EE 435
Lecture 41

Phased Locked Loops and VCOs
Over Sampled Data Converters

Final Exam:

Scheduled on Final Exam Schedule:

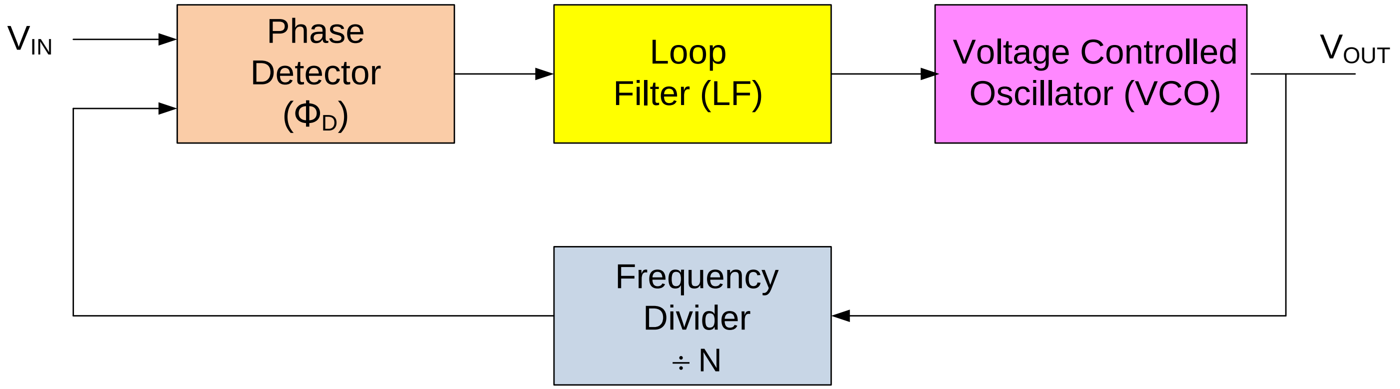
Wednesday May 5 9:45 a.m.

Revised Final Exam:

- Take-home format – open book and open notes
- Will be posted on course WEB site by late Friday April 30
- Due at 5:00 p.m. on Wednesday May 5 : Upload as pdf file into Canvas

If anyone has any constraints of any form such as internet access or other factors that makes it difficult to work with this revised format, please contact Professor Geiger by 5:00 p.m. on Wednesday April 28

Basic PLL Architecture



Applications include:

Frequency Demodulation

Frequency Synthesis

Clock Synchronization

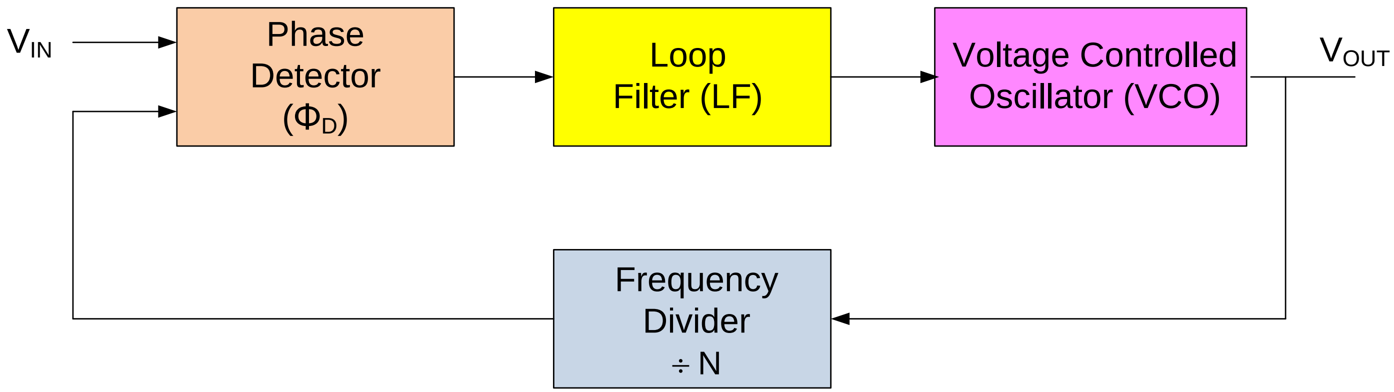
Noise filtering (extreme)

Tracking and calibrated filters

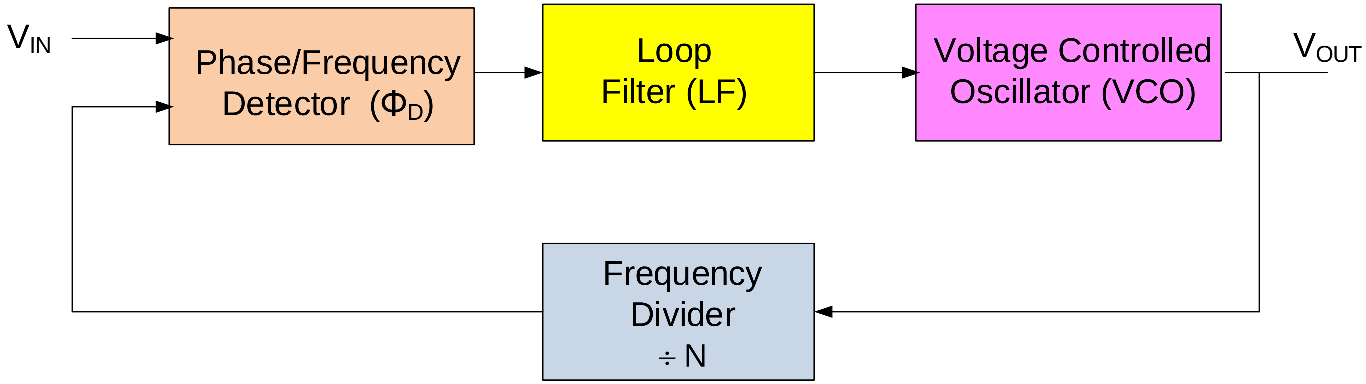
.....

- **One of the most widely used analog blocks**
- **Many SoC systems include multiple PLLs**
- **Closely related to Delay Locked Loop (DLL)**

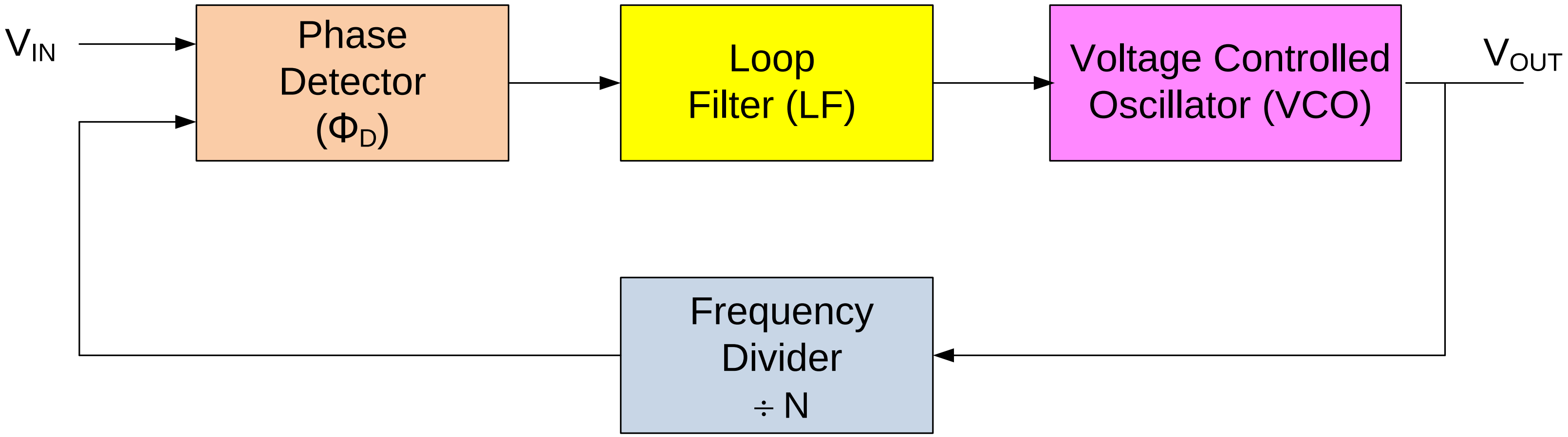
Basic PLL Architecture



Often actually use phase/frequency detector but still termed PLL



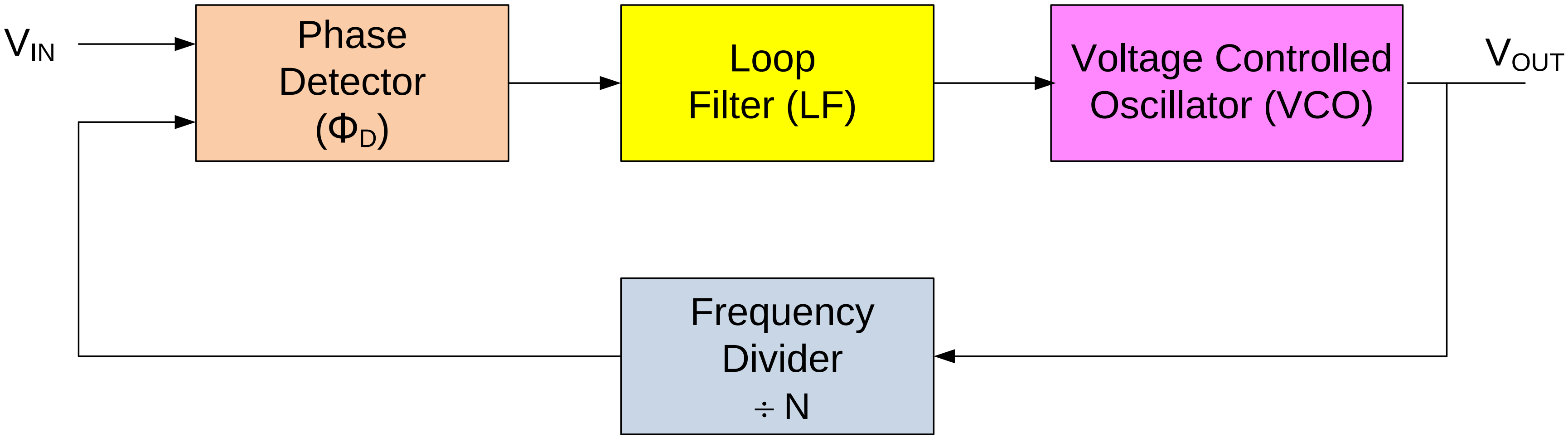
Basis PLL Architecture



Applications by subcategory:

- Clock and Data Recovery
- Recovering signals when SNR <<<1
- Timing generators in digital systems

Desired Operation of PLL



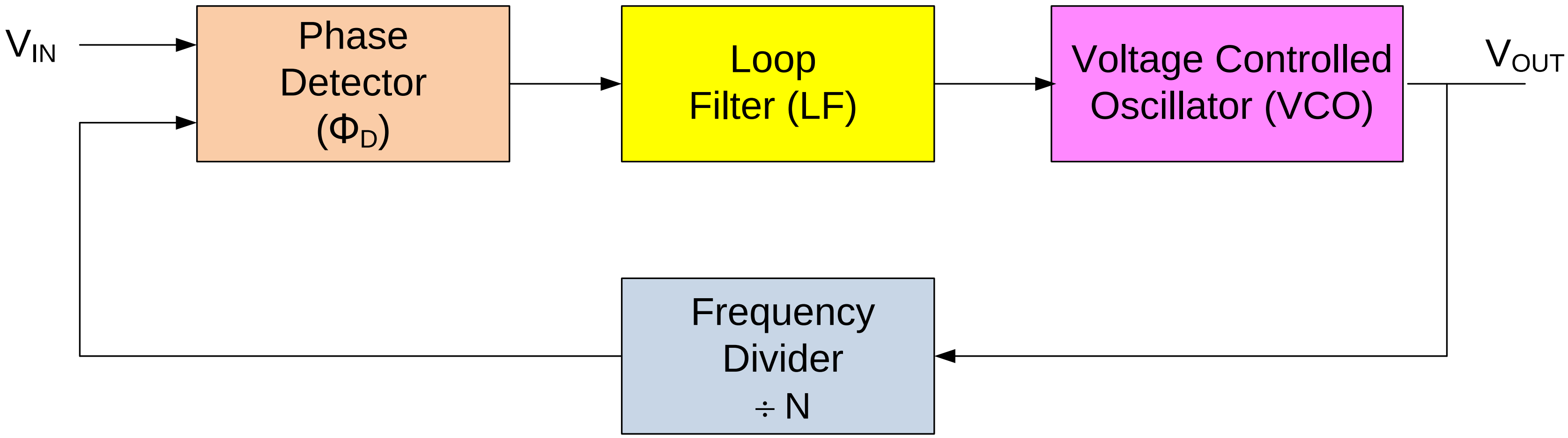
$$V_{IN}=V_M\sin(\omega_{IN}t+\phi_{IN})$$

