

ROBUST DESIGN OF VERY HIGH-ORDER ALLPASS DISPERSION FILTERS

Jonathan S. Abel*

Universal Audio, Inc.
Santa Cruz, CA 94060 USA
abel@uaudio.com

Julius O. Smith

CCRMA, Stanford University
Stanford, CA 94305 USA
jos@ccrma.stanford.edu

ABSTRACT

A nonparametric allpass filter design method is presented for matching a desired group delay as a function of frequency. The technique is useful in physical modeling synthesis of musical instruments and emulation of audio effects devices exhibiting dispersive wave propagation. While current group delay filter design methods suffer from numerical difficulties except at low filter orders, the technique presented here is numerically robust, producing an allpass filter in cascaded biquad form, and with the filter poles following a smooth loop within the unit circle.

The technique was inspired by the observation that a pole-zero pair arranged in allpass form contributes exactly 2π radians to the integral of group delay around the unit circle, regardless of the (stable) pole location. To match a given group delay characteristic, the method divides the frequency axis into sections containing 2π total area under the desired group-delay curve, and assigns a pole-zero allpass pair to each. In this way, the method incorporates an order selection technique, and by adding a pure delay to the desired group delay, allows the trading of increased filter order for improved fit to the frequency-dependent group delay. Design examples are given for modeling the group delay of a dispersive string (such as a piano string), and a dispersive spring, such as in a spring reverberator.

1. INTRODUCTION

In many media, waves propagate *dispersively*, with different frequencies traveling at different speeds. For instance, as described in Fletcher and Rossing [1], high frequencies slightly outrun low frequencies on “stiff” strings, making the higher overtones somewhat sharp. A pulse propagating on such a string is eventually “smeared out” along the string, as illustrated in Figure 1, with its high frequencies leading the way, and its low frequencies lagging.

To simulate such a system, waveguide synthesis models [2] are often used for their accuracy and efficiency. A waveguide model of a dispersive string segment is shown in Figure 2; propagation losses and dispersion are lumped into filtering at the waveguide ends. For musical strings such as for the piano, the attenuation filter $A(z)$ can be low-order and minimum-phase, yet perceptually exact, because there is very little loss in the string itself. The dispersion filter, in contrast, needs to be a high-order allpass whose job it is to delay different frequencies by different amounts according to the frequency-dependent propagation speed.

There are several approaches to designing allpass filters to achieve a prescribed delay vs. frequency. We now give a brief overview of some prior literature on this problem.

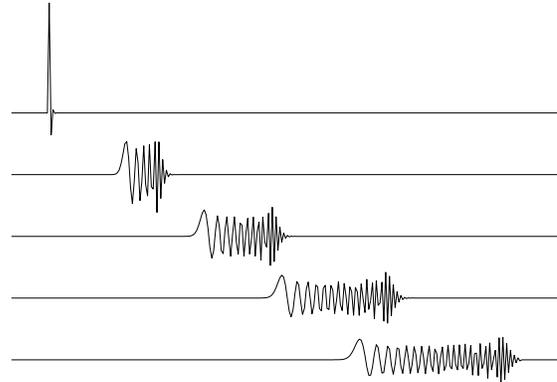


Figure 1: *Dispersive string propagation.*

Hilbert-transform methods, such as [3, 4], make use of the fact that the log-magnitude and phase of a minimum-phase spectrum form a Hilbert transform pair [5]. A group-delay filter can then be designed by integrating the group delay to form a desired phase response. An all-pole filter is fit to a minimum-phase frequency response having log magnitude equal to the Hilbert transform of half the desired phase response. The allpass filter is then formed by inverting the poles to generate corresponding maximum-phase zeros.

These nonparametric Hilbert-transform methods have a few drawbacks. They are not optimal, and they use the FFT in a manner that suffers significant time aliasing when the poles and zeros get too close to the unit circle in the complex plane. Perhaps a more serious difficulty is that the filter designed by these methods is in direct form, as opposed to factored form (which is often preferred in applications). It is often costly and numerically difficult to factor a high-order allpass filter form for implementation as cascade second-order sections. In addition, because the round-off error in direct-form filter coefficients can have a very large effect on the locations of poles and zeros, these methods can break down at extremely high model orders. Moreover, the internal all-pole filter-design methods used (such as LPC in [3]) can itself suffer numerical difficulties at the needed very high orders. Finally, there is no built-in model-order selection in these methods.

