SOUND SYNTHESIS BY PHYSICAL MODELING USING THE FUNCTIONAL TRANSFORMATION METHOD: EFFICIENT IMPLEMENTATIONS WITH POLYPHASE-FILTERBANKS

Stefan Petrausch, Rudolf Rabenstein

Multimedia Communications and Signal Processing
University of Erlangen-Nuremberg, Germany
stepe@lnt.de, rabe@lnt.de

ABSTRACT

The Functional Transformation Method (FTM) is a recently introduced method for sound synthesis by physical modeling. Based on integral transformations, it provides a parallel system description for any linear physical model, usually described by a set of partial differential equations. Such parallel descriptions can be directly implemented by a set of recursive systems in full rate. In this paper we present a new and very ef£cient method for this implementation which bene£ts from the spectral decomposition of the system. All recursive systems are working at a subsampled rate and are summed up by the application of a polyphase £lterbank. Performance measurements on a real time implementation show, that a ¤exible and ef£cient realization is achieved. Compared to the direct implementation it is over nine times faster at the cost of nine milliseconds of delay and even faster with more delay.

1. INTRODUCTION

Physical Modeling is the most recent sound synthesis technique, where not the sound itself but the sound production mechanism is modeled to achieve more natural and ¤exible sounds. In this context, the Functional Transformation Method (FTM) is a suitable method for the solution of any physical model described by Partial Differential Equations (PDEs). It starts with the mathematical description of the sounding object, in terms of a PDE with several initial and boundary conditions, and it provides a discrete implementation in parallel form, consisting of a set of £rst order conjugate complex recursive systems. For a detailed description of sound synthesis with the FTM see [1].

In this paper we will take advantage of the spectral analysis properties of the FTM. The recursive systems will be grouped together to band limited signals, from which we can generate the output function by a synthesis £lterbank. According to the theory of polyphase-£lterbanks (see [2]), it is suf£cient to work with subsampled versions of the band limited signals and therefore it is suf£cient to work with subsampled recursive systems.

As the subsampling of the complex modes, in contrast to subsampling of real modes, is a one-to-one mapping, the £lterbank can be directly applied to the subsampled complex recursive systems. A dual £lterbank approach with one additional shifted £lterbank avoids distortion for modes with frequencies near the band edges of the £lterbanks. The ef£cient implementation of the £lterbank by polyphase synthesis £lterbanks together with the subsampling of the recursive systems gives us a signi£cant reduction of the computational effort. We achieve a ¤exible and ef£cient implementation of physical models for real time sound synthesis.

The paper is organized as follows. In section 2 a short review of the FTM is given, which particularly describes the advantages of the FTM concerning the application of the £lterbank. Section 3 provides a short introduction to the principal properties of polyphase-£lterbanks along the lines of [2] and [3]. Conditions for the combination of the £lterbank and the FTM, and the actual implementation are described in section 4. Some details on the trade-off between delay, distortion and computational effort are given in section 5, supported by measurements on a real-time implementation in the audio application *Mustajuuri* (see [4]). Section 6 concludes this paper.

2. REVIEW OF THE FTM

The FTM provides a method for the solution of multi-dimensional systems. It is based on the transformation to the frequency domain in time and space. Its general procedure is depicted in £gure 1, for detailed information refer to [1] for instance.

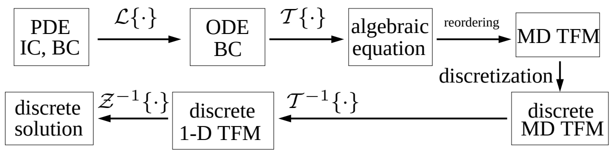

Figure 1: *General procedure of the FTM solving initial-boundary-value problems de£ned in form of PDEs and initial conditions (IC) and boundary conditions (BC). Further abbreviations are explained in the remainder of this section.*

2.1. Partial Differential Equations

The starting point is the description of the physical system in terms of several partial differential equations (PDEs) together with the systems initial (IC) and boundary conditions (BC). By introduction of several additional variables, one can always rewrite the PDEs by a system of £rst order equations, denoted by the vector description in equation (1).

