Phase-Locked Loop (PLL) and Carrier Synchronization

Fuyun Ling

fling@twinclouds.com

Outline

- Part 1: Basics of Phase-locked loop (PLL)
  - PLL overview,
  - Analog, Digital and Mixed Signal PLLs
  - Application of PLL in Digital Communications
- Part 2: Carrier Synchronization
  - Overview of carrier synchronization in digital communication systems
  - Carrier synchronization in wireline communication systems
  - Carrier synchronization in wireless communication systems
    - Examples: Single carrier and OFDM
- What Have Been Discussed

fling@twinclouds.com

Part 1: BASICS OF PHASE LOCKED LOOP (PLL)

fling@twinclouds.com

PLL Overview

- Phase-locked loops (PLLs) are important components for carrier and timing synchronization in digital communication systems
  - PLLs are close-loop feedback systems that can perform accurate synchronization with low precision requirement for implementation
- PLLs were mostly implemented in analog form in the past
- Its digital forms become more and more popular recently
  - We shall mainly discuss and analyze digital PLLs (DPLLs)
  - DPLL can be implemented using mixed components (A & D)
- PLLs are low order feedback systems, their analysis are quite simple but still very important
  - Some people are mostly relied on simulation nowadays. However, I feel some simple linear analyses can provide us a lot of insights!

fling@twinclouds.com

PLL Overview (cont.)

- The main components of DPLLs are:
  - Phase error detector
  - Loop filter
  - Correction feedback controller
- The order of a loop filter determines the order of the PLL
  - Mostly used forms of PLLs are first and second order ones
    - First order PLL can completely correct a constant phase error with no residual phase error on average (bias) in steady state
    - However, it cannot correct a linearly increasing error, e.g., a constant frequency offset, without a steady state bias
    - A second order PLL can correct a linearly increasing phase error
    - Higher order PLLs are only used for special purposes
- Phase Detector
  - Phase detector has a variety of forms depending on applications
  - They be discussed in detail when dealing with such applications

fling@twinclouds.com

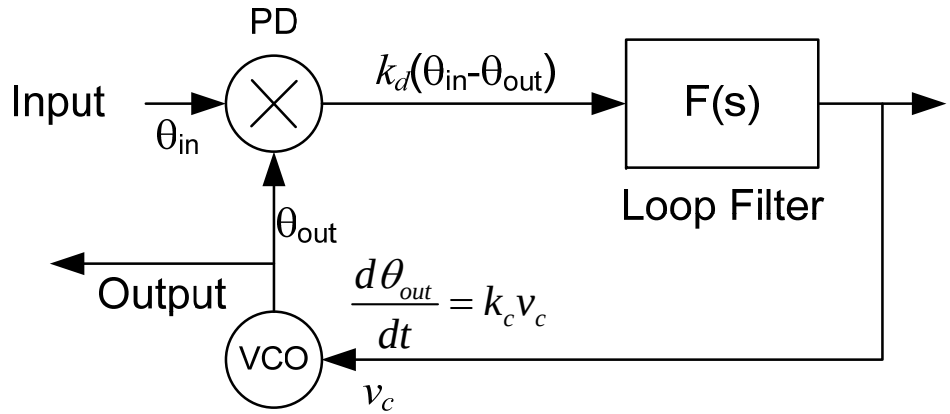
PLL Overview (cont.)

- Correction feedback controller
  - The correction controller can take analog and digital forms
  - They are application specific and will be discussed later
- The most important parameters are: Time constant, damping factor, loop noise bandwidth and pull-in range
  - Time constant determines the converging speed of PLL
  - Loop noise bandwidth determines the phase noise attenuation by the PLL
    - It is approximately inversely proportional to the time constant
  - Damping factor is a parameter of second order loops
    - Determines if and how the loops oscillate
  - pull-in range determines tolerable initial frequency offset

fling@twinclouds.com

Analog PLL (APLL)

- Basic APLL:



  - PD – Phase-error Detector (to be discussed in detail later)
  - VCO – Voltage Controlled Oscillator
- The output of VCO is phase-locked to the input (reference)
- Type of loop filter's F(s) determines the order of APLL
  - Closed loop transfer function:

$$H(s) = \frac{\theta_{out}(s)}{\theta_{in}(s)} = \frac{k_c k_d F(s)}{s + k_c k_d F(s)}$$

fling@twinclouds.com

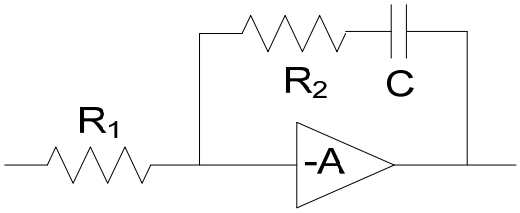
APLL (cont.)

- First order APLL – F(s) is a constant A

$$H(s) = \frac{K}{s+K} \qquad K = k_c k_d A$$

  - It has an exponential impulse response
- Second order APLL – A simple example

Loop Filter



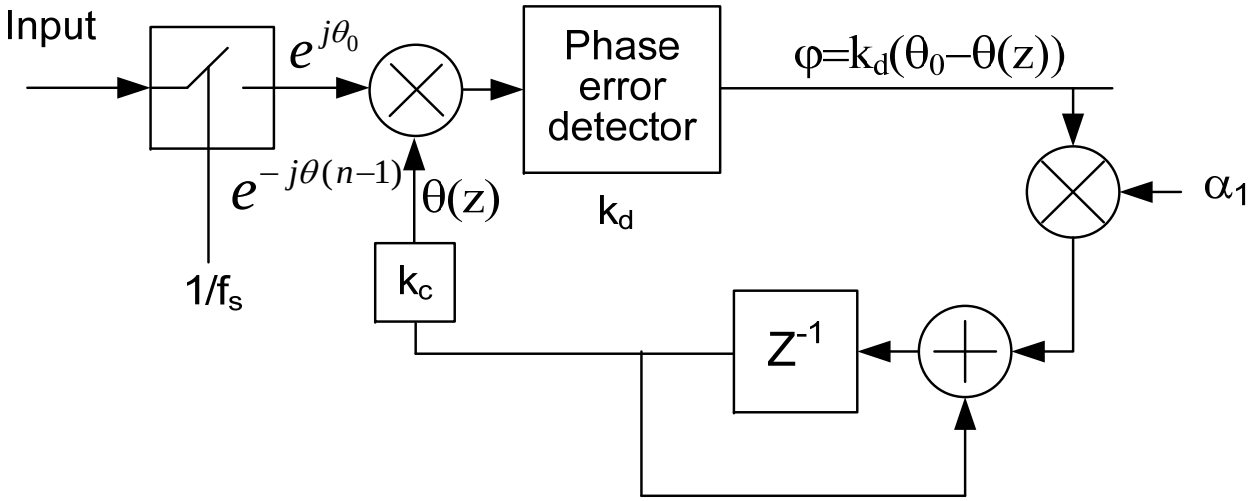
$$F(s) = \frac{-A(sCR_2+1)}{sCR_2+1+(1+A)sCR_1} \cong \frac{sCR_2+1}{sCR_1}, \text{ for } A >> 1$$

$$H(s) = \frac{k_c k_d (s\tau_2+1)/\tau_1}{s^2+s(k_c k_d \tau_2/\tau_1)+k_c k_d/\tau_1} \cong \frac{2\zeta\omega_n s+\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

  - ω_n: natural frequency, ζ: damping factor, where τ_1 = R_1C, τ_2 = R_2C
  - PLL characteristics mainly determined by the denominator of H(s)
  - Can also be implemented using only passive components (R, C)

fling@twinclouds.com

First Order Digital PLL (DPLL)



- System sampled at a rate of f_s, (one sample every $1/f_s$ seconds)
- The linearized system equation is:

$$\frac{\theta(z)}{\theta_0(z)} = \frac{\alpha_1 k_c k_d z^{-1}}{1-(1-\alpha_1 k_c k_d)z^{-1}} = \frac{k_1 z^{-1}}{1-(1-k_1)z^{-1}} = \frac{k_1}{z-(1-k_1)}$$

  - k_d is the phase detector (PD) gain with a unit of radius/number-unit
  - k_c is the phase conversion gain with a unit of number-unit/radius
  - $k_1 = \alpha_1 k_c k_d$ is the *loop gain*

fling@twinclouds.com

First Order DPLL (cont.)

