Advanced AD/DA converters

ΔΣ DACs

Pietro Andreani
Dept. of Electrical and Information Technology
Lund University, Sweden

Overview

- Why ΔΣ DACs

- Architectures for ΔΣ DACs

- Interpolation filters

- Smoothing filters

Motivations

Commercially, ΔΣ DACs are as important as ΔΣ ADCs, if not more!

Motivation for noise shaping in DAC: same as in ADC: with 3V full scale and 18-bit resolution, LSB is approx. 12uV → maximum error should not be higher than ½ LSB → impossible in a conventional Nyquist-rate DAC without trimming and/or very long conversion times

---
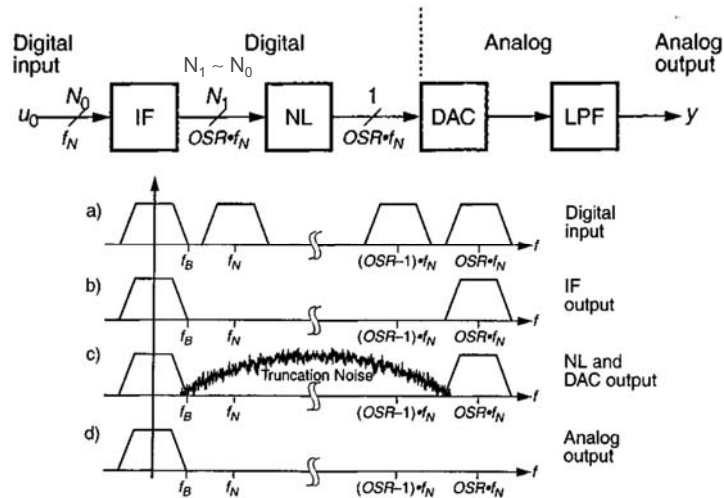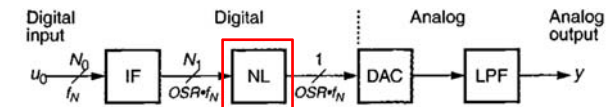
System overview

Below: typical implementation → interpolation filter IF: raises the sampling frequency to OSR·f_N (allowing subsequent noise shaping), and suppresses the spectral replicas centered at f_N, $2f_N$, ... $(OSR-1)f_N$ – this suppression reduces the out-of-band power at the input of the noise-shaping loop NL → good for the dynamic range of the NL, and analog LPF easier to make, since it needs to suppress less out-of-band noise, and also needs less linearity, since the amount of out-of-band noise that can be folded back into the baseband is smaller – suppression does not need to be very accurate, since truncation error in the NL will unavoidably introduce noise in the same frequency range

N_0 = 16-24 bits

IF: Interpolation Filter
NL: Noise-shaping Loop
DAC: Digital-Analog Converter
LPF: Lowpass Filter
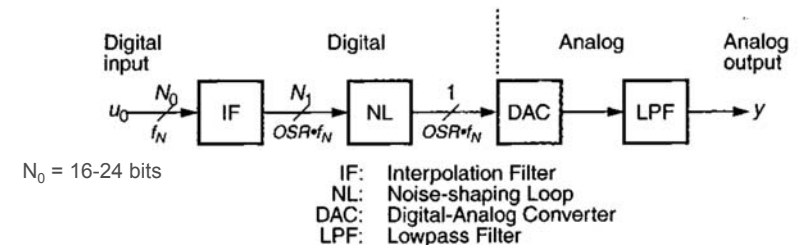
Spectra

System overview

The NL reduces the wordlength of the signal from 18b or more to 6b or less, down to 1b

1 bit → requirements on DAC linearity are greatly relaxed; however, the slew-rate requirements become very high, and the LPF filter harder to implement because of the high out-of-band noise – the overall trade-off between complexity, power consumption and chip area favors a multi-bit solution

The NL output is a faithful reproduction of the input in the baseband, plus a more or less large amount of truncation noise

The DAC is, ideally, a sample-and-hold stage → *sinc* response

LPF must smooth the signal without introducing additional distortion (quite difficult for single-bit signals)

System overview

In principle, the ΔΣ modulator is the same whether it is used for A/D or D/A conversion; however:

1) in D/A conversion, the ΔΣ modulator contains only digital signals → no internal data conversion is needed!

2) for this reason, data processing in the ΔΣ loop is highly accurate, and is not affected by "analog" imperfections → efficient implementation can be used which are impractical in analog modulators!!

