https://xkcd.com/912/

Control Systems

Dan Boschen

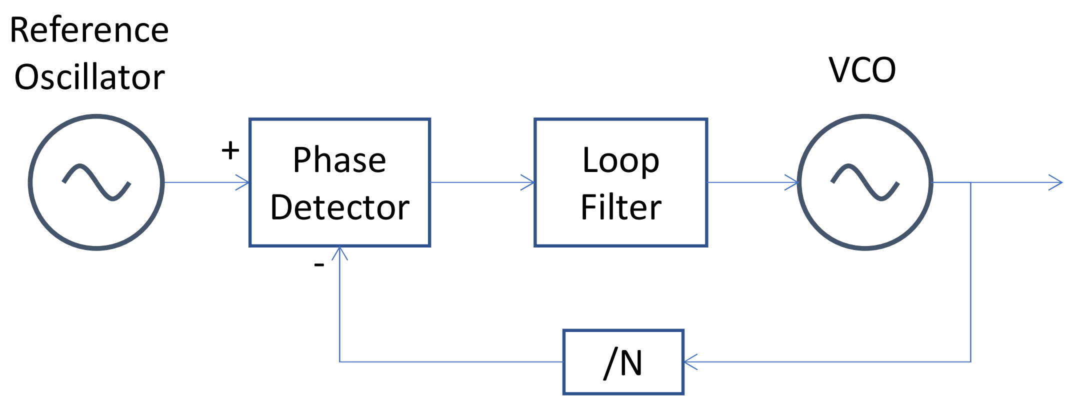
September 2024

Content is derived from courses on DSP and Python taught by Dan Boschen

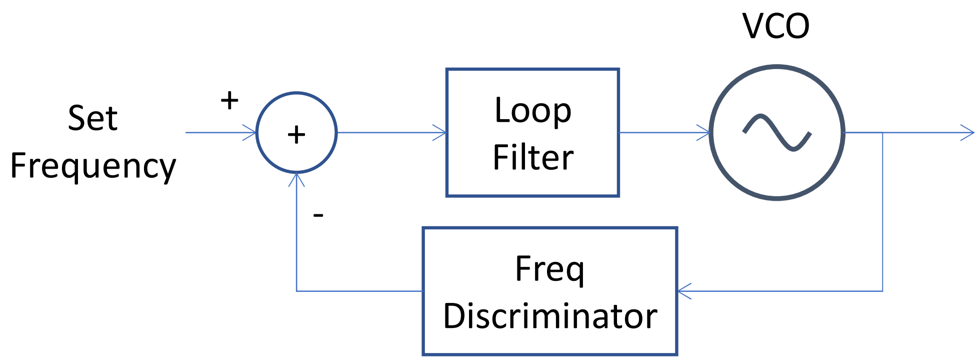
Outline

- Relevant Examples
- Fundamental background for Analog Control Systems
- Analog Control Systems
- Mapping to Digital and the Z-Transform
- Digital Control Systems
- Using Python for Control System Modelling

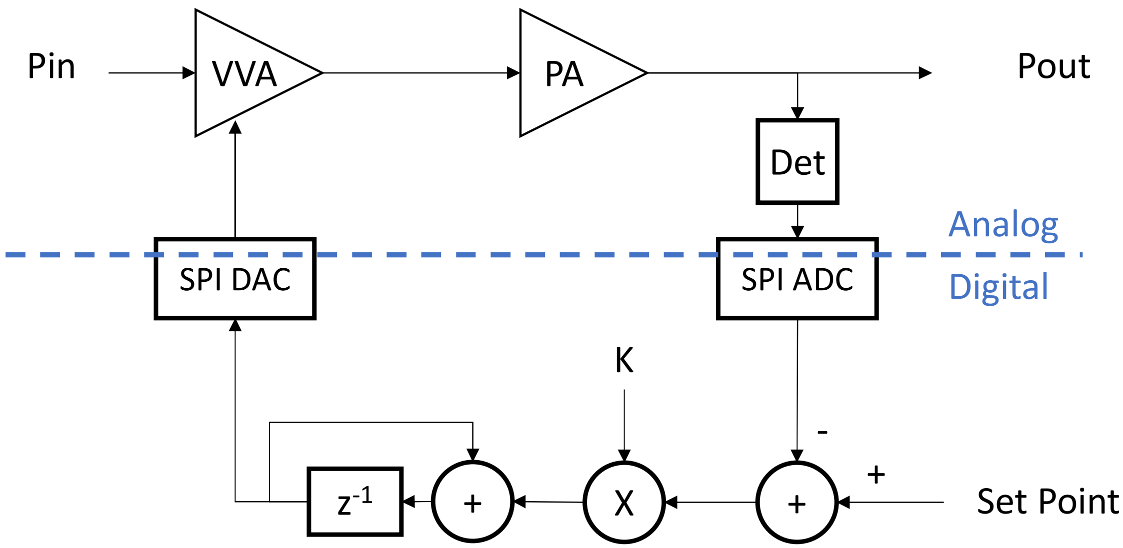
Examples



Integer-N Phase Lock Loop



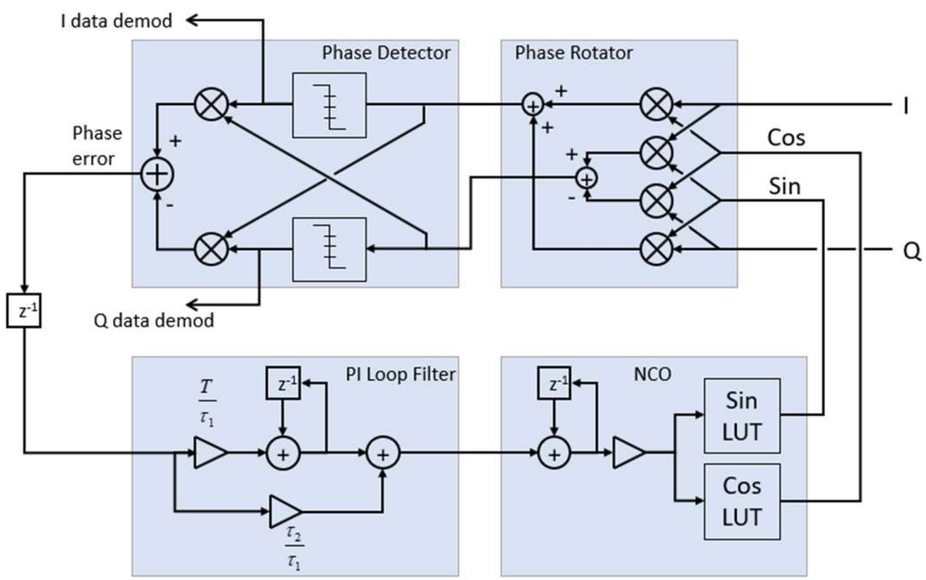
Frequency Lock Loop



Mixed Signal Power Control Loop

Motivating Application

Reduce This:



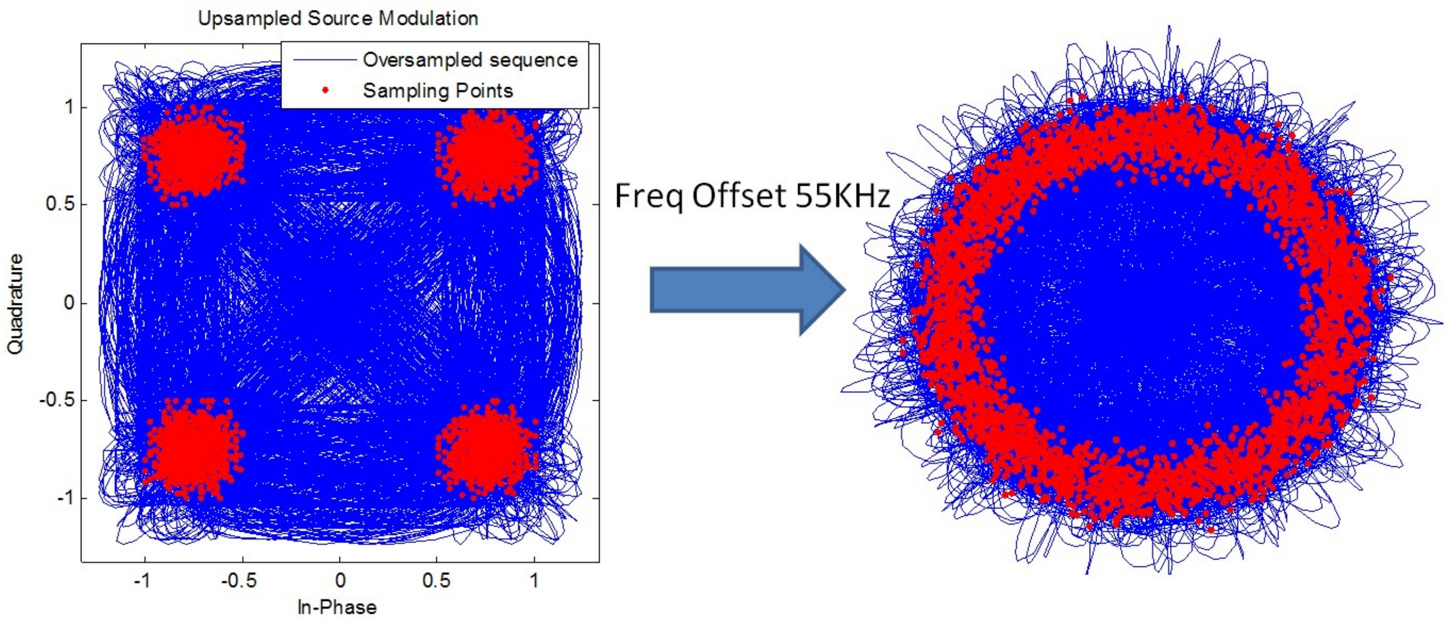
Decision Directed Carrier Recovery

To This:

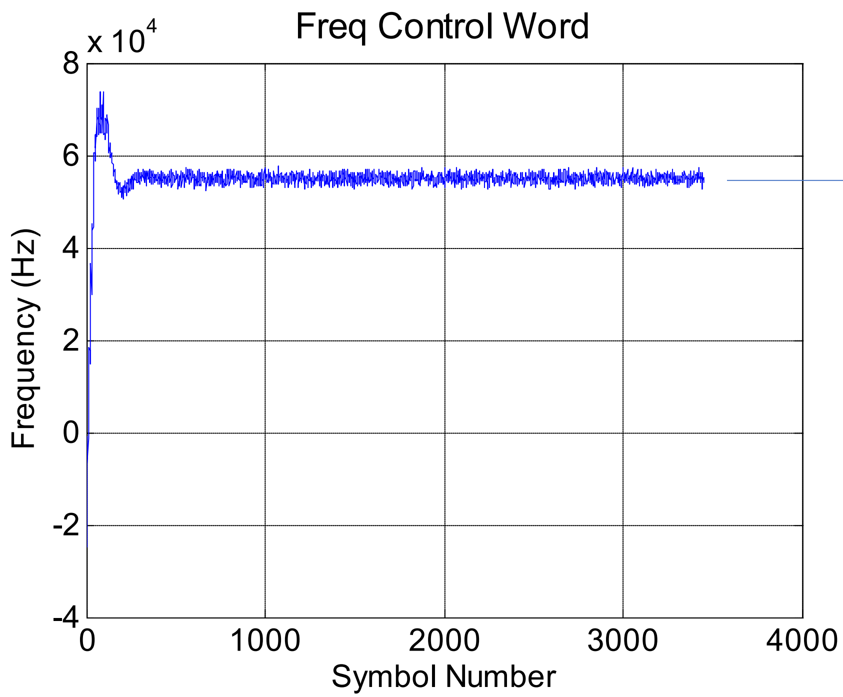
$$G_{OL}(z) = 0.0324\frac{(z-0.9695)}{(z-1)^2}$$

Further details of getting to $G_{OL}(z)$ are in the "DSP for Software Radio" Course.
Bottom line shown here as a motivating intro.

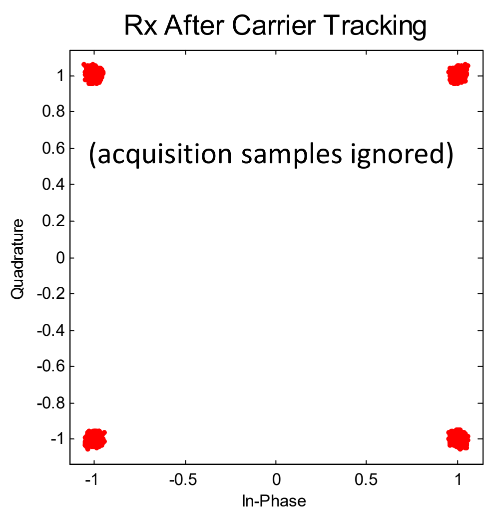
Example Application



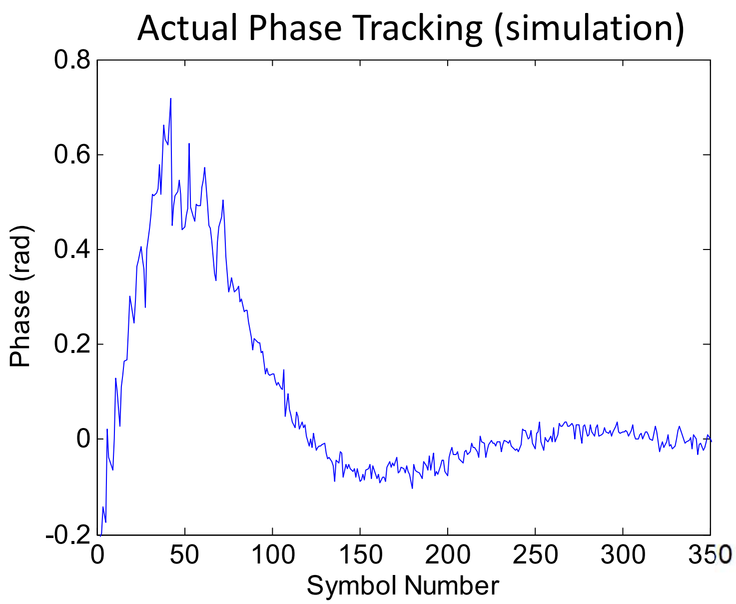
Example Application

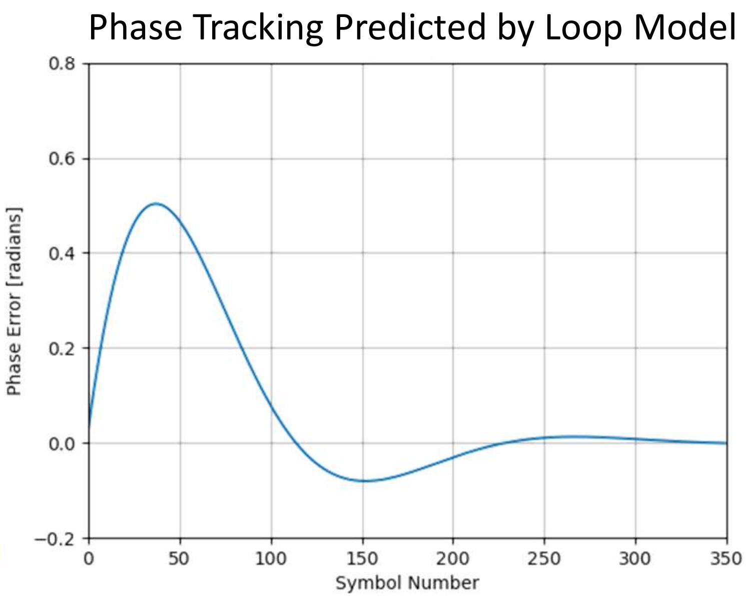


Tracked to 55KHz
Frequency Offset



Example Application





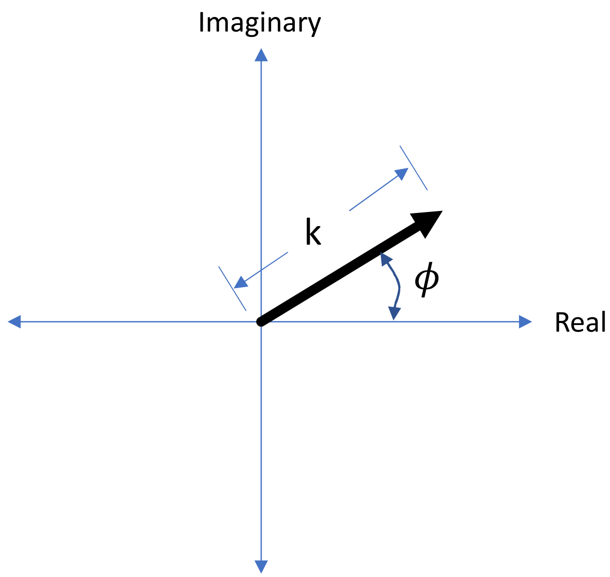
$$G_{OL}(z) = 0.0324\frac{(z - 0.9695)}{(z-1)^2}$$

```
gol = con.tf([0.0324, -0.03142], [1, -2, 1]
mag = 2*np.pi*55e3/12e6
int = con.tf([1, 0],[1, -1], T)
t, yout = con.step_response(mag*int/(1+Gol), T = np.arange(350)*T)
```