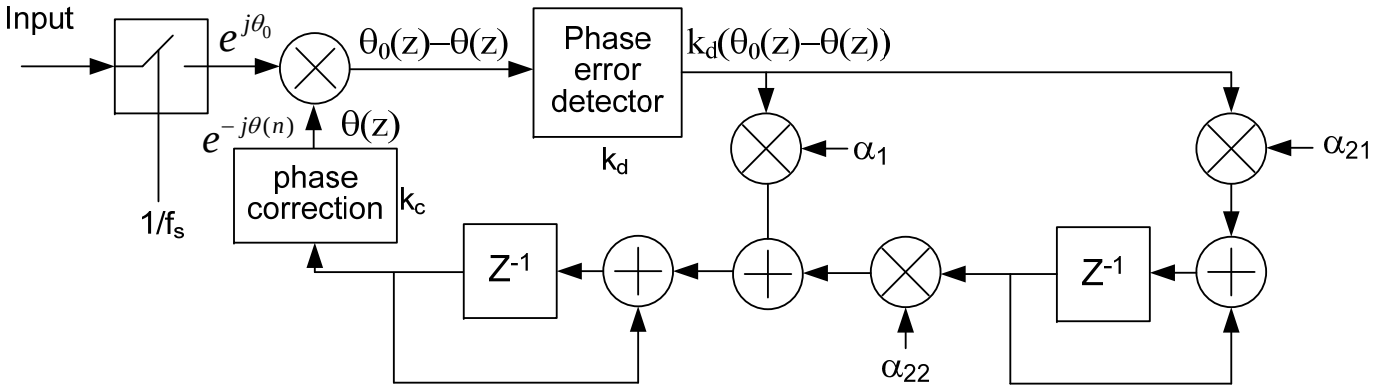
- Linear System analysis:
  - This is a first order system. It is stable as long as the loop gain k_1 <1.
  - If θ_0 is a constant, θ(n) will exponentially converge to θ_0 at $(1-k_1)^n$.
  - Its (analog) time constant is T_s/k_1 (seconds).
  - Its (single-sided) noise bandwidth is equal to $0.25k_1/T_s$ (Hz).
    - The total phase noise at the PLL output is in order of $4/k_1$ time of that at PD output
  - The first order loop can compensate for a constant phase offset with the steady state average phase error equal to zero
  - There will be a steady state phase error if there is a frequency offset
    - It is balanced if $2\pi\,\Delta f / f_s = k_1 \Delta\theta_{steady\ state}$, thus,

$$\Delta\theta_{steady\ state} = \frac{2\pi\Delta f}{k_1 f_s}\ (\text{radius})$$

  - The lock-in and tracking ranges are $\Delta f < \Delta\theta_{\max} k_1 f_s / 2\pi$
    - In most cases $\Delta\theta_{\max} < \pi$

fling@twinclouds.com

Second Order DPLL



- The linearized system equation is:

$$\frac{\theta(z)}{\theta_0(z)} = \frac{k_1 z + (k_2 - k_1)}{z^2 - (2 - k_1)z + (1 - k_1 + k_2)}$$

  - $k_1 = \alpha_1 k_c k_d$ and $k_2 = \alpha_2 k_c k_d$ ($\alpha_2 = \alpha_{21}\alpha_{22}$)
  - k_d is the phase detector gain with a unit of radius/number-unit
  - K_c is phase conversion gain with a unit of number-unit/radius

fling@twinclouds.com

Second Order DPLL (cont.)

- Using time impulse invariance mapping $z = (1-s)^{-1}$ we convert the system equation to s-domain:

$$\frac{\theta(s)}{\theta_0(s)} = \frac{(k_1 - k_2)\frac{1}{1-s} - k_1}{\left(\frac{1}{1-s}\right)^2 - (2+k_1)\frac{1}{1-s} + (1+k_1+k_2)} = \frac{-k_2 + (k_2+k_1)s - k_1 s^2}{k_2 + (k_1 - 2k_2)s + s^2(1-k_1+k_2)}$$

- The two roots are: $r_{1,2} = \frac{1}{2(1-k_1+k_2)}\left[-(k_1-2k_2) \pm \sqrt{k_1^2 - 4k_2}\right]$
  - System is stable as long as $k_1 > k_2 > 0$
  - System critical damped if $k_1^{\,2} = 4k_2$, i.e., $k_1 = 2\sqrt{k_2}$
  - System underdamped (oscillating when converging) if $k_1 < 2\sqrt{k_2}$

fling@twinclouds.com

Second Order DPLL (cont.)

- Denominator in standard second order linear system form:

$$s^2 + k_1 s + k_2 \cong s^2 + 2s\zeta\omega_n + \omega_n^2$$

- The natural frequency is:

$$\omega_n^2 = \frac{k_2}{1 - k_1 + k_2}, \text{ or } \omega_n = \sqrt{\frac{k_2}{1 - k_1 + k_2}}$$

- The system is critical damped if $\zeta = 1$. It can be shown that $\zeta = 1$ corresponds to $k_1 = 2\sqrt{k_2}$ – the same as the z-domain result

fling@twinclouds.com

Second Order DPLL (cont.)

- Characteristics of second order DPLL
  - For a second order loop, the time constant and noise bandwidth depend on both k_1 and k_2. However, they largely depend on k_1.
  - When $k_2 << k_1 << 1$, as usually the case, they take forms close to that in the first order loop. The initial converging time constant and the noise bandwidth are approximately equal to T_s/k_1 and $0.25(k_1+k_2/k_1)/T_s \approx 0.25k_1/T_s$.
  - When $k_2 << k_1 << 1$, we have $\omega_n \approx \sqrt{k_2}$ and $\zeta = k_1/2\sqrt{k_2}$. For critical damping, $k_1 = 2\sqrt{k_2}$.
  - *A critical damped loop will still be critical damped if k increase/ decrease by a factor of A and k_2 by a factor of the square root of A*.
    - This relationship is especially useful when changing time constant during real time operation, e.g., training, while keeping loop critically damped.
  - The natural frequency is useful for APLL in control applications. In modem applications, k_2 mainly controls the damping factor.

fling@twinclouds.com

Second Order DPLL (cont.)

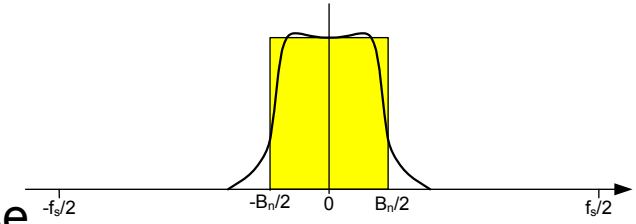
- Characteristics (cont.)
  - For the above analog PLL approximation to be accurate, the time constant need to be much longer than the sampling time interval.
    - The ratio should be at least ten times or more, i.e., k_1 should be less than 0.1, preferably 0.02 or even less.
  - Pull in range:
    - Since $k_2 << k_1$, when the PLL starts, the phase correction value is mainly determined by k_1. k_2 helps only marginally.
    - The phase correction is equal to $k_1 x \Delta\theta$ per update.
    - The phase error for each sampling period due to frequency offset is equal to $2\pi \times f_{offset} / f_s$. Thus, the frequency error can be corrected is or $2\pi f_{offset} / f_s < k_1 \Delta\theta_{\max}$, i.e., $f_{offset} < k_1 f_s \Delta\theta_{\max} / 2\pi$
    - $\Delta\theta_{max}$ is determined by the phase detector implementation. It is usually less than π, but in some cases it can be larger
    - The tracking range can be larger if “cycle slipping” is allowed

fling@twinclouds.com

Second Order DPLL (cont.)

- Characteristics (cont.)
  - Noise Bandwidth B_n
    - There are different ways to define the bandwidth of a PLL
    - The most useful measure of PLL's bandwidth is the *equivalent noise bandwidth.* Denoting the *two sided* noise bandwidth by B_n, we have

$$B_n = \int_{-f_s/2}^{f_s/2} \left|H_D(f)\right|^2 df \Big/ \left|H_D(0)\right|^2 \quad \text{(Hz)}$$



where, $H_D(f)$ is the DPLL's frequency response.

