• Switched-Capacitor Filters
  – "Analog" sampled-data filters:
    • Continuous amplitude
    • Quantized time
  – Applications:
    • First commercial product: Intel 2912 voice-band
      CODEC chip, 1979
    • Oversampled A/D and D/A converters
    • Stand-alone filters
      E.g. National Semiconductor LMF100

Switched-Capacitor Filters
• Emulating resistor via switched-capacitor network
• 1st order switched-capacitor filter
• Switch-capacitor filter considerations:
  – Issue of aliasing and how to avoid it
  – Tradeoffs in choosing sampling rate
  – Effect of sample and hold
  – Switched-capacitor filter electronic noise
  – Switched-capacitor integrator topologies

Switched-Capacitor Resistor
• Capacitor C is the "switched capacitor"
• Non-overlapping clocks \( \phi_1 \) and \( \phi_2 \)
  control switches S1 and S2, respectively
• \( v_{\text{IN}} \) is sampled at the falling edge of
  \( \phi_1 \)
  – Sampling frequency \( f_s \)
• Next, \( \phi_2 \) rises and the voltage across
  C is transferred to \( v_{\text{OUT}} \)
• Why does this behave as a resistor?

Switched-Capacitor Resistors
• Charge transferred from \( v_{\text{IN}} \) to
  \( v_{\text{OUT}} \) during each clock cycle is:
  \[ Q = C(v_{\text{IN}} - v_{\text{OUT}}) \]
• Average current flowing from
  \( v_{\text{IN}} \) to \( v_{\text{OUT}} \) is:
  \[ i = \frac{Q}{T_1} = Q \cdot f_s \]
Substituting for \( Q \):
  \[ i = f_s C(v_{\text{IN}} - v_{\text{OUT}}) \]
Switched-Capacitor Resistors

\[ i = f \frac{C}{v_{\text{IN}} - v_{\text{OUT}}} \]

With the current through the switched-capacitor resistor proportional to the voltage across it, the equivalent "switched capacitor resistance" is:

\[ R_{eq} = \frac{T}{f_2} \]

Example:

\[ f = 1 \text{MHz}, C = 1 \text{pF} \]
\[ \rightarrow R_{eq} = 1 \text{Mega}\Omega \]

Switched-Capacitor Filter

• Let’s build a “switched-capacitor” filter …

• Start with a simple RC LPF

• Replace the physical resistor by an equivalent switched-capacitor resistor

• 3-dB bandwidth:

\[ f_{-3dB} = \frac{1}{2\pi R_{eq} C_2} = f_1 \frac{C_1}{C_2} \]

\[ f_{-3dB} = \frac{1}{2\pi f_1 C_2} \]

Switched-Capacitor Filters Advantage versus Continuous-Time Filters

- Corner freq. proportional to:
  - System clock (accurate to few ppm)
  - C ratio accurate \( \rightarrow < 0.1\% \)

- Corner freq. proportional to:
  - Absolute value of Rs & Cs
  - Poor accuracy \( \rightarrow 20\text{ to } 50\% \)

\( \rightarrow \) Main advantage of SC filters \( \rightarrow \) inherent corner frequency accuracy

Typical Sampling Process
Continuous-Time(CT) => Sampled Data (SD)
Uniform Sampling

Nomenclature:

- Continuous time signal: $x(t)$
- Sampling interval: $T$
- Sampling frequency: $f_s = 1/T$
- Sampled signal: $x(kT) = x(k)$

- Problem: Multiple continuous time signals can yield exactly the same discrete time signal
- Let's look at samples taken at 1µs intervals of several sinusoidal waveforms ...