Fundamentals

Complex Representation

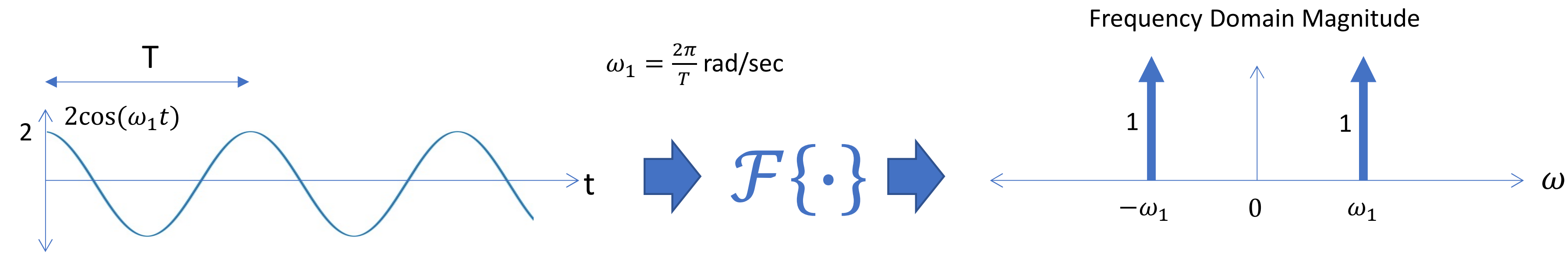
$$\mathrm{k}\, e^{j\phi}$$

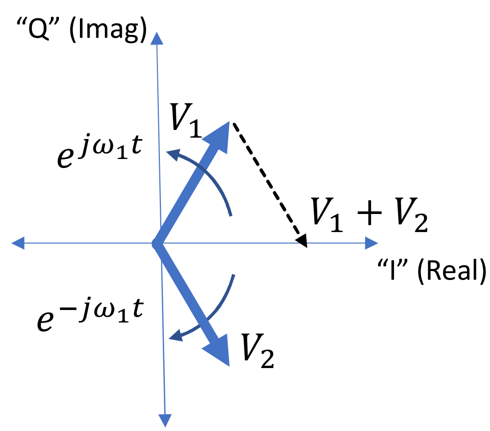


Example Complex Signals

$\mathrm{k(t)}e^{j\phi}$	Magnitude changes with time
$\mathrm{k}e^{j\phi(t)}$	Phase changes with time
$\mathrm{k(t)}e^{j\phi(t)}$	Magnitude and Phase changes with time
$\mathrm{K}(\omega)e^{j\Phi(\omega)}$	Magnitude and Phase changes with frequency

Positive and Negative Frequencies



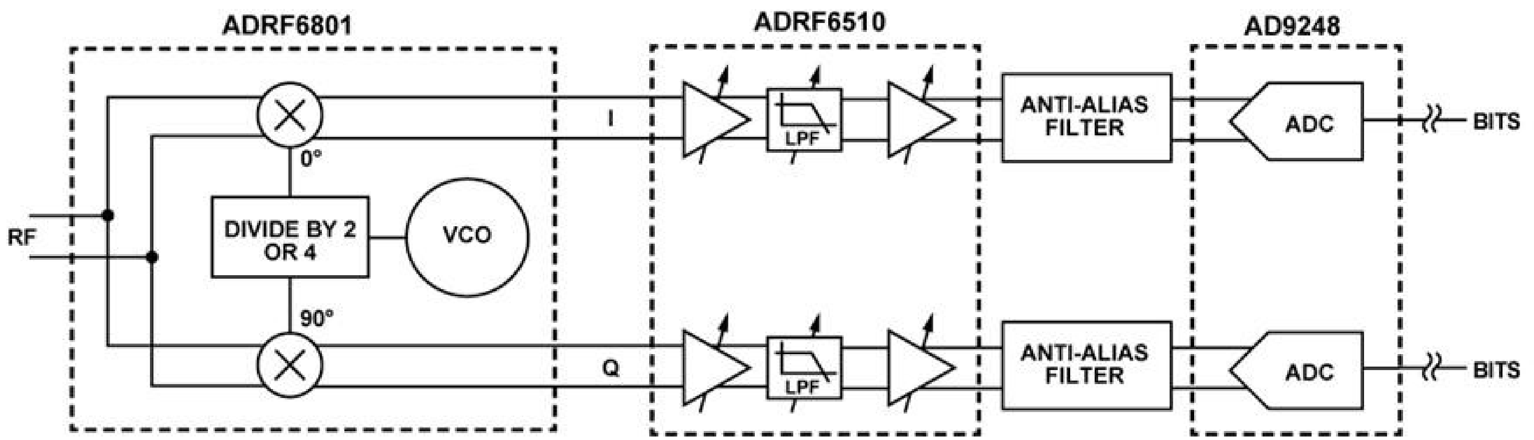


$$2\cos(\omega_1 t) = e^{j\omega_1 t} + e^{-j\omega_1 t}$$

Do Complex Signals "Exist" in the "Real" World??

$$Ke^{j\theta} = K\cos(\theta) + jK\sin(\theta)$$

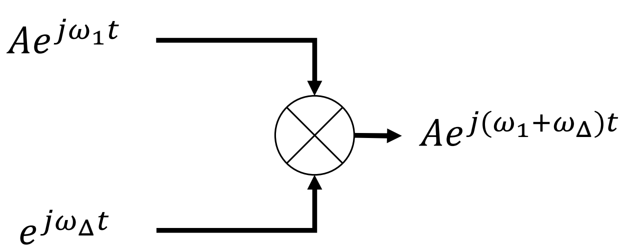
$$= \quad I \quad + \quad jQ$$



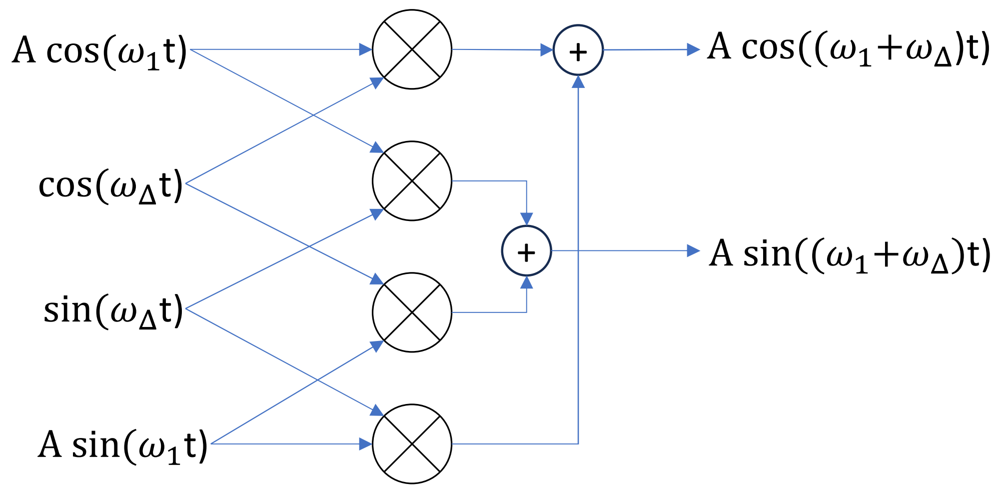
A Motivation for Using $Ke^{j\theta}$

Try explaining how a full complex multiplier can be used to shift frequency...

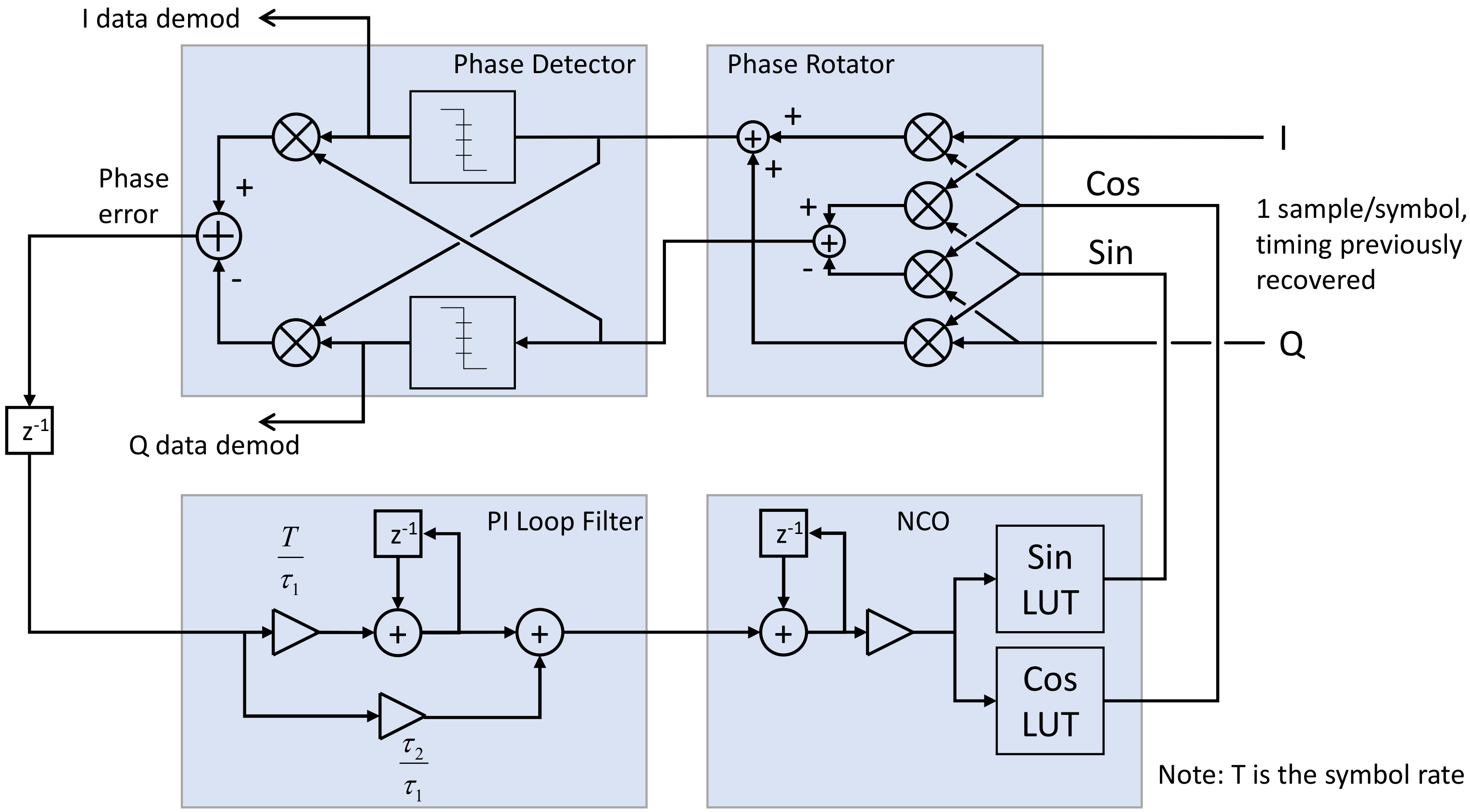
Using Exponentials



Using Sines and Cosines

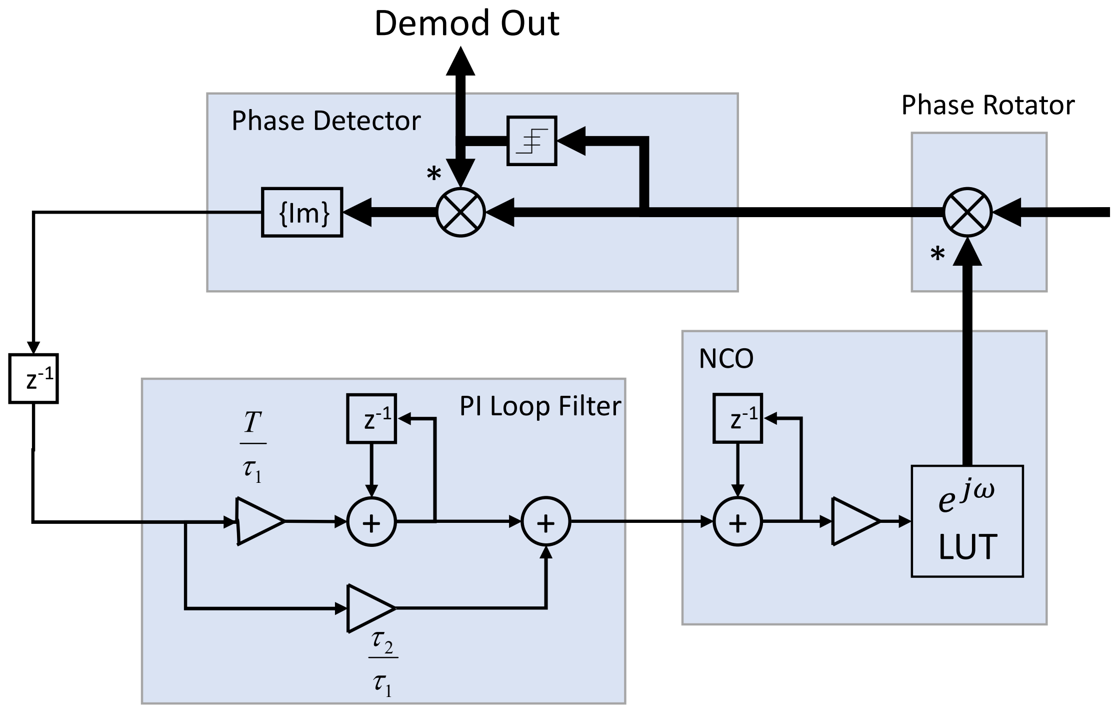


Decision Directed CR Loop



Note: T is the symbol rate

Decision Directed CR Loop



Note: T is the symbol rate

Fourier and Laplace Transforms

Fourier and Laplace

$$X(\omega) \equiv \int_{t=-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

$$X(s) \equiv \int_{t=0}^{\infty} x(t)e^{-st}dt$$

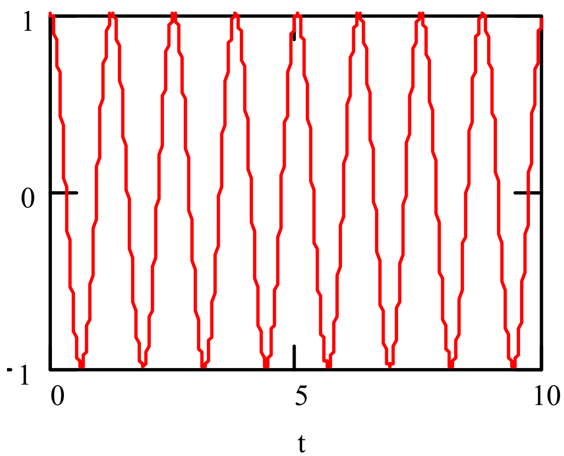
e^{st}

s is a complex variable:

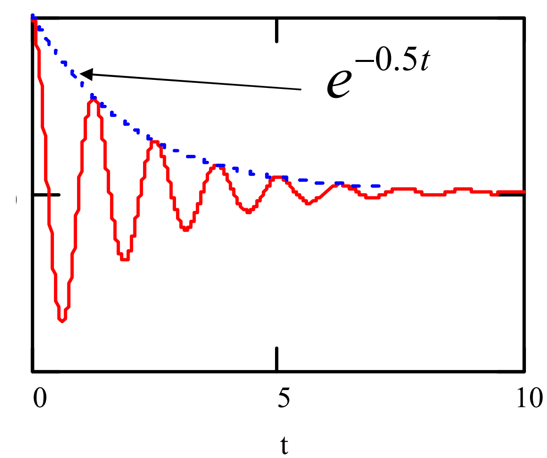
$$s = \sigma + j\omega$$

$$e^{-st} = e^{-(\sigma + j\omega)t} = e^{-\sigma t} e^{-j\omega t}$$

$\mathrm{Re}[e^{-st}], \sigma = 0, \omega = 2\pi(0.8)$

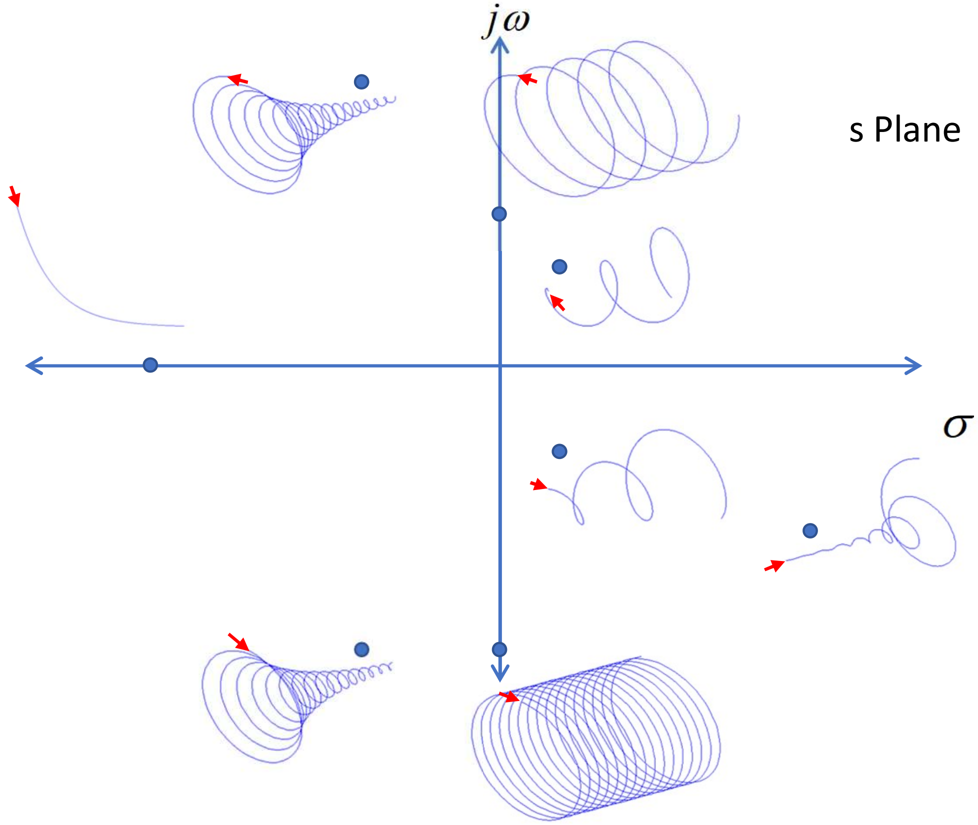


$\mathrm{Re}[e^{-st}], \sigma = 0.5, \omega = 2\pi(0.8)$



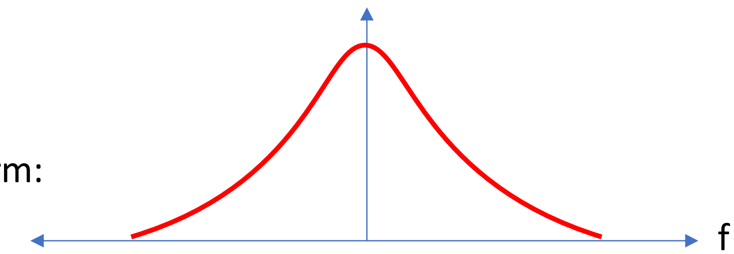
The s and z Planes

e^{st} for all values of s

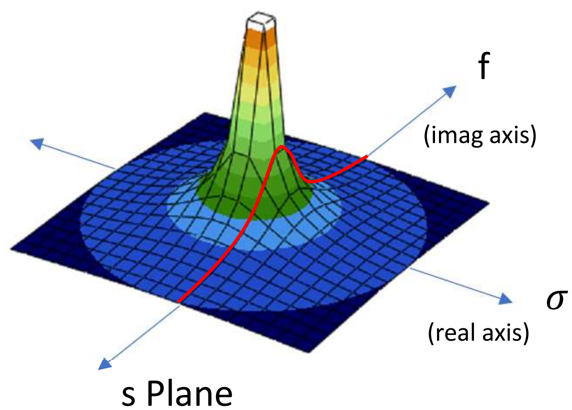


The s-plane

Graphing Magnitude of the Fourier Transform:

