MAGNITUDE-COMPLEMENTARY FILTERS FOR DYNAMIC EQUALIZATION

Federico Fontana

University of Verona
Dipartimento di Informatica
strada Le Grazie, 14
37134 Verona - Italy
fontana@sci.univr.it

Matti Karjalainen

Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
P. O. Box 3000
FIN-02015 HUT, Espoo, Finland
Matti.Karjalainen@hut.fi

ABSTRACT

Discrete-time structures of first-order and second-order equalization filters are proposed. They turn to be particularly useful in applications where the equalization parameters are dynamically varied, such as in contexts of audio virtual reality. In fact, their design allows a simplified and more direct control of the filter coefficients, at the cost of some more computation cycles that are required, during each time step, by implementations on real-time processing devices.

1. INTRODUCTION

Recent developments in multimedia audio and in auditory display have created a special interest for models, whose parameters can be easily and directly driven by higher-level control stages. Although these models may result in sub-optimal algorithms and structures in terms of computational cost, easy access to their parameters simplifies the design of the intermediate control stages, and sometimes allows realizing more effective user interfaces. If such a design philosophy once was heavily limited by unavoidable hardware requirements, now current real time hardware can provide, without loss of performance, the necessary resources to models where the control layer has been optimized, despite a sub-optimal realization of the signal processing layer.

Physical models have been a typical example of this different approach in the design of sound synthesis algorithms for electronic musical instruments: in some cases, their adoption has made human-machine interaction possible in this application field [1, 2]. More recently, special attention has been deserved by the auditory display community for models capable of reproducing "sounding objects", and their interactions with the human being in the context of a multi-modal, interactive virtual environment: for this type of human-object interaction, the effectiveness of control parameters like impact force or object position is even more important than the accuracy of the sound produced by the object itself, that is generated by a synthesis algorithm working at the signal processing layer [3, 4].

Audio effects are facing a similar change in the design philosophy. Audio virtual reality requires that physical and geometrical quantities, such as the sound source position or wall reflections in a virtual listening room, drive the processing algorithm as directly as possible. This can be achieved if the control layer complexity, that is usually devoted to map these quantities into parameters such as filter coefficients, has been minimized. Such goal is critical if those quantities are time-varying, as it happens for instance in application of human-computer interaction, where the decisions taken by the user determine changes in the system parameters.

Linear equalization is a processing technique which is well-known by the effect designer. Graphic equalizers, both analog and digital, are widely used to change the "color" of sounds. Their use is very general, and their tuning so immediate to understand that they can be successfully used in interactive audio systems, whenever the human-system interaction involves changes of the sound color, or "presence" [5].

Tunable equalization filters [6], classified as first-order (shelving) equalization filters and second-order (parametric) equalization filters, are perhaps the most common building blocks that are used in the design of digital graphic equalizers. They provide an easy and direct access to the equalization parameters, i.e.,

1. low-frequency gain and cutoff frequency for first-order low-frequency (LF) equalization filters, or *LF shelving filters*; symmetrically, high-frequency gain and cutoff frequency for first-order high-frequency (HF) equalization filters, or *HF shelving filters*;
2. center-frequency gain, selectivity and center frequency for *second-order equalization filters*; these second-order filters are in particular *peaking* filters when the center-frequency gain is positive, or *notch* filters when the center-frequency gain is negative.

Such low-, center- and high-frequency gains are tuned driving one parameter (denoted with K) of these filters, so that the gain in dB is simply found out by computing the value $20 \log_{10} K$. As a result of this ease of control, a graphic equalizer made by a series connection of one LF shelving filter (denoted with L), N second-order equalization filters, and one HF shelving filter (denoted with H), gives an immediate visual feedback to the user: provided a proper cutoff frequency and selectivity for each equalization filter, its overall equalization curve is described by a gain response having magnitude K_L at LF, magnitude K_H at HF, and magnitudes $K_1, \ldots, K_N$ that correspond to the N selective gains at the prescribed center frequencies, each one provided by a second-order equalization filter.

