SCALING OF MULTISTAGE INTERPOLATORS

Mattias Olsson, Per Löwenborg, and Håkan Johansson

Division of Electronics Systems
Department of Electrical Engineering
Linköping University
email:{matol, perl, hakanj}@isy.liu.se

ABSTRACT

In this paper we present a novel method for scaling of multistage interpolators. When a signal is upsampled it becomes Cyclo-Wide-Sense Stationary (CWSS) which prevents the use of common algorithms for scaling. Our method is based on multirate identities and polyphase decomposition and avoids these problems.

1. INTRODUCTION

When designing systems for digital signal processing we sometimes need to implement a device that changes the sampling frequency. A straightforward solution would be to first reconstruct the analog signal and then sample it again, however, this is usually far too costly and leads to low precision. Methods for performing this operation in the digital domain are, nowadays, well known. It can be shown that it is advantageous [1] from a complexity point of view, to do the interpolation in multiple stages. This, however, leads to a couple of design problems that have not yet been thoroughly investigated. In this paper we will focus on one of those problems, namely scaling of multistage interpolators.

Scaling is needed to avoid or reduce the impact of overflow. The idea behind scaling is to introduce scaling factors prior to multiplications and later scale the output with the inverse. Scaling in single-rate filters has been treated in many publications, see for example [2]. Different strategies for finding the scaling factors exist, for example safe scaling and scaling using the L_2-norm [2]. In the case of cascaded filters, each filter has to be scaled properly. Unfortunately, problems arise in interpolators because the stationarity is lost when upsamplers are used. The statistical properties of signals in multirate systems have been treated in [3], however the results are rather complicated and cumbersome to use in practice. In this paper we present a more practical method to scale the filters in multistage interpolators.

The outline of this paper is as follows. First, the concept of scaling in single-rate filters will be recapitulated followed by an introduction to interpolators. Then, we will present a novel method for scaling of multistage interpolators. Finally, we will give some concluding remarks.

2. SCALING

Interpolators contain filters and therefore we will recapitulate scaling in such structures.

The main operations performed in digital filters are multiplication and addition. The problem with these is that their result might overflow and hence cause distortion. If an arithmetic system like two's complement is used, it does not matter if the intermediate additions overflow as long as the overall calculation is within the allowed range. Multiplications do not have this property and therefore, the inputs to the multiplicators must be scaled.

To prevent overflow in a certain critical node $v(n)$ we can scale the input signal with a factor c and the output with $1/c$, see Fig. 1. The scaling must be done in such a way that the transfer function of the system is not altered, except possibly for a change in overall gain.

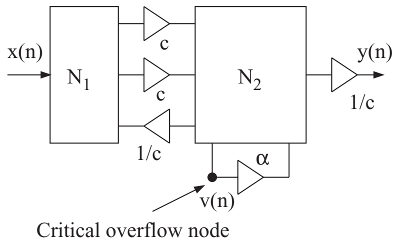

Figure 1: Scaling of the critical node $v(n)$ with a factor c.

To find the factor c, different strategies can be used. One is to simply forbid all overflows, which is called safe scaling. However, safe scaling is rather pessimistic and only suitable for FIR-filters with short impulse response length which have a high probability of overflow. Another strategy more suitable for longer filters with wideband input signals is based on the so called L_2-norm. The L_2-norm of a signal $x(n)$ with frequency function $X(e^{j\omega T})$ is defined as

$$||X||_2 = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi} |X(e^{j\omega T})|^2 d\omega T} = \sqrt{\sum_{n=-\infty}^{\infty} |x(n)|^2} \quad (1)$$

and is the root-mean-squared (rms) value of the signal.

L_2-norm scaling is done as follows. First, calculate $||F_i||_2$, which is the L_2-norm of the frequency response $F_i(e^{j\omega T})$ from the input to the critical node. The input to the system is then multiplied by $c = 1/||F_i||_2$ to reduce the risk of overflow in the critical node. If the output signal is fed back it would accordingly have to be multiplied by $1/c$ for the overall system to be unaffected. In the situation where the output is not fed back, we do not need the multiplication with $1/c$. Instead we can place a scaling constant at the final output to achieve the desired amplification.