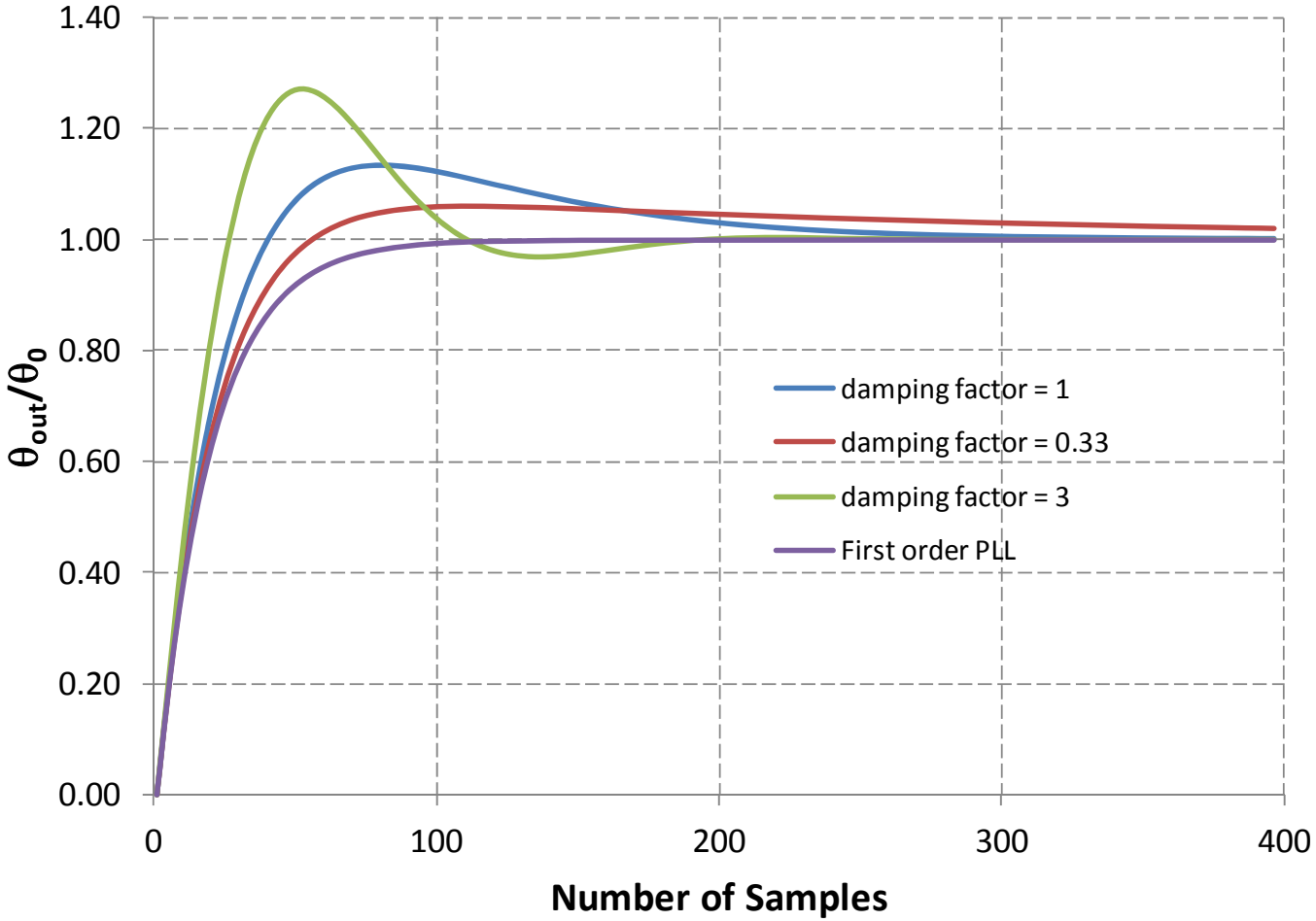
    - Assuming the phase jitter is white with a power denisity of δ_0, the variance of the phase jitter at the DPLL output is equal to $\delta_0 B_n/f_s$.
    - For second order PLL

$$B_n = \omega_n \left(\zeta + \frac{1}{4\zeta} \right)$$

    - With $\omega_n = \sqrt{k_2}/2$ and $\zeta = k_1/2\sqrt{k_2}$ we have $B_n = 0.25\left(k_1 + k_2/k_1\right) \approx 0.25k_1$ for k2 << k1

Second Order DPLL Initial Convergence

(k_1 = 0.05)



fling@twinclouds.com

DPLL Implementation considerations

- The values in delay elements of the PLLs is usually stored as integers, which may be represented in fractional form in the range of [-1, 1)
- The first order accumulator (left delay element in the 2nd PLL)
  - the entire value can wrap around when overflow, and corresponding to $-\pi$ to π (there's an inherent scale factor of π)
- The second order accumulator (2nd PLL's right delay element)
  - The value in the delay element determines the maximum frequency offset that can be handled.
  - It is desirable to split the second order coefficient to two parts: α_{21} and α_{22}, as shown in the system block diagram.
    - Denote the maximum value in the delay element to be A.
    - Every $1/f_s$ the maximum value can be added to the first order loop is equal to α_{22} A, i.e., $2\pi\alpha_{22}$ A (radius)
    - The maximum frequency offset can be handled is $\alpha_{22} A f_s$

fling@twinclouds.com

DPLL Implementation considerations (cont.)

- It is usually convenient to partition the m-bit integer value in the delay element in two parts (M:m-M bits)
- In the first order portion the top M bits are used for the phase correction
  - M determines the phase correction resolution
  - Small M reduces multiplier or look-up table size but also the resolution
  - The bottom m-M bits are for accumulation of phase error
    - When k_1 is small this reduces the possibility of underflow and bias
    - This only require a longer accumulator – less resource consuming
  - We consider the value in the range of [-1, 1), corresponding to a phase range of [$-\pi$, π)
    - It is used for table look-up to generate the sine and cosine values in the sample phase rotator
    - There is a inherent gain factor of π when compute the loop gain

fling@twinclouds.com

DPLL Implementation considerations (cont.)

- For the second order portion, the top M bits is for phase correction added every sample interval
  - It determines the maximum frequency offset that can be handled and the frequency resolution
  - Since k_2 is even smaller than k_1, the underflow possibility is even larger
    - m – M may need to be even longer, e. g., m = 32 and M = 16
    - Unbiased rounding may be used to further reduce bias in frequency error estimation and compensation
- These details need to be dealt with carefully in the computation of the loop gains k_1 and k_2 for DPLL design
- If delays are introduced by one or more components, the change of PLL characteristics will be insignificant as long as the delay is relatively short relative to its time constant.

fling@twinclouds.com

Applications of PLL in Digital Communications

- Carrier and timing synchronization
  - The carrier and timing frequency and phase are estimated during the initial acquisition stage and used for their PLL initialization
  - Their PLLs then enter training stage
    - PLLs usually have short time constant to speed up training
  - Once trained satisfactory, the receiver enters data mode
    - PLLs usually have long time constants to reduce estimated phase error
  - The carrier and timing are synchronous to the remote transmitter
  - Details will be discussed below and in future presentations
- Tx frequency syntheses
  - Once synchronized, the receiver of the terminal has perfect knowledge of the carrier frequency of the remove transmitter
  - PLL is used to synthesize the terminal transmitter frequency
  - An example is give below

fling@twinclouds.com

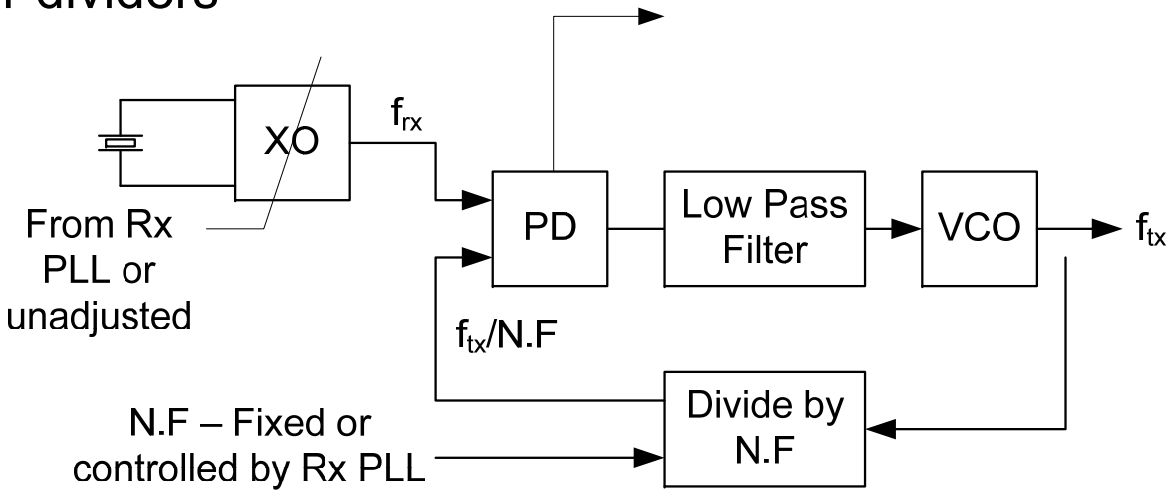
An Example: Frequency Synthesis

- In a mobile communication system, BTS (base-station) usually have a quite accurate transmitter carrier frequency
  - Base stations can afford good crystal oscillator
  - They can synchronize to GPS
- Mobile usually have a lower quality crystal oscillator to save cost and power
- Mobile receiver can learn BTS carrier frequency on average through carrier synchronization
  - There will be some time-varying error due to channel fading
- Accurate mobile Tx carrier frequency can be achieved based on the receiver clock synchronized to BTS Tx
  - Mobile Tx frequency is synthesized based on the recovered receiver clock

fling@twinclouds.com

Mobile TX Frequency Synthesis

- Assume the receiver generate an accurate clock at f_{rx} that is synchronous to the BTS Tx clock.
- The task is to generate and local Tx clock f_{tx} that is synchronous to f_{rx} but differ by a ratio of N.F
  - N and F are the integer and fractional parts, respectively
  - Can be implemented using a hardware fractional divider or switching between two or more integer dividers, e.g. using N and N+1 dividers



fling@twinclouds.com

Mobile TX Frequency Synthesis (cont.)

- Mobile Tx clock is generated using a VCO (with LC or ceramic resonators) at or higher than desired carrier frequency
- The reference frequency is generated from
  - An Rx VCXO that is adjusted to lock to the recovered BTX clock
  - or a fixed frequency Rx XO (unadjustable)
- The Tx VCO output is divided by N.F to compare with the Rx VCXO/XO output
  - For Rx VCXO synchronous to the BTX clock, the N.F divider can be pre-computed
  - For the fixed frequency XO, the divided down number is computed based on the information recovered by the Rx
  - The (VC)XO output may also be divided down to improve the resolution of the Tx frequency

fling@twinclouds.com

Mobile TX Frequency Synthesis (cont.)

