ELEC 4190 – Digital Communications

Phase and Timing Recovery

Notes:

Outline

- Phase and timing recovery
- Phase-locked loop (PLL)
- Carrier phase estimation
- Symbol synchronization

- Recommended reading: Proakis and Salehi – Chapter 8
- Extra reading: Lathi and Ding – Chapter 4

Notes:

sBJksDRPrf96kfrB1XsIog

Outline

- Phase and timing recovery
- Phase-locked loop (PLL)
- Carrier phase estimation
- Symbol synchronization

Notes:

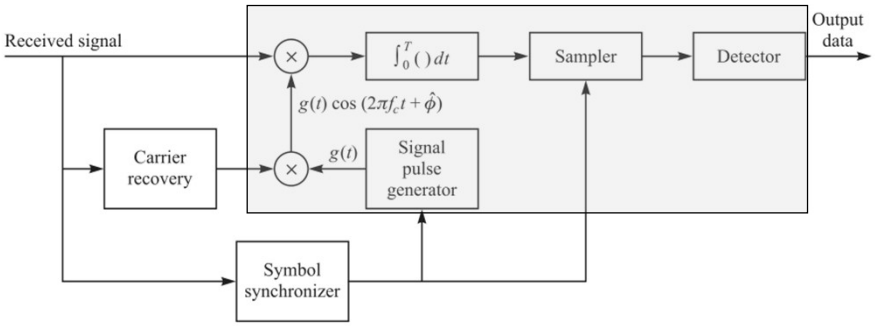
Phase and Timing Recovery

- Propagation delay from the transmitter to the receiver is generally unknown at the receiver
  - Symbol timing must be derived from the received signal in order to synchronously sample the output of the demodulator
  - Carrier offset must be estimated at the receiver if the detector is phase-coherent
- This can be done in two ways:
  - Data-directed: a periodic preamble is transmitted with information bits
    - Less complex but lowers the throughput and power efficiency
  - Non-data-directed: extract relevant information from the degraded data
    - Better throughput and power efficiency but more complex
    - We focus on this form: carrier phase recovery and symbol synchronization

Notes:

ASK/BPSK Receiver Structure with Phase and Timing Recovery

- Note: correlators can be replaced by matched filters

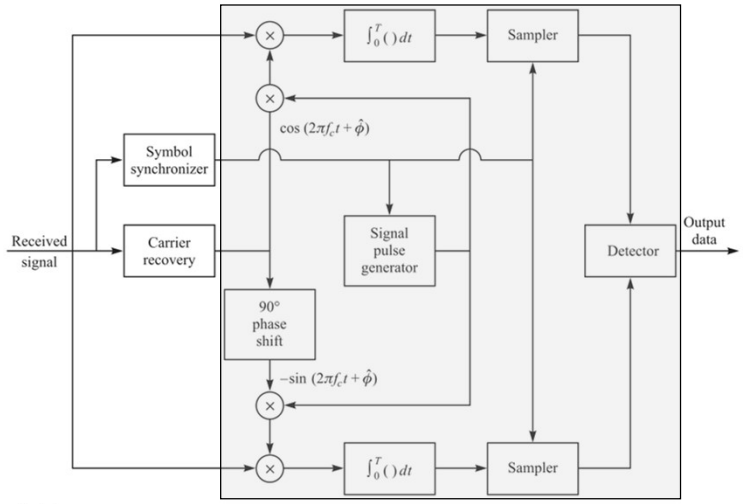


Notes:

sBJksDRPrf96kfrB1XsIog

QAM/PSK Receiver Structure with Phase and Timing Recovery

- Note: correlators can be replaced by matched filters



Notes:

sBJksDRPrf96kfrB1XsIog

Notes on Scrambling

- A long sequence of 1s or 0s may cause the bit synchronizer to lose synchronization momentarily
- Scrambling makes data more random by eliminating long sequences of 1s or 0s which is helpful in timing recovery
  - Scramblers are primarily used for preventing unauthorized access to the data
- A scrambler is used at the transmitter manipulates the input stream and a descrambler is used at the receiver to unscrambles data

Notes:

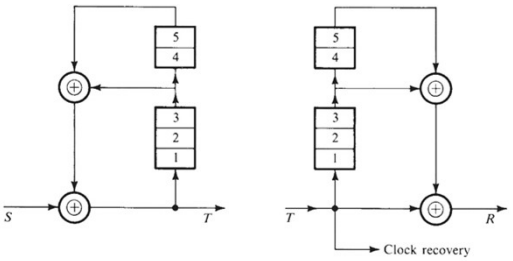
sBJksDRPrf96kfrB1XsIog

Example: Scrambler and Descrambler

- The scrambler consists of a feedback shift register, and the matching descrambler has a feedforward shift register
- Each stage in the shift register delays a bit by one unit

$$T = S \oplus D^3T \oplus D^5T$$

$$\begin{aligned} R &= T \oplus D^3T \oplus D^5T \\ &= S \oplus D^3T \oplus D^5T \oplus D^3T \oplus D^5T \\ &= S \end{aligned}$$



Notes:

sBJksDRPrf96kfrB1XsIog

Example

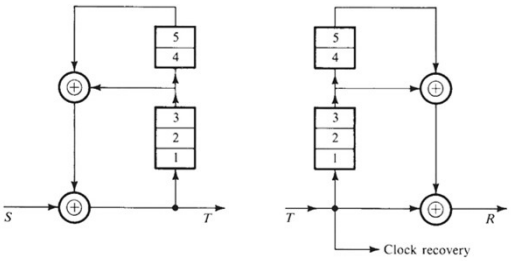
Find the scrambler output if data stream 1010000011 is fed to the input. Assume zero initial state. What happens to the sequence of 0s?

S	State					*T*
1	0	0	0	0	0	1
0	1	0	0	0	0	0
1	0	1	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	1	0	0
0	0	1	1	0	1	0
0	0	0	1	1	0	1
0	1	0	0	1	1	1
1	1	1	0	0	1	0

T	State					*R*
1	0	0	0	0	0	1
0	1	0	0	0	0	0
1	0	1	0	0	0	1
1	1	0	1	0	0	0
0	1	1	0	1	0	0
0	0	1	1	0	1	0
1	0	0	1	1	0	0
1	1	0	0	1	1	0
0	1	1	0	0	1	1

$$T = S \oplus D^3T \oplus D^5T$$

$$R = T \oplus D^3T \oplus D^5T$$



Notes:

sBJksDRPrf96kfrB1XsIog

Outline

- Phase and timing recovery
- Phase-locked loop (PLL)
- Carrier phase estimation
- Symbol synchronization

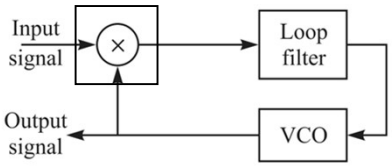
- Recommended reading: Proakis and Salehi – Chapter 8
- Extra reading: Lathi and Ding – Chapter 4

Notes:

sBJksDRPrf96kfrB1XsIog

Phase-Locked Loop (PLL)

- A very important device typically used to track the phase and the frequency of the carrier component of an incoming signal
- A PLL has three basic components:
  - A voltage-controlled oscillator (VCO)
    - An oscillator whose frequency can be linearly controlled by an input voltage
  - A multiplier, serving as a phase error detector (PD) or a phase comparator
  - A loop filter $G(f)$
    - Lowpass filter to remove high-frequency components
- The goal is to generate a VCO output whose:
  - frequency is the same as the frequency of the received signal and
  - phase is different from the phase of the received signal by 90 degrees



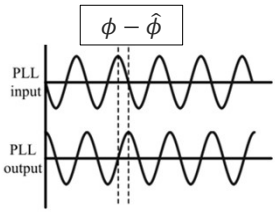
Notes:

sBJksDRPrf96kfrB1XsIog

PLL (cont.)

