SITRANO: A MATLAB APP FOR SINES-TRANSIENTS-NOISE DECOMPOSITION OF AUDIO SIGNALS

Leonardo Fierro and Vesa Välimäki

Acoustics Lab, Dept. of Signal Processing and Acoustics
Aalto University
Espoo, Finland
leonardo.fierro@aalto.fi | vesa.valimaki@aalto.fi

ABSTRACT

Decomposition of sounds into their sinusoidal, transient, and noise components is an active research topic and a widely-used tool in audio processing. Multiple solutions have been proposed in recent years, using time–frequency representations to identify either horizontal and vertical structures or orientations and anisotropy in the spectrogram of the sound. In this paper, we present *SiTraNo*: an easy-to-use MATLAB application with a graphic user interface for audio decomposition that enables visualization and access to the sinusoidal, transient, and noise classes, individually. This application allows the user to choose between different well-known separation methods to analyze an input sound file, to instantaneously control and remix its spectral components, and to visually check the quality of the separation, before producing the desired output file. The visualization of common artifacts, such as birdies and dropouts, is demonstrated. This application promotes experimenting with the sound decomposition process by observing the effect of variations for each spectral component on the original sound and by comparing different methods against each other, evaluating the separation quality both audibly and visually. SiTraNo and its source code are available on a companion website and repository.

1. INTRODUCTION

Sound decomposition into sinusoidal, transient, and noise components[1] has been a topic of interest for many years now. It has proved itself as an useful tool in many application, such as beat tracking and tempo estimation [4], tonality estimation [5], spectral complexity reduction in cochlear implants [6], virtual bass systems [2], restoration of drum sounds and other music information retrieval tasks [7]. The sines–transients–noise (STN) separation is also applied in time-scale modification methods [8], having lately been combined with fuzzy logic to improve [1, 9] or evaluate time–stretching performance [3].

The idea behind STN separation is that any audio signal can be described as a sum of tonal events (sines), impulsive events (transients), and a residual part that does not clearly belong to either one of the other two classes (noise). Verma and Meng [10] first introduced the idea of a three-way decomposition, motivating that, in the context of signal analysis and synthesis, the cemented sines + noise model could highly benefit from the inclusion of a third component, transients. The motivation was to counter the transient smearing that was typically leading to mediocre sound quality in sines + noise modeling [10]. The sines + transients + noise model was also used by Levine and Smith [11] for data compression and for time and pitch scale modification.

Fitzgerald showed that it was possible to extract spectral masks via horizontal and vertical median filtering of the STFT (Short-Time Fourier Transform) to decompose an audio signal into its sinusoidal and transient component [12]. Later, Driedger et al. updated Fitzgerald's technique by introducing again a noise class to retrieve spurious information from the other two [13]. A follow-up method involving the use of spectral tensors to find predominant orientation angles and anisotropy in the time-frequency signal representation has been proposed by Füg et al. to improve the separation quality when sinusoidal structures with vibrato are involved [14]. More techniques for sines–transients separation have been recently proposed, with different approaches based on kernel additive matrix [15], non-negative matrix factorization [16], sinusoidal modelling [17, 18], and neural networks [19]. However, these methods do not involve a third class for the noise component.

While both [13] and [14] applied hard binary masks to define the sines, transients, and noise classes, STN decomposition was extended with the concept of fuzzy logic by Damskägg and Välimäki [1] in the context of time-scale modification. Fuzzy classification allows spectral bins to contribute simultaneously to the three classes, hence providing a more refined basis for the separation. This decomposition method was then developed by Moliner et al. [2] to retrieve perfect reconstruction by ensuring that the three spectral masks sum up to unity.

Typically, STN-decomposed sound quality is degraded by leakage in-between different components, loss of sinusoidality, or other artifacts, e.g. musical noise. To evaluate the quality of STN separation, common metrics for blind audio source separation performance assessment such as SDR (Signal-to-Distortion Ratio), SAR (Signal-to-Artifacts Ratio), and SIR (Signal-to-Interference Ratio) have often been used [13, 14, 20]. However, this evaluation process presents substantial drawbacks. Tests usually require at least three different sources to be mixed into one which is subsequently decomposed again for the separation quality assessment: this prevents non-synthetic audio inputs, e.g. music or speech, to be put under test. Also, unless dealing with perfectly sinusoidal / impulsive / noisy sources, the input sounds themselves are a composition of unknown sinusoidal, transient, and noise parts. While listening

[1] In recent literature, these components have been often referred to as "Harmonic–Percussive–Residual", but this terminology can be misleading, as the first class aims at identifying sinusoidal structures more than harmonical ones. For this reason, the authors prefer the "Sines–Transients–Noise" definition, following the Sines+Transients+Noise model formulation and being consistent with previous publications [1, 2, 3].

tests are uncommon and other objective measures have not been proposed yet, matching visual and sound comparison of the separation outputs can be a suitable way to investigate and conduct a lead-in evaluation of these decomposition techniques.

In this work, we propose a MATLAB application designed for this specific purpose. Named SiTraNo after the sines + transients + noise model, this toolbox enables instantaneous control over the desired separation method and the amount of sines, transients, and noise in the audio input. The graphical user interface (GUI) shows the current output waveform based on the chosen mix of spectral components and features a selection of spectral tools, such as a spectrum analyzer and the individual spectrograms for each of the three classes. A simple visual detection of artifacts in sinusoidal and transient classes based on small local maxima in the time-frequency representation is implemented to provide visual information to match the audible cues used to subjectively assess the quality of the decomposition. While sines–transients separation is already available in open–source libraries and applications, such as Librosa [21], Zen [22] or the TSM Toolbox [23], SiTraNo stands out for its goal of providing a visual tool for decomposition that is immediate and both education- and evaluation-oriented.