Sampling Sine Waves

$v(t) = \sin[2\pi(101000)t]$  
$T = 1\,\mu s$  
$f_s = 1/T = 1\,MHz$  
$f_{in} = 101\,kHz$

$v(t) = -\sin[2\pi(899000)t]$  
$T = 1\,\mu s$  
$f_s = 1\,MHz$  
$f_{in} = 899\,kHz$

$v(t) = \sin[2\pi(1101000)t]$  
$T = 1\,\mu s$  
$f_s = 1\,MHz$  
$f_{in} = 1101\,kHz$
Sampling Sine Waves

Problem:

Identical samples for:

\[ v(t) = \sin(2\pi f_{in} t) \]
\[ v(t) = \sin(2\pi (f_{in} + f_s) t) \]
\[ v(t) = \sin(2\pi (f_{in} - f_s) t) \]

→ Multiple continuous time signals can yield exactly the same discrete time signal

Frequency Domain Interpretation

Signal scenario before sampling

Signal scenario after sampling & filtering

Key point: Signals @ \( n f_s \neq f_{max, signal} \) fold back into band of interest → Aliasing

Aliasing

- Multiple continuous time signals can produce identical series of samples
  - The folding back of signals from \( n f_s \neq f_{sig} \)
    down to \( f_{fin} \) is called aliasing
  - Sampling theorem: \( f_s > 2f_{max, Signal} \)
- If aliasing occurs, no signal processing operation downstream of the sampling process can recover the original continuous time signal
How to Avoid Aliasing?

- Must obey sampling theorem:
  \[ f_{\text{max, Signal}} < \frac{f_s}{2} \]
- Two possibilities:
  1. Sample fast enough to cover all spectral components, including "parasitic" ones outside band of interest
  2. Limit \( f_{\text{max, Signal}} \) through filtering

Anti-Aliasing Filter Considerations

Case 1: \( B = f_{\text{max, Signal}} = \frac{f_s}{2} \)
- Non-practical since an extremely high order anti-aliasing filter (close to an ideal brickwall filter) is required
- Practical anti-aliasing filter \( \rightarrow \) Nonzero filter "transition band"
- In order to make this work, we need to sample much faster than 2x the signal bandwidth
  \( \rightarrow \) "Oversampling"

Practical Anti-Aliasing Filter

Case 2: \( B = f_{\text{max, Signal}} < \frac{f_s}{2} \)
- More practical anti-aliasing filter
- Preferable to have an anti-aliasing filter with:
  \( \rightarrow \) The lowest order possible
  \( \rightarrow \) No frequency tuning required
  (If frequency tuning is required then why use switched-capacitor filter, just use the prefilter?)
Tradeoff
Oversampling Ratio versus Anti-Aliasing Filter Order

Maximum Aliasing Dynamic Range

Tradeoff: Sampling speed versus anti-aliasing filter order


Effect of Sample & Hold

•Using the Fourier transform of a rectangular impulse:

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

Effect of Sample & Hold on Frequency Response

More practical

Sample & Hold Effect (Reconstruction of Analog Signals)

Magnitude droop due to \(\sin x/x\) effect
Sample & Hold Effect
(Reconstruction of Analog Signals)

Magnitude droop due to \( \sin \frac{x}{x} \) effect:

Case 1) \( f_{in} = \frac{f_s}{4} \)
Droop = \(-1dB\)

Case 2) \( f_{in} = \frac{f_s}{32} \)
Droop = \(-0.0035dB\)

\( \Rightarrow \) High oversampling ratio desirable

Sampling Process Including S/H

1st Order Filter
Transient Analysis

SC response: extra delay and steps with finite rise time.

No problem
1st Order Filter
Transient Analysis

- ZOH: Emulates an ideal S/H → pick signal after settling (usually at end of clock phase)
- Adds delay and $\text{sin}(x)/x$ distortion
- When in doubt, use a ZOH in periodic ac simulations

Magnitude Response

Periodic AC Analysis

- SPICE frequency analysis
  - ac for linear, time-invariant circuits
  - pac for linear, time-variant circuits

- SpectreRF statements
  
  \[
  \text{V1 ( Vi 0 ) vsource type=dc dc=0 mag=1 pacmag=1 PSS1 pss period=1u erpreset=conservative PAC1 pac start=1 stop=1M lin=1001}
  \]

