EE247
Lecture 9

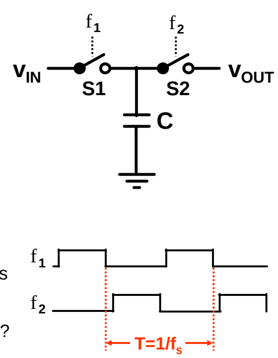
- Switched-Capacitor Filters
  - "Analog" sampled-data filters:
    - Continuous amplitude
    - Quantized time
  - Applications:
    - First commercial product: Intel 2912 voice-band CODEC chip, 1979
    - Oversampled A/D and D/A converters
    - Stand-alone filters
      E.g. National Semiconductor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

Switched-Capacitor Filters 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 ϕ_1 and ϕ_2 control switches S1 and S2, respectively
- v_{IN} is sampled at the falling edge of ϕ_1
  - Sampling frequency f_S
- Next, ϕ_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:

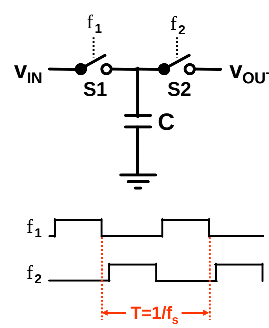
$$Q = C(v_{IN} - v_{OUT})$$

- Average current flowing from v_{IN} to v_{OUT} is:

$$i = Q/t = Q \, . \, f_s$$

Substituting for Q:

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



Switched-Capacitor Resistors

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

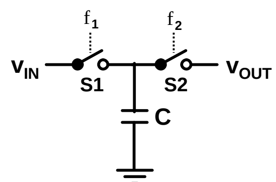
With the current through the switched-capacitor resistor proportional to the voltage across it, the equivalent "switched capacitor resistance" is:

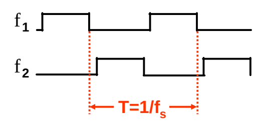
$R_{eq} = \frac{1}{f_S C}$

Example:

$f_S = 1MHz, C = 1pF$

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



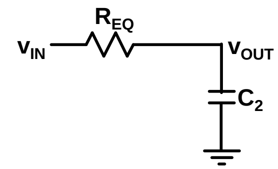


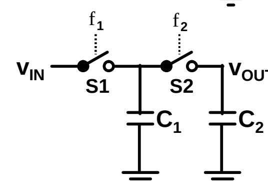
Switched-Capacitor Filter

- 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:

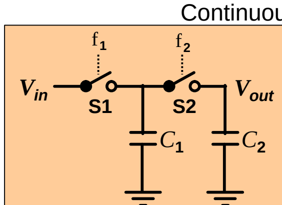
$w_{-3dB} = \frac{1}{R_{eq}C_2} = f_s \times \frac{C_1}{C_2}$

$\boxed{f_{-3dB} = \frac{1}{2p} f_s \times \frac{C_1}{C_2}}$



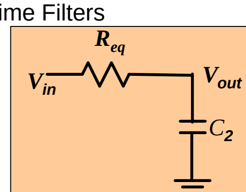


Switched-Capacitor Filters Advantage versus Continuous-Time Filters



$f_{-3dB} = \frac{1}{2p} f_s \times \frac{C_1}{C_2}$

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



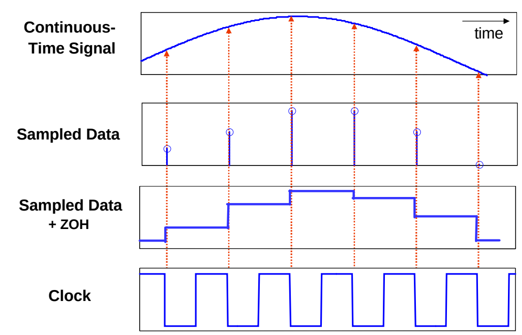
$f_{-3dB} = \frac{1}{2p} \times \frac{1}{R_{eq}C_2}$

- Corner freq. proportional to:
  Absolute value of Rs & Cs
  Poor accuracy → 20 to 50%

Main advantage of SC filters→ inherent corner frequency accuracy

Typical Sampling Process

Continuous-Time(CT) ⇒ Sampled Data (SD)

