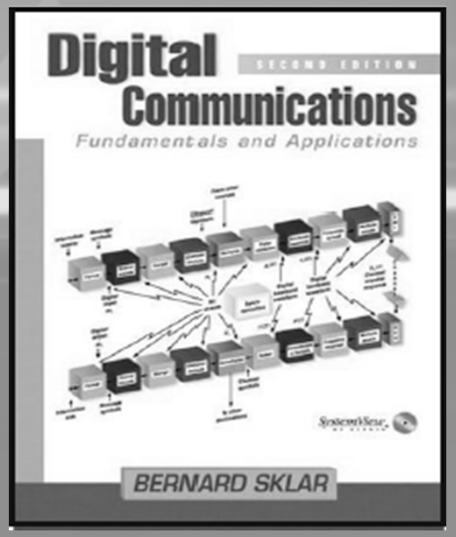


ENE 467
Digital Communications

TEACHING BY

ASST. PROF. SUWAT PATTARAMALAI, PH.D.

10. Synchronization

- Outcome
  - Understand the importance of Synchronization (Cost, Approach)
  - Know how to do Receiver Synchronization (Frequency, Phase, Steady state tracking, Non-linear loop analysis, Costas loop, etc.)
  - Know how to do Network Synchronization (Open loop, Close loop)

10.1.1 Synchronization Defined

In almost every discussion of receiver or demodulator performance, some level of signal synchronization is assumed, although this assumption is often not explicitly stated. For example, in the case of coherent phase demodulation (PSK), the receiver is assumed to be able to generate reference signals whose phases are identical (except perhaps for a constant offset) to those of the signaling alphabet at the transmitter. These reference signals are compared with the incoming signals in the process of making maximum-likelihood symbol decisions.

In order to be able to generate these reference signals, the receiver has to be in synchronization with the received carrier. This means that there has to be phase concurrence between the incoming carrier and a replica of it in the receiver. In other words, if there were no information modulated on the incoming carrier, the incoming carrier and the replica in the receiver would pass through zero simultaneously. This is what is known as being in *phase lock* and is a condition that must be closely approximated if coherently modulated signals are going to be accurately demodulated at the receiver. Being in phase lock means that the receiver's local oscillator is synchronized in both frequency and phase with the received signal. If the information-bearing signal is not modulated directly on the carrier but indirectly through the use of a subcarrier, both the phase of the carrier and that of the subcarrier must be determined. If the carrier and subcarrier are not kept in phase synchronism by the transmitter (they typically are not), this will require the generation of a replica of the subcarrier by the receiver, where the phase of the subcarrier replica is controlled separately from that of the carrier replica. This will enable the receiver to achieve phase lock on both the carrier and subcarrier.

It is also assumed that the receiver has accurate knowledge of when an incoming symbol started and when it is over. This knowledge is required in order to know the proper symbol integration interval—the interval over which energy is integrated prior to making symbol decisions. Clearly if the receiver integrates over an interval of an inappropriate length, or over an interval that spans two symbols, the ability to make accurate symbol decisions will be degraded.

It can be seen that symbol synchronization and phase synchronization are similar in that they both involve producing in the receiver a replica of a portion of the transmitted signal. For phase synchronization, it was an accurate replica of the carrier. For symbol synchronization, it is a square wave at the symbol transition rate. The receiver must, in effect, be able to produce a square wave that will transition through zero simultaneously with the incoming signal's transitions between symbols. A receiver that is able to do this can be said to have symbol synchronization, or to be in *symbol lock*. Since there are typically a very large number of carrier cycles per symbol period, this second level of synchronization is much coarser than phase synchronization and is usually done with different circuitry than that used for phase synchronization.

In many communication systems an even higher level of synchronization is required. This is usually called *frame synchronization.* Frame synchronization is required when the information is organized in blocks, or messages of some uniform number of symbols. This will occur, for example, if a block code is used for forward error control, or if the communications channel is being time-shared, on a regular basis, by several users (TDMA). In the case of block coding, the decoder needs to know the location of boundaries between code words in order to decode the message correctly. In the case of a time-shared channel, it is necessary to know where the location of boundaries between channel users are, in order to route the information appropriately. Similar to symbol synchronization, frame synchronization is equivalent to being able to generate a square wave at the frame rate, with the zero crossings coincident with the transitions from one frame to the next.

Most digital communications systems using coherent modulation require all three levels of synchronization: phase, symbol, and frame. Systems using noncoherent modulation techniques will typically require symbol and frame synchronization, but since the modulation is not coherent, accurate phase lock is not required. Instead, noncoherent systems require *frequency synchronization.* Frequency synchronization differs from phase synchronization in that the replica of the carrier that is generated by the receiver is allowed to have an arbitrary constant phase offset from the received carrier. Receiver designs can be simplified by removing the requirement to determine the exact value of the incoming carrier phase. Unfortunately, as is shown in the discussion of modulation techniques, this simplification carries a penalty in terms of degraded performance versus signal-to-noise ratio. The relative trade-offs of synchronization levels versus performance and system versatility are discussed further in the next section.

All of the discussion thus far had been oriented toward the receiving end of a communication link. There are instances, however, when the transmitter assumes the more active role in synchronization—by varying the timing and frequency of its transmissions to correspond to the expectations of the receiver. An example of this situation is a satellite communication network, where many terrestrial terminals are beaming signals toward a single satellite receiver. In most of these cases the transmitter relies on a return path from the receiver to determine the accuracy of its synchronization. Thus, transmitter synchronization often implies two-way communications or a network in order to be successful. Thus, transmitter synchronization is often called *network synchronization.* Transmitter or network synchronization is discussed later in this chapter.

Synchronization

10.1.2 Costs versus Benefits

There is a cost associated with the need for receiver synchronization. Each additional level of synchronization implies more cost. The most obvious cost is in the need for additional hardware or software in the receiver for acquisition and tracking. Possibly less obvious costs lie in the extra time required to achieve synchronization before commencing communications, or in the energy expended by the transmitter on signals to be used at the receiver as acquisition or tracking aids. In the face of these costs to the system, one might question why a communications system designer would consider a system design requiring a high degree of synchronization. The answer: improved performance and versatility.

Consider a standard commercial analog AM radio. This radio may be considered part of a broadcast communication system involving a central transmitter and many receivers. This communication system involves no synchronization. However, the receiver passband must be wide enough to accommodate not only the information-bearing signal, but also any fluctuations in the carrier, due perhaps to Doppler shift* or drift in the transmitter's frequency reference. This requirement in the receiver passband means that additional noise energy is passed to the detector, over and above the amount theoretically required by the bandwidth of the information. A somewhat more complicated receiver that employs a carrier frequency tracking loop would be able to keep a narrow passband filter centered about the carrier, thereby substantially reducing the detected noise energy and improving the received signal-to-noise ratio. Thus, although a standard radio may be perfectly adequate for reception of signals from large transmitters a few tens of kilometers distant, it may prove totally inadequate under less benign conditions.

For digital communications, examples of the trade-off between performance and receiver complexity are often seen in the choice of modulation. Among the simplest digital receivers are those designed to be used with noncoherently detected binary FSK. The only synchronization requirements are bit timing and frequency tracking. however, the same bit error probability could be achieved with approximately 4 dB less signal-to-noise ratio if the modulation is coherent BPSK. The disadvantage of BPSK is that the receiver requires accurate phase tracking. which can present a complex design problem if the signals experience high Doppler rates** or fading. (See Chapter 15.)

A third cost-versus-performance trade-off involves the use of error-control coding. As was established in earlier chapters, there are substantial performance advantages in the use of appropriate error-control coding techniques. The cost, however, measured in receiver complexity, can be high. For a block decoder to operate properly requires the receiver to achieve block, message, or frame synchronism. This is a procedure over and above the usual decoding procedure, although some error-correcting codes have been designed with block synchronization aids built in [1]. Convolutional codes also require some degree of additional synchronization in order to provide optimum performance. Although the performance analysis of convolutional codes often makes the assumption that the input data sequence is infinitely long, in practice it is not. In order to provide the minimum error probability, the decoder must know the beginning state (usually all zeros) when the data sequence will begin, the eventual ending state, and when the ending state is to be reached. Knowing when the beginning state was left and when the ending state is to be reached, however, is equivalent to having frame synchronization. In addition, the decoder will have to know how to group the channel symbols in order to make branch decisions. This is also a synchronization requirement.

The trade-offs discussed thus far have been in terms of the performance versus complexity of individual links and receivers. The ability to synchronize has a large potential consequence in terms of system efficiency and versatility as well. Frame synchronization allows the use of advanced, versatile, multiple-access techniques, such as the variety of demand-assignment-multiple-access (DAMA) schemes, which have become increasingly popular as communication channel resources become increasingly scarce. In addition, the use of spread-spectrum techniques—both as multiple access schemes and for interference rejection—requires a high level of system synchronization. (Spread-spectrum techniques are treated in Chapter 12.) It will be seen that these techniques provide the potential for a great deal of system versatility, which is a very valuable feature if the system encounters changing or unstable conditions, such as the effects of intentional and unintentional interference from external sources.

Synchronization

10.1.3 Approach and Assumptions

There have been at least two substantial developments in the general area of synchronization since the first edition of this text. One has been the emergence, and then near total dominance, of signal processing (including synchronizers) by sampled data techniques. The other has been the publication of several book-length treatments of synchronization [2–4]. This single chapter will not attempt the treatment of a full-length text. The goal here is to provide a broad intuitive understanding of the issues, rather than attempt to describe a catalog of synchronizer-design methods. Thus, we will generally follow a traditional analog development knowing that these principles apply equally well to sampled-data systems, even if the implementation of the synchronizers differ. Phase-locked-loops are commercially available as relatively small gate-count chips, or as one part of a larger signal-processing device. It is assumed that the reader interested in modern design implementations will be able to make the reasonable straightforward transition from the principles presented here to the basic sampled data representations.

10.2 RECEIVER SYNCHRONIZATION

All digital communication systems require some degree of synchronization to incoming signals by the receivers. In this section the fundamentals of the various levels of receiver synchronization are discussed. The discussion begins with the basic levels of synchronization required for coherent reception—frequency and phase synchronization—and a brief discussion of the principles of phase-locked-loop (PLL) operation and design. The discussion then broadens into the topic of symbol synchronization. Some degree of symbol synchronization is required for all digital communications reception, either coherent or noncoherent. The final topics in the section are receiver frame synchronization and techniques for achieving and maintaining it.

10.2.1 Frequency and Phase Synchronization

At the heart of nearly all synchronization circuits is some version of a phase-locked-loop (PLL). In modern digital receivers this loop may be difficult to recognize, but the functional equivalent is essentially always present. A schematic diagram of the basic PLL is given in Figure 10.1. Phase-locked loops are servo-control loops, whose controlled parameter is the phase of a locally generated replica of the incoming carrier signal. Phase-locked loops have three basic components: a phase detector, a loop filter, and a voltage-controlled oscillator (VCO). The phase detector is a device that produces a measure of the difference in phase between an incoming signal and the local replica. As the incoming signal and the local replica change with respect to each other, the phase difference (or phase error) becomes a time-varying signal into the loop filter. The loop filter governs the PLL's response to these variations in the error signal. A well-designed loop should be able to track changes in the incoming signal's phase but not be overly responsive to receiver noise. The VCO is the device that produces the carrier replica. The VCO, as the name implies, is a sinusoidal oscillator whose frequency is controlled by a voltage level at the device input. In Figure 10.1, the phase detector is shown as a multiplier, the loop filter is described by its impulse response function $f(t)$, with Fourier transform $F(\omega)$, and the VCO is so indicated.

A VCO is an oscillator whose output frequency is a linear function of its input voltage over some range of input and output. A positive input voltage will cause the VCO output frequency to be greater than its uncontrolled value, ω_0, while a negative voltage will cause it to be less. Phase lock is achieved by feeding a filtered version of the phase difference (i.e., the phase error) between the incoming signal $r(t)$ and the output of the VCO, $x(t)$, back to the input of the VCO, $y(t)$.

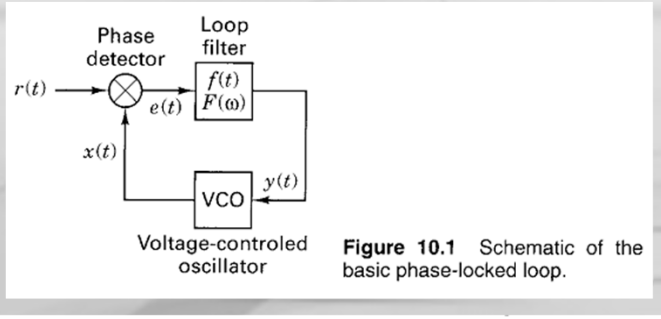


Figure 10.1 Schematic of the basic phase-locked loop.

In the case of modern digital receivers, the error detector may be mathematically much more complicated than the simple multiplier shown in Figure 10.1. For example, the error detector might be a set of matched-filter correlators, each matched to a slightly different phase offset feeding a weighting or decision function. The output of the weighting function would be the phase error estimate. Such a function might be mathematically very complex, but it would be easily approximated using modern digital technology. The VCO may not appear to be a sinusoidal oscillator, but it may be implemented as a read-only memory whose pointers are controlled by a combination of a clock and the output of the error estimator. The feedback path may not be continuous (as shown in Figure 10.1), but phase corrections may only be applied once per frame, or once per packet, depending on the signal structure. A special header or known sequence of symbols may be inserted into the information stream for the expressed purpose of aiding the estimation process. These obvious differences notwithstanding, the basic principles are still illuminated with the simple model of Figure 10.1.

Consider a normalized input signal of the form

$$r(t) = \cos\left[\omega_0 t + \theta(t)\right] \tag{10.1}$$

where ω_0 is the nominal carrier frequency and $\theta(t)$ is a slowly varying phase. Similarly, consider a normalized VCO output of the form

$$x(t) = -2\sin\left[\omega_0 t + \hat{\theta}(t)\right] \tag{10.2}$$

These signals will produce an output error signal at the phase detector output of the form

$$\begin{aligned} e(t) = x(t)r(t) &= 2\sin\left[\omega_0 t + \hat{\theta}(t)\right]\cos\left[\omega_0 t + \theta(t)\right] \\ &= \sin\left[\theta(t) - \hat{\theta}(t)\right] + \sin\left[2\omega_0 t + \theta(t) + \hat{\theta}(t)\right] \end{aligned} \tag{10.3}$$

Synchronization

Assuming that the loop filter is low pass, the second term on the right hand side of Equation (10.3) will be filtered out and can be ignored. This low-pass assumption is a reasonable loop design decision. A low-pass filter provides an error signal that is solely a function of the difference in phases between the input [Equation (10.1)] and the VCO output [Equation (10.2)]. This is exactly the error signal that is needed. The VCO output frequency is the time derivative of the argument of the sine function in Equation (10.2). If we make the assumption that ω_0 is the uncontrolled frequency of the VCO (the output frequency when the input voltage is zero), we can express the difference in the VCO output frequency from ω_0 as the time differential of the phase term $\hat{\theta}(t)$. The output frequency of the VCO is a linear function of the input voltage. Therefore, since an input voltage of zero produces an output frequency of ω_0, the difference in the output frequency from ω_0 will be proportional to the value of the input voltage $y(t)$, or

$$\begin{aligned}\Delta\omega(t) &= \frac{d}{dt}[\hat{\theta}(t)] = K_0 y(t) \\ &= K_0 e(t) * f(t) \\ &\approx K_0[\theta(t) - \hat{\theta}(t)] * f(t)\end{aligned} \tag{10.4}$$

where $\Delta\omega(t)$ denotes the frequency difference, the notation $*$ indicates the convolution operation (see Appendix A), and the small-angle approximation [i.e., $e(t) = \sin[\theta(t) - \hat{\theta}(t)] \cong \theta(t) - \hat{\theta}(t)$] has been used in the last line of Equation (10.4). The small-angle approximation will be accurate when the output phase error is small (the loop is close to phase lock). This will be the situation when the loop is operating normally. The factor K_0 is the gain of the VCO, and $f(t)$ is the loop-filter impulse response. This linear differential equation in $\hat{\theta}(t)$ (utilizing the small-angle approximation) is known as the linearized loop equation. It is the single most useful relationship in determining loop behavior during normal operation (where the phase error is small).