$$\begin{aligned} [\mathrm{L} + \mathbf{C}\mathrm{D}_t]\, \mathbf{y}\,(\mathbf{x}, t) &= \mathbf{v}(\mathbf{x}, t), \quad \mathbf{x} \in V \\ \mathbf{f}_b^T \mathbf{y}(\mathbf{x}, t) &= \phi(\mathbf{x}, t), \quad \mathbf{x} \in \partial V \\ \mathbf{f}_i^T \mathbf{y}(\mathbf{x}, t)|_{t=0} &= \mathbf{y}_\mathrm{i}(\mathbf{x}), \quad \mathbf{x} \in V \end{aligned} \tag{1}$$

Thereby t is the scalar time, $\mathbf{x}$ denotes the vector of space coordinates de£ned within the spatial volume V, bounded by ∂V.

$\mathbf{y}(\mathbf{x}, t)$ indicates the vector of unknown quantities, $\mathbf{v}(\mathbf{x}, t)$ is a vector of excitations. The operator D_t denotes £rst order temporal derivative, while $\mathbf{C}$ is a mass or capacitance matrix. The spatial differentiation operator L is a concise notation for

$$\mathrm{L} = \mathbf{A} + \mathbf{B}\nabla \,, \tag{2}$$

where ∇ denotes £rst-order spatial derivative. The matrix $\mathbf{A}$ contains loss terms and matrix $\mathbf{B}$ combines all expressions with £rst-order derivatives.

The initial conditions are given by the vector operator $\mathbf{f}_\mathrm{i}$ and the initial values $\mathbf{y}_i$, and the boundary conditions are given by the vector operator $\mathbf{f}_b$ and the boundary functions ϕ.

2.2. Integral Transformations

The FTM adapts the well known idea of integral transformations for the solution of one-dimensional systems to multi-dimensional systems. So, to remove the temporal derivatives in equation (1) the Laplace transformation is used, which yields an Ordinary Differential Equation (ODE) with boundary conditions. The following Sturm-Liouville transformation (SLT, denoted by the operator $\mathcal{T}\{\cdot\}$ in £gure 1) has the same effect for the spatial derivatives.

It is adopted to the speci£c boundary-value problem and uses the eigenfunctions $\mathbf{K}(\mathbf{x}, \beta)$ of the operator L as transformation kernels. It transforms the ODE into a simple algebraic equation which can be reordered to obtain a multi-dimensional transfer function model (MD TFM).

$$\bar{Y}(\tilde{\beta}_\mu, s) = \frac{1}{s + \tilde{\beta}_\mu} \underbrace{\left[\bar{\mathbf{V}}(\tilde{\beta}_\mu, s) + \bar{\mathbf{y}}_\mathrm{i}(\tilde{\beta}_\mu) + \bar{\Phi}_\mathrm{b}(\tilde{\beta}_\mu, s)\right]}_{=\bar{F}_\mathrm{e}(\tilde{\beta}_\mu, s)} \tag{3}$$

Equation (3) displays this transfer function model. The bars denote the application of the SLT and upper case letters denote the application of the Laplace transformation. The spatial frequency variable $\tilde{\beta}$ is always restricted to several discrete values $\tilde{\beta}_\mu$ due to the boundary conditions.

2.3. Discretization and inverse Transformation

Application of the impulse-invariant transformation on (3) (see [5] for instance) leads to a discrete multi-dimensional transfer function model (discrete MD TFM) as depicted in £gure 1.

$$\bar{Y}^\mathrm{d}(\tilde{\beta}_\mu, z) = \frac{z}{z - e^{-\tilde{\beta}_\mu T}} \bar{F}_\mathrm{e}^\mathrm{d}(\tilde{\beta}_\mu, z) \tag{4}$$

The inverse SLT is simply the summation over all possible values of $\tilde{\beta}$ of the discrete MD TFM (see equation 4) weighted by the transformation kernel $\mathbf{K}(\mathbf{x}, \tilde{\beta}_\mu)$ and the norm factor of the transformation kernel N_μ. Together with the inverse Z-transformation we achieve a discrete solution of the physical system as written in equation (5).

