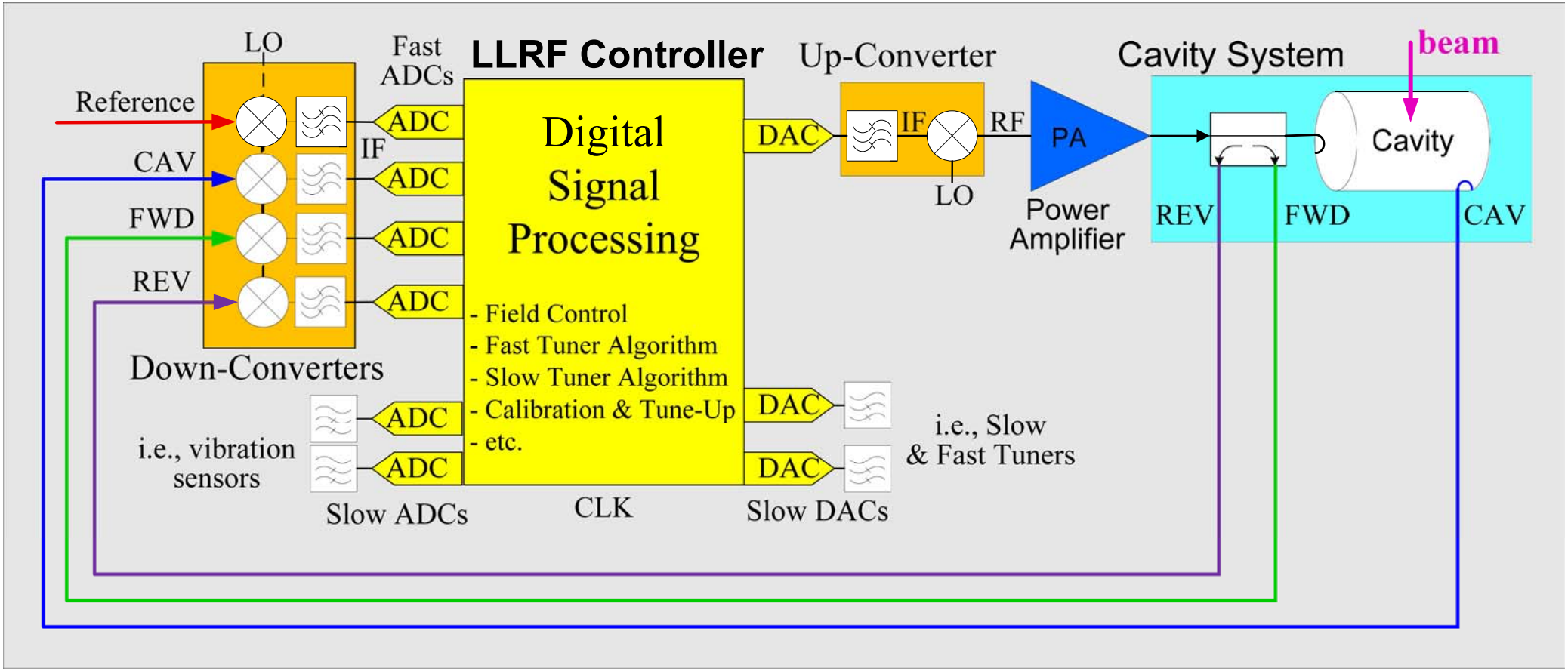
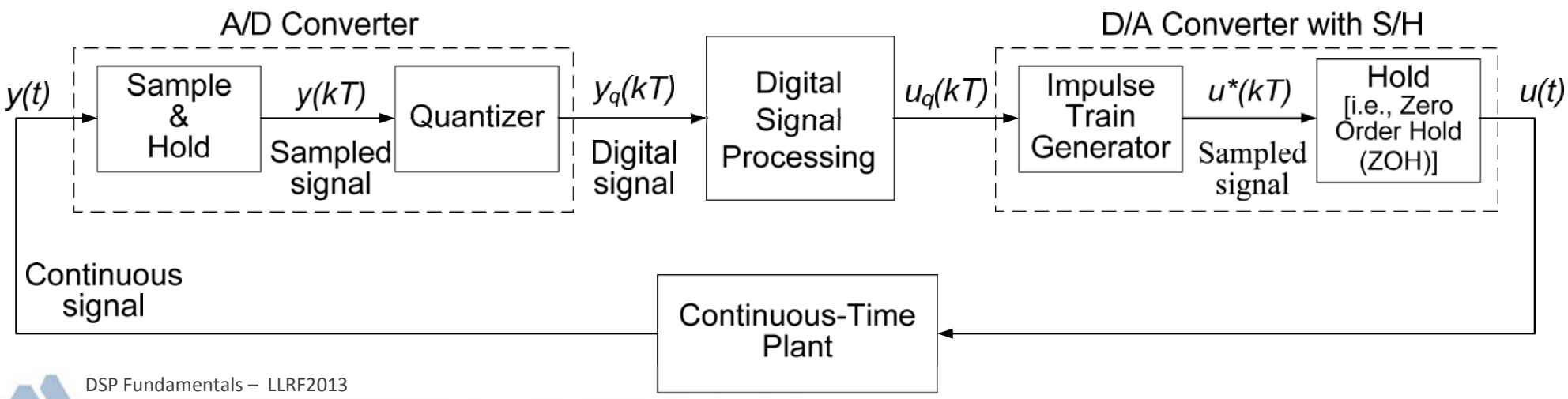
Digital Signal Processing: Fundamentals

Tim Berenc
Argonne National Laboratory, APS

Motivation


LO
Fast ADCs
LLRF Controller
Up-Converter
Cavity System
beam
Reference
CAV
FWD
REV
IF
ADC
ADC
ADC
ADC
Digital Signal Processing
- Field Control
- Fast Tuner Algorithm
- Slow Tuner Algorithm
- Calibration & Tune-Up
- etc.
DAC
IF
RF
LO
PA
Power Amplifier
Cavity
REV
FWD
CAV
Down-Converters
ADC
ADC
i.e., vibration sensors
Slow ADCs
CLK
DAC
DAC
i.e., Slow & Fast Tuners
Slow DACs

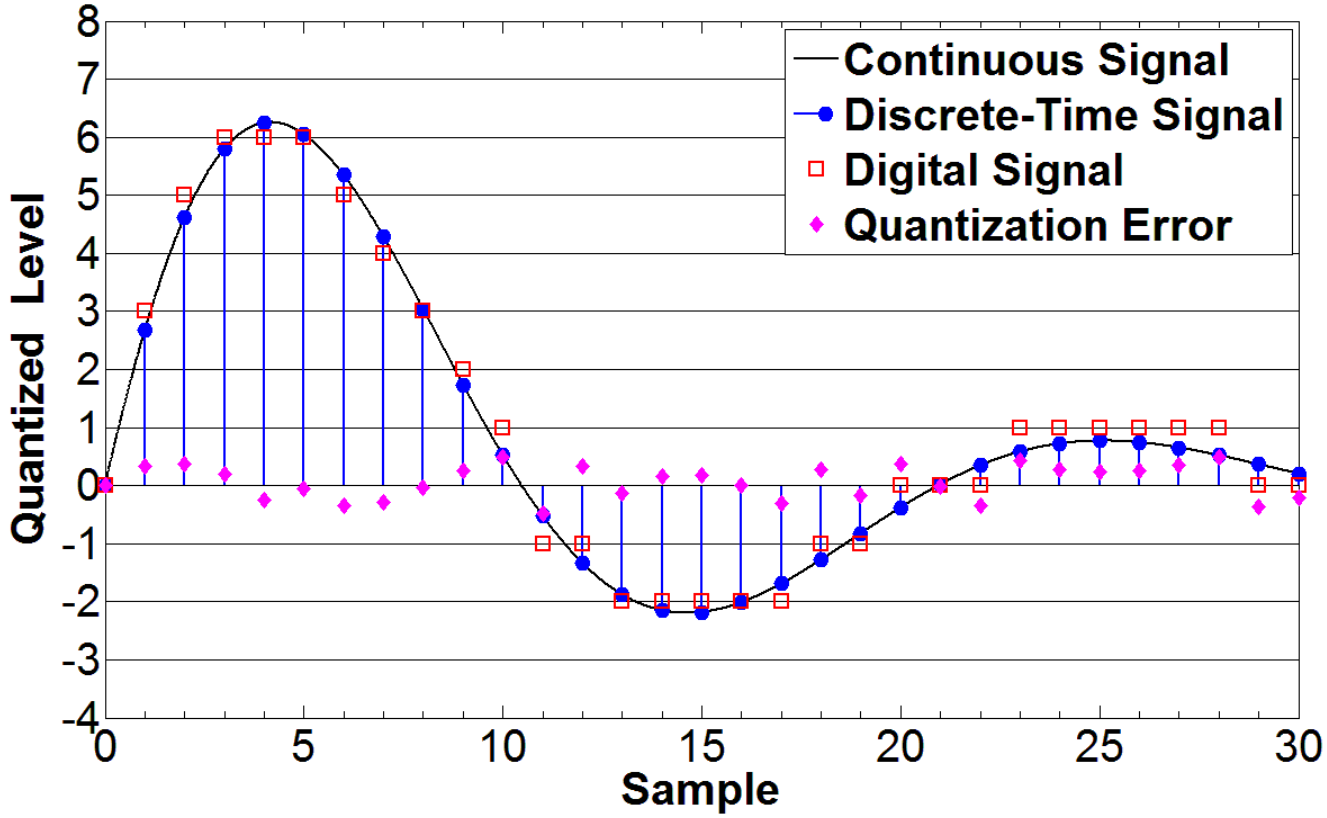

A/D Converter
D/A Converter with S/H
y(t)
Sample & Hold
y(kT)
Sampled signal
Quantizer
yq(kT)
Digital signal
Digital Signal Processing
uq(kT)
Impulse Train Generator
u*(kT)
Sampled signal
Hold [i.e., Zero Order Hold (ZOH)]
u(t)
Continuous signal
Continuous-Time Plant

Outline

- Sampling
- Transforms
- Digital Filters
- System Discretization
- The Cavity System (continuous-time plant)

Sampling

Signal Definitions

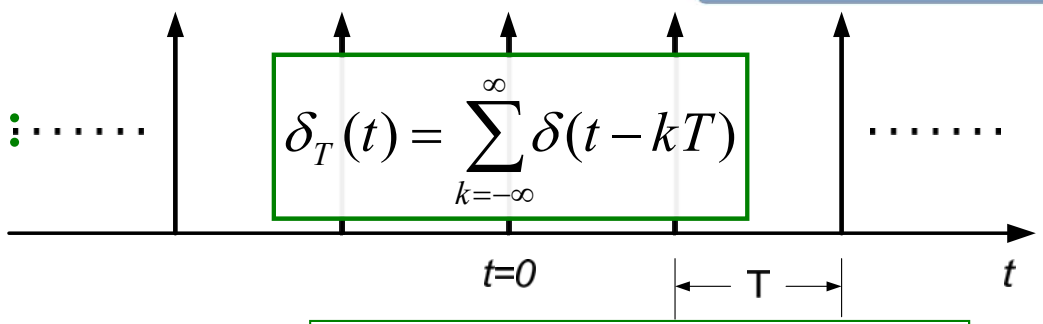


- Continuous Signal: Function of time => $y(t)$
- Discrete-Time Signal: Result of sampling y(t) at discrete times => $y^*(kT)$
- Digital Signal: Result of quantizing the discrete-time signal => $y_q(kT)$
- Quantization Error: $y^*(kT) + \varepsilon(kT) = y_q(kT)$

For ***SMALL*** quantization steps, linear analysis can be used and quantization errors investigated separately. Otherwise the analysis becomes non-linear.

Sampling

Impulse Train: $\cdots\cdots$ $\boxed{\delta_T(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT)}$ $\cdots\cdots$



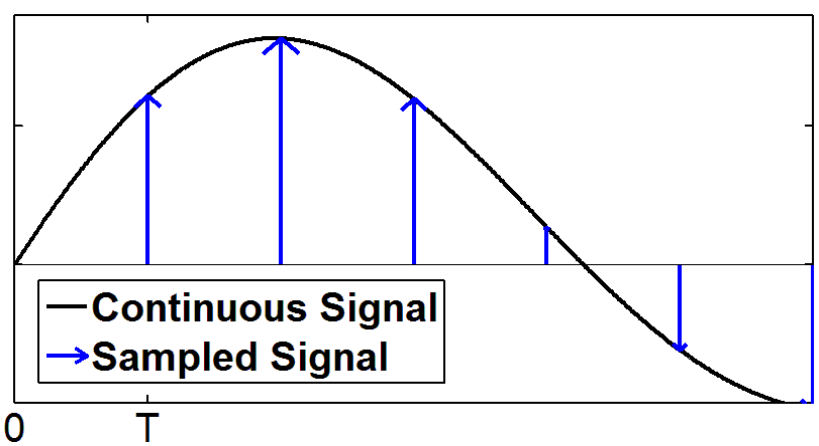
Fourier Series of $\delta_T(t)$

$$\boxed{\delta_T(t) = \sum_{m=-\infty}^{\infty} \alpha_m \, e^{i\,m\frac{2\pi}{T}t}} \;,\; \boxed{\alpha_m = \frac{1}{T}\int_{-T/2}^{T/2} \delta(t)\, e^{i\,m\frac{2\pi}{T}t}\, dt = \frac{1}{T}} \;,\; \boxed{\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} e^{i\,m\frac{2\pi}{T}t}}$$

Fourier transform of a constant: $1 \overset{\mathscr{F}}{\Leftrightarrow} \delta(f)$

modulation property: $y(t)\cdot e^{i2\pi f_o t} \overset{\mathscr{F}}{\Leftrightarrow} Y(f - f_o)$

$$\Rightarrow \quad \boxed{\delta_T(t) \overset{\mathscr{F}}{\Leftrightarrow} \frac{1}{T}\sum_{n=-\infty}^{\infty} \delta(f - n f_s)} \quad , \quad f_s = \frac{1}{T}$$



Sampled Signal:

$$\boxed{y^*(t) = y(t)\cdot\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} y(t)\, e^{i\,m\,f_s t}}$$

sum of modulations

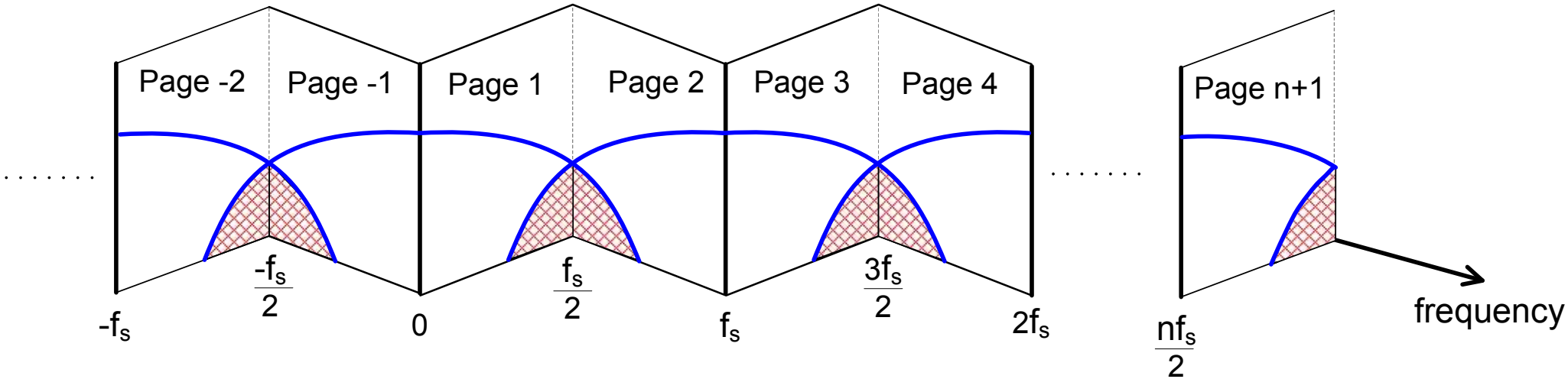
$$\boxed{y^*(t) \overset{\mathscr{F}}{\Leftrightarrow} \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - n f_s)}$$

source of aliasing

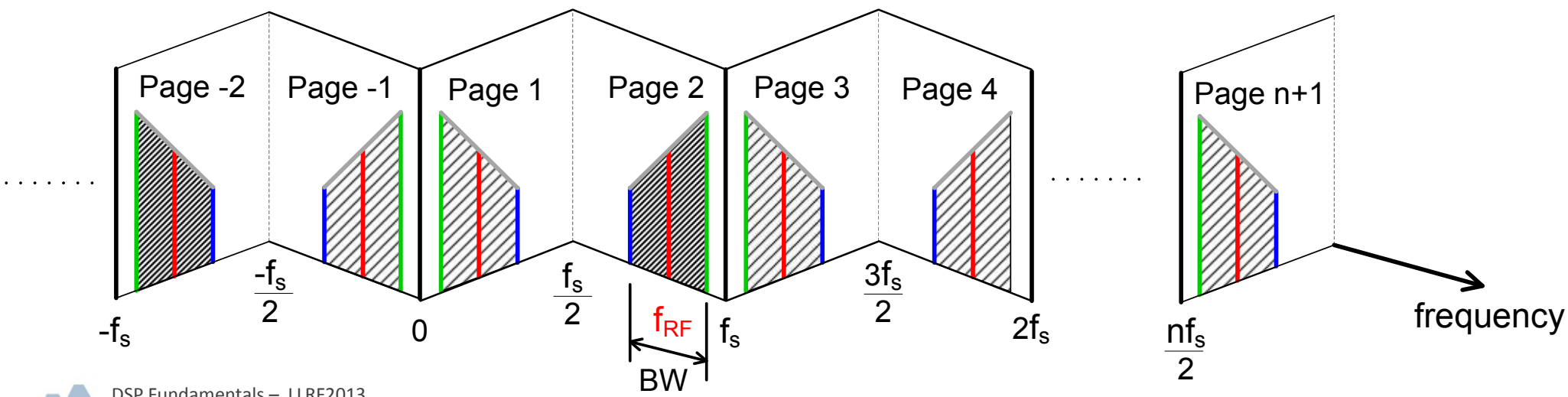
Sampling

$$y^*(t) \overset{\mathscr{F}}{\Leftrightarrow} \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - n f_s)$$

Bad Aliasing

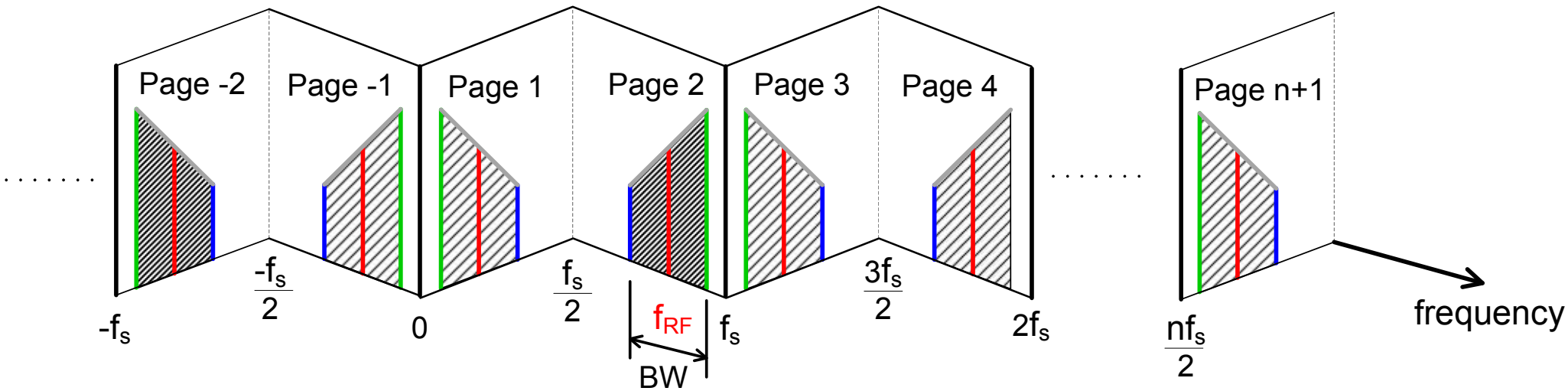


Usable Aliasing: Undersampling



Sampling

$$y^*(t) \overset{\mathscr{F}}{\Leftrightarrow} \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - nf_s)$$



$$\frac{n-1}{2} f_s < f_{RF} - \frac{BW}{2} \quad \text{and} \quad \frac{n}{2} f_s > f_{RF} + \frac{BW}{2}$$