The remainder of this paper is structured as follows. In Section 2, methods available in SiTraNo for STN decomposition are briefly reviewed. Section 3 shows the GUI of the MATLAB application, its functionalities, controls, plots and related user-controlled parameters. Some applications for which SiTraNo can be used are described in Section 4. Finally, Section 5 concludes this paper with a summary and general remarks.

2. SEPARATION METHODS

The separation methods implemented in SiTraNo are based on a time-frequency representation of the input signal. For this reason, the STFT of the audio signal x is computed as:

$$X(m,k) = \sum_{n=-M/2}^{M/2} x(n+mH)\, w(n)\, e^{-j\omega_k n}, \tag{1}$$

where $x(n)$ is the input signal, $w(n)$ is the analysis window, m is the frame index, k is the spectral bin, H is the hop size, M is the frame length in samples, j is the imaginary unit, and ω_k is the normalized central frequency of the k^{th} spectral bin. This section presents the four methods included in our current implementation.

2.1. Harmonic–Percussive (HP) separation

The HP method aims at separating the sinusoidal component from the transient component. Fitzgerald noted that sinusoids compose flat lines in time direction in the spectrogram; vice versa, impulsive events appear as flat lines in the frequency direction [12]. Hence, horizontal (time-oriented) and vertical (frequency-oriented) median filtering can be applied on the signal STFT to highlight the desired component and suppress the other [12]:

$$X_h(m,k) = \\ = \text{median}\Big[|X(m-\frac{L_h}{2}+1,k)|, ..., |X(m+\frac{L_h}{2},k)|\Big], \tag{2}$$

$$X_v(m,k) = \\ = \text{median}\Big[|X(m,k-\frac{L_v}{2}+1)|, ..., |X(m,k+\frac{L_v}{2})|\Big], \tag{3}$$

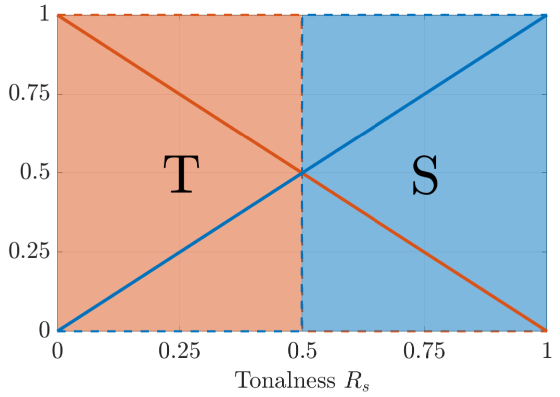


Figure 1: *Hard (dashed) and soft masks (solid lines) used for transients (red) and sines (blue) in Harmonic-Percussive separation.*

where X_h and X_v are the resulting horizontally-enhanced STFT and vertically-enhanced STFT, respectively, and L_f and L_t are the median filter lengths (in samples) in frequency and time directions, respectively. The two median-filtered spectrograms X_h and X_v are then used to extract the tonalness R_s and transientness R_t:

$$R_s(m,k) = \frac{X_h(m,k)}{X_h(m,k)+X_v(m,k)}, \tag{4}$$

$$R_t(m,k) = 1 - R_s(m,k) = \frac{X_v(m,k)}{X_h(m,k)+X_v(m,k)}. \tag{5}$$

In [12], R_s and R_t are used directly as soft spectral masks on X to obtain the relative spectral component:

$$X_i(m,k) = R_i(m,k)\; X(m,k) \qquad i = s, t. \tag{6}$$

Alternatively, a hard binary masking approach is possible:

$$S(m,k) = \begin{cases} 1, & \text{if } R_s > R_t \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

$$T(m,k) = \begin{cases} 1, & \text{if } R_t > R_s \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

$$\begin{aligned} X_s(m,k) &= S(m,k)\; X(m,k) \\ X_t(m,k) &= T(m,k)\; X(m,k). \end{aligned} \tag{9}$$

The relationship between the different masks is shown in Fig. 1.

2.2. Harmonic–Percussive–Residual (HPR) separation

Driedger et al. [13] improved Fitzgerald's technique by introducing a controllable separation factor β and a third class (noise), meant to describe those parts of the sound that are neither sinusoidal nor transient. From (4) and (5), a new set of spectral masks S (sinusoidal), T (transient), N (noise) can be derived by extending (7) and (8):

$$S(m,k) = \begin{cases} 1, & \text{if } R_s/R_t > \beta \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

$$T(m,k) = \begin{cases} 1, & \text{if } R_t/R_s > \beta \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

