Publication V

Heidi-Maria Lehtonen, Jukka Rauhala, and Vesa Välimäki. 2005. Sparse multi-stage loss filter design for waveguide piano synthesis. In: Proceedings of the 2005 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2005). New Paltz, NY, USA. 16-19 October 2005. Pages 331-334. ISBN 0-7803-9154-3.

SPARSE MULTI-STAGE LOSS FILTER DESIGN FOR WAVEGUIDE PIANO SYNTHESIS

*Heidi-Maria Lehtonen, Jukka Rauhala, Vesa Välimäki**

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
P.O. Box 3000, FI-02015 TKK, Espoo, Finland
hml@acoustics.hut.fi

ABSTRACT

A new filter structure and a design method are proposed for the loss filter that is used in digital waveguide synthesis. The main application of this work is the sound synthesis of the piano, but the methods are also applicable to the synthesis of other struck or plucked string instruments. The filter structure is an extension of a sparse FIR filter called the ripple filter, which has been proposed previously for waveguide synthesis of keyboard instruments. The new structure is based on a cascade of sparse FIR filters, which are designed one after the other on subbands that are integer fractions of the audio range. We show by examples that a cascade of three digital filters provides possibilities to exactly match the decay rate of a finite number, such as 50, lowest-order partials, or to approximately match the general trend and some variations of many partials of a piano tone. The subfilters can easily be designed using standard techniques.

1. INTRODUCTION

Physical modeling of musical instruments has been a popular research field over the past decade. Digital waveguide modeling is an efficient way to implement physics-based synthesis of string instruments, such as the guitar and the piano. This technique originates from the idea of discretizing the traveling-wave solution of the wave equation [1]. The transfer function of the synthesis model can be written as $1/[1 - z^{-L}H(z)]$, where L is the delay line length and the filter $H(z)$ consists of three different filters: a fractional delay filter, which is responsible for the accurate tuning of the string, the dispersion filter, which models the inharmonicity, and the loss filter, which determines the decay rate of the partials. In the case of the piano, which has a characteristic inharmonic and complex sound, the two latter filters are particularly important. In this paper, we concentrate on the design of the loss filter for piano synthesis.

Usually, the loss filter is a low-order FIR or IIR filter [2, 3]. For good accuracy, a high-order filter is needed because the variations of the magnitude response from one partial to the next one cannot be matched with a low-order filter. A combination of a one-pole filter and an FIR comb filter called the ripple filter has been proposed as a reduced-complexity loss filter [4]. The main idea was to first design the one-pole filter to imitate the general trend of the specified magnitude response, and then to use the comb filter to accurately tune the loop gain (and decay rate) of a selected partial. We have recently extended this concept by adding more than one feedforward path in cascade with a one-pole filter. This is called the multi-ripple filter [5]. A custom-made design algorithm, which appropriately smooths and simplifies the magnitude response specification, enables modeling the main features of the decay pattern of piano tones. An accuracy that otherwise calls for an FIR filter with a few hundred coefficients can be obtained with a cascade of one-pole and multi-ripple filters that has the computational complexity of a fifth-order FIR filter [5].

In loss filter design, a major problem is that the desired magnitude response is usually specified on a very narrow frequency band. For example, for the lowest keys of the piano, the partials below 2 kHz affect the sound most and it is difficult to accurately estimate the decay rate of high-frequency partials. However, the sampling rate used in high-quality piano synthesis is usually 44.1 kHz. The traditional filter design methods cannot face the challenge that the important frequency band is only 10% of the audio range.

The method presented in this paper facilitates the loss filter design problem, especially when the fundamental frequency of the tone is low. Its benefit is that it is easy to use, since standard filter design techniques are employed on a limited frequency band. The method is related to the frequency sampling [6] and the IFIR filter techniques [7]. The loss filter is designed at a reduced sampling rate and it is up-sampled for implementation. This paper extends our multi-ripple loss filter design technique [5]. The major differences are that the new filter structure consists of a cascade of three or more subfilters instead of two, and that standard filter design tools can be applied, although the resulting subfilters are non-standard, sparse filters with many zero coefficients. This paper presents the general method, a specific filter structure and a design method for piano tones, and design examples for a selected piano tone. The obtained results are compared with the existing methods.