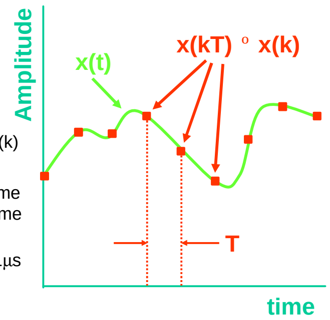


Uniform Sampling

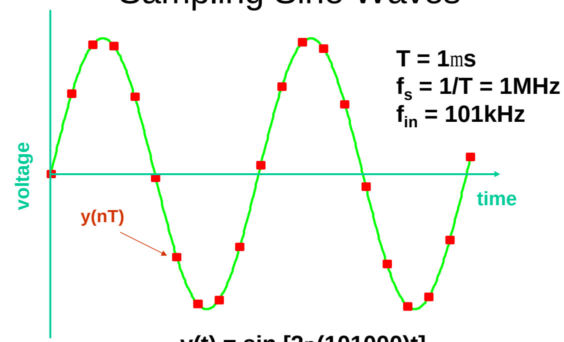
<u>Nomenclature:</u>

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 …

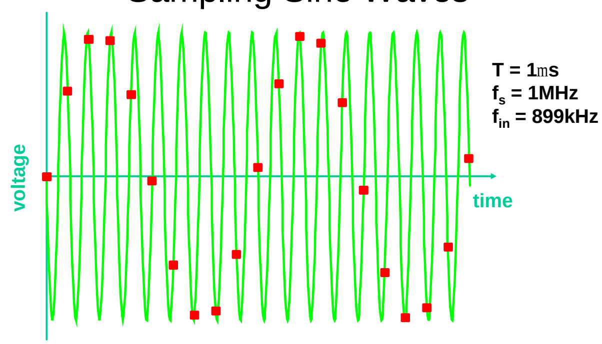


Sampling Sine Waves



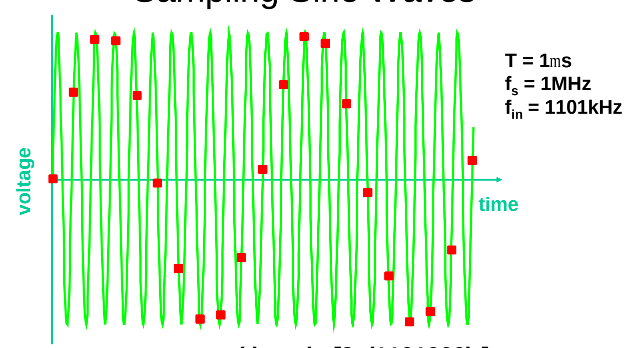
$v(t) = \sin[2\pi(101000)t]$

Sampling Sine Waves



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

Sampling Sine Waves



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

Sampling Sine Waves

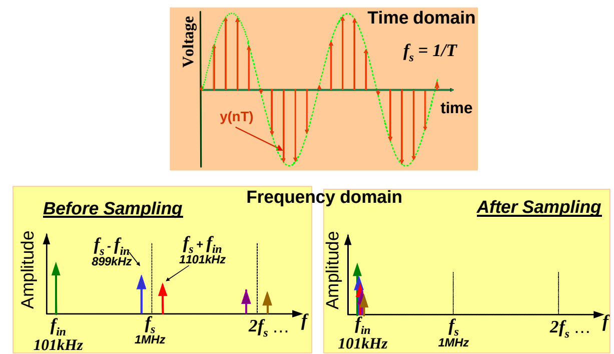
Problem:

Identical samples for:

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

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

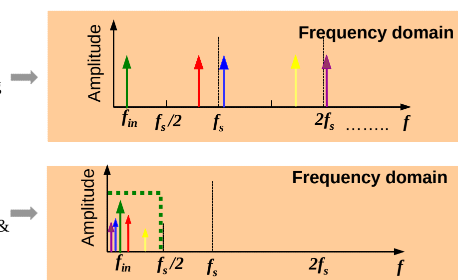
Sampling Sine Waves
Frequency Spectrum



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?