$$y^\mathrm{d}(\mathbf{x}, k) = \sum_{\mu=0}^{\infty} \frac{1}{N_\mu} \mathbf{K}(\mathbf{x}, \tilde{\beta}_\mu) e^{\tilde{\beta}_\mu k T} * \bar{f}_\mathrm{e}^\mathrm{d}(\tilde{\beta}_\mu, k) \tag{5}$$

2.4. Discrete Realization

The discrete realization of equation (5) is obviously an in£nite set of £rst order recursive systems. However, for a given frequency range (normally from zero up to the sampling frequency) we always achieve a £nite set (N_h) of recursive systems, each representing one harmonic of the resulting sound. Figure 2 illustrates the implementation of equation (5) for the set of spatial points $\mathbf{x}_\mathbf{a}$.

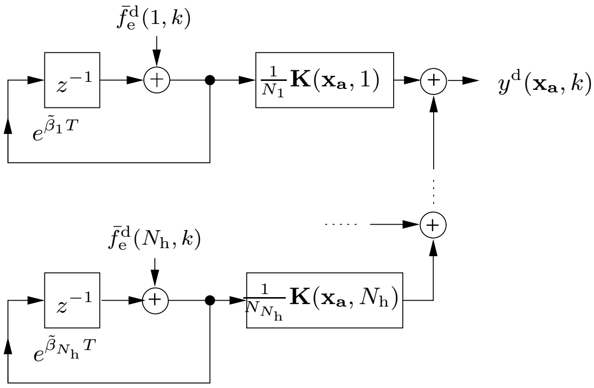

Figure 2: *Basic structure of the FTM simulations derived from vector PDEs with several complex £rst-order resonators in parallel.*

2.5. Related Methods with a Parallel System Description

At this point it is of interest to compare the FTM with another physical modeling technique, the modal analysis [6, 7, 8]. Modal synthesis can be interpreted as an extension of additive synthesis, i.e. a parallel arrangement of damped resonators. In additive synthesis, the frequencies and the damping coef£cients can be chosen completely arbitrary. In modal synthesis these parameters are derived from approximations to the physical behavior of vibrating structures. To reduce the number of vibrational modes in the model, a spatial discretization is used, where the number of spatial nodes corresponds to the required number of modes [6]. This spatial discretization yields only approximate values of the vibrational modes. Alternatively, also measurements from real-world musical instruments can be used to obtain the frequencies and damping coef£cients of the resonators.

In contrast to modal synthesis, the FTM uses a physical model of the vibrational body in the form of a PDE with appropriate initial and boundary conditions. It allows to calculate the exact values of an arbitrary number of vibrational modes. Furthermore, since no spatial discretization is involved, the coef£cients of the digital resonators are expressed directly by the physical parameters of the PDE model.

In short, the FTM shares with modal synthesis as well as with additive synthesis the basic structure of the discrete-time model, i.e. the well known parallel form realization of digital £lters [5]. FTM differs from the other synthesis methods in the determination of the model parameters. The application of suitable integral transformations to the underlying PDEs allows a direct expression of the model parameters in terms of the physical constants. For a more detailed comparison between FTM, modal synthesis, and other synthesis methods, see [1].

3. POLYPHASE-FILTERBANKS

As all harmonics in equation (5) are strictly band limited, it is easy to group them together to several (N in the sequel) band limited signals $y_n[k]$ with $0 \leq n < N$. Such signals represent the spectral analysis of the signal $y^{\mathrm{d}}(\mathbf{x}, k)$ ($y[k]$ in the sequel) and can be synthesized with a synthesis £lterbank. Ef£cient implementations of such £lterbanks are the well known polyphase £lterbanks (see [2] or [3]) which will be brie¤y described here.

3.1. Subsampled Filterbank

As the signals $y_n[k]$ are strictly band limited to a band width of $\Delta\Omega = \frac{2\pi}{N}$, it is save to subsample them with the factor N. As well known from signal theory (see [3] for instance), perfect reconstruction of the original signal is possible with suitable bandpass £lters $h_n[k]$.

