VPS VIBRATIONS IN PHYSICAL SYSTEMS

DOI: 10.21008/j.0860-6897.2024.1.03
Vibrations in Physical Systems, 2024, 35(1), 2024103

Refining consonance sensation through spectral component control in additive resynthesis

Marek PLUTA

AGH University of Krakow, Department of Mechanics and Vibroacoustics, al. Mickiewicza 30, 30-059 Krakow, Poland

Corresponding author: Marek PLUTA, email: pluta@agh.edu.pl

Abstract A previous study explored the potential to influence the perception of consonance by controlling the spectral components of simultaneous pitches within a chord in sound synthesis using the additive method. This approach involves considering spectral components across in all simultaneously sounding pitches constituting a chord. Components within the range of beatings and roughness are adjusted to gradually enhance or weaken both phenomena. So far, the idea and method were implemented in a basic additive synthesizer, producing simple, abstract, time-invariant timbres. The current study extends this research and addresses challenges associated with implementing the aforementioned consonance-altering mechanism in a more advanced additive synthesizer, applied to the task of resynthesizing the sound of selected acoustic instruments. This study deals with natural, independent evolution of spectral components, wherein consonance undergoes gradual variations over time. Additionally, it presents a solution to mitigate the impact of the mechanism on the overall signal level, arising from the attenuation of selected spectral components.

Keywords: consonance, dissonance, sound timbre, additive synthesis, resynthesis.

1. Introduction

Compared to acoustic instruments, sound synthesizers provide better-defined means to control timbre-related sound qualities. It may be the control over parameters of partials and their envelopes in additive synthesis, control over filter parameters and their evolution in subtractive synthesis, but also an introduction of modulation and other effects [1]. All of these can constitute timbre of individual sounds or notes. There is, however, a quality that emerges only in simultaneous multi-pitch constructions, such as harmonic intervals or chords: the consonance, and its opposite – the dissonance. Due to its nature and its basis in direct relations between frequencies of component intervals, it is commonly attributed not to individual sounds, but to vertical musical structures, and as such it is omitted from timbre controls in synthesizers.

Recent study [2, 3] suggests a possibility to include the consonance as a controlled effect in additive synthesis. It is based on two observations. Firstly, that frequency ratios between partials of sounds produced by acoustic instruments usually diverge slightly from natural numbers [4, 5], and secondly, that due to complex problem of commas musical systems employ intervals similar, but usually not equal to natural ratios [6, 7]. Perfect intervals, i.e. the strongest consonances, need to have natural frequency ratios between their components, otherwise their higher partials are not aligned, introducing a phenomenon of beating or roughness, which is characteristic quality of a dissonance [8-9]. Thus even in intervals considered as consonances there are traces of dissonance sensation. This sensation is stronger in imperfect consonances, and becomes prominent in intervals classified as dissonances. Gradation of dissonance has been a subject of discussion and study [10–13]. It allowed to propose a solution to control the sensation of dissonance in any chord by attenuating partials that would cause beating or roughness due to proximity of their frequencies.

Proposed mechanism has been studied in a very basic additive synthesizer, and proved to be a viable possibility [2, 3]. There is an audible effect of attenuation of selected partials on the overall sensation of consonance, and it can be gradually controlled over a sounding chord, making dissonances sound more consonant, or almost perfect consonances sound like perfect consonances. However, simplicity of the synthesizer used in prior study might have helped the effect to become audible: there was a set of 12

distinct, equidistant partials constituting a single pitched sound, and apart from common short (10 ms) fade-in and fade-out, partials did not change over time – neither their amplitude, nor frequency.

The current study carries out the research further, into the area of sounds with natural characteristics. It discusses problems of implementing aforementioned consonance-altering mechanism in a complex additive synthesizer applied to faithfully resynthesize sound of selected acoustic instruments. The study deals with natural evolution of spectral components, where amount of dissonance-causing effects varies with time. It presents solutions to several issues, such as large number of partials, many of which fall within a range of beating and roughness, or mitigation of impact of the mechanism on a total signal level when selected components are attenuated. Unlike the previous implementation, which was a hybrid of I/O patch programmed in graphical environment PureData [14] and a computational part programmed in C language, both communicating through a network protocol, current solution utilises a more practical approach. It is programmed in C++ language as a Virtual Studio Technology (VST) plug-in for digital audio workstation (DAW) programs, and implements all of currently planned features. This allows to evaluate its computational performance and possibility to use the effect in real time with complex, multiple-partial, multiple-envelope sounds.

2. Concept and principle of operation

Sensation of dissonance evoked by an interval or chord of several simultaneous pitched sounds is related to presence of specific pairs of components [2, 3, 10–12]: if any two of sufficiently strong partials constituting such musical structure differ in frequency by a small amount, it produces an amplitude-fluctuation effect. Very small differences produce slow fluctuations that are not perceived as a dissonance, but slightly larger differences of about 10 Hz or more produce faster beating, and even larger differences of about 20-30 Hz or more are perceived as roughness [15]. Both of the latter are connected with sensation of dissonance and can be used to control its degree.

