STATISTICAL SINUSOIDAL MODELING FOR EXPRESSIVE SOUND SYNTHESIS

Henrik von Coler

Audio Communication Group
TU Berlin
Berlin, Germany
voncoler@tu-berlin.de

ABSTRACT

Statistical sinusoidal modeling represents a method for transferring a sample library of instrument sounds into a data base of sinusoidal parameters for the use in real time additive synthesis. Single sounds, capturing an instrument in combinations of pitch and intensity, are therefor segmented into attack, sustain and release. Partial amplitudes, frequencies and Bark band energies are calculated for all sounds and segments. For the sustain part, all partial and noise parameters are transformed to probabilistic distributions. Interpolated inverse transform sampling is introduced for generating parameter trajectories during synthesis in real time, allowing the creation of sounds located at pitches and intensities between the actual support points of the sample library. Evaluation is performed by qualitative analysis of the system response to sweeps of the control parameters pitch and intensity. Results for a set of violin samples demonstrate the ability of the approach to model dynamic timbre changes, which is crucial for the perceived quality of expressive sound synthesis.

1. INTRODUCTION

A system capable of expressive sound synthesis reacts to dynamic control input with the desired or appropriate changes in sound. In analysis-synthesis systems this means that the perceived timbral qualities of the synthesized sound emulate the behavior of the analyzed instrument as close as possible. Such systems thus need to capture the individual sound of an instrument and allow manipulations based on a limited set of control parameters. In order to achieve this, the synthesis approach presented in this paper, entitled *statistical sinusoidal modeling*, combines a sample based approach with a novel method for sinusoidal modeling.

Sample based synthesis in its basic form is able to capture individual sounds very accurately but does not offer manipulation techniques necessary for an expressive synthesis [1]. Sinusoidal modeling, on the other hand, is capable of wide-ranging means for sound manipulation. A key problem of sinusoidal modeling approaches, however, is the mapping of control parameters to the large amount of synthesis parameters. Statistical sinusoidal modeling can be considered a way of mapping the control parameters *pitch* and *intensity* to the parameters of a sinusoidal model. This reduced set of control parameters is often considered the central input for similar sound synthesis systems.

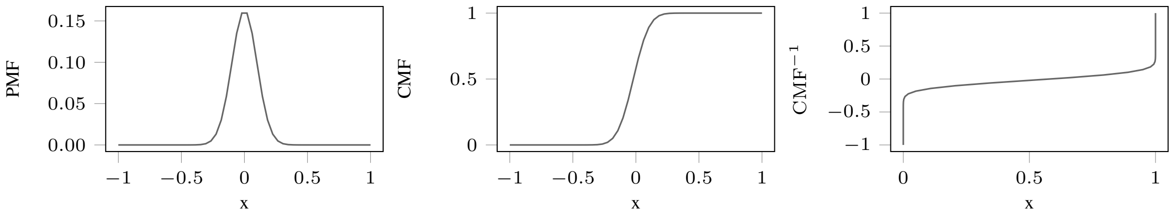
Different approaches aim at improving sample based sound synthesis. Among them are *granular synthesis* and *corpus based concatenative synthesis* [2]. Combined with spectral manipulation techniques, the flexibility of these approaches is further increased. Such combinations have proven to be effective for expressive sound synthesis. Examples include *spectral concatenative synthesis* [3] and *reconstructive phrase modeling* [4].

An *extended source-filter model* has been proposed by Hahn et al. [5, 6]. Partial and noise parameters are modeled in dependency of the control parameters pitch and intensity, by means of tensor-product B-splines. Two separate filters are used, one representing the instrument-specific features by partial index and another one capturing the frequency dependent partial characteristics. Wessel et al. [7] present a system for removing the temporal axis from the analysis data of sinusoidal models by the use of neural networks and memory-based machine learning. These methods are used to learn mappings of the three control parameters pitch, intensity and brightness to partial parameters. A system combining *corpus based concatenative synthesis* with *audio mosaicing* [8] has been proposed by Wager et al. [9]. This approach is able to synthesize an expressive target melody with arbitrary sound material by target-to-source mapping, using the features pitch, RMS energy and the modulus of the windowed Short- Time Fourier Transform.

A key feature in an expressive re-synthesis of many instruments, especially of bowed strings, are the so called spectral envelope modulations (SEM) [10]. The amplitude of each partial is modulated by its frequency in relation to the underlying frequency response of the instrument's resonant body. A vibrato in string instruments thus creates a periodic change in the relative partial amplitudes. At the typical vibrato frequencies of 5-9 Hz this effect is perceived as a timbral quality, rather than a rhythmical feature. This phenomenon, perceptually also referred to as *Sizzle* [11], contributes to the individual sound of instruments to a great extent. Using spectral modeling techniques for manipulations of instrument sounds, this effect is also considered essential for improving the quality [12]. Glissandi also result in spectral envelope modulations in the same way.

