Time domain attack and release modeling - applied to spectral domain sound synthesis

A Time Domain Sound Attack and Release Model in Combination with a Parametric Spectral Synthesis Model

Cornelia Kreutzer[1], Jacqueline Walker[1], and Michael O'Neill[2]

[1] University of Limerick, Department of Electronics and Computer Engineering
Limerick, Ireland
{Cornelia.Kreutzer, Jacqueline.Walker}@ul.ie

[2] University College Dublin, School of Computer Science and Informatics
Dublin, Ireland
{m.oneill}@ucd.ie

Abstract. We introduce a time-domain model for the synthesis of attack and release parts of musical sounds. This approach is an extension of a spectral synthesis model we developed: the Parametric Synthesis Model (PSM). The attack and release model is independent from a preceeding spectral analysis as it is based on the time domain sustain part of the sound. We apply a shaping function to this sustain part to obtain the sound attack and the release. The model has been tested with linear and polynomial shaping functions and produces good results for three different instruments. The time-domain approach also overcomes the problem of synthesis artifacts that often occur when using spectral analysis/synthesis methods for sounds with transient events. Moreover, the model can be combined with any synthesis model of the sustain part and offers the possibility to determine the duration of the attack and release parts of the sound.

1 Introduction

In the standard sinusoidal model used for speech [1] and musical sounds [2, 3], the harmonic part of a given signal is modeled as a sum of sinusoidal components with time-varying amplitude, frequency and phase. The remaining sound components are then usually added to the model by using some type of noise model. However, these methods are not sufficient to model transient parts of the signal. Transients mainly occur during the onset of a sound and have long been known to be important for our perception of timbre [4, 5].

A number of methods have been introduced to provide a sinusoidal sound model that is also capable of modeling transients more accurately. Jensen [6] proposed an amplitude model in the frequency domain where the amplitude envelope of each harmonic partial is fitted with appropriate functions. Verma and Meng [7] proposed an extension of the Spectral Modeling Synthesis (SMS) framework to model transients by performing sinusoidal modeling in the frequency domain. This is based on the observation that transient components of a signal show the same behavior in the frequency domain as sinusoidal components in the time domain. In [8] the standard sinusoidal model of the sound is extended by using exponentially damped sinusoids to model transient events more accurately. Similar approaches, also using exponentially damped sinusoids, have been proposed by others [9, 10] as well. Meillier and Chaigne [11] applied an autoregressive model which improved the spectral analysis of percussive sounds compared to the standard FFT approach. In [12] the spectral analysis is improved by synchronizing the analysis window to transient events. This overcomes the problem that transient events, which occur at a certain time, become diffused during the synthesis process when using the standard sinusoidal model.

All these approaches focus on improving the sinusoidal sound model in the spectral domain. Thus, the transient of the analysed sound is captured more accurately and artifacts during the synthesis process are reduced.

In contrast to that, we propose a time domain model for attack and release parts of musical sounds. We use this model in combination with a spectral synthesis model: the Parametric Synthesis Model (PSM). We model the sound attack and release independent from a former spectral analysis of these signal parts. Therefor, we exclude artifacts that might occur when using a transient analysis-synthesis model. This artifacts are due to interpolations of the sound partials between signal frames when it comes to the synthesis process.

The time domain approach also leads to a reduction in computational requirements, because it does not require to analyse and model the amplitude envelope of each partial individually.

The paper is organized as follows. After a brief overview of the SMS synthesis and analysis procedure in Section 2, we give a description of the PSM synthesis model in Section 3. In Section 4 we introduce the Time Domain Attack and Release Model. This is followed by a discussion of synthesis results obtained by the combination of the PSM and the proposed Time Domain Attack and Release Model. Finally, we give a summary of the presents work and an outlook on future work in Section 6.

2 Spectral Modeling Synthesis

Spectral Modeling Synthesis (SMS) is a framework for spectral analysis, synthesis and transformations of musical sounds introduced by Serra [2]. The approach is to analyse the spectral content of a given sound sample in order to perform a spectral synthesis using these analysis results.

The SMS framework consists of a deterministic and a stochastic model, where the deterministic model is used for the sinusoidal parts of the sound and the stochastic model is applied to the residual part - sometimes also referred to as the noise part - of the sound. Both models are independent from each other which allows a flexible analysis and re-synthesis process.

Since its proposal, the original framework has been further developed [13, 14], and a number of extensions have been proposed, such as additional models for transient parts of the sound as described in [7] or feature based sound transformation methods as described in [15].

2.1 SMS Analysis

There are several different ways to implement the spectral analysis for the SMS framework. In the following only the methods relevant for the present work will be discussed. In particular, we are using an SMS implementation described in [14]. For further details on alternative methods please consult [13].

During the analysis the time varying spectral parameters of the sound are detected and represented with time varying sinusoids. These sinusoidal partials are then subtracted from the original sound and the remaining *residual* is modeled with the stochastic model, e.g., as time varying white noise.

The first analysis step consists of applying the Short Term Fourier Transform (STFT) to the input signal to obtain the magnitude and the phase spectra of the sound. For the presented work a 1024 point STFT with a 256 point window overlap and a Blackman–Harris window is used.

