Lecture 19

Phase Locking & Timing/Carrier Recovery

March 14, 2024

JOHN M. CIOFFI

Hitachi Professor Emeritus (recalled) of Engineering

Instructor EE379A – Winter 2024

Announcements & Agenda

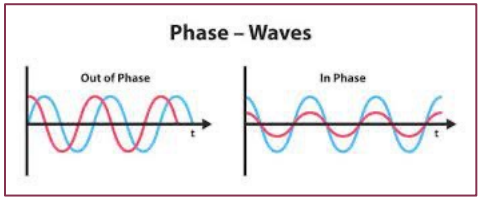
- Announcements
  - We distribute the final at end of lecture today.
  - Due tomorrow at 5:30pm – if not special arrangement, send via email to Ethan by then.
    - You can use your own favorite paper as easier to scan.

- Today
  - Phase Error Detection and VCO's, slide 5
  - Phase locking
    - continuous
    - Discrete-time
  - PLL Analysis
  - Symbol Timing
  - Carrier Recovery

- Feedback PS8
  - 6-10 hours
  - FIR
    - W combines MF and FF filter (MS-WMF)
    - B is best one-sided channel length
  - D-transform (set $D = Z^{-1}$) ; D=delay
    - Water-filling & Diversity (thank you for trying)

Check the Clocks!

- What's wrong ???
  - The modulator checks.
  - The detector checks.
  - The channel is easy.
  - But the results are awful.

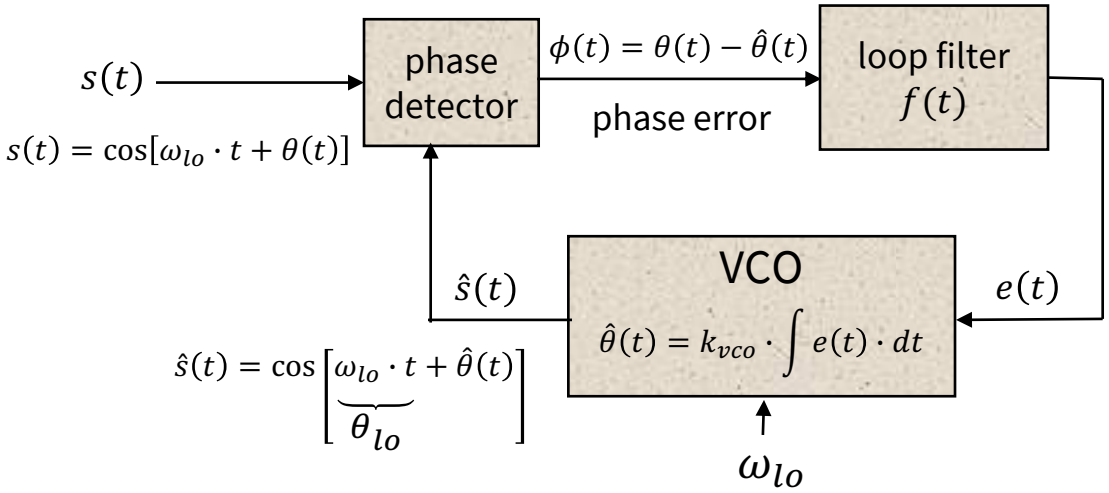


- Check the clocks!
  - Symbol/subsymbol clock
  - Sampling clock $\frac{1}{T'} = \frac{p}{q \cdot T}$ exactly.
  - Carrier frequency
    - Small percentage error on high frequency can be huge w.r.t. the baseband signal's bandwidth.

Phase Detection & Voltage-Controlled Oscillators

Section 6.1

The Phase Lock Loop



$$\theta(t) = cos^{-1}(s(t) - \theta_{lo})$$

$$\hat{\theta}(t) = cos^{-1}(\hat{s}(t) - \theta_{lo})$$

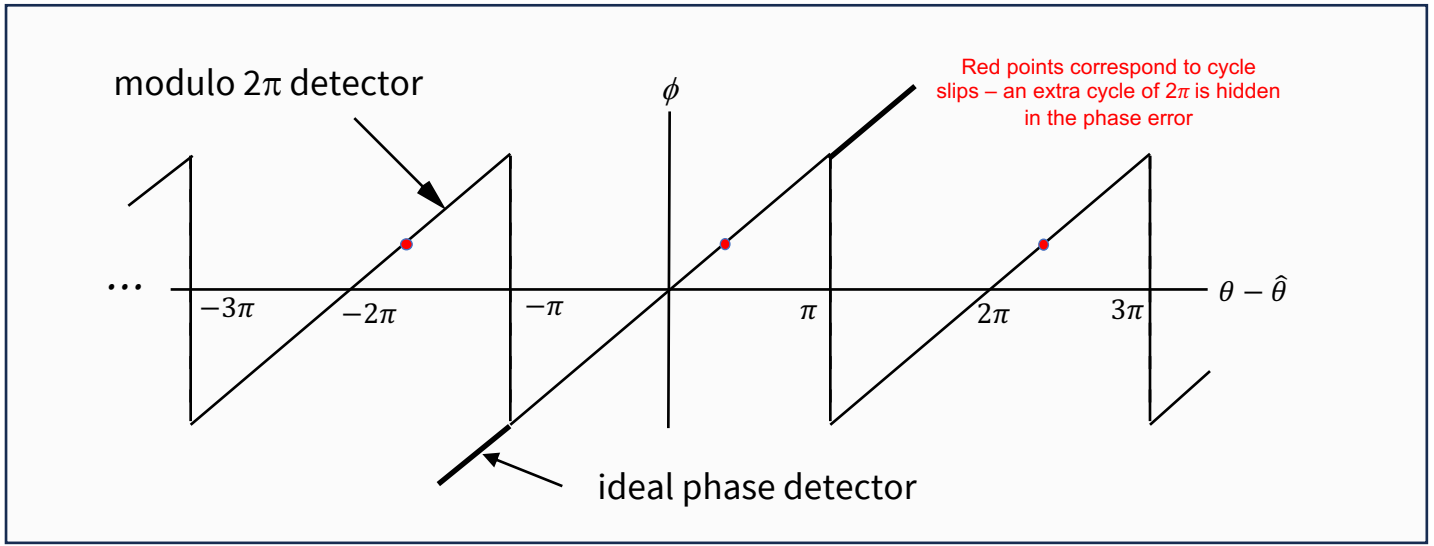
$$\phi(t) = \theta(t) - \hat{\theta}(t)$$

