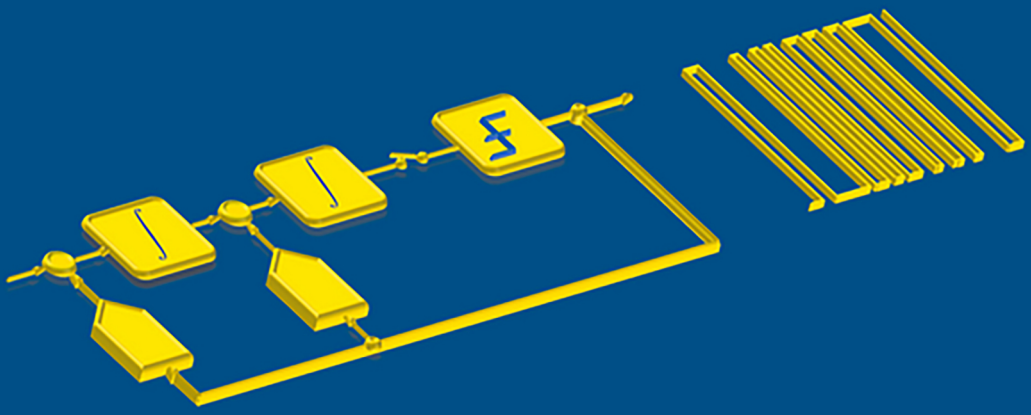
IEEE Press Series on Microelectronic Systems

R. Jacob Baker, Series Editor

Understanding Delta-Sigma Data Converters

Second Edition



Shanthi Pavan
Richard Schreier
Gabor C. Temes

IEEE PRESS

WILEY

UNDERSTANDING DELTA-SIGMA DATA CONVERTERS

IEEE Press
445 Hoes Lane
Piscataway, NJ 08854

UNDERSTANDING DELTA-SIGMA DATA CONVERTERS

SECOND EDITION

SHANTHI PAVAN
RICHARD SCHREIER
GABOR C. TEMES

IEEE Press Series on Microelectronic Systems

IEEE PRESS

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data is available.

ISBN: 978-1-119-25827-8

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

CONTENTS

PREFACE

The earlier incarnation of this book[1] was aimed at describing the principles and operation of delta-sigma ($\Delta\Sigma$) modulators, as used in analog-to-digital (A/D) and digital-to-analog (D/A) converters, in simple conceptual terms, without relying on complicated mathematics. It also provided practical design information for both industrial and academic designers of $\Delta\Sigma$ converters. The book was well received, dubbed the green book, and sold many copies internationally. It was translated into Japanese, and reprinted in China. It is cited currently about 170 times annually in the literature. In view of this continued popularity, why did we embark on creating this new avatar?

The answer is that twelve years have gone by since the green book was published. The interest of $\Delta\Sigma$ converter designers has shifted significantly during this period, in the wake of many new applications for data converters at the extreme ends of the frequency spectrum. Continuous-time $\Delta\Sigma$ ADCs with GHz clocks, both for lowpass and bandpass signals, were required for wireless applications. At the other extreme of the spectrum, multiplexed ADCs with very narrow (sometimes only 10 Hz wide) signal bands, but very high accuracy, were needed e.g. in the interfaces of biomedical or environmental sensors. Often, the optimal converter for these specifications is the incremental ADC, which is basically a $\Delta\Sigma$ ADC that is periodically reset and restarted.

To reflect the changed needs of designers, this book includes much new material on both theory and design techniques. The emphasis of topics in the existing material has also been changed. New chapters have been added on the cascade (MASH) architecture, on

[1] Understanding Delta-Sigma Data Converters, R. Schreier and G. C. Temes, IEEE Press and Wiley-Interscience, 2005.

DAC mismatch effects and their mitigation, as well as expanded chapters on continuous-time $\Delta\Sigma$ ADCs and their nonidealities, on circuit design techniques for both sampled-data and continuous-time ADCs, and on incremental ADCs.

During the past decade, several new books that deal with special aspects of $\Delta\Sigma$ ADCs have been published. A recent book by de la Rosa and del Rio[2] provides an encyclopedic collection of practical information on $\Delta\Sigma$ ADCs. It is a valuable addition to the literature, highly recommended for designers. By contrast, the purpose of our work (as the title implies) is to give a basic understanding of the operation of these converters, and to provide general design techniques. We can think of several possible scenarios for using this book in a classroom setting. Chapters 1 through 6 form the core theory. A semester-long course focusing on discrete-time $\Delta\Sigma$ ADCs should, in addition, cover chapters 7, 12, 13 and 14. A course focusing on CT$\Delta\Sigma$Ms would cover Chapters 1-6, 8-11, and 14.

Several colleagues, from academia and industry, reviewed drafts of the book at various stages. It is our pleasure to acknowledge their assistance. Thanks are due to Trevor Caldwell (Analog Devices), Rakshit Datta (Texas Instruments), Ian Galton (University of California at San Diego), John Khoury (Silicon Laboratories), Victor Kozlov (Analog Devices), Saurabh Saxena (Indian Institute of Technology Madras), and Nan Sun (University of Texas at Austin). Their careful and astute comments have, in our opinion, helped improve the quality of the book. Amrith Sukumaran's editorial assistance is also appreciated.

To stick to limits imposed by space and time, some topics had to be omitted altogether, while others had to be given short shrift. Nevertheless, we hope that this book will be useful both for teaching and for self-education purposes.

SHANTHI PAVAN
Chennai, India

RICHARD SCHREIER
Toronto, Canada

GABOR C. TEMES
Corvallis, USA

[2]CMOS Delta-Sigma Converters, J. M. de la Rosa and R. del Rio, IEEE Press and Wiley-Interscience, 2013.