In [6], high quality stiff-string sounds were demonstrated using high-order allpass filters to simulate dispersion in a digital waveguide model. In [7], this work was extended by applying a least-squares allpass-design method [8] and a spectral Bark-warping technique [9] to the problem of calibrating an allpass filter of arbitrary order to recorded piano strings. They were able to correctly tune the first several tens of partials for any natural piano

*Jonathan Abel is also a consulting professor at CCRMA.

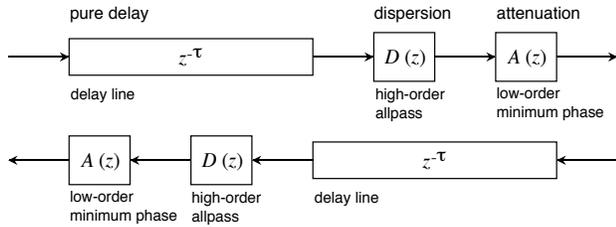


Figure 2: Dispersive waveguide section.

string with a total allpass order of 20 or less. Additionally, minimization of the L^∞ norm [10] has been used to calibrate a series of allpass-filter sections [11], and a dynamically tunable method, based on Thiran allpass filters, has recently been proposed [12].

Optimal allpass design methods are limited by numerical precision and computational cost. In particular, Bensa [11] reports difficulty in matching a piano-string dispersive delay using a model order greater than five. In a least-squares method for allpass design [8], the maximum order was 15 in five design examples, and numerical difficulties were noted when designing a particular delay equalizer at orders greater than 10. A detailed analysis of numerical issues is given in [10], along with techniques for roughly doubling the order that can be reliably designed. In particular, the numerical improvements enabled an order increase from 10 to 20 for a delay equalizer for a 7th-order lowpass filter. In [13], a particular order 80 lowpass delay equalizer was designed using linear programming methods; the order of another example in that paper was extended from 25 to 150 using the numerical improvements of [10], but this was classified as a “special case”, and is the highest order we have seen reported to date by such methods. In all of these methods, the allpass order is set *a priori*, rather than being automatically determined by the design method in some way.

The technique described in this paper, first presented at [14], is essentially nonparametric, with the allpass filter computed in *factored biquad form* directly from the desired group delay function. The method is based on an invariant feature of allpass group delays: A (complex) pole-zero pair arranged as an allpass filter will generate a group delay that peaks at the pole-zero frequency, and that has a constant area of 2π , irrespective of the pole location within the unit circle.

The group delay of a cascade of filter sections is the sum of the section group delays. So, to match a given group-delay characteristic, the frequency axis may be divided into *bands* such that the area under the desired group delay curve in every band is 2π radians. A pole-zero allpass section may then be assigned to each band, and the pole radius adjusted so that a specified portion of its group-delay area occurs within the band. An allpass filter approximating the desired group delay characteristic is formed by cascading a number of such allpass sections, covering the desired total frequency band.

Details of the method are given in § 3, with various design tradeoffs discussed in § 4. Applications to physical modeling synthesis of a stiff string and to emulation of a spring reverberator are considered as design examples. Finally, § 6 contains our summary and conclusions.

We begin by reviewing properties of the group delay of a pole-zero pair arranged as an allpass filter.

2. FIRST-ORDER ALLPASS PROPERTIES

Consider a pole p in the z -plane at normalized radian frequency¹ θ and radius ρ , with a complementary zero at $\zeta = 1/\bar{p}$. That is, the zero is at the same frequency θ , but inverse radius $1/\rho$. Then the pole $p = \rho \exp(j\theta)$ and zero $\zeta = (1/\rho) \exp(j\theta)$ form an allpass pair

$$G(z) = \frac{-\rho e^{-j\theta} + z^{-1}}{1 - \rho e^{j\theta} z^{-1}}, \quad (1)$$

as shown in Figure 3. The group delay of any linear time-invariant filter with transfer function $G(z)$ is defined as the negative derivative of the phase response $\angle G[\exp(j\omega)]$ with respect to frequency [15], *i.e.*,

$$\tau(\omega) \triangleq -\frac{d}{d\omega} \Im\{\log G(z)\}\Big|_{z=e^{j\omega}}, \quad (2)$$

where $\omega \in [-\pi, \pi)$, and is given in this case by

$$\tau(\omega) = \frac{1 - \rho^2}{1 + \rho^2 - 2\rho \cos(\omega - \theta)}. \quad (3)$$