- The phase-difference between the divided-down Tx VCO clock and the clock of the (divided down) Rx VCXO/XO output is generated in the Phase-Detector (PD)
- The PD output proportional to the phase difference drives the Tx VCO to form a negative feedback loop
- Noises (from electrical circuitries) and jitters (generated by the N.F divider) are reduced by the low pass filter
- This entire loop is a *mixed signal PLL* (with analog and digital components) which is in steady state once the VCO and the Rx clock are frequency and phase synchronized.
  - The lowpass filter will have only minor impact to the overall characteristics if its delay is relatively short to the PLL constant, i.e., the PLL bandwidth is narrower than the filter bandwidth

fling@twinclouds.com

Phase Detector in Frequency Synthesizer

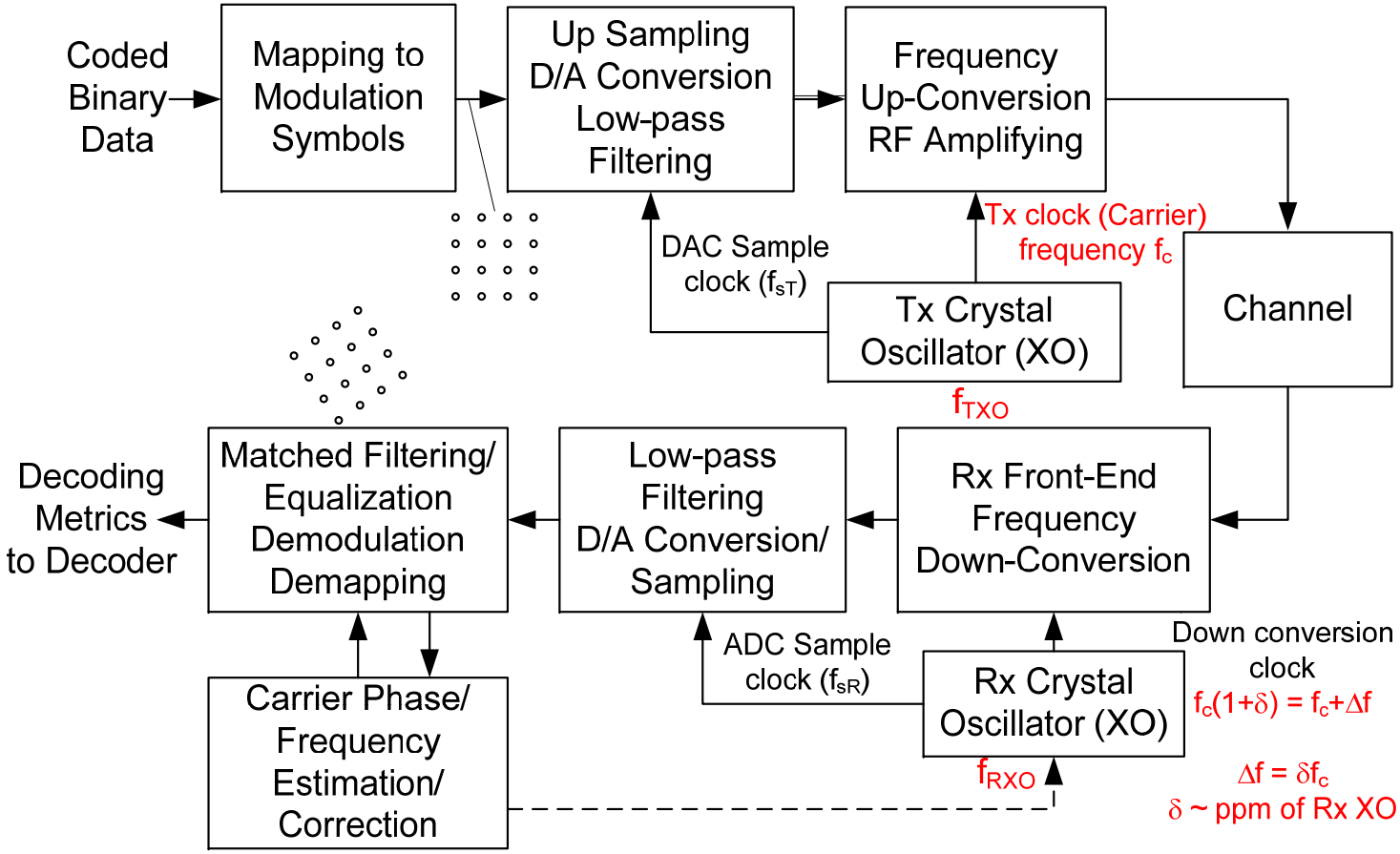
- Multiplier Phase Detector
  - Reference: $u_1(t) = U_1 \sin(\omega t + \theta_1)$,
  - Divider Output: $u_2(t) = U_2 \text{rect}\left((\omega + \Delta\omega)t + \theta_2\right)$ "rect" is an *even* rectangular function whose first harmonic is a cosine function)
  - PD output: $v(t) = u_1(t) \times u_2(t) = c_0 \sin(-\Delta\omega + \theta_1 - \theta_2) +$ high order harmonics
  - For $\Delta\omega \approx 0$, and with low-pass filtering out the high order terms, $v(t)$ is proportional to the phase difference between u_1 and u_2
- Digital (EXOR or Flip-Flop) Phase Detector
  - By converting $u_1(t)$ to rectangular waveform $\tilde{u}_1(t)$ the phase difference between the rising or falling edges of $\tilde{u}_1(t)$ and $u_2(t)$ can be detected using EXOR, Flip-Flop circuits or other means.
  - If a divider with fractional portion is used, the rectangular clocks generated are not uniform, the phase difference detected will be jittery. Low-pass filtering should be used to smooth out the jitter.

fling@twinclouds.com

Part 2: Carrier Synchronization

fling@twinclouds.com

Block Diagram of a Digital Communication System



fling@twinclouds.com

The importance of Carrier Synchronization

- There are two aspects of carrier synchronization between transmitter and receivers of a communication link: phase and frequency synchronizations
- To achieve coherent detection it is essential for receiver to know the phase of the received symbols
  - Coherent communication system can achieve better performance than a non-coherent one, even with pilot overhead and channel estimation error in channel estimation
- Non-coherent detection can tolerant phase offset but there's a limit for acceptable frequency offset
- Large frequency offset will degrade initial acquisition performance
- It is usually desirable for the Tx and Rx are frequency locked in the same device

fling@twinclouds.com

Sources of Carrier Out of Synch

- Carrier frequency offset due to modulation/demodulation frequency mismatch
  - This is the main source of carrier out-of-synch
  - If the demodulation is performed in the receiver, the carrier frequency offset is caused by that the Rx local oscillator has a slightly different nominal frequency from the remove Tx frequency
    - Crystal oscillators (XO) always have their natural frequencies different from the nominal values (in unit of *ppm*, i.e., parts-per-million or 10^{-6})
    - Temperature change will cause oscillator frequency change
      - To save cost, non-temperature compensated (TC) XO may be used
      - XO could be heated up by Tx amplifiers during transmission
    - Crystal will aging, i.e., the oscillating frequency will change with time, even it was calibrated at some point
    - *Carrier offset frequency is proportional to the carrier frequency*
    - Timing frequency offset is linearly related to the carrier frequency offset
    - This is most likely case and mainly considered in this presentation

fling@twinclouds.com

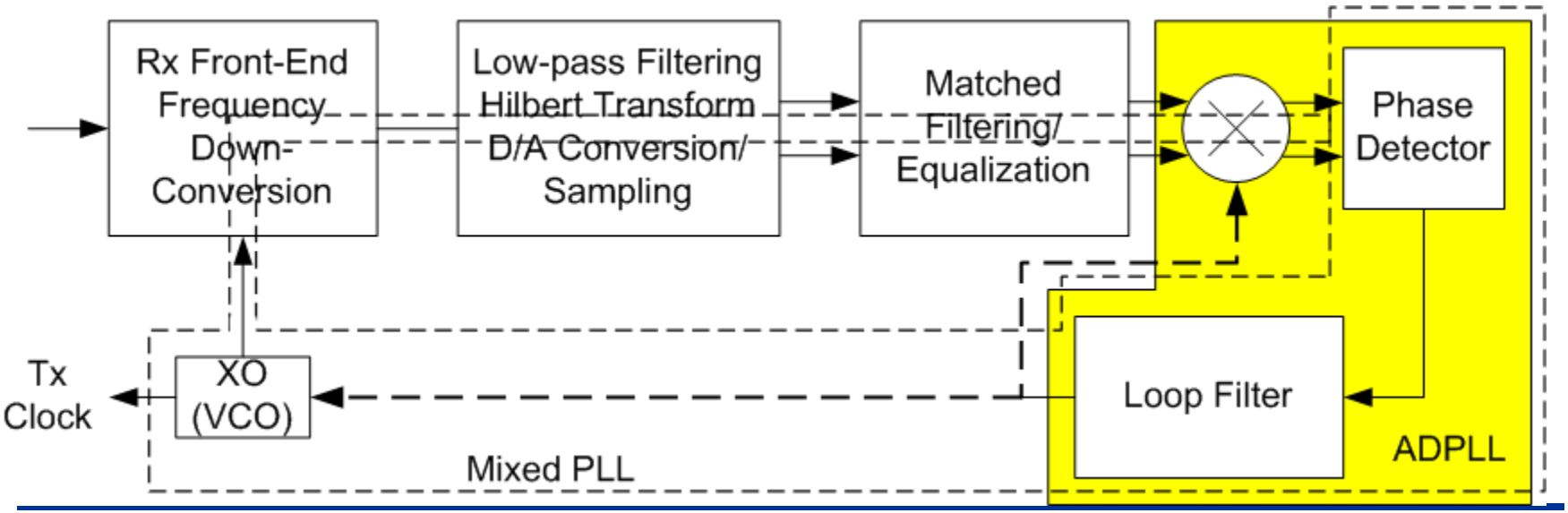
Sources of Carrier Out of Synch (cont.)