CHAPTER 1

THE MAGIC OF DELTA-SIGMA MODULATION

The aim of this introductory chapter is to motivate the need for oversampling data converters, and to give a bird's-eye view of the topics covered in this book. Towards the end of the chapter, we give a brief overview of the origins of $\Delta\Sigma$ data conversion and trends in this exciting area.

1.1 The Need for Oversampling Converters

Computational and signal processing tasks are now performed predominantly by digital means, since digital circuits are robust and can be realized by extremely small and simple structures that can in turn be combined to obtain very complex, accurate, and fast systems. Every year, the speed and density (of transistors) of digital integrated circuits (ICs) increase, thereby enhancing the dominance of digital methods in almost all areas of communications and consumer products. Since the physical world nevertheless remains stubbornly analog, data converters are needed to interface with the digital signal processing (DSP) core. As the speed and capability of DSP cores increases, so too must the speed and accuracy of the converters associated with them. This presents a continual challenge to the lucky few engineers dedicated to the design of data converters!

Figure 1.1 illustrates the block diagram of a signal processing system with analog input and output signals, plus a central digital engine. As shown, the analog input signal (usually after some amplification and filtering) enters an analog-to-digital converter

Understanding Delta-Sigma Data Converters, *Second Edition*. By Shanthi Pavan, Richard Schreier, and Gabor C. Temes.

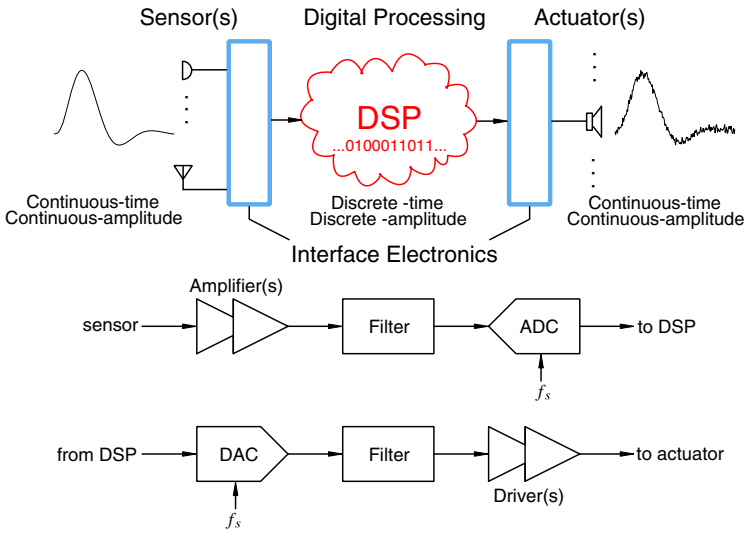


Figure 1.1 ADCs and DACs interface the real world to the virtual world.

(ADC), that transforms the input signal into a digital data stream. This stream is processed by the DSP core, and the resulting digital output signal is reconverted into analog form by a digital-to-analog converter (DAC). The DAC output is usually also filtered and amplified to obtain the final analog output signal.

Data converters (both ADCs and DACs) can be classified into two main categories: Nyquist-rate and oversampled converters. In the former category, there exists a one-to-one correspondence between the input and output samples. Each input sample is separately processed, regardless of the earlier input samples; in other words, the converter has no memory. Applying a digital input word containing bits $b_1, b_2, \cdots, b_N$ to a Nyquist-rate DAC ideally results in an analog output

$$V_{out} = V_{ref}(b_1 2^{-1} + b_2 2^{-2} + \cdots + b_N 2^{-N}), \tag{1.1}$$

where V_{ref} is the reference voltage, regardless of any previous input word. The accuracy of conversion can be evaluated by comparing the actual value of V_{out} with the ideal value given by (1.1).

As the name implies, the sampling rate f_s of a Nyquist-rate converter can be as low as Nyquist's criterion requires, i.e., twice the bandwidth B of the input signal. (For practical reasons, the actual rate is usually somewhat higher than this minimum value.)

In most cases, the linearity and precision of a Nyquist-rate converter is determined by the matching accuracy of the analog components (resistors, current sources, or capacitors) used in the implementation. For example, in the N-bit resistor-string DAC shown in Figure 1.2, the resistors must have a relative matching error less than 2^{-N} to guarantee an integral nonlinearity (INL)[1] less than 0.5 LSB. Similar matching requirements prevail for ADCs and DACs constructed from current sources or switched-capacitor (SC) branches.

[1]INL is simply the difference between the actual output and the ideal output.

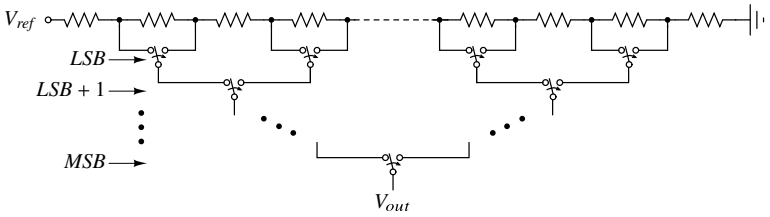


Figure 1.2 A resistor-string DAC. LSB denotes the least significant bit and MSB the most significant bit of the digital input.

Practical conditions restrict the matching accuracy to about 0.02%, and hence the effective number of bits (ENOB) to about 12, for such converters.

In many applications (e.g., digital audio), higher resolution and linearity are required, even as much as 18 or 20 bits. The only Nyquist-rate converters capable of such accuracy are the integrating or counting ones. These, however, require at least 2^N clock periods to convert a single sample, and hence, they are too slow for most signal processing applications.

