MAKING OF A COMPUTER CARILLON

Matti Karjalainen, Paulo A. A. Esquef, and Vesa Välimäki

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland
matti.karjalainen@hut.fi

ABSTRACT

In this paper we present an analysis, modeling, and synthesis approach to bell sounds. Recorded bell sounds are first analyzed by high-resolution frequency-zooming modal estimation technique, where each partial is described typically by 2-3 submodes in order to include the warble characteristics of envelope beating in the partials. The bells are then modeled by inharmonic digital waveguides (DWGs) in order to achieve a highly efficient yet parametrically controllable synthesis model. The third step is to make a set of bells and related parametric controls in order to build a "computer carillon". Special means are needed to approximate the strongly inharmonic bell sounds by DWGs, paying particular attention to the accuracy of the lowest partials. The control of bell models in the computerized carillon is through triggering of wavetables that store the initial part of the residual signal obtained in the modal decomposition process. Changing or modulating the modal parameters allows for sound effects that are not possible in real physical bells.

1. INTRODUCTION

The aim of this study is to analyze and model bell sounds for efficient real-time synthesis and playing control of a set of bell models, forming a computerized carillon. Full-scale physical models are too complex for the task, so we search for efficient semiphysical or DSP based techniques.

The most characteristic features of bell sounds are an inharmonic set of decaying partials, typically showing temporal envelope fluctuation, resulting in warble sound [1, 2]. Such beating in partials is due to two or more eigenmodes of vibration very close in frequency. Fig. 1 shows the waveform and spectrum of a small handbell, also the spectrum zoomed to the first partial around 1.3 kHz, demonstrating two mode peaks with spacing of about 2.5 Hz.

In Section 2 we first present a parametric analysis technique that is able to estimate with high accuracy the modal parameters of each partial. In Section 3 different synthesis modeling strategies are discussed and their applicability and efficiency is studied to find the best strategies for different bell types. In Section 4, realization of a computerized carillon for real-time synthesis is described, followed by summary and conclusions.

2. HIGH-RESOLUTION ANALYSIS OF BELL SOUNDS

The first task is to analyze recordings of target bell sounds to form the basis for synthesis models. We have developed a high-resolution parametric estimation technique based on frequency-zooming autoregressive moving-average method (FZ-ARMA) [3, 4], yielding accurate values for modal frequencies and decay times.

The first step of the procedure requires spectral analysis of the bell sound to roughly determine the frequencies of its prominent

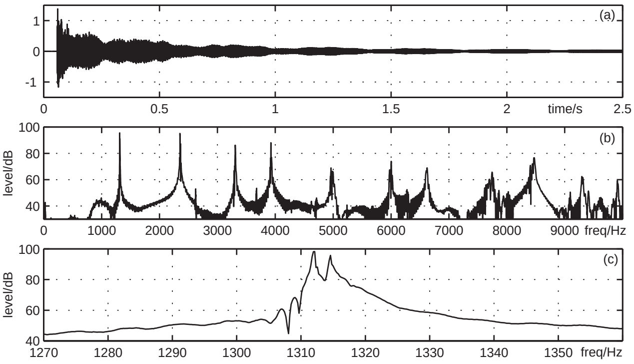

Figure 1: Analysis and modeling of a small bell sound: (a) recorded time-domain signal, (b) magnitude spectrum up to 10 kHz, (c) magnitude spectrum in the modal region around 1313 Hz.

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, the following steps are taken:

1. Down-modulate the original signal to place the spectral peak of interest around 0 Hz, obtaining a complex signal.
2. Apply low-pass filtering to isolate this particular partial. The bandwidth of the filter must be narrow enough to reject neighboring partials.
3. Maximally decimate the low-pass filtered complex signal.
4. Fit a low-order complex-valued ARMA model to the 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, i.e., the frequencies are related to the angles of the poles whereas the decay times are related to the pole radii. More details on the FZ-ARMA analysis can be found in [3]. 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 in bell sounds, see Fig. 1.

As an example, the sound waveform of a hand-bell in Fig. 1, sampled at 44.1 kHz, is analyzed via the FZ-ARMA modeling scheme. The decimated complex-valued signals associated with each partial are modeled through 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 partial modes (two modes per partial in this case). Figure 2 plots the decay envelopes of the lowest partials. Beating is found particularly in the first and second partial. High partials decay relatively fast so that their details are not so important from a perceptual point of view.