Figure 1 shows, as an example, the equalization curve of a graphic equalizer ($N = 2$) having $K_L = -2$ dB, $K_1 = 1$ dB centered at -40 dB, $K_2 = -2$ dB centered at -20 dB, $K_H = 1$ dB. The Nyquist frequency has been normalized to unity. The shelving filters provide an overall shift of the gain level, in a way that the action of the second-order equalization filters are "shelved" by this shift.

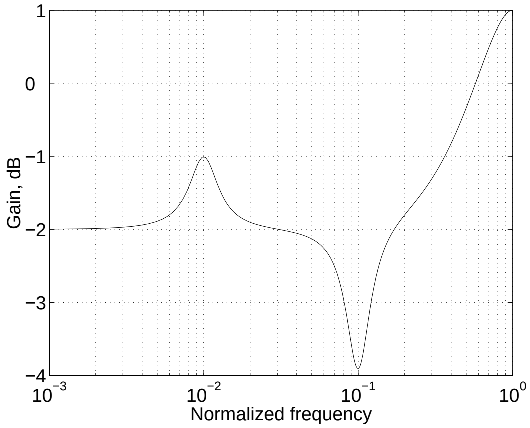

Figure 1: *Equalization curve of a graphic equalizer (N=2) having* $K_L = -2$ *dB,* $K_1 = 1$ *dB @* -40 *dB,* $K_2 = -2$ *dB @* -20 *dB,* $K_H = 1$ *dB. Nyquist frequency normalized to unity.*

2. MAGNITUDE-COMPLEMENTARY EQUALIZATION

It would seem quite obvious that reciprocating the K value results in a magnitude-complementary gain response, exactly canceling the former one. Unfortunately, this is not true: two tunable equalization filters having inverse K values do not have responses that are symmetrical around 0 dB. For example, Figure 2 shows gain responses coming from an LF shelving filter, providing respectively a cut of 5 dB ($K = 0.562$, dashed line) and a boost of 5 dB ($K = 1.778$, solid line). Evidently, they are not complementary. A similar example is given in the case of a second-order equalizer (Figure 3). Notice also that a bandpass response can be obtained only by setting K to be greater than one, in any case.

Complementary gain responses can be in principle obtained, due to the minimum-phase property of the transfer function realized by this class of filters, if another parameter, denoted with α, accounting for the filter cutoff frequency (first-order case) or the selectivity (second-order case), is tuned according to the value of K [6]. Tuning α together with K allows in practice to select a new value for the cutoff frequency or the selectivity each time the gain varies. Hence, complementary gain responses are provided if the user-level control operates in general on both K and α. This means that a system, having the functionality of synthesizing magnitude-complementary equalization functions, must be provided with an intermediate stage, which maps each selective gain control from the user interface into two parameters of the tunable equalization filter.

Parameters K and α must be mapped, in their turn, into filter coefficients. The number of the mapped coefficients is at least two, in realizations of the tunable equalization filter where α maps, directly and independently from K, into one filter coefficient. This happens when a lattice structure realizes the allpass block of the filter [5]. In general cases this number is higher, up to four, for example in realizations involving biquadratic filters that minimize the computational cost of the processing algorithm, meanwhile sacrificing the ease of control [7].

In the next Section we present a different approach to the inverse tunable equalization filter design. This approach minimizes the cost of the coefficients control, and the memory that is required for storing the filter coefficient values during fast lookup operations. As a rule of thumb, this solution increases the number of computation cycles that are required for computing a signal sample, although this increase is far from being dramatic.