Oversampling data converters are able to achieve over 20 ENOB resolution at reasonably high conversion speeds by relying on a trade-off. They use sampling rates much higher than the Nyquist rate, typically higher by a factor between 8 and 512, and generate each output utilizing numerous preceding input values. Thus, the converter incorporates memory elements in its structure. This property destroys the one-to-one relation between input and output samples. With oversampling converters, only a comparison of the complete input and output waveforms can be used to evaluate the converter's accuracy, either in the time or in the frequency domain.

A common measure of a converter's accuracy is the signal-to-noise ratio (SNR) for a sine-wave input. The relationship between ENOB and SNR (expressed in dB) for an ideal Nyquist converter with a full-scale sine-wave excitation is $SNR = 6.02\,ENOB + 1.76$. The inverse relationship is often applied to oversampling converters to convert the SNR into an effective number of bits.

As will be shown in later chapters, the implementation of oversampling converters requires a considerable amount of digital circuitry, in addition to some analog stages. Both need to be operated faster than the Nyquist rate. However, the accuracy requirements on the analog components are relaxed compared to those associated with Nyquist-rate converters. The price paid for high accuracy thus includes faster operation and added digital circuitry; both of these are getting cheaper as digital IC technology advances. Hence, the trade-off offered by $\Delta\Sigma$ converters continues to improve. As a result, they are gradually taking over in many applications previously dominated by Nyquist-rate converters.

1.2 Nyquist and Oversampling Conversion by Example

To better understand the difference between Nyquist and oversampled analog-to-digital conversion, consider the following illustrative examples.

1.2.1 The Coffee Shop Problem

A student visits a coffee shop on campus every morning to get her fix of caffeine, so that she can get through the day. A coffee grande at the campus cafe costs \$3.47. What are the ways in which the student can pay this rather inconvenient sum? (The old-fashioned cafe does not accept credit cards). The "Nyquist" way of paying would be for the student to carry coins of the right denominations every day. She could, however, pay with a \$5 bill and expect to shop assistant to return \$1.53. The cafe, unfortunately, is severely short of small coins, and the shop assistant is not in a position to entertain this practice. Nevertheless, the shop assistant and the student come to an understanding that will allow the latter to pay with a \$5 bill, while at the same time not under or overpay the cafe. It exploits the fact that the student visits the cafe *every day*. This is the $\Delta\Sigma$ way, described below.

The agreement between the two parties is the following. On any day, if the student owes the cafe more than \$2.50, she hands a \$5 bill to the shop assistant. When instead, she owes less than \$2.50, she pays nothing. The student keeps track of how much she owes the cafe. The transactions for the first three days are shown in Figure 1.3.

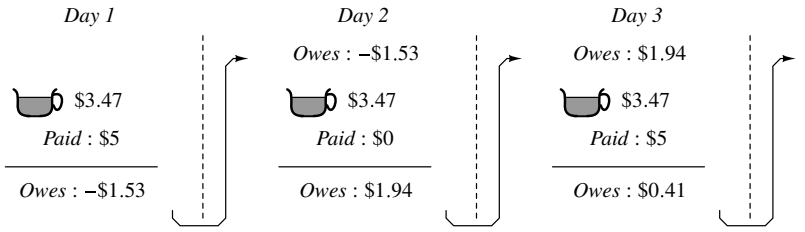


Figure 1.3 The $\Delta\Sigma$ way of paying \$3.47 for a coffee grande, with only \$5 bills.

On the first day, the student pays \$5, as agreed upon. She notes, at the end of the day, that she owes −\$1.53 to the cafe. The minus sign indicates that the student has overpaid.

While ordering at the cafe on the second day, the student reminds the shop assistant of the overpayment the previous day. The student needs to only pay \$(3.47 − 1.53)=\$1.94. As agreed upon, she pays nothing, again noting that the cafe is owed \$1.94.

On the third day, the student needs to pay \$5.41, and as per the understanding with the shop assistant, hands over a \$5 bill. She notes that the cafe is owed \$0.41. This continues every day *ad infinitum*.

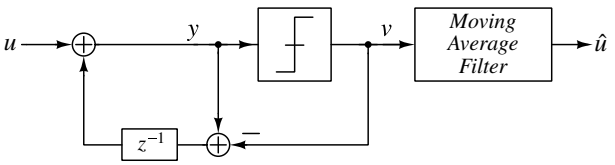


Figure 1.4 The algorithm of Figure 1.3. The u represents the cost of a coffee grande, and $v[n]$ is the payment made by the student on the nth day.

When the scheme above is cast into a signal flow diagram, Figure 1.4 results. In the figure, u presents the price of the coffee grande; $y[n]$, which is the input of the quantizer, represents the total amount owed by the student while ordering on the nth day; $v[n]$, which

is the quantizer output, represents the students payment on the nth day, and takes the value 0 or 5. Therefore, $(y[n] - v[n])$ is the amount owed by the student to the cafe after making the payment on the nth day. The z^{-1} block in Figure 1.4 denotes a delay of one day.

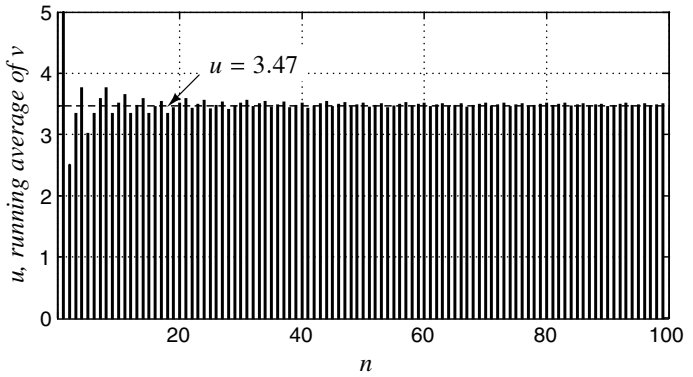


Figure 1.5 The running average of v approaches u for large n.

