Control of Frequency and Decay in Oscillating Filters Using Multirate Techniques

Frode Holm
Department of Computer Science
Center for Computer Music Research and Composition
University of California Santa Barbara, CA 93106
ho.l@cs.ucsb.edu

Abstract

Algorithms and theory for controlling frequency, bandwidth and decay-rate in the Karpus-Strong string model are presented. The methods rely on resampling the delay-line and the loop-gain filter instead of the more commonly used filters for fractional-delay and decay-stretching/shortening. This leads to systems with markedly different characteristics from those where a fixed rate is used.

The main focus of this paper is the new parameterization that the multirate view leads to.

1. Introduction

Traditionally, control of frequency in oscillating filter structures, like e.g. in the Karpus-Strong (K-S) algorithm [Karpus and Strong, 1983], has been accomplished through a combination of changing the length of the delay-line and the use of an FIR or IIR tuning filter for obtaining fractional delays [Jaffe and Smith, 1983; Karjalainen and Laine, 1991]. Various tradeoffs, usually linear phase vs. flat frequency response, are involved in choosing between FIR and IIR filters for this purpose.

Comparatively little attention has been given to the technique of using sample-rate conversion for achieving the same end, although a good example can be found in [Vercoe, 1988]. On the surface, the main difference between the two approaches is that in the multirate case the delay-length is kept constant, with frequency controlled through familiar table lookup techniques. An interpolation step in the form of an FIR filter is usually required in this case as well to obtain sample-values at non-integer positions. The associated tradeoffs now concern signal-to-noise ratios and aliasing rather than phase and spectrum issues, since this step does not modify the signal in the delay-line. These topics will not be discussed further in this paper (for an in-depth study of aliasing the reader is referred to [Vaidyanathan, 1993]).

What is perhaps not so well known is that with the multirate method one gets an additional parameter for "free" - the length of the delay-line can now be used to control bandwidths and decay-rate without introducing additional filters. This paper outlines how to construct practical algorithms from these considerations.

In the following (except section 2), linear interpolation is always assumed for the rate-conversion. For more accurate results, a method such as that proposed in [Smith and Gouyet, 1984] should be considered.

2. Frequency domain representation

It can be shown, [Openheim and Schaffer, 1989] and [Crochiere and Rabiner, 1983], that if the output sample-rate is kept constant, the effect of changing the sample-rate of a signal by a rational factor $f = M/L$, is to scale its spectrum by this factor (assuming aliasing constraints are met). Thus, if we have a signal, represented by $y(n)$, and resample it through $y(nf)$ (assuming a perfect interpolation), the frequencies in $y$ will be shifted accordingly. This is illustrated in fig.1. The gain factor $c$ depends on the interpolation step used with $X_c$ (sample-rate increase).

$$X_c(e^{jw}) = cX(e^{jw/f})$$

$$y(n) \rightarrow X_c(e^{jw}) = X_c(e^{jw/f})$$

Of course, this fact is well-known from the table lookup oscillator and commercial samplers, but what is important to note here is that this reasoning also applies to whole systems including recursive ones [Holm, 1994]. If we take the classic K-S algorithm as an example and "resample" the
3. Karglau-Strong revisited

A straightforward way to implement K-S using sample-rate change, is to use table lookup to get output samples from the delay-line and for each pass through the delay, i.e. for each fundamental period, ensure that all the samples in the delay-line have been filtered. This can either be done concurrently with the lookup operation or as a separate step. The former is illustrated in fig.2 and an example of the latter can be found in the 'pluck' unit generator in CSOUND [Vcerce, 1988].

\[ F = \frac{R}{p + 1/2} \text{ Hz}; \quad f = \frac{F(p + 1/2)}{R} \]  

This is the same formula as the one used for calculating the frequency increment in a table lookup oscillator.

In this picture, F is now independent of both p and R, since any changes in these quantities will be compensated for by f. Assuming a constant output sampling rate R, we now need to know what the effect of changing p has. To answer this, it is sufficient to consider the scaling of the frequency response of the K-S averaging filter by f. This filter falls off to 0 at the Nyquist rate, \( p\pi/2 \). By resampling, this frequency becomes

\[ f'f = \frac{R}{2} = \frac{F(p + 1/2)}{2} \]  

Fig.2 Filtering in lock-step with lookup, f=1.

