Oversampling Analog to Digital Converters
21st International Conference on VLSI Design, Hyderabad

Shanthi Pavan
Nagendra Krishnapura

Department of Electrical Engineering
Indian Institute of Technology, Madras
Chennai, 600036, India

4 January 2008
Outline

- Introduction to sampling and quantization
  - Quantization noise spectral density
  - Oversampling
  - Noise shaping-ΔΣ modulation

- High order multi bit ΔΣ modulators

- Stability of ΔΣ A/D converters

- Implementation of ΔΣ A/D converters
  - Loop filter design
  - Multi bit quantizer design
  - Excess delay compensation
  - Clock jitter effects

- Mitigation of feedback DAC mismatch
  - Dynamic element matching
  - DAC calibration

- Case study
  - 15 bit continuous-time ΔΣ ADC for digital audio
Signal processing systems

Sensor(s) → Digital Processing (DSP) → Actuator(s)

Interface Electronics
(Signal Conditioning)
(A-D and D-A Conversion)

Continuous-time
Continuous-amplitude

Discrete-time
Discrete-amplitude

Continuous-time
Continuous-amplitude
Signal processing systems

- Natural world: continuous-time analog signals
- Storage and processing: discrete-time digital signals
- Data conversion circuits interface between the two
- Wide variety of precision and speed
Continuous time signals

Signals defined for all $t$

Signals can take any value in a given range
Signals defined for discrete instants $n$
Signals can take any value in a given range
Digital signals

Signals defined for discrete instants $n$

Signals can take discrete values $k V_{\text{LSB}}$
Sampling and quantization

- A segment of a continuous-time signal has an infinite number of points of infinite precision.
- Discretization of time (sampling) and amplitude (quantization) results in a finite number of points of finite precision.
- Sampling and quantization = Analog to digital conversion.
- Errors in the process?
Signals in time and frequency domains

- Continuous time signal $x_{ct}(t)$
- Frequency domain representation using its Fourier transform $X_{ct}(f)$

$$X_{ct}(f) = \int_{-\infty}^{\infty} x_{ct}(t) \exp(-j2\pi ft) dt$$

- Discrete time signal $x_d[n]$
- Frequency domain representation using its Fourier transform $X_d(\nu)$

$$X_d[\nu] = \sum_{n=-\infty}^{\infty} x_d[n] \exp(-j2\pi \nu n)$$

$X_d[\nu]$ periodic with a period of 1
Signals in time and frequency domains

Continuous–time analog signal

Fourier transform of a continuous–time signal

\[ X_{ct}(f) = \int_{-\infty}^{\infty} x_{ct}(t) \exp(-j2\pi ft) dt \]

- Signal bandwidth \( f_b \): \( |X_{ct}(f)| = 0 \) for \( f > f_b \)
Signals in time and frequency domains

Discrete time signal

Fourier transform of a discrete time signal

\[ X_d[\nu] = \sum_{n=-\infty}^{\infty} x_d[n] \exp(-j2\pi \nu n) \]

- \( X_d[\nu] \) periodic with a period of 1
- \( X_d[\nu], 0 \leq \nu \leq 0.5 \) completely defines real \( x_d[n] \)
Sampling an analog signal

Analog signal sampled to obtain a discrete-time signal

\[ x_d[n] = x_{ct}(nT_s) \]
Sampling

Sampling

![Sampled analog signal]

**Fourier transform of a sampled signal with** $f_s = 2f_b$

![Fourier transform of a sampled signal]

$$X_d[\nu] = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X_{ct}(\nu f_s - n)$$

- Copies of signal spectrum at $nf_s = n/T_s$
- Perfect reconstruction possible for $f_s \geq 2f_b$
Sampling without aliasing

Fourier transform of a sampled signal with $f_s = 2f_b$
Reconstruction from sampled signal

Fourier transform of a sampled signal with $f_s = 2f_b$
Aliasing during sampling

Fourier transform of a sampled signal with $f_s = f_b$
Sampling followed by quantization

Quantized Sampled analog signal

- 7V_{LSB}
- 6V_{LSB}
- 5V_{LSB}
- 4V_{LSB}
- 3V_{LSB}
- 2V_{LSB}
- V_{LSB}
- 0
Quantization followed by sampling

Sampled continuous–time quantized signal

0 T_s  2T_s  3T_s  4T_s  5T_s  6T_s  7T_s  8T_s  9T_s  10T_s

7V_{LSB}  6V_{LSB}  5V_{LSB}  4V_{LSB}  3V_{LSB}  2V_{LSB}  V_{LSB}  0

Shanthi Pavan  Nagendra Krishnapura  Oversampling Analog to Digital Converters
Quantization

Nonlinearity results in harmonic distortion
Harmonics folded about the sampling frequency
Spectra of quantized sinewave before and after sampling

Quantized sampled sinewave spectrum
Sampling and Quantization-Spectral density

- $f_s/f_{in} = p/q$, large $p, q$: Closely spaced tones $\sim$ noise
- $f_s/f_{in}$ irrational: Continuous spectrum
- Approximated by a constant spectral density
Quantization error model

Modelled as an additive error

\[ e = v - y \]
Quantization error in the range $[-V_{\text{LSB}}/2, V_{\text{LSB}}/2]$
- Uniform distribution
- Mean squared value of $V_{\text{LSB}}^2/12$
Sampling and Quantization-Error

- Fully correlated to the input signal
- Statistics independent of the input signal
  - Uniform distribution; mean = 0; variance = $V_{\text{LSB}}^2/12$
- White spectral density
- Modelled as uncorrelated additive white noise

\[
V_{\text{LSB}}^2/6f_s
\]

\[
S_e(f)
\]

\[
0 \quad f_s/2
\]

\[
V_{\text{LSB}}^2/6
\]

\[
0 \quad 1/2
\]

\[
V_{\text{LSB}}^2/12
\]

\[
S_e(v)
\]
2\(^N\) level quantizer with \(V_{LSB}\) spacing

- Full scale sinewave input—amplitude \((2^{N-1} V_{LSB})\)
- Mean squared signal: \((2^{N-1} V_{LSB})^2 / 2\)
- Mean squared noise: \(V_{LSB}^2 / 12\)
- \(SNR = \frac{3}{2} 2^N = 6.02 N + 1.78\) dB
Sampling and Quantization

Fourier transform of a continuous–time signal

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Fourier transform of a sampled signal with $f_s = 2f_b$.
Signal and quantization noise

\[ V_{\text{LSB}}^2 / 6f_s \]

\[ f_b \]

\[ f_s \]

\[ 2f_s \]
Fourier transform of a sampled signal with $f_s = 4f_b$
Oversampling and Quantization

Signal and quantization noise

\[ \frac{V_{LSB}^2}{6f_s} \]

\[ f_b \]

\[ f_s \]

\[ 30 \]
Oversampling

- Sample at $f_s \gg 2f_{in}$
- Oversampling ratio $OSR = f_s / 2f_{in}$
- Filter the noise using a filter of bandwidth $f_b$
- Mean squared value of error $= V_{LSB}^2 / 12 / OSR$
- Increased signal to quantization noise ratio
2^N level quantizer with $V_{LSB}$ spacing

Full scale sinewave input—amplitude = $2^{N-1} V_{LSB}$

Oversampling ratio $OSR$

Mean squared signal: $(2^{N-1} V_{LSB})^2 / 2$

Mean squared noise: $V_{LSB}^2 / 12 / OSR$

$SNR = \frac{3}{2} 2^{2N} OSR = 6.02 N + 10 \log OSR + 1.76$ dB
Oversampling and Quantization