All modulators discussed in A/D context (CIFB, etc) are applicable to D/A as well – of course, accumulators replace integrators, etc

Analogously, the designer has to solve the same issues on SNR vs. loop order, stability, optimum dynamic range, signal scaling, etc – the signal wordlength needed in the various locations of the loop must be carefully considered (usually the wordlength may be chosen shorter towards the output of the loop)

Hardware can be saved by choosing simple coefficients containing only a few terms, each term being a power of 2
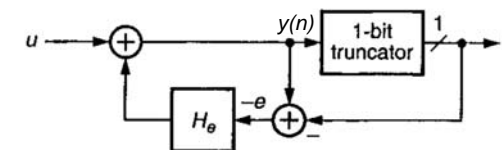
Error-feedback loop

We have already seen that this architecture is not good in analog modulators (filter in the feedback path!) – however, very good in digital ones!

$$V(z) = U(z) + [1 - H_e(z)]E(z) \rightarrow STF = 1, \; NTF = 1 - H_e$$

1st-order → $NTF = 1 - z^{-1} \rightarrow H_e = z^{-1}$ → simple delay

2nd-order → $H_e = z^{-1}(2 - z^{-1})$ → two delays, 1 shift (implementing a multiplication by 2), and two adders


Instability causes the input to the truncator to grow beyond the range of the digital circuitry, resulting either in saturation of *y(n)* at the highest /lowest value, which is usually acceptable, or in wrap-around, when *y(n)* overflows, which must be avoided

Error feedback loop

Loop implementation
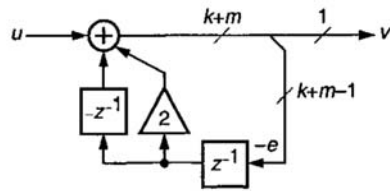
A digital limiter avoids signal overflow (with wrap-around and subsequent large errors) in higher-order loops
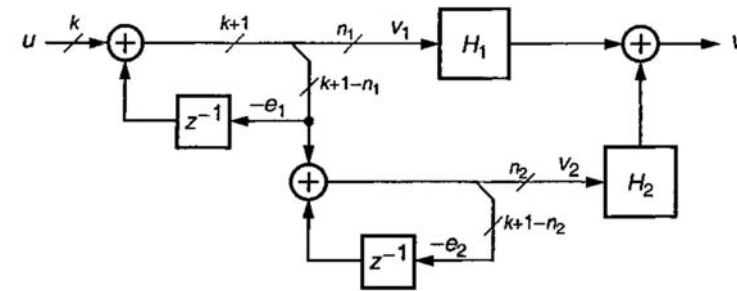
MASH ΔΣ DACs
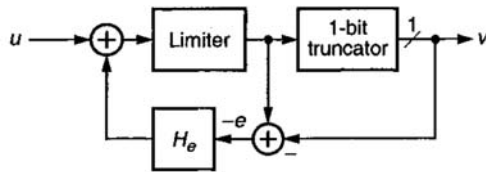
MASH ΔΣ DACs predate MASH ΔΣ ADCs!

Here: 1-1 MASH

Post-filter H_1 replicates STF_2 → simple delay → wordlength does not need to be increased; however, H_2 reproduced NTF_1 → increases the wordlength n_2 of signal v_2 → $H_1V_1 + H_2V_2$ further increases the output wordlength → multi-bit output → multi-bit DAC needed

MASH ΔΣ DACs – II

Alternative → separate DAC in each stage → less complicated DACs → analog mismatch results in leakage of 1st-stage truncation error to the output, but does not effect the linearity of the conversion, since the signal is present only in the first loop, and therefore the linearity is determined by the 1st-stage DAC

If DAC2 before H_2 → resolution of DAC2 can be reduced to 1b (as for DAC1), but an analog H_2 cannot reproduce NTF_1 in a perfect way (however, zeros of H_2 at dc can be realized very accurately in the analog domain with series capacitors in the signal path!)
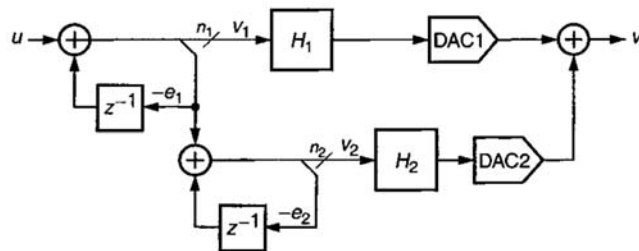
MASH ΔΣ DACs – III

Other option: split H_2 into a digital stage preceding DAC2, and an analog stage following DAC2 → the large truncation noise receives full shaping from NTF_2 and H_2, while the much smaller noise caused by DAC2 errors will be shaped only by the analog part of H_2 → this scheme is more accurate than having a fully analog H_2
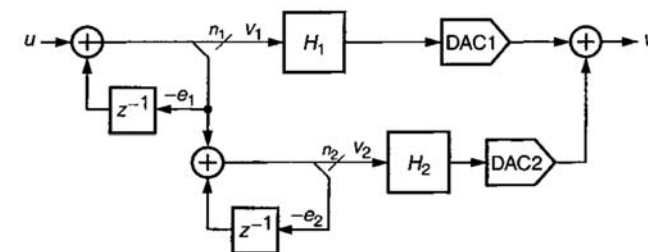
Multi-bit internal DACs

