Preserving the LTI system topology in s- to z-plane transforms

Vadim Zavalishin*

May 5, 2008

Abstract

A method for preserving the LTI system topology during the application of an s- to z-plane transform is described. The produced discrete-time systems keep the important characteristics of the original continuous-time systems in cases of modulated parameters. A further extension to the case of contained nonlinearities is made.

1 Introduction

In music DSP applications, two approaches for conversions of continuous-time (s-plane) LTI systems into the discrete-time (z-plane) case are widely used.

The first approach is a general DSP one. It assumes the application of an s- to z-plane transform substitution, usually the bilinear one, to the system's transfer function $H(s)$. The resulting discrete-time transfer function $H(z)$ is subsequently decomposed into biquad sections, with optional 1st-order sections, which are then implemented in one of the classic discrete-time forms such as direct or canonic one, including transposed versions of those. The transfer function $H(z)$ is obviously in the desired relation to $H(s)$, which in case of the bilinear transform can be considered 'reasonably perfect'. However, the specific set of state variables and the topology of the original continuous-time system are completely ignored in this approach. As a result, the obtained discrete-time systems usually exhibit strong undesired artifacts if the system parameters are modulated, especially at a high rate, which is not unlikely in the music DSP area. Besides, the completely different system topology makes it impossible to model the nonlinearities contained in the original system.

The second approach is rather music DSP specific. It starts with decomposing the s-plane system into lower-order elements such as integrators or one-pole filters, which are then converted to the z-plane using the transform approach. For those lower-order systems, one can often construct topologies, which feature nice modulation behaviors. These lower-order discrete-time blocks are then connected to each other in exactly the same way as their continuous-time counterparts. Therefore, the state variables and the system topology are preserved to a good extent, which usually ensures good modulated-case behavior and also makes it possible to clone nonlinear elements present in the original system. Unfortunately, the resulting systems often turn out to have delayless feedback loops, which means they are not directly implementable. A trivial approach is commonly in use here, to simply put additional z^{-1} delays into the offending feedback paths. This often results in acceptable transfer functions, except for the high frequency areas and/or certain values of system parameters, where the transfer function gets distorted beyond reasonable, often leading to unstable systems. The usual ways to counteract these effects are to clip the parameters to safe ranges, to design specific methods for the parameter prewarping ([1]), and/or to simply increase the sampling rate.

Apparently, a third possiblity exists, which combines the benefits of the first two. Surprisingly, it is much less known. Particularly, the important works on music DSP filter design, such as [1] and [2], make no mention of it. It seems, as if only brief hints to this

*Native Instruments GmbH

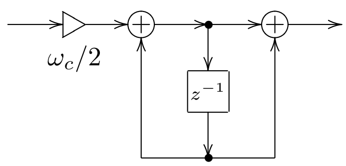

Figure 1: Bilinear integrator.

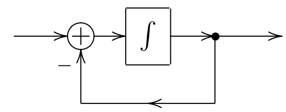

Figure 2: Equivalent state-variable representation of a 1-pole lowpass RC-filter. The cutoff of the integrator is set to $\omega_c = 1/RC$.

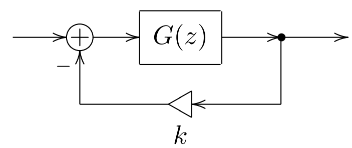

Figure 3: A generic discrete-time system with feedback.

approach, such as [3] exist.[1,2] This article is supposed to fill the gap.

2 The basics

We begin by mentioning the fact, that any differential LTI system can be 'implemented' in a structure diagram consisting only of adders, gain elements and integrators, with a possibility to extend the spectrum of such systems by allowing delay elements. Of these, adders, gain elements, and delays are naturally present in discrete-time systems as well. The integrators need to be converted by means of some s- to z-plane transform, the bilinear one being probably the best choice. A good possible topology for a bilinear-transformed integrator $H(s) = \omega_c/s$, where ω_c is the cutoff, is shown in Fig. 1 (in some cases, the opposite order of the IIR and FIR parts could be used).

Thus, all basic elements are available for the discrete-time case as well. So, for a given continuous-time structure we simply need to replace the integrators by their digital models. Notice, that the transform used to convert the integrators thereby becomes the transform applied to the entire system.

Let's take a basic 1-pole lowpass RC-filter as an example. Such filter can be completely equivalently represented in the state-variable form (Fig. 2). Replacing the integrator with its digital model, such as the one in Fig. 1, we obtain a bilinear-transformed digital implementation of the original filter. Notice that the transfer function $H(z)$ of the resulting filter is exactly the result of bilinear-transform substitution into $H(s)$. However, since the integrator in Fig. 1 contains a delayless path from input to the output, our digital filter is not directly implementable. This is pretty much the intermediate result we would have obtained from the second approach mentioned in the introduction.

3 The method

Now, instead of inserting a z^{-1} delay into the feedback, we are going to claim that the digital filter corresponding to Fig. 2 *is* implementable. We can even extend it to a somewhat more general case, pictured in Fig. 3.