- Carrier frequency offset due to modulation/demodulation frequency mismatch (cont.)
  - Demodulation operation may be performed before the analog frontend of the receiver, e.g., in wireline transmission with carrier frequency translation
    - The frequency offset in carrier and timing are not linearly related
    - Timing frequency recovery need to be performed independently of the carrier frequency recovery
- Channel may also introduce frequency offset
  - Frequency introduced due to Doppler is random and frequency dependent so it cannot be compensated by carrier recovery
  - When at high vehicle speed and in line of sight conditions, e.g., on open space highway, we may see single tone Doppler that can be compensated by carrier recovery

fling@twinclouds.com

Carrier Synchronization Subsystem

- Typical carrier synchronization subsystem
  - It is essentially an overall phase-locked-loop (PLL),
    - containing: phase detector, loop filter and phase/frequency adjustment device
      - Phase/frequency adjust in analog domain – mixed PLL
      - Phase/frequency adjusted in digital domain – all digital PLL (ADPLL)



fling@twinclouds.com

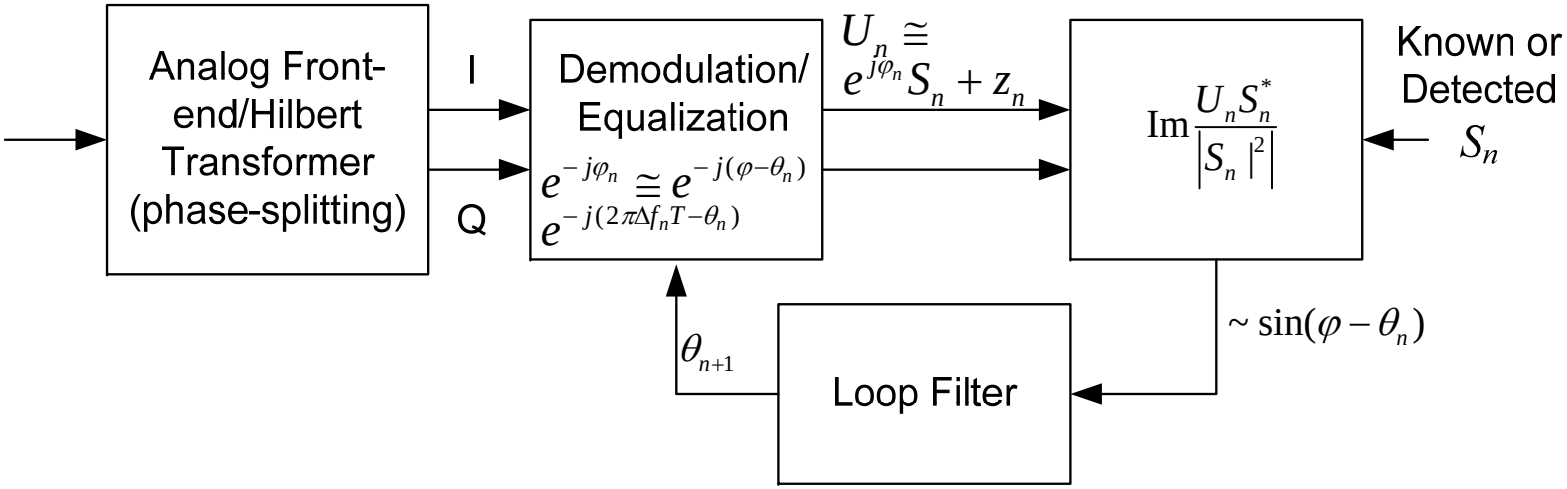
Wireline Modem Carrier Synchronization

- Characteristic and Assumptions of Wireline Modem Carrier Synchronization
  - Wireline communication are mainly over stationary channels
    - Frequency and phase offsets do not change rapidly
  - Training sequences are usually introduced at the beginning of transmission for initial training
    - Commonly no frequent insertion of training sequences
  - Due to the relative stable carrier characteristics, the carrier synch loop can (and should) have long time constant
  - In most modern designs data-aided carrier recovery is used for wireline modems.
    - Non-data-directed methods are mostly used in analog modems

fling@twinclouds.com

Wireline Modem Carrier Synch (cont.)

- Data-Aided Carrier Recovery



  - Data were converted to baseband before processing (carrier recovery using passband data can be done similarly)
  - Assuming sampling timing already (roughly) accurate
  - Assuming corresponding Tx data symbols are either known (training mode) or can be recovered at low error rate (data mode).

fling@twinclouds.com

Wireline Modem Carrier Synch (cont.)

- Data-aided carrier recovery
  - The carrier frequency and phase are first accurately estimated during initial training period with known training sequences before entering data mode
  - Timing synchronization has also been performed when entering data mode (to be discussed in timing-synch session)
    - Carrier and timing synchs mainly perform tracking in data mode
  - Impact of symbol detection error to carrier synch is minor if the error rate is low and the PLL time constant is long
    - It is desirable to use symbols from decoding (low error rate)
    - Decoding delay would only have minor impact to synch performance if the delay is short relative to the PLL's time constant

fling@twinclouds.com

Wireline Modem Carrier Synch (cont.)

- Data-aided carrier recovery (cont.)
  - During the high error period due to burst noise/interference, it may be desirable to freeze the equalizer and PLL update and waiting the data symbols can be recovered
  - For none phase-uniform constellations, e.g., QPSK, 16/64 QAM, it is possible to use constellation shape (corners) to perform carrier synchronization
    - Useful in the burst error recovery period
    - Useful for blind equalization receivers

fling@twinclouds.com

Wireline Modem Carrier Synch (cont.)

- Non-data-aided systems
  - Can be used for both digital and analog communication systems
  - It is possible due to the cyclo-stationary property of the data communication signals
  - For double-sideband signal, square law or other non-linear devices can be used to generate spectral lines at the twice of the carrier frequency ($2f_c$).
    - PLL will operate at $2f_c$ for error smoothing (averaging)
    - The output frequency is divided by 2 to generate the carrier (f_c)
  - For QAM/QPSK systems, a fourth power device is used to generate spectral lines at $4f_c$.
    - PLL will operate at $4f_c$
    - The output is divided by 4 in frequency to generate the carrier (f_c)

fling@twinclouds.com

Wireless Modem Carrier Synchronization

- Overview
  - Wireless communications are almost always over fading channels
    - Carrier phase of the received signal changes constantly relative to that of the transmitted signal
  - The Tx frequency of base-stations (BTS/Node-B) are very accurate
    - Good quality XOs are used and/or synchronized to GPS
    - Received carrier frequency has short term fluctuations due to channel fading but does not change on average
  - Carrier frequency is usually initialized during the modem initial acquisition stage
  - Short or continuous training sequences/symbols (pilots or references) are inserted into data bursts and/or data streams
    - To be used to constantly estimate the time varying channel phase
    - Also to be used for tracking average carrier frequency of the remotely transmitted signal

fling@twinclouds.com

Wireless Modem Carrier Synch. (cont.)