The effect works by the way of finding and attenuating sine components of multi-pitch complexes that would produce beating or roughness. The depth of attenuation is controlled by a user. Without attenuation the chord has its original, most dissonant character. Introduction of attenuation weakens effects that cause the sensation of dissonance, thus changing character of the chord to more consonant, without changing pitch of its notes.

This kind of control would be impractical or unfeasible in case of sound recordings or in any method of sound synthesis with rough spectral control, such as subtractive, FM, or sampling synthesis. Additive synthesis, on the other hand, provides the most precise control over individual sound partials [1], and is therefore an ideal basis for implementation of consonance-dissonance control effect: it produces a signal by mixing a number of sine components. These components can be controlled in terms of frequency and amplitude either as simple parameters, or using envelopes (Fig. 1).

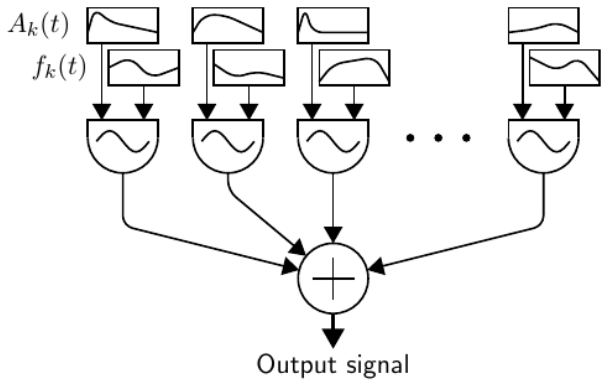

Figure 1. Additive synthesis method – a general diagram with individual amplitude and frequency envelopes $A_k(t)$, $f_k(t)$ for each sine component, after [1].

Moreover, additive synthesis is one of few methods that allow to resynthesize an existing sound [1], i.e. there are methods of analysis of sound recordings that would produce a complete set of parameters required to synthesize a faithful counterpart with possibility to control its selected characteristics (Fig. 2). Thus the additive method not only allows to implement the effect of consonance-dissonance control, but also to test it on real instrument sounds, which is the aim of current study. In the prior work [2, 3] the synthesizer produced very simple, artificial sounds.

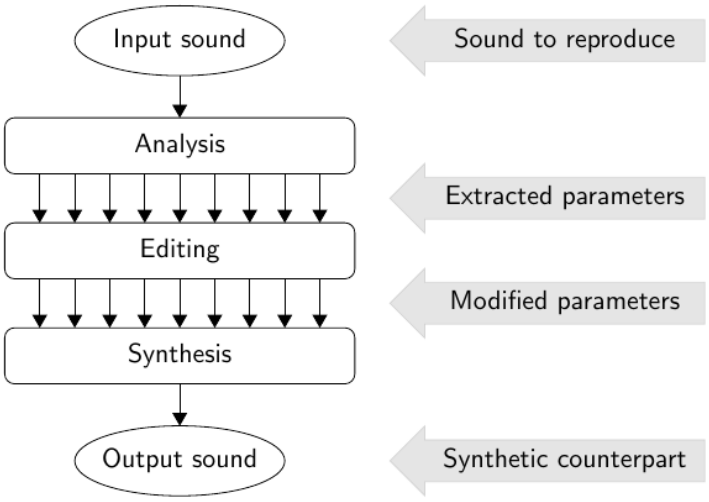

Figure 2. Principle of resynthesis, after [1].

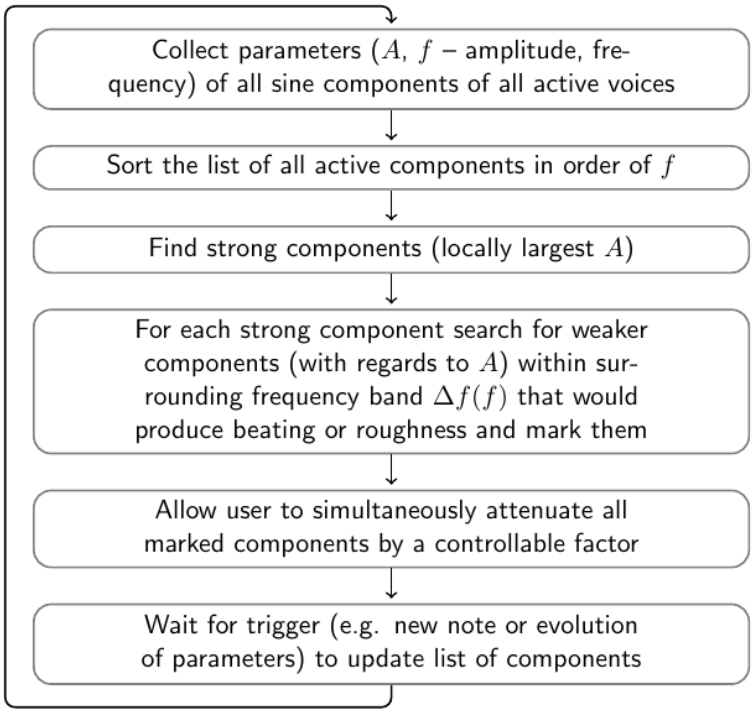

Figure 3. Operating principle of the effect of consonance-dissonance control.