Desired output when locked:

$$V_{OUT}=V_X\sin(N\omega_{IN}t+\phi_{OUT})$$

- Relationship between V_M and V_X is of little concern
- Frequency relationship is critical
- ϕ_{OUT} is often critical too
- Waveshape of V_{IN} and V_{OUT} is often of little concern
  May be highly distorted or even square waves

Desired Operation of PLL



$$V_{IN}=V_M\sin(\omega_{IN}t+\phi_{IN})$$

Desired output when locked:

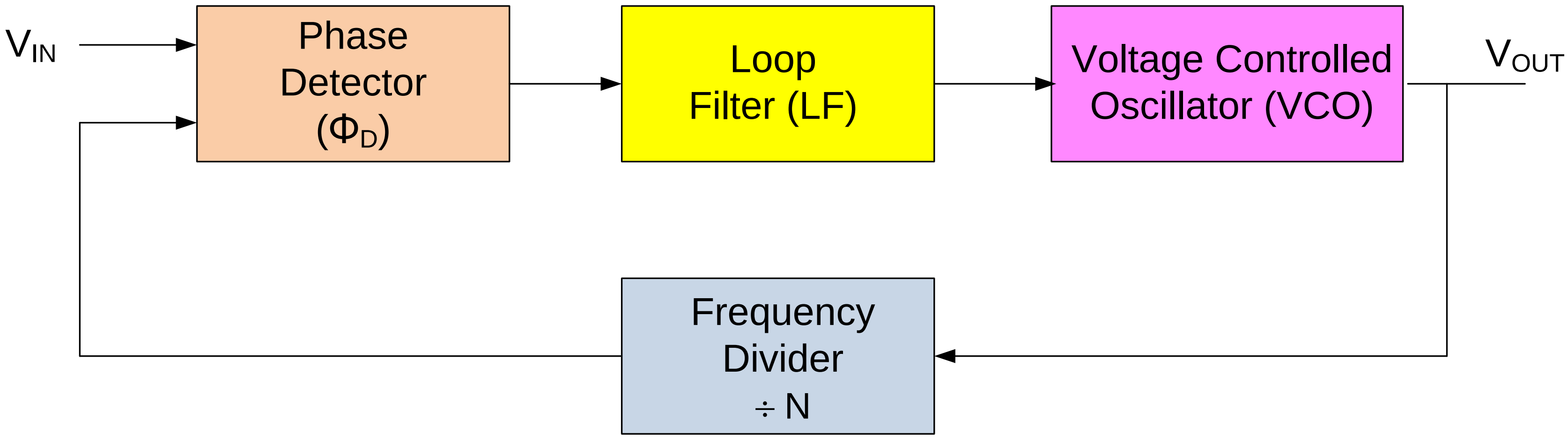
$$V_{OUT}=V_X\sin(N\omega_{IN}t+\phi_{OUT})$$

Some Terminology of PLLs

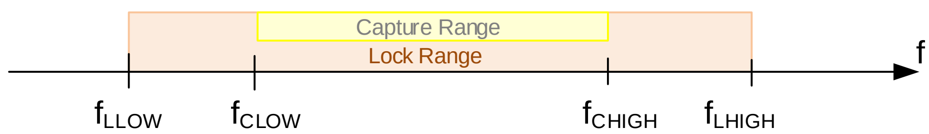
- Locked / Unlocked — Locked when V_{OUT} assumes desired value
- Lock Range — $f_{LLOW} < f_{IN} < f_{LHIGH}$ If locked, will remain locked for f_{IN} in lock range
- Capture Range — $f_{CLOW} < f_{IN} < f_{CHIGH}$ If unlocked, will lock for f_{IN} in capture range
- Free-running frequency — frequency of VCO when not locked
- Harmonic/Subharmonic Lock

Desired Operation of PLL

$V_{IN}=V_M\sin(\omega_{IN}t+\phi_{IN})$ $V_{OUT}=V_X\sin(N\omega_{IN}t+\phi_{OUT})$



Capture range always less than lock range $f_{LLOW} < f_{CLOW} < f_{CHIGH} < f_{LHIGH}$

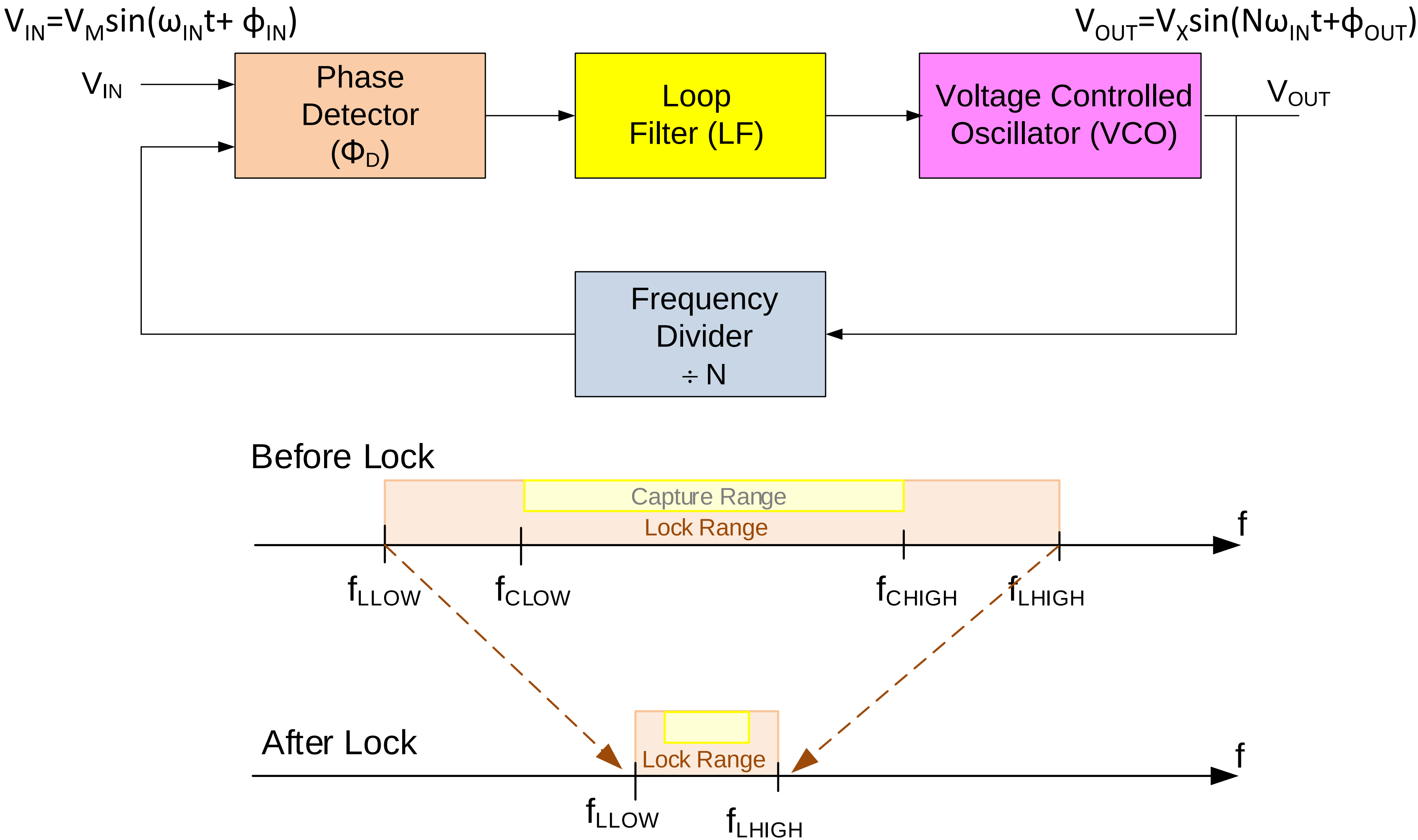


Loop filter controls capture and lock range

Jitter in VCO output strongly dependent upon lock range
(large lock range results in high jitter, low lock range in low jitter)

Loop filter is often dynamic with wide bandwidth prior to lock and narrow BW after lock

Desired Operation of PLL



Loop filter is often dynamic with wide bandwidth prior to lock and narrow BW after lock

Conceptual Operation of PLL

Consider a signal defined for $-\infty < t < \infty$ expressed as

$$V(t) = V_M \sin(\phi(t))$$

If the signal is sinusoidal with frequency ω, the argument ϕ can be expressed as

$$\phi(t)=\omega t+\theta$$

where ϕ is defined to be the phase of the signal and θ is the phase offset on the time axis from the time reference t=0

Taking the time derivative of ϕ(t) , we obtain

$$\frac{d\phi}{dt} = \omega$$

Taking the Laplace Transform, we have

$$\phi_S = \frac{\omega}{s}$$

Is the second statement “If the signal is sinusoidal ….” redundant ?

Is the second statement “If the signal is sinusoidal ….” redundant ?

Consider a signal defined for $-\infty < t < \infty$ expressed as

$$V(t) = V_M \sin(\phi)$$

If the signal is sinusoidal with frequency ω, the argument ϕ can be expressed as

$$\phi = \omega t + \theta$$

Consider any signal f(t) defined for all time (not necessarily periodic but could be)

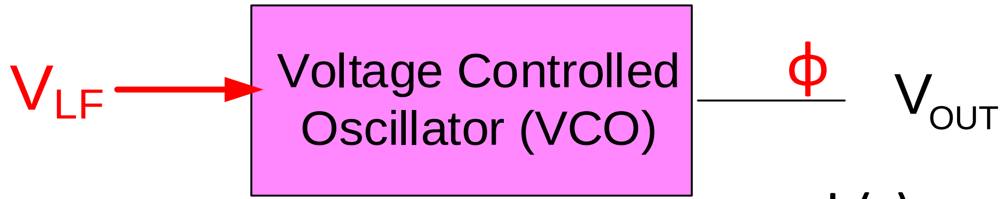
Define ϕ(t) by the expression $\phi(t) = \sin^{-1}\left(\frac{f(t)}{V_M}\right)$

It follows that $V(t) = V_M \sin(\phi(t)) = V_M \sin\left(\sin^{-1}\left(\frac{f(t)}{V_M}\right)\right) = f(t)$

Thus, the first statement gives NO information about the signal V(t)