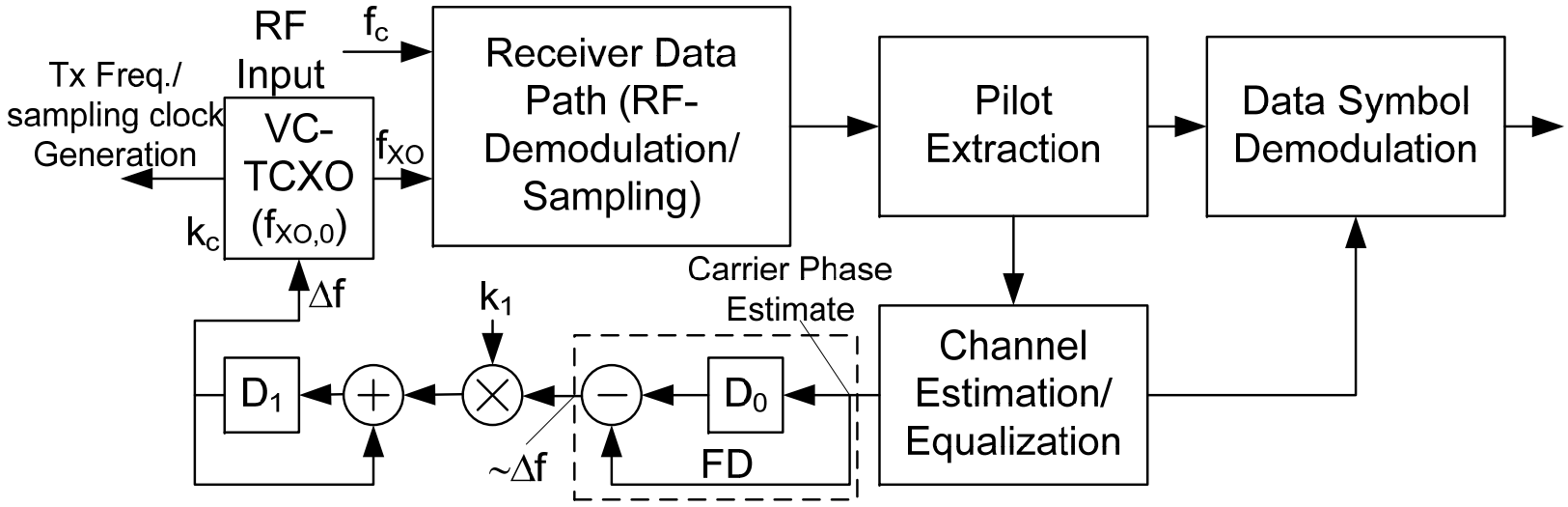
- Overview (cont.)
  - In most wireless modems, frequency and phase synchronization are performed separately:
    - Channel phase are estimated from data directly or channel estimates
    - First order feedback loop is used for tracking remote BTS Tx frequency
      - It is called frequency locked loop (FLL) or frequency tracking loop (FTL)
      - The differences between consecutive carrier phase estimates are used to drive FLL/FTL
        - » The phase difference can be estimated by *delay-and-correlation* method

        e.g., for $ch(n) = Ae^{j\theta_n}$, $(\theta_n - \theta_{n-1}) \approx \mathrm{Im}\left[ch(n) \times ch^*(n-1)\right] / \left|A\right|^2$
        - » The phase difference divided by the time lapse from one burst to next is an estimate of the frequency error
  - Once terminal carrier frequency synch is achieved (recovered), it will be approximated equal to the BTX Tx frequency on average
    - It will be quite accurate as locked to the accurate remove BTS carrier
    - Terminal Tx clock is usually derived from the recovered carrier frequency
    - Average carrier frequency is relatively stable

fling@twinclouds.com

Exemplary Wireless Modem Carrier Synchronization Realizations

- VC-TCXO Based Mixed Components Implementation



  - VC-TCXO: Voltage controlled-temperature compensated crystal oscillator

fling@twinclouds.com

Exemplary Wireless Modem Carrier Synch.

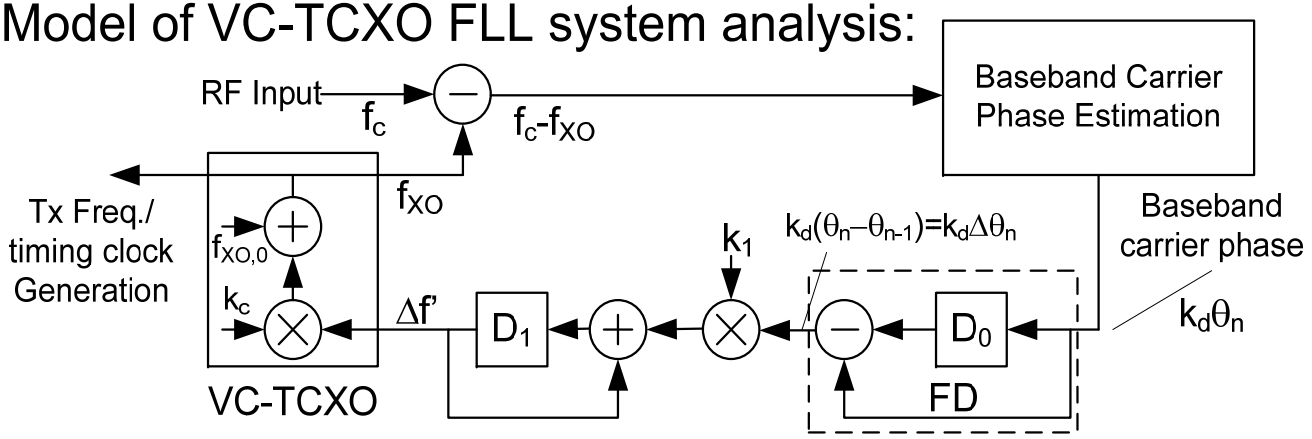
- VC-TCXO Based FLL Implementation (cont.)
  - Operations in frequency tracking mode:
    - The VC-TCXO has a initial frequency of $f_{XO,0}$. The baseband carrier frequency generated in the demodulation block is equal to $\Delta f = f_c - f_{XO,0}$.
    - The baseband carrier phase is estimated in the channel estimation block. The difference between the current and previous estimated carrier phase is computed in the FD block and proportional to Δf.
    - The difference is accumulated to generate a scaled estimate of Δf ($\Delta f'$).
    - $\Delta f'$ stored in D_1 is feed to VC-TCXO to change its generated frequency
    - The output of VC-TCXO is used for demodulation
    - Once converges, the FD block output is zero on average. The value of the frequency offset estimate in D_1 will be a constant. The VC-TCXO output is locked to the (accurate) remote BTS Tx frequency
    - FLL can and should have a long time constant
    - The VC-TCXO is used to synthesize the device Tx frequency and for the sampling clock (both are accurate)

fling@twinclouds.com

Exemplary Wireless Modem Carrier Synch.

- VC-TCXO Based FLL Implementation (cont.)
  - Linear system analysis and implementation considerations:
    - Denote baseband carrier phase estimate by $k_d\theta_n$ (θ_n in radius)
    - The output of FD is $k_d(\theta_n-\theta_{n-1})$. Note: $(\theta_n-\theta_{n-1}) \sim \Delta f \times 2\pi T_s$
      - Using channel estimate $ch(n) = Ae^{j\theta_n}$, $(\theta_n - \theta_{n-1}) \approx \mathrm{Im}\left[ch(n) \times ch^*(n-1)\right] / |A|^2$
    - Denote the gain of VC-TCXO by $k_c = \Delta f_{XO}/\Delta f'$.
    - The PLL analysis given above applies to FLL and its the loop gain k equal to $2\pi T_s k_d k_1 k_c$
      - Note: for scaling purposes, k_1 can be distributed in a few places in the loop

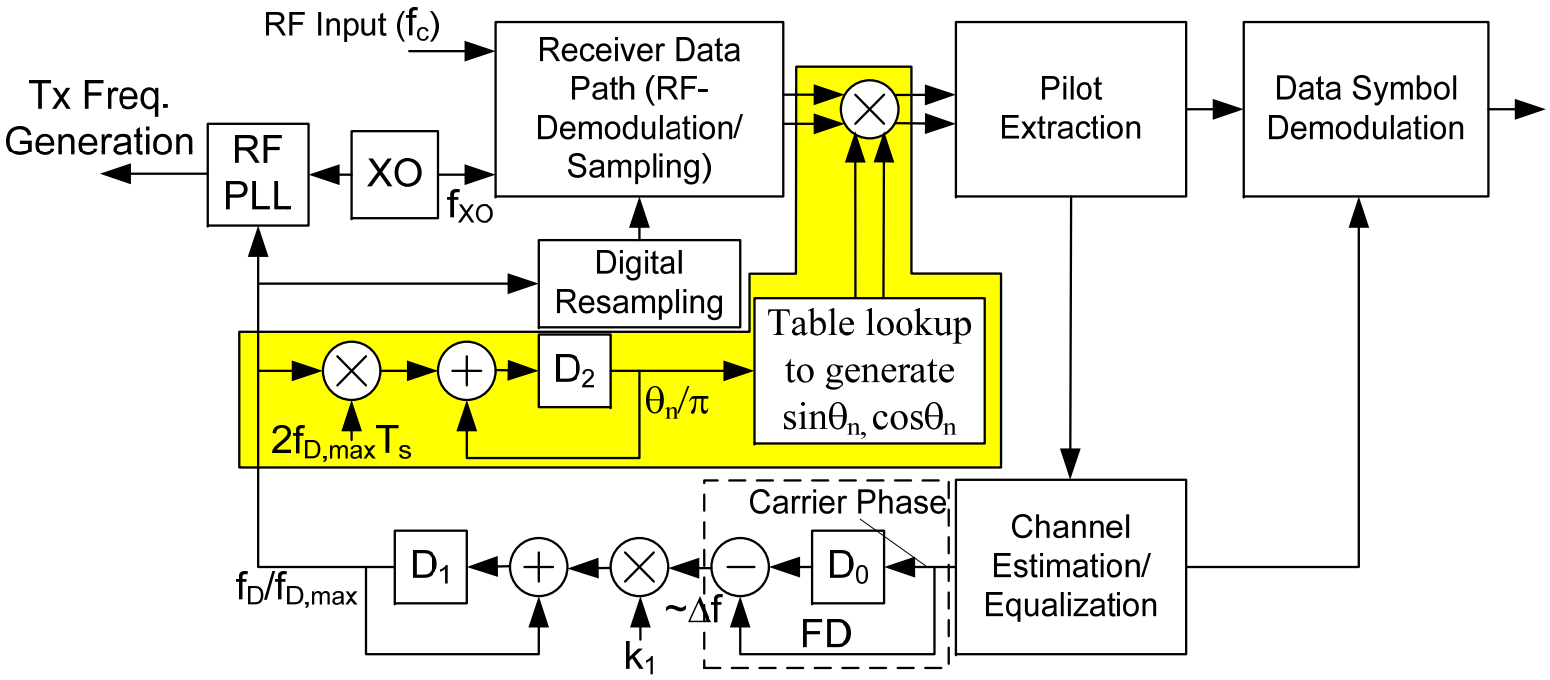
Model of VC-TCXO FLL system analysis:



fling@twinclouds.com

Exemplary Wireless Modem Carrier Synch.