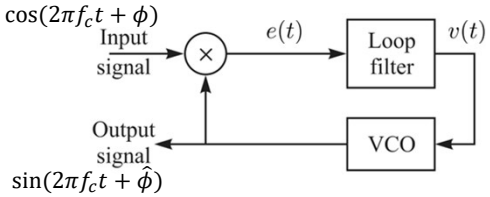
- Assume the input signal is $\cos(2\pi f_c t + \phi)$ and the output signal is $\sin(2\pi f_c t + \hat{\phi})$ where $\hat{\phi}$ represents the estimate of ϕ
- The input to the loop filter (output of PD) is

$$e(t) = \sin(2\pi f_c t + \hat{\phi})\cos(2\pi f_c + \phi)$$
$$= \frac{1}{2}\sin(\hat{\phi} - \phi) + \frac{1}{2}\sin(4\pi f_c t + \hat{\phi} + \phi)$$



- The loop filter will reject the second term at $2f_c$ and its output, which controls voltage for the VCO, is

$$v(t) = \frac{1}{2}\sin(\hat{\phi} - \phi)$$



Notes:

sBJksDRPrf96kfrB1XsIog

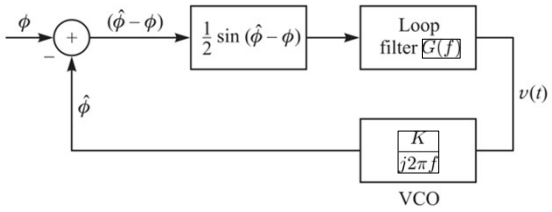
PLL (cont.)

- The output of the VCO is a sinusoidal signal generator with an instantaneous phase given by

VCO sensitivity

$$\sin(2\pi f_c t + \hat{\phi}) \quad \rightarrow \quad 2\pi f_c t + \hat{\phi} = 2\pi f_c t + K\int_{-\infty}^{t} v(t)dt$$

- The model below is an equivalent model to the PLL
  - In normal operation, the loop is tracking the phase and $\sin(\hat{\phi} - \phi) \approx \hat{\phi} - \phi$
  - In this case, model becomes linear with a closed-loop transfer function *H*(*f*)



$$H(s) = \frac{\hat{\Phi}(s)}{\Phi(s)} = \frac{0.5KG(s)}{s + 0.5KG(s)}$$

System is stable only when the real part of the poles is negative (i.e., left half of s-plane)

Notes:

sBJksDRPrf96kfrB1XsIog

Case Study: A General First-order Loop Filter

- Let's assume a general first-order filter such that

$$G(s) = \frac{1+\tau_2 s}{1+\tau_1 s} \quad \rightarrow \quad H(s) = \frac{1+\tau_2 s}{1+(\tau_2 + 2/K)s + 2\tau_1/K s^2}$$

- The equivalent noise bandwidth is

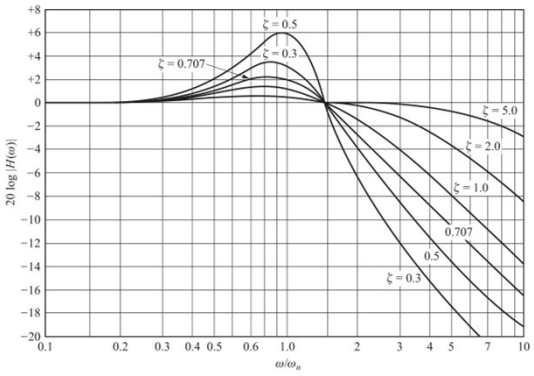
$$B_{\text{eq}} = \frac{1}{2|H(0)|^2}\int_{-\infty}^{+\infty}|H(f)|^2 \mathrm{d}f = \frac{1+(\tau_2\omega_n)^2}{8\zeta/\omega_n}$$

$\zeta = \omega_n(\tau_2 + 2/K)/2$ loop damping factor

$\omega_n = \sqrt{K/2\tau_1}$ loop natural frequency

The equivalent noise bandwidth is the bandwidth of a fictitious ideal LPF with the same area as $|H(f)|^2$