Example 10.1 Linearized Loop Equation

Show that for appropriately chosen K_0 and $f(t)$ the linearized loop equation [Equation (10.4)] demonstrates a tendency toward phase lock—that is, the phase difference between the incoming signal and the VCO output tends to decrease.

Solution

Consider the case where the phase of the input signal, $\theta(t)$, is slowly varying with time. It can be seen that if the phase difference on the right-hand side of Equation (10.4) is positive [i.e., $\theta(t) > \hat{\theta}(t)$], then by appropriate choice of K_0 and $f(t)$, the time derivative of $\hat{\theta}(t)$ will be positive, so that $\hat{\theta}(t)$ will increase with time, which will tend to reduce the magnitude of the difference $|\theta(t) - \hat{\theta}(t)|$. On the other hand, if the phase difference is negative, $\hat{\theta}(t)$ will decrease with time, which will also reduce the magnitude of the phase difference. Finally, if $\theta(t) = \hat{\theta}(t)$, then Equation (10.4) indicates that $\hat{\theta}(t)$ will not change with time, and the equality will be maintained.

Consider the Fourier transform of Equation (10.4),

$$j\omega\hat{\Theta}(\omega) = K_0[\Theta(\omega) - \hat{\Theta}(\omega)]F(\omega) \tag{10.5}$$

where the capitalized functions of ω are the Fourier transforms of the lowercase functions of t in Equation (10.4). That is, $\hat{\Theta}(\omega) \leftrightarrow \hat{\theta}(t)$, $\Theta(\omega) \leftrightarrow \theta(t)$, and $F(\omega) \leftrightarrow f(t)$. Reorganizing Equation (10.5) provides

$$\frac{\hat{\Theta}(\omega)}{\Theta(\omega)} = \frac{K_0 F(\omega)}{j\omega + K_0 F(\omega)} = H(\omega) \tag{10.6}$$

The term $H(\omega)$ is known as the closed-loop transfer function of the PLL. This term is very useful in characterizing the transient response of a PLL. The order of a PLL is defined to be the order of the highest-order term in $j\omega$ in the denominator of $H(\omega)$. Equation (10.6) indicates that this is always one more than the order of the loop filter $F(\omega)$. This is because when $F(\omega)$ is expressed analytically as $F(\omega) = N(\omega)/D(\omega)$, the denominator of $H(\omega)$ when expressed as a polynomial in $j\omega$ will have the term $j\omega D(\omega)$, which must have a term in $j\omega$ that is one order higher than the highest-order term in $D(\omega)$ alone. The order of a PLL is critical for determining the loop's steady-state response to a steady-state input. This is discussed in the next section.

Synchronization

10.2.1.1 Steady-State Tracking Characteristics

By reorganizing Equation (10.6), we can obtain an expression for the Fourier transform of the phase error:

$$\begin{aligned} E(\omega) &= \mathcal{F}\{e(t)\} \\ &= \Theta(\omega) - \hat{\Theta}(\omega) \\ &= [1 - H(\omega)]\Theta(\omega) \\ &= \frac{j\omega\Theta(\omega)}{j\omega + K_0 F(\omega)} \end{aligned} \tag{10.7}$$

Equation (10.7) can be used in conjunction with the final value theorem of Fourier transforms to determine the steady-state error response of a loop to a variety of possible input characteristics. The steady-state error is the residual error after all transients have died away, and thus provides a measure of a loop's ability to cope with various types of changes in the input. The final value theorem states that

$$\lim_{t\to\infty} e(t) = \lim_{j\omega\to 0} j\omega E(\omega) \tag{10.8}$$

Combining Equations (10.7) and (10.8) yields

$$\lim_{t\to\infty} e(t) = \lim_{j\omega\to 0} \frac{(j\omega)^2\Theta(\omega)}{j\omega + K_0 F(\omega)} \tag{10.9}$$

Example 10.2 Response to a Phase Step

Consider a loop's steady-state response to a phase step at the loop input.

Solution

Assuming that the PLL was originally in phase lock, a phase step will throw the loop out of lock. Having abruptly changed, however, the input phase again becomes stable.

This should be the easiest type of phase disturbance for a PLL to deal with. The Fourier transform of a phase step will be taken to be

$$\begin{aligned} \Theta(\omega) &= \mathcal{F}\{\Delta\phi\, u(t)\} \\ &= \frac{\Delta\phi}{j\omega} \end{aligned} \tag{10.10}$$

where $\Delta\phi$ is the magnitude of the step and $u(t)$ is the unit step function

$$\begin{aligned} u(t) &= \begin{cases} 1 & \text{for } t > 0 \\ 0 & \text{for } t < 0 \end{cases} \\ &= \int_{-\infty}^{t} \delta(\tau)\, d\tau \end{aligned}$$

in which $\delta(\tau)$ is the Dirac delta function. From Equation (10.9) and (10.10),

$$\lim_{t\to\infty} e(t) = \lim_{j\omega\to 0} \frac{j\omega\Delta\phi}{j\omega + K_0 F(\omega)} = 0$$

assuming that $F(0) \neq 0$. Thus the loop will eventually track out any phase step that appears at the input if the loop filter has a nonzero dc response. This means that for any loop filter with the property that $F(\omega) = N(\omega)/D(\omega)$ and $N(0) \neq 0$, the PLL will automatically tend to recover phase lock if the input is displaced by a constant phase. This is clearly a very desirable loop characteristic.

Synchronization

Example 10.3 Response to a Frequency Step

Next, consider a loop's steady-state response to a frequency step at the input.

Solution

A frequency step can approximate the effect of a Doppler shift in the incoming signal frequency due to relative motion between the transmitter and the receiver. Thus, this is an important example for systems with mobile terminals. Since phase is the integral of frequency, the input phase will change linearly as a function of time for a constant input-frequency offset. The Fourier transform of the phase characteristic will be the transform of the integral of the frequency characteristic. Since the frequency characteristic is a step, and the transform of an integral is the transform of the integrand divided by the parameter $j\omega$, it follows that

$$\Theta(\omega) = \frac{\Delta\omega}{(j\omega)^2} \tag{10.11}$$

where $\Delta\omega$ is the magnitude of the frequency step. Substituting Equation (10.11) into Equation (10.9) yields

$$\lim_{t\to\infty} e(t) = \lim_{j\omega\to 0} \frac{\Delta\omega}{j\omega + K_0 F(\omega)} = \frac{\Delta\omega}{K_0 F(0)} \tag{10.12}$$

The steady-state result in this case depends on more properties of the loop filter than merely a nonzero dc response. If the filter is "all-pass," then

$$F_{ap}(\omega) = 1 \tag{10.13}$$

If it is low-pass, then

$$F_{\ell p}(\omega) = \frac{\omega_1}{j\omega + \omega_1} \tag{10.14}$$

or if it is a lead-lag, then

$$F_{\ell\ell}(\omega) = \left(\frac{\omega_1}{\omega_2}\right)\frac{j\omega + \omega_2}{j\omega + \omega_1} \tag{10.15}$$

Equation (10.12) indicates that the loop will track the input phase ramp with a constant steady-state error whose value will depend on the gain term K_0 and the magnitude of the frequency step. Using any of $F_{ap}(\omega)$, $F_{\ell p}(\omega)$, or $F_{\ell\ell}(\omega)$ for $F(\omega)$ in Equation (10.12) yields

$$\lim_{t\to\infty} e(t) = \frac{\Delta\omega}{K_0}$$

Notice that a product of several filters with filter characteristics of the form of Equation (10.13), (10.14), or (10.15) would still produce this result. This steady-state error which is called the *velocity error,* will exist regardless of the order of the filter. unless the denominator of $F(\omega)$, contains $j\omega$ as a factor [$\omega_1 = 0$ in the denominator of Equation (10.14) or (10.15) with the appropriate renormalization in the numerators]. Having $j\omega$ as a factor of $D(\omega)$ is equivalent to having a perfect integrator in the loop filter. It is not possible to build a perfect integrator, but one may be closely approximated either digitally or by using active integrated circuits [5]. Thus if the system design requires the tracking of Doppler shifts with zero steady-state error the loop filter design must contain an approximation to a perfect integrator. It should be noted that even with a nonzero velocity error, the frequency is still being tracked: there are important applications where tracking to zero phase error is not important. Noncoherent signaling, such as the standard use of FSK modulation, is an example. For noncoherent signaling it is actually frequency tracking that is required, and the absolute value of phase is unimportant.

Synchronization

Example 10.4 Response to a Frequency Ramp

Consider a loop's steady state response when the input frequency is changing linearly with time (a frequency ramp function).

Solution

This example corresponds to the effect of a step change in the time derivative of the input frequency. This would approximate a change in the Doppler rate, which could model acceleration in the motion between a satellite or an aircraft and a ground receiver. In this case, the Fourier transform of the phase characteristic is given by

$$\Theta(\omega) = \frac{\Delta\dot{\omega}}{(j\,\omega)^3} \tag{10.16}$$

where $\dot{\omega}$ is the magnitude of the rate of frequency change. In this case, Equation (10.9) yields

$$\lim_{t\to\infty} e(t) = \lim_{j\omega\to 0} \frac{\Delta\dot{\omega}/j\,\omega}{j\,\omega + K_0 F(\omega)} = \lim_{j\omega\to 0} \frac{\Delta\dot{\omega}}{j\,\omega K_0 F(\omega)} \tag{10.17}$$

If the loop has a nonzero velocity error—that is, if the right-hand side of Equation (10.12) is not equal to zero—Equation (10.17) shows the steady-state phase error to be unbounded due to a frequency ramp. This says that a PLL with loop filters given by any of Equations (10.13) to (10.15) will not be able to track a frequency ramp. In order to track a frequency ramp, the denominator of the loop filter transform $D(\omega)$ must have $j\omega$ as a factor. From Equation (10.17) it can be seen that a loop filter with a transfer function of the type $F(\omega) = N(\omega)/j\omega D_1(\omega)$ will allow the PLL to track a frequency ramp with a constant phase error. This implies that in order to track a signal with a linearly changing Doppler shift (constant relative acceleration), the receiver must have a PLL that is second order or higher. To track a frequency ramp with zero phase error, the loop filter would be required to have a transfer function with $(j\omega)^2$ as a factor of the denominator, $F(\omega) = N(\omega)/(j\omega)^2 D_2(\omega)$. This implies a PLL that is third order or higher. Thus high-performance aircraft that need to track phase accurately through violent maneuvers may require third- or higher-order PLLs. In all cases, frequency lock is available with a loop of one order less than that required for phase lock. Steady-state error analysis is therefore a useful indicator of the required complexity of the loop filters.

In practice, the vast majority of PLL designs are second order. This is because a second-order loop can be made to be unconditionally stable [5]. Unconditionally, stable loops will always try to track the input. No set of input conditions—regardless of how extreme—will cause the loop to respond in the inappropriate direction to changes in the input. Second-order loops will track out the effect of a frequency step (Doppler shift), and they are relatively easy to analyze, since the closed-form results obtained for first-order loops are good approximations for second-order loop performance. Third-order loops are used for some special applications [e.g., some Global Positioning System (GPS) navigation receivers and some airborne receivers have third-order PLLs], but loop performance for third-order loops is relatively difficult to determine, and third- and higher-order loops are only conditionally stable. Often if the signal dynamics are expected to be such that high-order loops would be required for coherent demodulation, noncoherent demodulation is used instead.

Synchronization

10.2.1.2 Performance in Noise

The steady-state analysis of the preceding section tacitly assumed that the input signal was noise free. In some situations this may be approximately correct, but as in other parts of communication analysis, the more general case would include the effects of noise.

Reconsider the normalized loop input signal of Equation (10.1) and Figure 10.1. With the inclusion of normalized narrowband additive Gaussian noise $n(t)$, the expression for the input becomes

$$r(t) = \cos(\omega_0 t + \theta) + n(t) \tag{10.18}$$

where, for the moment, we consider the input phase offset, θ to be a constant. The noise process $n(t)$, assumed to be a zero-mean narrowband Gaussian process, can be expanded into quadrature components about the carrier frequency as [6]

$$n(t) = n_c(t)\cos\omega_0 t + n_s(t)\sin\omega_0 t \tag{10.19}$$

where both $n_c(t)$ and $n_s(t)$ are zero-mean Gaussian random processes and are statistically independent. Now the output of the phase detector can be written as [see Equation (10.3)]

$$e(t) = x(t)r(t) \tag{10.20}$$

$$= \sin(\theta - \hat{\theta}) + n_c(t)\cos\hat{\theta} + n_s(t)\sin\hat{\theta} + (\text{terms at twice the carrier frequency})$$

As before, the loop filter eliminates the twice-carrier-frequency terms. Denoting the second and third terms of Equation (10.20) as

$$n'(t) = n_c(t)\cos\hat{\theta} + n_s(t)\sin\hat{\theta} \tag{10.21}$$

we see that it is easy to verify that the variance of $n'(t)$ is identical to the variance of $n(t)$. This variance will be denoted by σ_n^2.

Consider the autocorrelation function of $n'(t)$,

$$\begin{aligned} R(t_1, t_2) &= \mathbf{E}\{n'(t_1)n'(t_2)\} \\ &= \mathbf{E}\{n_c(t_1)n_c(t_2)\}\cos^2\hat{\theta} + \mathbf{E}\{n_s(t_1)n_s(t_2)\}\sin^2\hat{\theta} \\ &\quad + [\mathbf{E}\{n_c(t_1)n_s(t_2)\} + \mathbf{E}\{n_s(t_1)n_c(t_2)\}]\sin\hat{\theta}\cos\hat{\theta} \end{aligned} \tag{10.22}$$

where $\mathbf{E}\{\cdot\}$ denotes the expected value. The cross-terms on the right-hand side of Equation (10.22) are equal to zero because n_c and n_s are mutually independent and have zero means [6]. With the assumption of wide-sense stationarity [7], we have

$$R(\tau) = R_c(\tau)\cos^2\hat{\theta} + R_s(\tau)\sin^2\hat{\theta} \tag{10.23}$$

where $\tau = t_1 - t_2$. Taking Fourier transforms, the power spectral density of $n'(t)$ is seen to be

$$\begin{aligned} G(\omega) &= \mathcal{F}[R(t)] \\ &= G_c(\omega)\cos^2\hat{\theta} + G_s(\omega)\sin^2\hat{\theta} \end{aligned} \tag{10.24}$$

where G_c and G_s are the Fourier transforms of R_c and R_s, respectively. But from Equation (10.19), it can be seen that the spectra G_c and G_s are made of shifted versions of the spectra of the original noise process $n(t)$. Therefore, because of our construction [8],

$$G_s(\omega) = G_c(\omega) = G_n(\omega_0 - \omega) + G_n(\omega_0 + \omega)$$

where $G_n(\omega)$ is the spectral density of the original bandpass noise process $n(t)$. Equation (10.24) can be rewritten as

$$G(\omega) = G_n(\omega_0 - \omega) + G_n(\omega_0 + \omega) \tag{10.25}$$

For the special case of white noise, we have $G_n(\omega) = N_0/2$ Watts/Hertz, where N_0 is the single-sided spectral density of the white noise. Thus, from Equation (10.25), for this important special case,

$$G(\omega) = N_0 \tag{10.26}$$

The value in this development is that for the same small-angle approximations that were made in the preceding section, the spectral density of the VCO phase, $G_{\hat{\theta}}$, is related to the spectral density of the noise process through the loop transfer function [Equation (10.6)]. That is,

$$G_{\hat{\theta}}(\omega) = G(\omega)\,|H(\omega)|^2 \tag{10.27}$$

where $G(\omega)$ is as given in Equation (10.25) and $H(\omega)$ as defined in Equation (10.6). The variance of the output phase is then

$$\sigma_{\hat{\theta}}^2 = \frac{1}{2\pi}\int_{-\infty}^{\infty} G(\omega)|H(\omega)|^2\,d\omega \tag{10.28}$$

Synchronization

10.2.1.3 Nonlinear Loop Analysis

All of the PLL discussion in the previous sections has utilized what is called the linearized PLL model. This model is shown schematically in Figure 10.2. The model makes use of the small-angle approximation

$$\sin(\theta - \hat{\theta}) \approx \theta - \hat{\theta} \tag{10.32}$$

which is accurate when the loop is "in lock" and performing as desired (i.e., with small phase errors). Clearly, these conditions form only part of the picture. A complete analysis of PLL performance must allow for the times when Equation (10.32) is not accurate. When the small-angle approximation is inaccurate, an appropriate model is the one shown schematically in Figure 10.3. From Equations (10.4), (10.20), and (10.21) and Figure 10.3, the model can be described by the differential equation

$$\frac{d}{dt}[\hat{\theta}(t)] = K_0 f(t) * \sin[\theta(t) - \hat{\theta}(t)] + K_0 f(t) * n'(t) \tag{10.33}$$

where, as before, * denotes the convolution operation. In spite of the best efforts of many researchers, this differential equation has resisted general solution for many years. However, Viterbi [8] derived a closed-form solution for an important special case.