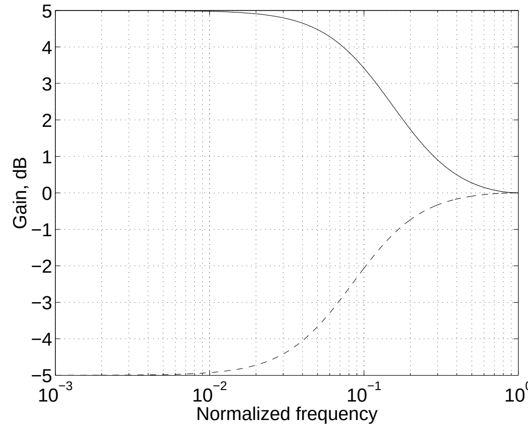

Figure 2: *Gain responses of an LF shelving filter. Dashed line:* 5 *dB cut (*$K = 0.562$*); solid line:* 5 *dB boost (*$K = 1.778$*). Nyquist frequency normalized to unity.*

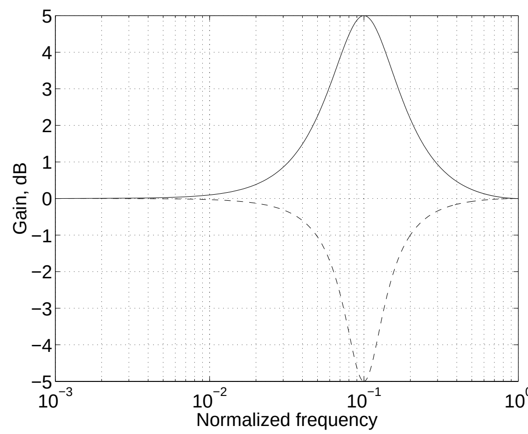

Figure 3: *Gain responses of a second-order equalizer. Dashed line:* 5 *dB cut (*$K = 0.562$*); solid line:* 5 *dB boost (*$K = 1.778$*). Nyquist frequency normalized to unity.*

3. DESIGN OF THE INVERSE FILTER

Consider the transfer function $H(z)$ of a bandstop tunable equalization filter, H_{BS}. It provides a gain response such as the one in dashed line of Figure 2 or Figure 3. $H_{\text{BS}}(z)$ can be put in the following form [6]:

$$H_{\text{BS}}(z) = \frac{1+K}{2}\left\{1 + \frac{1-K}{1+K}\,A(z)\right\}\ ,\ 0 < K < 1 \qquad (1)$$

where $A(z)$ is an allpass filter. When $A(z)$ is a first-order allpass,

$$A(z) = \frac{\alpha - z^{-1}}{1 - \alpha z^{-1}} \qquad (2)$$

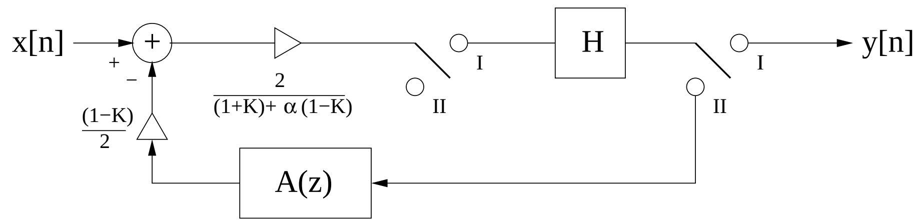

Figure 4: *Structure for the computation of the bandpass tunable equalization filter. The terms* $1-K$ *must be changed into* $K-1$ *when realizing an HF shelving filter.*