- Note that $\tau_1 >> \tau_2$



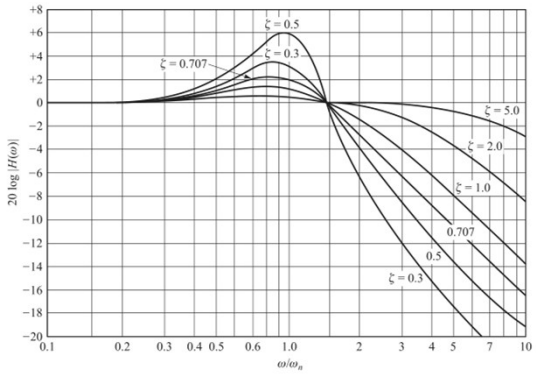
Notes:

sBJksDRPrf96kfrB1XsIog

Case Study: A General First-order Loop Filter (cont.)

- Design parameters control the loop bandwidth and involves a trade-off between speed of response and noise in phase estimate
  - Wide bandwidth tracks any time variations of the phase, but will allow more noise to pass into the loop

$$B_{\text{eq}} = \frac{1 + (\tau_2 \omega_n)^2}{8\zeta/\omega_n}$$



Notes:

sBJksDRPrf96kfrB1XsIog

Other Implementations of Loop Filter

- A simple gain that produces a first-order PLL

$$G(s) = K_1$$

- A first-order lowpass filter that produces a second-order PLL

$$G(s) = \frac{K_1}{1 + \tau_1 s}$$

- A proportional-plus-integrator filter that produces a second-order PLL

$$G(s) = K_p + \frac{K_i}{s}$$

Notes:

sBJksDRPrf96kfrB1XsIog

Outline

- Phase and timing recovery
- Phase-locked loop (PLL)
- Carrier phase estimation
- Symbol synchronization

Notes:

Carrier Phase Estimation

- Two basic approaches for dealing with carrier synchronization at the receiver:
  - Multiplex an unmodulated carrier component along with the information-bearing signal
    - The receiver uses a PLL to acquire and track the carrier component
    - Synchronize the local oscillator to the carrier frequency and phase
    - PLL should have a narrow bandwidth so that it is not significantly affected by the presence of frequency components from the information-bearing signal
  - Derive the carrier phase estimate directly from the modulated signal
    - More prevalent in practice
    - Total transmitter power is allocated to the transmission of the information-bearing signal
    - Now, we will confine our attention to the second approach

Notes:

sBJksDRPrf96kfrB1XsIog

Carrier Phase Estimation

- Assume the following received signal

$$\begin{aligned} r(t) &= s(t) + n(t) \\ &= A(t)\cos(2\pi f_c t + \phi) + n(t) \end{aligned}$$

  where $A(t)$ carries the digital information and has a zero mean
- Assume no frequency component at the carrier frequency
  - Have to extract carrier from the information-bearing signal itself

Notes:

sBJksDRPrf96kfrB1XsIog

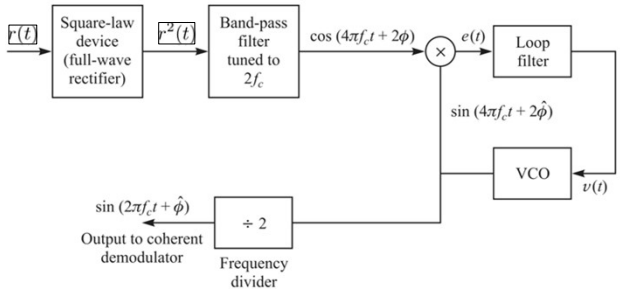
Squaring Loop

- A widely used method in practice to generate a carrier component from the received signal
  - First, square the received signal to generate a frequency component at $2f_c$
  - Then, use the component to drive a PLL tuned to $2f_c$

$$r^2(t) = \frac{1}{2}A^2(t) + \frac{1}{2}A^2(t)\cos(4\pi f_c t + 2\phi) + n'(t)$$

Non-zero power at $2f_c$

- Note:
  - The BPF is narrowband to isolate the desired double-frequency component
  - PLL operates in the same way discussed earlier



Notes:

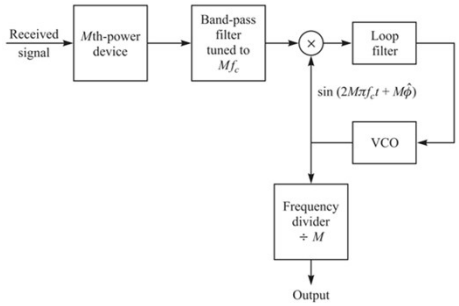
sBJksDRPrf96kfrB1XsIog

M-th Power Loop

- Used when the digital information is transmitted via *M*-phase modulation of a carrier
  - First, raise the received signal to *M*-th power to generate a frequency component at Mf_c
  - Then, use the component to drive a PLL tuned to Mf_c

$$r^M(t) = \frac{2}{2^M}A^M(t) + \frac{2}{2^M}A^M(t)\cos(2\pi M f_c t + M\phi) + n'(t)$$

Non-zero power at Mf_c



- Note:
  - The BPF is narrowband to isolate the desired frequency component
  - PLL operates in the same way discussed earlier

Notes:

sBJksDRPrf96kfrB1XsIog

Notes on Noise Enhancement

- The squaring (and *M*-th power) operation leads to increasing the noise power at the input to the PLL

$$r^2(t) = s^2(t) + \underbrace{2s(t)n(t) + n^2(t)}$$

Noise components

- Both components have spectral power in the frequency band centered at $2f_c$
  - So, the BPF also passes noise due to these two terms

Notes:

Costas Loop

- First, the received signal is multiplied in the in-phase and quadrature channels, then the signals are filtered out to remove higher frequency components

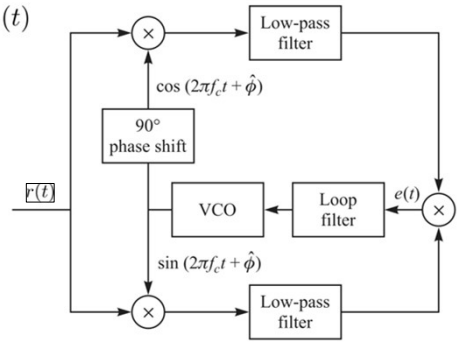
$$y_c(t) = \frac{1}{2}A(t)\cos(\hat{\phi} - \phi) + \frac{1}{2}A(t)\cos(4\pi f_c t + \hat{\phi} + \phi) + n_c(t)$$

$$y_s(t) = \frac{1}{2}A(t)\sin(\hat{\phi} - \phi) + \frac{1}{2}A(t)\sin(4\pi f_c t + \hat{\phi} + \phi) + n_s(t)$$

$$e(t) = \frac{1}{8}A^2(t)\sin(2(\hat{\phi} - \phi)) + n''(t)$$

$$v(t) = k\sin(2(\hat{\phi} - \phi)) + n''(t)$$

- The error signal then drives the VCO to minimize the phase error



Notes:

sBJksDRPrf96kfrB1XsIog

Outline

- Phase and timing recovery
- Phase-locked loop (PLL)
- Carrier phase estimation
- Symbol synchronization

Notes:

Symbol Synchronization

- In a digital communication system, the output of the demodulator must be sampled periodically at the symbol rate
- The sampling time instants should be $t_m = mT+\tau$, where T is the symbol interval and τ is the propagation delay
- Symbol synchronization (or time recovery) is the process of extracting the clock signal used for period sampling at the receiver
- Note that: the receiver must know not only the sampling frequency ($1/T$), but also where to take the samples within each symbol interval
  - The choice of sampling instant within the symbol interval of duration T is called the timing phase

Notes:

sBJksDRPrf96kfrB1XsIog

Symbol Synchronization (cont.)