Consider the case where $\theta(t)$, the input phase as a function of time, is a constant θ. We can now define a new phase variable

$$\phi(t) = [\theta - \hat{\theta}(t)] \text{ modulo } 2\pi \tag{10.34}$$

Because θ is constant, Equation (10.33) can be rewritten as

$$\frac{d}{dt}[\phi(t)] = K_0 f(t) * \sin\phi(t) + K_0 f(t) * n'(t) \tag{10.35}$$

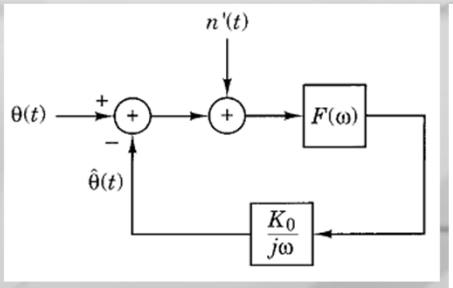


Figure 10.2 Schematic of linearized PLL model.

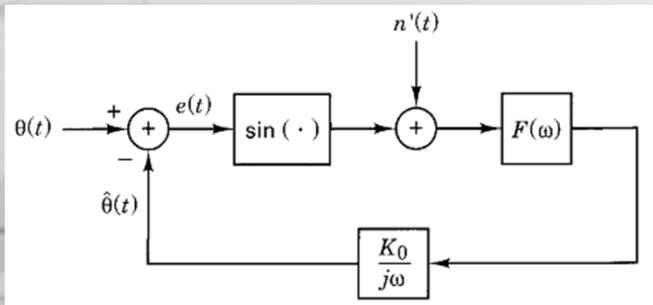


Figure 10.3 Schematic of nonlinearized PLL model.

Since, from Equation (10.35), $\phi(t)$ is a function of the random process $n'(t)$, $\phi(t)$ itself is a random process. Because $\phi(t)$ is defined modulo 2π, it can be shown [5] that $\phi(t)$ is stationary in the limit when all transient effects have died down (i.e., θ is a constant). Viterbi [8] determined that for a first-order PLL (i.e., the loop filter is a short circuit, or equivalently $f(t) = \delta(t)$), the probability density function of ϕ is of the form

$$p(\phi) = \frac{\exp(\rho\cos\phi)}{2\pi I_0(\rho)} \quad \text{for } |\phi| \le \pi \tag{10.36}$$

where $\rho = 1/\sigma_{\hat{\theta}}^2$ [see Equation (10.31)] is the normalized (to unit signal energy) loop signal-to-noise ratio, and $I_0(\rho)$ is the zeroth-order modified Bessel function of the first kind, evaluated at ρ. The phase variance, modulo 2π, can now be computed using Equation (10.36). The resulting value of the phase variance will be exact for first-order loops, and is an extremely useful approximation for the behavior of many second-order loops [5]. It has also been shown to be an exact form for higher-order loops under a modified definition of ρ [9].

The change of variable from a phase that can take any real value to a phase that is modulo 2π results in the concept of loop cycle slips. A cycle slip occurs when the magnitude of the original phase error, $|\theta - \hat{\theta}(t)|$, exceeds 2π radians. This will cause the value of ϕ [Equation (10.34)] to abruptly change from about 2π to about 0. This event can be thought of as a momentary loss of lock with an almost immediate reacquisition. The statistics of cycle slips can be as important an indicator of PLL performance as phase variance—especially at low-loop signal-to-noise ratios, when cycle slips may occur frequently.

By manipulating his phase-distribution results, Viterbi [8] derived an expression for the mean time to the first cycle slip, T_m, beginning at some arbitrary reference time:

$$T_m = \frac{\pi^2 \rho I_0^2(\rho)}{2B_L} \tag{10.37}$$

For large ρ, this expression can be approximated by

$$T_m \approx \frac{\pi \exp(2\rho)}{4B_L} \tag{10.38}$$

As was true with the probability density function of Equation (10.36), these results were derived for first-order loops, but they are useful approximations for the behavior of second-order loops, and they provide an upper bound to second-order loop performance at medium and large loop signal-to-noise ratios. In addition, computer simulations and laboratory measurements [5] indicate that the time T between cycle slips is exponentially distributed:

$$P(T) = 1 - \exp\left(-\frac{T}{T_m}\right) \tag{10.39}$$

This is to say that the probability that a loop will cycle-slip within time T, starting from zero phase error, is given by Equation (10.39).

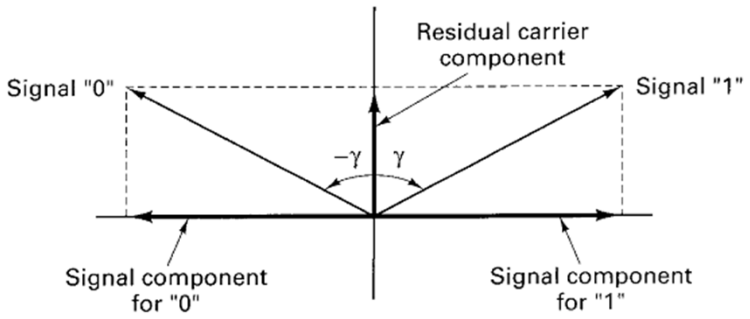


Figure 10.4 Residual carrier binary phase modulation.

10.2.1.4 Suppressed Carrier Loops

The discussion of PLLs to this point has presumed that the carrier input is a fairly stable sinusoid with some known positive average energy. In the case of a phase modulated communication system, if the carrier phase variation due to the modulation is less than $\pi/2$ radians, there will be positive energy at the carrier frequency. This is called a system design that has a residual carrier component, and all of the discussion of PLL development to this point would apply directly to this residual component. A diagram of the signal space for a binary phase modulated system with a residual carrier component is given in Figure 10.4, for a modulating angle of $\gamma \leq \pi/2$. At one time, most phase modulated systems were designed in this way. However, the residual carrier component is, in a sense, wasted energy—in the sense that the energy in the residual carrier is not being used to transmit the information, only to transmit the carrier. Thus most modern phase modulated systems are suppressed carrier systems. This means that there is no average energy transmitted at the carrier frequency. All of the transmitted energy goes into the modulation. Unfortunately, this means that there is no longer any signal for the basic PLL of Figure 10.1 to track.

Consider, as an example, a BPSK signal

$$r(t) = m(t) \sin(\omega_0 t + \theta) + n(t) \tag{10.40}$$

where $m(t) = \pm 1$ with equal probability. This is a suppressed carrier transmission—the average energy at radian frequency ω_0 is zero. This situation is represented graphically in Figure 10.4, when $\gamma = \pi/2$. The figure indicates that for this case the vertical carrier component will vanish. To acquire and track the phase of the carrier, the effects of the modulation must be eliminated. One way to eliminate the modulation is to square the signal:

$$\begin{aligned} r^2(t) &= m^2(t) \sin^2(\omega_0 t + \theta) + n^2(t) + 2n(t)m(t) \sin(\omega_0 t + \theta) \\ &= \tfrac{1}{2} - \tfrac{1}{2}\cos(2\omega_0 t + 2\theta) + n^2(t) + 2n(t)m(t) \sin(\omega_0 t + \theta) \end{aligned} \tag{10.41}$$

Here, we have made of the fact that $m^2(t) = 1$. The second term on the right-hand side of Equation (10.41) is a carrier-related term (at twice the original carrier

Synchronization

frequency) that can be acquired and tracked with a basic PLL of the type illustrated in Figure 10.1. Such an arrangement is illustrated in Figure 10.5. When the incoming suppressed-carrier waveform is squared, the resulting twice-carrier component can be acquired and tracked by a PLL of standard design. Some of the problems with this procedure can be inferred from Equation (10.41). The first problem is simply that all phase angles have been doubled. Thus, the phase noise and phase jitter has been doubled, and the phase error variance (related to the phase noise squared) is larger by a factor of 4 than that of the original signal. This angle doubling is offset by the divide-by-2 circuit at the VCO output, and, therefore, does not directly affect the accuracy of the loop's output signal that is used by the data demodulator. However, this larger internal variation will cause the PLL to require a 6-dB-larger carrier signal-to-noise ratio than a residual carrier system in order to maintain phase lock. In addition, now there are two effective noise terms interfering with loop operation, because of the cross-correlation term between noise and signal in Equation (10.41). For cases of medium or low loop signal-to-noise ratio, these two noise terms will reduce the available signal-to-noise ratio even further relative to the original unmodulated carrier. This additional loss due to signal-times-noise and noise-times-noise terms is called the *loop squaring loss* S_L. Gardner [5] shows that if the input noise process $n(t)$ is a narrowband Gaussian noise of bandwidth B_i, the squaring loss is upper bounded by

$$S_L \leq 1 + N_0 B_i \tag{10.42}$$

where, as before, N_0 is the single-sided power spectral density of the prefiltered, normalized white Gaussian noise process. Equation (10.42) is an upper bound because the filter bandwidth B_i is tacitly assumed to be wide enough to pass the signal undistorted. In an actual design, signal distortion can be traded for squaring loss, as is shown in [10].

Since the normalization in Equation (10.42) is with respect to the signal powers, the second term is proportional to a signal-to-noise ratio

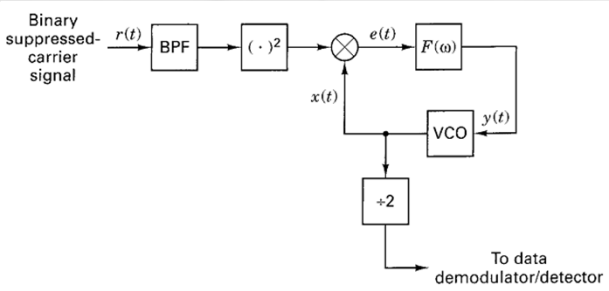


Figure 10.5 Basic squaring loop schematic.

$$\rho_i = \frac{1}{2N_0 B_i} \tag{10.43}$$

where ρ_i is the signal-to-noise ratio in the input filter bandwidth. For large loop signal-to-noise ratios, the output phase variance can now be expressed as

$$\sigma_{\hat{\theta}}^2 = 2N_0 B_L S_L = 2N_0 B_L \left(1 + \frac{1}{2\rho_i}\right) \tag{10.44}$$

The leading term on the right-hand side of Equation (10.44) can be seen to be identical to that of Equation (10.31), the phase variance of the standard PLL. It can also be seen that for large input signal-to-noise ratios, the second term in the squaring loss will vanish, and we are left with the phase variance of the standard PLL.

Another potentially serious problem, associated mainly with suppressed carrier loops, is that of *false lock* [5, 11–13]. This can be a problem especially during acquisition or reacquisition of carrier phase. The interaction of the data stream with the loop nonlinearities (especially the squaring circuit) and loop filters will produce sidebands in the spectrum that is input to the phase detector. These sidebands can contain stable frequency components. Care must be taken that these stable components are not allowed to capture the tracking loop. If the loop is captured, it will appear to be operating correctly; the VCO control signal $y(t)$ will be small but the VCO output will be offset in frequency from the correct carrier component. This is false lock. The loop is tracking a sideband frequency component, and the loop filter is filtering out the real carrier. False lock is a hardware-implementation problem that typically sets an effective lower limit on the bandwidth of the loop filters. Because they have fewer nonlinear elements, false locking is not usually a problem with residual carrier loops.

Synchronization

10.2.1.5 Costas Loops

An important form of a suppressed carrier loop is the Costas loop, shown schematically in Figure 10.6. This loop design is important because it eliminates the square-law device, which can be difficult to implement at carrier frequencies, and replaces it with a multiplier and relatively simple low-pass filters. Although the appearance of the circuits in Figures 10.5 and 10.6 is quite different, their theoretical performance can be shown to be the same [5]. The main remaining implementation problem with Costas loops is that to achieve the theoretically optimum performance, the two low-pass arm filters must be perfectly matched. This can only be approximated in any analog hardware implementation. If the arm filters are implemented digitally, there will be no problem keeping them matched, but the designer will confront the usual sampled data design issues. Thus the decision as to whether to implement a Costas loop or the classical design of Figure 10.5 amounts to a design decision between the difficulty of implementing the squaring device and the difficulty of implementing closely matched arm filters. This design decision will depend on the parameters and requirements of the particular receiving system, and cannot be generalized here.

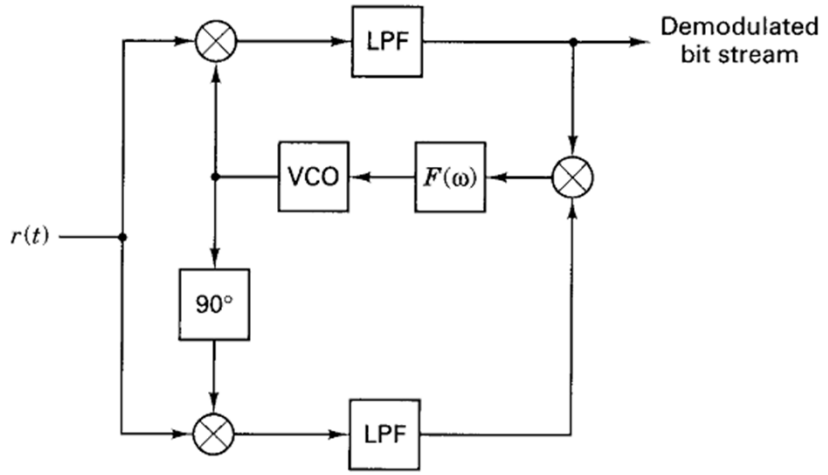


Figure 10.6 Costas loop.

Synchronization

10.2.1.6 High-Order Suppressed-Carrier Loops

Binary phase-shift keying is not the only type of suppressed-carrier modulation. In fact, assuming that all signals are equally likely *a priori,* any modulation scheme whose average amplitude, averaged over the signal set, is zero will have no average energy in the transmitted carrier. Perhaps the most common nonbinary suppressed-carrier modulation is quadrature phase-shift keying, or QPSK (4-ary PSK). If a QPSK signal is squared, the result "looks like" a BPSK signal. Thus, for equally likely QPSK signals, the carrier is still suppressed. However, squaring the signal a second time—equivalent to taking the original signal to the fourth power—can be seen to produce a term with a carrier component at four times the transmitted carrier's frequency. As in the binary case, operating on the incoming signal with a power-law device produces cross products among the noise terms and signal terms, and introduces the equivalent of a "squaring loss." Under the assumption that the noise bandwidth will pass the signal undistorted, the loss for fourth-power loops is upper bounded by [5]

$$S_L \leq 1 + \frac{9}{\rho_i} + \frac{6}{\rho_i^2} + \frac{3}{2\rho_i^3} \tag{10.45}$$

As was the case with the squaring loop, for sufficiently high input signal-to-noise ratios, ρ_i, Equation (10.45) indicates that the additional loss terms vanish, and the loop performance approaches that of the basic loop. As was also the case for the squaring loop, there are Costas loop designs equivalent to fourth-order loops [5, 14, 15] that may exhibit hardware implementation advantages. Their theoretical performance, however, is the same as that of the straightforward fourth-power design.

Example 10.5 Squaring Loss Bounds

Compare the upper bounds on squaring loss S_L given by Equations (10.42) and (10.45) for second- and fourth-power loops, respectively, for an input loop signal-to-noise ratio ρ_i of 10 dB.

Solution

A 10dB signal-to-noise ratio is also 10 in terms of its power ratio. Therefore, from Equations (10.42) to (10.44), for the squaring loop,

$$S_L = 1 + \frac{1}{2\rho_i} = 1.05 = 0.2 \text{ dB}$$

From Equation (10.45), for the fourth-power loop,

$$S_L = 1 + 0.9 + 0.06 + 0.0015 = 1.9615 = 2.9 \text{ dB}$$

Thus, while an input signal-to-noise ratio of 10 dB is adequate to keep losses small for the squaring loop, the same signal-to-noise ratio may allow significant losses for the fourth-power loop.