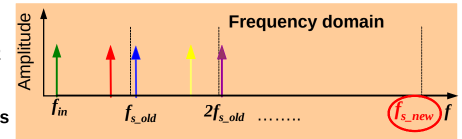
- Must obey sampling theorem:

$$f_{max\_Signal} < f_s/2$$

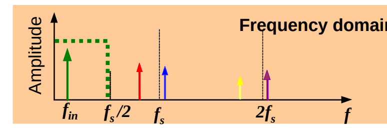
- Two possibilities:
  1. Sample fast enough to cover all spectral components, including "parasitic" ones outside band of interest
  2. Limit f_{max_Signal} through filtering

How to Avoid Aliasing?

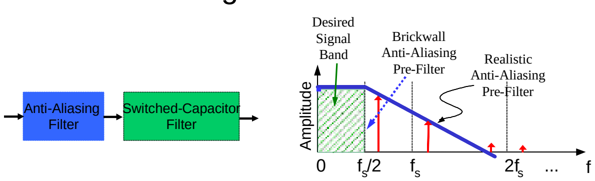
1- Push sampling frequency to x2 of the highest freq.
→ In most cases not practical



2- Pre-filter signal to eliminate signals above $f_s/2$ then sample



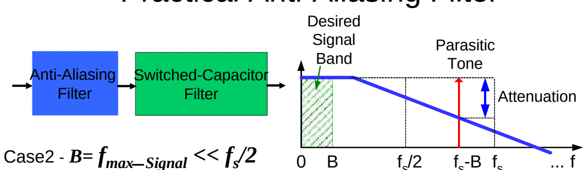
Anti-Aliasing Filter Considerations



Case1- $B = f_{max-Signal} = f_s/2$

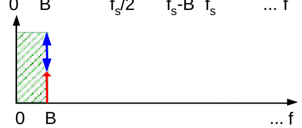
- Non-practical since an extremely high order anti-aliasing filter (close to an ideal brickwall filter) is required
- Practical anti-aliasing filter →Nonzero filter "transition band"
- In order to make this work, we need to sample much faster than 2x the signal bandwidth
  →"Oversampling"

Practical Anti-Aliasing Filter

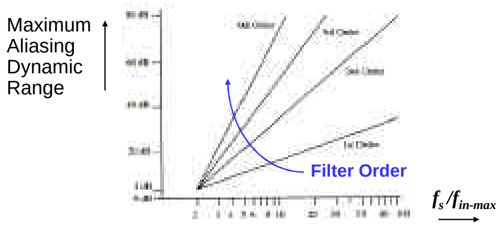


Case2 - $B = f_{max-Signal} << f_s/2$

- More practical anti-aliasing filter
- Preferable to have an anti-aliasing filter with:
  →The lowest order possible
  →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

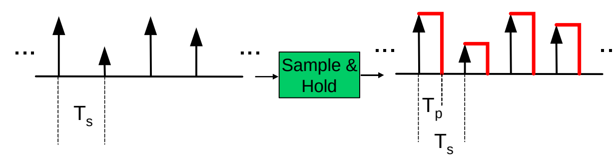


* *Assumption→ anti-aliasing filter is Butterworth type (not a necessary requirement)*

→Tradeoff: Sampling speed versus anti-aliasing filter order

Ref: R. v. d. Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, 2nd ed., Kluwer publishing, 2003, p.41]

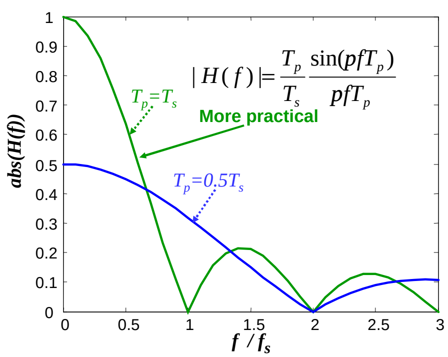
Effect of Sample & Hold



- Using the Fourier transform of a rectangular impulse:

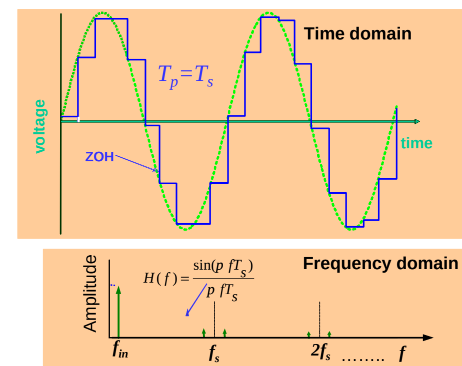
$$\left|H(f)\right| = \frac{T_p}{T_s}\frac{\sin(pfT_p)}{pfT_p}$$