The use of L_2-norm scaling for white-noise input signals ensures that the variance at the critical node equals that of the input. In particular, when the input is Gaussian with variance σ_x^2, the critical node is also Gaussian with the same variance. This implies that the probability of overflow at the critical node is the same as the probability of overflow at the input.

This is the reason why L_2-norm scaling is commonly used for scaling white-noise inputs and also general wide-band input signals, both random and deterministic ones.

One requirement for the statement above to hold is that the input is Wide-Sense Stationary (WSS). A random process $X(n)$ is said to be WSS if its mean-value is constant and its autocorrelation function $r_{xx}(n,k) = E[X(n)X^*(n-k)]$ only depends on k. An important property of a filter that is linear and time invariant (LTI) is that the output will still be WSS if the input is WSS [3]

In the next subsection we will show how scaling can be extended to cascaded FIR filters.

2.1 Scaling of Cascaded FIR Filters

A digital filter can usually be split into several cascaded filters by factorization of the transfer function. Such a situation is depicted in Fig. 2. As we stated before, scaling must be used to prevent overflow within the filters and therefore we have introduced a scaling constant c_i in front of each filter. We assume that the filters are FIR and realized in a direct-form structure, which limits the need for scaling constants to the output of each filter. The constant c_1 is used to scale the

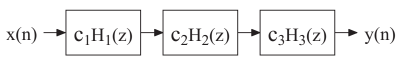

Figure 2: Scaling of cascaded direct-form FIR filters.

output of $H_1(z)$. Then, when c_2 is to be calculated, we need to take c_1 into account since when the signal reaches c_2 it has already been scaled by c_1. The constant c_3 is used to scale the output from $H_3(z)$ and must take c_1 and c_2 into account. Let

$$F_1 = H_1 \quad F_2 = H_1H_2 \quad F_3 = H_1H_2H_3 \tag{2}$$

and thus the expressions used to calculate c_i are equal to

$$c_1 = \frac{1}{||F_1||_2} \quad c_2 = \frac{1/c_1}{||F_2||_2} \quad c_3 = \frac{(1/c_1)(1/c_2)}{||F_3||_2} \tag{3}$$

In practice the scaling factors can be propagated into the filters and be combined with the multiplying constants within the filters. In Section 4 scaling will be extended to multistage interpolators.

3. INTERPOLATORS

To increase the sampling frequency of a signal, interpolation is used. Interpolation usually consists of two operations, upsampling and filtering. We will start this section by recapitulating the structure of an interpolator and, in particular, the polyphase representation that we will use later.

3.1 Upsampling and Filtering

The upsampling operation inserts zeros between the samples. If a signal is upsampled by a factor L it can be written as

$$y(m) = \begin{cases} x\left(\frac{m}{L}\right) & \text{if } m = 0, \pm L, \pm 2L, \ldots \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The upsampling operation scales the frequency axis and since the spectrum of a digital signal is periodic the spectrum will contain $L-1$ copies of the original spectrum. These

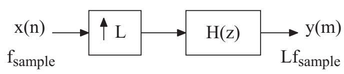

Figure 3: Interpolator consisting of an upsampler and a filter.

copies are unwanted and therefore a lowpass or interpolation filter is placed after the upsampler, see Fig. 3.

Unfortunately, stationarity will not be preserved when a signal is upsampled. It can be shown that a WSS signal becomes Cyclo-WSS (CWSS) if it is upsampled. A random process is said to be CWSS with a period of L if [3]

$$E[X(n)] = E[X(n+kL)], \ \forall n, \ \forall k$$
$$r_{XX}(n,k) = r_{XX}(n+L,k), \ \forall n, \ \forall k$$

The problem with signals that are CWSS, in this situation, is that the L_2-norm no longer can be used directly, because different samples have different statistical properties. For example $x(2n)$ may be more likely to overflow than $x(2n+1)$. One way to analyze cyclo-stationary signals is to use bispectrum masks [4] which are based on two-dimensional Fourier transforms. In this paper we use a different and simpler approach; we will use the identity in Fig. 4 and polyphase representation.