Sinusoidal partials. The deterministic or harmonic parts of the sound are defined as a sum of sinusoids with a slowly varying amplitude and frequency. Each sinusoid within this model is described by an amplitude and a frequency function.

Once the sound spectrum is obtained by means of the STFT, the prominent spectral peaks are detected and tracked using a peak tracking algorithm. The objective of this algorithm is to detect the magnitude, frequency and phase of the sinusoidal partials. In case the sound is pseudo-harmonic a pitch detection method can be used to integrate information about the fundamental frequency and improve the peak tracking procedure. After computing the frequency spectrum of the sound, the next step is to detect the most prominent spectral peaks. Peaks are defined as local maxima in the magnitude spectrum. Starting from the largest one these maxima are detected, where the number of peaks to be picked is user defined. In the present case the number of peaks is set to 100.

In theory, sinusoidal partials have a well defined frequency representation which allows them to be easily distinguished from other signal components. However, most natural sounds are not perfectly periodic, show interactions between their components and are not free from prominent noise components. Which makes the detection of the actual sinusoidal partials a more difficult task. Because of these issues the peak detection process detects as many peaks as possible and leaves the decision of which peak defines a sinusoidal partial to the next analysis step: the peak tracking algorithm.

Pitch detection. Before starting to track the detected peaks, it can be useful to search for the underlying fundamental frequency of the sound to improve the results of the peak tracking procedure. Therefore, a simple pitch detection method [16] is applied. The method uses the two–way mismatch algorithm [17] and is based on the assumption that the fundamental frequency, or at least the first few of its partials, exist. In case no pitch is detected by the pitch detection algorithm, the sound is supposed to be inharmonic and the peak tracking is performed without considering any information about the fundamental frequency.

Note, that we do not distinguish between the fundamental frequency and the pitch of the sound here. This is due to the assumptions being made that the fundamental frequency is the pitch. However, it is only the fundamental frequency and not the perceived pitch value we are referring to.

Partial tracking. The peak detection method returns magnitude, frequency and phase values of the strongest peaks of a signal frame sorted by frequency. Once the spectral peaks of a frame have been detected a partial tracking method is applied to determine the sinusoidal sound partials.

If a fundamental frequency has been detected the frequencies are set according to the harmonic series of the fundamental frequency. For inharmonic sounds the tracking algorithm starts off with the highest peaks detected for the first signal frame. The trajectories for the sinusoidal partials are then computed as follows. Each spectral peak is assigned to the trajectory closest to it. When a signal frame does not provide a peak value close to an existing track, the track is terminated. If this happens the next most prominent peak within this frame is used to start off a new track. The spectral peaks for each harmonic partial are then connected by linear interpolation to form the overall shapes of the amplitude tracks.

2.2 SMS Synthesis

The synthesis part is a combination of additive synthesis for the sinusoidal partials, and subtractive synthesis for the residual part of the sound.

The deterministic component is generated with additive synthesis, similar to the sinusoidal synthesis that was part of the analysis, with the difference that now the phase trajectories are discarded. By not considering the phase trajectories, the synthesis is done in the time domain as described in the following.

The instantaneous amplitude $\hat{A}(m)$ of a particular partial is obtained by linear interpolation,

$$\hat{A}(m) = \hat{A}^{l-1} + \frac{\hat{A}^{l} - \hat{A}^{l-1}}{S} m \tag{1}$$

where $m = 0, 1, \ldots, S-1$ denotes the time sample in the l^{th} signal frame. The instantaneous radian frequency $\hat{\omega}(m)$ is obtained by linear interpolation. Subsequently the instantaneous phase is taken to be the integral of the frequency

$$\hat{\omega}(m) = \hat{\omega}^{l-1} + \frac{\hat{\omega}^{l} - \hat{\omega}^{l-1}}{S} m \tag{2}$$

The instantaneous phase for the r^{th} signal partial is defined as

$$\hat{\theta}_r(m) = \hat{\theta}_r(l-1) + \hat{\omega}_r(m) \tag{3}$$

Which leads to the final synthesis equation given by

$$d^l(m) = \sum_{r=1}^{R} \hat{A}_r^l(m) cos \left[\hat{\theta}_r^l(m) \right] \tag{4}$$

where $\hat{A}(m)$ and $\hat{\theta}(m)$ are the calculated instantaneous amplitude and phase, respectively.

The synthesis of the residual part is usually carried out by generating a noise signal that has the frequency and amplitude characteristics that have been obtained by the spectral analysis of the stochastic signal part. In order to generate the residual sound segments, an inverse STFT of their spectral envelopes is computed. The time varying sound parameters are then calculated by an overlap-add process like the one used for the deterministic synthesis.

To compute the SMS synthesis results for comparison with the PSM results we did not apply such a specific noise model, but applied an inverse FFT to the residual part of the signal, such as described in [14].