Consider a VCO where the output is sinusoidal

Assume the output of interest is the phase ϕ



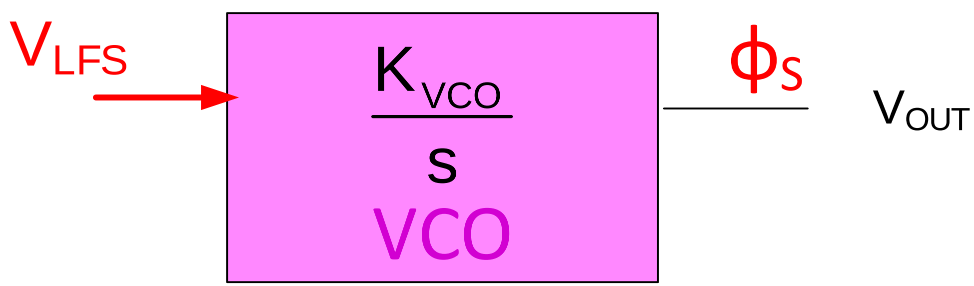
$\phi(t) = \omega t + \theta$

$$V_{OUT} = V_m \sin(\omega t + \theta)$$

$$\omega = V_{LF} \bullet K_{VCO}$$

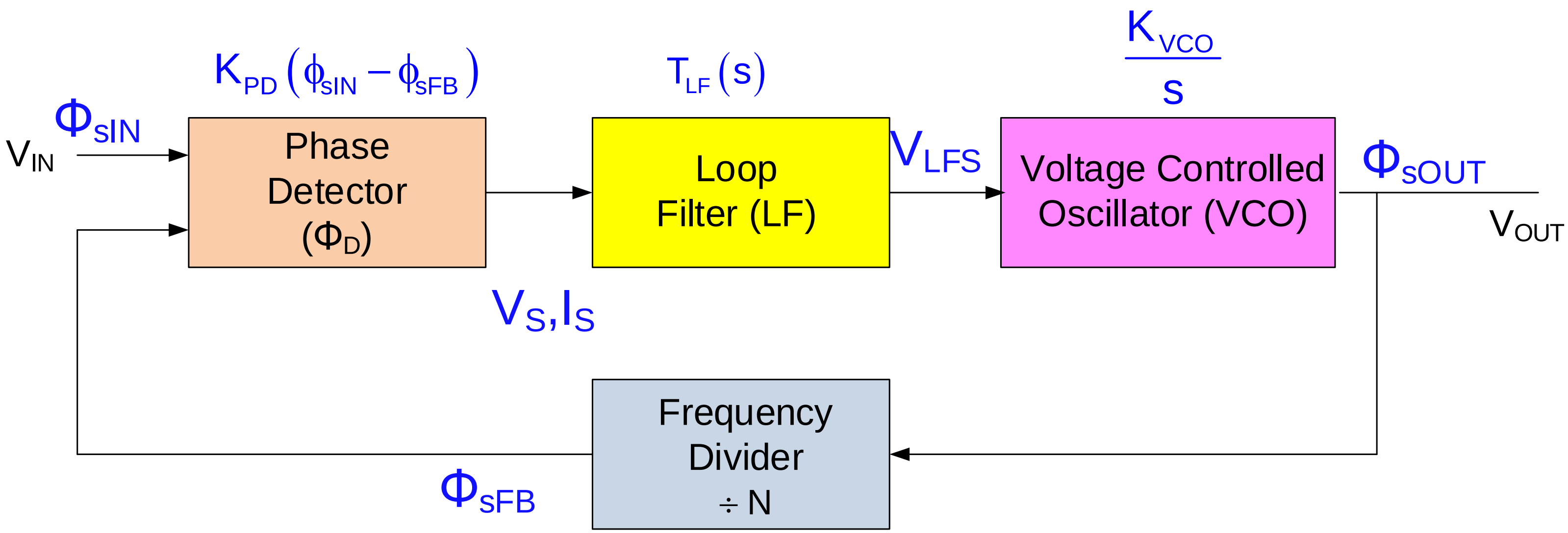
$$\frac{d\phi}{dt} = \omega = V_{LF} K_{VC0}$$

Taking Laplace Transform: $s\phi_S = V_{LFS} K_{VC0} \quad \Rightarrow \quad \phi_S = V_{LFS} \dfrac{K_{VC0}}{s}$



Conceptual Operation of PLL

- When locked, PLL can be modeled as a linear system
- Small-signal s-domain analysis when PLL is locked



Note: Dimensions of variables in loop are not the same

$$V = K_{PD}(\phi_{sIN} - \phi_{sFB})$$

$$V_{LF} = T_{LF}(s)V$$

$$\phi_{sOUT} = V_{LF}\frac{K_{VCO}}{s}$$

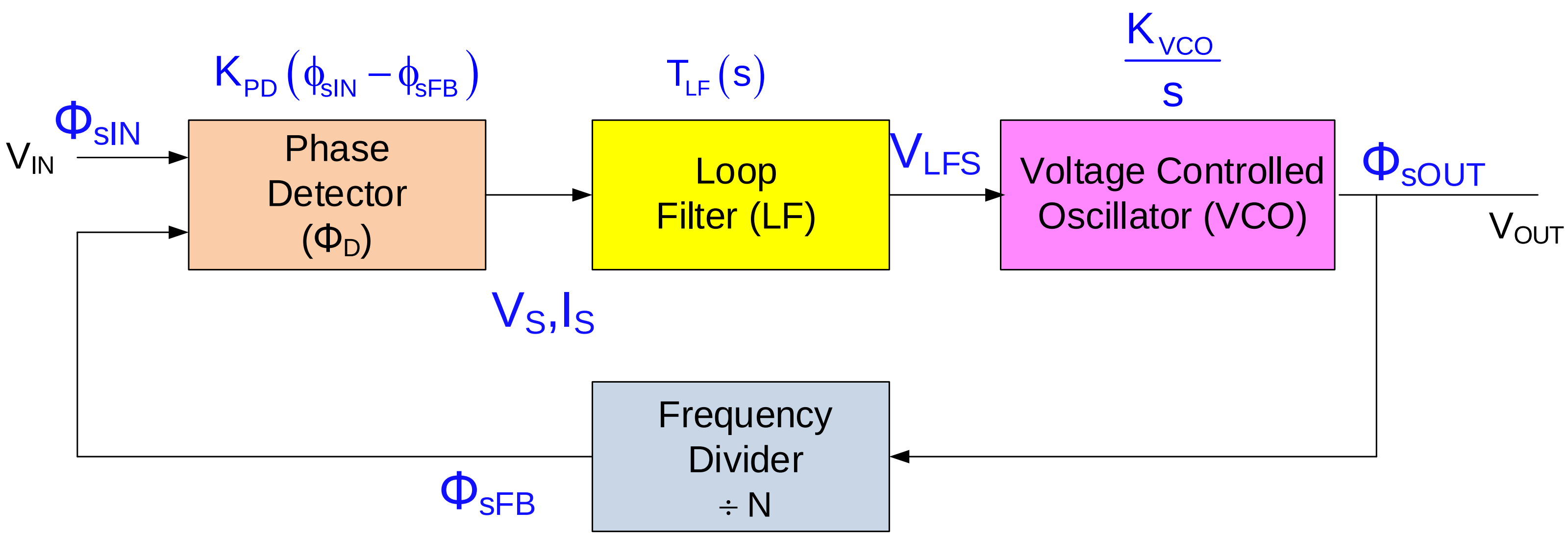
$$\phi_{sFB} = \frac{\phi_{sOUT}}{N}$$

Solving, we obtain

$$T_{PLL}(s) = \frac{\phi_{sOUT}}{\phi_{sIN}} = \frac{T_{LF}(s)K_{PD}K_{VCO}}{s + T_{LF}(s)\frac{K_{VCO}K_{PD}}{N}}$$

Often the LF is low order

Example: Assume N=1 and $T_{LF}(s) = \frac{1}{1+RCs}$

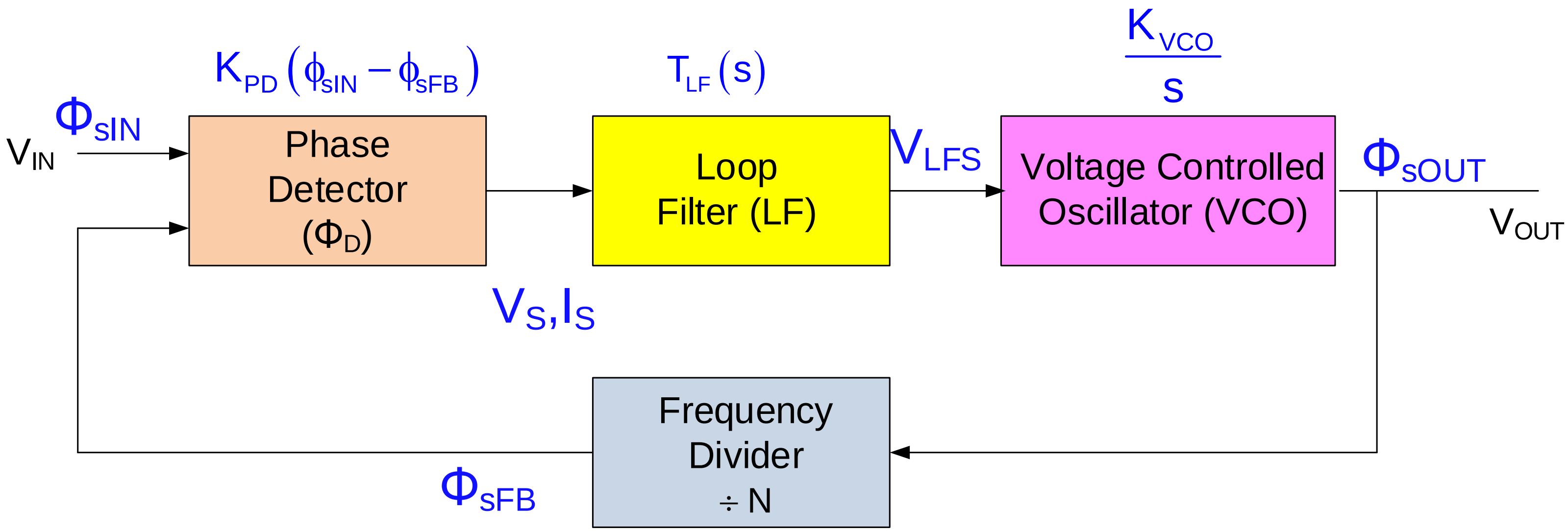


$$T_{PLL}(s) = \frac{T_{LF}(s)K_{PD}K_{VCO}}{s+T_{LF}(s)K_{VCO}K_{PD}} = \frac{K_{PD}K_{VCO}}{s(1+RCs)+K_{VCO}K_{PD}}$$

$$T_{PLL}(s) = \frac{\frac{K_{PD}K_{VCO}}{RC}}{s^2 + s\frac{1}{RC} + \frac{K_{VCO}K_{PD}}{RC}}$$

Often the LF is low order

Example: Assume N=1 and $T_{LF}(s) = \dfrac{1}{1+RCs}$



$$T_{PLL}(s) = \frac{\dfrac{K_{PD}K_{VCO}}{RC}}{s^2 + s\dfrac{1}{RC} + \dfrac{K_{VCO}K_{PD}}{RC}}$$

$$s^2 + s\frac{1}{RC} + \frac{K_{VCO}K_{PD}}{RC} = s^2 + s\frac{\omega_0}{Q} + \omega_0^2$$

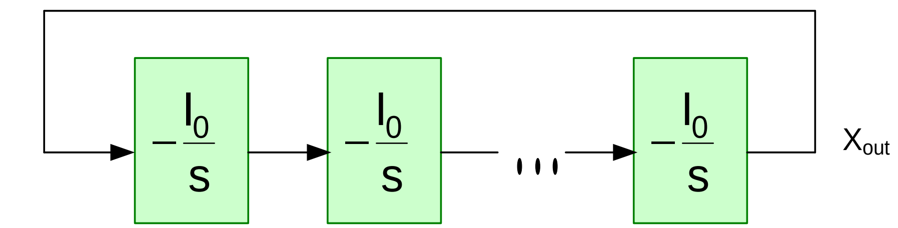
$$\omega_0 = \sqrt{\frac{K_{VCO}K_{PD}}{RC}}$$

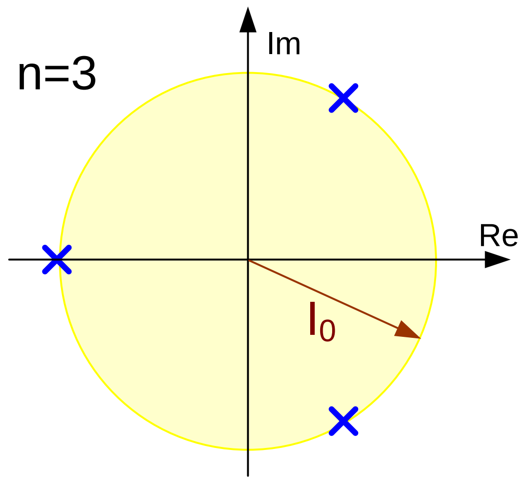
$$Q = \sqrt{RCK_{VCO}K_{PD}}$$

Voltage Controlled Oscillators

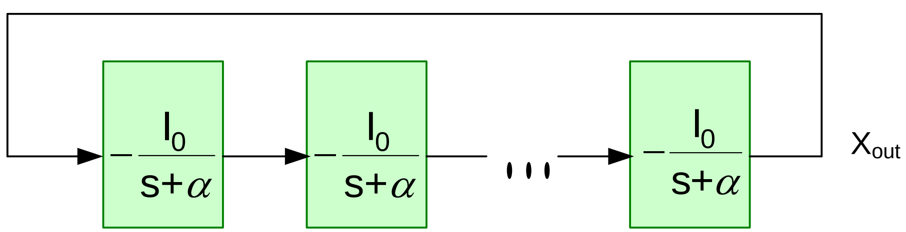
Voltage Controlled Oscillator (VCO)