2. METHOD

The loss filter presented in [5] consists of two cascaded subfilters. A first-order all-pole filter takes care of modeling the general trend of the piano tone's decay rate and an N^{th}-order

* This work was funded by the Academy of Finland (project no. 104934).

0-7803-9154-3/05/$20.00 ©2005 IEEE

feedforward comb filter models the decay rate variations from one partial to the next one. The principal difference in the loss filter structure presented here is that in addition to the parallel feedforward structure it is possible to insert feedforward blocks in cascade. Also the design of the loss filter differs from the technique presented in [5].

For waveguide synthesis of piano tones, we propose a structure that has three subfilters: an equalizer, an anti-imaging filter, and a multi-ripple filter. The equalizer is responsible for the general trend of the piano tone's decay, the anti-imaging filter attenuates the image frequency responses caused by up-sampling, and the multi-ripple filter is designed to fit the data as well as possible at low frequencies.

The purpose of the equalizer and the anti-imaging filter is basically the same as the anti-imaging filter in the IFIR technique. In the IFIR technique, it is important to eliminate the image frequency responses that result from the compression of the frequency response. In this case, it is not desired to completely get rid of the image frequency responses, since there are partials on higher frequencies as well. Instead, we need to be sure that these partials do not dominate the sound but are appropriately attenuated.

The block diagram of the N^{th}-order loss filter is shown in Fig. 1. The blocks H_1, H_2 and H_3 are the equalizer, the anti-imaging filter, and the multi-ripple filter, respectively. The system of Fig. 1 can be implemented in a time-reversed form, since then the delay blocks of the anti-imaging filter and the multi-ripple filter can share delay elements with the long delay line of length L in the waveguide model.

2.1. Equalizer Design

The equalizer is designed as a first-order FIR filter in a way that the magnitude response matches two given data points, such as those at the fundamental frequency and at the Nyquist frequency. The result should imitate the general trend of the piano tone's decay rate in the band 0 – 22.05 kHz. The phase is not linear, but since the changes in phase delay response are not significant, about 0.05% of the delay line length L with practical parameter values, this does not cause any problems. The dashed line in Fig. 2 presents the magnitude response of the equalizer.

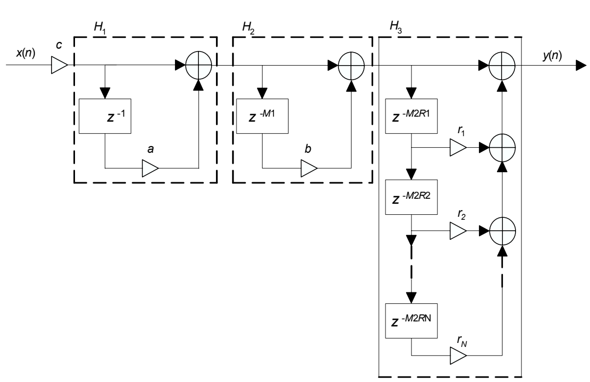

Figure 1: *Block diagram of the N^{th}-order loss filter consisting of three subfilters: H_1 is the equalizer, H_2 is the anti-imaging filter, and H_3 is the multi-ripple filter.*

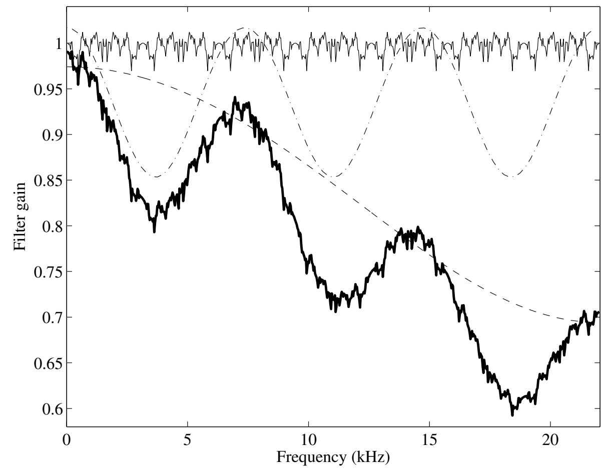

Figure 2: *The magnitude responses of the subfilters presented in Fig. 1: the equalizer (dashed line), the anti-imaging filter with M_1 = 6 (dash-dotted line), the multi-ripple filter with M_2 = 14 (solid line), and the resulting loss filter (thick line).*

2.2. Anti-Imaging Filter Design

The anti-imaging filter has two important tasks. Firstly, it takes care of attenuating the resulting image frequency responses and secondly, it can be used for easing the modeling of the important frequency band.

