SIGNAL GAIN AND HEADROOM IN RADIO DSP

Torbjorn Larsson
Paradiddle Communications Inc.
torbjorn@paradiddle.us

INTRODUCTION

In two preceding white papers [1] [2], we introduced the concept of *digital noise figure*, i.e. the noise figure of a DSP network, and then described how this fundamental parameter can be used in the design of DSP-based radio solutions. Simply stated, the noise figure of a DSP network is a convenient way to specify the amount of round-off noise generated inside the network. We illustrated this by showing how the noise figure of a digital down-converter (DDC) can be expressed as a function of a small number of bitwidths in its signal path. Not surprisingly, it was found that the noise figure of a digital system, just like the noise figure of an analog system, can be improved by increasing the signal gain in the system. This brings into focus another important issue in radio DSP design, namely *linearity*. Since there is no gain compression in a digital network, ensuring linearity is simply a matter of avoiding overflow (including saturation). However, to achieve this objective without investing an inordinate amount of hardware requires careful control of the headroom in the signal path.

In this third installment of our series on radio DSP design, we focus on the relationship between headroom, signal gain and noise figure in a DSP network. Based on past experience with analog systems, the reader may be used to think of noise figure as being in conflict with linearity; that is, to improve the noise figure of an analog system it is usually necessary to increase the signal gain, which inevitably will bring the signal closer to compression. However, digital systems have the distinct advantage that the gain factor, and thus also the noise figure, can be controlled *independently* of the signal headroom. To demonstrate this, we first define several fundamental parameters, including fractional RMS value, fractional gain and headroom. A quick introduction to headroom estimation, one of the more difficult tasks in signal path design, is also given. We then proceed with a detailed study of signal path design for three common DSP structures: a gain stage, an FIR filter and an IIR filter. Finally, we derive the noise figures of the same three structures and also investigate how the noise figure of an IIR filter is affected by the choice of filter realization (direct form 1 vs. direct form 2).

SIGNAL PATH AND DATAPATH

In radio DSP applications, we are primarily concerned with the design of DSP algorithms defined in integer arithmetic. Such algorithms can often be described as a network of *signal paths* with specified bitwidths. Figure 1a shows a simple example, an FIR (Finite Impulse Response) filter. Along the signal path in the filter we find various integer operations such as `add`, `multiply`, `saturate` and `round`. Because the signal samples are integers, the bitwidth must be known at every node in the signal path in order for the algorithmic behavior to be completely specified. The algorithmic behavior defines the precise relationship between the input and output samples.

To implement a DSP algorithm, the signal path must be mapped onto a physical *datapath* (with associated state machine). In some cases, the mapping is one-to-one, meaning that the datapath looks just like the signal path. In other cases, the datapath is organized in a way that is quite different from the signal path. Either way, it is essential that the mapping does not violate the bitwidth settings in the signal path, lest the algorithmic behavior be affected. For example, the datapath shown in Figure 1b is a multiply-accumulate implementation of the signal network in Figure 1a. It is clear that in order for this datapath to produce the same output as the signal path, the accumulator bitwidth w_{acc} must be greater than or equal to the bitwidth w_{sum} in the signal path in Figure 1a.

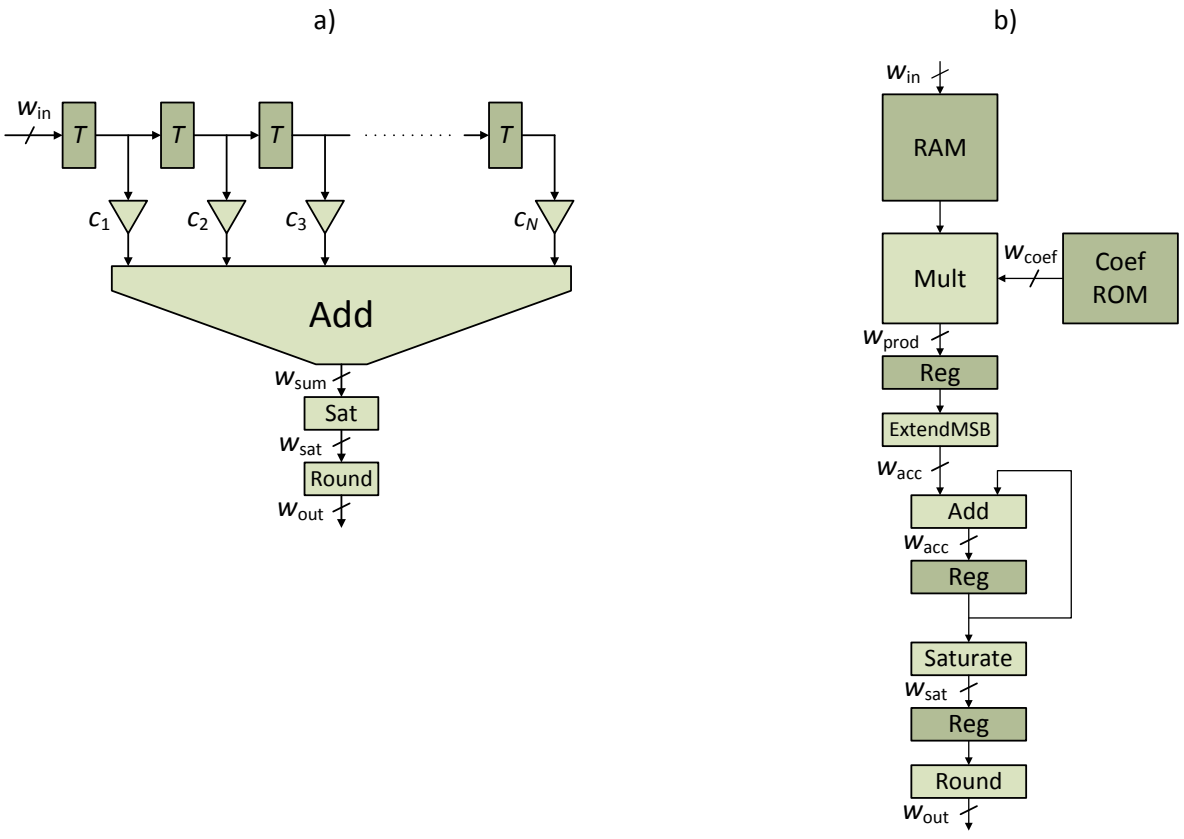

Figure 1 A FIR filter and its multiply-accumulate implementation. a) Signal path. b) Datapath architecture.

The organization of the datapath, the *datapath architecture*, is by definition transparent to the algorithmic behavior. If our implementation of an algorithm does not produce the expected output, this does not mean there is anything wrong with the architecture. We have simply implemented the *wrong algorithm*.

GAIN AND HEADROOM IN LINEAR DSP NETWORKS

Consider the linear DSP network in Figure 2. At the input to the network is a signal with bitwidth w_{in}, power P_{in} and peak amplitude $\hat{A}_{\text{in}}$. Recall that power in the digital domain is the same as mean square value and the square root of power is root-mean square (RMS) value. In response to the input signal, the network produces an output signal with bitwidth w_{out}, power P_{out} and peak amplitude $\hat{A}_{\text{out}}$. When setting the bitwidths in a DSP network, we are primarily concerned with three things: the amount of signal gain provided, how close the output peak value is to the full scale (FS) value, and the amount of noise introduced. We will begin by focusing on the first two issues.