Undersampling Constraint:

$$\frac{2f_{RF} + BW}{n} < f_s < \frac{2f_{RF} - BW}{n-1}$$

$$\frac{2f_{\max}}{n} < f_s < \frac{2f_{\min}}{n-1}$$

Note for n=even:

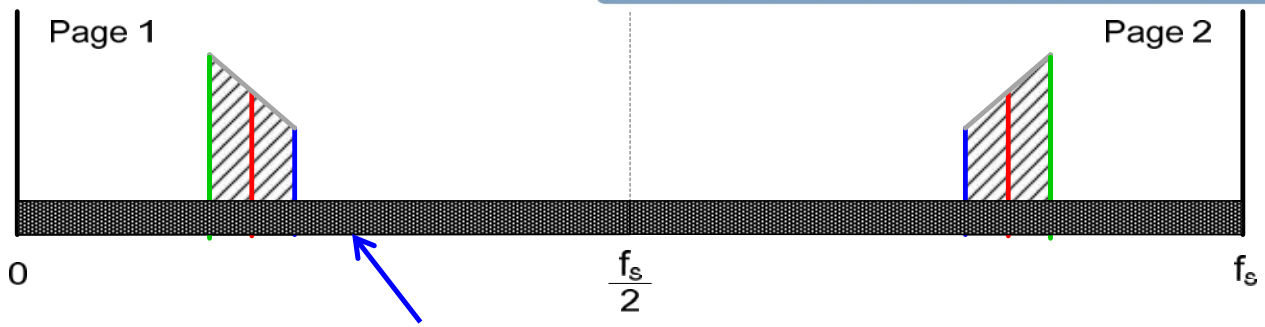
frequency reversal

$\Rightarrow$ single sideband at $f_{rf} + f_{ssb}$ becomes $f_{rf} - f_{ssb}$, sines flip sign

$\Rightarrow$ Counter/clock wise rotations become clock/Counter wise

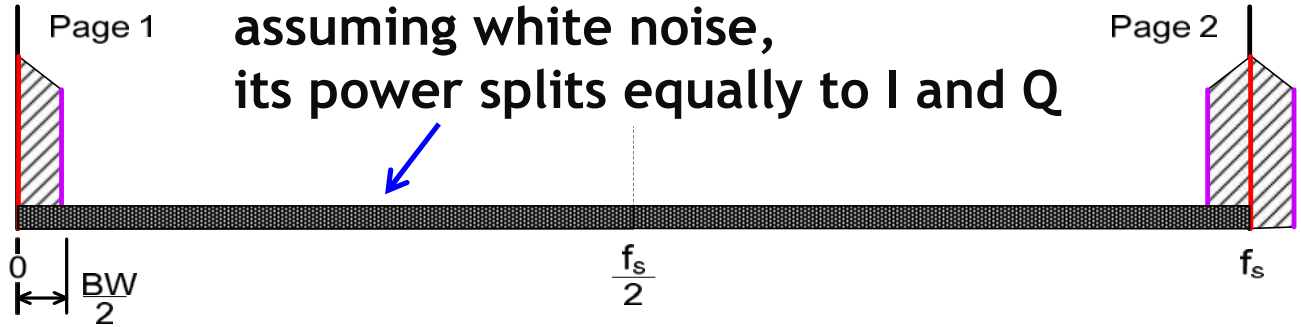
Sampling - Decimation

aliased IF spectrum

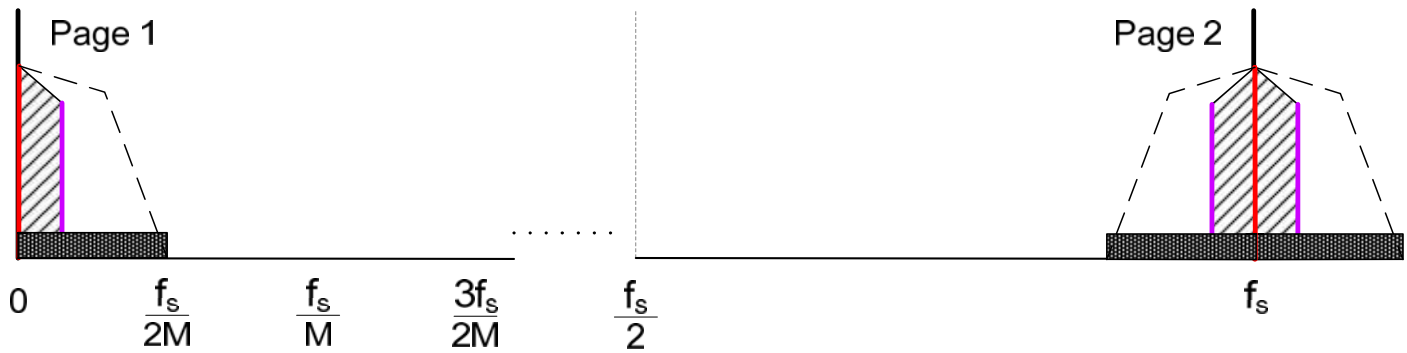


After digital down-conversion (DDC) (single quadrature channel)

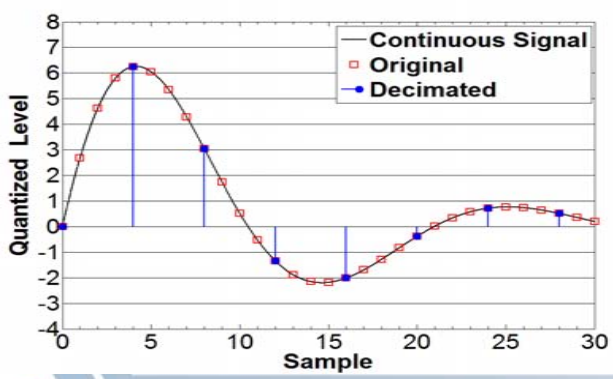
assuming white noise,
its power splits equally to I and Q

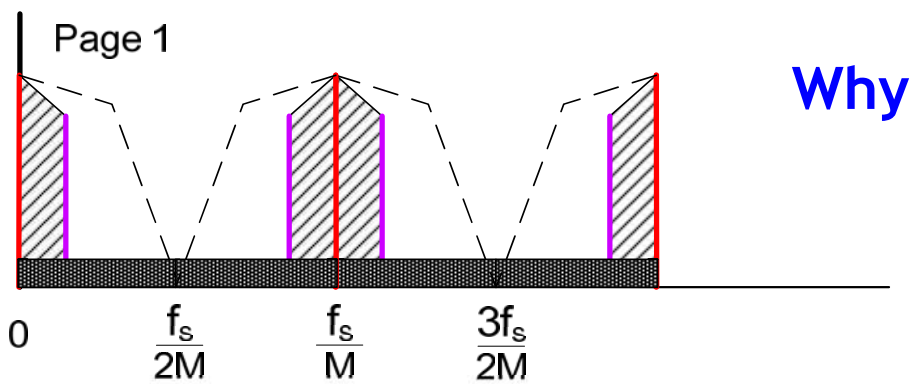


filter to prevent aliasing at new sampling rate



Keep every M^{th} sample for new rate

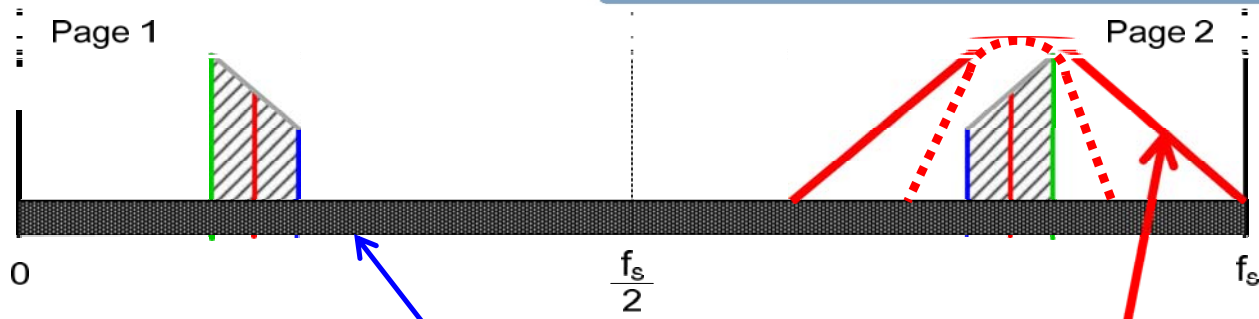




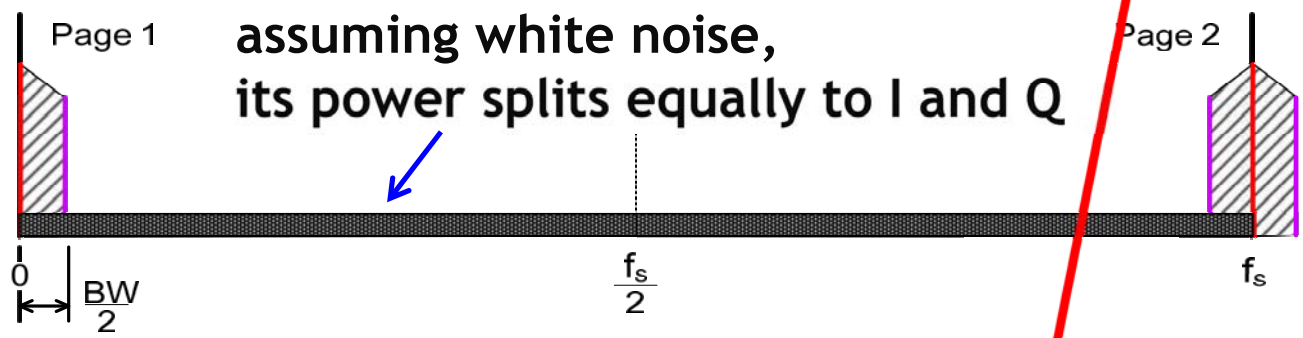
Why oversample ??

Sampling - Decimation

aliased IF spectrum

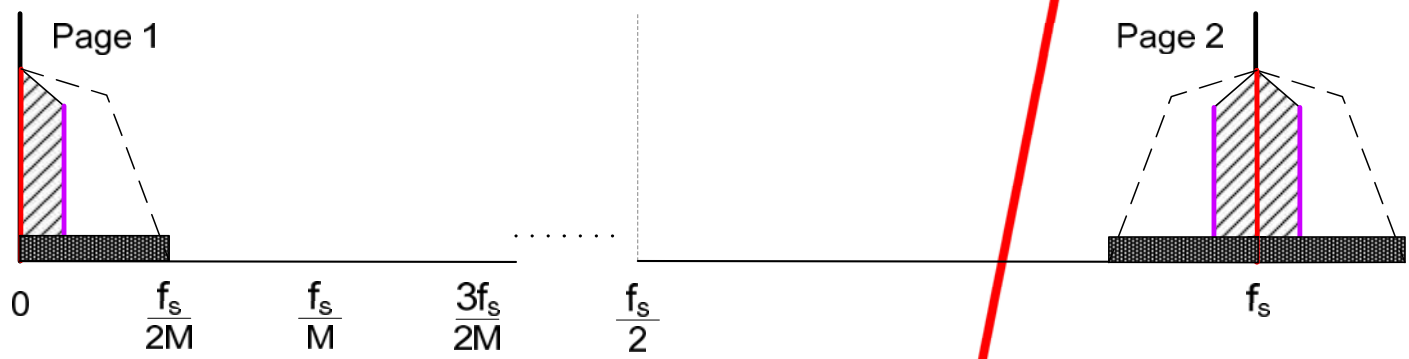


After digital down-conversion (DDC) (single quadrature channel)

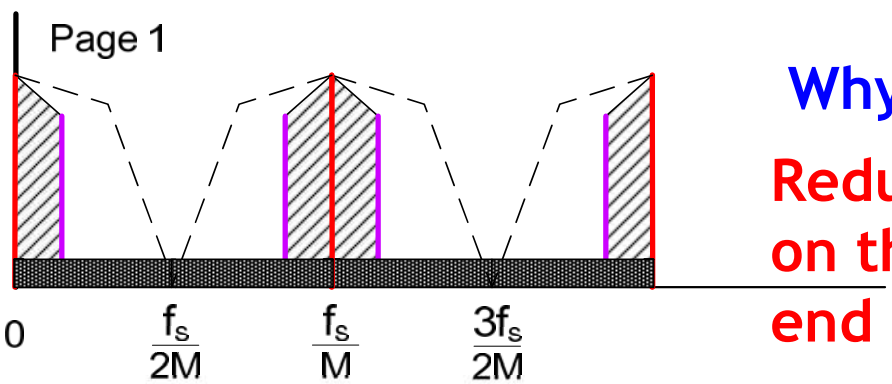


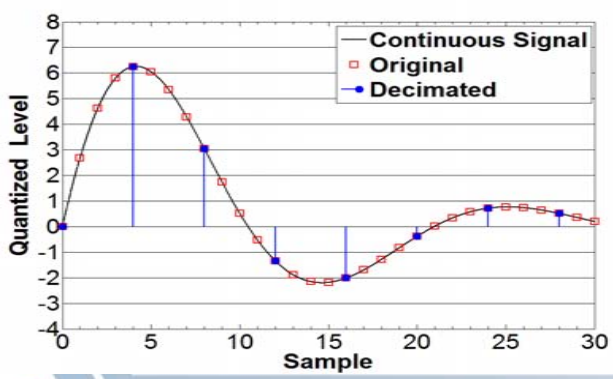
assuming white noise,
its power splits equally to I and Q

filter to prevent aliasing at new sampling rate



Keep every M^{th} sample for new rate



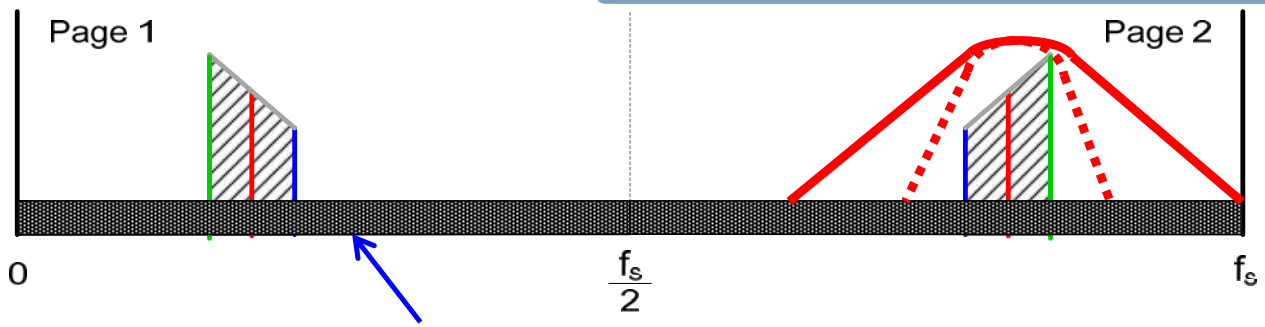


Why oversample ??

Reduces constraints on the analog front-end anti-aliasing filter

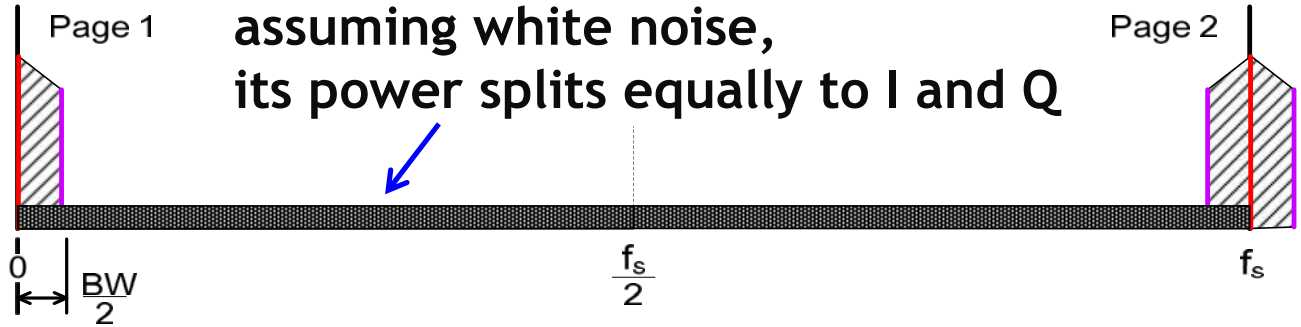
Sampling - Decimation

aliased IF spectrum

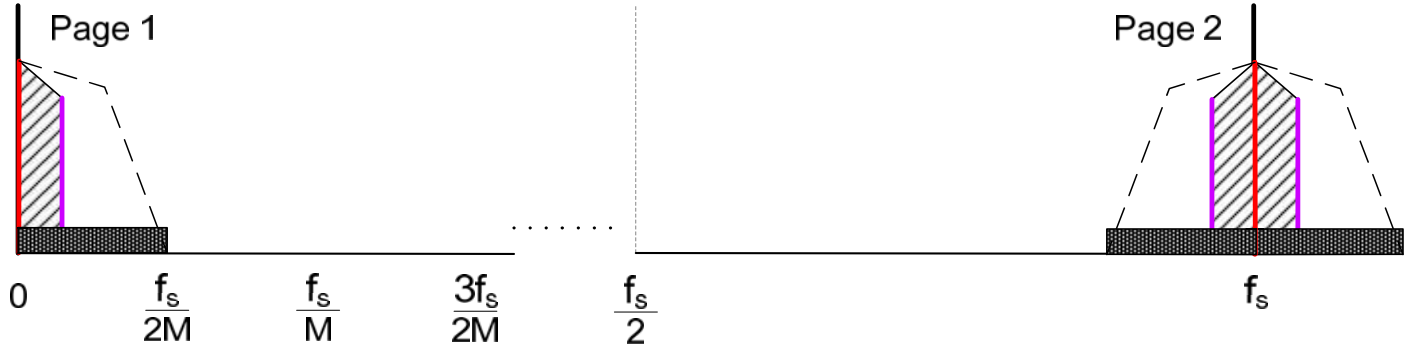


After digital down-conversion (DDC) (single quadrature channel)

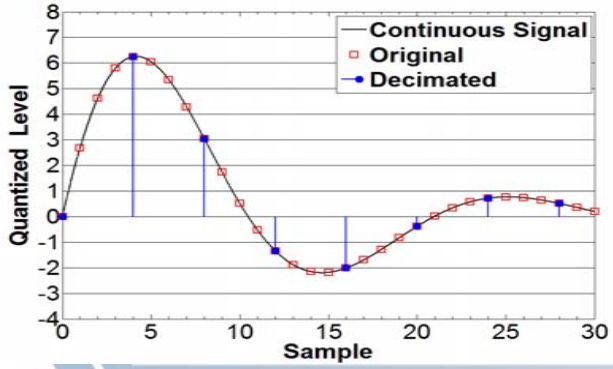
assuming white noise,
its power splits equally to I and Q

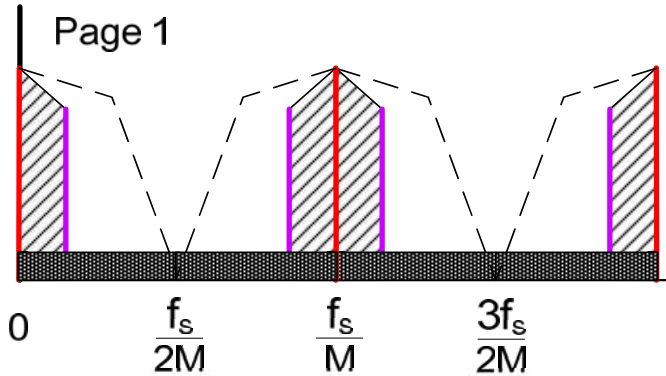


filter to prevent aliasing at new sampling rate



Keep every M^{th} sample for new rate





Why oversample ??

Reduces our system's Signal-to-Noise Ratio (SNR) →

Sampling - ADC Quantization Noise

Signal to Noise Ratio (SNR) over what bandwidth ??