In the original K-S algorithm the fundamental frequency, F, is given by (with sampling rate R and delay length \( p \))

\[ F = \frac{R}{p + 1/2} \]  

With re-sampling by a factor of \( f \), we get

It now becomes clear that by varying \( p \), first of all, the initial bandwidth of the resulting signal is affected. A larger value of \( p \) means a wider filter response applied to the signal used to fill the delay buffer at start-up time, usually white noise. Furthermore, as this filter is applied over and over in the course of oscillation, higher frequencies take a longer time to decay the larger \( p \) is (and vice versa). This is readily verified by contemplating figs. 3 & 4 and eq. (3). It may be a fortuitous circumstance that this parameter has a behaviour that corresponds so well with the physical reality observed in strings and trees, maybe not (physical modelling purses: take note!). The harder the pluck...
(by increasing \( p \)), the more initial high-frequency energy and the longer the decay of the harmonics - the more gentle the attack. For \( p = 2 \), the more the opposite is true.

In [Mw	= 1990] the decay characteristics for the basic K-S algorithm is tabulated. To obtain the decay values in the multirate view, the same reasoning as that detailed in [Saffe and Smith, 1983] can be applied, bearing in mind the required variable changes. The attenuation factor \( a(t) \) is given by:

\[
\begin{align*}
\frac{\partial a}{\partial t} &= \cos(\frac{\pi p f}{F}) \\
\frac{\partial a}{\partial t} &= \cos(\frac{\pi f}{F})
\end{align*}
\]

(4)

Values for the time it takes the fundamental to decay -40dB is listed in Table 1 for various values of \( p \) and \( F \).

<table>
<thead>
<tr>
<th>( p )</th>
<th>50</th>
<th>100</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>17.6</td>
<td>3.8</td>
<td>1.8</td>
<td>0.9</td>
</tr>
<tr>
<td>50</td>
<td>48.0</td>
<td>24.9</td>
<td>4.8</td>
<td>2.4</td>
</tr>
<tr>
<td>100</td>
<td>189.4</td>
<td>94.7</td>
<td>18.9</td>
<td>9.5</td>
</tr>
<tr>
<td>200</td>
<td>752.1</td>
<td>376.1</td>
<td>75.2</td>
<td>37.6</td>
</tr>
</tbody>
</table>

Table 1. -40 dB decay in sec. for fundamental \( F \).

Since changing \( p \) determines bandwidth and decay-rate in (3) given \( F \), the same must also be true for the fundamental frequency \( F \) if \( p \) is kept constant. In practical implementation \( p \) must consequently be adjusted with pitch accordingly to obtain consistent timbres.

4. Variations on a theme.

By extending the reasoning and methods suggested above it is possible to obtain more degrees of freedom with only a negligible increase in computational costs. In particular, it is straightforward to separate \( F \) completely from the control of the decay-rate (but not the initial bandwidth).

This can be done by introducing an additional parameter, \( G \) Hz, which specifies how many times per second the filter must complete a loop. The rate at which the low-pass filter is applied, and consequently the speed of the decay, is now an independent quantity. In pseudo-C such an algorithm could look something like the following

\( i \) is the table increment associated with \( G \). Please note that the quoted statements encapsulate a whole range of operations concerning modulus, table access, interpolation, possible anti-aliasing filters etc. that would have to be addressed in a real implementation.

```c
for (n=0; n < Duration; n++)
{
    Compute filter at rate q.
    cos(\frac{i+1}{G})*\cos(\frac{n*q}{G});
    y[i] = 0.5*x*y[i]+y[(i-p+1)];
    Compute output at rate \( f \)
    *output y[n*f];
}
```

There is only one slight complication to this scheme, which is due to the fact that the averaging filter delays the signal period by a 1/2 sample each time through the loop. This must be corrected for in calculating the frequency increment \( F \) in order to avoid a "doppler" effect. For the two-point averaging filter, eq. (3) must be recast as:

\[
F = \frac{1}{2}(p+G) \quad R
\]

(5)

i.e. the delay in the buffer due to the filter is now determined by the relative speeds of the two operations, expressed by the ratio \( G/F \).

To use this algorithm in a practical setting, one would first choose \( p \), given a fundamental \( F \), to set the retimed bandwidth, then choose an appropriate value for \( G \) to set the decay-rate according to (4) with \( G \) replacing \( F \). One point that some might object to is the fact that one period of the output waveform now could contain a mixture of 2 or even more "generations" of the filter. For realistic settings of the various parameters involved, this does not seem to have any objectionable audible consequences (that I can hear), but some caution might be in order.

5. Efficiency issues

In the original K-S implementation, the computational cost is fixed and sampling rate \( R \). With multirate techniques this is no longer true, as the loop-filter in this case operates independently of \( R \). Since an operation comparable to linear interpolation at \( R \) times per sec. is involved for frequency control in both cases, efficiency for the two methods, without decay control, will be identical when \( R=F(p+1) \).

Depending on how bright one needs the string to
sound, the cost with multirate methods will be either less (low bandwidth) or more (high bandwidth).