10.2.1.7 Acquisition

In most of the discussion thus far, the assumption has been that the PLL is in lock. This was the justification for assuming that the phase error $|\theta - \hat{\theta}|$ was small. At one time or another, however, every loop must acquire lock—that is, it must be brought into lock. Acquisition can be accomplished with the aid of external circuits or signals (aided acquisition) or in some cases by an unaided PLL (self-acquisition) [5].

Acquisition is an inherently nonlinear operation and therefore is difficult to analyze in general. However, some intuition may be obtained by considering a noise-free, first-order loop. Such a loop is shown schematically in Figure 10.3, where $n'(t) = 0$ (noise-free) and $F(\omega) = 1$ (first-order). Denote the input phase as

$$\theta(t) = \omega_i t$$

and the output phase as

$$\hat{\theta}(t) = \omega_0 t + \int_0^t K_0 \sin e(t)\, dt + \hat{\theta}(0) \tag{10.46}$$

where ω_i and ω_0 are the radian frequencies of the input and output signals, respectively. Thus the phase error is given by

$$\begin{aligned} e(t) &= \theta(t) - \hat{\theta}(t) \\ &= (\omega_i - \omega_0)t + \int_0^t K_0 \sin e(t)\, dt + \hat{\theta}(0) \end{aligned} \tag{10.47}$$

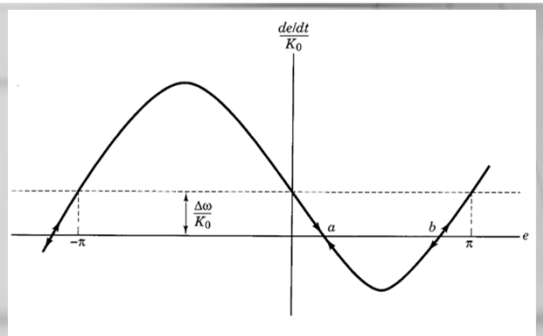


Figure 10.7 Phase-plane plot of first-order loop.

Differentiating both sides and letting $\Delta\omega = \omega_i - \omega_0$ provides

$$\frac{de}{dt} = \Delta\omega - K_0 \sin e \tag{10.48}$$

where the time dependence of the function $e(t)$ has been suppressed to ease notation. This differential equation describes the behavior of the first-order noise-free PLL. The loop being in lock requires that

$$\frac{de}{dt} = 0 \tag{10.49}$$

Equation (10.49) is a necessary, but not a sufficient, condition for phase lock. This can be verified by observing the phase plane diagram of Figure 10.7. This figure is obtained by dividing both sides of Equation (10.48) by the gain term K_0, and plotting the results. First observe point *a*. If the phase error is displaced a little to the left or right of point *a*, the sign of the derivative term is such that the phase error *e*, will be driven back toward *a*. Thus, point *a* is a stable point of the system, a point where phase lock can be obtained and will be maintained. Now consider the case of point *b*. If the phase error is exactly at *b*, Equation (10.49) will be satisfied. However, if there is any slight offset from *b*, the sign of the derivative term will be such that the error will be driven away from *b*. Thus *b* is a point of marginal stability for the loop, a point where Equation (10.49) is satisfied, but not a stable lock point.

The amount of time required for a loop to come into lock can be a very important system design consideration. By observing Equation (10.48), we can see that the requirement of Equation (10.49) for phase lock cannot be met unless

$$\frac{|\Delta\omega|}{|K_0|} \leq 1 \tag{10.50}$$

This is because sinusoidal functions have a maximum amplitude of unity. This range of frequency difference, $-K_0 < \Delta\omega < K_0$, is sometimes called the lock-in range of the loop. Assuming that Equation (10.50) holds, Gardner [5] gives a rule of thumb of $3/K_0$ seconds for the time required for loop acquisition. Actual values can be obtained from Equation (10.47) for well-defined sets of initial conditions, or by extensive computer simulation. It can be seen from the phase plot of Figure 10.7 that the time required will vary widely as a function of the initial phase error. For phase errors very close to point *b*, the driving force $(de/dt)/K_0$ will be very small. Thus, for this worst-case phase error, the error could "linger" in the vicinity of *b* for a long time. This phenomenon is called terminal loop hang-up [16] and can be a serious problem for system designs that depend on self-acquisition.

Synchronization

Perhaps the most important operational difference between first-order and higher-order loops is the higher-order loop's ability to "pull in" from frequency differences that are larger than the lock-in range. A first-order loop with a frequency error larger than the lock-in range will drift toward lock but never quite lock in. Why? Second- and higher-order loops can pull in and achieve phase lock because of their more complicated phase-plane characteristics. (Interested readers should consult Viterbi [8] and other texts on PLLs for more details [5, 9, 17–19].)

The study of self-acquisition for phase-locked loops is mostly of academic interest. Gardner [5] states that loops using self-acquisition can be guaranteed to acquire in reasonable time only under very benign circumstances. This, unfortunately, is rarely the case in practice.

Acquisition aiding drives the loop through the region of phase space expected to contain the lock-in region by means of some external driving signal. This is the most common means of achieving acquisition. Aiding can be implemented by simply applying a voltage ramp to the input of the VCO. This driving signal will cause the VCO output frequency to vary linearly with time. As was shown earlier [Equation (10.17)], loops with loop filters that do not contain $j\omega$ as a factor of their transfer function's denominator cannot track a frequency ramp with finite phase error. Therefore, if frequency sweeping is to be employed with a first-order loop or a second-order loop without this transfer function characteristic, the rate of frequency sweep must be slow enough so that when the loop achieves lock, the presence of phase lock can be detected and the sweeping signal removed before it drives the loop back out of lock. With loops that contain $j\omega$ as a factor of $D(\omega)$, it may not be necessary to remove the sweeping signal at all, because, at least in theory, the loop will be able to track out the frequency ramp. In any case, the sweep rate must not be too large, or the loop will be driven through the lock point so fast that it will fail to acquire. For a second-order loop with loop transfer function [see Equation (10.6)],

$$H(\omega) = \frac{1}{-(j\omega/\omega_n)^2 + 2\zeta(j\omega/\omega_n) + 1} \tag{10.51}$$

Gardner [5] indicates that the maximum sweep rate, $\Delta\dot{\omega}$, must be in the vicinity of

$$\Delta\dot{\omega} \approx \frac{1}{2}\omega_n^2 (1 - 2\sigma_{\hat{\theta}}) \tag{10.52}$$

where $\sigma_{\hat{\theta}}$ is as defined in Equation (10.31) and ω_n, implicitly defined in Equation (10.51), is called the *natural frequency* of a second-order PLL and is related to the loop bandwidth B_L and loop damping factor ζ [9] by

$$\omega_n = \frac{8\zeta}{4\zeta^2 + 1} B_L$$

Blanchard [17] gives more detailed results for aided phase acquisition.

10.2.1.8 Phase Tracking Errors and Link Performance

If a loop is unable to track out all phase errors, the received symbol-error probability will be degraded relative to what is theoretically achievable. The analysis required to determine the amount of the degradation is very involved, but for most of the standard coherent signaling systems, curves are available [14, 15, 20]. Figure 10.8 is an example of such a performance curve for a residual carrier-phase tracking loop operating on a signal with BPSK modulation in additive Gaussian noise. It can be seen that for signal-to-noise ratios of moderate value, small phase errors produce very little degradation. It is only when the standard deviation of phase error exceeds 0.3 that the degradations become significant. This means that the inherent degradation in performance caused by a well-designed loop operating in benign conditions can generally be ignored. The curve also indicates that if conditions are such that the phase variance is large, increasing the data signal-to-Gaussian noise ratio may not be effective in reducing the detected error probability. It should be noted that the presence of an irreducible error in these situations is a characteristic of residual carrier designs with constant loop signal-to-noise ratios ρ_i. Suppressed carrier tracking loops tend not to have irreducible errors, because an increase in the data signal-to-noise ratio will increase the signal-to-noise ratio of the suppressed carrier tracking loop, reducing the tracking error.

Example 10.6 PLL Signal-to-Noise Ratio

Develop an integral expression for the effect on link bit error probability of slowly varying phase tracking errors for a residual carrier BPSK link. Compare the effect of a normalized loop signal-to-noise ratio ($\rho = 1/\sigma_{\hat{\theta}}^2$) of 20 dB with one of 10 dB on error performance at a desired bit error probability of 10^{-5} using Figure 10.8.

Solution

From Chapter 4, the theoretically possible bit error probability for a BPSK link in additive white Gaussian noise of single-sided spectral density N_0 Watts/Hertz is given by

$$P_B = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

where E_b is the received energy per bit time. From the derivation of this expression for error probability, it can be shown that if there is a slowly varying (with respect to the data rate) phase-tracking error of β radians, the resulting probability of error will be given by

$$P_B(\beta) = Q\left(\sqrt{\frac{2E_b \cos\beta}{N_0}}\right)$$

Now if the phase error β is the result of tracking errors caused by system noise, β will be described stochastically by some probability density function $p(\beta)$. Then the expected bit-error probability is given by

$$P_B = \int_0^{2\pi} P_B(\beta)\, p(\beta)\, d\beta$$

For the special case of a first-order loop, the probability density function is given by Equation (10.36). Then the final expression for the expected bit-error probability is given by

$$P_B = \int_0^{2\pi} Q\left(\sqrt{\frac{2E_b \cos\beta}{N_0}}\right) \frac{\exp(\rho\cos\beta)}{2\pi I_0(\rho)}\, d\beta$$

A loop signal-to-noise ratio (ρ_i) of 20 dB will correspond to a standard deviation of phase noise of $\sigma_{\hat{\theta}} = 0.1$ rad. From Figure 10.8, this small amount of phase noise produces no appreciable degradation in the bit-error probability. A loop ρ_i of 10 dB, however, corresponds to a phase noise standard deviation of $\sigma_{\hat{\theta}} = 0.32$ rad. It can be seen from Figure 10.8 that for a bit-error probability of 10^{-5}, this phase noise standard deviation will require a data SNR of somewhat more than 11 (10.4 dB), rather than a data SNR of 9.1 (9.6 dB) for perfect phase tracking. Thus, this loop signal-to-noise will cause an error-performance degradation of somewhat more than 0.8 dB, at an error probability of 10^{-5}. It should be noted that for loop SNRs less than about 10 dB, the degradation in performance increases very rapidly. Thus, 10 dB is something of a threshold for reasonable system performance for residual carrier designs. Suppressed carrier designs, having no problem with irreducible error, may do better.

Synchronization

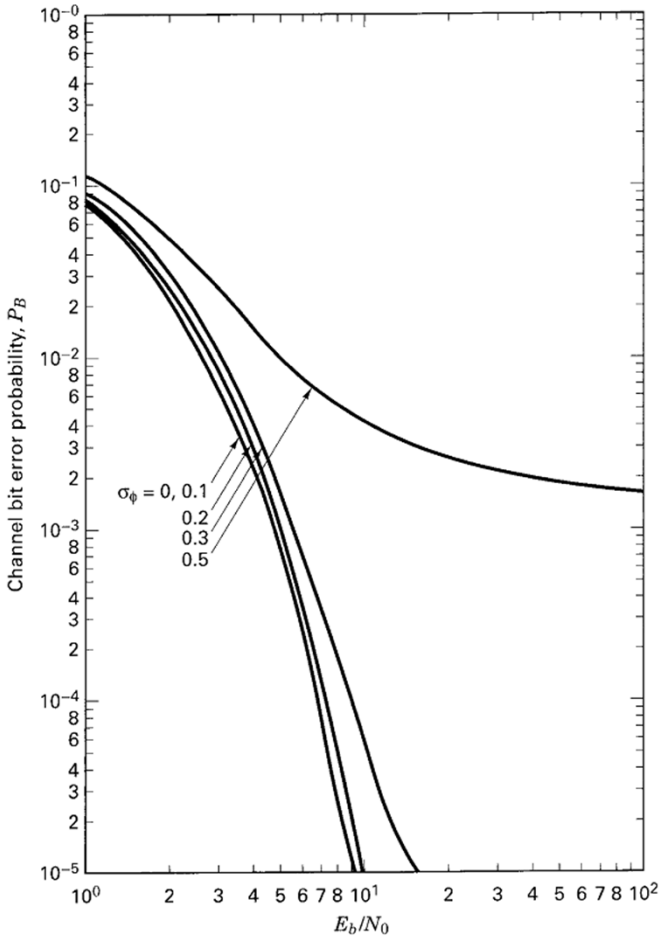


Figure 10.8 Channel bit error probability versus E_b/N_0 for BPSK with imperfect carrier synchronization. (Reprinted with permission from J. J. Stiffler, *Theory of Synchronous Communications*, Prentice-Hall, Inc., Englewood Cliffs, N.J., Fig. 9.1, p. 270.)

10.2.1.9 Spectrum Analysis Techniques

The techniques we have considered thus far belong to a class of synchronizers sometimes called *spectral line techniques.* These techniques all either use an existing spectral line at the carrier frequency or produce such a line at the carrier frequency or a multiple thereof, as a crucial part of the error determination. There is another set of techniques that are especially useful in carrier-frequency estimation or tracking that utilize the shape of the signal's passband spectrum. These techniques have solid roots in maximum-likelihood estimation theory [4], but they also are intuitively appealing and will be approached here from the intuitive direction.

Possibly the most intuitive technique in this class is a simple bank of matched filters, with each filter matched to the expected signal with a different carrier frequency offset. Such a bank of filters could be implemented directly, or by performing a weighting and combining operation on the output of a Fast Fourier Transform. In any case, the filter with the maximum output would be associated with the signal's frequency offset. Such a frequency detector is shown notionally in Figure 10.9. Depending on the signal design and its sensitivity to frequency errors, and on the density of the frequency offsets, either the largest output could be taken as the frequency estimate directly, or additional processing to refine the estimate could be performed. In either case, it is clear that a filter bank that spans the range of possible frequency offsets could be designed in concept, and that such a design would provide a quick and reliable estimate of the carrier frequency offset.

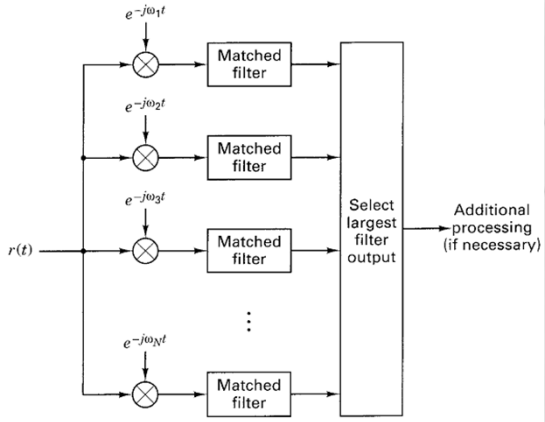


Figure 10.9 Matched filter-bank frequency estimator.

An advantage of the filter bank approach discussed previously is the width of the frequency uncertainty that can be easily accommodated. A disadvantage is the granularity of the initial estimate. A second spectral technique, sometimes called *band-edge filtering,* can provide a mush more accurate estimate, at the cost of a reduction in the initial frequency uncertainty that can be accommodated. The idea can be easily seen through a graphical example.

In the upper graph shown in Figure 10.10, the signal-bandpass spectrum is shown as the wide shaded region, centered on a nominal carrier frequency ω_0. Also shown in this graph are two narrower passband filters at the edges, or roll-off regions of the signal spectrum. If, as is shown in the second graph, the detected signal in both of the two band-edge filters is equal, the signal spectrum must be centered between them, and the nominal carrier frequency error is zero. However, if, as is shown in the third and fourth graphs, the input signal spectrum is shifted relative to the band-edge filters, one of the filters will have more detectable signal, and an error measure can be formed. This error measure could be used to drive a control loop, or it could be used to compute a frequency correction directly. The primary advantage of this kind of technique is that no noise-enhancing nonlinearity is required. The disadvantages are that it requires more knowledge of the signal spectrum, and the implementation of two narrow filters with well-matched passband characteristics. Narrow well-matched filters such as these could represent a design challenge if done with analog circuitry, but, in concept, it could be easily accomplished with digital techniques.