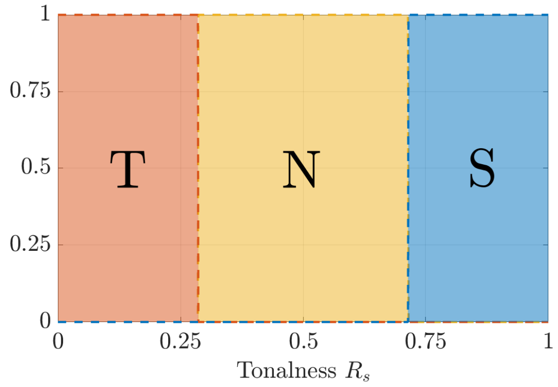


Figure 2: *Hard masks used for transient (red), noise (yellow), and sinusoidal (blue) in HPR separation, $\beta = 2.5$.*

$$N(m,k) = 1 - S(m,k) - T(m,k) \tag{12}$$

and their relationship for a chosen β is shown in Fig. 2. The spectral masks are then used on X to obtain the three spectral components:

$$\begin{aligned} X_s(m,k) &= S(m,k)\, X(m,k), \\ X_t(m,k) &= T(m,k)\, X(m,k), \\ X_n(m,k) &= N(m,k)\, X(m,k). \end{aligned} \tag{13}$$

It was observed that the quality of the HPR separation largely varies for sines and transients depending on the choice of the analysis window length M [24, 13]. A large window length for the STFT, ensuring good frequency resolution but poor time resolution, results in a good sines extraction but a low-quality transient output; vice versa, a small value of M leads to a better transients extraction but a worse sinusoidal description. To overcome this limitation, the decomposition process can be divided in two cascaded iterations [13]. In the first round, a larger analysis window is applied to extract the sinusoidal component, while transients and noise are mixed together:

$$\begin{aligned} x_s(n) &= \text{ISTFT}\Big[S_1(m,k)\; X(m,k)\Big], \\ x_{\text{res}}(n) &= \text{ISTFT}\Big[(T_1(m,k) + N_1(m,k))X(m,k)\Big]. \end{aligned} \tag{14}$$

Subsequently, the residual from the first round is separated again using a shorter analysis window, leading to the final decomposition:

$$\begin{aligned} x_t(n) &= \text{ISTFT}\Big[T_2(m,k)\; X_{\text{res}}(m,k)\Big], \\ x_n(n) &= \text{ISTFT}\Big[(S_2(m,k) + N_2(m,k))X_{\text{res}}(m,k)\Big]. \end{aligned} \tag{15}$$

2.3. HPR using Spectral Tensor (ST)

Füg et al. [14] noted that sounds exhibiting a vibrato, which carry sinusoidal information and are perceived as sines, do not present a strictly horizontal structure in the spectrogram, resulting in a leakage of energy in-between different spectral components when HPR separation is implemented. The strictness of the median filtering can be overcome using a structure tensor, a common image processing tool, to obtain a measure of the frequency change rate and local anisotropies in the spectrogram, which will then be used as features to define the spectral masks. The orientation angles $\alpha(m,k)$ and the anisotropy $C(m,k)$ of the spectral bins are obtained as described in [14]. Instantaneous frequency change rate $R(m,k)$ is computed for each bin from the orientation angles:

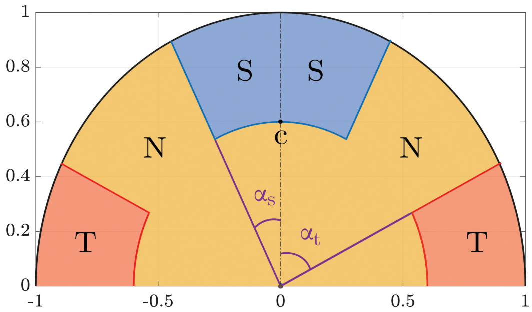


Figure 3: *Relationship between transient (red), noise (yellow) and sines (blue) masks for ST, $c = 0.6$, $r_s = r_t = 10$ kHz. Angles α_s and α_t are related to r_s and r_t, respectively, according to* (16).

$$R(m,k) = \frac{{f_s}^2}{HM} \tan\alpha(m,k). \tag{16}$$

The spectral masks are then obtained as follows:

$$S(m,k) = \begin{cases} 1, & \text{if } |R(m,k)| \leq r_s \;\wedge\; C(m,k) > c \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

$$T(m,k) = \begin{cases} 1, & \text{if } |R(m,k)| \geq r_t \;\wedge\; C(m,k) > c \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where c is the anisotropy threshold and r_s, r_t are the frequency rate thresholds. $N(m,k)$ is again computed as in (12); the relationship between the spectral masks and chosen values of parameters c, r_s, and r_t is shown in Fig. 3. The spectral components are then derived as in (13).

2.4. Fuzzy Separation

Damskägg and Välimäki [1] introduced the concept of fuzzy classification of the spectral bins, which was later extended by Moliner et al. [2] to ensure perfect reconstruction. In [2], a third membership function for noisiness is derived from (4) and (5):

$$R_n(m,k) = 1 - \sqrt{|R_s(m,k) - R_t(m,k)|}. \tag{19}$$

The spectral masks are computed as:

$$\begin{aligned} S(m,k) &= R_s(m,k) - \frac{1}{2}R_n(m,k), \\ T(m,k) &= R_t(m,k) - \frac{1}{2}R_n(m,k), \\ N(m,k) &= 1 - S(m,k) - T(m,k) = R_n(m,k). \end{aligned} \tag{20}$$

and their relationship is shown in Fig. 4. The spectral masks are once again used on X to obtain the spectral components, as in (13).

