EE247
Lecture 11

• Filters (continued)
  – Example: Switched-capacitor filters in CODEC integrated circuits
  – Switched-capacitor filter design summary
  – Comparison of various filter topologies

• New Topic: Data Converters

---

Summary
Last Lecture

• Switched-capacitor filters
  – Switched-capacitor integrators
    • LDI integrators
    • Effect of parasitic capacitance
    • Bottom-plate integrator topology
  – Resonators
  – Bandpass filters
  – Lowpass filters
    • Termination implementation
    • Transmission zero implementation
  – Switched-capacitor filter design considerations
  – Effect of non-idealities
  – Switched-capacitor filters utilizing double sampling technique
Switched-Capacitor Filter Application
Example: Voice-Band CODEC (Coder-Decoder) Chip


CODEC Transmit Path
Lowpass Filter Frequency Response

Note: $f_s = 128$ kHz
Low Q bandpass (Q<1) filter shape → Implemented with lowpass followed by highpass
**CODEC Transmit Path**

**Clocking & Anti-Aliasing Scheme**

First filter (1st order RC type) performs anti-aliasing for the next S.C. biquad

The 1st & 2nd stage filters form 3rd order elliptic LPF with corner frequency @ 32kHz \(\rightarrow\) Anti-aliasing for the next lowpass filter

The stages prior to the high-pass perform anti-aliasing for high-pass

- Notice gradual lowering of clock frequency \(\rightarrow\) Ease of anti-aliasing

---

**SC Filter Summary**

- Pole and zero frequencies proportional to
  - Sampling frequency \(f_s\)
  - Capacitor ratios
    - High accuracy and stability in response
    - Long time constants realizable without requiring large value \(R\)
- Compatible with transconductance amplifiers
  - Reduced circuit complexity, power dissipation
- Amplifier bandwidth requirements less stringent compared to CT filters (low frequencies only)
- Issue: Sampled-data filters \(\rightarrow\) require anti-aliasing prefiltering
Switched-Capacitor Filters versus Continuous-Time Filter Limitations

Considering overall effects of:

- Opamp finite slew rate
- Opamp finite unity-gain-bandwidth
- Opamp settling issues
- Clock feedthru
- Switch+ sampling cap. finite time-constant

→ Limited switched-capacitor filter performance frequency range

Summary
Filter Performance versus Filter Topology

<table>
<thead>
<tr>
<th></th>
<th>Max. Usable Bandwidth</th>
<th>SNDR</th>
<th>Freq. tolerance w/o tuning</th>
<th>Freq. tolerance + tuning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opamp-RC</td>
<td>~10MHz</td>
<td>60-90dB</td>
<td>+30-50%</td>
<td>1-5%</td>
</tr>
<tr>
<td>Opamp-MOSFET-C</td>
<td>~5MHz</td>
<td>40-60dB</td>
<td>+30-50%</td>
<td>1-5%</td>
</tr>
<tr>
<td>Opamp-MOSFET-RC</td>
<td>~5MHz</td>
<td>50-90dB</td>
<td>+30-50%</td>
<td>1-5%</td>
</tr>
<tr>
<td>Gm-C</td>
<td>~100MHz</td>
<td>40-70dB</td>
<td>+40-60%</td>
<td>1-5%</td>
</tr>
<tr>
<td>Switched Capacitor</td>
<td>~10MHz</td>
<td>40-90dB</td>
<td>&lt;&lt;1%</td>
<td>–</td>
</tr>
</tbody>
</table>
Material Covered in EE247
Where are We?