- The 3 basic PLL functions are the:
  - **Phase detector** – finds phase error $\phi(t)$ between current $\hat{\theta}(t)$ and desired $\theta(t)$,
  - **Loop filter** – smooths noise/jitter from **phase error,**
  - **Voltage-controlled oscillator (VCO)** – accumulates filtered phase errors to update phase estimate $\hat{\theta}(t)$.

Well, but, if I have $s(t)$, who needs to estimate phase – I already have it ???

Ideal Phase Detector

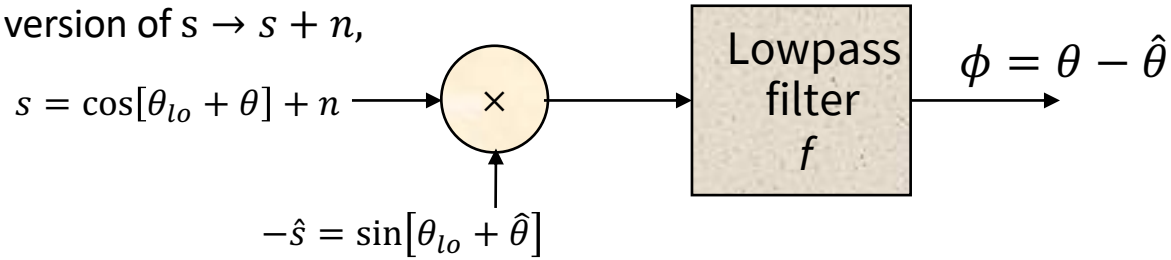
- An **ideal phase detector** would be nice to have but is not realizable – also, θ may be random.
  - And thus $\phi(t) = \theta(t) - \hat{\theta}(t)$ is random also.
  - $\mathbb{E}[\phi(t)] = 0$ is desirable.
  - **Phase jitter** $\sqrt{\mathbb{E}[\phi^2(t)]}$; slight distinction from **timing jitter** where $\delta t \triangleq$ time offset from correct zero crossing, $\sqrt{\mathbb{E}[(\delta t)^2]}$
- **Modulo-2π** Phase detector



Avoid cycle slips with local oscillater frequency within factor of 2x (1/2x to 2x). This recovers actual frequency.

Generate the phase-error directly

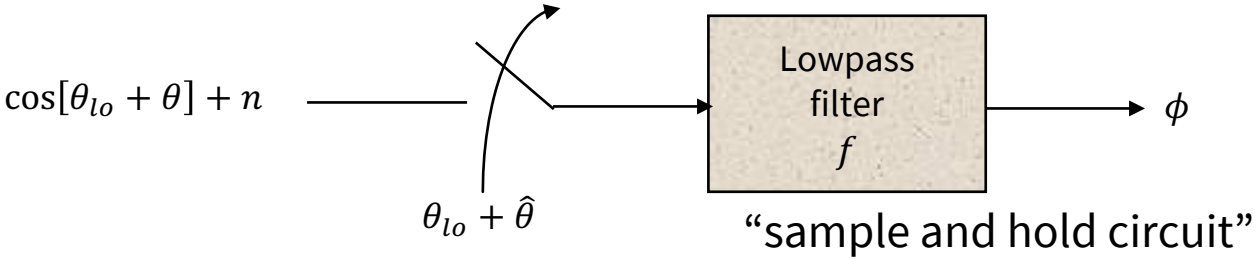
- We might have (or generate) a noisy version of $s \rightarrow s + n$,
  - Mix with local phase for phase-error:



$$-\cos(\omega_{lo}t + \theta) \cdot \sin\left(\omega_{lo}t + \hat{\theta}\right) = \frac{1}{2}\left\{-\sin\left(2\omega_{lo}t + \theta + \hat{\theta}\right) + \sin\left(\theta - \hat{\theta}\right)\right\}$$

- So the lowpass recovers $\phi(t)$, or really $\sin\phi(t) \cong \phi(t)$ for small $\phi(t)$.
- The low frequency noise in $s + n$ will also pass to phase error.

