Rubato4Rubato: A Brief Manual

Benjamin Honeray Lee[a*] and Guerino Mazzola[b]

September 16, 2024

1 Introduction

Rubato4Rubato is a software plugin for the Rubato Composer software that acts as a concrete implementation of the symbolic operator model of rubato. This document serves as a detailed manual and can serve as a convenient reference for all of the various tools found in the plugin.

1.1 Useful Links

- **The software.** A runnable `.jar` file containing Rubato and the plugin Rubato4Rubato can be found at `https://github.com/honeray/rubato4rubato`.
- **License.** A license for commercial usage of Rubato4Rubato and its patented intellectual property can be found at `https://license.umn.edu/product/rubato4rubato-a-software-platform-for-analyzing-rubato`.
- **Video Tutorial.** Rubato4Rubato was presented at the annual Música Analítica conference in 2023, and the video demonstration from this conference can be found at `https://youtu.be/lWt9-VRv_ug`. The video covers most features of Rubato4Rubato more extensively explains features useful to those who plan to use Rubato4Rubato themselves.

*Corresponding author. Email: lee02819@umn.edu

Figure 1: **The Rubato4Rubato GUI.** The Rubato4Rubato GUI allows a user to plot points along a symbolic score that cause accelerando or ritardando depending on the points plotted. Other features include tempo-based modification and direct viewing/export of the spline parameters used to implement the user-defined rubato.

2 Main GUI

The main GUI of the Rubato4Rubato plugin is comprised of three graphs superimposed on top of one another that the user can use to plot points. Because rubato involves changing both the symbolic onsets and offsets of notes, two sets of these graphs are provided (one to modify onset, the other offset). Without loss of generality, we detail the onset aspects of the GUI here; the offset aspects work similarly.

2.1 Symbolic Onset-Pitch Graph

The first of the three superimposed graphs is the onset-pitch graph shown in Figure 2. This graph is designed to contain a visual representation of a symbolic score's onset and pitch. As the name suggests, the graph's x-axis is the symbolic onset of a note, and the graph's y-axis is the symbolic pitch of a note. For the sake of user convenience, the y-coordinate of the origin of the graph is shifted to

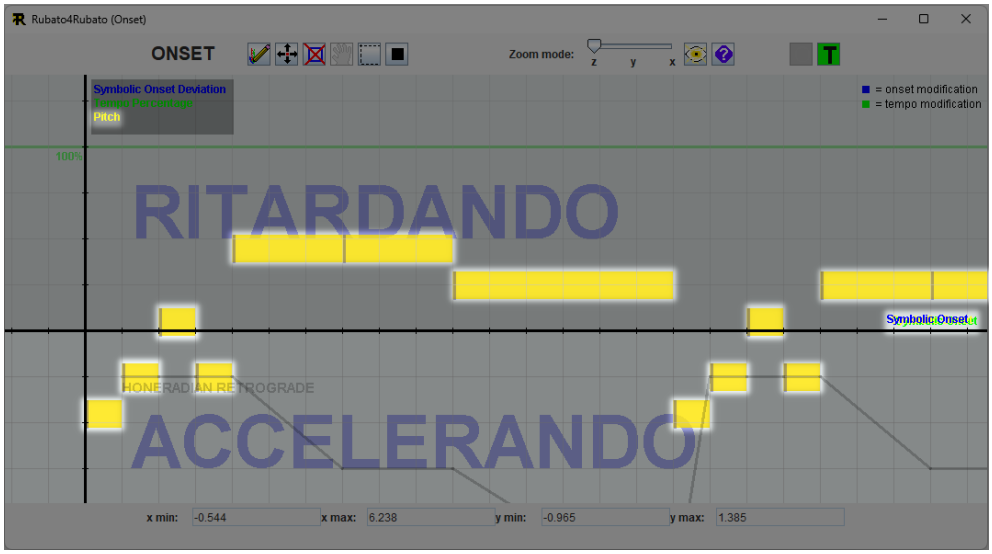

Figure 2: **The symbolic onset-pitch portion of the Rubato4Rubato GUI.** Other elements are darkened for clarity. The symbolic score to be modified by the user is shown in piano-roll notation similar to those seen in Digital Audio Workstations (DAWs).

the average pitch of all the notes of the score so that the user's view is centered. This view of the score is similar to the piano-roll notation used in Digital Audio Workstations (DAWs). Additionally, depending on whether one is working with symbolic onset or offset, a darkened line will appear at the beginning or end of each note, respectively.

