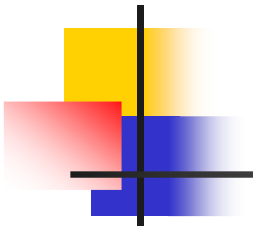
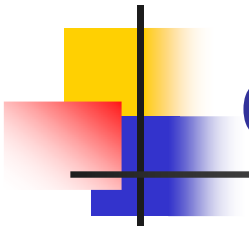


S-72.333 Postgraduate Course in Radio Communications
Fall 2004/Spring 2005

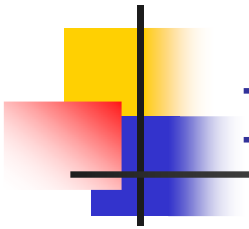
Timing and Carrier Recovery

Mervi Berner 11.01.2005
mervi.berner@iki.fi



Contents

- **Introduction - Receiver Loops: AGC, timing recovery, carrier recovery, channel equalization**
- **Timing recovery**
  - **Structure of a timing recovery loop**
  - **Early-late gate, Mueller and Muller, Gardner algorithms and other methods**
- **Carrier recovery**
  - **Phase locked loops:** Phase Detector and Loop Filter characteristics
  - **Costas loop**
  - **Carrier recovery algorithms**
- **Conclusion**



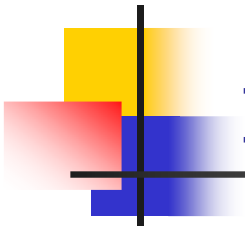
Introduction – Receiver Loops:

Stage 1 – Automatic Gain Control (AGC)

- scales the signal to a known power level
- typically handled in the analog domain (A/D converters have a limited dynamic range)
  - if received signal strength is too high, clipping.
  - if the signal strength is too low, distortion will occur because of severe quantization.

Stage 2 – Timing Recovery

- purpose of the timing recovery loop is to obtain symbol synchronization
- Two quantities must be determined by the receiver to achieve symbol synchronization:
  - sampling frequency - requires estimating the symbol period (samples can be taken at correct rate), quantity should be known (e.g. system's symbol rate 20MHz)
  - sampling phase - determining the correct time within a symbol period to take a sample, sampling the symbol at the center of the symbol period (peak) results in the best signal-to-noise-ratio and will ideally eliminate intersymbol interference.



Introduction – Receiver Loops (cont.):

Stage 3 – Carrier Recovery

At the transmitter:

- an oscillator generates a sinusoidal carrier signal that ideally exists at some known carrier frequency. The actual frequency will deviate slightly from the ideal value. This carrier is multiplied by the data to modulate the signal up to a passband center frequency.

At the receiver:

- the passband signal is multiplied by a sinusoid generated by the local oscillator.
- In practice, the frequencies of the local and transmitter oscillators differ and, instead of demodulation bringing the signal to baseband, the signal will be near baseband with some frequency offset.
- The presence of this frequency offset will cause the received signal constellation to rotate. This "spinning" effect must be removed before accurate symbol decision can be made.

⇒ The purpose of the carrier recovery loop is to *remove the frequency offset* so that the signal can be processed directly at baseband.

Stage 4 – Channel Equalization

- Transmitting a signal through a multipath channel results in a received signal that consists of several delayed and scaled versions of the transmitted signal. The multipath channel can be viewed as a linear filter.

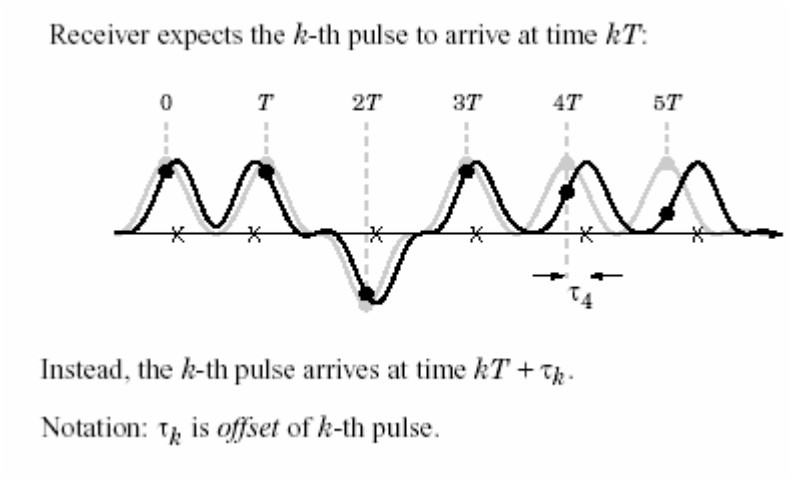
⇒ The equalizer is an adaptive filter that attempts to *remove intersymbol interference* by undoing the filtering effects of the multipath channel.