The parameters of the digital loop in ΔΣ DACs are much more accurately controlled than those in the analog loop in ΔΣ ADCs → some of the basic arguments for multi-bit quantization in ADCs (e.g. slew-rate in opamps, non-linearity, clock jitter, etc.) are not applicable in ΔΣ DACs!! Nevertheless, the stability issue is still present, and another one emerges: the desire of relaxed requirements on the analog smoothing filter (LPF) following the DAC → for a single-bit DAC, the input to the LPF is a fast-slewing two-level voltage, with most of the power contained in the large out-of-band q-noise → this noise must be filtered without appreciable distortion, otherwise the large out-of-band noise will be folded in-band; also, clock jitter will modulate the DAC output → the problems of single-bit modulations are shifted onto the analog back-end!

LPF for single-bit → e.g. cascade of 1) a high-order SC filter, 2) a SC buffer stage, and 3) a CT post-filter stage → power hungry!

Nowadays multi-bit ΔΣ DACs are more popular → same linearization techniques as with ΔΣ ADCs → treated in the next slides

Dual-truncation DAC topology

Single-bit in the D/A conversion of the signal; multi-bit in the D/A of the q-error – topology similar to the Leslie-Singh A/D, the large e_1 is cancelled at the output

The non-linearity of the M-bit DAC is shaped by H_2, which duplicates the NTF of the 1b loop → in-band power of non-linearity is suppressed
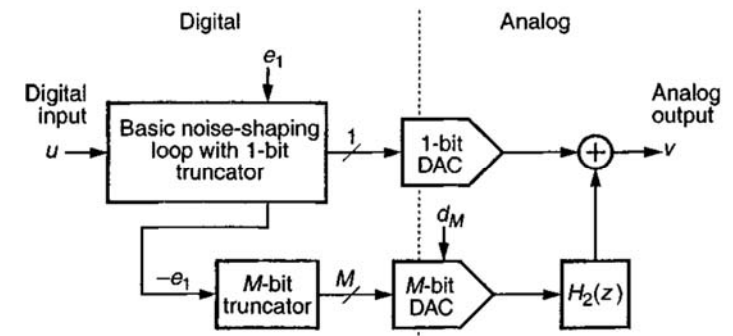
Dual-truncation MASH

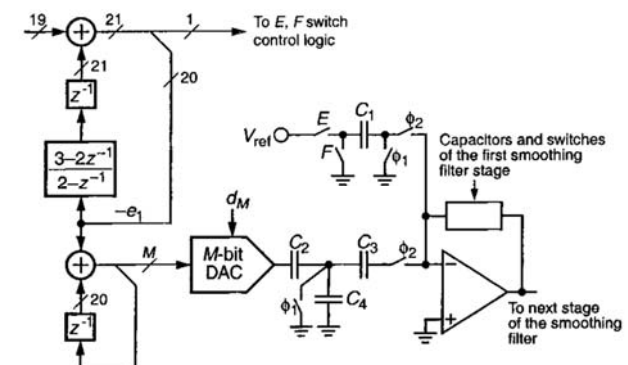Single-bit in the D/A conversion of the signal; multi-bit in the D/A of the q-error – noise shaping in both stages
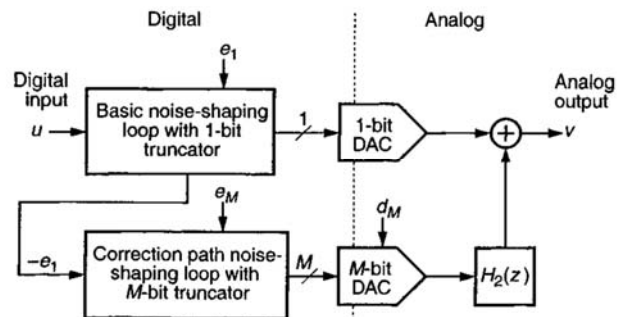
Dual-truncation 2-1 MASH DAC

C_1 implements the 1-bit DAC; C_2, C_3 and C_4 implement the analog filter H_2 Both loops are of the error-feedback kind: the first loop has a 2nd-order loop filter with a pole at z=0.5 (improved stability); the 2nd loop has a simple 1st-order filter