3 Parametric Spectral Model

3.1 Frequency estimation

To determine the frequency values within the synthesis model we use a flexible model that is not based directly on a preceding spectral analysis but on the basic knowledge about the sound. The fundamental frequency, or pitch, as well as the number of harmonic partials are user defined values. This is particularly important if the synthesized sound lies outside the range of the instrument the model is supposed to mimic. Also, within the range of an instrument there is no restriction of the pitch value or the number of harmonics that can be chosen, since both values are entirely user defined. Consequently we can model whole tones, semitones or quarter tones of an instrument as well as other notes whose pitch values is anywhere in between or outside these tones.

We apply a random walk to several frequency partials in order to reconstruct the naturalness of the sound. Figure 1 shows a representative result of the SMS partial tracking algorithm: in this particular case the result for a flute note (A4, played forte, non Vibrato [18, Instrument Nr.33, Flute:Sankyo]). As illustrated, some of the partials, especially the upper ones, show a certain amount of variation or *noisiness*. Due to this noisiness a reconstruction of the sinusoidal parts of the sound does keep the sound characteristics of the original recording, although the residual part of the signal is neglected for the reconstruction.

Because of this observation we incorporate this noisiness into the sinusoidal partials of our synthesis model rather than defining a separate noise model. This is achieved by the use of a one-dimensional random walk [19] to determine the frequencies of the upper harmonic partials.

A one-dimensional random walk can be described as a path starting from a certain point, and then taking successive steps on a one-dimensional grid. The

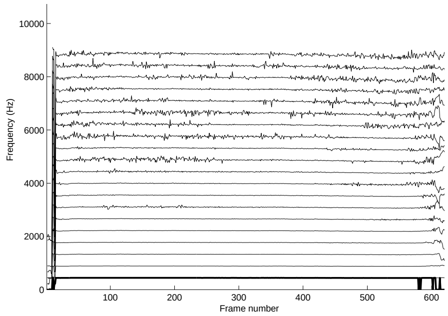

Fig. 1. SMS frequency analysis result of a flute note recording (A4 forte, non Vibrato)

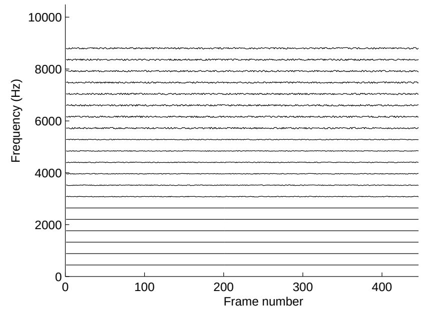

Fig. 2. Estimated frequency tracks for a note with 20 harmonics (fundamental frequency: 440 Hz)

step size is constant and the direction of each step is chosen randomly with all directions being equally likely.

For the purpose of our synthesis model random walks are applied to certain harmonic partials in the following way. First, the harmonic partials are divided into three groups, where each group represents a third of the overall number of harmonics. This follows from the results of the SMS analysis which shows different levels of variations for the lower, the middle and the upper harmonics. Concerning the lowest third of the harmonic partials - starting from the fundamental frequency - no random walk is applied as the analysis of these lower partials shows very little variation.

For the middle and the upper harmonic partials a random walk is applied, where the starting point of the random walk is determined by the basic frequency of the harmonic partial. Basic frequency in that case means the integer multiple of the fundamental frequency. Again, from the analysis result it can be seen that the upper harmonics show more variation than the middle ones. Due to that, and after testing several levels of noisiness, the step size of the random walk was set to 30 Hz for the upper harmonics and to 15 Hz for the middle ones. Figure 2 shows the estimated frequency tracks for the synthesis model with the same conditions as the flute sound in Figure 1 (a fundamental frequency of 440 Hz and 20 sinusoidal partials).

3.2 Amplitude estimation

In contrast to the frequency estimation which is not directly taken from the sound analysis results, we use SMS analysis results as a basis for estimating the amplitude values of the harmonic partials.

However, we reduce the number of parameters to provide a flexible synthesis model that is mostly independent from the preceding sound analysis process. The reduced number of parameters also reduces the computational complexity of the synthesis process. Additionally, our main concern is to keep the quality and naturalness of the musical sound after the synthesis process in order to mimic real instruments. Therefore, three different methods have been applied to the analysis amplitude data. In particular we have carried out amplitude estimation by means of local optimization, lowpass filter estimation and polynomial fitting. A detailed discussion of all these methods will be provided in the following sections after a short description of the applied analysis procedure.

We start by applying a standard SMS analysis as described in Section 2.1 to obtain the amplitude parameters. To increase the number of spectral samples per Hz and improve the accuracy of the peak detection process, we apply zero-padding in the time domain - using a zero-padding factor of 2. The STFT was performed with a sampling rate of 44.1 kHz and a Blackman-Harris window with a window size of 1024 points and a hop size of 256 points. From the resulting frequency spectrum, 100 spectral peaks were detected and subsequently used to track the harmonic partials of the sound. The number of partials to be tracked was set to 20. This analysis has been applied to sound samples taken from the RWC database [18]. In particular to all notes over the range of a flute, a violin and a piano. Given the amplitude tracking results only one representative note for each instrument has been chosen to provide the basis for the amplitude values of the PSM. However, this could be changed in the future into using more than one amplitude template, e.g., using different templates for the low notes and the high notes within the range of an instrument.