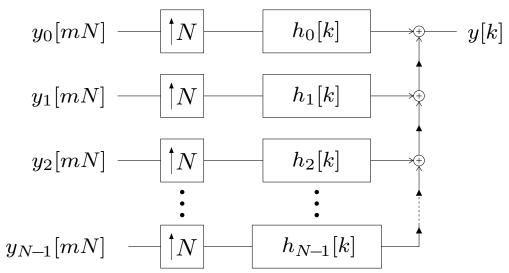

Figure 3: *Implementation of the £lterbank with an IDFT.*

Figure 3 depicts the basic idea of polyphase £lterbanks. The signals $y_n[k]$ are given in subsampled rate $m \cdot N$. They are upsampled with the factor N and afterward £ltered with the bandpass £lters $h_n[k]$ to achieve the original signals. The last step is the summation of all $y_n[k]$ to achieve the output function $y[k]$.

3.2. Ef£cient Implementation

If the bandpass £lters $h_n[k]$ are equidistant shifted versions of the so called prototype lowpass £lter $h_0[k]$, which means

$$h_n[k] = e^{jn\frac{2\pi}{N}} h_0[k] \ , \tag{6}$$

then we can replace the N full rate bandpass £lters in £gure 3 by one inverse discrete Fourier transformation (IDFT) and N £lters in subsampled rate.

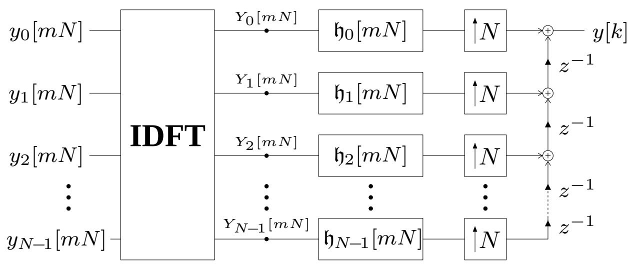

Figure 4: *Implementation of the synthesis £lterbank with an IDFT, £lters and a signal multiplexer. z^{-1} denotes delay by one sample.*

This much more ef£cient implementation of the £lterbank is depicted in £gure 4. The £lters $\mathfrak{h}_n[mN]$ are shifted and subsampled versions of the prototype lowpass £lter $h_0[k]$. They can be obtained by

$$\mathfrak{h}_n[mN] = h_0[mN + n] \ . \tag{7}$$

3.3. Analysis-Filterbank

To the synthesis £lterbank of £gure 4 corresponds an analysis £lterbank with a similar structure. Its components are the same as in £gure 4 but the order is reversed. The £rst step is the signal demultiplexer, realized by the delays and downsampling. The subsequent elements are the £lters $\mathfrak{h}_n[mN]$ with the same coef£cients as given in equation (7), and £nally the IDFT.

4. THE IMPLEMENTATION

A few adaptations have to be made to implement equation (5) with a polyphase £lterbank. These adaptations and further properties will be described in this section.

4.1. Complex versus Real Signals

As the FTM is a physical modeling method, we are always modeling real valued systems. The PDE in equation (1) is working for the real valued output $\mathbf{y}(\mathbf{x}, t)$. Therefore equation (5) has to be real too. Any complex eigenfrequency $\tilde{\beta}_\mu$ and thus any complex harmonic has its conjugate complex counterpart. Thus equation (5) describes a set of real harmonics.

However it is not possible to use the real harmonics in combination with a £lterbank; it is not even possible to use subsampled versions of the real harmonics. Although the computation of real harmonics is twice as effective, we have to use the complex ones.

If one tries to subsample a real harmonic whose frequency is near to any integer multiple of the new sampling frequency, it is no longer possible to separate frequency and amplitude correctly. The spectrum of the subsampled signal aliases with its conjugate complex part in such a manner that even destructive interference of the signal is possible (see [9] for another approach to avoid this problem).

Nevertheless, for computational ef£ciency it is not necessary to calculate the conjugate complex harmonics separately. The extraction of the real part of the £lterbank output has the same effect, as the computation of all harmonics together with their conjugate complex counterparts.