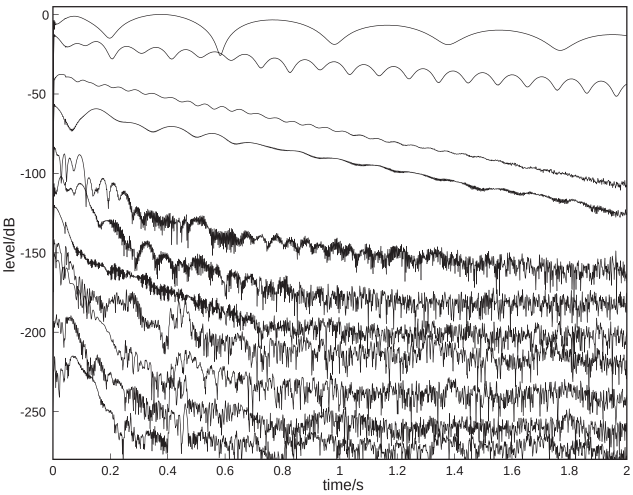

Figure 2: Decay envelopes for partials in bell sound (Fig. 1) from top to down plotted on the dB scale with 20 dB offsets for clarity. Higher partials reach the measurement noise floor.

3. EFFICIENT SYNTHESIS MODELS FOR BELLS

When high-quality recordings of bell sounds are available, they can be used directly in sampling synthesis, i.e., in replay from wavetables. This has two drawbacks, however. One is the need of memory for long samples, even tens of seconds. In many cases this is not necessarily a drawback anymore with modern memory technologies. Another problem is the inflexibility of sample-based synthesis if sound features have to be changed at runtime. This was the main motivation to experiment with different efficient parametric synthesis techniques.

3.1. Modal filterbank synthesis

A straightforward method of sound synthesis is to use modal parameters from the FZ-ARMA analysis to build a modal filterbank synthesizer [4]. This is relatively efficient in the handbell case of Fig. 1 where appoximately 11 partials is enough for a frequency band of 10 kHz. If a partial consists of two modes, each realized by a filter of order two, the overall filterbank order is about 44. An impulse can be used as the excitation of such a synthesis model. For large low-pitch bells there can be need for filterbank orders of several hundreds. Thus it is attractive to search for even more efficient models.

3.2. Inharmonizing a digital waveguide

Digital waveguides (DWG) are known as highly efficient models for synthesizing musical instruments that have harmonic spectra [5]. Now we study how to make them inharmonic. DWGs consist basically of a delay loop (see two of them in Fig. 3) which create a comb filter type of spectrum, so that the frequencies of modal resonances are located at frequencies where the phase shift is integer multiple of 2π. Inharmonic relationship of modes means nonlinear phase function, which can be obtained by cascading an integer-length (L) delay with an allpass filter:

$$H_{\mathrm{delay}}(z) = \frac{a_N + \cdots + a_1 z^{-N+1} + z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}\, z^{-L}. \qquad (1)$$

The phase delay will be

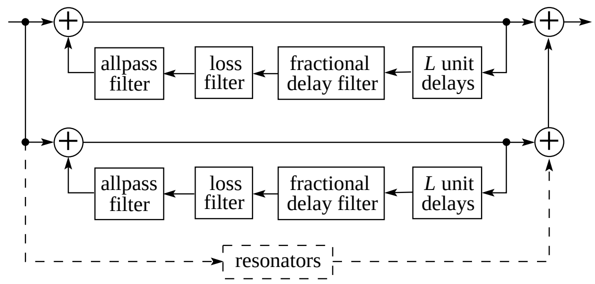

Figure 3: Bell model of dual DWG filters with additional parallel resonators. Each DWG consist of integer delay (L), fractional delay for fine tuning of pitch, loss filter, and inharmonizing allpass filter.