Effect of Sample & Hold on Frequency Response



Sample & Hold Effect
(Reconstruction of Analog Signals)

$T_p=T_s$

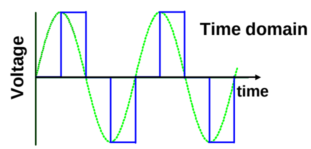


Magnitude droop due to *sinx/x* effect

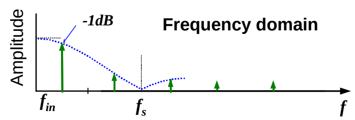
Sample & Hold Effect (Reconstruction of Analog Signals)

Magnitude droop due to *sinx/x* effect:

Case 1) $f_{sig}=f_s/4$



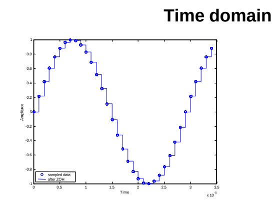
Droop= *-1dB*



Sample & Hold Effect (Reconstruction of Analog Signals)

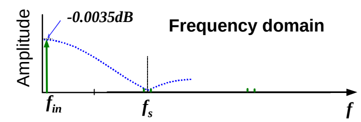
Magnitude droop due to ***sinx/x*** effect:

Case 2) $f_{sig}=f_s/32$

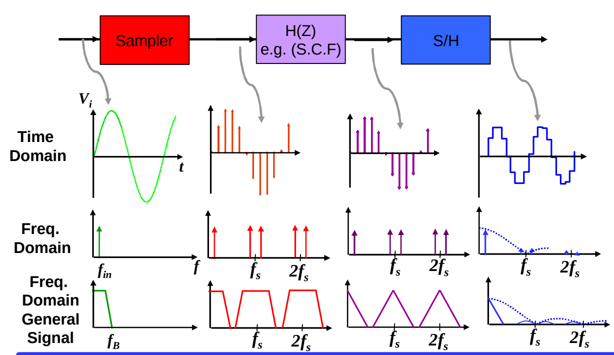


Droop= *-0.0035dB*

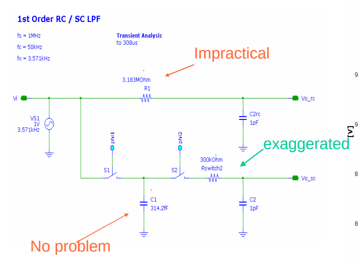
→ High oversampling ratio desirable

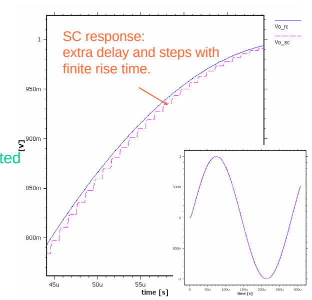


Sampling Process Including S/H

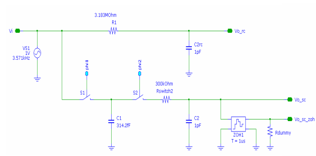
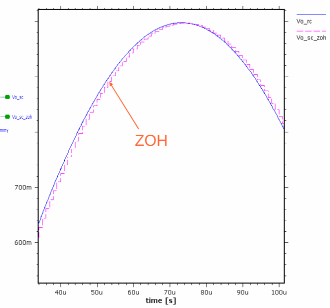