The principle of operation upon which the effect is based, when applied in additive synthesizer, is presented in Fig. 3. The first stage of the algorithm is to extract current frequency and amplitude values of all sine components in all active synthesizer voices, and store them together along with their voice number and original partial number within a voice. Next, the list is sorted in order of increasing frequency, so the components of different voices will be mixed, but their stored original partial and voice number will allow to identify them. In the following stage the list is searched for components stronger, i.e. having larger amplitude, than their surroundings. The extent of the surrounding depends on frequency, as it is calculated in relation to Equivalent Rectangular Bandwidth auditory filter (ERB) [2, 3]. The surrounding is then searched for any weaker components that would – introducing amplitude modulation with the strongest component – produce beating or roughness effect. These components are marked for attenuation. Having the components marked, the algorithm checks user's setting of the depth of attenuation. In practice it can be mapped to a MIDI modulator wheel or a similar real-time controller to allow to use it as a means of expression, or to some automatic controller, such as envelope generator (EG) or low frequency oscillator (LFO), to make the sound evolve in terms of dissonance-consonance feeling. All marked components are

attenuated simultaneously. The process is repeated. In case of simple additive synthesizer with constant sound (no frequency or amplitude envelopes), as presented in [2, 3], it is enough to repeat it on each note-on and note-off MIDI event. But if the sound evolves, i.e. frequencies and amplitudes change according to some envelopes, which is the case of synthesizer variant presented in this manuscript, it needs to be regularly updated even if the chord is constantly kept on.

3. Analysis stage of the resynthesis

For the sake of clarity, prior study [2, 3] discussed the effect using a very simplified additive synthesizer with 12 constant partials, without any amplitude or frequency evolution. While it did allow to observe and perceive the effect itself, it produced artificial sound, without many features characteristic for real instruments. The current study addresses this issue using resynthesis (Fig. 2) on the basis of recordings of selected acoustic instruments. For this purpose the synthesizer has been rewritten and significantly extended to handle virtually unlimited number of partials, each one controlled with two envelopes – for frequency and amplitude.

Several signal analysis techniques may be applied to obtain required parameters for the additive synthesizer. The most basic are channel or phase vocoders [16–19], but McAulay-Quatieri algorithm (MQ) [19, 20] is able to provide more precise data, therefore it has been applied in current study. It estimates amplitude and frequency for each signal component in subsequent locations in time determined by short-time Fourier transform (STFT) hop size. Component parameters are estimated from FFT peaks searched for within adjustable surroundings. Peak continuation algorithm produces parameter trajectories that will be used as envelopes in the synthesizer. Component durations can be independent. The analysis has been carried out using the SPEAR program [21]. Fig. 4 presents graphical result of analysis for a bassoon note D4 (actual data used in current study). A frequency band is limited for the clarity of presentation, although for the purpose of resynthesis a full band has been utilised. Frequency trajectories are represented by lines, while the line intensity represents the amplitude.