Let's consider a fixed discrete time moment n, at which we consider the output signal value $y = y[n]$ as a function of the input signal value $x = x[n]$. For any difference LTI system, and for $G(z)$ in Fig. 3 in particular, this function will have the linear form:

$$y = f(x) = gx + s$$

where x is the system input, y is the system output, the offset s is defined by the state of the system *and* by the system parameters, and the instant gain g is

[1] While [3] is the first known to the author reference to the discussed approach, the author would like to mention that he has come up with a similar idea independently.

[2] Due to the limited author's acquaintance with the respective literature, it might as well be that the approach *is* already documented elsewhere. Anyway, the author hopes that this article will not be useless.

defined by the system parameters. Both s and g are fixed at a given n. Typically, $g > 0$. Often, $g \leq 1$.

Therefore, for the entire system in Fig. 3 one could write:

$$y = f(x - ky) = g(x - ky) + s \qquad (1)$$

where x and y are this time the input and output of the entire system. Instead of rushing to solve the above equation, we first examine its meaning.

In the analog world, the signals cannot abruptly change their value. Each element of the circuit, including wires, would have a (possibly negligibly small, but still nonzero) complex impedance, all such impedances working together to smooth the sudden changes.

It is suggested to treat the digital structures with a similar approach. E.g. in the case of (1) we would not assume that y instantly takes the value equal to the solution of the equation. Initially y would retain its value from the previous sample $y_0 = y[n-1]$. Then $f(x - ky_0)$ would be computed according to (1) producing a new *desired* value for y. Now the value of y will start changing in a *continuous* way towards this new value. Simultaneously $f(x - ky)$ will be changing further, correcting the 'destination' value for y until both of them coincide, in which case the system has reached the state corresponding to the solution of (1), all of that happening within the duration of the current sample.

Solving (1) according to the above spirit we get

$$y = \begin{cases} \dfrac{gx + s}{1 + gk} & \text{if } 1 + gk > 0 \\ +\infty \cdot \operatorname{sgn}[(gx + s) \\ \qquad\qquad - (1 + gk)y_0] & \text{otherwise} \end{cases} \qquad (2)$$

Notice that for a typical $G(z)$, the system would be unstable for $k \leq -1$ anyway, although if $1 + gk > 0$ such instability would take more than one sample to reach the infinity.

Thus we have just described a way to treat the digital system in Fig. 3 as *implementable*. If the feedback path doesn't contain any nonlinearities, then, by incorporating the explicit computation of (2) into the system structure, the latter can be converted into an equivalent structure without delayless feedback. Often however, one might simply use (2) during the implementation of the system in program code.

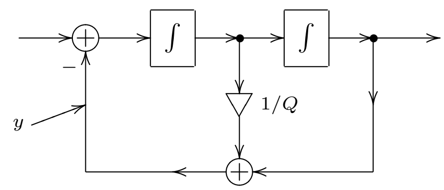

Figure 4: 2-pole state-variable lowpass filter.

An extension of the method to the cases of multiple feedback paths is simple, if all paths share at least one common point. E.g. for a 2-pole state-variable filter (Fig. 4) we simply can (temporarily) choose the feedback signal as y, again resulting in an equation of the form (1) (with $k = 1$).

The case of a system, where some interdependent delayless feedback loops do not have a common point, is somewhat more complicated. Still, it is always possible to write a linear equation or a system of linear equations analogous to (1). The extension of the approach to this case is clear.

Also, often it is possible to separate the 'smaller' loops into LTI *subsystems*, which can be converted to equivalent structures without delayless feedback as mentioned earlier. E.g. the ladder filter structure can be decomposed into a number of serially connected 1-pole filters, where each one of those would be converted independently first, thereafter the entire system having only a single delayless feedback loop.

4 Nonlinearities

The described method can be extended to the case of systems containing nonlinear elements. We use the ladder filter emulation ([1], [2]) as an example. To reduce the problem complexity, including the computational one, we choose to use a single saturation stage at the feedback mix-in point (Fig. 5). This should be the best point, if only a single saturator is to be used. Notice that the structure in Fig. 5 has a generic character, not restricted to the ladder filter.

Let the saturator in Fig. 5 be a memoryless wave-

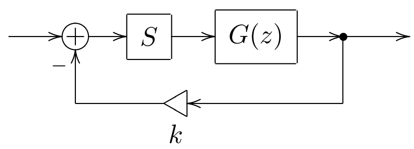

Figure 5: Saturation in feedback ('S' is a memoryless saturator).

shaper defined by the function

$$x_{\text{out}} = S(x_{\text{in}}) \tag{3}$$

In this case (1) becomes

$$y = gS(x - ky) + s \tag{4}$$

If $S'(x) \geq 0\ \forall x$ and if $k \geq 0$ and $g \geq 0$ then (4) always has a unique solution, this solution also being in agreement with the 'nonzero-impedance' paradigm. Therefore, we can simply solve (4).

If (3) is a second-order polynomial equation relative to x_{in} and x_{out}, then (4) can be easily solved analytically. The same obviously holds, if (3) defines a curve which can be broken into multiple second-order segments (although a search for the applicable segment will be necessary). This is particularly the case for parabolic:

$$S(x) = \begin{cases} \dfrac{x}{4} \cdot (4 - |x|) & \text{if } |x| < 2 \\ \operatorname{sgn} x & \text{otherwise} \end{cases}$$

and hyperbolic:

$$S(x) = \frac{x}{1 + |x|}$$

saturation curves.