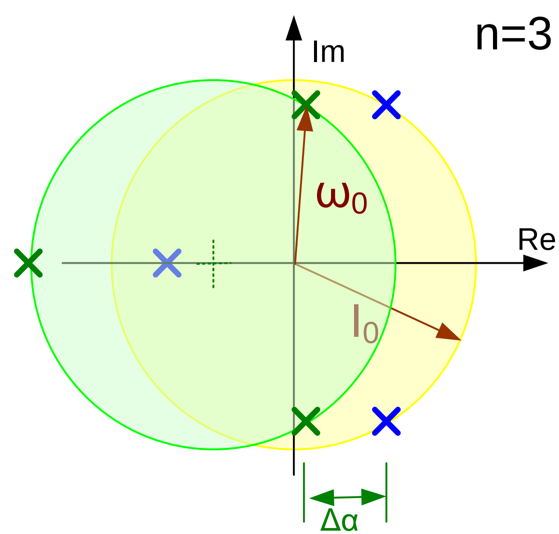
Many different VCOs can be used





Integrator-Based VCO

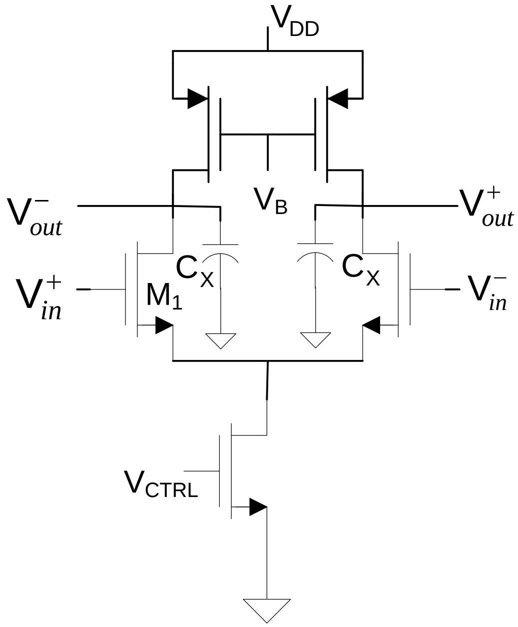


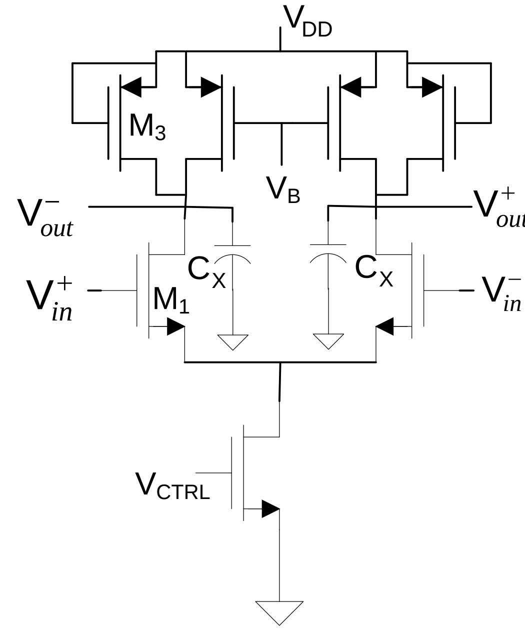


Lossy Integrator-Based VCO

Voltage Controlled Oscillators

Voltage Controlled Oscillator (VCO)





$$I_{0d}(s) = \frac{g_{m1}}{sC_X}$$

$$I_0 = \frac{g_{m1}}{C_X}$$

Integrator for: Integrator-based VCO

$$I_{0d}(s) = \frac{g_{m1}}{sC_X + g_{m3}}$$

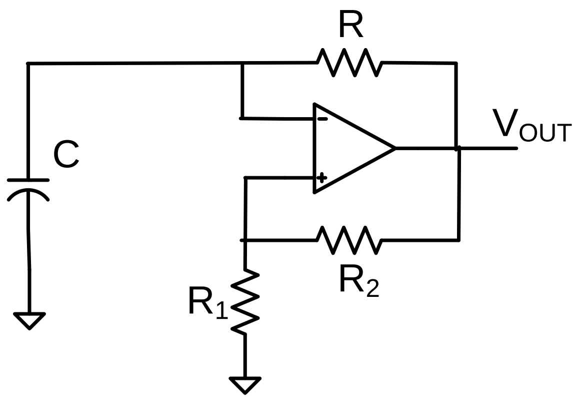
$$I_0 = \frac{g_{m1}}{C_X} \qquad \alpha = \frac{g_{m3}}{C_X}$$

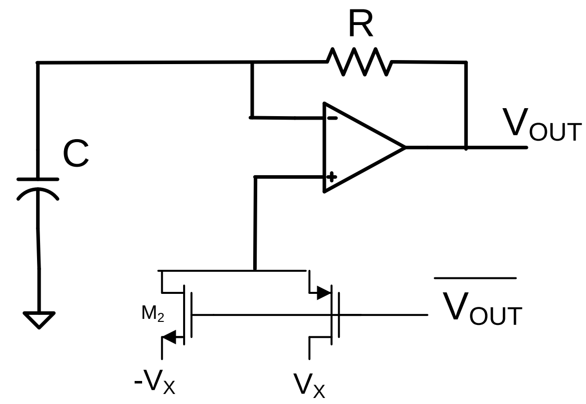
Integrator for: Lossy Integrator-based VCO

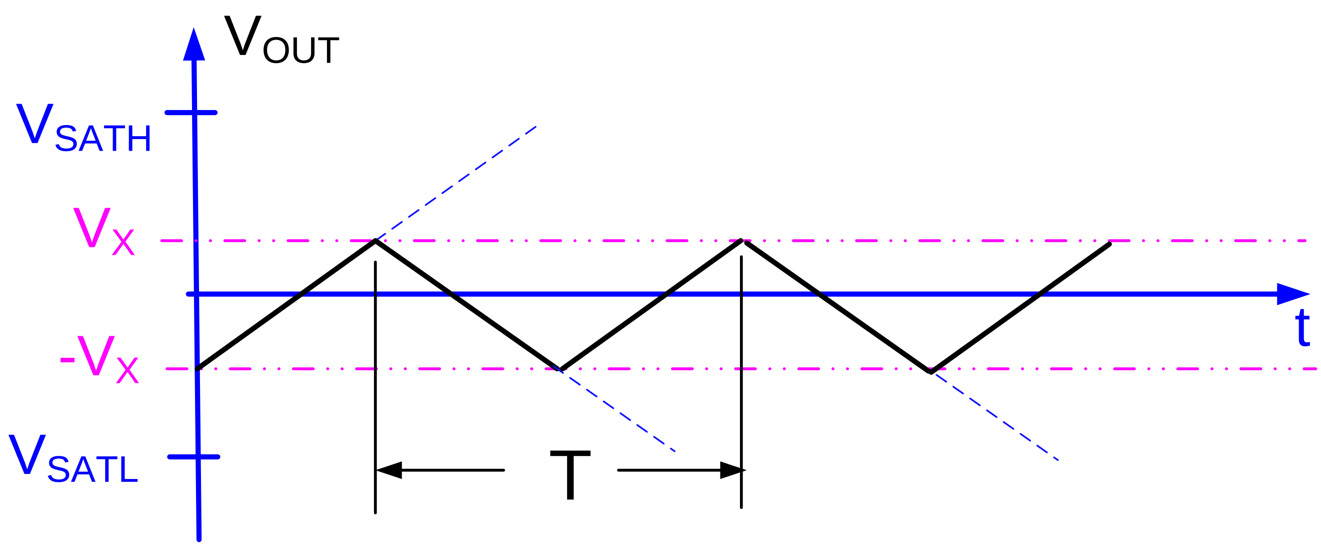
Voltage Controlled Oscillators

Voltage Controlled Oscillator (VCO)

Relaxation Oscillator Derived VCO

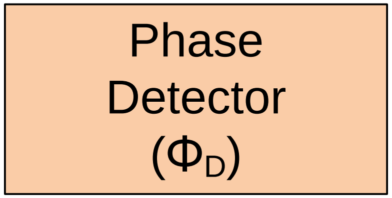






Can have either triangle wave or square wave ouputs

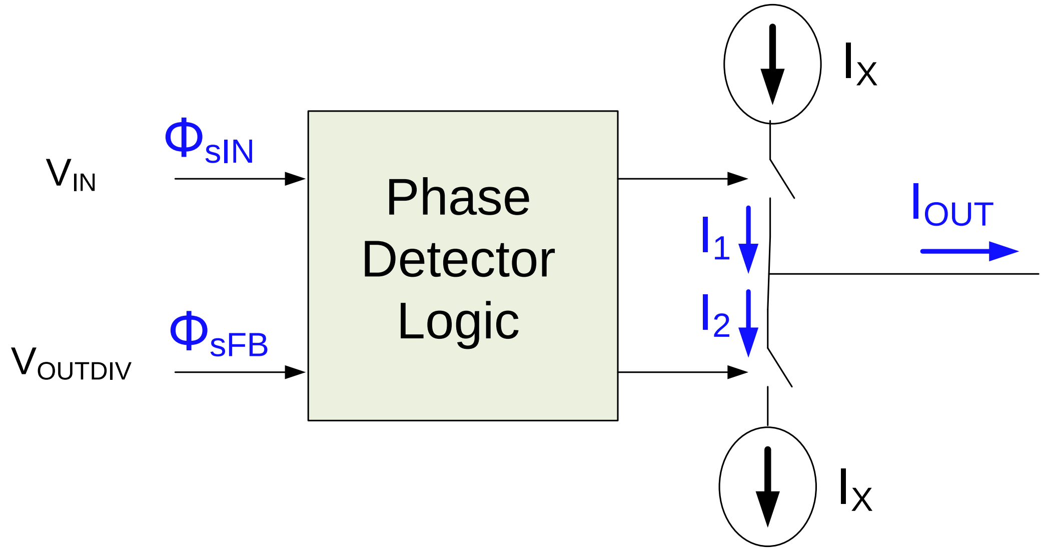
Phase Detectors



Many different Phase Detectors can be used

Some Popular Phase Detector Circuits

- Analog Multiplier
- Exclusive OR Gate
- Sample and Hold
- Charge Pump



Charge-pump based Phase Detector

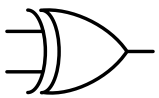
Average I_{OUT} is the average phase

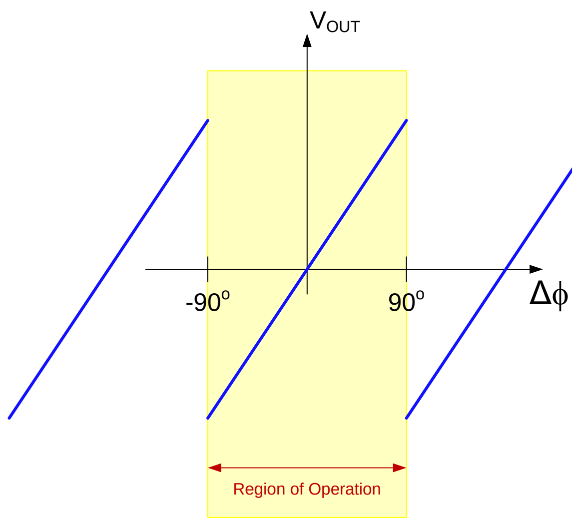
Phase Detectors

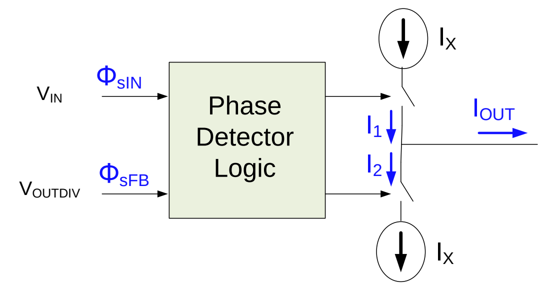
Phase
Detector
(Φ_D)