If we add in decay-control, the balance shifts somewhat in favor of the multirate case since the only added cost here is the loss of some minor optimization possibilities, whereas with original K-S, the loop-filter becomes twice as expensive [Jaffe and Smith, 1983]. Now the two methods will be approximately equal in cost when 2R=G(p+1/2).

Again, depending on the desired decay-rate value, the multirate method may be more or less efficient.

6. Extensions

It remains to be seen how useful multirate techniques can be when considering larger and more complex systems, such as those found in complex instrument or vocal tract simulations. The part of such systems that control stationary formants will most likely not benefit much from resampling since, as we have seen, this would move the spectral shape of the affected filters.

Still, useful success of such methods in the physical modeling field have been reported. [Wu et al., 1987] and [Wright and Owens, 1993], and it is likely that more examples and theories will be forthcoming in the near future.

7. Summary

Methods for controlling frequency, bandwidth and decay-rate in the Karplus-Strong string synthesis model using multirate algorithms were presented. It was shown how this leads to a different parameterization of the model and how this makes filters for tuning and decay-sharpening/stretching unnecessary, thus eliminating problems like non-linear phase and amplitude response that is not flat. Instead issues of table-lookup noise and aliasing have to be addressed. Two possible algorithms based on these considerations were outlined.

It was also argued that, on average, the two methods have the same computational cost, the exact measure depending on actual parameter-settings which determine number of operations per sec. in the multirate case.

One method is not necessarily “better” than another, except in situations where application constraints may favor one over the other.

References


ICMC Proceedings 1994 375 Sound Synthesis Techniques
Connections between Feedback Delay Networks and Waveguide Networks for Digital Reverberation

Julius O. Smith
Center for Computer Research
in Music and Acoustics (CCRMA)
Stanford University
Email: jos@ccrma.stanford.edu

Davide Rocchesso
Centro di Sonologia Computazionale
Dipartimento di Elettronica e Informatica
Università degli Studi di Padova
Email: roc@di.unipd.it

Abstract

The order N feedback delay network (FDN) has been proposed for digital reverberation. Also proposed with similar advantages is the digital waveguide network (DWN). This paper notes that the FDN is isomorphic to a (normalized) waveguide network consisting of one (parallel) scattering junction joining N reflectively terminated branches. This correspondence gives rise to new generalizations in both cases.

The feedback delay network (FDN), depicted in Fig. 1, has been proposed for digital reverberation applications (see the companion paper [Rocchesso and Smith, 1994] for references). These structures are characterized by a set of delay lines connected in a feedback loop through a "feedback matrix."

![Figure 1: Order 3 Feedback delay network.](image)

Digital waveguide networks (DWN) have also been proposed as a starting point for digital reverberator development [Smith, 1985]. Like FDNs, DWNs make it easy to construct high-order lossless systems. Ordinarily, the lossless prototype reverberator is judged for the quality of white noise it generates in response to an impulse signal. For smooth reverberation, the white noise should sound uniform in every respect. Subsequent introduction of lowpass filters into the prototype network (e.g., applied to junction pressure) serves to set the desired reverberation time vs. frequency. Since FDNs and DWNs appear to present very different approaches for constructing lossless prototypes, it is natural to ask what connections may exist between them, and whether there may be unique advantages of one over the other.

Figure 2 illustrates an N-branch DWN which is structurally equivalent to an N-th order FDN.

The waves traveling into the junction are associated with the FDN delay line outputs $s_j(n)$, and the length of each waveguide is half the length of the corresponding FDN delay line $m_j$ (since a traveling wave must traverse the branch twice to complete a round trip from the junction to the termination and back). When $m_j$ is odd, we may replace the reflecting termination by a unit-sample delay, or we may define the branch medium such that the speed of propagation is slightly faster in one direction.

![Figure 2: Waveguide network consisting of a single scattering junction to which N branches are connected. The far end of each branch is terminated with a perfect, non-inserting reflection, indicated by a black dot.](image)

As discussed in greater detail in the companion paper, the delay-line inputs (outgoing traveling waves) are computed by multiplying the delay-line outputs (incoming traveling waves) by the $N$-by-$N$ feedback matrix $A = [a_{ij}]$:

$$s_j(n + m_j) = \sum_{j=1}^{N} a_{ij} s_i(n)$$

The above notation coincides with that used in the companion paper. By defining $s_j^T = s_j(n)$,
$r^n = n(n + m)$, and $A = [a_{ij}]$, we obtain the more usual DWN notation

$$e^n = A e^n$$

where $e^n$ is the vector of incoming travelling-wave samples arriving at the junction at time $n$, and $e^n$ is the vector of outgoing travelling-wave samples leaving the junction at time $n$, and $A$ is the scattering matrix associated with the junction.