Figure 1.5 shows the running average of v, given by

$$\frac{1}{n}\sum_{k=1}^{n} v[k]. \tag{1.2}$$

The running average represents the price paid by the student per unit coffee grande, on average during the preceding days. As n becomes large, we see it approaches u, which is \$3.47.

In the beginning of this discussion, it might have seemed surprising that the student would be able to pay an inconvenient sum of \$3.47 with only \$5 bills. The $\Delta\Sigma$ way exploits the fact that u remains substantially the same from sample to sample. It uses feedback to make v approximate u *on average*. An individual sample of v has no meaning – one can determine u from v only by averaging many samples. Why does this scheme work? It is perhaps easier to see this by redrawing the diagram of Figure 1.4 as in Figure 1.6. We see that $y[n]$ is the total amount owed by the student (from the beginning of time) after grabbing her coffee for the day. As long as this is bounded, it must follow that the average of the accumulator's (Σ) input must be close to zero. Since the input to the accumulator is the difference (Δ) between the input and feedback sequence, it should follow that v and u would be equal, on average. Thus, by embedding a (very coarse) 2-level quantizer into a negative feedback loop, and sufficiently averaging the output sequence, the digital estimate $\hat{u}$ can be a very good representation of u.

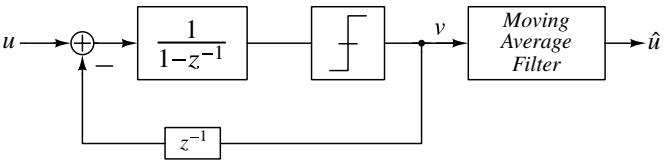


Figure 1.6 The system of Figure 1.4, redrawn.

The feedback loops of Figures. 1.4 and 1.6, both equivalent, represent a first-order $\Delta\Sigma$ modulator. The first structure is called the error-feedback structure, while the latter is the more traditional (and immediately recognizable) error accumulating structure.

1.2.2 The Dictionary Problem

A student visiting a bookstore begins to wonder about the thickness of that venerable tome, the *Webster's International Dictionary of the English Language*. An immediate way of finding the thickness is to get hold of a 6-inch ruler (this is a bookstore, after all) and measure the dictionary's thickness, as illustrated in Figure 1.7. Since the ruler has markings at every eighth of an inch, the worst-case error in measuring the thickness would amount to one-sixteenth of an inch. This is the "Nyquist" way, where the distance between successive marks on the ruler would correspond to the LSB. Measurement uncertainty (quantization error, in data-conversion parlance) can only be reduced by using a ruler with more finely spaced markings. The effort involved in making such a ruler is decidedly higher, not to mention the difficulty in discerning the marking that best corresponds to the height of the tome. Note, however, that the measurement is made in one shot – meaning that one use of the ruler is sufficient for measurement.

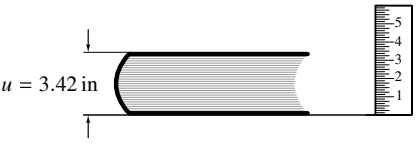


Figure 1.7 Measuring the thickness of *Webster's Dictionary* the Nyquist way.

The student finds focusing on the finely spaced levels a strain on the eyes, and he begins to wonder if it is at all possible to measure the book's thickness without having to look at the marks on the ruler at all. In other words, is it possible to find the thickness to within one-sixteenth of an inch (or even better) using only the fact that this is a 6-inch ruler? At first, this may seem like an impossible task – how is it possible to measure to within a fraction of an inch with a scale whose only "marking" is 6 in?

The student, being resourceful, exploits the fact that the bookstore has any number of copies of the *Webster's Dictionary* that he can put at his disposal. He contrives the following algorithm, which, he reasons, should allow him to determine the dictionary's thickness to arbitrary precision. The algorithm involves a sequence of operations, and proceeds as follows as illustrated in Figure 1.8.

Markings are made on the wall, from the floor up, at intervals of 6 inches using the (6-inch) ruler. The student places a copy of the *Webster's Dictionary* on the floor. The action of placing the book causes the level corresponding to the top of the stack of dictionaries (which contains just one instance at this time) to cross the lowest (6-inch) mark on the wall, which is at the floor level. The result of this experiment, denoted by v, is decreed to be 6 (corresponding to the 6-inch ticks on the wall).

A copy of *Webster's* is placed on the first, as shown in Figure 1.8(b). Since the action of adding the second copy causes the height of the stack to cross a marking on the wall, the result of this experiment is also deemed to be 6. This mode of operation continues *ad infinitum*. v at the end of every step, therefore, is 6 if the addition of a new copy causes

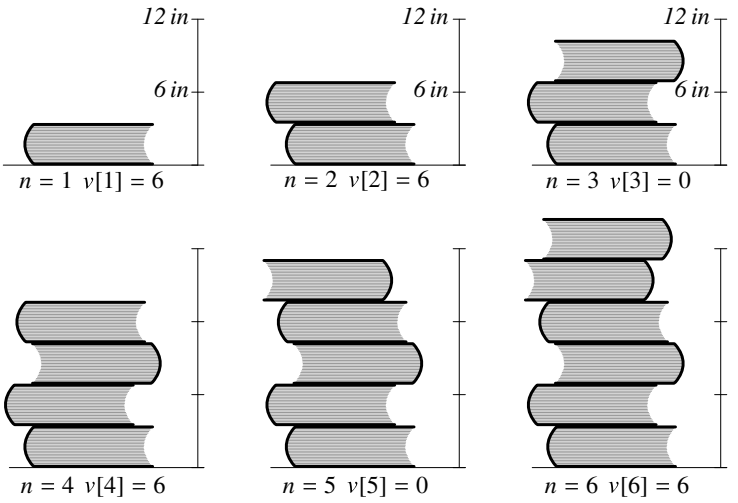


Figure 1.8 Measuring the thickness of *Webster's Dictionary* the $\Delta\Sigma$ way.