✓ Filters
  – Continuous-time filters
    • Biquads & ladder type filters
    • Opamp-RC, Opamp-MOSFET-C, gm-C filters
    • Automatic frequency tuning
  – Switched capacitor (SC) filters

• Data Converters
  – D/A converter architectures
  – A/D converter
    • Nyquist rate ADC- Flash, Pipeline ADCs,….
    • Oversampled converters
    • Self-calibration techniques

• Systems utilizing analog/digital interfaces

Data Converters
Suggested Reference Texts


Data Converter Basics

- DSPs benefited from device scaling
- However, real world signals are still analog:
  - Continuous time
  - Continuous amplitude
- DSP can only process:
  - Discrete time
  - Discrete amplitude
  \[\rightarrow\] Need for data conversion from analog to digital and digital to analog

A/D & D/A Conversion

A/D Conversion

- Analog In
- Anti-alias Filtering
- Sampling
- Quantization
- Digital Filter
- Digital Coding
- Digital Out

D/A Conversion

- Digital In
- Digital Filter
- Digital Decoding
- DAC
- Analog Hold
- Reconstruction Filtering
- Analog Out
Data Converters

• Stand alone data converters
  – Used in variety of systems
  – Example: Analog Devices AD9235 12bit/ 65Ms/s ADC- Applications:
    • Ultrasound equipment
    • IF sampling in wireless receivers
    • Various hand-held measurement equipment
    • Low cost digital oscilloscopes

Data Converters

• Embedded data converters
  – Integration of data conversion interfaces along with DSPs and/or RF circuits → Cost, reliability, and performance
  – Main issues
    • Feasibility of integrating sensitive analog functions in a technology typically optimized for digital performance
    • Down scaling of supply voltage as a result of downscaling of feature sizes
    • Interference & spurious signal pick-up from on-chip digital circuitry and/or high frequency RF circuits
    • Portable applications dictate low power consumption
Example: Typical Cell Phone

Contains in integrated form:

- 4 Rx filters
- 4 Tx filters
- 4 Rx ADCs
- 4 Tx DACs
- 3 Auxiliary ADCs
- 8 Auxiliary DACs

Total: Filters → 8
ADCs → 7
DACs → 12

D/A Converter Transfer Characteristics

- An ideal digital-to-analog converter:
  - Accepts digital inputs $b_1-b_n$
  - Produces either an analog output voltage or current
  - Assumption (will be revisited)
    - Uniform, binary digital encoding
    - Unipolar output ranging from 0 to $V_{FS}$

Nomenclature:

- $N = \text{# of bits}$
- $V_{FS} = \text{full scale output}$
- $\Delta = \text{min. step size} \rightarrow \text{ILSB}$
- $\Delta = \frac{V_{FS}}{2^N}$
- $or \ N = \log_2 \frac{V_{FS}}{\Delta} \rightarrow \text{resolution}$
D/A Converter Transfer Characteristics

\[ N = \# \text{of bits} \]
\[ V_{FS} = \text{full scale output} \]
\[ \Delta = \text{min. step size} \rightarrow \text{LSB} \]
\[ \Delta = \frac{V_{FS}}{2^N} \]

\[ V_0 = V_{FS} \sum_{i=1}^{N} b_i 2^{-i} \]
= \Delta \times \sum_{i=1}^{N} b_i 2^{-N-i}, \quad b_i = 0 \text{ or } 1

binary-weighted

\[ V_0 = V_{FS} \sum_{i=1}^{N} b_i 2^{-i} \]
\[ = \Delta \times \sum_{i=1}^{N} b_i 2^{-N-i}, \quad b_i = 0 \text{ or } 1 \]

Note: \( D(b_i = 1, \text{all} i) \)
\[ \rightarrow V_0^{\text{max}} = V_{FS} - \Delta \]
\[ \rightarrow V_0^{\text{max}} = V_{FS} \left(1 - \frac{1}{2^N}\right) \]

Example: D/A with 3-bit Resolution

Example: \( N = 3 \)
Assume \( V_{FS} = 0.8V \)
Input code is 101
\[ V_0 = \Delta \left(b_1 \times 2^2 + b_2 \times 2^1 + b_3 \times 2^0\right) \]
Then: \( \Delta = V_{FS} / 2^3 = 0.4V \)
\[ \rightarrow V_0 = 0.4V \left(1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0\right) = \]
\[ \rightarrow V_0 = 0.5V \]
Note: MSB \( \rightarrow V_{FS} / 2 \) \& LSB \( \rightarrow V_{FS} / 2^N \)
**Ideal 3-Bit D/A Transfer Characteristic**

