LOW-ORDER ALLPASS INTERPOLATED DELAY LOOPS

Nelson Lee and Julius O. Smith III

CCRMA,
Stanford University
Stanford, USA
{nalee,jos}@ccrma.stanford.edu

ABSTRACT

This paper presents empirical and theoretical results for a delay line cascaded with a second-order allpass filter in a feedback loop. Though such a structure has been used for years to model stiff vibrating strings, the complete range of behavior of such a structure has not been fully described and analyzed. As shown in this paper, in addition to the desired behavior of providing a frequency-dependent delay line length, other phenomena may occur, such as "beating" or "mode splitting." Associated analysis simulation results are presented.

1. INTRODUCTION

In physical-modeling synthesis, the audible inharmonicity produced from stiff-stringed instruments such as the guitar and piano need to be reproduced. In [1, 2], formulae for coefficients of inharmonicity perceived by test subjects were presented. In that work, it was established that high-fidelity physical models of relatively stiff strings require some stiffness simulation in the low-frequency range, certainly for most piano strings, and even for the lowest notes of an acoustic guitar. In [3], the parameters of the physical-model of wave propagation along a string are related to the parameters of a digital waveguide (ie., the inharmonicity coefficient is related to the desired phase response of the waveguide). [4] casts a complementary mathematical formulation to [3] and extends the digital waveguide model to account for a point source excitation of a portion of the modeled string.

In the literature, there are various solutions for designing an allpass filter to simulate the dispersive wave propagation of a stiff string. In [5], convincing stiff-string tones were obtained using Yegnanarana's method [6] for designing rather high-order allpass filters with a prescribed group delay. In [7], a computationally simpler method was introduced based on cascading real, identical, first-order allpass filters, and a physical interpretation involving masses and springs was given. In [8], a weighted-least-squares formulation adapted from [9] was used to fit the coefficients of an allpass filter to a desired phase response over a Bark-warped frequency scale, yielding accurate tuning of the first several tens of piano-string overtones using an allpass order of 20 or less. In [10], an optimization procedure proposed by M. Lang [11] was applied to obtain the optimal Chebyshev allpass filter, where the optimality was with respect to the phase response weighted by inverse frequency; due to numerical difficulties, it was necessary to divide the desired phase by N, find the optimal allpass for that case, and use a cascade of N identical allpass filters for the final model. An elementary and robust method for designing allpass filters of very high order was presented in [12], and a more complete summary of numerical difficulties with prior methods is given there. Additionally, extremely accurate results were reported in [12] for simulating the note F1 on a piano using an order 128 allpass in a delay loop (designed directly as a cascade of 64 second-order biquad sections).

In [13, 14], closed-form Thiran allpass filters were used to model a desired phase-delay by minimizing the difference between the theoretical prediction of a stiff string's harmonic frequencies and those of an allpass-interpolated delay loop. Similarly, in [15], the residual signal of an inverse-FIR-filtered inharmonic piano tone was minimized with respect to phase delay, thereby matching the partials of the recorded tone. In [16], a delay line cascaded with a second-order allpass filter in a feedback loop was investigated for modeling inharmonic bell-sounds.

In view of the above-cited literature, we know how to design very high-order allpass filters that model dispersive strings with great accuracy (e.g., [12]), and we have several methods for lower order allpass design that yield perceptually good models, but which require significant computation, and/or present numerical difficulties.

This paper addresses *low-order* dispersion allpass design by means of a *local* perturbation in the tuning of a partial overtone using a second-order allpass section. The basic idea is to tune the allpass pole close to the string overtone in order to "pull" it sharp, as needed for string stiffness simulation. In other words, we use a second-order allpass filter with real coefficients in a delay loop to adjust the tuning of a single overtone. Theoretical and simulated results are presented. In particular, a graphical solution technique and root-locus analyses are presented, and a physical interpretation is noted.