4.2. Dual-Filterbank Approach

As already mentioned in section 3.1, the £lterbank in £gure 4 is simply a set of equidistant shifted versions of the prototype lowpass £lter $h_0[k]$. However, as we have to use causal and linear phase systems this £lter must be a Finite Impulse Response (FIR) £lter. This £nite length (let it be $L \cdot N$) causes £nite transition bands and a £nite suppression of the mirrored harmonics. Therefore, as we do not want to restrict the frequencies of the harmonics, we use a dual £lterbank approach.

We de£ne valid zones (bounded by the dashed lines in £gure 5) for the harmonics, so that we can be sure, that any mirrored frequency (included between the dotted lines in £gure 5) is below a tolerated level. The non-valid zones in the *even* £lterbank are covered by the valid zones of the *odd* £lterbank (lower two frequency responses in £gure 5), which is just a shifted version of the even £lterbank. Working with complex harmonics this shifting causes no further problems, for real signals a separation of the conjugate complex part of the spectrum would be necessary.

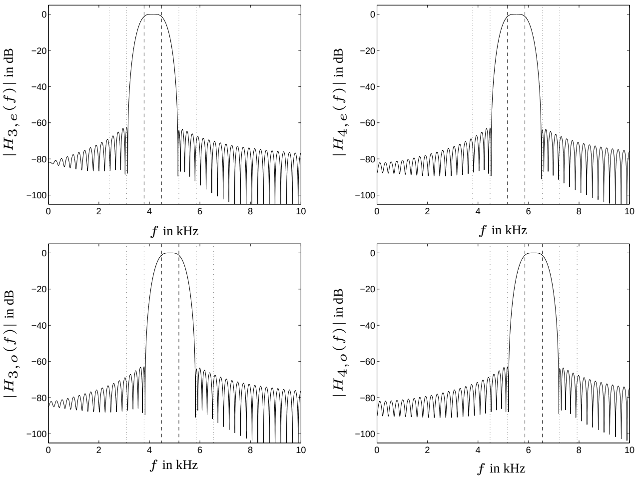

Figure 5: *Frequency response of bank number 3 (left) and 4 (right) of the even £lterbank (upper) and the odd £lterbank (lower) for $N = 32$, $L = 6$. The window function is the Kaiser window.*

4.3. Analysis- and Synthesis- Filterbank

Application of the polyphase synthesis £lterbank to the subsampled version of the FTM output in equation (5) poses the problem of aliasing for the exciting signal $\tilde{f}_{\mathrm{e}}^{\mathrm{d}}(\tilde{\beta}_\mu, k)$. The complex harmonics are strictly band limited, the exciting signal is not.

Therefore we have to create band limited excitation signals via the corresponding polyphase analysis £lterbank. An overview of the complete resulting system for the even £lterbank is depicted in £gure 6. We consider one input and one output function, nevertheless multiple inputs and multiple outputs are possible too.

The odd £lterbank system is nearly identical to £gure 6, except for the modulated £lter coef£cients, the exchange of the £ltering, and the evaluation of the real part in the synthesis £lterbank.

4.4. Filter design

Furthermore, we take a look at the design of the prototype lowpass £lter $h_0[k]$, as this £lter mainly determines the behavior of the £lterbank.

As already mentioned in section 4.2, we have to use a FIR £lters for the prototype lowpass. So, one of the main properties of the lowpass £lter is its £nite length. Since integer multiple of the £lterbank order N are useful (but not essential) for reasons of computational effort, we de£ne the length of the lowpass £lter $h_0[k]$ to $L \cdot N$.

Longer £lters are causing steeper transition bands and higher aliasing suppression, as can be seen in £gure 7 for $L = 4$ and $L = 8$. However longer £lters are also causing longer delays. For a lowpass £lter of length LN we achieve a delay of $\frac{LN}{2}$ samples.

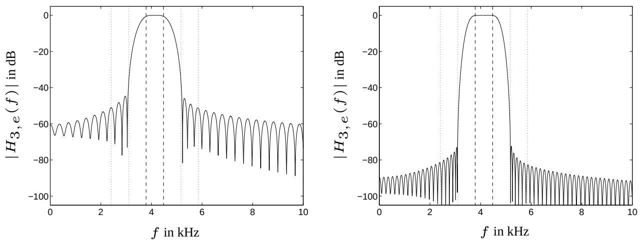

Figure 7: *Frequency response of band 3 of the even £lterbank with N=32 and $L = 4$ (left) resp. $L = 8$ (right).*

FIR £lters are usually truncated si-signals ($\frac{\sin x}{x}$) weighted by an appropriate window function. There are no restrictions for the choice of the window function. Selection criteria for this choice are the ¤atness of the valid passband (between the dashed lines in £gure 7) and the suppression of aliasing components in the stopband (between the doted lines in £gure 7). The behavior in between these bands is not important.

