Switched-Capacitor Filters

Today

• 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
- $v_{IN}$ 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 $v_{OUT}$
- Why does this behave as a resistor?

Switched-Capacitor Resistors

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

$$i = f_S C (v_{IN} - v_{OUT})$$

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

$$R_{eq} = \frac{1}{f_S C}$$

Example:

$$f_S = 1MHz, C = 1pF$$

$$\rightarrow R_{eq} = 1Mega\Omega$$

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:

$$\omega_{-3dB} = \frac{1}{R_{eq} C_2} = f_s \times \frac{C_1}{C_2}$$

$$f_{-3dB} = \frac{1}{2\pi} f_s \times \frac{C_1}{C_2}$$
Switched-Capacitor Filters Advantage versus Continuous-Time Filters

\[ f_{-3dB} = \frac{1}{2\pi} f_s \times \frac{C_1}{C_2} \]

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

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

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

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

Typical Sampling Process
Continuous-Time(CT) \( \rightarrow \) Sampled Data (SD)

Continuous-Time Signal

Sampled Data

Sampled Data + ZOH

Clock
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 1\( \mu \)s intervals of several sinusoidal waveforms …

Sampling Sine Waves

- \( T = 1\mu s \)
- \( f_s = 1/T = 1MHz \)
- \( f_{in} = 101kHz \)

\( v(t) = \sin [2\pi(101000)t] \)
Sampling Sine Waves

\[ v(t) = -\sin[2\pi(899000)t] \]

\[ T = 1\mu s \]
\[ f_s = 1\text{MHz} \]
\[ f_{in} = 899\text{kHz} \]

Sampling Sine Waves

\[ v(t) = \sin[2\pi(1101000)t] \]

\[ T = 1\mu s \]
\[ f_s = 1\text{MHz} \]
\[ f_{in} = 1101\text{kHz} \]
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 @ \( nf_s \pm f_{max_{signal}} \) fold back into band of interest → **Aliasing**

---

**Aliasing**

- Multiple continuous time signals can produce identical series of samples.
- The folding back of signals from \( nf_s \pm f_{sig} \) 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

1- Push sampling frequency to \( x2 \) of the highest freq.
   \( \rightarrow \) In most cases not practical

2- Pre-filter signal to eliminate signals above \( f_s/2 \) then sample
Anti-Aliasing Filter Considerations

Case 1: \( B = f_{\text{max}} \) — Signal = \( f_s/2 \)
- 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: \( B = f_{\text{max}} - \text{Signal} \ll f_s/2 \)
- 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

Filter Order

Tradeoff: Sampling speed versus anti-aliasing filter order

* Assumption: anti-aliasing filter is Butterworth type (not a necessary requirement)

Effect of Sample & Hold

*Using the Fourier transform of a rectangular impulse:

\[ |H(f)| = \frac{T_p}{T_s} \frac{\sin(\pi f T_p)}{\pi f T_p} \]
Effect of Sample & Hold on Frequency Response

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

More practical

Sample & Hold Effect (Reconstruction of Analog Signals)

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

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

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

Case 1) \( f_{\text{sig}} = \frac{f_s}{4} \)

Droop = -1dB

Case 2) \( f_{\text{sig}} = \frac{f_s}{32} \)

Droop = -0.0035dB

→ High oversampling ratio desirable
Sampling Process Including S/H

Time Domain

Freq. Domain

Freq. Domain General Signal

1st Order Filter
Transient Analysis

Impractical

No problem

SC response: extra delay and steps with finite rise time.
1st Order Filter
Transient Analysis

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

Periodic AC Analysis
**Magnitude Response**

1. RC filter output
2. SC output after ZOH
3. Input after ZOH
4. Corrected output
   - $(2)$ over $(3)$
   - Repeats filter shape around $nf_s$
   - Identical to RC for $f < f_s/2$

**Periodic AC Analysis**

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

- **SpectreRF statements**
  
  ```
  V1 ( Vi 0 ) vsource type=dc dc=0 mag=1 pacmag=1
  PSS1 pss period=1u errpreset=conservative
  PAC1 pac start=1 stop=1M lin=1001
  ```

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

rc_pac
simulator lang=spectre
ahdl_include "zoh.def"

S1 ( Vi c1 phi1 0 ) relay ropen=100G rclosed=1 vt1=-500m vt2=500m
S2 ( c1 Vo_sc phi2 0 ) relay ropen=100G rclosed=1 vt1=-500m vt2=500m
C1 ( c1 0 ) capacitor c=314.159f
C2 ( Vo_sc 0 ) capacitor c=1p
R1 ( Vi Vo/rc ) resistor r=3.1831M
C2rc ( Vo/rc 0 ) capacitor c=1p
CLK1_Vphi1 ( phi1 0 ) vsource type=pulse val0=-1 val1=1 period=1u
  width=450n delay=50n rise=10n fall=10n
CLK1_Vphi2 ( phi2 0 ) vsource type=pulse val0=-1 val1=1 period=1u
  width=450n delay=550n rise=10n fall=10n
V1 ( Vi 0 ) vsource type=dc dc=0 mag=1 pacmag=1
PSS1 pss period=1u errpreset=conservative
PAC1 pac start=1 stop=3.1M log=1001
ZOH1 ( Vo_sc_zoh 0 Vo_sc 0 ) zoh period=1u delay=500n aperture=1n tc=10p
ZOH2 ( Vi_zoh 0 Vi 0 ) zoh period=1u delay=0 aperture=1n tc=10p

