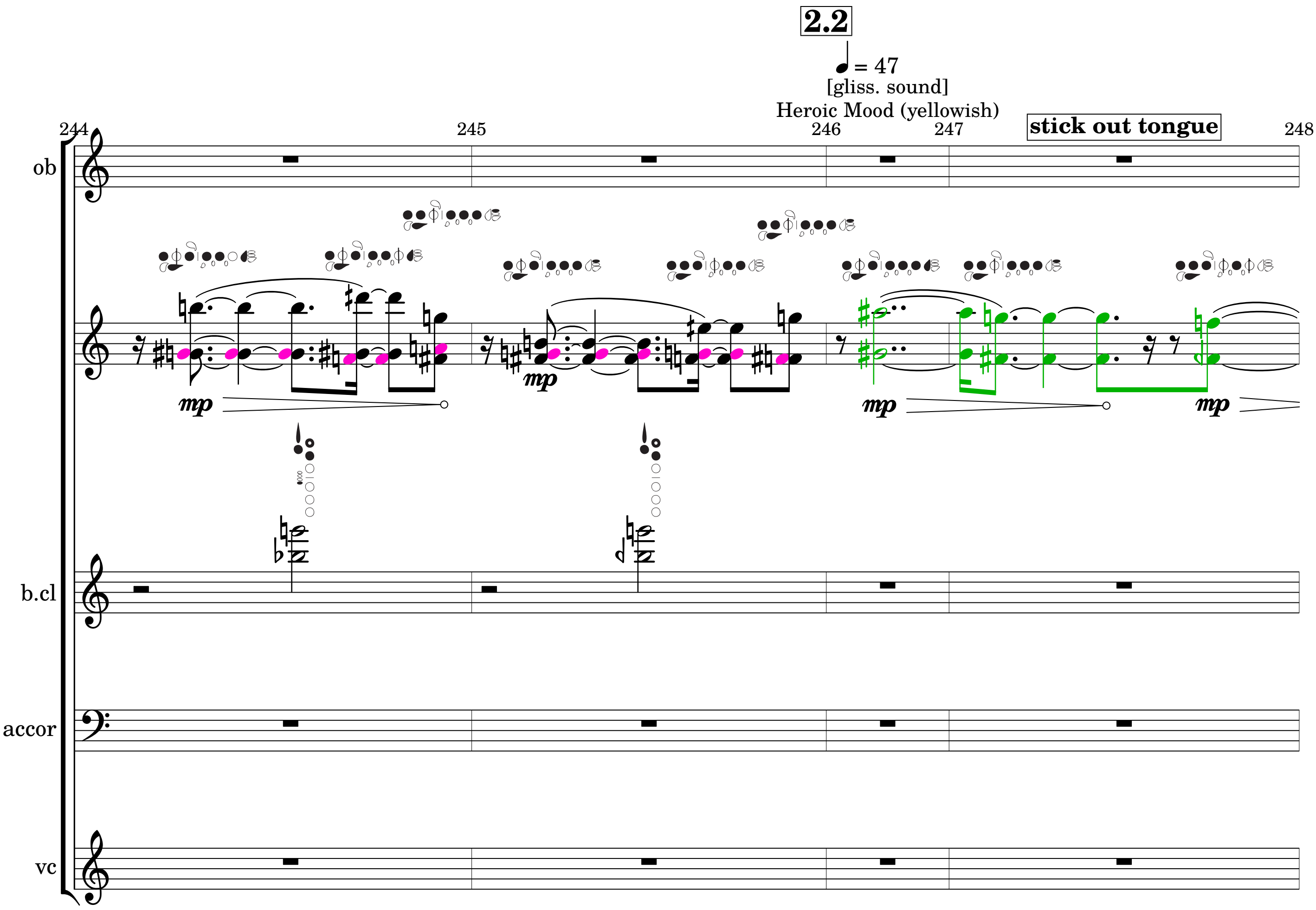

216
217
218
219
220
221
ob
b.cl
accor
vc
p

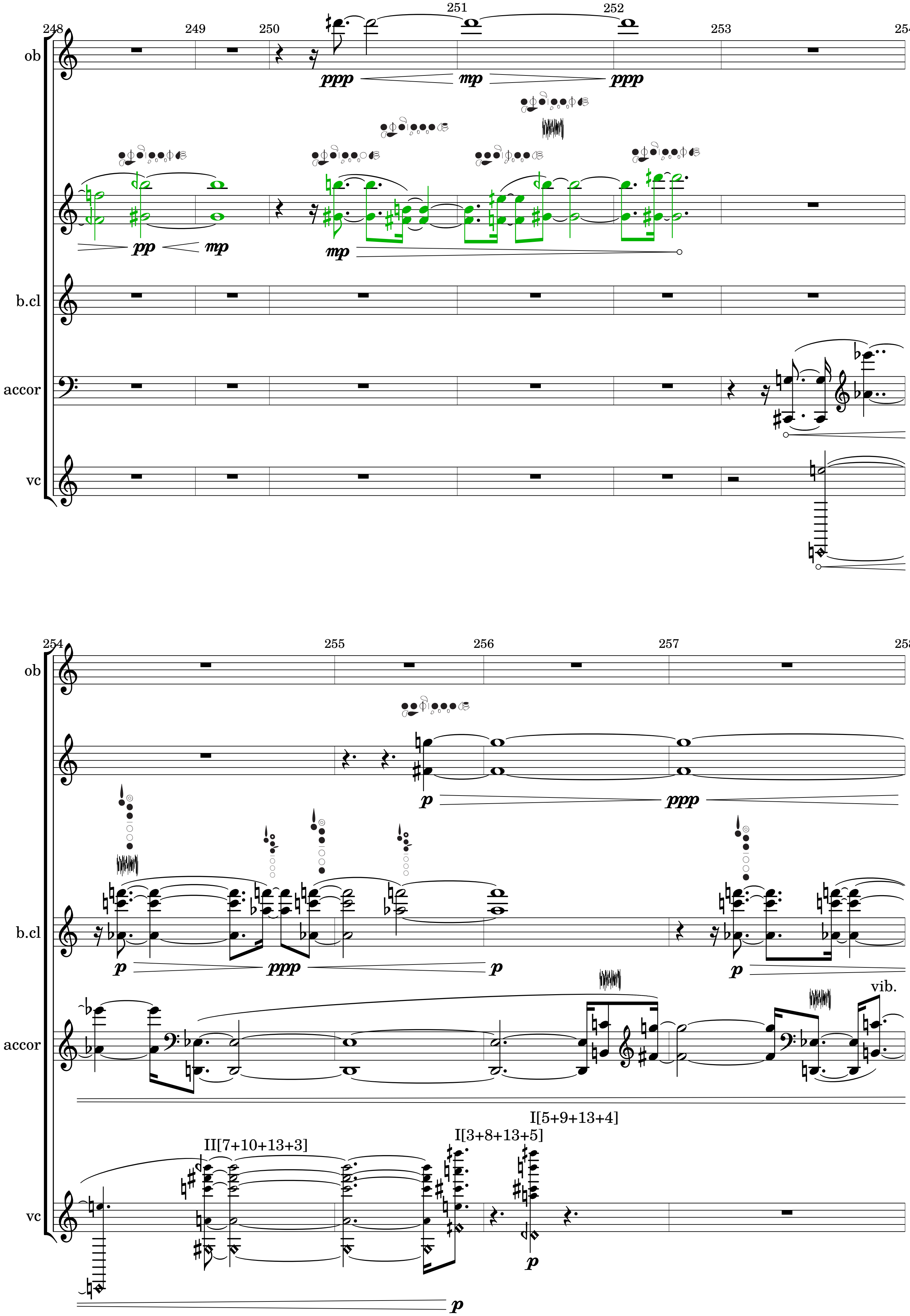

221
222
223
224
225
ob
b.cl
accor
vc
p

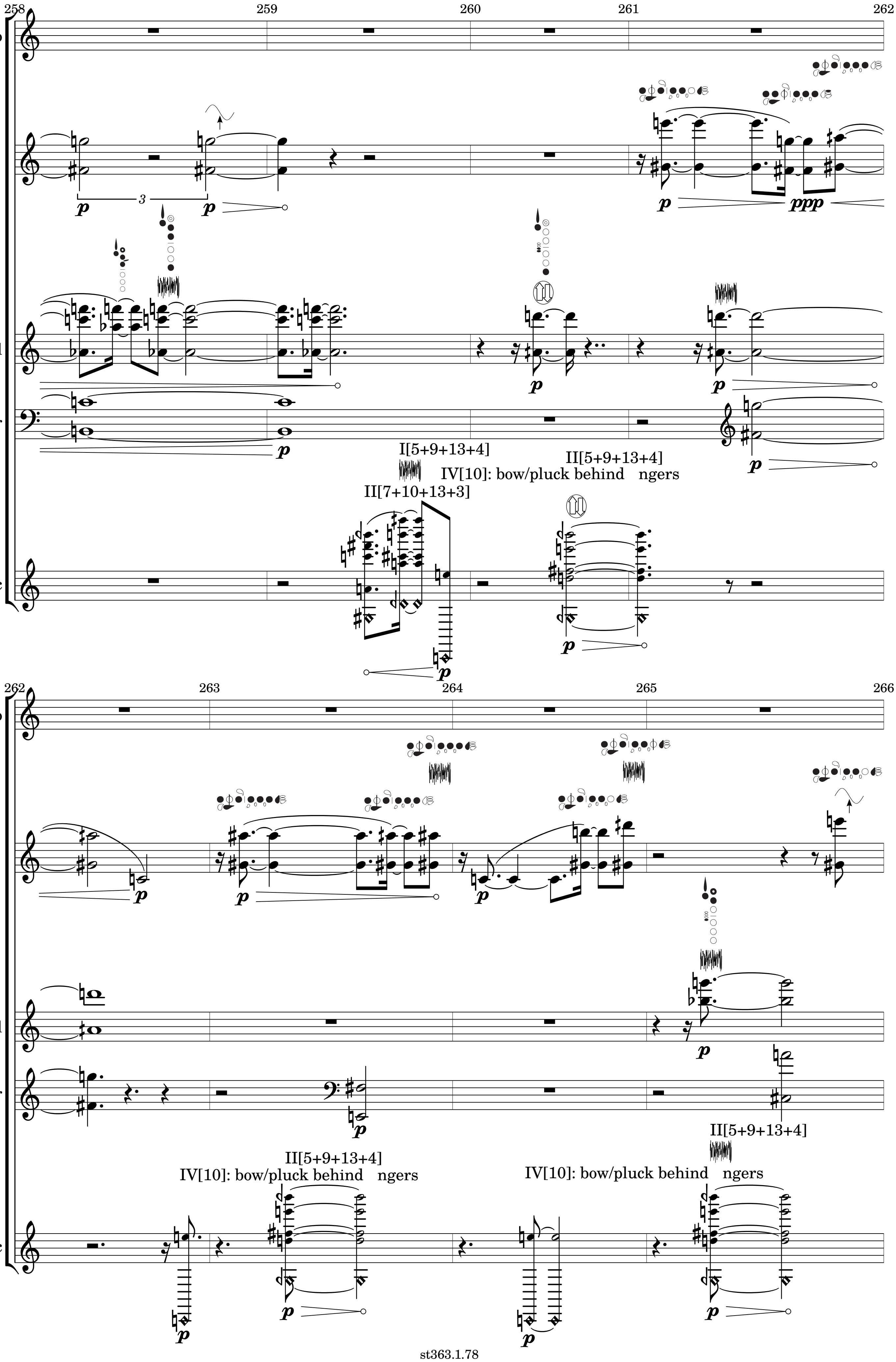

2.1.2
♩ = 48
[background sound]
225
226
227
228
229
ob
b.cl
accor
vc
p
III[4]
I:pull
III[4]
IV[6+11+5]
229
230
231
232
IV[7+10+13+3]
I:pull


232
233
234
235
236
ob
b.cl
accor
vc
p
mp
II[7+10+13+3]
IV[7+10+13+3]
I:pull
IV[6+11+5]
236
237
238
239
I[6+11+5]
I[2]
vib.
IV[10]: bow/pluck behind ngers
8


2.1.3
= 67
239
240
241
242
243
244
ob
b.cl
accor
vc
mp
pp
mp
mp
mp


2.2
= 47
[gliss. sound]
Heroic Mood (yellowish)
stick out tongue
244
245
246
247
248
ob
b.cl
accor
vc
mp
mp
mp
mp