The signal gain in the network can be defined as $G = \sqrt{P_{\text{out}}/P_{\text{in}}}$. In most cases, the calculation of signal gain in radio DSP applications is a straightforward affair. This is because we are dealing with the gain experienced by a "user" signal that occupies a specific portion of the frequency axis, the "user band". The network may be frequency selective, i.e. the gain may be frequency dependent, but is normally designed with a passband that is wide enough to contain the user band. Because the frequency response within the passband is virtually flat (ignoring small variations like ripple and droop), the network appears to the user signal as a frequency nonselective system. Usually a good approximation to the signal gain can be obtained by evaluating the gain at the center of the passband. For a lowpass filter this is the DC gain, which is especially simple to calculate.

Because the possibility of overflow is a major concern in DSP systems, it is often desirable to measure the RMS value relative to the full-scale (FS) value. We refer to this important parameter as the *fractional RMS value*. The fractional RMS value at a signal node with bitwidth w and RMS value $\sqrt{P}$ is defined as $\rho = \sqrt{P}/2^{w-1}$. A fundamental characteristic of a DSP network is its *fractional gain*, given by

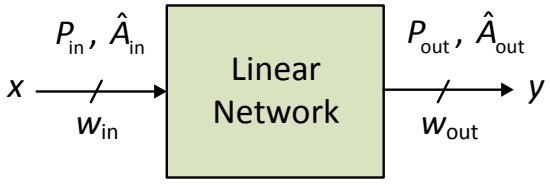

Figure 2 A linear DSP network with one input and one output.

$$\gamma = \frac{\rho_{\text{out}}}{\rho_{\text{in}}} = \frac{\sqrt{P_{\text{out}}}/2^{w_{\text{out}}-1}}{\sqrt{P_{\text{in}}}/2^{w_{\text{in}}-1}} = G \cdot 2^{w_{\text{in}}-w_{\text{out}}} \quad (1)$$

The fractional gain measures the change in signal level relative to the FS value from input to output. From Eq. (1), it is seen that the signal gain can be written as

$$G = \gamma \cdot 2^{w_{\text{out}}-w_{\text{in}}} \quad (2)$$

The conclusion is that a DSP network has two types of gains, signal gain and fractional gain. Somewhat oversimplified, it can be said that the signal gain affects the ability of the system to handle small signal levels without compromising the required SNR, whereas the fractional gain affects the ability to withstand large signal levels without non-linear behavior (overflow).

Just like we defined fractional RMS value, we can also define the *fractional peak value*, given by $\hat{\alpha} = \hat{A}/2^{w-1}$. The two parameters are related through the *peak-to-average ratio* (PAR), defined as $\hat{\alpha}/\rho$. Because the fractional peak value is the ultimate indicator of how close the signal is to overflow, we are often interested in estimating this parameter. Note that the fractional peak value can be expressed as $\hat{\alpha} = \rho \cdot PAR$. The estimation of fractional peak value can therefore be broken into two separate tasks: the estimation of fractional RMS value and the estimation of PAR. It should be mentioned that the peak value is often defined in a statistical sense, as the amplitude value that exceeds or equals a certain percentage of all the amplitude samples. For example, if a peak value measurement is based on the 99.99 percentile, then the reported peak value $\hat{A}$ is the amplitude that makes 99.99% of the amplitude samples in the measurement less than or equal to $\hat{A}$.

As an alternative to working with fractional peak value, it is common to use *headroom*. Headroom is defined as the ratio of FS value to peak value and is therefore simply the inverse of the fractional peak value. However, here we will make the additional distinction that headroom is always given in dB scale, denoted by *HR* and defined as $HR = -20\log_{10}\hat{\alpha}$. When estimating the headroom at the output of a linear filter, it is usually best to ignore transients and instead focus on the steady-state response, as

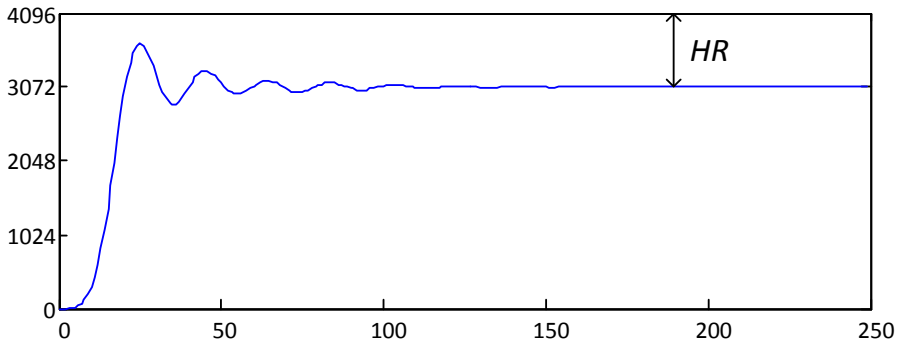

Figure 3 When estimating the headroom at the output of a filter, focus on the steady-state response.

illustrated in Figure 3 for an IIR filter. Transients are much harder to characterize, and overflow (in the form of clipping) can sometimes be accepted during extreme and very rare transients. In addition, wireless transmitters are usually required to ramp up the signal power in a controlled manner when starting a transmission, which eliminates many transients that would otherwise appear in the receiver. Our recommended approach is to estimate the output headroom assuming steady-state operation, but ensure that there is enough headroom to also handle normal transients.

An important property of a DSP network is the amount by which the headroom changes from input to output:

$$HR_{\text{out}} - HR_{\text{in}} = -20\log_{10}\hat{\alpha}_{\text{out}} + 20\log_{10}\hat{\alpha}_{\text{in}} = -20\log_{10}\left(\hat{\alpha}_{\text{out}}/\hat{\alpha}_{\text{in}}\right) \tag{3}$$

Compared to calculating signal gain, calculating the headroom change can be exceedingly difficult. This is because signals on *all* frequencies contribute to the peak value, not just the signal in the user band. Typically in a radio receiver, there are strong unwanted signals present at various frequencies outside the user band. If the DSP network is frequency selective (a filter), all these input signals will experience different gains, thus changing the signal composition at the output. Estimating what this will do to the fractional peak value can be a very challenging task. We can see this more clearly by expressing the peak values in Eq. (3) in terms of fractional RMS value and PAR,

$$HR_{\text{out}} - HR_{\text{in}} = -20\log_{10}\left(\frac{\rho_{\text{out}}}{\rho_{\text{in}}}\cdot\frac{PAR_{\text{out}}}{PAR_{\text{in}}}\right) \tag{4}$$

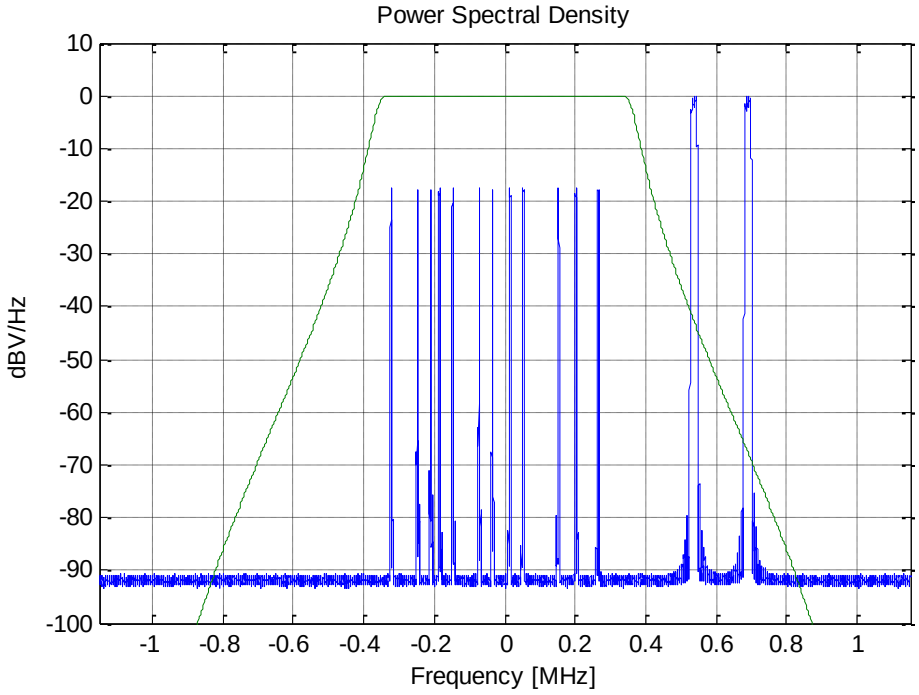

Figure 4 Filter frequency response and input signal spectrum.