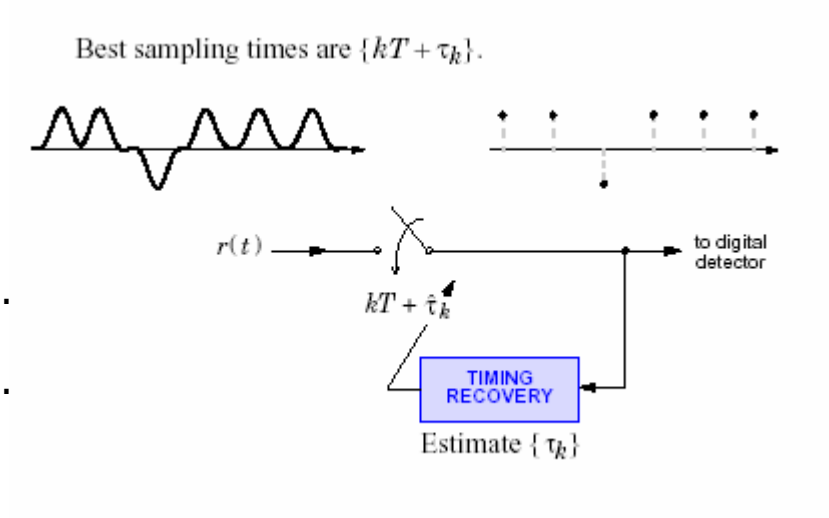
Timing Recovery Problem

The reality is that the receiver never knows the precise arrival times of the pulses. The receiver must know the sample frequency and where to take the samples within each symbol interval.

Timing recovery consists of:
• *Timing measurement* means the estimation of the timing error (s. algorithms).
• *Timing correction* can be performed in several ways:
o adjusting the timing phase of a voltage -controlled oscillator
o using an interpolation filter
o adjusting the receive filter

Sampling
• The optimal sampling times are $\{kT + \tau_k\}$.
• The job of a timing recovery scheme is to estimate the timing offsets before sampling.





Timing Recovery Loop (1)

- The interpolator is able to generate samples in between those actually sampled by the A/D (i.e., it interpolates). By generating these intermediate samples, the interpolator can adjust the effective sampling frequency and phase. The interpolated signal is a smoothed version of the original signal and it contains *N* times as many samples.
- Matched filtering technique is used to estimate the transmitted signal. The receiver filter's shape is "matched" to the transmitted signal's pulse shape. The matched filter's pulse shape is a time-reversed version of the transmit pulse shape. The output of the filter is then sampled at time *T*. Such processing has two advantages:
  - typical pulse shapes have a low-pass response. By filtering the received signal with such a filter at the receiver, the frequencies containing the data signal are passed while the remaining frequencies are attenuated. This matched filtering limits the amount of the noise spectrum that is passed on to subsequent stages in the receiver.
  - matched filter correlates the received signal with the transmit pulse shape over the symbol period *T*

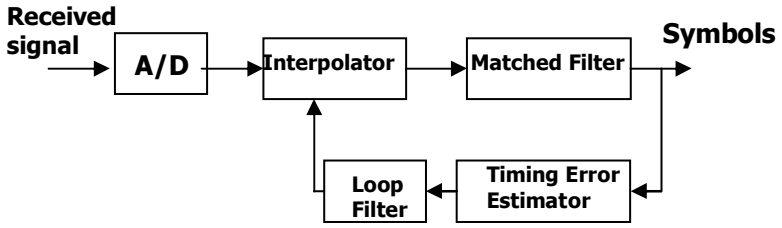


Fig. 1. Example of an all-digital timing recovery loop.

Timing Recovery Loop (2)

- Matched filtering provides the receiver with a stronger signal to work with (processing gain) compared to directly sampling the received signal. The received signal is severely distorted by noise, but the matched filter's output is still close to its ideal value for the case of no noise. This result is possible because the matched filter filters out the higher frequency noise and then integrates the remaining lower frequency noise over a time period of T. Because AWGN is zero-mean, integration effectively averages out the noise.
- It is important to sample the matched filter's output exactly at time T to obtain the sample with the highest SNR. Sampling the matched filter's output at some time $T + \Delta$ (where Δ represents a receiver timing offset) will significantly reduce the effective SNR seen by subsequent receiver blocks.

  $\Rightarrow$ the use of matched filtering gives the optimum performance in the presence of AWGN

  $\Rightarrow$ important to keep Δ as close to zero as possible and thus provides motivation for the inclusion of a timing recovery loop in the receiver
- The output of the matched filter is sent to a timing error estimator that can use a number of different algorithms to generate a timing error.

Timing Recovery Loop (3)

- The control signal for the interpolator is formed by filtering this error signal using a standard second-order loop filter containing a proportional and an integral section.
- The second-order loop filter consists of two paths:
  - proportional path:
    - multiplies the timing error signal by a proportional gain *Kp*
    - used to track out a phase error
  - integral path
    - multiplies the error signal by an integral gain *Ki* and then integrates the scaled error using an adder and a delay block.
    - used to track out a frequency error

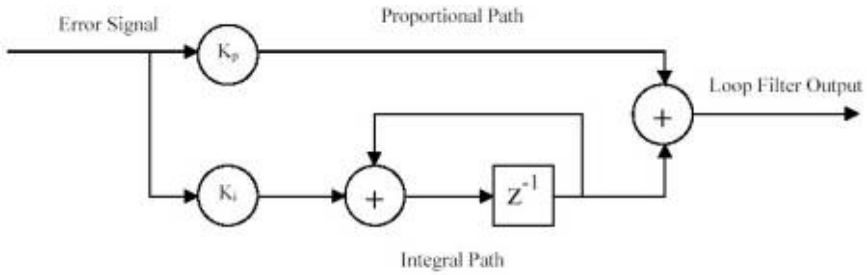


Fig. 2. Structure of a typical second-order loop filter.