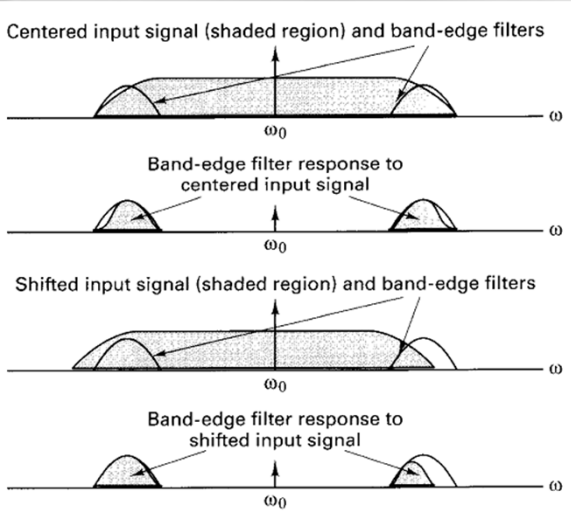


Figure 10.10 Band edge filter example.

Synchronization

10.2.2 Symbol Synchronization—Discrete Symbol Modulations

All digital receivers need to be synchronized to the incoming digital symbol transitions in order to achieve optimum demodulation. In the discussion that follows, we will consider several of the basic types of designs of symbol or data synchronizers. The discussion will center on a random binary baseband signal, for ease of terminology and notation, but the extension to nonbinary baseband signals should be apparent.

The presentation in this section assumes that nothing is known about the actual data sequence. this class of synchronizers is called non-data-aided (NDA) synchronizers. There is another class of symbol synchronizers that use known information about the data stream. This knowledge may be obtained by feeding back decisions on received data, or because a known sequence has been injected into the data stream. Data-aided (DA) techniques have become more important and prevalent with the increasing use of bandwidth-efficient modulation. This is especially true with the class of continuous phase modulations. Data-aided techniques will be considered in somewhat more detail in the succeeding section.

The symbol synchronizers that will be considered here can be classified into two basic groups. The first group consists of the open-loop synchronizers. These circuits recover a replica of the transmitter data clock output directly from operations on the incoming data stream. The second group comprises the closed-loop synchronizers. Closed-loop data synchronizers attempt to lock a local data clock to the incoming signal by use of comparative measurements on the local and incoming signals. Closed-loop methods tend to be more accurate, but they are much more costly and complex.

10.2.2.1 Open-Loop Symbol Synchronizers

Open-loop symbol synchronizers are also occasionally called nonlinear filter synchronizers [20], a very descriptive title. This class of synchronizers generates a frequency component at the symbol rate by operating on the incoming baseband sequence with a combination of filtering and a nonlinear device. The operation is analogous to carrier recovery in a suppressed carrier-tracking loop. In the present case, the desired frequency component, at the data symbol rate, is isolated with a bandpass filter, and "shaped" with a high-gain saturating amplifier. The shaping recovers the square-wave appearance of the data clock signal.

Three examples of open-loop bit synchronizers are shown in Figure 10.11. In the first example (Figure 10.11a), the incoming signal $s(t)$ is filtered with a matched filter. The output of this filter will be the autocorrelation function of the input signal shape. For square-wave signaling, for example, the output will be the familiar isosceles-triangular waveshape. The sequence of bit autocorrelation waveshapes is then "rectified" by some type of memoryless even-law nonlinearity—a square-law device, for example. The resulting waveform will have positive amplitude peaks that correspond, to within a time delay, with the input symbol transitions. This sequence of processes is illustrated in Figure 10.12. Thus the output waveform from the even-law device will contain a Fourier component at the fundamental frequency of the data clock. This frequency component is isolated from its harmonics with a bandpass filter (BPF) and shaped with an ideal saturating amplifier, with transfer function

$$\operatorname{sgn} x = \begin{cases} 1 & \text{for } x > 0 \\ -1 & \text{otherwise} \end{cases} \tag{10.53}$$

The second example in Figure 10.11 produces a Fourier component at the data clock frequency by means of a delay and multiply. The delay shown in Figure 10.11b is half a bit period, which is the best value because it provides the strongest Fourier component [20]. The waveform $m(t)$ will always be positive in the second half of every bit period, but will have a negative first half if there has been a state change in the incoming bit stream, $s(t)$. This produces a square-wave signal with spectral components at the data rate and all harmonics, as in Figure 10.11a. As before, the appropriate spectral component can be isolated with a BPF and shaped.

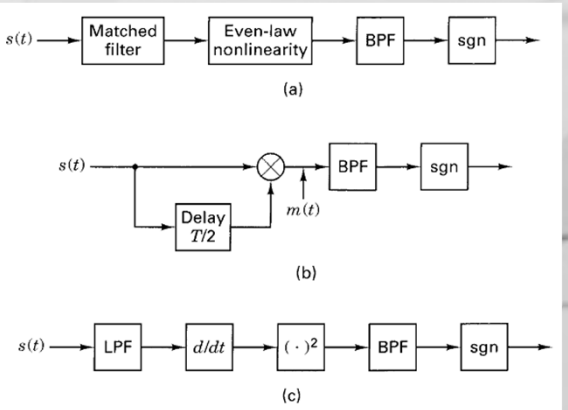


Figure 10.11 Three types of open-loop bit synchronizers.

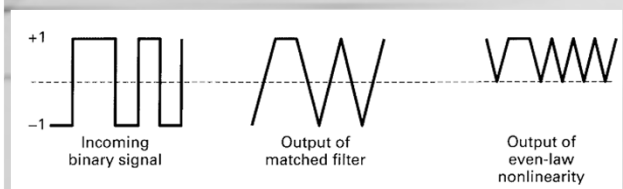


Figure 10.12 Open-loop bit synchronizer illustration.

Clearly, there will be some hardware delay associated with the signal processing steps illustrated in Figure 10.11. Wintz and Luecke [21] have shown that for a BPF that effectively averages K input symbols (bandwidth = $1/KT$), the magnitude of the fractional mean time error (delay) is approximated by

$$\frac{\overline{|\varepsilon|}}{T} \approx \frac{0.33}{\sqrt{KE_b/N_0}} \quad \text{for } \frac{E_b}{N_0} > 5, \quad K \geq 18 \tag{10.54}$$

where T is the bit period, E_b the detected energy per bit, and N_0 the single-sided received noise spectral density. Wintz and Luecke have also shown that at high signal-to-noise ratios the fractional standard deviation of the fractional timing error is given by

$$\frac{\sigma_\varepsilon}{T} \approx \frac{0.411}{\sqrt{KE_b/N_0}} \quad \text{for } \frac{E_b}{N_0} > 1 \tag{10.55}$$

Thus, for a given BPF, when the received signal-to-noise ratio is sufficiently large all of the techniques shown in Figure 10.11 will provide accurate bit timing.

10.2.2.2 Closed-Loop Symbol synchronizers

The primary disadvantage of open-loop symbol synchronization methods is that there is an unavoidable non-zero-mean tracking error. This error can be made small for large signal-to-noise ratios, but since the synchronization signal waveform depends directly on the incoming signal, the error will never vanish.

Closed-loop, symbol-data synchronizers use comparative measurements on the incoming signal and a locally generated data-clock signal to bring the locally generated signal into synchronism with the incoming data transitions. The procedure is essentially the same as that used for closed-loop carrier tracking.

Among the most popular of the closed-loop symbol synchronizers is the early/late-gate synchronizer. An example of such a synchronizer is shown schematically in Figure 10.13. The synchronizer operates by performing two separate integrations of the incoming signal energy over two different $(T-d)$ second portions of a symbol interval. The first integration (the early gate) begins integration at the loop's best estimate of the beginning of a symbol period (the nominal time zero) and integrates for the next $(T-d)$ seconds. The second integral (the late gate) delays the start of its integration for d seconds, and then integrates to the end of the symbol period (the nominal time T). The difference in the absolute values of the outputs of these two integrations, y_1 and y_2, is a measure of the receiver's symbol-timing error, and it can be fed back to the loop's timing reference to correct loop timing.

The action of the early/late-gate synchronizer can be understood by referring to Figure 10.14. In the case of perfect synchronization, Figure 10.14a shows that both gates are entirely within a signal symbol interval. In this case, both integrators will accumulate the same amount of signal, and their difference (the error signal e, in Figure 10.13) is zero. Thus, when the device is synchronized, it is stable—there is no tendency to drive itself away from synchronization. The case shown in Figure 10.14b is for a receiver whose data clock is early relative to the incoming data. In this case the first portion of the early gate falls in the previous bit interval, while the late gate is still entirely inside the current symbol. The late-gate integrator will accumulate signal over its entire $(T-d)$ integration interval, as in the case in Figure 10.14a; but the early-gate integrator will end up with energy accumulated only over $[(T-d)-2\Delta)]$, where Δ is the portion of the early-gate interval falling in the previous bit interval. Thus, for this case, the error signal will be $e=-2\Delta$, which will lower the input voltage to the VCO in Figure 10.13. This will reduce the VCO output frequency and retard the receiver's timing to bring it back toward the incoming signal's bit timing. Using Figure 10.14 as a guide, it can be seen that if the receiver's timing had been late, the amounts of energy integrated in the early gate and late gate would be reversed, as would the sign of the error signal. Thus, late receiver timing produces an increase in the VCO input voltage, increasing the output frequency and advancing the receiver's timing toward that of the incoming signal.

The example illustrated in Figure 10.14 tacitly assumes that there will be data state changes before and after the channel symbol of interest. If there are no transitions, it can be seen that the early gate and late gate will have the same integrated energy. Thus, there will be no error signal generated for cases where there is no data-state change. This is a practical implementation consideration in the use of all symbol synchronizers. Reconsider Figure 10.13. It is not possible to build two integrators that are exactly the same. Thus, the signals from the two arms of the early/late-gate loop will contain an offset with respect to each other, even when they should be identical. This offset will be small for well-designed integrators but will cause the loop to drift out of synchronism if there are long sequences of identical data symbols. There are two common responses to this problem. The first, and perhaps most obvious, is to format the data in a manner which ensures that there will be no transitionless intervals that are long enough to allow the loop to break lock. The second response is to modify the loop design so that it contains a single integrator. An example of this type of modified design is the tau-dither loop, considered in conjunction with the synchronization of spread-spectrum systems in Chapter 12.

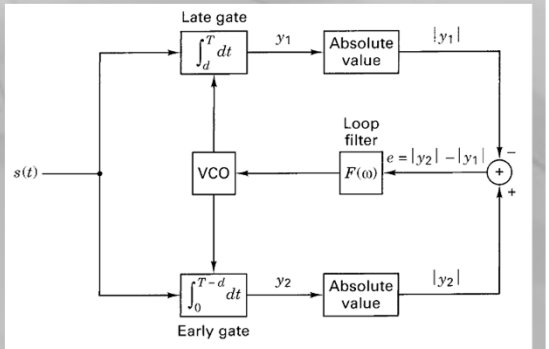


Figure 10.13 Early/late-gate data synchronizer.

Another loop design issue is the integration interval of the two gates. The example illustrated in Figure 10.14 shows the gates to occupy about three-fourths of a symbol period. Actually, this interval can vary from half a symbol interval to nearly a whole symbol interval. Why not less than half? The trade-off is between the amount of integrated noise and interference in a gate versus the amount of signal. As was true with the nonlinear model of phase-locked loops, loops of this type are difficult to analyze; the determination of performance is usually via computer simulation. This will be especially true for overlapping gates, as in Figure 10.14, because the noise samples in the two gates will be correlated. Gardner [5] has shown that for a normalized incoming signal of one volt, additive white Gaussian noise, random data (the probability of a transition is $\frac{1}{2}$), and early and late gates that are half a bit interval in duration, for large-loop signal-to-noise ratios, the fractional timing jitter is approximated by

$$\frac{\sigma_e^2}{T^2} = 2N_0B_L \tag{10.56}$$

where N_0 is the (normalized) noise power spectral density, T is the symbol interval, and B_L is the loop bandwidth.

Synchronization

10.2.2.3 Symbol Synchronization Errors and Symbol Error Performance

The effect of symbol-synchronization error on bit-error probability for a BPSK signal in additive white Gaussian noise is shown in Figure 10.15. It can be seen from the figure that the degradation is less than about 1 dB in signal-to-noise ratio for a fractional timing jitter of less than 5%. Comparing symbol timing error effects with the effect of phase noise (see Figure 10.8), it can be seen that the symbol synchronization error, taken as a fraction of the symbol interval, does not affect system performance as strongly as does phase noise taken as a fraction of a cycle. In both cases, however, the degradation increases with increases in error.

Example 10.7 Effect of Timing Jitter

Through the use of Figure 10.15, determine the effect of a 10% symbol-fractional timing jitter on a system required to maintain a 10^{-3} bit error probability.

Solution

It can be seen from Figure 10.15 that a 10^{-3}-bit-error probability will require a SNR of about 6.7 dB in the absence of all timing jitter. The same figure indicates that for a fractional timing jitter of 10% ($\sigma_\epsilon / T = 0.1$), a SNR of about 12.9 dB is required. Thus, the ability to accommodate this large timing jitter would require a 6.2-dB higher signal-to-noise ratio than that needed to maintain a 10^{-3}-bit-eror probability without jitter. This illustrates a use to which Figure 10.15 can be put; however, this example is clearly extreme. No communication system would be designed with over four times the nominally required power level in order to accommodate a large symbol-synchronization error. Some other answer would be found, such as redesigning the system filtering to increase the value of K in Equation (10.55), which will reduce the symbol timing jitter.

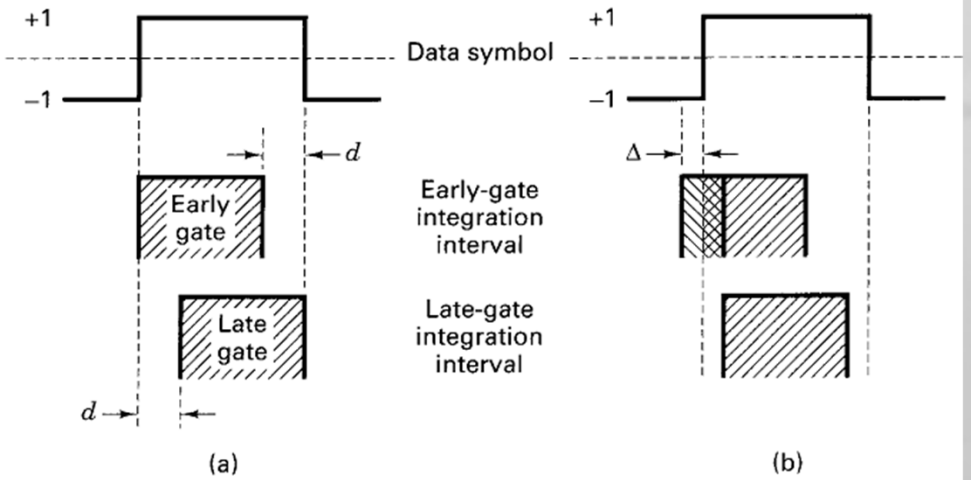


Figure 10.14 (a) Correct receiver timing. (b) Early receiver timing.

10.2.3 Synchronization with Continuous-Phase Modulations (CPM)

10.2.3.1 Background

Continuous-Phase Modulations (CPM) have grown from a research topic to an increasingly important signaling technique because of their bandwidth efficiency. As bandwidth becomes increasingly dear, their importance will continue to increase. These modulations raise new issues in synchronization, especially symbol synchronization. The bandwidth efficiency of CPM is obtained by increasing the smoothness of the waveforms in the time domain. If done properly, this smoothness will concentrate the signal's energy in a narrower bandwidth, reducing the amount of bandwidth required to pass the signal, and allowing adjacent signals to be packed closer together. However, this smoothness in the time domain also tends to eliminate the symbol transition features upon which many symbol synchronization schemes depend. A related problem is that, in general, it is difficult with CPM to separate the effects of carrier-phase error from symbol-timing error, making the phase and timing tasks interrelated. A mitigating feature of the smoothed waveforms is that in most cases of practical interest, the performance of the receivers is relatively insensitive to moderate timing errors [3].