Figure 4: Identity for filters and upsamplers [5].

3.2 Polyphase Representation

In practice the interpolator is using a polyphase decomposed structure. We will now present some results that leads to the polyphase representation.

By using so called polyphase representation the transfer function $H(z)$ can be written as a sum of downsampled and delayed transfer functions,

$$H(z) = \sum_{i=0}^{L-1} z^{-i} H_i(z^L) \tag{5}$$

For an FIR filter it is always easy to rewrite $H(z)$ in the polyphase form of (5). Not all IIR filters can easily be rewritten in polyphase form, however, in this paper we will only treat FIR filters.

If (5) is used together with the multirate identity in Fig. 4 the interpolator can be realized as in Fig. 5. The advantage with this structure is that the sampling frequency in each branch is lower than in the original structure.

The polyphase structure will be used in Section 4 to transform each stage in the multistage interpolator into structures that can more easily be used for scaling.

3.3 Multistage Interpolators

The complexity needed to perform interpolation can often be reduced by doing the interpolation in multiple steps, so called multistage interpolation [5]. In Fig. 6 a multistage interpolator is shown. The sampling frequency is increased stepwise.

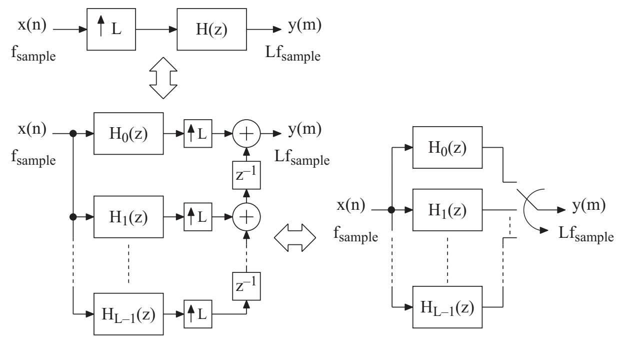

Figure 5: Polyphase interpolator.

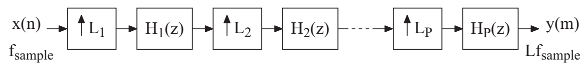

Figure 6: Multistage interpolator.

4. PROPOSED METHOD FOR SCALING OF MULTISTAGE INTERPOLATORS

This section introduces the proposed method for scaling of multistage interpolators. As discussed in Section 3, the dilemma is that the output of the upsampler is not WSS but CWSS. This means that different output samples have different statistical properties. In the single-stage interpolator case, it means in particular that the variance at the output of the interpolator is time-varying and periodic with the period L. To scale the output of the interpolator, we therefore divide the output into L subsequences which are WSS and hence can be scaled using the principles explained in Section 2 for single-rate filters. In the multistage interpolator case, we use the same idea but applied to each subinterpolator i as seen from the input to the output of stage i. The WSS subsequences are the outputs of the corresponding polyphase components, which are found by making use of the identity in Fig. 4 and polyphase decomposition. Details of the proposed method follows below.

A single-stage interpolator can be scaled using polyphase representation as follows. In Fig. 5 we see that each branch in the polyphase structure consists of a single rate filter. By calculating the L_2-norm for each branch $H_i(z)$ we can find the scaling constants for the filters. All the filters must use the same scaling constant, otherwise the amplification at the output would become time-varying. Choose the largest value $\max\{||H_i||_2\}$ and let the scaling factor c_i to be equal to $1/\max\{||H_i||_2\}$. For some of the polyphase branches it will inevitably be overly pessimistic.[1]

Consider next the multistage case. We will illustrate the method through an example. A three-stage interpolator that increases the sampling frequency 24 times can be constructed as in Fig. 7. In practice the interpolator in Fig. 7 is usually

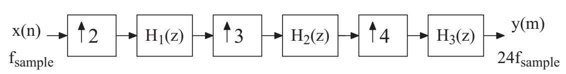

Figure 7: Multistage interpolator example.