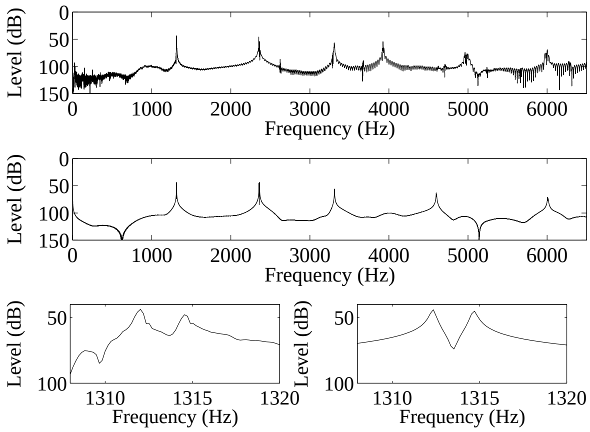

Figure 4: Magnitude spectra of the original handbell (top) and synthesized bell (middle). Zoom in the first partial: original (bottom-left) and synthesized (bottom-right).

$$\tau_{\mathrm{ph}}(\omega) = L + N - \frac{2}{\omega} \arctan \left\{ \frac{\sum_{k=1}^{N} a_k \sin(k\omega)}{1 + \sum_{k=1}^{N} a_k \cos(k\omega)} \right\}, \quad (2)$$

where a_k are the coefficients of the allpass filter, constrained to $a_0 = 1$. For efficient realization the allpass order N must be low. The most interesting case is to use a second order allpass ($N = 2$).

In [6] we developed an automated iterative procedure to fit an inharmonic DWG model, i.e., a feedback loop with a delay plus a second-order allpass loop filter to tune the three lowest partials of a given bell response.

After tuning the modal frequencies, the decay times must be adjusted by a low-pass filter in the waveguide loop to have proper losses at each modal frequency. When a low-order filter is used, typically of first or second order, it can only approximate the general trend of decay time vs. frequency, paying most attention to adjusting the long-ringing lowest partials to decay properly.

Two such inharmonic DWGs must be connected in parallel in order to have two modes for each partial, having a proper frequency difference and decay times to make a desired beating (warble sound). If only one or two partials show remarkable beating, such as in the case of Fig. 1, the second DWG can be avoided by using extra parallel resonators to create the beating for the desired partials in the main DWG.

Figure 4 shows the original and a modeled magnitude spectrum for the handbell introduced in Fig. 1.

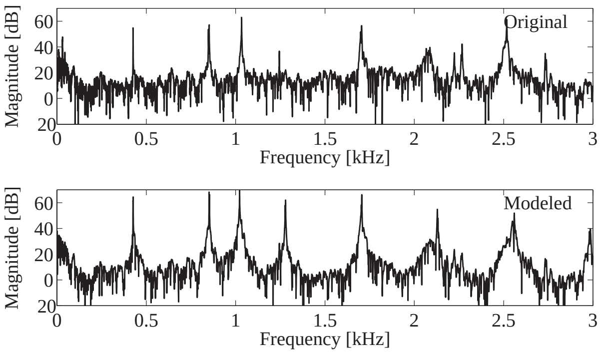

Figure 5: Magnitude spectra for a tuned bell: original (top) and modeled (bottom).

3.3. DWG synthesis of tuned bells

The handbell modeled above is an example of bell making where no specific effort is done to increase the harmonicity of partials. In large bells it is common to shape the bells so that many partials are in approximately harmonic frequency relationships [2]. This improves to perceive a definite pitch. The remaining inharmonic ones make the bell-like timbre.

It was found that designing inharmonic DWG models for tuned bells is more difficult than for the handbell using the technique described above. It turns out that the inharmonic partials between approximately harmonic ones tend to get too high Q values, i.e., to decay much slower than desired. This could be compensated for by a loop filter with a magnitude response dip at that frequency, but this leads to high-order loop filters and increased complexity of design.

One choice of compromise could be to let the harmonic components become somewhat less harmonic, yet getting a clearly bell-like sound. This is however not acceptable if a given tuned bell sound must be modeled with a perceptually clear pitch.

A better choice for efficient and relatively accurate modeling of tuned bells is to use digital waveguides only for the realization of the harmonically related partials and to implement remaing low-frequency inharmonic partials by separate parallel resonators.

The upper curve in Fig. 5 depicts the spectral range of 0–3 kHz of a bell from a Belfort bell set[1]. It can be seen that partials 1, 2, 4, 5, etc., form a nearly harmonic set, while partial 3 (just above 1 kHz) is clearly inharmonic. In this case the simplest choice is to model partial 3 as a pair of separate modal resonances and approximate all others by a pair of digital waveguides, designed as in the small bell case. Pairwise modes are needed to realize proper beating envelopes in the partials. The bottom curve in Fig. 5 plots the magnitude spectrum of a model designed this way. The most notable difference is the stronger fourth partial in the synthetic one, which is due to the analysis of model excitation described below.