Variance $\sigma^2 = \frac{q^2}{12} = \frac{1}{2^{2N} \cdot 12}$

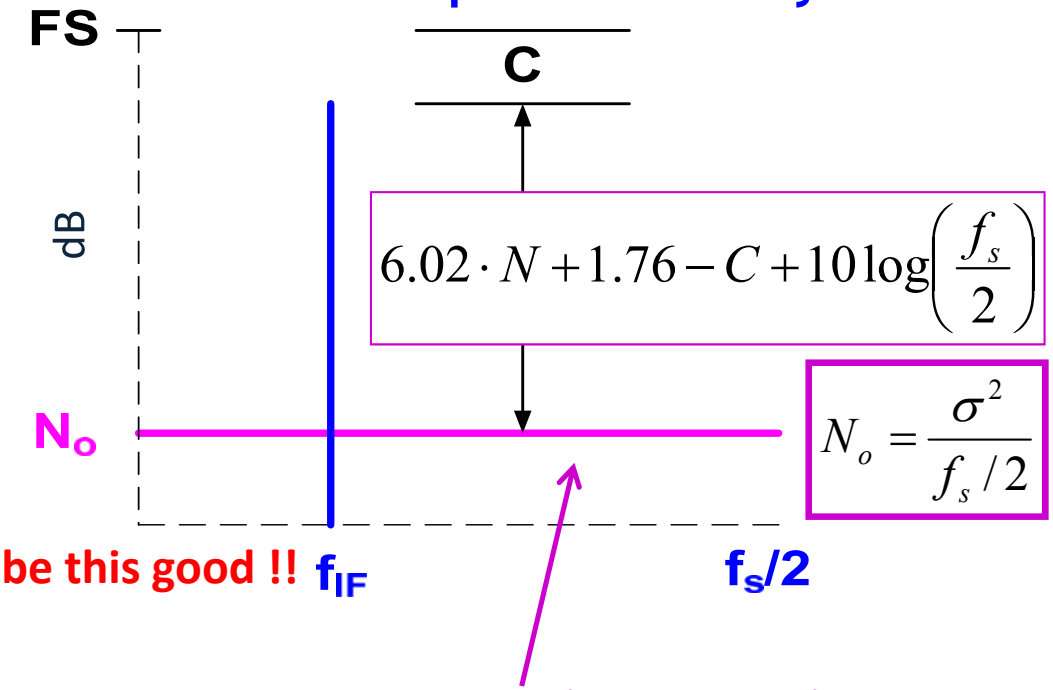
$$SNR = 6.02 \cdot N + 1.76 - C \quad \text{[dB]}$$

N = # of ADC bits

C = dB carrier is below full scale

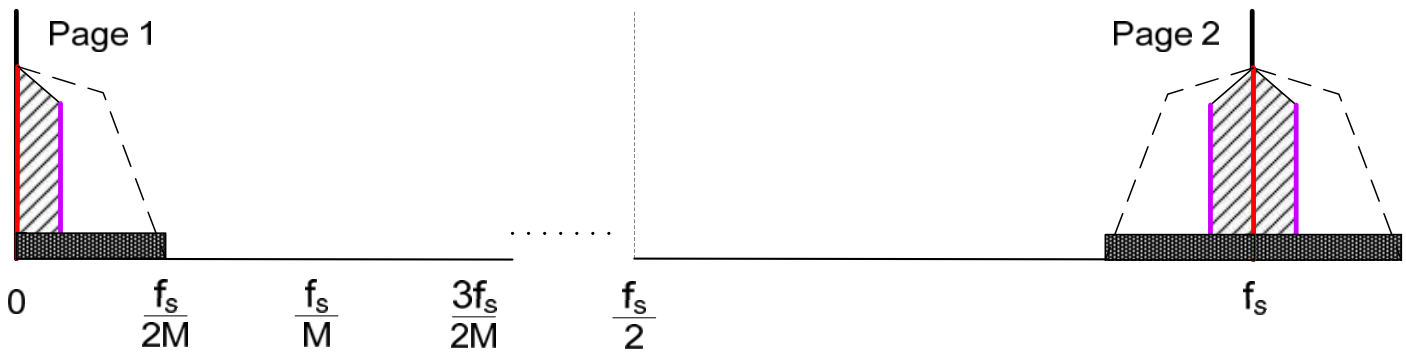
!! Real-world 14-bit and 16-bit SNR's will not be this good !! Look at datasheets

Power Spectral Density



Quantization noise power is spread across the ***Nyquist bandwidth***

Therefore, oversampling increases the SNR over the ***System Bandwidth***



Oversampling

- ***reduces SNR***
- ***relaxes constraints on the analog front-end anti-aliasing filters***

Decimation - reduces downstream computational burden

Sampling - ADC Quantization Noise Impact on Phase Noise

phase noise variance $\sigma_\phi^{\,2} = \int_0^{f_{max}} S_\phi(f)\,df$

14 bit ADC

$\sigma_\phi^{\,2} = -85$ dB rad²

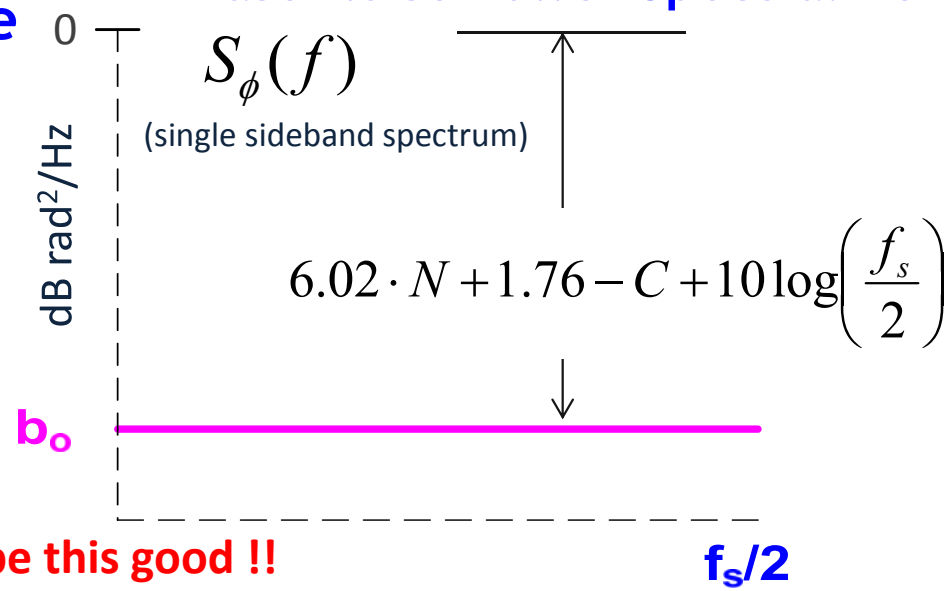
$\sigma_\phi = 0.0032$ deg rms

16 bit ADC

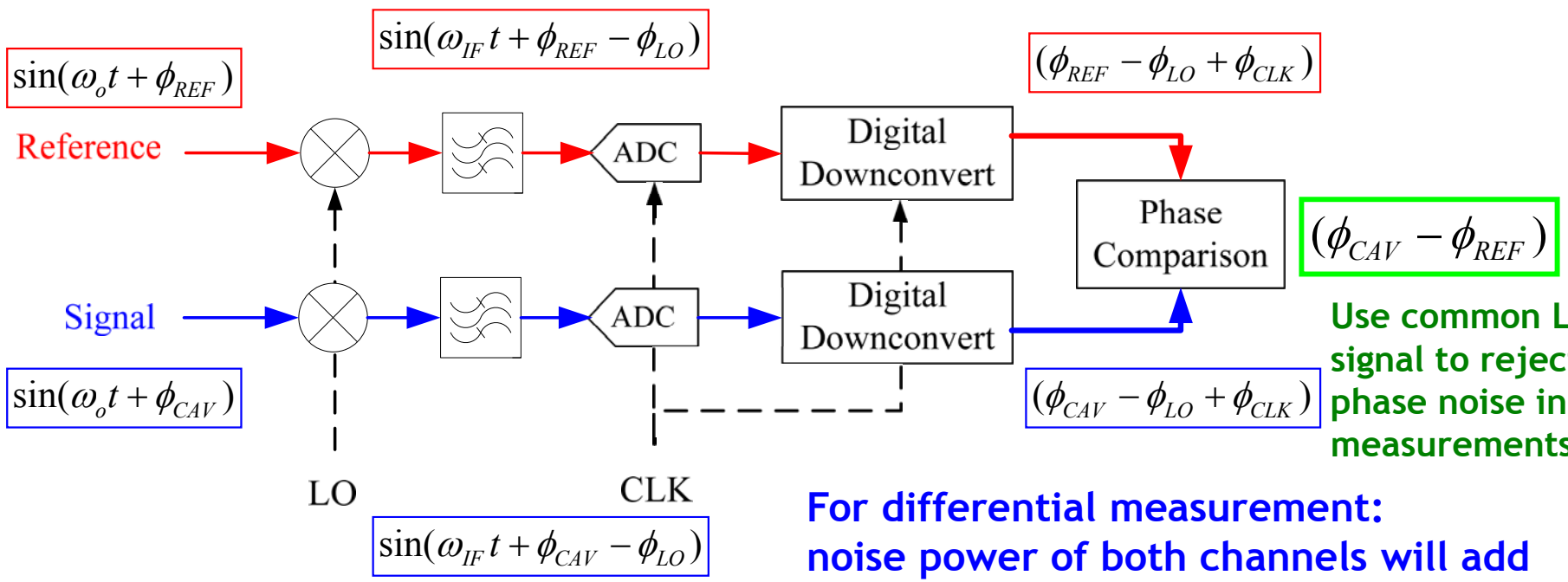
$\sigma_\phi^{\,2} = -97$ dB rad²

$\sigma_\phi = 0.0008$ deg rms

Phase Noise Power Spectral Density



!! Real-world 14-bit and 16-bit SNR's will not be this good !!
Look at datasheets



Use common LO and CLK signal to reject their phase noise in differential measurements

For differential measurement:
noise power of both channels will add

Sampling - ADC Aperture Jitter

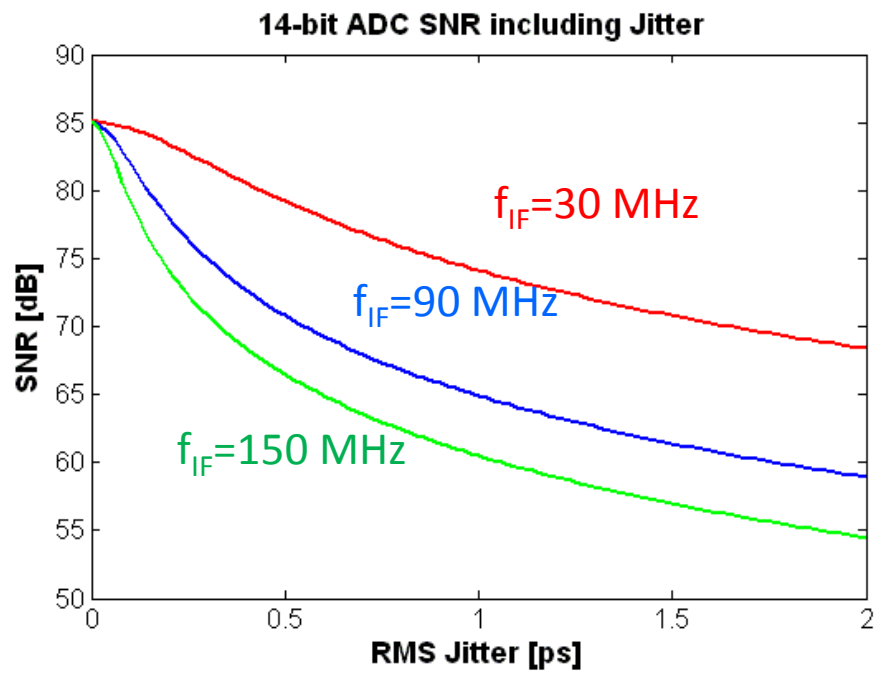
$$\cos(\omega_{IF}(t+\varepsilon)) = \cos(\omega_{IF} t + \omega_{IF}\varepsilon)$$

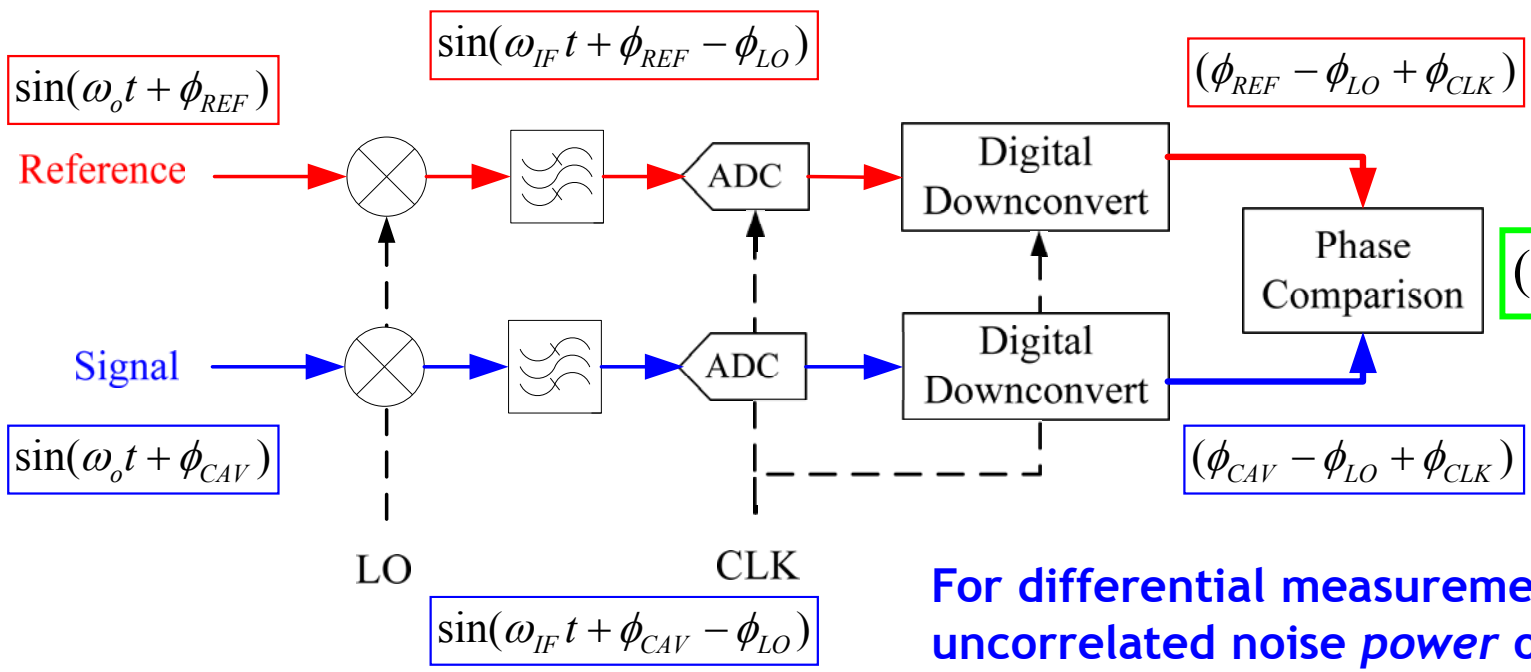
Shows up as Phase noise

Proportional to IF frequency

$$SNR_{jitter} = -20 \cdot \log(\omega_{IF} \cdot \sigma_{jitter}) \quad \text{[dB]}$$

σ_{jitter} rms aperture jitter





Use common LO and CLK signal to reject their phase noise in differential measurements

For differential measurement: uncorrelated noise *power* of both channels add

Sampling - ADC Aperture Jitter

phase noise variance $$\sigma_\phi^{\,2} = \int_0^{f_{max}} S_\phi(f)\,df$$

14 bit ADC, 0.5psec rms

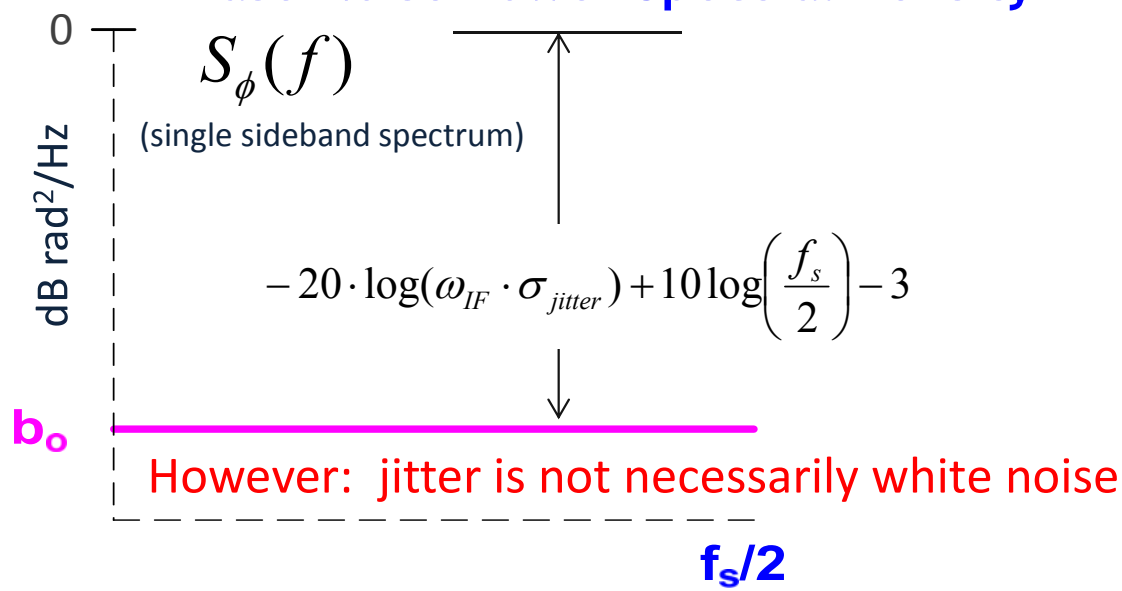
30 MHz IF

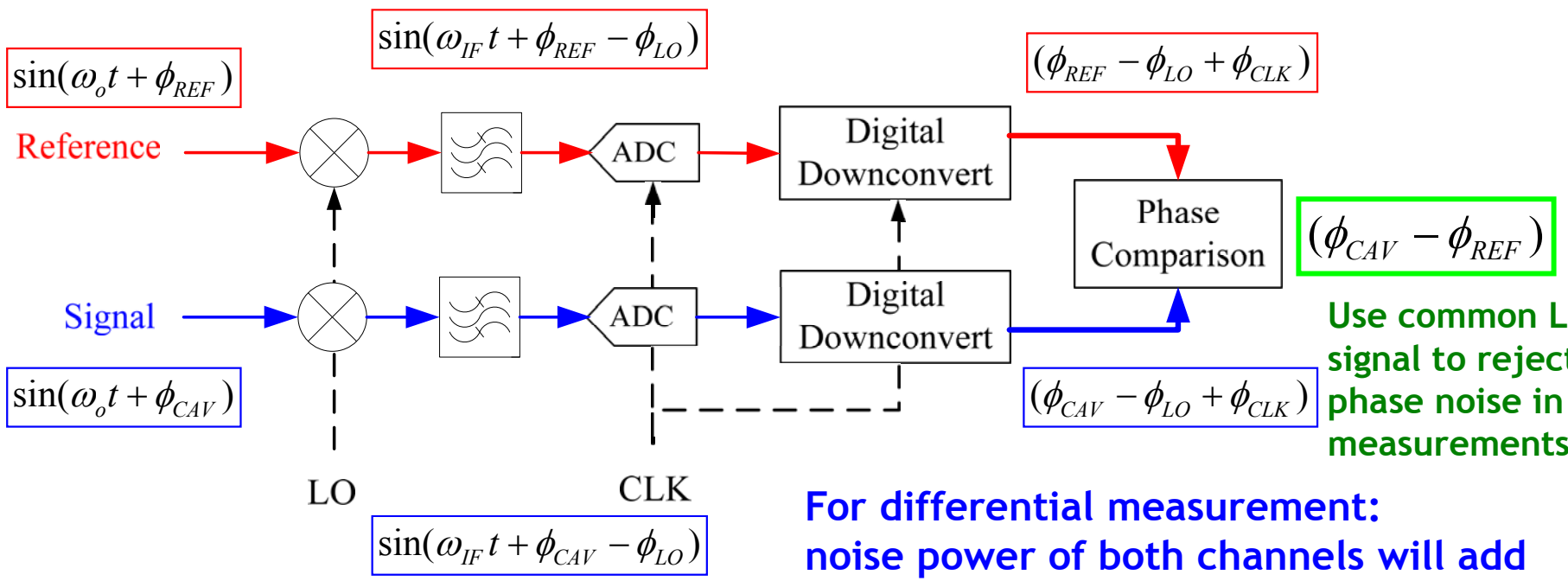
$\sigma_\phi \approx 0.008$ deg rms

90 MHz IF

$\sigma_\phi \approx 0.023$ deg rms

Phase Noise Power Spectral Density



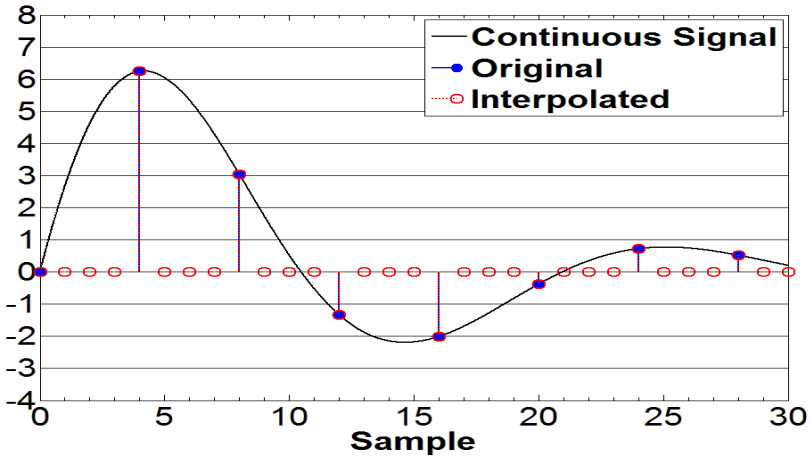


Use common LO and CLK signal to reject their phase noise in differential measurements