[1]Another possible solution could be to calculate the mean of the L_2-norms and use its inverse as the scaling factor, but that will not be considered in this paper.

implemented as in Fig. 8, because of the parallel structure which lowers the overall computation burden. We will use a double index, $H_{i,j}$, to denote that the polyphase branch j orginates from filter i. Notice the similarities between Fig.

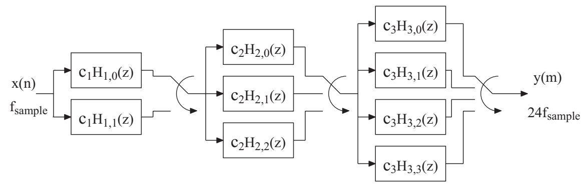

Figure 8: Multistage interpolator realized using polyphase representation.

2 and Fig. 8. The situation is the same, we need to find the constants c_i that prevent overflow at the output of each stage.

The method is general, but as an illustration we assume that we have filters with the following transfer functions

$$H_1(z) = \frac{1+2z^{-1}+z^{-2}}{4} \tag{6}$$

$$H_2(z) = \frac{1+z^{-1}+z^{-2}}{3} \tag{7}$$

$$H_3(z) = \frac{1+z^{-1}+z^{-2}+z^{-3}}{4} \tag{8}$$

Step 1: We now use polyphase representation to rewrite the first upsampler and filter $F_1(z) = H_1(z)$ as in Fig. 9. The

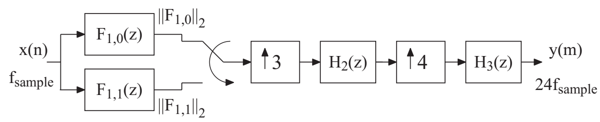

Figure 9: Interpolator scaling - step 1.

transfer functions for the polyphase branches can, using (5), be found to be

$$F_{1,0}(z) = \tfrac{1+z^{-1}}{4} \quad F_{1,1}(z) = \tfrac{1}{2} \tag{9}$$

We now calculate the L_2-norm for these two transfer functions.

$$||F_{1,0}||_2 = \tfrac{1}{2\sqrt{2}} \quad ||F_{1,1}||_2 = \tfrac{1}{2} \tag{10}$$

Now c_1 is calculated as

$$c_1 = \frac{1}{\max\{||F_{1,0}||_2, ||F_{1,1}||_2\}} = \frac{1}{1/2} = 2 \tag{11}$$

Step 2: Now use the identity in Fig. 4 to switch places between $H_1(z)$ and the upsampler by three in Fig. 7. The result is shown in Fig. 10.

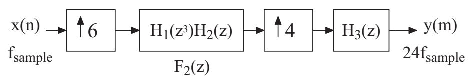

Figure 10: Interpolator scaling - step 2a.

Let

$$F_2(z) = H_1(z^3)H_2(z) \tag{12}$$

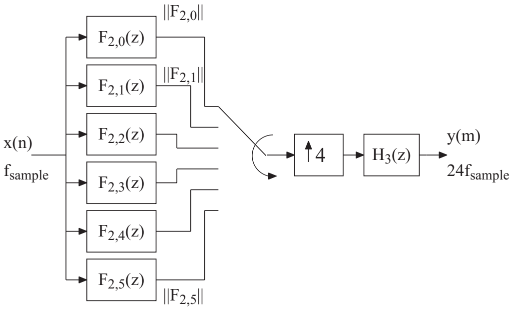

Figure 11: Polyphase representation - step 2b.

and divide the upsampler by six and $F_2(z)$ in Fig. 10 into six polyphase branches as in Fig. 11. The transfer functions for each branch becomes

$$F_{2,\{0\ldots2\}}(z) = \tfrac{1+z^{-1}}{12} \quad F_{2,\{3\ldots5\}}(z) = \tfrac{1}{6} \tag{13}$$

As before, we calculate the L_2-norm for each branch

$$||F_{2,\{0..2\}}||_2 = \tfrac{1}{6\sqrt{2}} \quad ||F_{2,\{3\ldots5\}}||_2 = \tfrac{1}{6} \tag{14}$$

and select the largest one. The constant c_2 then becomes equal to

$$c_2 = \frac{1/c_1}{\max\{||F_{2,\{0\ldots5\}}||\}} = \frac{1/2}{1/6} = 3. \tag{15}$$

Step 3: Use the multirate identity to switch places between $H_1(z^3)H_2(z)$ and the upsampler by four. The result is Fig. 12.