Local optimization. The SMS analysis provides one amplitude value for each harmonic partial and for each frame of a given sound signal. We reduce that parameter size by determining the local maxima of each amplitude track. This reduces the number of parameters to about a third of the SMS analysis result. For example, for the flute note (A4, played forte, non Vibrato) the SMS analysis consists of 12680 amplitude values. This is reduced to 3015 values which represent all the local maxima of the 20 harmonic partials.

We determine the local maxima of each amplitude envelope by using the first derivative of the amplitude envelope function f_e. Suppose we want to determine if f_e has a maximum at point x. If x is a maximum of f_e, then f_e is increasing to the left of x and decreasing to the right of x. The same principle applies for local minima of f_e. If x is a minimum of f_e, then f_e is decreasing to the left of x and increasing to the right of x. In contrast, if f_e is increasing or decreasing on both sides of x, then x is not a maximum or a minimum. In terms of the first derivative of f_e this means, that f_e is increasing when the derivative is positive, and that f_e is decreasing when the derivative is negative.

To compute the shape of each amplitude track, necessary for the synthesis process, we then perform a one-dimensional linear interpolation between the local maxima of the track.

Figures 4, 8 and 12 illustrate the estimated amplitude tracks for the three considered instruments. As can be seen the shape of the tracks are close to the SMS analysis result shown in Figures 3, 7 and 11. However, this is not the case for the attack and the release part of the sound.

Lowpass filter estimation. The second curve fitting method uses a lowpass filter to estimate the overall amplitude envelope of each partial. We apply a 3^{rd} order Butterworth lowpass filter to the analysis data. We perform zero-phase digital filtering by processing the input data in both the forward and reverse directions. After filtering in the forward direction, the filtered sequence is reversed and runs back through the filter. The resulting sequence has precisely zero-phase distortion and double the filter order.

As shown in Figures 5 and 8 the envelope shapes of the estimated amplitude tracks are similar to the local optimization estimation. However, the estimation takes significantly longer to be performed. Similar to the local optimization method, no sufficient estimate for the synthesis of the attack and the release of the sound signal is obtained.

Polynomial interpolation. Additionally to the local optimization and the lowpass filter method we performed polynomial fitting to obtain an estimate for the several amplitude tracks. For each amplitude envelope the coefficients of a polynomial of degree 6 are computed that fit the data - in our case the analysis result - in a least squares sense. This computation is performed using a Vandermonde matrix [20]

$$V = \begin{bmatrix} 1 & \alpha_1 & \alpha_1 & \dots & \alpha_1^{n-1} \\ 1 & \alpha_2 & \alpha_2 & \dots & \alpha_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_m & \alpha_m & \dots & \alpha_m^{n-1} \end{bmatrix} \tag{5}$$

since solving the system of linear equations $Vu = y$ for u with V being an $n \times n$ Vandermonde matrix is equivalent to finding the coefficients u_j of the polynomial

$$P(x) = \sum_{j=0}^{n-1} u_j x_j \tag{6}$$

of degree $\leq n-1$ with the values y_i at α_i [20].

The estimated amplitude envelopes are shown in Figures 6, 10 and 13. In contrast to the two other methods being used, the results are very smooth amplitude envelopes. That is, all the small variations that can be seen in the SMS analysis result are missing. Nevertheless, the synthesized sounds preserve the timbre of the particular instrument and the sound quality of the original recordings. Regarding the flute and the violin, the polynomial estimation also gives a sufficient estimate for the attack and the release part of the sound signal.

3.3 Spectral synthesis

With the calculated frequency and amplitude parameters we synthesize a new sound using an additive synthesis method. This method is based on the inverse FFT as described in [14, 21]. Compared to the traditional use of oscillator banks

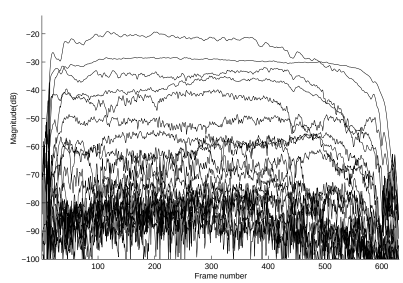

Fig. 3. SMS amplitude analysis result of a flute note recording (A4, played forte, non Vibrato)

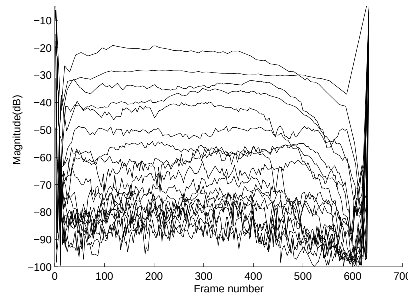

Fig. 4. Estimated amplitude tracks for a flute note (A4, f0:440 Hz) using local maxima estimation

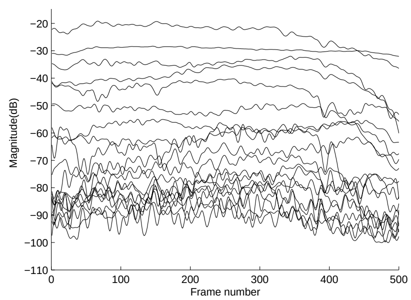

Fig. 5. Estimated amplitude tracks for the sustain part of a flute note (A4, f0:440 Hz) using an LP filter