For differential measurement: noise power of both channels will add

Sampling - Interpolation

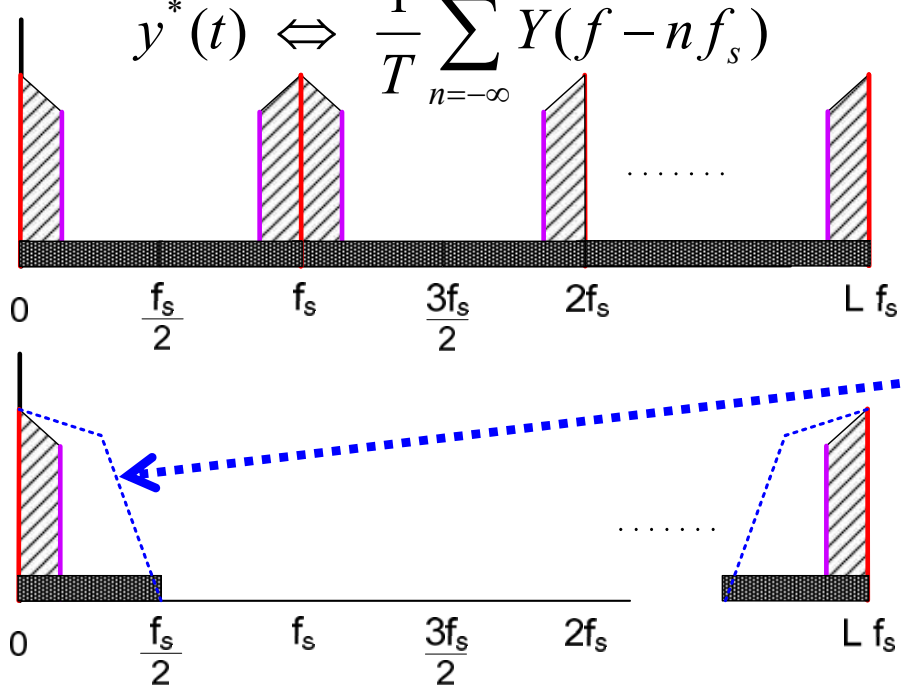
If we want to increase a sample rate to Lf_s , what values to insert? . . . . zeros



$$y^*(t) = y(t)\cdot\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} y(t)\, e^{i\, m\, f_s\, t}$$

The sampled signal is defined to be zero at all times other than $T=1/f_s$

$$y^*(t) \Leftrightarrow \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - n f_s)$$



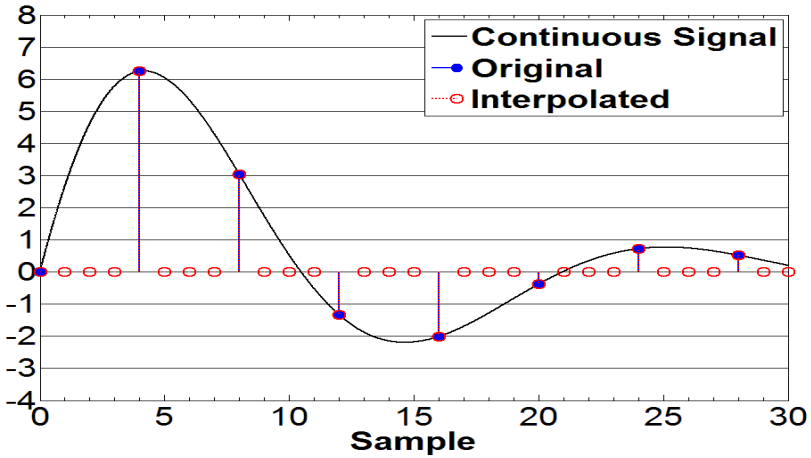
Thus, its spectrum is unchanged if we add zeros in between, and the spectrum repeats at f_s

After adding zeros, use a low-pass filter (LPF) operating at the new rate (Lf_s)

Why interpolate ??

Sampling - Interpolation

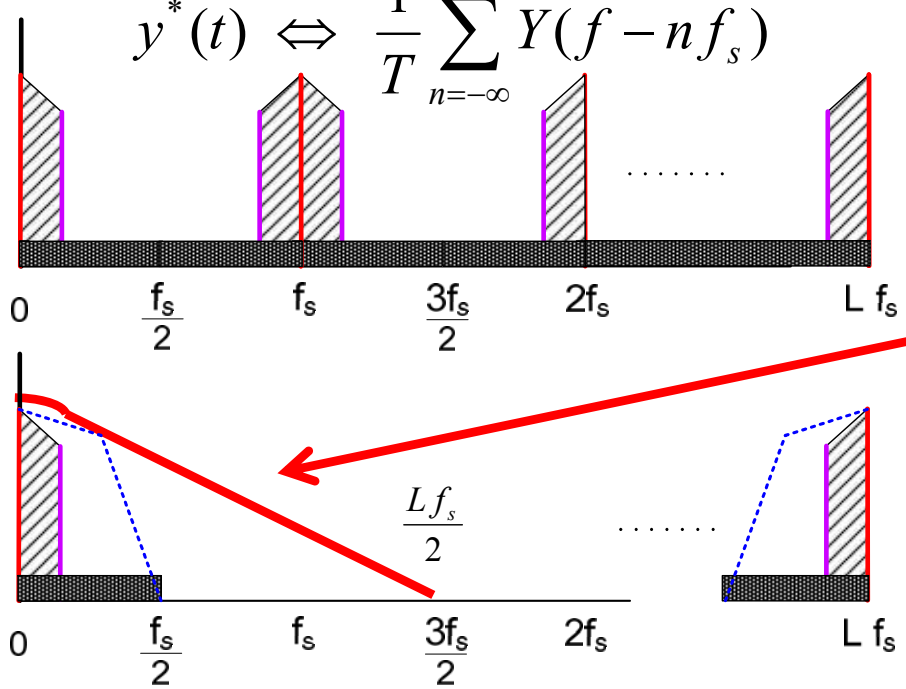
If we want to increase a sample rate to Lf_s , what values to insert? . . . . zeros



$$y^*(t) = y(t)\cdot\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} y(t)\, e^{i\, m\, f_s\, t}$$

The sampled signal is defined to be zero at all times other than $T=1/f_s$

$$y^*(t) \Leftrightarrow \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - n f_s)$$

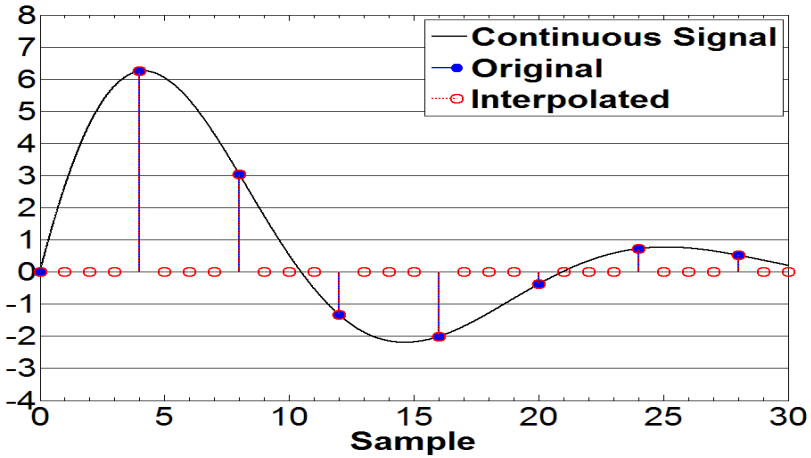


Reduces constraints on the analog reconstruction filter

Why interpolate ??

Sampling - Interpolation

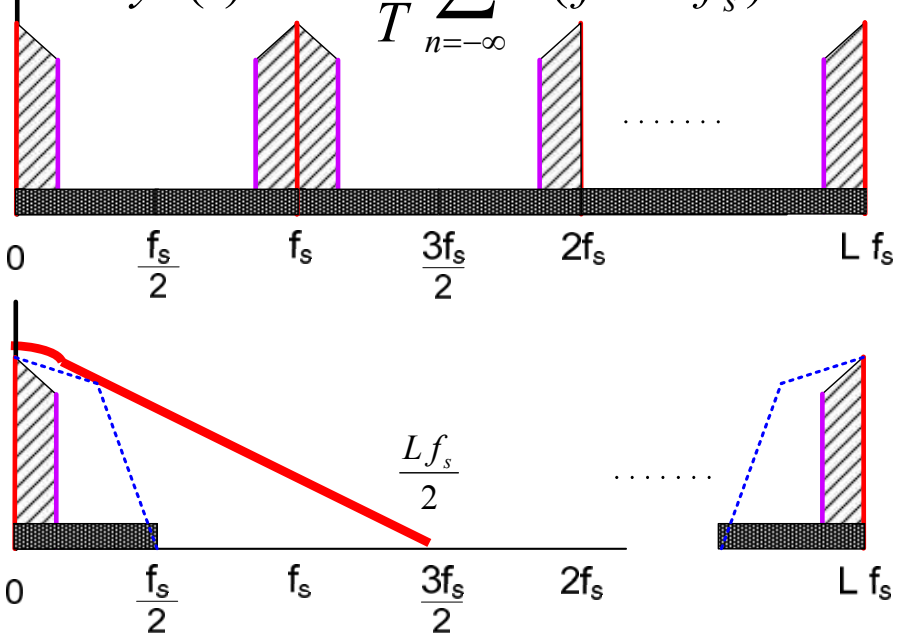
If we want to increase a sample rate to Lf_s , what values to insert? . . . . zeros



$$y^*(t) = y(t)\cdot\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} y(t)\, e^{i\, m\, f_s\, t}$$

The sampled signal is defined to be zero at all times other than $T=1/f_s$

$$y^*(t) \Leftrightarrow \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - n f_s)$$

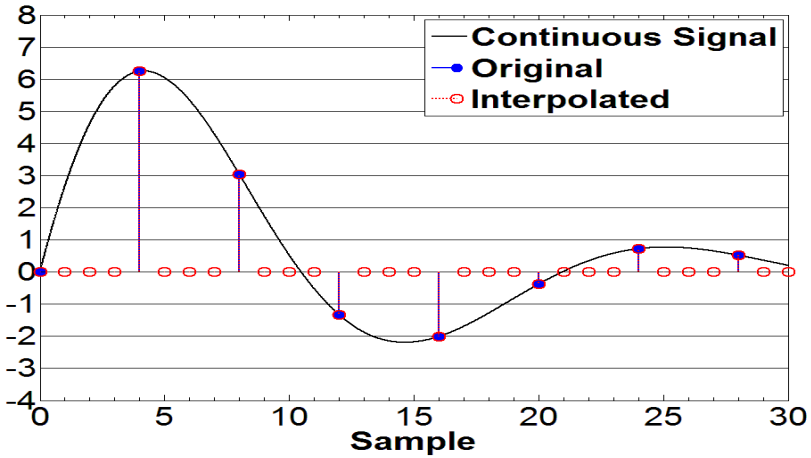


Similare to ADC's, it reduces the output SNR

Why interpolate ??

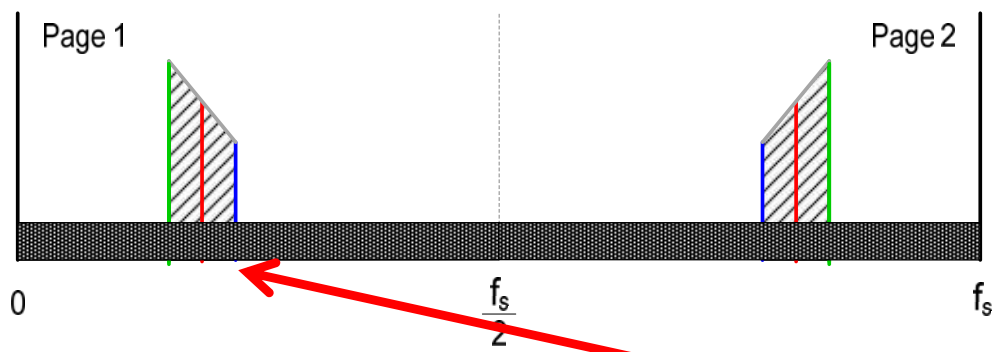
Sampling - Interpolation

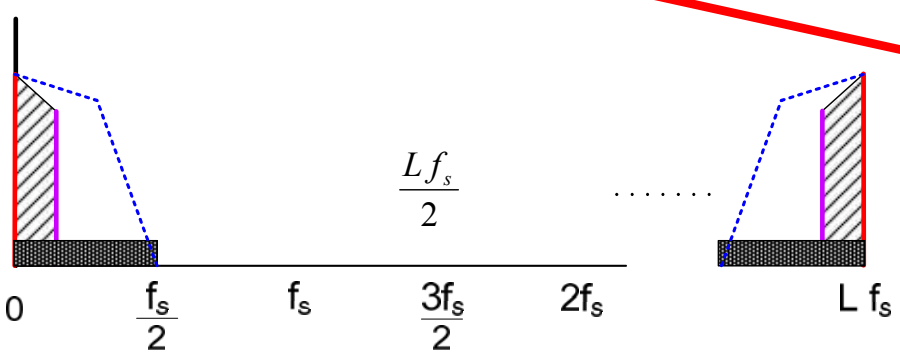
If we want to increase a sample rate to Lf_s , what values to insert? . . . . zeros



$$y^*(t) = y(t)\cdot\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} y(t)\, e^{i\, m\, f_s\, t}$$

The sampled signal is defined to be zero at all times other than $T=1/f_s$



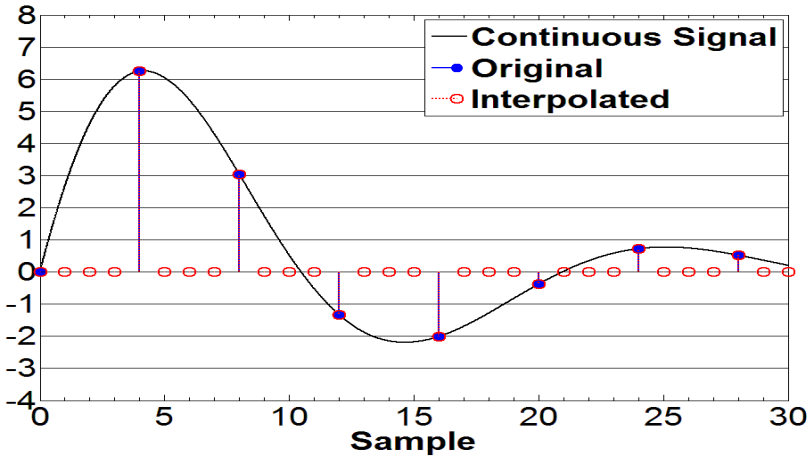


Eventually need to up-convert back to our digital IF frequency

Why interpolate ??

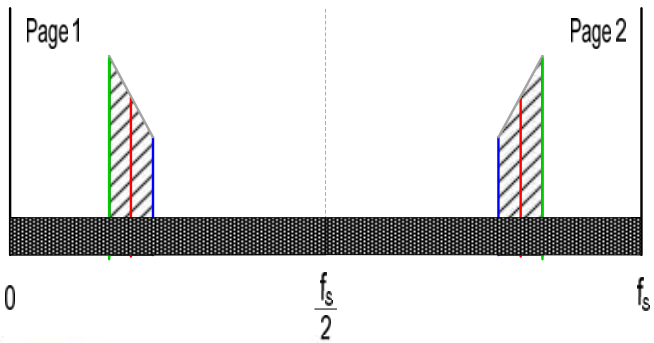
Sampling - Interpolation

If we want to increase a sample rate to Lf_s , what values to insert? . . . . zeros



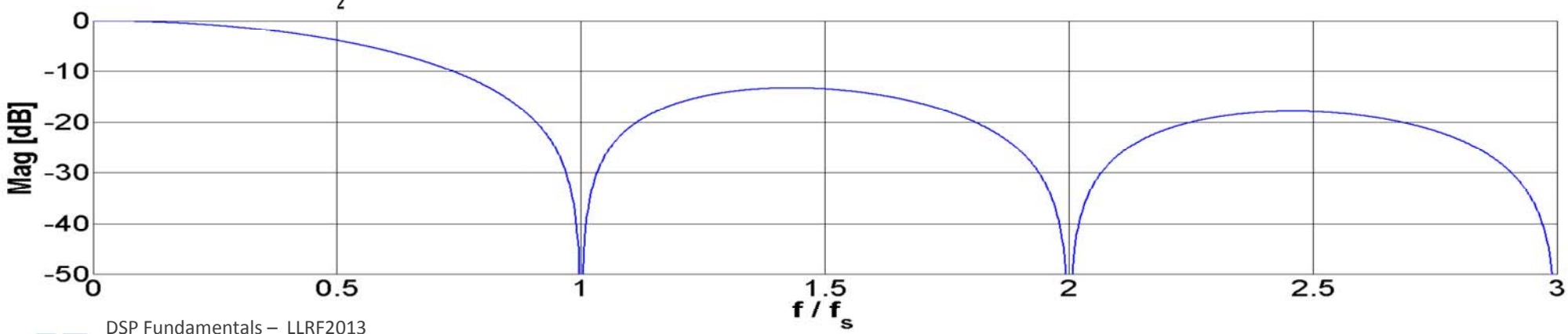
$$y^*(t) = y(t)\cdot\delta_T(t) = \frac{1}{T}\sum_{m=-\infty}^{\infty} y(t)\, e^{i\, m\, f_s\, t}$$

The sampled signal is defined to be zero at all times other than $T=1/f_s$



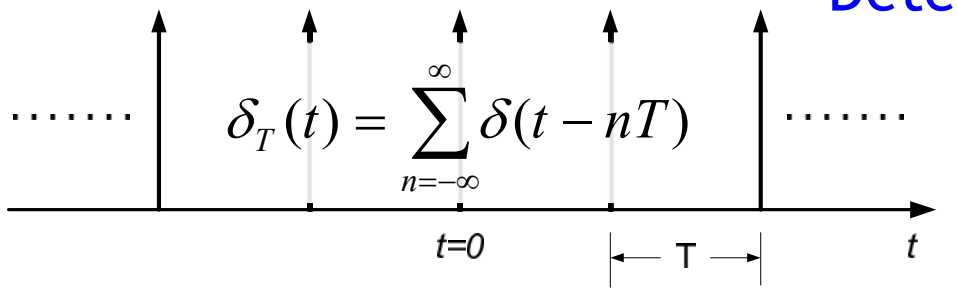
Why interpolate ??

