DESIGN OF AN EFFICIENT INHARMONIC DIGITAL WAVEGUIDE FILTER FOR SYNTHESIS OF HAND-BELL SOUNDS

Paulo A. A. Esquef and Vesa Välimäki

Helsinki University of Technology – Laboratory of Acoustics and Audio Signal Processing
P.O.Box 3000, FIN-02015 HUT, Espoo, Finland
{paulo.esquef,vesa.valimaki}@hut.fi

ABSTRACT

This paper addresses the use of an inharmonic digital waveguide (IDWG) filter as an efficient means to synthesize hand-bell sounds. The IDWG is obtained by inserting a second-order allpass filter to the feedback path of a waveguide string model. An automated iterative procedure for estimation of the allpass filter coefficients and the length of the delay line is proposed. Limitations of the devised design method as well as possible extensions are also discussed.

1. INTRODUCTION

The characteristic sound of bells is dominated basically by two features: inharmonicity and the presence of beating on the amplitude evolution of the tone partials. In particular, the beating of partials indicates that some partials are composed of two or more resonant modes, whose frequencies are very close to each other (see Table 1 and Fig. 6).

Synthesis of inharmonic sounds, such as those produced by stiff strings or bars, has been investigated before [1, 2, 3]. Following the same rationale of [1, 2] we have proposed in [4] using an inharmonic digital waveguide (IDWG) filter to synthesize hand-bell sounds. As shown in Figure 1, an IDWG can be achieved by inserting an allpass filter into the feedback path of a standard digital waveguide (DWG) to account for the non-linear phase characteristic needed for generating inharmonic sounds.

In [4], the coefficients of a second-order allpass filter and the length of the delay line were manually adjusted to match the first resonances of the IDWG filter to the frequencies of the lower partials of the featured bell sound. For psychoacoustic reasons only the frequencies of the first three partials were matched. Aiming at the same goal, in this paper we present an automated procedure to estimate both the coefficients of the allpass filter and the length of the delay line.

2. ANALYSIS OF BELL SOUNDS

The estimation of the frequencies and decay times of the partial modes of the bell sound can be performed through the so-called frequency-zooming autoregressive moving-average (FZ-ARMA) analysis, described in [5].

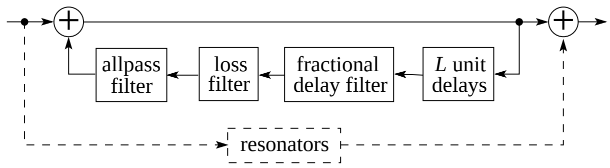

Figure 1. Inharmonic waveguide plus resonators.

The first step of the procedure requires spectral analysis of the bell sound to roughly determine the frequencies of its prominent spectral peaks or partials. The idea behind the frequency-zooming modeling is to analyze each partial of the tone separately. To accomplish that for a given partial of frequency f_i, the following steps are taken:

1. Down-modulate original signal $s(n)$ to place the spectral peak to be analyzed around DC (zero frequency). That implies a modulated complex-valued signal $s_1(n) = \exp(-j\Omega_{\rm m} n)s(n)$, where $\Omega_{\rm m} = 2\pi f_i/f_{\rm s}$ and $f_{\rm s}$ is the sample frequency.
2. Lowpass filter signal $s_1(n)$ to isolate this particular partial, i.e., $s_2(n) = s_1(n) * h_{\rm lp}(n)$. The bandwidth of the lowpass filter must be narrower than twice the minimum frequency difference between two consecutive partials.
3. Maximally decimate signal $s_2(n)$.
4. Fit a low-order complex-valued ARMA model to the complex-valued decimated signal.

From the poles of the estimated model one can retrieve the frequencies and decay times of the resonant modes that are present in the analyzed partial. More specifically, the frequencies are related to the angles of the poles whereas the decay times to the pole radii. More details on the FZ-ARMA analysis can be found in [5]. The advantage of the FZ-ARMA analysis is that it provides means to solve resonance frequencies that occur very close to each other, as it happens with the resonance frequencies pertaining to separate bell partials.

As an example, the sound waveform of a hand-bell, sampled at 44.1 kHz, is analyzed via the FZ-ARMA modeling scheme [5]. The decimated complex-valued signals

The work of Paulo A. A. Esquef has been supported by a scholarship from the Brazilian National Council for Scientific and Technological Development (CNPq-Brazil) and by the ALMA (IST-2001-33059) project.