the stack to cross a new 6-inch mark, and zero otherwise. Denoting the thickness by u, the height of the stack in the nth instance is given by

$$\sum_{k=1}^{n} u = nu. \tag{1.3}$$

This is compared with the *next* 6-inch mark on the wall, whose height is given by

$$\sum_{k=1}^{n-1} v[k]. \tag{1.4}$$

Thus,

$$v[n] = \begin{cases} 6, & \sum_{k=1}^{n} u \geq \sum_{k=1}^{n-1} v[k], \\ 0, & \textit{otherwise}. \end{cases}$$

The student argues that at the end of n operations,

$$0 < \sum_{k=1}^{n} v[k] - \sum_{k=1}^{n} u < 6, \tag{1.5}$$

since the height of the stack and the mark immediately above the top of the stack can differ by at most 6 inches. This means that

$$\frac{1}{n}\sum_{k=1}^{n} v[k] - \frac{6}{n} < u < \frac{1}{n}\sum_{k=1}^{n} v[k]. \tag{1.6}$$

An estimate of u can therefore be obtained by simply averaging the sequence $v[n]$. As n approaches infinity, the average of the output sequence approaches the true height of our venerable tome, which is about 3.42 inches.

When the student's scheme is translated into the language of electrical engineering, the diagram shown in Figure 1.9 results. The input u is summed in a delay-free integrator.

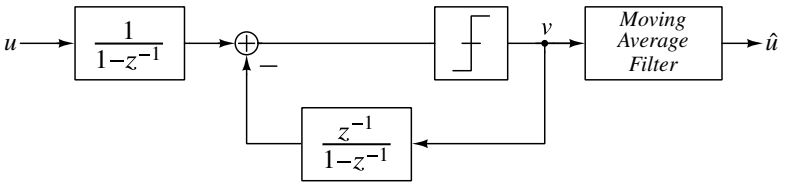


Figure 1.9 Equivalent representation of the algorithm in Figure 1.8.

The output sequence v is summed (Σ) using a delayed integrator, since the current decision depends on the sum of the previous decisions. The difference (Δ) between the two accumulated results is quantized to one of two levels (0 and 6 in our example). The resulting output sequence v is averaged (by a moving-average filter) to estimate the input u. The averaging filter acts on a digital input and is, therefore, a digital filter.

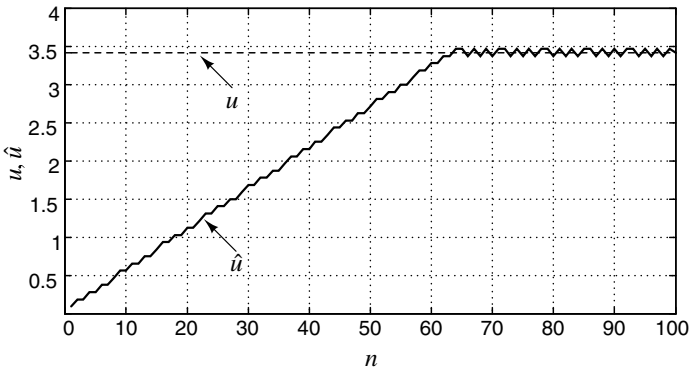


Figure 1.10 First 100 samples of the output of the moving average filter in Figure 1.11, for $u = 3.42$. A 64-tap filter (with all taps equal) is assumed.

Figure 1.10 shows the first hundred samples of $\hat{u}$ at the output of a 64-tap moving-average filter. In steady state, $\hat{u}$ happens to be within 0.05 inches of u. At first sight, it indeed seems remarkable that one can resolve to a small fraction of an inch with a scale marked only at 6 inches!

It is instructive to compare the Nyquist and $\Delta\Sigma$ ways of measurement. The former is a one-shot process, with the accuracy of measurement depending on the fineness and precision of the marks on the ruler. The latter, in contrast, is an iterative process. It involves feedback, since the outcome $v[n]$ of the nth iteration depends on the results of previous experiments. The $\Delta\Sigma$ method relies on the fact u does not change between successive iterations. This means that u is heavily *oversampled*. Moreover, $v[n]$ is not representative of u; u can only be inferred by averaging the outcomes of a large number of iterations. Measurement accuracy generally improves as n is increased. Averaging 1000 samples reduces the error to 0.006 inches.

A practical problem with the realization of Figure 1.9 is that the outputs of both integrators keep increasing with n. In our bookstore example, the pile of dictionaries in Figure 1.8 would risk hitting the ceiling due to lack of headroom. Likewise, electronic integrators have limits on their maximum allowable output. This can be circumvented by

simply moving the integrators into the loop, as shown in Figure 1.11. In the figure, $\hat{u}$ is a digital representation of u, and the system converts the continuous valued input u into a quantized output. This is achieved by embedding a coarse quantizer (which, in our example, has only two levels – 0 and 6 inches) in a negative feedback loop. The feedback loop of Figure 1.11 is called a $\Delta\Sigma$ modulator (or $\Delta\Sigma$ converter). More precisely, it represents a first-order, 2-level $\Delta\Sigma$ modulator. The integrator, whose output is quantized, is often referred to as the *loop filter*.

The discussion in this section was a (hopefully) gentle introduction to the basic idea behind $\Delta\Sigma$ modulation. A more detailed development of the first-order $\Delta\Sigma$ loop, its analysis and alternative ways of realizing the same functionality are given in Chapter 2.