信号和量化噪声

\[ \text{Move quantization error to filter stopband?} \]
Quantizer

Hard nonlinearity
- Modelled as additive error

\[ e = v - y \]
Linearization of soft nonlinearity

Negative feedback loop

Loop gain → ∞ ⇒ Error $u - v \to 0$
Linearization of hardnonlinearity

Quantizer output cannot equal the input

Loop gain $\to \infty \Rightarrow$ Error $|u - v| \to \infty$
Reduce error to zero only in the signal band

Negative feedback loop with dc loop gain $\rightarrow \infty$

Small loop gain at high frequencies

Error $|u - v| \rightarrow 0$ at low frequencies
First order $\Delta \Sigma$ modulator

Loop filter is an accumulator

Error $|u - v| \rightarrow 0$ at low frequencies

Differencing followed by accumulation–$\Delta \Sigma$ modulator
Noise and Signal transfer functions

![Diagram showing the noise and signal transfer functions.](image)

\[
STF = \frac{V}{U} = \frac{z^{-1}/1 - z^{-1}}{1 + z^{-1}/1 - z^{-1}}
\]

\[
= z^{-1}
\]

\[
NTF = \frac{V}{E} = \frac{1}{1 + z^{-1}/1 - z^{-1}}
\]

\[
= 1 - z^{-1}
\]
Noise transfer function

First order noise transfer function

Shanthi Pavan Nagendra Krishnapura
Oversampling Analog to Digital Converters
Output noise spectral density

\[ S_{Ve}(\nu) = S_e(\nu) |1 - \exp(-j2\pi \nu)|^2 \]
\[ = 4S_e(\nu) \sin^2(\pi \nu) \]
\[ S_{Ve}(f) = 4S_e(f) \sin^2(\pi f / f_s) \]
Output noise in the signal band

\[ v_e^2 = \int_0^{f_b} S_{v_e}(f) df \]

\[ = 4 \frac{V_{LSB}^2}{6f_s} \int_0^{f_b} \sin^2(\frac{\pi f}{f_s}) df \]

\[ \approx 4 \frac{V_{LSB}^2}{6f_s} \int_0^{f_b} (\frac{\pi f}{f_s})^2 df \]

\[ = \frac{V_{LSB}^2 \pi^2}{12} \left( \frac{2f_b}{f_s} \right)^3 \]

\[ = \frac{V_{LSB}^2 \pi^2}{12} \left( \frac{1}{OSR} \right)^3 \]
Oversampling with noise shaping

- Output noise $\propto OSR^{-3}$ with first order noise shaping
- Output noise $\propto OSR^{-1}$ with no noise shaping
- Output noise $\propto OSR^{-(2L+1)}$ with $L^{th}$ order noise shaping

Tremendous increase in signal to noise ratio with oversampling
• $2^N$ level quantizer with $V_{\text{LSB}}$ spacing
• Full scale sinewave input—amplitude = $2^{N-1} V_{\text{LSB}}$
• Oversampling ratio $OSR$
• First order noise shaping
• Mean squared signal: $(2^{N-1} V_{\text{LSB}})^2 / 2$
• Mean squared noise: $(V_{\text{LSB}}^2 / 12)(\pi^2 / 3)1 / OSR^3$
• $SNR = \frac{9}{2\pi^2} 2^{2N} OSR^3 = 6.02 N + 30 \log OSR - 3.4 \text{ dB}$
Noise transfer functions

- $1 - z^{-1}$ for a first order $\Delta\Sigma$ modulator
- Higher order differencing ($\sim (1 - z^{-1})^N$) in higher order modulators
- Crucial quantity in the design of delta sigma modulators
Analog to digital converter (Flash) in the forward path
Digital to analog converter in the feedback path
Output noise in signal band suppressed by noise shaping
Output of the analog to digital converter is the oversampled digital output $v$
Sampling preserves the signal if \( f_s \geq 2f_b \)

Quantization adds an error \( V_{\text{LSB}}^2/12 \)

Quantization error modelled as additive white noise

Oversampling and filtering reduces quantization error in the signal band

Oversampling, noise shaping, and filtering provides a much higher reduction of quantization error in the signal band
High Order NTFs

For the first order loop

\[ V(z) = X(z) + (1 - z^{-1}) E(z) \]

\[ \text{STF} = 1, \text{NTF} = 1 - z^{-1} \]

Can we do better?
High Order NTFs

\[ V(z) = X(z) + (1 - z^{-1})^2 E(z) \]

- Second Order Noise Shaping
- Can be extended to higher orders
High Order NTFs

In-band quantization noise for a first order NTF is

\[ Q \approx \frac{\Delta^2}{12\pi} \int_0^{\pi/\text{OSR}} \omega^2 d\omega = \frac{\Delta^2}{36\pi} \left( \frac{\pi}{\text{OSR}} \right)^3 \]

What if the NTF was of the form \((1 - z^{-1})^N\) ?

\[ Q \approx \frac{\Delta^2}{12\pi} \int_0^{\pi/\text{OSR}} \omega^{2N} d\omega = \frac{\Delta^2}{12(2N + 1)\pi} \left( \frac{\pi}{\text{OSR}} \right)^{2N+1} \]

Increasing order can dramatically reduce in-band quantization noise.
High Order NTFs

- Higher order $\Rightarrow$ Reduced in-band noise
- NTF gain increases at high frequencies (around $\omega \approx \pi$).
- Why can't one go on increasing order?
Stability of $\Delta\Sigma$ Modulators

$Y(z) = L_0(z)U(z) + L_1(z)V(z)$

$v$ is the quantized version of $y$. 
Quantizer is modeled as an additive noise source.

\[ V(z) = U(z)STF(z) + E(z)NTF(z) \]

\[ Y(z) = U(z)STF(z) + E(z)(NTF(z) - 1) \]

In the signal band, \( STF(z) \approx 1 \)

Quantizer Input \( \approx (\text{ADC input}) + (\text{Shaped Noise}) \)
Stability of $\Delta \Sigma$ Modulators

Quantizer input for OBG=1.5 and OBG=3.5
Gain of a Nonlinear Characteristic

Assume an infinite precision quantizer with saturation.

- What is its gain?
- Gain depends on signal.
- Black sinewave: Gain = 1
- Red sinewave: Gain < 1
Gain of a Nonlinear Characteristic

\[ \text{Gain} = \frac{E(v.y)}{E(y.y)}. \]

- Makes intuitive sense.
- \( E(v.y) \) is the average value of \( v.y \).
- \( E(v.y) \) is a measure of how much the output “resembles” the input.
Gain of a Nonlinear Characteristic

If input to the quantizer exceeds the quantizer range
  • Quantizer gain falls.
  • If quantizer gain falls, system poles can move out of the unit circle.
  • Modulator will become unstable.
  • Signal level dependent loop stability has to be expected.
Intuition about Loop Stability

- Loop becomes unstable if the quantizer saturates.
- Saturation occurs if the quantizer input exceeds the quantizer range.
- Quantizer Input = ADC Input + Shaped Noise.