Zero-order hold of the output DAC corresponds to time-domain convolution with a rectangular pulse which corresponds to multiplication in the frequency domain with a sinc function



Transforms

Transforms

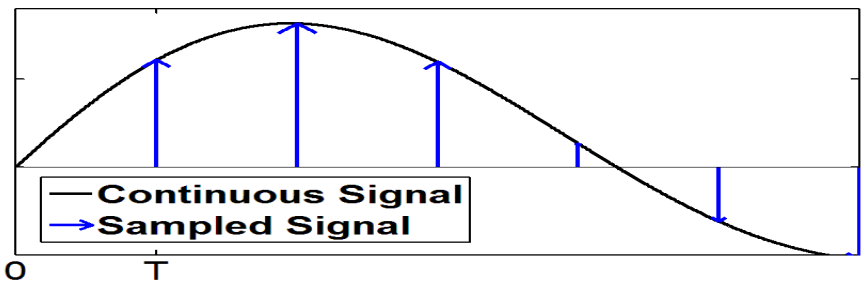


Determine an explicit formula for the DTFT

$$Y^*(f) = \frac{1}{T}\sum_{n=-\infty}^{\infty} Y(f - nf_s) \quad = ?$$

This time, don't use Fourier series for $\delta_T(t)$

$$\boxed{y^*(t) = y(t)\cdot\delta_T(t) = \sum_{k=-\infty}^{\infty} y(kT)\,\delta(t-kT)}$$



$$\delta(t) \overset{\mathscr{F}}{\Leftrightarrow} 1$$

delay property: $x(t-t_o) \Leftrightarrow e^{-i2\pi f t_o} X(f)$

$$\Rightarrow \boxed{y^*(t) \overset{\mathscr{F}}{\Leftrightarrow} Y^*(f) = \sum_{k=-\infty}^{\infty} y(kT)e^{-i2\pi f kT}}$$

Continuous function of frequency

Discrete-Time Fourier Transform (DTFT)

$$\boxed{IDTFT = \frac{1}{f_s}\int_0^{f_s} Y^*(f)e^{i2\pi f kT}\,df}$$

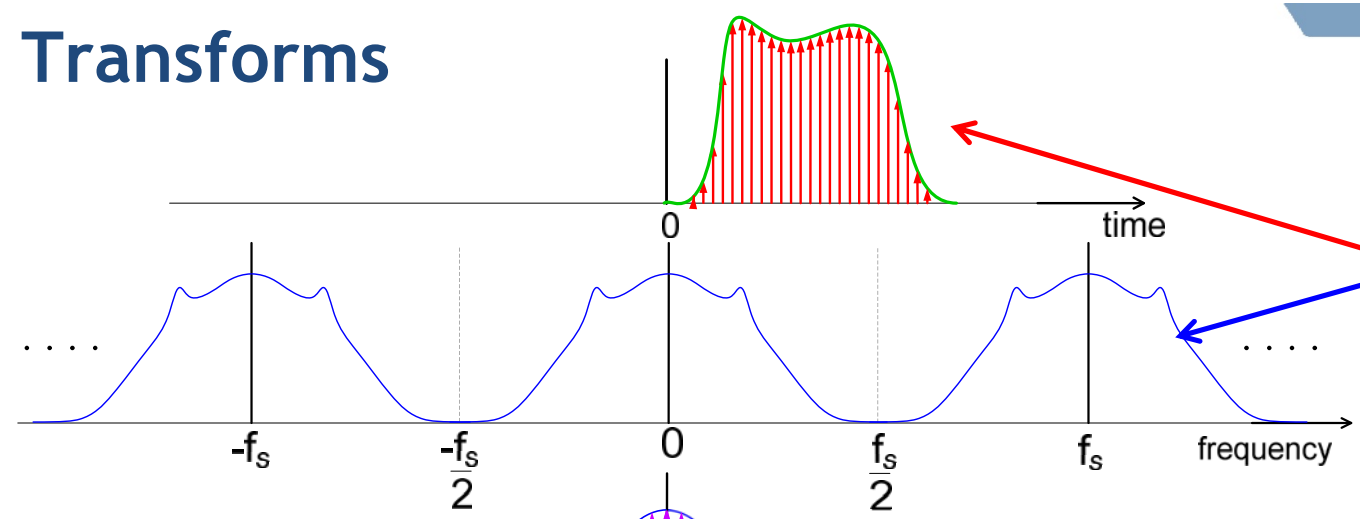
only need to integrate over 1 period since Y*(f) is periodic

Comparing DTFT to previous formula:

$$\boxed{\sum_{n=-\infty}^{\infty} Y(f - nf_s) = T\cdot\sum_{k=-\infty}^{\infty} y(kT)e^{-i2\pi f kT}}$$

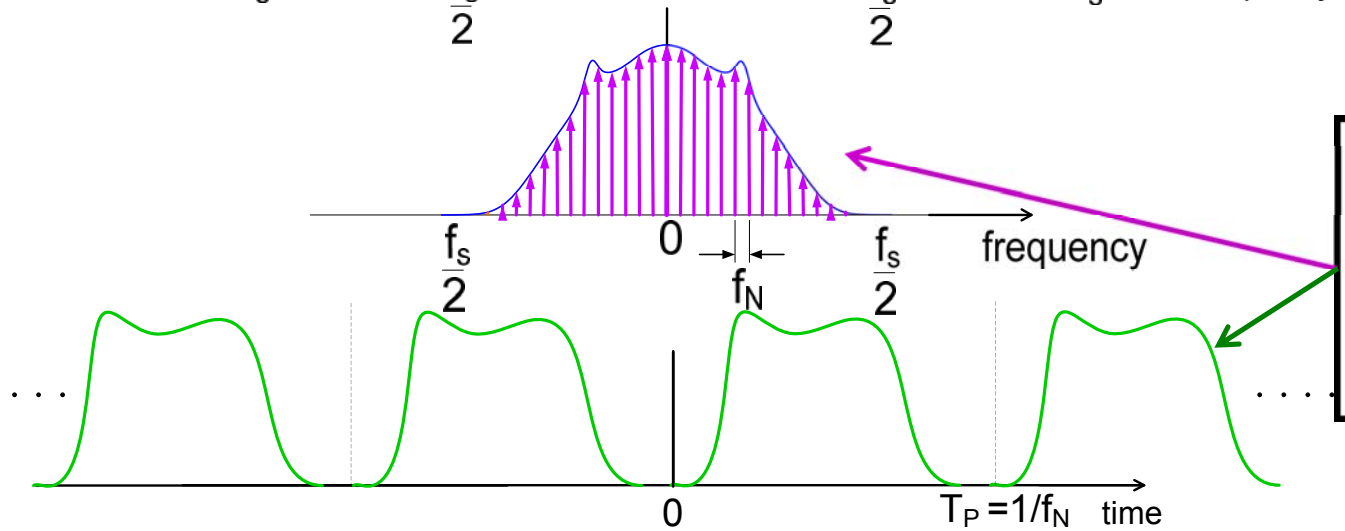
This is a *Poisson Summation formula*: *The periodic summation of a function's Fourier transform is given by discrete samples of that function*

Transforms



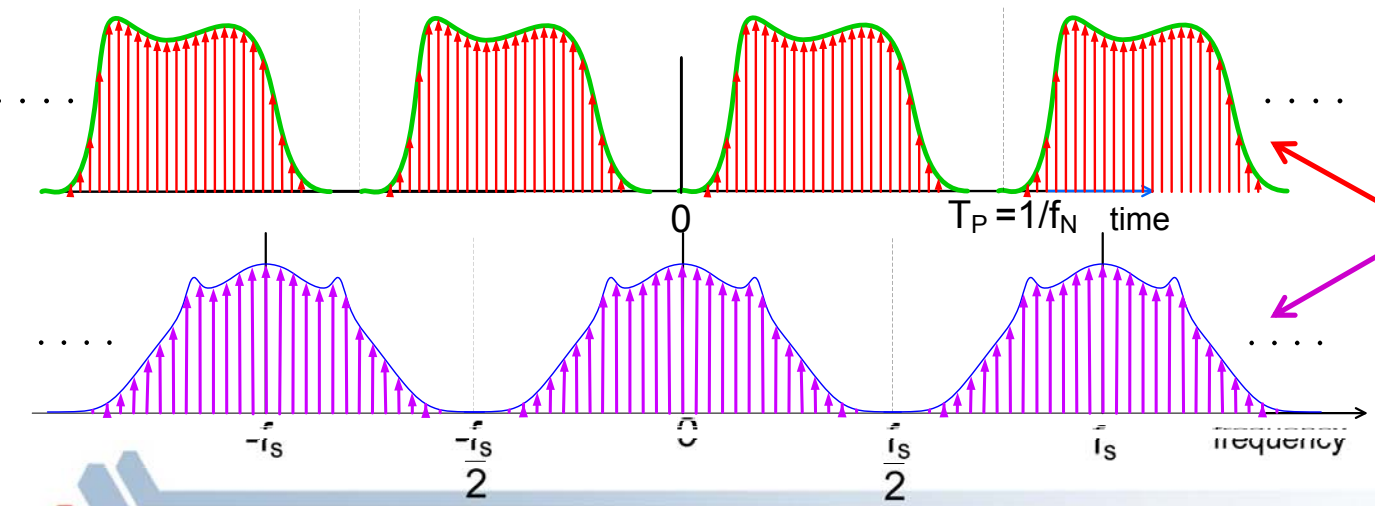
The periodic summation of a function's Fourier transform is given by discrete samples of that function

Can lead to freq aliasing



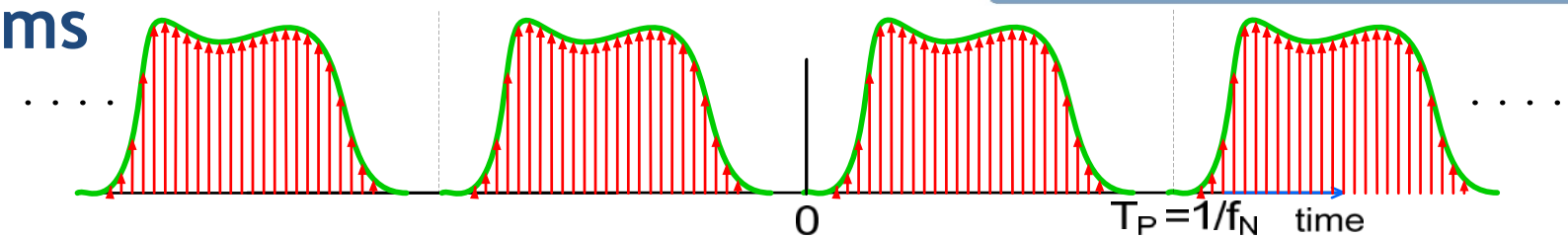
By duality, the periodic summation of a time function is given by discrete samples of its Fourier Transform

Can lead to time aliasing



Again, the periodic summation of a function's Fourier transform is given by discrete samples of that function

Transforms



Consider a K-periodic sequence discrete-time signal: $\boxed{y(t+KT)\,\delta_T(t) = y(t)\,\delta_T(t)}$

Fourier Series coefficients are given as:

$$\alpha_n = \frac{1}{KT}\int_0^{KT} y(t)\delta(t-kT)e^{-i\frac{2\pi n}{KT}t}\,dt \qquad \boxed{\alpha_n = \frac{1}{KT}\sum_{k=0}^{K-1} y(kT)e^{-i\frac{2\pi n}{K}k}} \qquad \boxed{\alpha_{n+mK} = \alpha_n}$$

The coefficients are also K periodic

Sample the DTFT and note that we only need K-samples to get a value at f:

$$Y^*(f)\Big|_{f=n\frac{f_s}{K}=\frac{n}{KT}} = \sum_{k=-\infty}^{\infty} y(kT)e^{-i2\pi\frac{n}{K}k} \qquad y((k+K)T)e^{-i2\pi\frac{n}{K}(k+K)} = y(kT)e^{-i2\pi\frac{n}{K}k}$$

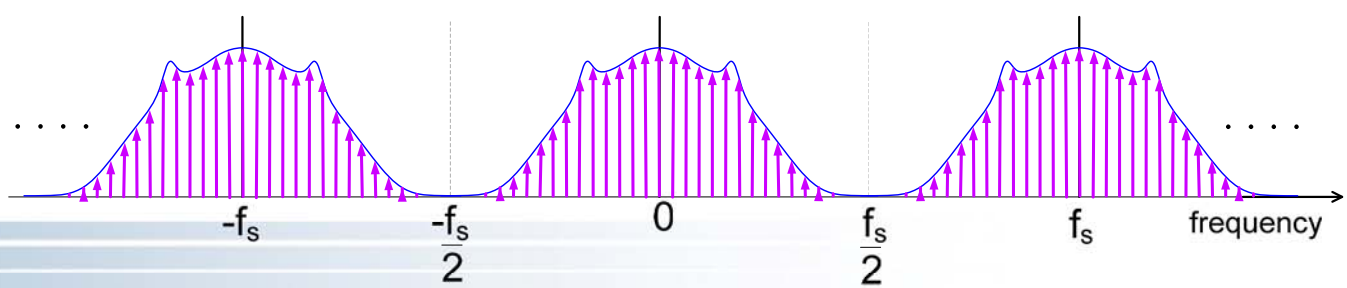
$$\Rightarrow \quad \boxed{Y_n = \sum_{k=0}^{K-1} y[k]e^{-i2\pi\frac{n}{K}k}}$$

Discrete Fourier Transform

A summation of K is sufficient.
Terms are consistent with the Fourier Series coefficients.

$$\boxed{y[k] = \frac{1}{K}\sum_{n=0}^{K-1} Y_n\, e^{i2\pi\frac{n}{K}k}}$$

Inverse Discrete Fourier Transform



Transforms - Discrete Fourier Transform

- **Caution:** While convolution in the time domain transforms to multiplication of Fourier/LaPlace transforms in the frequency domain**, multiplication of DFT's results in *circular convolution* as opposed to linear convolution. This can be understood as the result of the DFT's representing periodic signals. Linear convolution can be obtained from circular convolution by appropriate zero-padding.**

- **Fast Fourier Transform is a computationally efficient method for the DFT**

- **What is it used for??**
  - Filtering entirely in the frequency domain (FFT => freq shape => IFFT)
  - Designing digital filters via the Frequency Sampling method to obtain a frequency response equal to specified values at specific frequency sample points
  - For signal analysis – i.e., analyzing amplitude and phase noise, Caution: watch out for windowing/spectral leakage effects
  - FFT filter banks
  - Compression algorithms
  - Polyphase Channelizers

Transforms

Fourier Transform

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{-i2\pi f t} dt \quad \Leftrightarrow \quad y(t) = \int_{-\infty}^{\infty} Y(f) e^{i2\pi f t} df$$

DTFT
Discrete-Time
Fourier Transform

$$Y^*(f) = \sum_{k=-\infty}^{\infty} y(kT) e^{-i2\pi f kT} \Leftrightarrow y(kT) = \frac{1}{f_s} \int_0^{f_s} Y^*(f) e^{i2\pi f kT} df$$

$f_N = \frac{f}{f_s} = fT$ Typically substitute a normalized freq. to generalize independent of T

DFT
Discrete
Fourier Transform

$$Y_n = \sum_{k=0}^{K-1} y(kT) e^{-i\,2\pi \frac{n}{N} k} \Leftrightarrow y(kT) = \frac{1}{K} \sum_{n=0}^{K-1} Y_n e^{i\,2\pi \frac{n}{N} k}$$

Laplace Transform

$$Y(s) = \int_{0^-}^{\infty} y(t) e^{-st} dt \quad \Leftrightarrow \quad y(t) = \frac{1}{i\,2\pi} \int_{c-i\infty}^{c+i\infty} Y(s) e^{st} ds$$

z-Transform

$$\mathrm{Y}_z(z) = \sum_{k=0}^{\infty} y[k] z^{-k} \quad \Leftrightarrow \quad y[k] = \frac{1}{i\,2\pi} \oint_C \mathrm{Y}_z(z) z^{k-1} dz$$

Just as the ***DTFT is the discrete-time equivalent of the Fourier transform***, the ***z-Transform is the discrete-time equivalent of the Laplace transform***. It can be derived in a similar fashion to the DTFT making the substitution $\boldsymbol{z=e^{sT}}$

(or $z=e^s$ when using normalized frequency)

z-Transform

$$\mathrm{Y}_z(z) = \sum_{k=0}^{\infty} y[k] z^{-k} \quad \overset{\mathcal{Z}}{\Leftrightarrow} \quad y[k] = \frac{1}{i\,2\pi} \oint_C \mathrm{Y}_z(z) z^{k-1} dz$$

The inverse transform is derived from complex analysis by recognizing $Y_z(z)$ as a Laurent series. The closed contour must lie within the region of convergence (ROC) of F(z). For further details, see texts on complex analysis and transforms.

DTFT

$$\mathrm{Y}_z(e^{i2\pi f_N}) = \sum_{k=0}^{\infty} y[k] e^{-i2\pi f_N k}$$

When the unit circle lies within the (ROC), which is the case for stable $Y_z(z)$, then the DTFT and IDTFT are given by the z-Transform and inverse z-Transform evaluated along the unit circle, $z=e^{i2\pi f}$.

$$y[k] = \frac{1}{i\,2\pi} \oint_C \mathrm{Y}_z(e^{i2\pi f_N}) e^{i2\pi f_N (k-1)} dz = \int_0^1 \mathrm{Y}_z(e^{i2\pi f_N}) e^{i2\pi f_N k} df_N$$

IDTFT

Just as the ***Derivative Property*** of the **Laplace transform** is exploited to solve ***differential equations***,

$$y^{(n)} \overset{\mathcal{L}}{\Leftrightarrow} s^n Y(s) - \sum_{k=1}^{n-1} y^{(n-k)}(0)$$

the ***Shift Properties*** of the **z-transform** are exploited to solve ***difference equations***.

Right Shift: $$y[k-m] \overset{\mathcal{Z}}{\Leftrightarrow} z^{-m} Y_z(z) + \sum_{k=1}^{m} y[-k] z^{k-m}$$

Left Shift: $$y[k+m] \Leftrightarrow z^{m} Y_z(z) - \sum_{k=0}^{m-1} y[k] z^{m-k}$$

Digital Filters

Digital Filters

Consider an nth order difference equation for a single-input single-output (SISO) sys.

$$\boxed{y(k) = \sum_{j=1}^{n} a_j y(k-j) + \sum_{j=0}^{m} b_j u(k-j)}$$

$a_j = 0 \; \forall \; j \quad \Longrightarrow$ Non-recursive filter, i.e., no previous output values used

Otherwise, it's a recursive filter

Finite Impulse Response (FIR) filters can be implemented in recursive form, thus FIR and non-recursive are NOT synonymous.

Infinite Impulse Response (IIR) filters are *almost* synonymous with recursive filters (i.e., recursive filters can be FIR)

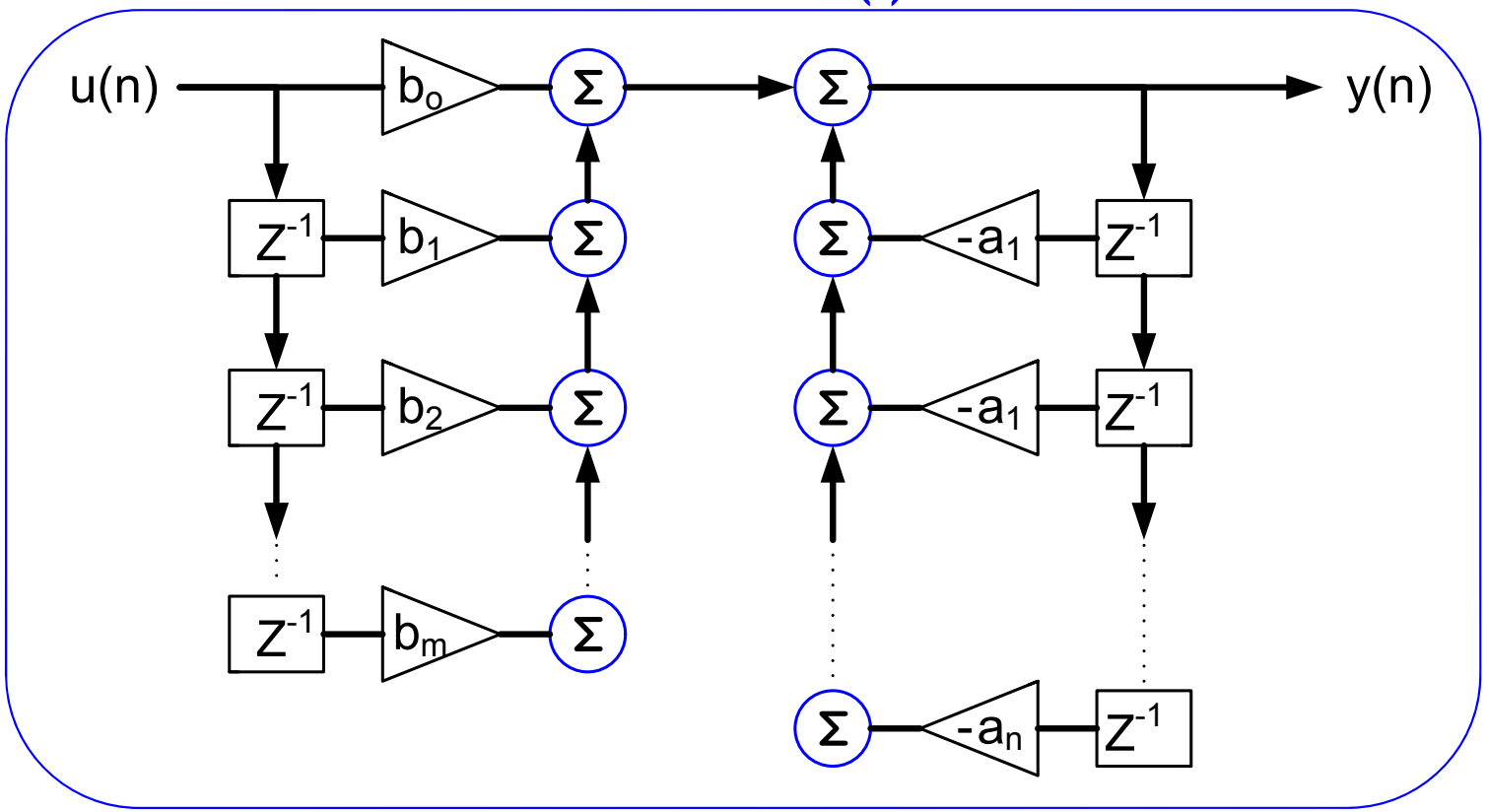
- ***IIR* does not imply instability**, a stable IIR asymptotically approaches zero (i.e., exponential decay response)