3.4. Excitations

Different ways of obtaining the excitation signal to trigger the hitting of a model are used in different cases. The easiest case is the modal filterbank model, where, for a properly designed ARMA model, the excitation is simply an impulse. If only poles are used to make an all-pole (AR) model, the excitation is obtained by inverse filtering the recording and truncating the result to contain the main energy part.

[1]The recordings were kindly provided by Marc Leman.

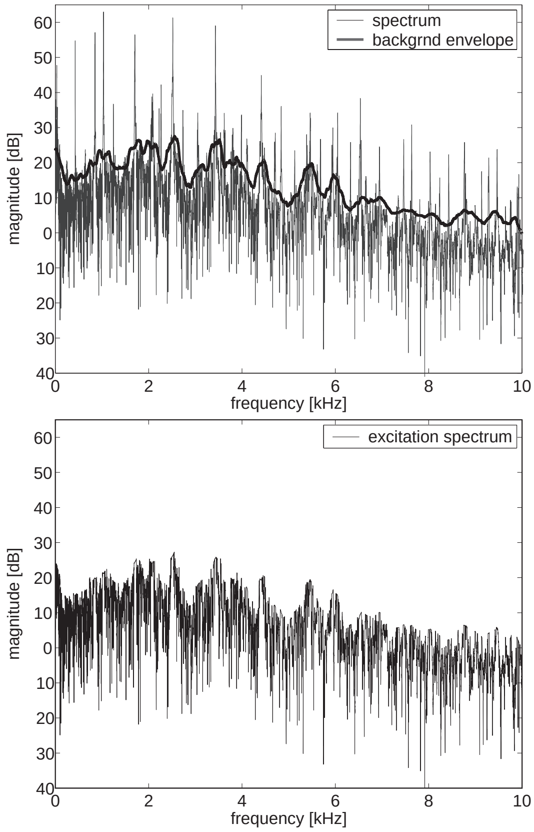

Figure 6: Original and excitation spectra of an inharmonic bell. The excitation is obtained by removing the modal resonances from the complex spectrum of the original sound.

In the digital waveguide cases we have tried the following techniques to get the content of excitation wavetable. A simple way is to use the inverse-filtered residual as mentioned above and to shape it spectrally (e.g., by low-order AR modeling) to yield a proper spectral envelope for the synthetic sound.

Another technique that was applied in the inharmonic DWG modeling of the tuned bell above is to get the excitation from the original sound in the following way:

1. First a long FFT ($2^{17} = 131072$) of the original bell sound is computed;
2. The magnitude and phase are computed and the phase is stored for future use;
3. An estimate of the background spectrum is obtained using the two-pass split-window technique [7];
4. The FFT bins associated with the spectral peaks are replaced with the values of the background spectrum;
5. The modified magnitude spectrum and the original phase are transformed back into the time domain via IFFT;
6. Finally, the resulting excitation signal is windowed: the second half of a Hanning window of length 2^{17} is used for this purpose.

Figure 6 illustrates the procedure used to modify the spectrum of the original bell sound to generate the excitation signals. The magnitude modification is as simple as adopting the spectral envelope curve as an upper magnitude limit excluding the modal resonances.

4. REAL-TIME SYNTHESIS AND CONTROL

Real-time synthesis and playing control of the bell models has been carried out in the BlockCompiler modeling software [8, 9], developed particularly for experimental physical modeling purposes. The BlockCompiler is based on the Common Lisp language for creating and manipulating of block-based models, 'patches', as well as C-code generation and compilation of the patches for efficient run-time execution of the models.

As an example of creating patches in BlockCompiler the modal filterbank model of the handbell can be made by the following script:

```
(defvar bellpatch
  (patch ((wt (.wtable :data *bell-wtable*))
          (iir (.iir :denom *bell-coeffs*))
          (da (.da)))
    (-> wt iir (inputs da))
    (defun hit () (trigger wt))))
```

where the bell model object created by the form starting from `patch` is assigned to the variable `bellpatch`. A wavetable for model excitation is assigned to local variable `wt`, then the modal filterbank is made to `iir`, and furthermore sound output (DA-conversion) is bound to `da`. The line starting by `(-> ...)` makes inteconnections between block outputs and inputs. Finally the function `hit` is compiled so that when it is called, it triggers the wavetable to feed excitation to the filterbank. Wavetable data `*bell-wtable*` and IIR filterbank denominator coefficients, `*bell-coeffs*`, are assumed to be preanalyzed from a given target sound. When a command `(run-patch bellpatch)` is executed, the model starts streaming real-time synthesis, and the bell can be triggered by function form `(hit)`.

