Fundamentals of Synchronization

6 Fundamentals of Synchronization **1005**
6.1 Phase computation and regeneration . . . 1007
6.1.1 Phase Error Generation . . . 1007
6.1.2 Voltage Controlled Oscillators . . . 1010
6.1.2.1 The Voltage-Controlled Crystal Oscillator (VCXO) . . . 1011
6.1.2.2 Basic Jitter effect . . . 1011
6.1.3 Maximum-Likelihood Phase Estimation . . . 1012
6.2 Analysis of Phase Locking . . . 1013
6.2.1 Continuous Time . . . 1013
6.2.1.1 First-Order PLL . . . 1013
6.2.1.2 Second-Order PLL . . . 1014
6.2.2 Discrete Time Synchronization . . . 1015
6.2.2.1 First-Order Phase Lock Loop . . . 1016
6.2.2.2 Second-Order Phase Lock Loop . . . 1018
6.2.3 Phase-locking at rational multiples of the provided frequency . . . 1019
6.3 Symbol-Timing Synchronization . . . 1021
6.3.1 Open-Loop Timing Recovery . . . 1021
6.3.2 Decision-Directed Timing Recovery . . . 1023
6.3.3 Pilot Timing Recovery . . . 1025
6.3.4 Fractionally Spaced Digital Timing Recovery . . . 1025
6.4 Carrier Recovery . . . 1028
6.4.1 Open-Loop Carrier Recovery . . . 1028
6.4.2 Decision-Directed Carrier Recovery . . . 1029
6.5 Phase Locking with Multicarrier Methods . . . 1031
6.5.1 Baseband Multicarrier - Wireline DMT . . . 1031
6.5.1.1 Pilots and Synchronization in DMT . . . 1031
6.5.2 Pilot Carrier Recovery . . . 1031
6.5.3 Searcing for the guard period or pilots . . . 1031
6.5.4 Passband Multicarrier - Wireless OFDM . . . 1032
6.5.4.1 The channel-gain . . . 1032
6.5.4.2 Timing Recovery . . . 1032
6.5.4.3 Carrier Recovery . . . 1032
6.5.4.4 Pilot and Synch Symbol use in Various OFDM Standards . . . 1032
6.6 Continuous Phase Modulation - Digital Frequency Modulation . . . 1033
6.6.0.1 The Frequency Pulse Response . . . 1034
6.6.0.2 Finite-length Frequency Pulse Responses . . . 1035
6.6.1 CPM passband and baseband representations . . . 1036
6.6.2 Common CPM filters . . . 1036
6.6.2.1 The rectangular (REC) frequency filter family . . . 1036
6.6.2.2 The raised-cosine frequency filter family . . . 1039
6.6.2.3 The spectrally raised-cosine (SRC) frequency filter family . . . 1042

6.6.2.4 The Gaussian Minimum Phase-Shift Keying frequency pulse-response filter family . . . 1043
6.6.2.5 Tamed Frequency Modulation . . . 1045
6.6.3 Non-coherent CPM Detection . . . 1046
6.6.3.1 Non-Coherent Decision Feedback or known state - symbol-by-symbol detection . . . 1047
6.6.4 Calculation of CPM Spectra . . . 1048
6.6.4.1 REC Spectra . . . 1051
6.6.4.2 RC Spectra . . . 1060
6.6.4.3 SRC Spectra . . . 1064
6.6.4.4 GMSK Spectra . . . 1067
6.6.4.5 TFM Spectrum . . . 1069
6.6.4.6 Some other spectrum observations . . . 1070
6.6.5 Coherent Maximum Likelihood Detection . . . 1074
6.6.6 Performance of Coherent Detection . . . 1076
6.6.6.1 Symbol-by-Symbol Case . . . 1076
6.6.6.2 Maximum Likelihood Case . . . 1077
6.7 Frame Synchronization in Data Transmission . . . 1080
6.7.1 Autocorrelation Methods . . . 1080
6.7.1.1 Synchronization Patterns . . . 1080
6.7.1.2 The channel response . . . 1081
6.7.1.3 The noise . . . 1082
6.8 Pointers and Add/Delete Methods . . . 1083
Exercises - Chapter 6 . . . 1085

Bibliography **1090**

Index **1091**

Chapter 6

Fundamentals of Synchronization

The analysis and developments of Chapters 1-5 presumed that the modulator and demodulator are synchronized. That is, both modulator and demodulator know the exact symbol rate and the exact symbol phase, and where appropriate, both also know the exact carrier frequency and phase. In practice, the common (receiver/transmitter) knowledge of the same timing and carrier clocks rarely occurs unless some means is provided for the receiver to synchronize with the transmitter. Such synchronization is often called **phase-locking.**

In general, phase-locking uses three component operations as generically depicted in Figure 6.1:[1]

1. **Phase-error generation** - this operation, sometimes also called "phase detection," derives a phase difference between the received signal's phase $\theta(t)$ and the receiver estimate of this phase, $\hat{\theta}(t)$. The actual signals are $s(t) = cos\left(\omega_{lo}t + \theta(t)\right)$ and $\hat{s}(t) = cos\left(\omega_{lo}t + \hat{\theta}(t)\right)$, but only their phase difference is of interest in synchronization. This difference is often called the **phase error**, $\phi(t) = \theta(t) - \hat{\theta}(t)$. Various methods of phase-error generation are discussed in Section 6.1.

2. **Phase-error processing** - this operation, sometimes also called "loop filtering" extracts the essential phase difference trends from the phase error by averaging. Phase-error processing typically rejects random noise and other undesirable components of the phase-error signal. Any gain in phase-error generation is assumed to be absorbed into the loop filter's gain. Both analog and digital phase-error processing, and the general operation of what is known as a "phase-lock loop," are discussed in Section 6.2.

3. **Local phase reconstruction** - this operation, which in some implementations is known as a "voltage-controlled oscillator" (VCO), regenerates the local phase from the processed phase error in an attempt to match the incoming phase, $\theta(t)$. That is, the phase reconstruction tries to force $\phi(t) = 0$ by generation of a local phase $\hat{\theta}(t)$ so that $\hat{s}(t)$ matches $s(t)$. Various types of voltage controlled oscillators and other all-digital methods of regenerating local clock are also discussed in Section 6.1

Any phase-locking mechanism will have some finite delay in practice so that the regenerated local phase will try to project the incoming phase and then measure how well that projection did in the form of a new phase error. The more quickly the phase-lock mechanism tracks deviations in phase, the more susceptible it will be to random noise and other imperfections. Thus, the communication engineer must trade these two competing effects appropriately when designing a synchronization system. The design of the transmitted signals can facilitate or complicate this trade-off. These design trade-offs will be generally examined in Section 6.2. In practice, unless pilot or control signals are embedded in the transmitted signal, it is necessary either to generate an estimate of transmitted signal's clocks or to generate an estimate of the phase error directly. Sections 6.3 and 6.4 specifically examine phase detectors for situations where such clock extraction is necessary for timing recovery (recovery of the symbol clock) and carrier recovery (recovery of the phase of the carrier in passband transmission),

[1] These operations may implemented in a variety of ways.

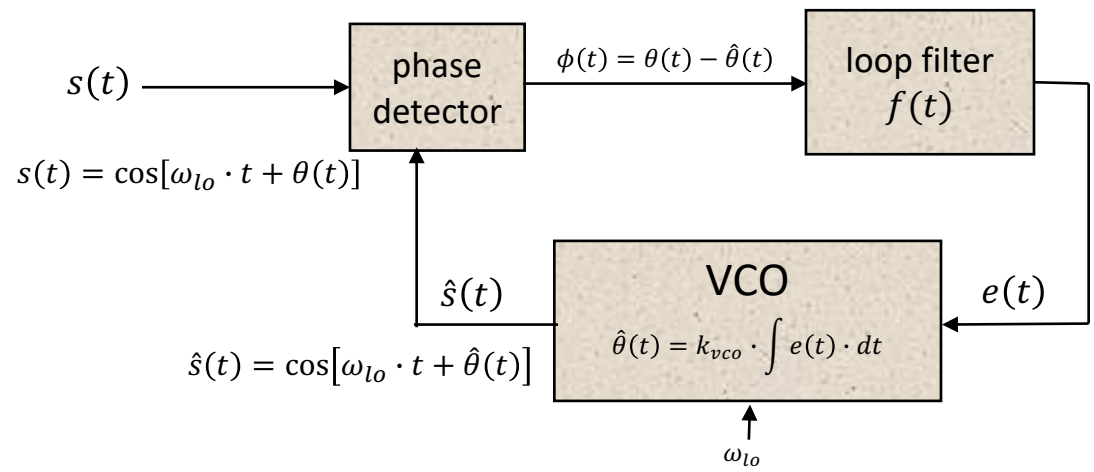


Figure 6.1: General phase-lock loop structure.

respectively. Multicarrier methods, particularly in wireless, create many challenges for synchronization. The methods for timing recovery for baseband (wireline) systems like copper or fiber transmission systems are first discussed in Section 6.5 before progressing to the more difficult joint timing and carrier recovery for wireless systems, particularly in mobile systems with channel variation.

Section 6.6 investigates digital FM and Continuous Phase Modulation where essentially a VCO is used to modulate (and sometimes also to demodulate) message-bearing communication signals, where the messages are conveyed by discrete phase shifts (or frequency shifts) from one symbol to the next. These systems can be very important in transmission applications where energy savings is of paramount concern. In addition to symbol timing, data is often packetized or framed. Methods for recovering frame boundary are discussed in Section 6.7. Pointers and add/delete (bit or byte robbing/stuffing) are specific mechanisms of local phase reconstruction that allow the use of asynchronous clocks in the transmitter and receiver while effectively synchronizing the transmitter and receiver. These methods find increasing use in digital integrated-circuit or software implementations of receivers and are discussed in Section 6.8.

6.1 Phase computation and regeneration

This section details Figure 6.1's phase detector and voltage-controlled oscillator (VCO) operation.

6.1.1 Phase Error Generation

Phase-error generation can be continuous or discrete-time at specific sampling instants. This subsection's discussion therefore does not use a time argument or sampling index on phase signals. That is $\theta(t) \to \theta$.

Ideally, the two phase angles, θ, the phase of the input sinusoid s, and $\hat{\theta}$, the estimated phase produced at the VCO output, would be available. Then, their difference ϕ could be computed directly.

Definition 6.1.1 *[ideal phase detector] A device that can compute exactly the difference $\phi = \theta - \hat{\theta}$ is called an* **ideal phase detector**. *Ideally $E[\phi(t)] = 0$ and the phase error has standard deviation $\sqrt{E[\phi^2(t)]}$, which is known as the* **phase jitter**. **Timing jitter** *can be expressed also as $\sqrt{E[(\delta t)^2]}$ where δt represents a time (not phase) offset from the correct/desired zero crossings of $s(t)$.*

Ideally, the receiver would have access to the sinusoids $s = \cos(\theta_{lo} + \theta)$ and $\hat{s} = \cos(\theta_{lo} + \hat{\theta})$ where θ_{lo} is the common phase reference that depends on the local oscillator frequency; θ_{lo} disappears from the ensuing arguments, but in practice the general frequency range of $\frac{d\theta}{dt} \approx \omega_{lo}$ will affect implementations[2]. A seemingly straightforward method to compute the phase error would then be to compute θ, the phase of the input sinusoid s, and $\hat{\theta}$, the estimated phase, according to

$$\theta = \pm \arccos[s] - \theta_{lo} \;\; ; \;\; \hat{\theta} = \pm \arccos[\hat{s}] - \theta_{lo} \quad . \tag{6.1}$$

Then, $\phi = \theta - \hat{\theta}$, to which the value θ_{lo} is inconsequential in theory. However, a reasonable implementation of the arccos function can only produce angles between 0 and π, so that ϕ would then always lie between $-\pi$ and π. Any difference of magnitude greater than π would be thus effectively be computed modulo $(-\pi, \pi)$. The arccos function could then be implemented with a look-up table.

Definition 6.1.2 *[mod-2π phase detector] The* arccos *look-up table implementation of the phase detector is called a* **modulo-2π phase detector**[3].

The characteristics of the mod-2π phase detector and the ideal phase detector are compared in Figure 6.2.

[2] Local oscillators often use a timing reference that can be based on carefully cut crystals with resonant frequencies guaranteed to be "close" to the nominal frequency, say with ±2 to ±50 ppm (parts per million), depending on application requirements. If both sides use such crystals, ω_{lo} will be close. Other timing references can be obtained from satellite or other wireless beacons available to both transmitter and receiver, again at least to get close.

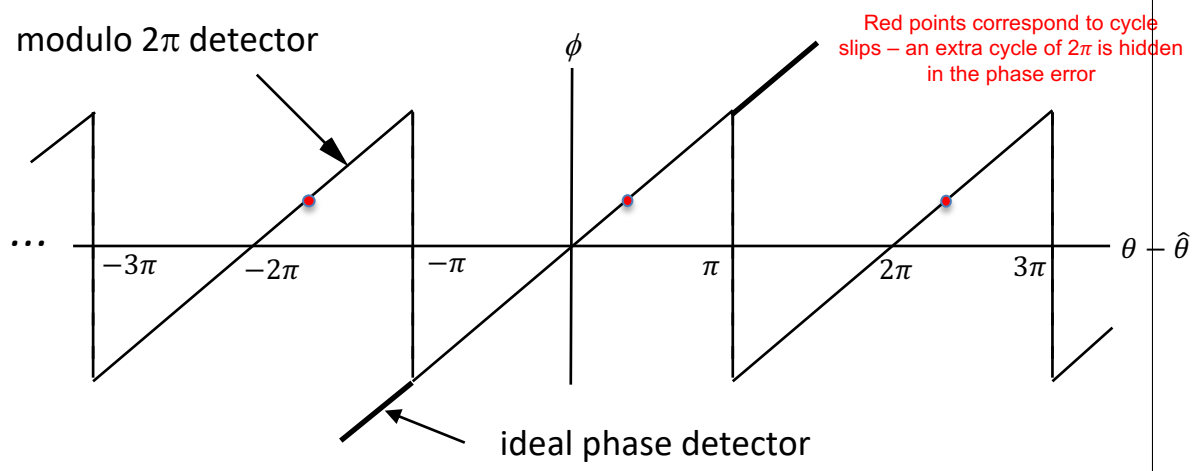


Figure 6.2: Comparison of ideal and mod-2π phase detectors.

If the phase difference does exceed π in magnitude, the large difference will not be exhibited in the phase error ϕ - this phenomenon is known as a "cycle slip," meaning that the phase lock loop missed (or added) an entire period (or periods) of the input sinusoid. This is an undesirable phenomena in most applications, so after a phase-lock loop with mod-2π phase detector has converged, one tries to ensure that $|\phi|$ cannot exceed π. In practice, a good phase lock loop should keep this phase error as close to zero as possible, so the condition of small phase error necessary for use of the mod-2π phase detector is met. The means for avoiding hidden cycle slips is to ensure that the local-oscillator frequency is less than double the desired frequency and greater than 1/2 that desired.

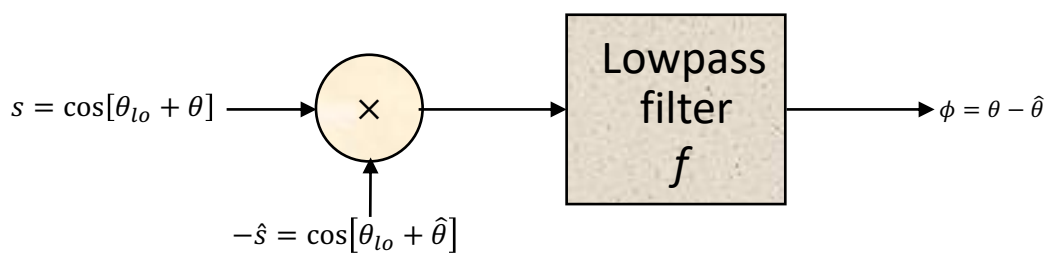


Figure 6.3: Demodulating phase detector (or "mixer").

Definition 6.1.3 *[demodulating phase detector] Another commonly encountered phase detector, both in analog and digital form, is the* **demodulating phase detector**[4] *shown in Figure 6.3, where*

$$\phi = f * \left[-\sin(\theta_{lo} + \hat{\theta}) \cdot \cos(\theta_{lo} + \theta) \right] \quad , \tag{6.2}$$

and f is a lowpass filter that is cascaded with the phase-error processing in the phase-locking mechanism.

The basic concept arises from the relation

$$-\cos\left(\omega_{lo}t+\theta\right)\cdot\sin\left(\omega_{lo}t+\hat{\theta}\right)=\frac{1}{2}\left\{-\sin\left(2\omega_{lo}t+\theta+\hat{\theta}\right)+\sin\left(\theta-\hat{\theta}\right)\right\} \quad , \tag{6.3}$$

where the sum-phase term (first term on the right) can be eliminated by lowpass filtering; this lowpass filtering can be absorbed into the loop filter that follows the phase detector (see Section 6.2). The phase detector output is thus proportional to $\sin(\phi)$. The usual assumption with this type of phase detector is that ϕ is small ($\phi << \frac{\pi}{6}$), and thus

$$\sin(\phi)\approx\phi \quad . \tag{6.4}$$

When ϕ is small, generation of the phase error thus does not require the arcsin function.

Sampling Phast Detector: Another way of generating the phase-error signal is to use the local sinusoid to

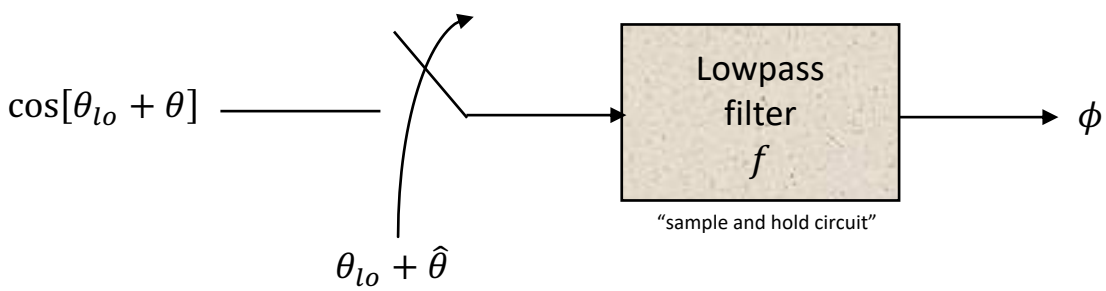


Figure 6.4: Sampling phase detector.

sample the incoming sinusoid as shown in Figure 6.4. If the rising edge of the local sinusoid is used for the sampling instant, then

$$\theta_{lo}+\hat{\theta}=-\frac{\pi}{2} \quad . \tag{6.5}$$

(At this phase, $\hat{s}=0$.) Then, at the time of this rising edge, the sampled sinusoid s has phase

$$\theta_{lo}+\hat{\theta}+\theta-\hat{\theta}=-\frac{\pi}{2}+\phi \quad , \tag{6.6}$$

so that $s(t)$ at these times t_k is

$$s(t_k)=cos(-\frac{\pi}{2}+\phi)=\sin(\phi)\approx\phi \quad . \tag{6.7}$$

Such a phase detector is called a **sampling phase detector.** The lowpass filter in Figure 6.4 "holds" the sample value of phase, and is sometimes known as a **sample-and-hold circuit**.