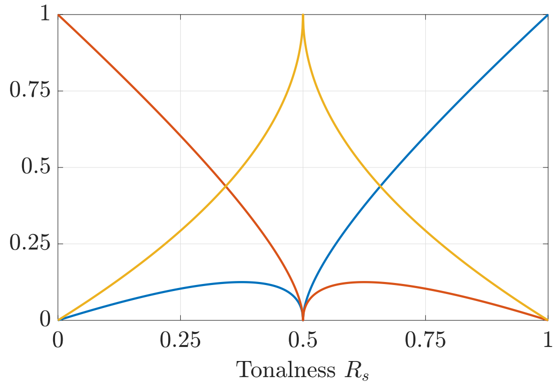


Figure 4: *Transient (red), noise (yellow) and sinusoidal (blue) membership functions for Fuzzy separation.*

3. IMPLEMENTATION AND GUI

3.1. Overview

SiTraNo aims at giving an immediate and easy access to STN spectral components. When an input file is imported, it is decomposed into sines, transients, and noise signals according to the chosen separation method, using ISTFT (Inverse STFT) to generate time-domain sounds. Through individual sliders, the amount of each component in the output mix can be chosen continuously between 0% and 100%. The mixed audio can be reproduced at any time, and saved as a Wave file.

The application puts its emphasis on the visualization of the decomposition process: the user is presented with the current form of the output waveform, plus the percentage of energy of each component and the spectrum of the signal at the selected time location; the user can also access spectrograms of sines, transients, and noise individually, which are complemented with a visual evaluation of possible artifacts. The SiTraNo GUI is shown in Fig. 5. The elements of the interface can be grouped in three main blocks: the *Control* (right) panel, where user controls and application status are accessible to the user; the *Audio* (bottom left) panel, where the mixed waveform and energy percentages are presented and the time location can be modified; and the *Analysis* (top left) panel, comprising a group of spectral plots.

3.2. Architecture and Dependencies

SiTraNo has been developed through MATLAB's App Designer [25], with dependencies on the Image Processing Toolbox [26] and the Audio Toolbox [27]. It has been tested for compatibility back to MATLAB R2018b. Further plans for this application are described in Section 5. The application installer and its source code can be accessed and downloaded both from the companion page [28] and the GitHub repository [29]. At the moment of writing, it is advised to use SiTraNo with audio inputs with a maximum duration of around one minute.

3.3. Standard Parameters

The application is set to work at a sample rate $f_s = 44.1$ kHz. The single-sided FFT (Fast Fourier Transform) is implemented to compute STFTs. A standard window length $M = 2048$ samples is applied, except for two-stage HPR that uses $M = 8192$ samples for the first decomposition and $M = 512$ samples for the second one. Window lengths have been chosen accordingly to [13, 14]. The hopsize is fixed at $M/8$ samples, and the number of FFT points is equal to the used window length. The Hann window function is used, following [1]. The horizontal median filter length is 200 ms (4 time frames at 44.1 kHz, $M = 2048$ samples), vertical median filter length is 500 Hz (23 frequency bins at 44.1 kHz, $M = 2048$ samples). For HPR, a separation factor $\beta = 2.5$ is used. Regarding ST, a Sobel filter is used as a differentiation kernel. Standard parameters include anisotropy threshold $c = 0.2$, frequency change rate thresholds $r_s = r_t = 10000$ Hz, Gaussian filter variance $\sigma = 2.8$, and tolerance $\epsilon = 20$, following [14].

3.4. Control Panel

The right panel in Fig. 5 is where most of the GUI controls reside. The user has access to a scroll-down window to select one among the separation methods described in Section 2 (double-round HPR is set as default) and to three sliders, which determine the amount of sines, transients, and noise to be mixed in the output sound. The number of FFT points for the spectrogram computation and artifacts threshold (see Section 3.6) can be arbitrarily selected by the user: valid candidate values for such thresholds usually lie in the $[-90, -80]$ dB range. The spectral plots can be refreshed at any time by means of an apposite button. Four buttons in the lower section of the panel allow the user to, respectively, play and stop the audio mix, open a new input file and save the current mix into an output Wave file. The sound reproduction can be looped by ticking a checkbox. The right panel also presents a lamp to monitor the application status. The lamp has three possible conditions: red, while main processing is ongoing after a new signal is imported or the separation method is changed; yellow, while the spectrograms are updated; green, while the application is active and running. Interactions with the GUI are disabled during yellow and red phases to avoid conflicting conditions to appear.