Another important effect for an expressive re-synthesis is the connection between intensity and the spectral features of the instrument's sound. Increases in intensity usually cause significant changes in the spectral distribution, respectively spectral skewness and spectral flatness [13], as well as in the tonal/noise ratio.

The proposed system is designed to encompass the above mentioned effects with simple means, enabling an efficient real time implementation. Details on the analysis, sinusoidal modeling and statistical modeling are presented in Section 2. Section 3 explains the statistical sinusoidal synthesis process in detail, followed by the evaluation of synthesis results in Section 4. The conclusion in Section 5 summarizes the findings and lists perspectives for further development.



(a) PMF

(b) CMF

(c) Inverted CMF

Figure 1: Exemplary probability mass function (PMF) with derived cumulative mass function (CMF) and inverted CMF.

2. ANALYSIS AND MODELING

2.1. Sample Library

The focus of the presented synthesis system rests on excitation-continuous melody instruments, using a violin as source material for the analysis stage. The *TU-Note Violin Sample Library*[14] is used for generating the statistical model. Featuring 336 single sounds and 344 two-note sequences, it has been specifically designed for this purpose. For the use in this project, the single sounds are reduced to a total of 204, consisting of 51 unique, equally spaced pitches, each captured at four dynamic levels. In the remainder, this two-dimensional space will be referred to as the *timbre plane*. It must be noted that, depending on the instrument, additional dimensions for timbre control need to be added to this space. For the violin, limited to standard techniques, the proposed reduction is acceptable. MIDI values from 0 to 127 are used to organize the dimensions pitch and velocity. Pitches range from the lowest violin note at MIDI 55 $\sim$ ***G3*** = 197.3341 Hz (443 tuning frequency) to the note at MIDI 105 $\sim$ ***A7*** = 3544 Hz. The dynamic levels ***pp***, ***mp***, ***mf*** and ***ff*** are captured in the *timbre plane*. The material has been recorded at 96 kHz with 24 bit resolution and can be downloaded using a static repository [15].

2.2. Sinusoidal Analysis

The *sinusoids+noise* model [16] is used for extracting the tonal and noise parameters for each single sound. Analysis and modeling is carried out offline, prior to the synthesis stage. Monophonic pitch tracking is performed using the YIN [17] and SWIPE [18] algorithms. Tests with more recent, real-time capable approaches [19] did not improve the performance. Based on the extracted fundamental frequency trajectories, partial trajectories are obtained by peak picking in the short-time Fourier transform (STFT), using a hop-size of 256 samples (2.76 ms) and a window size of 4096 samples, zero-padded to 8192 samples.

Quadratic interpolation (QIFFT), as presented by Smith et al. [20], is applied for estimating amplitude and frequency of up to 80 partials in each frame. The partial phases φ_i are obtained by finding the argument of the minimum when subtracting each partial with the individual amplitude a_i and frequency f_i from the complete frame x at different phases $\varphi*$:

$$\varphi_i = \arg\min \left[\sum_{n=1}^{L} \left(x(n) - a_i sin(2\pi f_i t(n) + \varphi^*) \right) \right], \qquad \varphi^* = -\pi \dots + \pi \tag{1}$$

After all partial parameters are extracted, the residual is created by subtracting the tonal part with original phases from the complete sound in the time domain. Modeling of the residual component is performed with a filter bank, based on the Bark scale, as proposed by Levine et al. [21]. The instantaneous energy trajectories of all bands are calculated using a sliding RMS with the hop-size of the sinusoidal analysis (2.76 ms) and a window length of 21.33 ms.

For each single sound, the analysis results in up to 80 partial amplitude trajectories, 80 partial frequency trajectories, 80 partial phase trajectories and 24 noise band energy trajectories. Since the original phases are not relevant for the proposed synthesis algorithm, they are not used for the further modeling steps.

2.3. Segmentation

The *TU-Note Violin Sample Library* includes manually annotated segmentation labels, based on a transient/steady state discrimination. For the single sounds they define the attack, sustain and release portion of each sound. Trajectories during attack and release segments are stored completely and additionally modeled as parametric linear and exponential trajectories. Details on the modeling and synthesis of attack and release segments are not subject to this paper. The sustain part is synthesized with the statistical sinusoidal modeling approach, explained in detail in the following section.