Binary Phase Detector: Figure 6.5 illustrates the **binary phase detector**. In the binary phase detector, the phase difference between the two sinusoids is approximately the width of the high signals at point C in Figure 6.5. The hard limiters are used to convert the sinusoids into 1/0 square waves or binary signals. The adder is a binary adder. The lowpass filter just averages (integrates) the error signal, so that its output is proportional to the magnitude of the phase error. The important sign of the phase error is determined by "latching" the polarity of $\hat{s}$ when s goes high (leading-edge triggered D flip-flop).[5]

[5]When A and B are phase aligned, then $\phi=0$, so that the race condition that exists in the clocking and data set-up on line B should not be of practical significance if sufficiently high-speed logic is used.

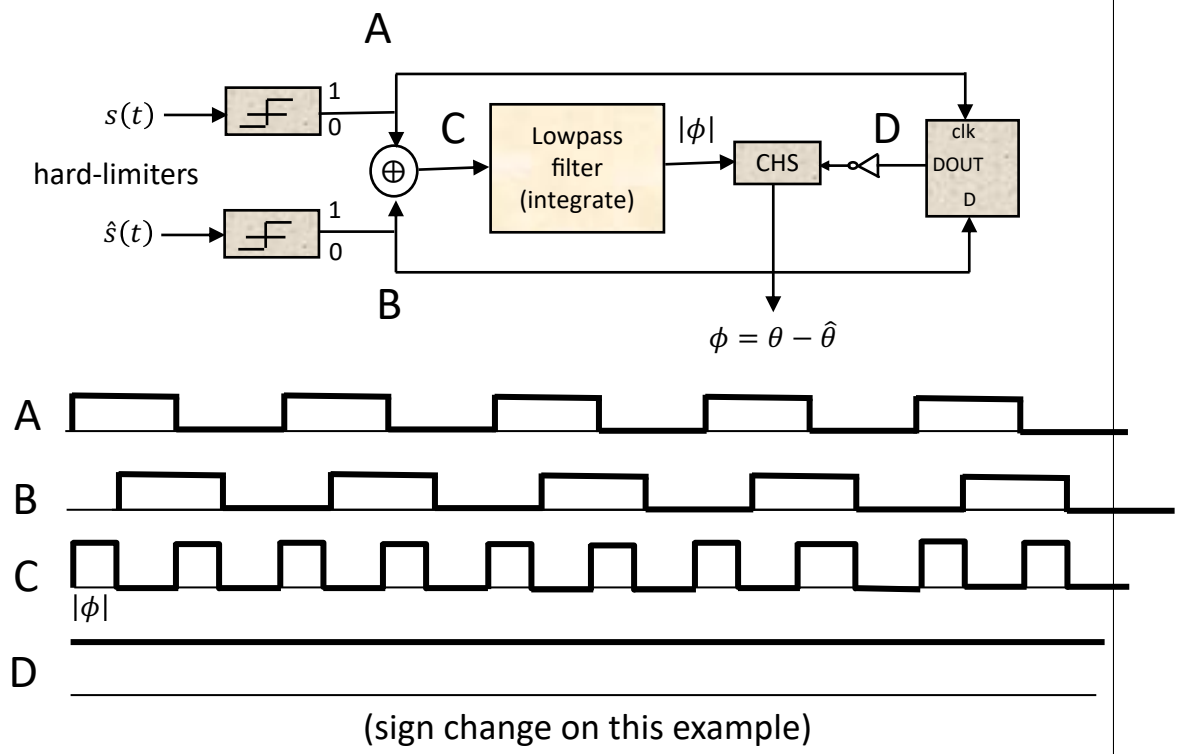


Figure 6.5: Binary phase detector.

6.1.2 Voltage Controlled Oscillators

The voltage controlled oscillator basically generates a sinusoid with phase difference (or derivative) proportional to the input control voltage $e(t)$.

Definition 6.1.4 *[Voltage Controlled Oscillator] An ideal* **voltage controlled oscillator (VCO)** *has an output sinusoid with phase $\hat{\theta}(t)$ that is determined by an input* **error** *or* **control** *signal according to*

$$\frac{d\hat{\theta}}{dt} = k_{vco} \cdot e(t) \quad , \tag{6.8}$$

in continuous time, or approximated by

$$\hat{\theta}_{k+1} = \hat{\theta}_k + k_{vco} \cdot e_k \quad . \tag{6.9}$$

in discrete time.

Analog VCO physics are beyond the scope of this text, so it will suffice to just state that devices satisfying (6.8) are readily available in a variety of frequency ranges. When the input signal e_k in discrete time is digital, the VCO can also be implemented with a look-up table and adder according to (6.9) whose output is used to generate a continuous time sinusoidal equivalent with a DAC. This implementation is often called a **numerically controlled oscillator** or NCO.

Yet another implementation that can be in digital logic on an integrated circuit is shown in Figure 6.6.

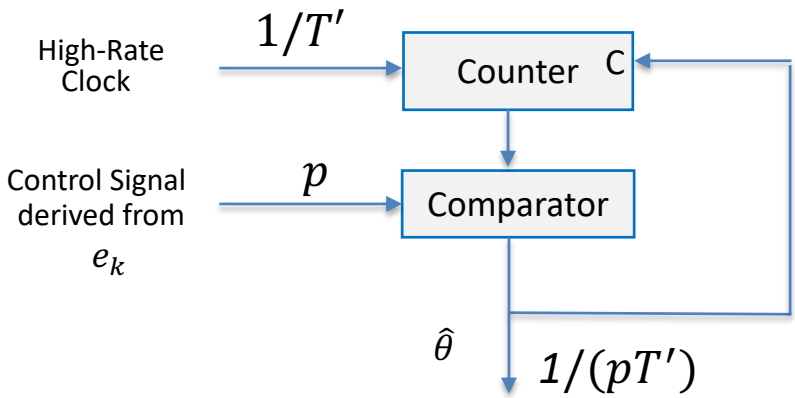


Figure 6.6: Discrete-time VCO with all-digital realization.

The high-rate clock is divided by the value of the control signal (derived from e_k) by connecting the true output of the comparator to the clear input of a counter. The higher the clock rate with respect to the rates of interest, the finer the resolution on specifying $\hat{\theta}_k$. If the clock rate is $1/T'$, then the divider value is p or $p+1$, depending upon whether the desired clock phase is late or early, respectively. A maximum phase change with respect to a desired phase (without external smoothing) can thus be $T'/2$. For 1% clock accuracy, then the master clock would need to be 50 times the generated clock frequency. With an external analog smoothing of the clock signal (via bandpass filter centered around the nominal clock frequency), a lower frequency master clock can be used. Any inaccuracy in the high-rate clock is multiplied by p so that the local oscillator frequency must now lay in the interval $\left(f_{lo} \cdot (1 - \frac{1}{2p}), f_{lo} \cdot (1+p)\right)$ to avoid cycle slipping. (When $p = 1$, this reduces to the interval mentioned earlier.) See also Section 6.2.3 on phase-locking at rational multiples of a clock frequency.

6.1.2.1 The Voltage-Controlled Crystal Oscillator (VCXO)

In many situations, the approximate clock frequency to be derived is known accurately. Conventional crystal oscillators usually have accuracies of 50 parts per million (ppm) or better. Thus, the VCO need only track over a small frequency/phase range. Additionally in practice, the derived clock may be used to sample a signal with an analog-to-digital converter (ADC). Such an ADC clock should not jitter about its nominal value (or significant signal distortion can be incurred). In this case, a VCXO normally replaces the VCO. The VCXO employs a crystal (X) of nominal frequency to stabilize the VCO close to the nominal value. Abrupt changes in phase are not possible because of the presence of the crystal. However, the high stability and slow variation of the clock can be of crucial importance in digital receiver designs. Thus, VCXO's are used instead of VCO's in designs where high stability sample clocking is necessary.

6.1.2.2 Basic Jitter effect

Oscillator "timing" jitter distortion δx for a waveform $x(t) = s(t) + n(t)$ has linearized approximation

$$\delta x \approx \frac{dx}{dt} \cdot \delta t \quad , \tag{6.10}$$

so that

$$(\delta x)^2 \approx \left(\frac{dx}{dt}\right)^2 (\delta t)^2 \quad . \tag{6.11}$$

A signal-to-jitter-noise ratio can be defined by

$$\text{SNR} = \frac{\mathbb{E}\left[x^2\right]}{\mathbb{E}\left[(\delta x)^2\right]} = \frac{\mathcal{E}_{\boldsymbol{x}}}{(dx/dt)^2 \cdot \mathbb{E}\left[(\delta t)^2\right]} \quad . \tag{6.12}$$

For the highest frequency component of $x(t)$ with frequency f_{max}, the SNR becomes

$$\text{SNR} = \frac{\mathcal{E}_{\boldsymbol{x}}}{4\pi^2 \cdot (f_{max} \cdot \delta t)^2} \quad , \tag{6.13}$$

which illustrates a basic time/frequency uncertainty principle: If $(\delta t)^2$ represents timing jitter in squared seconds, then jitter frequirements must become smaller (tighter) as the signal frequency f_{max} increases. An SNR of 20 dB (factor of 100 in jitter) with a signal with maximum frequency of 1 MHz would suggest that jitter be below 16 ns.

6.1.3 Maximum-Likelihood Phase Estimation

A number of detailed developments on phase lock loops attempt to estimate phase from a likelihood function:

$$\max_{x,\theta} p_{y/x,\theta} \quad . \tag{6.14}$$

Maximization of such a function can be complicated mathematically, often leading to a series of approximations for various trigonometric functions that ultimately lead to a quantity proportional to the phase error that is then used in a phase-lock loop. Such approaches are acknowledged here, but those interested in the ultimate limits of synchronization performance are referred elsewhere. Practical approximations are included here under the category of "decision-directed" synchronization methods in Sections 6.3.2 and 6.4.2.

6.2 Analysis of Phase Locking

Tis section analyzes both continuous-time and discrete-time PLL's. Subsections 6.2.1 and 6.2.2 illustrate both first-order and second-order PLL's for continuous- and discrete-time respectively. First-order loops are found to track only constant phase offsets, while second-order loops can track both phase and/or frequency offsets. Throughout this section, the continuous-time-frequency Laplace transform variable s should not be confused with the sinusoid to be phase locked, $s(t)$. The appropriate interpretation should be obvious in context.

6.2.1 Continuous Time

Figure 6.7's continuous-time PLL has a phase estimate that follows the differential equation

$$\dot{\hat{\theta}}(t) = k_{vco} \cdot f(t) * \Big(\theta(t) - \hat{\theta}(t)\Big) \quad . \tag{6.15}$$

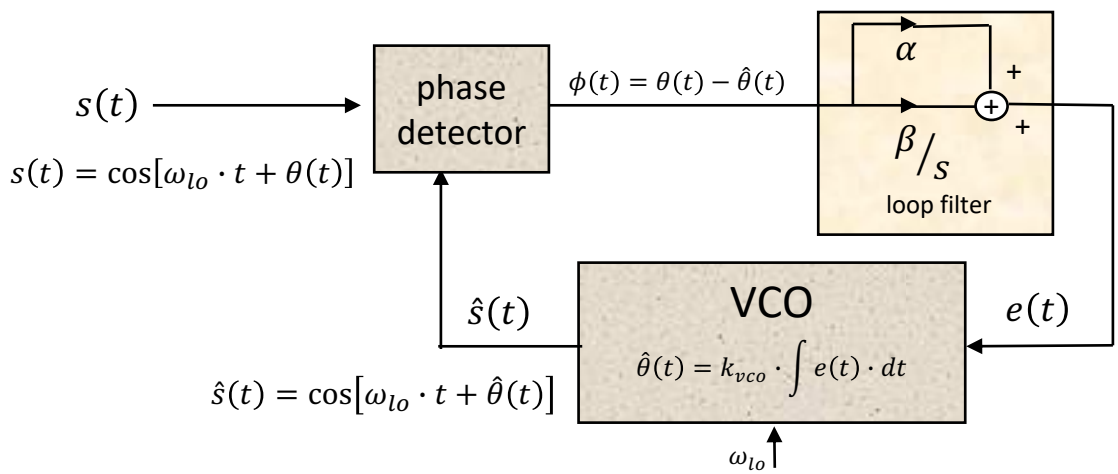


Figure 6.7: Continuous-time PLL with loop filter.

The transfer function between PLL output phase and input phase is

$$\frac{\hat{\theta}(s)}{\theta(s)} = \frac{k_{vco} \cdot F(s)}{s + k_{vco} \cdot F(s)} \tag{6.16}$$

with s the Laplace transform variable. The corresponding transfer function between phase error and input phase is

$$\frac{\phi(s)}{\theta(s)} = 1 - \frac{\hat{\theta}(s)}{\theta(s)} = \frac{s}{s + k_{vco} \cdot F(s)} \quad . \tag{6.17}$$

The cases of both first- and second-order PLL's are shown in the upper-right box in Figure 6.7 with $\beta = 0$, reducing the diagram to a first-order loop.

6.2.1.1 First-Order PLL

The first order PLL has

$$k_{vco} \cdot F(s) = \alpha \tag{6.18}$$

(a constant) so that phase errors are simply integrated by the VCO in an attempt to set the phase $\hat{\theta}$ such that $\phi = 0$. When ϕ =0, there is zero input to the VCO and the VCO output is a sinusoid at frequency

ω_{lo}. Convergence to zero phase error will only happen when $s(t)$ and $\hat{s}(t)$ have the same frequency (ω_{lo}) with initially a constant phase difference that can be driven to zero under the operation of the first-order PLL, as is subsequently shown.

The response of the first-order PLL to an initial phase offset (θ_0), or a waveform precisely as $\theta_0 \cdot u(t)$ with Laplace transform θ_0/s, with

$$\theta(s) = \frac{\theta_0}{s} \tag{6.19}$$

is

$$\hat{\theta}(s) = \frac{\alpha \cdot \theta_0}{s(s+\alpha)} \tag{6.20}$$

or

$$\hat{\theta}(t) = \left(\theta_0 - \theta_0 \cdot e^{-\alpha t}\right) \cdot u(t) \tag{6.21}$$

where $u(t)$ is the unit step function (=1, for $t > 0$, $= 0$ for $t < 0$). For stability, $\alpha > 0$. Clearly

$$\hat{\theta}(\infty) = \theta_0 \quad . \tag{6.22}$$

An easier analysis of the PLL final value is through the final-value theorem:

$$\begin{aligned} \phi(\infty) &= \lim_{s \to 0} s \cdot \phi(s) && (6.23) \\ &= \lim_{s \to 0} s \cdot \frac{\theta_0}{s+\alpha} && (6.24) \\ &= 0 \quad , && (6.25) \end{aligned}$$

so that the final phase error is zero for a unit-step phase input. For a linearly varying phase (that is a constant frequency offset), $\theta(t) = (\theta_0 + \Delta \cdot t) \cdot u(t)$, or

$$\theta(s) = \frac{\theta_0}{s} + \frac{\Delta}{s^2} \quad , \tag{6.26}$$

where Δ is the frequency offset

$$\Delta = \frac{d\theta}{dt} - \omega_{lo} \quad . \tag{6.27}$$

In this case, the final value theorem illustrates the first-order PLL's eventual phase error is

$$\phi(\infty) = \frac{\Delta}{\alpha} \quad . \tag{6.28}$$

A larger "loop gain" α causes a smaller the offset, but a first-order PLL cannot drive the phase error to zero. The steady-state phase instead lags the correct phase by Δ/α. Increase oif α reduces phase lag. However, larger α increases the PLL bandwidth. Any small noise in the phase error then will pass through the PLL into the phase estimate with less attenuation, leading to a more noisy phase estimate. As long as $|\Delta/\alpha| < \pi$, then the modulo-2π phase detector functions with constant non-zero phase error. The range of Δ for which the phase error does not exceed π is known as the **pull range** of the PLL

$$\text{pull range} = |\Delta| < |\alpha| \cdot \pi \quad . \tag{6.29}$$

That is, any frequency deviation less than the pull range will result in a constant phase "lag" error. Such a non-zero phase error may or may not present a problem for the associated transmission system. A better method by which to track frequency offset is the second-order PLL.

6.2.1.2 Second-Order PLL

The second order PLL uses the loop filter to integrate the incoming phase errors as well as pass the errors to the VCO. The integration of errors eventually supplies a constant signal to the VCO, which in turn forces the VCO output phase to vary linearly with time. That is, the frequency of the VCO can then be shifted away from ω_{lo} permanently, unlike the operation with first-order PLL.