A normalized CPM signal can be represented in complex notation as

$$s(t) = \exp\{j[\omega_0 t + \theta + \psi(t - \tau, \alpha)]\} \tag{10.57}$$

where ω_0 is the carrier frequency, θ is the carrier phase (measured relative to the phase of the receiver), and $\psi(t, \alpha)$ is called the *excess phase* of $s(t)$. It is $\psi(t, \alpha)$ that carries the information in the signal. It is also $\psi(t, \alpha)$ that determines the amount of bandwidth the signal will need. The required bandwidth is sometimes referred to as the *bandwidth occupancy* of the signal. When considering the goal of reducing or minimizing the required bandwidth from the standpoint of Fourier Theory, it can be seen that relatively high frequency components are associated with relatively abrupt changes in the time domain signal [22]. Thus, in order to reduce or eliminate the high frequency components, one must smooth out all the rough edges or abrupt changes in the time domain signal. In CPM signaling, this is accomplished through a combination of three techniques:

1. Use signal pulses that have several orders of continuous derivatives.
2. Allow individual signal pulses to occupy multiple signal time intervals (i.e., intentionally inject some intersymbol interference between symbols).
3. Reduce the maximum allowed phase change per symbol interval.

Not all CPM schemes use all of these techniques, but they all use some. For CPM schemes, it should be noted that at the beginning of each symbol interval, the excess phase, $\psi(t, \alpha)$, is a Markov Process [4], in that it depends only on the phase at the beginning of the symbol and the current symbol value. The phase value at the beginning of the symbol is a consequence of some number of previous symbols. Therefore, for the practical case where there are a finite number of possible phase states, the result is a finite state channel. Thus, the excess phase can be defined as

$$\psi(t, \alpha) = \eta(t, \mathbf{C}_k, \alpha_k) + \Phi_k \qquad kT \le t \le (k+1)T \tag{10.58}$$

where

$$\eta(t, \mathbf{C}_k, \alpha_k) = 2\pi h \sum_{i=k-L+1}^{k} \alpha_i q(t - iT) \tag{10.59}$$

$\mathbf{C}_k$ is called the correlative state, k is a time index, and α_k is the kth information symbol drawn from the alphabet $\{\alpha_k\} = \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$. This alphabet allows for the general case of M-ary (rather than just binary) signaling. The parameter h is the modulation index, and $q(t)$ is called the *modulation phase response*, defined outside of the region $0 < t < LT$, as

$$q(t) = \begin{cases} 0 & \text{for } t \le 0 \\ 1/2 & \text{for } t \ge LT \end{cases} \tag{10.60}$$

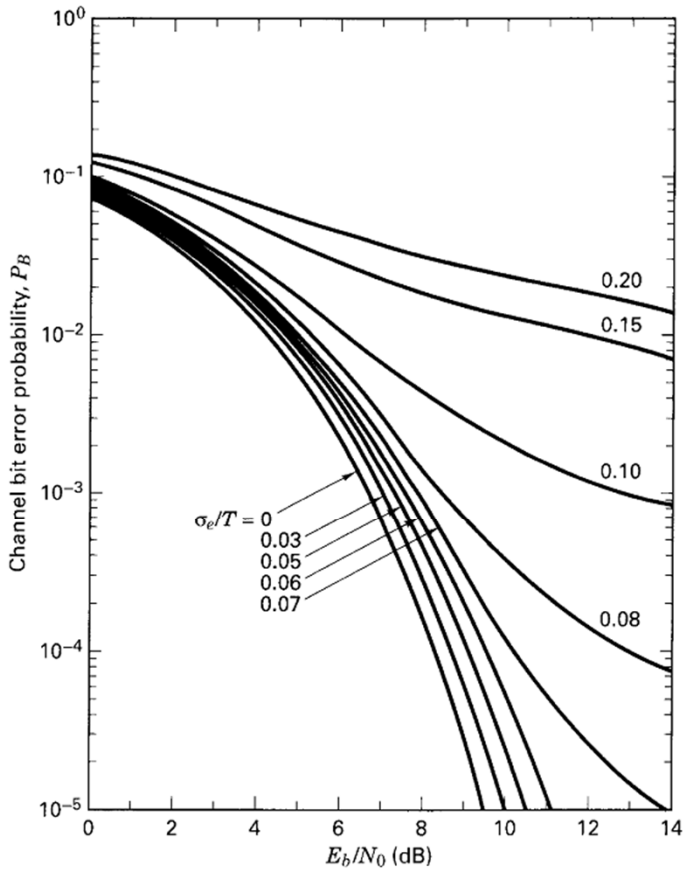


Figure 10.15 Channel bit error probability versus E_b/N_0 with the standard deviation of the symbol sync error σ_e as a parameter. (Reprinted from W. C. Lindsey and M. K. Simon, *Telecommunication Systems Engineering,* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1973, courtesy of W. C. Lindsey and Marvin K. Simon.)

where L is called the correlation length. The correlation length is the number of information-symbol periods of length T seconds that are affected by a single information symbol. This is a measure of the amount of intentional intersymbol interference. When $L = 1$, we refer to the signaling as *full response.* This is the condition that was assumed in earlier chapters dealing with modulation. In full-response signaling, each pulse is confined to its own time boundaries. However, when $L > 1$, the signaling is called *partial response,* which means that each pulse is not restricted to its own symbol interval, but rather it is "smeared" into $L - 1$ neighboring symbol intervals. This is the vehicle used in many CPM schemes for purposely injecting controlled intersymbol interference between symbols and thereby increasing bandwidth efficiency. Classical minimum-shift-keying (MSK), one of the earliest examples of CPM (see Chapter 9), does not use multiple symbol intervals per pulse. Thus, classical MSK is an example of full-response signaling. Observing Equations (10.60), it can be seen that the consequence of $q(LT) = \frac{1}{2}$ is that the maximum possible phase change over an LT interval is $(M - 1)\,\pi h$, as can be seen from Equation (10.58) and (10.59).

The vector $\mathbf{C}_k$, which is called the *correlative state,* is a sequence of information symbols $\{\alpha_k\}$, starting from the earliest time that can affect the signal's phase at the current time k, as follows:

$$\mathbf{C}_k = (\alpha_{k-L+1}, \cdots, \alpha_{k-2}, \alpha_{k-1})$$

The term Φ_k in Equation (10.58), expressed as

$$\Phi_k = \pi h \sum_{i=0}^{k-L} \alpha_i \mod 2\pi \tag{10.61}$$

is called the *phase state.* The phase state is one of a set of discrete phases that the signal can take as a consequence of the values of past symbols. Starting the phase transition for the next symbol from this phase state is a necessary condition for continuous phase. In the context of a trellis diagram, Φ_k can be viewed as an initial state or node, and $\mathbf{C}_k$ as defining a path to one of the other nodes. The definition of $q(t)$ in the interval $(0 < t < LT)$ is what gives the modulation its particular characteristics. MSK has the parameters $h = \frac{1}{2}$, $L = 1$, $M = 2$ and $q(t) = t/(2T)$ in the interval $(0 < t < T)$. The frequency response, defined as $g(t) \triangleq dq(t)/dt$, is clearly rectangular for MSK:

$$g(t) = \begin{cases} 1/(2T) & 0 \le t \le T \\ 0 & t < 0, \quad t > T \end{cases} \tag{10.62}$$

Gaussian MSK (GMSK), another example of CPM, is defined as having a frequency response that is the convolution of this rectangle with a Gaussian shaped pulse.

Synchronization

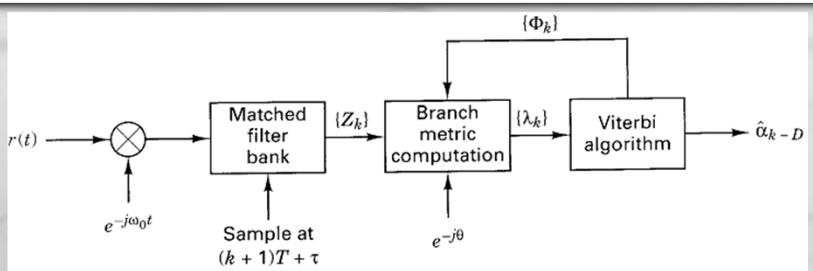


Figure 10.16 CPM receiver structure.
(Note: $\hat{\alpha}_{k-D}$ is the k-th output symbol with processing delay, D)

Many of the synchronization techniques we have described in previous sections are based on *ad hoc* methods. People who had gained a fair amount of intuition in synchronization invented them by trying things that seemed to make sense. With a few exceptions, intuition has been less successful with CPM. Most techniques are based on classical estimation theory principles, the most popular being maximum likelihood estimation. The principles involved are the same as those developed for maximum likelihood signal detection.

Maximum likelihood estimation involves the maximization of conditional probabilities, and it is based on Baysian Theory [7]. Let $s(t, \gamma)$ represent a signal with some set of unknown parameters, γ. The parameters could be carrier phase, or symbol-timing offset, or the values of the transmitted information symbols, or possibly other parameters. Let

$$r(t) = s(t, \gamma) + n(t) \tag{10.63}$$

represent a received signal, where $n(t)$ is some additive receiver noise process. Let $R(t)$ represent a realization of the process $r(t)$. Then the maximum-likelihood estimate for the unknown parameter set γ is the value of γ that maximizes the likelihood $p[r(t) = R(t)|\,\gamma]$ over all γ. As has been developed in Chapter 3, a realization of a maximum-likelihood detector for a known signal is a filter matched to that signal. For the case of CPM, this leads to a receiver structure as shown in Figure 10.16.

In the primary signal detection process, the carrier frequency ω_0, carrier phase θ, and symbol timing offset τ are assumed known. The receiver structure is effectively a bank of matched filters, each filter matched to an L-symbol signal realization, which feeds a Viterbi Algorithm. The number of filters is M^L, and the number of nodes in the branch metric computation is PM^{L-1}, where P is the number of phase states $\{\Phi_k\}$. Both of these numbers can be awkwardly large, and so simpler receiver structures are often used in practice [3, 4, 22]; however, the structure is still useful as a basis for synchronization.

From the preceding CPM description, the individual filters in the filter bank will have impulse responses given by

$$h^{(\ell)}(t) \triangleq \begin{cases} e^{-j\eta_\ell(T-t, \mathbf{C}_0^{(\ell)}, \alpha_0^{(\ell)})} & 0 \le t \le T \\ 0 & \text{elsewhere} \end{cases} \tag{10.64}$$

with $(\ell = 1, 2, \ldots, M^L)$ denoting the generic L-symbol string $(\mathbf{C}_0^{(\ell)}, \alpha_0^{(\ell)}) = (\alpha_{-L+1}^{(\ell)}, \ldots, \alpha_{-1}^{(\ell)}, \alpha_0^{(\ell)})$, where each $\alpha_k^{(\ell)}$ is selected from the signal alphabet and ℓ denotes a particular path (symbol sequence) in the set of M^L possible paths. Similar to the earlier notation,

$$\eta_\ell(t, \mathbf{C}_0^{(\ell)}, \alpha_0^{(\ell)}) = 2\pi h \sum_{i=-L+1}^{0} \alpha_i^{(\ell)} q(t - iT) \tag{10.65}$$

From Figure 10.16, it can be seen that the individual filter outputs are given by

$$Z_k^{(\ell)}(\mathbf{C}_k, \alpha_k, \tau) \triangleq \int_{\tau+kT}^{\tau+(k+1)T} r(t)\, h^{(\ell)}(t - \tau - kT) e^{-j\omega_0 t}\, dt \tag{10.66}$$

This set of outputs $\{Z_k\}$, along with the carrier phase estimate $\hat{\theta}$ and the phase state $\{\Phi_k\}$, are used to compute the path metrics and, ultimately, to determine the Viterbi Algorithm's decisions.

10.2.3.2 Data-Aided Synchronization

Techniques for synchronizing CPM receivers can be divided into those that rely on knowledge of the information symbols, and those that do not. Those that rely on such knowledge are called data-aided techniques. Those that do not are termed non-data-aided (NDA). Clearly this is a distinction that could be applied to receiver synchronization for all modulation techniques, but data aided techniques appear especially useful and popular with CPM. There are two ways in which the data symbols could be known: either the symbols under consideration are a part of a known header or training sequence inserted into the data stream, or decisions from the Viterbi Algorithm are being fed back into the synchronization process. If decision feedback is to be employed, clearly the decisions must be fairly reliable, which implies that the receiver must be close to lock. Initial acquisition based on decision feedback is not likely to be a practical approach.

Given that the transmitted symbol string is known over some observation interval, L_0, the index (ℓ) in Equation (10.66) can be dropped. With the usual assumptions of Gaussian noise processes and equal energy signals, the likelihood function, $\Lambda(R|\,\hat{\theta}, \hat{\tau})$, associated with θ and τ, the unknown phase and time offsets, respectively, is given by [3]

$$\Lambda(R|\,\hat{\theta}, \hat{\tau}) = \exp\left\{\sum_{k=0}^{L_0-1} \mathrm{Re}\left[Z_k(\mathbf{C}_k, \alpha_k, \hat{\tau})e^{-j(\hat{\theta}+\Phi_k)}\right]\right\} \tag{10.67}$$

where unimportant constant factors have been dropped and Re{·} refers to the real part of the complex argument. It is clear that the right-hand side of Equation (10.67) will be maximized when the summation is maximized. Therefore, taking the partial derivatives of the summation with respect to $\hat{\theta}$ and $\hat{\tau}$ and then setting the results to zero yields

$$\sum_{k=0}^{L_0-1} \mathrm{Im}\left[Z_k(\mathbf{C}_k, \alpha_k, \hat{\tau})e^{-j(\hat{\theta}+\Phi_k)}\right] = 0 \tag{10.68}$$

and

$$\sum_{k=0}^{L_0-1} \mathrm{Re}\left[Y_k(\mathbf{C}_k, \alpha_k, \hat{\tau})e^{-j(\hat{\theta}+\Phi_k)}\right] = 0 \tag{10.69}$$

where $Y_k = \partial Z_k/\partial\hat{\tau}$, and Im{·} refers to the imaginary part of the complex argument. Mengali [3] points out that the left-hand side of Equation (10.69) can be obtained in two ways: either by taking the derivative in the straightforward manner, or by implementing a set of "derivative filters." Which is the preferred implementation would depend on the exact case at hand.

Unfortunately, Equations (10.68) and (10.69) do not provide much in the way of intuition or insight, and there are no known closed-form solutions. The equations must be solved numerically through some jointly iterative procedure for $\hat{\theta}$ and $\hat{\tau}$. Mengali suggests an iterative procedure, where the successive terms in each summation are used to form error terms for a successive approximation. That is,

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \gamma_P e_P(k-1) \tag{10.70}$$

$$\hat{\tau}_{k+1} = \hat{\tau}_k + \gamma_T e_T(k-1) \tag{10.71}$$

where e_P and e_T are higher order terms from the left-hand sides of Equations (10.68) and (10.69), respectively, and γ_P and γ_T are "gains," selected to assure that the process converges. Clearly, this iterative procedure is more appropriate in a decision-feedback operation than with a fixed-length training sequence.

10.2.3.3 Non-data-Aided Synchronization

One of the first tenets of Information Theory is that having more information is better than having less. In the current context this means that knowing the symbol sequence will allow better estimates of carrier phase and symbol timing than not knowing. There may be cases, however, where training sequences are impractical or inconvenient, and the decision process is not sufficiently reliable for decision feedback. In these cases, non-data-aided (NDA) synchronization processes are called for. Two techniques of general applicability will be discussed, and one power-law technique that can be applied in a large number of cases.

The first technique is a direct extension of the development in the previous section. Clearly, if the symbol sequence $(\mathbf{C}_k, \alpha_k)$ is not known, a new likelihood function similar to Equation (10.67) can be written to accommodate that fact:

$$\Lambda(R|\,\hat{\mathbf{C}}_k, \hat{\alpha}_k, \hat{\theta}, \hat{\tau}) = \exp\left\{\sum_{k=0}^{L_0-1} \mathrm{Re}\left[Z_k(\hat{\mathbf{C}}_k, \hat{\alpha}_k, \hat{\tau})e^{-j(\hat{\theta}+\Phi_k)}\right]\right\} \tag{10.72}$$

Since the likelihood function is proportional to a conditional probability, the chain rule of conditional probabilities can be applied to get back to a likelihood function dependent on $\hat{\theta}$ and $\hat{\tau}$ alone. The chain rule states that [7]

$$p(r(t) = R(t)|\gamma) = \int_{\text{all }\beta} p[r(t) = R(t)|\gamma, \beta]\, p(\beta)\, d\beta \tag{10.73}$$

which implies that the desired likelihood function is given by

$$\Lambda'(R|\,\hat{\theta}, \hat{\tau}) = \frac{1}{M^L} \sum_{\text{all }\{\hat{C}_k, \hat{\alpha}_k\}} \Lambda(R|\,\hat{\mathbf{C}}_k, \hat{\alpha}_k, \hat{\theta}, \hat{\tau}) \tag{10.74}$$

where the assumption has been made that all symbol sequences are equally likely. The likelihood function on the right-hand side of Equation (10.74) can now be differentiated to produce two equations analogous to Equations (10.68) and (10.69). Clearly, the result is considerably more computationally complicated than the results in Equations (10.68) and (10.69), and simpler if sub-optimal techniques may be desired. Mengali [3] discusses some approximations which lead to a somewhat simplified estimator for $\hat{\tau}$.