Many different Phase Detectors can be used

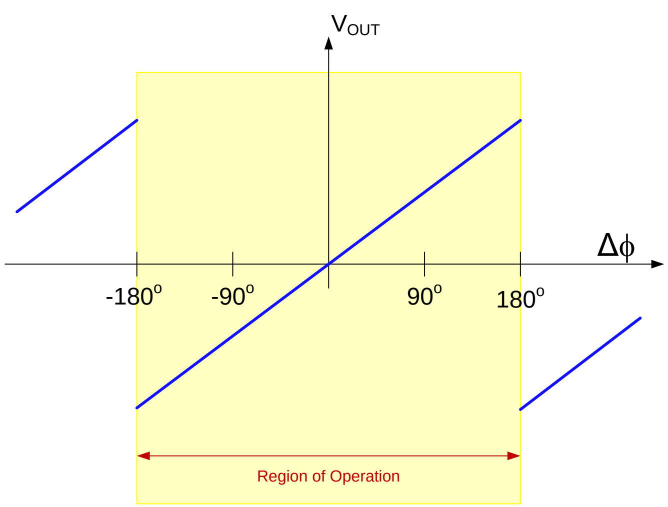
Could be simply

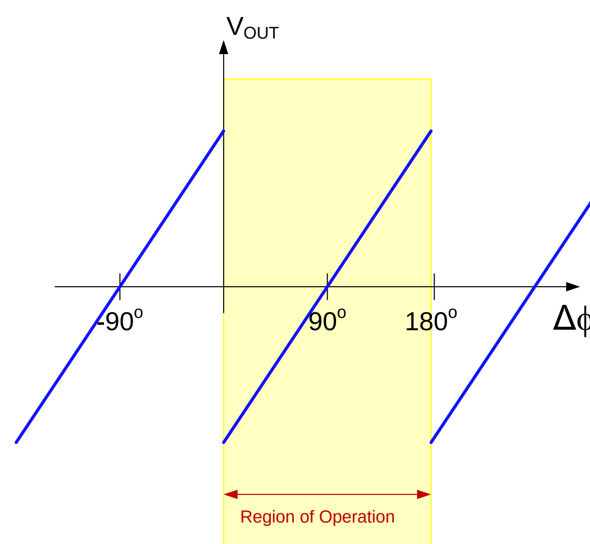




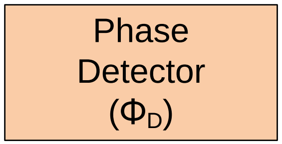


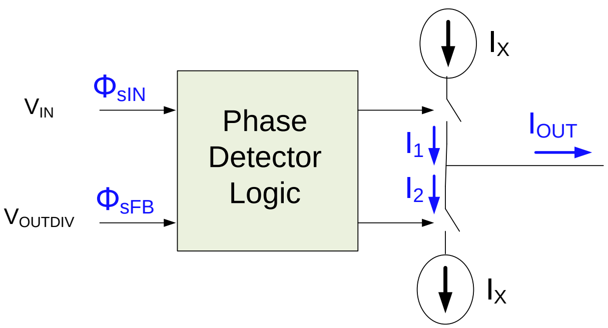
Average I_{OUT} is the average phase



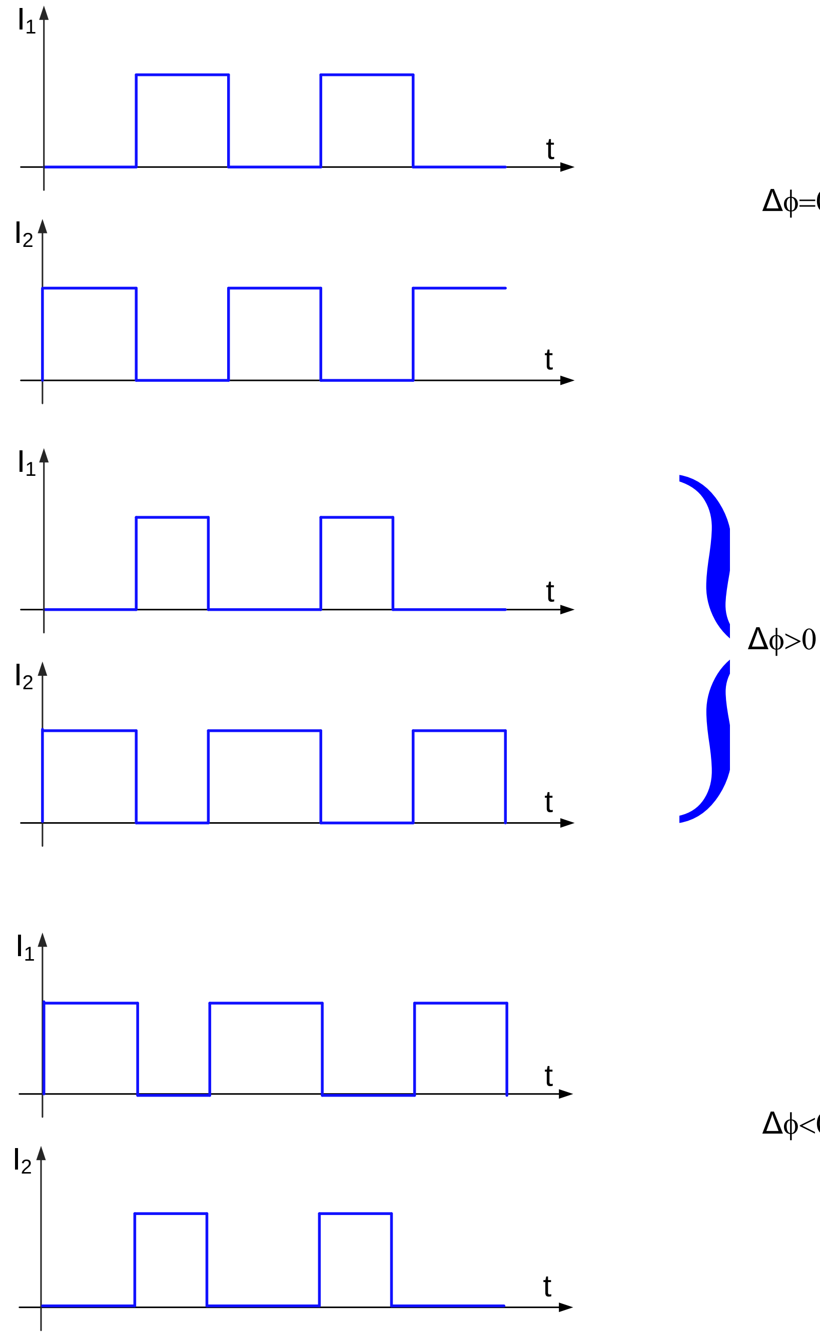


Phase Detectors

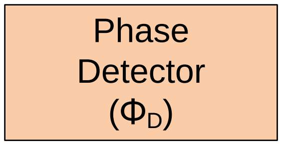


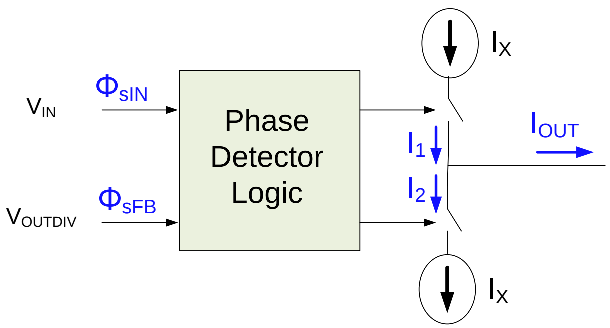


Average I_{OUT} is the average phase

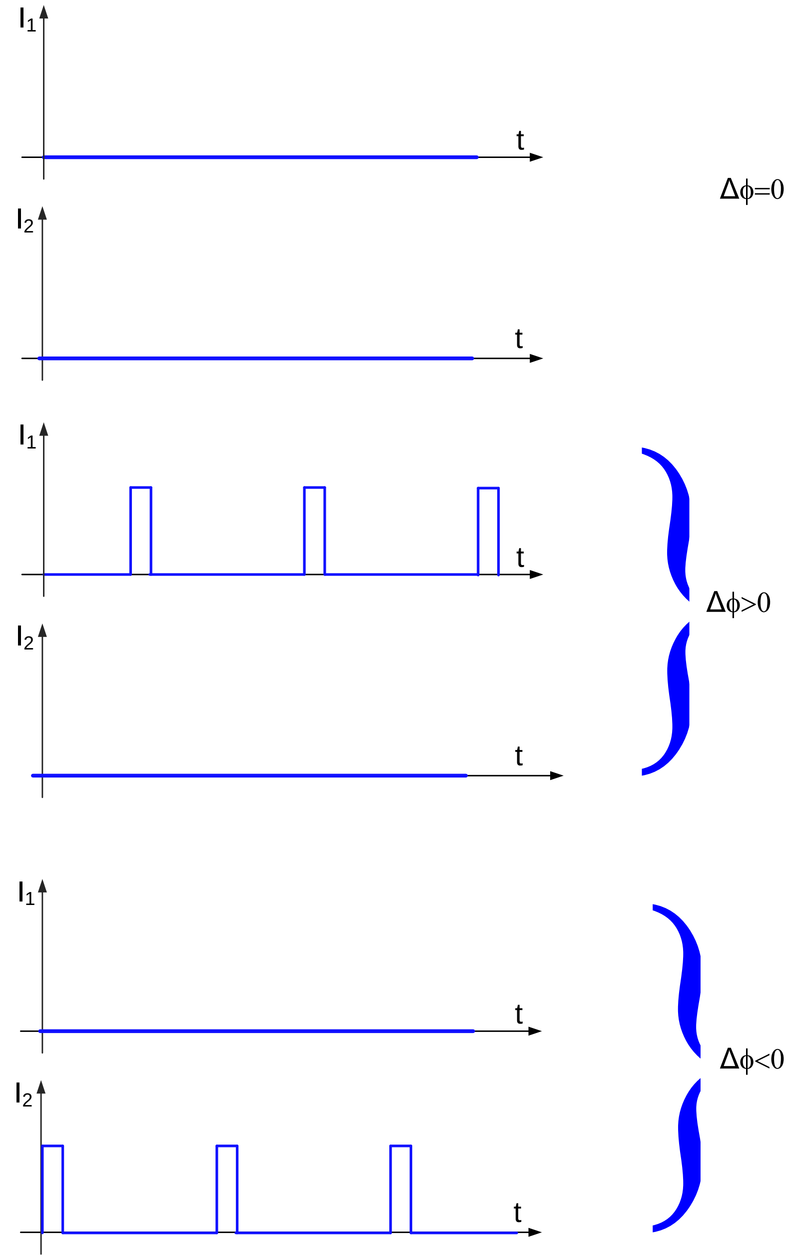


Phase Detectors





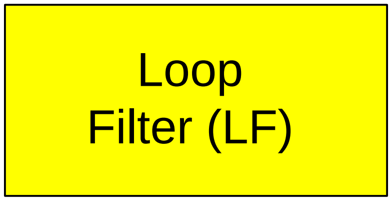
Average I_{OUT} is the average phase



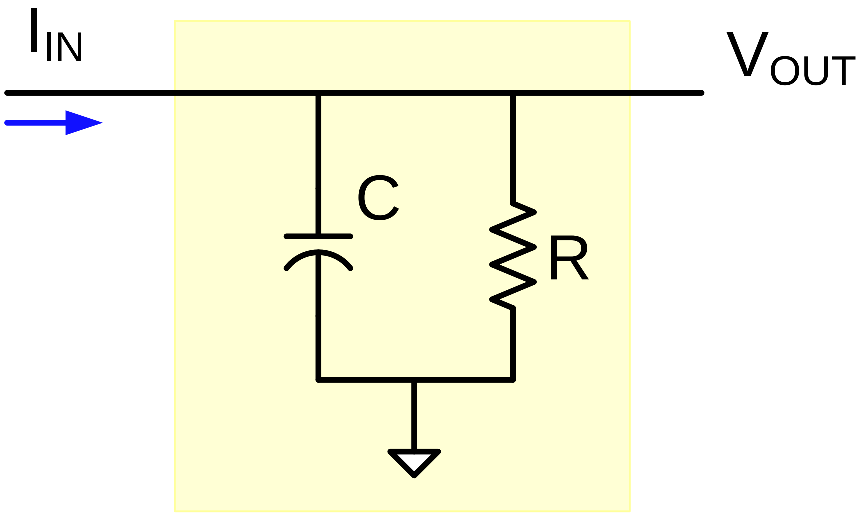
Vulnerable to dead zone problem

Loop Filters

$T_{LF}(s)$



Many different Phase Detectors can be used
Often the loop filter is first or second order
Usually the loop filter circuit is very simple



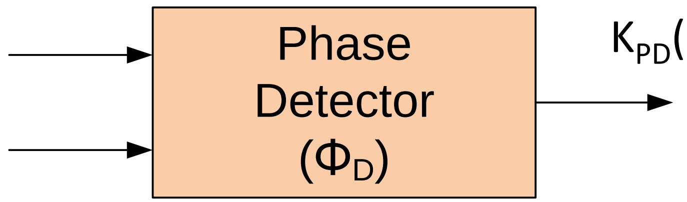
$$\frac{V_{OUT}}{I_{INAVG}} = T_{LF}(s) = \frac{R}{1+RCs}$$

Basic first-order LF with average current difference as input

What is the phase of a signal?