It is seen that in order to estimate the change in headroom, we must estimate both the change in fractional RMS value and the change in PAR. Estimating the change in fractional RMS value is relatively straightforward, assuming that we know the frequency location and power of each input signal (a worst-case constellation of input signals can usually be defined for this purpose). The real issue, however, is the PAR estimation. Note that the fractional RMS value and the PAR are essentially unrelated and do not necessarily move in the same direction; it is perfectly possible for a filter to reduce the fractional RMS value and simultaneously increase the PAR.

As an illustration, consider Figure 4, which shows the frequency response of a filter together with the PSD of its composite input waveform. The individual signals that make up the input waveform are all QPSK-modulated carriers with raised-cosine spectral shaping and 40% roll-off, but the two signals outside the filter passband are generated by transmitters that operate in a higher power class and at a higher symbol rate. Clearly, the fractional RMS value will drop significantly when the two high-power signals are suppressed by the filter (the drop is easily calculated to be roughly 18 dB), but what about the PAR? One way to determine the change in the PAR is by simulation. Because a simple floating-point simulation model is sufficient for this purpose, the simulation can be run before the bitwidths in the signal path have been specified. Another approach is to analyze the signal composition at input and output and use tabulated PAR data for the modulation formats involved to estimate the change in the PAR. We can apply this method to the case in Figure 4 with the help of Table 1, which gives the 99.99% PAR for a superposition of QPSK-modulated carriers as a function of the number of carriers. At the input of the filter, the composite waveform is dominated by the two high-power signals and the PAR should therefore be roughly equal to that of two equal-power, QPSK-modulated carriers (8.2 dB). At the output of the filter, the two high-power signals are no longer present and the output waveform is now composed entirely of 12 carriers with roughly equal power. Hence, we can expect that the output PAR is similar to that of a sum of 12 equal-power, QPSK-modulated carriers (11.7 dB). In the simple case shown in Figure 4, this method produces fairly accurate results: the actual input and output PAR values, when measured by simulation, were found to be 8.5 dB and 11.7 dB, respectively. Other cases are not so clear-cut. In practice, a healthy margin should always be added to the required headroom in order to account for the inaccuracy in the PAR estimate.

Table 1 99.99 percentile PAR of *N* superimposed equal-power, QPSK-modulated carriers with raised-cosine pulse shaping and 40% roll-off.

N	**PAR [dB]**	***N***	**PAR [dB]**	***N***	**PAR [dB]**
1	6.0	5	10.6	9	11.5
2	8.2	6	11.0	10	11.6
3	9.3	7	11.2	11	11.7
4	10.1	8	11.4	12	11.7

There is one case in which the calculation of headroom change is very straightforward, namely the case with a frequency nonselective network. Here the signals on all frequencies experience the same gain, which means that $PAR_{out} = PAR_{in}$ and consequently

$$HR_{out} - HR_{in} = -20\log_{10}\left(\rho_{out}/\rho_{in}\right) = -20\log_{10}\gamma \tag{5}$$

Calculating the fractional gain γ is usually a simple task for this type of network. For example, in [1] we studied the gain of a CORDIC rotator [3] and found that the fractional gain was approximately equal to 0.82 (with only a weak dependence on the number of iterations in the algorithm). From this we can conclude that a CORDIC rotator increases the signal headroom by $-20\log_2(0.82) = 1.7$ dB.

GAIN CALCULATIONS

Simple Gain Stage

With the above definitions in mind, let us calculate the gains of some common DSP networks, starting with the simple gain stage in Figure 5a. This is a frequency nonselective network where the input signal is multiplied by a positive constant c_0. The multiplication leads to an expansion of the signal bitwidth from to w_{in} to $w_{prod} = w_{in} + n_0$, where $n_0 = \text{ceil}(\log 2(c_0))$ is the smallest number of bits required to represent c_0. Good design practice dictates that the signal bitwidth should always be expanded when adding and multiplying to ensure that overflow cannot occur for *any* input waveform. Overflow in 2's complement arithmetic leads to wrap-around, a catastrophic type of behavior that should generally be avoided (except under certain circumstances that are outside the scope of this paper). It might be tempting to use various assumptions about the input signal, including assumptions regarding the input headroom, to limit the bitwidth growth in the signal path and "save some hardware". However, should these assumptions change later in the project (as they have a tendency to do), this could violate the whole design and lead to errors that are hard to identify. The only safe approach is to assume that the input signal has fractional peak value of 1 (i.e. zero headroom), which in the case of the gain stage in Figure 5a means that the smallest number of bits required to avoid overflow after multiplication with c_0 is $w_{in} + \text{ceil}(\log_2 c_0)$. Because the expanded bitwidth typically provides more resolution than necessary, a `round` operation is subsequently used to delete a certain number of the least significant bits. Rounding is employed instead of simple truncation in order to avoid a bias, or DC offset, in the output signal.

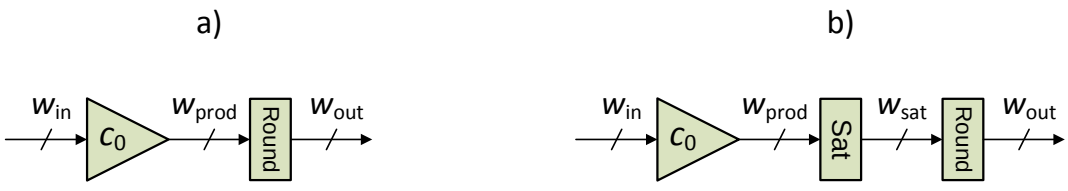

Figure 5 Gain stage. a) Without saturation. b) With saturation.

Figure 6 depicts the mathematical model of a round operation that deletes *n* bits. It is seen that the round operation acts as an attenuator that scales the samples by a factor $1/2^n$, followed by an additive noise source. Under normal conditions, the noise samples generated by the round operation will form a stationary white process with power spectral density (PSD) given by

$$\eta_0 = \frac{1}{6f_s} = \frac{T_s}{6} \tag{6}$$

where $f_s = 1/T_s$ is the sample rate. A more thorough discussion of round-off noise is given in reference [1]. For the calculation of signal gain, however, we can ignore the noise and focus on the scale factor in the `round` operation. In the gain stage of Figure 5a, the number of bits deleted in the `round` operation is $w_{\text{prod}} - w_{\text{out}}$. It follows that the total signal gain in the network is given by

$$G = c_0 \cdot 2^{-(w_{\text{prod}} - w_{\text{out}})} = c_0 \cdot 2^{-(w_{\text{in}} + n_0 - w_{\text{out}})} = \frac{c_0}{2^{n_0}} \cdot 2^{w_{\text{out}} - w_{\text{in}}} \tag{7}$$

Comparing Eq. (7) with Eq. (2), we see that the fractional gain in this case is $\gamma = c_0 / 2^{n_0}$. Note that $0.5 < \gamma \leq 1$ due to the definition of n_0. Hence, this DSP network can only increase the headroom, not reduce it. Suppose for instance that $c_0 = 6$, which gives $n_0 = 3$. The fractional gain is then 6/8 ≈ −2.5 dB and the change in headroom achieved by the gain stage is +2.5 dB. Notice that the fractional gain is determined only by our choice of coefficient c_0, whereas the signal gain *G* also depends on the difference between the output and input bitwidths. In a typical design scenario, coefficients like c_0 are calculated well *before* the bitwidths in the signal path, which means that the fractional gain is specified first, with the aim to achieve a desired headroom change. The signal gain is specified last and is primarily controlled by setting the input and output bitwidths.