2.4. Statistical Modeling

After the segmentation, the above obtained trajectories of the partials and noise bands during the sustain portion of the sound are transformed into statistical distributions. Probability mass functions (PMF) with 50 equally spaced bins are created and transformed to cumulative mass functions (CMF):

$$CMF(x) = \sum_{x_i=0}^{x} PMF(x_i) \tag{2}$$

Inverse transform sampling relies on the inverted CMF for generating random number sequences with the given distribution. Figure 1 shows an exemplary PMF with the derived CMF and inverted CMF. For the synthesis algorithm, CMFs and their inversions are calculated and stored for all partial and noise trajectories during the sustain parts. $CMFs$ for the first five partials' amplitudes and frequencies are shown in Figure 2, respectively Figure 3. $CMFs$ for the first five noise band energies are shown in Figure 4. Additionally, the mean, median and standard deviation of all distributions are stored with the model.

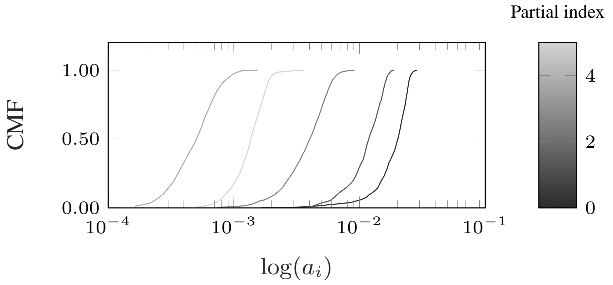


Figure 2: CMFs for the first five partial amplitudes.

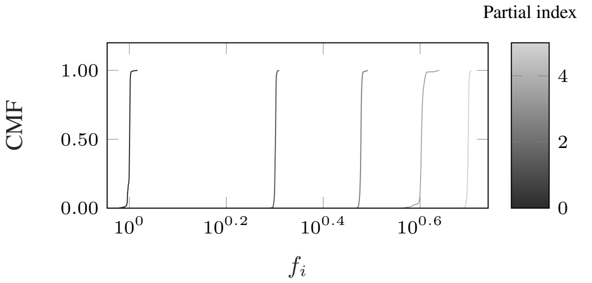


Figure 3: CMFs for the first five partial frequencies.

3. SYNTHESIS

3.1. Algorithm Overview

The implementation of the algorithm is included in a C^{++} based framework, using the JACK API [22]. Synthesis is performed in the time domain, with a non-overlapping approach and a frame size related to the buffer size of the audio interface. On the test system, a buffer size of 128 samples was used at a sampling rate of $f_s = 48\,\text{kHz}$, which allows a responsive use of the synthesizer. For generating a single sound, a maximum of 160 partial parameters and 24 bark band energies have to be generated each synthesis frame. The full number of 80 partials, however, is only synthesized for pitches below 600 Hz at a sampling rate of 96 kHz, respectively below 300 Hz at 48 kHz. Figure 5 shows the number of synthesized partials in dependency of sampling rate and fundamental frequency.

Listing 1: Pseudo code for the synthesis algorithm.

```
for each frame:

    get control inputs

    for all partials:
        generate random frequency value
        generate random amplitude value
        generate linear amplitude ramp
        synthesize sinusoid
        add sinusoid to output

    for all Bark bands:
        generate random band energy
        generate linear energy ramp
        apply band filter to noise signal
        add band signal to output
```

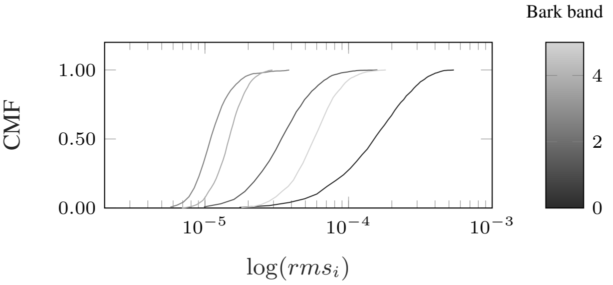


Figure 4: CMFs for the first five bark energy trajectories.

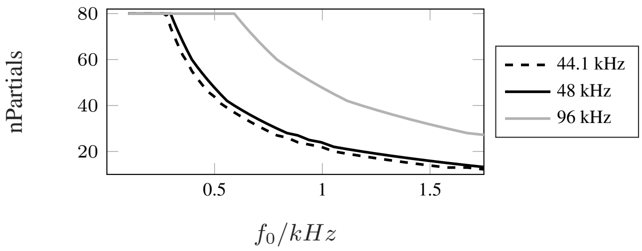


Figure 5: Number of partials synthesized, depending on sampling rate and fundamental frequency.

For each frame of the synthesis output, a new set of support points is generated, as shown in Listing 1. Interpolation trajectories are generated for the connection to the preceding values of partial amplitudes and noise band energies. Partial frequencies are piece-wise constant.