2. PHASE, PHASE DELAY, OR GROUP DELAY?

In the above-cited literature, allpass design techniques were proposed that explicitly optimize (1) the unwrapped phase response [9, 10], (2) the group delay [6, 12], or (3) the phase delay of the allpass [13, 14, 15]. These three functions of the allpass phase response are related but not equivalent. The specific choice is typically dictated by the mathematics of the design method itself. In this section, we present a derivation to show that it is *phase delay* to be directly optimized when designing dispersion allpass filters.

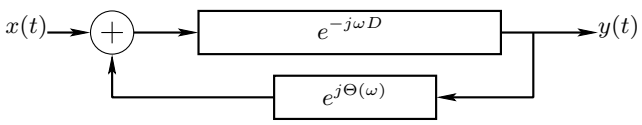

Figure 1: Diagram of Allpass Delay Loop to be Analyzed.

Figure 1 shows a basic diagram of a delay line with an allpass filter in the feedback loop. For simplicity, we consider the continuous-time case. The frequency response of any allpass filter may be expressed as $\exp(j\Theta(\omega))$, where $\Theta(\omega)$ denotes the phase response (since the magnitude frequency response is 1 at all frequencies for an allpass filter). The frequency response of a delay-line D seconds long is $\exp(-j\omega D)$, where ω denotes radian frequency. Thus, the frequency response of the system in Fig. 1 is given by Mason's rule (or direct derivation) by

$$H(j\omega) = \frac{Y(j\omega)}{X(j\omega)} = \frac{e^{-j\omega D}}{1 - e^{-j\omega D}e^{j\Theta(\omega)}} \quad (1)$$

$$= \frac{1}{e^{j\omega D} - e^{j\Theta(\omega)}} \quad (2)$$

$$= \frac{e^{-j\Theta(\omega)}}{e^{j\omega\left[D - \frac{\Theta(\omega)}{\omega}\right]} - 1} \quad (3)$$

$$= \frac{e^{-j\Theta(\omega)}}{e^{j\omega[D+P(\omega)]} - 1} \quad (4)$$

where $P(\omega)$ denotes the *phase delay* of the allpass filter, defined by

$$P(\omega) = -\frac{\Theta(\omega)}{\omega}. \quad (5)$$

Thus, the phase delay effectively adds to the delay line length, giving it a frequency-dependent length $D + P(\omega)$ as desired.

The poles of the system are all on the $j\omega$ frequency axis at radian frequencies ω_k for which

$$\omega_k \cdot [D + P(\omega_k)] = k \cdot 2\pi, \quad (6)$$

for $k = 0, 1, 2, \ldots$. Thus, we have

$$\begin{aligned}
\omega_0 &= 0 \\
\omega_1 &= \frac{2\pi}{D + P(\omega_1)} \\
\omega_2 &= 2 \cdot \frac{2\pi}{D + P(\omega_2)} \\
&\cdots \\
\omega_k &= k \cdot \frac{2\pi}{D + P(\omega_k)} \\
&\cdots
\end{aligned}$$

and so on. An exact solution is nontrivial since ω_k appears in two places, and $P(\omega_k)$ is nonlinear, in general. One solution is to reformulate the problem in discrete time and factor the closed-loop transfer-function denominator polynomial $\tilde{A}(z) - z^{-N}A(z)$, where N is the delay-line length and the (finite-order) allpass transfer function is $\tilde{A}(z)/A(z)$. However, polynomial factoring is a nonlinear, iterative algorithm, and for large N, numerical root-finders can run into trouble. A general graphical solution technique is to plot the left- and right-hand sides of the equation and find all points of intersection to find all solutions.