Improved Gain Stage

By introducing a `saturate` operation after the coefficient multiplier, as shown in Figure 5b, we obtain a network that can also reduce the signal headroom, i.e. raise the signal level relative to the FS value. Generally, the purpose of a `saturate` operation is to remove a fixed number (typically just one or two) of the most significant bits, while ensuring that no wrap-around occurs. The latter is achieved

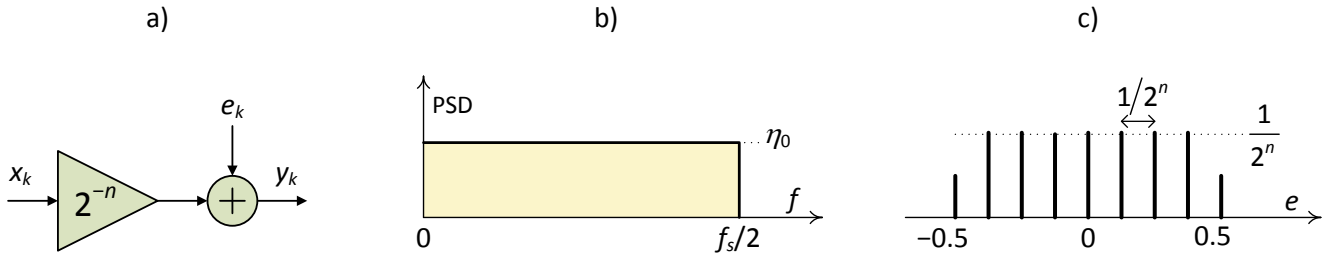

Figure 6 Mathematical representation of rounding. a) Equivalent model. b) Power spectral density of round-off noise. c) Probability mass function of round-off noise (*n* = 3).

by limiting (i.e. clipping) those sample values that cannot be represented in the saturation bitwidth w_{sat}. Although clipping is a non-linear operation, it is preferable to wrap-around, and by setting the saturation bitwidth appropriately, the probability of clipping can be made very small. Let $\delta \geq 0$ denote the number of bits removed by the `saturate` operation, so that $w_{sat} = w_{in} + n_0 - \delta$. The signal gain of the DSP network in Figure 5b is then given by

$$G = c_0 \cdot 2^{-(w_{sat} - w_{out})} = c_0 \cdot 2^{-(w_{in} + n_0 - \delta - w_{out})} = \frac{c_0}{2^{n_0 - \delta}} \cdot 2^{w_{out} - w_{in}} \tag{8}$$

We see that the fractional gain is now $\gamma = 2^{\delta} \cdot c_0 / 2^{n_0}$, which is greater than 1 for any non-zero δ. For example, suppose that the input headroom is known to be 5 dB. By choosing c_0 = 6 and δ = 1, we obtain a fractional gain of γ = 6/4 (3.5 dB), which corresponds to a headroom change of −3.5 dB. The resulting output headroom is therefore 5 − 3.5 = 1.5 dB. Clearly, we would not want to let δ = 2 in this case, since this would not only force the output headroom to zero, but most likely also produce an unacceptably high probability of clipping.

FIR Filter

We are now ready to tackle the FIR filter in Figure 1a. Following the same principle as before, we let the bitwidth grow inside the arithmetic operations wherever it is necessary to ensure that no input waveform can cause an overflow condition. After the last adder, the bitwidth is reduced again by a `saturate` operation followed by a `round` operation. Notice that by first expanding the bitwidth in the arithmetic signal path and then reducing it again in the saturate operation, we have replaced the possibility of a catastrophic non-linear behavior (wrap-around) with the possibility of a more well-behaved non-linear behavior (clipping). The choice of saturation bitwidth may involve assumptions about the input signal, such as headroom. However, should these assumptions change later in the project, then at least the performance degradation is guaranteed to be graceful, and the remedy is clear: increase the saturation bitwidth. To avoid overflow in the adder tree, the signal bitwidth at the output of the last adder (w_{sum}) must equal $w_{in} + n_1$ where $n_1 = \text{ceil}\left(\log_2\left(G_1\right)\right)$ and

$$G_1 = \sum_{k=1}^{N} |c_k| \tag{9}$$

Observe that FIR filter coefficients in general are signed numbers. By expanding the bitwidth in this way, we have ensured that no overflow can occur even in the worst-case scenario where the samples stored in the shift register all have FS amplitude and signs that make all the products add constructively in the adder tree.

When setting the saturation bitwidth, we can ignore this extreme case and instead focus on the steady state behavior of the filter. A conservative approach is to assume that the headroom at the filter input is zero and that all input signals occupy frequency bands that fall within the passband of the filter. The number of bits required to represent the (steady state) signal at the output of the adder tree is then

$w_{in} + n_0$, where $n_0 = \text{ceil}(\log_2 G_0)$ and G_0 is the gain at the center of the passband. Assuming that we are dealing with a lowpass filter, this is the DC gain, given by

$$G_0 = \sum_{k=1}^{N} c_k \tag{10}$$

We refer to this as the *nominal* setting of the saturation bitwidth. Notice that if $n_0 = n_1$, there is no saturation. However, it is common to have $n_0 = n_1 - 1$, in which case the saturation operation removes one bit. The total DC gain in the FIR filter, including the scale factor in the round operation, is obtained as

$$G = G_0 \cdot 2^{-(w_{sat} - w_{out})} = G_0 \cdot 2^{-(w_{in} + n_0 - w_{out})} = \frac{G_0}{2^{n_0}} \cdot 2^{w_{out} - w_{in}} \tag{11}$$

We see that the fractional gain is $\gamma = G_0 / 2^{n_0}$, which cannot exceed 1. Hence, this FIR filter can increase the headroom, but not reduce it. Since there is rarely any need to use the fractional gain to increase the headroom at the output of a filter, it is common to scale the filter coefficients so that $G_0 \approx 2^{n_0}$ and $\gamma \approx 1$.

FIR Filter with Headroom Adjustment

The nominal setting of the saturation bitwidth rests on the assumption that all input signal components see the passband gain when passing through the filter. In many cases, this is far too conservative. The strongest input signals may in fact fall in the stopband of the filter and thus become virtually eliminated. The result is a significant drop in the output signal level, or equivalently, an increase in the output headroom. To avoid wasting dynamic range, we may want to compensate for the suppression of signal components by increasing the fractional gain in the passband. Just like in the case with the gain stage above, the `saturate` operation can be used to achieve a fractional gain greater than 1. This is done by modifying the saturation bitwidth to $w_{sat} = w_{in} + n_0 - \delta$ where δ is a small non-negative integer. We refer to δ as a *headroom adjustment*. The DC gain in the FIR filter is now given by

$$G = G_0 \cdot 2^{-(w_{sat} - w_{out})} = G_0 \cdot 2^{-(w_{in} + n_0 - \delta - w_{out})} = \frac{G_0}{2^{n_0 - \delta}} \cdot 2^{w_{out} - w_{in}} \tag{12}$$

with a fractional gain of $\gamma = 2^{\delta} \cdot G_0 / 2^{n_0}$. For every additional bit removed by the saturate operation, the fractional gain increases by a factor of two (6 dB). It is rare that the headroom needs to be reduced by more than 6 dB, so the most common headroom adjustments are 0 and 1.