As illustrated in Figure 3, the group delay $\tau(\omega)$ is symmetric in frequency about the pole-frequency θ , with a peak delay of

$$\max_{\omega} \{\tau(\omega)\} = \frac{1 + \rho}{1 - \rho}, \quad (4)$$

which is achieved at the pole frequency $\omega = \theta$, and a minimum delay of

$$\min_{\omega} \{\tau(\omega)\} = \frac{1 - \rho}{1 + \rho}, \quad (5)$$

which is attained half way around the unit circle at $\omega = \theta + \pi$.

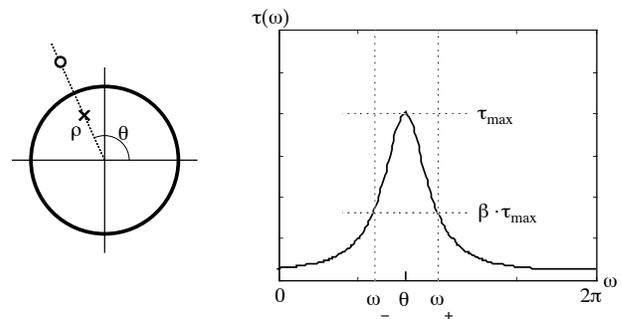


Figure 3: Allpass filter pole, zero. Group delay.

Now, what happens to the group delay $\tau(\omega)$ as the pole radius ρ is varied? Referring to Figure 4, as the pole moves towards the unit circle, the peak delay increases, while the bandwidth over which the delay is large decreases, concentrating around the pole frequency θ .

We might expect the group-delay peak to increase as the pole approaches the unit circle, but why does it also become narrow? What’s going on is that there is only so much group delay to “go around”—more precisely, the integral of the group delay around the unit circle is always 2π . Since the group delay is the negative derivative of the phase response with respect to frequency, its

¹We will assume the sampling rate is 1 so that the normalized radian frequency θ ranges between $-\pi$ and π .

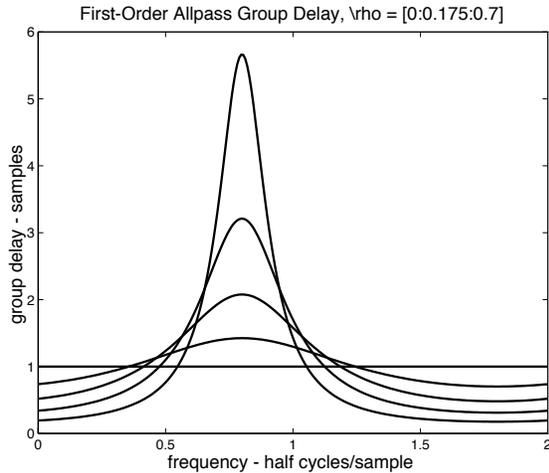


Figure 4: First-order allpass group delay, various ρ .

integral around the unit circle is simply minus the phase accumulated during one traversal of the unit circle, which is 2π per pole, regardless of where the pole is inside the unit circle:

$$\int_0^{2\pi} \tau(\omega) d\omega = \varphi(0) - \varphi(2\pi) = 2\pi \quad (6)$$

Note that only the pole contributes to this integral because the zero lies outside the unit circle. These facts can be readily seen by considering the graphical method for evaluating the phase response of a digital filter [15].

3. DISPERSION FILTER DESIGN

Recall that the goal is to design an allpass filter $D(z)$ to match a desired frequency-dependent delay, $\delta(\omega)$. The approach taken here constructs the allpass filter from (complex) first-order sections, taking advantage of the fact that each section has a 2π integrated delay, and a delay peak which may be arbitrarily located and scaled. The filter is formed by dividing the desired group delay into frequency bands, each having area 2π , as shown in Figure 5, and then modeling each delay band with its own allpass section.