	Mode 1		Mode 2	
Partial #	Freq./Hz	T_{60}/s	Freq./Hz	T_{60}/s
1	1312.0	8.02	1314.5	7.04
2	2353.3	3.93	2362.9	4.53
3	3306.5	1.94	3309.4	0.30

Table 1. Mode parameters of the first three partials.

associated with each one of the first three partials are modeled through $\mathrm{ARMA}(2,4)$, i.e. two complex poles and four complex zeros. From the pole locations of the models we obtained the frequencies and decay times of the modes associated with first three partials (two modes per partial in this case). The estimated values are summarized in Table 1.

3. INHARMONIC DIGITAL WAVEGUIDE

The structure of the inharmonic waveguide is depicted in Figure 1. The optional resonators used to simulate beating between partial modes are drawn with dashed lines.

3.1. Effect of the allpass filter

To start with, we consider a simple DWG model without the allpass and fractional-delay filters, and in which the loss filter is reduced to a simple gain factor $g < 1$. The phase response of the feedback path (open loop) of this DWG is given by

$$\phi_{\mathrm{ol,DWG}}(\omega) = -L\omega, \qquad (1)$$

where ω is the normalized angular frequency and L is the integer length of the delay line. Such DWG will have L resonances whose frequencies are equally spaced on the frequency range from 0 to 2π. Indeed, the resonance frequencies ω_i occur at the values of $\phi_{\mathrm{ol,DWG}}(\omega_i) = -2\pi i$, for $i = 0, 1, \ldots, (L-1)$.

Now, if we insert an Nth-order allpass filter into the feedback path of the previous DWG filter, the phase response of the allpass filter will add a non-linear contribution to the phase response of the open loop, i.e.,

$$\phi_{\mathrm{ol,IDWG}}(\omega) = -L\omega + \phi_{\mathrm{ap}}(\omega), \qquad (2)$$

where $\phi_{\mathrm{ap}}(\omega)$ is the phase response of the allpass filter, whose transfer function is given by

$$H_{\mathrm{ap}}(z) = \frac{a_N + \cdots + a_1 z^{-N+1} + a_0 z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}. \qquad (3)$$

The phase response of an Nth-order allpass filter can be written as

$$\phi_{\mathrm{ap}}(\omega) = -N\omega + 2\arctan\left\{ \frac{\sum_{k=1}^{N} a_k \sin(k\omega)}{1 + \sum_{k=1}^{N} a_k \cos(k\omega)} \right\}, \qquad (4)$$

where a_k are the coefficients of the allpass filter, constrained to $a_0 = 1$.

The resonances of the IDWG system still appear at $\phi_{\mathrm{ol,IDWG}}(\omega_i) = -2\pi i$. However, with the allpass filter included, the phase response of the open loop becomes non-linear. As a result, the resonances no longer appear equally spaced in frequency, leading to inharmonicity.

Experiments have been conducted in [4] using second-order allpass filters. The results revealed that the effect of the allpass filter in the open loop consists of attracting the partials toward the frequency of the pole of the allpass filter. The attraction power increases with the radius of the pole of the allpass filter and is stronger near the location of the allpass pole.

The presence of the allpass filter inside the open loop changes not only the frequency of the poles of the DWG filter but also their radius. In fact, the pole radii r_i of the IDWG system depend on both the gain $g(\omega)$ and the group delay $\tau_{\mathrm{ol,IDWG}}(\omega)$ of the open loop, evaluated at the resonance frequencies ω_i of the system, i.e.,

$$r_i = [g(\omega_i)]^{1/\tau_{\mathrm{ol,IDWG}}(\omega_i)}. \qquad (5)$$

4. DESIGN PROCEDURE

4.1. Allpass filter

The phase response of an IDWG, containing only a delay line and an allpass filter in the feedback path, depends on the length of the delay line L and the coefficients of the allpass filter a_k. For a second-order allpass filter we have to estimate two coefficients, since $a_0 = 1$. Besides, there is an additional degree of freedom given by the choice of L. Therefore, it seems reasonable that, within certain limits, it is possible to match the frequencies of three partials using such a system.

If L is not to be estimated and a set of $G = N$ specification values for the desired phase is provided, it is possible to construct a set of G linear equations and solve for a_k. This equation system is given by [6]

$$\sum_{k=1}^{N} a_k \sin(\beta_i + k\omega_i) = -\sin\beta_i, \qquad (6)$$

with

$$\beta_i = -\frac{1}{2}\left[\Phi_{\mathrm{loop,spec}}(\omega_i) + N\omega_i + L\omega_i\right], \qquad (7)$$

where $\Phi_{\mathrm{loop,spec}}(\omega_i)$ are the phase specifications at ω_i. If ω_i are related to the frequencies of the partials, then the values of $\Phi_{\mathrm{loop,spec}}(\omega_i)$ should be integer multiples of -2π.