Example: FIR Filter for Passband Droop Compensation

To illustrate what we have covered so far, let us consider the FIR filter with three taps shown in Figure 7. This is a filter used in combination with CIC (Cascaded Integrator Comb) filters to compensate for the passband droop caused by such filters [4]. Notice that the last `add` operation is actually a subtractor, so the coefficients in the filter are (−1, c_0, −1). The parameter values for this particular example, taken from the Digital Down-Converter (DDC) design in [1], are given in Table 2. To avoid overflow, the bitwidth after the last `add` operation must be set to $w_{sum} = w_{in} + n_1$, where $n_1 = \text{ceil}\left(\log_2(1+7+1)\right) = 4$. The DC gain at the same point in the signal path is $G_0 = -1+7-1 = 5$, which gives $n_0 = 3$. Since $n_0 < n_1$ in this case, we need a `saturate` operation after the last adder. The saturation bitwidth should be set to $w_{sat} = w_{in} + n_0 = w_{in} + 3$, and the resulting signal (DC) gain can be calculated as

Table 2 Parameter Settings For the FIR Filter in Figure 7.

Parameter	Value
c_0	7
w_{in}	16
w_{sum}	20
w_{sat}	19
w_{out}	15

$$G = G_0 \cdot 2^{-(w_{sat}-w_{out})} = 5 \cdot 2^{-(w_{in}+3-w_{out})} = \frac{5}{8} \cdot 2^{w_{out}-w_{in}} \tag{13}$$

With w_{in} = 16 and w_{out} = 15, the signal gain evaluates to G = 5/16 ≈ 0.31. Notice that the fractional gain is 5/8 = 0.625, so this filter actually increases the signal headroom by 4.1 dB. Such a big increase in the headroom is generally undesirable, and in most other cases we would have preferred to raise the fractional gain by re-scaling the three filter coefficients. However, in this particular case, re-scaling the coefficients would compromise the main advantage of the filter structure: its extreme simplicity. As pointed out earlier, two of the coefficients equal −1 and therefore require no multiplier. Moreover, in a custom DSP implementation the multiplication with c_0 = 7 only requires a single adder (strictly speaking, a subtractor), since the constant multiplication 7·x is synthesized as 8·x – x. It should also be observed that the filter has no actual stopband, so a headroom adjustment is not motivated, unless we can assume that there already exists a headroom of at least 1.9 dB at the input.

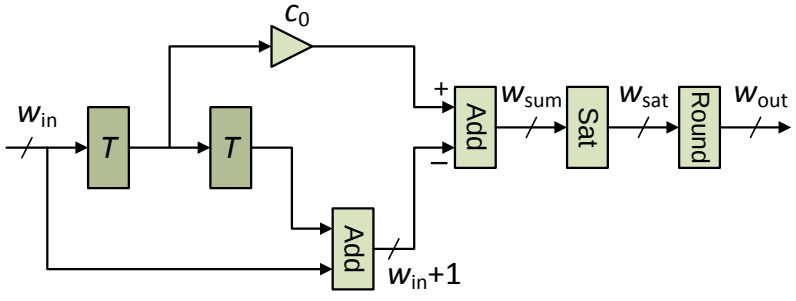

Figure 7 FIR filter for passband droop compensation.

Second-Order Lowpass IIR Filter

Figure 8 shows a lowpass IIR (Infinite Impulse Response) filter that implements the difference equation $y_n = x_n + a_1 y_{n-1} - a_2 y_{n-2}$. The feedback coefficients a_1 and a_2 are unsigned fixed-point numbers with bitwidth w_a. Stability requires that $a_1 < 2$ and $a_2 < 1$, which means that both coefficients can be given in the format x.xxx…x with one bit in the integer part and $w_a - 1$ bits in the fraction. In the signal path, multiplication with the two feedback coefficients is implemented as multiplication with the integer equivalents $\tilde{a}_1 = a_1 2^{w_a - 1}$ and $\tilde{a}_2 = a_2 2^{w_a - 1}$ followed by a `round` operation that removes $w_a - 1$ bits (i.e. divides by $2^{w_a - 1}$). As usual, we let the signal bitwidth grow in the `add` and `multiply` operations to eliminate the possibility of overflow, and then use a `saturate` operation (in addition to the `round` operation) to remove the superfluous bits. The bitwidth after adding the input sample is $w_{out} + 3$, which leaves 3 bits to be removed by the `saturate` operation.

The output bitwidth must be set so as to accommodate the increase in signal level in the feedback loop, given by the DC gain

$$G_{FB} = \frac{1}{1 - a_1 + a_2} \tag{14}$$

Hence, we must have $w_{out} = w_{in} + n_a$ where $n_a = \text{ceil}\left(\log_2 \left(G_{FB} \right) \right)$. In this filter structure, there is no `round` operation at the output, so the signal gain G is equal to the feedback loop gain G_{FB}, which is independent of the input and output bitwidths. However, we can still calculate the fractional gain according to Eq. (1),

$$\gamma = G \cdot 2^{w_{in} - w_{out}} = G_{FB} \cdot 2^{-n_a} = \frac{1 / \left(1 - a_1 + a_2 \right)}{2^{n_a}} \tag{15}$$

By the definition of n_a, the fractional gain in Eq. (15) is ≤ 1. In general, it is a good idea to have a decent

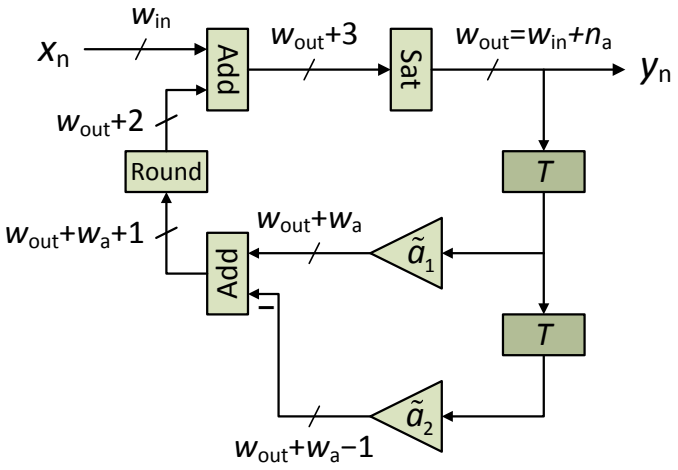

Figure 8 Second-order IIR filter.

amount of headroom inside the feedback loop of an IIR filter (say 2 dB) to be able to handle transients without clipping. If the headroom is deemed insufficient, it is always possible to increment n_a, thereby reducing the fractional gain in the loop by 6 dB. The extra bit can be removed by another `saturate` operation placed *after* the feedback loop.