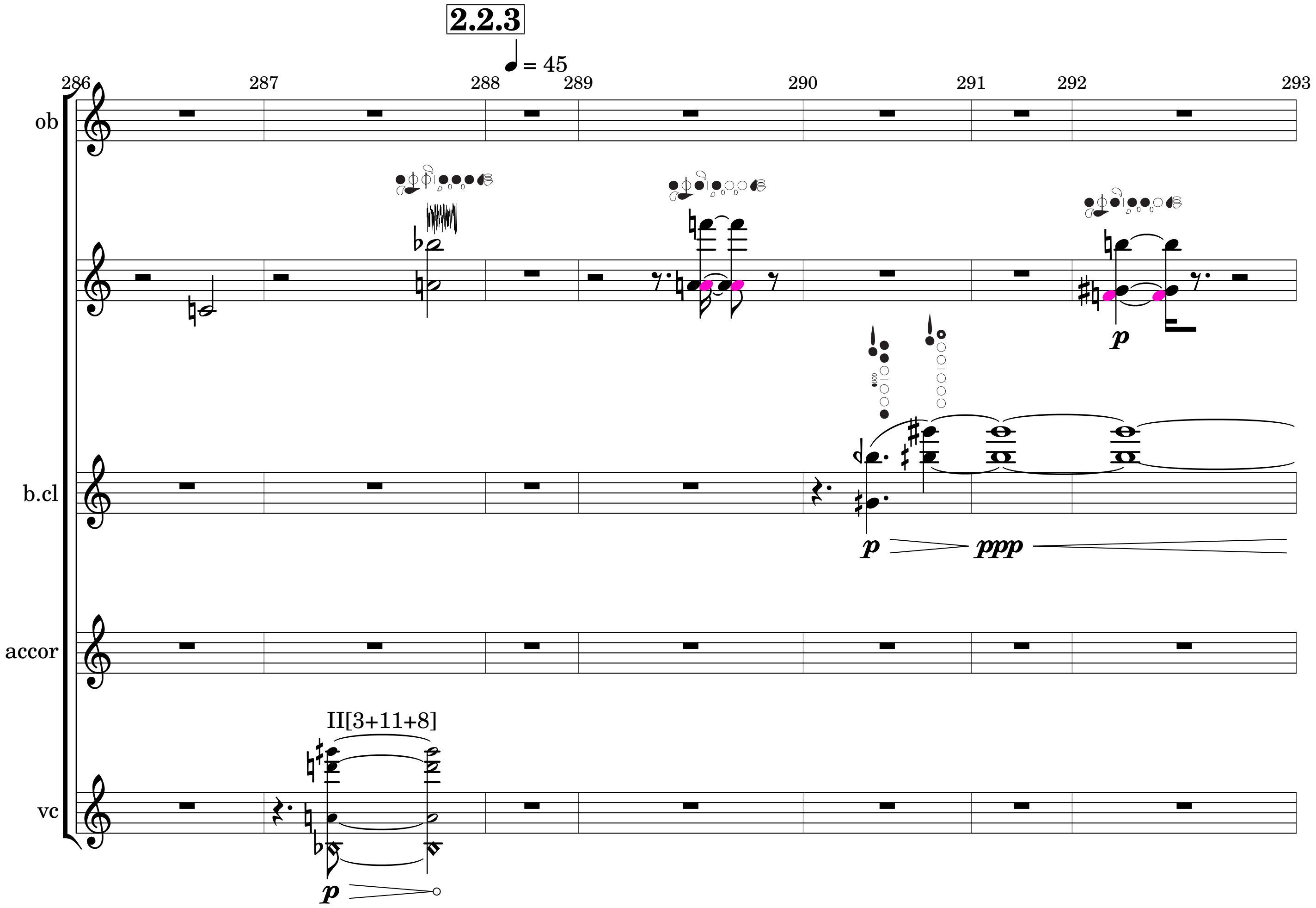

ob
b.cl
accor
vc
ppp
mp
ppp
pp
mp
mp
p
ppp
p
ppp
p
p
vib.
II[7+10+13+3]
I[3+8+13+5]
I[5+9+13+4]
p
p

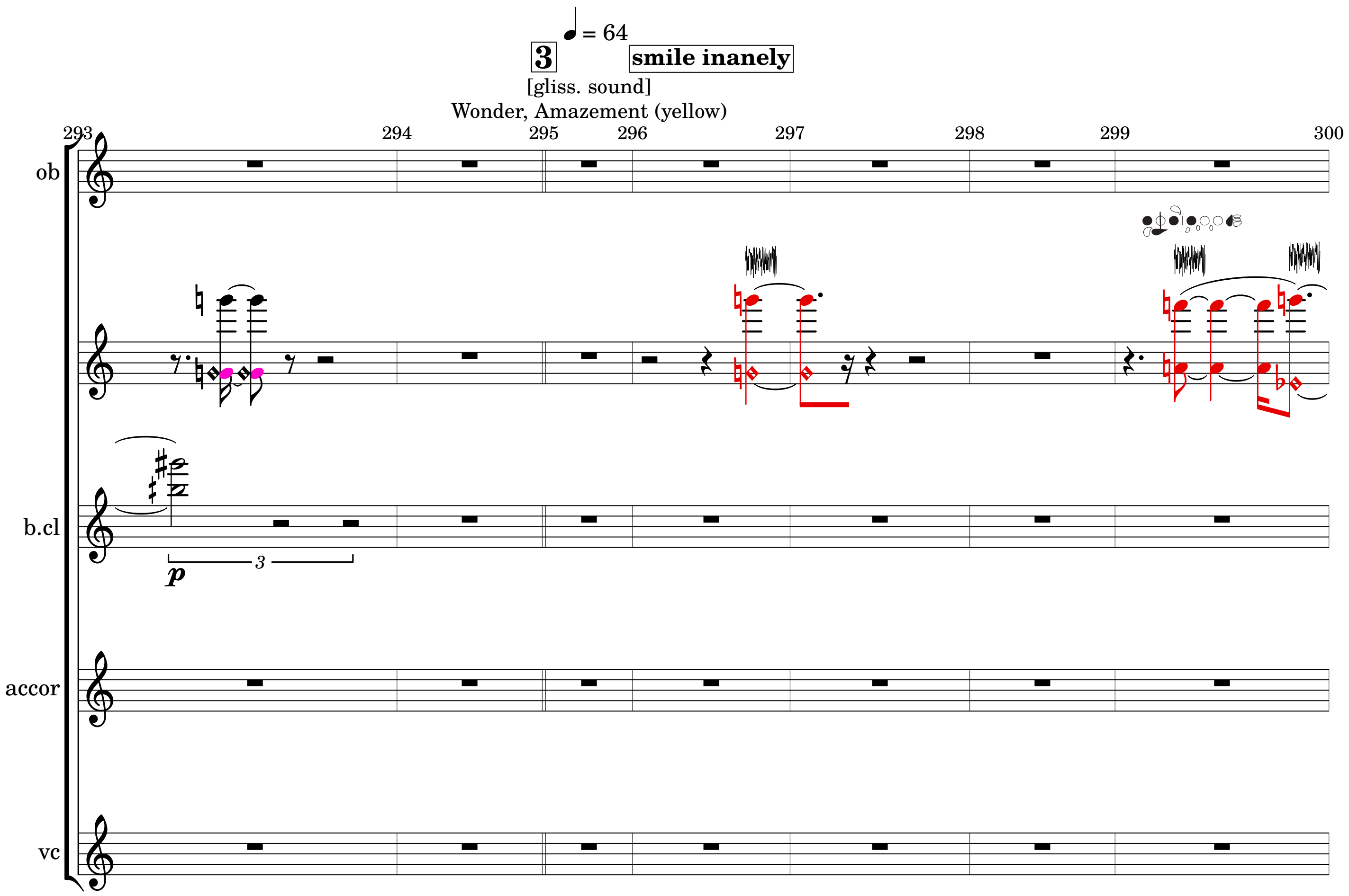

ob
b.cl
accor
vc
I[5+9+13+4]
IV[10]: bow/pluck behind ngers
II[7+10+13+3]
II[5+9+13+4]
IV[10]: bow/pluck behind ngers
II[5+9+13+4]
IV[10]: bow/pluck behind ngers
II[5+9+13+4]