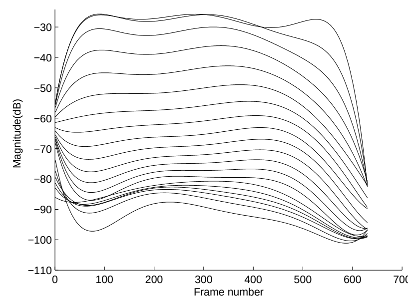

Fig. 6. Estimated amplitude tracks for a flute note (A4, f0:440 Hz) using a polynomial fit

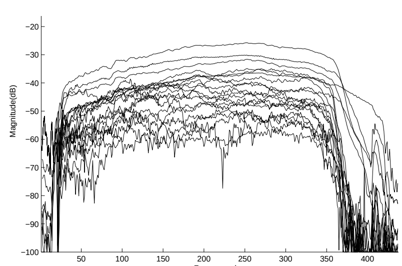

Fig. 7. SMS amplitude analysis result of a violin note recording (A#3, played forte, non Vibrato)

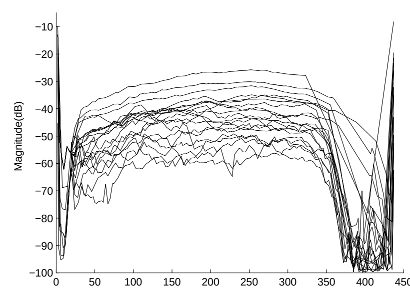

Fig. 8. Estimated amplitude tracks for a violin note (A#3, f0:233.08 Hz) using local maxima estimation

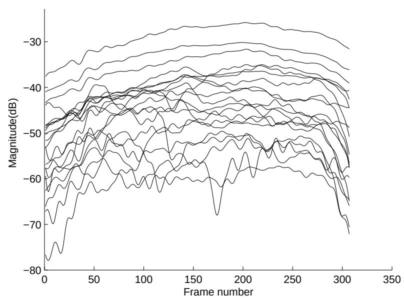

Fig. 9. Estimated amplitude tracks for the sustain part of a violin note (A#3, f0:233.08 Hz) using an LP filter

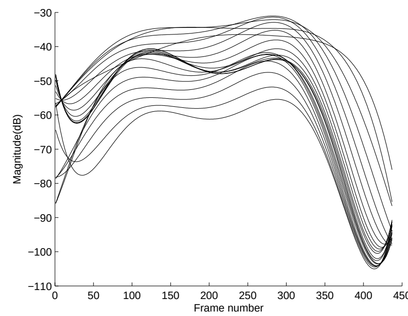

Fig. 10. Estimated amplitude tracks for a violin note (A#3, f0:233.08 Hz) using a polynomial fit

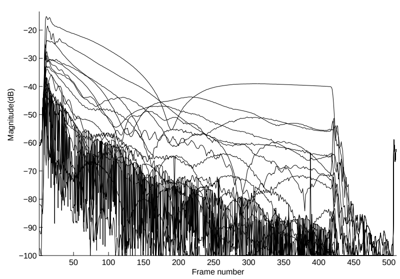

Fig. 11. SMS amplitude analysis result of a piano note recording (C#4, played forte, non Vibrato)

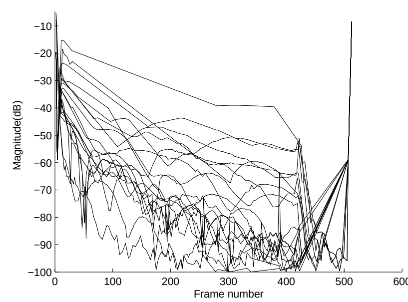

Fig. 12. Estimated amplitude tracks for a piano note (C#4, f0:277.18 Hz) using local maxima estimation

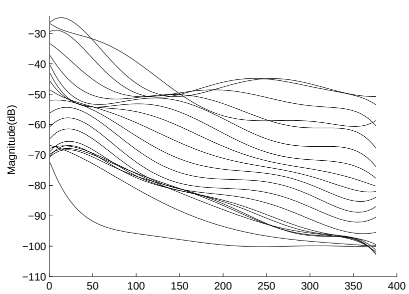

Fig. 13. Estimated amplitude tracks for the sustain part of a piano note (C#4, f0:277.18 Hz) using a polynomial fit

for additive synthesis, this is a more efficient and faster approach. The faster computation results from the fact that a sinusoid is a sinc-type function in the frequency domain, using the transform of the window, and not all samples in these functions have the same weight [14]. So it is only necessary to calculate the main lobe samples of the window transform with the corresponding amplitude, frequency and phase values to generate a sinusoid in the spectral domain. All the main lobes of the sinusoids we want to compute are then placed into an FFT buffer and by performing an inverse FFT we obtain the synthesized time-domain signal. Applying an overlap-add method then gives the time varying characteristics of the sound.