Timing recovery: Early-late gate algorithm

- generates its error by using samples that are early and late compared to the ideal sampling point.
- The generation of the error requires at least three samples per symbol. Thus, it is impractical for systems with high data rates.

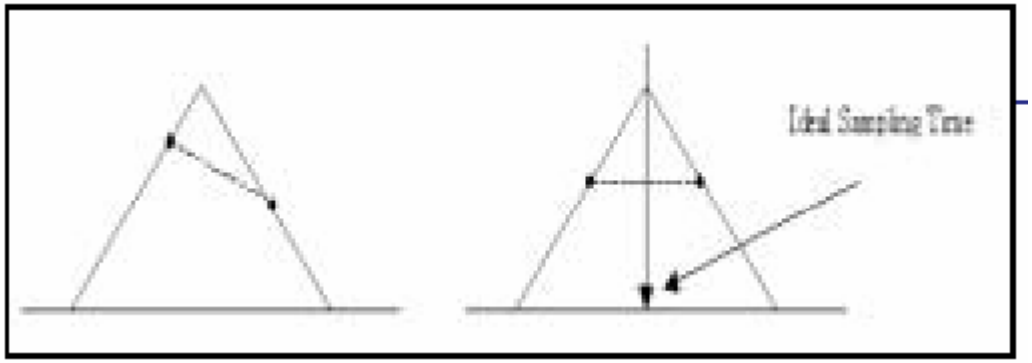


Figure 3. Method of generating error for early-late gate algorithm. The left plot shows where sampling is occurring too late. When sampling occurs at the right time, the early and late samples will be at the same amplitude.

Timing recovery: Mueller and Muller algorithm

- requires one sample per symbol. The error term is computed using the following equation: $e_n = (y_n \bullet y_{n-1}) - (y_n \bullet y_{n-1})$,

  where y_n is the sample from the current symbol and y_{n-1} is the sample from the previous symbol.
- sensitive to carrier offsets; carrier recovery must be performed prior to the Mueller and Muller timing recovery.
- examples of the value for the Mueller and Muller error for the cases of different timing offsets are shown in Figures 4.

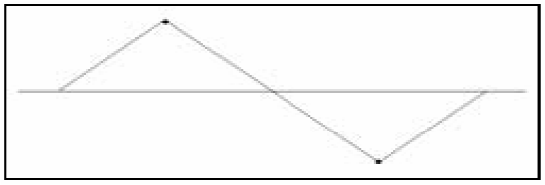


Figure 4.1. Correct timing: $e_n = (-1 \bullet 1) - (-1 \bullet 1) = 0$.

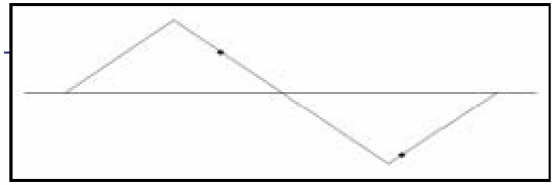


Figure 4.2. Timing is fast: $e_n = (-0.8 \bullet 1) - (-1 \bullet 0.5) = -0.3$.

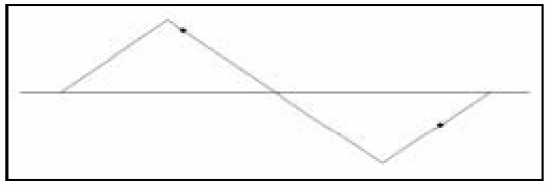


Figure 4.3. Timing is slow: $e_n = (-0.5 \bullet 1) - (-1 \bullet 0.8) = 0.3$.

Timing recovery: Gardner algorithm

- widespread use
- uses two samples per symbol
- insensitive to carrier offsets, the timing recovery loop can lock first, therefore simplifying the task of carrier recovery.
- error for the Gardner algorithm is computed using the following equation: $e_n = (y_n - y_{n-2})\, y_{n-1}$,where the spacing between y_n and y_{n-2} is T seconds, and the spacing between y_n and y_{n-1} is $T/2$ seconds.
- figures illustrate how the sign of the Gardner error can be used to determine whether the sampling is correct, late or early
- Gardner error is most useful on symbol transitions

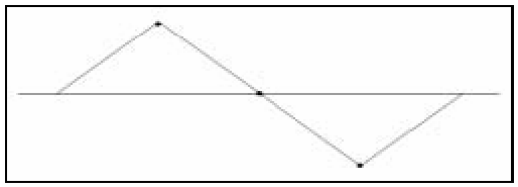


Figure 5.1. Correct timing: $e_n = (-1\ -1) \bullet 0 = 0$.