Digital Filters

Consider an nth order single-input single-output (SISO) system:

$$y(k) = \sum_{j=1}^{n} a_j y(k-j) + \sum_{j=0}^{m} b_j u(k-j)$$

Direct Form (I)



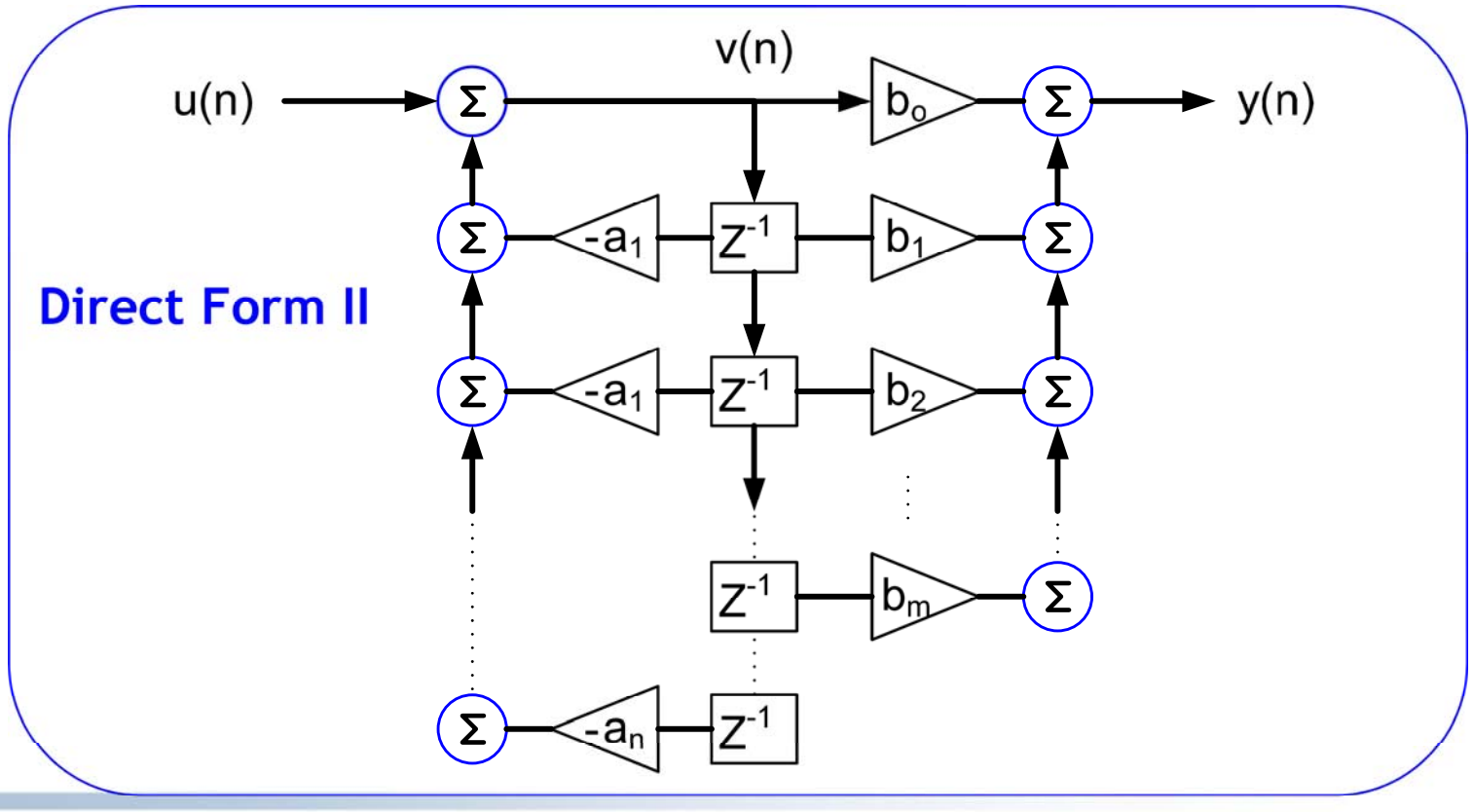
Digital Filters

Consider an nth order single-input single-output (SISO) system:

$$\boxed{y(k) = \sum_{j=1}^{n} a_j y(k-j) + \sum_{j=0}^{m} b_j u(k-j)} \overset{\mathcal{Z}}{\Leftrightarrow} \boxed{Y_Z(z) = \sum_{j=1}^{n} a_j z^{-j} Y_Z(z) + \sum_{j=0}^{m} b_j z^{-j} U_Z(z)}$$

(assuming zero initial conditions)

$$\Rightarrow \quad Y_Z(z) = \frac{\sum_{j=1}^{m} b_j z^{-j}}{1 - \sum_{j=1}^{n} a_j z^{-j}} \cdot U_Z(z) = \left(\frac{V_Z(z)}{U_Z(z)} U_Z(z) \right) \cdot \frac{Y_Z(z)}{V_Z(z)} = \left(\frac{1}{1 - \sum_{j=1}^{n} a_j z^{-j}} \right) U_Z(z) \cdot \left(\sum_{j=1}^{m} b_j z^{-j} \right)$$



Digital Filters

Consider an nth order single-input single-output (SISO) system:

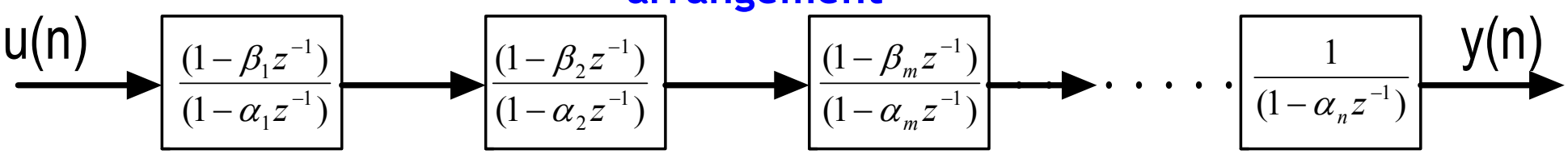
$$y(k) = \sum_{j=1}^{n} a_j y(k-j) + \sum_{j=0}^{m} b_j u(k-j) \overset{\mathcal{Z}}{\Leftrightarrow} Y_Z(z) = \sum_{j=1}^{n} a_j z^{-j} Y_Z(z) + \sum_{j=0}^{m} b_j z^{-j} U_Z(z)$$

(assuming zero initial conditions)

$$\Rightarrow \frac{Y_Z(z)}{U_Z(z)} = \frac{K \prod_{j=1}^{m} (1-\beta_j z^{-1})}{\prod_{j=1}^{n} (1-\alpha_j z^{-1})}$$

Products of sequences

Series/Cascade arrangement



Digital Filters

Consider an nth order single-input single-output (SISO) system:

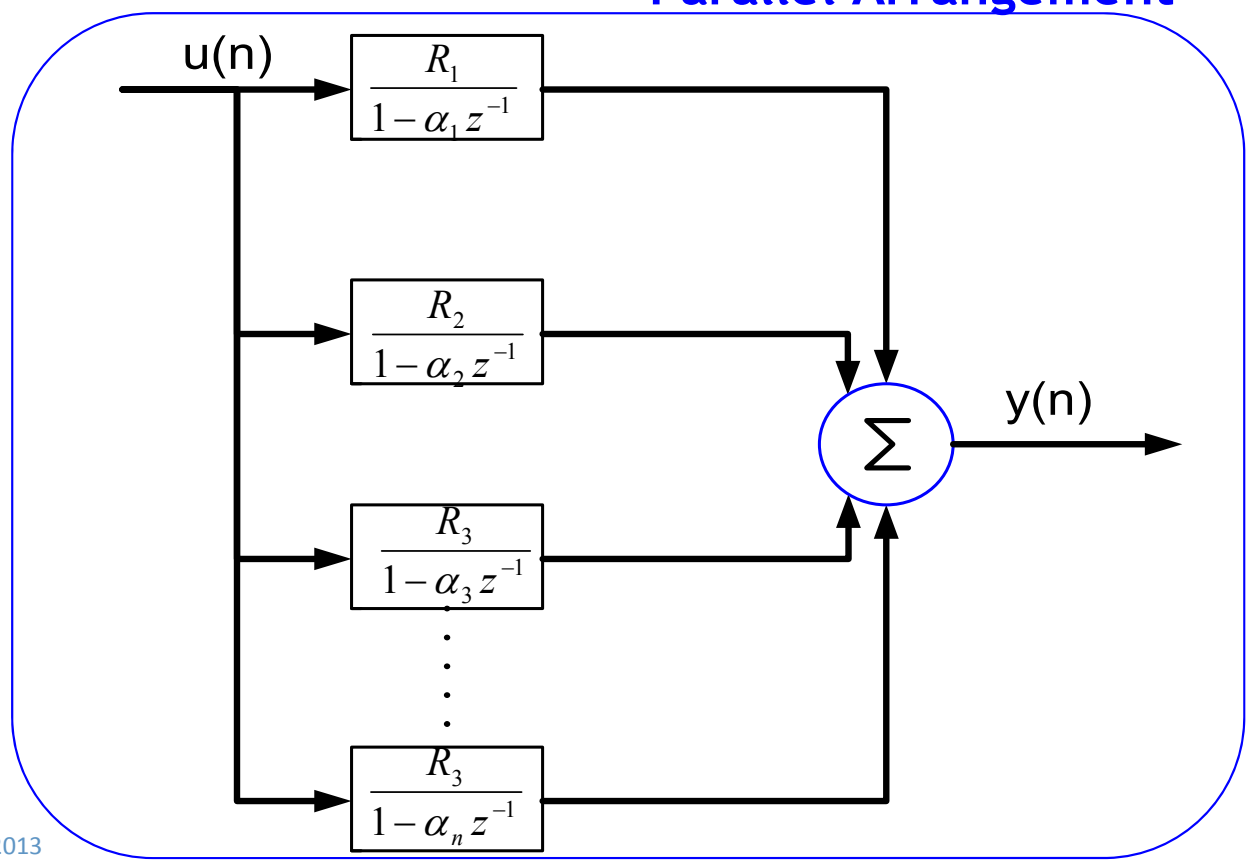
$$\boxed{y(k) = \sum_{j=1}^{n} a_j y(k-j) + \sum_{j=0}^{m} b_j u(k-j)} \overset{\mathcal{Z}}{\Leftrightarrow} \boxed{Y_Z(z) = \sum_{j=1}^{n} a_j z^{-j} Y_Z(z) + \sum_{j=0}^{m} b_j z^{-j} U_Z(z)}$$

(assuming zero initial conditions)

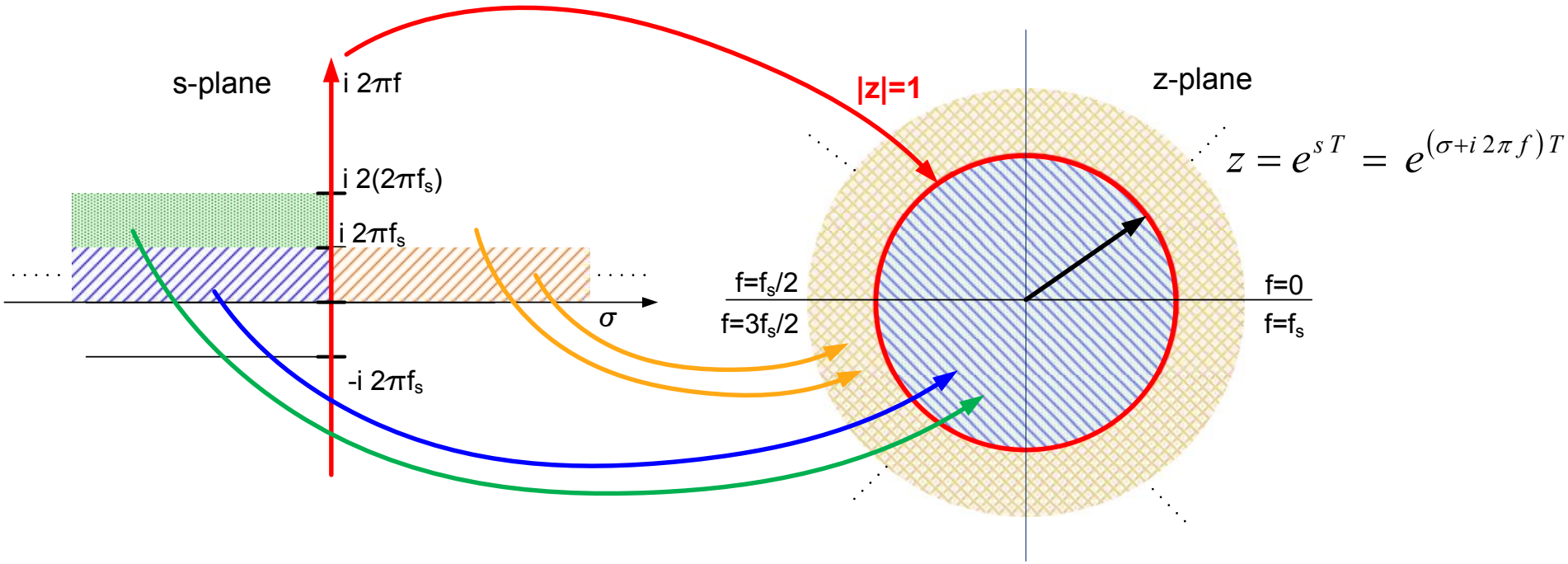
$$\Rightarrow \frac{Y_Z(z)}{U_Z(z)} = \sum_{j=1}^{n} \frac{R_i}{1-\alpha_j z^{-1}}$$

Partial Fraction Expansion

Parallel Arrangement



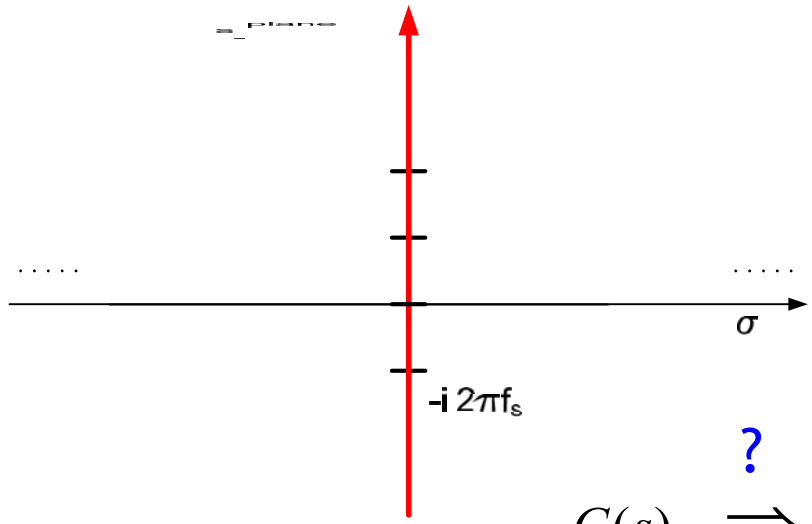
Digital Filters - mappings

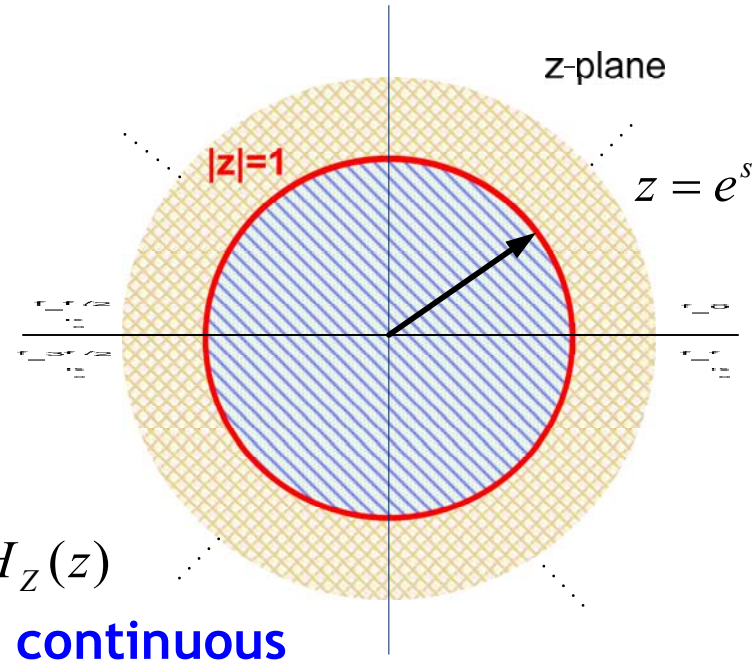


It is not a one-to-one mapping

Stability requires all the poles of the transfer function to be within the unit circle |z|<1

Digital Filters - mappings





$$z = e^{sT} = e^{(\sigma + i\,2\pi f)T}$$

?

$$G(s) \Rightarrow H_Z(z)$$

Transformation from continuous filter design to a digital filter

Numerical approximation of differential equations

Numerical approx. to integration:

Euler's Forward Method: $s = \dfrac{z-1}{T}$

Euler's Backward Method: $s = \dfrac{z-1}{Tz}$

Tustin's bilinear transform: $s = \dfrac{2(z-1)}{T(z+1)}$

Numerical approx. to differentiation:

Forward difference: $s = \dfrac{z-1}{T}$

Backward difference: $s = \dfrac{z-1}{Tz}$

Central difference: $s = \dfrac{z^2-1}{2Tz}$

Digital Filters

Some design methods:

- Design Continuous Filter and the use a numerical approximation mapping
  - frequency pre-warping can also be used on the continuous filter before conversion
- Impulse Invariant Method
  - Uses the z-transform definition as a contour integration (summation of residues) resulting from the LaPlace transform of a sampled signal
- Matched z-Transform
  - Uses direct definition of z to map from s, but with care taken for zeros of transfer function
- Direct Design for i.e., arbitrary magnitude response
  - Uses computer aided optimization techniques
- FIR Filters
  - Fourier Series Method
  - sum (low-pass) and difference (high pass) filter, moving average (CIC can be considered a special implementation of a moving average filter), comb filters, modified comb filters
  - NOT all FIR's have linear phase !! (linear requires symmetric or anti-symmetric pulse response)
- DFT using the Frequency Sampling

Digital Filters

Quantization Effects to Be Aware of:

- **Input Signal Quantization**
- **Quantization of the coefficients** (can affect filter stability, and freq. response)
- **Quantization of products** (under fine quantization assumptions, can analyze effects by adding an appropriate noise source at the output of each product)

Coarse quantization can lead to limit cycles

Be Aware of Overflow and use proper scaling for dynamic range:

Overflow can cause limit cycles

System Discretization

System Discretization - continuous-time linear systems

$$\dot{x}(t) = A(t)\,x(t) \qquad x(t_o) = x_o$$

State equation, no inputs or outputs

The sequence of approximating functions (*Picard Iteration*):

$$x_o(t) = x_o \;,\; x_1(t) = x_o + \int_{t_o}^{t} A(\sigma_1)x_o(\sigma_1)d\sigma_1 \;,\; x_2(t) = x_o + \int_{t_o}^{t} A(\sigma_1)x_1(\sigma_1)d\sigma_1 \;,\; .. \; x_k(t) = x_o + \int_{t_o}^{t} A(\sigma_1)x_{k-1}(\sigma_1)d\sigma_1$$

converges to:

$$x(t) = x_o + \int_{t_o}^{t} A(\sigma_1)x_o d\sigma_1 + \int_{t_o}^{t} A(\sigma_1)\int_{t_o}^{\sigma_1} A(\sigma_2)x_o d\sigma_2 d\sigma_1 + \cdots \int_{t_o}^{t} A(\sigma_1)\int_{t_o}^{\sigma_2} A(\sigma_2)\int_{t_o}^{\sigma_3} A(\sigma_3)\cdots \int_{t_o}^{\sigma_{k-1}} A(\sigma_k)x_o d\sigma_k d\sigma_{k-1}\cdots d\sigma_1 + \cdots$$

term-by-term differentiation leads to:

$$\dot{x}(t) = A(t)\,x(t)$$

thus the above *formula for x(t)* satisfies the state equation !!