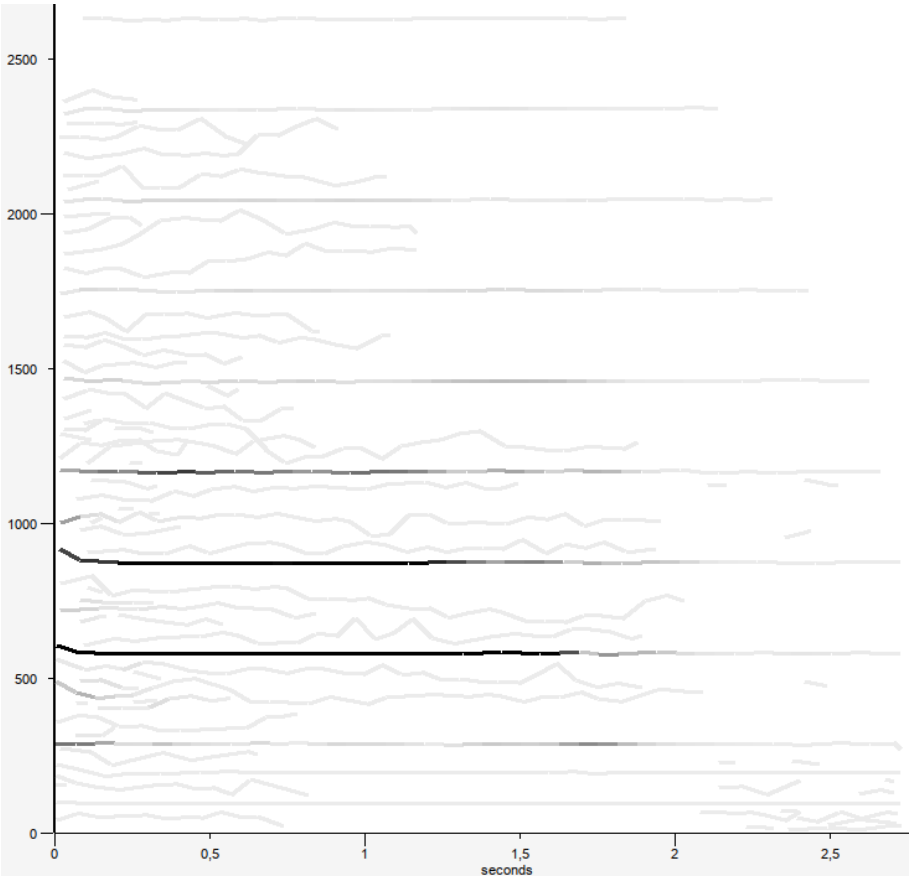

Figure 4. MQ analysis results (frequency band limited for clarity of presentation) obtained with SPEAR program [21]; vertical axis – frequency [Hz], line intensity indicates amplitude.

The results of the analysis are stored as partials data. Each partial is represented by an ordered set of triplets representing subsequent values of time, frequency, and amplitude. Time positions are common for all partials, although not all of them are present in each partial – individual partials can start later or end earlier, as can be seen in Fig. 4. In this particular case of a bassoon note lasting 2.72 s the algorithm produced 235 partials using default settings.

4. Implementation and improvements

The prior works [2, 3] demonstrated the effect on the basis of a hybrid implementation: I/O patch programmed in PureData and computational part programmed in C, both communicating through a network protocol. In order to efficiently handle much larger amount of data from MQ analysis and to allow operation under the supervision of digital audio workstations, current solution is designed as a VST plug-in and programmed entirely in C++.

The first limitation of the original implementation was a small, fixed number of partials (12). Current implementation utilises dynamic C++ data structures to allow to create necessary number of partials on runtime, depending on data in MQ analysis files. The limitation on number of partials is therefore only a matter of processing power, and it does not affect the solution with hundreds of partials, as tested with current data files. For the sake of computational efficiency sine oscillators are implemented as wavetable generators [1].

Significantly larger amount of partials required an improved algorithm for marking weaker partials to attenuate. Previously, with only 12 distant partials (harmonics) per voice, the resulting spectrum was sparse, therefore it was entirely sufficient to perform a single browse through frequency-ordered partials and compare adjacent pairs, marking weaker in pair to attenuate. MQ however, produces much denser spectra (Fig. 4), which required a different approach. New algorithm is presented in Fig. 5.

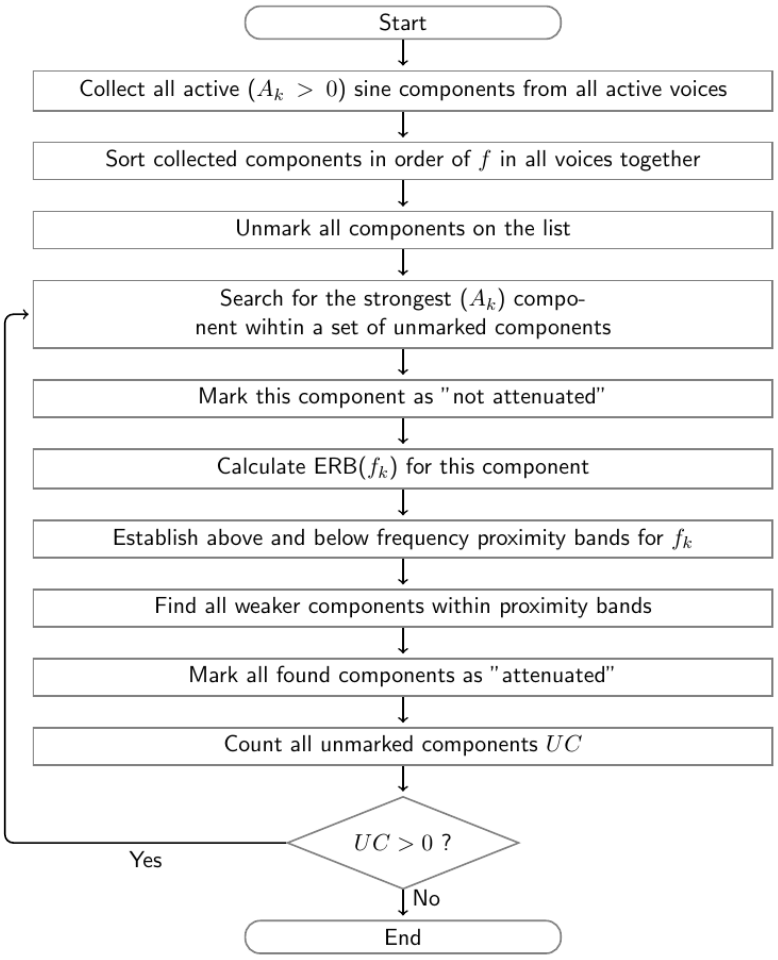

Figure 5. Algorithm marking weaker partials to be attenuated in proximity of stronger ones to avoid beating or roughness.