Conclusions -

- The maximum ADC input must be smaller than the quantizer range. (called the Maximum Stable Amplitude (MSA)).
- More “shaped” noise → More likelihood of instability.
- More shaped noise → Lesser in-band noise.
- An aggressive NTF will have a reduced MSA.
Estimating Maximum Stable Amplitude (MSA)

- Simulation is the best way.
- Keep stepping up the input sinewave amplitude.
  - For every amplitude, compute in-band SNR.
  - Beyond the MSA, the closed loop poles move out of the unit-circle.
  - Noise shaping is lost $\Rightarrow$ In-band SNR falls.
  - Quantizer input tends to infinity.
- Time consuming.
Estimating MSA Without Sinewave Inputs

- Originally proposed by Lars Risbo.
- Put a slowly increasing ramp into the ADC.
  - Beyond the MSA, the closed loop poles move out of the unit-circle.
  - Quantizer input tends to infinity very rapidly.
  - The value of the ADC input when the quantizer input *blows up* is the MSA.
- Found (empirically) to result in an MSA close to that predicted by the sinewave method.
- Much quicker than the sinewave technique.
Estimating MSA Without Sinewave Inputs

\[ \text{Very Slow Ramp (0 to 1 over 1 second)} \]
Estimating MSA Without Sinewave Inputs

log(Quantizer Input) versus ADC Input

MSA is about 90% of the quantizer range
MSA vs OBG for a Third Order NTF

Out of Band Gain

Maximum Stable Amplitude

4-bit quantizer

3-bit quantizer

Shanthi Pavan  Nagendra Krishnapura  Oversampling Analog to Digital Converters
A Systematic NTF Design Procedure

- NTFs of the form $(1 - z^{-1})^N$ have stability problems.
- Why?
- The OBG is too high ($2^N$).
- This saturates the quantizer even for small inputs, causing instability.
- The MSA is small.
- Worse for low quantizer resolutions.
A Systematic NTF Design Procedure Solution

- Introduce poles into the NTF.

\[ NTF(z) = \frac{(1 - z^{-1})^N}{D(z^{-1})}. \]

- Recall that \( NTF(\infty) = 1. \)

\[ \Rightarrow D(z = \infty) = 1. \]
Why do poles help?

- Properly chosen poles reduce OBG of the NTF, enhancing stability.
- However, stability comes at the expense of increased in-band noise.
A Systematic NTF Design Procedure

- Commonly used pole positions: Butterworth, Chebyshev, Inv. Chebyshev etc.
- Coefficients for these approximations readily gotten from MATLAB.
- Schreier’s Delta-Sigma Toolbox is an invaluable design aid.
- One should understand what the toolbox does.
A Systematic NTF Design Procedure

- Choose the order of the NTF.
- OSR, number of levels \((n)\) and desired SNR are known.
  - Example: Order = 3, OSR = 64, \(n = 16\), SNR = 115 dB.
- Basically, the NTF is a high-pass filter transfer function.
  - Example: Choose a Butterworth Highpass.
- Choose the 3 dB corner of the high pass filter -
  - Example: \(\omega_{3dB} = \frac{\pi}{8}\).
  - For a Butterworth NTF, specifying the cutoff specifies the complete transfer function.
A Systematic NTF Design Procedure

- Get the transfer function from MATLAB
  
  \[ [b,a] = \text{butter}(3, 1/8, 'high') \]

  \[ H(z) = \frac{0.6735 - 2.0204z^{-1} + 2.0204z^{-2} - 0.6735z^{-3}}{1 - 2.2192z^{-1} + 1.7151z^{-2} - 0.4535z^{-3}} \]

  - MATLAB sets \(|H(e^{j\pi})| = 1\).

- Recall that for \(H(z)\) to be a valid NTF, \(H(\infty) = 1\).
A Systematic NTF Design Procedure

- Scale $H(z)$ by $\frac{1}{0.6735}$ to obtain $NTF(z)$.

$$NTF(z) = \frac{(1 - 3z^{-1} + 3z^{-2} - z^{-3})}{1 - 2.2192z^{-1} + 1.7151z^{-2} - 0.4535z^{-3}}$$
A Systematic NTF Design Procedure

- Find loop filter using $\frac{1}{1+L(z)} = NTF(z)$.
- Simulate the equations describing the modulator.
- Compute the peak SNR.
  - In our example, we obtain SNR=102 dB after simulation.
  - MSA = 0.85.
A Systematic NTF Design Procedure

- If SNR is not enough, repeat the entire procedure above with a higher cutoff frequency for the Butterworth high pass filter.
  - This will increase the OBG (intuition on this later).
  - The MSA will reduce.

- If SNR is too high, repeat the entire procedure above with a lower cutoff frequency for the Butterworth high pass filter.
  - This will decrease the OBG (intuition on this later).
  - The MSA will increase.
A Systematic NTF Design Procedure

- SNR obtained with 3 dB cutoff of $\frac{\pi}{8}$ is inadequate.
- So, we increase the cutoff frequency to $\frac{\pi}{4}$.
- The peak SNR is around 116 dB.
- OBG = 2.25, MSA = 0.8.
- We are done.
- This iterative process is coded into `synthesizeNTF` in Schreier’s toolbox.
A Systematic NTF Design Procedure: Remarks

- **Butterworth** is one of several candidate high pass filters.
  - All the zeros of transmission are at the origin.
- Another useful family is the inverse Chebyshev approximation.
  - Has complex zeros (on the unit circle).

![Graph showing Butterworth and Inverse Chebyshev curves](image-url)
The Sensitivity of a Feedback Loop

- $E$ is a disturbance injected into the feedback loop.
- $V(z) = X(z) \frac{L(z)}{1+L(z)} + E(z) \frac{1}{1+L(z)}$.
- If $L(z) = \infty$, $V(z) = X(z)$.
- The loop rejects $E(z)$, or the loop is *insensitive* to $E(z)$.
The Sensitivity of a Feedback Loop

$L(z)$ cannot be $\infty$ at all frequencies.

$V(z) = X(z) \frac{L(z)}{1+L(z)} + E(z) \frac{1}{1+L(z)}$.

The loop rejects $E$ at frequencies where the loop gain is high.

How effectively this is done is called the sensitivity function.

Sensitivity is $\frac{1}{1+L(e^{j\omega})}$.
The Sensitivity of a Feedback Loop

- In a $\Delta \Sigma$ loop, sensitivity is the same as the NTF.
- Recall: The first sample of the NTF impulse response is 1.
- Equivalent to $NTF(\infty) = 1$
- The NTF can be written as
  \[
  \frac{(1+a_1 z^{-1})(1+a_2 z^{-1}+a_3 z^{-2})\cdots}{(1+b_1 z^{-1})(1+b_2 z^{-1}+b_3 z^{-3})\cdots}
  \]
- Poles must be within the unit circle (for a stable loop).
- The zeroes are on the unit circle (or inside).
The Sensitivity of a Feedback Loop

It can be shown that \( \int_0^\pi \log(|1 + a_1 e^{-j\omega}|) \, d\omega = 0 \), if \(|a_1| \leq 1\).

The area above the 0 dB in the log magnitude plot is equal to the area below the 0 dB line.
The Sensitivity of a Feedback Loop

\[ \int_{0}^{\pi} \log(|1 + a_2 e^{-j\omega} + a_3 e^{-j2\omega}|) \, d\omega = 0 \]

if the roots of \( 1 + a_2 z^{-1} + a_3 z^{-2} \) lie within (or on) the unit circle.