Factoring x_o from *x(t)* leads to the Transition Matrix (*Peano-Baker series)*

$$\boxed{\Phi(t,\tau) = I + \int_{\tau}^{t} A(\sigma_1)d\sigma_1 + \int_{\tau}^{t} A(\sigma_1)\int_{\tau}^{\sigma_1} A(\sigma_2)d\sigma_2 d\sigma_1 + \cdots \int_{\tau}^{t} A(\sigma_1)\int_{\tau}^{\sigma_2} A(\sigma_2)\int_{\tau}^{\sigma_3} A(\sigma_3)\cdots \int_{\tau}^{\sigma_{k-1}} A(\sigma_k)d\sigma_k d\sigma_{k-1}\cdots d\sigma_1 + \cdots}$$

$$\boxed{x(t) = \Phi(t,t_o)\,x_o}$$

System Discretization - continuous-time linear systems

Special Case 1: A(t) = A , constant matrix

typical Peano-Baker term becomes:

$$\int_{\tau}^{t} A(\sigma_1)\int_{\tau}^{\sigma_2} A(\sigma_2)\int_{\tau}^{\sigma_3} A(\sigma_3)\cdots\int_{\tau}^{\sigma_{k-1}} A(\sigma_k)d\sigma_k d\sigma_{k-1}\cdots d\sigma_1 = A^k\int_{\tau}^{t}\int_{\tau}^{\sigma_2}\int_{\tau}^{\sigma_3}\int_{\tau}^{\sigma_{k-1}} 1\, d\sigma_k d\sigma_{k-1}\cdots d\sigma_1 \boxed{= \frac{A^k(t-\tau)^k}{k!}}$$

thus:

$$\Rightarrow \boxed{\Phi(t,\tau) = \sum_{k=0}^{\infty}\frac{A^k(t-\tau)^k}{k!} = e^{A(t-\tau)}}$$

$$\boxed{e^{At} \equiv \sum_{k=0}^{\infty}\frac{A^k t^k}{k!}}$$

Special Case 2: A(t) commutes with its integral $A(t)\int_{\tau}^{t} A(\sigma_1)d\sigma_1 = \int_{\tau}^{t} A(\sigma_1)d\sigma_1\, A(t)$

$$\Rightarrow \boxed{\Phi(t,\tau) = \sum_{k=0}^{\infty}\frac{1}{k!}\left(\int_{\tau}^{t} A(\sigma)d\sigma\right)^k = e^{\int_{\tau}^{t} A(\sigma)d\sigma}}$$

System Discretization - continuous-time linear systems

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \qquad x(t_o) = x_o$$ State equation with input

$$y(t) = C(t)x(t) + D(t)u(t)$$ output equation

change of variables $\boxed{z(t) = \Phi^{-1}(t,t_o)x(t)} \Rightarrow x(t) = \Phi(t,t_o)z(t)\ ,\ z(t_o) = x_o$

$$\Rightarrow \quad \dot{\Phi}(t,t_o)z(t) + \Phi(t,t_o)\dot{z}(t) = A(t)\Phi(t,t_o)z(t) + B(t)u(t)$$

$$\Rightarrow \quad A(t)\Phi(t,t_o)z(t) + \Phi(t,t_o)\dot{z}(t) = A(t)\Phi(t,t_o)z(t) + B(t)u(t)$$

$$\Rightarrow \quad \boxed{\dot{z}(t) = \Phi^{-1}(t,t_o)B(t)u(t)} \quad \Rightarrow \quad \boxed{\int_{t_o}^{t} \dot{z}(\sigma) = \int_{t_o}^{t} \Phi^{-1}(\sigma,t_o)B(\sigma)u(\sigma)\,d\sigma}$$

$$\Rightarrow \quad \boxed{x(t) = \Phi(t,t_o)x_o + \int_{t_o}^{t} \Phi(t,\sigma)B(\sigma)u(\sigma)\,d\sigma}$$

After substitution, rearranging, and using transition matrix properties

zero-input response

zero-state response

$$\Rightarrow \quad \boxed{y(t) = C(t)\Phi(t,t_o)x_o + \int_{t_o}^{t} C(t)\Phi(t,\sigma)B(\sigma)u(\sigma)\,d\sigma + D(t)u(t)}$$

System Discretization - continuous-time linear systems

Special Case 1: time-invariant system $\Phi(t,t_o) = e^{A(t-t_o)}$

letting t_o=0

$$\Rightarrow \quad x(t) = e^{At}x_o + \int_0^t e^{A(t-\sigma)}Bu(\sigma)d\sigma$$

$$\Rightarrow \quad y(t) = Ce^{At}x_o + \int_0^t Ce^{A(t-\sigma)}Bu(\sigma)d\sigma + Du(t)$$

let $\tau = t-\sigma$

$$\Rightarrow \quad \boxed{y(t) = Ce^{At}x_o + \int_0^t Ce^{A\tau}Bu(t-\tau)d\tau + Du(t)}$$

for a single unit-impulse input $u(t) = \delta(t)$

$$\boxed{h(t) \equiv y(t)\Big|_{\substack{u(t)=\delta(t)\\ x_o=0}} = Ce^{At}B + D\delta(t)}$$

zero-state
unit-impulse response

$$\Rightarrow \quad \boxed{y(t)\Big|_{x_o=0} = \int_0^t h(\tau)u(t-\tau)d\tau}$$

THIS IS CONVOLUTION !!

Output due to an arbitrary input is given by convolution with the impulse response

System Discretization - continuous-time linear systems

Laplace Transform: $$L\{f(t)\} = F(s) = \int_0^\infty f(t)e^{-st}dt$$

derivative property: $$L\{f'(t)\} = sF(s) - f(0)$$

thus, because $$\frac{d}{dt}e^{At} = Ae^{At} \quad \Rightarrow \quad L\{e^{At}\} = (sI - A)^{-1}$$

convolution property: $$L\left\{\int_0^t f_1(\tau)f_2(t-\tau)d\tau\right\} = F_1(s)F_2(s)$$

$$y(t) = Ce^{At}x_o + \int_0^t Ce^{A\tau}Bu(t-\tau)d\tau + Du(t)$$

$$\Rightarrow \quad Y(s) = C(sI - A)^{-1}x_o + \left\{C(sI - A)^{-1}B + D\right\}U(s)$$

$$\Rightarrow \quad H(s) = \left.\frac{Y(s)}{U(s)}\right|_{x_o = 0} = C(sI - A)^{-1}B + D$$

Transfer Function

note: for the multi-input case, unit-pulses need to be applied individually to piece together H(s)

System Discretization - continuous-time linear systems

Transfer Function Concept

$$\boxed{y(t)\big|_{x_o=0} = \int_0^t h(\tau)u(t-\tau)\,d\tau}$$

let $$\boxed{u(t) = \mathrm{Re}(u_o e^{i\omega_o t}) = u_o\cos(\omega_o t) = \frac{u_o}{2}\left[e^{i\omega_o t} + e^{-i\omega_o t}\right]}$$

as $t \to \infty$ $$\Rightarrow \quad y(t) = \frac{u_o}{2}\int_0^\infty h(\tau)e^{i\omega_o(t-\tau)}d\tau + \frac{u_o}{2}\int_0^\infty h(\tau)e^{-i\omega_o(t-\tau)}d\tau$$

$$\Rightarrow \quad y(t) = \frac{u_o}{2}\left(\int_0^\infty h(\tau)e^{-i\omega_o\tau}d\tau\right)e^{i\omega_o t} + \frac{u_o}{2}\left(\int_0^\infty h(\tau)e^{i\omega_o\tau}d\tau\right)e^{-i\omega_o t}$$

$$\Rightarrow \quad y(t) = \frac{u_o}{2}\left[H(i\omega_o)e^{i\omega_o t} + H^*(i\omega_o)e^{-i\omega_o t}\right]$$

LaPlace Transform at $s=i\omega_o$

$$\Rightarrow \quad y(t) = u_o\left[\mathrm{Re}\{H(i\omega_o)\}\cos(\omega_o t) - \mathrm{Im}\{H(i\omega_o)\}\sin(\omega_o t)\right]$$

$$\boxed{y(t) = \mathrm{Re}\left\{H(i\omega_o)u_o e^{i\omega_o t}\right\} = u_o\left|H(i\omega_o)\right|\cos(\omega_o + \angle H(i\omega_o))}$$

use $\boxed{H(s)\big|_{s=i\omega_o}}$ to find steady state response

System Discretization - discrete-time linear system

$$\boxed{\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k) \\ y(k) &= C(k)x(k) + D(k)u(k) \end{aligned}}$$

Linear State equation

Solution:

$k = k_o$ $\quad x(k_o+1) = A(k_o)x(k_o) + B(k_o)u(k_o)$

$k = k_o+1$ $\quad x(k_o+2) = A(k_o+1)x(k_o+1) + B(k_o+1)u(k_o+1)$

$$x(k_o+2) = A(k_o+1)A(k_o)x(k_o) + A(k_o+1)B(k_o)u(k_o) + B(k_o+1)u(k_o+1)$$

$k = k_o+2$ $\quad x(k_o+3) = A(k_o+2)A(k_o+1)A(k_o)x(k_o)$

$$+ A(k_o+2)A(k_o+1)B(k_o)u(k_o) + A(k_o+2)B(k_o+1)u(k_o+1) + B(k_o+2)u(k_o+2)$$

$$\boxed{\Phi(k,j) = \begin{cases} A(k-1)A(k-2)\cdots A(j), & k \geq j+1 \\ I, & k = j \end{cases}}$$

State Transition Matrix

Can write as one line (upper) by considering an the empty product (when k=j) as the identity matrix

$$\Rightarrow \boxed{x(k) = \Phi(k,k_o)x(k_o) + \sum_{j=k_o}^{k-1} \Phi(k,j+1)B(j)u(j), \quad k \geq k_o+1}$$

again,
sum of a zero-input and a zero-state response

$$\Rightarrow \boxed{y(k) = \begin{cases} C(k_o)x(k_o) + D(k_o)u(k_o), & k = k_o \\ C(k)\Phi(k,k_o)x(k_o) + \sum_{j=k_o}^{k-1} C(k)\Phi(k,j+1)B(j)u(j) + D(k)u(k), & k \geq k_o+1 \end{cases}}$$

System Discretization - discrete-time linear system

Special Case: Time-invariant (constant coefficient matrices)

$$\boxed{\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + Du(k) \end{aligned}}$$

$$\Phi(k,j) = \begin{cases} A(k-1)A(k-2)\cdots A(j), & k \geq j+1 \\ I, & k = j \end{cases}$$

$$\Rightarrow \quad \boxed{\Phi(k,j) = A^{k-j}, \quad k \geq j}$$

Can set $k_o = 0$ without loss of generality

$$x(k) = A^k x_o + \sum_{j=0}^{k-1} A^{k-j-1} Bu(j), \quad k \geq 1$$

$$y(k) = A^k x_o + \sum_{j=0}^{k-1} CA^{k-j-1} Bu(j) + Du(k), \quad k \geq 1$$

define functions to allow $k \geq 0$:

$$R(k) = \begin{cases} 0, & k = 0 \\ A^{k-1}B, & k \geq 1 \end{cases} \quad \Rightarrow \quad x(k) = A^k x_o + \overbrace{\sum_{j=0}^{k} R(k-j)u(j)}^{\text{discrete convolution}}, \quad k \geq 0$$

$$H(k) = \begin{cases} D, & k = 0 \\ CA^{k-1}B, & k \geq 1 \end{cases} \quad \Rightarrow \quad y(k) = CA^k x_o + \sum_{j=0}^{k} H(k-j)u(j), \quad k \geq 0$$

System Discretization - discrete-time linear system

Special Case: Time-invariant

$$y(k) = CA^k x_o + \sum_{j=0}^{k} H(k)u(k-j) \;, \quad k \geq 0 \quad \overset{\mathcal{Z}}{\Leftrightarrow} \quad Y_Z(z) = Cz(zI-A)^{-1}x_o + H_Z(z)U_Z(z)$$

similar to continuous time, H(k) can be measured as the unit-pulse response

note: for the multi-input case, unit-pulses need to be applied individually to piece together H(k)

for a single unit-impulse input: $u(k) = \delta(k)$

$$h(k) = y(k)\Big|_{\substack{u(k)=\delta(k)\\ x_o=0}} = \sum_{j=0}^{k} H(k)\delta(k-j) = H(k)$$

convolution

What is the z-transform of H(k) ?

$$H(k) = \begin{cases} D, & k=0 \\ CA^{k-1}B, & k \geq 1 \end{cases} \qquad Z\{H(k)\} = CZ\{\underbrace{(0, I, A, A^2, A^3, \cdots)}_{\text{sequence } A^k \text{ shifted to the right}}\}B + Z\{\underbrace{(D, 0, 0, 0, 0, \cdots)}_{\text{unit pulse sequence}}\}$$

$$\Longrightarrow \boxed{z^{-1}Z\{A^k\}}$$

What is the z-transform of A^k ?

$$x(k+1) = Ax(k) \quad \Leftrightarrow \quad X_Z(z) = z(zI-A)^{-1} \qquad x(0) = I$$

by uniqueness: $A^k \quad \overset{\mathcal{Z}}{\Leftrightarrow} \quad Z\{A^k\} = z(zI-A)^{-1}$

$$\Longrightarrow \quad \boxed{H_Z(z) = C(zI-A)^{-1}B + D}$$

Transfer Function

similarly: $R_Z(z) = (zI-A)^{-1}B$

System Discretization - discrete-time linear system

Transfer Function Concept

$$\boxed{y(k) = CA^k x_o + \sum_{j=0}^{k} H(k)u(k-j) \quad , \quad k \geq 0}$$

let $$\boxed{u(k) = \mathrm{Re}(u_o e^{i\omega_o kT}) = u_o \cos(\omega_o kT) = \frac{u_o}{2}\left[e^{i\omega_o kT} + e^{-i\omega_o kT}\right]}$$

as $k \to \infty$ $$\Rightarrow \quad y(k) = \frac{u_o}{2}\left\{\sum_{n=0}^{k} H(k) e^{-i\omega_o nT}\right\} e^{i\omega_o kT} + \frac{u_o}{2}\left\{\sum_{n=0}^{k} H(k) e^{i\omega_o nT}\right\} e^{-i\omega_o kT}$$

$$\Rightarrow \quad y(k) = \frac{u_o}{2}\left\{\sum_{n=0}^{k} H(k) z^{-n}\right\}\Bigg|_{z=e^{i\omega_o T}} e^{i\omega_o kT} + \frac{u_o}{2}\left\{\sum_{n=0}^{k} H(k) z^{n}\right\}\Bigg|_{z=e^{-i\omega_o T}} e^{-i\omega_o kT}$$

z-transform of H(k)

$$\Rightarrow \quad y(k) = \frac{u_o}{2}\left[H_Z(z = e^{i\omega_o kT}) \cdot e^{i\omega_o kT} + H_Z^*(z = e^{i\omega_o kT}) \cdot e^{i\omega_o kT}\right]$$

$$\Rightarrow \quad y(k) = u_o\left(\mathrm{Re}\left\{H_Z(z = e^{i\omega_o kT})\right\} \cdot \cos\omega_o kT - \mathrm{Im}\left\{H_Z(z = e^{i\omega_o kT})\right\} \cdot \sin\omega_o kT\right)$$

$$\boxed{y(k) = \mathrm{Re}\left\{H_Z(z = e^{i\omega_o kT}) \cdot u_o e^{i\omega_o kT}\right\} = u_o \left|H_Z(z = e^{i\omega_o kT})\right| \cdot \cos(\omega_o kT + \angle H_Z(z = e^{i\omega_o kT}))}$$

$\Rightarrow$ use $\boxed{H_Z(z)\big|_{z = e^{i\omega_o kT}}}$ evaluate on the unit circle to find steady state response

System Discretization - discrete-time equivalent

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

$$x(t) = e^{At}x_o + \int_0^t e^{A(t-\sigma)}Bu(\sigma)d\sigma$$

$$x(kT+T) = e^{AT}e^{kT}x_o + \int_0^{kT+T} e^{A(kT+T-\sigma)}Bu(\sigma)d\sigma$$

$$\Rightarrow \quad x(kT+T) = e^{AT}e^{kT}x_o + e^{AT}\int_0^{kT} e^{A(kT-\sigma)}Bu(\sigma)d\sigma + \int_{kT}^{kT+T} e^{A(kT+T-\sigma)}Bu(\sigma)d\sigma$$

$$\Rightarrow \quad x(kT+T) = e^{AT}x(kT) + \int_{kT}^{kT+T} e^{A(kT+T-\sigma)}d\sigma \cdot Bu(kT)$$

when the output of the DAC is a zero-order hold

$$\Rightarrow \quad \boxed{x(k+1) = A^*x(k) + B^*u(k)} \qquad \boxed{y(k) = Cx(k) + Du(k)}$$

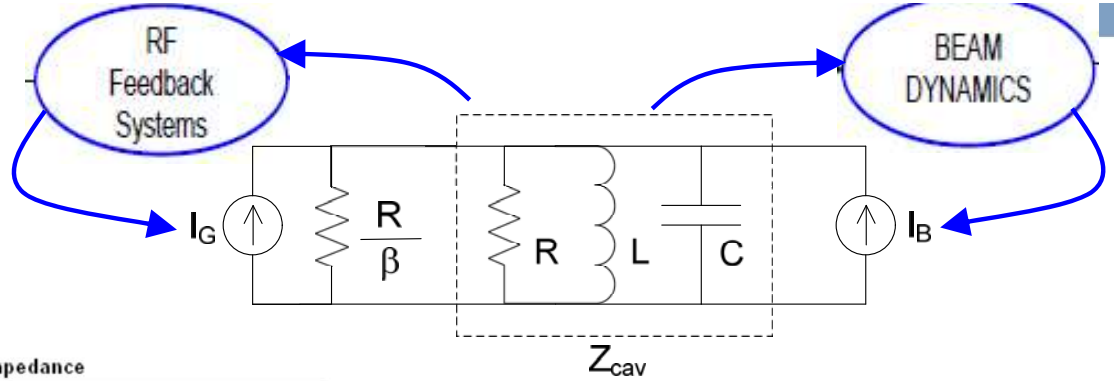
$$\boxed{A^* = e^{AT}} \qquad \boxed{B^* = \int_0^T e^{A\tau}d\tau \cdot B}$$

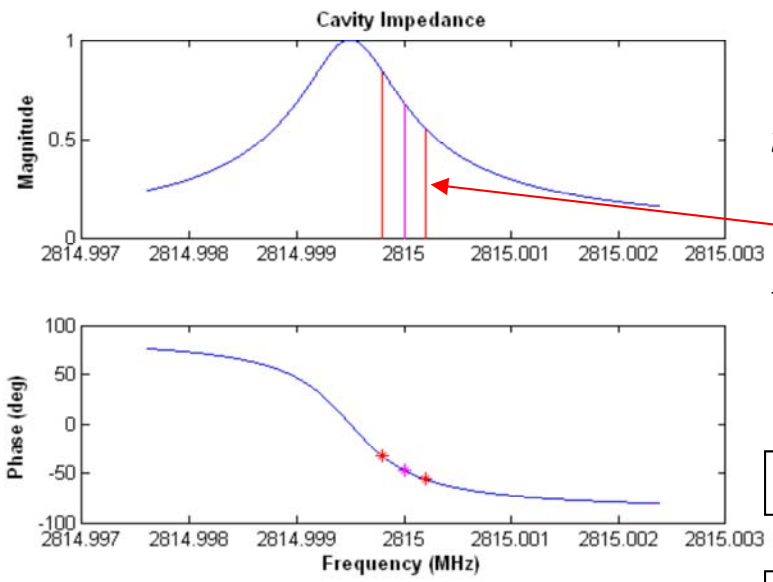
$$B^* = A^{-1}(e^{AT} - I)\cdot B$$ when A is non-singular

$$B^* = \left[IT + \frac{AT^2}{2!} + \frac{A^2T^3}{3!} + \cdots \frac{A^kT^{k+1}}{(k+1)!} + \cdots\right]\cdot B$$ otherwise

The Cavity System

The System





$$\hat{i} = I \cdot \left(1 + \frac{a}{2}\left[e^{j\omega_{am} t} + e^{-j\omega_{am} t} \right] \right)$$ In-Phase modulated Drive Current:

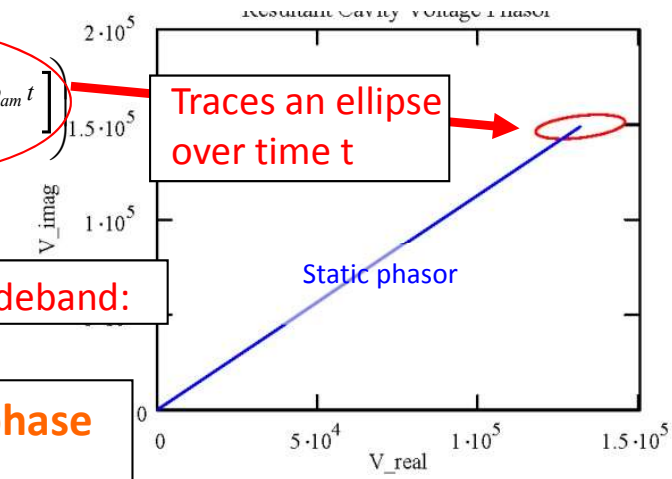
$$\hat{v} = I \cdot \left(Z(i\omega_{RF}) + \frac{a}{2}\left[Z^{+} e^{j\omega_{am} t} + Z^{-} e^{-j\omega_{am} t} \right] \right)$$

Z at ω_{RF}:

Z at upper sideband:

Z at lower sideband:

Traces an ellipse over time t



Resultant V phasor has both **In-phase** and **Quadrature** components (**if cavity is detuned**)

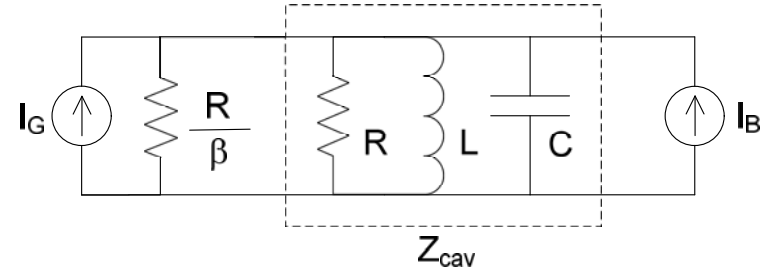
$$\hat{v} = I\frac{a}{2} \cdot \left[\left\{ \left[Z^{+}_{\mathrm{Re}} + Z^{-}_{\mathrm{Re}} \right] \cdot \cos\omega_{am} t - \left[Z^{+}_{\mathrm{Im}} - Z^{-}_{\mathrm{Im}} \right] \cdot \sin\omega_{am} t \right\} + j \cdot \left\{ \left[Z^{+}_{\mathrm{Im}} + Z^{-}_{\mathrm{Im}} \right] \cdot \cos\omega_{am} t - \left[Z^{-}_{\mathrm{Re}} - Z^{+}_{\mathrm{Re}} \right] \cdot \sin\omega_{am} t \right\} \right]$$

$$\begin{bmatrix} \hat{v}_i(s) \\ \hat{v}_q(s) \end{bmatrix} = \begin{bmatrix} G_{ii}(s) = \frac{1}{2}\left[Z(j\omega_{RF} + s) + Z^{*}(j\omega_{RF} - s) \right] & G_{qi}(s) = -G_{iq}(s) \\ G_{iq}(s) = -\frac{j}{2}\left[Z(j\omega_{RF} + s) - Z^{*}(j\omega_{RF} - s) \right] & G_{qq}(s) = G_{ii}(s) \end{bmatrix} \begin{bmatrix} \hat{i}_i(s) \\ \hat{i}_q(s) \end{bmatrix}$$

The System

State-Space Model

$$\begin{bmatrix} \dot{V}_i \\ \dot{V}_q \end{bmatrix} = \begin{bmatrix} -\sigma & -\Delta\omega \\ \Delta\omega & -\sigma \end{bmatrix} \begin{bmatrix} V_i \\ V_q \end{bmatrix} + \begin{bmatrix} \sigma R_p & 0 \\ 0 & \sigma R_p \end{bmatrix} \begin{bmatrix} I_i \\ I_q \end{bmatrix}$$



About nominal operating point, can use perturbation technique to study microphonics, otherwise system is time-varying. Most thorough treatment found in [9]

$$\begin{bmatrix} \dot{V}_i \\ \dot{V}_q \end{bmatrix} = \begin{bmatrix} -\sigma & -\Delta\omega \\ \Delta\omega & -\sigma \end{bmatrix} \begin{bmatrix} V_i \\ V_q \end{bmatrix} + \begin{bmatrix} 0 & -\delta\omega(t) \\ \delta\omega(t) & 0 \end{bmatrix} \begin{bmatrix} V_i \\ V_q \end{bmatrix} + \begin{bmatrix} \sigma R_p & 0 \\ 0 & \sigma R_p \end{bmatrix} \begin{bmatrix} I_i \\ I_q \end{bmatrix}$$

$$\Delta\omega \equiv \omega_r - \omega_o \qquad R_p = R/(\beta+1)$$

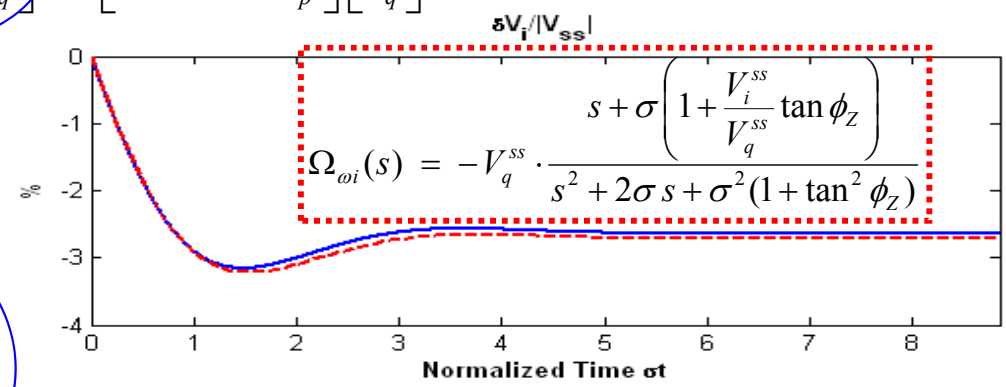
Perturbation Technique

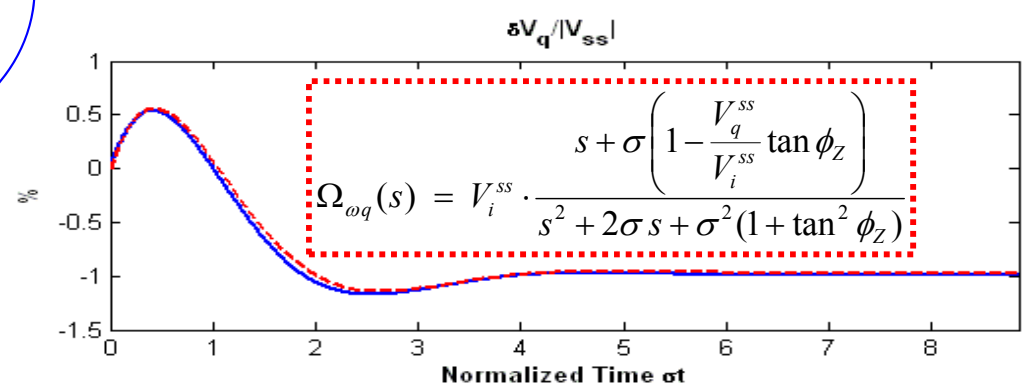
$$\dot{\hat{V}} = (A + \delta\tilde{\Omega}(t))\,\hat{V} + B\,\hat{I}$$

$$\begin{bmatrix} \dot{v}_i \\ \dot{v}_q \end{bmatrix} = \begin{bmatrix} -\sigma & -\Delta\omega \\ \Delta\omega & -\sigma \end{bmatrix} \begin{bmatrix} v_i \\ v_q \end{bmatrix} + \begin{bmatrix} \sigma R_p & 0 \\ 0 & \sigma R_p \end{bmatrix} \hat{u}$$

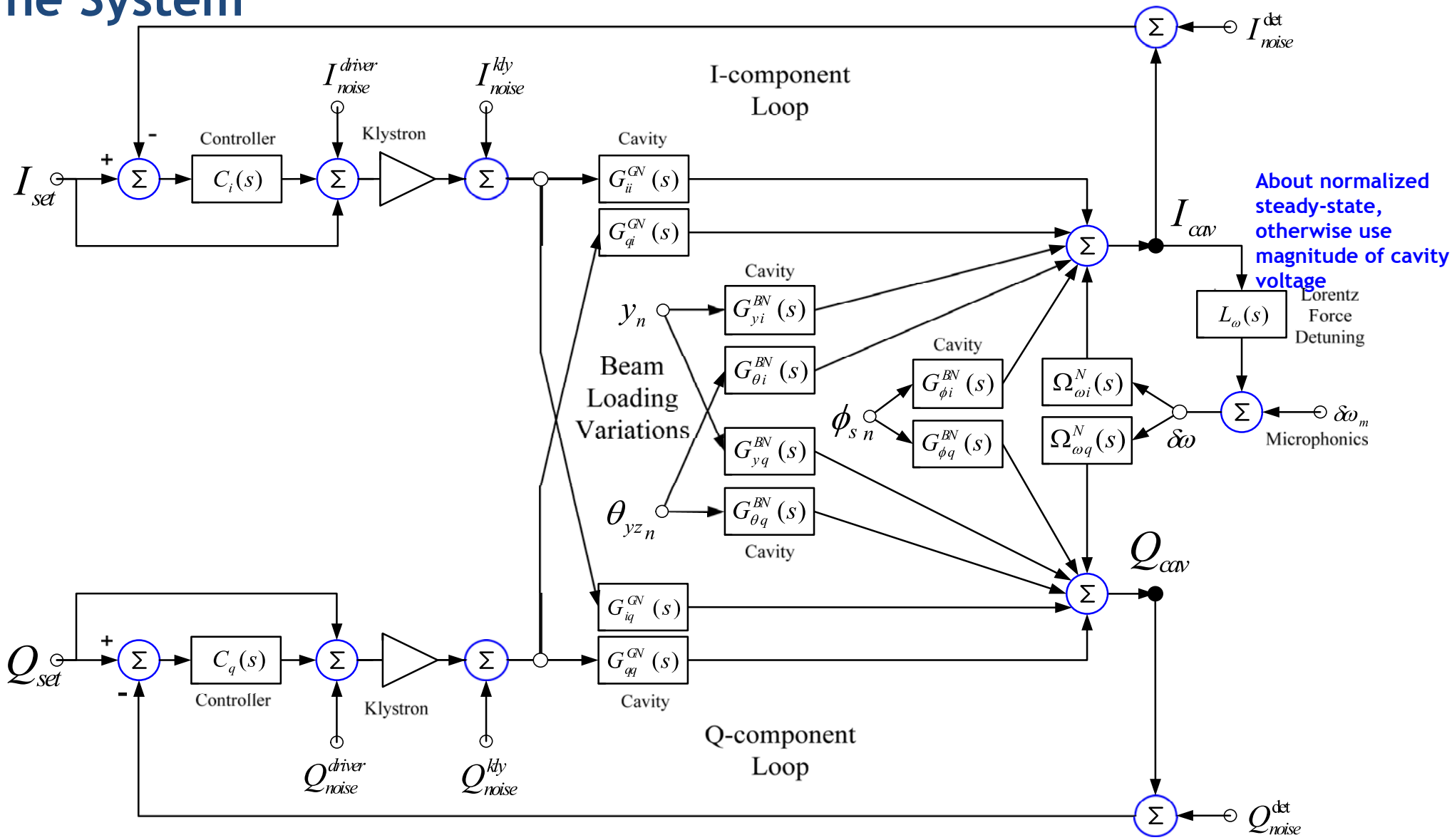
$$\hat{u} = \frac{\delta\omega(t)}{\sigma R_p} \cdot \begin{bmatrix} -V_Q^{ss} \\ V_I^{ss} \end{bmatrix}$$

$$\Omega(s) = \begin{bmatrix} \Omega_{\omega i}(s) \\ \Omega_{\omega q}(s) \end{bmatrix} = \frac{1}{\sigma R} \cdot \begin{bmatrix} -V_Q^{ss} \cdot G_{ii}(s) + V_I^{ss} \cdot G_{qi}(s) \\ V_I^{ss} \cdot G_{qq}(s) - V_Q^{ss} \cdot G_{iq}(s) \end{bmatrix}$$





The System

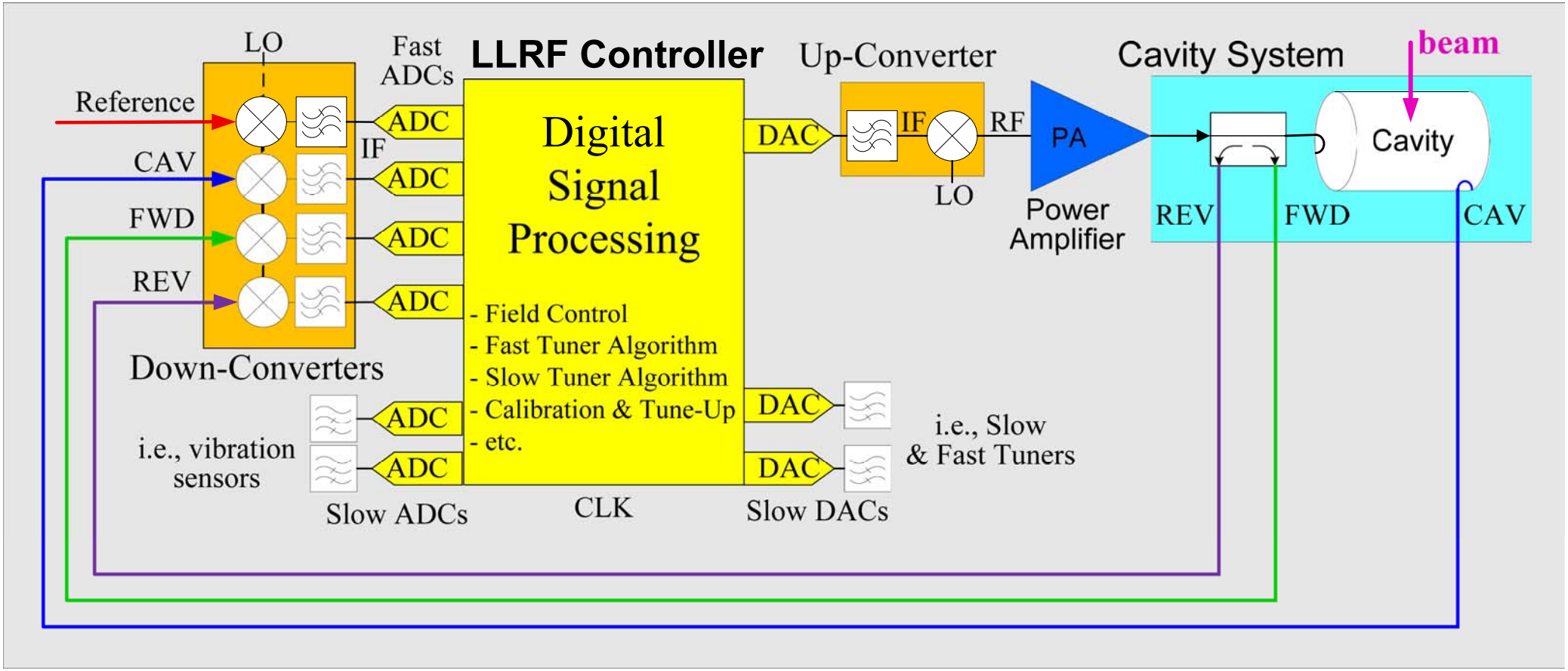


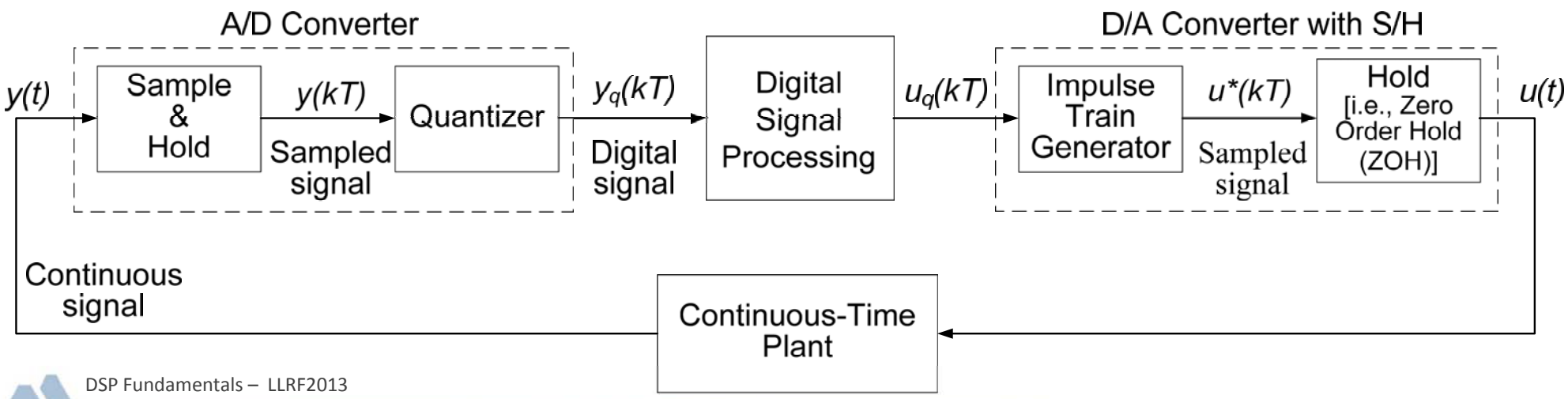
This is shown to gain an appreciation for the complexity of a single cavity system. State-space representation starts looking a lot more attractive than above.

For deflecting cavities: beam-loading is a function of offset, tilt, arrival phase, and current

Note, this does not include beam dynamics

Motivation


LO
Fast ADCs
LLRF Controller
Up-Converter
Cavity System
beam
Reference
CAV
FWD
REV
IF
ADC
ADC
ADC
ADC
Digital Signal Processing
- Field Control
- Fast Tuner Algorithm
- Slow Tuner Algorithm
- Calibration & Tune-Up
- etc.
DAC
IF
RF
LO
PA
Power Amplifier
Cavity
REV
FWD
CAV
Down-Converters
ADC
ADC
i.e., vibration sensors
Slow ADCs
CLK
DAC
DAC
i.e., Slow & Fast Tuners
Slow DACs


A/D Converter
D/A Converter with S/H
y(t)
Sample & Hold
y(kT)
Sampled signal
Quantizer
yq(kT)
Digital signal
Digital Signal Processing
uq(kT)
Impulse Train Generator
u*(kT)
Sampled signal
Hold [i.e., Zero Order Hold (ZOH)]
u(t)
Continuous signal
Continuous-Time Plant

References

[1] W. S. Levine, et. al., "*Control System Fundamentals,*" "*The Control Handbook,*" 2nd Edition, CRC Press, 2011.
[2] E. Cunningham, "*Digital Filtering, An Introduction,*" Houghton Mifflin Company, 1992.
[3] W. J. Rugh, "*Linear System Theory,*" 2nd Edition, Prentice Hall, 1996.
[4] G. Cooper, C. McGillem, "*Probabilistic Methods of Signal and System Analysis*," 2nd Edition, Prentice Hall, 1996.
[5] J. Reid, "*Linear System Fundamentals, Continuous and Discrete, Classic and Modern,*" McGraw-Hill, 1983.
[6] E. Rubiola, "*Phase Noise and Frequency Stability in Oscillators,*" Cambridge University Press, 2009.
[7] W. Kester, "*Aperture Time, Aperture Jitter, Aperture Delay Time – Removing the Confusion,*" Analog Devices MT-007 Tutorial, Rev.A, Oct. 2008.
[8] R. Gray, "*Quantization Noise Spectra,*" IEEE Transactions on Information Theory, Vol. 36, No. 6, Nov. 1990.
[9] J. O. Smith III, "*Introduction to Digital Filters with Audio Applications,*" Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, https://ccrma.stanford.edu/~jos/filters/.
[8] E. B. Saff, A. D. Snider, "*Fundamentals of Complex Analysis for Mathematics, Science, and Engineering,*" 2nd Edition, Prentice-Hall, 1993.
[9] M. Doleans, "*Studies in Reduced-Beta Elliptical Superconducting Cavities (General Design Considerations, Dynamic Detuning, and Longitudinal Dynamics),*" Universite Paris 7 – Denis Diderot, Doctorat Physique et Technologie des Grands-Instruments, Sept. 2003. http://neutrons.ornl.gov/APGroup/Papers/DoleansThesis.pdf
[10] R. Bellman, "*Perturbation Techniques in Mathematics, Engineering, & Physics*," Dover Publications, Inc., NY, 1972.