The design procedure is as follows:

1. Add a constant delay to the desired frequency-dependent delay $\delta(\omega)$ so that it integrates to a desired multiple of 2π , call it N , where N is the desired allpass order.
2. Starting at DC, divide $\delta(\omega)$ into 2π -area frequency bands, as illustrated in Figure 5.
3. Fit a first-order (complex) allpass section $G_n(z)$ to each band as described below.
4. Cascade the first-order sections to form the allpass filter,

$$D(z) = \prod_{n=1}^N G_n(z). \quad (7)$$

What remains is to find the pole location for each band. The idea is to have the pole-zero pair delay $\tau(\omega)$ approximate the group

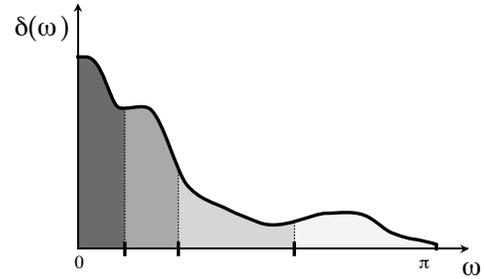


Figure 5: Segmenting $\delta(\omega)$ into 2π -area bands.

delay $\delta(\omega)$ in that band, and be small outside the band. The pole frequency is taken to be the band midpoint,

$$\theta = (\omega_+ + \omega_-)/2, \quad (8)$$

where the frequencies ω_{\pm} denote the left and right band edges. The pole radius is chosen so that the group delay at either band edge is a fraction β of the peak group delay, as illustrated in Figure 3. This controls the width of the group delay peak in each band, and determines a tradeoff between smoothness of fit to the desired group delay and the maximum “slew rate” of the allpass group delay (*i.e.*, its ability to follow small-bandwidth features in the desired group delay). Setting

$$\tau(\omega_{\pm}) = \beta \cdot \max_{\omega} \tau(\omega), \quad (9)$$

and using (3) and (4), we find

$$\rho(\beta) = \eta - [\eta^2 - 1]^{\frac{1}{2}}, \quad (10)$$

where

$$\eta = \frac{1 - \beta \cos \Delta}{1 - \beta}, \quad \Delta = (\omega_+ - \omega_-)/2. \quad (11)$$

Note that in the second step of the design procedure above, the initial band edge was set to zero. As the group delay $\delta(\omega)$ is even in frequency for real filters, this choice leads to first-order allpass sections appearing as complex conjugate pairs which may be combined to form biquads having real coefficients. Allpass filters with real coefficients also result by choosing the first band to be centered on DC. This can be done by setting the first band edge frequency to that at which the integral of $\delta(\omega)$ from DC is π . In this case, there will be two first-order allpass sections with real poles (one at DC and one at the Nyquist limit), and the rest will appear as complex conjugate pairs.

The list of band edges encodes all relevant delay information, with the band filters separately computed from their band-edge frequencies and a user-supplied β . As a result the design method is very efficient, requiring little more than an evaluation of (10) and (8) per designed biquad. It is also numerically robust, with numeric requirements nearly independent of model order.

A fair number of bands is often needed to capture the behavior of the desired delay, and in this case, the bands will be everywhere sufficiently narrow that $\rho(\beta)$ may be approximated by

$$\rho(\beta) \approx 1 - \left[\frac{\beta}{1 - \beta} \right]^{\frac{1}{2}} \Delta, \quad \Delta \ll 1. \quad (12)$$

The overlap parameter β is generally supplied by the user and independent of frequency, so that the root $[\beta/(1-\beta)]^{\frac{1}{2}}$ may be pre-computed. Under these conditions, the filter design is less costly than its implementation, and real-time manipulation of the frequency-dependent delay is inexpensive. Furthermore, it is well known that the coefficients of stable biquad sections can be interpolated from one to another without obtaining unstable intermediate filters.

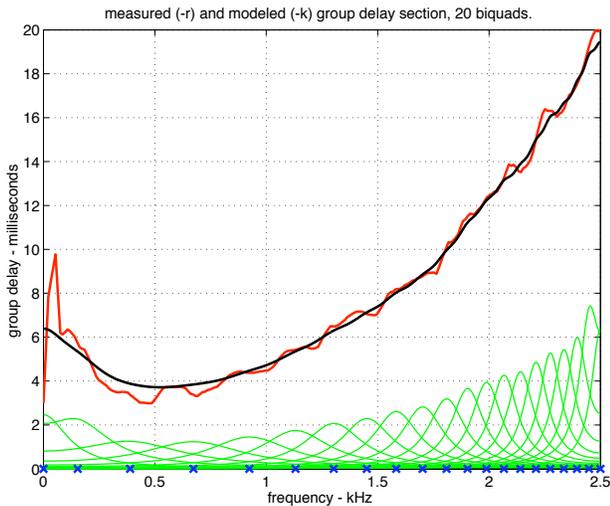


Figure 6: Spring element group delay model.