Straightforward to derive, if one accepts the previous result.
The Sensitivity of a Feedback Loop

\[ \int_{0}^{\pi} \log |NTF(e^{j\omega})| d\omega = \]

\[ \int_{0}^{\pi} \log \left| \frac{(1 + a_1 e^{-j\omega})(1 + a_2 e^{-j\omega} + a_3 e^{-2j\omega}) \cdots}{(1 + b_1 e^{-j\omega})(1 + b_2 e^{-j\omega} + b_3 e^{-3j\omega}) \cdots} \right| = \]

\[ \int_{0}^{\pi} \log(|1 + a_1 e^{-j\omega}|) d\omega + \int_{0}^{\pi} \log(|1 + a_2 e^{-j\omega} + a_3 e^{-2j\omega}|) d\omega - \]

\[ \int_{0}^{\pi} \log(|1 + b_1 e^{-j\omega}|) d\omega - \int_{0}^{\pi} \log(|1 + b_2 e^{-j\omega} + b_3 e^{-2j\omega}|) d\omega + \cdots \]
The Sensitivity of a Feedback Loop

\[ \int_{0}^{\pi} \log |NTF(e^{j\omega})| d\omega = \]

\[ \int_{0}^{\pi} \log \left| \frac{(1 + a_1 e^{-j\omega})(1 + a_2 e^{-j\omega} + a_3 e^{-2j\omega}) \cdots}{(1 + b_1 e^{-j\omega})(1 + b_2 e^{-j\omega} + b_3 e^{-3j\omega}) \cdots} \right| = \]

\[ \int_{0}^{\pi} \log(|1 + a_1 e^{-j\omega}|) d\omega + \int_{0}^{\pi} \log(|1 + a_2 e^{-j\omega} + a_3 e^{-2j\omega}|) d\omega - \]

\[ \int_{0}^{\pi} \log(|1 + b_1 e^{-j\omega}|) d\omega - \int_{0}^{\pi} \log(|1 + b_2 e^{-j\omega} + b_3 e^{-2j\omega}|) d\omega + \cdots \]

= Zero
The Bode Sensitivity Integral

\[ \int_{0}^{\pi} \log |NTF(e^{j\omega})| d\omega = 0 \]

The Integral of the Log Magnitude of an NTF is 0
Good inband performance at the expense of poor out-of-band performance.
Complex zeros better than choosing all NTF zeros at the origin.
The Bode Sensitivity Integral

Complex zeros better than choosing all NTF zeros at the origin.
The Bode Sensitivity Integral

Higher order $\implies$ less in-band noise.

\[ 20 \log |NTF| \]
Loop Filter Architectures

- Remember: A quantizer = ADC + DAC.
- Needs ONE DAC.
- Loop filter gain goes to infinity at DC, with order 2.
- Both NTF zeros at DC ($z = 1$).
- Called CIFF (Cascade of Integrators Feed Forward)
Loop Filter Architectures

Remember: A quantizer = ADC + DAC.

Needs TWO DACs.

Loop filter gain goes to infinity at DC, with order 2.

Both NTF zeros at DC (z = 1).

Called CIFB (Cascade of Integrators Feed Back).
Loop Filter Architectures

- CIFF loop with complex zeros.
- NTF zeros are at $1 \pm j\sqrt{\gamma}$.
Loop Filter Architectures

- CIFB loop with complex zeros.
- NTF zeros are at $1 \pm j\sqrt{\gamma}$.
Loop Filter Implementation

- Traditionally done in discrete-time.
- Implemented using switched-capacitor techniques.
- Switched capacitor circuits have several advantages.
  - Exact nature of settling is irrelevant, only the settled value matters.
  - Pole-zero locations of the loop filter are set by capacitor ratios, which are extremely accurate.
  - Insensitive to clock jitter, as long as complete settling occurs.
  - Easier to simulate.
Loop Filter Implementation Switched capacitor loop filters have disadvantages too -

- Difficult to drive from external sources due to the large spike currents drawn.
- Upfront sampling: requires an anti-alias filter.
- Integrator opamps consume more power than continuous-time counterparts.
- Require large capacitors to lower $kT/C$ noise.
Continuous-time Loop Filters

$V_{in}(t)$  
$\Sigma$  
$L(s)$  
DAC  
ADC  
$V_{out}[n]$  
$V_{dac}(t)$

What is the NTF?
How does one design such a loop?
How does this compare with a discrete-time loop filter?
The input to the DAC is a digital code $a_k$ that changes every $T_s$.

The DAC output is an analog waveform.

$$\text{Output} = \sum_k a_k p(t - kT_s)$$

$p(t)$ is called the pulse-shape.

Commonly used shapes are the Non-Return to Zero (NRZ) and Return-to-Zero (RZ) pulses.
Loop Modeling

- Set input to zero.
- Replace ADC-DAC with quantization noise $e(n)$.
- DAC is modeled as a filter with impulse response $p(t)$.
Break the loop after the sampler.
Apply a discrete time impulse.
What comes back is $l[n] = p(t) \ast l(t)|_{kT_s}$.
The z-transform of $l[n]$ is the equivalent discrete time loop filter.
A First Order Example

Discrete-time equivalent impulse response of the loop filter
0, 1, 1, 1, 1 …

\[ L(z) = \frac{z^{-1}}{1-z^{-1}} \]

\[ NTF(z) = \frac{1}{1+L(z)} = 1 - z^{-1} \]

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
A Second Order Example

Say we need $NTF(z) = (1 - z^{-1})^2$.

Discrete-time impulse response through $k_1$

$$k_1(r_1(t) - r_1(t - 1)) = \{0, k_1, k_1, k_1, k_1 \cdots \}$$

Discrete-time impulse response through $k_2$

$$k_2(r_2(t) - r_2(t - 1)) = \frac{1}{2}\{0, k_2, 3k_2, 5k_2 \cdots \}$$
A Second Order Example

- Discrete-time impulse response through $k_1$
  \[ k_1(r_1(t) - r_1(t - 1)) = \{0, k_1, k_1, k_1, k_1, \ldots\} \Rightarrow \frac{k_1z^{-1}}{1 - z^{-1}}. \]

- Discrete-time impulse response through $k_2$
  \[ k_2(r_2(t) - r_2(t - 1)) = \frac{1}{2}\{0, k_2, 3k_2, 5k_2, 7k_2, \ldots\} \Rightarrow \frac{k_2z^{-1}}{(1 - z^{-1})^2} - \frac{0.5k_2z^{-1}}{1 - z^{-1}}. \]

- \[ L(z) = \frac{(k_1 + 0.5k_2)z^{-1} + (-k_1 + 0.5k_2)z^{-2}}{(1 - z^{-1})^2}. \]
A Second Order Example

\[ L(z) = \frac{(k_1 + 0.5k_2)z^{-1} + (-k_1 + 0.5k_2)z^{-2}}{(1 - z^{-1})^2}. \]

To achieve \( NTF(z) = (1 - z^{-1})^2 \), we need

\[ L(z) = \frac{2z^{-1} - z^{-2}}{(1 - z^{-1})^2}. \]

\[ \Rightarrow k_1 = 1.5, k_2 = 1. \]
Continuous-time Sigma-Delta Summary

- It is possible to “emulate” a D-T loop filter with a C-T one.
- The equivalence depends on the DAC pulse shape.
- The technique can be extended to high order NTFs -
  - From the desired $NTF(z)$, find $L(z)$
  - Convert $L(z)$ into $L(s)$ using the DAC pulse shape
  - The MATLAB command `d2c` will do it for you, for an NRZ DAC.
  - Implement $L(s)$ using any one of the loop filter topologies.
- A CT loop filter has several other advantages ... listen on.
The Anti-Aliasing Feature of CT $\Delta\Sigma$ Modulators

\[ V_{\text{in}}(t) \xrightarrow{\Sigma} L(s) \xrightarrow{\text{DAC}} V_{\text{out}}[n] \]

- Move $L(s)$ outside the loop
The Anti-Aliasing Feature of CT $\Delta \Sigma$ Modulators

$V_{in}(t) \rightarrow L(s) \rightarrow \Sigma \rightarrow ADC \rightarrow V_{out}[n]$

$V_{in}(t) \rightarrow L(s) \rightarrow \Sigma \rightarrow ADC \rightarrow V_{out}[n]$

$\bullet$ Move the sampler outside the loop
Replace the cascade of the DAC and $L(s)$ by the equivalent discrete-time filter $L(z)$. 
The Anti-Aliasing Feature of CT $\Delta \Sigma$ Modulators

\[ V_{in}(t) \xrightarrow{L(s)} \Sigma \xrightarrow{L(z)} V_{out}[n] \]

\[ e[n] \]

\[ NTF(z) = \frac{1}{1 + L(z)} \]

105
The Anti-Aliasing Feature of CT $\Delta\Sigma$ Modulators

Consider a tone at frequency $\Delta f$ in the signal band. Response to frequency $\Delta f$ is $L(\Delta f)NTF(\Delta f)$.