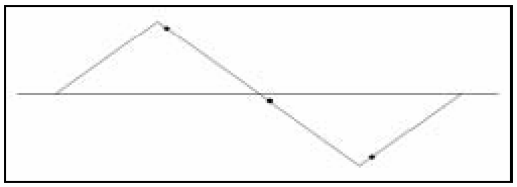


Figure 5.2. Timing is late: $e_n = (-0.8\ -0.8) \bullet (-0.2) = 0.32$.

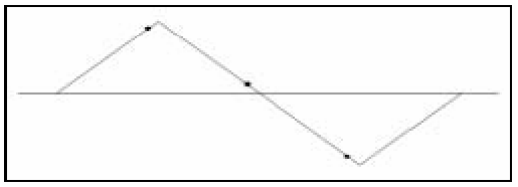
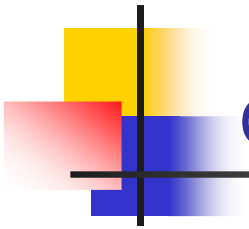
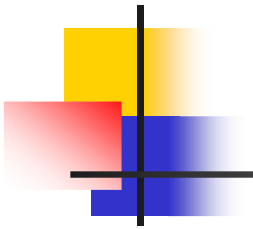


Figure 5.3. Timing is early: $e_n = (-0.8\ -0.8) \bullet (0.2) = -0.32$.



Other methods of timing recovery

- The ideal case: the transmitter and receiver running off of the same clock. This situation is typically impossible in a wireless communications system. In wired systems, such as computer networks, a timing recovery loop is not needed because synchronization is explicit.
- Alternative: the clock frequency transmitted along with the data. The receiver can recover this clock signal with a narrowband bandpass filter tuned to that frequency. This method is generally inefficient because the transmission of the clock signal consumes both bandwidth and transmitter power that could have otherwise been used for sending user data.
- Other algorithms exist for generating an error signal for a timing recovery.

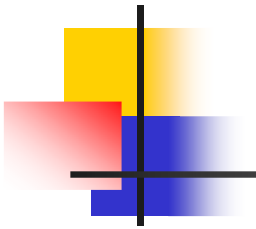


Carrier Recovery

Carrier recovery typically entails two subsequent steps:

- carrier synchronization parameters are estimated
  - *Carrier frequency offset* is mainly caused by two mechanisms – the frequency instability in either the transmitter or receiver oscillator, and the Doppler effect when the receiver is in motion relative to the transmitter.
  - *Carrier phase offset* is the result of three major components: the phase instability in oscillators, the phase due to transmission delay, and thermal noise (such as AWGN).
- the received carrier signal is corrected according to the estimates made

- Necessary for downconversion
- Compensates for clock mismatch and component drift
- Accomplished with a Phase Locked Loop (PLL). The carrier-recovery loop is a specific use of a PLL.
  - Needs training signal
  - Matches carrier frequency and phase
- Alternative: a feedforward (FF) digital carrier recovery technique.



Concepts of a simple Phase Locked Loop (PLL)

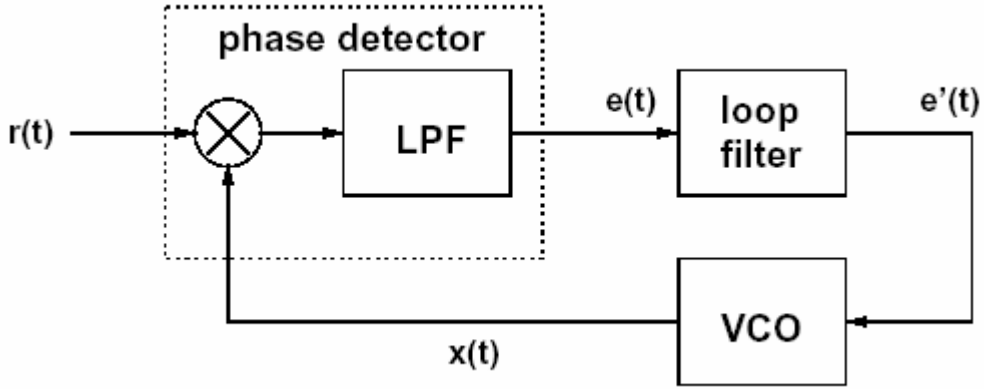
- PLL is a closed loop frequency controlled system in which functioning is based on the phase sensitive detection of phase difference between the input and output signals of a controlled oscillator.
- The term 'phase locking' means the task of aligning the output phase of the oscillator voltage with the phase of the reference voltage.
- Phase locking is achieved by
    1. Changing the frequency of the oscillator momentarily.
    2. A means of comparing the phases of the oscillator and reference signals.
- The unique property of a PLL is that during the phase locked condition the frequency of the input and the output signal are the same.

Phase Locked Loop

In Simple Terms: Phase locked loop is an oscillator whose frequency is locked onto some frequency component of an input signal, which is done with a feedback control loop.