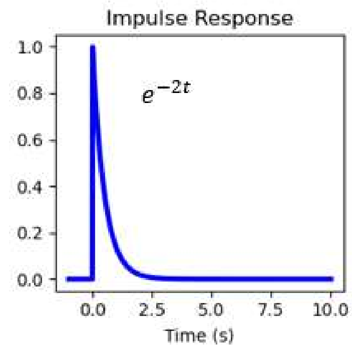


Graphing Magnitude of the Laplace Transform:

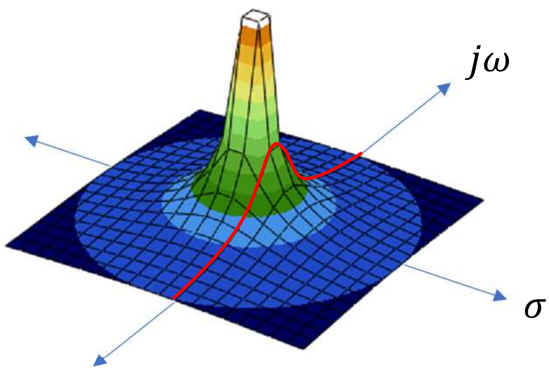


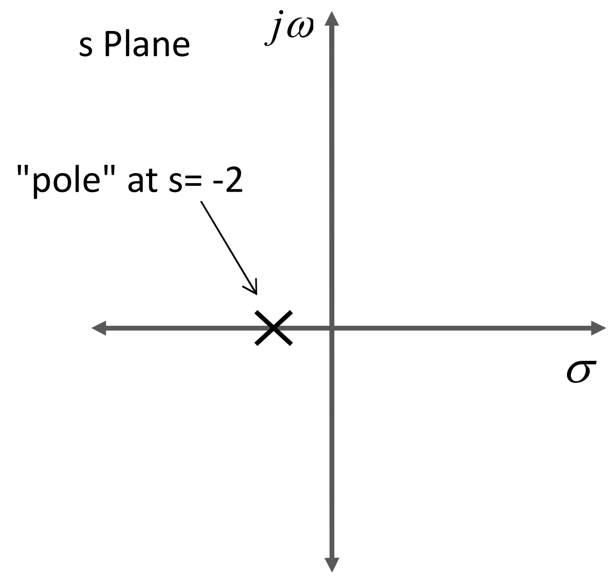
The s-plane

$$x(t) = e^{-2t}$$



$$X(s) = \frac{1}{s+2}$$

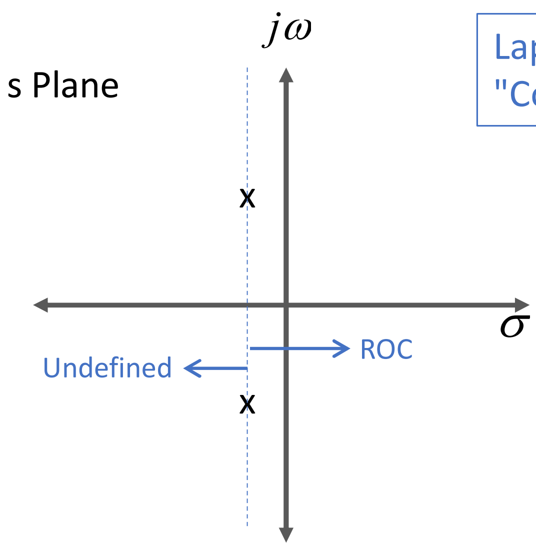


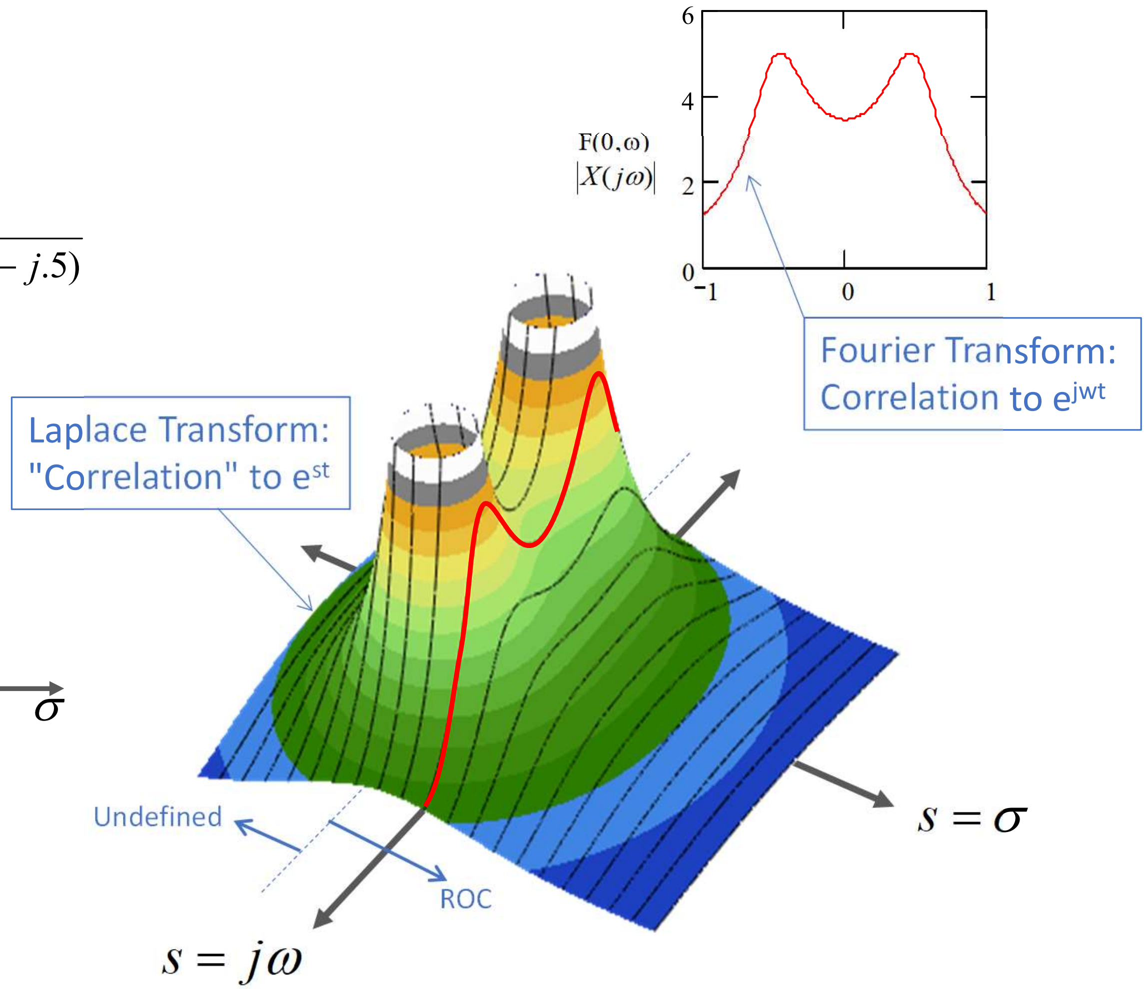


Filter Example

$$X(s)=\frac{1}{(s+.2+j.5)(s+.2-j.5)}$$

(poles at $s=-.2\pm j.5$)





Fourier and Laplace – Discrete Time

$X(\omega) \equiv \int_{t=-\infty}^{\infty} x(t)e^{-j\omega t}dt$

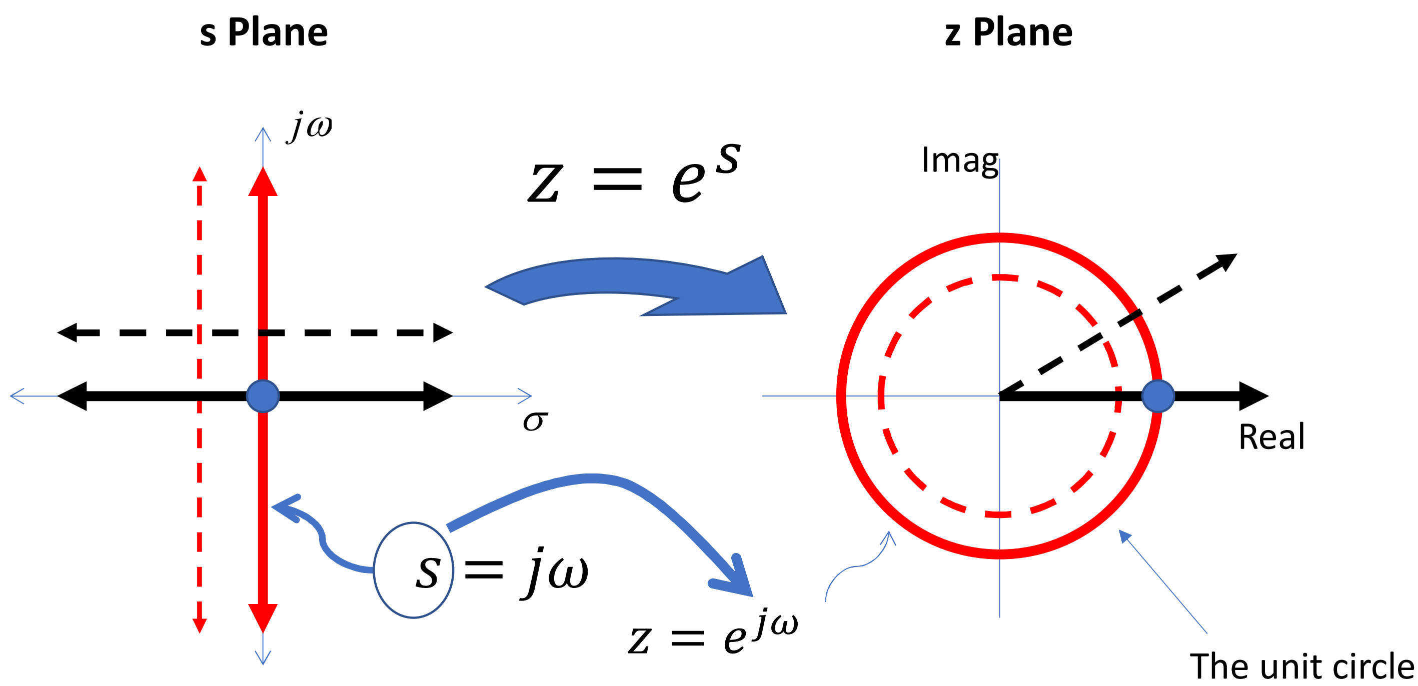
$$X(\omega) \equiv \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

DTFT

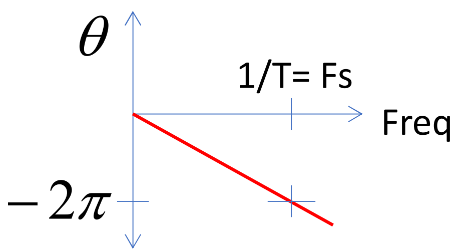
$X(s) \equiv \int_{t=0}^{\infty} x(t)e^{-st}dt$

$$X(s) \equiv \sum_{n=0}^{\infty} x[n]e^{-sn} \rightarrow z = e^s \rightarrow X(z) \equiv \sum_{n=0}^{\infty} x[n]z^{-n}$$

z Transform!

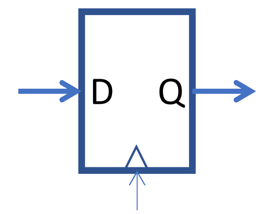

s Plane
z Plane
$j\omega$
σ
$z = e^{s}$
Imag
Real
$s = j\omega$
$z = e^{j\omega}$
The unit circle

z^{-1}



$$e^{sT} = z$$

$$e^{-sT} = z^{-1}$$

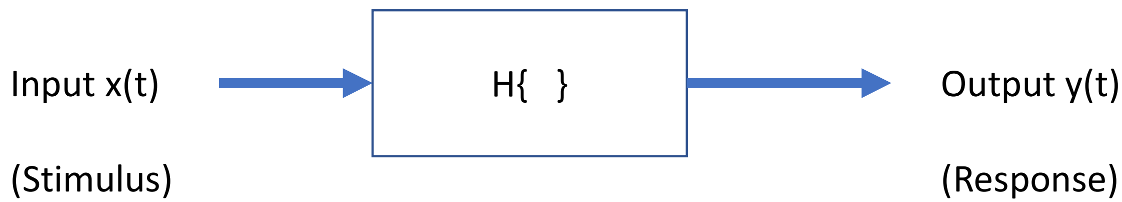


Laplace Transform for a constant delay of time T, x(t-T)

z Transform for a unit sample delay, where Fs = 1/T

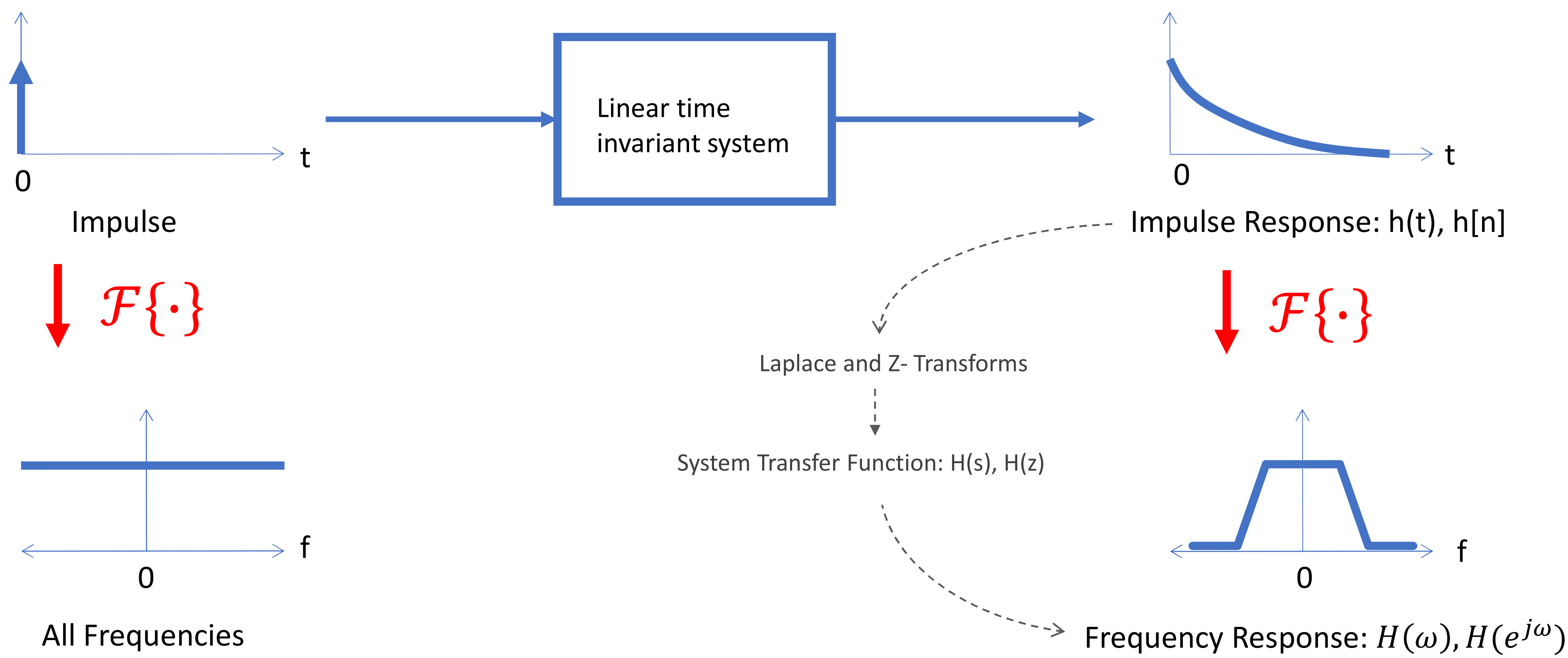
Control Systems

Systems

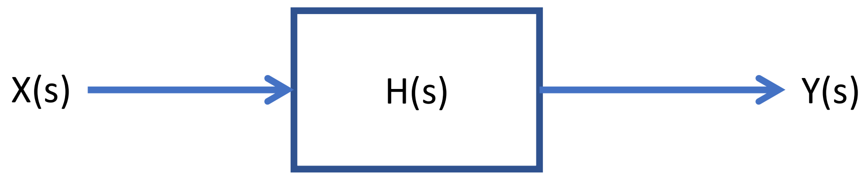


We will be covering the (simpler) world of **linear**, **causal, time-invariant** systems

Impulse Response and Frequency Response

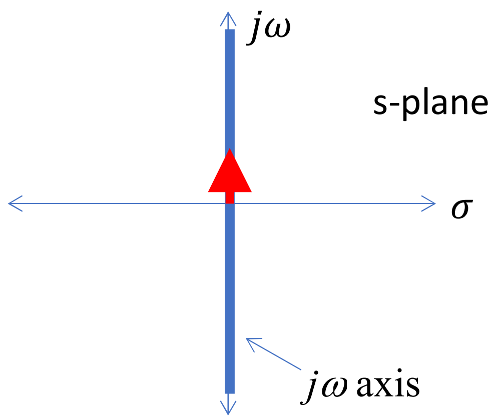


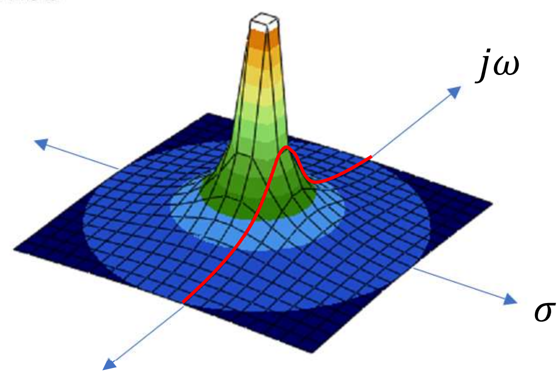
Transfer Functions



Stability: pole locations
Dynamics, impulse response, step response

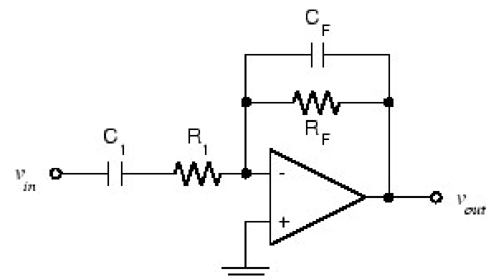
When H(s) is evaluated when s = jw : Frequency Response