In a general ADC, a tone $(\Delta f + f_s)$ can alias as $\Delta f$.

What about a CTDSM?

Response to frequency $(\Delta f + f_s)$ is $L(\Delta f + f_s)NTF(\Delta f)$
The Anti-Aliasing Feature of CT $\Delta \Sigma$ Modulators

![Diagram showing alias rejection in CT $\Delta \Sigma$ modulators](image)

- Alias rejection is $|\frac{L(\Delta f)}{L(\Delta f + f_s)}|$
- Implicit anti-aliasing without an explicit filter!
- Valuable feature of CT Delta-Sigma modulators.
Effect of Time-Constant Variations in the Loop Filter

- On-chip RC’s vary with process and temperature.
- On an integrated circuit, ratios of like elements are tightly controlled.
- We need to only worry only about quantities with “dimensions”.
- What happens due to absolute variation of RC time constants?
Effect of RC Variations: Intuitive explanation

If all RC time-constants decrease

- Loop filter bandwidth increases.
- In-band loop gain increases.
- Lower in-band quantization noise - better in-band NTF.
- NTF must be worse out-of-band - higher OBG.
Effect of RC Variations: Intuitive explanation

If all RC time-constants decrease

- Higher OBG for the NTF.
- Reduced maximum stable amplitude.
- Closer to instability.
Effect of RC Variations: Intuitive explanation

If all RC time-constants increase

- Loop filter bandwidth decreases.
- In-band loop gain decreases.
- Higher in-band quantization noise - poorer in-band NTF.
- NTF must be better out-of-band - lower OBG.
Effect of RC Variations: Intuitive explanation

If all RC time-constants increase

- Lower OBG for the NTF.
- Increased maximum stable amplitude.
- “More” stable.
Effect of RC Variations on the NTF

Nominal NTF: Maximally flat with an OBG=3

|\text{NTF} (e^{j\omega})| \text{(dB)}
\begin{align*}
k_p &= 0.7 \\
k_p &= 1 \\
k_p &= 1.3
\end{align*}
Effect of RC Variations: Time Domain Intuition

Nominal NTF: Maximally flat with an OBG=3
Effect of RC Variations: Time Domain Intuition

Nominal NTF: Maximally flat with an OBG=3

Nominal

SLOW LOOP

Shanthi Pavan, Nagendra Krishnapura

Oversampling Analog to Digital Converters
Excess Delay in CT $\Delta\Sigma$ Modulators

Why is there excess loop delay?

- Quantizer needs time to make a decision.
- Finite operational amplifier gain-bandwidth product.
- DEM logic delay in multibit converters.
Excess Delay in CT $\Delta \Sigma$ Modulators

A First Order Example

$\text{Vin} \rightarrow \frac{1}{s} \rightarrow \text{T}_s = 1 \rightarrow \text{Vout}$

- Loop filter is an integrator.
- An NRZ DAC is used.
- Sampling Rate = 1 Hz
Excess Delay in CT $\Delta \Sigma$ Modulators

Discrete-time equivalent impulse response of the loop filter $0, 1, 1, 1, 1 \ldots$

$L(z) = \frac{z^{-1}}{1-z^{-1}}$

$NTF(z) = \frac{L(z)}{1+L(z)} = 1 - z^{-1}$
Excess Delay in CT $\Delta \Sigma$ Modulators

In practice, the quantizer needs time to make a decision.
Equivalent to a delay $t_d$ in the loop.
What happens to the NTF of the loop?
Excess Delay in CT \( \Delta \Sigma \) Modulators

- Discrete-time equivalent impulse response of the loop filter
  \( \{0, 1 - t_d, 1, 1, 1 \ldots\} = \{0, 1, 1, 1, 1 \ldots\} + \{0, -t_d, 0, 0, 0 \ldots\} \)

- \( L(z) = \frac{z^{-1}}{1-z^{-1}} - t_d z^{-1} \)

- \( NTF(z) = \frac{L(z)}{1+L(z)} = \frac{1-z^{-1}}{1-t_d z^{-1}+t_d z^{-2}} \)
The order of the system is increased.

Becomes unstable for $t_d = 1$

Not surprising - a delay in a feedback loop is always problematic.

Aggressive NTF designs are more sensitive to excess delay.
Fix for Excess Delay: Basic Idea

- Impulse response of the loop filter with delay
  \[ \{0, t_d, 1, 1, 1 \cdots \} = \{0, 1, 1, 1, 1 \cdots \} + \{0, -t_d, 0, 0, 0 \cdots \} \]

- Add a path with discrete-time response \( \{0, t_d, 0, 0, 0 \cdots \} \) to the loop filter.
Fix for Excess Delay: Basic Idea

- Implementation of feedforward path in the loop.
Fix for Excess Delay: Basic Idea

Vin \rightarrow + \frac{1}{s} \rightarrow + \rightarrow t_d \rightarrow Vout

\begin{align*}
e(n) & \quad T_s = 1 \\
\end{align*}

- Equivalent implementation of loop filter feedforward.
Fix for Excess Delay : Basic Idea

- Eliminate path from the input (small compared to the integrator output).
- Excess delay can be compensated by adding a direct path around the quantizer.
Excess Delay Compensation: Summary

- Direct path around the quantizer.
- Modification of $H(s)$ (coefficient tuning).
- General approach valid even for high order modulators.
- Determining coefficients and $k$ best done numerically.
Clock Jitter in Discrete-time $\Delta\Sigma$ ADCs

Jittery Sampling

$V_{in}(t) \xleftrightarrow{\text{Sampling}} V_{in}[n] \xrightarrow{\Sigma} \L(z) \xrightarrow{\text{ADC}} V_{out}[n]$

- The input is sampled outside the modulator
Clock Jitter in Discrete-time $\Delta \Sigma$ ADCs

- Treat the input as a sinusoid with maximum amplitude $A$.
- Error due to jitter at the sampling instant is $\Delta t \frac{dA \sin(2\pi f_{in}t)}{dt}$.
- Assume white clock jitter with RMS value $\sigma_j$.
- RMS value of noise due to jitter in the signal bandwidth is $\sigma_j \sqrt{2A\pi f_{in}/OSR}$.
Clock Jitter in Continuous-time $\Delta \Sigma$ ADCs

The input is sampled inside the modulator.
The Ideal Sampler/Quantizer

- Input is sampled in the ADC.
- ADC output code is sampled by the DAC.
The Ideal Sampler/Quantizer

- DAC output analog waveform - fed back into the loop filter.
- No delay in the quantizer, no clock jitter.
- ADC output code is the modulator output.
ADC needs a finite time for conversion.

DAC is clocked $t_{\text{del}}$ later.