In the second-order PLL,

$$k_{vco} \cdot F(s) = \alpha + \frac{\beta}{s} \quad . \tag{6.30}$$

Then

$$\frac{\hat{\theta}(s)}{\theta(s)} = \frac{\alpha \cdot s + \beta}{s^2 + \alpha \cdot s + \beta} \quad , \tag{6.31}$$

and

$$\frac{\phi(s)}{\theta(s)} = \frac{s^2}{s^2 + \alpha \cdot s + \beta} \quad . \tag{6.32}$$

One easily verifies with the final value theorem that for either constant phase or frequency offset (or both) that

$$\phi(\infty) = 0 \tag{6.33}$$

for the second-order PLL. For stability,

$$\alpha > 0 \tag{6.34}$$

$$\beta > 0 \quad . \tag{6.35}$$

6.2.2 Discrete Time Synchronization

This subsection examines **discrete-time**

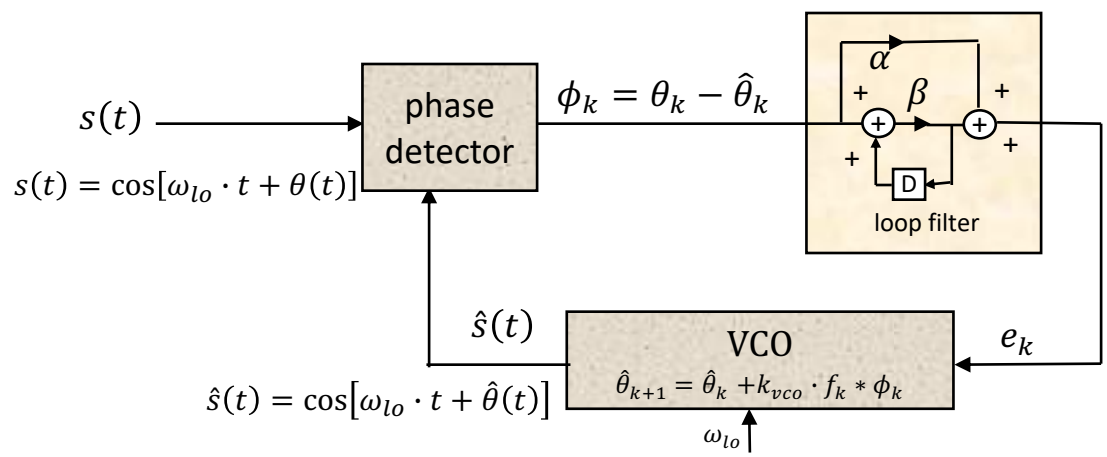


Figure 6.8: Discrete-time PLL.

first-order and second-order phase-lock loops. Figure 6.8 is the discrete-time equivalent of Figure 6.1. Again, the first-order PLL only recovers phase and not frequency. The second-order PLL recovers both phase and frequency. The discrete-time VCO follows

$$\hat{\theta}_{k+1} = \hat{\theta}_k + k_{vco} \cdot f_k * \phi_k \quad , \tag{6.36}$$

and generates $\cos\left(\omega_{lo} \cdot kT + \hat{\theta}_k\right)$ as the local phase at sampling time k. This expression assumes discrete "jumps" in the phase of the VCO. In practice, such transitions will be smoothed by any VCO that produces a sinusoidal output and the analysis here is only approximate for $kT \leq t \leq (k+1)T$, where T is the sampling period of the discrete-time PLL (which often is the symbol period in digital-transmission systems, but need not be). Taking the D-Transforms of both sides of (6.36) equates to

$$D^{-1} \cdot \hat{\Theta}(D) = [1 - k_{vco} \cdot F(D)] \cdot \hat{\Theta}(D) + k_{vco} \cdot F(D) \cdot \Theta(D) \quad . \tag{6.37}$$

The transfer function from input phase to output phase is

$$\frac{\hat{\Theta}(D)}{\Theta(D)} = \frac{D \cdot k_{vco} \cdot F(D)}{1 - [1 - k_{vco} \cdot F(D)] \cdot D} \quad . \tag{6.38}$$

The transfer function between the error signal $\Phi(D)$ and the input phase is thus

$$\frac{\Phi(D)}{\Theta(D)} = \frac{\Theta(D)}{\Theta(D)} - \frac{\hat{\Theta}(D)}{\Theta(D)} = 1 - \frac{D \cdot k_{vco} \cdot F(D)}{1 - [1 - k_{vco} \cdot F(D)] \cdot D} = \frac{D - 1}{D \cdot [1 - k_{vco} \cdot F(D)] - 1} \quad . \tag{6.39}$$

$F(D)$ determines whether the PLL is first- or second-order.

6.2.2.1 First-Order Phase Lock Loop

In the first-order PLL, the loop filter is a (frequency-independent) gain α, so that

$$k_{vco} \cdot F(D) = \alpha \quad . \tag{6.40}$$

Then

$$\frac{\hat{\Theta}(D)}{\Theta(D)} = \frac{\alpha \cdot D}{1 - (1 - \alpha) \cdot D} \quad . \tag{6.41}$$

For stability, $|1 - \alpha| < 1$, or

$$0 \leq \alpha < 2 \quad . \tag{6.42}$$

The closer α to 2, the wider the bandwidth of the overall filter from $\Theta(D)$ to $\hat{\Theta}(D)$, and the more (any) noise in the input sinusoid can distort the estimated phase. The first-order loop can track and drive to zero any phase difference between a constant θ_k and $\hat{\theta}_k$. To see this effect, a constant input phase is modeled by

$$\theta_k = \theta_0 \quad \forall \;\; k \geq 0 \quad , \tag{6.43}$$

which has D-Transform

$$\Theta(D) = \frac{\theta_0}{1 - D} \quad . \tag{6.44}$$

The phase-error sequence then has transform

$$\begin{aligned} \Phi(D) &= \frac{D - 1}{D \cdot (1 - k_{vco} F(D)) - 1} \cdot \Theta(D) && (6.45) \\ &= \frac{(D - 1) \cdot \theta_0}{(1 - D) \cdot (D \cdot (1 - \alpha) - 1)} && (6.46) \\ &= \frac{\theta_0}{1 - (1 - \alpha) \cdot D} && (6.47) \end{aligned}$$

Thus

$$\phi_k = \begin{cases} \theta_0 \cdot (1 - \alpha)^k & k \geq 0 \\ 0 & k < 0 \end{cases} \quad , \tag{6.48}$$

and $\phi_\infty \to 0$ if (6.42) is satisfied. This result can also be obtained by the final-value theorem for one-sided D-Transforms

$$\lim_{k \to \infty} \phi_k = \lim_{D \to 1} (1 - D) \cdot \Phi(D) = 0 \quad . \tag{6.49}$$

Values of α between 1 and 2 lead to oscillating decay around the steady-state value. This large swinging of the loop intuitively implies magnification of noise.

The first-order loop will exhibit a constant phase offset, at best, for any frequency deviation between θ_k and $\hat{\theta}_k$. To see this constant-lag effect, the input phase can be set to

$$\theta_k = \Delta \cdot k \quad \forall \;\; k \geq 0 \quad , \tag{6.50}$$

where $\Delta = \omega_{offset}T$, which has D-Transform[6]

$$\Theta(D) = \frac{\Delta \cdot D}{(1-D)^2} \quad . \tag{6.51}$$

The phase-error sequence then has transform

$$\begin{aligned}
\Phi(D) &= \frac{D-1}{D \cdot [1 - k_{vco} \cdot F(D)] - 1} \cdot \Theta(D) && (6.52)\\
&= \frac{D-1}{D \cdot [1 - k_{vco} \cdot F(D)] - 1} \cdot \frac{\Delta \cdot D}{(1-D)^2} && (6.53)\\
&= \frac{\Delta \cdot D}{(1-D) \cdot [1 - D \cdot (1-\alpha])} && (6.54)
\end{aligned}$$

This steady-state phase error can also be computed by the final value theorem

$$\lim_{k\to\infty} \phi_k = \lim_{D\to 1} (1-D) \cdot \Phi(D) = \frac{\Delta}{\alpha} \quad . \tag{6.55}$$

This constant-lag phase error is analogous to the same effect in the continuous-time PLL. Equation (6.55) can be interpreted in several ways. The main result is that the first-order loop cannot track a nonzero frequency offset $\Delta = \omega_{offset}T$, in that the phase error cannot be driven to zero. This result also illuminates the value of $\alpha = 1$: Values larger than 1 lead to a negative (but less than unit magnitude) geometric constant so the PLL will converge (as long as $\alpha < 2$). However, values greater than 1 (but stable) cause the loop to oscillate about the final steady-state phase as the loop converges (decays to that steady-state phase). The value $\alpha = 1$ is a special case where essentially the new phase estimate is the current value of the actual phase (the update of the phase equation degenerates to $\hat{\theta}_{k+1} = \theta_k$). This essentially means "noise is passed through unchanged" to the next phase estimate. Indeed phase noise increases in magnitude when $1 < \alpha < 2$ as will be discussed below, so this range might well be avoided. These values also lead to a reduction in the frequency offset in (6.55), while values less than one (without oscillating) decay actually increase the frequency offset Δ in the lag.

For very small frequency offsets, say a few parts per million of the sampling frequency or less, the first-order loop will incur only a very small penalty in terms of residual phase error (for reasonable α satisfying (6.42)). In this case, the first-order loop may be sufficient in terms of magnitude of phase error, in nearly tracking the frequency offset. To stay within the linear range of the modulo-2π phase detector (thereby avoiding cycle slips) after the loop has converged, the magnitude of the frequency offset $|\omega_{offset}|$ must be less than $\frac{\alpha\pi}{T}$. The reader is cautioned not to misinterpret this result by inserting the maximum α (=2) into this result and concluding than any frequency offset can be tracked with a first-order loop as long as the sampling rate is sufficiently high. Increasing the sampling rate $1/T$ at fixed α, or equivalently increasing α at fixed sampling rate, increases the bandwidth of the phase-lock loop filter. Any noise on the incoming phase will be thus less filtered or "less rejected" by the loop, resulting in a lower quality estimate of the phase. A better solution is to often increase the order of the loop filter, resulting in the following second-order phase lock loop.

An example considers a PLL attempting to track a 1 MHz clock with a local oscillator clock that may deviate by as much as 100 ppm, or equivalently 100 Hz in frequency. The designer may determine that a phase error of $\pi/20$ is sufficient for good performance of the receiver. Then,

$$\frac{\omega_{offset} \cdot T}{\alpha} \leq \frac{\pi}{20} \tag{6.56}$$

or

$$\alpha \geq 40 \cdot f_{offset} \cdot T \quad . \tag{6.57}$$

Then, since $f_{offset} = 100$ Hz, and if the loop samples at the clock speed of $1/T = 10^6$ MHz, then

$$\alpha > 4 \times 10^{-3} \quad . \tag{6.58}$$

[6]Using the relation that $(D)\frac{dF}{dD} \leftrightarrow kx_k$, with $x_k = \Delta \; \forall \; k \geq 0$.

Such a small value of α is within the stability bound of $0 < \alpha < 2$. If the phase error or phase estimate are relatively free of any "noise," then this value is probably acceptable. However, if either the accuracy of the clock is less or the sampling rate is less, then an unacceptably large value of α can occur.

Noise Analysis of the First-Order PLL: If the phase input to the PLL (that is θ_k) has some zero-mean "noise" component with variance σ_θ^2, then the component of the phase error caused by the noise is

$$\Phi_n(D) = \frac{1-D}{1-[1-\alpha]\cdot D} \cdot N_\theta(D) \tag{6.59}$$

or equivalently

$$\phi_{n,k} = (1-\alpha)\cdot\phi_{n,k-1} + n_{\theta,k} - n_{\theta,k-1} \quad . \tag{6.60}$$

By squaring the above equation and finding the steady-state phase-jitter variance $\sigma_{\phi,k}^2 = \sigma_{\phi,k-1}^2 = \sigma_\phi^2$ and setting $E\left[n_{\theta,k}\cdot n_{\theta,k-l}\right] = \sigma_\theta^2 \cdot \delta_l$ via algebra,

$$\sigma_\phi^2 = \frac{\sigma_\theta^2}{1-\alpha/2} \quad . \tag{6.61}$$

Larger values of α create more rapid response response to the input phase deviations, but lead to larger phase noise as reflected in the denominator of (6.61) that acts to increase noise variance. If $\alpha \to 2$, the phase error variance becomes infinite. Small values of α reduce the variance but never below the value σ_ϕ^2 (which occurs only when $\alpha = 0$ so no tracking at all). Thus, the noise is always increased in the first-order PLL. The solution when there is any possibility of significant noise in the phase is to use the second-order PLL of the next subsection.

6.2.2.2 Second-Order Phase Lock Loop

In the second-order PLL, the loop filter is an accumulator of phase errors, so that

$$k_{vco}\cdot F(D) = \alpha + \frac{\beta}{1-D} \quad . \tag{6.62}$$

This equation is perhaps better understood by rewriting it in terms of the second-order difference equations for the phase estimate (where $\hat{\Delta}_k$ is an intermediate variable that estimates the radian frequency offset)

$$\hat{\Delta}_k = \hat{\Delta}_{k-1} + \beta\cdot\phi_k \tag{6.63}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha\cdot\phi_k + \hat{\Delta}_k \tag{6.64}$$

In other words, the PLL accumulates phase errors into a frequency offset (times T) estimate $\hat{\Delta}_k$, which is then added to the first-order phase update at each iteration. Then

$$\frac{\hat{\Theta}(D)}{\Theta(D)} = \frac{(\alpha+\beta)\cdot D - \alpha\cdot D^2}{1-(2-\alpha-\beta)\cdot D + (1-\alpha)\cdot D^2} \quad , \tag{6.65}$$

which has poles $\frac{1}{(1-\frac{\alpha+\beta}{2})\pm\sqrt{(\frac{\alpha+\beta}{2})^2-\beta}}$. Interestingly, the value of $\alpha = 1$ causes a single pole at $\frac{1}{2-\alpha-\beta}$ in the transfer function, and corresponds to $\hat{\theta}_{k+1} = \theta_k + \hat{\Delta}_k$, again suggesting noise is passed through to the new phase estimate. However, the noise in the frequency-offset estimate may be significantly reduced for small β, perhaps leading to an acceptable situation. Prob 6.6 investigates such a situation when the initial frequency offset is random but small. For stability, α and β must satisfy,

$$0 \le \alpha < 2 \tag{6.66}$$

$$0 \le \beta < 1 - \frac{\alpha}{2} - \sqrt{\frac{\alpha^2}{2} - 1.5\alpha + 1} \quad . \tag{6.67}$$

Typically, $\beta < \left(\frac{\alpha+\beta}{2}\right)^2$ for real roots, which makes $\beta << \alpha$ since $\alpha+\beta < 1$ in most designs.

The second-order loop will track for any frequency deviation between θ_k and $\hat{\theta}_k$. To see this effect, the input phase is again set to

$$\theta_k = \Delta \cdot k \quad \forall \; k \geq 0 \quad , \tag{6.68}$$

which has D-Transform

$$\Theta(D) = \frac{\Delta \cdot D}{(1-D)^2} \quad . \tag{6.69}$$

The phase-error sequence then has transform

$$\begin{aligned} \Phi(D) &= \frac{D-1}{D \cdot [1 - k_{vco} \cdot F(D)] - 1} \cdot \Theta(D) && (6.70) \\ &= \frac{(1-D)^2}{1-(2-\alpha-\beta)\cdot D + (1-\alpha)\cdot D^2} \cdot \frac{\Delta \cdot D}{(1-D)^2} && (6.71) \\ &= \frac{\Delta \cdot D}{1-(2-\alpha-\beta)\cdot D + (1-\alpha)\cdot D^2} && (6.72) \end{aligned}$$

This steady-state phase error can also be determined through the final value theorem

$$\lim_{k\to\infty} \phi_k = \lim_{D\to 1} (1-D)\cdot\Phi(D) = 0 \quad . \tag{6.73}$$

Thus, as long as the designer chooses α and β within the stability limits, a second-order loop should be able to track any constant phase or frequency offset. One, however, must be careful in choosing α and β to reject noise, equivalently making the second-order loop too sharp or narrow in bandwidth, can also make its initial convergence to steady-state very slow. The trade-offs are left to the designer for any particular application.

Noise Analysis of the Second-Order PLL The phase-jitter's power transfer function from any phase noise component at the input to the second-order PLL to the output phase error is found to be:

$$\frac{|\Phi(D)|^2}{|\Theta(D)|^2} = \frac{|\, 1 - e^{-\jmath\omega} \,|^4}{|\, 1 - (2-\alpha-\beta)\cdot e^{-\jmath\omega} + (1-\alpha)\cdot e^{-2\jmath\omega} \,|^2} \quad . \tag{6.74}$$

This transfer-function can be calculated and multiplied by any input phase-noise power spectral density to get the phase noise at the output of the PLL. Various stable values for α and β (i.e., that satisfy Equations (6.66) and (6.67)) may be evaluated in terms of noise at the output and tracking speed of any phase or frequency offset changes. Such analysis becomes highly situation dependent.

Problem 6.6 is a good example of the trade-offs of this type of design for the interested reader (or student to which the problem was assigned as homework!).

Phase-Jitter Noise Specifically following the noise analysis, a phase-jitter noise may have a strong component at a specific frequency. Phase jitter often occurs at either the power-line frequency (50-60 Hz) or twice the power line frequency (100-120 Hz) in many systems.[7] In other situations, other radio-frequency or ringing components can be generated by a variety of devices operating within the vicinity of the receiver. In such a situation, the choices of α and β may be such as to try to cause a notch at the specific frequency. A higher-order loop filter might also be used to introduce a specific notch, but overall loop stability should be checked as well as the transfer function.

6.2.3 Phase-locking at rational multiples of the provided frequency

Clock frequencies or phase errors may not always be computed at the frequency of interest. Figure 6.9 illustrates the translation from a phase error measured at frequency $\frac{1}{T}$ to a frequency $\frac{p}{q} \cdot \frac{1}{T} = \frac{1}{T'}$ where

[7] This is because power supplies in the receiver may have a transformer from AC to DC internal voltages in chips or components and it is difficult to completely eliminate leakage of the energy going through the transformer via parasitic paths into other components.

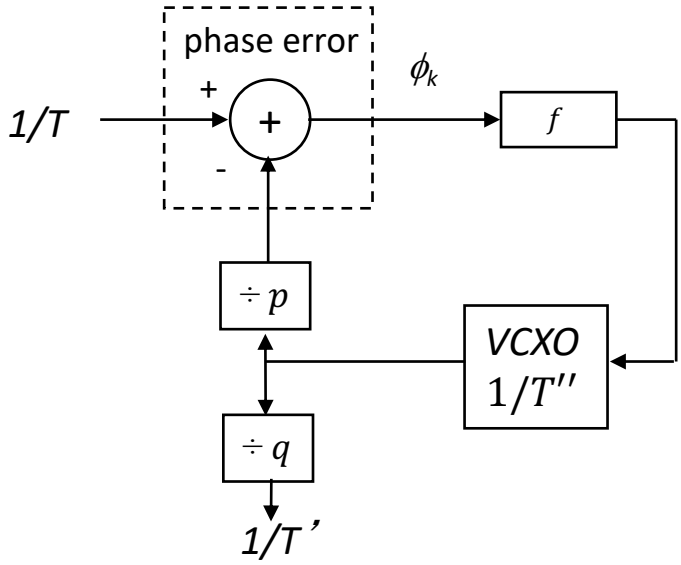


Figure 6.9: Phase-locking at rational fractions of a clock frequency.

p and q are any positive co-prime integers. The output frequency to be used from the PLL is $1/T'$. A high frequency clock with period

$$T'' = \frac{T}{p} = \frac{T'}{q} \tag{6.75}$$

is used so that

$$p \cdot T'' = T \tag{6.76}$$

$$q \cdot T'' = T' \quad . \tag{6.77}$$

The divider in Figure 6.9 uses the counter circuits of Figure 6.6. To avoid cycle slips, the local oscillator should be between $f_{lo} \cdot \big((1 - \frac{p}{2}), (1 + p)\big)$.