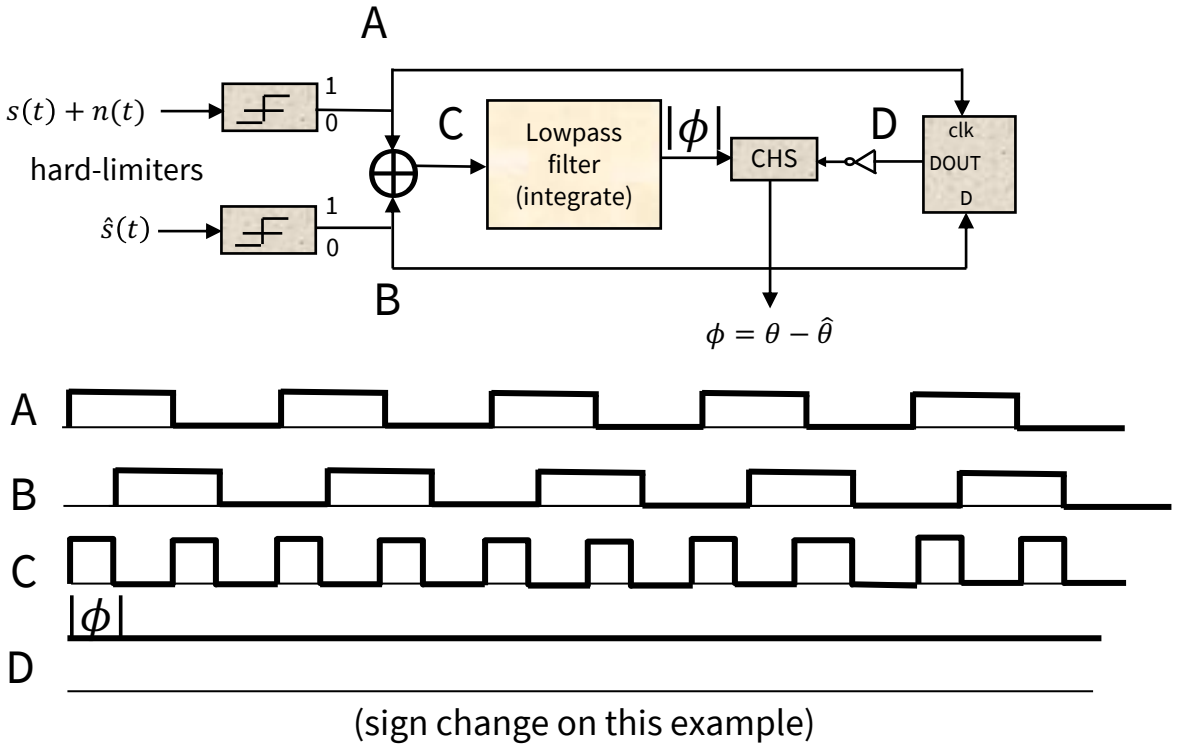
Sampling Phase Detector



- Sample on the rising edge of local sinusoid, at time where phase satisfies $\theta_{lo} + \theta = -\frac{\pi}{2}$.
- Therefore, the input sinusoid has phase $\theta_{lo} + \theta + \widehat{\theta} - \hat{\theta} = \theta_{lo} + \widehat{\theta} + \theta - \hat{\theta} = -\frac{\pi}{2} + \phi$
  - Remove offset (add $\pi/2$ - same as use sin instead of cosine).
- Again, this assumes small error so $\sin(\phi(t)) \cong \phi(t)$.

Binary Phase Detector

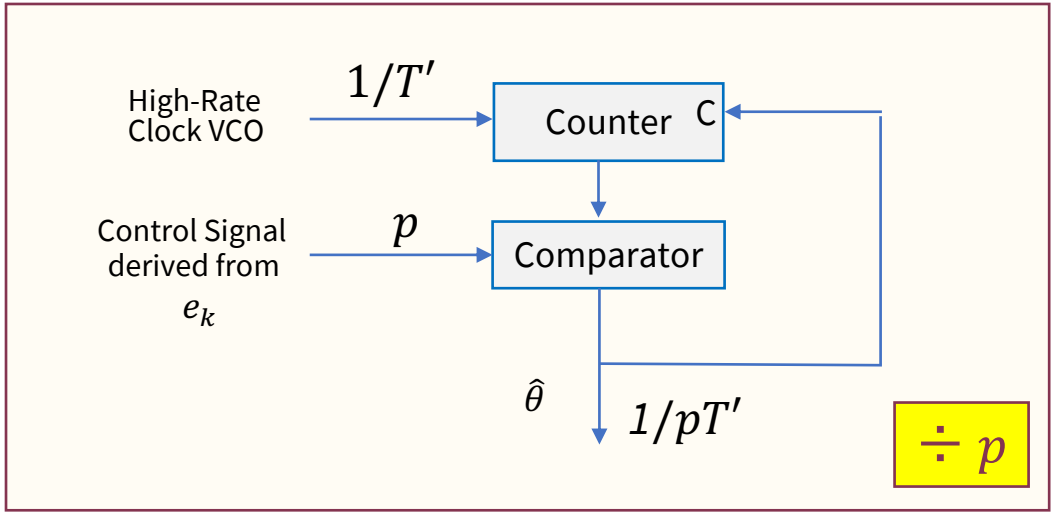
- Adjusts overlap:



Voltage-Controlled Oscillator

- Continuous
  - Purchase it – hardware. $\frac{d\hat{\theta}}{dt} = k_{vco} \cdot e(t)$
- Discrete-time
  - Look-up table and adder to generate corresponding sinusoid $\hat{\theta}_{k+1} = \hat{\theta}_k + k_{vco} \cdot e_k$

- Divider circuit: if VCO frequency is not the desired frequency.
- The clock rate is much higher than the frequency of interest, and a counter reduces it to obtain a phase estimate at desired frequency.