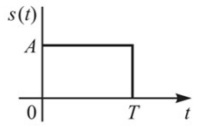
- Symbol synchronization can be accomplished in one of several ways
  - Transmitter and receiver clocks are synchronized to a master clock
    - Master clock provides a very precise timing signal
    - Receiver must estimate and compensate for the propagation delay
  - Transmitter simultaneously transmits the clock frequency $1/T$ (or a multiple of $1/T$) along with the information signal
    - Receiver employs a narrowband filter to extract the clock signal for sampling
    - Simple and reliable, but at the expense of wasting transmit power
  - Extracting clock signal from the received data signal (i.e., self-synchronization)
    - We will focus on this approach
    - We will discuss baseband signals and same approaches apply to bandpass signals

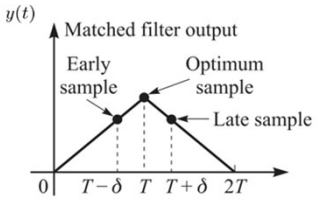
Notes:

sBJksDRPrf96kfrB1XsIog

Early-Late Gate Synchronizers

- Let's consider rectangular pulse below
- Output of the filter matched to $s(t)$ is symmetric and maximum at $t=T$
  - This is true for any pulse shape, so the argument is valid for other shapes
- The proper time for sampling is clearly at $t = T$ at the peak
  - It is difficult to identify the peak value of the signal in the presence of noise
- Suppose we sampled early at $t = T-\delta$ (or late at $t = T+\delta$)
  - The absolute values of the early samples $|y[m(T-\delta)]|$ (and late samples $|y[m(T+\delta)]|$) will be equal and smaller (on average in the presence of noise) than samples of the peak value $|y[mT]|$
  - So, proper sampling time is the midpoint



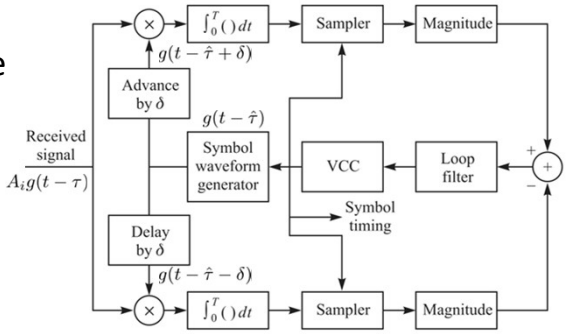


Notes:

sBJksDRPrf96kfrB1XsIog

Structure of Early-Late Gate Synchronizers

- The correlators are equivalent to matched filters
- The two correlators integrate over the symbol interval *T*, but
  - One correlator starts integrating δ seconds early relative to the estimated optimum sampling time
  - The other integrator starts integrating δ seconds late relative to the estimated optimum sampling time
- The error (i.e., difference between the absolute values of the two correlator outputs) passes through a LPF to smooth the noise
  - If the timing is off, the output of the LPF is nonzero



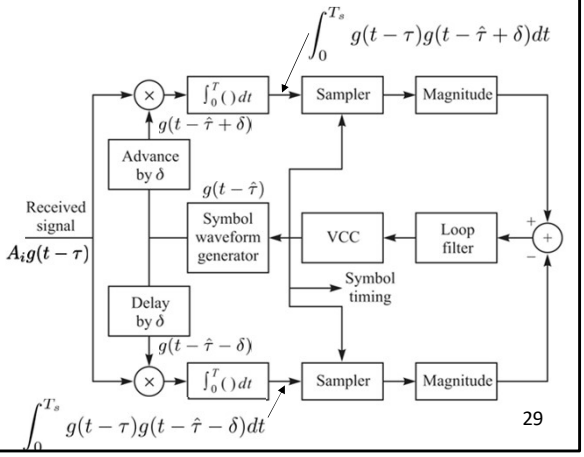
Notes:

sBJksDRPrf96kfrB1XsIog

Structure of Early-Late Gate Synchronizers (cont.)

- The smoothed error signal is used to drive a VCC that produces the desired clock signal for sampling and pulse waveform generation
  - Sign of error determines if the clock signal is retarded or advanced
    - $\hat{\tau} = \tau$: error is zeros → maintain
    - $\hat{\tau} > \tau$: error is negative → decrease
    - $\hat{\tau} < \tau$: error is positive → increase

- This pulse waveform is advanced and delayed and then fed to the two correlators



Notes:

sBJksDRPrf96kfrB1XsIog

Summary

- By now you should know:
  - How PLLs work
  - How to estimate carrier phase and sampling time

Notes:

sBJksDRPrf96kfrB1XsIog