- Output
  - Divide results by $\text{sinc}(f_s f)$ to correct for ZOH distortion
Spectre Circuit File

// Copy from the SpectreRF Primer
module zoh (Pout, Nout, Pin, Nin) (period,
delay, aperture, tc)
node [V,I] Pin, Nin, Pout, Nout;
parameter real period=1 from (0:inf);
parameter real delay=0 from [0:inf);
parameter real aperture=1/100 from (0:inf);
parameter real tc=1/500 from (0:inf);
{
integer n; real start, stop;
node [V,I] hold;
analog {
// determine the point when aperture
begins
n = ($time() - delay + aperture) / period
+ 0.5;
start = n*period + delay - aperture;
$break_point(start);
// determine the time when aperture ends
n = ($time() - delay) / period + 0.5;
stop = n*period + delay;
$break_point(stop);
// Implement switch with effective
// series resistance of 1 Ohm
if ( ($time() > start) && ($time() <= stop))
I(hold) <- V(hold) - V(Pin, Nin);
else
I(hold) <- 1.0e-12 * (V(hold) - V(Pin, Nin));
// Implement capacitor with an effective
// capacitance of tc
I(hold) <- tc * dot(V(hold));
// Buffer output
V(Pout, Nout) <- V(hold);
// Control time step tightly during
// aperture and loosely otherwise
if (($time() >= start) && ($time() <= stop))
$bound_step(tc);
else
$bound_step(period/5);
}
}

ZOH Circuit File

// Implement switch with effective series
// resistance of 1 Ohm
if ( ($time() < period/2) && ($time() >= period/2))
I(hold) <- V(hold);$$;
else
I(hold) <- 1.0e-12 * (V(hold) - V(Phi, Nin));
// Implement capacitor with an effective
// resistance of tc
I(hold) <- tc * dot(V(hold));
// Buffer output
V(Phi, Nin) <- V(hold);
// Control time step tightly during
// aperture and loosely otherwise
if (($time() >= period/2) && ($time() <= period))
$bound_step(tc);
else
$bound_step(period/5);

Sampled-Data Filters

Anti-aliasing Requirements

- Frequency response repeats at $f_s$, $2f_s$, $3f_s$, ...
- High frequency signals close to $f_s$, $2f_s$, ..., folds back into passband (aliasing)
- Most cases must pre-filter input to a sampled-data filter to remove signal at $f > f_s/2$ ($\text{nyquist} \Rightarrow f_{\text{max}} < f_s/2$)
- Usually, anti-aliasing filter included on-chip as continuous-time filter with relaxed specs. (no tuning)
Example: Anti-Aliasing Filter Requirements

- Voice-band SC filter $f_{AMB} = 4\text{kHz}$ & $f_s = 256\text{kHz}$
- Anti-aliasing filter requirements:
  - Need $40\text{dB}$ attenuation at clock frequency
  - Incur no phase-error from 0 to 4kHz
  - Gain error 0 to 4kHz < 0.05dB
  - Allow +/-30% variation for anti-aliasing corner frequency (no tuning)

Need to find minimum required filter order

Example: Anti-Aliasing Filter Specifications

- Normalized frequency for $0.05\text{dB}$ droop: need perform passband simulation
  $0.34 \rightarrow 4\text{kHz} / 0.34 = 12\text{kHz}$
- Set anti-aliasing filter corner frequency for minimum corner frequency $12\text{kHz}$ Nominal corner frequency $12\text{kHz} \times 0.7 = 17.1\text{kHz}$
- Check if attenuation requirement is satisfied for widest filter bandwidth
  $17.1 \times 1.3 = 22.2\text{kHz}$
- Normalized filter clock frequency to max. corner freq.
  $256 / 22.2 \approx 11.48$ make sure enough attenuation
- Check phase-error within 4kHz bandwidth. simulation