5. PERFORMANCE AND RESULTS

The performance of this implementation is a trade-off between computational effort, delay and distortion. Their interrelations and measurements at a real-time implementation are discussed in this section.

5.1. Delay, distortion and computational effort

Delay and distortion are directly related to the £lterbank. Convolution with a FIR lowpass £lter of length LN causes a delay of $\frac{LN}{2}$ samples. Since the complete implementation in £gure 6 consists of an analysis and a synthesis £lterbank a delay of $\delta_{\mathrm{t}} = LNT$ occurs, where T denotes the sampling interval.

As the downsampling produces images of the complex harmonics and as the length of the FIR £lters is £nite, we have to tolerate a minor distortion (see also £gure 9 in the next section). The intensity of the distortion is mainly determined by the £lter length multiplier L. A £lter length of $4N$ suppresses any image by at least $S_{\min} = 45\mathrm{dB}$, a £lter length of $8N$ by at least $S_{\min} = 75\mathrm{dB}$ (see £gure 7). This suppression is independent from the order N of the £lterbank.

The aim of the £lterbank implementation is the reduction of the computational cost. To estimate the computational cost we can calculate the number of needed multiplications per sample (MPS). For the direct implementation we need three real multiplications per real harmonic and sample,

$$\mathrm{MPS_d} = 3 \cdot N_{\mathrm{h}} \; . \tag{8}$$

The computational cost of the £lterbank implementation is divided into the implementation of the subsampled complex harmonics and into the constant computational effort for the £lterbank. The recursive systems themselves, that implement the real harmonics, need $6 \cdot N_{\mathrm{h}}$ real multiplications per subsampled time step. Both £lterbank implementations need two Fast Fourier Transformations (FFTs) and $2N$ £lter implementations of length L per subsampled time stamp. However, as the upper half of the DFTs input is zero, we can skip the £rst $\frac{N}{2}$ butter¤ies of the FFTs, so what we only need $\frac{N}{2}\,(\log_2 N - 1)$ complex multiplications per FFT. The