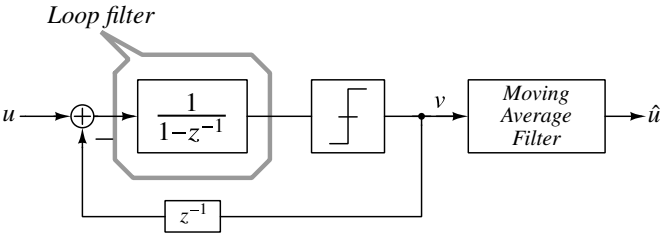


Figure 1.11 Addressing "headroom problems" of the system of Figure 1.9 by moving the integrators into the loop. Averaging v yields an estimate $\hat{u}$ of u.

The reader might wonder why the measurement must proceed in the iterative fashion shown in Figure 1.8. Why not stack 64 dictionaries, and measure the height of the stack (to the nearest 6 inch mark) and divide by 64? To understand this, we denote the error introduced by the quantizer of Fig 1.11 in the nth iteration by $e[n]$. It is easy to see that

$$v[n] = u[n] + e[n] - e[n-1]. \tag{1.7}$$

The output of the M-tap moving-average filter (with weights being equal) is given by

$$\hat{u} = \frac{1}{M}\sum_{k=r+1}^{M+r} v[k] = u + \frac{1}{M}\left(e[r+M+1] - e[r]\right). \tag{1.8}$$

It is easy to see that $\hat{u}$ is what one would obtain by stacking up M dictionaries, measuring the height of the stack to the nearest 6 inches, and dividing the result by M. From the equation above, we observe that the estimation error in $\hat{u}$ is due to the e in the first and last of the 64 (assuming $M = 64$) samples being processed by the moving-average filter. This suggests that quantization error can be reduced by weighting $v[n]$ non-uniformly – that is, by attaching more importance to the middle set of samples than those toward the end. This intuition is confirmed by filtering the output sequence of the modulator with a 64-tap moving-average filter with a triangular impulse response. From Figure 1.12, we see that the peak-to-peak excursion of the output of such a filter is much smaller than that in the case where all the samples of $v[n]$ are equally weighted. Thus, there is merit to observing the height of the stack every time an additional dictionary is added, as this enables the use of arbitrary moving-average filters. Recall that measuring the height of a stack of 64 dictionaries (to the closest 6-inch mark) and dividing by 64 is equivalent to uniformly weighting the samples of v. To summarize, there is more to choosing the post-filter that processes the modulator's output than simply averaging the output. To understand how

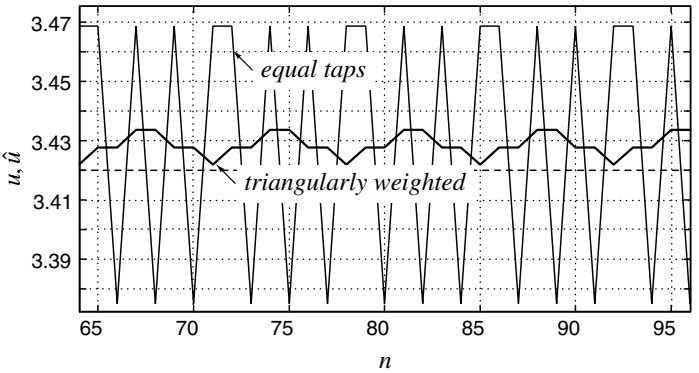


Figure 1.12 Outputs of moving-average filters with equal weights, and a triangularly weighted response.

one designs a post-filter, it is helpful to examine a $\Delta\Sigma$ modulator in the frequency domain, which we will do going forward. Before that, we wish to draw the reader's attention to the following.

The example above considered the modulator's input u to be constant. In practice, the input to be digitized has a nonzero bandwidth (which is much smaller than the sampling rate). Then, the output of the digital post-filter (which is a sequence at the sampling rate) can be downsampled, so that the output sample rate can equal the Nyquist rate corresponding to the input signal. Figure 1.13 shows the system model of an ADC employing

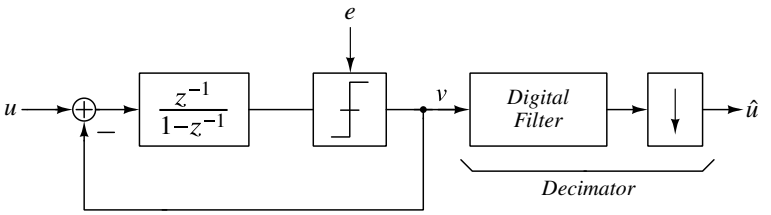


Figure 1.13 System model of an ADC with a first-order $\Delta\Sigma$ modulator.

a first-order $\Delta\Sigma$ modulator. The delay element in the feedback path of the modulator of Figure 1.11 has been pushed into the forward path. The (benign) consequence of this is to delay the input by one sample. The combination of the the digital post-filter and downsampler is called the *decimation filter* or decimator.