1st Order Filter Transient Analysis



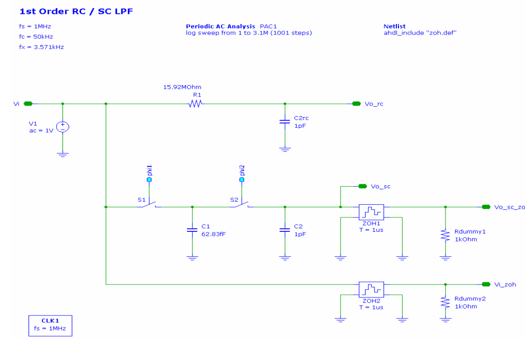


1^{st} Order Filter Transient Analysis

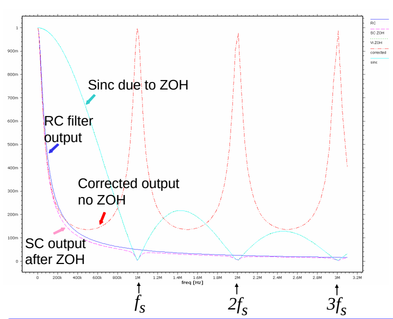



- ZOH: Emulates an ideal S/H→ pick signal after settling (usually at end of clock phase)
- Adds delay and sin(x)/x distortion
- When in doubt, use a ZOH in periodic ac simulations

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 sinc(f/f_s) to correct for ZOH distortion

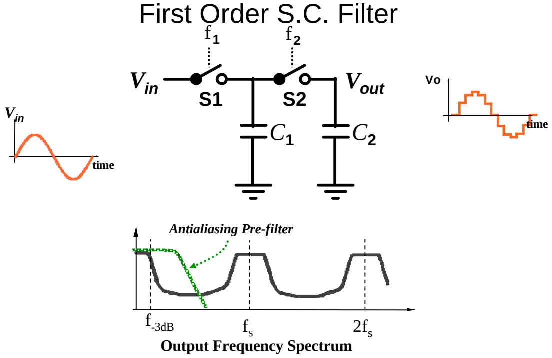
Spectre Circuit File

```
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;
  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
    // resistence 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);
  }
}
```


First Order S.C. Filter

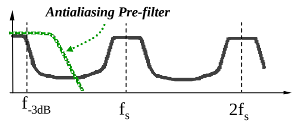


Switched-Capacitor Filters → problem with aliasing

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 $f > f_s/2$ *(nyquist* → $f_{max} < f_s/2$ *)*
- Usually, anti-aliasing filter included on-chip as continuous-time filter with relaxed specs. (no tuning)

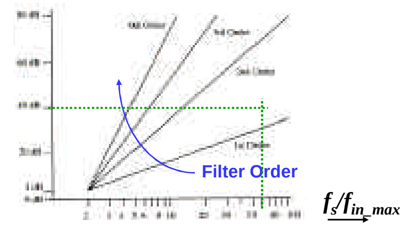
Example : Anti-Aliasing Filter Requirements



- 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



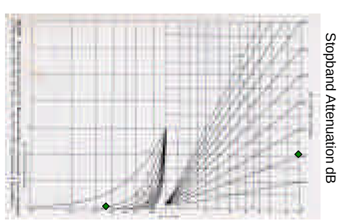
* *Assumption→ anti-aliasing filter is Butterworth type*

→2nd order Butterworth

→Need to find minimum corner frequency for mag. droop < 0.05dB

Example : Anti-Aliasing Filter Specifications

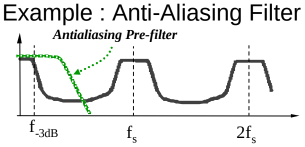
- Normalized frequency for 0.05dB droop: need perform passband simulation→ 0.34→ 4kHz/0.34=12kHz
- Set anti-aliasing filter corner frequency for minimum corner frequency 12kHz → Nominal corner frequency 12kHz/0.7=17.1kHz
- Check if attenuation requirement is satisfied for widest filter bandwidth → 17.1x1.3=22.28kHz
- Normalized filter clock frequency to max. corner freq. →256/22.2=11.48→ make sure enough attenuation
- Check phase-error within 4kHz bandwidth: simulation



Normalized w

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)

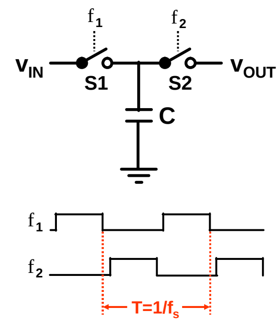
→2-pole Butterworth LPF with nominal corner freq. of 17kHz & no tuning (12kHz to 22kHz corner frequency)

Summary

- Sampling theorem → $f_s > 2f_{max\_Signal}$
- Signals at frequencies $nf_S \pm f_{sig}$ fold back down to desired signal band, f_{sig}
  → This is called <u>aliasing</u> & 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 <u>*sinx/x*</u>
  → Need to pay attention to droop in passband due to <u>*sinx/x*</u>
- If the above requirements are not met, CT signal can NOT be recovered from SD or DT without loss of information