2.1. Graphical Solution

$$H(z) = \frac{\rho^2 - 2\rho\cos(\theta)z^{-1} + z^{-2}}{1 - 2\rho\cos(\theta)z^{-1} + \rho^2 z^{-2}} \quad (7)$$

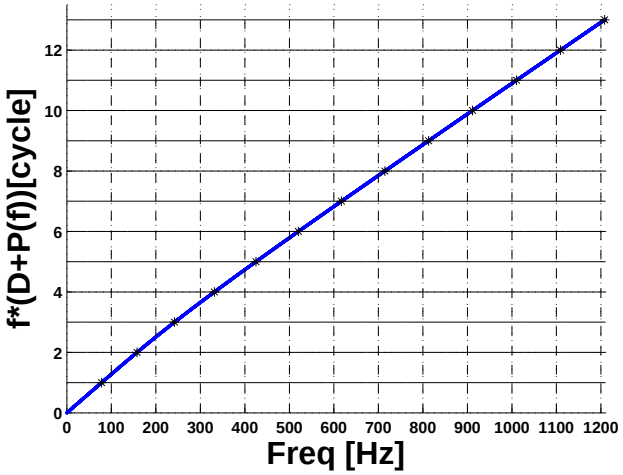

Figure 2: Use of the graphical method to determine the frequencies of the poles of the allpass interpolated delay loop. In this example, $D = (100Hz)^{-1}$. The pole of the allpass filter has frequency 100Hz and radius 0.9.

To illustrate the graphical solution technique for delay $D = (100Hz)^{-1}$, we used a second-order allpass with real coefficients. The form of the allpass filter is shown in Equation 7. Note that there are two parameters: the pole radius and pole frequency.

In Figure 2, the pole frequency is 100Hz, and the pole radius is 0.9. The single bold curve in the figure plots $f \cdot [D + P(f)]$, and the horizontal lines plot k for $k = 1, 2, ..., 13$. From the intersection points (crosses), we can determine the pole locations of our allpass interpolated delay loop.

Figure 3 shows solution values f_k for $k = 1, 2, ..., 5$. As shown, the peak at the fundamental (100Hz) has been split into a lower and higher frequency, while all higher partials have been shifted higher ("stretched"). As interpreted in Figure 3, ω_1 and ω_2, computed using the graphical solution technique, correspond to a split of the first mode as the allpass filter introduces two poles (one at positive and one at negative frequencies). Thus, for clarity, frequencies corresponding to the intersection of 1 and 2 with $f \cdot [D + P(f)]$ in Figure 2 are plotted as two peaks for the first *mode* of the system in Figure 3 and the frequencies of the higher *modes* are pulled sharp.

3. SIMULATION RESULTS

To verify the graphical solution, we simulated the allpass-interpolated delay loop to obtain 30 seconds of the impulse response with $f_s = 10$kHz.

We then performed an FFT of length 2^{20}, and measured the spectral magnitude peaks using quadratic interpolation. These interpolated peak locations are thus estimates of the system pole frequencies, and are compared with the frequencies determined by the graphical solution method of Section 2. As Figure 4 shows, the error for the first 13 harmonics is less than 10^{-4} Hz. The average error for the first 49 harmonics is 3.45×10^{-5} Hz.

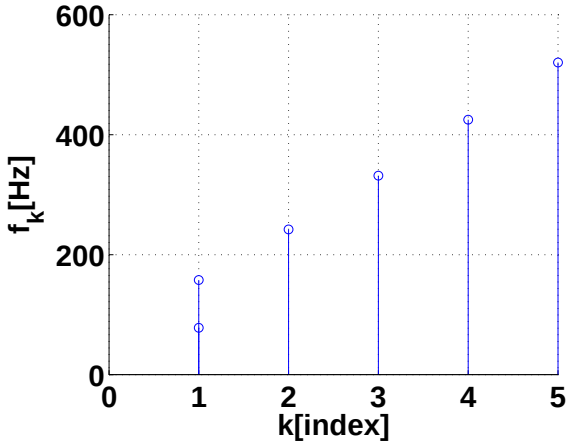

Figure 3: Values of f_k for $k = 1, 2, ..., 5$ determined from use of the graphical method for the example in Figure 2.