Jitter and distortion

- Noisy sinusoid is $x(t) = s(t) + n(t)$.
- Analysis linearizes about derivative $\delta x = \left(\frac{dx}{dt}\right) \cdot \underbrace{\delta t}_{\substack{timing \\ error}}$.
- Distortion "SNR" for the signal error related to timing error is:

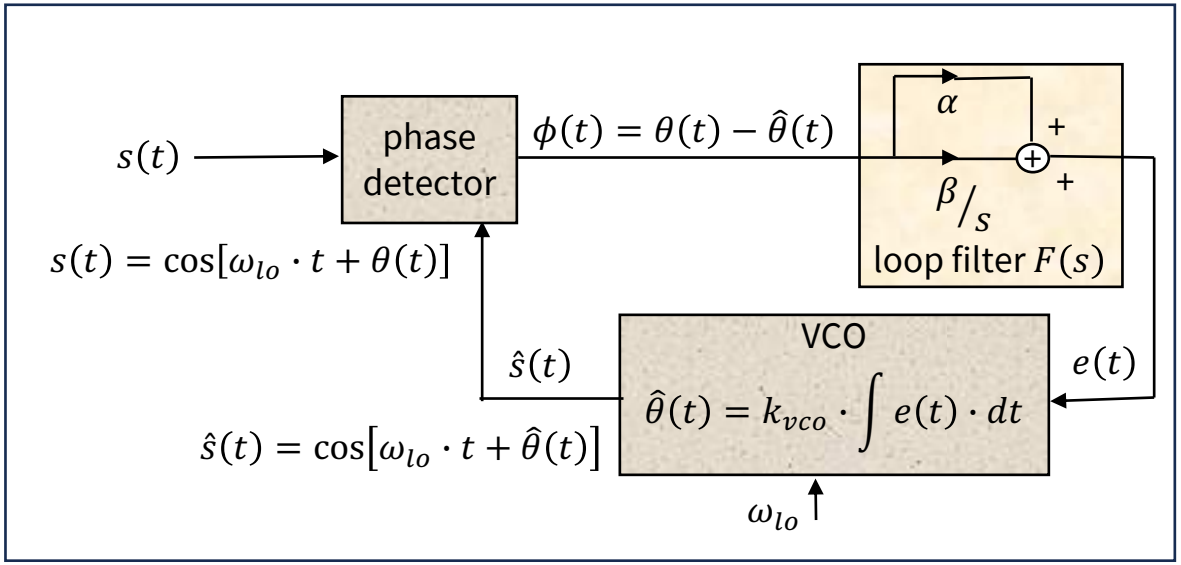
$$\text{SNR} = \frac{\mathbb{E}\left[x^2\right]}{\mathbb{E}\left[(\delta x)^2\right]} = \frac{\mathcal{E}_{\boldsymbol{x}}}{(dx/dt)^2 \cdot \mathbb{E}\left[(\delta t)^2\right]}$$

- For sinusoid nominal (or max) frequency f_{max}
  - $SNR = \frac{\mathcal{E}_x}{4\pi^2 \cdot (f_{max} \cdot \delta t)^2}$.
- To hold phase-distortion SNR constant as frequency increases, jitter must decrease
  - Requirements are tougher as bandwidth increases.

Phase Locking

Section 6.2

Continuous-Time PLLs



$$\frac{\hat{\theta}(s)}{\theta(s)} = \frac{k_{vco} \cdot F(s)}{s + k_{vco} \cdot F(s)}$$

$$\frac{\phi(s)}{\theta(s)} = 1 - \frac{\hat{\theta}(s)}{\theta(s)} = \frac{s}{s + k_{vco} \cdot F(s)}$$

Absorb k_{vco} into α and β.

- First-order PLL has $\beta = 0$:
  - $\frac{d\hat{\theta}}{dt} = \alpha \cdot \phi(t)$; can only track a constant phase offset.
- Second-order PLL has $\beta > 0$:
  - $\frac{d\hat{\theta}}{dt} = \alpha \cdot \phi(t) + \beta \cdot \int \phi(t) \cdot dt$; can track a linearly varying phase offset or frequency offset & a constant phase offset

PLL Convergence

First-Order PLL

- Constant phase input $\theta_0 \cdot u(t)$ → Laplace transform $\frac{\theta_0}{s}$.

$$\hat{\theta}(s) = \frac{\alpha}{s+\alpha} \cdot \frac{\theta_0}{s} \rightarrow \hat{\theta}(\infty) = \lim_{s \to 0} s \cdot \hat{\theta}(s) = \theta_0$$

- Equivalently $\theta_0 \cdot u(t)$ → $\phi(\infty)$=0 .
  - So if $\alpha > 0$, any constant phase offset is driven to zero by the PLL.

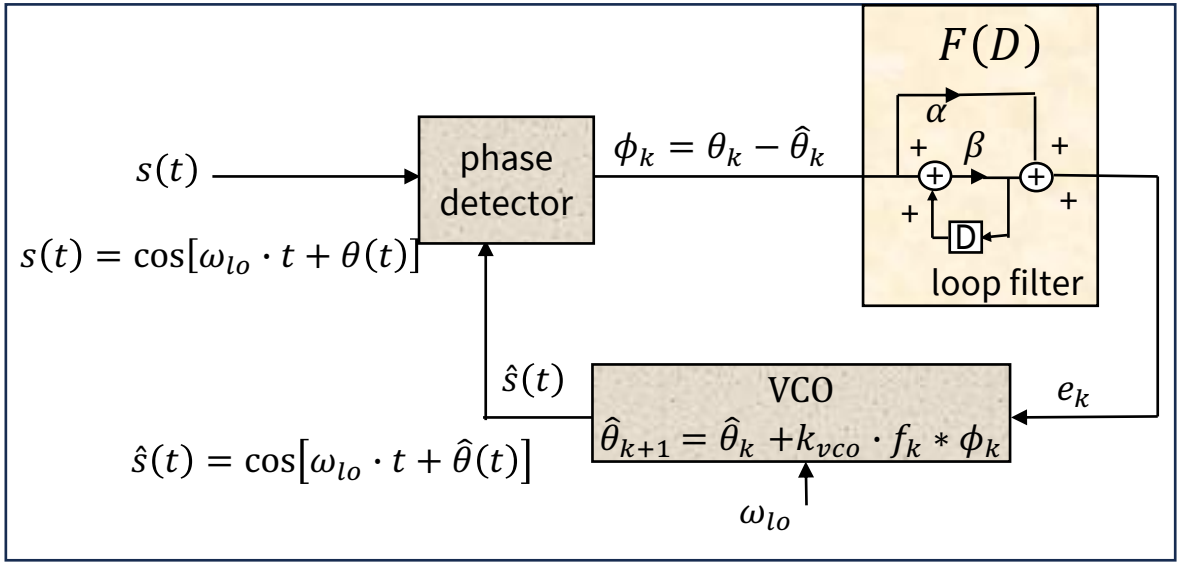
- Affine phase input $(\theta_0 + \Delta \cdot t) \cdot u(t)$, where $\Delta = \frac{d\theta}{dt} - \omega_{lo}$,→ Laplace transform $\frac{\theta_0}{s} + \frac{\Delta}{s^2}$.