The idea is to search for the strongest component, and then mark weaker components in its proximity to attenuate. Next, to search for the strongest among the remaining ones, mark its surrounding for attenuation and repeat this procedure until all components are marked either as "not attenuated", or "attenuated". Proximity of a strong component is established on the basis of observations discussed in prior works [2, 3], in particular:

- slow fluctuations of amplitude do not affect sensation of dissonance, therefore partials above and below the strongest one, but closer than a fixed frequency limit f_L (in Hz) remain unattenuated; the limit is user-defined, initially 11 Hz,
- very fast fluctuations do not cause roughness, but produce a sensation of new pitch (differential tone), therefore partials more distant than a certain value f_H are not attenuated; this limit is related to ERB value, so it depends on the centre frequency f_C (frequency of the currently processed strongest partial); it is also user-defined and initially set to a quarter of ERB,
- partials in two areas above and below the currently processed strongest partial falling between two limits mentioned above ($[f_C\text{-}f_H, f_C\text{-}f_L]$ and $[f_C\text{+}f_L, f_C\text{+}f_H]$) produce either distinct beating or roughness, and can be attenuated to affect sensation of dissonance.

The original implementation generated fixed partials, with no amplitude or frequency envelopes: frequencies were natural multiplies of a fundamental frequency (harmonics), and amplitudes were set to user-defined values that did not change during a note. With resynthesis, both, frequencies and amplitudes, need to be controlled with individual envelopes according to data obtained from MQ analysis (as in Fig. 4). Envelopes of all partials have common break-points (nodal points for which A_k and f_k values are provided). Values between break-points are calculated using linear interpolation. Distance between breakpoints depends on parameters of MQ analysis, but in case of data in Fig. 4 it is approximately 12.5 ms, which translates to 600 signal samples with sampling rate of 48 kHz. Once a subsequent break-point is reached new ΔA_k and Δf_k are calculated for every partial to update (interpolate) momentary values of A_k and f_k on a sample-basis in the following envelope segment. It has to be noted that break-point distance is independent from another time-related value, i.e. processing buffer size. In audio plugins signal is processed or generated in subsequent data chunks of equal size (buffers). These are set on the system or DAW level and affect the latency. Values can be either lower (e.g. 64 samples) or higher (e.g. 2048 samples) than break-point distance.

In the previous approach, without parameter envelopes, it was sufficient to mark partials for attenuation only on note-on and note-off MIDI events. With parameters controlled through envelopes beating and roughness situation changes during a single note, therefore more frequent updates are necessary. However, both dissonance-related sensations require time to emerge (actual values depend on several factors and are difficult to establish in case as general as a freely user-controlled synthesizer), so it is not necessary to perform updates on a sample-basis. It has been assumed that an update on every new sample buffer is sufficient: a buffer of 256 samples with 48 kHz sample rate lasts approximately 5.3 ms, which translates to almost 200 Hz buffer rate. Therefore even a relatively long 1024-sample buffer would keep only a single period of a very fast (roughness) fluctuation, and at least a few periods are required for a sensation to emerge. Thus, buffer-size update rate seems a reasonable trade-off.

Due to evolving values of partial amplitudes and frequencies, partials marked for attenuation would frequently switch on and off causing perceptible signal distortion. In order to avoid it, additional ramped envelopes are applied instead of binary switching. The additional envelope has a momentary value and target value. Target value is calculated using a binary state (attenuated or not attenuated) and attenuation depth (controlled by a user). Additional user-controlled parameter – ramp rate – increases or decreases momentary value on a sample-basis. Thus for instruments with a lot of internal details the rate can be set to quickly reach target value so the dissonance-control effect could be perceived, and for slow, smooth sounds it can be adjusted to eliminate possible discontinuities in slowly evolving partials. Initial value is set to reach target attenuation in 100 ms.

Attenuating larger numbers of partials will reduce output signal level. To avoid it a total amplitude of attenuated signals is calculated in relation to the original signal level and a correction is calculated. It is not applied immediately, but with ramped envelope analogous to those applied in attenuated partials, sharing their ramp rate.