ZOH 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;
begin
  // determine the point when aperture begins
  n = (Time()) - delay + aperture) / period + 0.5;
  start = n * period + delay - aperture;
  break_point(start);
  // 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);
end
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 $f_{3dB} = 4kHz$ & $f_s = 256kHz$
- Anti-aliasing filter requirements:
  - Need 40dB 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

Oversampling Ratio versus Anti-Aliasing Filter Order

- Assumption → anti-aliasing filter is Butterworth type

→ 2nd order Butterworth
→ Need to find minimum corner frequency for mag. droop < 0.05dB
Example : Anti-Aliasing Filter Specifications

- Normalized frequency for 0.05dB droop: need perform passband simulation $\rightarrow 0.34 \rightarrow 4kHz/0.34 = 12kHz$
- Set anti-aliasing filter corner frequency for minimum corner frequency 12kHz $\rightarrow$ Nominal corner frequency 12kHz/0.7 = 17.1kHz
- Check if attenuation requirement is satisfied for widest filter bandwidth $\rightarrow 17.1 \times 1.3 = 22.28kHz$
- Normalized filter clock frequency to max. corner freq. $\rightarrow 256/22.2 = 11.48 \rightarrow$ make sure enough attenuation
- Check phase-error within 4kHz bandwidth: simulation

From: Williams and Taylor, p. 2-37

Example: Anti-Aliasing Filter

- Voice-band SC filter $f_{-3dB} = 4kHz$ & $f_s = 256kHz$
- Anti-aliasing filter requirements:
  - Need 40dB 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 17kHz & no tuning (12kHz to 22kHz 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 \( \frac{\sin x}{x} \).
  - Need to pay attention to droop in passband due to \( \frac{\sin x}{x} \).
- 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 \( S_1 \) produces a noise voltage on \( C \) with variance \( kT/C \).
- The corresponding noise charge is \( Q^2 = C^2V^2 = kTC \).
- This charge is sampled when \( S_1 \) opens.
Switched-Capacitor Noise

- Resistance of switch S2 contributes to an uncorrelated noise charge on C at the end of φ₂.

- Mean-squared noise charge transferred from v_{IN} to v_{OUT} each sample period is $Q^2 = 2kTC$.

\[ Q^2 = 2kTC \]

Switched-Capacitor Noise

- The mean-squared noise current due to S1 and S2’s kT/C noise is:

\[ i^2 = \left( Qf_s \right)^2 = 2k_BT Cf_s^2 \]

- This noise is approximately white and distributed between 0 and $f_s/2$ (noise spectra → single sided by convention)

The spectral density of the noise is:

\[ \frac{i^2}{\Delta f} = \frac{2k_BT C f_s^2}{f_s/2} = 4k_BT C f_s = \frac{4k_BT}{R_{\varepsilon}} \]

using $R_{\varepsilon} = \frac{1}{f_s C}$

→ S.C. resistor noise equals a physical resistor noise with same value!
Periodic Noise Analysis

Sampling Noise from SC S/H

SpectreRF PNOISE: check
noisetype=timedomain
noisetimepoints=[...]
as alternative to ZOH.
noiseskipcount=large
might speed up things in this case.

Netlist

ahdl_include "zoh.def"
Vclk 100ns
Vrc
Vrc_hold

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 \( \mu \text{V} \) rms
(expect 64 \( \mu \text{V} \) for 1pF)

Switched-Capacitor Integrator

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

Continuous-Time versus Discrete Time Design Flow

**Continuous-Time**
- Write differential equation
- Laplace transform \( F(s) \)
- Let \( s = j\omega \) \( \rightarrow F(j\omega) \)
- Plot \(|F(j\omega)|, \text{phase}(F(j\omega))\)

**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

\[ V_o = -Q_I/C_s \] & \[ V_i = Q_s/C_s \] 

\[ C_s V_i[(n-1)T_s] = C_s V_i[(n-3/2)T_s] \]
Discrete Time Design Flow

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

$$
\begin{align*}
ntS & \rightarrow 1 \\
[(n-1)tS] & \rightarrow Z^{-1} \\
[(n-1/2)tS] & \rightarrow Z^{-1/2} \\
[(n+1)tS] & \rightarrow Z^1 \\
[(n+1/2)tS] & \rightarrow Z^{1/2}
\end{align*}
$$