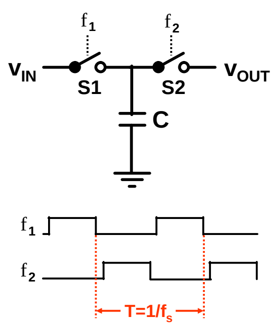
Switched-Capacitor Noise

- Resistance of switch S1 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 ϕ_2
- Mean-squared noise charge transferred from v_{IN} to v_{OUT} each sample period is $Q^2=2kTC$



Switched-Capacitor Noise

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

$$\overline{i^2} = (Qf_s)^2 = 2k_BTCf_s^2$$

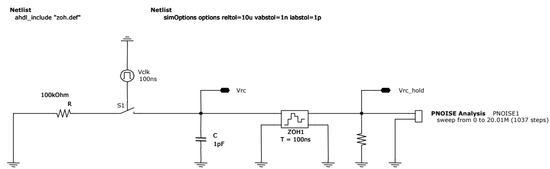
- This noise is approximately white and distributed between 0 and $f_s/2$ (noise spectra → single sided by convention)
  The spectral density of the noise is:

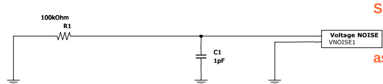
$$\frac{\overline{i^2}}{\Delta f} = \frac{2k_BTCf_s^2}{f_s/2} = 4k_BTCf_s = \frac{4k_BT}{R_{EQ}} \qquad using \qquad R_{EQ} = \frac{1}{f_sC}$$

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

Periodic Noise Analysis

Sampling Noise from SC S/H

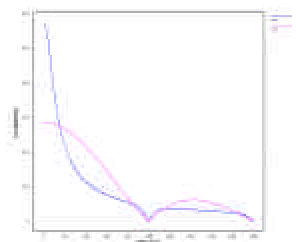




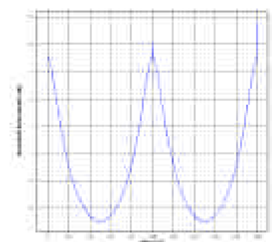
SpectreRF PNOISE: check
noisetype=timedomain
noisetimepoints=[…]
as alternative to ZOH.
noiseskipcount=*large*
might speed up things in this case.

```
PSS pss period=100n maxacfreq=1.5G errpreset=conservative
PNOISE ( Vrc_hold 0 ) pnoise start=0 stop=20M lin=500 maxsideband=10
```


Sampled Noise Spectrum

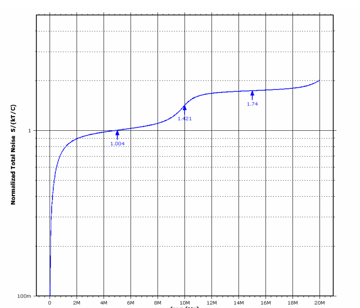


Density of sampled noise including sinc distortion



Sampled noise normalized density corrected for sinc distortion

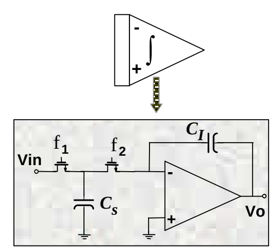
Total Noise

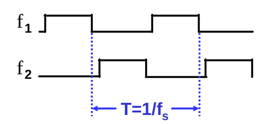


Sampled noise in
0 … $f_s/2$: 62.2µV rms

(expect 64µV for 1pF)

Switched-Capacitor Integrator





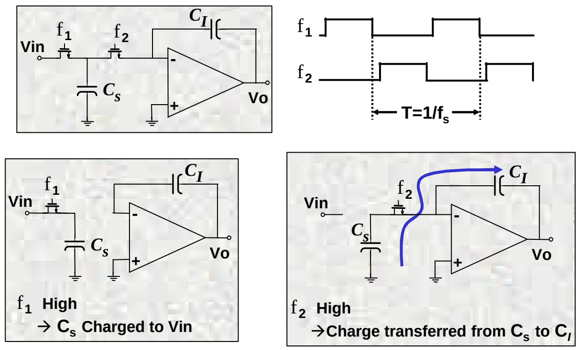
$$for \quad f_{signal} << f_{sampling}$$

$$\rightarrow V_0 = \frac{f_s \times C_s}{C_I} \int V_{in}\, dt$$

$$w_0 = f_s \times \frac{C_s}{C_I}$$

Main advantage: No tuning needed
→ 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=jω → F(jω)
- Plot |F(jω)|, phase(F(jω))