In the end, a signal sample for a discrete moment of time t is calculated according to the following expression:

$$y(t) = \big(1 + D \cdot C(t)\big)\Big(\sum_{k=1}^{M} A_k\,(t)\big(1 - D \cdot a_k(t)\big) osc(f_k(t), t)\Big), \tag{1}$$

where $D \in [0, 1]$ is the user-controlled effect depth, $C(t)$ is the momentary value of global amplitude correction, k is the index of partial within a voice, M it the number of partials within a voice, $A_k(t)$ is the momentary value of partial amplitude envelope, $f_k(t)$ is the momentary value of partial frequency envelope, $a_k(t)$ is the momentary value of partial attenuation, and $osc(f_k(t), t)$ is the output of a single sine oscillator for a given frequency and time moment.

During performance user is controlling four parameters, as described in Table 1. All of them affect the signal in real time.

Table 1. User-controlled parameters.

Parameter	Symbol	Initial value	Range
Effect depth	D	0	0 – 1
Close proximity limit	f_L	11 Hz	1 – 60 Hz
Far proximity limit	f_H	0.25 ERB	0.1 – 3 ERB
Ramp rate	R	100 ms	5 – 300 ms

5. Results

The synthesizer has been tested with sounds of three acoustic instruments: bassoon, flute, and oboe (recorded directly from real instruments). In case of bassoon MQ analysis produced 219 break-points and 236 partials. For flute it was 325 break-points and 122 partials, and for oboe – 229 break-points and 160 partials. All instruments worked without any problems that could indicate lack of processing power, such as missed buffers, and did not show any distortion with ramp rate set to 100 ms.

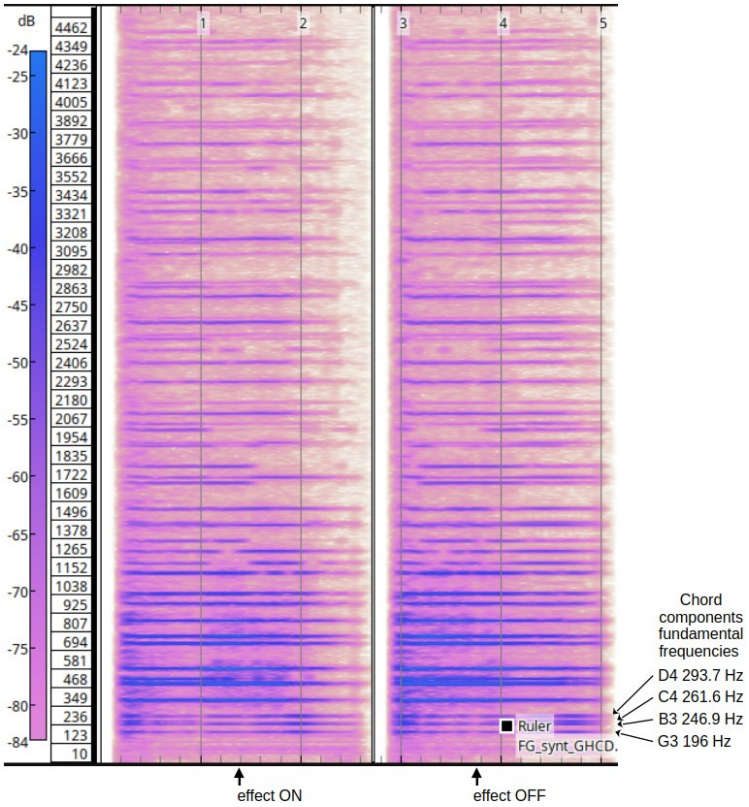

Figure 6. Spectrogram of the effect applied to synthesized bassoon (X-axis – time [s], Y-axis – frequency [Hz]); in the first chord the effect is turned on in the middle, in the second – it is turned off in the middle.