3.2. Statistical Value Generation

The statistical sinusoidal synthesis offers two different modes for generating parameter trajectories. For the three synthesis parameter types (partial amplitude, partial frequency and noise band energy) the mode can be selected, individually.

3.2.1. Mean/Median Mode

In the *mean/median mode*, the individual distribution functions are not used. Support points of the parameter trajectories are generated using the mean or median values stored in the model. For a constant control input, the resulting parameter trajectories remain constant, too. Variations in parameters are thus induced only through modulations of the input parameters.

3.2.2. Inverse Transform Sampling

Inverse transform sampling is a method for generating random number sequences with desired distributions from uniformly distributed random processes [23]. The inverted CMF, as shown in Figure 1c, maps the uniform distribution $\boldsymbol{\mathcal{U}}(0,1)$ to the target distribution. The method can be implemented using a sequential search method [24, p. 85], without actually inverting the distribution functions in advance. For a random value $0 \leq r \leq 1$ from the uniform distribution, the corresponding value $\tilde{r}$ from the target distribution can be obtained as the argument of the minimum of the difference to the relevant cumulative mass function, as shown in Figure 1b:

$$\tilde{r} = \arg\min\left[CMF(x) - r\right] \tag{3}$$

In the implementation, this is realized using a vector search for Equation 3. Binary search trees can increase the efficiency of this approach and lookup tables or *guide tables* for the individual distributions are even more efficient [24]. For the chosen amount of parameters, the sequential search showed to be efficient enough to run the synthesis smoothly with 80 partials on an *Intel® Core™ i7-5500U CPU at 2.40GHz.*

3.3. Timbre Space Interpolation

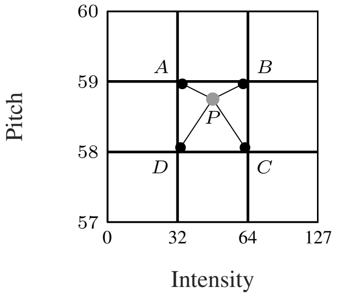


Figure 6: Interpolation in the *timbre plane*.

For the use of expressive input streams for pitch and intensity, arbitrary points in the *timbre plane* need to be synthesized. Interpolation between the support points generated by analyzing the sample library is possible for the mean/median mode as well as for the inverse transform sampling mode. Figure 6 shows a point P located in the square $ABCD$ between four support points. The weights for the parameters at each support point are calculated by the following distance-based formulae:

$$w_A = (1-x)(1-y) \tag{4}$$
$$w_B = x(1-y) \tag{5}$$
$$w_C = x \cdot y \tag{6}$$
$$w_D = (1-x)y \tag{7}$$

In the average mode, the weights w_i can be directly applied to the mean or median values m_i corresponding to the parameter values at the given points A, B, C and D for obtaining the interpolated average $\tilde{m}$:

$$\tilde{m} = w_A m_A + w_B m_B + w_C m_C + w_D m_D \tag{8}$$

In the case of inverse transform sampling, the interpolation is performed as presented in Figure 7. A single random value r is generated from a uniformly distributed random process $\boldsymbol{\mathcal{U}}(0,1)$. This value is then used to generate four random values $\tilde{r}_i$ using the *CMFs* at the four support points. These resulting values are finally multiplied by the weights from Equations 5–7 and summed to obtain the interpolated random value $\tilde{r}^*$:

$$\begin{aligned}\tilde{r}^* &= w_A\tilde{r}(CMF_A) + w_B\tilde{r}(CMF_B)\\ &+ w_C\tilde{r}(CMF_C) + w_D\tilde{r}(CMF_D)\end{aligned} \tag{9}$$

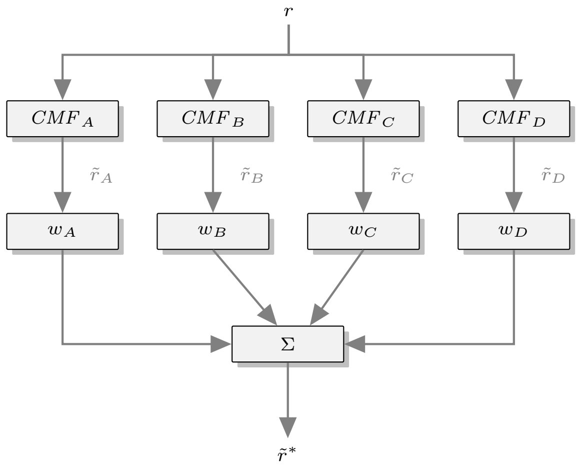


Figure 7: Interpolated inverse transform sampling.