$V_1=V_{M1}\sin(\omega_1 t+ \phi_1)$

$V_2=V_{M2}\sin(\omega_2 t+ \phi_2)$



$K_{PD}(\phi_1-\phi_2)$

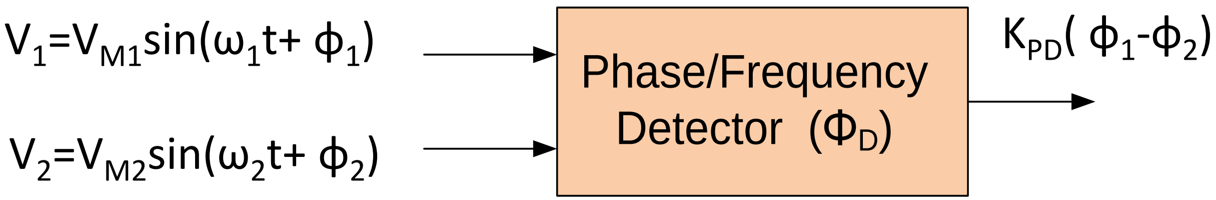
Assume $\omega_1 = \omega_2=\omega$

If V_I can be expressed as $V_1=V_{M1}\sin(\omega t+ \phi_1)$ the phase is ϕ_1

But what is the phase if ω is time varying? Or what is the “phase” if this functional form does not really characterize V(t) ? Or what if $\omega_1 \neq \omega_2$?

What does a phase detector do if the two inputs are not at the same frequency?

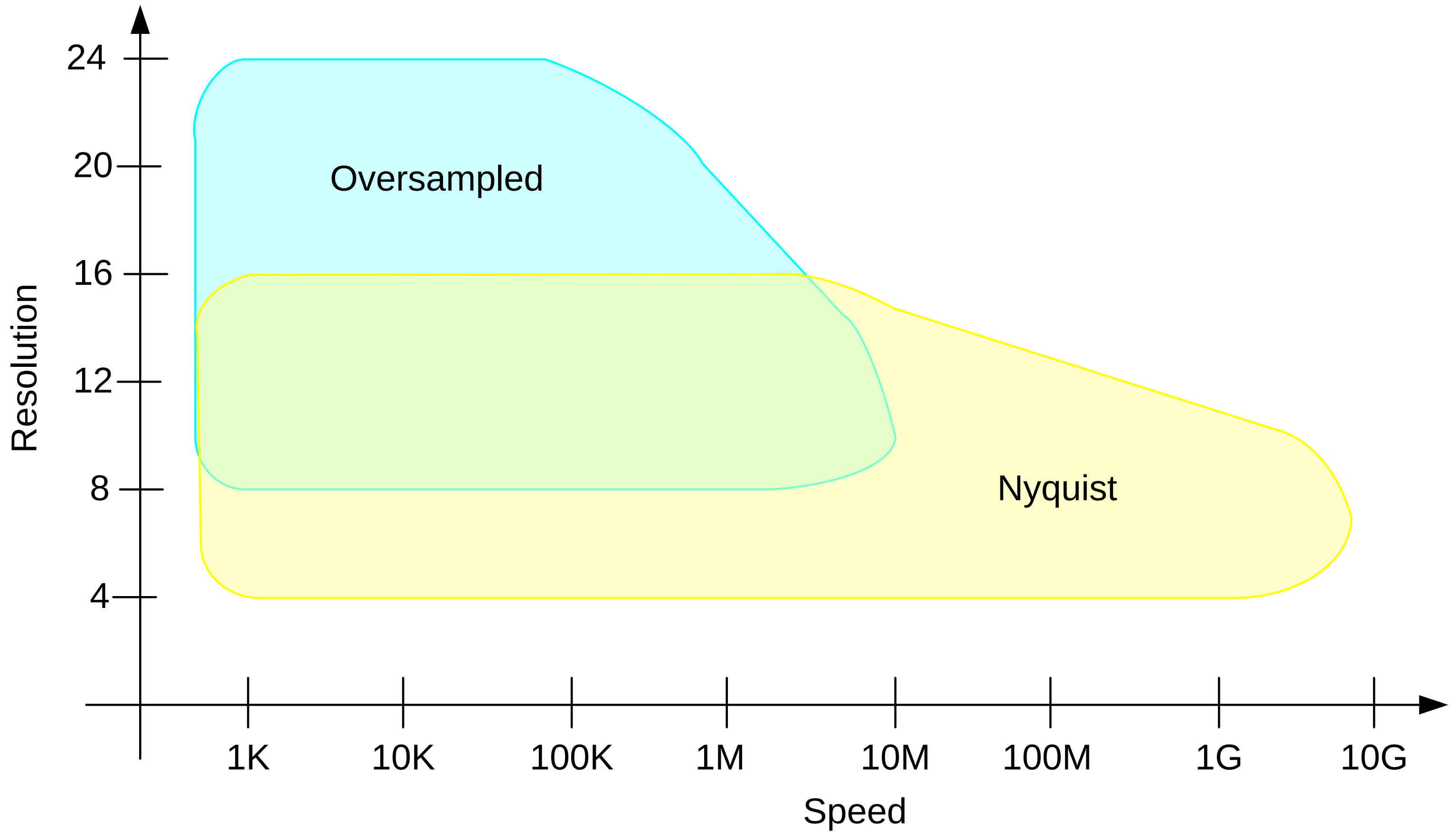
What is the phase of a signal?



Most Phase Detectors are actually Phase/Frequency Detectors

- Large output when frequency difference exists
- Also provides output when phase difference exists after frequencies are matched