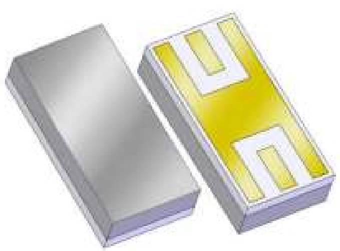




Transfer Functions

Example Bandpass Analog Filters with a Transfer Function given as $H(s)$

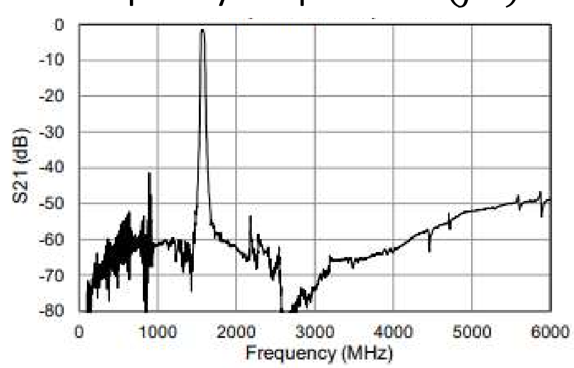




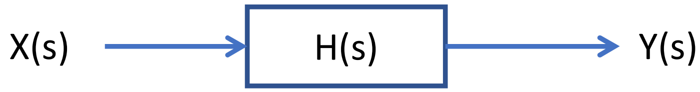
CSP: 3.26 X 1.60 X 0.84 mm

(Qorvo 880273 GPS Bandpass Filter)

Frequency Response: $H(j\omega)$



Loop Transfer Function

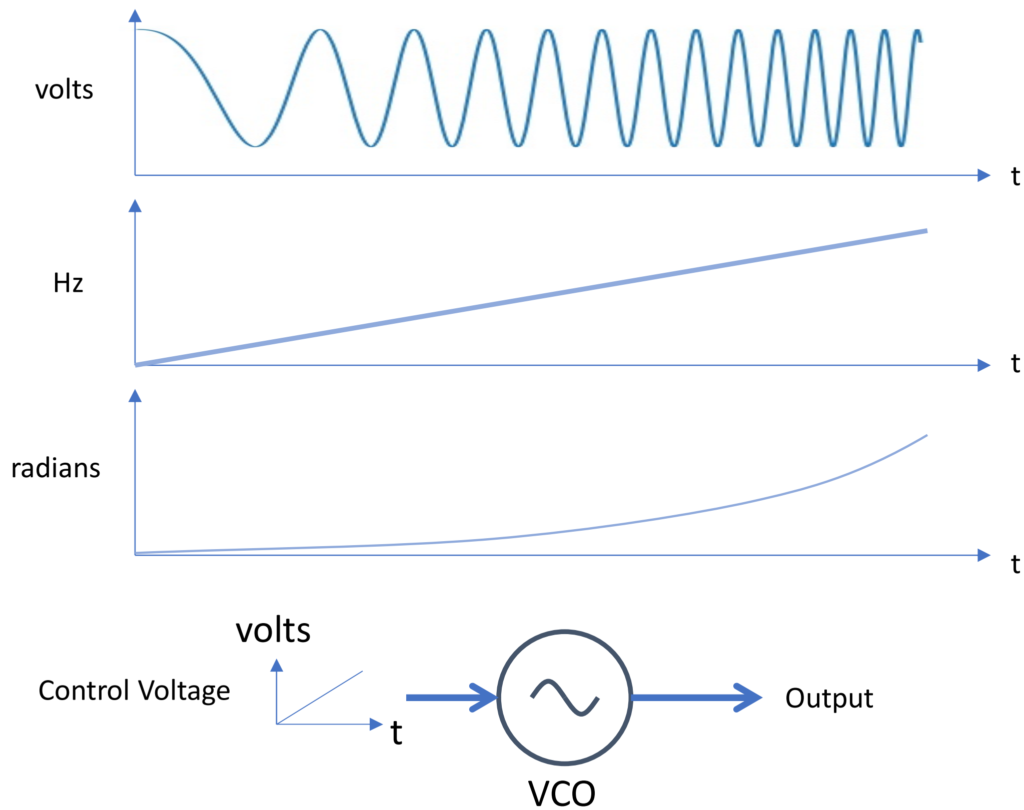


Understand the transfer function of each element involved

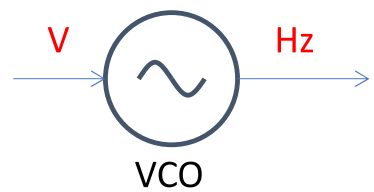
Pay careful attention to units to derive correct transfer function

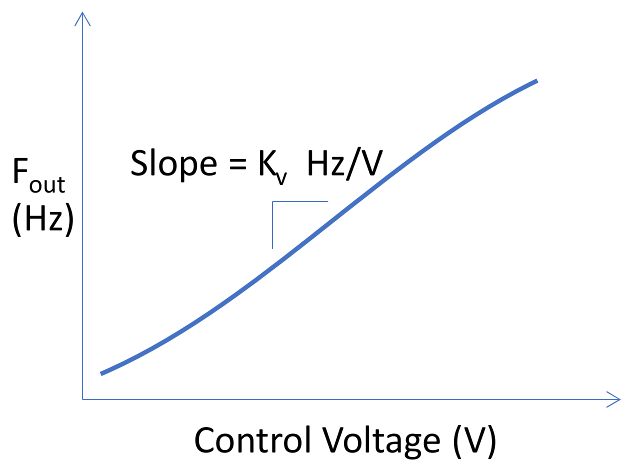
Example: VCO …

VCO Units



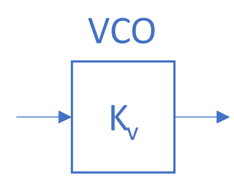
VCO Transfer Function





When frequency is unit of interest,
VCO converts Volts to Hz

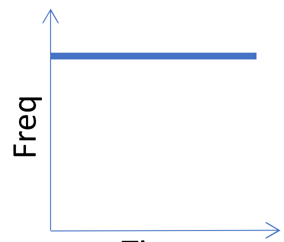
Transfer function = K_v (Hz/V)



Constant "Gain" of Kv, and unit conversion from Volts to Hz

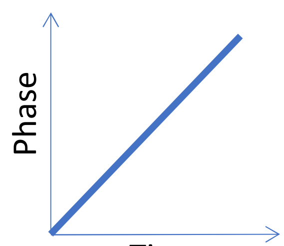
VCO Transfer Function

Freq vs Time



$$\theta(t) = \int_0^t f(t)dt$$

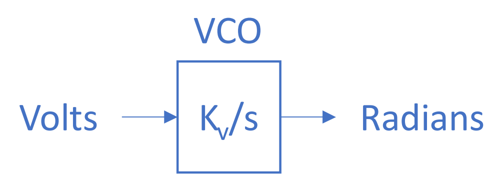
Phase vs Time



$$\theta(t) = \int_0^t f(t)dt \overset{\mathcal{L}}{\Rightarrow} \frac{1}{s}F(s)$$

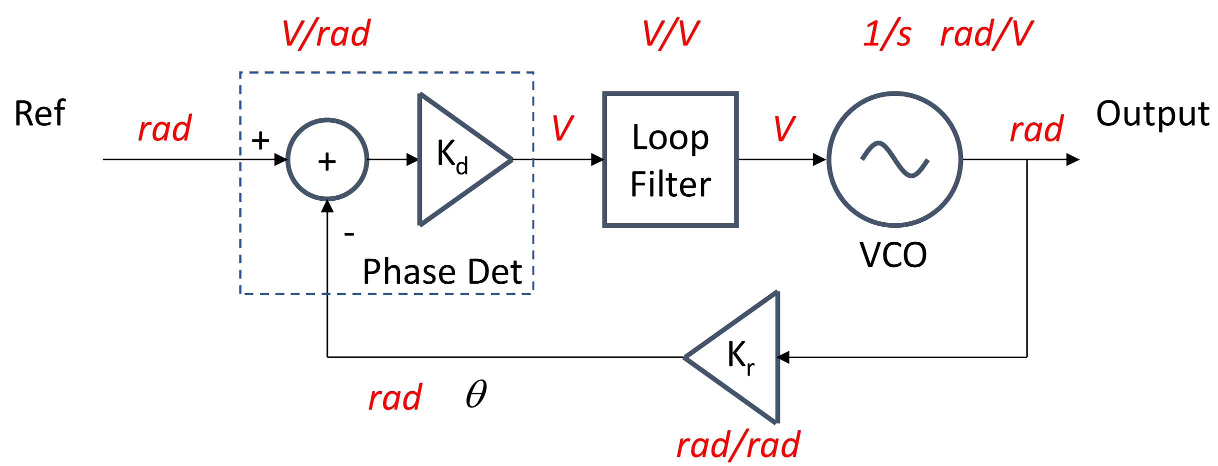
When phase is unit of interest, VCO is a "lowpass" converting Volts to Radians

Transfer function = K_v/s (Radians/V)



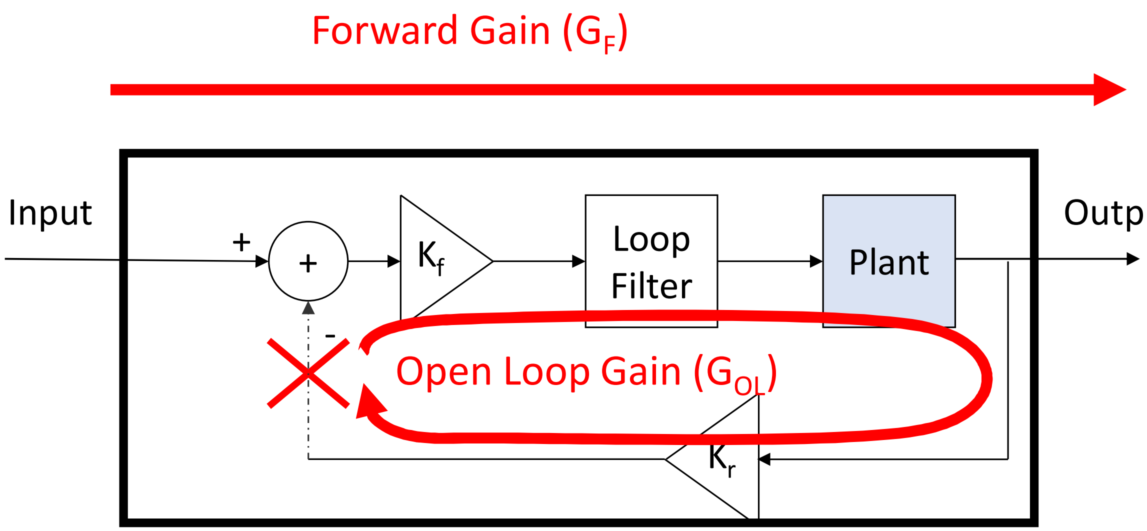
Lowpass filter with gain K_v /s and unit conversion from Volts to Radians

Phase Lock Loop



Keep track of units!

Loop Transfer Function



Closed Loop Gain: $$G_{CL} = \frac{G_F}{1 + G_{OL}}$$

Loop Transfer Function

Stability:

$G_{CL}(s)$ is unstable for any poles in the RHP

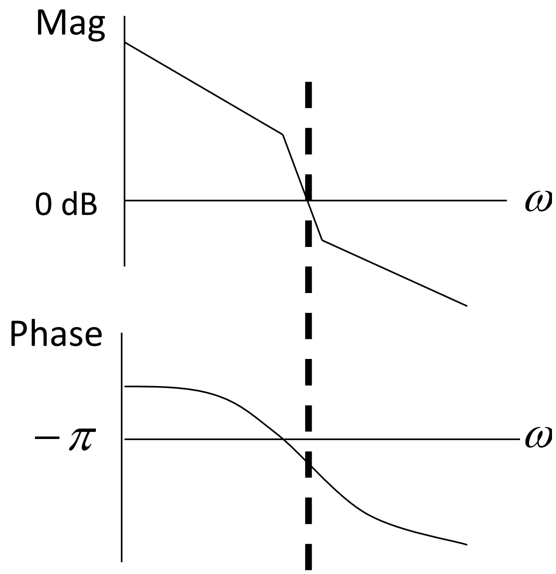
$$G_{CL} = \frac{G_F}{1 + G_{OL}}$$

Characteristic Equation

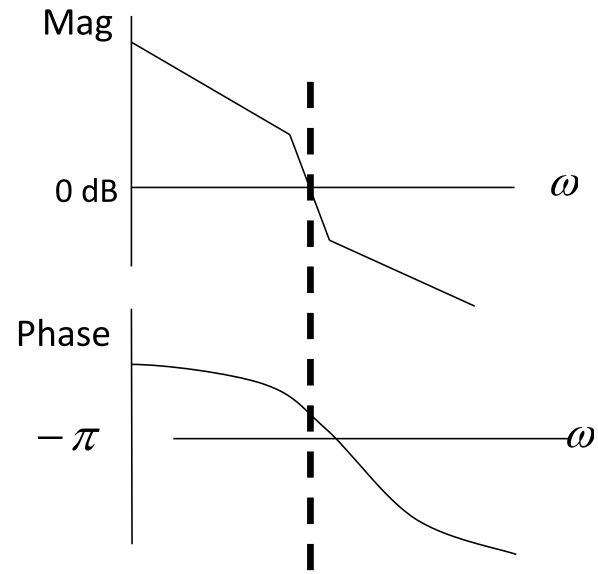
The zeros (roots) of $1+G_{OL}(s)$ are the poles of $G_{CL}(s)$

Stability – Bode Plot

Determined from Open Loop Gain: When $|G_{OL}| \geq 1, \angle G_{OL} < -\pi$

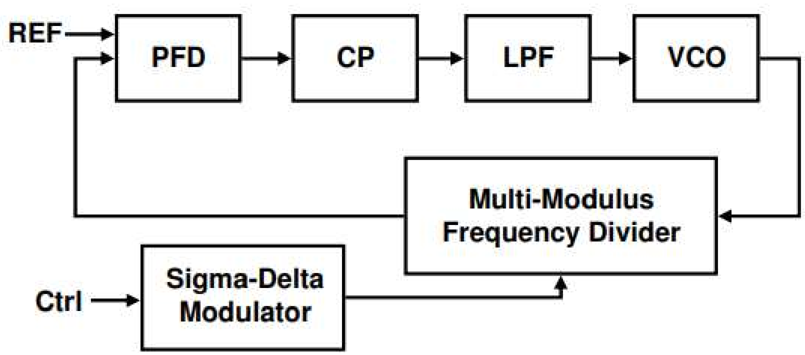


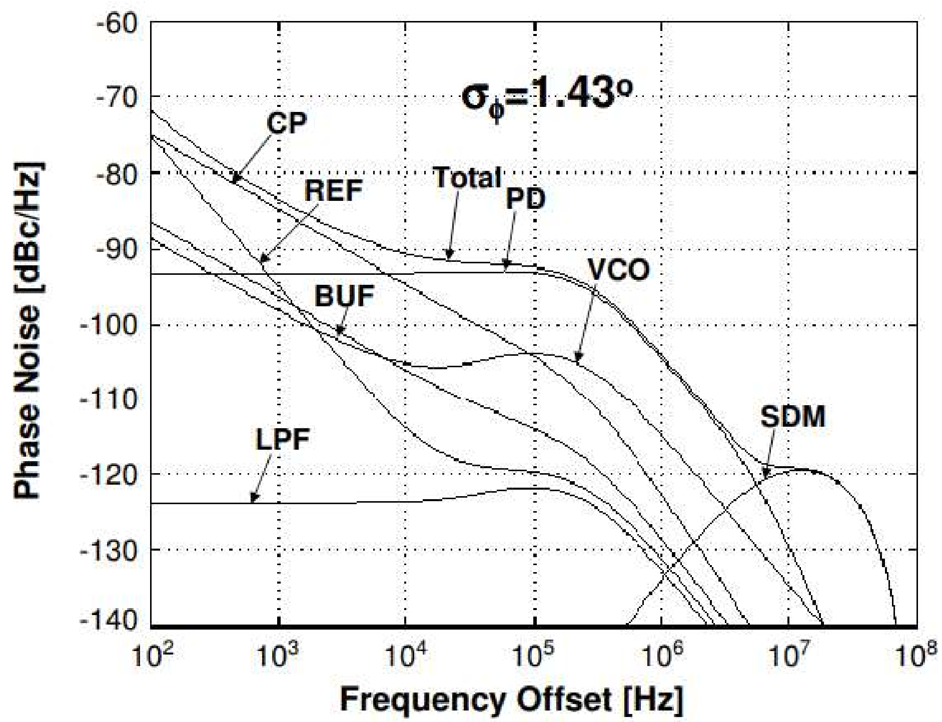
Unstable!



Stable!