3.4. Smoothing

The inverse transform sampling method as presented above does not consider the recent state for generating new support points. Hence, it does not capture the frequency characteristics of the analyzed trajectories. As a result, rapid changes may occur in the synthesized trajectories, which are not included in the original signals, although the resulting distribution functions are correct. For that reason, an adjustable low-pass filter is inserted after the random number generators for smoothing the trajectories. It should be noted that this filtering process narrows the distribution functions.

4. MEASUREMENTS

For evaluating the ability of the proposed synthesis algorithm to react to expressive control streams, the responses to sweeps in the frequency and intensity dimension are captured and analyzed by qualitative means. Only the deterministic component is used for this evaluation, discarding the noise.

4.1. Frequency Sweeps

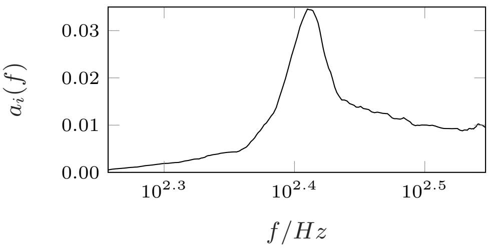
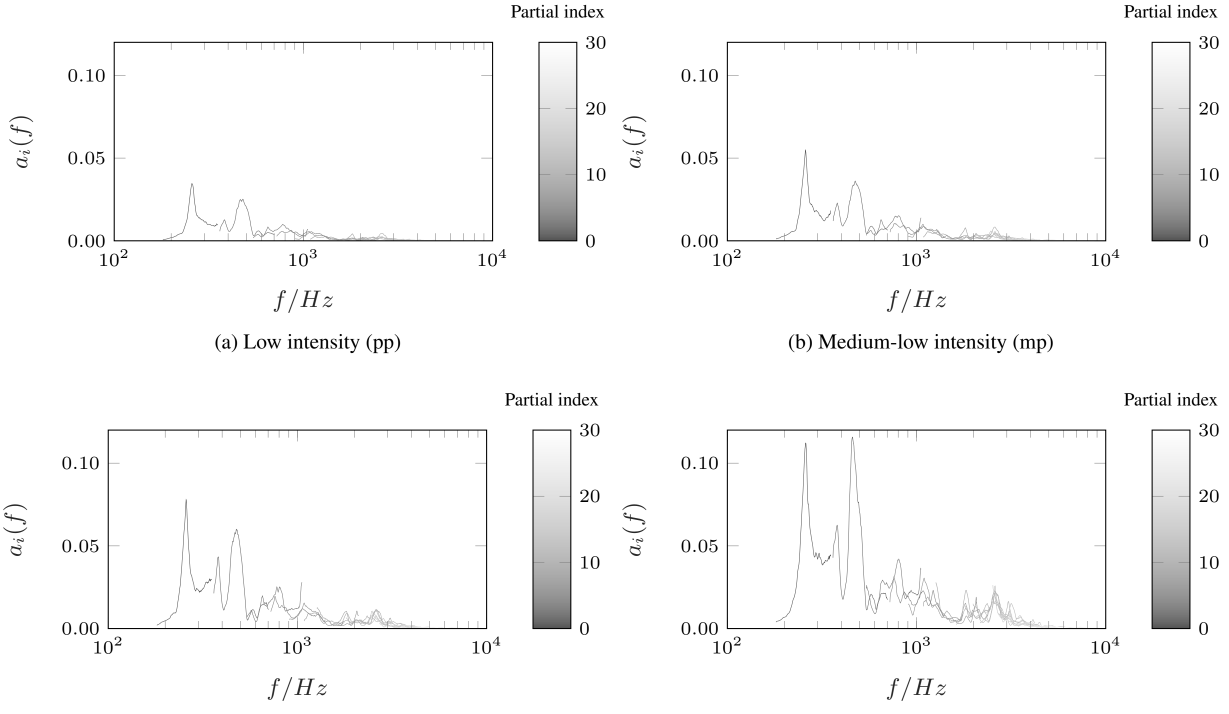


Figure 8: SEM for an octave sweep of the first partial.

For analyzing the effect of the spectral envelope modulations, a frequency sweep of one octave is sent to the synthesis system at four different intensities. The sweep ranges from the lowest tone of MIDI=55 (***G3***, 197.33 Hz) to MIDI=67 (***G4***, 394.67 Hz). The responses of all active partials to the frequency sweeps are recorded as separate signals for an analysis.



(a) Low intensity (pp)

(b) Medium-low intensity (mp)

(c) Medium-high intensity (mf)

(d) High intensity (ff)

Figure 9: Representations of the frequency response through spectral envelope modulations of 30 partials by a one octave sweep.