Switched-Capacitor Integrator

$$
\begin{align*}
-C_lV_o(nT_s) &= -C_l V_o[(n-1)t_s] + C_s V_{in}[(n-1)t_s] \\
V_o(nT_s) &= V_o[(n-1)t_s] - \frac{C_s}{C_l} V_{in}[(n-1)t_s] \\
V_o(Z) &= Z^{-1}V_o(Z) - Z^{-1}\frac{C_s}{C_l} V_{in}(Z) \\
\frac{V_o}{V_{in}}(Z) &= -\frac{C_s}{C_l} \frac{Z^{-1}}{1-Z^{-1}} \\
\text{DDI (Direct-Transform Discrete Integrator)}
\end{align*}
$$
z-Plane Characteristics

- Consider variable $Z = e^{sT}$ for any $s$ in left-half-plane (LHP):
  
  \[
  S = -a + jb \\
  Z = e^{-aT} \cdot e^{jbT} = e^{-aT} (\cos bT + jsin bT) \\
  |Z| = e^{-aT}, \angle(Z) = bT \\
  \rightarrow \text{For values of } S \text{ in LHP } |Z| < 1 \\
  \rightarrow \text{For } a = 0 \text{ (imag. axis in s-plane) } |Z| = 1 \text{ (unit circle)} \\
  \text{if } \angle(Z) = \pi = bT \text{ then } b = \pi / T = \omega \\
  \text{Then } \omega = \omega_s / 2
  \]

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

z-Domain Frequency Response

- The angle to the pole is equal to 360° (or \( 2\pi \) radians) times the ratio of the pole frequency to the sampling frequency
DDI Integrator

Pole-Zero Map in z-Plane

$Z = 1 \rightarrow Z = -1$

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

$V_o / V_{in} (Z) = \frac{C_s}{C_l} \frac{Z^{-1}}{1 - Z^{-1}}$

$V_o / V_{in} (Z) = \frac{C_s}{C_l} \frac{1}{e^{j\omega T} - 1}$

Series expansion for $e^x$

$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$

DDI Switched-Capacitor Integrator

$V_o / V_{in} (\omega) = \frac{C_s}{C_l} \frac{1}{1 - (\omega T)^2 + (\omega T)^4 + \ldots}$

for $\omega T << 1$

$V_o / V_{in} (\omega) = \frac{C_s}{C_l} \frac{1}{j\omega T}$

Since $T = 1/f_s$

$V_o / V_{in} (\omega) = \frac{C_s}{C_l} \frac{1}{j\omega} = \frac{1}{C_l f_s \omega}$

$\rightarrow$ ideal integrator
DDI Switched-Capacitor Integrator

\[ \frac{V_o}{V_{in}}(Z) = -\frac{C_o}{C_i} \times \frac{Z^{-1}}{1-Z^{-1}}, \quad Z = e^{j\omega F} \]

\[ = -\frac{C_o}{C_i} \times \frac{1}{1-e^{j\omega F}} = \frac{C_o}{C_i} \times e^{-j\omega F/2}e^{j\omega F/2} \]

\[ = -\frac{C_o}{C_i} \times e^{-j\omega F/2} \times \frac{jT}{2} \times e^{j\omega F/2} \]

Example: Mag. & phase error for:

1. \( f / f_s = 1/12 \)  
   - Mag. Error = 1% or 0.1dB
   - Phase error = 15 degree
   - \( Q_{mag} = 3.8 \)

2. \( f / f_s = 1/32 \)  
   - Mag. Error = 0.16% or 0.014dB
   - Phase error = 5.6 degree
   - \( Q_{mag} = -10.2 \)

DDI Integrator  
- magnitude error no problem
- phase error major problem
Switched Capacitor Filter
Build with DDI Integrator

Example: 5th Order Elliptic Filter
Singularities pushed towards RHP due to integrator excess phase

Coarse View

Fine View

s-plane

jω

s

Pole

Zero

Continuous-Time Prototype

|H(jω)| Passband Peaking

SC DDI based Filter

Zeros lost!
Modified Switched-Capacitor Integrator

DDI Integrator

LDI Integrator

Sample output $\frac{1}{2}$ clock cycle earlier

→ Sample output on $\phi_2$

Switched-Capacitor Integrator

$\Phi_2 \rightarrow Q_i[(n-1)T_s] = C_s V_i[(n-1)T_s]$,
$Q_i[(n-1)T_s] = Q_i[(n-3/2)T_s]$

$\Phi_2 \rightarrow Q_i[(n-1/2)T_s] = 0$,
$Q_i[(n-1/2)T_s] = Q_i[(n-3/2)T_s] + Q_i[(n-1)T_s]$

$\Phi_1 \rightarrow Q_i[nT_s] = C_s V_i[nT_s]$,
$Q_i[nT_s] = Q_i[(n-1)T_s] + Q_i[(n-1)T_s]$

$\Phi_2 \rightarrow Q_i[(n+1/2)T_s] = 0$,
$Q_i[(n+1/2)T_s] = Q_i[(n+1/2)T_s] + Q_i[(n+T_s)]$