- XO Based All-Digital FLL Implementation
  - Block Diagram



fling@twinclouds.com

Exemplary Wireless Modem Carrier Synch.

- XO Based All-Digital Implementation (cont.)
  - Operations in frequency tracking mode:
    - The XO has a fixed frequency f_{XO}. The baseband carrier frequency offset of the data samples is $f_{D,0} = f_c - f_{XO}$ at the modulator output.
    - The digital rotator rotates the sample phase by $-2\pi f_D(n)T_s$ at nT_s.
      - This is equivalent to make the frequency offset in the samples at the rotator output to be $f_{D,0} - f_D(n)$
    - The detection of phase differences between samples and their accumulation to generate scaled $f_D(n+1)$ is the same as in the VC-TCXO FLL case.
    - The scaled $f_D(n+1)$ is feed to the yellow block to generate phase for rotating the samples at $(n+1)T_s$.
      - This block is a "digital frequency corrector" (DFC) as described below
    - In steady state, the value stored in D_1 is a scaled value of $f_c - f_{XO}$
    - The estimated frequency offset value in D_1 is used to synthesize the local Tx clock and for sample timing generation

fling@twinclouds.com

Exemplary Wireless Modem Carrier Synch.

- Operation and Design of All-Digital Frequency Correction (ADFC) block
  - The frequency estimate at nT_s, $f_D(n)$ is stored in register D_1 as $d_1(n)$
    - The range of $d_1(n)$ is [–1, 1) corresponding to the values of $f_D(n)$ from $-f_{D,max}$ to $f_{D,max}$, i.e., $d_1(n) = f_D(n)/f_{D,max}$
  - The register D_2 contains value ($d_2(n)$) for correcting the carrier phase offset due to $f_c - f_{XO}$, from n to n+1.
    - The range of $d_2(n)$ is [–1, 1) corresponding to phase $\theta_n \in [-\pi, \pi)$
  - The value $d_2(n)$ generates $\sin\theta_n/\cos\theta_n$ by table lookup
    - There is an inherent operation of scaling by π
  - $d_2(n)$ decrements by an amount of $2T_s f_{D,max} d_1(n) = 2f_D(n)T_s$
  - Sample phase is decremented by $2\pi f_D(n)T_s$, from n to n-1
  - At steady state, $d_1(n) = (f_c - f_{XO})/ f_{D,max}$
    - Effect of $f_c - f_{XO}$ are entirely corrected
    - Frequency offset detector FD output will be zero on average

fling@twinclouds.com

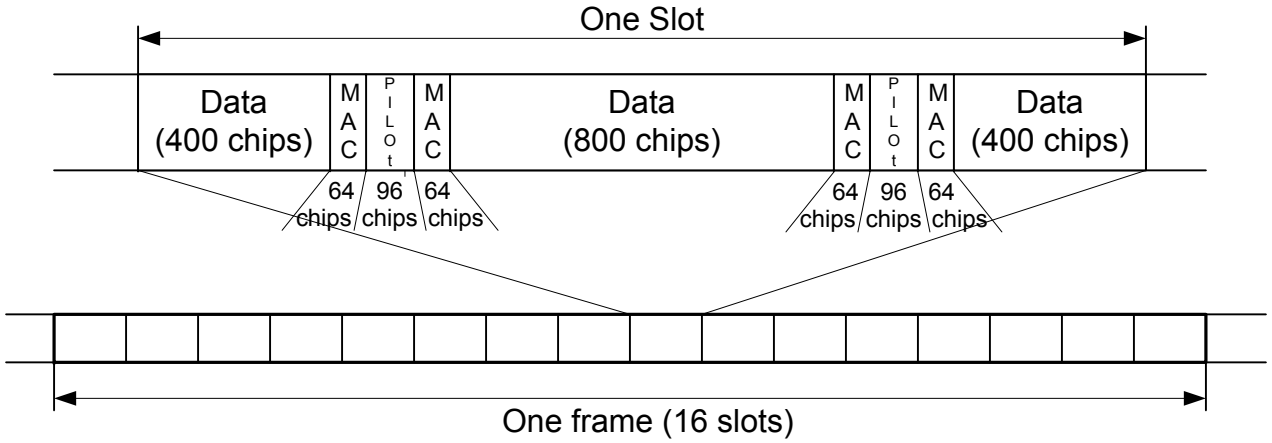
Exemplary Wireless Modem Carrier Synch.

- Comparison and Discussion
  - Advantages of VC-TCXO based design:
    - Conceptually simple in design: all clocks (system, Tx, sampling) are derived from VC-TCXO
    - VC-TCXO is usually quite accurate to make initial acquisition easier
  - Advantages of XO based all digital design:
    - Lower cost;
    - Reduce interactions between different modes in a multimode system
      - Solves the problem of "who's in control of VC-TCXO?"
  - For XO based FLL, the scaling between D_1 and D_2 need to be accurate to ensure accuracy of estimated frequency offset
    - Very important for generating accurate Tx and sampling clocks
  - Initialization and training
    - The value of D_1 can be initialized with available offset estimate, e.g. from initial acquisition
    - Larger k_1 can be used to speed up training
      - It is then reduced to normal value to ensure low steady state jitter

fling@twinclouds.com

Example: Carrier Synchronization in EV-DO

- Pilot Channel Structure cdma2000 EV-DO
  - Unlike cdma2000 1x and WCDMA, which have continuous pilot channels, EV-DO has a TDM pilot channel



- During initial acquisition, the receiver correlated the received signal with the known 96 chip pilot sequences
  - Once the pilot bursts are found, the receiver starts FLL initialization

fling@twinclouds.com

Example: Carrier Synchronization in EV-DO

- FLL Training state
  - Each burst has 96 chips. Each can be divided into 2 half bursts.
  - Each of the bursts generates a channel estimate
  - The two channel estimates are separated by 48 chips (≈ 39 μs)
  - The phase difference between the two estimates are used for initial training of FLL
    - The frequency offset can be determined without alias is 12800 Hz
  - Once the FLL is determined to converge to a smaller offset, the FLL enters the tracking state
- FLL Tracking state
  - The channel estimate from two adjacent pilot bursts can be used to compute the phase difference to drive the FLL
  - The separation between two pilot bursts is equal to 1/1200 sec.
    - The FLL has to converge to within 600 Hz before entering tracking state

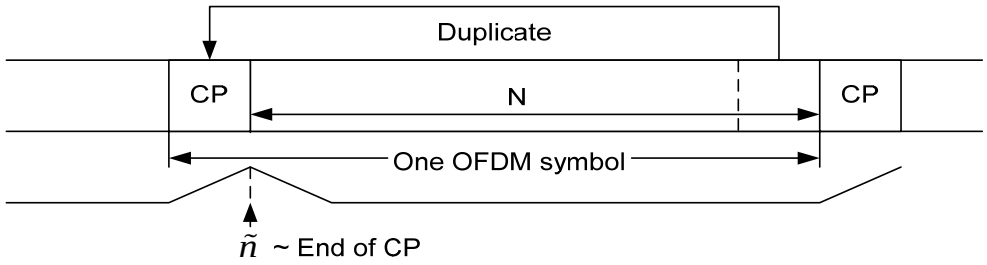
fling@twinclouds.com

Example: Carrier Synch. in OFDM Systems – Cyclic Prefix (CP) Based Method

- Delay-and-correlation using Cyclic Prefix (CP)
  - CP is a repetition of the last portion of the OFDM symbol
  - delay-and-correlate method:

$$D(n) = \sum_{k=0}^{N_{CP}-1} r^*(n+k) \times r(n+k+N), \quad N = \text{FFT size}$$

- Steps of Frequency offset Estimation
  - Determine symbol timing
  - Determine fractional F_{sb} carrier frequency offset ($|\Delta f_{fr}| < 0.5F_{sb}$)
  - Determine integer F_{sb} carrier frequency offset (mF_{sb}, $\Delta f = mF_{sb} + \Delta f_{fr}$)
- Determine OFDM symbol timing with: $D(\tilde{n}) = \max_n \left| D(n) \right|$



fling@twinclouds.com

CP Based OFDM Carrier Synch. (cont.)