- With $\beta = 0$; $\hat{\theta}(\infty) = \lim_{s \to 0} s \cdot \phi(s) = \frac{\Delta}{\alpha}$, so this is a nonzero phase error.

- With $\beta > 0$; $\phi(s) = \frac{s^2}{s^2+\alpha \cdot s+\beta} \cdot \left(\frac{\theta_0}{s} + \frac{\Delta}{s^2}\right)$ → $\lim_{s \to 0} s \cdot \phi(s) = 0$.

Second-Order PLL

Discrete-Time PLLs



- Loop Filter – VCO combination yields

$$\frac{\hat{\Theta}(D)}{\Theta(D)} = \frac{D \cdot k_{vco} \cdot F(D)}{1 - [1 - k_{vco} \cdot F(D)] \cdot D}$$

$$\frac{\Phi(D)}{\Theta(D)} = \frac{D - 1}{D \cdot [1 - k_{vco} \cdot F(D)] - 1}$$

Discrete PLL operation

- 1st Order tracks well only phase offset.
  - Discrete-time constant phase is:
    - $\theta_k = \theta_0 \cdot u_k$ or $\frac{\theta_0}{1-D}$.
  - Phase error is $\phi_k = \theta_0 \cdot (1-\alpha)^k \cdot u_k$.
  - → 0 for $k \to \infty$ if $0 < \alpha < 1$.
  - Final Value $\phi_\infty = \lim_{D \to 1} (1-D) \cdot \Phi(D)$.
- 1st order with frequency offset
  - Discrete-time frequency offset is:
    - $\theta_k = \Delta \cdot k \cdot u_k$ or $\frac{\Delta \cdot D}{(1-D)^2}$.
  - Phase error is $\phi_\infty \to {}^{\Delta}/_{\alpha} > 0$.
  - Big α , converge fast, but more jitter.
- Second Order tracks also freq offset
  - constant frequency offset
  - → 0 for $k \to \infty$ if poles outside unit circle.
  - Choose α and β for tracking speed,
  - AND ALSO for jitter limit.

First-Order PLL

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha \cdot \underbrace{(\theta_k - \hat{\theta}_k)}_{\phi_k} \qquad \frac{\Phi(D)}{\Theta(D)} = \frac{1-D}{1-(1-\alpha)\cdot D}$$

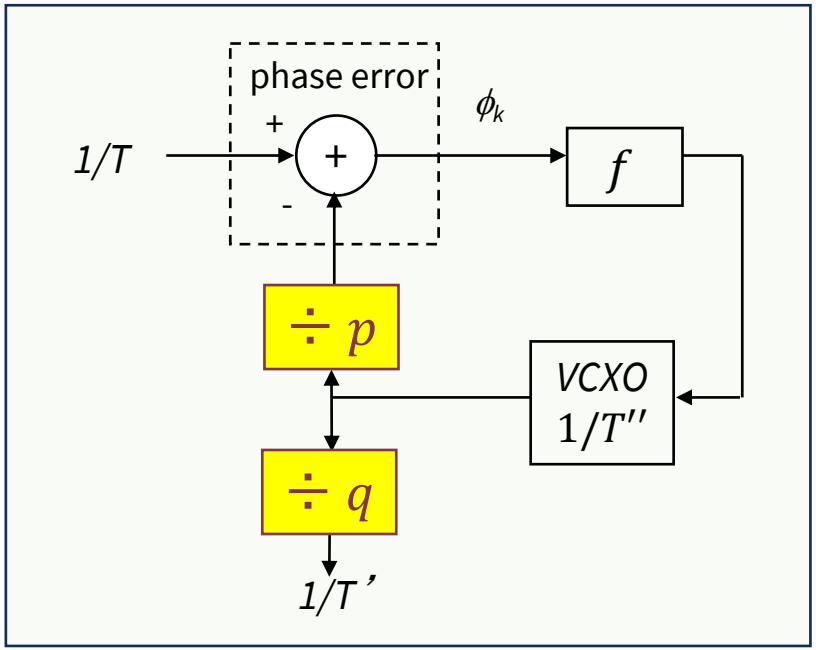
$$\hat{\theta}_{k+1} = \hat{\theta}_k + \hat{\Delta}_k + \alpha \cdot \underbrace{(\theta_k - \hat{\theta}_k)}_{\phi_k}$$

$$\hat{\Delta}_k = \hat{\Delta}_{k-1} + \beta \cdot \underbrace{(\theta_k - \hat{\theta}_k)}_{\phi_k}$$

Second-Order PLL

$$\frac{\Phi(D)}{\Theta(D)} = \frac{(1-D)^2}{1-(2-\alpha-\beta)\cdot D - (1-\alpha)\cdot D^2}$$

Phase-Locking with interpolation



This very useful for systems with packets of subsymbols.