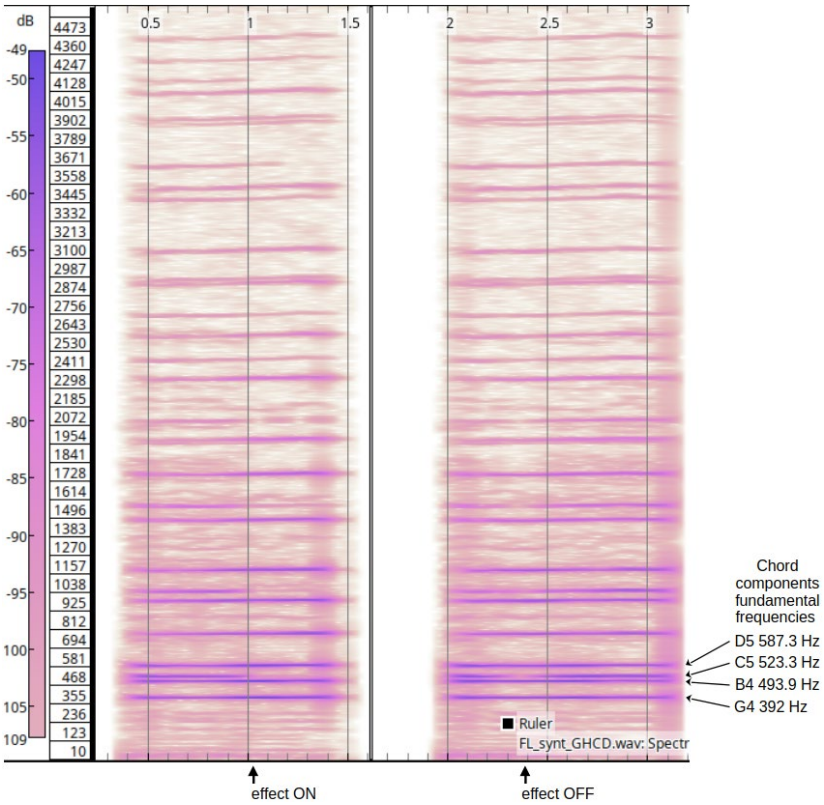

Figure 7. Spectrogram of the effect applied to synthesized flute (X-axis – time [s], Y-axis – frequency [Hz]); in the first chord the effect is turned on in the middle, in the second – it is turned off in the middle.

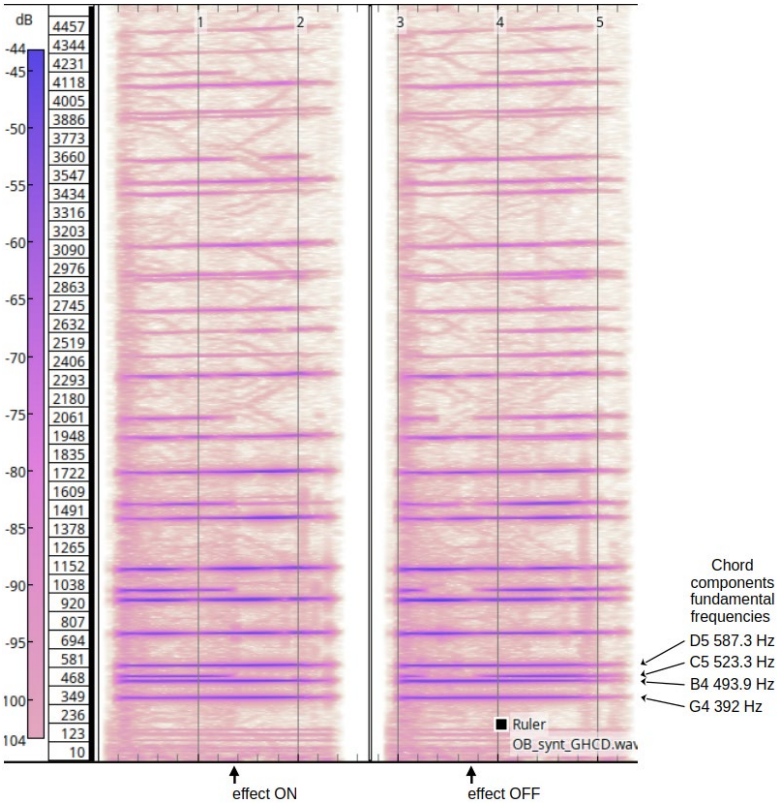

Figure 8. Spectrogram of the effect applied to synthesized oboe (X-axis – time [s], Y-axis – frequency [Hz]); in the first chord the effect is turned on in the middle, in the second – it is turned off in the middle.

Even though signal variability characteristic for natural instruments and represented by partials envelopes may – to some degree – mask the consonance-dissonance control effect, it was still clearly audible, and could be easily boosted by adjusting parameters, particularly proximity limits f_L and f_H. Moreover, audibility strongly depends on the chord played – more dissonant ones allow to introduce larger contrast, as the effect essentially reduces the amount of dissonance sensation, which needs to be present in the original sound. The effect did not introduce audible distortion. Therefore the effect is applicable not only to simple, artificial sounds, as shown in previous works, but can be successfully utilised with other musical sounds as well.

Figures 6–8 show the results of synthesizer operation for above mentioned instruments (frequency band is limited for visual clarity). All recordings consist of a repeated chord (notes G3, B3, C4, D4 for bassoon, and G4, B4, C5, D5 for flute and oboe) played manually using MIDI keyboard with effect depth controlled by modulator wheel. Ramp rate and f_L were at default initial values, but f_H has been increased to 0.5 ERB for the effect to be more pronounced in the charts. The first chord starts without the effect, but in the middle of sound the wheel turns the effect to the maximum depth. Contrary, the second chord starts with maximum effect depth, which in the middle is turned to zero. The affected partials are clearly visible.

6. Conclusions

While the prior study [2, 3] introduced the concept of controlling consonance-dissonance balance within a chord in the additive synthesizer by attenuating weaker partials that would cause beating or roughness sensation, it was not clear whether the same mechanism can be applied to produce practical effect for more complex sounds, like ones generated by acoustic instruments. The current paper addressed this issue by presenting much more advanced variant of synthesizer, able to reproduce fine details characteristic for acoustic instruments.