In matrix form, the equation system becomes $\mathbf{Ta} = \mathbf{b}$, where the elements of $\mathbf{T}$ are

$$T_{ij} = \sin(\beta_i + k\omega_i), \qquad (8)$$

for $i = 1, 2, \ldots, G$ and $k = 1, 2, \ldots, N$. The vectors $\mathbf{a}$ and $\mathbf{b}$ are given, respectively, by $\mathbf{a} = [a_1\ a_2\ \cdots\ a_N]^{\mathrm{T}}$ and $\mathbf{b} = -[\sin(\beta_1)\ \sin(\beta_2)\ \cdots\ \sin(\beta_N)]^{\mathrm{T}}$.

If $G = N$, the solution for $\mathbf{a}$ is given by $\mathbf{a} = \mathbf{T^{-1}b}$. For $G > N$ the system becomes overdetermined and an exact solution is no longer available. However, approximations within a certain error can be obtained, e.g., using a weighted least-squared equation error (LSEE) formulation [6]. In our case, we want to match 3 partials with a second-order filter. This implies $G > N$ and thus the LSEE solution becomes

$$\mathbf{a} = [\mathbf{T}^{\mathrm{T}}\mathbf{W}\mathbf{T}]^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{W}\mathbf{b}, \tag{9}$$

where $\mathbf{W}$ is a diagonal matrix containing non-negative weight values. The optimum weighting values that form the matrix $\mathbf{W}$ depend on $\mathbf{a}$, which is to be estimated. Therefore an iterative algorithm has been proposed in [6] as a solution to this problem. The procedure consists of initializing the iterations with an arbitrary weight function and estimating $\mathbf{a}$ through Eq. (9). Then, the weight matrix is corrected based on the just estimated allpass filter coefficients. The iterations continue until the norm of difference between the actual and previously estimated $\mathbf{a}$ becomes smaller than a constant. In our simulations we arbitrarily set the value of this constant to 1×10^{-6}.

4.2. Allpass filter and delay line

In addition to the allpass coefficients, we also have the value of L as a free parameter. Here, the value of L will not be constrained to be an integer number, since a tuning filter can be added to the IDWG to implement the fractional part of the delay line. An iterative procedure can also be devised to jointly estimate $\mathbf{a}$ and L.

A first estimate for L can be $L = 2\pi/\omega_1 - 2$, where ω_1 is the frequency of the first partial and the -2 accounts for the delay the second-order allpass introduces itself. With this value of L and with the phase specifications for the frequencies of the first three partials ω_i, i.e., $\phi_{\mathrm{ol,spec}}(\omega_i) = -2\pi i$ for $i = 1, 2, 3$, we proceed iteratively as follows:

1. Estimate the coefficients of the second-order allpass filter using the LSEE method.

2. Obtain a new estimate for L such that the frequency of the first partial be perfectly matched, i.e.,

$$L^{(n+1)} = \frac{-\phi_{\mathrm{ol,spec}}(\omega_1) + \phi_{\mathrm{ap}}^{(n)}(\omega_1)}{\omega_1}, \tag{10}$$

   where n is the iteration index and $\phi_{\mathrm{ap}}^{(n)}(\omega_1)$ is the phase response of the allpass filter at iteration n, evaluated at the frequency of the first partial.

3. Go to step 1 if

$$\frac{||\phi_{\mathrm{ol,spec}}(\omega_i) - \phi_{\mathrm{ol,IDWG}}^{(n)}(\omega_i)||}{||\phi_{\mathrm{ol,spec}}(\omega_i)||} > \varepsilon, \tag{11}$$

   where ε is a tolerance factor and $\phi_{\mathrm{ol,IDWG}}^{(n)}(\omega_i) = \phi_{\mathrm{ap}}^{(n)}(\omega_i) - L^{(n)}\omega_i$ is the phase response of the open loop at iteration n. Otherwise, stop the iterations.

4.3. Loss filter

The design of the loss filter has to be carried out after that of the allpass filter, since the presence of the latter may also imply a frequency-dependent loss due to the changes on the pole radius governed by Eq. (5).