The output noise due to the quantization error in the $\Delta\Sigma$ modulator is $q[n] = e[n] - e[n-1]$. In the z-domain, this becomes $Q(z) = (1-z^{-1})E(z)$, and in the frequency domain, after z is replaced by $e^{j\omega}$, the power spectral density (PSD) of the output noise is found to be

$$S_q(\omega) = 4\sin^2\left(\frac{\omega}{2}\right) S_e(\omega). \tag{1.9}$$

Here, $S_e(\omega)$ is the one-sided PSD of the quantization error (noise) of the internal ADC. For "busy" (i.e., rapidly and randomly varying) input signals, one may approximate e with white noise of mean-square value $\Delta^2/12$, where Δ is the step size of the quantizer, and thus

obtain

$$S_e(\omega) = \frac{\Delta^2}{12\pi}. \tag{1.10}$$

The filtering function $(1 - z^{-1})$ is called the noise transfer function (NTF). The squared magnitude of the NTF as a function of frequency is illustrated in Figure 1.14. As the

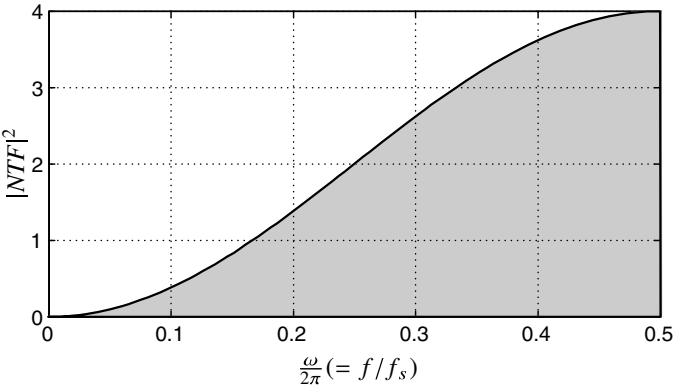


Figure 1.14 Noise-shaping function for the ΔΣ modulator shown in Figure 1.13.

figure shows, the NTF of the ΔΣ modulator is a highpass filter function. It suppresses e at frequencies around 0, but the NTF also enhances e at frequencies around $\omega = \pi$.

We introduce next the *oversampling ratio*

$$OSR = \frac{f_s}{2f_B}, \tag{1.11}$$

where f_B is the maximum signal frequency, which is the signal bandwidth. OSR defines how much faster we sample in the oversampled modulator than in a Nyquist-rate converter.

It turns out that the in-band component of quantization noise at the output of the modulator is given by

$$q_{rms}^2 = \frac{\pi^2}{3}\frac{e_{rms}^2}{OSR^3}. \tag{1.12}$$

As expected, the in-band noise decreases with increasing OSR. However, this decrease is relatively slow; doubling the OSR reduces the noise only by 9 dB, and hence it enhances the ENOB by only about 1.5 bits.

The discussion in this chapter is intended merely as an introduction, a whetting of the appetite – the topics of sampling, oversampling and the first-order ΔΣ modulator are covered in detail in Chapter 2.

1.3 Higher-Order Single-Stage Noise-Shaping Modulators

As the reader might have anticipated, a way to increase the resolution (i.e., the ENOB) of a ΔΣ modulator is to use a higher-order loop filter. By adding another integrator and feedback path to the modulator of Figure 1.13, the structure of Figure 1.15 results. Linearized

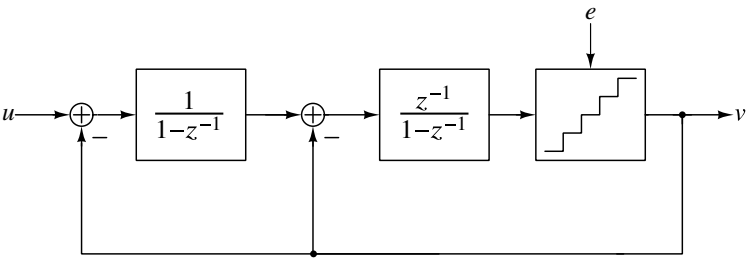


Figure 1.15 A second-order ΔΣ modulator.

analysis yields

$$V(z) = z^{-1}U(z) + (1 - z^{-1})^2 E(z). \tag{1.13}$$

This indicates that the NTF is now $(1 - z^{-1})^2$ in the z-domain, which applies a shaping function of $(2\sin(\omega/2))^4$ to the PSD of e. It follows that the in-band noise power is (to a good approximation for $OSR \gg 1$)

$$q_{rms}^2 = \frac{\pi^4 e_{rms}^2}{5\,OSR^5}. \tag{1.14}$$

Doubling OSR, therefore, results in about 2.5 bits of additional resolution. This is a much more favorable trade-off than that of the first-order modulator. A more detailed analysis of second-order ΔΣ modulators, and alternative ways of realizing them, are discussed in Chapter 3.

In principle, by adding more integrators and feedback branches to the loop, even higher-order NTFs can be obtained. For an Lth-order loop filter resulting in $NTF(z) = (1 - z^{-1})^L$, the in-band noise power is approximately

$$q_{rms}^2 = \frac{\pi^{2L} e_{rms}^2}{(2L + 1)\,OSR^{2L+1}}, \tag{1.15}$$

and the number of bits added to the resolution by doubling the OSR is given by $(L + 0.5)$.

From the discussion above, it appears as if using a ΔΣ loop with an appropriately chosen (very high-order) NTF can attain arbitrarily high SNRs, even for small *OSR*. This sounds too good to be true, and as the wise reader should suspect, something that *sounds* too good to be true *is probably* too good to be true. It turns out that for high-order loops, stability considerations, which have thus far been ignored, reduce the achievable resolution to a lower value than that given by the equations above. For high-order single-bit modulators, the difference is substantial, amounting to more than 60 dB for a fifth-order modulator. Higher-order ΔΣ modulators, their stability, trade-offs involved in their design, and various means of realizing them will be discussed in detail in Chapter 4.

1.4 Multi-Stage and Multi-Quantizer Delta-Sigma Modulators

The philosophy behind using a high-order loop to suppress in-band quantization noise is to *divide* noise by a large loop-gain, obtained by incorporating more integrators in the loop. An alternative strategy to accomplish the same objective is to *cancel* the quantization error

by measurement and subtraction. It turns out that this approach eases the stability problems associated with high-order modulators. The resulting structures are called cascade modulators, and also referred to as multi-stage or MASH (for Multi-stAge noise-SHaping) modulators. This, and other techniques based on this fundamental idea, form the subject of Chapter 5.