The clock is jittery.
Effect of ADC Sampling Jitter

- Modelled as an error preceding the ADC.
- Noise shaped by the loop.
Effect of DAC Reconstruction Jitter

- Modelled as an error following the DAC.
- Equivalent to an error at the modulator input.
- Degrades performance.
Types of DACs: NRZ versus RZ

- DAC INPUT CODE
- NRZ DAC OUTPUT
- RZ DAC OUTPUT

Shanthi Pavan Nagendra Krishnapura
Oversampling Analog to Digital Converters
Modeling Clock Jitter in NRZ DACs

JITTERY DAC OUTPUT

\[ y(n) - y(n-1) \Delta t_n \]

\[ y(n+1) - y(n) \Delta t_{n+1} \]

IDEAL OUTPUT

ERROR

\[ y(n) \]

\[ y(n-1) \]

\[ y(n+1) \]
Modeling Clock Jitter in RZ DACs

\[
y(n) = y(n-1) + \Delta t_{n-1} \cdot 2y(n-1) + \Delta t_{n+1} \cdot 2y(n+1)
\]

JITTERY DAC OUTPUT

IDEAL OUTPUT

ERROR

\[2y(n+1)\Delta t_{n+1}\]

\[2y(n-1)\Delta t_{n-1}\]

\[2y(n-1)\Delta t_{n-1/2}\]

\[2y(n+1)\Delta t_{n+1}\]
Clock Jitter in NRZ versus RZ DACs

- Error depends on the height & number of transitions in the DAC output waveform.
- NRZ DACs have a transition height $y(n) - y(n - 1)$, one transition every $T_s$.
- RZ DACs have a transition height $2y(n)$, two transitions every $T_s$.
- RZ DACs are MUCH more sensitive to clock jitter!
Clock Jitter in Modulators with NRZ DACs

\[ y(n) \]

\[ \Delta T_n \]

\[ \Delta T_{n+2} \]

\[ y(n+1) \]

\[ y(n) \]
Effect of Jitter on SNR

\[ e_j(n) = [y(n) - y(n - 1)] \frac{\Delta t(n)}{T} \]

\[ \sigma_{e_j}^2 = \sigma_{dy}^2 \frac{\sigma_{\Delta t}^2}{T^2} \]

\[ y(n) = v_{in}(n) + e_q(n) * h(n) \]

- \( v_{in} \) is the input.
- \( e_q \) is the quantization noise sequence.
- \( h(n) \) is the impulse response corresponding to the NTF.

\[ y(n) - y(n - 1) = v_{in}(n) - v_{in}(n - 1) + (e_q(n) - e_q(n - 1)) * h(n) \]

Due to oversampling, \( v_{in}(n) \approx v_{in}(n - 1) \)
\[ y(n) - y(n-1) \approx (e_q(n) - e_q(n-1)) \ast h(n) \]

\( e_q(n) \) is a white sequence with mean square value \( \sigma_{lsb}^2 \).

\[
\sigma_{dy}^2 \approx \sigma_{lsb}^2 \frac{\pi}{\pi} \int_0^\pi |(1 - e^{-j\omega}) NTF(e^{j\omega})|^2 d\omega
\]

The in-band noise due to jitter (\( J \)) is

\[
J \approx \frac{\sigma_{\Delta T_s}^2}{T^2} \frac{\sigma_{lsb}^2}{\pi OSR} \int_0^\pi |(1 - e^{-j\omega}) NTF(e^{j\omega})|^2 d\omega
\]
Effect of Jitter on SNR

\[ J = \frac{\sigma^2_{\Delta T_s}}{T^2} \cdot \frac{\sigma^2_{lsb}}{\pi OSR} \int_0^{\pi} (1 - e^{-j\omega}) NTF(e^{j\omega})^2 d\omega \]  \hspace{1cm} (1)

- **Observation**: The NTF at high frequencies (close to \( \omega = \pi \)) contributes the most to \( J \).
- \( \Rightarrow \) NTFs with high OBG result in more jitter noise.
- Smaller LSB, less jitter noise \( \rightarrow \) multibit modulator less sensitive to jitter.
Example Calculation

- Audio modulator, 24 kHz bandwidth.
- OSR = 64 \( (f_s = 3.072 \text{ MHz}) \), 4-bit quantizer.
- Quantizer input range is 2 V.
- LSB size is 2/16 \( \rightarrow \sigma_{lsb}^2 = \frac{(2/16)^2}{12} \)
- Assume 100 ps RMS jitter.
- J = \((1.28 \mu V)^2\).
- Maximum Signal Amplitude is 0.83 V peak.
- Signal to Jitter Noise Ratio is \(20 \log\left(\frac{0.83/\sqrt{2}}{1.28 \mu V}\right) = 113 \text{ dB}\)
- Conclusion: 100 ps RMS Jitter is not an issue for 15 bit resolution.
Feedback DAC nonlinearity
Typically 4 bits (16 levels) or less in the quantizer
quantizer output $v = d_{2.0} \text{[binary]} = b_{1.7} \text{[thermometer]}$

$I_{DAC} = kI_{LSB}, \ k=\{0,1,\ldots,7\}$

- Flash quantizer gives a thermometer coded output
- Thermometer coded DAC: high accuracy and small loop delay
Switched capacitor (discrete-time) $\Delta \Sigma$ modulator

- Array of $M$ capacitors for $M + 1$ levels
- Flash quantizer output $v$
- $v$ capacitors charged to $V_{ref}$ and $M - v$ to zero volts

$b_{1-8} = \text{thermometer coded } v$
Continuous-time $\Delta \Sigma$ modulator

- Array of $M$ resistors for $M + 1$ levels
- Flash quantizer output $v$
- $v$ resistors connected to $V_{ref}$ and $M - v$ to ground
Continuous-time $\Delta\Sigma$ modulator

- $b_1I_{\text{LSB}}$
- $b_2I_{\text{LSB}}$
- $\ldots$
- $b_8I_{\text{LSB}}$

$b_{1:8} = \text{thermometer coded } v$

- Array of $M$ current sources for $M + 1$ levels
- Flash quantizer output $v$
- $v$ current sources turned on and $M - v$ turned off
Multi bit versus single bit quantizer

- Multi bit: smaller LSB ⇒ lower quantization noise
- Single bit: larger LSB ⇒ higher quantization noise
Multi bit versus single bit quantizer

- Multi bit quantizer
  - Clearly defined gain
  - Conforms to prediction using linear models

- Single bit quantizer
  - Signal dependent quantizer gain
  - Deviates from prediction using linear models

straight line fit

which one?
Multi bit versus single bit quantizer

\[ I_k = I_{LSB} + \Delta I_k \]

\[ I_{DAC} = I_{LSB} + \sum_k \Delta I_k \]

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Multi bit quantizer
- Characteristics not linear due to mismatch

Single bit quantizer
- Characteristics always linear
Effect of DAC nonlinearity

- DAC output equals the input $u$
- $v$ related to the input $u$ by inverse nonlinearity of the DAC
Nonlinear DAC driven by an ideal $\Delta\Sigma$ modulator and its output $w$ analyzed.
Multi bit feedback DAC nonlinearity

\[ I_k = I_{\text{LSB}} + \Delta I_k \]

\[ b_1 \cdot I_1 \quad b_2 \cdot I_2 \quad \cdots \quad b_8 \cdot I_8 \]

I\(_{\text{DAC}}\)

full scale

quantizer output \( v \)

INL

quantizer output \( v \)

156

Shanthi Pavan  Nagendra Krishnapura  Oversampling Analog to Digital Converters
Multi bit feedback DAC nonlinearity

- $I_{out}[0] = 0$
- $I_{out}[8] = \sum_{n=1}^{8} I_n$
- $I_{LSB} = 1/8 \sum_{n=1}^{8} I_n$
- DNL $\Delta I_k = I_k - I_{LSB}$
- INL $I_{ek} = \sum_{n=1}^{k} I_n - nI_{LSB} = \sum_{n=1}^{k} \Delta I_k$
Effects of DAC nonlinearity

\[ \sigma_I / I_{\text{LSB}} = 0.001 \ (0.1\%) \]