As an example design, Figure 6 shows a desired delay characteristic $\delta(\omega)$, the frequency-dependent portion of the measured time delay for a single traversal of a spring reverberator element. Frequencies in the neighborhood of 500 Hz arrive first, with delay increasing away from 500 Hz.

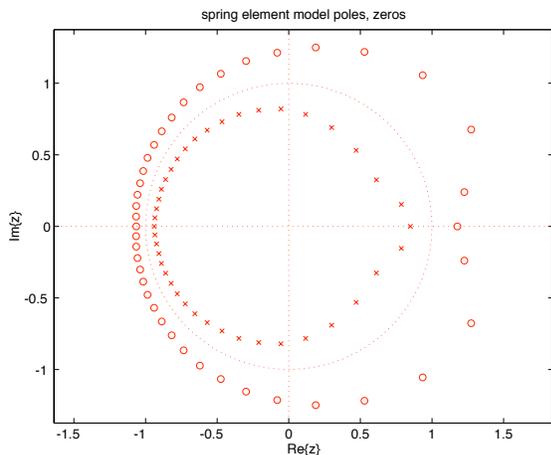


Figure 7: Spring element model poles and zeros.

Also shown in Figure 6 is the group delay of a 20-biquad allpass filter designed to match the spring element delay; it's essentially a smoothed version of the desired delay. The pole frequencies of the designed allpass are shown on the frequency axis, along

with their associated group delays. In Figure 7, the allpass pole and zero locations are shown in the z -plane, the poles tracing a smooth loop inside the unit circle. Where the delay is large, the poles and zeros are more closely spaced, and are closer to the unit circle. This is consistent with the notion that the greater the delay, the smaller the bandwidth required to achieve a 2π delay integral. That the poles are closer to the unit circle in the presence of narrowly spaced bands can be seen from (12), where the pole distance to the unit circle is roughly proportional to bandwidth.

4. DISPERSION FILTER DESIGN TRADEOFFS

There are several parameters which may be adjusted to provide an improved fit to the desired group delay characteristic.

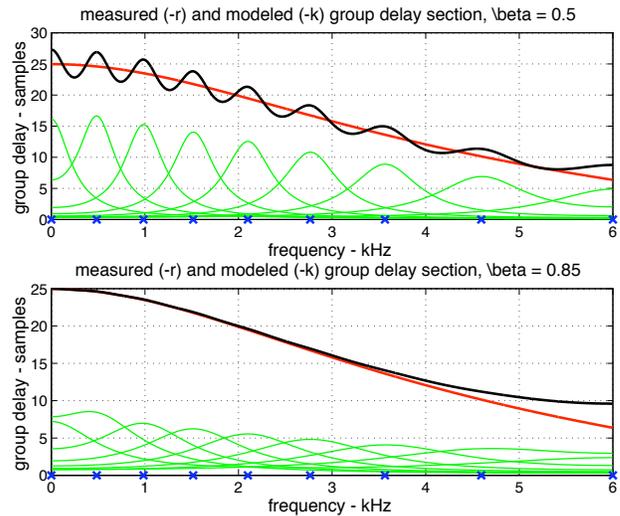


Figure 8: Example designs, $\beta = \{0.5, 0.85\}$.

The parameter β determines the extent to which successive band group delays overlap. Values of β close to one produce smooth group delays, whereas small values of β produce rippled group delays, but allow the designed filter to track narrow-bandwidth features such as sharp transitions from one delay to another. This is seen in Figure 8, showing the frequency-dependent delay of a length of piano string modeled using $\beta = 0.5$ and $\beta = 0.85$. Note how the higher β gives a smoother fit, but a greater “tracking error” near the Nyquist limit.

Since the band allpass sections are separately computed, there is no barrier to making β a function of frequency. For instance, β could be adjusted in proportion to a local measure of the smoothness of $\delta(\omega)$.