The window used for the transform is a Blackman-Harris window. Though, such a window does not overlap perfectly to a constant in the time domain without using very high overlap factors. To overcome this problem we divide by the window in the time domain and apply a triangular window before the overlap-add process [14, 21].

4 Time Domain Attack and Release Model

As discussed in Section 3, it is difficult to obtain an efficient estimate for the attack and release part of the sound signal using amplitude estimation methods alone. Therefore, PSM synthesis results resemble the sound quality and the timbre of the original sound signal during the sustained portion of the note but have audible artefacts during the attack and the release portions of the note.

To overcome this problem and gain better synthesis results, we extend the synthesis model with a time domain attack and release model. The approach we are using to obtain the attack and release portion of the sound corresponds to multiplying the sustain portion of the sound by a time domain window. Thus, we accomplish the desired transformation in the frequency domain. This reduces the complexity of the model significantly, as the alternative is to map each amplitude partial in detail through the attack and release stages in the frequency domain.

Figure 14 illustrates the computation of the sound attack and release portions in the time domain. Details of the procedure and the two different shaping functions we have tested will be described in the following.

4.1 Linear Shape

To synthesize the attack and release portions of the sound we require the sustain part of the PSM synthesized signal in the time domain and the durations of the attack and the release parts we want to model. Both duration times may be user defined and thus can be changed according to the signal length and the instrument.

The attack is computed as follows. From the beginning of the sustain signal we take a part with the same length as the attack duration. This is the part of the signal that is shaped to gain the attack portion. Then, we carry out a point wise multiplication of this part with a linear shaping function $y_{att}(n) =$

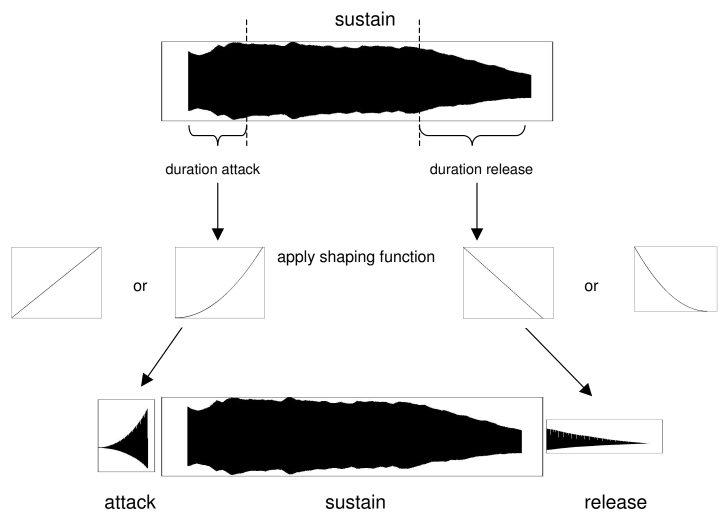

Fig. 14. Diagram of the Time Domain Attack and Release Model

$k * x(n)$, for $n = \{1, 2, \ldots, N\}$. This can be compared to the application of a time domain window. The parameters of y_{att} are set according to the given signal, with $y_{att}(1) = 0$ to ensure that the sound starts at 0. The length N of the shaping function is equal to the duration of the attack. The slope k of the function is determined by $k = y_{att}(N) - y_{att}(1)/(N - 1)$. Thus, $y_{att}(N) = 1$. This allows for a smooth transition between the attack and the sustain portion of the sound when they are joined.

For the release part of the sound the procedure is similar to the attack, but here we perform the shaping at the end of the sustain signal. From the end of the sustain signal a part with the same length as the release duration is taken. To compute the sound release we carry out a point wise multiplication of this signal part with a linear shaping function $y_{rel}(m) = -k * x(m)$, for $m = \{1, 2, \ldots, M\}$. The function length M is equal to the duration of the release and $y_{rel}(M) = 0$ to ensure that the sound terminates to 0. The negative function slope k is determined by $k = -y_{att}(N) - y_{att}(1)/(N - 1)$. To ensure a smooth transition between the sustain and the release part of the sound the function parameters are set so that $y_{rel}(1) = 0.5$. Although setting $y_{rel}(1) = 0.5$ works well for the three different instruments we have tested so far, it must be noted that this value is largely dependent on the shape of the given sustain signal.

After the computing attack and the release portions, both are connected to the original sustain part of the PSM synthesized sustain signal. To do so, the three seperate waveforms are concatenated in the order attack - sustain - release. This results in the overall synthesized time domain signal.

4.2 Polynomial Shape

To obtain a more smooth and realistic attack and release, we also used a second order polynomial as a shaping function. Setting the function parameters and computing the particular attack and release signals has been performed similarly to the linear shape. The synthesis result of the PSM model and the user defined duration of the sound attack and release parts provide the basis for the model. To compute the attack a part of length N - equal to the attack duration - is taken from the beginning of the sustain signal. This waveform is then pointwise multiplied with the polynomial function $y_{att}(n) = k * x(n)^2$, for $n = \{1, 2, \ldots, N\}$. As with the linear shaping function, the function parameters are set to ensure $y_{att}(1) = 0$ and $y_{att}(N) = 1$ Therefore, the sound starts at 0 and we gain a smooth transition between the attack and the sustain portion of the synthesized waveform.