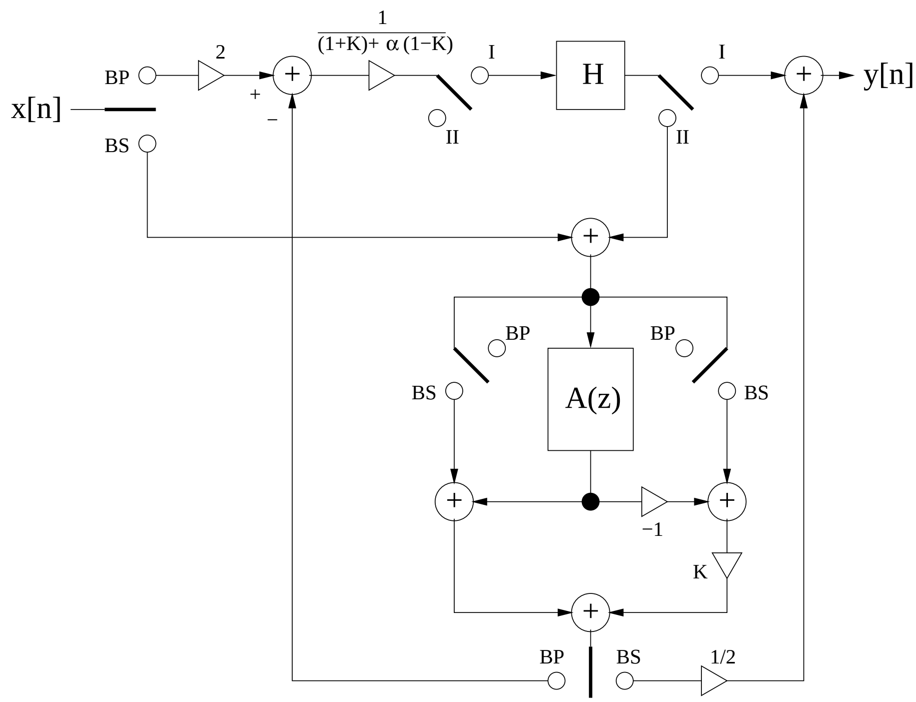

Figure 5: *Bandpass/bandstop equalizer. When switches are set to position BS a bandstop transfer function is provided. When switches are set to position BP a complementary (bandpass) transfer function is provided. HF shelving is realized by swapping the branch containing the multiplier by* K *with the one terminating at the adder common to both of them.*

we have an LF shelving filter. When $A(z)$ is a second-order allpass,

$$A(z) = \frac{\alpha - \beta(1+\alpha)z^{-1} + z^{-2}}{1 - \beta(1+\alpha)z^{-1} + \alpha z^{-2}} \tag{3}$$

then we have a second-order equalizer. An LF shelving filter is turned into an HF shelving filter by changing the sign of $1-K$ in (1).

Let $H_{\mathrm{BP}}(z)$ be the inverse of $H_{\mathrm{BS}}(z)$, this being possible due to the minimum phase property of the bandstop transfer function:

$$\begin{aligned} H_{\mathrm{BP}}(z) &= \frac{1}{H_{\mathrm{BS}}(z)} \\ &= \frac{2}{1+K}\,\frac{1}{1+\frac{1-K}{1+K}A(z)}\,,\ 0<K<1 \end{aligned} \tag{4}$$

We see that K is still factored out from the allpass block in the denominator of the bandpass transfer function. This suggests that the independence of gain tuning might be preserved in the inverse filter, too. Unfortunately, this filter does not straightforwardly result in a structure preserving the nature of the rational function $H_{\mathrm{BP}}(z)$, since it contains a delay-free loop, which is known to be non-computable.

Recent research [8] has shown that, for rational functions containing delay-free loops like the one seen above, techniques exist to solve the non-computability, while preserving the structural properties of the function. In this case, using the techniques just mentioned, the inverse (bandpass) filter preserves the allpass block (where the α coefficient is embedded) that is contained in the original tunable equalization filter.

The algorithm that computes the bandpass filter consists of the following three steps that are repeated at each time step n:

1. feed the allpass block with zero, computing the value $l_0[n]$;

2. calculate the output $y[n]$ of the bandpass tunable equaliza-

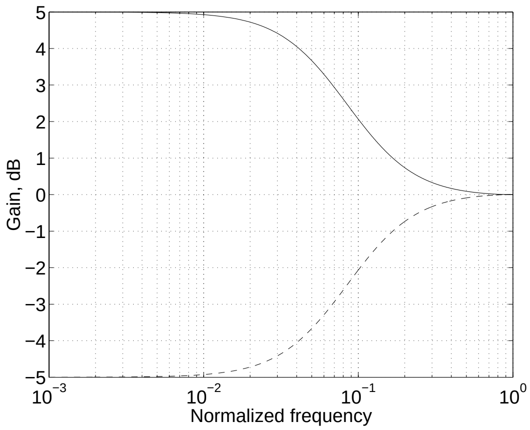

Figure 6: *Gain responses of the proposed system in LF shelving configuration ($K = 0.562$). Dashed line: 5 dB cut; solid line: 5 dB boost. Nyquist frequency normalized to unity.*