Also when "direct conversion" is used, so carrier and symbol/sampling clocks all derive from single master clock source (VCXO)
For instance, 4G/5G use $1/T''$=30.72· *integer* MHZ.

See https://en.wikipedia.org/wiki/Crystal_oscillator_frequencies

- Main loop phase locks higher speed clock $^1/_T = {}^1/_{p \cdot T''}$, but divides by q also, so that $^1/_{T'} = {}^1/_{q \cdot T''}$
- Thus, $^1/_T = {}^q/_p \cdot {}^1/_{T'}$ where $^q/_p$ can be any rational fraction.

PLL Analysis

Section 6.2.2

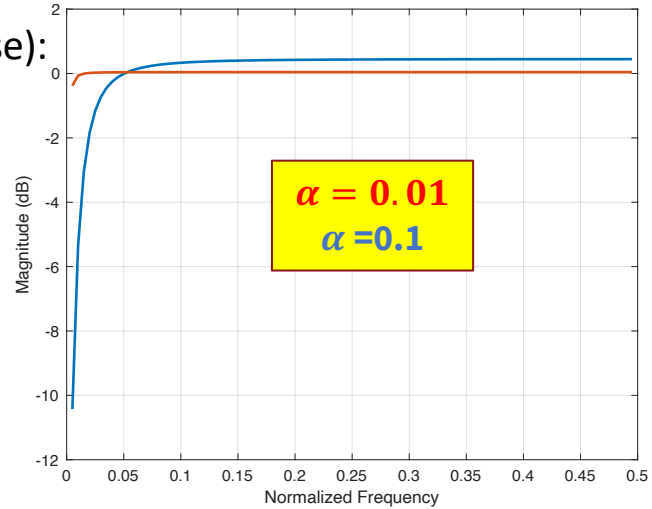
Generation of $s(t)$ and/or $\phi(t)$ has noise

- Even if it is a pilot sinusoid, there is still added channel noise.
- Data-dependent jitter (data patterns cause clock to appear variable over short term.), Secs 6.3 and 6/4.
- 1st-order PLL Noise Response is (high-pass ~ all-pass, increases noise):

$$\Phi_n(D) = \frac{1-D}{1-(1-\alpha)\cdot D} \cdot N_\theta(D)$$

$$\phi_{n,k} = (1-\alpha)\cdot\phi_{n,k-1} + \left(n_{\theta,k} - n_{\theta,k-1}\right)$$

  - DC is blocked (essentially reflects $\theta_\infty = \theta_0$).
  - "Anything else" passes to phase error almost equally.



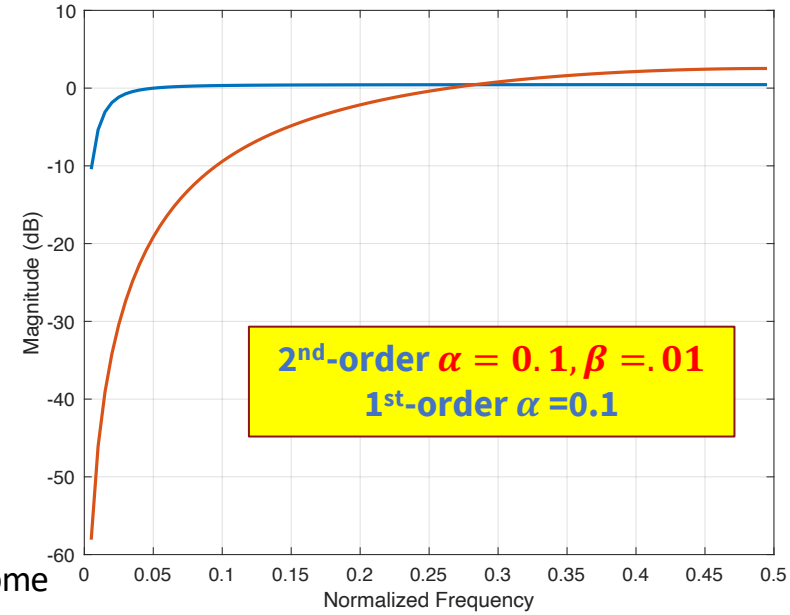
- With white noise input jitter $\sigma_\theta^2 = \mathbb{E}[\theta]$, the steady-state variance is $\sigma_\phi^2 = \mathbb{E}[\phi^2] = \frac{\sigma_\theta^2}{1-\alpha/2}$ (increases noise).
- Larger α permits faster tracking, but admits more jitter; 1st-order PLL always increases jitter.

Second Order Noise

- 2nd order PLL – Sec 6.2.2.2, Problem PS8.5 (6.2)

$$\Phi_n(D) = \frac{(1-D)^2}{1-(2-\alpha-\beta)\cdot D+(1-\alpha)\cdot D^2}\cdot N_\theta(D)$$

- Try for real roots (stability) typically ($\beta << \alpha$)
  - although can design to attempt lower (notch) gain at some freqs
    - E.g., 60 or 120 Hz jitter



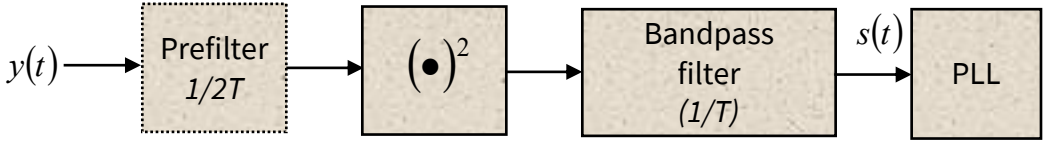
- Designer can adjust parameters
  - or use more complex filters. See Problem 6.9's 3rd-order loop for some fun (tracks Doppler).
  - Higher-order PLLs can introduce instability as some roots can approach stability boundary (unit circle) in overall feedback system.