- Ideal DAC introduces no error!
- One-to-one mapping from input to output

<table>
<thead>
<tr>
<th>Digital Input Code</th>
<th>Analog Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>$V_{FS}$</td>
</tr>
<tr>
<td>001</td>
<td>$V_{FS}/2$</td>
</tr>
<tr>
<td>010</td>
<td>$V_{FS}/8$</td>
</tr>
<tr>
<td>011</td>
<td>$V_{FS}/8$</td>
</tr>
<tr>
<td>100</td>
<td>$V_{FS}/8$</td>
</tr>
<tr>
<td>101</td>
<td>$V_{FS}/8$</td>
</tr>
<tr>
<td>110</td>
<td>$V_{FS}/8$</td>
</tr>
<tr>
<td>111</td>
<td>$V_{FS}/8$</td>
</tr>
</tbody>
</table>

**Step Height (1 LSB $\Delta$)**


**A/D Converter Transfer Characteristics**

- An ideal analog-to-digital converter:
  - Accepts analog input in the form of either voltage or current
  - Produces digital output either in serial or parallel form
- Assumption (will be revisited)
  - Unipolar input ranging from 0 to $V_{FS}$
  - Uniform, binary digital encoding

$V_{FS} =$ full scale output

\[ \Delta = \min \text{ resolvable input } \rightarrow \text{ILSB} \]

\[ \Delta = \frac{V_{FS}}{2^N} \]

or

\[ N = \log_2 \frac{V_{FS}}{\Delta} \rightarrow \text{resolution} \]
Ideal A/D Transfer Characteristic

- Ideal ADC introduces error with max peak-to-peak: $\rightarrow (+1/2 \Delta)$
  $\Delta = V_{FS}/2^N$

  $N =$ # of bits

- This error is called "quantization error"

Non-Linear Data Converters

- So far data converter characteristics studied are with uniform, binary digital encoding

- For some applications to maximize dynamic range non-linear coding is used e.g. Voice-band telephony,
  - Small signals $\rightarrow$ larger # of codes
  - Large signals $\rightarrow$ smaller # of codes
Example: Non-Linear A/D Converter
For Voice-Band Telephony Applications

Non-linear ADC and DAC used in voice-band CODECs

- To maximize dynamic range without need for large # of bits
- Non-linear Coding scheme called A-law & μ-law is used
- Also called companding


Data Converter Performance Metrics

- Data Converters are typically characterized by static, time-domain, & frequency domain performance metrics:
  - Static
    - Monotonicity
    - Offset
    - Full-scale error
    - Differential nonlinearity (DNL)
    - Integral nonlinearity (INL)
  - Dynamic
    - Delay, settling time
    - Aperture uncertainty
    - Distortion-harmonic content
    - Signal-to-noise ratio (SNR), Signal-to-(noise+distortion) ratio (SNDR)
    - Idle channel noise
    - Dynamic range & spurious-free dynamic range (SFDR)
Typical Sampling Process

CT ⇒ SD ⇒ DT

Continuous Time

Sampled Data (e.g. T/H signal)

Clock

Discrete Time

Physical Signals

"Memory Content"

Discrete Time Signals

• A sequence of numbers (or vector) with discrete index time instants

• Intermediate signal values not defined (not the same as equal to zero!)

• Mathematically convenient, non-physical

• We will use the term "sampled data" for related signals that occur in real, physical interface circuits
Uniform Sampling

- Samples spaced T seconds in time
- Sampling Period $T \iff$ Sampling Frequency $f_s = 1/T$
- Problem: Multiple continuous time signals can yield exactly the same discrete time signal (aliasing)

Data Converters

- ADC/DACs need to *sample/reconstruct* to convert from continuous-time to discrete-time signals and back
- Purely mathematical discrete-time signals are different from "sampled-data signals" that carry information in actual circuits
- Question: How do we ensure that sampling/reconstruction fully preserve information?
Aliasing

- The frequencies $f_x$ and $nf_s \pm f_x$, $n$ integer, are indistinguishable in the discrete time domain.

- Undesired frequency interaction and translation due to sampling is called aliasing.

