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 \( \frac{1}{T_s} = \frac{p}{q \cdot T} \) 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**

- 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 ???
An ideal phase detector would be nice to have but is not realizable – also, \( \theta \) 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-\(2\pi\) Phase detector

Avoid cycle slips with local oscillator frequency within factor of 2x (1/2x to 2x). This recovers actual frequency.
We might have (or generate) a noisy version of $s \rightarrow s + n$,
- Mix with local phase for phase-error:
  \[ s = \cos[\theta_{lo} + \theta] + n \]
  \[ -\hat{s} = \sin[\theta_{lo} + \hat{\theta}] \]

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

- So the lowpass recovers $\phi(t)$, or really $\sin \phi(t) \approx \phi(t)$ 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 \( \theta_{lo} + \theta + \hat{\theta} - \hat{\theta} = \theta_{lo} + \hat{\theta} + \theta - \hat{\theta} = -\frac{\pi}{2} + \phi \)
  - 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:

\[ \phi = \theta - \hat{\theta} \]

- \( s(t) + n(t) \)
- Hard limiters
- \( \hat{s}(t) \)

Diagram:

- A
- B
- C: Lowpass filter (integrate)
- D: clk DOUT

Waveforms:

- A
- B
- C: \( |\phi| \)
- D: (sign change on this example)
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.

---

Section 6.1.2

Dashboard

Stanford University

March 14, 2024
Jitter and distortion

- Noisy sinusoid is \( x(t) = s(t) + n(t) \).

- Analysis linearizes about derivative \( \delta x = \left( \frac{dx}{dt} \right) \cdot \delta t \).

- Distortion “SNR” for the signal error related to timing error is:

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

- For sinusoid nominal (or max) frequency \( f_{\text{max}} \)
  - \( SNR = \frac{\mathcal{E}_x}{4\pi^2 \cdot f_{\text{max}} \cdot (\delta t)^2} \).

- 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

First-order PLL has $\beta = 0$:
- $\frac{d\theta}{dt} = \alpha \cdot \phi(t)$; 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

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

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

Absorb $k_{vco}$ into $\alpha$ and $\beta$. 

VCO

Phase detector

Loop filter $F(s)$

\[ s(t) = \cos[\omega_{lo} \cdot t + \theta(t)] \]

\[ \hat{s}(t) = \cos[\omega_{lo} \cdot t + \hat{\theta}(t)] \]

\[ \phi(t) = \theta(t) - \hat{\theta}(t) \]

\[ \omega_{lo} \]

\[ s(t) \rightarrow \text{phase detector} \rightarrow \phi(t) = \theta(t) - \hat{\theta}(t) \]

\[ + \]

\[ \beta / \alpha \]

\[ s(t) \]

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

\[ \frac{\phi(s)}{\theta(s)} = 1 - \frac{\hat{\theta}(s)}{\theta(s)} = \frac{s}{s + k_{vco} \cdot F(s)} \]
 PLL Convergence

- **Constant phase input** $\theta_0 \cdot u(t) \rightarrow \text{Laplace transform} \frac{\theta_0}{s}$.
  
  $$\hat{\theta}(s) = \frac{\alpha \cdot \theta_0}{s+\alpha} \rightarrow \hat{\theta}(\infty) = \lim_{s \rightarrow 0} s \cdot \hat{\theta}(s) = \theta_0$$

- Equivalently $\theta_0 \cdot u(t) \rightarrow \phi(\infty)=0$.
  - 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 $\Delta = \frac{d\theta}{dt} - \omega_l$, $\rightarrow \text{Laplace transform} \frac{\theta_0}{s} + \frac{\Delta}{s^2}$.

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

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

First-Order PLL

\[ \hat{\theta}_{k+1} = \hat{\theta}_k + \alpha \cdot (\theta_k - \hat{\theta}_k) \]

\[ \Phi(D) = \frac{1 - D}{1 - (1 - \alpha) \cdot D} \]

Second-Order PLL

\[ \hat{\theta}_{k+1} = \hat{\theta}_k + \Delta_k + \alpha \cdot (\theta_k - \hat{\theta}_k) \]

\[ \Delta_k = \Delta_{k-1} + \beta \cdot (\theta_k - \hat{\theta}_k) \]

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

1st Order tracks well only phase offset.
- Discrete-time constant phase is:
  \[ \theta_k = \theta_0 \cdot u_k \text{ or } \frac{\theta_0}{1 - D} \cdot u_k \]
  Phase error is \( \phi_k = \theta_0 \cdot (1 - \alpha)^k \cdot u_k \).
  \[ \rightarrow 0 \text{ for } k \to \infty \text{ if } 0 < \alpha < 1. \]
  Final Value \( \phi_\infty = \lim_{D \to 1} (1 - D) \cdot \Phi(D) \).

1st order with frequency offset
- Discrete-time frequency offset is:
  \[ \theta_k = \Delta \cdot k \cdot u_k \text{ or } \frac{\Delta \cdot D}{(1 - D)^2} \cdot u_k \]
  Phase error is \( \phi_\infty \to \Delta / \alpha > 0. \)
  Big \( \alpha \), converge fast, but more jitter.