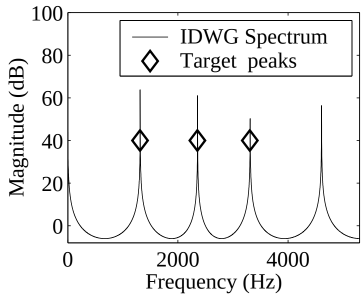

Figure 2. Detail of the magnitude response of the IDWG filter and target frequencies (Mode 1).

Given the specifications of the decay time of the partial modes, expressed in T_{60}, the compensated target gains for the magnitude of the loss filter at the mode frequencies can be obtained by

$$g(\omega_i) = e^{\tau_{\mathrm{ol}}(\omega_i)\log(0.001)/(T_{60_i} f_{\mathrm{s}})}, \quad i = 1, 2, \ldots \tag{12}$$

where $\tau_{\mathrm{ol}}(\omega_i)$ are the measured group delays of the open loop at ω_i and T_{60_i} are the prescribed decay times of the partial modes.

Linear-phase finite impulse response (FIR) filter is a suitable choice for implementing the loss filter, since its contribution to $\tau_{\mathrm{ol}}(\omega)$ is constant for all ω. In our simulations, we used a second-order FIR filter. Moreover, we decide to meet only the decay time specifications of the first two partial modes. This restriction allows not only designing the loss filter in a closed form, but also meeting the magnitude response requirement exactly. The resulting filter is guaranteed to be lowpass if $g(\omega_2) < g(\omega_1)$.

4.4. Tuning filter

As discussed in Section 4.1, the value of L was optimized to perfectly tune the first partial. In general, the value of L is a real number and a fractional-delay filter may be needed in the IDWG. For this task we used a first-order allpass filter whose coefficient is given by [7]

$$a_{\mathrm{t}} = \frac{\sin[(1-\delta)\omega_1/2]}{\sin[(1+\delta)\omega_1/2]}, \tag{13}$$

where ω_1 is the frequency of the first partial in radians and $0 < \delta < 1$ is the fractional part of the delay line. We can compute $\delta = L - L_{\mathrm{I}}$, with $L_{\mathrm{I}} = \lfloor L - \tau_{\mathrm{lf}}(\omega_1) - \tau_{\mathrm{ap}}(\omega_1) \rfloor$, where L_{I} is the integer part of the delay line, $\tau_{\mathrm{lf}}(\omega_1)$ is the group delay contribution of the loss filter at ω_1. Operator $\lfloor \cdot \rfloor$ means *the greatest integer less than or equal to.*

5. SIMULATION EXAMPLES

In this section, we verify through examples the capability of the IDWG to match specified partial frequencies only. Therefore, the design of the loss filter will be neglected, since it concerns controlling the decay time of the partials.

5.1. Real-life hand-bell

The 'Mode 1' frequencies of the first three partials (see Table 1) are taken as targets. Then, the procedure described in Section 4.1 is employed to estimate the coefficients of the inharmonizing second-order allpass filter and the length of the delay line. In the sequel, we implemented an IDWG filter as shown in Figure 1. However, the loss filter was set to a unit gain and the parallel resonators were absent. The magnitude response of the resulting IDWG filter is plotted in Figure 2. For comparison purposes, the target resonance frequencies are marked arbitrarily at 40 dB. A successful IDWG design was also attained for matching the frequencies of the second modes of the partials.

5.2. Artificial examples

Here, we aim at studying the inharmonicity limitations of the IDWG filter. Some initial remarks are worth mentioning before showing the examples:

1. The role of an allpass filter within the feedback loop is to attract the partial frequencies close to the frequency of the allpass pole [4]. Thus, a typical situation in which the automated design method is likely to work occurs when the frequencies of the second and third partials are both less (but not too much) than 2 and 3 times the frequency of the first partial, respectively. For example, $\omega_2 = 1.7\omega_1$ and $\omega_3 = 2.4\omega_1$.

2. If only the second partial varies from its original harmonic value, the situation is characterized by a partial stretching (from the second partial with respect to either the first or the third partial). As a consequence, the estimation procedure may lead to useless results.

We start with an example in which the target partial frequencies are related harmonically, i.e., $\omega_2 = 2\omega_1$ and $\omega_3 = 3\omega_1$. The fundamental frequency is set to $f_1 = 1.0$ kHz. A successful IDWG is attained, although the allpass filter is not really necessary in this case.