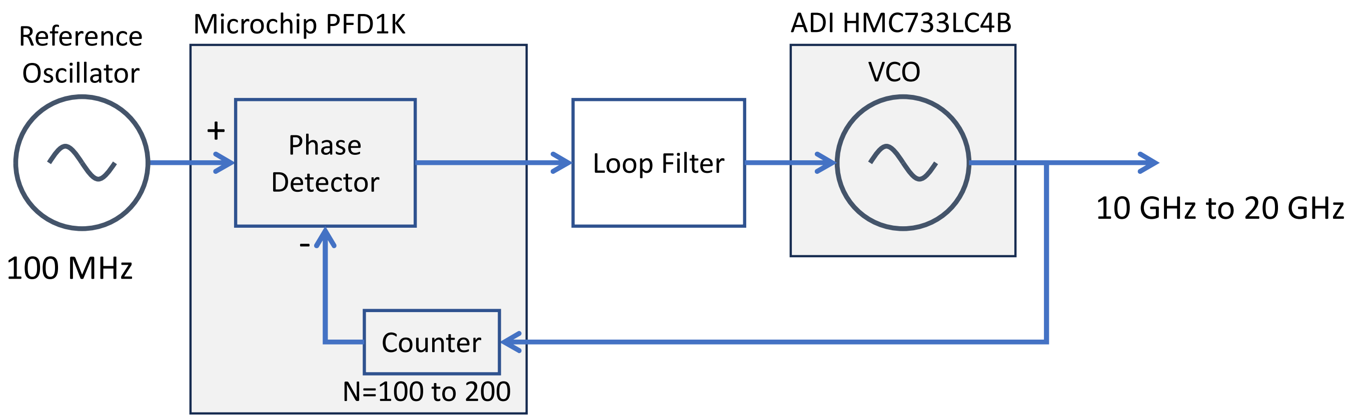
Composite Phase Noise in Frac-N PLL





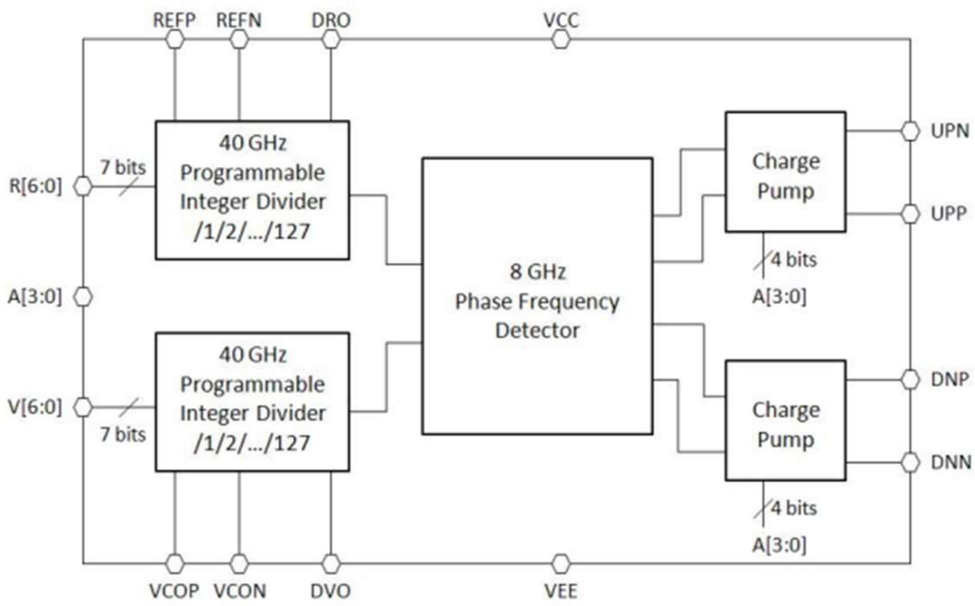
Source: "Analytical Phase Noise Modeling and Charge Pump Optimization for Fractional-N PLLs", Frank Herzel et al, IEEE Transatctions on Circuits and Systems-I: Regular Papers, Vol. 57, No. 8, August 2010.

Example Analog PLL

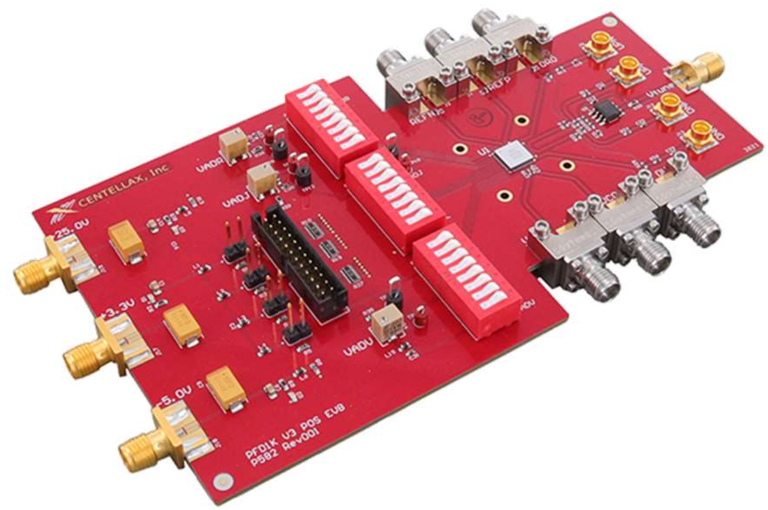


Integer-N Phase Lock Loop

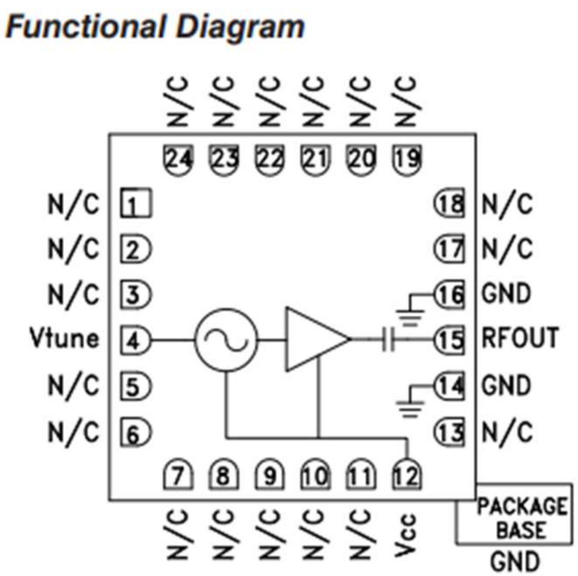
Microchip 8 GHz Phase Freq Detector PFD1K

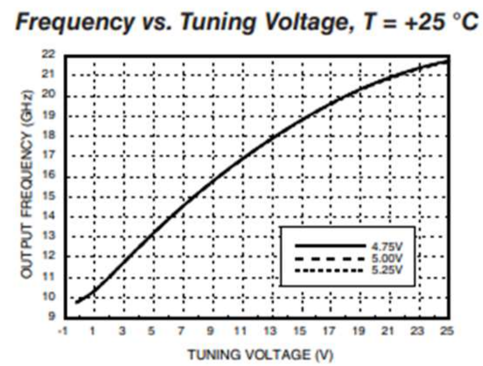


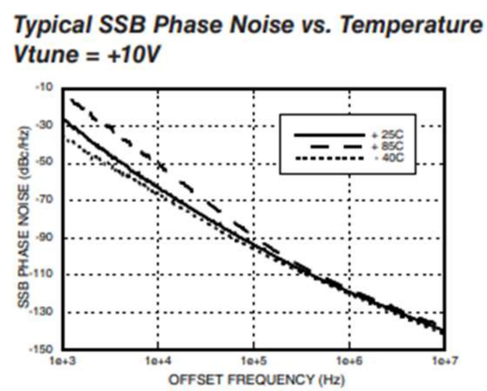
Functional Block Diagram



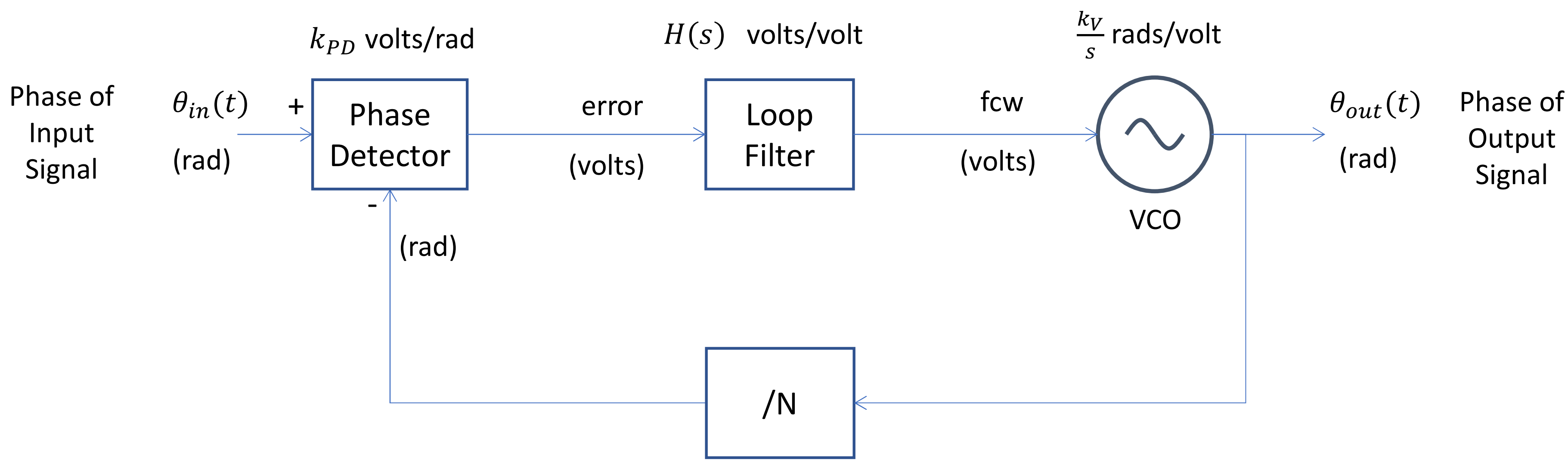
ADI 10 – 20 GHz VCO HMC733LC4B





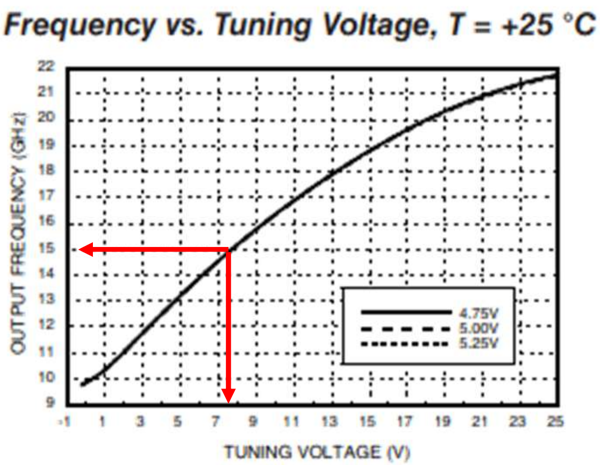


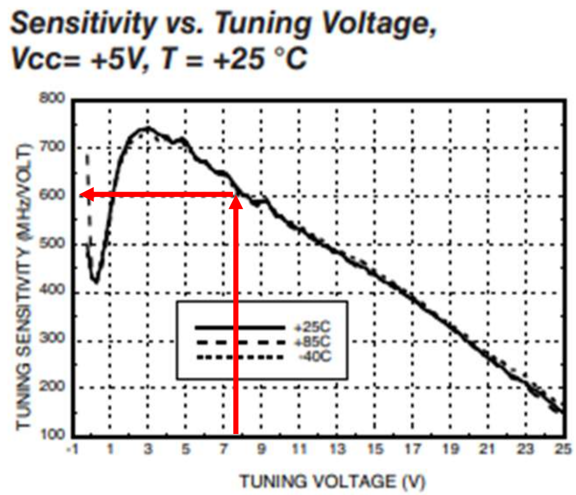
Analog PLL Loop Model



VCO Loop Gain

slope of Freq vs Tuning is → k_V



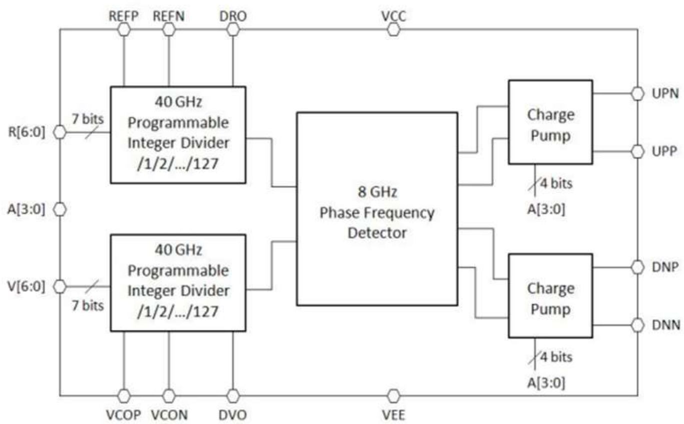


VCO Loop Model is

$$\frac{k_V}{s}$$ rads/volt

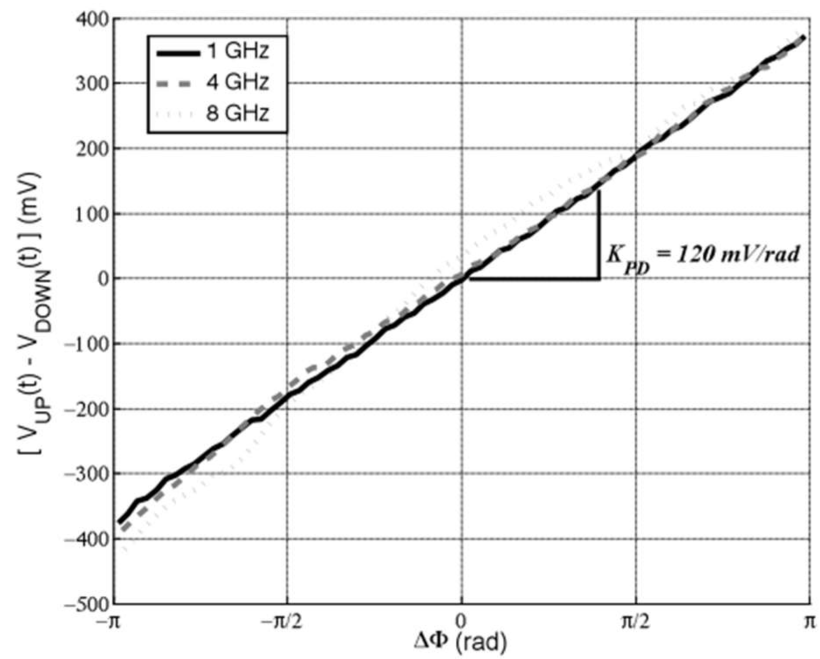
At 15 GHz, slope is 600 MHz/Volt. At this operating point $k_V = 2\pi 600e6\ (rad/\sec)/volt$

Phase Detector Gain

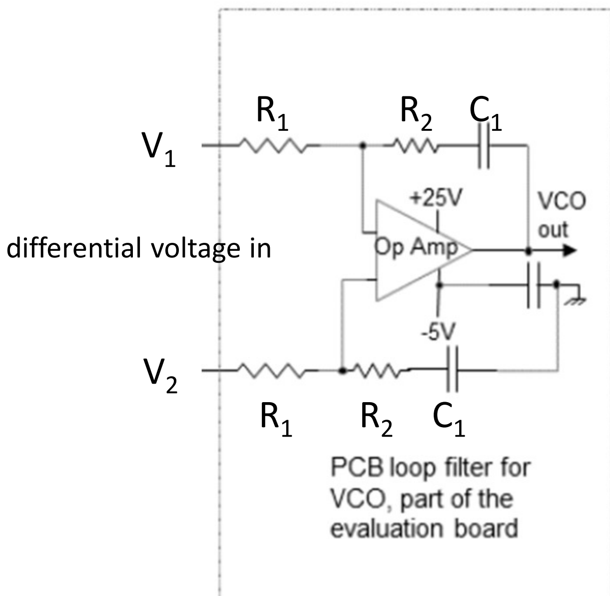


Functional Block Diagram

Figure 1-7. Diff. Output Voltage vs. Frequency (0 dBm Pin)



Differential PI Loop Filter



Differential amplifier w/o C_1 is $V_{out} = \frac{R_2}{R_1}(V_2 - V_1)$

$$V_{out} = \frac{R_2 + \frac{1}{sC_1}}{R_1}(V_2 - V_1)$$

$$\frac{Vout}{V_{in}} = H(s) = \frac{1 + sR_2C_1}{sR_1C_1} = \frac{1 + s\tau_2}{s\tau_1}$$

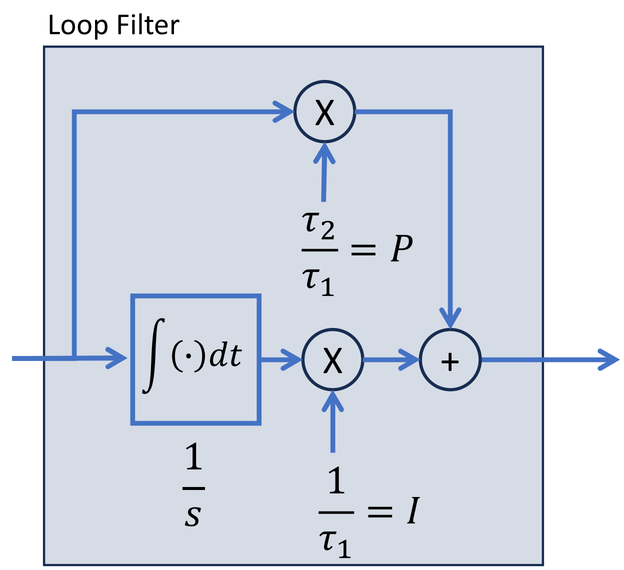
$$H(s) = \frac{1}{s}\frac{1}{\tau_1} + \frac{\tau_2}{\tau_1} = \frac{1}{s}I + P$$