6.3 Symbol-Timing Synchronization

Usually in data transmission, the received signal does not include a "pilot," that is, a sinusoid synchronized to the symbol rate. The receiver instead derives this sinusoid from the received noisy signal $y(t)$. Thus, the unabetted PLL's studied so far would not be sufficient to recover the symbol rate. **Timing recovery** generates this symbol-rate sinusoid from the received channel signal for input to the PLL. There are two types of timing recovery. The first type is **open loop** timing recovery and does not use the receiver's message-related decisions. The second type is called **decision-directed** or **decision-aided** and uses the receiver's decisions. Both types approximate Equation (6.14)'s ML synchronization. Since the recovered symbol rate is used to sample the incoming waveform in most systems, care must be exerted in the higher-performance decision-directed methods that not too much delay appears between the sampling device and the decision device. Such delay can seriously degrade receiver performance or even render the phase-lock loop unstable.

Subsection 6.3.1 begins with the simpler open-loop methods and Subsection 6.3.2 then progresses to decision-directed methods.

6.3.1 Open-Loop Timing Recovery

Probably the simplest and most widely used timing-recovery method is Figure 6.10's **square-law timing-recovery** method.

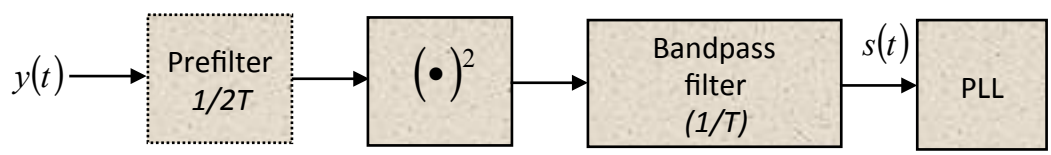


Figure 6.10: Square-law timing recovery.

The present analysis ignores the optional prefilter momentarily, in which case the nonlinear squaring device produces at its output

$$y^2(t) = \left[\sum_m x_m \cdot h(t - mT) + n(t)\right]^2 \quad . \tag{6.78}$$

The expected value of the square output (assuming, as usual, that the successively transmitted data symbols x_m are independent of one another) is

$$\begin{aligned} \mathbb{E}\left\{y^2(t)\right\} &= \sum_m \sum_n \mathcal{E}_{\boldsymbol{x}} \cdot \delta_{mn} \cdot h(t - mT) \cdot h(t - nT) + \sigma_n^2 & (6.79) \\ &= \mathcal{E}_{\boldsymbol{x}} \cdot \sum_m h^2(t - mT) + \sigma_n^2 \quad , & (6.80) \end{aligned}$$

which is periodic, with period T. The bandpass filter attempts to replace the statistical average in (6.80) by time-averaging. Equivalently, the output of the squaring device is the sum of its mean value and a zero-mean noise fluctuation about that mean,

$$y^2(t) = \mathbb{E}\left\{y^2(t)\right\} + \left(y^2(t) - \mathbb{E}\left\{y^2(t)\right\}\right) \quad . \tag{6.81}$$

The second term can be thought of as noise as far as the recovery of the timing information from the first term. The bandpass filter tries to reduce this noise. The instantaneous values for this noise depend upon the transmitted data pattern and can exhibit significant variation, leading to what is sometimes called

"data-dependent" timing jitter. The bandpass filter and PLL try to minimize this jitter. Sometimes "line codes" (or basis functions) are designed to insure that the underlying transmitted data pattern results in significantly less data-dependent jitter effects. Line codes are special sequential encoders (see Chapter 2) that use the system state to amplify clock-components in any particular data-symbol stream. Most (well-designed) systems typically do not need such codes[8] to force very low error probabilities (once in century levels). These systems must avoid a long string of constant transmitted (written) signal, and this is the main purpose of constrained codes, although they can also be used to help with avoidance of patterns that lead to nonlinear ISI that may not be otherwise easily removed (although there are methods to do so that often are better choices than constrained codes). For more information on this area, see [2], and they are thus not addressed in the current version of this text.

Generating the "2nd Harmonic:" While the signal in (6.80) is periodic, its amplitude may be small or even zero, depending on $h(t)$. This amplitude is essentially the value of the frequency-domain convolution $P(f) * P(f)$ evaluated at $f = 1/T$ – when $P(f)$ has small energy at $f = 1/(2T)$, then the output of this convolution typically has small energy at $f = 1/T$. If this desired energy is too small, other "even" nonlinear operations (absolute value, 4th power, etc.) can replace the squaring circuit. Use of these functions may be harder to analyze, but basically their use attempts to provide the desired sinusoidal output. The effectiveness of any particular nonlinear-operation choice depends upon the channel pulse response. The prefilter in Figure 6.11 can be used to eliminate signal components that are not near $1/2T$ so as to reduce noise further. The PLL at the end of sqare-law timing recovery works best when the sinusoidal component of $1/2T$ at the prefilter output has maximum amplitude relative to noise amplitude. This maximum amplitude occurs for a specific symbol-clock phase. The frequency $1/2T$ is the "bandedge". Symbol-timing recovery systems often try to select a timing phase (in addition to recovering the correct clock frequency) so that the bandedge component is maximized. Square-law timing recovery by itself does not guarantee a maximum bandedge component.

Envelope Timing Recovery: For QAM data transmission, Figure 6.11 illustrates square-law timing recovery's equivalent, **envelope timing recovery.**

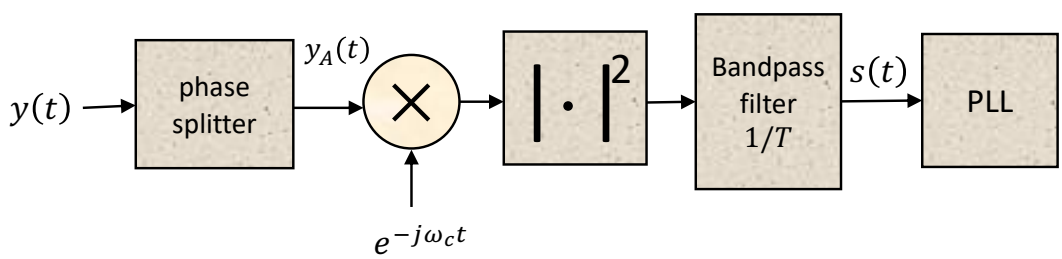


Figure 6.11: Envelope timing recovery.

Envelope-timing-recovery's analysis is basically the same as the real baseband case. The nonlinear element's choice produces real output, for instance generalization of absolute value is magnitude (square root of the sum of squares of the real and imaginary inputs). A problem with envelope timing is its à priori need for the carrier.

[8]These "constrained" codes are most heavily used in storage systems where higher-level systems cannot "resend" a lost information packet if the system determines that message packet has failed - which will happen eventually for any system with non-zero P_e. Storage systems therefore cannot fail (or "crash") and thus use error-correcting codes as in Chapters ??chapter8

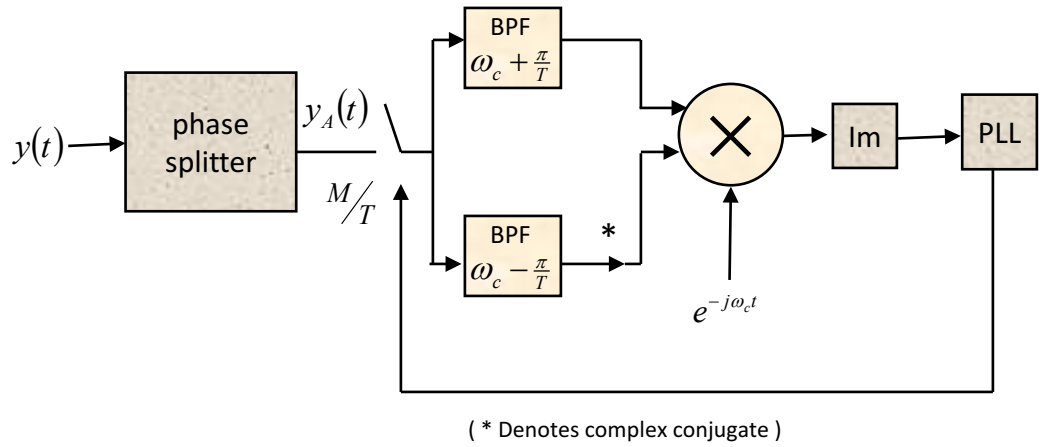


Figure 6.12: Bandedge timing recovery.

Bandedge Timing Recovery: Figure 6.12 illustrates one widely used method that avoids the carrier-frequency dependency. This is"bandedge" timing recovery. The two narrowband bandpass filters are identical, except for their center frequency. Chapter 's earlier discussion about fractionally spaced equalizers notes that timing-phase errors can lead to aliased nulls within the Nyquist band. The correct choice of timing phase should lead (approximately) to a maximum of the energy within the two bandedges. A Nyquist-Criterion-satisfying channel response is real at least after aliasing and summing of frequency-domain translates. Many Nyquist pulses are purely real and desirable. An equalized system would have equalizers, and timing phases, that could try to force this real channel response by adjusting phase among the possible choices to minimize MSE. While one equalizer could have a strange situation of exactly equal energy at frequencies $1/2T \pm \epsilon_f$ and $-1/2T \mp \epsilon_f$ (where ϵ_f is small frequency deviation) and opposite phases that would cancel exactly in aliasing, any imaginary component as $\epsilon_f \to 0$ would need also to be zero exactly at the Nyquist frequency or the MSE would increase. This is the basis of finding the best (MMSE) timing phase (for whatever equalizer might be present, but presumably a MMSE equalizer also for consistency of criteria at MMSE). At this band-edge-maximizing timing phase, the multiplier output following the two bandpass filters in Figure 6.12 should be maximum, meaning that quantity should be purely real. The PLL forces this timing phase by using the samples of the imaginary part of the multiplier output as the error signal for the phase-lock loop. While the carrier frequency is presumed to be known in the design of the bandpass filters, existence of a sinusoid at the carrier frequency is otherwise absent in "bandedge" timing recovery, making it a preferred method where the carrier frequency and symbol frequency may not be generated from the same source, that is - they do not have a known rational-fraction relationship.

6.3.2 Decision-Directed Timing Recovery

Decisions can also be used in timing recovery. A common decision-directed timing recovery method minimizes the mean-square error, over the sampling time phase, between the equalizer (if any) output and the decision, as in Figure 6.13. That is, the receiver chooses τ to minimize

$$J(\tau) = \mathbb{E}\left\{|\hat{x}_k - z(kT+\tau)|^2\right\} \quad , \tag{6.82}$$

where $z(kT+\tau)$ is the equalizer output (LE or DFE) at sampling time k corresponding to sampling phase τ. The update uses a stochastic-gradient estimate of τ in the opposite direction of the (unaveraged) derivative of $J(\tau)$ with respect to τ. This derivative is (letting $\epsilon_k \triangleq \hat{x}_k - z(kT+\tau)$)

$$\frac{dJ}{d\tau} = \Re\left[\mathbb{E}\left\{2\epsilon_k^* \cdot (-\frac{dz}{d\tau})\right\}\right] \quad . \tag{6.83}$$

The (second-order) update is then

$$\tau_{k+1} = \tau_k + \alpha \cdot \Re\{\epsilon_k^* \cdot \dot{z}\} + T_k \tag{6.84}$$
$$T_k = T_{k-1} + \beta \cdot \Re\{\epsilon_k^* \cdot \dot{z}\} \quad . \tag{6.85}$$

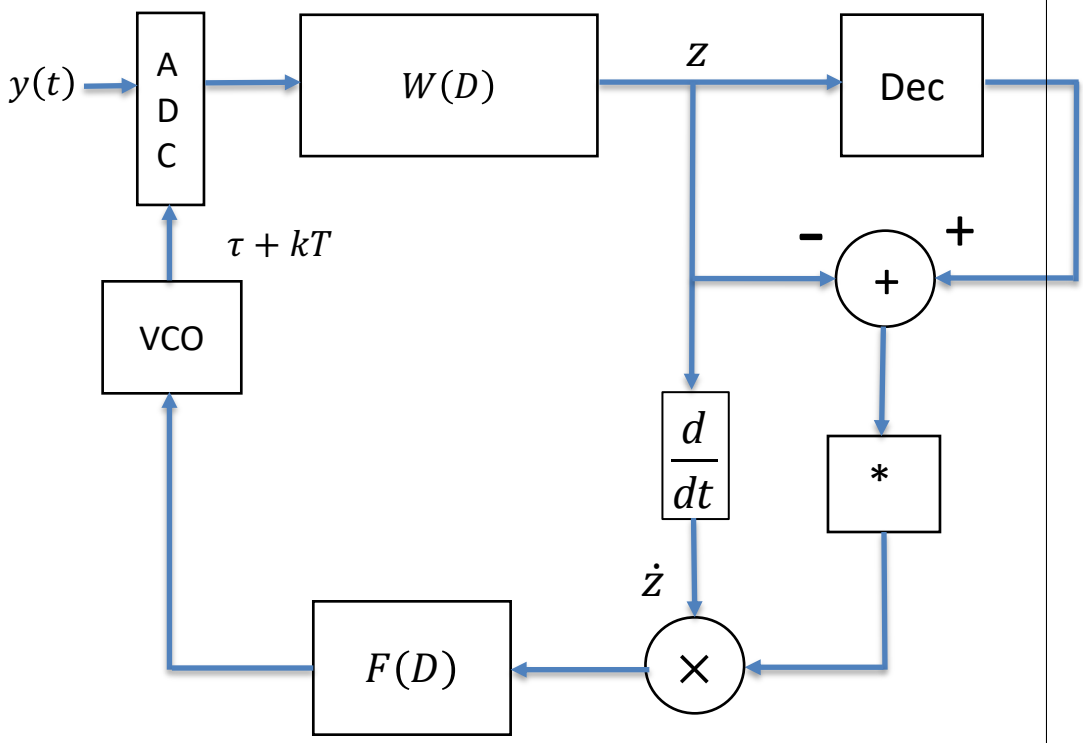


Figure 6.13: Decision-directed timing recovery.

Figure 6.13 illustrates this type of decision-directed phase-lock loop. There is one subtle decision-directed implementation problem, the differentiator. Differentiator implementation is simpler if the system's sampling rate is significantly higher than the symbol rate. Then the differentiation can be approximated by simple differences between adjacent decision inputs. However, a higher sampling rate can significantly increase system costs.

Another symbol-rate sampling approach is to assume that z_k corresponds to a bandlimited waveform within the Nyquist band:

$$z(t+\tau) = \sum_m z_m \cdot \text{sinc}(\frac{t+\tau-mT}{T}) \quad . \tag{6.86}$$

Then the derivative is

$$\frac{d}{dt} z(t+\tau) \triangleq \dot{z}(t+\tau)$$
$$= \sum_m z_m \cdot \frac{d}{dt}\text{sinc}(\frac{t+\tau-mT}{T}) \tag{6.87}$$
$$= \sum_m z_m \cdot \frac{1}{T} \cdot \left[\frac{\cos\left(\frac{\pi(t+\tau-mT)}{T}\right)}{\frac{t+\tau-mT}{T}} - \frac{\sin\left(\frac{\pi(t+\tau-mT)}{T}\right)}{\pi\left(\frac{t+\tau-mT}{T}\right)^2} \right] \tag{6.88}$$

which if evaluated at sampling times $t = kT - \tau$ simplifies to

$$\dot{z}(kT) = \sum_m z_m \cdot \left(\frac{(-1)^{k-m}}{(k-m)T} \right) \tag{6.89}$$

or

$$\dot{z}_k = z_k * g_k \quad , \tag{6.90}$$

where

$$g_k \triangleq \begin{cases} 0 & k = 0 \\ \frac{(-1)^k}{kT} & k \neq 0 \end{cases} \quad . \tag{6.91}$$

The problem with such a realization is the length of the filter g_k. The delay in realizing such a filter could seriously degrade the overall PLL performance. Sometimes, the derivative can be approximated using the two terms g_{-1} and g_1 by

$$\dot{z}_k = \frac{z_{k+1} - z_{k-1}}{2T} \quad . \tag{6.92}$$

The Timing Function and Baud-Rate Timing Recovery: The timing function is defined as the expected value of the error signal supplied to the loop filter in the PLL. For instance, in the case of the decision-directed loop using $\dot{z}_k$ in (6.92) the timing function is

$$\mathbb{E}\left\{u(\tau)\right\} = \Re\left[E\left\{\epsilon_k^* \cdot \frac{z_{k+1} - z_{k-1}}{T}\right\}\right] = \Re\frac{1}{T}\left[\mathbb{E}\left\{\hat{x}_k^* \cdot z_{k+1} - \hat{x}_k^* \cdot z_{k-1}\right\} + \mathbb{E}\left\{z_k^* \cdot z_{k-1} - z_k^* \cdot z_{k+1}\right\}\right] \tag{6.93}$$

the real parts of the last two terms of (6.93) cancel, assuming that the variation in sampling phase, τ, is small from sample to sample. Then, (6.93) simplifies to

$$\mathbb{E}\{u(\tau)\} = \mathcal{E}_{\boldsymbol{x}} \cdot [h(\tau - T) - h(\tau + T)] \quad , \tag{6.94}$$

essentially meaning the phase error is zero at a symmetry point of the pulse. The expectation in (6.94) is also the mean value of

$$u(\tau) = \hat{x}_k^* \cdot z_{k-1} - \hat{x}_{k-1}^* \cdot z_k \quad , \tag{6.95}$$

which can be computed without delay.

In general, the timing function can be expressed as

$$u(\tau) = G(\hat{\boldsymbol{x}}_k) \bullet \boldsymbol{z}_k \tag{6.96}$$

so that $\mathbb{E}[u(\tau)]$ is as in (6.94). The choice of the vector function G for various applications can depend on the type of channel and equalizer (if any) used, and $\boldsymbol{z}_k$ is a vector of current and past channel outputs.

6.3.3 Pilot Timing Recovery

In pilot timing recovery, the transmitter inserts a sinusoid of frequency equal to q/p times the the desired symbol rate. The PLL of Figure 6.9 can be used to recover the symbol rate at the receiver. Typically, pilots are inserted at unused frequencies in transmission. Section 6.5 will address more detailed use of pilots, which is typical in multi-carrier systems where one of the tones can be used.

6.3.4 Fractionally Spaced Digital Timing Recovery

Chapter 3 relates that fractionally spaced equalizers (FSEs) will automatically adjust for any constant timing-phase offset (and constant carrier-phase offset) so only symbol frequency is important to recover when there is an FSE. Usually fractionally spaced sampling implies some level of equalization.