3.5. Audio Panel

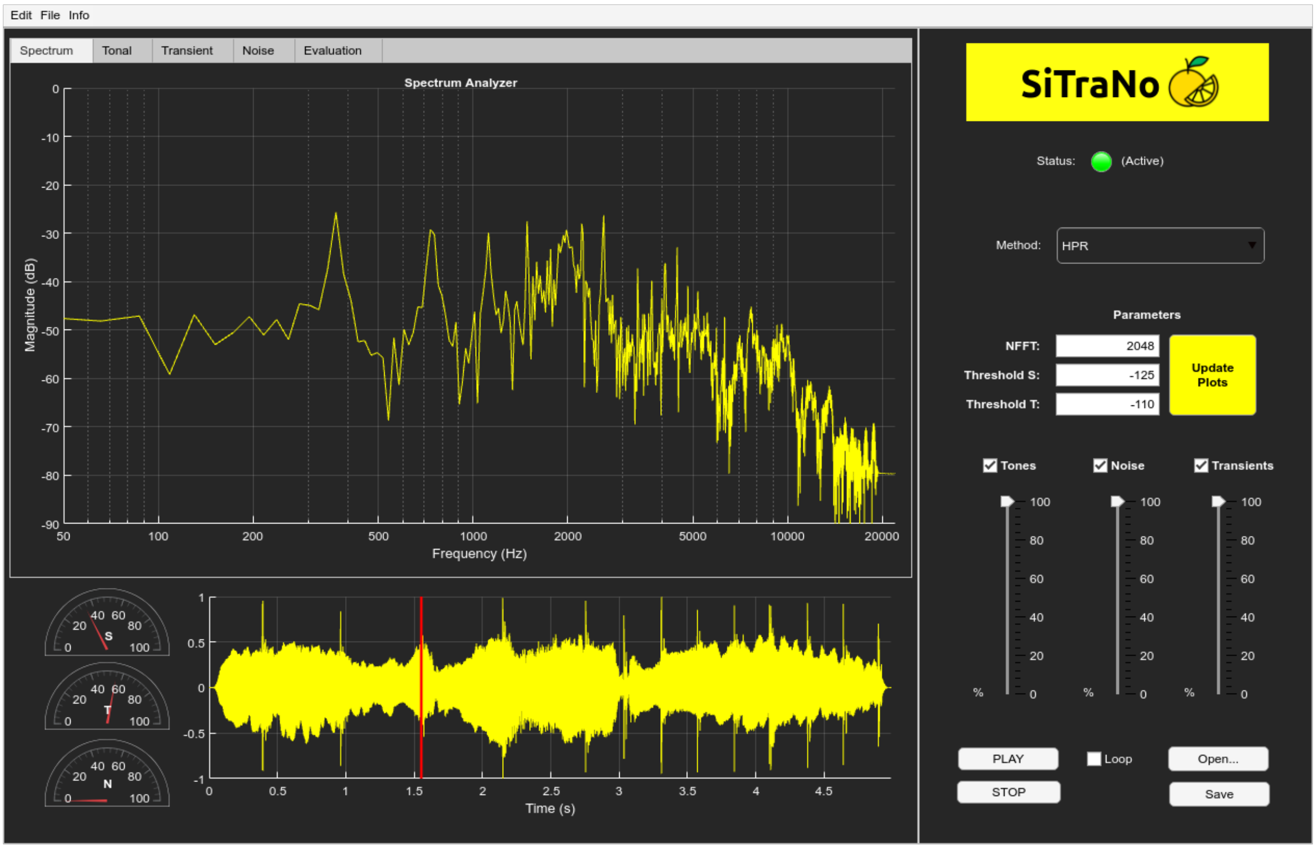
The bottom panel in Fig. 5 features the time-domain plot of the output mix, which is updated every time that either one of the slider values or the separation method are changed, and three gauges, used to visualize the amount of energy percentage per spectral component at the current time frame, which is indexed by a red cursor on top of the time-domain plot. Such a cursor automatically updates its position during audio playback, but can also be manually moved by the user.

3.6. Analysis Panel

The top left panel in Fig. 5 comprises a group of five tabs relative to the spectral analysis and evaluation of the audio signal. The first tab features a spectrum analyzer. The following three tabs show the spectrograms of, respectively, the sinusoidal, transient, and noise component. The last tab shows data relative to the evaluation of the decomposition.

3.6.1. Spectrum Analyzer

The first entry in the tab group shows a spectrum analyzer, also shown in Fig. 5, which is refreshed every time the time cursor is



Status: ● (Active)

Method: HPR

Parameters

NFFT: 2048

Threshold S: -125

Threshold T: -110

☑ Tones ☑ Noise ☑ Transients

□ Loop

Figure 5: *SiTraNo GUI, comprising the Analysis panel (top left), the Control panel (right) and the Audio panel (bottom left).*

updated. A sliding window centered in the current time location is used to compute the single-sided FFT of the audio mix, which is plotted on a logarithmic frequency axis with a $[-90, 0]$ dB magnitude scale. The quickness of the component mixing using the sliders, combined with the promptness of the time navigation, makes the spectrum analyzer a powerful tool for the visualization of each spectral component's contribution to the sound mix. For example, it is very easy to spot when a harmonic structure is present or to investigate how the spectral behavior changes during an impulsive event.

3.6.2. STN Spectrograms

In the following tabs in SiTraNo, the logarithmic spectrograms for, respectively, the sines, transients, and noise component are shown. Some examples of sinusoidal-only spectrograms and transient-only spectrograms are shown, respectively, in the left side and the the right side of Fig. 7. The STFT window length can be modified from the Control panel. The spectrograms are updated only when a different separation method is selected, or the user manually presses the refresh button. Time-frequency representations of the individual spectral components are particularly useful from a visual analysis point of view, as it makes very easy to compare the quality of transient and sinusoidal extraction among multiple methods.