Components of a PLL

- Phase Detector
- Voltage Controlled Oscillator (VCO)
- Low Pass Filter



Applications of PLL

- Carrier Recovery
- Clock Recovery
- Tracking Filter
- Frequency Demodulation
- Phase Demodulation
- Frequency Synthesis

Figure 6. Simple PLL. The goal of this PLL is to produce a locally generated sinusoid *x(t)* whose instantaneous angle is equal to the instantaneous angle of the received signal *r(t)*.

Carrier Recovery

Consider a received signal shown below, consisting of bursts of a sinusoid. When a burst occurs, the PD (Phase Detector) has a chance to compare the phase of output with the input. Any error produces a voltage V_d that is applied to the VCO to correct the phase.

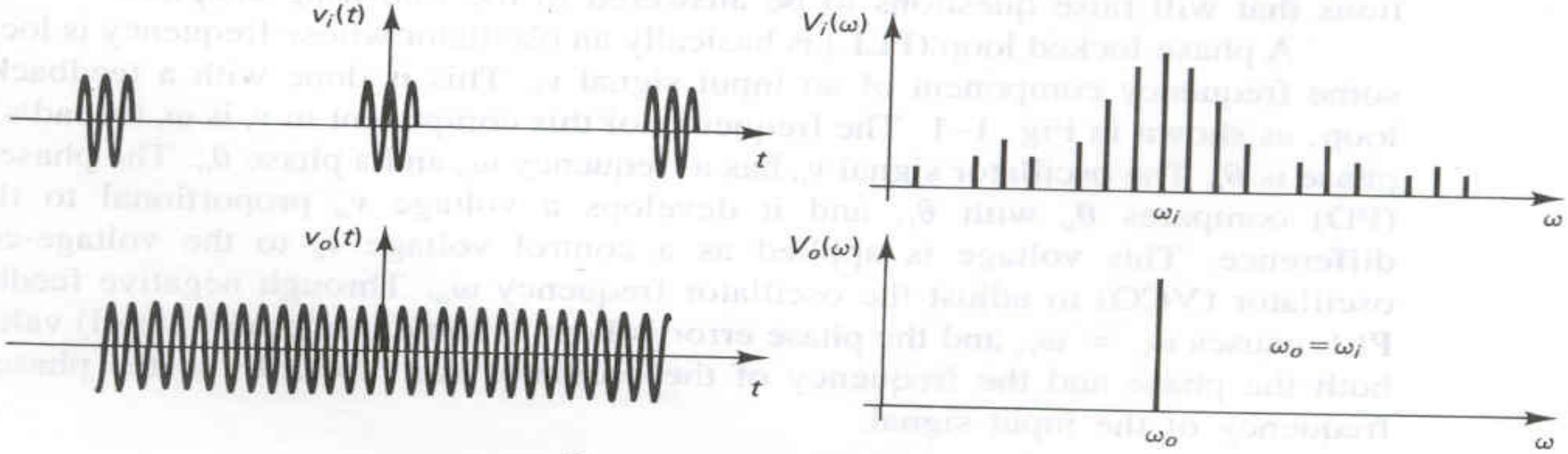
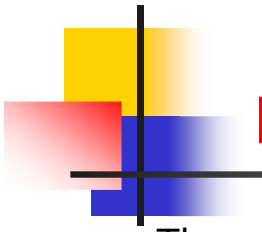


FIGURE 1–2 Carrier recovery

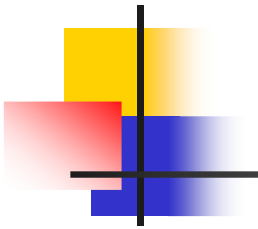


Phase Detector Characteristics

The main design choices to be made when implementing a PLL are the *type of phase detector* used and the *order and bandwidth of the loop filter*:

Phase detector (PD)

- Purpose: to generate an error signal that drives the PLL. It measures the difference between phase of the local oscillator and the input carrier. The error should be proportional to the phase difference between the signals *r(t)* and *x(t)*.
- Types of phase detectors:
  - ***Ideal PD*:** computes an error e(t) that is the difference between the instantaneous angles of *r(t)* and *x(t)*. It cannot be implemented in practice.
  - ***Sinusoidal PD*:** an approximation of the ideal phase error e*(t)* is formed by multiplying the input signal r*(t)* by the locally generated signal x*(t)* and passing the multiplier output through a lowpass filter. The multiplier output thus has two components, a high frequency term representing the sum of the angles and a low frequency term representing the difference of the angles. The purpose of the lowpass filter in the PD is to filter out the high frequency component. -> The lowpass-filtered output of the multiplier is the sine of the ideal phase detector's error.
  - ***Quadrature PD*:** A quadrature signal is a complex signal composed of two sinusoids at the same frequency but with a $\pi/2$ radian phase offset between them. The quadrature PD generates its error by multiplying the two signals together and taking the imaginary part of the output. -> The quadrature phase detector's output is the sine of the ideal phase detector's error.
- The type of phase detector used depends on the application, implementation complexity and the "quality" (how closely the PD's error approximates the error generated by an ideal phase detector) of the error generated.