Data Converter Type Chart

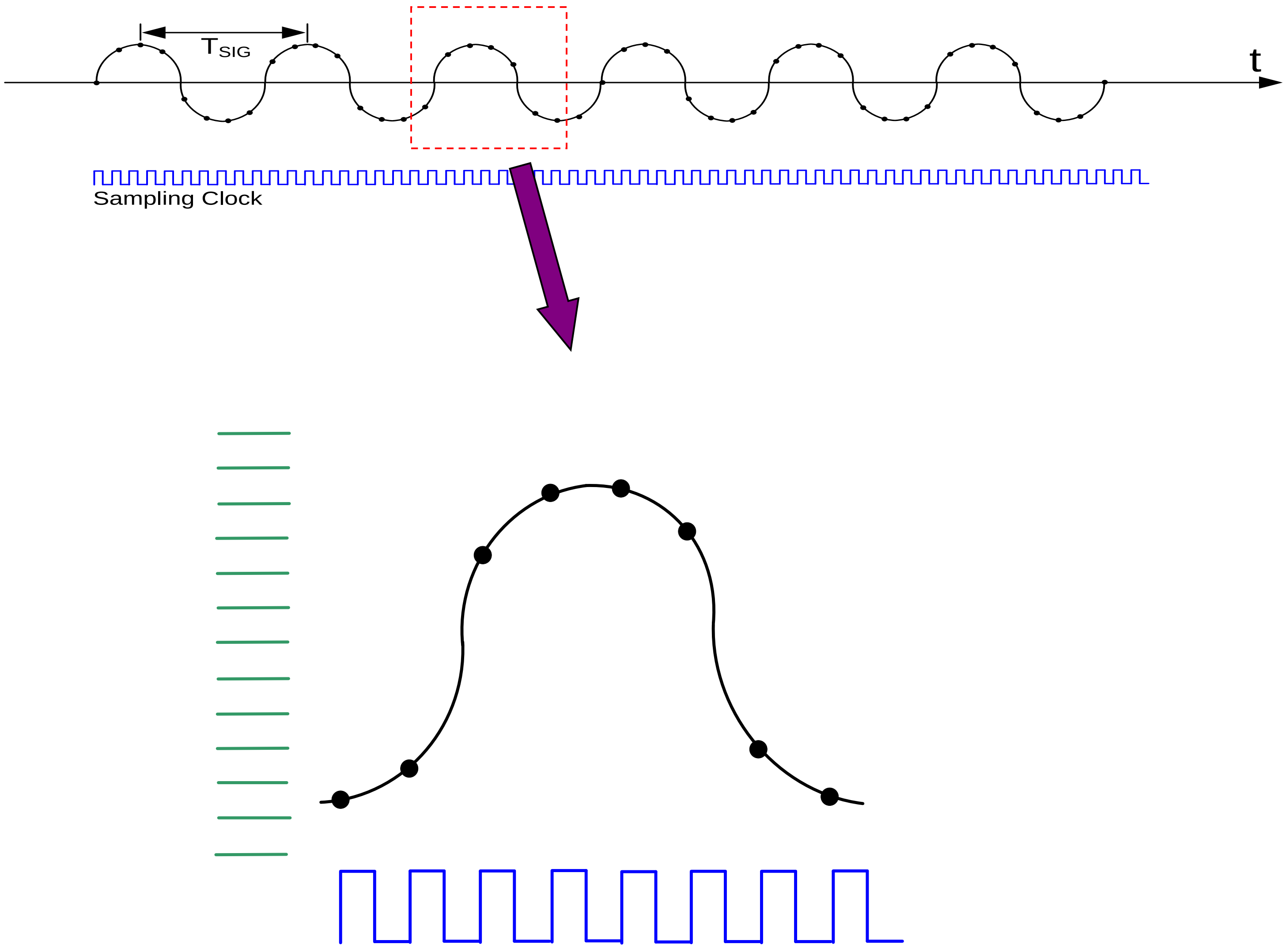


Over-Sampled Data Converters

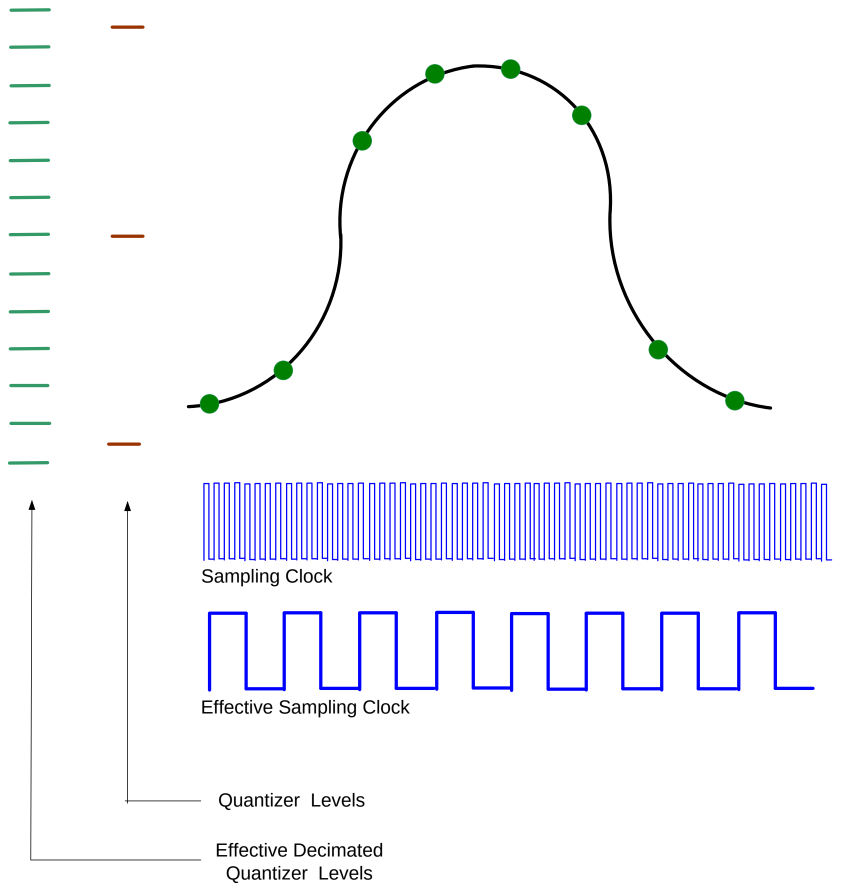
General Classes

- Single-bit
- Multi-bit
- First-order
- Higher-order
- Continuous-time

Nyquist Rate



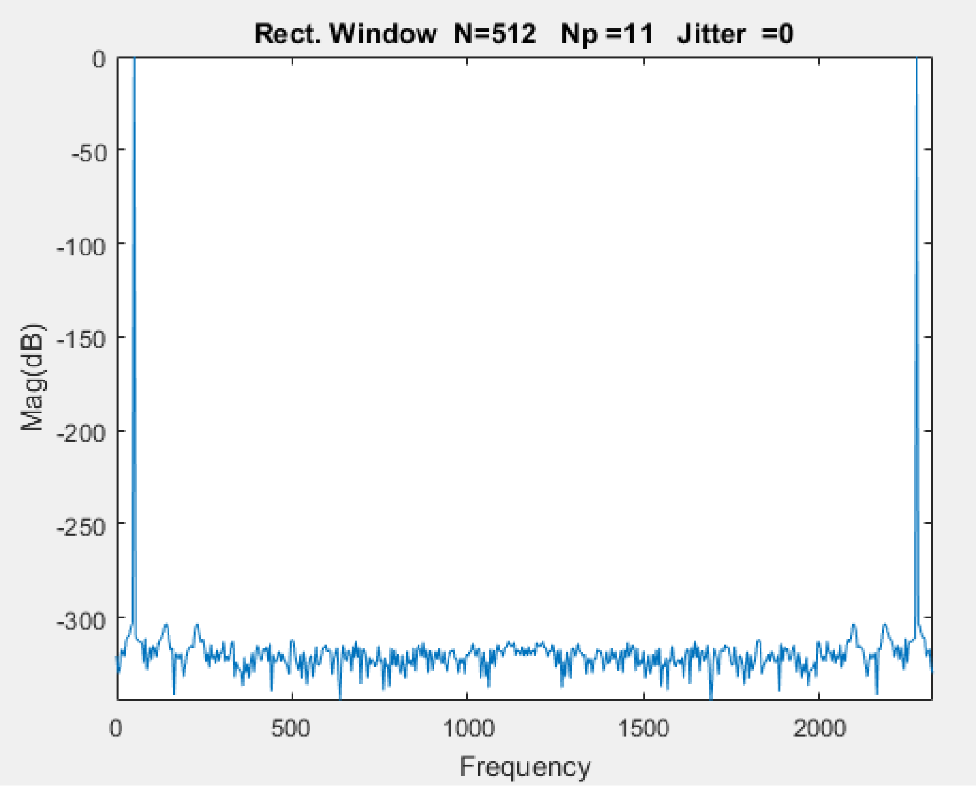
Over-Sampled



Over-sampling ratios of 128:1 or 64:1 are common
Dramatic reduction in quantization noise effects
Limited to relatively low frequencies

Recall:

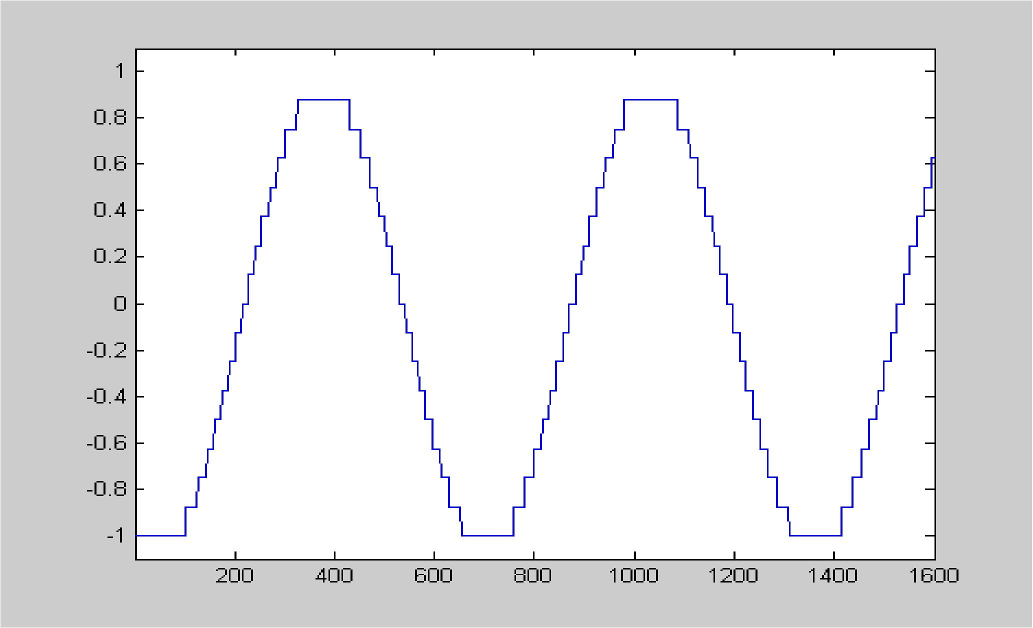
f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=2.3KHz
Oversampled: 23:1



MatLab Results

Recall:

Quantization Effects



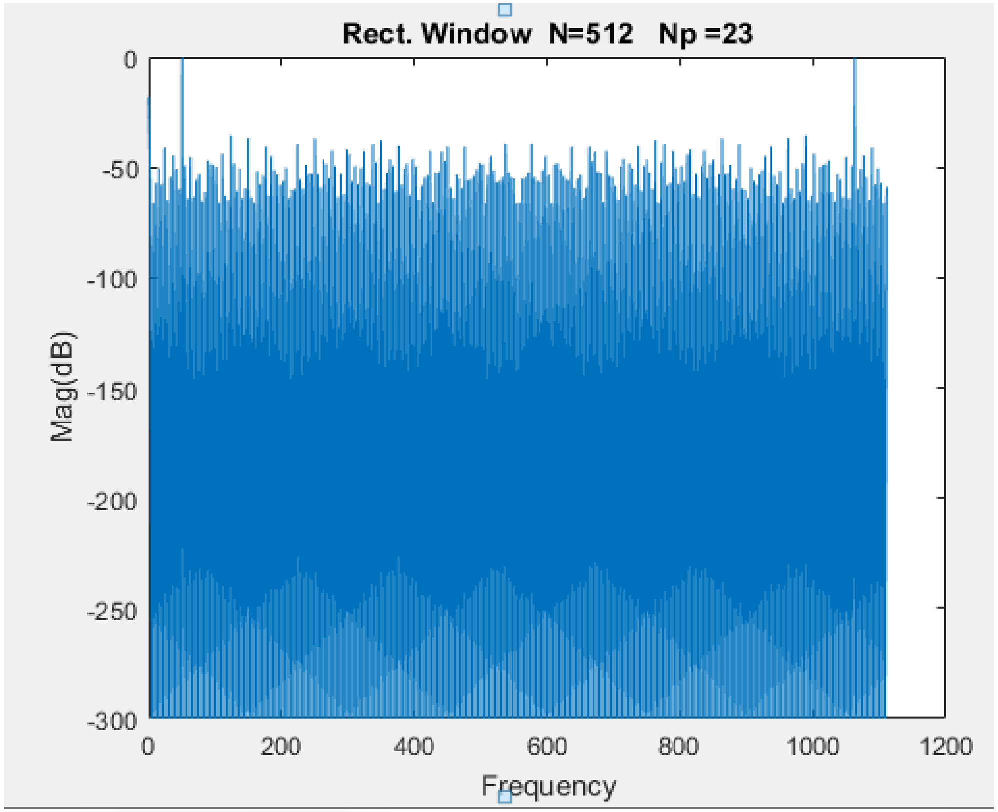
Simulation environment:

N_P=23
f_{SIG}=50Hz

Recall:

Quantization Effects

Res = 4 bits



f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=1113KHz
Oversampled: 11:1

RMS Quantization Noise:

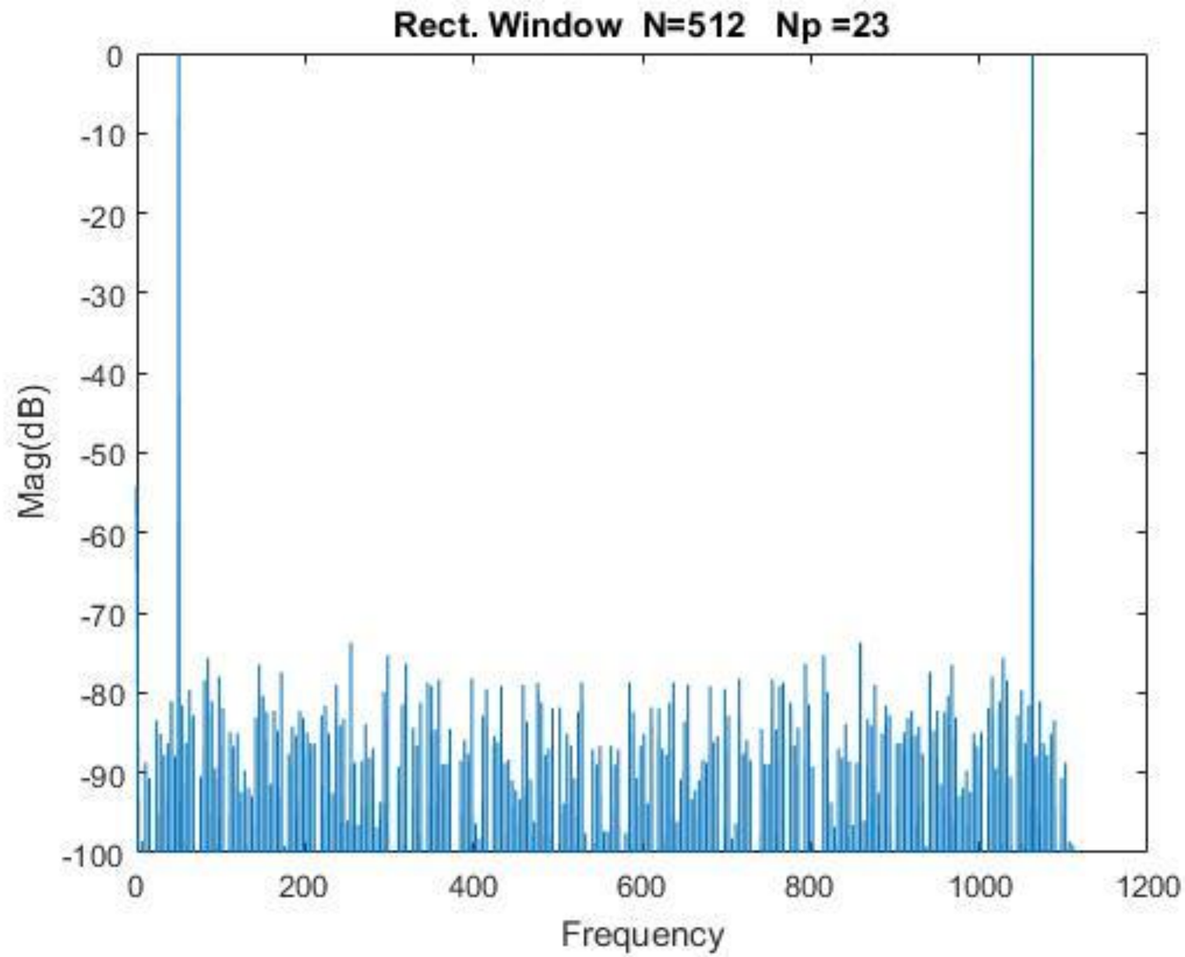
$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

Lets now increase resolution

Recall:

Quantization Effects

Res = 10 bits



f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=1113KHz
Oversampled: 11:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