Single-stage dual-truncation DAC

Again, very similar to the ADC counterpart – ideally, the large 1-bit truncation error is cancelled at the output, while the use of a 1-bit DAC results in linear operation – mismatch between ideal and real H_4 and H_5 results in an imperfect cancellation of the 1-bit truncation error, but linearity is preserved
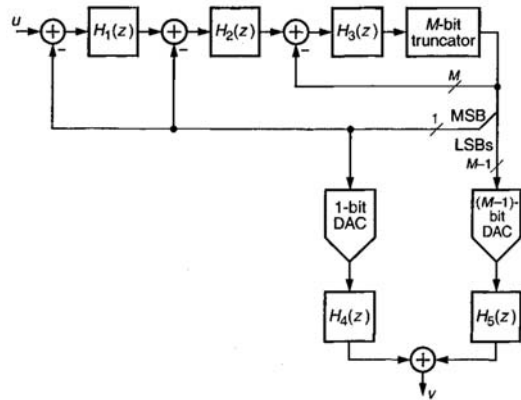
Issues in multi-bit ΔΣ DACs

Multi-bit ΔΣ ADCs: the number of bits N is generally not larger than 4, since for N=5 the internal ADC needs 32 comparators with associated circuitry (→ substantial extra power consumption and silicon area) – thus, N=2-4, and the complexity of the feedback DAC + digital circuitry for DEM is low, no need for simplification

Multi-bit ΔΣ DACs: no internal ADC is required → N > 4 can be chosen – however, DAC and error-correction circuitry grows exponentially with N → N > 4 is again impractical

To simplify analog filtering, we would like to have N > 4 → possible approaches discussed in the following, in terms of a 2nd-order 6-bit ΔΣ DAC

Segmented ΔΣ DAC

An obvious solution is to adopt a segmented architecture: 6 bits are obtained as 3 MSBs + 3 LSBs – the two segments are both thermometer-coded and scrambled – the complexity is $2 \cdot 2^3 = 16$ instead of $2^6 = 64$

The problem is that both the MSB and the LSB signals have large distortion components (truncation is a strongly non-linear operation!) that ideally cancel each other if they are recombined exactly – however, the MSB DAC will not be exactly 8 times the LSB DAC → large distortion unavoidable (independently of the scrambling of the individual DACs)
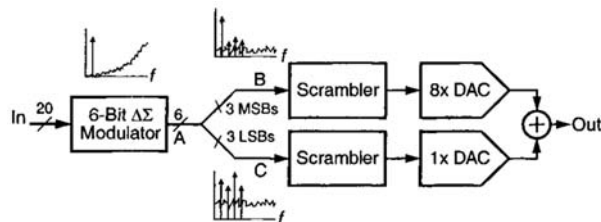
Possible solution

This problem can be overcome by cascading a 1st-order modulator to the main modulator, reducing the wordlength from 6 to 4 bits and shaping the "segmented" signals – if MOD1 has NTF=H_1 and truncation error E_1, the two segmented signals are the 4-bit output B = A + H_1E_1 of the 1st-order loop, and C = -H_1E_1, which is the (negative of) the 3-bit shaped truncation error of the same loop – the analog output is ideally B+C = A, as required

The complexity is $2^3 + 2^4 = 24$ (still much lower than $2^6 = 64$), and both B and C are shaped signals → mismatch in the recombination will result in some additional in-band noise, but not distortion – for an OSR of 128, a 1% DAC element mismatch error still allows an SNR of 110dB
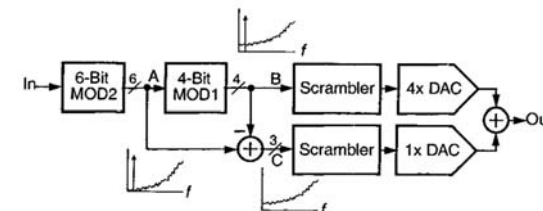
Other possible solution

Simple 1st-order noise-shaping loop (without segmentation as in the previous example): the L-bit LSB word is compressed into a B-bit word by an error-feedback loop and fed to an adder – if N=6 and L=4 we can have B=2, which, combined with the 2-bit MSB, results in a 4-bit DAC – with a high OSR, the accuracy may be sufficient
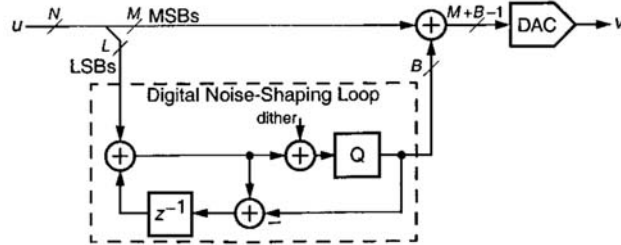
Digital correction

Power-up calibration: the RAM stores the digital equivalent of the *actual* analog outputs of the DAC for all codes – the circuit works as follows:

1) RAM and DAC have the same input → therefore, RAM and DAC must have the same output, since the RAM contains the actual outputs of the DAC

2) The feedback loop forces the in-band signal of the RAM output to follow the digital input u