This graph simply shows a user where they are in the score, and the score does not change in any way from this display process. To actually begin modifying the timings of notes in the score, the user must plot points on the symbolic onset/offset deviation graph.

2.2 Onset/Offset Deviation Graph

To shift the symbolic onset/offset of a note, the user must plot a point on the onset/offset deviation graph. As shown in Figure 3, this graph has the same x-axis as the symbolic onset-pitch graph (symbolic onset), but its y-axis represents how much the onset/offset should change. For example, if a user plots a point

Figure 3: **The onset/offset deviation portion of the Rubato4Rubato GUI.** Other elements are darkened for clarity. Points with positive onset/offset deviation lead to ritardando, while those with negative onset/offset deviation lead to accelerando.

with an x-coordinate corresponding to the onset of the fifth note in a score and a y-coordinate of 0.75, then the symbolic onset of the fifth note in the score will be delayed by three quarter notes.

Since the user likely intends for there to be a ripple effect when delaying a single note, all other notes after a shifted note will also be shifted. This means that in the example in the previous paragraph, all notes after the fifth note in the score will also be delayed by 3 quarter notes.

Since positive onset deviations delay notes and negative onset deviations make notes come earlier, the positive y-axis of this graph is labeled as ritardando, while the negative y-axis of the graph is labeled as accelerando. This references our definition of rubato in as a symbolic operator—accelerando and ritardando are treated as modifications to the symbolic onsets and offsets of notes rather than to tempo.

Figure 4: **The tempo percentage portion of the Rubato4Rubato GUI.** Other elements are darkened for clarity. The tempo percentage line begins at 100% and increases or decreases depending on the ritardando and/or accelerando specified by the user.

2.3 Tempo Percentage Graph

The last of the three graphs in the Rubato4Rubato GUI is the tempo percentage graph shown in Figure 4. Like the other two graphs, the x-axis of the tempo percentage graph is symbolic onset, and like the name suggests, the y-axis of the tempo-percentage graph is the percentage of the original tempo that some note has given some modification to its symbolic onset and/or offset.

While Rubato4Rubato mainly focuses on our new definition of rubato as changes to the symbolic onset and offset of notes, we also acknowledge that rubato as tempo modification is a much more well-known model of rubato. Therefore, we have taken the time to implement formulas and methods for approximating tempo modifications equivalent to changes to the symbolic onset and/or offset of notes in the score.

Briefly, for initial note onsets E_i and final, post-modification onsets E_f, the modified tempo of the nth note $T_{f,n}$ in the score equivalent to the original onset-modification can be determined through the below equation.

$$T_{f,n} = \frac{E_{i,n} - E_{i,n-1}}{E_{f,n} - E_{f,n-1}} T_{i,n}, \quad n > 1$$

Conversely, for initial tempos T_i and final, post-modification tempos T_f, the onset modification ΔE_n of the nth note in the score equivalent to the original tempo change can be determined through the equation below.

$$\Delta E_n = \frac{1}{c_n}(E_{i,n} - E_{i,n-1}) + E_{f,n-1} - E_{i,n}, \quad n > 1,\ c_n = \frac{T_{f,n}}{T_{i,n}},\ E_{f,0} = E_{i,0} = 0$$

It should be noted that in practice, this tempo modification equivalent to the user-defined changes to the symbolic onset and offset of notes (and vice versa) is an approximation rather than a conversion. The precision of this approximation can be adjusted by using the spline parameters windows as described in Section 3.1.