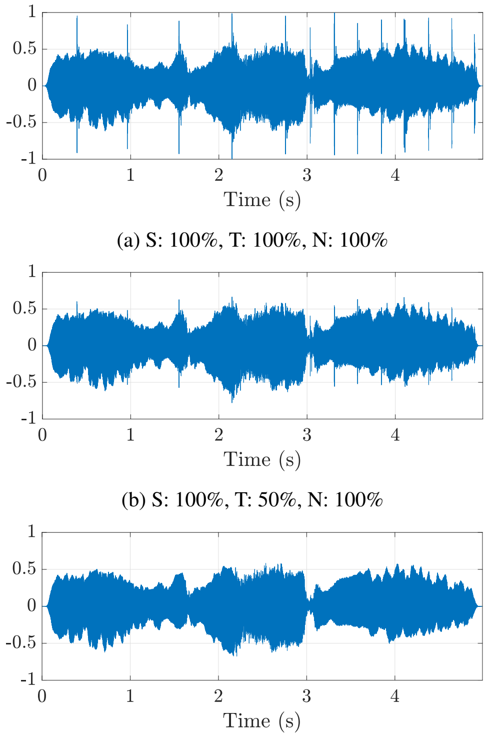
A basic artifact visualization tool is also provided for the sines and transients spectrograms (due to the component's nature, there is not much sense in looking for artifacts in noise). Energy bins showing local maxima below a certain threshold (that can be set in the Control panel) are marked in red in the spectrogram. In this way, a visual evaluation, combined with the audio playback, can be performed: intuitively, the denser a cluster of flagged bins is, the higher is the chance that an artifact is audible. The simplicity of this detection method may cause relevant content to be visually classified as an artifact if the threshold is ill-chosen, and that is why the use of the audio cues remains necessary.

3.6.3. Evaluation

In the last tab, some comprehensive statistics are displayed. In particular, each component's energy percentage is compared with the total amount of energy for all the flagged bins. Also, the number of flagged bins in each spectrogram per time frame is plotted. Again, intuitively, a tall peak in the number of flagged bins in a single time frame suggests the presence of a strong artifact.

4. APPLICATIONS

In this section, a few relevant cases of use for the proposed app are discussed. SiTraNo is particularly useful for educational purposes, as the visualization of the STN decomposition process greatly simplifies its understanding, when paired with the instantaneous audible cues. SiTraNo also proves useful to perform dynamic range



(a) S: 100%, T: 100%, N: 100%

(b) S: 100%, T: 50%, N: 100%

(c) S: 100%, T: 0%, N: 20%

Figure 6: *Dynamic range reduction of the cast-viol excerpt, achieved by acting on the transient and noise sliders.*

compression and to compare different separation methods against each other. Provided examples are discussed below and in a video tutorial available at the companion webpage [28].

4.1. Dynamic Range Reduction

In the context of audio mixing, it is often useful to pre-process sounds by applying a compressor to reduce the signal dynamics: this enables a great set of operation, e.g. making quieter parts of music audible in the mix, boosting the perceived loudness or de-essing [30]. Often, the dynamics of a sound are defined by impulsive or high-energy percussive events. A straightforward yet effective way of taking advantage of the mixing capabilities of SiTraNo is to decrease the amount of transients in the sounds while leaving sines and noise untouched. An example of this application is shown in Fig. 6. A mixture of castanets and violins (*cast-viol*) is imported in SiTraNo and STN decomposition is applied using HPR. As shown in Fig. 6a, the castanet events are occupying the entirety of the signal dynamics. By acting on the transient and noise sliders, it is possible to see how the magnitude of the impulsive events can arbitrarily reduced (Fig. 6b) to the point of being completely suppressed. (Fig. 6c). Audio examples for the different mixes described above available on the companion webpage [28].

4.2. Comparison of Decomposition Methods

A single-ended evaluation metric for STN separation has not been proposed yet, while double-ended evaluation metrics, such as SDR, SIR, and SAR cannot be applied unless the original sources that compose the three spectral classes are known, as discussed in Section 1. SiTraNo allows for a visual and audible comparison of the output of different separation methods by matching the visual information provided by the spectrograms and the flagged bins with the audible cues. In Fig. 7, a full comparison between single-round HPR, double-round HPR, ST, and Fuzzy is reported for audio excerpt *cast-viol*. A window size $M = 2048$ samples and artifact thresholds $T_s = T_t = -85$ dB have been used for this example.

Sinusoidal spectrograms show a common pattern of artifacts and interference appearing towards high frequencies, which seems to be more sparse for single-round HPR (Fig. 7a) and double-round HPR (Fig. 7c), while denser clusters of flagged bins emerge in ST (Fig. 7e) and Fuzzy (Fig. 7g). Looking at transient spectrograms, artifacts and interference are very noticeable in single-round HPR (Fig. 7b) and Fuzzy (Fig. 7h) for low frequencies. Double-round HPR (Fig. 7d) returns a better separation, although spurious energy is still detected between the impulsive events. This energy is mostly musical noise coming from the spectral subtraction process [31]. ST (Fig. 7f) provides the best isolation of the transients, but heavy energy dropouts can be spotted in low and mid-low frequencies, which contribute to a partial loss of bass presence and prominence [32]. Audio excerpts and video tutorials are available in the companion page [28].

5. CONCLUSIONS

In this work, a MATLAB app with GUI for decomposition of audio signals into sines, transients, and noise has been proposed. The implementation gives control over the amount of each spectral component to be mixed into the output signal and allows to select several among the most recent sines-transients-noise separation methods. Additionally, users have access to spectral representations of the signal to investigate the behavior of individual components and the quality of the separation. A simple visualization overlay highlights the presence of artefacts in each spectrogram. Examples of relevant uses of this app, such as dynamic range reduction and comparison of decomposition quality for multiple separation methods, have been reported. SiTraNo is available on the companion website [28] and on its GitHub repository [29].