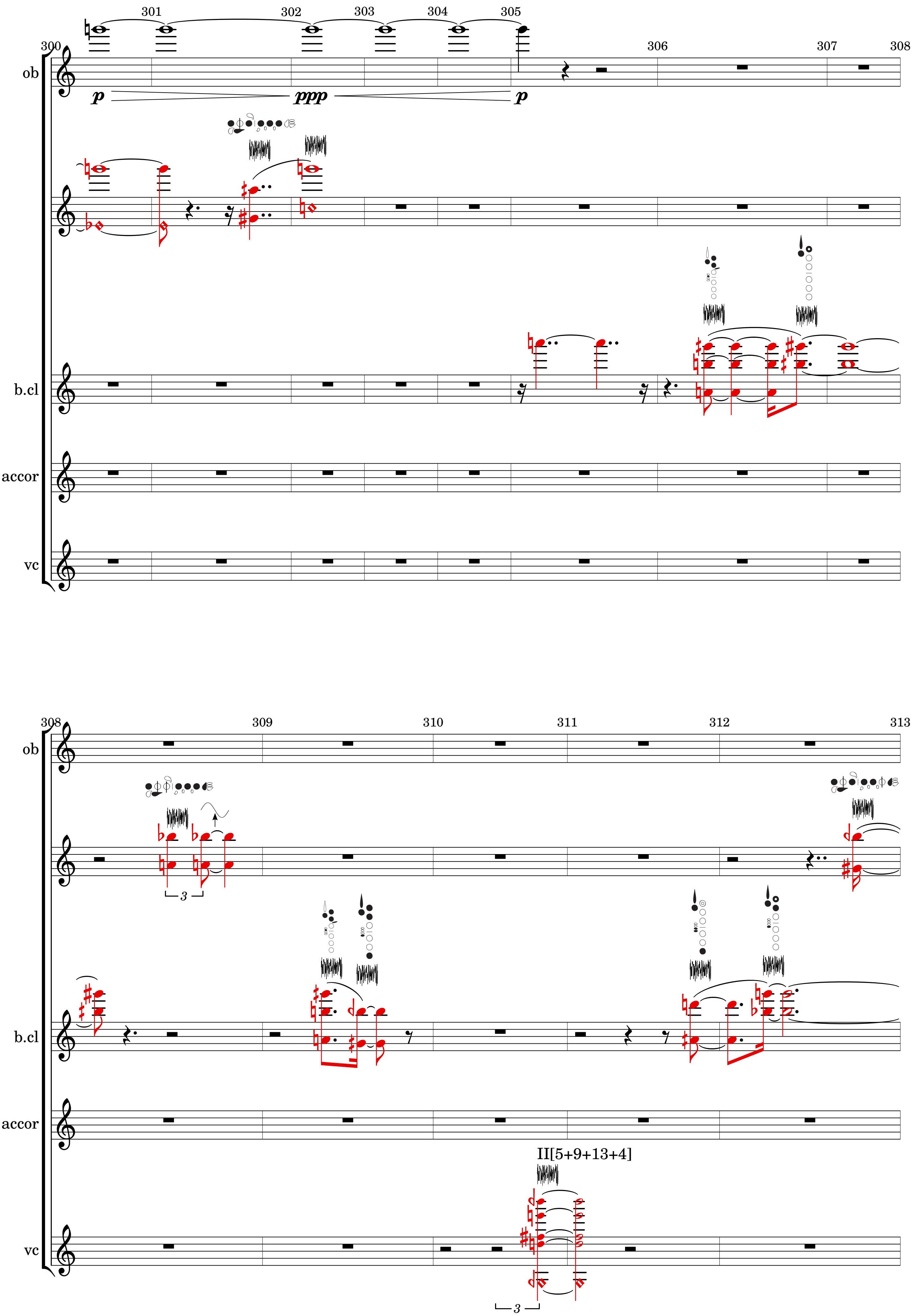

266
267
268
269
270
ob
b.cl
accor
vc
271
272
273
274
IV[10]: bow/pluck behind ngers