Biquad Filter

By combining the feedback section in Figure 8 with a feed-forward section as shown in Figure 9, we obtain a complete second-order system with two poles and two zeros, often referred to as a *biquad* [5]. This system realizes the difference equation $y_n = b_0 x_n + b_1 x_{n-1} + b_2 x_{n-2} + a_1 y_{n-1} - a_2 y_{n-2}$. Again we confine the discussion to lowpass filters, in which case the three *b*-coefficients are all positive. As before, the bitwidth in the feedback loop is $w_{FB} = w_{in} + n_a$ where n_a is the number of bits required to represent the feedback loop gain G_{FB} in Eq. (14). To guarantee that no overflow can occur, the bitwidth in the feed-forward section must grow by n_b bits, where $n_b = \text{ceil}(\log_2 G_{FF})$ and $G_{FF} = b_0 + b_1 + b_2$. Observe that even if there is a decent amount of headroom in the feedback loop, we should ignore this when setting parameter n_b, because a transient may still produce a full-scale value at the input to the feed-forward section. The bitwidth at the output of the last `add` operation in the biquad is now $w_{in} + n_a + n_b$, which is guaranteed to eliminate overflow, but may in some instances create more headroom than desired. To determine whether a headroom adjustment is necessary, we first note that the total DC gain in the feedback and feed-forward sections is

$$G_0 = G_{FB} \cdot G_{FF} = \frac{b_0 + b_1 + b_2}{1 - a_1 + a_2} \tag{16}$$

Hence, the signal at the output of the feed-forward section should occupy no more than $w_{in} + n_0$ bits, where $n_0 = \text{ceil}(\log_2 G_0)$ and G_0 is given by Eq. (16). If $n_0 < n_a + n_b$, we need a saturate operation to

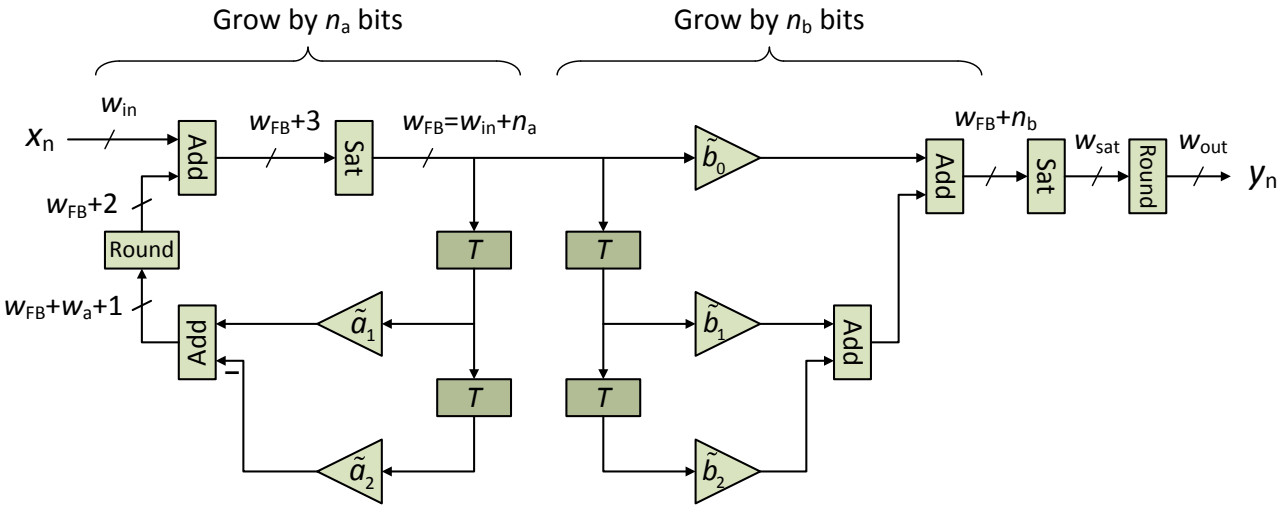

Figure 9 Biquad (direct form 2).

reduce the bitwidth to $w_{sat} = w_{in} + n_0$ bits, as shown in the figure. Combining the DC gain G_0 with the scale factor in the `round` operation, the signal gain in the biquad can be obtained as

$$G = G_0 \cdot 2^{-(w_{sat}-w_{out})} = G_0 \cdot 2^{-(w_{in}+n_0-w_{out})} = \frac{b_0+b_1+b_2}{1-a_1+a_2} \cdot 2^{-n_0} \cdot 2^{w_{out}-w_{in}} \tag{17}$$

As usual we find that the fractional gain cannot be made to exceed 1, unless we make a headroom adjustment and let $w_{sat} = w_{in} + n_0 - \delta$ for some positive integer δ. Allowing for this modification, the general expression for the fractional gain of a second-order IIR filter becomes

$$\gamma = \frac{b_0+b_1+b_2}{1-a_1+a_2} \cdot 2^{-n_0+\delta} \tag{18}$$

Notice that although the a-coefficients are completely constrained by the filter design (pole locations), we are free to scale the *b*-coefficients to obtain any desired fractional gain. In standard lowpass designs, the *b*-coefficients are always set to achieve a double zero at $z = -1$, which means that $(b_0, b_1, b_2) = c \cdot (1,2,1)$ for some scale factor *c*. The DSP network for this case is shown in Figure 10, where we have also eliminated the redundant sample storage. The fractional gain is now given by

$$\gamma = \frac{4c}{1-a_1+a_2} \cdot 2^{-n_0+\delta} \tag{19}$$

As an illustration, let us consider a Chebyshev design with passband edge $0.05 \cdot f_s$ and 0.25 dB ripple. A coefficient bitwidth of $w_a = 12$ was chosen for this example and the resulting feedback coefficients are given in Table 3. The frequency response is shown in Figure 11. Evaluating first the DC gain in the feedback loop according to Eq. (14), we find that $G_{FB} = 6.3$, which gives $n_a = 3$. The fractional gain in the feedback loop is 0.79, calculated using Eq. (15). Now, suppose that an overall fractional gain in the range 0.9 – 1.0 is desired. By evaluating Eq. (19) for $c = 1,2,3,\ldots$, we then find that the smallest value of

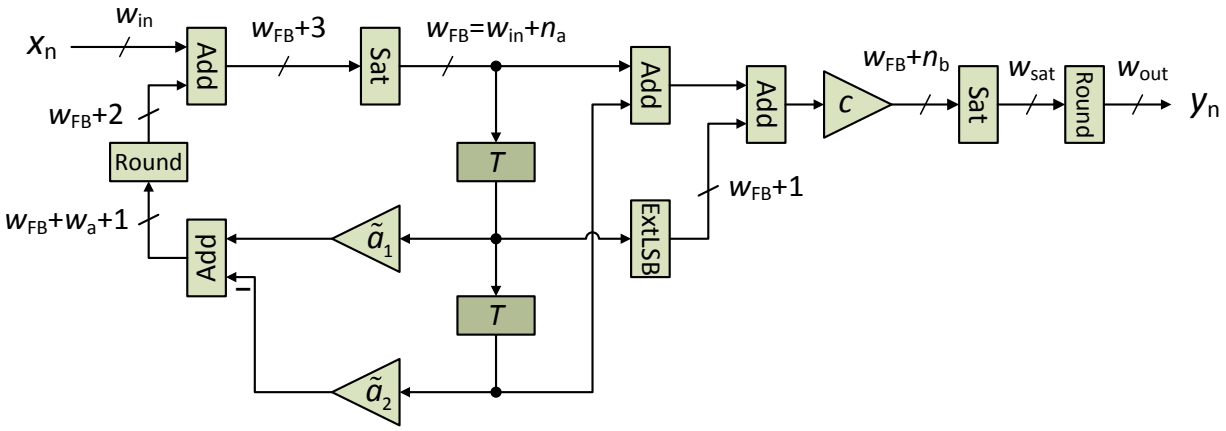

Figure 10 Lowpass biquad with a double zero at $z = -1$ (direct form 2).

Table 3 Parameter Settings For the Biquad Filter in Figure 10.

Parameter	Value
w_a	12
a_1	1.41601562500
a_2	0.57470703125
c	5
n_a	3
n_b	5
n_0	7

c that meets this requirement is $c = 5$, which gives $G_{FF} = 4c = 20$ and $G_0 = 6.3 \cdot 20 = 126$ and $n_0 = 7$. The resulting fractional gain in the biquad is $\gamma = 126/2^7 \approx 0.985$. Notice that $n_a + n_b = 8$, which means that the `saturate` operation at the output is indeed necessary in this case (since $n_0 = 7$). Naturally, when setting the fractional gain in a filter close to 1, we must be certain that the headroom at the filter input is sufficient to handle transients.