For the sound release we also use a second order polynomial, but this time with a negative slope. From the end of the sustain signal a part of length M - equal to the release duration - is taken. Subsequently, this waveform is pointwise multiplied with the polynomial function $y_{rel}(m) = -k * x(m)^2$, for $m = \{1, 2, \ldots, M\}$. The function parameters are set so that $y_{rel}(1) = 0.5$ and $y_{rel}(M) = 0$. This provides for a smooth transition between the sustain signal and the release portion and ensures that the sound terminates to 0. Again, note that the setting of $y_{rel}(1)$ depends on the shape of the given sustain signal. For the instruments we have tested so far, 0.5 has shown to be the most suitable setting.

After computing the sound attack and the release, both are connected to the original sustain part of the PSM synthesized sound to form the overall synthesized time domain signal.

5 Empirical Evaluation

Figures 15, 16 and 17 show comparisons of the original sound sample, the SMS synthesis result and the PSM synthesis result with the time-domain attack and release for the three different instruments being used. In all three cases the PSM amplitude values have been estimated using the local optimization method described in Section 3.2. For the SMS results we applied a standard SMS analysis/synthesis as described in Section 2.

The presented PSM model has been tested for notes covering the whole range of a flute (consisting in 37 notes), a violin (consisting in 64 notes) and a piano (consisting in 88 notes). An SMS analysis has been carried out for all these notes using recorded samples from the RWC database [18]. The analysis was done to find a representative note for the presented amplitude model and to compare the synthesis results obtained by our model with the standard SMS results.

As mentioned before, the amplitude analysis results of only one note per instrument have been used as a basis for the PSM synthesis model. This way the amplitude shape stays the same for all notes of an instrument and only

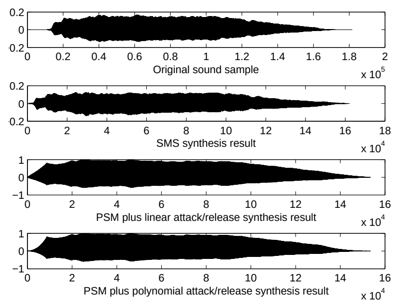

Fig. 15. Flute: original sound, SMS result and PSM result with attack/release model (time domain plots)

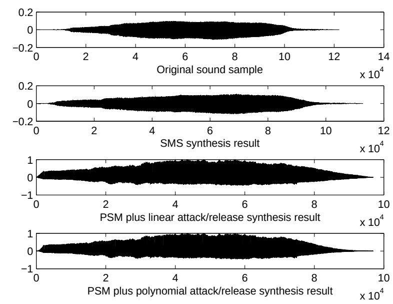

Fig. 16. Violin: original sound, SMS result and PSM result with attack/release model (time domain plots)

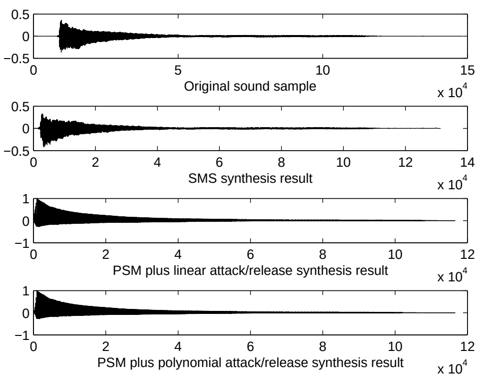

Fig. 17. Piano: original sound, SMS result and PSM result with attack/release model (time domain plots)

the frequencies are changed to obtain the synthesis results. However, the model also allows us to modify the amplitude data if desired. For example, different amplitude *templates* can be used for different parts of the range of an instrument.

The frequency estimation works very well and allows a large flexibility when choosing the fundamental frequency. Due to the random walk that is applied to higher frequency partials the synthesised sound keeps the natural noisiness of the real instrument recording without the need for a separate noise model.

Concerning the three different amplitude estimation methods, all of them perform well when estimating the sustain part of the signal. Although, only the polynomial fit gives a satisfactory estimate for the attack and the release parts of the signal at the same time. The combination of the basic PSM model with the time domain attack and release model overcomes these difficulties and provides an efficient method to model the beginning and the end of the sound. Moreover, the attack/release model is independent from a preceeding spectral analysis and from the computation of the sustain portion of the sound. Using this approach we avoid artefacts that result from smoothing transient events, a problem connected with spectral transient analysis/synthesis methods. Together with the user defined duration, the new approach presented here allows for a flexible synthesis model. The proposed model is also less computationally intensive than a standard spectral analysis/synthesis approach. This is due to the use of fewer parameters.

Overall the synthesis results resemble the timbre and sound quality of the original recordings with slight perceptual differences only in the upper range of the instruments.

6 Conclusion and Future Work

We introduced a time domain attack and release model as a an extension of a Parametric Synthesis Model for musical sounds.