Inharmonic digital waveguide models can be created in a similar way, although scripting will be somewhat more elaborate. More than 100 such bell models can be executed simultaneously in real time on a 1 GHz computer.

When a set of bells is made by any of the methods described above, a computarized carillon is obtained by making the bells playable through some user interface. A MIDI keyboard can be mapped to the triggering inputs of the bells. Even a computer keyboard may be used in simple demonstrations. The carillon can be played also from sequencer programs through MIDI control or directly from specialized sequencer programs.

The advantage of parametric synthesis models it that sound features can be easily controlled also at runtime. This is important particularly when computer music is created where the potential of modifying sounds is utilized. With synthesis models it is relatively easy to produce sounds that are very difficult or impossible to create by real physical means. One such example is to modulate the frequencies of the modal resonances. Many other digital sound effects can be easily added to the model-based synthesis approach.

5. SUMMARY

In this paper we have investigated the analysis, efficient real-time synthesis, and control of bell sounds, i.e., making of a computerized carillon. The technique used for parametric analysis is based on high resolution FZ-ARMA modeling which yields modal parameters even for complicated partial behaviors.

For real-time synthesis we have used inharmonic digital waveguides, where the lowest partials are inharmonized by an additional allpass filter in the DWG loop. Two submodels and/or additional parallel resonators are needed to achieve proper decay envelopes of the partials, which is important particularly to make warble sound. Rapidly decaying higher partials do not need as accurate modeling since auditory perception of main timbral features is mostly based on the long-ringing low-frequency partials. There often seems to be only a couple of lowest partials that show strong beating in their envelopes. In such cases a good candidate for efficient and accurate synthesis is to use only one inharmonic digital waveguide and separate parallel resonators for the low-frequency modes to make the envelopes beating.

Finally we have discussed the implementation for real-time synthesis and control of the synthesis models using a physical modeling software called BlockCompiler.

Sound examples are available in:
`http://www.acoustics.hut.fi/demos/smac03/bells`

6. ACKNOWLEDGMENTS

The work of Paulo Esquef has been supported by the EU funded project ALMA (IST-2001-33059) and by the Brazilian National Council for Scientific and Technological Development (CNPq). Our thanks are due to Marc Leman who kindly provided the set of Belfort bell recordings.

7. REFERENCES

[1] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. New York, USA: Springer-Verlag, 1991.

[2] T. D. Rossing, *Acoustics of Bells*. Striudsburg, PA: Van Nostrand-Reinhold, 1984.

[3] 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, Dec. 2002.

[4] M. Karjalainen, V. Välimäki, and P. A. A. Esquef, "Efficient Modeling and Synthesis of Bell-Like Sounds," *Proc. DAFx-02*, Hamburg, Germany, pp. 181–186, Sept. 2002.

[5] J. O. Smith, *Principles of Digital Waveguide Models of Musical Instruments*, Ch. 10 in *Applications of Digital Signal Processing to Audio and Acoustics*, (ed. Kahrs and Brandenburg). Kluwer Academic Publishers, 1998.

[6] P. A. A. Esquef and V. Välimäki, "Design of an Efficient Inharmonic Digital Waveguide Filter for Synthesis of Hand.Bell Sounds," *Proc. FINSIG'03*, Tampere, Finland, May 2003.

[7] W. A. Struzinski and E. D. Lowe, "A Performance Comparison of Four Noise Background Normalization Schemes Proposed for Signal Detection Systems," *J. Acoust. Soc. Am.*, Vol. 76, no. 6, pp. 1738–1742, Dec. 1984.

[8] M. Karjalainen, "BlockCompiler: Efficient Simulation of Acoustic and Audio Systems," Preprint 5756 of *114th AES Convention*, Amsterdam, March 2003.

[9] M. Karjalainen, "Time-Domain Physical Modeling and Real-Time Synthesis Using Mixed Modeling Paradigms," *Proc. SMAC'2003*, Stockholm, Aug. 2003.