Figure 8 shows the amplitude of the first partial as a function of the fundamental frequency. The resulting trajectory shows no discontinuities, validating the interpolation process. It further shows a prominent peak at appropriately 257 Hz, caused by the spectral envelope modulations. Joining the amplitude over frequency trajectories of the first 30 partials, the frequency response of the instrument can be visualized through SEM. Results are shown for MIDI intensities 20 (***pp***), 50 (***mp***), 80 (***mf***) and 120 (***ff***) in Figure 9. With increasing partial index, the overlap with the neighboring partial trajectories increases. The approximated frequency responses are thus blurred for higher frequencies.

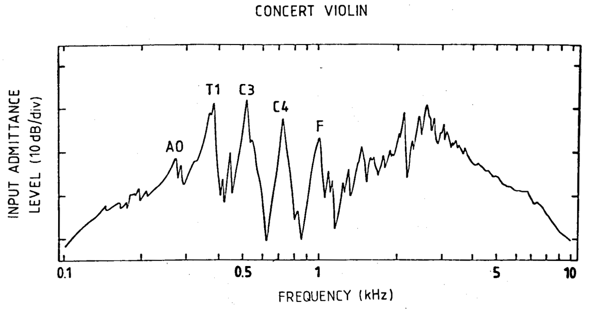


Figure 10: Input admittance at the bass bar side of an Andrea Guarneri violin [25].

All four representations of the frequency responses in Figure 9 show the same prominent peaks. These peaks correspond to the formants typical for violins. For a comparison, the input admittance of a Guarneri violin is shown in Figure 10.

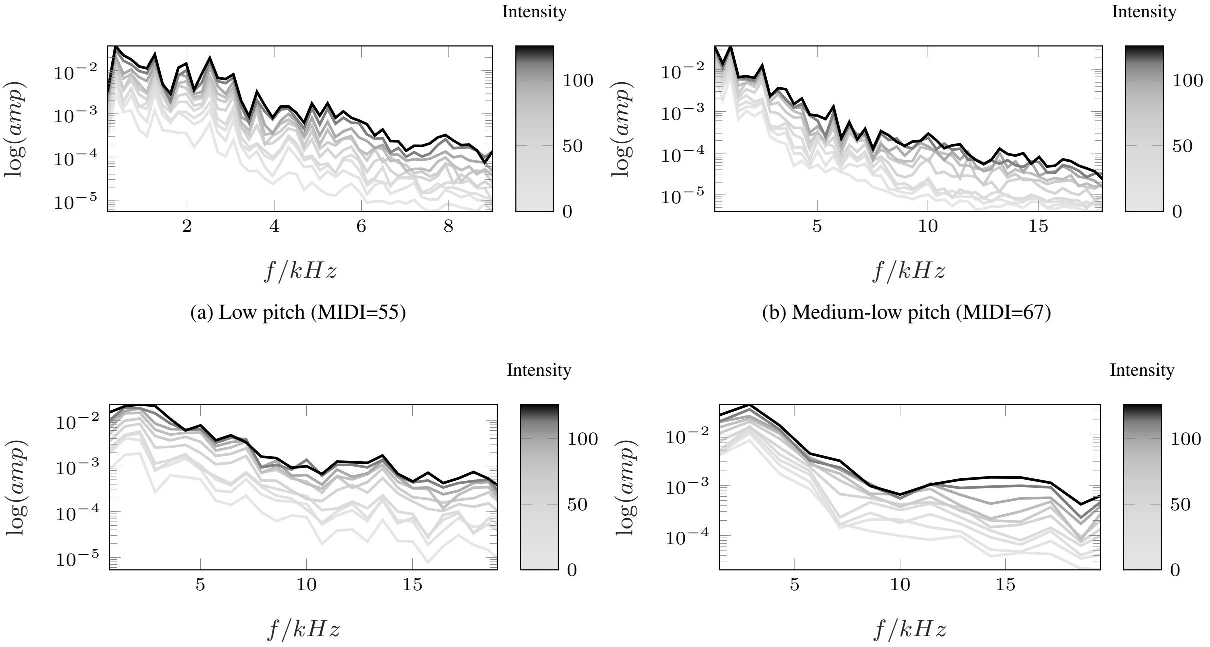
Characteristic resonances of violin bodies have been labeled inconsistently by different researchers. However, referring to Curtin et al. [11], the prominent resonances for Figure 10 are listed in Table 1. Plots 9a – 9d all show the f-hole resonance at 284 Hz and the main wood resonance, respectively the lowest corpus mode at 415 Hz. At higher intensities, the plots show peaks at 709 Hz, 872 Hz and 1170 Hz, related to the upper wood resonances and the lateral air motion. The so called *violin formant* is represented by a region of increased energy between 2000 Hz and 3000 Hz.