Another approach is indicated by the form of a sub-optimal receiver structure using Laurent filters [23, 24]. This approach approximates the CPM signal by a set of superimposed pulse amplitude modulated (PAM) waveforms. Considering only the first in the series, the expression is

$$e^{j\psi(t,\,\alpha)} \approx \sum_i a_{0,i}\, h_0(t - iT) \tag{10.75}$$

where $\psi(t, \alpha)$ is defined in Equation (10.58) and the coefficients $a_{0,i}$ are called *pseudo-symbols.* The pseudo-symbols, whose values depend on the past and present data symbols, are defined by

$$a_{0,i} = \exp\left(j\pi h \sum_{l=0}^{i} \alpha_l\right) \tag{10.76}$$

where the modulation index h can take on any noninteger value. For the important special case of MSK, where $h = \frac{1}{2}$, the expression in Equation (10.75) is exact for a filter function of the form

$$h_0(t) = \begin{cases} \sin\left(\dfrac{\pi t}{2T}\right) & 0 \le t \le 2T \\ 0 & \text{elsewhere} \end{cases} \tag{10.77}$$

For other modulations, the approximation will be more or less accurate, and the form of $h_0(t)$ will vary [23]. In any case, ignoring the noise process momentarily, the normalized signal can now be rewritten in the form

$$s(t) \approx e^{j(\omega_0 t + \theta)} \sum_i a_{0,i} h_0(t - iT - \tau) \tag{10.78}$$

From this expression, it is clear that the standard techniques for phase and symbol timing that were developed in the previous sections for linear modulations could be applied to this approximation. Mengali [3] points out that care must be taken in following this approach, however, because a filter actually matched to $h_0(t)$ might produce a very poorly shaped pulse. The issue is discussed by Kaleh [25].

Finally, in the special cases where the modulation index is rational, $h = k_1/k_2$, with (k_1, k_2) integers, power-law techniques can be applied [22]. For this case, Equation (10.57) can be rewritten as

$$s(t) = \exp\left\{j\left[\omega_0 t + \theta + 2\pi \frac{k_1}{k_2} \sum_{i=k-L+1}^{k} \alpha_i q(t - iT)\right]\right\} \tag{10.79}$$

where, for simplicity, Φ_k from Equation (10.58) has been absorbed into θ. Taking the k_2^{th} power of $s(t)$ gives

$$[s(t)]^{k_2} = \exp\left\{j\left[k_2(\omega_0 t + \theta) + 2\pi k_1 \sum_{i=k-L+1}^{k} \alpha_i q(t - iT)\right]\right\} \tag{10.80}$$

10.2.4 Frame Synchronization

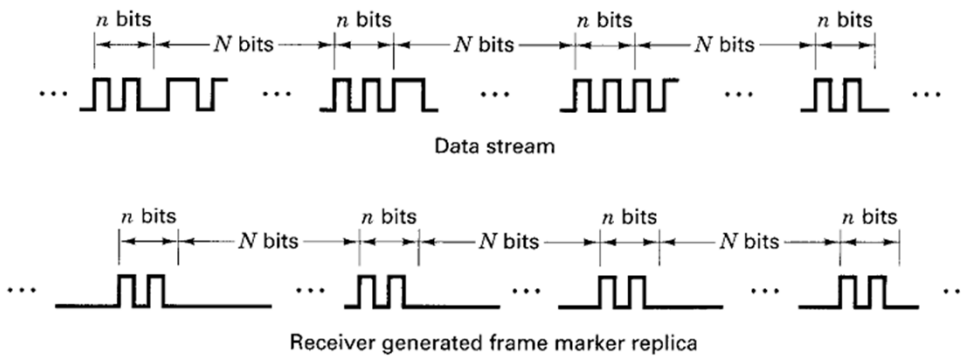
Almost all digital data streams have some sort of frame structure. This is to say that the data stream is organized into uniformly sized groups of bits. If the data stream is digitized TV, each pixel is represented by a word having several bits, which is further organized into horizontal raster scans, which is further organized in terms of vertical raster scans. Computer data are typically organized into words of some number of 8-bit bytes, and these, in turn, are organized into card images, packets, frames, or files. Any system that uses block-error control coding must be organized around the codeword length. Digital speech is typically transmitted in packets or frames which are indistinguishable from other digital data.

For a receiver to make sense of the incoming data stream, the receiver needs to be synchronized with the data stream's frame structure. Frame synchronization is usually accomplished with the aid of some special signaling procedure from the transmitter. This procedure may be very simple, or fairly involved, depending on the environment in which the system is required to operate.

Probably the simplest frame synchronization aid is the frame marker, illustrated in Figure 10.17. The frame marker is a single bit, or a short pattern of bits that the transmitter injects periodically into the data stream. The receiver must know the pattern and the injection interval. The receiver, having achieved data synchronization, correlates the known pattern with the incoming data stream at the known injection interval. If the receiver is not in synchronization with the framing pattern, the accumulated correlation will be low. When the receiver comes into frame synch, however, the correlation should be nearly perfect, blemished only by an occasional detection error.

The advantage of the frame marker is its simplicity. Even a single bit can suffice as a frame marker if a sufficient number of correlations are accumulated before deciding whether or not the system has achieved synchronization. The major drawback is that the sufficient number may be very large, and thus the expected time required to acquire synchronization would be long. Therefore, frame markers are most useful in systems that transmit data continuously, like many telephony and computer links, and would be inappropriate for systems that transmit in isolated bursts or systems that require rapid frame acquisition. A secondary drawback is that the inserted bit(s) may make the organization of the data stream awkward.

An example is the T1 carrier stream developed by Bell Labs and in general use in North American telephony systems. The T1 carrier structure includes a single bit frame marker after each set of 24 8-bit bytes, each byte representing one of 24 possible voice data streams. This yields a data structure that is an integer multiple of 193 bits—an unhandy number from the standpoint of most integrated circuits.



An approach for systems with inconsistent or bursty transmissions, or systems with rapid acquisition requirements, is a synchronization codeword. A synchronization codeword would typically be sent as part of a message header. The receiver must know the codeword and be constantly searching for it in the data stream, possibly with a matched filter correlator. Detection of the codeword would indicate a known position (typically the beginning) in the data frame. The advantage of this system is that frame acquisition can be essentially immediate. The only delay would be that required to process the incoming codeword. The disadvantage is that the codeword must be long relative to the frame marker, to keep the probability of false detections low. The complexity of the correlation operation is proportional to the length of the sequence, so the correlator may be relatively complicated.

A good synchronization codeword is one that has the property that the absolute value of its "correlation sidelobes" is small. A correlation sidelobe is the value of the correlation of a codeword with a time-shifted version of itself. Thus, the correlation sidelobe value for a k-symbol shift of an N-bit code sequence $\{X_i\}$ is given by

$$C_k = \sum_{j=1}^{N-k} X_j X_{j+k} \tag{10.81}$$

where X_i $(1 \le i \le N)$ is an individual code symbol taking values ± 1 and the adjacent data symbols (associated with index value $i > N$) are assumed to be zero. An example of correlation sidelobe computation is shown in Figure 10.18. The 5-bit

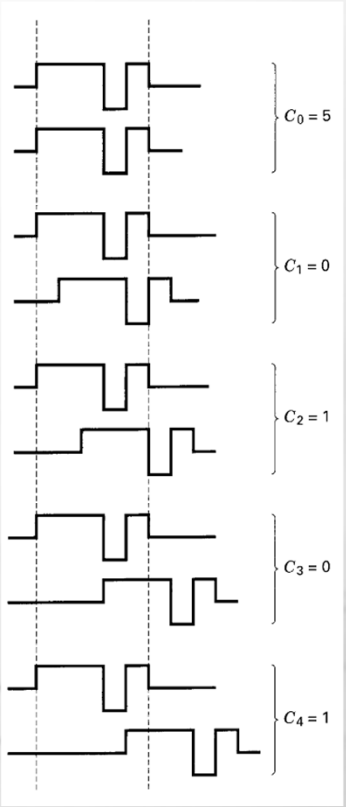


Figure 10.18 Correlation sidelobe example.

sequence in the example is seen to have good correlation properties, in that the largest sidelobe is one-fifth of the main lobe, C_0. Sequences like the example in Figure 10.18, with the property that their largest sidelobe has a magnitude of unity, are known as Barker sequences or Barker words [26]. There is no known constructive method for finding Barker words, and only 10 unique words are known, the longest of which has 13 symbols. The known unique Barker words are given in Table 10.1. Some thought should make it clear that a completely exhaustive list of known Barker sequences would include those sequences produced by inverting the sign of the symbols and those produced by reversing the time ordering of the symbols in the sequences of Table 10.1.

The sidelobe correlation properties of Barker codes are based on the assumption that the adjacent symbols have zero value. This is an approximation to the effect of equally likely random binary data adjacent to the Barker word, taking the values ±1. Unfortunately, the Barker sequences are too short for this approximation to provide the best codeword in random binary data in all cases. Willard [27] found the best sequences in terms of the minimum probability of false synchronization for random adjacent symbols for the Barker word lengths by the use of a computer simulation. The Willard sequences are shown in Table 10.2.

Two probabilities characterize the performance of a system using a synchronization word. These are the probability of a missed detection and the probability of false alarm. Clearly, the system designer would wish both probabilities to be as small as possible. These are conflicting desires. In order to decrease the probability of a miss, the system design may allow less-than-perfect correlation of an incoming synchronization word. That is, a word may be accepted even if it contains a small number of errors. This, however, enlarges the number of symbol patterns that will be accepted and thereby increases the probability of a false alarm. The probability of a miss for an N-bit word where k or fewer errors are accepted is given by

$$P_m = \sum_{j=k+1}^{N} \binom{N}{j} p^j (1-p)^{N-j} \tag{10.82}$$

where p is the probability of a detector bit error. The probability of a false alarm generated by N bits of random data is given by

$$P_{FA} = \sum_{j=0}^{k} \frac{\binom{N}{j}}{2^N} \tag{10.83}$$

It can be seen that for small p, P_m will decrease roughly exponentially with increasing k. Unfortunately, P_{FA} increases roughly exponentially with increasing k. To obtain acceptable values of both P_m and P_{FA} for a given value of p, the system designer often needs values of N larger than those provided by the Barker and Willard sequences. fortunately, there is a fairly large body of literature dealing with longer sequences. Most of these sequences were discovered through exhaustive computer searches. Spilker [20] lists sequences of up to $N = 24$ found by Newman and Hofman [28] and mentions that their original paper had sequences to $N = 100$. Wu [29] provides a list of Maury-Styles sequences to length $N = 30$, a list of Linder sequences to length 40. He also provides a fairly complete discussion of the topic of synchronization sequences, including constructive techniques for reasonable but non-optimum sequences, and insight into the frame synchronization procedures of some operational satellite digital communication systems.

TABLE 10.1 Barker Synchronization Codewords

N	Barker Sequence
1	+
2	++ or +−
3	++−
4	+++− or ++−+
5	+++−+
7	+++−−+−
11	+++−−−+−−+−
13	+++++−−+++−+−+

TABLE 10.2 Willard Synchronization Codewords

N	Willard Sequences
1	+
2	+−
3	++−
4	++−−
5	++−+−
7	+++−+−−
11	+++−++−+−−−
13	+++++−−+−+−−−

Synchronization

10.3 NETWORK SYNCHRONIZATION

For systems using coherent modulation techniques, one-direction communications such as broadcast channels, or single-link communications, such as most microwave links, land-line links, or fiber-optics links, the synchronization architecture that makes the most sense is to make synchronization totally a receiver function. For communications systems using noncoherent modulation techniques, or that involve many users accessing a central communication node, such as many satellite communication systems, it often makes sense for synchronization to be mostly or entirely a terminal function. This means that the terminal transmitter parameters are modified to achieve synchronization, rather than modifying the central node's receiver parameters. This must be the approach if the system uses time-division multiple access (TDMA). In TDMA, each user is allotted a segment of time in which to transmit its information. The terminal transmitter must be synchronized with the system in order for its transmitted burst of data to arrive at the central node at the time when the node is prepared to receive the data. Synchronization of the terminal transmitter also makes sense with systems that combine signal processing at the central node with frequency-division multiple access (FDMA). If the terminals precorrect their transmission to be synchronized with the central node, the node can use a fixed set of channel filters and a single timing reference for the processing of all channels. Otherwise, the node would require a separate time and frequency acquisition and tracking capability for each incoming channel, and it would need to deal with the possibility of varying amounts of adjacent channel interference. It seems clear that terminal transmitter synchronization is often the cleaner, more reasonable system approach to synchronizing a network.

Transmitter synchronization procedures may be classified s being either open loop or closed loop. Open-loop techniques do not rely on any measurement of the arriving signal parameters at the central node. The terminal pre-corrects its transmission based on stored knowledge of link parameters that have been provided by some external authority but may possibly be modified by observations of a return signal from the central node. Open-loop techniques rely on link parameters being accurately known and predictable. They work best when link geometry is nominally fixed, and the links themselves operate continually for relatively long periods, once established. They tend to be difficult to use efficiently when the link geometry is not static or when the terminals access the system sporadically.

The main advantages of the open-loop methods are that acquisition is fast—the procedure can work without a return link, and the amount of real-time computation that is required is small. The disadvantages of open-loop methods are that they require the existence of the external authority that provides knowledge of the required link parameters, and that they are relatively inflexible. The lack of any direct real-time measure of system characteristics means that the system cannot adjust quickly to any unplanned change in conditions.

Closed-loop techniques, on the other hand, require little in the way of *a priori* knowledge of link parameters. Knowledge would be useful in reducing the time required for acquisition, but need not be precise as is required by open-loop methods. Closed-loop methods involve measurements of the synchronization accuracy of the incoming transmissions from the terminal upon their arrival at the central node, and the return of the results of these measurements to the terminal via a return path. Thus, closed-loop methods require a return path that provides a response to the terminal's transmission, the ability in the terminal to recognize the response for what it is, and the ability in the terminal to modify the transmitter characteristics appropriately, based on the response. This amounts to a requirement for a relatively large amount of real-time processing in the terminal, and two-way links between every terminal and the central node. The disadvantages of closed-loop methods are that they require a relatively large amount of real-time processing, require two-way links to every terminal, and that acquisition can take a relatively long time. The advantages are that no external source of knowledge is required for the system to work, and the responses on the return link allow the system to adapt easily and quickly to changing geometries and link conditions.

10.3.1 Open-Loop Transmitter Synchronization

Open-loop systems can be further subdivided into systems that employ information gained by observing a return link, and those that do not. Those that do not are the simplest of all, in terms of real-time processing requirements, but communication performance for these simple terminals is clearly very dependent on stable link characteristics.

All transmitter synchronization schemes attempt to precorrect the timing and transmission frequency of the signal in such a manner that the signal will arrive at a receiver with the expected frequency and at the expected time. Thus, to precorrect time, a transmitter would divide the distance between itself and the receiver by the speed of light to get the transmission transit time, and then shift the message transmission timing that much ahead. By transmitting the signal early, it will arrive at the receiver at the appropriate time. The time of arrival at the node is given by

$$T_A = T_t + \frac{d}{c} \tag{10.84}$$

where T_t is the actual transmission start time, d is the transmit distance, and c is the speed of light. Similarly, to precorrect the transmission frequency, the transmitter must allow for the Doppler shift caused by relative motion between the transmitter and the intended receiver. To be received correctly, the required transmission radian frequency is

$$\omega \approx \left(1 - \frac{V}{c}\right)\omega_0 \tag{10.85}$$

where c is the speed of light, V is the relative velocity (positive for decreasing transmission distance), and ω_0 is the nominal transmission radian frequency.