Symbol Timing (generate a noisy s(t))

Section 6.3

Open-loop timing recovery

- Open loop does not use $\widehat{\boldsymbol{x}}$.



- The squaring produces a noisy periodic signal:

$$y^2(t) = \left[\sum_m x_m \cdot h(t - mT) + n(t)\right]^2$$

$$\begin{aligned} E\left\{y^2(t)\right\} &= \sum_m \sum_n \mathcal{E}_{\boldsymbol{x}} \cdot \delta_{mn} \cdot h(t - mT) \cdot h(t - nT) + \sigma_n^2 \\ &= \mathcal{E}_{\boldsymbol{x}} \cdot \sum_m h^2(t - mT) + \sigma_n^2 \quad , \end{aligned}$$

- So, the mean-square is periodic:

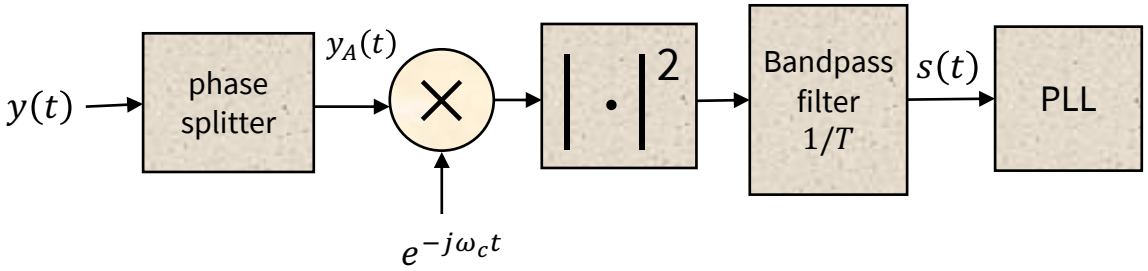
$$y^2(t) = \underbrace{\mathbb{E}[y^2(t)]}_{periodic, 1/T} + \underbrace{(y^2(t) - \mathbb{E}[y^2(t)])}_{\substack{deviation\ (noise) \\ data-dependent\ jitter}} .$$

Indeed any nonlinear function $f(y)$ that has nonzero power series term, $f''(y) \neq 0$, for can work to produce noisy periodic.

- Choose α and β accordingly , also the pre-filter @ 1/2T helps pass only frequencies that square to $1/T$.

Complex BB, QAM Version

- Envelope Timing:



- ET presumes carrier is already known, phase-locked.
- The “squaring” generalizes to complex magnitude (squared).
- Otherwise, the concept remains the same.

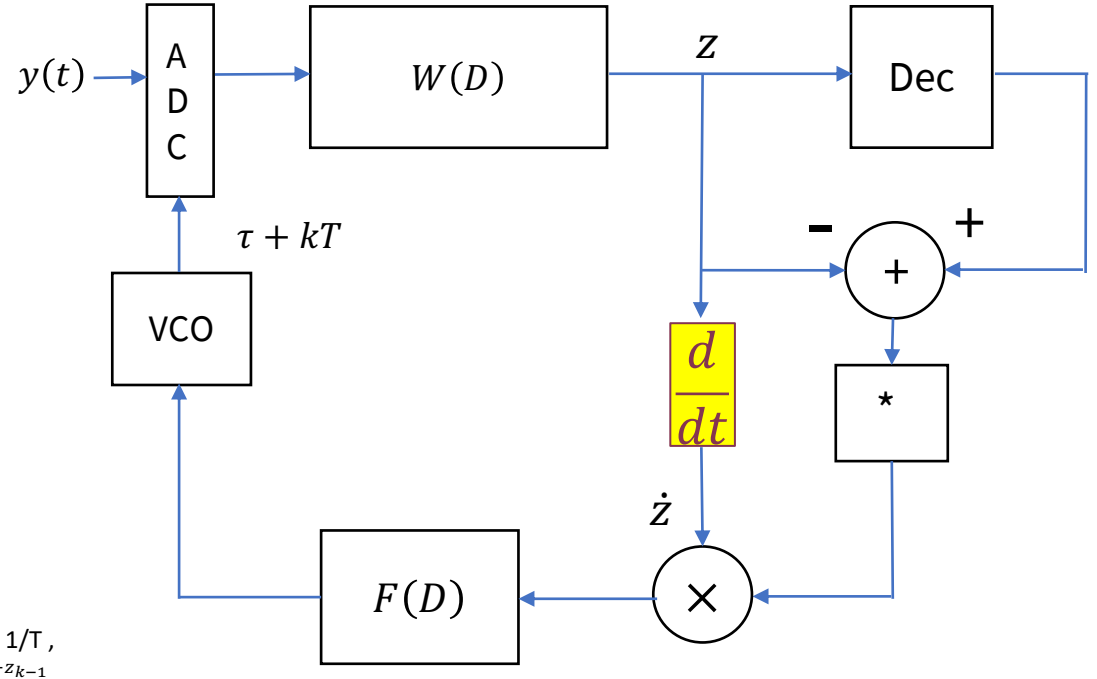
Decision-Directed Timing Recovery

- "Closed Loop"

$$J(\tau) = \mathbb{E}\left\{|\hat{x}_k - z(kT+\tau)|^2\right\}$$

$$\frac{dJ}{d\tau} = \Re\left[\mathbb{E}\left\{2\epsilon_k^* \cdot (-\frac{dz}{d\tau})\right\}\right]$$

$$\begin{aligned} \tau_{k+1} &= \tau_k + \alpha \cdot \Re\{\epsilon_k^* \cdot \dot{z}\} + T_k \\ T_k &= T_{k-1} + \beta \cdot \Re\{\epsilon_k^* \cdot \dot{z}\} \quad . \end{aligned}$$