3) Therefore, the DAC output must follow (in-band) the digital input u !
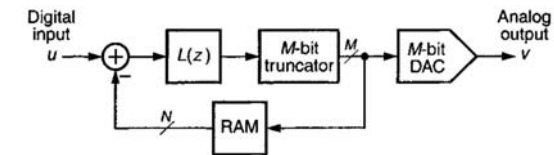
At power-up

Power-up calibration with auxiliary 1-bit ADC: a digital M-bit counter generates all input codes for the M-bit DAC (2^M codes), and each code is held at the DAC input at least 2^N clock periods, where N is the required linearity (in bits) of the DAC; the DAC output is converted by the ADC into a single-bit data stream, whose dc average is the DAC output in digital form – a digital LPF recovers this dc value, which is stored into the RAM at the address given by the counter output

Several other (background) calibration schemes are possible, as clear from a very reach literature on the subject
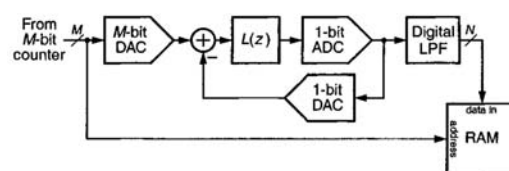
To summarize: single-bit or multi-bit ΔΣ DAC ?

Single-bit: much simpler internal DAC structure, no need for thermometer coding and digital mismatch shaping

Multi-bit: several advantages:

1) simpler noise-shaping loop, since more aggressive NTF can be used and truncation error is reduced;

2) less or no dithering, since tones are less likely to be generated (and since the amplitude of dithering is typically ½ LSB, which is (much) smaller in a multi-bit quantizer;

3) much simpler analog smoothing filter, since slewing and out-of-band noise in the DAC output are much reduced – also, the sensitivity to clock jitter is reduced, due to the reduced step size

All in all, it seems that the advantages of a multi-bit approach outweigh the disadvantages

Interpolation filter

Usually, multi stage interpolation filter – as an example, the interpolation filter (IF) in the 18-bit audio ΔΣ DAC below (single-bit internal DAC, published 1991)

spectrum (repeated beyond Nyquist)

Data at $8f_s$ + modulator clocked at $64f_s$ → implicit x8 S/H between IF and modulator
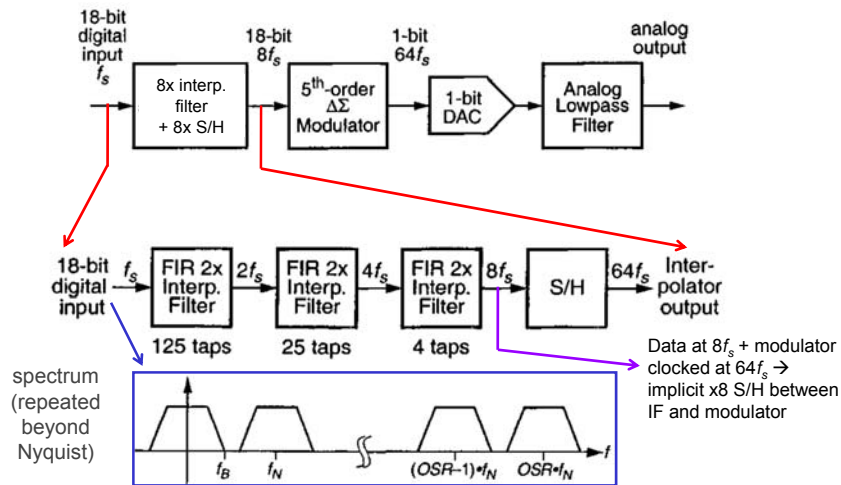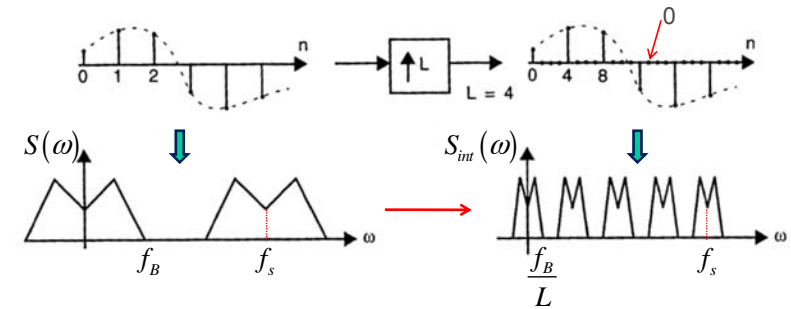

Interpolation

Upsampling by L → insert L-1 zeros between samples!