PI Loop Filter

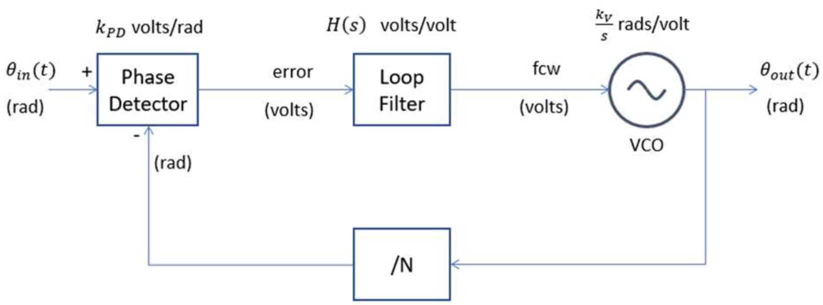
$$H(s) = \frac{1 + s\tau_2}{s\tau_1}$$

$$H(s) = \frac{1}{s}\frac{1}{\tau_1} + \frac{\tau_2}{\tau_1}$$

$$= \frac{1}{s}I + P$$



Open Loop Gain

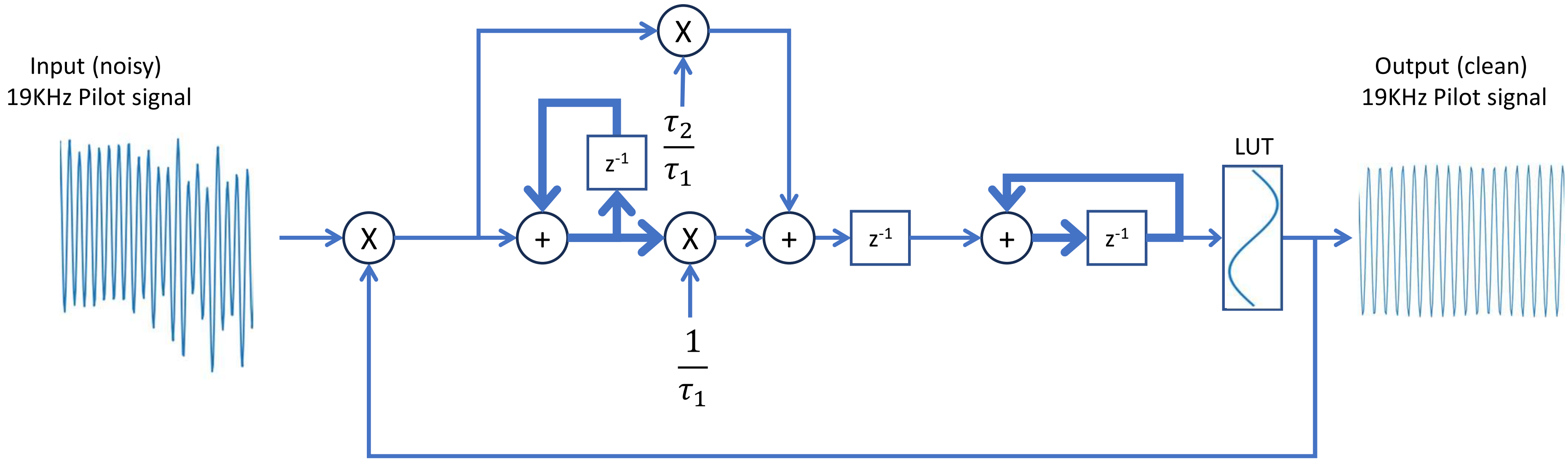


$$G_{OL}(s) = \frac{k_V k_{PD}}{Ns} H(s)$$

$$= \frac{k_V k_{PD}}{Ns} \frac{1 + s\tau_2}{s\tau_1}$$

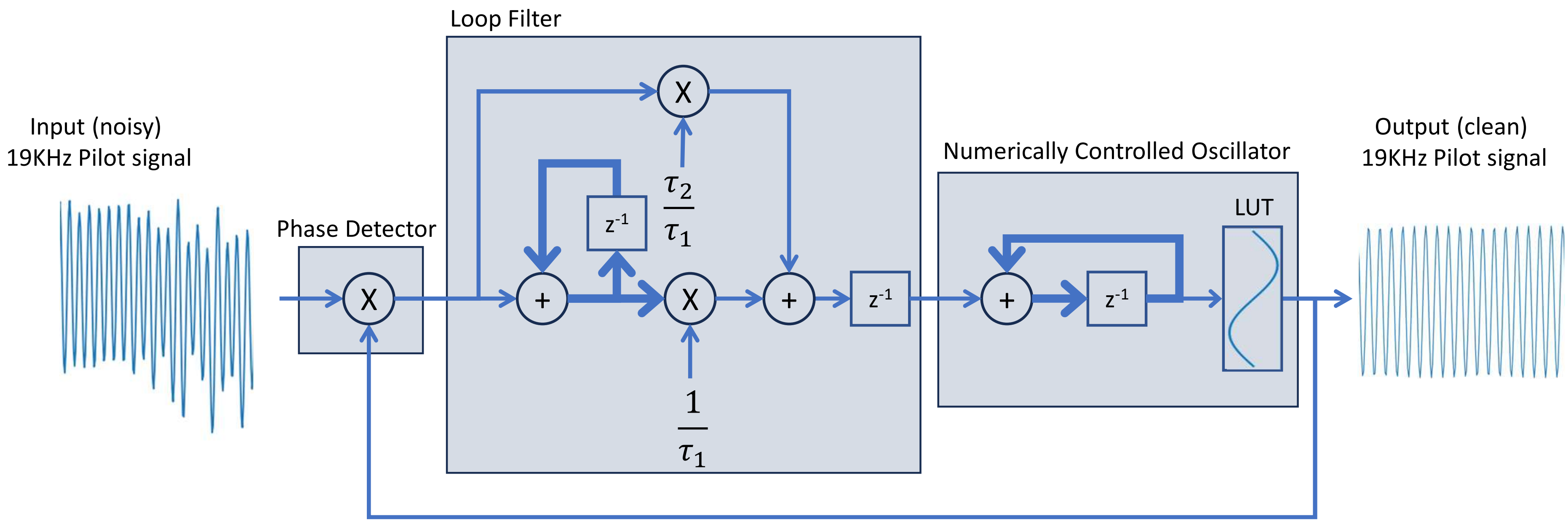
$$= \frac{k_V k_{PD}}{N\tau_1} \frac{1 + s\tau_2}{s^2}$$

Example Digital PLL



All Digital Phase Lock Loop

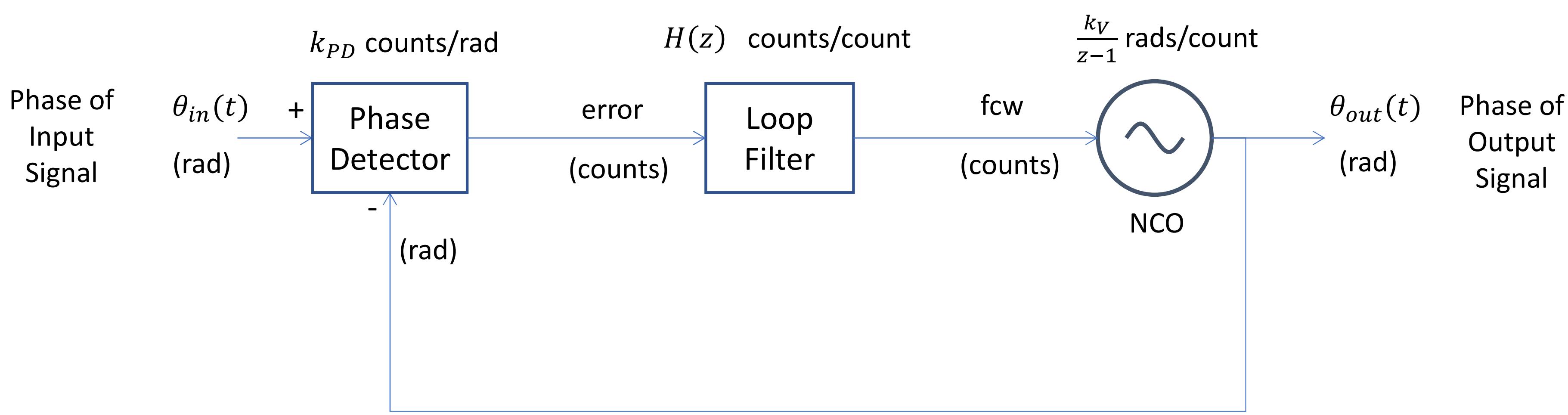
Update Rate 192 KHz



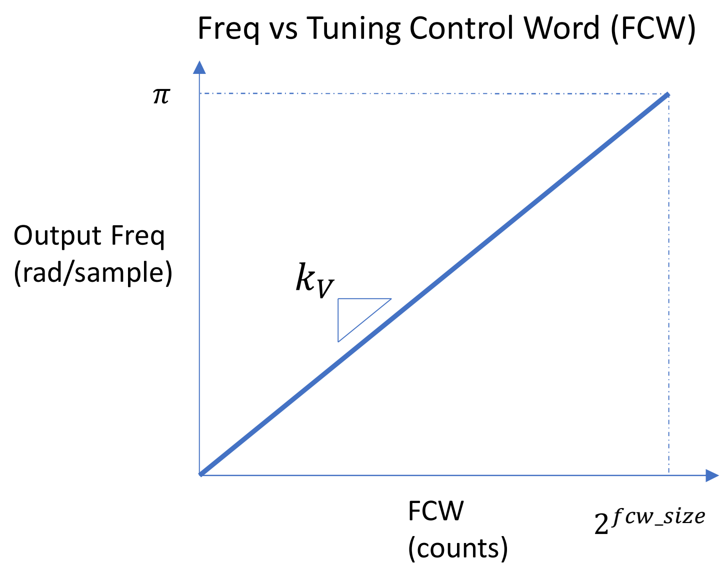
All Digital Phase Lock Loop

Update Rate 192 KHz

Digital PLL Loop Model



NCO Loop Gain

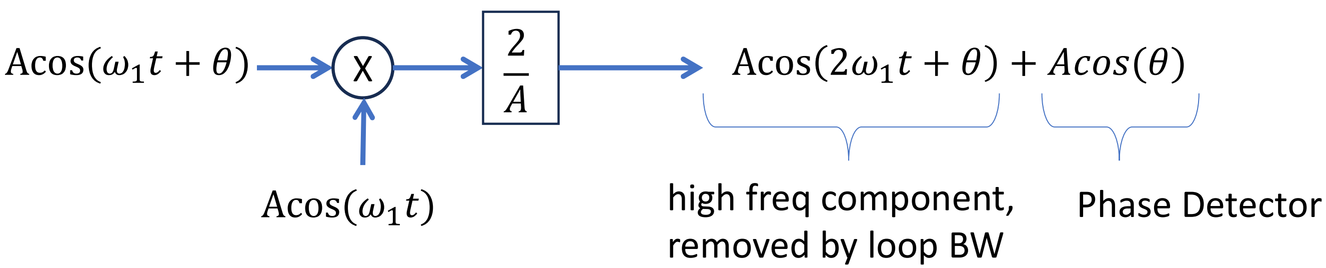


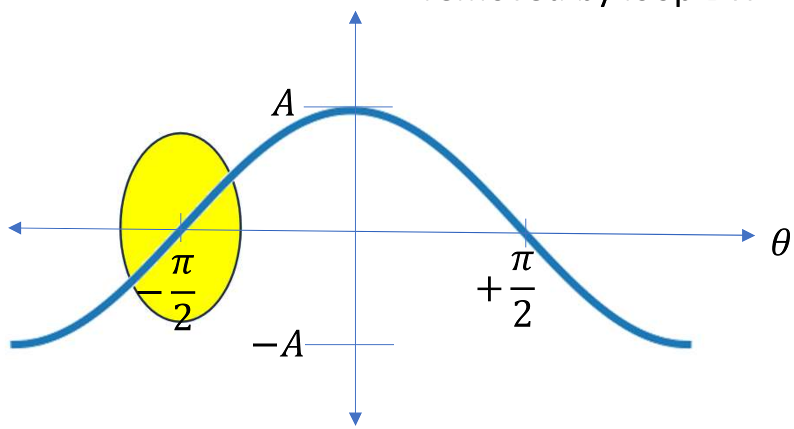
NCO Loop Model is

$$\frac{k_V}{z-1} \text{ rads/count}$$

Where $k_V = \frac{\pi}{2^{fcw\_size}}$

Multiplier as Phase Detector

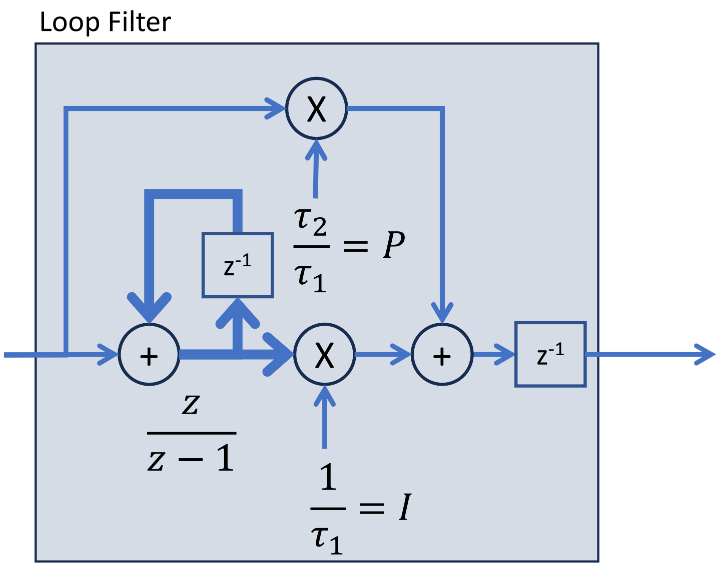




At zero crossing where phase detector is linear with slope:

$$k_{PD} = A \text{ counts/rad}$$

Loop Filter

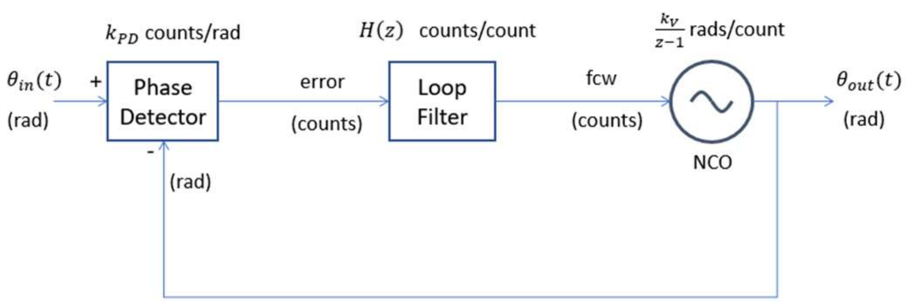


$$H(z) = P + I\frac{z}{z-1}(z^{-1})$$

$$= \frac{\tau_2}{\tau_1} + \frac{1}{\tau_1}\frac{1}{z-1}$$

$$H(z) = \frac{\tau_2 z + (1-\tau_2)}{\tau_1(z-1)}$$

Open Loop Gain



$$G_{OL}(z) = \frac{k_V k_{PD}}{z-1} H(z)$$

$$= \frac{k_V k_{PD}}{z-1} \frac{(\tau_2 z + (1-\tau_2))}{\tau_1 (z-1)}$$

$$= \frac{k_V k_{PD}}{\tau_1} \frac{(\tau_2 z + (1-\tau_2))}{(z-1)^2}$$

Python Control Systems Library

https://python-control.readthedocs.io/en/0.10.1/

Simulation Demonstration

Jupyter Notebook

Want More DSP??

http://ieeeboston.org/courses

Digital Signal Processing for Wireless Communications– First Video Release, October 10, 2024. Live Workshops: Thursdays, October 17, 24, 31, November 7, 14, 2024 – 6:00 – 7:30PM (Eastern Time)

http://dsprelated.com/courses

Python Applications for Digital Design and Signal Processing (America Times)

Attendees will gain an overall appreciation of using Python and quickly get up to speed in best practice use of Python and related tools specific to modeling and simulation for signal processing analysis and design.

Instructor: Dan Boschen

Early Registration Deadline: October 10, 2024

First Class Release Date: October 17, 2024

More Info / Register →

DSP For Wireless Communications (Europe / Asia Times)

Attendees will build a stronger intuitive understanding of the fundamental signal processing concepts involved with digital filtering and mixed signal analog and digital design. With this, attendees will be able to implement more creative and efficient signal processing architectures in both the analog and digital domains. The knowledge gained from this course will have immediate practical value for any work in the signal processing field.

Instructor: Dan Boschen

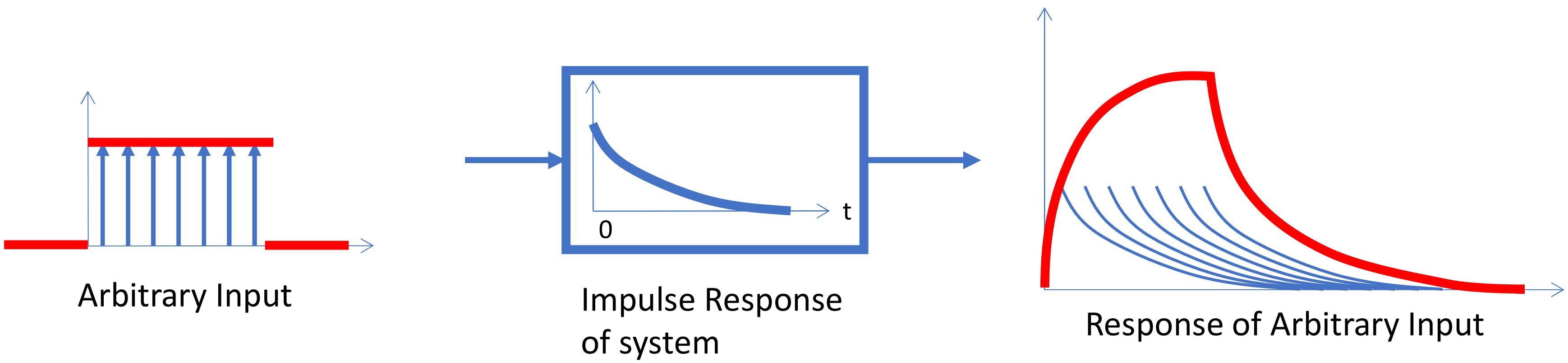
Early Registration Deadline: February 13, 2025