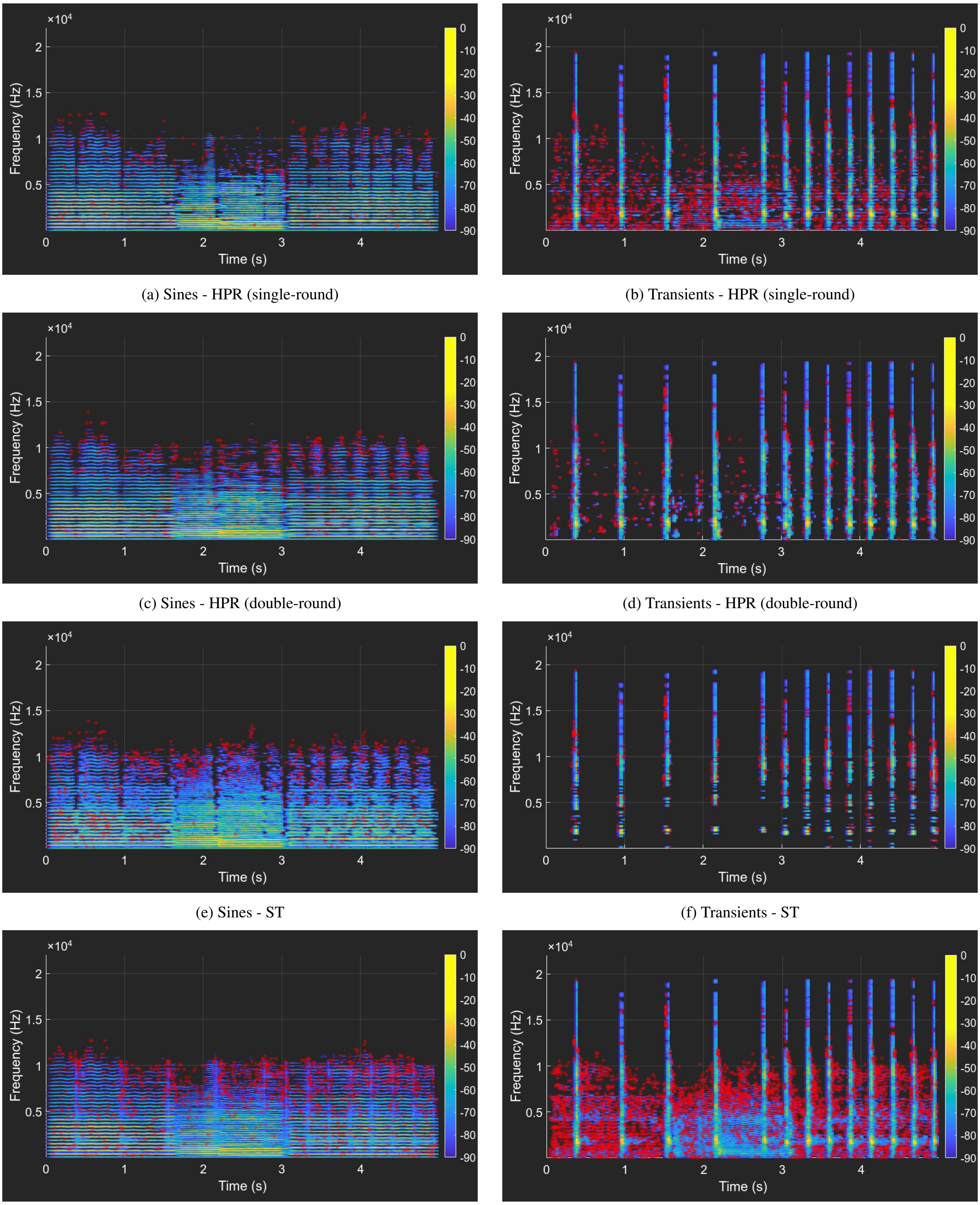
In future releases, the authors would like to include standard parameters control and more separation methods, to improve the artifacts detection and optimize the code implementation. A parallel version of SiTraNo as a real-time VST plugin is also under consideration, as it would enable to extend the evaluation to CPU usage and computational cost for each method, and to enable a real-time one-in three-out implementation for creative use.

6. ACKNOWLEDGMENTS

The work of Leonardo Fierro was funded by the Aalto ELEC Doctoral School. This work belongs to the activities of the “Nordic Sound and Music Computing Network—NordicSMC”, NordForsk project number 86892.

7. REFERENCES

[1] E.-P. Damskägg and V. Välimäki, “Audio time stretching using fuzzy classification of spectral bins,” *Appl. Sci.*, vol. 7, no. 12, pp. 1293, Dec. 2017.



(a) Sines - HPR (single-round)

(b) Transients - HPR (single-round)

(c) Sines - HPR (double-round)

(d) Transients - HPR (double-round)

(e) Sines - ST

(f) Transients - ST

(g) Sines - Fuzzy

(h) Transients - Fuzzy

Figure 7: *Parallel comparison of the single-round HPR, double-round HPR, ST, and Fuzzy methods using SiTraNo, with sinusoidal spectrograms in the left column and transients spectrograms in the right column. Artifacts are highlighted in red.*

[2] E. Moliner, J. Rämö, and V. Välimäki, "Virtual bass system with fuzzy separation of tones and transients," in *Proc. Digital Audio Effects (DAFx)*, Vienna, Austria, Sept. 2020.

[3] L. Fierro and V. Välimäki, "Towards objective evaluation of audio time-scale modification methods," in *Proc. Sound and Music Computing Conf. (SMC)*, Torino, Italy, June 2020, pp. 457–462.