Table 1: Main resonances of a violin body [25, 11].

Label	**Frequency**	**Description**
A0:	275 Hz	*f-hole resonance*
C2 (T1):	460 Hz	*main wood*
C3:	530 Hz	*second wood*
C4:	700 Hz	*third wood*
F:	1000 Hz	*lateral air motion*
	2000 - 3000 Hz	*violin formant, bridge hill*

4.2. Intensity Modulations

The response of the synthesis system to changes in intensity is captured at four different pitches. Intensity sweeps from 0 to 127 are used at MIDI pitches 55 (***G3***, 197.33 Hz), 67 (***G4***, 394.67 Hz), 79 (***G5***, 789.33 Hz) and 93 (***A6***, 1772.00 Hz). The plots in Figure 11 show the spectrum of the harmonic signal in dependency of the



(a) Low pitch (MIDI=55)

(b) Medium-low pitch (MIDI=67)

(c) Medium-high pitch (MIDI=79)

(d) High pitch (MIDI=93)

Figure 11: Amplitudes of the first 50 partials in dependency of intensity, captured for four different pitches.

intensity, sampled at the partial frequencies. For higher pitches, the number of partials is reduced, resulting in a lower frequency resolution. An increase in high frequency content is indicated for higher intensities at all pitches.

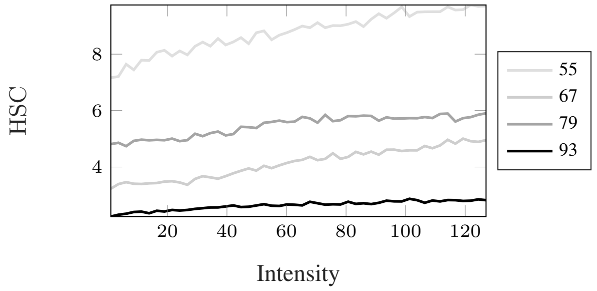


Figure 12: Harmonic spectral centroid as function of intensity for four different MIDI pitches (55, 67, 97, 93).

The harmonic spectral centroid (HSC) is calculated for 50 equally spaced points within all sweeps for analyzing the influence of the intensity on the harmonic component of the signal. Based on the spectral centroid, the HSC regards only the amplitudes a_i of the partials, resulting in a pitch independent measure for the spectral distribution of the partials:

$$\text{HSC} = \frac{\sum\limits_{i=1}^{i=N} i a_i}{\sum\limits_{i=1}^{i=N} a_i} \tag{10}$$

Figure 12 shows the HSC as a function of the intensity for four different pitches. All trajectories show a quasi monotonic increase in the HSC with increasing intensity. Changes in intensity thus result in changes in timbre, respectively in brightness.

5. CONCLUSION

The proposed statistical sinusoidal modeling system is capable of reacting to expressive gestures, using the input parameters pitch and intensity. Evaluations of frequency and intensity sweeps show the desired responses in timbral qualities, validating the interpolated inverse transform sampling. The next important step for improving the algorithm is the implementation of a Markovian inverse transform sampling, considering past values for the random sequence generation and thus preserving the frequency characteristics of the synthesis parameters.

Using the actual inverse cumulative mass functions during runtime could further improve the performance of the algorithm. At the current state the inverse transform sampling requires a search within an unsorted vector, whereas actual inverted functions can be used by simple indexing. The flexibility and compression rate of the model could be increased by using parametric distributions instead of stored distribution functions.

Since the presented approach aims at the synthesis of sustained signals, an integration of parametric transition models [26] and trajectory models for attack and release segments is necessary for completing the synthesis system. Future experiments aim at a perceptual evaluation of synthesized sounds and expressive phrases from the full system. User studies are planned for assessing the applicability of the synthesis approach in a real-time scenario.

6. REFERENCES

[1] Roger B. Dannenberg and Istvan Derenyi, "Combining Instrument and Performance Models for High-Quality Music Synthesis," *Journal of New Music Research*, pp. 211–238, 1998.

[2] Diemo Schwarz, "A System for Data-Driven Concatenative Sound Synthesis," in *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-00)*, Verona, Italy, 2000.

[3] Alfonso Perez, Jordi Bonada, Esteban Maestre, Enric Guaus, and Merlijn Blaauw, "Combining Performance Actions with Spectral Models for Violin Sound Transformation," in *International Congress on Acoustics*, Madrid, Spain, 2007.

[4] Eric Lindemann, "Music Synthesis with Reconstructive Phrase Modeling," *IEEE Signal Processing Magazine [80] March 2007*, 2007.

[5] Henrik Hahn and Axel Röbel, "Extended Source-Filter Model of Quasi-Harmonic Instruments for Sound Synthesis, Transformation and Interpolation," in *Proceedings of the 9th Sound and Music Computing Conference (SMC)*, Copenhagen, Denmark, 2012, pp. 434–441.