Second Order tracks also freq offset
- Constant frequency offset
- \( \rightarrow 0 \) for \( k \to \infty \) if poles outside unit circle.
- Choose \( \alpha \) and \( \beta \) for tracking speed,
  AND ALSO for jitter limit.
Main loop phase locks higher speed clock \( \frac{1}{T} = \frac{1}{p \cdot T''} \), but divides by \( q \) also, so that \( \frac{1}{T'} = \frac{1}{q \cdot T''} \).

Thus, \( \frac{1}{T} = \frac{q}{p} \cdot \frac{1}{T'} \), where \( \frac{q}{p} \) can be any rational fraction.

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 \( \frac{1}{T''} = 30.72 \cdot \text{integer MHZ} \).

See [https://en.wikipedia.org/wiki/Crystal_oscillator_frequencies](https://en.wikipedia.org/wiki/Crystal_oscillator_frequencies) Section 6.2.3
PLL Analysis

Section 6.2.2
Even if it is a pilot sinusoid, there is still added channel noise.


1\textsuperscript{st}-order PLL Noise Response is (high-pass ~ all-pass, increases noise):

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

\[
\phi_{n,k} = (1 - \alpha) \cdot \phi_{n,k-1} + (n_\theta,k - n_\theta,k-1)
\]

- 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 \( \alpha \) permits faster tracking, but admits more jitter; 1\textsuperscript{st}-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.

2nd-order $\alpha = 0.1, \beta = .01$

1st-order $\alpha = 0.1$
Symbol Timing

(generate a noisy $s(t)$)

Section 6.3
Open-loop timing recovery

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

- The squaring produces a noisy periodic signal:
  \[ y^2(t) = \left[ \sum_m x_m \cdot h(t-mT) + n(t) \right]^2 \]
  
  \[ E \{ y^2(t) \} = \sum_m \sum_n E_x \cdot \delta_{mn} \cdot h(t-mT) \cdot h(t-nT) + \sigma_n^2 \]
  
  \[ = E_x \cdot \sum_m h^2(t-mT) + \sigma_n^2 \]

- So, the mean-square is periodic:
  \[ y^2(t) = \mathbb{E}[y^2(t)] + (y^2(t) - \mathbb{E}[y^2(t)]) \]

  periodic, $1/T$
  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 $\alpha$ and $\beta$ accordingly, also the pre-filter @ 1/2T helps pass only frequencies that square to $1/T$. 

March 14, 2024

Section 6.3.1

Stanford University
Envelope Timing:

$y(t) \rightarrow$ phase splitter $y_A(t) \rightarrow | \cdot |^2 \rightarrow$ Bandpass filter $1/T \rightarrow s(t) \rightarrow$ PLL

$e^{-j\omega_c t}$

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

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

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

\[ \tau_{k+1} = \tau_k + \alpha \cdot \Re\{\epsilon_k^* \cdot \dot{z}\} + T_k \]

\[ T_k = T_{k-1} + \beta \cdot \Re\{\epsilon_k^* \cdot \dot{z}\} . \]

- **Implementation? (by filter)**
  - Filter for \( \frac{d}{dt} \) is easy design at sampling rate > 1/T,
  - but needs approximation at 1/T: \( \dot{\hat{z}}_k \approx \frac{z_{k+1} - z_{k-1}}{2T} \).

- 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.

Section 6.3.2-4
Carrier Recovery

Section 6.4
Open Loop Carrier Recovery

- Square the passband signal directly:

\[ y(t) \xrightarrow{\text{Prefilter}} (\cdot)^2 \xrightarrow{\text{Bandpass Filter } 2 \cdot f_c} s(t) \xrightarrow{\text{PLL}} \frac{\bullet}{2} \]

- Compute average/autocorrelation to analyze:

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

- Noisy sinusoid at twice the carrier

\[
\mathbb{E} \{ y^2(t) \} = \frac{1}{2} \Re [r_{yA}(0)] + \frac{1}{2} \Re \left[ r_{yA}(0) e^{2j(\omega_c t + \theta)} \right].
\]
Decision-directed carrier recovery

- This exploit constellation rotation:

\[
x_k = a_k + jb_k \quad ; \quad \hat{x}_k = \hat{a}_k + j\hat{b}_k ,
\]

\[
\frac{x_k}{\hat{x}_k} = \frac{|x_k|}{|\hat{x}_k|} \cdot e^{j\phi_k}
\]

\[
= \frac{a_k + jb_k}{\hat{a}_k + j\hat{b}_k}
\]

\[
= \left( a_k \hat{a}_k + b_k \hat{b}_k \right) + j \left( \hat{a}_k b_k - a_k \hat{b}_k \right)
\]

\[
\phi_k = \arctan \frac{\hat{a}_k b_k - a_k \hat{b}_k}{a_k \hat{a}_k + b_k \hat{b}_k}
\]

- Can initialize with known training sequence
End Lecture 19
Thank you all and good luck!