- If aliasing occurs, no signal processing operation downstream of the sampling process can recover the original continuous time signal!

Frequency Domain Interpretation

Signal scenario before sampling

Signal scenario after sampling $\rightarrow$ DT

$\rightarrow$ Signals $@ nf_s \pm f_{max\_signal}$ fold back into band of interest

$\rightarrow$ Aliasing
Brick Wall Anti-Aliasing Filter

Sampling at Nyquist rate \( (f_s=2f_{\text{signal}}) \) → required brick-wall anti-aliasing filters

Practical Anti-Aliasing Filter

- Practical filter: Nonzero "transition band"
- In order to make this work, we need to sample faster than 2x the signal bandwidth
- "Oversampling"
Practical Anti-Aliasing Filter

Data Converter Classification

- $f_s > 2f_{\text{max}}$ Nyquist Sampling
  - "Nyquist Converters"
  - Actually always slightly oversampled (e.g. CODEC $f_{\text{sig}\text{max}} = 3.4\text{kHz}$ &
  - ADC sampling $8\text{kHz} \rightarrow f_s/f_{\text{max}} = 2.35$)
  - Requires anti-aliasing filtering prior to A-to-D conversion

- $f_s >> 2f_{\text{max}}$ Oversampling
  - "Oversampled Converters"
  - Anti-alias filtering is often trivial
  - Oversampling is also used to reduce quantization noise, see later
    in the course...

- $f_s < 2f_{\text{max}}$ Undersampling (sub-sampling)
Sub-Sampling

- Sub-sampling → sampling at a rate less than Nyquist rate → aliasing
- For signals centered @ an intermediate frequency → Not destructive!
- Sub-sampling can be exploited to mix a narrowband RF or IF signal down to lower frequencies

Nyquist Data Converter Topics

- Basic operation of data converters
  - Uniform sampling and reconstruction
  - Uniform amplitude quantization
- Characterization and testing
- Common ADC/DAC architectures
- Selected topics in converter design
  - Practical implementations
  - Compensation & calibration for analog circuit non-idealities
- Figures of merit and performance trends
Where Are We Now?

• We now know how to preserve signal information in CT → DT transition

• How do we go back from DT → CT?

Ideal Reconstruction

\[ x(k) \Rightarrow x(t) \]

• The DSP books tell us:

\[ x(t) = \sum_{k=-\infty}^{\infty} x(k) \cdot g(t-kT) \quad g(t) = \frac{\sin(2\pi f t)}{2\pi f t} \]

• Unfortunately not all that practical...
Zero-Order Hold Reconstruction

- How about just creating a staircase, i.e. hold each discrete time value until new information becomes available?
- What does this do to the frequency content of the signal?
- Let's analyze this in two steps...

DT→CT: Infinite Zero Padding

Time Domain

Frequency Domain

Next step: pass the samples through a sample & hold block (ZOH)
Hold Pulse $T_p = T_s$ Transfer Function

\[ |H(f)| = \frac{\sin(\pi f T_s)}{\pi f T_s} \]

ZOH Spectral Shaping

- Continuous Time Pulse Train Spectrum
- ZOH Transfer Function ("Sinc Shaping")
- ZOH output, Spectrum of Staircase Approximation
Smoothing Filter

- Order of the filter required is a function of oversampling ratio.
- High oversampling helps reduce filter order requirement.
- Filter out the high frequency content associated with staircase shape of the signal.

Summary

- Sampling theorem $f_s > 2f_{max}$ usually dictates anti-aliasing filter.
- If theorem is met, CT signal can be recovered from DT without loss of information.
- ZOH and smoothing filter reconstruct CT signal from DT vector.
- Oversampling helps reduce order & complexity of anti-aliasing & smoothing filters.
Next Topic

- Done with "Quantization in time"

- Next: Quantization in amplitude

Ideal ADC ("Quantizer")

- Accepts & analog input & generates it’s digital representation
- Quantization step:
  \[ \Delta (= 1 \text{ LSB}) \]
- Full-scale input range:
  \[-0.5\Delta \ldots (2^N-0.5)\Delta \]
- E.g. \( N = 3 \) Bits
  \[ V_{FS} = -0.5\Delta \text{ to } 7.5\Delta \]
Quantization Error