Floyd Gardner[9] introduced a discrete-time phase-error generator that operates with sampling rate at (approximately) twice the symbol rate. This method is predicated on low intersymbol interference (ISI) to work properly (so it may imply the use of an FSE). The so-called "Gardner Loop" uses no

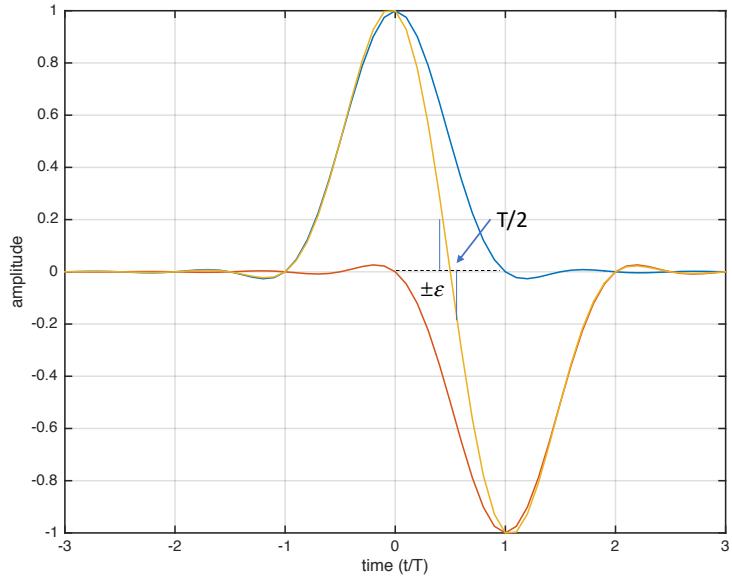


Figure 6.14: Illustration of timing error at $T/2$.

decisions and is independent of carrier frequency, and has found wide use in a number of systems that use "fractional spacing" (see Chapter 3).

Gardner's loop essentially uses the phase error

$$\phi_k = y_I(kT - \frac{T}{2}) \cdot [y_I(kT) - y_I(kT - T)] + y_Q(kT - \frac{T}{2}) \cdot [y_Q(kT) - y_Q(kT - T)] \quad . \tag{6.97}$$

A simple way to understand Gardner's phase error is to inspect the adjacent 100% raised-cosine pulses (of opposite sign) in Figure 6.14. When the pulse shape is symmetric (as Gardner assumed or an equalizer might ensure), the figure shows that if the difference (essentially ± 2) between two successive main symbol-rate sampling points is multiplied by the value of the sample half-way between those samples is a scaled timing-phase-error estimate ϵ). With symmetry, the error is zero when the zero crossing is exactly half way between the two main-symbol-rate samples (without ISI - or which would average to zero as it usually does). If there is lag the error (whether positive polarity follows negative or vice-versa) will be positive and if advance, the error is negative. This is true for both in-phase and quadrature samples independently, so they can be summed to get an overall phase estimate that is twice as good (signal to noise 3 dB higher). The Gardner Loop only works for BPSK and QPSK. If successive samples are of the same polarity, then the error is zero and no phase update occurs (because there is no timing information at that sampling instant).

While Gardner's 1986 derivation is more complex, it shows that an alternative error signal that has the same average as the one in Equation (6.97) is

$$\phi_k \approx |y(kT)|^2 - |y(kT - T)|^2 \quad . \tag{6.98}$$

The preferred implementation (less noise) would be that in Equation (6.97) although the squaring could be simpler in some implementations.

Gardner loops typically need the excess bandwidth to exceed 40% so that the intersymbol interference from other odd multiples of $T/2$ would be negligible in the error signal as is evident in the Figure (which is for 100% excess bandwidth raised cosine pulses). Gardner also showed that a (constant) phase offset

[9] May 1986, IEEE Trans on Comm, A BPSK/QPSK Timing -Error Detector for Sampled Receivers.

in carrier does not affect the phase error. The phase error independence can be seen by first writing the in-phase and quadrature components as

$$x_I(t) = a(t)\cos\Delta_\theta - b(t)\sin\Delta_\theta \tag{6.99}$$

and thus

$$x_Q(t) = a(t)\sin\Delta_\theta + b(t)\cos\Delta_\theta \quad . \tag{6.100}$$

Substitution of these into $\phi(t)$ yields

$$\begin{aligned}
\phi(t) &= [a(t-T/2)\cos\Delta_\theta - b(t-T/2)\sin\Delta_\theta]\cdot[a(t)\cos\Delta_\theta - b(t)\sin\Delta_\theta - a(t-T)\cos\Delta_\theta + b(t-T)\sin\Delta_\theta] \\
&+ [a(t-T/2)\sin\Delta_\theta + b(t-T/2)\cos\Delta_\theta]\cdot[a(t)\sin\Delta_\theta + b(t)\cos\Delta_\theta - a(t-T)\sin\Delta_\theta - b(t-T)\cos\Delta_\theta] \\
&= a(t-T/2)\cdot[a(t)-a(t-T)]\cos^2\Delta_\theta + b(t-T/2)\cdot[b(t)-b(t-T)]\sin^2\Delta_\theta \qquad (6.101) \\
&+ a(t-T/2)\cdot[a(t)-a(t-T)]\sin^2\Delta_\theta + b(t-T/2)\cdot[b(t)-b(t-T)]\cos^2\Delta_\theta \qquad (6.102) \\
&= a(t-T/2)\cdot[a(t)-a(t-T)] + b(t-T/2)\cdot[b(t)-b(t-T)] \quad . \qquad (6.103)
\end{aligned}$$

Thus, even with constant carrier-phase offset, the Gardner loop will have the same timing-phase error.

Richard Gitlin in 1987 instead[10] suggested a related method called "Center-Tap Tracking" that essentially recognizes that if an equalizer is already sampling at $T/2$, then the center tap position (which will be the largest tap in the real part of the equalizer when timing phase is correct) of the equalizer will move forward or backward when phase error occurs. Update of the largest tap position when $T/2$ spacing is used is essentially equivalent to movement of Gardner's $T/2$ error position by the same amount. In a more sophisticated version, the position tracked is the energy-median, meaning that the sum of squared tap magnitudes after this position equals the sum of squared tap magnitudes before this position. However, Gitlin's method works for all constellations (not just BPSK and QPSK). It is similarly be independent of constant carrier-phase offset as well because it is equivalent to the shift calculation above. Gitlin's Center-Tap Tracking need not use as much excess bandwidth because the center-tap movement should be independent of ISI on the off-symbol rate $T/2$ samples. However, it does require an equalizer be present, which Gardner loops do not strictly imply, although ISI will degrade the Gardner Loop.

[10] AT&T Technical Journal, Nov-Dec 1987, "Center-Tap Tracking Algorithms for Timing Recovery."

6.4 Carrier Recovery

Again in the case of carrier recovery, there are two basic types, open-loop and decision directed, as in Subsections 6.4.1 and 6.4.2 respectively.

6.4.1 Open-Loop Carrier Recovery

A model for the modulated signal with some unknown offset, θ, in the phase of the carrier signal is

$$y(t) = \Re\left\{\sum_m x_m \cdot p(t - mT) \cdot e^{\jmath(\omega_c t + \theta)}\right\} \quad , \tag{6.104}$$

or equivalently

$$y(t) = \frac{y_A(t) + y_A^*(t)}{2} \quad , \tag{6.105}$$

where

$$y_A(t) = \sum_m x_m \cdot p(t - mT) \cdot e^{\jmath(\omega_c t + \theta)} \quad . \tag{6.106}$$

The autocorrelation function for y is

$$\begin{aligned} r_y(\tau) &= \mathbb{E}\left\{y(t)y(t-\tau)\right\} && (6.107)\\ &= \mathbb{E}\left\{\left(\frac{y_A(t) + y_A^*(t)}{2}\right)\left(\frac{y_A(t-\tau) + y_A^*(t-\tau)}{2}\right)\right\} && (6.108)\\ &= \frac{1}{4}\mathbb{E}\left[y_A(t) \cdot y_A^*(t-\tau) + y_A^*(t) \cdot y_A(t-\tau)\right] \\ &+ \frac{1}{4}\mathbb{E}\left[y_A(t) \cdot y_A(t-\tau) + y_A^*(t) \cdot y_A^*(t-\tau)\right] && (6.109)\\ &= \frac{1}{2}\Re\left[r_{y_A}(\tau)\right] + \frac{1}{2}\Re\left[r_{y_A}(\tau) \cdot e^{2\jmath(\omega_c t + \theta)}\right] \quad . && (6.110) \end{aligned}$$

The average output of a squaring device applied to $y(t)$ is thus

$$\mathbb{E}\left\{y^2(t)\right\} = \frac{1}{2}\Re\left[r_{y_A}(0)\right] + \frac{1}{2}\Re\left[r_{y_A}(0)e^{2\jmath(\omega_c t + \theta)}\right] \quad , \tag{6.111}$$

which is a (constant plus a) sinusoid at twice the carrier frequency. The square-law carrier-recovery circuit is illustrated in

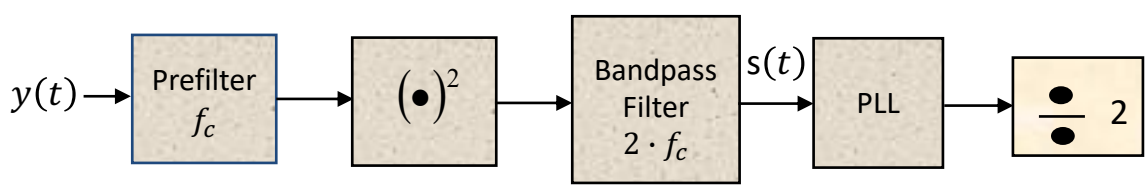


Figure 6.15: Open-loop carrier recovery.

Figure 6.15. Note the difference between this circuitry and the envelope timing recovery, where the latter squares the magnitude of the complex baseband equivalent for $y(t)$, whereas the circuit in Figure 6.15 squares the channel output directly (after possible prefiltering). The bandpass filter again tries to

average any data-dependent jitter components (and/or noise) from the squared signal, and the PLL can be used to tune further the sinusoidal phase accuracy. The output frequency is double the desired carrier frequency and is divided by 2 to achieve the carrier frequency. The division is easily implemented with a single "flip-flop" in digital circuitry. The pre-filter in Figure 6.15 can be used to reduce noise entering the PLL.

6.4.2 Decision-Directed Carrier Recovery

Decision-directed carrier recovery is more commonly encountered in practice than open loop carrier recovery. The basic concept used to derive the error signal for phase locking is illustrated

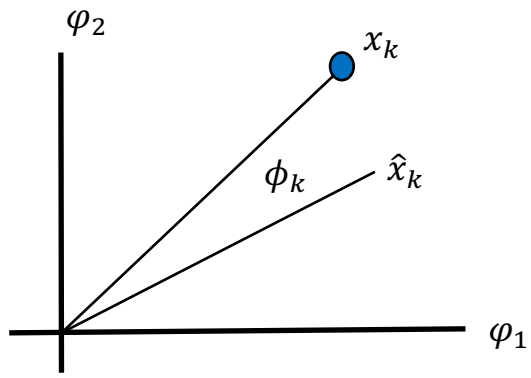


Figure 6.16: Decision-directed phase error.

in Figure 6.16. The decision-device input $\hat{x}_k$ is not exactly equal to the decision-device output x_k. Since

$$x_k = a_k + \jmath b_k \;\; ; \;\; \hat{x}_k = \hat{a}_k + \jmath \hat{b}_k \quad , \tag{6.112}$$

then

$$\frac{x_k}{\hat{x}_k} = \frac{|x_k|}{|\hat{x}_k|} \cdot e^{\jmath \phi_k} \tag{6.113}$$

$$= \frac{a_k + \jmath b_k}{\hat{a}_k + \jmath \hat{b}_k} \tag{6.114}$$

$$= \frac{\left(a_k \hat{a}_k + b_k \hat{b}_k\right) + \jmath \left(\hat{a}_k b_k - a_k \hat{b}_k\right)}{\hat{a}_k^2 + \hat{b}_k^2} \quad , \tag{6.115}$$

which leads to the result

$$\phi_k = \arctan \frac{\hat{a}_k b_k - a_k \hat{b}_k}{a_k \hat{a}_k + b_k \hat{b}_k} \tag{6.116}$$

$$\approx \arctan \frac{1}{\mathcal{E}_{\boldsymbol{x}}} \left(\hat{a}_k b_k - a_k \hat{b}_k\right) \tag{6.117}$$

$$\approx \frac{1}{\mathcal{E}_{\boldsymbol{x}}} \left(\hat{a}_k b_k - a_k \hat{b}_k\right) \tag{6.118}$$

$$\propto \left(\hat{a}_k b_k - a_k \hat{b}_k\right) \quad , \tag{6.119}$$

with the approximation being increasingly accurate for small phase offsets. Alternatively, a look-up table for arcsin or for arctan could be used to get a more accurate phase-error signal ϕ_k. For large constellations, the phase error must be smaller than that which would cause erroneous decisions most

of the time. Often, a training data pattern is used initially in place of the decisions to converge the carrier recovery before switching to unknown data and decisions. Using known training patterns, the phase error should not exceed π in magnitude since a modulo-2π detector is implied.

6.5 Phase Locking with Multicarrier Methods

6.5.1 Baseband Multicarrier - Wireline DMT

symbol and sample alignment

6.5.1.1 Pilots and Synchronization in DMT

In some transmission systems, the pilot is added at the Nyquist frequency exactly 90 degrees out of phase with the nominal +,-,+,- sequence that corresponds to the $1/T'$ DMT sampling clock. This insertion must be done in analog after the DAC if the system is not oversampled. Insertion 90 degrees out of phase means that the band-edge component is maximized when these samples have zero energy at this frequency (presuming the Nyquist frequency is not used for data transmission, which it rarely is in DMT systems anyway because it is only one real dimension available and tends to complicate loading-algorithm design for the same reason that DC is often avoided also in real-baseband DMT systems like all DSLs and Cable DOCSIS 3.1). This is the equivalent of band-edge timing (which when presented was for a passband complex system) shifted to baseband for the same observation of zero imaginary energy should appear at Nyquist when timing phase minimizes MMSE. Unfortunately, most DMT systems are designed so that the use of the full band to Nyquist is rare and only occurs on the shortest (least attenuation) channels so energy near the band-edge is very small. Thus other methods for timing recovery are also usually used (like pilots at other in-band frequencies or correlation of the cyclic prefix as in Section 6.5.

6.5.2 Pilot Carrier Recovery

In pilot carrier recovery, the transmitter inserts a sinusoid of frequency equal to q/p times the the desired carrier frequency. The PLL of Figure 6.9 can be used to recover the carrier rate at the receiver. Typically, pilots are inserted at unused frequencies in transmission. For instance, with OFDM and DMT systems in Chapter 4, a single tone may be used for a pilot. In this case, if the timing reference and the carrier are locked in some known rational relationship, recover of one pilot supplies both signals using dividers as in Figure 6.9. In some situations, the carrier frequencies may apply outside of the modem box itself (for instance a baseband feed of a QAM signal for digital TV) to a broadcaster who elsewhere decides the carrier frequency (or channel) and there provides the carrier translation of frequency band). The 5 pilors 802.11(a) WiFi systems also provide for the situation where the symbol clock and carrier frequency may not appear the same even if locked because of movement in a wireless environment (leading to what is called "Doppler Shift" of the carrier caused by relative motion). Then at least two pilots would be of value (and actually 5 are used to provide against loss of a few). The effect of jitter and noise are largely eliminated because the PLL sees no data-dependent jitter at the pilot frequency if the receiver filter preceding the PLL is sufficiently narrow.

6.5.3 Searcing for the guard period or pilots

Large noise essentially requires the cross-correlation of channel output and input as in the previous section. Repeated versions of the synch pattern may be required to get acceptable acquisition of packet boundary.

DMT's cyclic prefix in Chapter 4 is an example of a known pattern that can be used form frame (and timing) recovery. The relevant autocorrelation calculation of for a given DMT symbol size of $N+\nu$ samples with conjectured sampling times l and corresponding $l+N-1$ is (conjugate reduces to real part for real baseband)

$$\hat{R}_{\boldsymbol{yy}}(l) = \frac{1}{\nu} \cdot \sum_{n=0}^{\nu-1} y_{N-1-l-n} \cdot y^*_{0-l-n} \quad . \tag{6.120}$$

This should be maximum when $l=0$. The calculation can be averaged over M successive DMT symbols so

$$\hat{R}_{\boldsymbol{yy}}(l) = \frac{1}{\nu \cdot M} \cdot \sum_{m=0}^{M-1} \left[\sum_{n=0+m(N+\nu)}^{\nu-1+m(N+\nu)} y_{N-1-l-n} \cdot y^*_{0-l-n} \right] \quad . \tag{6.121}$$

This may also derive timing phase as well as frame boundary, as this correlation is (roughly) maximum when the timing phase in $(0T')$ is properly selected because this timing phase essentially makes the residual ISI energy outside the cyclic-prefix window smallest.