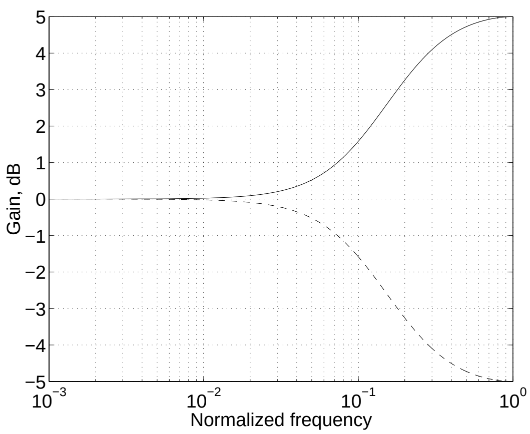

Figure 7: *Gain responses of the proposed system in HF shelving configuration ($K = 0.562$). Dashed line: 5 dB cut; solid line: 5 dB boost. Nyquist frequency normalized to unity.*

tion filter in the following way, x being the input signal:

$$y[n] = \frac{1}{1+\alpha\frac{1-K}{1+K}} \left\{ \frac{2}{1+K}\, x[n] - \frac{1-K}{1+K}\, l_0[n] \right\}$$
$$= \frac{2}{(1+K)+\alpha(1-K)} \left\{ x[n] - \frac{1-K}{2}\, l_0[n] \right\}$$

3. feed the allpass filter with $y[n]$ to update its state variables.

The algorithm that we have just outlined can be computed by the structure depicted in Figure 4. That structure contains in particular a hold block (labeled with H) devoted to retain the output sample $y[n]$, that must be fed back to the allpass block after the computation of the output itself. In more detail, such a structure performs the following procedure:

- the two switches are at the position labeled with I; steps 1 and 2 are computed. $y[n]$ is retained by the hold block H;
- the switches move to position labeled with II: H feeds the allpass block with $y[n]$ to update its internal state;

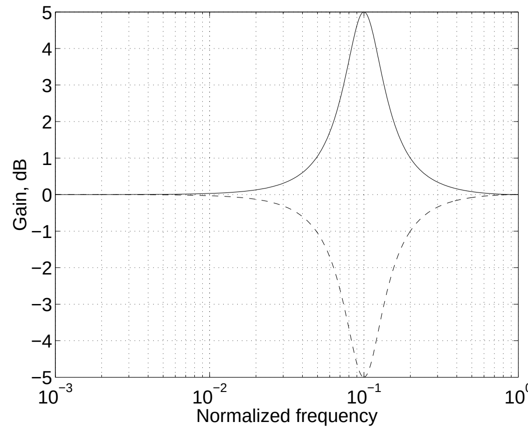

Figure 8: *Gain responses of the proposed system configured as a second-order equalizer ($K = 0.562$). Dashed line: 5 dB cut; solid line: 5 dB boost. Nyquist frequency normalized to unity.*

- the switches move back to position I. The system is ready to process the next input sample.

The same procedure can be computed also by a pure linear filter structure, both in the case of the shelving filters and the second-order tunable equalization filters. In this case, a reformulation of the algorithm leads to an allpass filter network that is free of non-computable loops. This technique has been successfully used in the solution of non-computable loops arising in Warped IIR (*WIIR*) filter structures [9].

The similarity of the inverse filter structure with the original tunable equalization filter is quite evident: the blocks that are present in both systems are the same, except for differences existing in the multipliers. The bandstop and the bandpass (inverse) filters can be embedded together in a single structure, containing some switches that are alternatively set according to the (bandstop or bandpass) configuration of the system.

Such a structure is shown in Figure 5, accounting for all the cases we treated. When the switches are in the position labeled with BS, the system implements a bandstop tunable equalization filter. When they are in the position labeled with BP, the system implements a complementary (bandpass) tunable equalization filter. Shelving filtering is provided using first-order allpass blocks, whereas second-order equalization filters come up using second-order allpass blocks as defined by (3).