Loop Filter Characteristics

Loop filter (LF)

- Purpose: to filter the phase error signal *e(t)* in order to provide a better signal to the VCO. The error signal generated by the PD is actually a noisy estimate of the phase error, i.e. the error signal consists of an error term and a noise term. The loop filter processes e*(t)* in order to generate a useful error while suppressing the effect of the noise as much as possible.
- An important design criterion for the loop filter is the *order of the filter*.
  A first-order loop filter can be used to completely track out a phase error. In order for the PLL to also track out a frequency offset, a PLL with a second-order loop filter is needed.
- The values of the gain parameters chosen for the loop filter control the *loop bandwidth* of the PLL. The size of the loop bandwidth determines the range of error signal frequencies that the loop filter will pass. The value for this bandwidth has a direct impact on the performance of the PLL.
  - If the value of the loop bandwidth is large, the loop filter can pass a wide range of frequencies for the error signal. A wide loop bandwidth will thus allow the PLL to track out large frequency errors. However, it will also pass a wider portion of the noise spectrum. The end result is a noisy control signal for the VCO and that translates into phase jitter on the locally generated signal *x(t)*.
  - A small value for the loop bandwidth will limit the amount of noise that passes through the filter. The narrow loop bandwidth will result in a cleaner control signal for the VCO. However, the drawback of using a narrow loop bandwidth is that the range of frequencies that the PLL can track out is reduced.

Costas Loop

- Training sequence, i.e., before the actual transmission of data, the transmitter sends a standard sequence of symbols that are known a prior by both the transmitter and the receiver. If a long enough training sequence is available, the receiver can lock onto the carrier during the training period. After the training period, the receiver has a good estimate of the carrier phase.
- If SNR is high, the probability of error is very small. Hence, the decisions made by the receiver are most likely to be correct. These "correctly decided symbols" are fed back to the PLL to continue to track the carrier phase.
- Differential encoding and decoding are needed.
- The Costas loop performs both phase-coherent suppressed carrier reconstruction and synchronous data detection within the loop:
- The upper loop is referred to as the quadrature,or tracking loop, and functions as a typical PLL, providing a data-corrupted error signal.
- The lower in-phase, or decisioning loop provides data extraction at the output of the lower mixer, and corrects the data corruption. The corrected error signal is applied through loop filter to the VCO, which yields a phase estimate. (The data signal is removed by multiplication before the loop filter.)

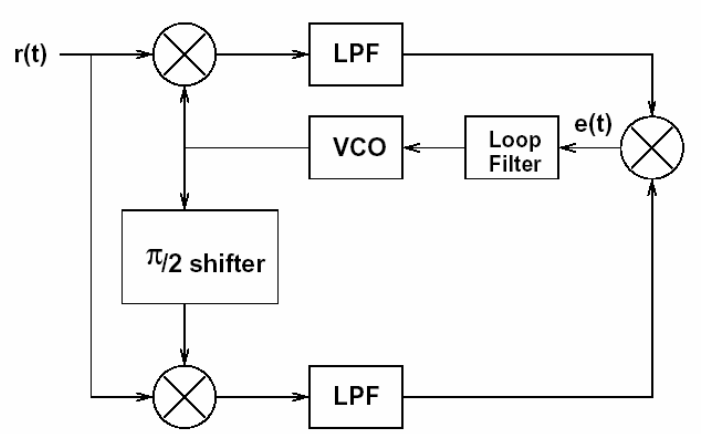


Figure 7: BPSK Costas loop

N-ary PSK Costas Loop

The basic Costas loop can be enhanced to perform carrier recovery and symbol detection for a N-ary PSK modulation scheme as shown in Fig. 8. Error signal to the loop filter is proportional to sinNϕ (ϕ is the phase error).

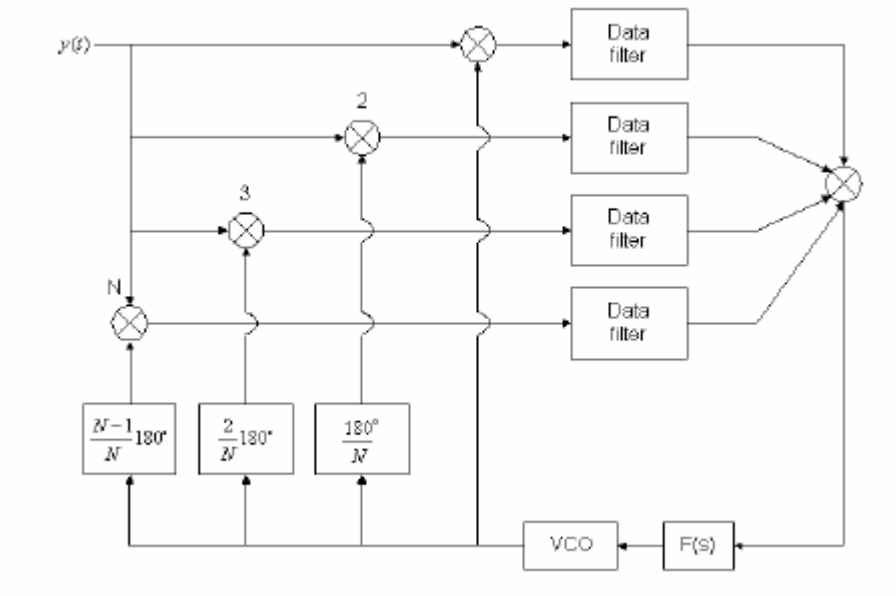
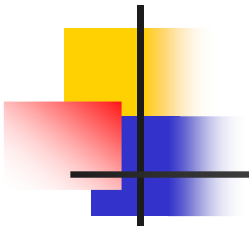