Now, we specify an inharmonic configuration by setting $\omega_2 = 1.8\omega_1$ and $\omega_3 = 3\omega_1$. This example lies in the category described in remark 2. The magnitude response of the resulting IDWG is displayed in Figure 3. In this example, convergence is achieved with the iterative design method. However, the poles of the allpass filter are on the unit circle. Therefore, instead of moving the frequencies of the partials, this solution creates a new peak at the location of the second partial. This resonance is useless since it has low magnitude. Moreover, an undesired peak is located close to the second partial remains present in the IDWG transfer function.

In the next example, we set again $\omega_2 = 1.8\omega_1$ but set $\omega_3 = 2.4\omega_1$, i.e., the target partials are moderately squeezed. For this configuration, the iterative estimation procedure works successfully as can be seen from the results shown in Figure 4.

The allpass filter design method does not guarantee a stable filter. Indeed, for more demanding cases where the partial squeezing becomes too strong, the resulting allpass can be unstable. Nevertheless, the iterative procedure used to estimate the allpass coefficients and the value of L may

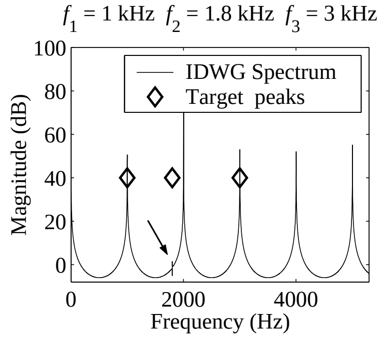

Figure 3. Inharmonic target frequencies showing partial stretch. The poles of the allpass are on the unit circle. The arrow indicates the weak extra resonance peak.

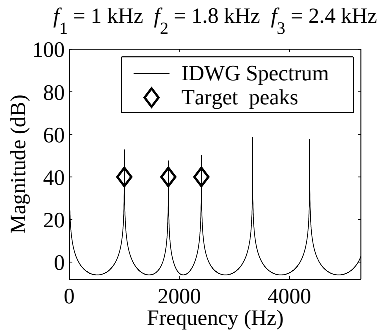

Figure 4. Inharmonic targets showing partial squeeze.

converge. Of course the IDWG cannot be implemented with an unstable allpass filter.

An example in which such situation occurs is shown in Figure 5. In this case, the frequency of the second partial is set to $\omega_2 = 1.5\omega_1$ and that of the third partial is set to $\omega_3 = 2.7\omega_1$, i.e., the difference in frequency between the second and third partials becomes larger than the fundamental frequency. As a consequence and similarly to the result shown in Figure 3, all target spectral peaks are matched, but an extra peak appears unnecessarily in the resulting magnitude spectrum of the attained IDWG. Furthermore, the estimated allpass filter is unstable, preventing the realization of the IDWG.

6. A COMPLETE EXAMPLE

In this section we go through the complete design of a hand-bell synthesizer. Besides the design of the IDWG,

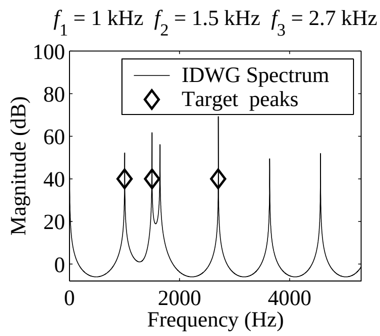

Figure 5. Inharmonic target frequencies. The attained allpass is unstable, hence the IDWG is useless.

Param.	Mode 1	Mode 2
a	$[1,\ -1.7114,\ 0.8308]$	$[1,\ -1.7097,\ 0.8314]$
$\lfloor L \rfloor$	29	29
a_{t}	0.0832	0.0636
h_{lp}	$[0.0175,\ 0.9649,\ 0.0175]$	$[0.0126,\ 0.9744,\ 0.0126]$

Table 2. Parameters of the two IDWG filters.

there are also those of the loss filter and the simulation of the typical warble sound associated with bell sounds.

6.1. Complete IDWG design

The IDWG design starts by specifying the phase of open loop at the frequencies of the first three partials (either those of Mode 1 or Mode 2 shown in Table 1) and running the iterative algorithm that estimates $\mathbf{a}$ and L, as described in Section 4.2. Next comes the design of the tuning filter. The coefficient of tuning filter, a_{t}, is computed straightforwardly via Eq. (13) with $\delta = L - \lfloor L \rfloor$.