2.4 Honeradian Retrograde Line

Figure 5: **The Honeradian retrograde line.** Other elements are darkened for clarity. The Honeradian retrograde line tells the user how much accelerando can be applied before changing the order of notes. If the user wishes to deliberately induce localized retrograde of notes, they can assign points of accelerando whose magnitudes pass the Honeradian retrograde line.

In addition to the three superimposed graphs, the Rubato4Rubato GUI also contains a line indicating Honeradian retrograde as shown in Figure 5. Honeradian retrograde is a type of rubato that was discovered in the process of developing Rubato4Rubato. In brief, *Honeradian retrograde* is a form of rubato that changes the order of notes in the score, resulting in asymptotic tempo modification. This is explained in much more depth in our manuscript "Retrograde is Rubato: Playable Asymptotic Tempo and Other Consequences of Rubato as a Symbolic Operator".

To avoid unintentional Honeradian retrograde, a line has been implemented that indicates to the user a boundary past which any placed points will result in the changing of the order of notes.

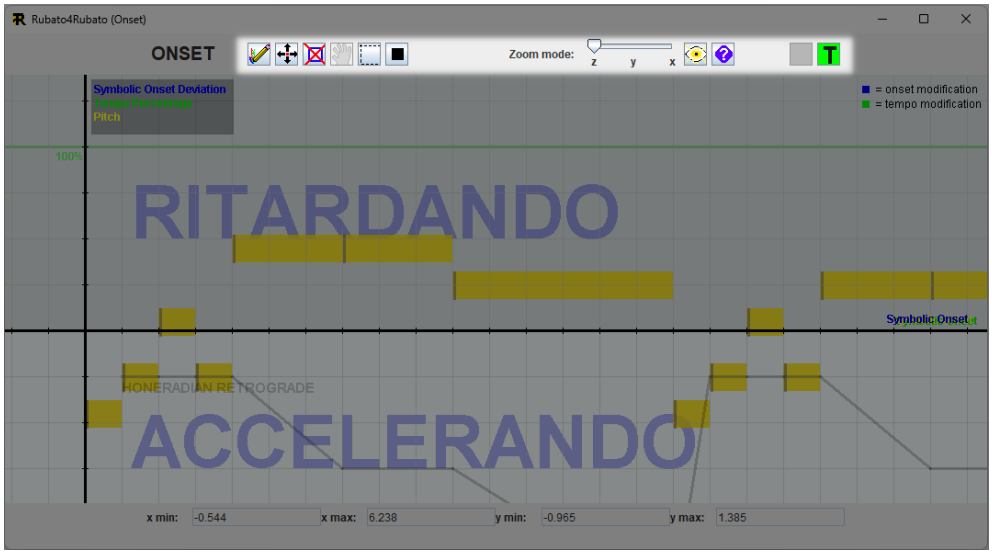

Figure 6: **Rubato4Rubato's user tools.** Other elements of the GUI are darkened for clarity. User tools include draw, move, erase, pan, select, stop point, zoom dimension slider, spline parameters window, help, onset modification mode, and tempo modification mode.

3 User Tools

To interact with Rubato4Rubato's GUI, the user has been supplied with several tools. From left to right in the highlighted section of Figure 6, these are

1. Draw: creates a point
2. Move: moves a selection of points
3. Erase: erases a point or selection of points
4. Pan: moves the user's view of the graph
5. Select: selects points
6. Stop Point: places a point to end a series of spline modifications at a specific note before the last note in the score
7. Zoom Dimension Slider: allows the user to specify whether they wish to magnify the x-axis, y-axis, or both (z)

8. Spline Parameters Window: see Spline Parameters Window.

9. Help: opens a help window with descriptions of all the tools

10. Onset Modification Mode: allows the user to implement rubato by modifying the symbolic onsets/offsets of notes. This is the default editing mode for Rubato4Rubato and is intended to be the main method of modifying timings of notes in the symbolic score.