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Effects of DAC nonlinearity

\[ \sigma_I/I_{\text{LSB}} = 0.001 \ (0.1\%) \]

![Graph showing effects of DAC nonlinearity]

Shanthi Pavan  Nagendra Krishnapura  Oversampling Analog to Digital Converters
Effects of DAC nonlinearity

- Distortion
- Increased in band quantization noise
Reducing DAC nonlinearity

- Reduce relative mismatch of DAC elements
- $\sigma_I/I_{LSB}, \sigma_C/C, \sigma_R/R \propto 1/\sqrt{WL}$
- $100 \times$ area increase to reduce relative mismatch by $10 \times$
- Sizing alone cannot help
Representing $v$ using a thermometer DAC

- $v$ current sources must be on—multiple possibilities
- $M!/M!(M - v)!$ combinations can represent $v$
- Only one possibility for $v = 0$ (all off) and $v = 8$ (all on)
Different combinations of unit cells for a given input

- \( v = 1 \) can be represented by turning on any one of \( I_{1-8} \)
- Average of all possibilities

\[
\frac{1}{8} \sum_{n=1}^{8} I_n = I_{LSB}
\]

is the ideal output!

- For all \( v \), averaging all possible combinations produces the ideal output
- Use different combinations to represent a given code
Different combinations of unit cells for a given input

\[ \sigma = 0.3 \text{ LSB} \]
Randomization

b₁-8: Thermometer coded

b₂-8: Control signals to DAC unit elements

cycle 1

cycle 2

cycle 3

cycle 4

Fixed connections

Randomized connections
Randomization

- $M \times M$ switching matrix
- In each cycle, randomly choose a set of connections
- Converts distortion to white noise
- $M!$ possible connections in the switch matrix ($9! = 362880$)—use a smaller subset
- Switch matrix introduces delay in the loop
Randomization-Butterfly scrambler

Each stage flips across 1, 2, or 4 positions
7 switches instead of 64
Only 128 combinations used—but good enough in practice
Randomization-results

$$\sigma_{I/LSB} = 0.001 \ (0.1\%)$$

---

**Ideal output**
- Blue line

**with DAC error**
- Green line

**with randomization**
- Red line

Shanthi Pavan Nagendra Krishnapura
Oversampling Analog to Digital Converters
ΔΣ modulator with randomization

- Extra delay in the loop
Randomization-summary

- Distortion components converted to noise
- Increased noise floor
- Additional loop delay
Data weighted averaging

Cycle through all the current sources as rapidly as possible
DAC nonlinearity

DAC output $I_{DAC}$ vs quantizer output $v$

INL

DNL
Data weighted averaging—dc input

- DAC output
- Error
- Pattern repeats after 8 cycles

Shanthi Pavan Nagendra Krishnapura
Oversampling Analog to Digital Converters
Data weighted averaging—dc input

- Accumulated error is zero after a small number of cycles
- Pattern repeats every $M$ cycles for an $M + 1$ level DAC
- Tones at $f_s/M$ and its harmonics for $\nu = 1$
Data weighted averaging—arbitrary inputs

\[ v \xrightarrow{\text{rotator}} D/A \]

\[ v \xrightarrow{\frac{1}{1-Z^{-1}}} D/A \xrightarrow{1-Z^{-1}} \]
Data weighted averaging—arbitrary inputs

\[ I_k = I_{\text{LSB}} + \Delta I_k \]

D/A input

INL

D/A input

176
Data weighted averaging—arbitrary inputs

accumulated $v$
1 3 5 8 11

quantizer output $v$
1 2 2 3 3

difference of successive outputs
177

Shanthi Pavan Nagendra Krishnapura Oversampling Analog to Digital Converters
Data weighted averaging—mismatch shaping

\[ \frac{1}{1-z^{-1}} \]

INL\( (v') \)

\[ \infty \]

D/A output error bounded by \( INL_{\text{max}} \)

Finite power at all frequencies

\[ 1 - z^{-1} \] at the output provides first order shaping

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Data weighted averaging—implementation

\[ b_{1-8} = \text{thermometer coded } v \]

- \( b_{1-8} \)
- Thermometer to binary converter
- Accumulator
- MUX
- M input barrel shifter driven by accumulated ADC output
- Loop delays from thermometer-binary converter, accumulator, barrel shifter

\{s_0,s_1,s_2\}  0: blue path
1: red path

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Data weighted averaging—results

$\sigma = 0.001 \ (0.1\%)$

Graph showing data weighted averaging results with $\sigma = 0.001 \ (0.1\%)$. The blue line represents the ideal case, the green line represents the case with no DWA, and the red line represents the case with DWA. The x-axis represents multiples of the base frequency ($f_b$, $2f_b$, $3f_b$, $18Qf_b$). The y-axis shows a logarithmic scale ranging from $-120$ to $0$.
ΔΣ modulator with data weighted averaging

Extra delay in the loop
- Provides first order mismatch shaping
- Potential for tones at $\approx f_s/M$ with an $M + 1$ level quantizer
- For low OSR, tones can be close to the signal band
- Additional loop delay
Individual level averaging

- Cycle through all current sources for each input code
- Separate pointer for each input code
- Lesser potential for tones than DWA
- More noise than DWA
Data weighted averaging—variants

Double Index Averaging

1 2 2 3 3 0 4 7
3 5 3 2

Bidirectional Data Weighted Averaging

1 2 2 3 3 0 4 7
3 5 3 2

184
Data weighted averaging—variants

- Bidirectional DWA: Opposite directions in each cycle
- Double index averaging: Separate pointers for $v > M/2$ and $v \leq M/2$
- DWA with randomization: Randomize the shifts once in every few cycles to break up tones
Mismatch shaped by the transfer function $H_{mismatch}$

Deviati0n from exact shaping due to the constraint $|sv| = |v|$

Complex hardware
Dynamic element matching: tradeoffs

- Mismatch error reduction
  - High order noise shaping (highest)
  - DWA
  - ILA
  - Randomization (lowest)
- Potential for tones
  - Randomization (lowest)
  - High order noise shaping
  - ILA
  - DWA (highest)
- Complexity
  - High order noise shaping (highest)
  - ILA, Randomization
  - DWA (lowest)
- Excess loop delay
  - High order noise shaping (highest)
  - ILA
  - DWA
  - Randomization (lowest)
Dynamic element matching: summary

- Data weighted averaging
  - Best compromise between complexity and performance
  - Works very well with high OSR
  - Potential for tones at low OSR
- ILA, other DWA variants
  - More complex, less potential for tones
- Randomization
  - Can also be used for DACs without noise shaping
- Measure DAC characteristics
- Duplicate its characteristics in the digital path
- \( v' = v + \epsilon; \ \epsilon \ll v; \) Lot more bits in \( v' \) than \( v \)
Calibration

- Store only the error to reduce register width
- Noise shaped quantization (digital $\Delta \Sigma$ modulator) to reduce decimator input width
Calibrate all current sources against a master source

Use $M + 1$ current sources and calibrate one at a time
Calibration: summary

- No additional components in the loop ⇒ no excess delay
- Measuring DAC characteristics inline is challenging
- Additional digital or analog complexity


CASE STUDY
A 15-bit Continuous-time ΔΣ ADC for Digital Audio Design Targets

- Audio ADC (24 kHz Bandwidth)
- 15 bit resolution
- OSR = 64 ($f_s = 3.072$ MHz)
- 0.18\(\mu\)m CMOS process, 1.8 V supply
Continuous-time versus Discrete-time A continuous-time implementation was chosen