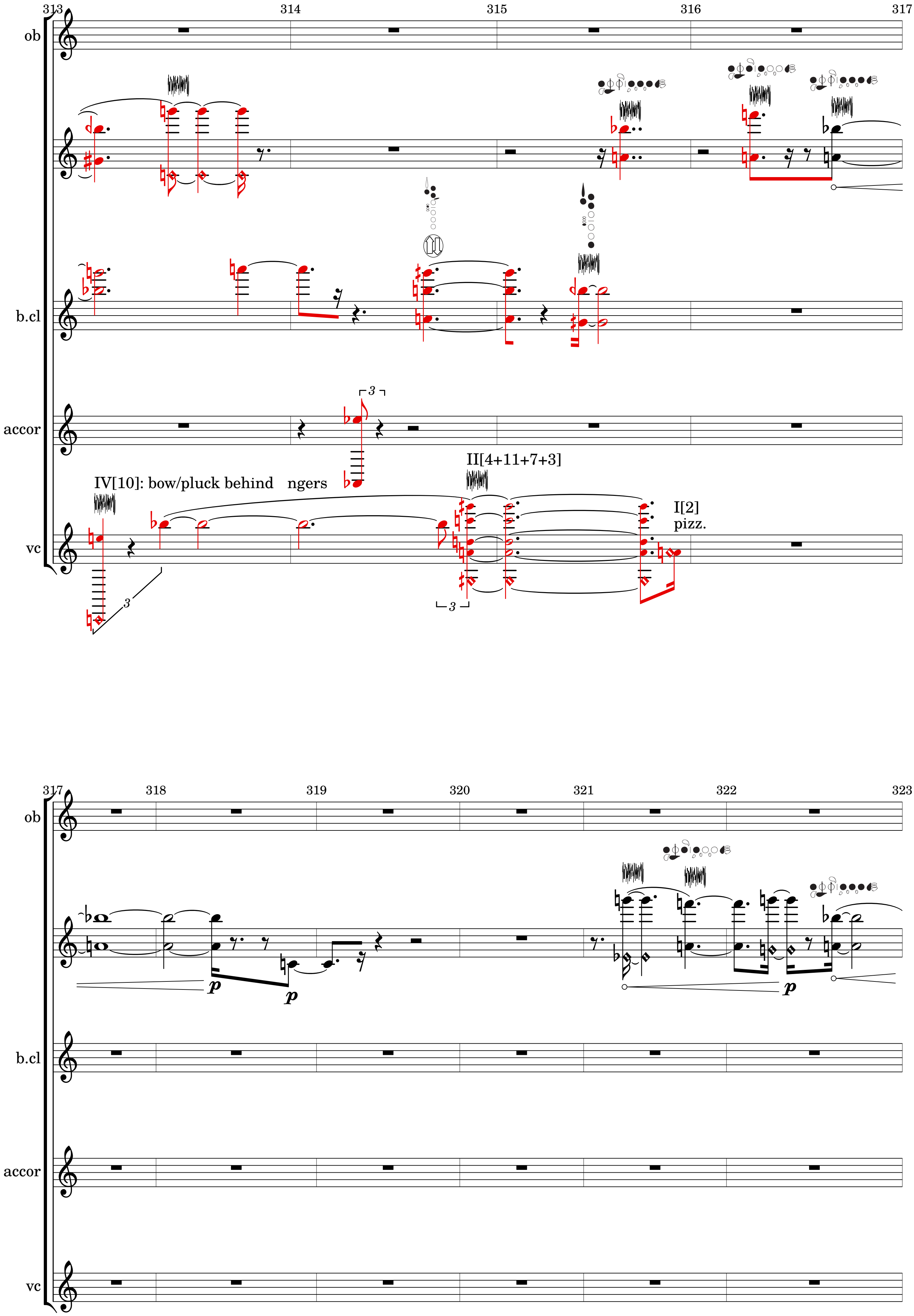
2.2.2

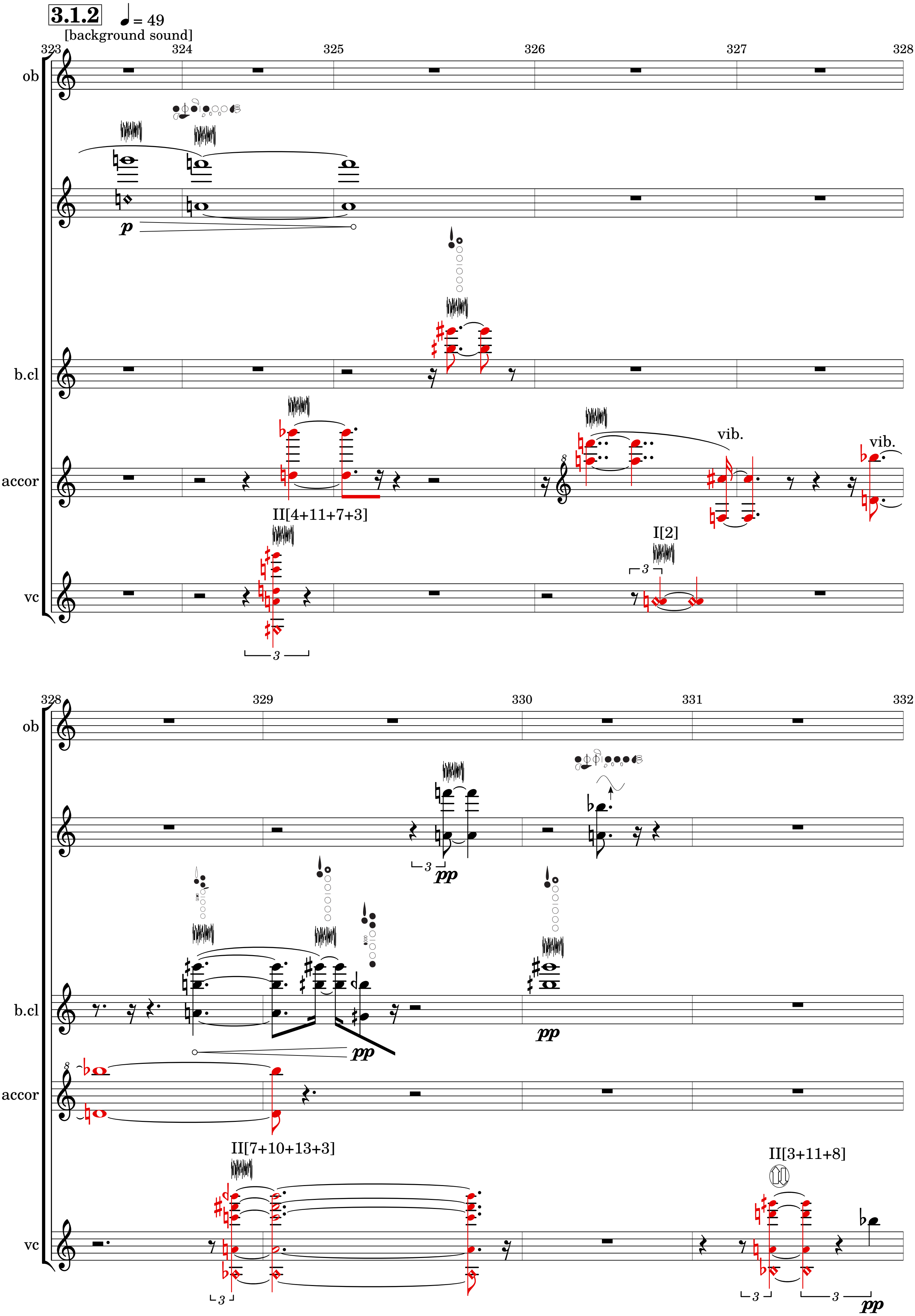

= 70
[background sound]
274
275
276
277
278
279
280
ob
b.cl
accor
vc
p
281
282
283
284
285
286
II[3+11+8]
II[7+10+13+3]