A very nice curve to use for such saturator is a hyperbolic tangent. It particularly occurs in differential amplifiers, such as ones used in the transistor ladder filter ([2]). In this case (4) takes the form

$$y = g \tanh(x - ky) + s \tag{5}$$

and is obviously not solvable analytically. Therefore one needs to resort to numeric methods.

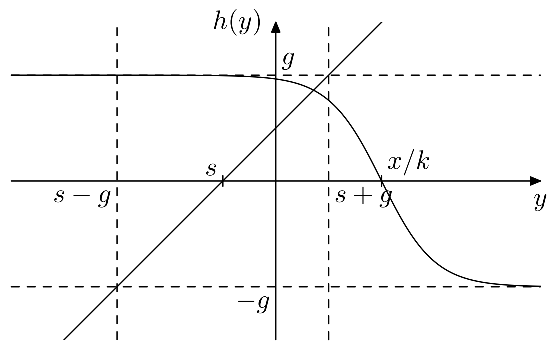

Figure 6: Graphical illustration of (5).

One possibility would be to break $\tanh x$ into second-order approximation segments. Another approach is to build an approximation of a function $F(u, v)$ defined as:

$$F(u, v) = \tanh(u - vF(u, v)) \tag{6}$$

where $u = x - ks$, $v = gk$. Otherwise the Newton-Raphson method looks like a good candidate, except that some convergence problems may occur for poorly chosen initial values. To come up with a reasonably good initial value choice we consider (5) as an intersection of a hyperbolic tangent curve $g \tanh(x - ky)$ with a straight line $y - s$ (Fig. 6).

Looking at the picture, we come up with the following rule. If $s - g < x/k < s + g$, then the initial point is $y = x/k$. Otherwise one of the points $s - g$ and $s + g$ (the one lying between s and x/k) is chosen. This choice ensures the *monotonous*[3] convergence of the algorithm. For $0 < g \leq 1$ and $0 \leq k \leq 8$, which are reasonable ranges (particularly for the ladder filter), it was found that about 5 iterations are generally necessary for the convergence to 10^{-5} precision. Yet another iteration will generally double the number of digits.

A number of tricks can be further employed to reduce the computation times. E.g. one could use an approximate computation with reduced precision at earlier iterations. A second-order segmented approximation of $\tanh x$ or an approximation of (6) might be

[3] Because the initial point is lying between $y = x/k$ and the true solution.

used improve the initial value choice. Once could also first check whether the previous output value $y[n-1]$ doesn't provide a better initial guess, although the convergence speed may significantly vary in this case.

Returning to the generic nonlinearity $S(x)$, we'd like to mention a cheap and dirty but computationally efficient approach, which is to simply solve (1) against $x - ky$, allowing the possibility of infinite results, and then compute $S(x - ky)$ just once.

The discussed nonlinear-case method can be also generalized to a case of multiple contained nonlinearities, However, the solution of the equation similar to (4) will become more computationally intensive, and in certain cases more challenging.

If $k < 0$ and/or if the saturation curve is having a complex shape, such that (4) can have multiple solutions, the 'nonzero-impedance' approach can be used to pick up the right one.

5 Conclusion

The described method allows the conversion of continuous-time LTI system prototypes to discrete-time implementations with the transfer function mapped according to the desired transform formula, while the system topology and state variables are preserved. Thus one can obtain implementations which have high quality amplitude and phase responses and simultaneously have nice behavior in the modulated-parameter case and (if desired) certain nonlinear elements of the original prototype. Obviously the same method also can be applied to the direct construction of discrete-time systems.

The practical cases of application of the method include filters, phasers, flangers, etc. Since even bilinear-transformed systems feature significant high-frequency warping in their responses compared to the original prototypes, sampling rates higher than 44kHz might still be desired.

The nonlinear case, often requiring iterative numeric methods, is significantly more computationally intensive. Taking the aliasing produced by the nonlinearities into account, an oversampling at a higher rate instead of solving (4) might often provide a reasonable compromise at better performance.

Acknowledgements

The author would like to thank NI, particularly Daniel Haver, Tom Kurth, Nicolas Gross, and Andreas Gloggengiesser for their support and for helping him to release this article. Very special thanks to Stephan Schmitt.

References

[1] Stilson T., Smith J.O. III "Analysing the Moog VCF with Considerations for Digital Implementation."
http://ccrma.stanford.edu/~stilti/papers

[2] Huovilainen A. "Non-linear Digital Implmentation of the Moog Ladder Filter." *Proc. of the 7th Int. Conference on Digital Audio Effects (DAFx'04), Naples, Italy, October 5-8, 2004*
http://dafx04.na.infn.it

[3] Landeholm E. "Zero delay feedback using sample prediction." *Posted Mar 8, 2002 to the* music-dsp *mailing list*
http://music.columbia.edu/pipermail/music-dsp