Relation between spectra:
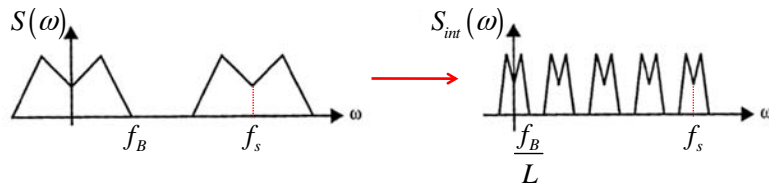
$$S(z) = x_0 + x_1 z^{-1} + x_2 z^{-2} + \ldots \qquad S_{int}(z) = x_0 + x_1 z^{-L} + x_2 z^{-2L} + \ldots$$

$$S_{int}(z) = S(z^L) \quad \rightarrow \quad S_{int}(e^{j\omega}) = S(e^{jL\omega})$$

Interpolation – II

On the other hand, $S(e^{j\omega})$ is periodic with period 2π, which means that $S_{int}(e^{j\omega}) = S(e^{jL\omega})$ contains L replicas of the spectrum of $S(e^{j\omega})$, placed at

$$\omega = 0 - \frac{2\pi}{L}, \quad \frac{2\pi}{L} - \frac{4\pi}{L}, \quad \ldots \quad \frac{2\pi(L-1)}{L} - 2\pi$$

Interpolation filter

The IF must suppress all replicas of the spectrum between baseband and OSR·f_s, where f_s is the sampling frequency of the digital input signal → this improves the dynamic range of the noise-shaping loop, and eases the selectivity and linearity requirements of the analog smoothing filter

Not all unwanted bands need to be completely cancelled, since truncation noise will be present at higher frequencies anyway

In principle, we can go from f_s to OSR·f_s in one step, and carry out all filtering at this frequency → not optimal, since all digital circuits would operate at the highest speed: higher power consumption and higher digital noise → interpolation is done in several steps, with the most filtering at low frequencies

Typically, the first stage of the IF is operated at 2·f_s, and suppresses the odd-order images of the original Nyquist-rate signal – the requirements on this filter stage are very tough: it needs a very flat passband with a very small gain variation (here: 0.001dB!), and a very sharp cutoff in order to suppress the adjacent image, which is very close!

Half-band filters

The 1st filter stage is a 125-tap half-band FIR filter → every 2nd tap weight is zero (apart the central one) → very economical implementation

Half-band filters are symmetrical around $f_s/4$: pass-band and stop-band must be symmetrical, and the ripple is the same in the two bands – this limitations are ok for the interpolation-by-2 needed here (also the 2nd and 3rd stages are half-band filters)
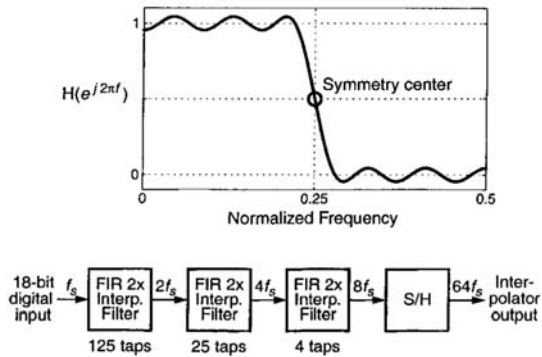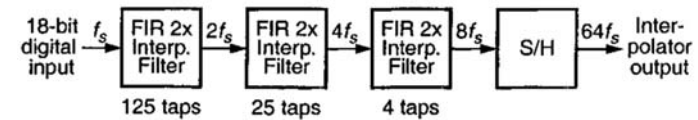
Interpolation filter

The 2nd stage can have a much less abrupt transition → 25-tap half-band

The 3rd stage has even more relaxed specifications → 4-tap half-band

Finally, the last stage is just a digital S&H stage, where one sample is held 8 times at the 8-times-higher output frequency → additional 1st-order sinc filtering with first notch at the frequency of the S&H (i.e., 8 times the original frequency)

FIR filter are very popular in audio because they can have a perfectly flat group delay (i.e. a perfectly linear in-band phase response) – IIR filter are less common, even though they can provide a higher stop-band attenuation for a given hardware complexity