6.5.4 Passband Multicarrier - Wireless OFDM

Timing (slope) vs carrier offset (y intercept)

6.5.4.1 The channel-gain

FEQ vs interpolation
Use all tones for decision directed

6.5.4.2 Timing Recovery

symbol alignment and sample alignment

6.5.4.3 Carrier Recovery

binning for gross estimate
sensitivity analysis

6.5.4.4 Pilot and Synch Symbol use in Various OFDM Standards

6.6 Continuous Phase Modulation - Digital Frequency Modulation

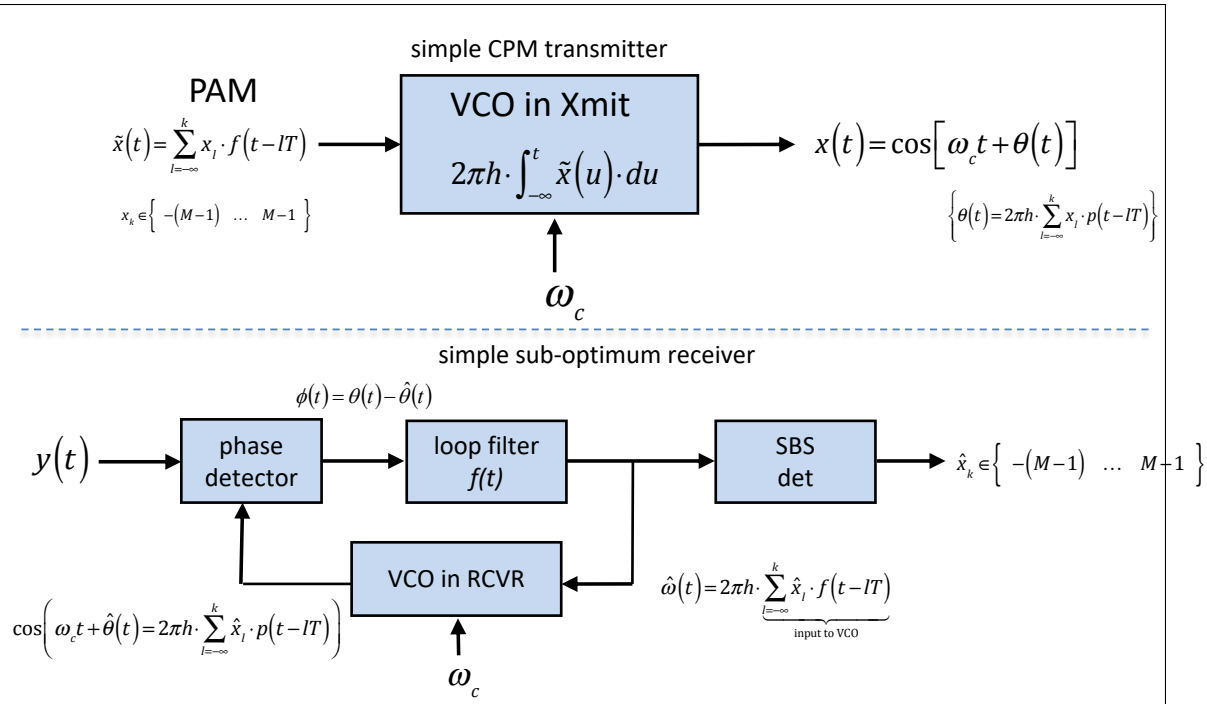


Figure 6.17: Simple CPM system.

Continuous Phase Modulation (CPM): CPM appears in this Chapter on synchronization because it is a form of modulation that combines elements of carrier-phase synchronization with modulation in a creative way. It is hard therefore to place CPM in any one topical area. Even the advanced all-digital modern versions of CPM will combine elements of detection with phase estimation. Thus, the author begs pardon of readers who find this placement of material somewhat unusual, and hopes the topic's address here (or anywhere else placed in this book) though will be appreciated and useful.

Earliest CPM forms can implement a modulator using a VCO as in the upper portion of Figure 6.17. This VCO's output-sinusoid's phase $\theta(t)$ varies according to a filtered PAM data signal with input signal levels $x_k = \pm 1, \dots \pm (M-1)$:

$$x(t) = \sqrt{2P_x} \cdot \cos\left[\omega_c t + \underbrace{2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)}_{\theta(t)}\right] \quad . \tag{6.122}$$

The quantity h is known as the **modulation index** and essentially scales phase-change level relative to the phase $\omega_c t$, the nominal **carrier frequency**'s (ω_c's) phase. The modulated waveform $x(t)$ can be viewed as a carrier-modulated signal (see Section 2.1) with inphase and quadrature components moving continuously in time along the circumference of a circle with radius $\sqrt{2P_x}$. P_x is the **average power** of the CPM signal. The sinusoidal signal $x(t)$ has frequency/phase that varies with the input message sequence and takes real amplitude values between $-\sqrt{2P_x}$ and $+\sqrt{2P_x}$.

The function $p(t)$ is known as the **phase pulse response** of the CPM signal. Typically, $p(t)$ is chosen as a continuous function so that $x(t)$'s phase changes are continuous. Smoother phase responses (those with one or more continuous derivatives) lead to more narrow spectra. Since a sinusoid has an infinite

number of continuous derivatives, the first discontinuous derivative of the phase pulse response $p(t)$ (via the chain rule for differentiation) leads to a corresponding discontinuous derivative of the modulated signal $x(t)$. If this is the nth derivative of phase, then this will correspond to $x(t)$'s $(n+1)^{th}$ derivative[11]. The Fourier transform of $x(t)$ then falls with increasing frequency as f^{n+2} for large f (and the power spectrum falls thus as f^{2n+4}). When $n = -1$, the phase itself is not continuous and phase changes abruptly. An example of $n = -1$ occurs for Chapter 1's PSK with a rectangular basis function. This PSK signal's power spectrum then falls as f^2 when $n = -1$. The pulse phase response is always chosen so that $p(\infty) = 1/2$, and its value is upper bounded close to $1/2$ for all t. Thus the maximum corresponding phase change is roughly $(M-1)h\pi$.

In addition to a narrow frequency spectrum, CPM signals have a (one-dimensional) peak-to-average ratio of 3 dB (and of 0 dB in two dimensions). This means that analog amplification can be very efficient in energy use, driving the transmit-amplifier gain into saturation or near-saturation so that maximum energy is transferred to the signal $x(t)$ itself rather than to bias currents in the amplifier electronics. CPM signals thus are often used in situations where power consumption is of paramount concern (often battery-powered devices or things) and particularly in wireless transmission where narrow spectrum also causes less interference to signals using adjacent frequencies. Some applications of CPM have been 2G wireless[12], Bluetooth systems where again battery-powering is common, and other very limited-power devices/communications between things and/or machines. The uplink of WiMAX (IEEE 802.16.1) also uses CPM, as do the Internet-of-Things low-power transmission standards like IEEE 802.15.4c and 802.15.4d.

6.6.0.1 The Frequency Pulse Response

Since the VCO modulator in Figure 6.17 and Equation (6.122) produces a sinusoidal output with frequency that is proportional to the frequency offset (time derivative of phase offset $\theta(t)$), CPM signals are often also described by a **frequency pulse response** $f(t)$ such that $\dot{p}(t) = f(t)$ or equivalently

$$p(t) = \int_{-\infty}^{t} f(u) \cdot du \quad . \tag{6.123}$$

The sinusoidal CPM signal's frequency is thus

$$\omega = \omega_c + \frac{d\theta}{dt} = \omega_c + 2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot f(t - lT) \quad . \tag{6.124}$$

Interpretting (6.124) directly, the name **Digital Frequency Modulation** is also commonly and correctly encountered for this type of modulation (so CPM and DFM may be used interchangeably in the literature).

A simple (but not often optimum) receiver could again use a VCO inside a PLL and simply have it track the received sinusoid's phase, as also shown in Figure 6.17. The receiver's VCO input (which can also be fed into a detector) is then the information-bearing signal

$$\hat{x}(t) = \sum_{l=0}^{k} x_l \cdot f(t - lT) \quad , \tag{6.125}$$

with corresponding respective VCO-output phase and frequency

$$\hat{\theta}(t) = 2\pi h \cdot \sum_{l} x_l \cdot p(t - lT) + n_\phi(t) \tag{6.126}$$

$$\hat{\omega}(t) = \sum_{l} x_l \cdot f(t - lT) + n_\omega(t) \quad , \tag{6.127}$$

[11] Strictly speaking point discontinuities in the phase derivatives could be allowed at the zero-crossings of the original sinusoid but this case is of little interest in practice and ignored here.

[12] so called GSM smartphones of the turn of the century when battery-powered portable phones had limited battery life

where $n_\phi(t)$ represents the noise in the phase-error for the receiver's PLL (and similarly $n_\omega(t)$ the noise in the frequency estimate). Clearly larger modulation index h could lead to a PLL requirement for a very fast response (large values of α and β as defined in the PLLs earlier in this chapter), but that increases the phase noise that passes to the PLL output phase estimate. The signal in (6.127) could be decoded like any ISI signal, where $f(t)$ is viewed as the pulse response. If $f(t)$ is for instance a raised-cosine waveform (a popular form of CPM), then simple symbol-by-symbol detection on the phase may be an attractive simple (possibly sub-optimum) detector. An even simpler receiver/detector is the detector often used for analog FM signals, which is the simple diode-capacitor "discriminator" circuit shown in Figure 6.18. The discriminator design selects the resistor R and capacitor C values (with knowledge of the carrier frequency) to hold the charge (set $RC << 1/\omega_c$) until the next (rectified) cycle recharges the capacitor to the current level of the received waveform. The discriminator output's amplitude will be proportional to the frequency offset (and thus no PLL is needed for the carrier, although a symbol clock will be necessary to detect the discriminator output data symbols x_l) after lowpass filtering. (Symbol timing can be recovered by any of the methods discussed earlier in this chapter.)

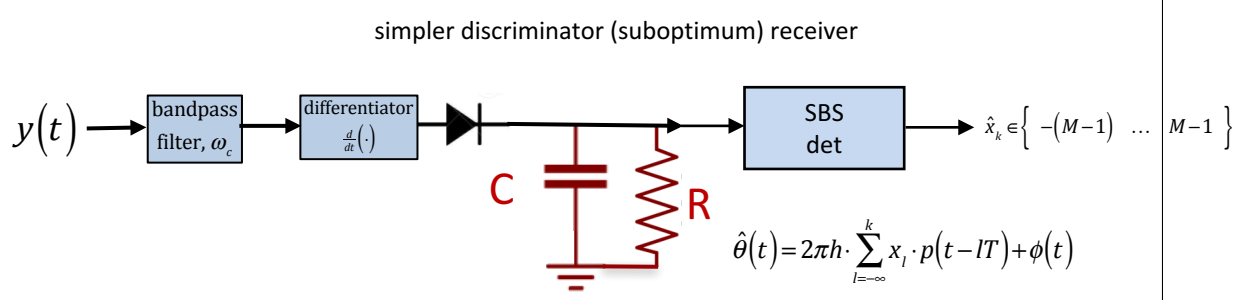


Figure 6.18: Simpler CPM discriminator receiver.

6.6.0.2 Finite-length Frequency Pulse Responses

When $f(t) = 0 \; \forall \; t \notin [0 \le t \le \nu T]$ with non-negative integer $\nu \in Z$, the frequency pulse response is **finite length**. It is called **partial response** when $\nu \ge 1$ and **full response** when $\nu = 0$. (QPSK and BPSK are examples of full response.) Finite-length systems have a finite number of waveforms that can be transmitted as long as the modulation index h is of the form

$$h = \frac{2\kappa}{q} \tag{6.128}$$

with k and q co-prime. The number of possible waveforms depends on past on x_{k-1}, x_{k-2}, ..., $x_{k-\nu+1}$ and also a phase (recalling that $p(t > \nu T) = 1/2$) for earlier values of symbol times (meaning samples $x_{k-(l \ge \nu)}$) that can only take a maximum of q possibilities (modulo 2π). This number of distant-past phase values derives from

$$\sum 2\pi \left(\frac{2\kappa}{q}\right) \cdot \left(p(t - l \ge \nu T) = \frac{1}{2}\right) = \sum \frac{2\kappa\pi}{q} \quad , \tag{6.129}$$

which when summed will be even-integer multiples of π/q. Such π/q multiples, modulo 2π, cannot assume more than q values (in special cases, the number of possible values can be less q). There is thus a set of states S that includes all $M^{\nu-1}$ possible values for the vector of samples $\boldsymbol{x}_{k-1:k-\nu} \triangleq [x_{k-1} \; x_{k-2} \; ..., x_{k-\nu+1}]$ that also could be paired (in S) with any of the up to q distant past phase values. Thus, the maximum number of such (noise-free) waveforms is

$$|S| = q \cdot M^{\nu-1} \quad . \tag{6.130}$$

In effect Chapter 1's **ML signal detector** (Section 1.3) need only have $|S|$ matched filters (even though the number of waveforms is infinitely countable). Subsection 6.6.5 will address such a finite-complexity

maximum-likelihood detection further, paralleling to some degree Chapter 9's Viterbi detectors of Chapter 9.

6.6.1 CPM passband and baseband representations

A CPM signal can be rewritten in passband form as

$$x(t) = \sqrt{2P_x} \cdot \cos\left[2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)\right] \cdot \cos(\omega_c t) - \sqrt{2P_x} \cdot \sin\left[2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)\right] \cdot \sin(\omega_c t) \quad , \tag{6.131}$$

leaving baseband equivalent

$$\begin{aligned} x_{bb}(t) &= \sqrt{2P_x}\left\{\cos\left[2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)\right] + \jmath \cdot \sin\left[2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)\right]\right\} && (6.132) \\ &= \sqrt{2P_x} \cdot e^{\jmath 2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)} && (6.133) \\ &= x_A(t) \cdot e^{-\jmath \omega_c t} \quad . && (6.134) \end{aligned}$$

The baseband equivalent mathematical model has artificially a factor of 2 more power, so as in Chapter 2, a phase-splitting demodulator that would demodulate the carrier would also scale by $1/\sqrt{2}$ leaving

$$\tilde{x}_{bb} = P_x \cdot e^{\jmath 2\pi h \cdot \sum_{l=-\infty}^{k} x_l \cdot p(t-lT)} \quad , \tag{6.135}$$

with a flat PSD (or energy per dimension at the output of a normalized receiver matched filter) of $\bar{\mathcal{E}}_{\boldsymbol{x}} = P_x T/2$.

6.6.2 Common CPM filters

6.6.2.1 The rectangular (REC) frequency filter family

REC frequency filters are indexed by the finite-length-response parameter ν. This positive integer ν is the filter memory in the same sense it was used for FIR channels (partial response and equalization) in Chapter 3, except now ν describes the non-zero length of the frequency pulse response. REC systems are partial-response and have a frequency pulse response analogous to those described in Section 3.8.2 of Chapter 3. The REC frequency pulse response is:

$$f_{\nu\text{REC}}(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2\nu T} & 0 \le t \le \nu T \\ 0 & t \ge \nu T \end{cases} \quad . \tag{6.136}$$

The corresponding phase pulse response is found through simple integration as

$$p_{\nu\text{REC}}(t) = \begin{cases} 0 & t < 0 \\ \frac{t}{2\nu T} & 0 \le t \le \nu T \\ \frac{1}{2} & t \ge \nu T \end{cases} \quad . \tag{6.137}$$

Figure 6.19 illustrates the REC phase response filters for several values of ν. The transitions between new input phase symbols x_k to their final values rise linearly from time kT up to time $(k+\nu)T$, which means the phase will be a function of the current symbol value and the last ν symbol values (of which there are $M^\nu \cdot q$ possibilities of phase evolution). When $\nu = 1$, there is a continuous linear ramp in phase from one symbol to the next. There are M possible previous phase symbol values, from which that linear transition could come. The **phase tree** or **phase trellis** of Figure 6.20 illustrates this for

$M = 2$ and $\nu = 1$ with $h = 1/2$, one of the simplest and most widely used forms of CPM known as **Minimum Shift Keying (MSK)**. When $M > 2$, but $\nu = 1$ and h remains at 0.5, this 1REC family has the special name **CPFSK (continuous-phase frequency-shift keying)**; for all other values of $\nu > 1$, it is known as νREC with the value of h specified[13].

[13] some authors use L or LREC where this text respectively uses ν or νREC for consistency within this text with earlier chapters' notation

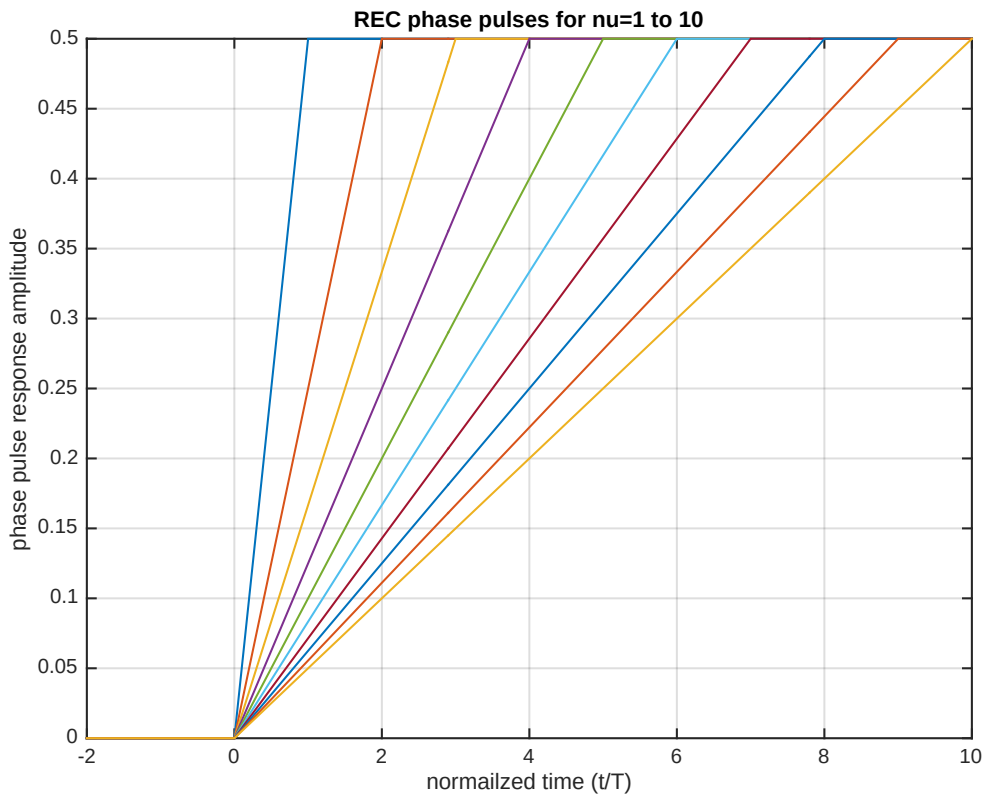


Figure 6.19: Phase pulse responses for rectangular CPM.

Figure 6.20's MSK phase tree illustrates the phase's continuous linear growth from each symbol instant to the next. While the phase is continuous, it's first derivative is not and thus $n = 0$, so that MSK signals will have spectra that decrease as f^4 in power at large f, which is better than nominal PSK's (or QAM's) drop with frequency as f^2 when for instance the phase changes abruptly from last value at each symbol instance. Larger values of h will increase the size of phase transitions and typically thus widen the REC frequency spectra.