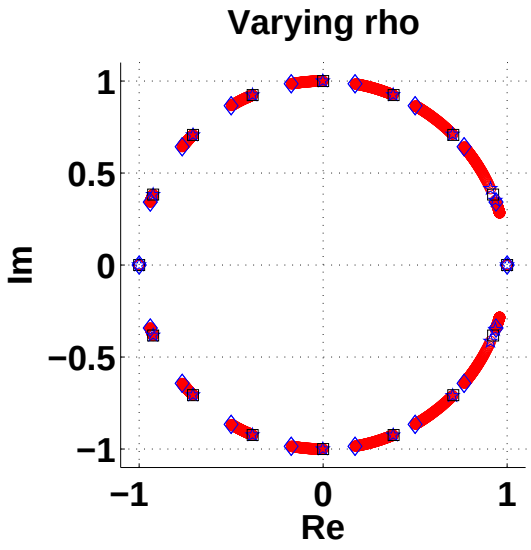

Figure 5: Varying ρ, the allpass pole's radius, from 0.0 to 1.0. The diamonds show pole locations of the system when $\rho = 0.0$ and the pentagrams show pole locations of the system when $\rho = 0.99$. Circled points show trajectories of system pole locations for intermediate values of ρ. Square points show pole locations when $\rho = 1.0$, meaning the pole and zeros of the allpass filter cancel.

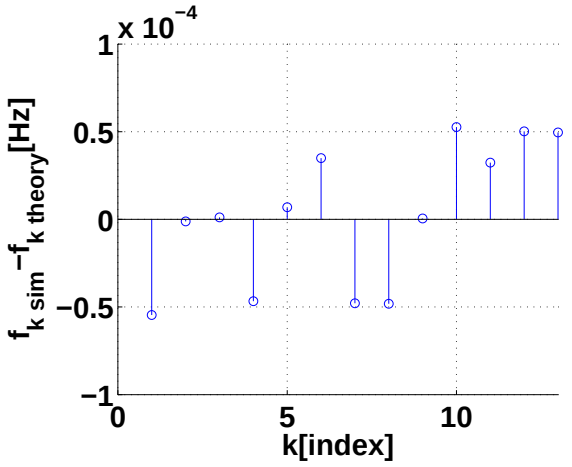

Figure 4: Error between theoretical values and simulated values of f_k for $k = 1, 2, ..., 13$ in Hz.

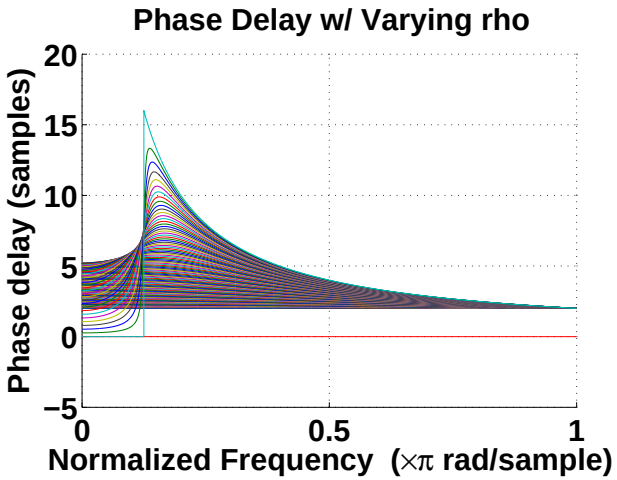

Figure 6: The phase delay of the allpass filter when $\rho = 0.0, 0.01, ..., 1.0$

4. ROOT LOCUS

Using the same second-order allpass filter as described in Section 2.1, we will present how the poles of the allpass delay loop change corresponding to different values of the pole radius. More specifically, we change ρ from 0.0 to 1.0 with a step-size of 0.01. In our system, we have a total of 16 poles from the delay line and we fix the pole location of our allpass filter to occur at $\pi/8$, corresponding to the location of the first pole of the delay line loop with non-zero frequency. Figure 5 plots the poles of the allpass interpolated delay loop with ρ varying from 0.0 to 1.0.