Adding a pure delay δ_0 to the desired group delay $\delta(\omega)$ allows additional, more closely spaced allpass sections to be used, and provides a more accurate fit. Figure 9 shows a dispersive delay modeled using five biquads (top) and ten biquads (bottom). The additional biquads provided when the pure delay is added result in a better fit across the band.

If computational resources are limited, it may be desired to model the group delay only in a band of interest. For instance, when designing audio dispersion filters for vibrating-string models, it is typically most cost-effective to obtain an allpass filter that

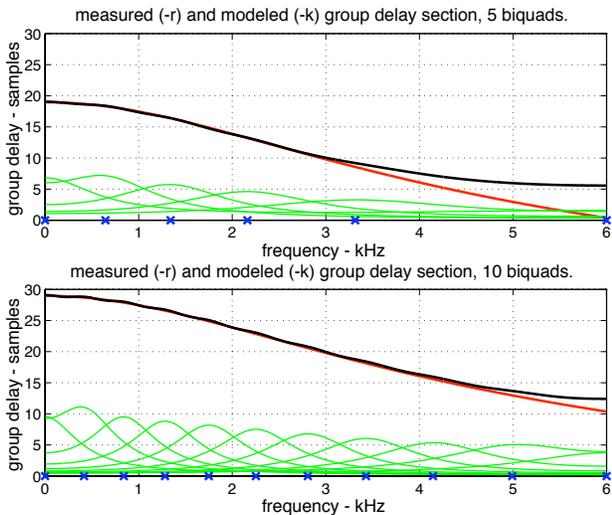


Figure 9: Example designs, $\delta_0 = \{0, 10\}$ samples.

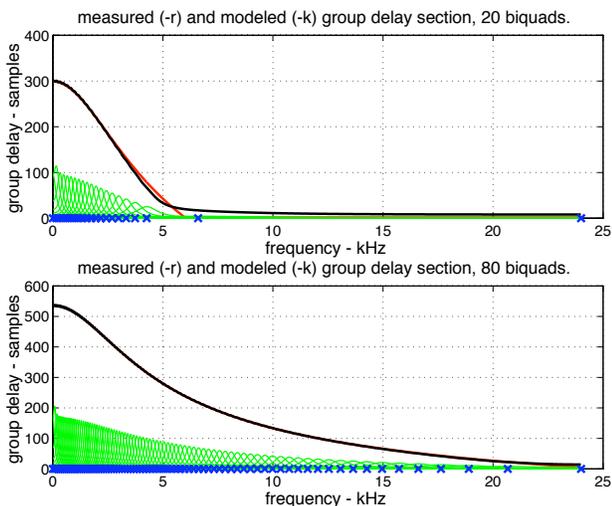


Figure 10: Example designs, $\{6, 24\}$ kHz bandwidth.

correctly tunes only the lowest-frequency partial overtones, where the number of partials correctly tuned is significantly less than the total number of partials present, as in [7]. In Figure 10, we see that 80 biquads are needed to model the delay out to 24 kHz, whereas only 20 biquads are needed to model the delay out to 6 kHz. Such a filter would be efficient for implementing the string dispersion needed for a low note on a piano.

5. PIANO-STRING DISPERSION FILTER

Figure 11 shows the impulse response based on measurements from note F1 of a piano-string (top) and that of the cascade of a minimum phase attenuation filter and an allpass dispersion filter (bottom) designed by the method of this paper using 64 biquad sections (see Figure 2 for string-model context). As can be seen,

the measured and model impulse responses appear virtually identical. Frequency-domain plots are omitted, as there is no visible error, except for a slight divergence (on the order of a tenth of a millisecond) at the Nyquist limit (12 kHz).

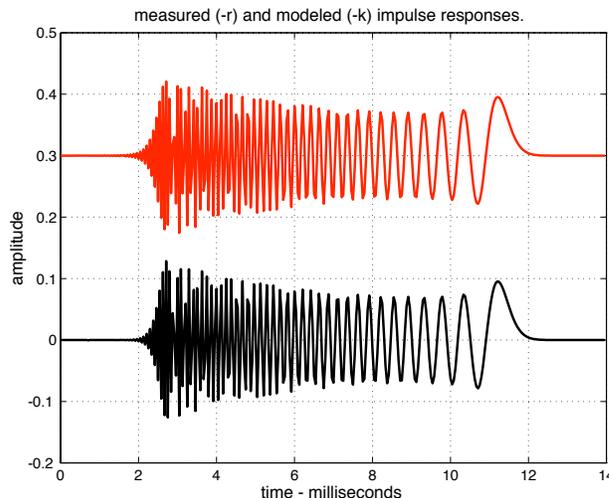


Figure 11: Order 128 impulse-response fit for a piano string, note F1.