Discrete-Time

- Write difference equation → relates output sequence to input sequence

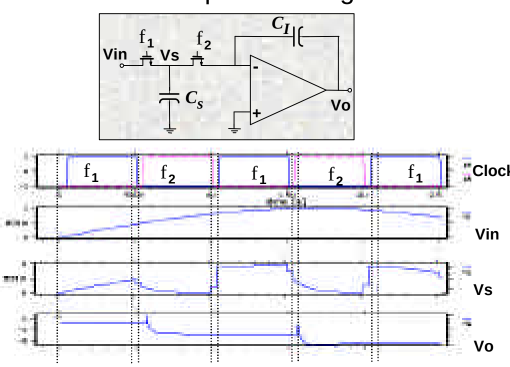
$$V_o(nT_s) = V_i\,[(n-1)T_s] - \ldots\ldots\ldots$$

- Use delay operator Z^{-1} to transform the recursive realization to algebraic equation in Z domain

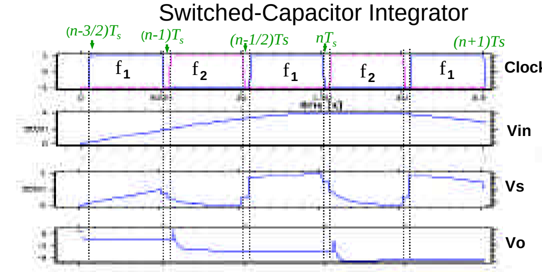
$$V_o(Z) = Z^{-1}V_i(Z)\ldots\ldots$$

- Set $Z = e^{jwT}$
- Plot mag./phase versus frequency

Switched-Capacitor Integrator



Switched-Capacitor Integrator



$\Phi_1 \rightarrow \quad Q_s[(n-1)T_s] = C_s V_i[(n-1)T_s]\,, \quad Q_I[(n-1)T_s] = Q_I[(n-3/2)T_s]$

$\Phi_2 \rightarrow \quad Q_s[(n-1/2)T_s] = 0 \quad , \quad Q_I[(n-1/2)T_s] = Q_I[(n-1)T_s] + Q_s[(n-1)T_s]$

$\Phi_1 \rightarrow \quad Q_s[nT_s] = C_s V_i[nT_s]\,, \quad \boxed{Q_I[nT_s] = Q_I[(n-1)T_s] + Q_s[(n-1)T_s]}$

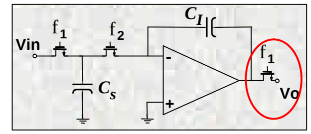
Since $V_o = -Q_I/C_I$ & $V_i = Q_s/C_s \rightarrow C_I V_o(nT_s) = C_I V_o[(n-1)T_s] - C_s V_i[(n-1)T_s]$

Discrete Time Design Flow

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

$$nT_S \ldots\ldots\ldots\ldots\ldots\ldots\ldots \rightarrow 1$$
$$[(n-1)T_S] \ldots\ldots\ldots\ldots \rightarrow Z^{-1}$$
$$[(n-1/2)T_S] \ldots\ldots\ldots \rightarrow Z^{-1/2}$$
$$[(n+1)T_S] \ldots\ldots\ldots\ldots \rightarrow Z^{+1}$$
$$[(n+1/2)T_S] \ldots\ldots\ldots \rightarrow Z^{+1/2}$$

Switched-Capacitor Integrator



$$-C_I V_o(nT_s) = -C_I V_o[(n-1)T_s] + C_s V_{in}[(n-1)T_s]$$
$$V_o(nT_s) = V_o[(n-1)T_s] - \frac{C_s}{C_I} V_{in}[(n-1)T_s]$$
$$V_o(Z) = Z^{-1} V_o(Z) - Z^{-1}\frac{C_s}{C_I} V_{in}(Z)$$
$$\frac{V_o}{V_{in}}(Z) = -\frac{C_s}{C_I} \times \boxed{\frac{Z^{-1}}{1-Z^{-1}}}$$