[4] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stajylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 421–424.

[5] Á. Faraldo Pérez, *Tonality Estimation in Electronic Dance Music: A Computational and Musically Informed Examination*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain, Mar. 2018.

[6] B. Lentz, A. Nagathil, J. Gauer, and R. Martin, "Harmonic/percussive sound separation and spectral complexity reduction of music signals for cochlear implant listeners," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 8713–8717.

[7] C. Dittmar, *Source Separation and Restoration of Drum Sounds in Music Recordings*, Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), June 2018.

[8] J. Driedger, M. Müller, and S. Ewert, "Improving time-scale modification of music signals using harmonic-percussive separation," *IEEE Signal Process. Letters*, vol. 21, no. 1, pp. 105–109, 2013.

[9] T. Roberts and K. K. Paliwal, "Time-scale modification using fuzzy epoch-synchronous overlap-add (FESOLA)," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoustics (WASPAA)*, New Paltz, NY, USA, Oct. 2019, pp. 31–34.

[10] T. S. Verma and T. Meng, "An analysis/synthesis tool for transient signals that allows a flexible sines+transients+noise model for audio," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Seattle, WA, USA, May 1998, vol. 6, pp. 3573–3576.

[11] S. N. Levine and J. O. Smith III, "A sines+ transients+ noise audio representation for data compression and time/pitch scale modifications," in *Proc. Audio Eng. Soc. 105th Conv.*, San Francisco, CA, USA, Sept. 1998.

[12] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. Digital Audio Effects (DAFx)*, Graz, Austria, Sept. 2010, vol. 13.

[13] J. Driedger, M. Müller, and S. Disch, "Extending harmonic-percussive separation of audio signals," in *Proc. ISMIR*, Taipei, Taiwan, Oct. 2014, pp. 611–616.

[14] R. Füg, A. Niedermeier, J. Driedger, S. Disch, and M. Müller, "Harmonic-percussive-residual sound separation using the structure tensor on spectrograms," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Shanghai, China, Mar. 2016, pp. 445–449.

[15] D. Fitzgerald, A. Liukus, Z. Rafii, B. Pardo, and L. Daudet, "Harmonic/percussive separation using kernel additive modelling," in *Proc. IET Irish Signals Syst. Conf. (ISSC)*, Limerick, Ireland, June 2014.

[16] F. J. Canadas-Quesada, D. Fitzgerald, P. Vera-Candeas, and N. Ruiz-Reyes, "Harmonic-percussive sound separation using rhythmic information from non-negative matrix factorization in single-channel music recordings," in *Proc. Digital Audio Effects (DAFx)*, Sept. 2017, pp. 276–282.

[17] J. Neri and P. Depalle, "Fast partial tracking of audio with real-time capability through linear programming," in *Proc. Digital Audio Effects (DAFx)*, Aveiro, Portugal, Sept. 2018.

[18] Y. Masuyama, K. Yatabe, and Y. Oikawa, "Phase-aware harmonic/percussive source separation via convex optimization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, May 2019.

[19] K. Drossos, P. Magron, Stylianos I. Mimilakis, and T. Virtanen, "Harmonic-percussive source separation with deep neural networks and phase recovery," in *Int. Workshop Acoust. Signal Enhancement (IWAENC)*, Sept. 2018, pp. 421–425.

[20] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, July 2006.

[21] B. McFee, D. Raffel, C.and Liang, D. PW. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python in Science Conf.*, July 2015, vol. 8, pp. 18–25.

[22] "*Zen* GitHub repository," `https://github.com/sevagh/Zen`, Accessed: 24-05-2021.

[23] J. Driedger and M. Müller, "TSM Toolbox: MATLAB Implementations of Time-Scale Modification Algorithms," in *Proc. Digital Audio Effects (DAFx)*, Sept. 2014, pp. 249–256.

[24] H. Tachibana, N. Ono, and S. Sagayama, "Singing voice enhancement in monaural music signals based on two-stage harmonic/percussive sound separation on multiple resolution spectrograms," *IEEE Trans. Audio Speech Lang. Process.*, vol. 22, no. 1, pp. 228–237, Oct. 2013.

[25] MathWorks Inc., "MATLAB App Designer Manual," `https://www.mathworks.com/help/matlab/app-designer.html`, Accessed: 09.04.2021.

[26] MathWorks Inc., "MATLAB Image Processing Toolbox," `https://www.mathworks.com/help/images/`, Accessed: 09.04.2021.

[27] MathWorks Inc., "MATLAB Audio Toolbox," `https://www.mathworks.com/help/audio/`, Accessed: 09.04.2021.

[28] L. Fierro and V. Välimäki, "*SiTraNo* Companion Web Page," `http://research.spa.aalto.fi/publications/papers/dafx21-STN/`, Accessed: 12-04-2021.

[29] L. Fierro, "*SiTraNo* GitHub repository," `https://github.com/himynameisfuego/SiTraNo/releases/latest`, Accessed: 09-04-2021.

[30] U. Zölzer (Ed.), *DAFX - Digital Audio Effects*, John Wiley & Sons, 2011.

[31] O. Cappé, "Elimination of the musical noise phenomenon with the Ephraim and Malah noise suppressor," *IEEE Trans. Speech Audio Proc.*, vol. 2, no. 2, pp. 345–349, Apr. 1994.

[32] B. Katz, *Mastering Audio: The Art and The Science*, Butterworth-Heinemann, 2003.