- Quantization error → Difference between analog input and output of the ADC converted to analog via an ideal DAC

- Called:
  - Quantization error
  - Residue
  - Quantization noise

![Diagram](attachment://quantization_error_diagram.png)

For an ideal ADC:
- Quantization error is bounded by $-\Delta/2 \ldots +\Delta/2$ for inputs within full-scale range

![Graph](attachment://quantization_error_graph.png)
ADC Dynamic Range

- Assuming quantization noise is much larger compared to circuit generated noise:

\[ D.R._{\text{Maximum}} = 10 \log \frac{\text{Full Scale Signal Power}}{\text{Quantization Noise Power}} \]

- Crude assumption: Same peak/rms ratio for signal and quantization noise!

\[ D.R._{\text{Maximum}} = 20 \log \frac{\text{Peak Full Scale}}{\text{Peak Quantization Noise}} \]

\[ = 20 \log \frac{V_{FS}}{\Delta} = 20 \log 2^N = 6.02 \times N \ [\text{dB}] \]

Question: What is the quantization noise power?

Quantization Error

Let us assume \( V_{in} \) is a ramp signal with amplitude equal to ADC full-scale

![Quantization Error Diagram]

Note: Quantization error waveform \( \rightarrow \) periodic and also ramp
Quantization Error

Need to find the *rms* value for quantization error waveform:

\[ \varepsilon_{eq}^2 = \frac{1}{T} \int_{-\Delta/2}^{+\Delta/2} (k \times t)^2 \, dt = \frac{\Delta^2}{k} \int_{-\Delta/2k}^{+\Delta/2k} t^2 \, dt \]

\[ = \frac{\Delta \times k^2}{k} \int_{-\Delta/2k}^{+\Delta/2k} t^2 \, dt \]

\[ \rightarrow \varepsilon_{eq}^2 = \frac{\Delta^2}{12} \rightarrow \text{Independent of } k \]

\[ \varepsilon_{eq} = \frac{\Delta}{\sqrt{12}} \]

In general above equation applies if:

- Input signal much larger than 1LSB
- Input signal busy
- No signal clipping

Quantization Error PDF

- Probability density function (PDF) Uniformly distributed from 
  \[-\Delta/2 \ldots +\Delta/2\] provided that:
  - Busy input
  - Amplitude is many LSBs
  - No overload
- Not Gaussian!

- Zero mean
- Variance

\[ \varepsilon = \int_{-\Delta/2\Delta}^{+\Delta/2\Delta} \frac{\Delta^2}{12} \, d\varepsilon \]


Signal-to-Quantization Noise Ratio

- If certain conditions the quantization error can be viewed as being "random", and is often referred to as "noise"

- In this case, we can define a peak "signal-to-quantization noise ratio", SQNR, for sinusoidal inputs:
  \[ SQNR = \left( \frac{1}{2N} \right)^2 \frac{\Delta^2}{12} = 1.5 \times 2^{2N} \]

  - Electronic noise
  - Deviations from the ideal quantization levels

Real converters do not quite achieve this performance due to other sources of error:
- Electronic noise
- Deviations from the ideal quantization levels

\[ SQNR = 6.02N + 1.76 \text{ dB} \quad \text{Accurate for } N > 3 \]

SQNR Measurement

- \[ 20 \log(SQNR) \]
- \[ SQNR_{peak} = 6.02N + 1.76 \text{ dB} \]
Static Ideal Macro Models

![Diagram of ADC and DAC](image1)

Cascade of Data Converters

![Diagram of ADC and DAC cascade](image2)
Static Converter Errors

Deviation of converter characteristics from ideal:
- Offset
- Full-scale error
- Differential nonlinearity $\rightarrow$ DNL
- Integral nonlinearity $\rightarrow$ INL

Full-Scale Error

ADC

Ideal full-scale point

Actual full-scale point

(1/2 LSB)

Digital Output Code

Analog Input Value (LSB)