Figure 8: N-ary PSK Costas loop. The carrier phase is obtained and the symbols detected in one loop.



Digital phase-locked loop (DPLL)

- In an all-digital receiver a digital phase-locked loop (DPLL) is required.
- This DPLL employs a second order infinite impulse response (IIR) loop filter. The two filter coefficients ki and kp control the filter corner frequency and damping ratio.
- In the digital implementation, the VCO is replaced with a direct digital synthesizer (DDS).
- The phase detector is implemented using the arc-tan functioned unit.

Decision-directed carrier recovery loop

The carrier-recovery loop here uses a decision-directed phase detector.

- Many blocks in a receiver use decision-directed algorithms: the algorithm processes the current received symbol and makes a decision as to what it thinks the corresponding transmitted symbol was. The decision is typically made by using a decision device known as a slicer.
- The slicer makes a decision by quantizing the received sample to the nearest constellation point, and that quantized symbol is used as an estimate of the actual transmitted symbol. Noise and other impairments can cause the slicer to make incorrect decisions. Most decision-directed algorithms can tolerate a few decision errors, but when several errors occur in a short period of time, the algorithms will often diverge.

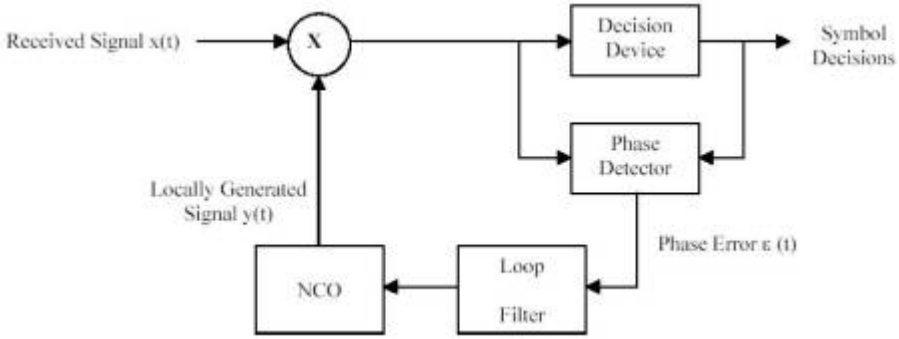


Figure 9: Architecture for a decision-directed carrier recovery loop.

Decision-directed carrier recovery loop (cont.)

Figure 10 shows the actual constellations before and after carrier recovery.
The received signal with a carrier offset has a rotating or spinning constellation. Once the carrier recovery loop has converged, this spinning effect is removed. At that point the slicer is able to make correct symbol decisions.
When operating a receiver over a channel that has multipath effects, an equalizer is often used at the receiver.The equalizer is a filter that removes the effects of a multipath channel. If a regular baseband equalizer is used, the equalizer is placed between the derotator and the slicer. Since the slicer is used to derive an error for carrier recovery, the equalizer is essentially included within the carrier recovery loop. The long delay of the equalizer filter requires lower values for the carrier recovery loop filter gains and thus reduces the tracking capabilities of the loop.