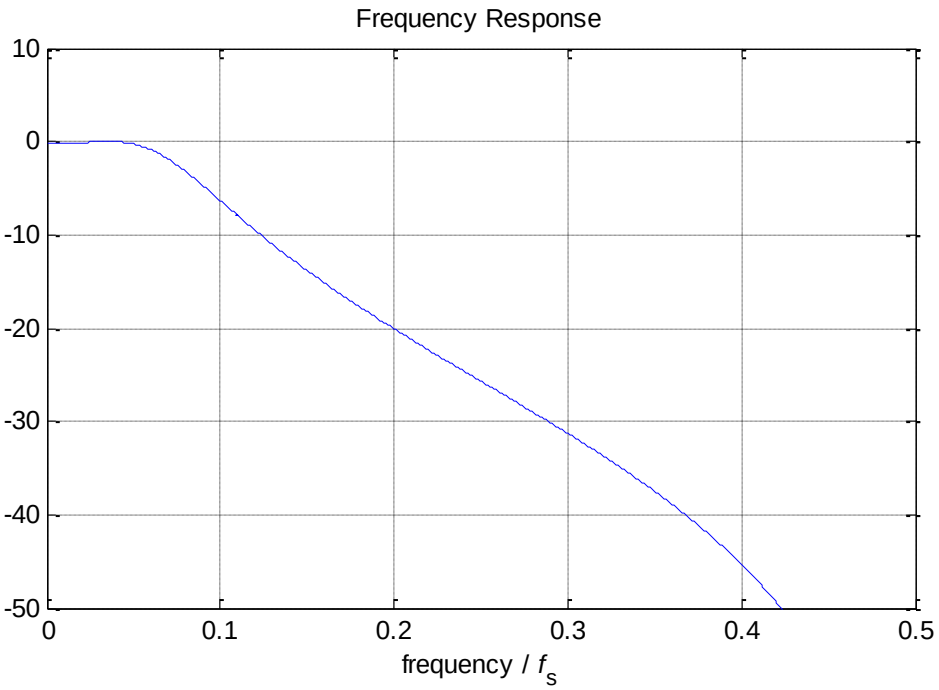

Figure 11 Frequency response of a biquad filter with parameter settings according to Table 3.

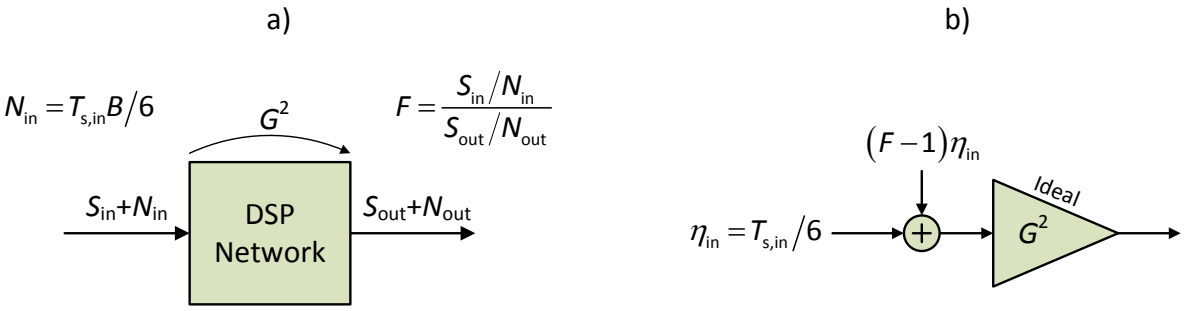

Figure 13 Noise figure in the digital domain. a) Basic setup. b) Equivalent system model with input-referred excess noise PSD.

NOISE FIGURE CALCULATIONS

In this section, we use the results derived above to calculate the noise figures of some common DSP structures. Consider first a DSP network with signal gain *G*, input sample rate $f_{s,in} = 1/T_{s,in}$ and output sample rate $f_{s,out} = 1/T_{s,out}$. The noise factor (noise figure in linear scale) of this network can be defined as the degradation of SNR from input to output, as illustrated in Figure 13a. It is easily shown [1] that the noise factor can be calculated using the formula

$$F = 1 + \frac{\eta_{ex}}{G^2 \eta_{in}} \tag{20}$$

where η_{ex} is the PSD of the *output-referred excess noise*, i.e. the contribution of internal noise sources to the noise level at the output, and η_{in} is the PSD of the "source" noise present at the input, given by $\eta_{in} = T_{s,in}/6$. Note that η_{in} is equivalent to the noise level *kT* in the definition of noise factor for analog networks (where *k* is Boltzmann's constant and *T* is the temperature). The excess noise is calculated by adding up the noise contributions from all `round` and `truncate` operations in the network.

It is often convenient to think of the excess noise as being generated by a fictitious noise source located at the input of an ideal (noiseless) system, as shown in Figure 13b. This theoretical noise source, with PSD given by $\eta_{ex}/G^2 = (F-1)\eta_{in}$, is referred to as the *input-referred excess noise*. Notice that since the input noise level η_{in} is a known quantity, the input-referred excess noise level is completely specified by the noise factor.

Let us begin with the general DSP structure in Figure 12, which consists of a noiseless signal path followed by a single round operation. Both the gain stage and the FIR filter discussed earlier take

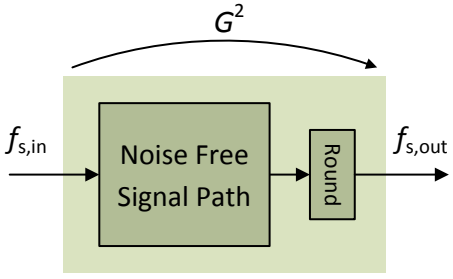

Figure 12 DSP network with one round operation.

this form. To calculate the noise factor of such a system, we first note that the only contributor to the excess noise is the `round` operation at the output. The noise PSD generated by a `round` operation is given by Eq. (6) and it follows immediately that $\eta_{ex} = T_{s,out}/6$. Substituting this result into Eq. (20), the noise factor is obtained as

$$F = 1 + \frac{T_{s,out}/6}{G^2 T_{s,in}/6} = 1 + \frac{1}{G^2 R} \tag{21}$$

where R is the rate conversion factor of the network, defined as $R = f_{s,out}/f_{s,in} = T_{s,in}/T_{s,out}$. To obtain an expression for the noise factor of a gain stage or a FIR filter, all we need to do is let $R = 1$ and substitute the general expression for signal gain in Eq. (2). The result is

$$F = 1 + \frac{1}{\gamma^2} 4^{w_{in} - w_{out}} \qquad \text{(FIR filter or gain stage)} \tag{22}$$

Notice that for any given input bitwidth, the noise factor can be made to approach 1 (i.e. a noise-free system) by making the output bitwidth large enough. From the discussion of gain in the previous section we have $\gamma = G_0 / 2^{n_0 - \delta}$, where G_0 is given by Eq. (10) for a FIR filter and $G_0 = c_0$ for a gain stage.

Not surprisingly, we find that there is a conflict between noise figure and headroom: increasing the fractional gain will reduce (improve) the noise figure but will also reduce (degrade) the headroom. However, in practice this should never be an issue. The fractional gain of a DSP block is controlled by scaling its coefficients or by the headroom adjustment, whereas the noise figure can be controlled by setting the input and output bitwidths. Therefore, when designing a DSP system, the fractional gain of each individual DSP block can (and should) be set *before* the bitwidths in the signal path have been specified. This effectively allows the designer to control the output headroom and the noise figure of a DSP network independently of each other.