IF stages and relative spectra

After ideal smoothing filter
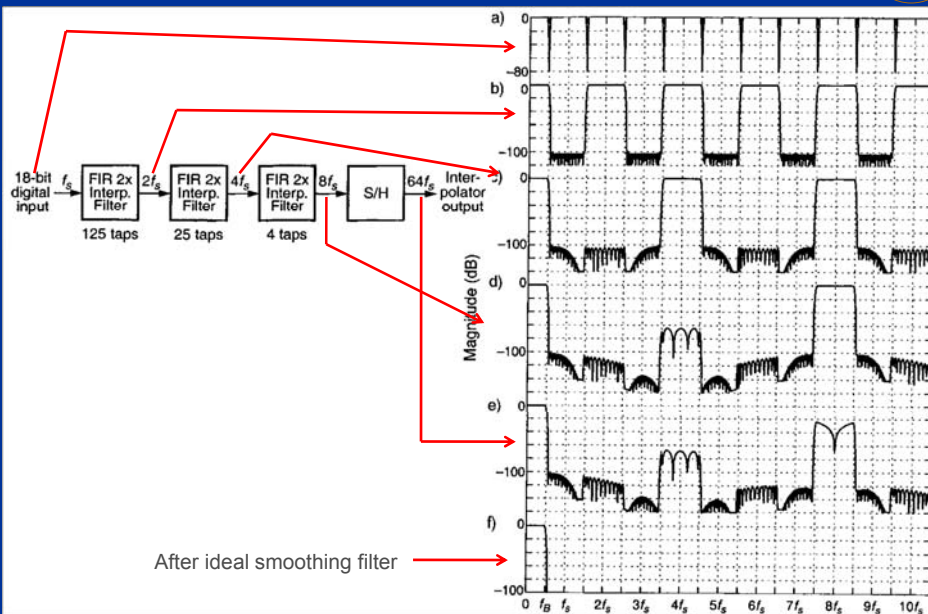
Another audio example – upsampling by 256 – spectra

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(\text{sinc2})^7$ | $(\text{sinc2})^5$ | $(\text{sinc2})^3$ | $(\text{sinc2})^2$ | sinc4
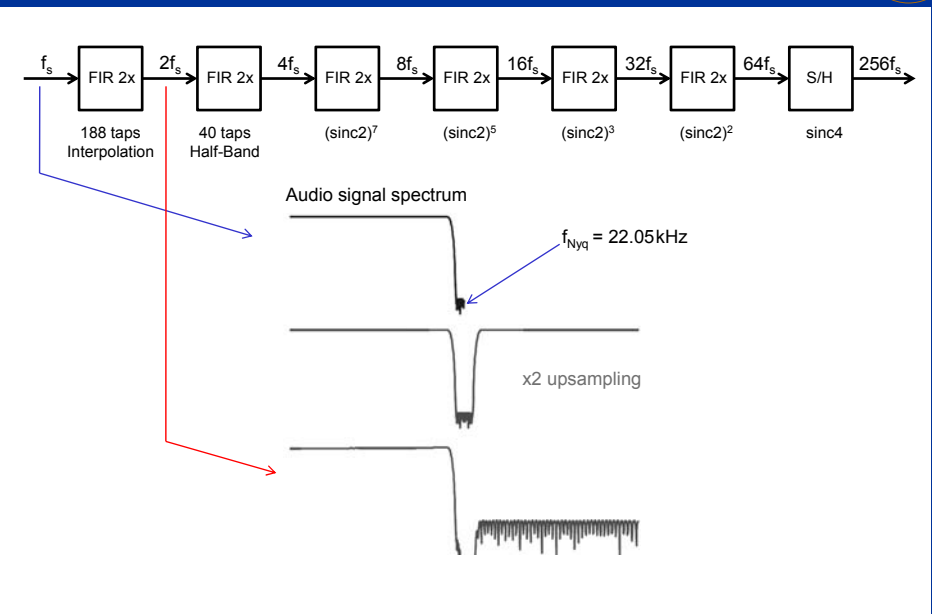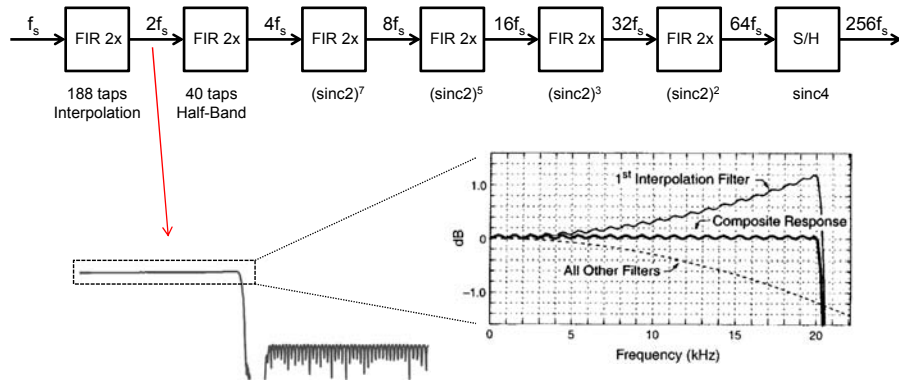
1.0 — 1st Interpolation Filter
dB — Composite Response
-1.0 — All Other Filters
0   5   10   15   20
Frequency (kHz)