Figure 6.20 shows that at any symbol instant, the modulator can be in one of two states. Sometimes those two states are 0 or $\pm\pi$ (say at even symbol-sample-time instants), while those two states are instead $\pm\frac{\pi}{2}$ on odd symbol-sample-time instants. This text calls such a diagram a "trellis diagram" in Chapter 9 and beyond. The number of states is equal to $2^{\nu-1} \cdot q = 4$, even though only 2 states are active at any symbol instant. **Maximum Likelihood Sequence Decoding (MLSD)** of encoded signals that conform to such a trellis can be simplified by a procedure introduced in Chapter 9 that is called the **Viterbi Algorithm** by communication engineers. Subsection 6.6.5 further discusses ML detectors in CPM. In effect, Figure 6.20's trellis diagram also plots the phase progression (which is linear for MSK) according to

$$\theta_k = \theta_{k-1} + \frac{\pi}{2T} \cdot (t - kT) \cdot x_k \quad . \tag{6.138}$$

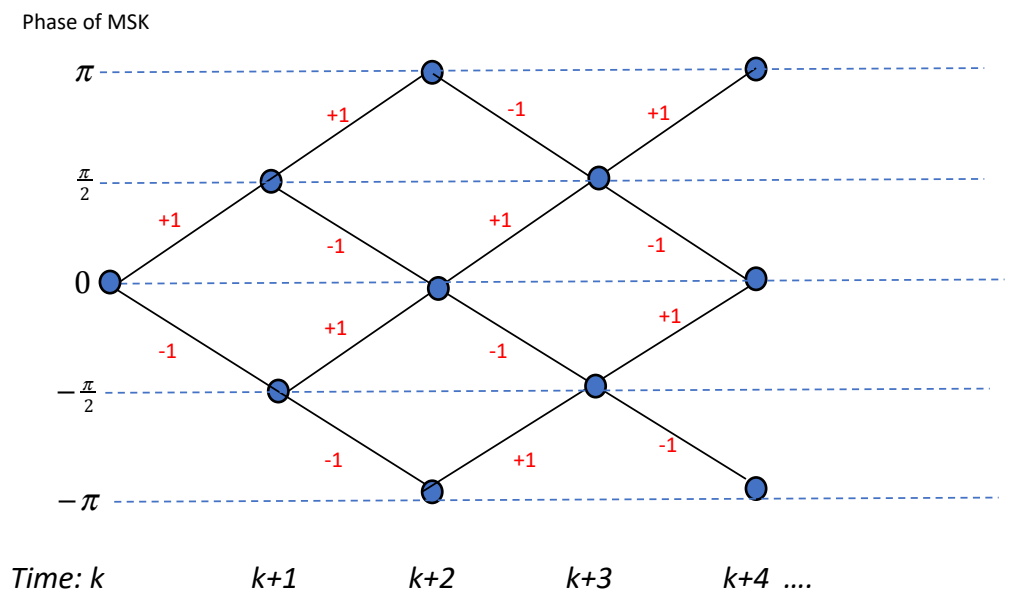


Figure 6.20: MSK Phase-Trellis Diagram.

For more general νREC with $h = 1/2$, there can be M^ν states at any stage (time) of the trellis, corresponding the previous phases that still affect the current state's phase value. M^ν at even times and M^ν different states at odd times, which is $M^{\nu-1} \cdot q = 2^{\nu-1} \cdot 4 = 2^{\nu+1}$ total states.

6.6.2.2 The raised-cosine frequency filter family

Raised Cosine (RC) frequency-pulse-response filters are also indexed by the same (as REC) parameter ν that characterizes the non-zero filter length in symbol periods from time $t = 0$ to time $t = \nu T$. The

RC frequency filter is:

$$f_{\nu\mathrm{RC}}(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{2\nu T} \cdot \left[1 - \cos\left(\frac{2\pi t}{2\nu T}\right)\right] & 0 \leq t \leq \nu T \\ 0 & t \geq \nu T \end{cases} . \tag{6.139}$$

The corresponding νRC phase pulse response is found through simple integration as

$$p_{\nu\mathrm{RC}}(t) = \begin{cases} 0 & t < 0 \\ \frac{t}{2\nu T} - \frac{1}{4\pi} \cdot \sin\left(\frac{2\pi t}{2\nu T}\right) & 0 \leq t \leq \nu T \\ \frac{1}{2} & t \geq \nu T \end{cases} . \tag{6.140}$$

νRC's phase pulse response is smoother than νREC's, and $p(t)$ has a continuous first derivatives so that $n = 1$ and RC spectra should decrease as f^6. Figure 6.21 plots the phase pulse responses, which can be compared to those for REC in Figure 6.19 to see the smoother transitions.

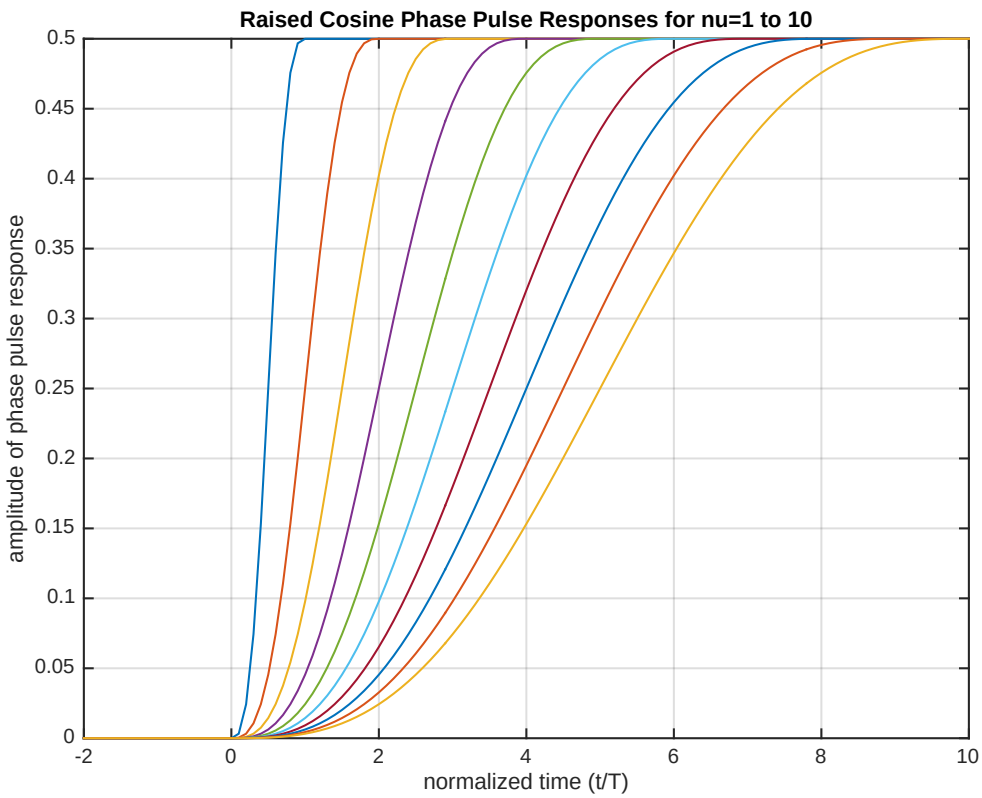


Figure 6.21: Phase pulse responses for Raised Cosine CPM.

A trellis diagram for $M = 2$, and $\nu = 1$ appears for RC in Figure 6.22. This is basically the same diagram as Figure 6.20, except this diagram has curved phase changes (the curves are not exact and meant to illustrate) between symbol instants that show the phase changes are not only continuous, but so are their derivatives, consistent with the design of a smoother phase change.

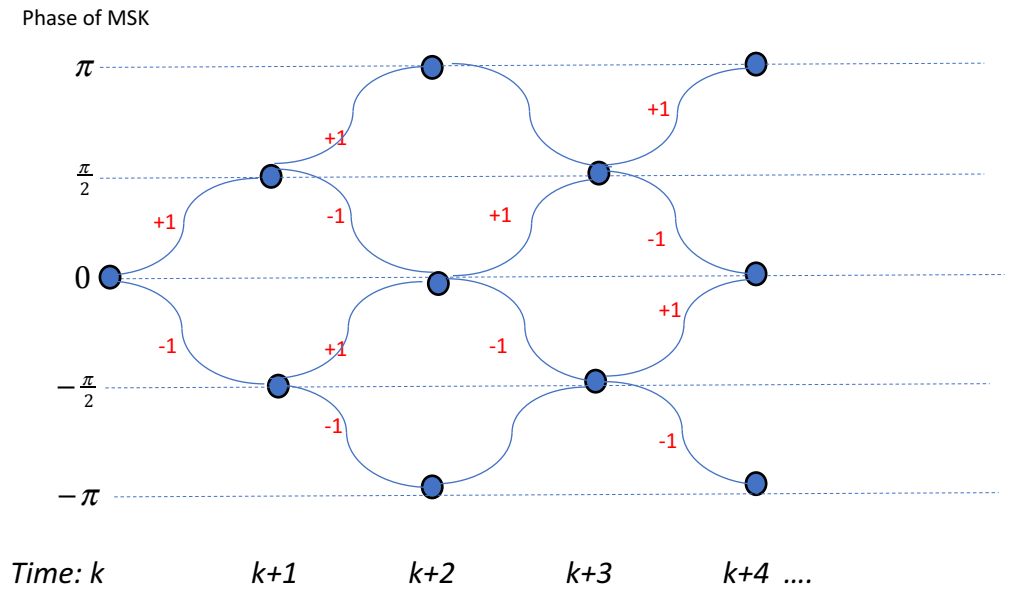


Figure 6.22: RC Phase-Trellis Diagram.

A related form of RC is the **HSC (Half-Cycle Sinusoid)** given by

$$f_{\nu\text{HCS}}(t) = \begin{cases} 0 & t < 0 \\ \frac{\pi}{4\nu T} \cdot \left[\sin\left(\frac{\pi t}{\nu T}\right)\right] & 0 \le t \le \nu T \\ 0 & t \ge \nu T \end{cases} . \tag{6.141}$$

Problem 6.11 studies various aspects of the HCS CPM waveform.

6.6.2.3 The spectrally raised-cosine (SRC) frequency filter family

Spectrally Raised Cosine (SRC) frequency pulse responses are given by

$$f_{n\text{SRC}}(t) = \frac{1}{nT} \cdot \text{sinc}\left(\frac{2t}{nT}\right) \cdot \frac{\cos\left(\frac{2\pi t}{nT}\right)}{1 - \left(\frac{4\alpha t}{nT}\right)^2} . \tag{6.142}$$

In this case, the raised cosine is in spectrum (not in time like the RC phase pulse responses). The parameter α is an excess bandwidth parameter between 0 and 1 as in Chapter 3. The memory parameter n is not exactly the filter length ν for SRC; however, SRC's main lobe in time is nT symbol periods wide. The overall response is infinite in length. One reasonable rule for truncation is to go back/forward an additional $\nu = 3n$ symbol periods in time before/after the main lobe, where the consequent truncation error is negligible. However, comparison with RC suggests that because of the main-lobe width, this characterization by $n = \nu$ is intuitively appealing. For this reason, some authors like Sundberg set

$\nu = n$ in their designation of νSRC, which really should be nSRC with ν reserved for the length of the partial-response in this textbook. Figure 6.23 plots the phase pulse responses for nSRC, which are yet more smooth ($p(t)$'s first 3 derivatives exist) and spectra decline with large frequency is as f^8. This figure uses 100% excess bandwidth in the phase pulse response.

6.6.2.4 The Gaussian Minimum Phase-Shift Keying frequency pulse-response filter family

Gaussian MSK essentially combines MSK (really 1REC) with a Gaussian pulse shape (with infinite number of derivatives) as in Figure 6.24.

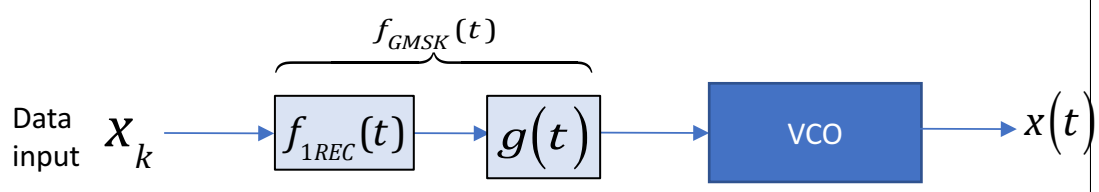


Figure 6.24: Simple GMSK modulator.

The first filter is the 1REC frequency-pulse-response filter of Equation (6.136), advanced by $T/2$, which can also be viewed as the difference between two integrators from $-\infty$ to $+T/2$ andfrom $-\infty$ to $-T/2$. The second filter is a Gaussian function with variance

$$\sigma_{MSK} = \frac{\ln 2}{4\pi^2 B^2} \tag{6.143}$$

where B is a bandwidth parameter such that $0 < BT < \infty$. Essentially B is a measure of $1/\nu$ that was used in earlier CPM phase-pulse-response waveforms. Thus, the second Gaussian filter is

$$g(t) = B \cdot \sqrt{\frac{2\pi}{\ln 2}} \cdot e^{-\frac{2\pi^2 B^2 t^2}{\ln 2}} \quad . \tag{6.144}$$

Recognizing that linear-time-invariant convolution is commutative, the GMSK frequency response can then also be directly written (using the well-known Q-function) as

$$f_{\text{GMSK}}(t) = \frac{1}{2T}\left[Q\left(2\pi B \cdot \frac{t+T/2}{\sqrt{\ln 2}}\right) - Q\left(2\pi B \cdot \frac{t-T/2}{\sqrt{\ln 2}}\right)\right] \quad . \tag{6.145}$$

Figure 6.25 plots the very smooth phase responses of GMSK for $B = 1/n,\ n = 1, ..., 10$.

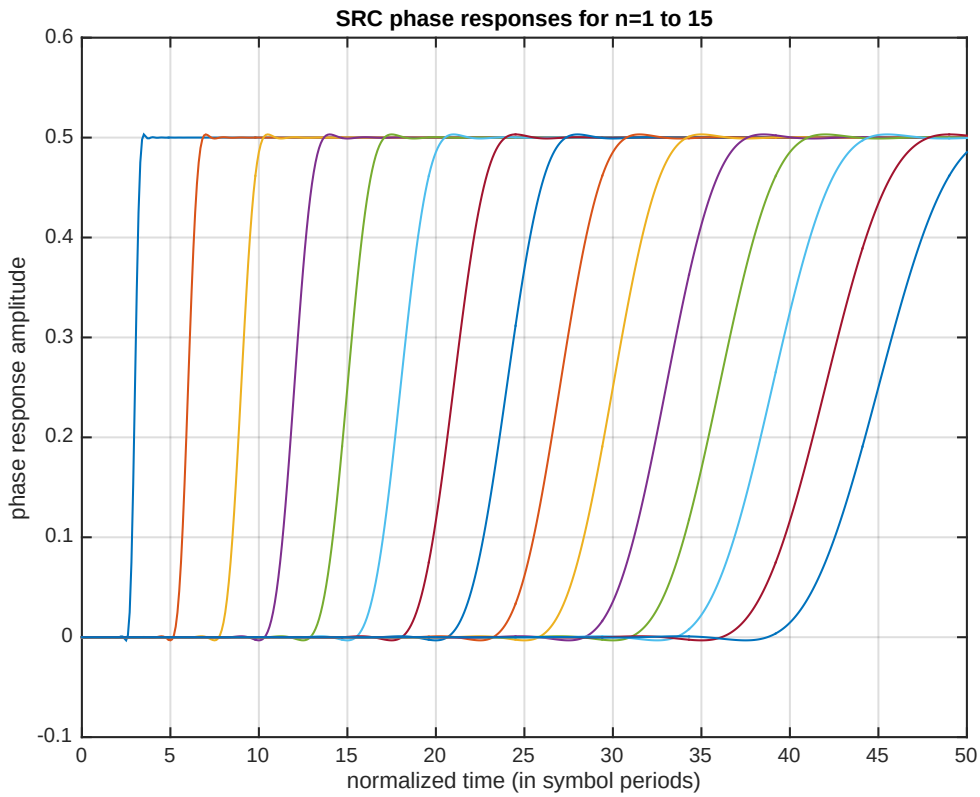


Figure 6.23: Phase pulse responses for Spectrally Raised Cosine CPM with 100% excess bandwidth.

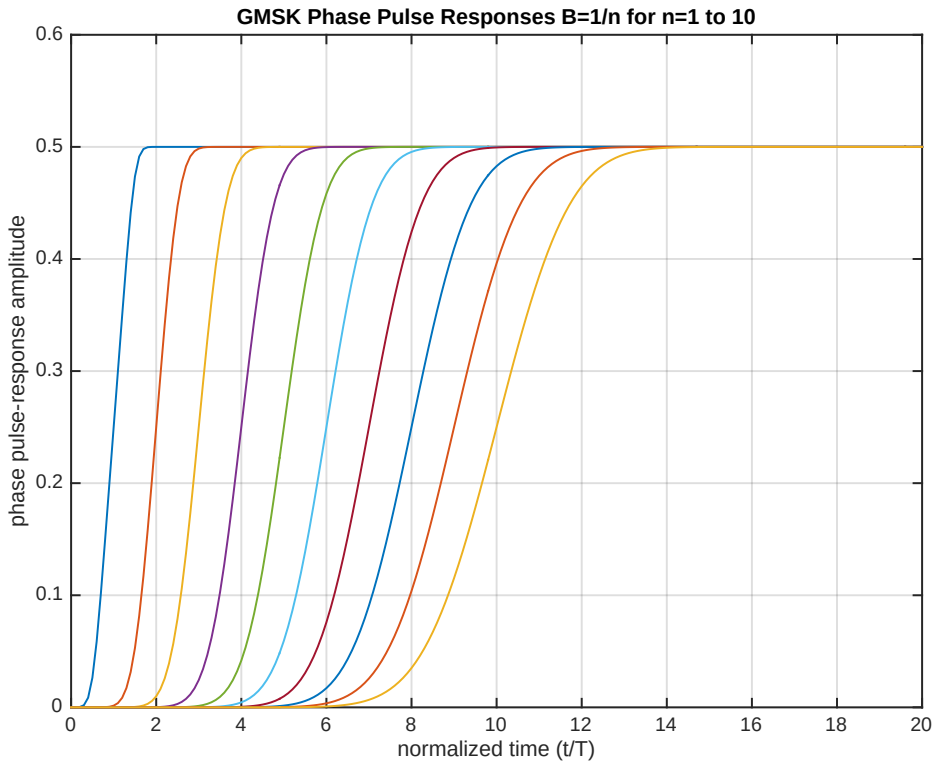


Figure 6.25: GMSK Phase Pulse Responses.

This author has found that a value for the memory length ν of a truncated GSM pulse (when $B = 1/n$ and n is a positive integer) is $\nu = 1/B + 10$.

Gaussian MSK is used in Bluetooth and also in the 2G wireless systems known as GSM.