To obtain the shape of the note onset and release we use linear and polynomial shaping functions. The parameters of these functions are determined to ensure that the signal starts and ends at 0. Additionally, the setting of the function parameters ensures that the transition between the attack and release and the sustain part of the signal are smooth.

The PSM model has been tested for notes covering the whole range of three different instruments; a flute, a violin and a piano. The timbre and the perceptual quality of the sound is preserved even for notes at the upper end of the instrument range and for sounds that are outside the range of the instrument. This is not always the case for traditional analysis/re-synthesis approaches, mostly due to the quality of the recorded sound samples and the complex analysis procedure. The synthesis of sounds outside the instrument range by means of an analysis/re-synthesis method also requires additional transformations after the analysis which can lead to artifacts in the synthesised sound too. For notes at the upper end of the instrument range a difference between the recorded sample and the synthesised sound of the presented model can be recognized, especially for the flute and the violin. However, there are no artifacts present in these sounds and the timbre and pitch are the same as in the original sample.

Future work will be focused on analysing the effects of the time domain model on the spectral representation of the signal. Moreover, we are planning to carry out listening tests to gain more detailed results for a comparison between the original recorded sound samples, the synthesis results of the SMS framework and the presented PSM model. We are also planning to extend the number of music instruments being used.

References

1. McAuley, R., Quatieri, T.: *Speech Analysis/Synthesis Based on a Sinusoidal Representation.* **34** (1986) 744–754 IEEE Transactions on Acoustics, Speech and Signal Processing.

2. Serra, X.: A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition. PhD thesis, Stanford University (1989)
3. Serra, X., Smith, J.: *Spectral Modeling Synthesis:A Sound Analysis/Synthesis Based on a Deterministic plus Stochastic Decomposition*. Computer Music Journal **14** (1990) 12–24
4. Grey, J.M.: *Multidimensional Perceptual Scaling of Musical Timbre*. Journal of the Acoustical Society of America **61** (1977) 1270–1277
5. McAdams, S., Cunibile, J.C.: *Perception of Timbral Analogies*. Physical Transactions of the Royal Society **336** (1992) Series B.
6. Jensen, K.: Timbre Models of Musical Sounds. PhD thesis, University of Copenhagen, Copenhagen, Denmark (1999)
7. Verma, T.S., Meng, T.H.Y.: *Extending Spectral Modeling Synthesis with Transient Modeling Synthesis*. Computer Music Journal **24** (2000) 47–59
8. Nieuwenhuijse, J., Heusdens, R., Deprettere, E.: *Robust Exponential Modeling of Audio Signals*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (1998) 3581–3584 Seattle, Washington, USA.
9. Boyer, R., Essid, S.: Transient Modeling with a Frequency–Transform Subspace Algorithm and "Transient+Sinusoidal" Scheme, IEEE International Conference on Digital Signal Processing (DSP) (2002) 865–868, vol.2 Thera, Greece.
10. Hermus, K., Verhelst, W., Lemmerling, P., Wambacq, P., van Huffel, S.: *Perceptual Audio Modeling with Exponentially Damped Sinusoids*. Signal Processing **85** (2005) 163–176 Elsevier North-Holland, Inc., Amsterdam, The Neatherlands.
11. Meillier, J.L., Chaigne, A.: *AR Modeling of Musical Transients*, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (1991) 3649–3652 Toronto, Canada.
12. Masri, P., Bateman, A.: Improved Modelling of Attack Transients in Music Analysis–Resynthesis, International Computer Music Conference (ICMC), IEEE (1996) 100–103 Hong Kong, China.
13. Serra, X.: 3. Musical Signal Processing. In: *Musical Sound Modeling with Sinusoids plus Noise*. Swets & Zeitlinger, Lisse, The Neatherlands (1997) 91–123
14. Amatriain, X., Bonada, J., Loscos, A., Serra, X.: chap. 10. In: *Spectral Processing* in *DAFx – Digital Audio Effects*. John Wiley & Sons (2002) 373–439, edited by Udo Zoelzer
15. Serra, X., Bonada, J.: *Sound Transformations Based on the SMS High Level Attributes*, Digital Audio Effects (DAFx) Workshop (1998) Barcelona, Spain.
16. Cano, P.: *Fundamental Frequency Estimation in the SMS Analysis*, Digital Audio Effects (DAFx) Workshop (1998) Barcelona, Spain.
17. Maher, R.C., Beauchamp, J.W.: *Fundamental Frequency Estimation of Musical Signals using a two-way Mismatch Procedure*. Journal of the Acoustical Society of America **95** (1994) 2254–2263
18. : *RWC Music Database – Musical Instrument Sound*. RWC–MDB–I–2001 No. 01–50, Tokyo, Japan. (2001)
19. Feller, W.: *Introduction to Probability Theory and its Applications*. 3rd edn. Wiley series in probability and mathematical statistics. John Wiley & Sons (1968)
20. Meyer, C.: 4. In: *Matrix Analysis and Applied Linear Algebra*. SIAM, Philadelphia, PA (2000)
21. Rodet, X., Depalle, P.: *Spectral Envelopes and Inverse FFT Synthesis*, 93rd AES Convention, San Francisco, AES Preprint No. 3393 (H-3) (1992)