[6] Henrik Hahn and Axel Röbel, "Extended Source-Filter Model for Harmonic Instruments for Expressive Control of Sound Synthesis and Transformation," in *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, 2013.

[7] David Wessel, Cyril Drame, and Matthew Wright, "Removing the Time Axis from Spectral Model Analysis-Based Additive Synthesis: Neural Networks versus Memory-Based Machine Learning," in *Proceedings of the International Computer Music Conference (ICMC)*, Ann Arbor, Michigan, 1998, p. 62–65.

[8] Jonathan Driedger, Thomas Prätzlich, and Meinard Müller, "Let it Bee – Towards NMF-Inspired Audio Mosaicing," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Malaga, Spain, 2015, pp. 350–356.

[9] Sanna Wager, Liang Chen, Minje Kim, and Christopher Raphael, "Towards Expressive Instrument Synthesis Through Smooth Frame-by-Frame Reconstruction: From String to Woodwind," in *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, USA, 2017, pp. 391–395.

[10] Ixone Arroabarren, Miroslav Zivanovic, Xavier Rodet, and Alfonso Carlosena, "Instantaneous Frequency and Amplitude of Vibrato in Singing Voice," in *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, China, 2003, p. 537–540.

[11] Joseph Curtin and Thomas D Rossing, "Violin," in *The Science of String Instruments*, Thomas Rossing, Ed., pp. 209–244. Springer, 2010.

[12] Axel Roebel, Simon Maller, and Javier Contreras, "Transforming Vibrato Extent in Monophonic Sounds," in *Proceedings of the 14 th International Conference on Digital Audio Effects (DAFx)*, Paris, France, 2011.

[13] Stefan Weinzierl, Steffen Lepa, Frank Schultz, Erik Detzner, Henrik von Coler, and Gottfried Behler, "Sound Power and Timbre as Cues for the Dynamic Strength of Orchestral Instruments," *The Journal of the Acoustical Society of America*, vol. 144, no. 3, pp. 1347–1355, 2018.

[14] Henrik von Coler, Jonas Margraf, and Paul Schuladen, "TU-Note Violin Sample Library," TU Berlin, `http://dx.doi.org/10.14279/depositonce-6747`, 2018, Data set.

[15] Henrik von Coler, "TU-Note Violin Sample Library – A Database of Violin Sounds with Segmentation Ground Truth," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx)*, Aveiro, Portugal, 2018.

[16] Xavier Serra and Julius Smith, "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition ," *Computer Music Journal*, vol. 14, no. 4, pp. 12–14, 1990.

[17] Alain de Cheveigné and Hideki Kawahara, "YIN, a Fundamental Frequency Estimator for Speech and Music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[18] Arturo Camacho, *Swipe: A Sawtooth Waveform Inspired Pitch Estimator for Speech and Music*, Ph.D. thesis, Gainesville, FL, USA, 2007.

[19] Orchisama Das, Julius Smith, and Chris Chafe, "Real-time Pitch Tracking in Audio Signals with the Extended Complex Kalman Filter," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx)*, Edinburgh, UK, 2017.

[20] Julius O. Smith and Xavier Serra, "PARSHL: An Analysis/Synthesis Program for Non-Harmonic Sounds Based on a Sinusoidal Representation," in *Proceedings of the International Computer Music Conference (ICMC)*, Barcelona, Spain, 2005.

[21] Scott Levine and Julius Smith, "A Sines+Transients+Noise Audio Representation for Data Compression and Time/Pitch Scale Modifications," in *Proceedings of the 105th Audio Engineering Society Convention*, San Francisco, CA, 1998.

[22] Henrik von Coler, "A Jack-based Application for Spectro-Spatial Additive Synthesis," in *Proceedings of the 17th Linux Audio Conference (LAC-19)*, Stanford University, USA, 2019.

[23] Luc Devroye, "Sample-based Non-uniform Random Variate Generation," in *Winter Simulation Conference*, Washington, D.C., USA, 12 1986, pp. 260–265.

[24] Luc Devroye, *Non-Uniform Random Variate Generation*, Springer, McGill University, 1986.

[25] J. Alonso Moral and E. Jansson, "Input Admittance, Eigenmodes and Quality of Violins," in *STL-QPSR*, vol. 23, pp. 60–75. KTH Royal Institute of Technology, 1982.

[26] Henrik von Coler, Moritz Götz, and Steffen Lepa, "Parametric Synthesis of Glissando Note Transitions – A user Study in a Real-Time Application," in *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx)*, Aveiro, Portugal, 2018.