The filter is designed on a reduced frequency band in the same manner as the equalizer. It is important that the up-sampling factor M_1 is selected in a way that the anti-imaging filter attenuates the frequency band below 5 kHz appropriately. We have found that that factor 6 is sufficient for the lowest piano tones (key index 1-10) whereas for the key indices about 10-20 factor 2 works well. In the middle range, where the harmonics cover a larger portion of the audio range, the anti-imaging filter is not essential. The resulting magnitude response is presented in Fig. 2 with a dash-dotted line.

2.3. Multi-Ripple Filter Design

The multi-ripple filter design process presented here results in the same multi-ripple filter structure as presented in [5]. On the other hand, the design processes are somewhat different since the design method presented here uses standard filter design techniques. The purpose of the loss filter is to model the decay rate of the partials accurately. The desired gain values can be determined from a real piano sound since the relation between the decay time constant of the kth partial and the corresponding filter gain value g_k can be expressed as a closed-form formula: $g_k = \exp[-1/(f_0\tau_k)]$, where f_0 is the fundamental frequency, and τ_k is the decay time constant.

The general trend of the obtained gain is decreasing at higher frequencies. This makes the task harder for the filter design algorithms. However, this problem can be made easier by multiplying the gain values with the inverse magnitude response values of the equalizer and the anti-imaging filter. Hence, the gain values are around one. This effect is compensated at the end with the equalizer and the anti-imaging filter.

The problem that the data covers only a small part of the audio range can be overcome by critical down-sampling. When the frequency of the highest specified partial is f_{max}, the up-sampling factor for critical down-sampling is M_2 = floor[44100/($2f_{max}$)] and the new sampling rate is f_s = 44100/M_2. The actual design method is based on frequency sampling [6]. First the impulse response (and thus the corresponding FIR filter coefficients) is obtained from the magnitude response by inverse discrete Fourier transform. After this, the N largest values are chosen and the other coefficients are set to zero. When we choose N to be the exact number of the partials, we can design a filter which models the given frequency response perfectly. When the impulse response is up-sampled, the result is a sparse FIR filter. In the frequency domain, the up-sampling means that $M_2 - 1$ image frequency responses follow the original frequency response. The magnitude responses of all three subfilters in the case of the exact fit at the lowest frequencies are presented in Fig. 2.

In the case of low filter orders, the fit in the data cannot be perfect, though, the N largest values of the impulse response fit the obtained magnitude response to the data best in the least-square sense. In the loss filter design, it is usually desired that especially the highest peaks are modeled accurately, since the gain values near the value one have the longest decay times. This fact can be taken into account in the design by emphasizing the largest gain values with a weighting function presented by Bank and Välimäki in [3].

3. DESIGN EXAMPLES

In the following, two design examples are presented. The first one describes how to do an exact fit and the second example is for the filter order 5. In both cases the filter is designed for the key index 3 (f_0 = 30.8677 Hz) with 50 lowest-order partials.

3.1. Perfect Match of 50 Partials

In the loss filter design, the case of an exact fit to the data has been considered particularly problematic. Laroche and Meillier [8] have presented a method for designing a perfect match: When all harmonics are measured correctly, the loss filter can be implemented so that every partial is modeled with its own resonator. The computational cost of the implementation becomes large at low fundamental frequencies, when there are many partials to be modeled. The approach taken here is somewhat different and it leads to a computationally more efficient implementation.

The first phase of the design process is to determine the new, reduced sampling rate. When the target is to model 50 lowest-order partials accurately for the key index 3, the highest frequency in the data is 1543 Hz. This means that the up-sampling factor M_2 is chosen to be 14 and the new sampling rate used during the design is 44100 Hz/14 = 3150 Hz.

The second phase is to design the equalizer and the anti-imaging filter. It is done in the same way as in Sections 2.1 and 2.2. The up-sampling factor M_1 for the anti-imaging filter is chosen to be 6.