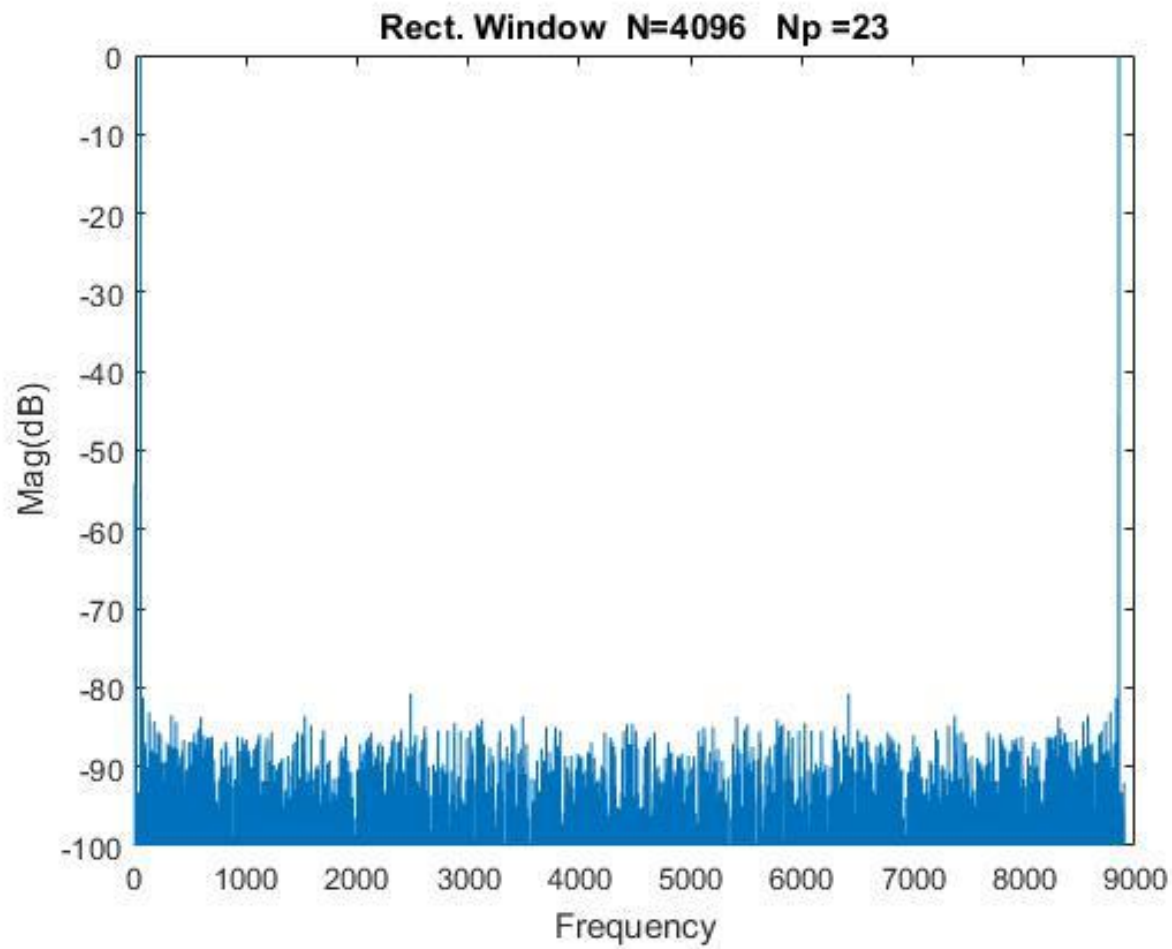
Quantization noise is much lower but still significant

Lets now increase oversampling ratio (i.e. number of samples)

Recall:

Quantization Effects

Res = 10 bits



f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

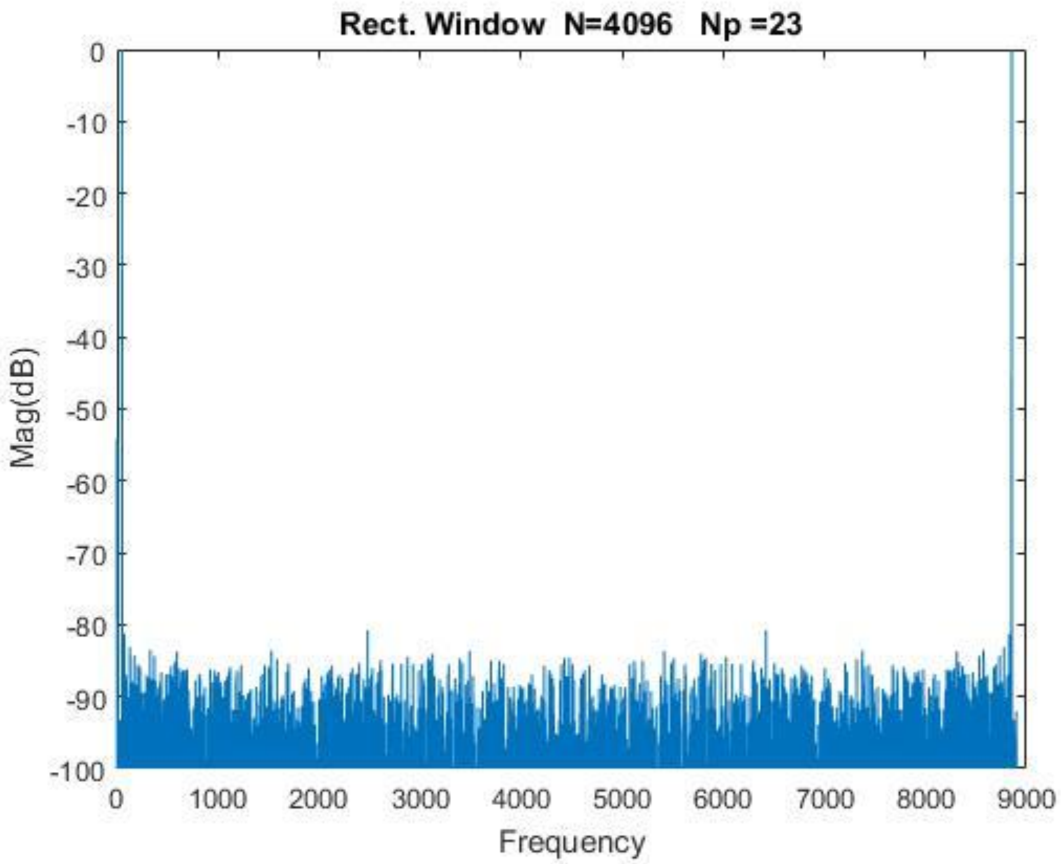
Compared to the previous slide, it appears that the quantization noise has gone down

But has it ?   Magnitude of quantization DFT terms decreased but E_{RMS} unchanged

Recall:

Quantization Effects

Res = 10 bits

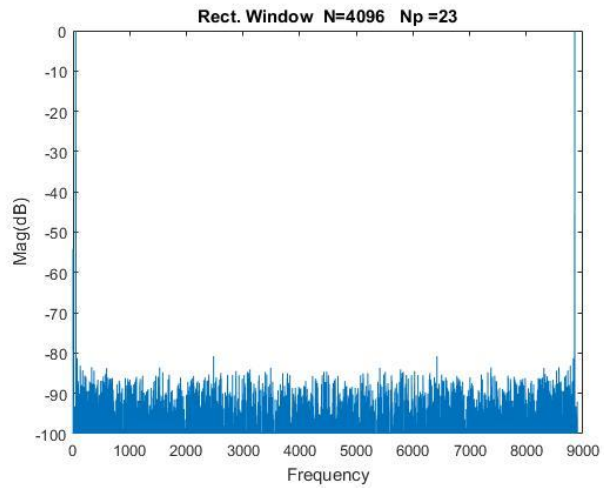


f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

Can any additional useful information about the input be obtained since we have many more samples than are needed?

Over-Sampling



Res = 10 bits

f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

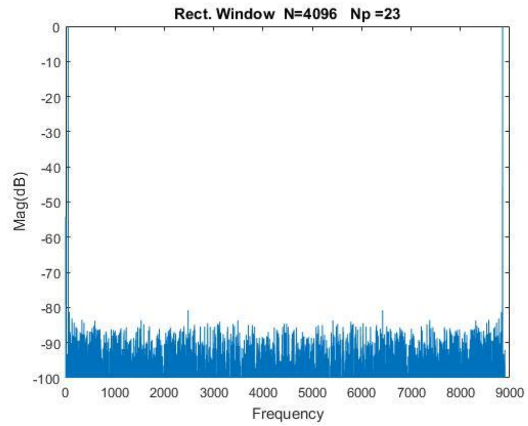
What would happen if we break the 4096 samples into groups of 20 samples and form?

$$\hat{X}_{OUT}(k \bullet 20T_{SAMP}) = \frac{1}{20}\sum_{j=1}^{20} x_{OUT}\left(jT_{SAMP} + 20kT_{SAMP}\right)$$

$$E_{RMS} = ?$$

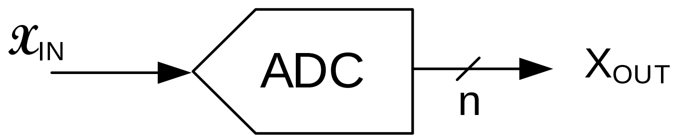
- Though the individual samples have been quantized to 10 bits, the arithmetic operations will have many more bits
- The effective sampling rate has been reduced by a factor of 20 but is still over 4 times the Nyquist rate
- Has the quantization noise been reduced (or equivalently has the resolution of the ADC been improved?
- Is there more information available about the signal?

Over-Sampling



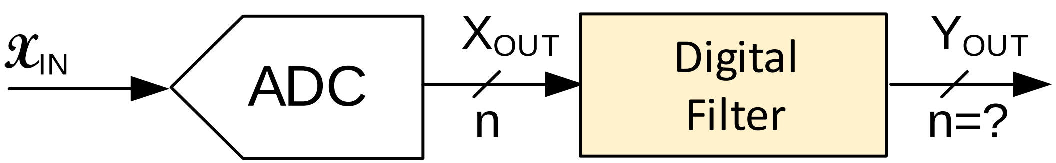
Res = 10 bits

f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1



$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

Since the quantization noise is at high frequencies, what would happen if filtered the Boolean output signal?

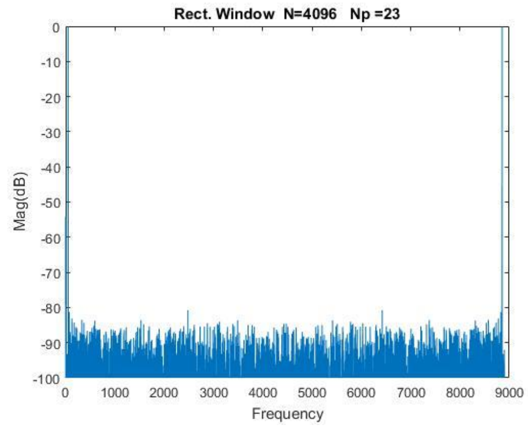


$$E_{RMS} = ?$$

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}\left(k - jT_{SAMP}\right)$$

Or $$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}\left(k - jT_{SAMP}\right) + \sum_{j=1}^{h} b_j Y_{OUT}\left(k - jT_{SAMP}\right)$$

Over-Sampling



Res = 10 bits

f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

What does this difference equation represent?

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}\left(k - jT_{SAMP}\right)$$

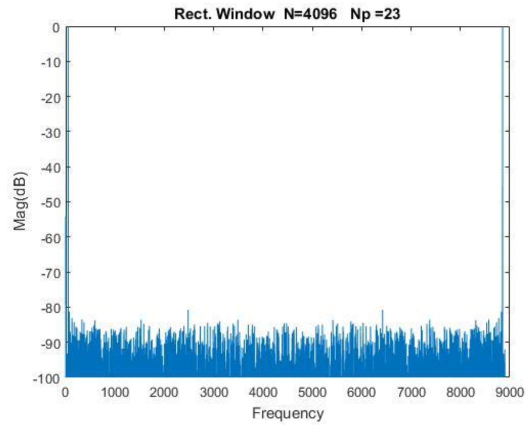
- Moving Average (MA) Digital Filter
- Filter shape (e.g. low-pass, band-pass, high-pass, … dependent upon <a_i> coefficients)

What does this difference equation represent?

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}\left(k - jT_{SAMP}\right) + \sum_{j=1}^{h} b_j Y_{OUT}\left(k - jT_{SAMP}\right)$$

- Auto Regressive Moving Average (ARMA) Digital Filter
- Filter shape (e.g. low-pass, band-pass, high-pass, … dependent upon <a_i> and <b_j> coefficients)

Over-Sampling



Res = 10 bits

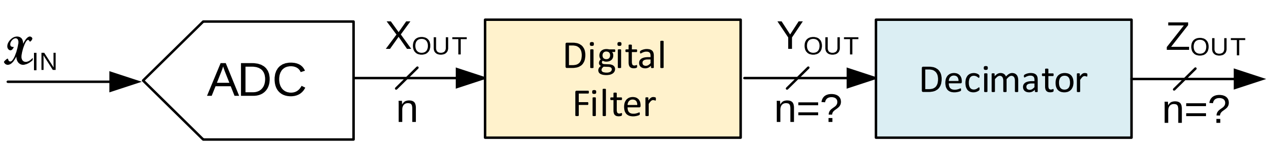
f_{SIG}=50Hz
f_{NYQ}=100Hz
f_{SAMP}=8904KHz
Oversampled: 89:1

$$E_{RMS} = \frac{x_{LSB}}{\sqrt{12}}$$

Since the quantization noise is at high frequencies, what would happen if filtered and decimated the Boolean output signal?

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}\left(k - jT_{SAMP}\right)$$

$$Y_{OUT}(kT_{SAMP}) = \sum_{j=0}^{m} a_j x_{OUT}\left(k - jT_{SAMP}\right) + \sum_{j=1}^{h} b_j Y_{OUT}\left(k - jT_{SAMP}\right)$$



$$E_{RMS} = ?$$

Stay Safe and Stay Healthy !

End of Lecture 41