From: Williams and Taylor, p. 2-37

Oversampling Ratio versus Anti-Aliasing Filter Order

Maximum Aliasing Dynamic Range

Filter Order $f_s / f_{in\_max}$

* Assumption: anti-aliasing filter is Butterworth type

$\rightarrow$ 2nd order Butterworth

$\rightarrow$ Need to find minimum corner frequency for mag. droop < 0.05dB

Example: Anti-Aliasing Filter

- Voice-band SC filter $f_{AMB} = 4\text{kHz}$ & $f_s = 256\text{kHz}$
- Anti-aliasing filter requirements:
  - Need $40\text{dB}$ attenuation at clock freq.
  - Incur no phase-error from 0 to 4kHz
  - Gain error 0 to 4kHz < 0.05dB
  - Allow +/-30% variation for anti-aliasing corner frequency (no tuning)

$\rightarrow$ 2-pole Butterworth LPF with nominal corner freq. of $17\text{kHz}$ & no tuning ($12\text{kHz}$ to $22\text{kHz}$ corner frequency)
Summary

- Sampling theorem \( f_s > 2f_{\text{max_signal}} \)
- Signals at frequencies \( nf_s \pm f_{\text{sig}} \) fold back down to desired signal band, \( f_{\text{sig}} \)
  - This is called aliasing & usually dictates use of anti-aliasing pre-filters
- Oversampling helps reduce required order for anti-aliasing filter
- S/H function shapes the frequency response with \( \sin x/x \)
  - Need to pay attention to droop in passband due to \( \sin x/x \)
- If the above requirements are not met, CT signal can NOT be recovered from SD or DT without loss of information

Switched-Capacitor Noise

- Resistance of switch S1 contributes to an uncorrelated noise charge on C at the end of \( \phi_2 \)
- Mean-squared noise charge transferred from \( v_{\text{IN}} \) to \( v_{\text{OUT}} \) each sample period is \( Q^2 = 2kTC \)

Switched-Capacitor Noise

- Resistance of switch S2 produces a noise voltage on C with variance \( kTC \)
- The corresponding noise charge is \( Q^2 = C V^2 = kTC \)
- This charge is sampled when \( S_1 \) opens

Switched-Capacitor Noise

- The mean-squared noise current due to S1 and S2's kT/C noise is:
  \[ i^2 = (Qf_s)^2 = 2kTCf_s^2 \]
- This noise is approximately white and distributed between 0 and \( f_s/2 \)
  (noise spectra \( \rightarrow \) single sided by convention)
- The spectral density of the noise is:
  \[ \frac{i^2}{\Delta f} = \frac{2kTCf_s}{f_s/2} = 4kTCf_s \]  \( \rightarrow \) S.C. resistor noise equals a physical resistor noise with same value!
Periodic Noise Analysis

Sampling Noise from SC S/H

SpectrRF PNOISE: check
noiseType=timedomain
noiseTimePoints=[...]
as alternative to ZOH.
noiseSkipCount=large

might speed up things in this case.

Vclk
100ns

Vrc
Vrc_hold

R
100kOhm

C
1pF

PNOISE Analysis

sweep from 0 to 20.01M (1037 steps)

Sampled Noise Spectrum

Density of sampled noise
including sinc distortion

Sampled noise normalized
density corrected for sinc distortion

Total Noise

Sampled noise in
0 ... \( f_s / 2 \): 62.2 \( \mu V \) rms
(expect 64 \( \mu V \) for 1pF)

Switched-Capacitor Integrator

Main advantage: No tuning needed
\( \rightarrow \) critical frequency function of ratio of caps & clock freq.

for \( f_{signal} < f_{sampling} \)

\[ V_0 = \int_{t_1}^{t_2} V_{in} \, dt \]

\[ \omega_0 = f_s \times \frac{C_1}{C_f} \]
Switched-Capacitor Integrator

Continuous-Time versus Discrete Time Design Flow

Continuous-Time
- Write differential equation
- Laplace transform \( (F(s)) \)
- Let \( s \rightarrow j\omega \) \( \Rightarrow F(j\omega) \)
- Plot \( |F(j\omega)|\), phase\( (F(j\omega)) \)