11. Tempo Modification Mode: allows the user to implement rubato by modifying the tempos of notes. This mode is intended to be a middle ground for those more accustomed to the tempo modification model of rubato. It is possible to approximate one editing mode using the other, but repeatedly switching between the two modes will cause the interpolated splines to become increasingly unstable.

In addition to these tools, several keyboard shortcuts have also been implemented, the most prevalent of which are undo (ctrl+Z), redo (ctrl+Y), copy (ctrl+C), paste (ctrl+V), and zoom in/out (scroll wheel). Other keyboard shortcuts are detailed in the help window.

3.1 Spline Parameters Window

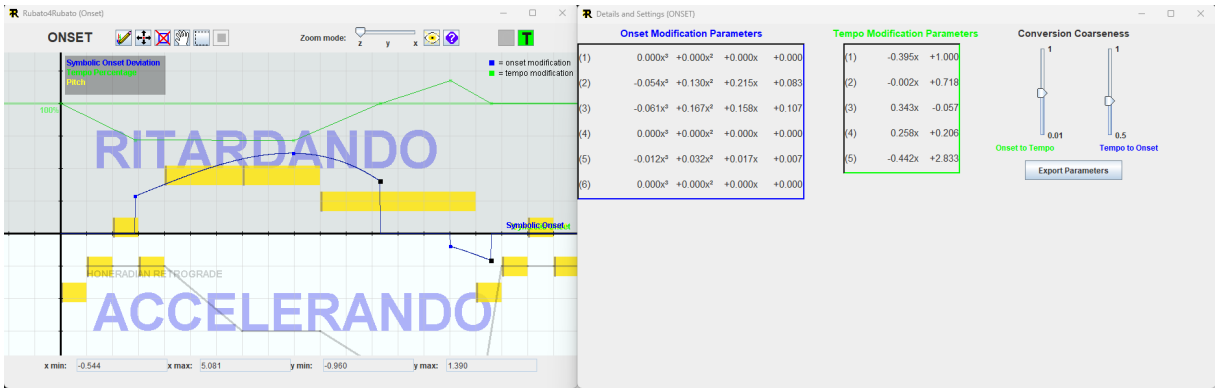

Figure 7: **Rubato4Rubato's spline parameters window.** Using Rubato4Rubato's spline parameters window to see the specific values (right) being used to generate the splines in the main GUI (left).

The spline parameters window is one of the most important tools in Rubato4Rubato's GUI. Here, one can see the actual values of the cubic spline parameters being used to implement rubato over a defined set of points. Referencing the example illustrated in Figure 7, the Onset Modification Parameters describe the splines (or lack of) used for each interval of user-defined points. Then, using the equations detailed in Section 2.3 to approximate the equivalent change to tempo, a list of lines and their parameters is shown in the Tempo Modification Parameters section.

Lastly, since the formula to find a tempo change equivalent to a symbolic onset/offset change (and vice versa) is an approximation, the resolution of this conversion can be changed in the Conversion Coarseness section of the spline parameters window. When increasing the resolution, Rubato4Rubato inserts imaginary notes in between the notes of the actual symbolic score, allowing for a finer approximation of tempo change by artificially splicing notes into smaller segments that tempo can change between.

3.2 Data Export

Once the user has implemented rubato on a symbolic score through some set of points, they may wish to save the data as a separate file from Rubato4Rubato for easier additional analysis. To accomplish this, the user can use the Export Parameters button found in Section 3.1, which will export the following as a `.csv` file:

1. The values of the cubic spline parameters used to generate the splines from user-defined points
2. The intervals over which the cubic splines are used, which is determined by the x-coordinates of the user-defined points
3. The values of the linear spline parameters used to generate the approximate tempo modifications equivalent to the symbolic onset/offset modifications
4. The intervals over which the linear splines are used (determined in the same manner as the intervals for the cubic splines)

3.3 Chords

In addition to simple monophonic melodies, Rubato4Rubato can also handle the edge case of symbolic scores with chords as shown in Figure 8. Since the onsets and offsets of the notes in a chord are typically the same, the cumulative shift approach typically used would offset the chords too much, since it would consider the onset and offset of each note separately. To circumvent this, the onsets of notes are compared to the onsets of their neighbors. If their onsets match, they are considered to be in a chord, and the cumulative effects of the individual onsets and offsets of the chordal notes are ignored.