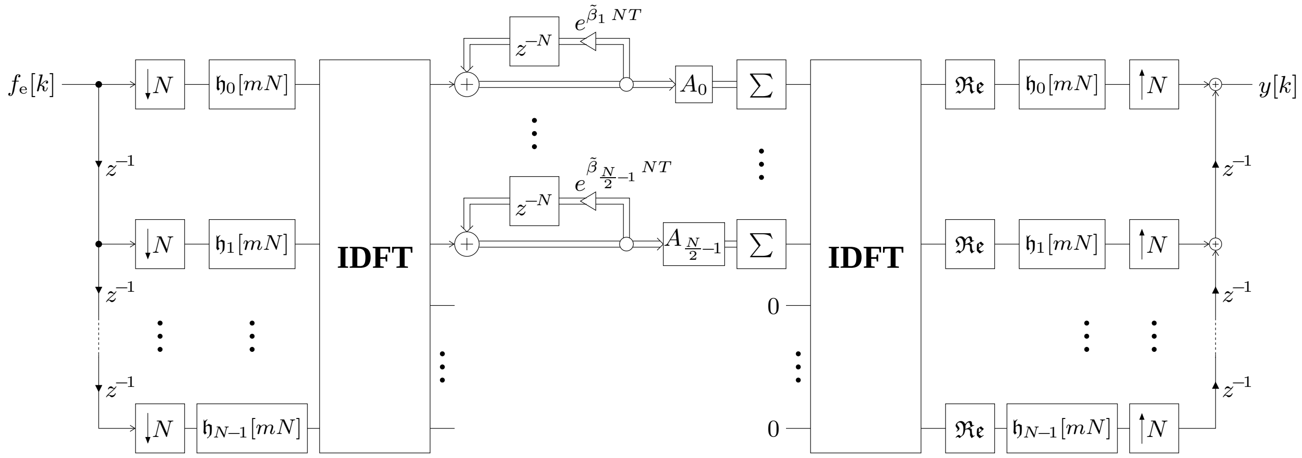

Figure 6: Overview of the complete even £lterbank system for excitation in one point with the excitation function $f_e[k]$ and one output function $y[k]$. Each channel includes a set of recursive systems, denoted by the thick lines. The weighting factors A_0 and $A_{\frac{N}{2}-1}$ include the excitation point dependency, the transformation kernel $\mathbf{K}(\mathbf{x}, \tilde{\beta}_\mu)$, and the norm factor N_μ (see section 2.3).

real analysis £lter implementations need L real multiplications per channel and time stamp. The synthesis £lters for the even £lterbank needs L real multiplications for each channel, as we only need the real part of its output. But the complex £lters of the odd £lterbank need $2L$ real multiplications. In summation the average and peak number of real Multiplications Per Sample $(\mathrm{MPS_f})$ for the complete system is calculated to

$$\mathrm{MPS_f} = 6\frac{N_\mathrm{h}}{N} + 8(\log_2 N - 1) + 5L\,. \qquad (9)$$

5.2. Measurements and Results

For the purpose of testing and veri£cation a real-time application was programmed. Both, the direct implementation as well as the £lterbank implementation of an oscillating membrane (see [10]) are realized as a plugin of the audio application *Mustajuuri* (see [4]).

In the £rst measurement, the number of membrane harmonics was reduced to one and the base pitch of the membrane was set to 200Hz for £gure 8 resp. 1kHz for £gure 9. Furthermore, the membrane was excited by an impulse, so that a sine wave was achieved. Figure 8 compares the temporal behavior of the resulting sine wave. For the direct implementation, the synthesized sine wave with a frequency of 200Hz starts immediately after the exciting impulse at $t = 0$. The £lterbank implementation with $N = 32$ channels and a £lter length multiplier of $L = 4$ is delayed by $\delta_\mathrm{t} = 5.8\mathrm{ms}$ (sampling interval $T = 1/44100s$), what is in perfect line with the results in section 5.1.

The next £gure illustrates the minimum suppression S_min of the aliasing components. As it can be seen in £gure 9 subsampling causes $\frac{N}{2} - 1 = 15$ images of the original 1kHz sine wave. For $L = 4$ the minimum suppression S_min is 45dB and for $L = 6$ it is 65dB, which is suf£cient for most listening conditions.

To conclude the presentation of the £lterbanks performance, the simulation of the lowest guitar string with about $N_\mathrm{h} = 268$ audible real harmonics was assumed and tested. The following table shows the trade-off between delay δ_t, distortion (denoted by the minimum aliasing suppression S_min in dB) and computational ef£ciency. The computational ef£ciency is denoted by G,

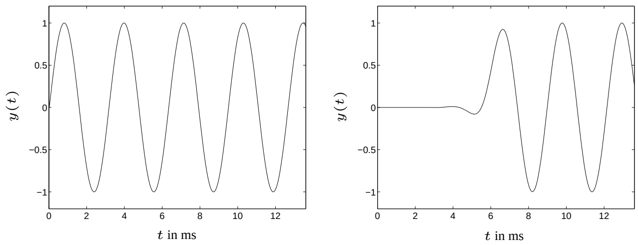

Figure 8: *Sine signal with a frequency of 200Hz, realized directly (left) and with a 32 channel polyphase £lterbank and $L = 8$ (right).*