The diamond points correspond to the locations of the poles of the system when $\rho = 0.0$. The pentagram points correspond to the locations of the poles when $\rho = 0.99$. As ρ increases in value from 0.0 to 0.99, the poles are pulled towards $\pi/8$ as denoted by circles between the diamond and pentagram points, showing the paths of each pole's movements, thus sketching a kind of *root locus* for the system poles. Lastly, the square points correspond to the locations of the poles when $\rho = 1.0$, when the poles and zeros of the allpass filter cancel.

To understand the delay effects of the second-order allpass, we need examine its phase delay, as discussed in Section 2. Figure 6 shows the phase delay of a second-order allpass filter having poles at frequencies $\omega T = \pm\pi/8$, and common pole radius varying from $\rho = 0.0$ to $\rho = 1.0$. As shown, when ρ is close to 1, the phase delay is small for $\omega T \ll \pi/8$ radians per sample, peaks (at around 16 samples of delay) near $\omega T = \pi/8$, and decreases toward two samples delay for $\omega T > \pi/8$.

In particular, when $\rho = 0.0$, the allpass filter acts as a pure delay of two samples. As ρ increases, the poles at frequencies greater than the allpass's pole location shift down in frequency. This results from the increasing phase delay at these frequencies as ρ increases (see Fig. 6). When ρ is nearly 1.0, as Fig. 6 shows, the phase delay remains flat at frequencies from dc up to the allpass's pole's location, where it peaks. The phase delay then decreases towards two samples at higher frequencies. The observations are consistent with the root locus plot of Fig. 5. Qualitatively, the second-order allpass filter introduces two poles into the system for ρ not equal to 1.0. Thus, the system examined has 16 poles total when $\rho = 1.0$ and 18 poles otherwise.

5. PHYSICAL INTERPRETATION

As shown in [7], a first-order allpass filter in a delay loop can be interpreted physically as a spring termination at one end of an ideal string (with the other end being rigidly terminated). Similarly, several cascaded first-order allpasses can be interpreted physically as a terminating mass-spring chain in which all poles are real (and equal in the case of that paper). In our use of second-order allpass filters with *complex poles*, we have the physical interpretation of an ideal string terminated by *resonant* mass-spring systems. In the case of a single second-order allpass used to retune only the fundamental frequency, say, we have an ideal string terminated by a mass-spring resonator, where the resonance is tuned very close to the fundamental frequency of the ideal string. In this situation, we should not be surprised to encounter *coupled resonator effects*, such as have been clearly mapped out (with regard to coupled piano strings) by Weinreich [17]. In particular, when the coupling impedance is purely reactive, as it is in our situation (no damping), we can expect mode splitting and beating, as occurs, for example, when coupled piano strings are tuned too far apart such that they "beat", or in the "wolf note" of a cello.

6. CONCLUSIONS

In this paper we explored aspects of a delay loop interpolated using second-order allpass sections to pull the tuning of individual modes. We presented an s-plane derivation showing that phase-delay error, suitably weighted psychoacoustically, is the direct metric for stiff string modeling. We investigated the effect of the phase delay on the poles of the overall system, and introduced a graphical technique for finding all the poles. Finally, a physical interpretation was noted that makes the observed results predictable and understood.

7. REFERENCES

[1] H. Järveläinen and M. Karjalainen, "Perceptibility of inharmonicity in the acoustic guitar," in *ACTA Acustica United with Acustica*, 2006, vol. 92, pp. 842–847.

[2] V. Välimäki H. Järveläinen and M. Karjalainen, "Audibility of inharmonicty in string instrument sounds, and implications to digital sound synthesis," in *Proceedings of the 1999 International Computer Music Conference*, Beijing, China, Oct. 22-27, 1999, pp. 359–362.

[3] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith III, "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides," in *Journal of the Acoustical Society of America*, 2003, vol. 114.