To obtain lossless FDN's, prior work has focused on unitary feedback matrices. A matrix $A$ is said to be unitary if $A^* A = I$, where $*$ denotes transposition and complex conjugation.

Since every lossless scattering junction provides a lossless FDN matrix, all of these matrices are unitary. The answer is immediately no: Unitary scattering matrices arise only in the case of normalized waves, e.g., pressure waves which are multiplied by the square-root of the wave admittance of the waveguide in which they travel [Smith, 1987]. In such cases, the scattering matrix can be expressed as a Householder reflection

$$A = 2\left[\begin{array}{c} u^T \\ I_n \end{array}\right]\left[\begin{array}{c} u^T \\ I_n \end{array}\right]^T - I, \text{ where } u^T = [u_1^T, \ldots, u_n^T],$$

and $I$ is the identity matrix. Thus, unnormalized scattering matrices can be expressed in the form of an "oblique" Householder reflection

$$A = 2u^T \left[\begin{array}{c} u \\ I \end{array}\right] - I, \text{ where } u^T = [u_1^T, \ldots, u_n^T].$$

This form, $e^n$, is reflected about $I$ and scaled by its "shadow" along $u$. From these forms, we see that all junctions of $N$ physical waveguides require only $O(N^2)$ computations and thus do not span all lossless scattering matrices without further generalization. What are all lossless scattering matrices?

From basic physical principles, a scattering matrix is lossless if and only if the total active complex power is scattering-invariant, i.e.,

$$e^n^T \Gamma e^n = e^n^T \Gamma e^n$$

$$= A^* A = I$$

where $\Gamma$ is a Hermitian, positive-definite matrix which can be interpreted as a generalized junction admittance. For unitary $A$, we have $\Gamma = I$. In the case of $N$ traveling pressure waves scattering at a "parallel" junction, we obtain $\Gamma = \text{diag}[\Gamma_1, \ldots, \Gamma_n]$. Unless all branch admittances are identical, the scattering matrix is never unitary. In general, the Cholesky factorization $\Gamma = U^T U$ gives an upper-triangular matrix $U$ which converts $A$ to a unitary matrix via similarity transformation $A^* A \Gamma = U^T \Gamma U \Rightarrow A^* A = I$, where $(\Gamma = U A U^{-1})$. Hence, the eigenvalues of every lossless scattering matrix lie on the unit circle. When $U$ is diagonal, a physical waveguide interpretation always exists with $U = \text{diag}[\Gamma]$. A generalized waveguide interpretation exists for all $U$ via a "power equivalent junction" [Smith, 1987] in which $U$ acts as an ideal transformer (in the classical network theory sense) on the time vector of all $\Gamma$ waveguide variables.

It readily follows from unitarity of $A$ that $A$ admits $N$ linearly independent eigenvectors. Consequently, nonzero $|\lambda| = 1$ for each eigenvalue $\lambda$, where $\lambda$ is a $\Gamma$-dependent eigenvector of $A$. Then the matrix $A^*$ diagonalizes $A$ in the sense $A^* \Gamma A = \text{diag}[\lambda_1^*, \ldots, \lambda_n^*] = D$, where $D = \text{diag}[\lambda_1^*, \ldots, \lambda_n^*]$. Hence, we obtain $A^* \Gamma A = D \Rightarrow A = \text{diag}[\lambda_1, \ldots, \lambda_n]$. Thus, Eq. (1) is satisfied for $\Gamma = T \Gamma^{-1}$ which is Hermitian and positive definite. We may summarize as follows:

**Theorem:** A scattering matrix (FDN feedback matrix) is lossless if and only if its eigenvalues lie on the unit circle and its eigenvectors are linearly independent.

Thus, lossless scattering matrices may be fully parameterized as $A = T \Gamma^{-1} D T^*$, where $D$ is any unit-norm diagonal matrix, and $T$ is any invertible matrix.

It can be quickly verified that all scattering matrices arising from the intersection of $N$ physical waveguides possess one-eigenvalue equal to 1 (corresponding to all incoming waves being equal) and $N - 1$ eigenvalues equal to -1 (corresponding to equal incoming waves on $N - 1$ branches, and a large opposite wave on the remaining branch which pulls the junction pressure to zero).

Since only a subset of all $N \times N$ unitary matrices is given by a physical junction of $N$ digital waveguides, (e.g., consider permutation matrices), the FDN point of view yields lossless systems outside the scope of single-junction Multipath networks. On the other hand, since only normalized waveguide junctions exhibit unitary scattering matrices, the DWN approach gives rise to new classes of lossless FDN's. Moreover, by considering more than one scattering junction, the DWN approach suggests a far larger class of lossless network topologies.

**References**