6. SUMMARY AND FUTURE WORK

In this paper, we described a new method for allpass dispersion filter design having the following features:

- Simple and fast
- Model order automatically determined
- Filters computed in factored biquad form
- Applicable to variable allpass designs in real time
- Effective at extremely high orders (numerically robust)

Example designs were shown for stiff string and spring-reverb modeling. Other applications to consider include stiff strings in other instruments (e.g., cello), springs, and dispersive acoustic tubes. More generally, the method can be useful for high-order group delay equalization.

For future work, we think it could be effective to use this method to initialize one of several optimal methods for group-delay allpass design, to see if their convergence and numerical performance can be improved. Another idea is to model the pole locus as points along a parametrized curve, such as a series of spline curves, and to minimize group-delay error with respect to certain parameters of such a “pole locus curve”; while such a constrained locus of poles cannot be expected to be optimal in any global sense, the resulting optimized design may yield a useful improvement over the noniterative starting-point described in this paper. For this and other variations, it can help to carry out optimizations in smaller band slices corresponding to cascaded allpass sections. Finally, we think time-varying group-delay filter applications appear promising.

7. REFERENCES

- [1] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments, 2nd Edition*. New York: Springer Verlag, 1998.
- [2] J. O. Smith, III, *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects*. [Online] <http://ccrma.stanford.edu/~jos/pasp/>, Mar. 2006.
- [3] B. Yegnanarayana, "Design of recursive group-delay filters by autoregressive modeling," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 30, no. 4, pp. 632–637, Aug. 1982.
- [4] G. R. Reddy and M. N. S. Swamy, "Digital all-pass filter design through discrete hilbert transform," in *Proc. IEEE Int. Conf. Acoust., Speech, and Sig. Proc. (ICASSP'90)*, Albuquerque, USA, 1998.
- [5] A. V. Oppenheim and R. W. Schaffer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.
- [6] A. Paladin and D. Rocchesso, "A dispersive resonator in real-time on MARS workstation," in *Proc. Int. Comp. Music Conf. (ICMC'92)*, San Francisco, USA, 1992, pp. 146–149.
- [7] D. Rocchesso and F. Scalton, "Accurate dispersion simulation for piano strings," in *Proc. Nordic Acoust. Meeting (NAM'96)*, Helsinki, Finland, June 12-14 1996, pp. 407–414.
- [8] 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—I: Fundamental Theory and Applications*, vol. 41, no. 1, pp. 40–48, 1994.
- [9] J. O. Smith III and J. Abel, "Bark and ERB bilinear transform," *IEEE Trans. Speech and Audio Proc.*, vol. 7, no. 6, pp. 697–708, Nov. 1999.
- [10] M. Lang, "Allpass filter design and applications," *IEEE Trans. Sig. Proc.*, vol. 46, no. 9, pp. 2505–2514, 1998.
- [11] J. Bensa, "Stiff piano string modeling: Computational comparison between finite differences and digital waveguide," in *Proc. 148th Meeting Acoust. Soc. Am.*, San Diego, USA, Nov. 2004, nov. 15–19.
- [12] J. Rauhala and V. Välimäki, "Tunable dispersion filter design for piano synthesis," *IEEE Sig. Proc. Letters*, vol. 13, no. 5, pp. 253–256, May 2006.
- [13] Z. Jing, "A new method for digital all-pass filter design," *IEEE Trans. Acoust., Speech, and Signal Proc.*, vol. 34, no. 11, pp. 1557–1564, Nov. 1987.
- [14] J. S. Abel, J. O. Smith, and J. Bensa, "An allpass filter design method with application to piano string synthesis," in *Proc. 149th Meeting Acoust. Soc. Am.*, Vancouver, Canada, May 2005.
- [15] J. O. Smith, III, *Introduction to Digital Filters*. [Online] <http://ccrma.stanford.edu/~jos/filters/>, Sept. 2005, online book.