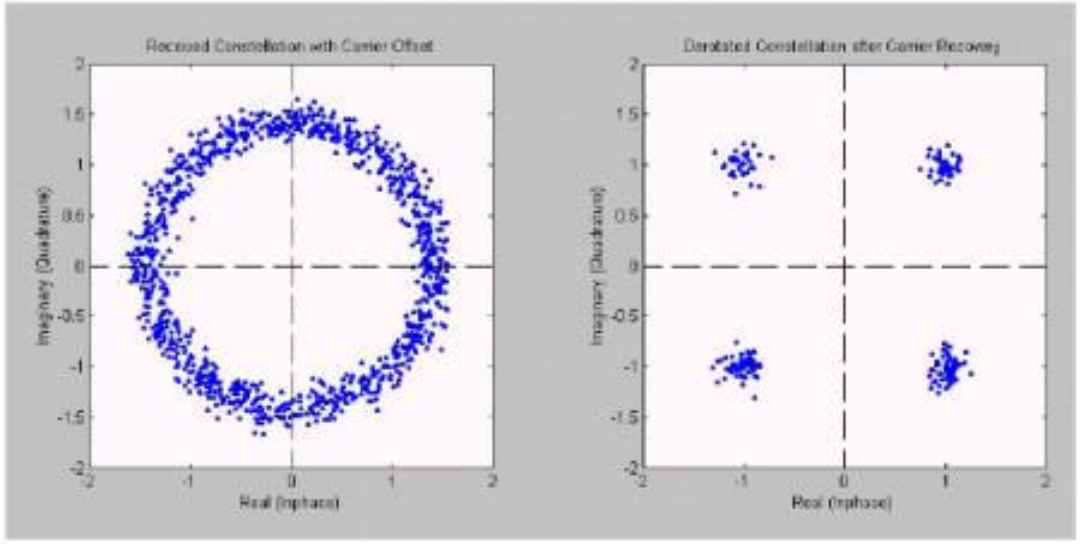
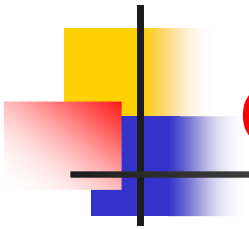
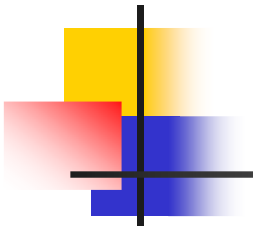


Figure 10: Simulation results for decision-directed carrier recovery loop. Plots shows QPSK constellation before (left) and after (right) carrier recovery.



Other Carrier Recovery Algorithms

- Other decision-directed algorithms
  - Example: uses the inverse tangent operation to find the difference between the received sample's phase and the phase of the slicer's output. Approximations can be used to avoid implementing an inverse tangent lookup table in hardware.
- Non-decision-directed techniques
  - Exist for use in situations where the signal-to-noise ratio (SNR) is so low that decision-directed techniques fail due to too many decision errors.
  - Example: power of N carrier recovery technique. This algorithm first raises the received signal to a power of N. The phase detector consists of a multiplication of this signal by the VCO output and then the imaginary part of the result is sent to the loop filter.



Conclusion

- The implementation of symbol timing synchronization is a vital process in obtaining the best SNR sampling point while also avoiding intersymbol interference.

- Even a small frequency or phase offset may highly impair the symbol detection: Symbols cannot not be correctly detected without proper compensation. For this reason, the receiver must be provided with accurate carrier frequency and phase estimates from the received signal itself by means of a carrier synchronizer, and then the received signal must be corrected according to the estimates made. Carrier frequency offset should be estimated and compensated prior to the phase offset compensation. This is because the initial carrier frequency offset can be large enough to cause phase rotation of a substantial part of a cycle in a single symbol and, as a result will prevent successful estimation of carrier phase offset.

- PLLs can be used to obtain timing and carrier synchronization.

- Only ’classical’ timing and carrier recovery schemes presented here, for further developments see e.g. reference [5] (+[1],[6]-[8] mentioned in the ref.)

References

[1] Louis Litwin, "Matched filtering and timing recovery in digital receivers", www.rfdesign.com, Sept. 2001

[2] Chris Dick Xilinx Inc., **Fred Harris** San Diego State University, **Michael Rice** Brigham Young University, **"Configurable logic for wireless communications: carrier and symbol synchronization", International IC – Korea Conference Proceedings, Sept. 2000**

[3] "Clock Recovery in Phase-Locked Loops", presentation by Avinash Balaji Praveen

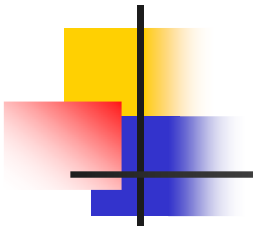
[4] "TAP Transceiver Design", Firmware Development and Hardware Integration, TAP Internal Review, Sept.2004

[5] John R. Barry, Aleksandar Kavicic, Steven W. McLaughlin, Aravind Nayak, and Wei Zeng, "Iterative Timing Recovery", IEEE Signal Processing Magazine, Jan.2004

[6] Louis Litwin, "Using PLLs to Obtain Carrier Synchronization: Part 1 and Part 2", http://www.commsdesign.com/, April 2003

[7] Jingxin Chen, "Carrier Recovery in burst-mode 16-QAM", Master of Science Thesis, University of Saskatchewan, June 2004

[8] Wong & Lok: *Theory of Digital Communications* " Ch. 3. Synchronization"



Homework

Timing Recovery

[1] a. Describe the Timing recovery problem.

b. What could be used instead of an interpolator?

c. What could be used instead of training signals?

Carrier Recovery

[2] a. Draw a QPSK Costas Loop.

b. Calculate the error generated by a quadrature phase detector.

(Hint: The error term for the ideal phase detector can be written as: $e(t) = \theta_x(t) - \theta_y(t)$. The quadrature input sinusoid can be written as: $x(t) = \cos(\theta_x(t))+j\sin(\theta_x(t))$ and the corresponding locally generated sinusoid can be written as: $y(t) = \cos(\theta_y(t))-j\sin(\theta_y(t))$. The quadrature phase detector generates its error by multiplying the two signals together and taking the imaginary part of the output.)