---

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(\text{sinc2})^7$ | $(\text{sinc2})^5$ | $(\text{sinc2})^3$ | $(\text{sinc2})^2$ | sinc4

x2 upsampling

---

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(\text{sinc2})^7$ | $(\text{sinc2})^5$ | $(\text{sinc2})^3$ | $(\text{sinc2})^2$ | sinc4

x2 upsampling

---

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(\text{sinc2})^7$ | $(\text{sinc2})^5$ | $(\text{sinc2})^3$ | $(\text{sinc2})^2$ | sinc4

x2 upsampling

Spectra – V

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(sinc2)^7$ | $(sinc2)^5$ | $(sinc2)^3$ | $(sinc2)^2$ | sinc4

x2 upsampling

Spectra – VI

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(sinc2)^7$ | $(sinc2)^5$ | $(sinc2)^3$ | $(sinc2)^2$ | sinc4

x2 upsampling

Spectra – VII

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(sinc2)^7$ | $(sinc2)^5$ | $(sinc2)^3$ | $(sinc2)^2$ | sinc4

x4 upsampling

Spectra – signal band

f_s → FIR 2x → $2f_s$ → FIR 2x → $4f_s$ → FIR 2x → $8f_s$ → FIR 2x → $16f_s$ → FIR 2x → $32f_s$ → FIR 2x → $64f_s$ → S/H → $256f_s$

188 taps Interpolation | 40 taps Half-Band | $(sinc2)^7$ | $(sinc2)^5$ | $(sinc2)^3$ | $(sinc2)^2$ | sinc4

amplitude (dB)

Frequency

phase (rad)

Frequency

Post filtering

The constant group-delay feature must be preserved in audio → linear-phase post-filtering, or quasi-linear-phase + compensation in the digital interpolation filter

Post-filtering is particularly tough in single-bit implementations – if a conventional CT active filter is used, its opamps (or transconductors) would need an impractically high slew-rate to avoid signal-dependent slewing (= distortion)

Other linearity issue: the waveforms generated by the DAC are themselves not perfect: the periodic samples $x(nT)$ of $x(t)$ may be (almost) ideal, while $x(t)$ may contain harmonics (again, due to finite slew-rate and other limitations in the DAC)

A switched-capacitor (SC) filter as first post-filtering stage alleviates both problems

Post filtering

A sampled-data SC filter needs only $x(nT)$ as input signal, and can remove most of the high-frequency power of $x(nT)$, thus reducing the step size at the output → when the step is small enough, a CT active filter can perform the final smoothing
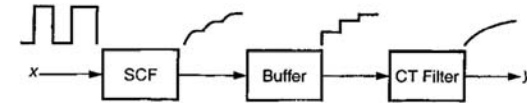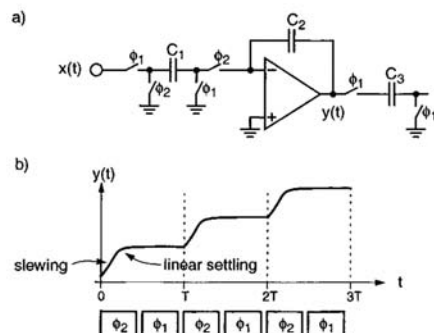
SC filter

In a SC filter a limited slew rate is not a problem, as long as the slew rate (together with the linear settling that follows slewing) is high enough to yield a correct output
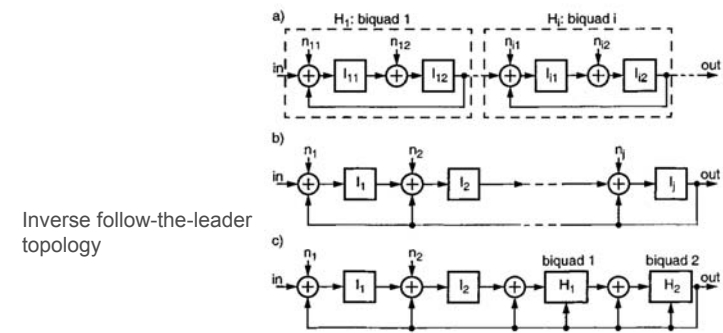
$$y(nT) = y(nT-T) + \frac{C_1}{C_2} x(nT-T)$$

at the end of the sampling period – the intermediate non-linear slewing does not matter (which would not be true in the case of a CT filter)

SC filter topology

A cascade of biquads is a very popular approach in any filter design – however, while the noise source n_{i2} in each biquad is shaped by the integrator I_{i1}, the noise source n_{i1} is not shaped, and is referred to the filter input by dividing it by $|H_1 H_2 \ldots H_1|^2$, which is ≤1 because of dynamic range scaling → the n_{i1} from all biquads is not shaped, very bad if we desire a very high SNR! → the topology in (b) is much better, and that in (c) allows the realization of finite transmission zeros as well

Inverse follow-the-leader topology

Cascade of biquads

Inverse follow-the-leader topology