Finally, note that switching between BS and BP in principle does not introduce transients in the output signal and the internal state. In fact, the configuration changes when $K = 1$: under this condition the switches positions are insignificant.

4. SIMULATIONS

Figure 6 shows, in dashed line, the gain response of the system in Figure 5 configured as an LF shelving filter, providing a cut of 5 dB ($K = 0.562$); clearly, this response equals the one given in Figure 2. When the system switches to position BP, a complementary boost of 5 dB is provided: the new gain response is depicted in solid line in the same figure. Figure 7 and Figure 8 show identical situations in the case of HF shelving and for the second-order equalizer.

Simulations confirm that the system provides complementary transfer functions.

It can be seen from Figure 5 that the overall number of computations in the bandpass configuration involves, for each time step, two sums and two multiplies more than the bandstop configuration. This means that our realization of the inverse tunable equalization filter is less efficient than the tunable equalization filter itself. Conversely, the control of the system is easy and efficient, since the synthesis of complementary transfer functions does not need access the allpass block. Moreover, it can be figured out that the proposed system synthesises bandpass transfer functions without making use of coefficients whose magnitude is greater than one.

One possible implementation (in Matlab™ code) of the routine that realizes, for example, an inverse LF shelving filter is reported in the following lines. Here, `N` is the number of time samples to be processed, and it is `a` $= \alpha$, `K` $= K$, `input` is the input signal vector, `out_sh` is the output signal vector, `S` is the state of the allpass filter (2). The first line inside the iteration computes steps 1 and 2 of the algorithm outlined in Section 3, and the second line computes step 3:

```
for i = 1:N

    out_sh(i) =
        2/((1+K)+a*(1-K))*input(i) -
        (1-K)/((1+K)+a*(1-K))*filter([a,-1],[1,-a], 0, S) ;

    [none, S] = filter([a, -1],[1, -a], out_sh(i), S) ;

end
```

5. SUMMARY

We have proposed a versatile structure that realizes shelving and second-order tunable equalization filters. Although slightly less efficient computationally than the original tunable equalization filter [6], it provides magnitude-complementary transfer functions. Moreover, its control layer allows an easy and direct access to the equalization parameters.

6. ACKNOWLEDGMENTS

This work has been partially funded by the "Sound Source Modeling" Project of the Academy of Finland.

7. REFERENCES

[1] G. Borin, G. De Poli, and A. Sarti, "Algorithms and structures for synthesis using physical models," *Computer Music Journal*, vol. 16, no. 4, pp. 30–42, 1992.

[2] J. O. Smith, "Physical modeling synthesis update," *Computer Music Journal*, vol. 20, no. 2, pp. 44–56, 1996.

[3] P. Djoharian, "Shape and material design in physical modeling sound synthesis," in *Proc. Int. Computer Music Conf.*, Berlin, Germany, Aug. 2000, pp. 38–45.

[4] F. Avanzini and D. Rocchesso, "Controlling material properties in physical models of sounding objects," in *Proc. Int. Computer Music Conf.*, La Habana, Cuba, Sept. 2001, In press.

[5] S. K. Mitra, *Digital Signal Processing. A computer-Based Approach*, McGraw-Hill, New York, 1998.

[6] P. A. Regalia and S. K. Mitra, "Tunable digital frequency response equalization filters," *IEEE Trans. on Speech and Audio Processing*, vol. ASSP-35, no. 1, pp. 118–120, Jan. 1987.

[7] R. Bristow-Johnson, "The Equivalence of Various Methods of Computing Biquad Coefficients for Audio Parametric Equalizers," *Audio Engineering Society Convention*, Nov. 1994, Preprint 3906.

[8] A. Härmä, "Implementation of frequency-warped recursive filters," *EURASIP Signal Processing*, vol. 80, no. 3, pp. 543–548, Mar. 2000.

[9] A. Härmä, M. Karjalainen, L. Savioja, V. Välimäki, U. K. Laine, and J. Huopaniemi, "Frequency-warped signal processing for audio applications," *Journal of the Audio Engineering Society*, vol. 48, no. 11, pp. 1011–1031, Nov. 2000.