To design the loss filter we need the compensated gain specifications, which are obtainable through Eq. (12). We can measure the group delay of the open loop without including the loss filter. We know that, as second-order FIR, it adds only a unit delay to the open loop. The loss filter is designed as highlighted in Section 4.3.

Finally, the loss filter is included in the feedback path and one unit delay is removed from the delay line to compensate for that. The design of the two IDWGs, one for each set of resonance frequencies (Mode 1 and Mode 2), is carried out separately.

6.2. Simulating beating

The IDWG filter accounts only for simulating the inharmonic spectrum of a bell sound. As proposed in [8] and used in [4], for instance, two second-order resonators were tuned to the second vibration modes of the first two partials and placed in parallel with the IDWG (see Figure 1). If more partials showed beating, a larger number of resonators in parallel with the IDWG would be needed. However, this would decrease the computational efficiency of the synthesizer. Alternatively, two (or more) IDWG filters, one for each of the resonance modes of the lower partials, can be placed in parallel.

Here, we decided for using two IDWG in parallel. This is justified by the precision with which each IDWG can be tuned to the resonance modes of the partials. The parameters of the two IDWGs, each one designed through the proposed method as to meet the specifications given in Table 1, are shown in Table 2. Once the IDWGs have been designed, synthesis is carried out by exciting the system. A short excitation (about 5 ms) can be used for this purpose. For instance, an amplitude decaying noise burst. Figure 6 compares the spectra of the original and synthesized hand-bell sound.

7. CONCLUSIONS

This paper discussed synthesis of bell sounds by means of an inharmonic digital waveguide filter. We presented an iterative method to automatically estimate the parameters of IDWG synthesis filter, such that its first three resonances match those of a target hand-bell.

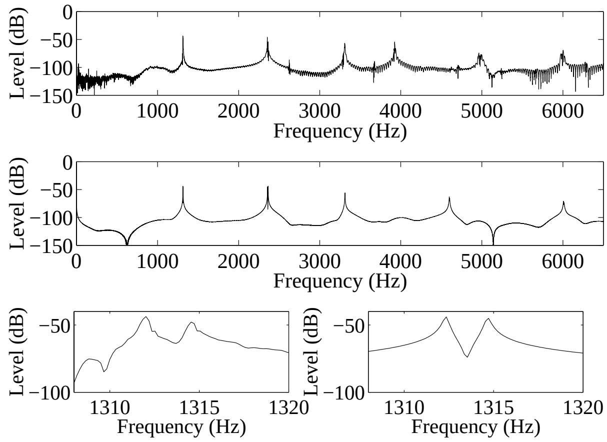

Figure 6. Magnitude spectra of the original hand-bell (top) and synthesized bell (middle). Zoom in the first partial: original (bottom-left) and synthesized (bottom-right).

8. REFERENCES

[1] S. Van Duyne and J. O. Smith, "A simplified approach to modeling dispersion caused by stiffness in strings and plates," in *Proc. ICMC'94*, 1994, pp. 407–414.

[2] D. Rocchesso and F. Scalcon, "Accurate dispersion simulation for piano strings," in *Proc. NAM'96*, 1996, pp. 407–414.

[3] G. Essl and P. R. Cook, "Banded waveguides: towards physical modeling of bar percussion instruments," in *Proc. ICMC'99*, 1999, pp. 321–324.

[4] M. Karjalainen, V. Välimäki, and P. A. A. Esquef, "Efficient modeling and synthesis of bell-like sounds," in *Proc. 5th DAFx-02*, 2002, pp. 181–186, URL: http://www.acoustics.hut.fi/demos/dafx02/.

[5] M. Karjalainen, P. A. A. Esquef, P. Antsalo, A. Mäkivirta, and V. Välimäki, "Frequency-zooming ARMA modeling of resonant and reverberant systems," *J. Audio Eng. Soc.*, vol. 50, no. 12, pp. 1012–1029, 2002.

[6] M. Lang and T. I. Laakso, "Simple and robust method for the design of allpass filters using least-squares phase error criterion," *IEEE Trans. Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 41, no. 1, pp. 40–48, Jan. 1994.

[7] K. Steiglitz, *A Digital Signal Processing Primer – with Applications to Digital Audio and Computer Music*, chapter 6, Addison-Wesley Publishing Company, Inc., 1996.

[8] B. Bank, V. Välimäki, L. Sujbert, and M. Karjalainen, "Efficient physics-based sound synthesis of the piano using DSP methods," in *Proc. EUSIPCO'00*, Tampere, Finland, Sept. 2000, vol. 4, pp. 2225–2228.