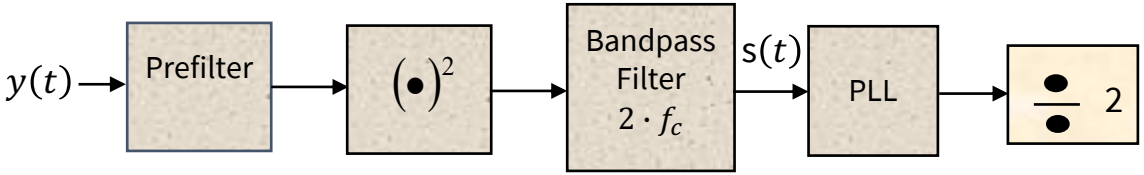
- $\frac{d}{dt}$ implementation? (by filter)
  - Filter for $\frac{d}{dt}$ is easy design at sampling rate > 1/T ,
  - but needs approximation at 1/T: $\dot{z}_k \approx \frac{z_{k+1} - z_{k-1}}{2T}$.
- Approximate $\frac{d}{dt}$ by (stochastic) gradient (sub gradient) , so $\frac{dJ}{d\tau} \approx \hat{x}_k \cdot z_{k-1} - \hat{x}_{k-1} \cdot z_k$
  - has same sign on average as true gradient, =0 when converged,
  - and presumes channel pulse-response symmetry about the optimum sampling point.

Carrier Recovery

Section 6.4

Open Loop Carrier Recovery

- Square the passband signal directly:



- Compute average/autocorrelation to analyze:

$$
\begin{aligned}
r_y(\tau) &= \mathbb{E}\{y(t)y(t-\tau)\} \\
&= \mathbb{E}\left\{\left(\frac{y_A(t)+y_A^*(t)}{2}\right)\left(\frac{y_A(t-\tau)+y_A^*(t-\tau)}{2}\right)\right\} \\
&= \frac{1}{4}\mathbb{E}\left[y_A(t)\cdot y_A^*(t-\tau)+y_A^*(t)\cdot y_A(t-\tau)\right] \\
&+ \frac{1}{4}\mathbb{E}\left[y_A(t)\cdot y_A(t-\tau)+y_A^*(t)\cdot y_A^*(t-\tau)\right] \\
&= \frac{1}{2}\Re\left[r_{y_A}(\tau)\right]+\frac{1}{2}\Re\left[r_{y_A}(\tau)\cdot e^{2\jmath(\omega_c t+\theta)}\right] \quad .
\end{aligned}
$$

- Noisy sinusoid at twice the carrier

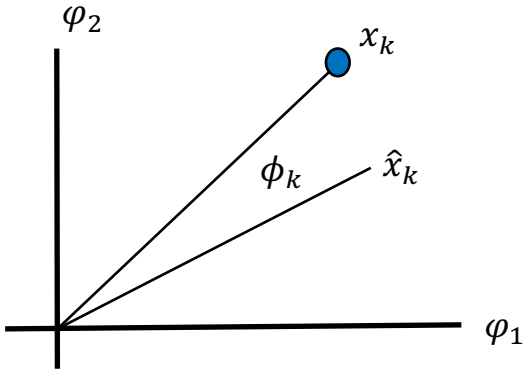
$$\mathbb{E}\left\{y^2(t)\right\} = \frac{1}{2}\Re\left[r_{y_A}(0)\right]+\frac{1}{2}\Re\left[r_{y_A}(0)e^{2\jmath(\omega_c t+\theta)}\right]$$

Decision-directed carrier recovery

- This exploit constellation rotation:

$$x_k = a_k + \jmath b_k \;\; ; \;\; \hat{x}_k = \hat{a}_k + \jmath \hat{b}_k \quad ,$$

$$\begin{aligned} \frac{x_k}{\hat{x}_k} &= \frac{|x_k|}{|\hat{x}_k|} \cdot e^{\jmath \phi_k} \\ &= \frac{a_k + \jmath b_k}{\hat{a}_k + \jmath \hat{b}_k} \\ &= \frac{\left(a_k \hat{a}_k + b_k \hat{b}_k\right) + \jmath \left(\hat{a}_k b_k - a_k \hat{b}_k\right)}{\hat{a}_k^2 + \hat{b}_k^2} \end{aligned}$$



$$\begin{aligned} \phi_k &= \arctan \frac{\hat{a}_k b_k - a_k \hat{b}_k}{a_k \hat{a}_k + b_k \hat{b}_k} \\ &\approx \arctan \frac{1}{\mathcal{E}_{\boldsymbol{x}}} \left(\hat{a}_k b_k - a_k \hat{b}_k\right) \\ &\approx \frac{1}{\mathcal{E}_{\boldsymbol{x}}} \left(\hat{a}_k b_k - a_k \hat{b}_k\right) \\ &\propto \left(\hat{a}_k b_k - a_k \hat{b}_k\right) \quad , \end{aligned}$$

- Can initialize with known training sequence

End Lecture 19
Thank you all and good luck!