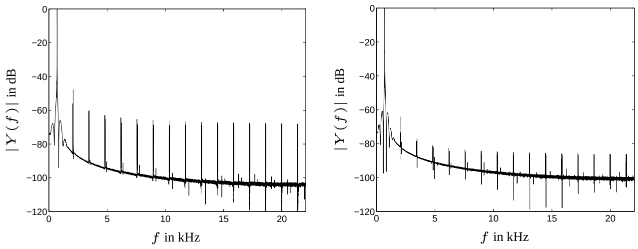

Figure 9: *Spectrum of a sine with a frequency of 1kHz, realized with a $N = 32$ channel polyphase £lterbank for $L = 4$ (left) resp. $L = 6$ (right).*

the quotient of the computational cost for the direct implementation divided by the computational cost for the £lterbank implementation. It is given as a theoretically achievable maximum value $G_\mathrm{max} = \frac{\mathrm{MPS_d}}{\mathrm{MPS_f}}$ and the value G_m which has been measured directly on the real-time implementation. To calculate G_max, the number of audible harmonics N_h (268 for the lowest guitar string), and the number of channels N, and the £lterlength multiplier L was inserted in equation (9) resp. equation (8) to achieve the number of real multiplications per sample (also included in the table). For the measurement of G_m, the number of membrane harmonics was simply set to 268.

parameters		performance				
N	L	δ_t	S_min	MPS	G_max	G_m
direct		0.02 ms	120 dB	804	1	1
16	4	1.45 ms	45 dB	145	5.5	3
16	6	2.18 ms	65 dB	155	5.2	2.8
32	4	2.9 ms	45 dB	103	7.8	5.4
32	6	4.35 ms	65 dB	113	7.1	5.2
32	8	5.8 ms	75 dB	123	6.5	5
64	4	5.8 ms	45 dB	86	9.3	9.1
64	6	8.7 ms	65 dB	87	9.2	9.1

The real-time implementation of the £lterbank is obviously slowed down by a non-negligible amount of overhead, which is not included in equation (9). Nethertheless, for large numbers of harmonics we achieve even more computational ef£ciency. We were able to simulate the membrane with over 10000 harmonics at the cost of 8.7ms of delay and non audible distortion (parameters $N = 64$, $L = 6$).

6. CONCLUSIONS

This paper introduced a new approach for the ef£cient implementation of sound synthesis by physical modeling. It described the solution of partial differential equations with the FTM and gave a polyphase £lterbank implementation of this solution. Measurements at a real-time implementation of an oscillating membrane have shown the practical relevance of this method and its advantages. The implementation with polyphase £lterbanks gives the system designer more freedom, for trade off between distortion, delay and computational effort. So it is possible to calculate spatial two-dimensional systems with over 10000 harmonics with non audible distortion ($< 65\mathrm{dB}$) and non audible delay ($< 10\mathrm{ms}$) in real-time.

This research has been funded by the European Commission within the IST-project ALMA (Algorithms for the Modelling of Acoustic interactions).

7. REFERENCES

[1] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling using Functional Transformation Models*, Kluwer Academic Publishers, New York, 2003.

[2] N. Fliege, *Multiraten-Signalverarbeitung*, B.G. Teubner, Stuttgart, 1993.

[3] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.

[4] Tommi Ilmonen, "Mustajuuri (Audio Application and Toolkit)," http://www.tml.hut.£/ tilmonen/mustajuuri/, 2002.

[5] Ronald W. Schafer Alan V. Oppenheim, *Digital Signal Processing*, Prentice Hall, Englewood Cliffs, 1975.

[6] J.M. Adrien, "Dynamic modeling of vibrating structures for sound synthesis, modal synthesis," in *Proc. AES 7th International Conf.*, Toronto, Canada, 1989, pp. 291–299, Audio Engineering Society.

[7] G. De Poli, A. Piccialli, and C. Roads, Eds., *Representations of Musical Signals*, The MIT Press, Cambridge, Massachusetts, USA, 1991.

[8] C. Roads, S. Pope, A. Piccialli, and G. De Poli, Eds., *Musical Signal Processing*, Swets & Zeitlinger, Lisse, 1997.

[9] Lutz Trautmann and Vesa Välimäki, "A Multirate Approach to Physical Modeling Synthesis using the Functional Transformation Method," in *IEEE Proc. Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, New York, Oct. 2003.

[10] Lutz Trautmann, Stefan Petrausch, and Rudolf Rabenstein, "Physical modeling of drums by transfer function methods," in *IEEE Proc. Int. Conf. on Acoustics, Speech & Signal Processing (ICASSP)*, Salt Lake City, Utah, May 2001.