DDI (Direct-Transform Discrete Integrator)

z-Plane Characteristics

- Consider variable $Z=e^{sT}$ for any S in left-half-plane (LHP):

$S= -a+jb$

$Z= e^{-aT} \cdot e^{jbT} = e^{-aT}(\cos bT + j\sin bT)$

$|Z|= e^{-aT}$, $angle(Z)= bT$

→ For values of S in LHP $|Z|<1$

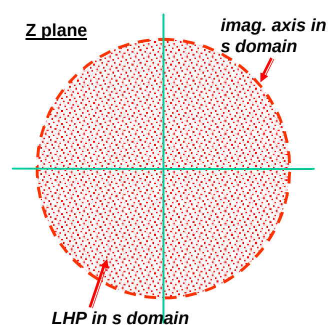
→ For $a=0$ (imag. axis in s-plane) $|Z|=1$ (unit circle)

if $angle(Z)=\pi=bT$ then $b=\pi/T=w$

Then $w=w_s/2$

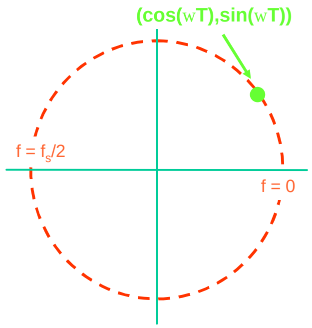
z-Domain Frequency Response

- 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



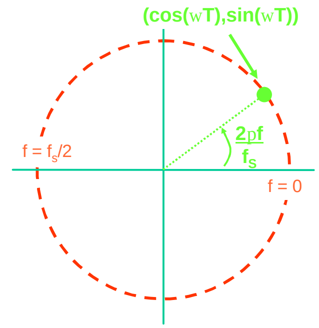
z-Domain Frequency Response

- Particular values:
  - f = 0 → z = 1
  - f = $f_s/2$ → 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 2π radians) times the ratio of the pole frequency to the sampling frequency



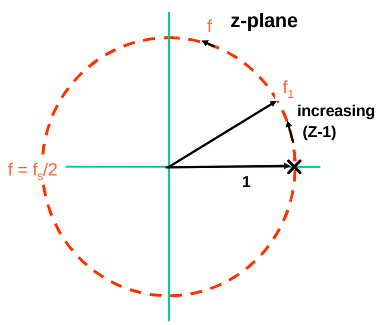
DDI Integrator Pole-Zero Map in z-Plane

Z-1=0 → Z=1 on unit circle

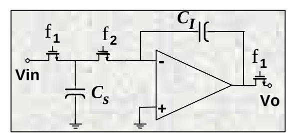
Pole from f→0 in s-plane mapped to z=+1

As frequency increases z domain pole moves on unit circle (CCW)

Once pole gets to (Z=-1),(f=f_s /2), frequency response repeats



DDI Switched-Capacitor Integrator



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

$$\frac{V_o}{V_{in}}(Z) = -\frac{C_s}{C_I}\times\frac{1}{Z-1}\ ,\ Z = e^{jwT}$$

$$= -\frac{C_s}{C_I}\times\frac{1}{e^{jwT}-1} =$$

Series expansion for e^x

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

$$\frac{V_o}{V_{in}}(w) = -\frac{C_s}{C_I}\times\frac{1}{\left[1+jwT+\frac{(jwT)^2}{2!}+\frac{(jwT)^3}{3!}+....\right]-1}$$

$$= -\frac{C_s}{C_I}\times\frac{1}{jwT-\frac{(wT)^2}{2!}+\frac{(wT)^3}{3!}+....}$$

for $wT << 1$

$$\frac{V_o}{V_{in}}(w) = -\frac{C_s}{C_I}\times\frac{1}{jwT}$$

Since $T = 1/f_s$

$$\frac{V_o}{V_{in}}(w) = -\frac{C_s}{C_I}\times\frac{f_s}{s} = -\frac{1}{C_I R_{eq} s}$$

$\rightarrow$ ideal integrator