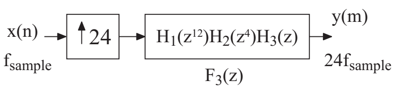

Figure 12: Interpolator scaling - step 3a.

Let

$$F_3(z) = H_1(z^{12})H_2(z^4)H_3(z) \tag{16}$$

and divide $F_3(z)$ and the upsampler by 24 into 24 polyphase branches as in Fig. 13. The transfer functions can be found

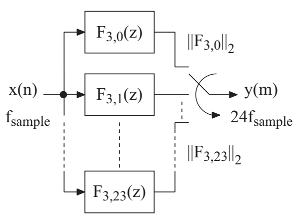

Figure 13: Polyphase representation - step 3b.

to be

$$F_{3,\{0\ldots11\}}(z) = \tfrac{1+z^{-1}}{24} \quad F_{3,\{12\ldots23\}}(z) = \tfrac{1}{12} \tag{17}$$

and the L_2-norms

$$||F_{3,\{0\ldots11\}}||_2 = \tfrac{1}{12\sqrt{2}} \quad ||F_{3,\{12\ldots23\}}||_2 = \tfrac{1}{12} \tag{18}$$

Finally, we calculate

$$c_3 = \frac{(1/c_1)(1/c_2)}{\max\{||F_{3,\{0\ldots23\}}||\}} = \frac{(1/2)(1/3)}{1/12} = 2 \tag{19}$$

Using the constants found we can now implement the filter as in Fig. 8. The method can be summarized in the following algorithm. We refer to Fig. 6 for a definition of the variables.

Algorithm *Scaling of multistage interpolators*

1. $F_1(z) \leftarrow H_1(z)$
2. $c_0 \leftarrow 1$
3. **for** $i \leftarrow 1$ **to** number of filters
4. Split $F_i(z)$ and the leftmost upsampler by L_i into L_i polyphase branches with transfer functions $F_{i,\{0\ldots L_i-1\}}(z)$.
5. Calculate $||F_{i,\{0\ldots L_i-1\}}||_2$.
6. $c_i \leftarrow \Pi_{j=0}^{i-1}(1/c_j)/\max\{||F_{i,\{0\ldots L_i-1\}}||_2\}$.
7. **if** $i =$ number of filters
8. **then**
9. **stop**
10. **else**
11. Exchange places for $F_i(z)$ and the upsampler by L_{i+1}
12. $F_{i+1}(z) \leftarrow F_i(z^{L_{i+1}})H_{i+1}(z)$
13. Combine the two leftmost upsamplers into a new upsampler

5. CONCLUDING REMARKS

We have presented a method for scaling multistage interpolators. Compared to earlier published methods it is easier to use in a practical situation. This method is, in its current form, restricted to multistage interpolators using direct-form FIR filters. This is because such filters only need scaling at their inputs. The method can be adopted to the general case where the filters also need scaling internally, eg. IIR filters.

Techniques similar to the one presented in this paper could also be used for roundoff noise calculations. Further, it could be used for decimators, however in that case the stationarity is preserved which simplifies the situation. An extension to interpolators or decimators with rational factors is possible given certain restrictions, but that will be treated in another paper.

REFERENCES

[1] R. R. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Prentice-Hall, 1983.

[2] L. B. Jackson, *Digital Filters and Signal Processing*, 3rd ed. Kluwer Academic Publishers, 1996.

[3] V. P. Sathe and P. P. Vaidyanathan, "Effects of Multirate Systems on the Statistical properties of Random Signals," *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 131–146, January 1993.

[4] S. Akkarakaran and P. P. Vaidyanathan, "Bifrequency and Bispectrum Maps: A New Look at Multirate Systems with Stochastic Inputs," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 723–736, March 2000.

[5] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Prentice-Hall, 1993.