In the next phase, inverse DFT is applied and the obtained impulse response is made minimum phase with Matlab's 'rceps' function [9]. After this, 50 largest impulse response values are selected to be the FIR filter coefficients. At the end, the impulse response is up-sampled by factor 14 and a sparse FIR filter is obtained.

In Fig. 3(a), it is seen that the resulted magnitude response (solid line) follows exactly the gain specification (dots). Fig. 3(b) presents the corresponding T_{60}-times, i.e. the time it takes for each harmonic to decay 60 dB. It can also be seen that in the high frequencies, around 7 kHz, the T_{60}-time is somewhat longer than in the surroundings, resulting from the effect of the anti-imaging filter. This is not a problem, since the T_{60}-time is only about 3 seconds, and thus it does not have a significant effect on the resulting sound. Without the anti-imaging filter, there would be large values around 3 kHz in the T_{60}-domain.

3.2. Low-Order Sparse Filter Design

When the selected filter order is low, some compromises in the fitting must be done. One solution is to smooth the original data so that a few important points are preserved. Smoothing the data simplifies the design task significantly as most of the minor details are ignored. In the following, the data is smoothed in the same way as in [5]: The amplitude maxima, loop gain maxima, and local maxima and minima are chosen as the points to be preserved, and these points are connected with a smooth polynomial function, which can be obtained by Matlab's 'polyfit', 'polyder', and 'interp1' functions [9].

After applying the inverse DFT, the four largest impulse response values are selected. These values can be optimized with the sparse weighted least squares technique presented in [10]. The optimized filter coefficients are calculated with $\boldsymbol{h} = (\boldsymbol{U}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{U})^{-1}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{H}$ [6], where $\boldsymbol{U}$ is the DFT matrix (see e.g. [11]), $\boldsymbol{W}$ is a diagonal weighting matrix whose kth diagonal element is w_k, and $\boldsymbol{H}$ is the target frequency response vector. The matrix $\boldsymbol{U}$ is modified so that only the elements corresponding to the selected impulse response values are nonzero [10]. The weighting function is the one proposed in [3], and it can be written as $w_k = (g_k - 1)^{-4}$.

An example design is presented in Fig. 4 with a thick line. The up-sampling factors, the equalizer, and the anti-imaging filters of the previous design example are used, because the gain specifications are again for the key index 3.

4. RESULTS AND COMPARISON

In the following, the proposed loss filter structure is compared to existing methods. The compared filters are our previous loss filter [5] of order 5, and an IIR filter with a first-order denominator and a numerator of order 201. The latter filter was designed using Matlabs 'invfreqz' function with the weighting function presented in [3]. The data in this example is the same that has been used in Sections 3.1 and 3.2. Fig. 4(a) shows the resulted magnitude responses and in Fig. 4(b) the results are presented in T_{60}-domain.

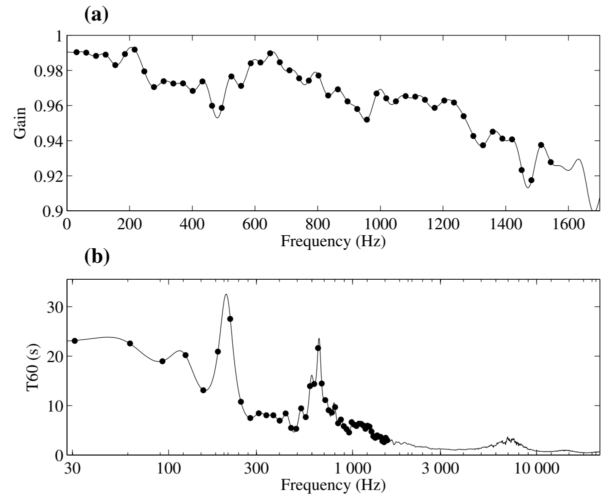

Figure 3: *(a) The gain specifications (dots) and the magnitude response (solid line) of the designed filter and (b) the corresponding reverberation time as a function of log frequency.*