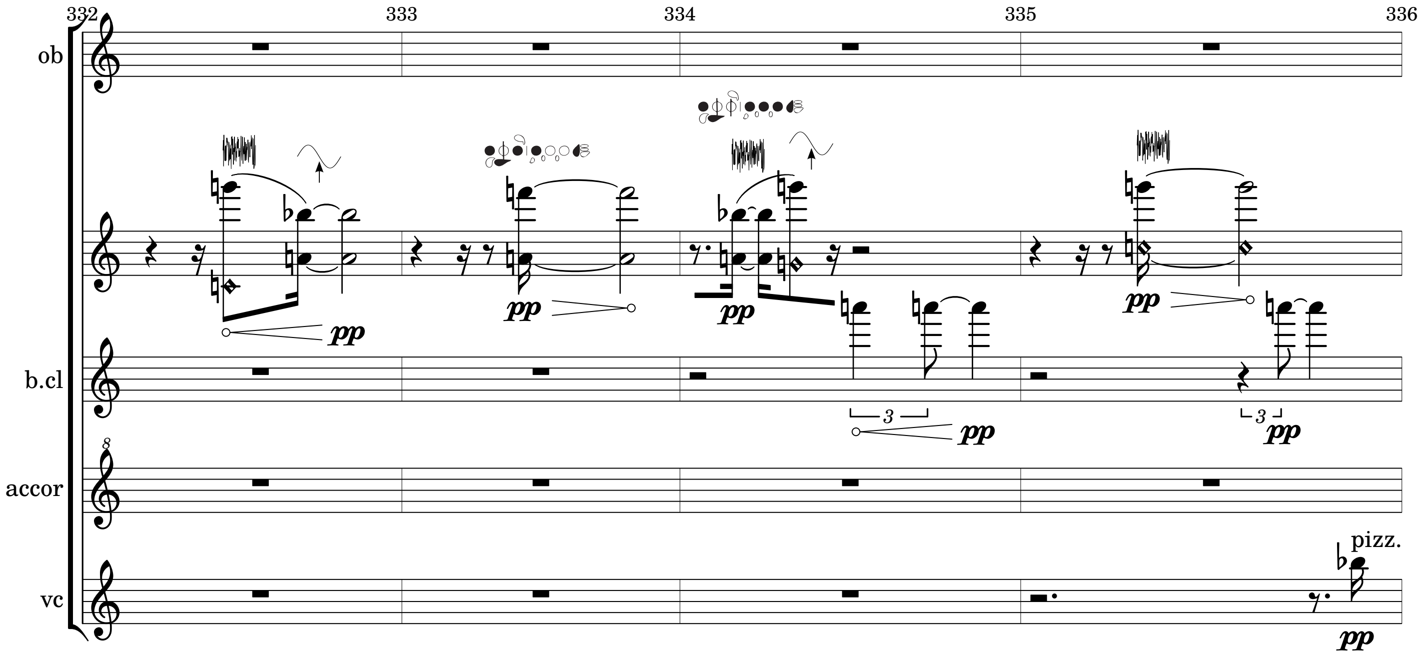

2.2.3
♩ = 45
286
287
288
289
290
291
292
293
ob
b.cl
accor
vc
II[3+11+8]
p
ppp
p

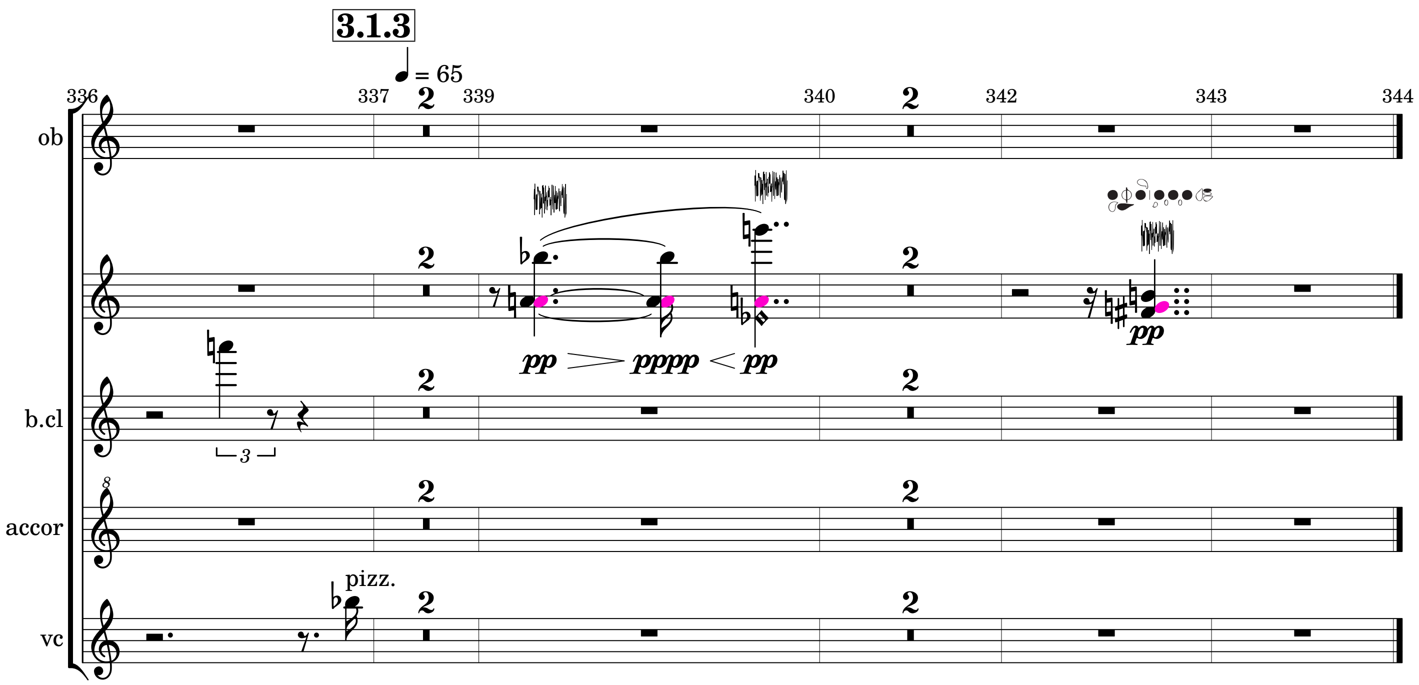

♩ = 64
3
smile inanely
[gliss. sound]
Wonder, Amazement (yellow)
293
294
295
296
297
298
299
300
ob
b.cl
accor
vc
3
p


301
302
303
304
305
300
306
307
308
ob
p
ppp
p
b.cl
accor
vc
308
309
310
311
312
313
ob
b.cl
accor
II[5+9+13+4]
vc
3
3


313
314
315
316
317
ob
b.cl
accor
vc
IV[10]: bow/pluck behind ngers
II[4+11+7+3]
I[2]
pizz.
317
318
319
320
321
322
323
ob
b.cl
accor
vc
p
p
p


3.1.2
♩ = 49
[background sound]
323
324
325
326
327
328
ob
b.cl
accor
vc
p
vib.
vib.
II[4+11+7+3]
I[2]
3
3
328
329
330
331
332
pp
pp
pp
pp
II[7+10+13+3]
II[3+11+8]
pp


332
333
334
335
336
ob
b.cl
accor
vc
pp
pizz.


3.1.3
= 65
336
337
339
340
342
343
344
ob
b.cl
accor
vc
2
pp
pppp
pizz.