The basic concept behind a cascade modulator is illustrated in Figure 1.16. The output

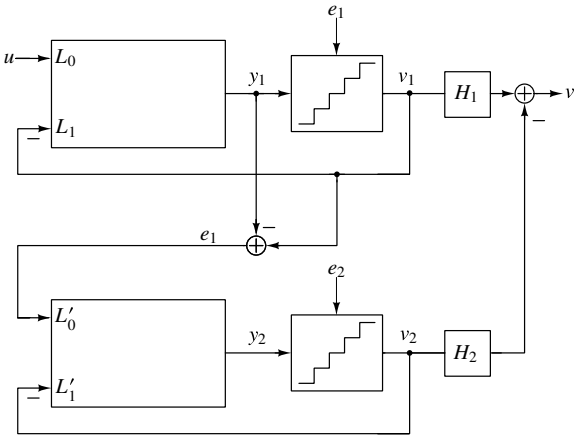


Figure 1.16 A multi-stage delta-sigma modulator.

signal of the first stage is given by

$$V_1(z) = STF_1(z)U(z) + NTF_1(z)E_1(z), \tag{1.16}$$

where STF_1 and NTF_1 are the signal and noise transfer functions, respectively, of the first stage. The second stage is added to improve the SNR beyond what NTF_1 can provide.

As shown in Figure 1.16, the quantization error e_1 of the input stage is found in analog form by subtracting the input to its internal quantizer from its output. e_1 is then fed to another $\Delta\Sigma$ loop forming the second stage of the modulator, and converted into digital form. Hence, the output signal of the second stage in the z-domain is given by

$$V_2(z) = STF_2(z)E_1(z) + NTF_2(z)E_2(z), \tag{1.17}$$

where STF_2 and NTF_2 are the signal and noise transfer functions, respectively, of the second stage. The digital filter stages H_1 and H_2 at the outputs of the two modulator loops are designed such that in the overall output v of the system, the first-stage error e_1 is canceled. By the equations above, this is achieved if the condition

$$H_1(z)NTF_1(z) = H_2(z)STF_2(z) \tag{1.18}$$

holds. The simplest (and usually most practical) choice for H_1 and H_2 that satisfies (1.18) is $H_1 = k \cdot STF_2$ and $H_2 = k \cdot NTF_1$, where k is constant chosen to give unity signal gain. Since STF_2 is often just a delay, H_1 is easily realized. The overall output is then given by

$$V(z) = k \cdot STF_1(z)STF_2(z)U(z) + k \cdot NTF_1(z)NTF_2(z)E_2(z). \tag{1.19}$$

In a typical case, both stages of the MASH modulator may contain a second-order loop, and their transfer functions may be given by $STF_1 = z^{-1}$, $STF_2 = 0.5z^{-2}$ and $NTF_1 = NTF_2 = (1 - z^{-1})^2$. Choosing $k = 2$, the output we obtain is then

$$V(z) = z^{-2}U(z) + 2(1 - z^{-1})^4 E_2(z). \tag{1.20}$$

Thus, the noise-shaping performance is essentially that of a fourth-order single-loop converter, but the stability behavior is that of a second-order one.

If the condition (1.18) is not exactly satisfied, for example, due to imperfections in the realization of the analog transfer functions, then $E_1(z)$ will appear at the output multiplied by $k \cdot [STF_2 NTF_{1a} - NTF_1 STF_{2a}]$, where the subscript a denotes the actual value of the analog transfer function. This is not surprising, since the efficacy of *any* technique based on cancellation is always degraded by mismatch. As will be shown in Chapter 5, mismatch can result in a serious deterioration of the noise performance of the converter.

1.5 Mismatch Shaping in Multi-Bit Delta-Sigma Modulators

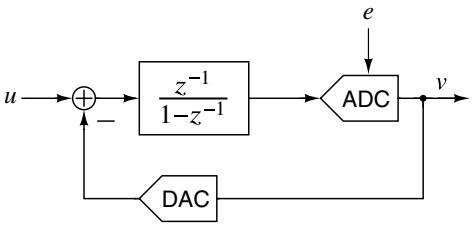


Figure 1.17 The quantizer in a ΔΣ modulator is implemented as a cascade of ADC and DAC.

A quantizer is implemented as a cascade of an ADC and DAC, as shown in Figure 1.17. The DAC appears in the feedback path of the ΔΣ modulator, and its nonlinearities result in comparable nonlinearities for the overall conversion. This occurs because the in-band part of the DAC output signal is forced by the feedback loop to follow the input signal u very accurately. Hence, if the DAC is nonlinear, its input will be distorted to give an accurate output. Since the DAC input is the output of the converter, the converter output is distorted.

It was this fact that forced early designers of ΔΣ modulators to use single-bit internal DACs in the ΔΣ loops. A single-bit DAC has the immensely important virtue of *inherent linearity*. Since the input to a one-bit DAC only takes on two values, the transfer characteristic of the DAC can be represented by two points in the input–output plane. Thus, a straight line that passes through those points models a 1-bit DAC exactly. In other words, the DAC is *exactly* described by an equation of the form $v_d = kv + offset$, where k and *offset* are constants. Since a system that obeys such a model does not introduce distortion, a 1-bit DAC is said to be *inherently linear*.

In contrast, single-bit ADCs (which are essentially comparators) have an ill-defined gain factor, as will be shown in Chapter 2. Also as Chapters 3 and 4 will show, loops containing one-bit quantizers must remain stable over a wide range of loop gains. This