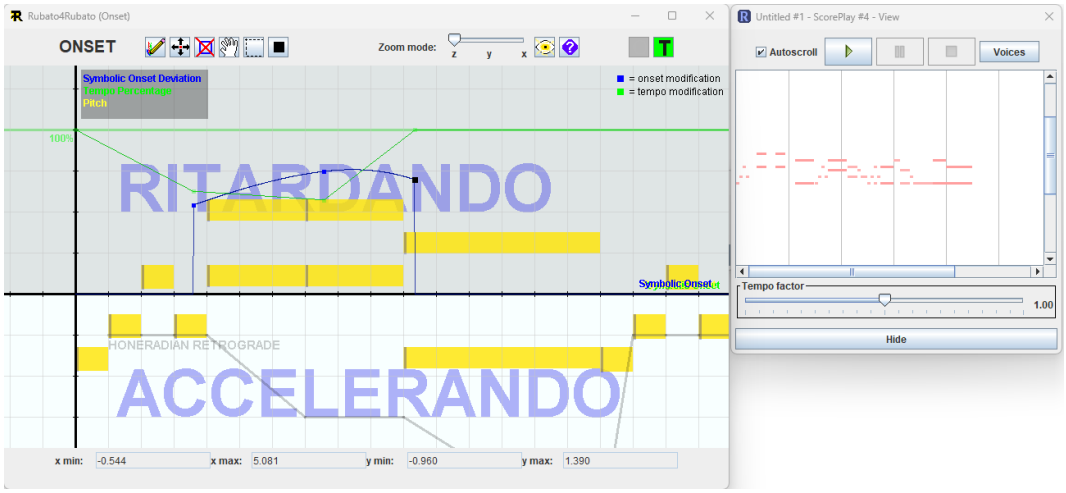

Figure 8: **Modifying a symbolic score containing chords using Rubato4Rubato.** The outputted MIDI file ignores redundant onset/offset shifts from chords.

4 Presets

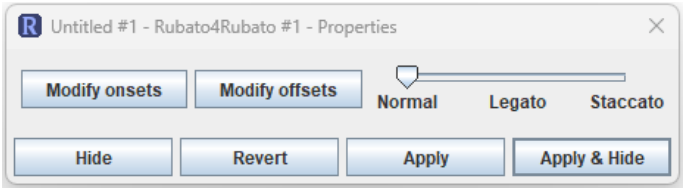

Figure 9: **Rubato4Rubato's properties window.** The presets available to the user are shown in the slider (right).

While allowing the user to modify the onset and offset of notes within the score is useful, there may be certain scenarios that a user would like at the literal push of a button. Therefore, the presets illustrated in Figure 9 have been implemented. More may be considered in the future.

4.0.1 Legato

Since legato by definition involves having the offset of every note match the onset of the next note, a preset could be defined such that the offset of the

ith note is set to the onset of the $(i+1)$th note. This would negate the offset calculations that are typically performed by Rubato4Rubato's algorithm.

4.0.2 Staccato

Since staccato by definition sets the offset of every note to a short time after the note's onset, a methodology similar to that of legato has been implemented to set the offset of each note to a constant, user-specified amount of time after the onset of each note.

4.1 Hierarchical Symbolic Onset Modification

Although not yet implemented in the software, the model of rubato as a symbolic operator is also capable of hierarchical modifications. In this case, the methodologies used to "split" notes for higher-resolution tempo-approximation described in Section 3.1 would be extended to modifications over different voices in the symbolic score. One could then plot spline changes for those specific voices over certain onset/offset intervals without modifying the overarching score-spanning spline changes, leading to children hierarchies of symbolic rubato. This feature has yet to be implemented in the software, but the theory points to this being a valuable feature that could be added in the future.