First Class Release Date: February 20, 2025

More Info / Register →

Backup Slides

Laplace Transform

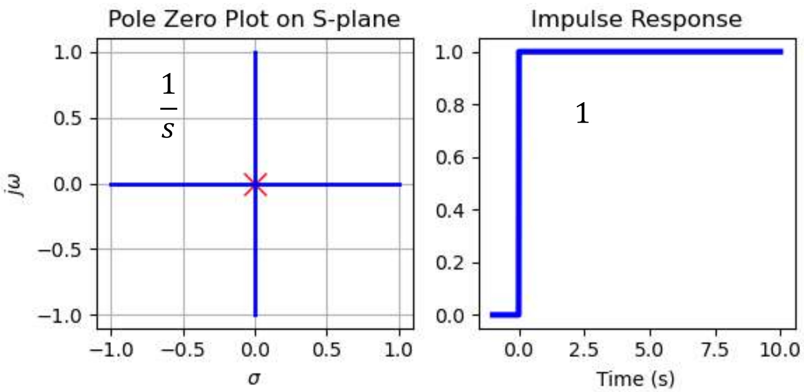


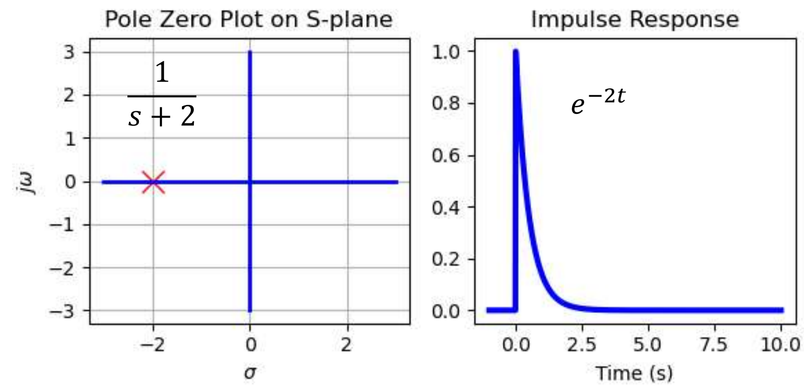
MOTIVATION FOR LAPLACE

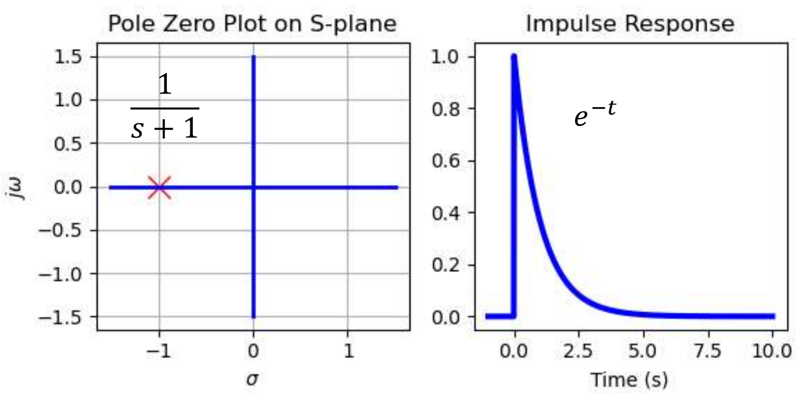
- Convolution in the Time Domain is a product in the Frequency and Laplace Domains.
- In many cases it is easier for us to do a product rather than convolution.
- Can determine both steady state and transient responses

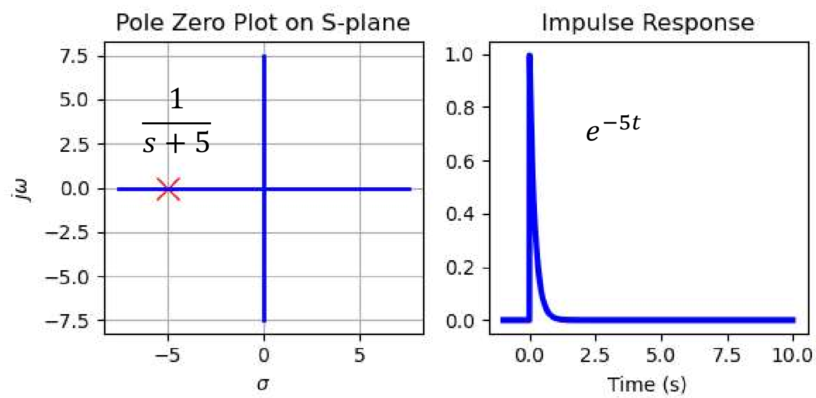
- The Laplace Transform of the system's impulse response provides useful behaviour insight.
- The Laplace Transform coverts integro-differential equations to simple algebra.

Laplace Transforms – First Order, All Pole

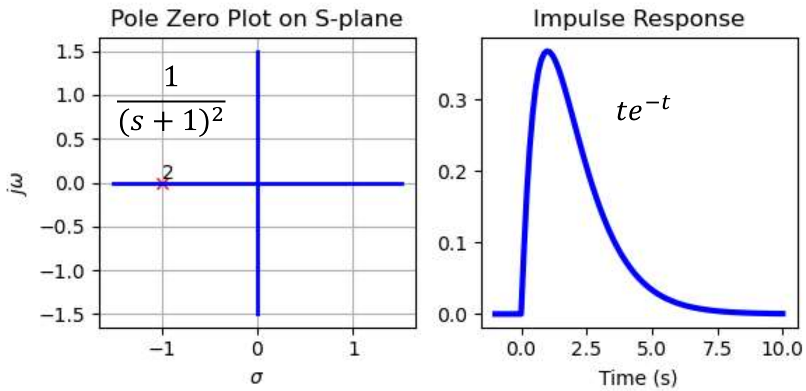


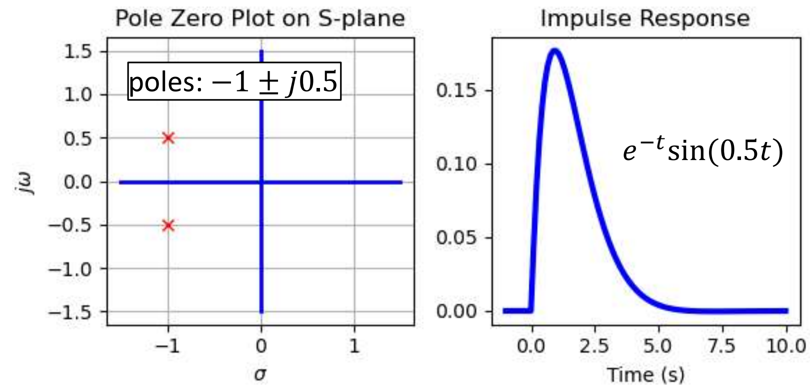


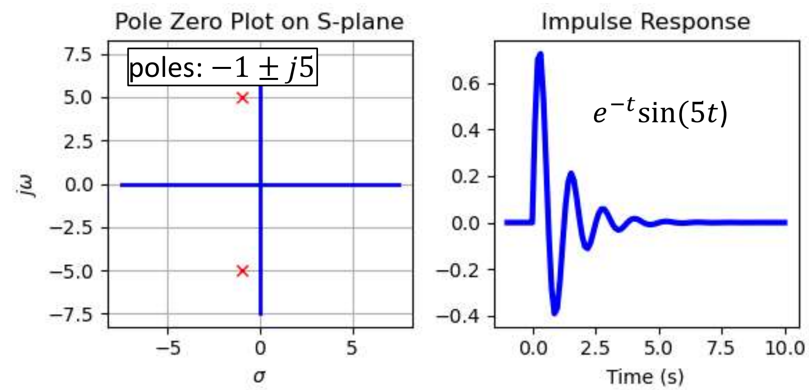




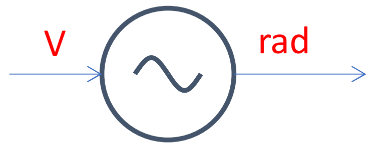
Example Laplace Transforms- 2nd Order, All Pole





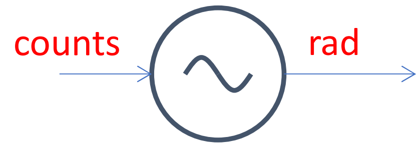


VCO

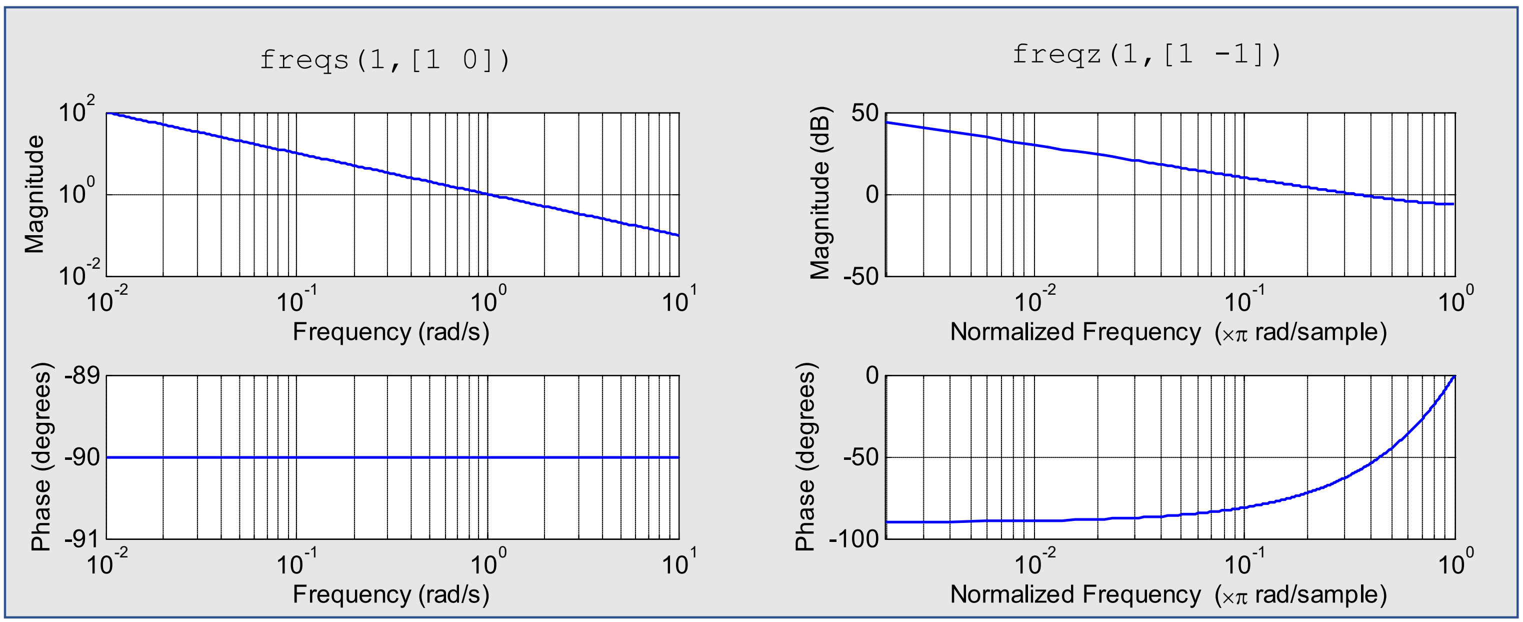


$$\frac{K_v}{s}$$

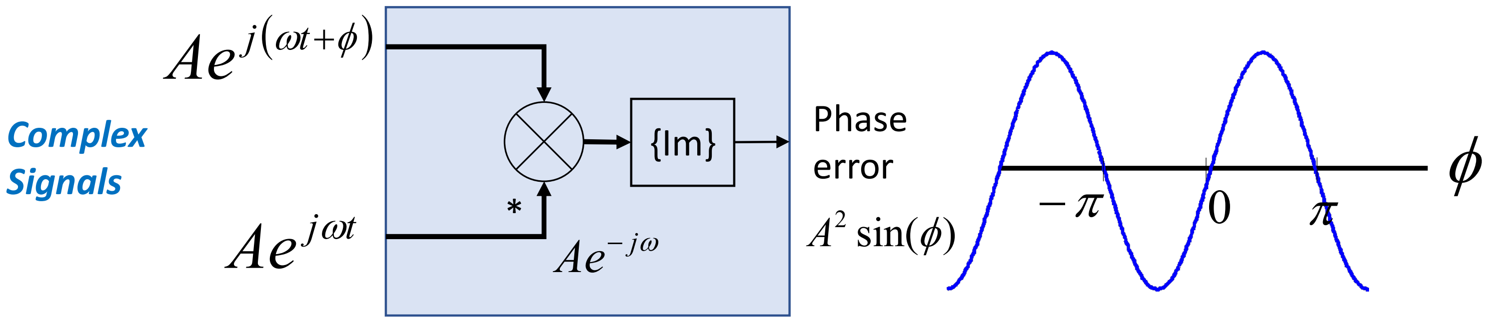
NCO



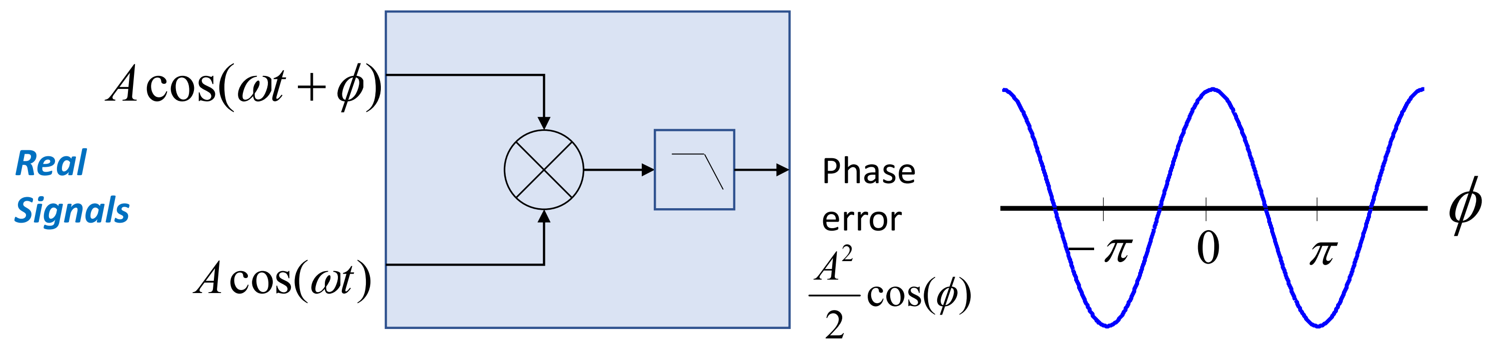
$$\frac{K_v}{1 - z^{-1}}$$



Phase Detectors



Or take Real output to get $\cos(\phi)$



2nd Order System with PI Loop Filter

From classical continuous-time 2nd order loop equations with a PI loop filter:

τ_1 and τ_2 time constants determined from natural frequency and damping ratio:

ω_n : natural frequency rad/s

ζ : damping ratio

$$\tau_1 = \frac{K_v K_d}{\omega_n^2}$$

$$\tau_2 = \frac{2\zeta}{\omega_n}$$

And ω_n determined from desired loop BW, ω_{3dB}:

$$\omega_n = \frac{\omega_{3dB}}{\sqrt{\alpha + \sqrt{\alpha^2 + 1}}} \qquad \alpha = 1 - 2\zeta^2$$

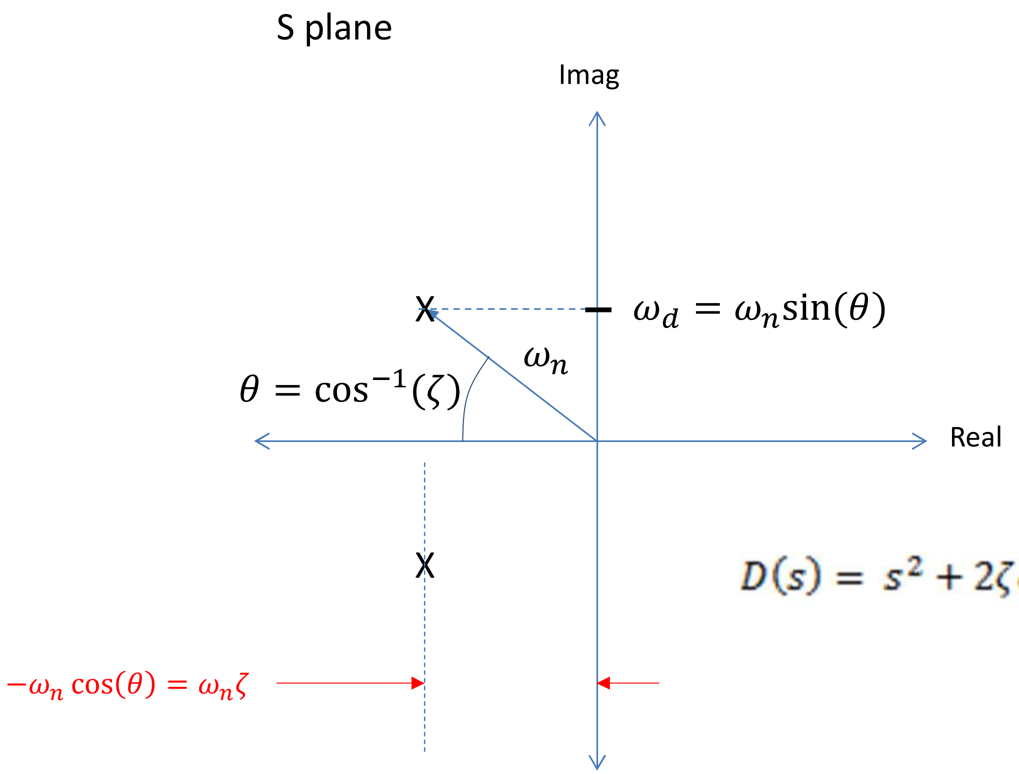
Applies when

$$G_{OL}(s) = \frac{K_v K_d (1 + s\tau_2)}{s^2 \tau_1}$$

From Closed Loop Canonical Form:

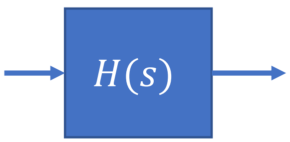
$$H_{CL}(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Ref: Floyd M. Gardner, *Phaselock Techniques*, John Wiley and Sons, 1979

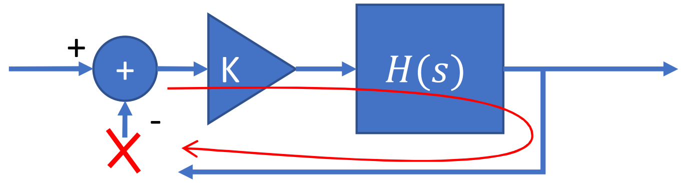


ω_n: natural frequency
ω_d: damped frequency
ζ: damping factor
D(s): denominator of transfer function for closed loop 2nd order system

$$D(s) = s^2 + 2\zeta\omega_n s + \omega_n^2$$

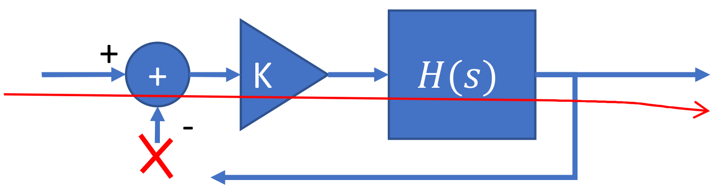


Open Loop Transfer Function



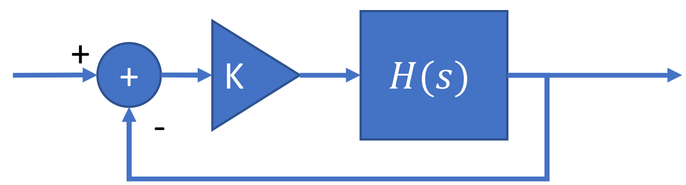
$$\boldsymbol{G_{OL}(s)} = KH(s)$$

Forward Path Transfer Function



$$\boldsymbol{G_F(s)} = KH(s)$$

Closed Loop Transfer Function



$$\boldsymbol{G_{CL}(s)} = \frac{G_F(s)}{1 + G_{OL}(s)} = \frac{KH(s)}{1 + KH(s)}$$