[4] É. Ducasse, "On waveguide modeling of stiff piano strings," in *Journal of the Acoustical Society of America*, 2005, vol. 118, pp. 1776–1781.

[5] A. Paladin and D. Rocchesso, "A dispersive resonator in real-time on mars workstation," in *Proceedings of the 1992 International Computer Music Conference*. 1992, pp. 146–149, Computer Music Association.

[6] B. Yegnanarayana, "Design of recursive group-delay filters by autoregressive modeling," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, no. 4, pp. 632–637, Aug. 1982.

[7] S. A. Van Duyne and J. O. Smith III, "A simplified approach to modeling dispersion caused by stiffness in strings and plates," in *Proceedings of the 1994 International Computer Music Conference*. 1994, pp. 407–410, Computer Music Association.

[8] D. Rocchesso and F. Scalcon, "Accurate dispersion simulation for piano strings," in *Proc. Nordic Acoustical Meeting*, Helsinki, Finland, June 12-14, 1996.

[9] M. Lang and T. I. Laakso, "Simple and robust method for the design of allpass filters usingleast-squares phase error criterion," in *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 1994, vol. 41, pp. 40–48.

[10] J. Bensa, S. Bilbao, R. Kronland-Martinet, J. O. Smith III, and T. Voinier, "Computational modeling of stiff piano strings using digital waveguides and finite differences," in *ACTA Acustica United with Acustica*, 2005, vol. 91, pp. 289–298.

[11] M. Lang, "Allpass filter design and applications," in *IEEE Transactions on Signal Processing*, 1998, vol. 46, pp. 2505–2514.

[12] J. S. Abel and J. O. Smith III, "Robust design of very high-order allpass dispersion filters," in *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, September 18-20, 2006.

[13] J. Rauhala and V. Välimäki, "Tunable dispersion filter design for piano synthesis," in *IEEE Signal Processing Letters*, May 2006, vol. 13, pp. 253–256.

[14] J. Rauhala and V. Välimäki, "Dispersion modeling in waveguide piano synthesis using tunable allpass filters," in *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06)*, September 18-20, 2006.

[15] H. M. Lehtonen, "Analysis of piano tones using an inharmonic inverse comb filter," in *Proc. of the 11th Int. Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, September 1-4, 2008.

[16] M. Karjalainen, V. Välimäki, and P. A. A. Esquef, "Efficient modeling and synthesis of bell-like sounds," in *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, September 26 2002, pp. 13–20.

[17] G. Weinreich, "Coupled piano strings," *Journal of the Acoustical Society of America*, vol. 62, no. 6, pp. 1474–1484, Dec 1977, see also [23] and *Scientific American*, vol. 240, p. 94, 1979.

[18] J. O. Smith III, *Introduction to Digital Filters with Audio Applications*, W3K Publishing, http://www.w3k.org/books/, 2007.

[19] E. Wang and B. T. G. Tan, "Application of wavelets to onset transients and inharmonicity of piano tones," in *Journal of the Audio Engineering Society*, May 2008, vol. 56, pp. 381–392.

[20] J. Rauhala, "The beating equalizer and its application to the synthesis and modification of piano tones," in *Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07)*, Bordeaux, France, September 10-15, 2007.

[21] J. Rauhala, H. M. Lehtonen, and V. Välimäki, "Fast automatic inharmonicity estimation algorithm," in *Journal of the Acoustical Society of America*, 2007, vol. 121.

[22] H. M. Lehtonen, V. Välimäki, and T. I. Laakso, "Canceling and selecting partials from musical tones using fractional-delay filters," in *Computer Music Journal*, 2008, vol. 32, pp. 43–56.

[23] A. Askenfelt, Ed., *Five Lectures on the Acoustics of the Piano*, Royal Swedish Academy of Music, Stockholm, 1990, lectures by H. A. Conklin, Anders Askenfelt and E. Jansson, D. E. Hall, G. Weinreich, and K. Wogram. Sound example CD included. Publication number 64. `http://-www.speech.kth.se/music/5_lectures/`.