New synthesizer is able to faithfully resynthesize any recorded sound, with hundreds of partials, each partial with own amplitude and frequency envelopes. The effect has been refined to accommodate evolving spectra, and the whole synthesizer has been rewritten as a VST plugin. Resynthesis has been studied for three instruments: bassoon, flute, and oboe. In every studied case synthesizer was able to control the consonance-dissonance balance producing easily perceivable result without introducing audible distortions, and without problems on the side of computational efficiency.

Further study shall include auditory tests as well as more thorough output signal analysis to establish usable ranges and mapping curves for parameter values with regards to type of synthesized instrument or sound.

The procedure applied to refine the sensation of consonance might be elaborate, but the underlying perceptual phenomenon has even more nuances that exceed the scope of this manuscript. There is vast research within the area, and recent findings might in future be applied to model the sensation of consonance and dissonance with better understanding of complex interactions and subtle effects.

Acknowledgments

The article was published as part of the research subsidy 16.16.130.942 of the Department of Mechanics and Vibroacoustics AGH-UST Krakow.

Additional information

The author(s) declare: no competing financial interests and that all material taken from other sources (including their own published works) is clearly cited and that appropriate permits are obtained.

References

1. M. Pluta; Sound synthesis for music reproduction and performance; Wydawnictwa AGH, Krakow, Poland, 2019
2. M. Pluta; Influence on chord consonance by controlling spectral components of its component pitches in additive sound synthesis (in Polish); Proceedings of the XXV Konferencja Inżynierii Akustycznej i Biomedycznej, Kraków-Zakopane, March 28–31, 2023
3. M. Pluta; Impact on the sensation of consonance by controlling spectral components of simultaneous pitches consisting a chord in sound synthesis using the additive method; Vibrations in Physical Systems, 2023, 34(2), 2023209; DOI: 10.21008/j.0860-6897.2023.2.09
4. A.H. Benade; Fundamentals of Musical Acoustics; Dover Publications, 1990

5. G. Wieliczko; The study of harmonics of sound's components produced by a string of an electric guitar (in Polish); Engineering thesis, AGH University of Science and Technology, Krakow, 2018
6. M. Toporowski, M. Pilch; Old temperaments. Basics of acoustics and practical implementation (in Polish); Wydawnictwo Akademii Muzycznej im. Karola Szymanowskiego w Katowicach, 2014
7. M. Pluta; Principles of music and musical notation (in Polish); Wydawnictwa AGH, Krakow, Poland, 2012
8. N. Di Stefano, P. Vuust, E. Brattico; Consonance and dissonance perception. A critical review of the historical sources, multidisciplinary findings, and main hypotheses; Physics of Life Reviews, 2022, 43, 273–304; DOI: 10.1016/j.plrev.2022.10.004
9. T. Pankovski, E. Pankovska; Emergence of the consonance pattern within synaptic weights of a neural network featuring Hebbian neuroplasticity; Biologically Inspired Cognitive Architectures, 2017, 22, 82–94; DOI: 10.1016/j.bica.2017.09.001
10. P.N. Vassilakis; Perceptual and Physical Properties of Amplitude Fluctuation and their Musical Significance; Doctoral Dissertation, Los Angeles: University of California, Los Angeles, 2001
11. P.N. Vassilakis; Auditory roughness as a means of musical expression; Selected Reports in Ethnomusicology, 2005, 12, 119–144
12. E. Terhardt; On the Perception of Periodic Sound Fluctuations (Roughness); Acustica, 1974, 30(4), 201–213
13. W. A. Sethares; Tuning, Timbre, Spectrum, Scale, 2nd ed.; Springer, London, 2005
14. M. Puckette; The Theory and Technique of Electronic Music; World Scientific, Singapore, 2007
15. E. Ozimek; Sound and its perception. Physical and psychoacoustical aspects (in Polish); Wydawnictwo Naukowe PWN, 2002
16. J.L. Flanagan, R. M. Golden; Phase Vocoder; Bell System Technical Journal, 1966, 45(9), 1493–1509
17. M.R. Portnoff; Implementation of the digital phase vocoder using the fast Fourier transform; IEEE Transactions on Acoustics, Speech, and Signal Processing, 1976, 24(3), 243–248
18. J.A. Moorer; The use of the phase vocoder in computer music applications; Journal of the Audio Engineering Society, 1978, 26(1/2), 42–45
19. M.K. Klingbeil; Spectral Analysis, Editing, and Resynthesis: Methods and Applications; PhD thesis, Columbia University, New York, NY, USA, 2009
20. R.J. McAulay, T.F. Quatieri; Speech analysis/synthesis based on a sinusoidal representation; IEEE Transactions on Acoustics Speech and Signal Processing, 1986, 34(4), 744–754
21. M. Klingbeil; Software for spectral analysis, editing, and synthesis; Proceedings of the 2005 International Computer Music Conference, ICMC 2005, Barcelona, Spain, September 4–10, 2005