000

0

5

6

7

111

110

111

101

000

0

1

2

3

4

5

6

7

DAC

Ideal full-scale point

Actual full-scale point

(-1 1/4 LSB)

Digital Input Code

Analog Output Value (LSB)

000

0

100

101

110

111

 Offset and Full-Scale Error

Note:

→ For further measurements (DNL, INL) connecting the endpoints & deriving ideal codes based on the non-ideal endpoints eliminates offset and full-scale error
Offset and Full-Scale Errors

- Alternative specification in % Full-Scale = 100% * (of LSB value)/ $2^N$
- Gain error can be extracted from offset & full-scale error
- Non-trivial to build a converter with extremely good full-scale/offset specs
- Typically full-scale/offset is most easily compensated by the digital pre/post-processor
- More critical: Linearity measures $\rightarrow$ DNL, INL

ADC Differential Nonlinearity

1. Endpoints connected
2. Ideal characteristics derived eliminating offset & full-scale error
3. DNL measured
ADC Differential Nonlinearity

- Ideal ADC transitions point equally spaced by 1LSB
- For DNL measurement, offset and full-scale error is eliminated
- DNL [k] (a vector) measures the deviation of each code from its ideal width
- Typically, the vector for the entire code is reported
- If only one DNL # is reported that would be the worst case

Example
Compute Offset, Full-Scale Error, & DNL

A 3bit ADC is designed to have an ideal:
\[ \text{LSB} = 0.1V \]

The measured transitions levels for the end product is shown in the table below, compute offset, full-scale, gain error, & DNL

1- Offset: (real transition-ideal)\(= -0.03V\),
in LSB\(\rightarrow -0.03/0.1 = -0.3\) LSB

2- Full-scale error (real last transition-ideal)
\[ = 0.68 - 0.65 = 0.03V \]
in LSB\(\rightarrow 0.03/0.1 = +0.3\) LSB

3- LSB after correcting for offset & full-scale error:
\[ \text{LSB} = \frac{(\text{Last transition-first transition})(2^N-2)}{2} \]
\[ \text{LSB} = (0.68 - 0.02)/6 = 0.11V \]

<table>
<thead>
<tr>
<th>Transition #</th>
<th>Ideal transition point [V]</th>
<th>Real transition point [V]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>2</td>
<td>0.15</td>
<td>0.15</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>0.2</td>
</tr>
<tr>
<td>4</td>
<td>0.35</td>
<td>0.37</td>
</tr>
<tr>
<td>5</td>
<td>0.45</td>
<td>0.42</td>
</tr>
<tr>
<td>6</td>
<td>0.55</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td>0.65</td>
<td>0.68</td>
</tr>
</tbody>
</table>
ADC Differential Nonlinearity Example

\[ V_{FS} = 2^N \times 0.11V = 0.88V \]

4- Gain relative to ideal
Gain = 0.8/0.88 = 0.9

Find all code widths
\[ \text{Width}[k] = \text{Transition}[k+1] - \text{Transition}[k] \]
- Divide code width by LSB \( W[k] \)

5- Find DNL:
\[ \text{DNL}[k] = W[k] - \text{LSB} \]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0.13</td>
<td>1.18</td>
<td>0.18</td>
</tr>
<tr>
<td>2</td>
<td>0.05</td>
<td>0.45</td>
<td>-0.55</td>
</tr>
<tr>
<td>3</td>
<td>0.17</td>
<td>1.55</td>
<td>0.55</td>
</tr>
<tr>
<td>4</td>
<td>0.05</td>
<td>0.45</td>
<td>-0.55</td>
</tr>
<tr>
<td>5</td>
<td>0.08</td>
<td>0.73</td>
<td>-0.27</td>
</tr>
<tr>
<td>6</td>
<td>0.18</td>
<td>1.64</td>
<td>0.64</td>
</tr>
<tr>
<td>7</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
ADC Differential Nonlinearity Examples

ADC DNL

- DNL=-1 implies missing code
- For an ADC DNL < -1 not possible → undefined
- Can show:

\[
\sum_{i} DNL[i] = 0
\]

- For a DAC DNL < -1 possible