- CP based Fractional F_{sb} carrier offset estimation

$$\text{Phase change in one OFDM symbol} \sim \text{Im}\left[D(\tilde{n})\right] \Big/ \left|D(\tilde{n})\right|$$

  - Phase difference (~frequency offset) is used to drive an FLL
  - FLL will converge to zero fraction F_{sb} (subcarrier spacing) but cannot determine offset with integer F_{sb} due to aliasing

$$r(n+k) \times r^*(n+k-N) \approx A_r^2 e^{2\pi j \Delta f / F_{sc}} = A_r^2 e^{2\pi j (mF_{sc} + \Delta f_{fr})/F_{sc}} = A_r^2 e^{2\pi j \Delta f_{fr}/F_{sc}}$$

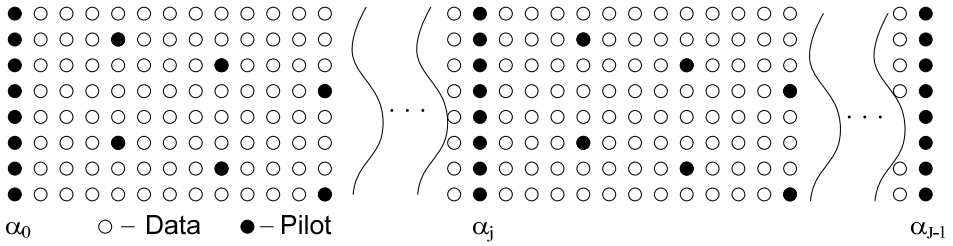
    - FLL will converge to $\Delta F = mF_{sc}$

- Integer F_{sb} carrier frequency offset determination
  - Initial frequency offset can be larger than F_{sc}
    - E.g., for LTE, at 2GHz and XO error ≤ ±10ppm, Δf can be up to ±20kHz > $F_{sc,\ LTE}$ = 15khz ~ ±7.5 kHz (m = 0, ±1)
  - With frequency offset of mF_{sb}, the indices of the demodulated OFDM symbol at the FFT output are shifted by m

fling@twinclouds.com

CP Based OFDM Carrier Synch. (cont.)

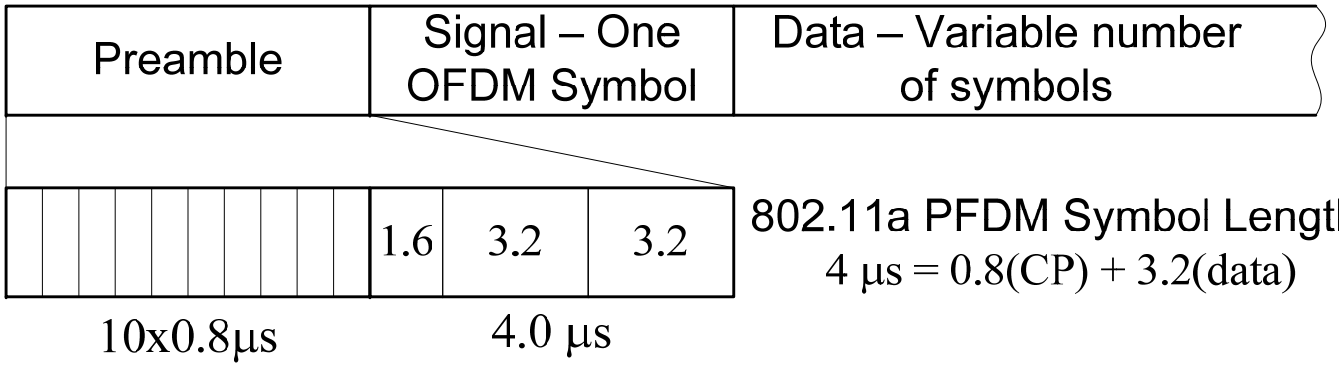
- Integer F_{sb} carrier frequency offset determination (cont.)
  - An example of based on known FDM Pilot: DVB-T
    - Take FFT of the samples with only integer F_{sc} frequency offset
    - Continuous pilots in DVB-T;
      - 2K and 8K FFT modes has 45 and 177 symbols, respectively per OFDM symbol
      - The continuous pilots are in the same subcarriers in each OFDM symbol
      - Their subcarrier indices are α_j, j = 0, …, J-1, J=45/177 for 2K/8K FFT modes
    - Compute $\sum_{j=0}^{J-1} P_{i,\alpha_j+m} P^*_{i+1,\alpha_j+m}, \quad m = 0, \pm 1, \pm 2, \ldots$
      - $P_{i,k}$ – The FFT Output of the k-th subcarrier of the i-th OFDM symbol
    - Pick the largest out of m
      - m is the integer offset
      - If k is a data carrier, the sum should be zero on average



fling@twinclouds.com

Example: Carrier Synch. in OFDM – Data Assisted

- Short time domain symbol based approach
  - Using shorter symbols, larger frequency offset can be estimated without alias
  - Example: 802.11a OFDM WIFI preamble design



  - Delay and correlation method for frequency offset estimation
    - Using CP of data symbol for correlation: maximum = ±156 kHz
    - Using short preamble for correlation: maximum = ±625 kHz
    - Frequency offset can be estimated in one shot
  - Any known time sequence can be used for Δf estimation by partition to multiple segments

fling@twinclouds.com

Data Assisted OFDM Carrier Synch. (cont.)

- Post FFT data-assisted carrier tracking
  - Using known OFDM modulation symbols – Pilots/Ref. Symbols
    - The product of subcarrier at FFT output multiplying by the conjugate of the modulation symbol generates the phase of this subcarrier
    - The phase difference $\Delta\phi$ of the same subcarrier in adjacent OFDM symbols is equal to $2\pi(\Delta f/F_{SC})(1+N_{CP}/N_{FFT})$
      - The maximum Δf (with no alias) is less than $0.5F_{SC}xN_{FFT}/(N_{FFT}+N_{CP})$
    - Pilots may not on the same subcarrier in adjacent OFDM symbols
      - The greater time gap, the smaller maximum Δf can be estimated
    - Any two known OFDM symbols on the same subcarrier and close in time can be used for carrier tracking, e.g. PSS, SSS and CRS in LTE
  - Using time domain channel estimates of adjacent OFDM symbols
    - The phase change $\Delta\phi$ of the same time tap in the channel estimates of adjacent OFDM symbols has the same relationship to Δf as above and has the same limitation of maximum Δf.
    - Does not require pilots on the same subcarrier in adjacent OFDM symbols

fling@twinclouds.com

What Have Been Discussed: PLL

- Phase Looked Loop (PLL) is an important component of carrier and timing synchronizations in receivers of digital communication systems
- First and second order PLLs are most widely used
  - The analyses are simple but important
  - Higher order PLLs are only used for special purposes
- The main components of PLLs are phase error detector, loop filter and phase correction controller
- The important parameters of PLLs are time constant, noise bandwidth, damping factor (for 2^{nd} order loop) and pull-in range
- Digital PLLs (DPLLs) are most widely used in modern digital communication receivers

fling@twinclouds.com

What Have Been Discussed: PLL (cont.)

- First and second order DPLLs are analyzed in detail
  - Loop gain determines the convergence time constant and noise bandwidth of first order PLL
  - The time constant and noise bandwidth is mainly determined by first order coefficient even in second order PLLs
    - The second order coefficient determines the damping factor
    - Approximate formulas are given for real time constant change while maintain the critical damping factor
- Implementation considerations are discussed
  - Digital and analog scaling determines loop characteristics
- Example of PLLs for frequency synthesis for transmitters for digital communication terminals was discussed

fling@twinclouds.com

What Have Been Discussed: Carrier Synch.

- Carrier synchronization includes phase and frequency synchronization
  - Essential to achieve best possible system performance
- The sources of carrier out-of-synch are the inaccuracy of local oscillator and channel Doppler fading
- Wireline channel is relatively stable – carrier phase and frequency offset can be viewed as constants
  - Second order PLL is usually used in modems to correct both
- Wireless channel has fast phase change but carrier frequency does not change rapidly on average
  - Frequency locked loop (FLL) is used to tracking the average offset between the receiver local oscillator and that of remote transmitter
  - Fast changing carrier phases are recovered by channel estimator and/or equalizer

fling@twinclouds.com

What Have Been Discussed: FLL

- FLL is a first order loop
  - Implementation is simple in principle
  - Frequency offset is computed by divide the carrier phase change between two sampling instants divided by the time between them
    - It is important to make sure the frequency offset detector can handle the maximum offset range, alias may exist and need to be handled properly
- Tx clock in wireless devices are derived from their Rx clock, which locks to the (accurate) remote BTS Tx clock
- We discussed two types of practical carrier synchronization subsystems: VC-TCXO and simple XO based designs
  - Details of design and implementation of FLL are presented
  - The design trade-offs are discussed
- FLL examples of single carrier and OFDM receivers are presented and discussed in detail

fling@twinclouds.com

THANK YOU

fling@twinclouds.com