Discrete-Time
- Write difference equation \( \Rightarrow \) relates output sequence to input sequence
- Use delay operator \( Z^{-1} \) to transform the recursive realization to algebraic equation in \( Z \) domain
- Set \( Z = e^{j\omega T} \)
- Plot mag./phase versus frequency

Switched-Capacitor Integrator

\[ \Phi_1 \rightarrow Q_1[nT] = C_s V_i[nT] \]
\[ \Phi_2 \rightarrow Q_2[nT] = 0 \]
\[ \Phi_1 \rightarrow Q_1[nT] = C_s V_i[nT] \]
\[ \Phi_2 \rightarrow Q_2[nT] = Q_1[nT] + Q_2[n-1T] \]

Since \( V_o = -Q_1/C_s \) & \( V_i = Q_2/C_s \)
\[ Q_s[nT] = C_s V_i[nT] \]
\[ Q_s[nT] = Q_1[nT] + Q_2[n-1T] \]
Discrete Time Design Flow

- Transforming the recursive realization to algebraic equation in Z domain:
  - Use Delay operator Z:

\[
\begin{align*}
\{nT\} \rightarrow 1 \\
\{n-1/2\} \rightarrow Z^{-1/2} \\
\{n+1/2\} \rightarrow Z^{1/2}
\end{align*}
\]

Switched-Capacitor Integrator

\[
\begin{align*}
-V_{in}nT \rightarrow -C_1V_0[n(n-1)T]+C_4V_{in}[n(n-1)T] \\
V_0(nT) = V_0[n(n-1)T] + \frac{C_4}{C_1}V_{in}[n(n-1)T] \\
V_{in}Z = Z^{-1}V_0(nT) - Z^{-1} \left( \frac{C_4}{C_1}V_{in}(Z) \right) \\
\frac{V_{in}}{V_{in}} = \frac{Z^{-1}}{Z^{-1} - Z^{-1} \frac{C_4}{C_1}V_{in}(Z)}
\end{align*}
\]

z-Plane Characteristics

- Consider variable \( Z = e^{sT} \) for any \( s \) in left-half-plane (LHP):
  \[
  S = -a + jb \\
  Z = e^{-at} e^{jbt} = e^{-at} (\cos bt + jsin bt) \\
  |Z| = e^{-at}, \angle(Z) = bt
  \]

  \( \rightarrow \) For values of \( S \) in LHP \( |Z| < 1 \)

  \( \rightarrow \) For \( a = 0 \) (imag. axis in s-plane) \( |Z| = 1 \) (unit circle)

  if \( \angle(Z) = \pi - bT \) then \( b = \pi T / \omega \)

  Then \( \omega = \omega / 2 \)

z-Domain Frequency Response

- LHP singularities in s-plane map into inside of unit-circle in Z domain
- RHP singularities in s-plane map into outside of unit-circle in Z domain
- The j\( \omega \) axis maps onto the unit circle
**z-Domain Frequency Response**

- Particular values:
  - \( f = 0 \rightarrow z = 1 \)
  - \( f = \frac{f_s}{2} \rightarrow z = -1 \)
- The frequency response is obtained by evaluating \( H(z) \) on the unit circle at \( z = e^{j\omega T} = \cos(\omega T) + jsin(\omega T) \)
- Once \( z = 1 (f = f_s/2) \) is reached, the frequency response repeats, as expected.

**DDI Integrator**

- Pole-Zero Map in z-Plane:
  - \( Z:1\rightarrow 1 \) on unit circle
  - Pole from \( f \rightarrow 0 \) in s-plane mapped to \( Z = 1 \)
  - As frequency increases z domain pole moves on unit circle (CCW)
  - Once pole gets to \( Z = -1 \), \( f = f_s/2 \), frequency response repeats.