Unfortunately, in practice, neither the time nor the frequency precorrection can be done exactly. Even satellites in nominally geostationary orbits move slightly with respect to a point on the earth, and the behavior of the time and frequency references in the terminal and the central node are never entirely predictable. Thus, there will always be some time and frequency precorrection error. The time error may be expressed as

$$T_e = \frac{r_e}{c} + \Delta t \tag{10.86}$$

where r_e is the error in the range estimate and Δt is the difference between the time reference at the terminal and the reference at the receiver. The frequency error may be expressed as

$$\omega_e = \frac{V_e \omega_0}{c} + \Delta\omega \tag{10.87}$$

where V_e is the error in the measured or predicted relative velocity of the transmitter and receiver (the Doppler error) and $\Delta\omega$ is the frequency difference between the transmitter and the receiver frequency references. There are many other sources of time and frequency error in addition to those mentioned here, but they are typically much less important. Spilker [20] gives a reasonably complete accounting of sources of time and frequency error for satellite systems.

The error terms Δt and $\Delta\omega$ are typically due to random fluctuations in frequency references. The time reference for a transmitter or receiver is generally obtained by counting cycles of the frequency reference, so errors in the accuracy of the time and frequency references are related. The fluctuations in a frequency reference are very difficult to characterize statistically, although the power spectral density of he fluctuations is approximated by a sequence of power-law segments [15]. Frequency references are often specified in terms of a maximum allowable fractional frequency change per day:

$$\delta = \frac{\Delta\omega}{\omega_0} \quad \text{hertz/hertz/day} \tag{10.88}$$

Typical values for δ range from 10^{-5} to 10^{-6}, for inexpensive crystal oscillators, to 10^{-9} to 10^{-11}, for high-quality crystal oscillators; to 10^{-12} for rubidium standards, to 10^{-13} for cesium standards. An effect of specifying system-frequency references by the maximum fractional frequency is that if there is no intervention, the offset form the nominal frequency ω_0 can grow linearly with time:

Synchronization

$$\Delta\omega(T) = \omega_0 \int_0^T \delta\, dt + \Delta t(0) = \omega_0\, \delta T + \Delta\omega(0) \quad \text{hertz} \tag{10.89}$$

For a cycle-counting time reference, however, the cumulative time offset is related to the cumulative phase error of the reference:

$$\begin{aligned}\Delta t(T) &= \int_0^T \frac{\Delta\omega(t)}{\omega_0}\, dt + \Delta t(0) \\ &= \int_0^T \delta t\, dt + \int_0^T \frac{\Delta\omega(0)}{\omega_0}\, dt + \Delta t(0) \\ &= \tfrac{1}{2}\delta T^2 + \frac{\Delta\omega(0)T}{\omega_0} + \Delta t(0)\end{aligned} \tag{10.90}$$

Thus, without intervention, a time-reference error can grow quadratically with time. For open-loop transmitter synchronization systems, this quadratic growth in time error often sets limits on how often the external authority must intervene, either to update the terminal's knowledge of receiver timing, or to reset both the receiver's and the transmitter's time references to nominal. The quadratic error growth usually means that timing errors are more of an operational problem than are frequency errors, although this will depend on the system design.

If the transmitter does not have information from measurements on a return link, the time and frequency offsets as modeled by Equations (10.86) to (10.90) will allow a system designer the ability to determine the maximum interval between interventions on the basis of a probability-of-error criterion. Time- and frequency-reference recalibration is often a burdensome procedure; it should be done rarely as possible.

If a terminal has access to a return link from the central node and the ability to make comparative measurements between the local reference and incoming signal parameters, the interval between recalibrations can be made much longer. Large satellite control stations can measure and model the orbital parameters of nominally geostationary satellites to an accuracy of a few tens of feet in range and a few feet/second in velocity relative to the ground terminal. Thus, for the important special case of a synchronous satellite as the central node, the first terms on the right-hand side of Equations (10.86) and (10.87) are usually negligible. When this is true, the differences between the incoming signal parameters and those generated by the terminal's time and frequency references will approximate the error terms Δt and $\Delta\omega$. These error terms measured on the downlink can be used to compute appropriate corrections to the uplink transmissions. On the other hand, if the time and frequency references are known to be accurate but the link geometry is somewhat in question—perhaps because the terminal is mobile or the satellite is non-geostationary—the same sort of return link measurement could be used to resolve range or velocity uncertainties. These measures of range or relative velocity can then be used to precorrect uplink timing and frequency.

The case where a terminal is able to utilize measurements made on a return link signal is sometimes called quasi-closed-loop transmitter synchronization. The quasi-closed technique is clearly more adaptable to uncertainties in the communication system than is the purely open-loop system. The purely open-loop system requires complete *a priori* knowledge of all important link parameters in order to operate successfully. Unanticipated changes in the links cannot be tolerated. The quasi-closed-loop system, on the other hand, requires *a priori* knowledge of all but one of the important parameters in each of time and frequency, but the remaining term can be determined from observations of the return link. This adds complexity to the terminal, but it also adds the ability to adapt to certain types of unplanned link changes. This degree of adaptability can greatly reduce the frequency of required system calibration.

10.3.2 Closed-Loop Transmitter Synchronization

Closed-loop transmitter synchronization involves the transmission of special synchronization signals that are used to determine the signal's time or frequency error relative to the desired timing or frequency when the signal arrives at the receiver. The results of this determination are then fed back to the transmitter on a return link. The determination of synchronization errors can be either implicit or explicit. If the central node has sufficient processing capacity, the central node may make an actual error measurement. Such a measurement might be the amount and direction of offset, or perhaps simply the direction alone. This information would be formatted and returned to the transmitter on a return link. If the central node has little processing capability, the special synchronization signal may simply be turned around and returned to the transmitter on the return link. In this case, it becomes part of the transmitter's task to interpret the returned signal for itself. The design of a special synchronization signal that lends itself to easy unambiguous interpretation can be a challenge.

The relative advantages and disadvantages of the two types of closed-loop systems have to do with the location of the signal-processing capability and the efficiency of channel usage. A major advantage of having the processing at the central node is that results of the error measurements that are transmitted on the return link can be a short digital sequence. This efficient use of the return link can be important if a single return link is time-division multiplexed between a large number of terminals. A second potential advantage is that the error-measuring capability in the central node can be shared by all terminals communicating through the node. This can amount to a large saving in system processing capability. The principal potential advantage in having the processing at the terminal is that the central node may not be easily accessible, and reliability considerations may dictate a simple design. This has typically been the case when the central node is a space satellite. With continuing improvements in satellite technology, simplicity requirements can be expected to be less dominant in the future than in the past. Another potential advantage to having the processing in the terminal is that the response can be quicker because there is little processing delay in the central node. This may be important if link parameters are changing very rapidly. The primary disadvantages are the inefficient use of the return channel and that the return signals may be difficult to interpret. This difficulty would arise when the central node is not just a simple repeater but makes symbol decisions and transmits these decisions on the return link. This symbol decision capability can greatly improve the terminal-to-terminal error performance, but it complicates the synchronization procedure. This is because the effects of a time or frequency offset are resident in the return signal indirectly—that is, only as they have affected the symbol decisions. Consider the example of a BFSK transmission to a central node that makes noncoherent bit decisions. The decisions will be dependent on the detected signal energy in the mark and space detectors. If the transmitted signal is an alternating sequence of marks and spaces, the signal at the central node can be modeled as

$$r(t) = \begin{cases} \sin[(\omega_0 + \omega_s + \Delta\omega)t + \theta] & 0 \le t \le \Delta t \\ \sin[(\omega_0 + \Delta\omega)t + \theta] & \Delta t < t \le T \end{cases} \tag{10.91}$$

where T is the symbol interval, ω_0 is one symbol frequency, $(\omega_0 + \omega_s)$ is the other symbol frequency, $\Delta\omega$ is the frequency error at the central node, Δt is the signal arrival time error at the central node, and θ is an arbitrary phase angle. Now, if

$$x = \frac{1}{T}\int_0^T r(t)\cos\omega_0 t\, dt \tag{10.92}$$

and

$$y = \frac{1}{T}\int_0^T r(t)\sin\omega_0 t\, dt \tag{10.93}$$

represent the detector quadrature components, then the detected signal energy can be expressed as

$$\begin{aligned} z^2 &= x^2 + y^2 \\ &= \left(\frac{\sin[(\omega_s + \Delta\omega)\Delta t/2]}{(\omega_s + \Delta\omega)T}\right)^2 + \left(\frac{\sin[\Delta\omega(T - \Delta t)/2]}{\Delta\omega T}\right)^2 \\ &\quad + \frac{\cos(\Delta\omega\Delta t) + \cos[\Delta\omega T - (\omega_s + \Delta\omega)\Delta t] - \cos(\Delta\omega T) - \cos(\omega_s \Delta t)}{2\Delta\omega(\omega_s + \Delta\omega)T^2} \end{aligned} \tag{10.94}$$

For the special case where the time error, Δt, is zero, Equation (10.94) simplifies to

$$z^2 = \left[\frac{\sin(\Delta\omega T/2)}{\Delta\omega T}\right]^2 \tag{10.95}$$

Synchronization

For the case where the frequency offset is zero,

$$z^2 = \left(\frac{T - \Delta t}{2T}\right)^2 + \left[\frac{\sin(\omega_s \Delta t/2)}{\omega_s T}\right]^2 \qquad (10.96)$$

The important thing to notice in Equations (10.94) to (10.96) is that any time error or frequency offset or combination of both will decrease the detected signal energy in the correct symbol detector and introduce signal energy into the incorrect signal detector. This will reduce the effective distance between signals in signal space and degrade error performance. A measurement of error performance, however, which is all that is available on the return link, gives no insight into whether the problem is a frequency offset, a time error, or a combination of both. Thus the transmission of standard signals is not likely to provide a useful response for synchronization.

A useful technique for determining the correct frequency precorrection for our example of BFSK-signaling is to transmit a constant tone whose frequency is the average of the two symbol frequencies. Such a tone should produce a random binary sequence on the return link with equal numbers of marks and spaces. A frequency offset from the average would produce predominately marks or spaces. Finding the center frequency in this way allows accurate frequency precorrection of the signals. Once the correct frequency is found, the transmitter can transmit an alternating sequence of marks and spaces in order to discover correct timing. By varying the timing of the transmission through a range of half-a-symbol interval, the transmitter can look for the timing that provides the worst error performance. When the transmission arrival at the central node is displaced from correct timing by half-a-symbol interval, the two detectors will detect equal amounts of energy, and the binary sequence on the return link will be random. Determining the time when the transmitted and return signals are decorrelated will allow the transmitter to compute the correct transmission timing. Notice that this procedure works better than attempting to find the point at which error performance is the best. Any well-designed system will have sufficient transmission energy to allow for slight timing offsets, so an error-free return signal could be achieved with less than perfect timing. In fact, the larger the signal-to-noise ratio, the worse a best-finding procedure works. A worst-finding system, however, will work well for any well-designed system, and it will improve in potential accuracy with increasing signal-to-noise ratio. This can be seen intuitively, because increased signal-to-noise ratios will allow the system to tolerate larger timing errors, so the improvement in error performance as the timing error decreases from half-a-symbol time will be more rapid in the large signal-to-noise case than in the smaller signal-to-noise ratio case. This will allow a more precise determination of the half-symbol timing position.

Synchronization

CONCLUSION

This chapter has outlined the fundamental problems and issues associated with synchronization in digital communications. The trade-offs are generally between expense and complexity on the one hand, and error performance on the other. We have discussed receiver synchronization and phase-locked loops (PLL) in particular. Typically, it is the receiver that takes the most active role in the synchronization of a communications link. Even in cases where a terminal's transmitter assumes the more active role, as in some satellite links, the process is often aided by a return path that has been acquired by the terminal's receiver. Thus, receiver synchronization is more fundamental. Phase-locked loops and their variations are the primary control circuits used to track variations in phase of an incoming signal. The mathematics needed to describe the response of a PLL to a given input involves the solution to a nonlinear differential equation. It was shown, however, that under steady-state conditions, a linearized model provides a useful approximation to system performance. In circumstances where the linearized model cannot be accurately applied, results by Viterbi [8] for first-order loops were introduced. Although exact for first-order loops only, these results have been shown to be useful approximations to the performance of higher order loops as well [5].

The extremely important special case of suppressed-carrier loops was discussed. Suppressed-carrier loops are required to track the phase of an incoming signal that has no average energy at the carrier frequency. The common example of such a signal is one that has been modulated with standard antipodal BPSK. In this situation, a harmonic of the suppressed carrier is produced through the use of a nonlinearity, and the harmonic is tracked.

The next higher level of synchronization treated here was symbol synchronization. Two primary classes of symbol synchronization were discussed. Open-loop synchronizers operate directly on the modulated signal to produce a symbol transition indication. Closed-loop synchronizers use a closed-cycle control loop to acquire and track the symbol transitions.

The highest level of synchronization considered was frame synchronization. To receive the data in a useful form, the receiver must determine which symbols belong to which frames. This knowledge is equivalent to having frame sync, which is usually accomplished by including with the data symbols some recognizable pattern known to the receiver. The receiver scans the incoming data until it recognizes the pattern. Synchronization can be checked by looking for periodic repetitions of the pattern.

This chapter has necessarily been only an outline of the important problems, issues, and results relating to the synchronization of digital communication systems. The interested reader will find that the references listed are excellent works that will provide much greater depth of coverage than space has allowed here.

Problems and Questions

- Problems

10.2. Consider a transmitter and receiver that are in relative motion as in Problem 10.1. Once again assume that the linearized loop equations hold. Under this assumption determine the PLL phase error as a function of time for the all-pass and low-pass loop filters of Equations (10.13) and (10.14). Demonstrate that the validity of the assumption of the linearized loop equations depends on the value of the gain K_0.

10.6. A first-order phase-locked loop with loop gain K_0 is disturbed by additive white Gaussian noise of normalized (to unit signal energy) two-sided power spectral density of $N_0/2$ watts/hertz. Determine the necessary relationship between noise power spectral density and loop gain if the loop is designed to cycle slip no more often than once per day.

10.12. A deep-space probe is moving away from the earth at a nominal velocity of 15.000 m/s with a velocity uncertainty of ± 3 m/s. The probe frequency reference is specified to have a drift rate of no more than 10^{-9} Hz/Hz/day. The nominal downlink transmission frequency is 8 GHz. After a 1-month (30-day) silence the probe begins a scheduled transmission toward an earth terminal. The earth terminal contains a cesium standard. What center frequency and frequency search bandwidth should be used by the ground station? Assuming that the range to the probe was accurately known at the beginning of the month, and that the uncertainty in the probe's time and frequency references were zero [$\Delta t(0) = 0$, $\Delta\omega(0) = 0$], what is the uncertainty in the time of arrival of the downlink transmission?

10.16. Consider the case of full-response MSK signaling with a synchronization "training sequence" of alternating ones and zeros (i.e., $\alpha_k = 1$ for k even, and -1 for k odd).

(a) Show that for this example there are only four distinct phase states $\{\Phi_k\}$.

(b) Derive the form of the filters $h^{(i)}(t)$, given in Equation (10.64).

(c) Using the results obtained in part b), derive expressions for Equations (10.68) and (10.69) for this simplified case.

Problems and Questions

- Questions

10.1. What is the definition of *synchronization* in the context of a digital communication system, and why is it important? (See Section 10.1.1.)

10.2. Why may a synchronization system that works well for a home radio receiver possibly be *inadequate* in a high performance aircraft? What modifications are likely to be required for adequate performance? (See Section 10.1.2.)

10.3. The *linearized loop equation* depends on an approximation. What is that approximation, why is it appropriate for loops that are in lock or near lock, and why is it not appropriate for acquisition analysis? (See Section 10.2.1.)

10.4. Second-order phase locked loops have several advantages in their performance, and as the basis for phase-tracking performance analysis. Name two such advantages. (See Section 10.2.1.1.)

10.5. Why are *continuous phase modulation* schemes of increasing importance in modern communications system, and what synchronization challenges do they pose? (See Section 10.2.3.1.)

10.6. What are the advantages and disadvantages of *data-aided* versus *non-data-aided* synchronizers? (See Section 10.2.3.2.)

10.7. Describe a situation in which it may be appropriate to require a transmitter to synchronize itself to the expectations of a receiver. (See Section 10.3.)