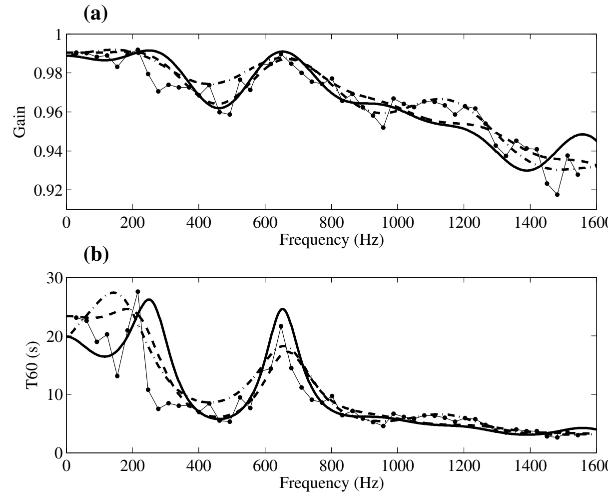

Figure 4: *(a) The gain specification (dots) and the results of three design examples: low-order loss filter presented here (thick), low-order loss filter presented in [5] (dashed), and high-order conventional filter (dash-dotted). (b) The corresponding reverberation time.*

The characteristics of the proposed loss filter are shown in Fig. 4 with a thick line. It can be seen that that the filter magnitude response follows the gain specification quite accurately at frequencies below 800 Hz. Also the highest peaks are modeled well. In the T_{60}-domain, which is more interesting from the perceptual point of view, the results are also quite sufficient.

The loss filter presented in [5] did also well in the comparison. Its magnitude response is presented in Fig. 4(a) with a dashed line. The high-order IIR filter is also practically as good as the two other filters presented, but the order of the filter is 40 times larger.

Obviously, the two loss filters presented here and in [5] are better for instrument synthesis purpose, since their computational cost is significantly smaller than the cost of the high-order filter with the same performance.

The differences of the proposed filter and the filter presented in [5] are minor when it comes to accuracy of the modeling. The computational costs are equally large. The major difference and improvement lies in the design. Since the filter proposed in this paper uses standard filter design techniques, it is somewhat easier to design.

5. CONCLUSIONS

A new method of the loss filter design for piano synthesis purposes was presented. The proposed filter structure is based on a cascade of sparse FIR filters, which are designed one after the other on subbands and are then up-sampled for implementation. The strengths of this method are its simplicity and good performance even with low filter orders. It is also possible to design a filter, which models the given gain specification perfectly. In comparison against a traditional, non-sparse filter, the proposed loss filter performs well with a significantly smaller computational cost.

6. REFERENCES

[1] J. O. Smith, “Physical modeling using digital waveguides,” *Computer Music J.*, vol. 16, no. 4, pp. 74–91, 1992.

[2] J. O. Smith III, “Techniques for Digital Filter Design and System Identification with Application to the Violin,” Ph.D. thesis, Dept. of Music, Stanford Univ., Stanford, CA, 1983.

[3] B. Bank and V. Välimäki, “Robust loss filter design for digital waveguide synthesis,” *IEEE Signal Processing Letters*, vol. 10, no. 1, pp. 18-20, 2003.

[4] V. Välimäki, H. Penttinen, J. Knif, M. Laurson, and C. Erkut, “Sound synthesis of the harpsichord using a computationally efficient physical model,” *EURASIP J. Applied Signal Processing*, vol. 4, no. 7, pp. 934–948, 2004.

[5] J. Rauhala, H.–M. Lehtonen, and V. Välimäki, “Multi-ripple loss filter for waveguide piano synthesis,” in *Proc. Int. Computer Music Conf.*, Barcelona, Spain, 2005.

[6] T. W. Parks and C. S. Burrus, *Digital Filter Design*. John Wiley & Sons, Inc., 1987.

[7] Y. Neuvo, C.-Y. Dong, and S. K. Mitra, “Interpolated finite impulse response filters,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 32, no. 3, pp. 563–570, 1984.

[8] J. Laroche and J.-L. Meillier, “Multichannel excitation/filter modeling of percussive sounds with application to the piano,” *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 329–344, 1994.

[9] Matlab version 7.0, The Mathworks Inc., 1984-2004.

[10] A. Tarczynski and V. Välimäki, “Modifying FIR and IIR filters for processing signals with lost samples,” in *Proc. IEEE Nordic Signal Processing Symp.*, Espoo, Finland, September 1996, pp. 359–362.

[11] S. K. Mitra, *Digital Signal Processing, a Computer-Based Approach*. McGraw-Hill, 2002.