For the droop-compensating FIR filter in Figure 7, we have $\gamma = 5/8$, $w_{in} = 16$ and $w_{out} = 15$ (see Table 2). With these parameter values, the noise factor in Eq. (22) evaluates to $F = 1 + 1/(5/8)^2 \cdot 4 = 11.24$, which corresponds to a noise figure of $NF = 10\log_{10}(11.24) = 10.5$ dB.

Next, consider the biquad structure in Figure 9. Recall that this is the so called “direct form 2” realization [6] of an IIR filter. Because there is no rate conversion, we have $T_{s,in} = T_{s,out} = T_s$. Here there are two round operations contributing to the excess noise. The noise generated by the left-most round operation will experience the full signal gain G on its way to the output, and the total excess noise PSD is therefore given by

$$\eta_{ex} = \left(G^2 + 1\right)\eta_0 \tag{23}$$

where $\eta_0 = T_s/6$ as before. After substitution into Eq. (20), this gives

$$F = 1 + \frac{\left(G^2+1\right)\eta_0}{G^2\eta_0} = 2 + \frac{1}{G^2} \tag{24}$$

Finally, by substituting the general expression for the signal gain in Eq. (2), we obtain the following noise factor for direct form 2:

$$F = 2 + \frac{1}{\gamma^2}\cdot 4^{w_{\text{in}}-w_{\text{out}}} \qquad \text{(biquad, direct form 2)} \tag{25}$$

The fractional gain γ for a biquad is given by Eq. (18). We see that the noise factor for direct form 2 cannot be made to approach that of a noise-free system, but instead will approach 2 (*NF* = 3 dB) when $w_{\text{out}} >> w_{\text{in}}$. Is this a fundamental limitation of biquad filters? Fortunately the answer is no. To see this, consider the biquad in Figure 14. Here the order of the feedback and feed-forward sections has been reversed, a filter realization known as "direct form 1" [6]. Notice that we still need to place a saturate operation and a round operation at the output in order to control the output headroom and signal gain, respectively. Changing the order of the two filter sections changes neither the fractional gain, nor the signal gain. However, it does change the noise factor. Since the left-most round operation will no longer see the feed-forward gain G_{FF} on its way to the output, the excess noise is now given by

$$\eta_{\text{ex}} = \left(G^2/G_{\text{FF}}^2+1\right)\eta_0 \tag{26}$$

and it follows that

$$F = 1 + \frac{\left(G^2/G_{\text{FF}}^2+1\right)\eta_0}{G^2\eta_0} = 1 + \frac{1}{G_{\text{FF}}^2} + \frac{1}{G^2} \tag{27}$$

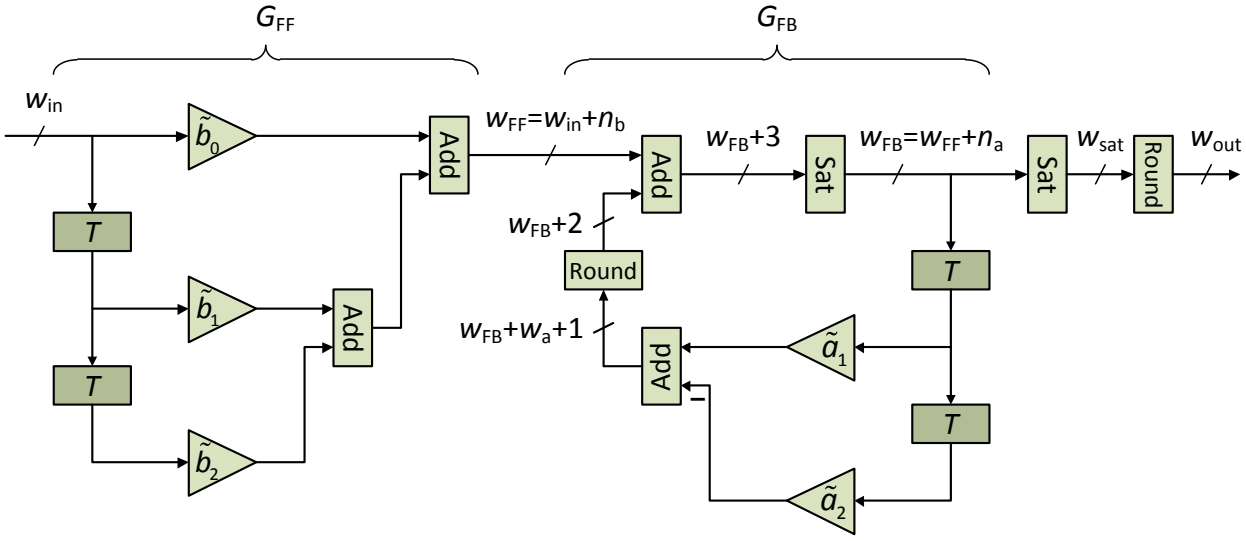

Figure 14 Biquad (direct form 1).

After substituting the general expression for signal gain, the noise factor for direct form 1 is obtained as

$$F = 1 + \frac{1}{G_{FF}^2} + \frac{1}{\gamma^2} 4^{w_{in} - w_{out}} \qquad \text{(biquad, direct form 1)} \qquad (28)$$

Recall that G_{FF} is a function only of the *b*-coefficients and is independent of the input and output bitwidths. By scaling the *b*-coefficients appropriately, we can therefore make the middle term in Eq. (28) small enough that $1 + 1/G_{FF}^2 \approx 1$. The result is that the noise factor approaches that of an FIR filter, given by Eq. (22). The performance improvement comes at a price, however. In the direct form 2 realization, the bitwidth in the feedback loop is $w_{in} + n_a$, whereas in the direct form 1 realization it is $w_{in} + n_b + n_a$. The greater bitwidth in direct form 1 leads to an increased path delay in the logic in the feedback loop, which reduces the speed at which the filter can operate. Direct form 2 may therefore be a better solution for filters operating close to the A/D converter, where the sample rate is high and the noise figure is less critical for the overall noise performance [1].

Finally, let us compare the noise factors obtained when a lowpass biquad with parameters as in Table 3 is implemented in direct form 1 and direct form 2. Note that the parameters in the table can be used with either filter form. From our earlier calculations, we already have $\gamma = 0.985 \approx 1$ and $G_{FF} = 20$ for this example. Substituting these results into Eq. (28) and Eq. (25), we obtain the following noise factors for the two filter realizations:

$$F \approx 1 + 4^{w_{in} - w_{out}} \qquad \text{(direct form 1)} \qquad (29)$$

$$F \approx 2 + 4^{w_{in} - w_{out}} \qquad \text{(direct form 2)} \qquad (30)$$

REFERENCES

[1] T. Larsson, *Noise Figure Calculations in DSP Systems*, Paradiddle Communications Inc., Online at www.paradiddle.us/white-papers/DSPNoiseFigure.pdf.

[2] T. Larsson, *Cascading Analog and Digital Noise Figures*, Paradiddle Communications Inc., Online at www.paradiddle.us/white-papers/CascadingNoiseFigures.pdf.

[3] U. Meyer-Baese, *Digital Signal Processing with Field Programmable Gate Arrays*, Springer, 2004.

[4] R. Lyons, "Turbocharging Interpolated FIR Filters", in *Streamlining Digital Signal Processing*, R. G. Lyons, Ed., IEEE Press, 2007.

[5] R. Lyons and A. Bell, "The Swiss Army Knife of Digital Networks", in *Streamlining Digital Signal Processing*, R. G. Lyons, Ed., IEEE Press, 2007.

[6] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing*, Prentice-Hall, 1996.