Loop Order and Type

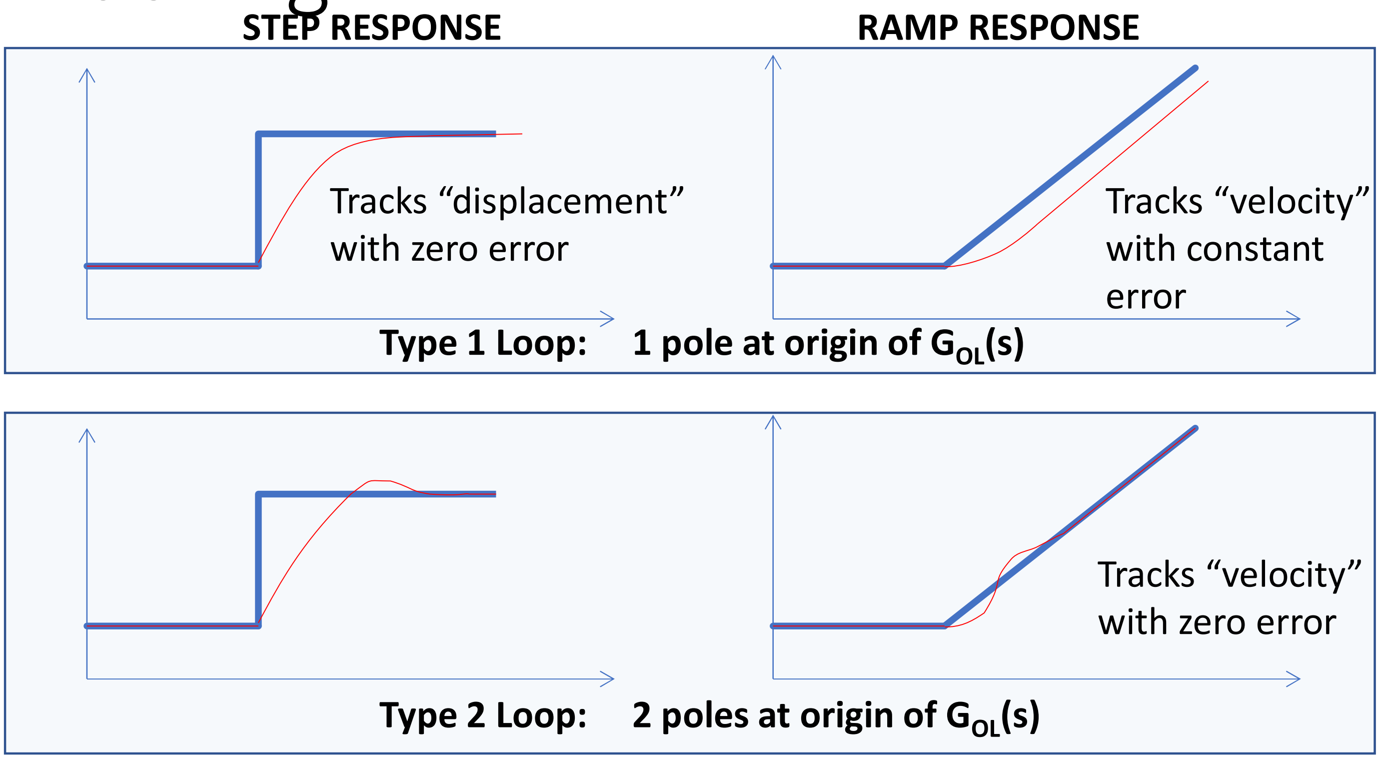
$$G_{CL}(s) = \frac{G_F(s)}{1 + G_{OL}(s)}$$

Loop Order: The degree of the characteristic equation

Loop Type: The number of poles at the origin in $G_{OL}(s)$

Example: 2^{nd} Order Type 1 Loop: $G_{OL}(s) = \frac{s+3}{s(s+2)}$

Loop Tracking



(Need a Type 3 Loop to track “acceleration”)

Loop Tracking

Final Value vs Input (Error) and Loop Type

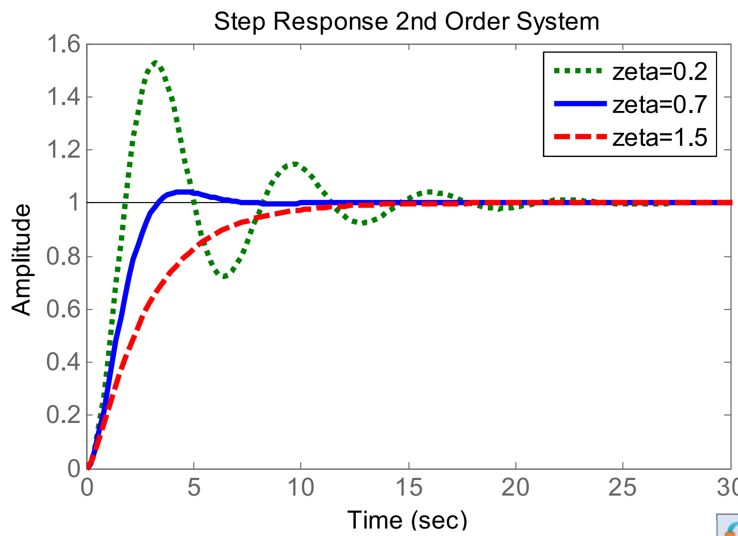
Input	Type 0	Type 1	Type 2	Type 3
Step (Constant)	Constant	0	0	0
Velocity (Ramp)	Ramp	Constant	0	0
Acceleration (x^2)	x^2	Ramp	Constant	0

2nd Order System

Canonical Form: $$H_{CL}(s) = \frac{\omega_n^{\;2}}{s^2 + 2\zeta\omega_n s + \omega_n^{\;2}}$$

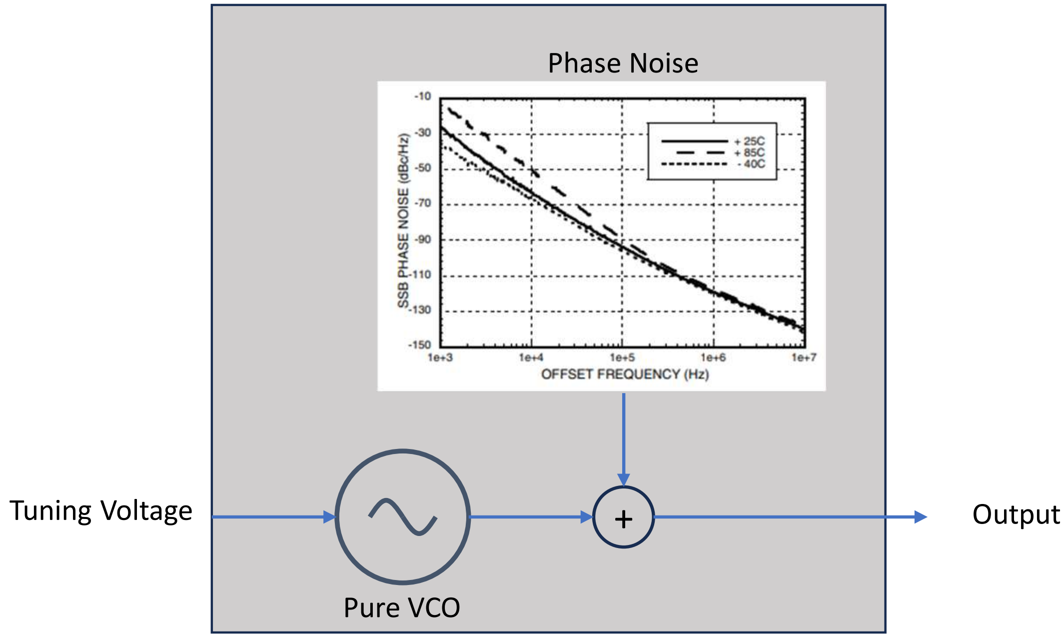
ω_n : natural frequency

ζ : damping ratio

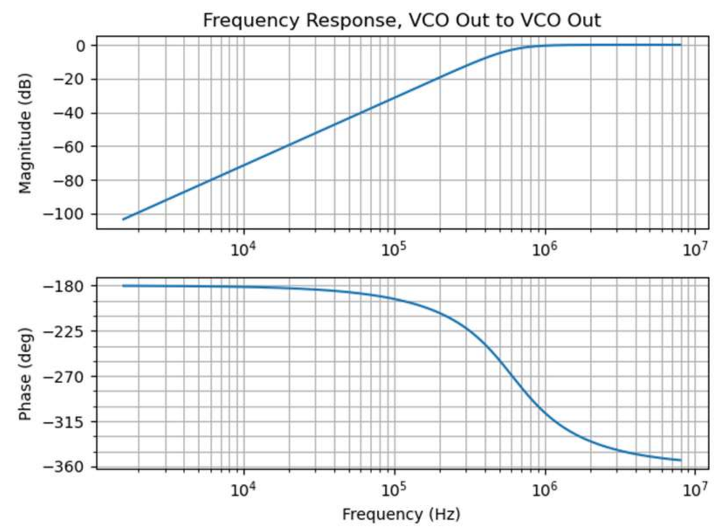



```
sys=tf(1,[1 2*.707 1])
step(sys)
```

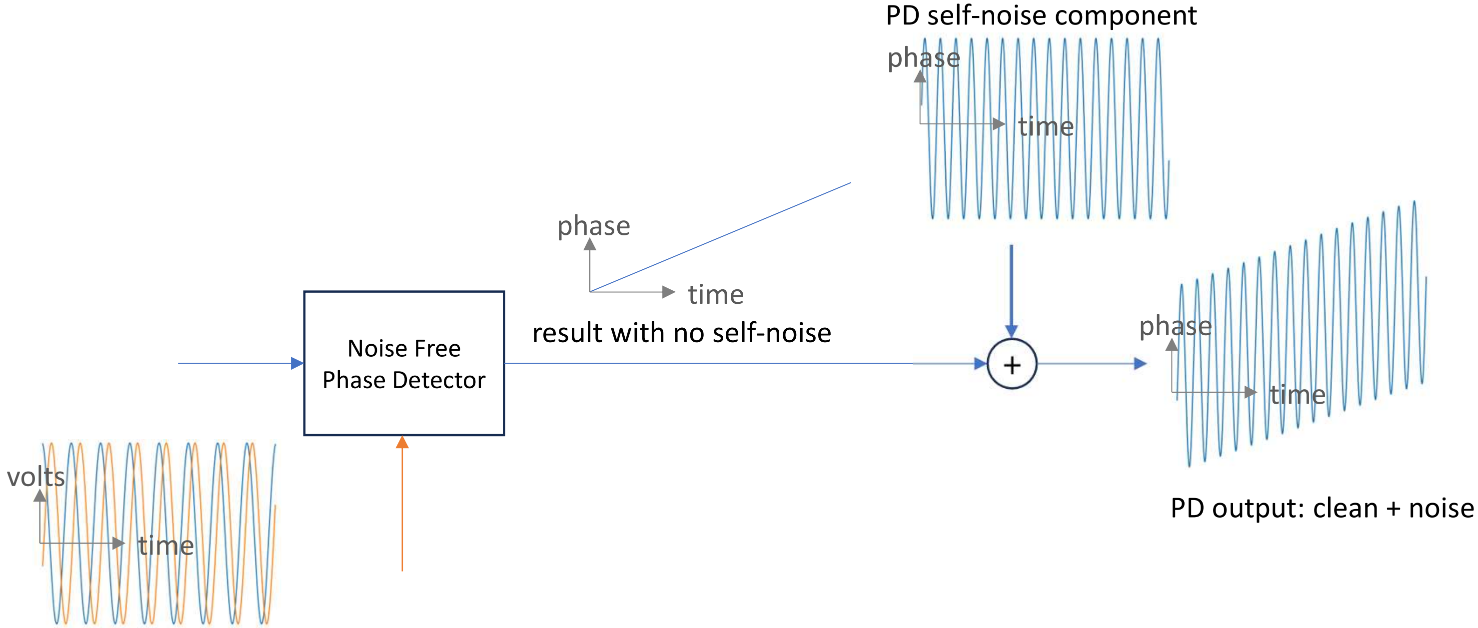
VCO Phase Noise Frequency Domain Model



Loop Transfer Function from Phase Noise Injection Node to Output



Noisy Phase Detector is modelled in
the loop as a noise-free phase detector
with an external noise input



A MOTIVATION FOR USING USING $e^{j\phi}$: SSB Modulator

Consider a single tone at freq ω_1 $\quad s_1(t) = e^{j\omega_1 t}$

We can shift the frequency by $\omega_\Delta$by multiplying (mixing) the signal with another single tone at freq ω_Δ, with no other filtering required after the product.

$$s_\Delta(t) = e^{-j\omega_\Delta t}$$

$$s_2(t) = s_1(t)s_\Delta(t) = e^{j\omega_1 t}e^{-j\omega_\Delta t} = e^{j(\omega_1 - \omega_\Delta)t}$$

Done!

Same processing but we will represent the signals with Sines and Cosines:

$$s_1(t) = \cos(\omega_1 t) + \mathrm{j}\sin(\omega_1 t)$$

$$s_\Delta(t) = \cos(\omega_\Delta t) - \mathrm{j}\sin(\omega_\Delta t)$$

$$s_2(t) = s_1(t)s_\Delta(t) = (\cos(\omega_1 t) + \mathrm{j}\sin(\omega_1 t))(\cos(\omega_\Delta t) - \mathrm{j}\sin(\omega_\Delta t))$$

$$= \cos(\omega_1 t)\cos(\omega_\Delta t) + \sin(\omega_1 t)\sin(\omega_\Delta t) + j(\sin(\omega_1 t)\cos(\omega_\Delta t) - \cos(\omega_1 t)\sin(\omega_1 t))$$

Ugh! Need I continue?

Motivation for using $e^{j\phi}$

De Moivre's Theorem:

$$(\cos(\theta) + j\sin(\theta))^n = \cos(n\theta) + j\sin(n\theta)$$

Same as:

$$\left(e^{j\theta}\right)^n = e^{jn\theta}$$

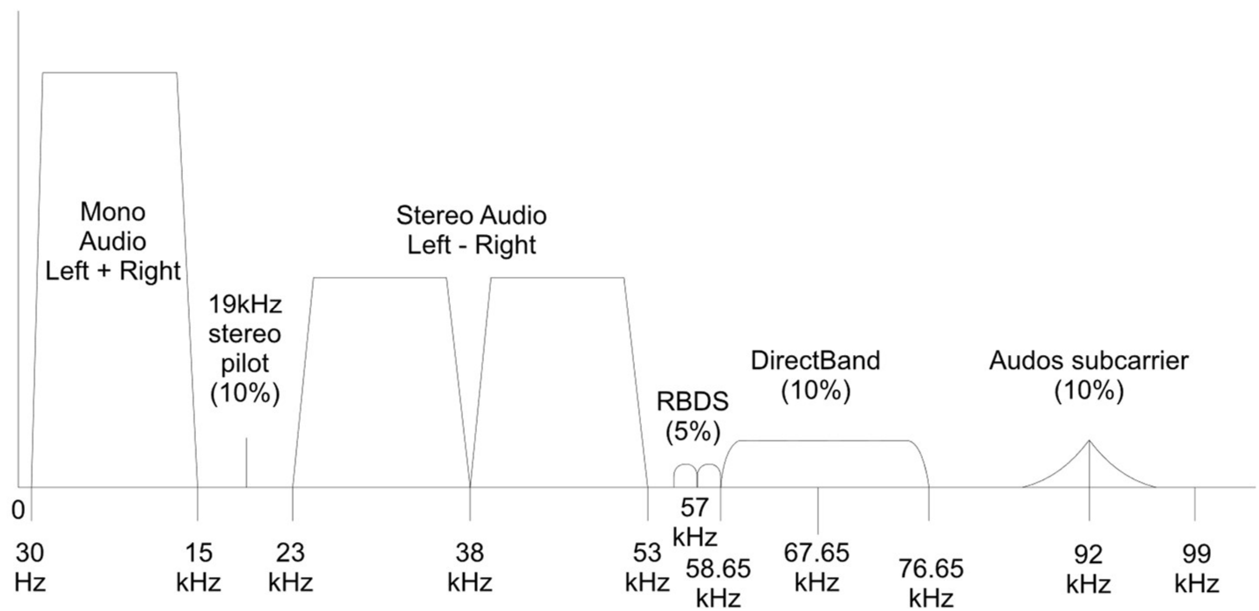
https://xkcd.com/912/

Control Systems

Quick Start with Python

Dan Boschen September 2024

FM Broadcast



By Arthur Murray - http://en.wikipedia.org/wiki/File:RDS_vs_DirectBand_FM-spectrum2.png, Public Domain, https://commons.wikimedia.org/w/index.php?curid=5594246

Control Cheat Sheet

import control as con

sys = con.tf([num], [den])	transfer function of s: example sys=con.tf(3, [1 2]) is 3/(s+2)
sys = con.tf([num], [den], ts)	transfer function of z, ts is sampling time interval
con.mineral(sys)	reduce transfer (good practice to always do this!)
con.bode(sys)	Bode plot (used with **open loop** tf to assess stability)
con.nyquist(sys)	Nyquist plot (used with **open loop** tf to assess stability)
con.rloclus(sys)	Root Locus plot (used with **open loop** tf to view closed loop poles vs gain)
con.pzmap(sys)	Map of poles and zeros
con.step(sys)	Step Response (used with **closed loop** tf)
con.impulse(sys)	Impulse Response (used with **closed loop** tf)
[num, den] = con.zp2tf(z, p, k)	zero-pole-gain to transfer function
[z, p, k] = con.tf2zp([num], [den])	transfer function to zero-pole-gain
con.sensitivity	sensitivity margin M for gain and phase margin for open loop TF shortest distance on Nyquist plot to -1 is 1/M reasonable values for M: 1.3 to 2
gcl = con.feedback(sys1,sys2)	closed loop transfer function from forward tf and backward (feedback) loop tf