- Implicit anti-aliasing
- Resistive input impedance
- Low power dissipation
Architectural Choices

- Single-bit versus multibit quantization ?
- Single loop versus MASH ?
- NTF ?
- Loop Filter Architecture ?
Architecture: Single-bit vs Multibit

**Single bit quantizer**
- Simple hardware
- Gentle NTF
- High jitter sensitivity
- Metastability
- Opamp slew rate

**Multibit quantizer**
- Complex hardware
- Aggressive NTF
- Low jitter sensitivity
- Metastability: no issue
- Reduced slew rate

A 4-bit quantizer is used.
Architecture : Single Loop vs MASH

Matching of transfer functions are needed in a MASH design

- More complicated
- Might require calibration

A single loop design is chosen.
Architecture: Choice of the NTF

A maximally flat NTF is chosen

Small OBG

- High in-band quantization noise
- Low jitter noise
- Increased Maximum Stable Amplitude (MSA)

Large OBG

- Low in-band quantization noise
- High jitter noise
- Reduced Maximum Stable Amplitude (MSA)

An OBG of 2.5 is chosen as a compromise
Effect Of OBG On Jitter And Quantization Noise

![Graph showing the relationship between Out of Band Gain and SNR (dB). The graph includes lines for Peak SQNR, Peak SJNR (50ps jitter), and Peak SJNR (100ps jitter).]
Effect Of Systematic RC Time Constant Variations On The NTF

\[ |\text{NTF}(e^{j\omega})| \]

- Nominal
- 30% Higher
- 30% Lower

\[ \omega/\pi \]

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
MSA And SQNR With Systematic RC Time Constant Variations

[Graph showing Peak SNR (dB) and Maximum Stable Amplitude (dBFS) vs. \([RC]_{\text{nom}}/[RC]\).]

Shanthi Pavan Nagendra Krishnapura
Oversampling Analog to Digital Converters
Feedforward versus Distributed Feedback Loopfilters

\[ \frac{\omega_1}{s} + \frac{\omega_2}{s} + \frac{\omega_3}{s} \]

(a) \( \omega_1 = 2.67, \omega_2 = 2.08, \omega_3 = 0.059 \)

(b) \( \omega_1 = 0.34, \omega_2 = 0.71, \omega_3 = 1.225 \)

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Feedforward versus Distributed Feedback Loop Filters

**Feedforward**
- First integrator is fastest.
- Third integrator is slowest.
- First opamp is power hungry (for noise reasons).
- Third opamp is low power (slowest integrator).
- Small capacitor area.

**Distributed Feedback**
- Third integrator is fastest.
- First integrator is slowest.
- First opamp is power hungry (for noise).
- Third opamp is power hungry (fastest integrator).
- Large capacitor area.

A feedforward loop filter is used.
Loop Filter

idacm

vip

vim

idacp

A1

C1

R1

100K

1.05346pF

R11

R21

R31

Csf

Rf

A2

C2

R2

400K

730fF

Rf

C3

R3

500K

8.6264pF

vop1

vom1

vop2

vom2

vop3

vom3

vop

vom

Rs

R11 = 337 K

R21 = 506 K

R31 = 112 K

Rf = 200K

Cs = 172fF
Excess Delay Compensation: Conventional
Excess Delay Compensation: Proposed
Second Opamp

CMFB

First Stage

Second Stage

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Flash ADC Block Diagram

Vtop

Vref<0>

Vbot

Vref<15>

Vref<0>

Vbot

Vtop

digital back end

4-bit data (ADC output)

15 to DEM/DAC

4-bit data to DEM/DAC

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters

213
Comparator

![Comparator Diagram](image)

(a)

(b)
Effect of Random Offset in the Comparators

SNR (dB)

$\sigma_{\text{offset}}$ (in LSB)

Shanthi Pavan Nagendra Krishnapura

Oversampling Analog to Digital Converters
Digital Backend

Therm. to Binary Converter

4 - Bit Adder

Barrel Shifter

Latch

DAC_in<0:14>

FLASH O/P (in<0:14>)

dem_clkd

EN

15

15

15

DAC_in<0:14> (DAC I/P)
Unit DAC Resistor

- **Vrefp**: 1.6 MΩ
- **Vrefm**: 1.6 MΩ

From Reference Generator

- **dacp**
- **dacm**

To input terminals of first opamp
Reference Generation Circuitry

(a) \[ \text{Vref} \]

\[ V_{\text{refp}} - V_{\text{cm}} \left( \frac{15}{R} \right) \]

\[ (V_{\text{cm}} - V_{\text{refm}}) \left( \frac{15}{R} \right) \]

(b) \[ \text{Vdd} \]

\[ R \]

\[ V_{\text{refp}} \]

\[ C_1 \]

\[ C_{\text{ext}} \]

\[ \text{gnd} \]

\[ \text{vdd} \]

\[ \text{gnd} \]

\[ \text{218} \]
Test Setup and Die Layout
Test Setup Schematic

- **ΣΔ Converter**
- **Vdd**
- **Ibias 500 nA**
- **Vcmref 0.9 V**
- **Clock (3.072 MHz)**
- **4 bits To Logic Analyzer**

**Connections:**
- **Differential Audio Source**
- **Vip**
- **Vim**
- **Vrefp**
- **Vrefm**
- **1 µF**
- **0.1 nF**
Measured Dynamic Range

SNR (dB) vs. Input Power (dB FS)

93.5 dB SNDR
In Band Spectrum

![In Band Spectrum Diagram](image-url)
Out of Band Spectrum

PSD (dB)

Frequency (kHz)

PSD (dB)

Frequency (kHz)
## Performance Summary

**Table:** Summary of Measured ADC performance.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Bandwidth/Clock Rate</td>
<td>24 kHz / 3.072 MHz</td>
</tr>
<tr>
<td>Quantizer Range</td>
<td>3 V&lt;sub&gt;pp, diff&lt;/sub&gt;</td>
</tr>
<tr>
<td>Input Swing for peak SNR</td>
<td>-1 dBFS</td>
</tr>
<tr>
<td>Dynamic Range/SNR/SNDR</td>
<td>93.5 dB/92.5 dB/90.8 dB</td>
</tr>
<tr>
<td>Active Area</td>
<td>0.72 mm&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>Process/Supply Voltage</td>
<td>0.18 μm CMOS/1.8 V</td>
</tr>
<tr>
<td>Power Dissipation (Modulator)</td>
<td>90 μW</td>
</tr>
<tr>
<td>Power Dissipation (Modulator and Reference Buffers)</td>
<td>121 μW</td>
</tr>
<tr>
<td>Figure of Merit (DR/SNR)</td>
<td>0.049 pJ/level, 0.054 pJ/level</td>
</tr>
</tbody>
</table>
Some References ...

- *Delta-Sigma Data Converters: Theory, Design and Simulation*
  S. Norsworthy, R. Schreier and G. Temes, *IEEE Press*

- *The Yellow Bible of ΔΣ ADCs*

- *Understanding Delta-Sigma Data Converters*
  R. Schreier and G. Temes, *IEEE Press*

- *The Green Bible of ΔΣ ADCs*

Both the above are essential reading!
Some References ...

- Theory, Practice, and Fundamental Performance Limits of High-Speed Data Conversion Using Continuous-Time Delta-Sigma Modulators
  J. Cherry, Ph.D Dissertation, Carleton University.
  Excellent reading on continuous-time Delta-Sigma modulator design.

- A Power Optimized Continuous-time $\Delta\Sigma$ ADC for Audio Applications
  Detailed description of the case study discussed in this tutorial.