6.6.2.5 Tamed Frequency Modulation

Tamed Frequency Modulation (TFM) is known for its steep frequency roll-off. It is used in IEEE 802.16.1 Uplink (in the "WiMax" series of standards, while the downlink is OFDM). The frequency pulse response is specified through use of the smooth function

$$f_0(t) = \frac{1}{T} \cdot \text{sinc}\left(\frac{t}{T}\right) \cdot \left[1 - \frac{2 - \frac{2\pi t}{T} \cdot \cot\left(\frac{\pi t}{T}\right) - (\pi t/T)^2}{24 \cdot (t/T)^2}\right] \quad , \tag{6.146}$$

as

$$f_{\text{GTFM}}(t) = \frac{1}{8} \cdot [a \cdot f_0(t-T) + b \cdot f_0(t) + a \cdot g_0(t+T)] \quad . \tag{6.147}$$

For TFM, $a = 1$ and $b = 2$. For Generalized TFM,

$$2 \cdot a + b = 4 \tag{6.148}$$

so that that then $p(\infty) = 0.5$. $f_0(t)$ is sometimes called a "Nqyuist-3" filter because it is a product of a "Nyquist-1" (flat frequency response, so sinc function in time domain) and another filter that satisfies the Nyquist criterion. The value of $f_0(t)$ is found in Problem 6.12 as $1 + \pi^2/72$. This author has found that $\nu = 11$ should be sufficient when truncating TFM (symmetrically about its center, so $5T$ on each side).

6.6.3 Non-coherent CPM Detection

Earliest approaches to DFM/CPM systems used the discriminator receiver of Figure 6.18. This receiver provided a noise advantage in early FM radio through its "differentiation" (essentially the received signal and noise are differentiated, which on the signal part leads to the desired baseband signal because differentiation of a sinusoid and stripping of the resultant sinusoidal waveform leaves a signal proportional to the frequency.[14]) When the noise has a flat (white) spectrum and the signal energy is concentrated at low frequencies (like an analog audio signal in early analog FM radio), then such a system improves the SNR[15]. However, if the information signal roughly occupies the entire bandwidth with a spectrum roughly similar to the noise power spectra, both signal and noise roughly experience the same filtering so there is no real SNR improvement. Such receivers have no need of carrier-clock recovery and are called **non-coherent**. Non-coherent detection nonetheless loses 3 dB of SNR because of the received signal's rectification, which now has a non-zero DC component that is 1/2 the received energy. Since DC carries no information, only the remaining half of the energy carriers the information-bearing part of the original transmitted signal.

The receiver could also use a PLL with VCO as a (sub-optimum) receiver where the VCO input is a noisy version of the CPM-input-frequency signal $\sum_l x_l \cdot f(t - lT)$. A simple PAM-like detector can be used on this noisy VCO-input signal (in parallel with the PLL). For some $\nu = 1$ frequency pulse responses previously described, there is zero or very little intersymbol interference in this signal. However, simple slicing with PAM-like decision regions is not an optimum detector (neither MAP nor ML) because the differentiated channel-output noise need no longer be white. Furthermore the two transmit dimensions of the original carrier-modulated signal $x(t)$ have been "compressed" to a one-dimensional signal that is essentially the data-modulated frequency. The receiver PLL must have sufficiently wide bandwidth to track the modulation in the CPM signal's phase. Thus, unlike a traditional PLL that recovers only a constant carrier frequency, such a VCO-based receiver necessarily passes noise with the same gain as the signal (the PLL parameters α and β can not be chosen too small). The receiver PLL phase error $\phi(t) = \theta(t) - \hat{\theta}(t)$ (presuming a little delay in the estimate through the loop and the noise is white) has twice the noise power. Another view is that the VCO-based receiver essentially loses a dimension that otherwise would have been used to increase minimum distance by 3 dB. Thus the VCO-based detector is thus also viewed as "non-coherent" in that the carrier is not really determined separately from the data signal and loses 3 dB.

Thus, both the discriminator and the VCO-based receivers lose 3 dB (at least). The performance analysis of this subsection is thus directed at this case. Subsection 6.6.5 will address a **coherent receiver** where the carrier recovery is achieved separately (i.e., just for ω_c and not the modulation too) where the 3dB is not lost. Both coherent and non-coherent detection will need to recover the symbol-clock, which can be achieved with any of the earlier methods in this chapter once the phase signal has been recovered by a VCO or a discriminator. Such conventional symbol-clock recovery will be assumed in the sequel.

[14] $d/dt \sin(\omega_c t) = \omega_c \cos(\omega_c t)$ so that when the amplitude of the resultant sinusoid is retained after the discriminator and low-pass filter, just the information-bearing frequency-proportional signal remains.

[15] Such a system effectively "wastes bandwidth to get a higher SNR"

6.6.3.1 Non-Coherent Decision Feedback or known state - symbol-by-symbol detection

The simplest forms of coherent (see Subsection 6.6.5) and non-coherent detection (after DC removed) assume knowledge of the previous ν phases[16] transmitted at the receiver. Such a decision-feedback receiver is rarely optimal ML. When previously transmitted phase values are known (or equivalently previous $x_{k-1}...x_{k-\nu}$ are known), and since the phase-pulse response is known also, the effect of previous phases can be subtracted from the current phase. Essentially the remaining phase-signal is a PAM signal with an amplitude that is really only important in relationship to whatever noise now appears in that same phase signal. For the non-coherent detectors of this subsection, this is the same as applying a DFE with the decision regions for M PAM to the discriminator output (or equivalently to the receiver VCO input's parallel path to detection). These non-coherent DFE detectors will have a performance reduced by at least 3 dB with respect to a coherent optical detector as discussed further in Subsection 6.6.5. However, error propagation can be more significant in CPM because of the substantial contribution of controlled phase/frequency from previous symbols, particularly as ν grows. A DFE can never outperform an ML detector – in CPM with large ν, the assumption of correct decisions can lead to the (incorrectly) projected performance being higher than ML – thus, the design must be cognizant to check for this. Section 6.6.6 will discuss minimum distances with ML detection and how they may be informative to bounding the performance projections of this section (when too optimistic because correct previous decisions are assumed).

Focusing on the symbol-by-symbol PAM detector here, the usual PAM formula $P_e = \frac{3}{M-1} Q\left(\sqrt{\frac{3 \cdot \text{SNR}}{\sigma_\theta^2}}\right)$ might be applied. However, one of the major motivations for CPM use is low power and another is low spectral leakage into adjacent signals in frequency. Thus, the noise from other signals (presumably also CPM) is determined by how low CPM's "sidebands" are relative to the signal power of the in-band signal. When other adjacent (presumably) CPM signals occupy adjacent frequencies, the receiver is limited by white noise when those other signals' sidelobe energy is well below the white noise power. For the VCO-based non-coherent receiver, the loss is basically the 3 dB coherent-detector loss; however the discriminator losses yet more because of the noise differentiation. In both cases, the signal power of the detected output signal is

$$\mathcal{E}_{cpm} = (2\pi h)^2 \frac{M^2-1}{3} = \frac{4\pi^2 h^2 (M^2-1)}{3} \quad . \tag{6.149}$$

The differentiated white noise of the discriminator output has power (normalizing symbol rate to $T = 1$) that is multiplied by the differentiator power transfer $|\jmath 2\pi f|^2$ or

$$\mathcal{E}_{dis} = 4\pi^2 \frac{\mathcal{N}_0}{2} \cdot \int_{-1/2}^{1/2} f^2 \cdot df = \frac{\mathcal{N}_0 \pi^2}{6} \quad , \tag{6.150}$$

which is larger than the doubled noise of the VCO-based noise power spectral density $\mathcal{N}_0$ by the factor $\pi^2/6 = 1.65$ or 2.16 dB. Thus, the discriminator may be the simplest form of detector but tends to lose another roughly 2 dB beyond the 3 dB lost to non-coherent detection. Even with no error propagation, the CPM signal to noise ratio needs to be at least 10.5 (BPSK) + 3 (non-coherent) = 13.5 dB for a 2-level PAM CPM signal to be detected with $P_e \leq 10^{-6}$. For the discriminator receiver, this required SNR would need to increase to roughly 15 dB. For QPSK, then 18 dB. The non-coherent-disriminator 5 dB loss could be acceptable because the transmit peak power of the CPM signal can often be greater by several dB (for the same battery power) than a linearly modulated signal (for instance a signal like OFDM). As Subsection 6.6.5 will show, the 3 dB loss can be avoided and indeed it is possible to do yet a few dB better by using a (more complicated) maximum likelihood detector that exploits CPM's phase transition history. These statements largely apply only to channels with little intersymbol interference. Subsection 6.6.6 will further review additional losses that can be bounded by ML minimum distance arguments.

CPM, as stated previously, makes most sense when there is a severe limitation on transmit energy and/or adjacent channel interference from (or into) adjacent channel asynchronous signals is of paramount concern. Chapter 4 showed that DMT/OFDM zero entirely adjacent channel interference,

[16] For SRC, GMSK, and GTFM, some appropriate truncated length of the infinite-length phase responses is assumed.

but this is when all the channels are presumed synchronized to the same carrier and data-symbol clocks. In some situations such synchronization may be physically impossible because different transmitters are in physically distinct locations (as well as are the receivers) while they share the same transmission resources (bandwidth). CPM can make sense in such situations, but white noise will not be limiting – instead it is the "sideband energy" of the adjacent receivers that is important. There is really no ideal theory that can be applied in this case, so CPM researchers have used reasonable insight to suggest that the CPM-carrier spacing (for same number of bits/Hz) can be equal to the point at which 99% of the total PSD energy is contained, leaving the remaining 1% presumably in the next few adjacent side bands. From the perspective of the on-channel signal, this means an SNR of roughly 20dB or better (so better than the 18 dB above for 2 bits/Hz or $\bar{b} = 1$, QPSK-equivalent data rates at $M = 4$ and better yet for $M = 2$).

This point can be computed for the various methods, once the power spectra are known. The next subsection discusses the calculation of CPM power spectra, and then the 99% frequencies are tabulated for each type of CPM.

6.6.4 Calculation of CPM Spectra

This section roughly follows a spectrum-calculation method that was introduced by T. Aulin and C.E. Sundberg[17], as "An Easy Way to Calculate Power Spectra for Digital FM." The word "easy" may have many interpretations, but the authors there meant really "less difficult" than full implementation or simulation of the CPM modulator and measuring its output. At several points, this development will deviate from their work to avoid multiple levels of integration that often cause difficulty in Matlab programs. The corresponding Matlab source code for implementation appears in Appendix E.

Appendix A relates that the wide-sense stationary (time averaged over one symbol period, and one carrier period where pertinent, as well as statistically averaged over data symbol and/or noise distributions) baseband-equivalent autocorrelation function for a signal $x(t)$ and the original signal's autocorrelation are related through:

$$r_x(\tau) = \frac{1}{2}\Re\left\{r_{bb}(\tau)e^{\jmath\omega_c\tau}\right\} = \frac{1}{2}\Re\left\{r_A(\tau)\right\} \quad . \tag{6.151}$$

Consistently then, the original CPM signal has power $r_x(0) = P_x$ with $r_{bb}(0) = 2P_x$. This means that recovery of the signal $\tilde{x}_{bb}(t)$ with Chapter 2's scaled demodulator will have $\tilde{r}_{bb}(\tau)$ with power $\tilde{r}_{bb}(0) = P_x$ ad energy/dimension $\bar{\mathcal{E}}_{\boldsymbol{x}} = \mathcal{E}_{\boldsymbol{x}}/2 = P_x T/2$ in each dimension, as well as noise energy per dimension $\frac{\mathcal{N}_0}{2}$. It will be convenient here to compute the baseband autocorrelation function $\tilde{r}_{bb}(\tau)$ directly from the CPM signal and then to compute its corresponding power spectrum through the Fourier Transform $\tilde{S}_{bb}(f) = \mathcal{F}\left[\tilde{r}_{bb}(\tau)\right]$. Thus, (with angle brackets denoting time average)

$$\tilde{r}_{bb}(\tau) = P_x \left\langle E\left\{e^{\jmath[\theta(t)-\theta(t-\tau)]}\right\}\right\rangle \quad , \tag{6.152}$$

which with wide-sense stationarity is the same as

$$\tilde{r}_{bb}(\tau) = P_x \left\langle E\left\{e^{\jmath[\theta(t+\tau)-\theta(t)]}\right\}\right\rangle \quad . \tag{6.153}$$

Substitution into the CPM signal's phase expression with the phase pulse response yields (assuming *i. i. d.* uniformly distributed PAM symbol values for x_l).

$$\tilde{r}_{bb}(\tau) = P_x \left\langle E\left\{e^{\jmath 2\pi h\cdot\left[\sum_{l=-\infty}^{k} x_l\cdot(p(t+\tau-lT)-p(t-lT))\right]}\right\}\right\rangle \tag{6.154}$$

$$= P_x \left\langle \prod_{l=-\infty}^{k} \frac{1}{M} \sum_{\substack{i=-(M-1)\\ i\text{ odd}}}^{M-1} e^{\jmath 2\pi h\cdot i\cdot(p(t+\tau-lT)-p(t-lT))}\right\rangle \tag{6.155}$$

$$= \frac{P_x}{MT}\cdot\int_0^T \prod_{l=-\infty}^{k} \sum_{\substack{i=-(M-1)\\ i\text{ odd}}}^{M-1} e^{\jmath 2\pi h\cdot i\cdot(p(t+\tau-lT)-p(t-lT))}\, dt \quad . \tag{6.156}$$

[17] Proc IEEE, Vol. 130, Part F, No. 6, October 1983 pp. 519-526.

The lag τ will be indexed by the multiple of T and an offset $\tau' \in [0, T]$ so that

$$\tau = \tau' + mT \quad m = 0, 1, 2, ... \tag{6.157}$$

Thus,

$$\tilde{r}_{bb,m}(\tau') = \frac{P_x}{MT} \cdot \int_0^T \prod_{l=-\infty}^{k} \sum_{\substack{i=-(M-1) \\ i \text{ odd}}}^{M-1} e^{j2\pi h \cdot i \cdot \left(p(t+\tau'-[l-m]T)-p(t-lT)\right)} \, dt \quad m = 0, 1, 2, ... \quad . \tag{6.158}$$

Further simplification arrives through including only non-unity product terms. For the time range of the integral ($[0, T]$), there are only some product index values (for l) that cause the difference $p(t + \tau - [l - m]T) - p(t - lT) \neq 0$. For large negative time arguments, these arguments become $0 - 0 = 0$ and for large positive time arguments, this difference becomes $1/2 - 1/2 = 0$.

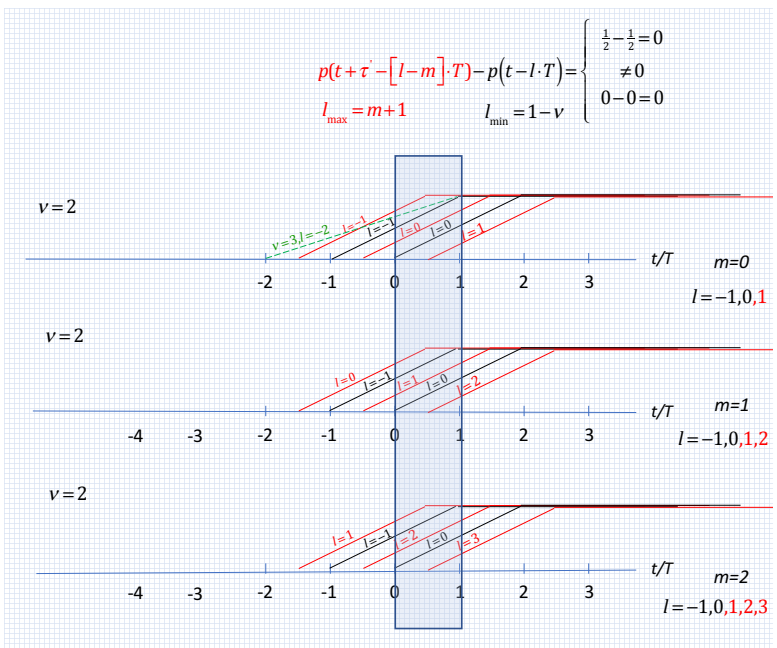


Figure 6.26: Illustration of phase-pulse-response differences that overlap the interval $[0, T]$.

Figure 6.26 illustrates the non-zero differences for the integral interval of $[0, T]$ for a 2REC pulse ($\nu = 2$). τ' is set at 0.5 in this graph for example purposes. In the upper portion for $m = 0$, there are two black pulses corresponding to $p(t+T)$ and $p(t)$ for $l = -1$ and $l = 0$ respectively, where these are the only two translates of $p(t - lT)$ that are not constant (at values 0 or 1/2) in the integral's interval $[0, T]$ (which is shown shaded). The 3 red pulses shown are translates (again for $m = 0$ in the upper graph) of $p(t + \tau' - [m - l]T)$ that also are neither 0 nor 1/2 in the interval $[0, T]$. For $m = 0$, the possibly l values (translates) are $l = -1, 0$ for the black lines and $l = 0, 1$ for the red lines, so that there are only 3 indices that contribute in this interval (other than irrelevant $0 - 0 = 1/2 - 1/2 = 0$ values).

Again in Figure 6.26 for $m = 1$, essentially the same 5 waveforms (2 black and 3 red) contribute, but the red indices have changed to $0, 1, 2$. For $m = 2$, the 5 curves now correspond to (as for all m values) the same two black translate values of $l = -1$ and 0, but now the red translates are $l = 1, 2, 3$. In general, the 5 translates for $\nu = 2$ pulses are captured by the range $l \in 1 - \nu, ..., 0, ...m + 1$. The green curve for $\nu = 3$ is shown just to illustrate that larger ν will lead to more translates that overlap

the shaded $[0,T]$ integral range. Thus, Equation (6.158) can be rewritten

$$\tilde{r}_{bb,m}(\tau') = \frac{P_x}{MT} \cdot \int_0^T \prod_{l=1-\nu}^{m+1} \sum_{\substack{i=-(M-1) \\ i \text{ odd}}}^{M-1} e^{\jmath 2\pi h \cdot i \cdot \left(p(t+\tau'-[l-m]T)-p(t-lT)\right)} dt \quad m = 0,1,2,\ldots \quad . \tag{6.159}$$

The product-sum inside the integral in Equation (6.159) is a function of the four inputs τ', t, T, and m and is a subroutine in the Matlab programs of Appendix E.

While the analysis used REC pulses, any CPM phase pulses can be used as long as ν is chosen sufficiently large in truncating such a possibly (otherwise) infinite-length pulse response so that its values prior to time zero are approximately constant at 0, and its values after time νT are approximately constant at 1/2. For REC and RC pulses, this is exactly true, but for SRC, GMSK, and TFM pulses, this is approximated by having sufficient numbers of symbol periods in the truncated response used to approximate these pulses (which we also will call ν from 0 to νT).

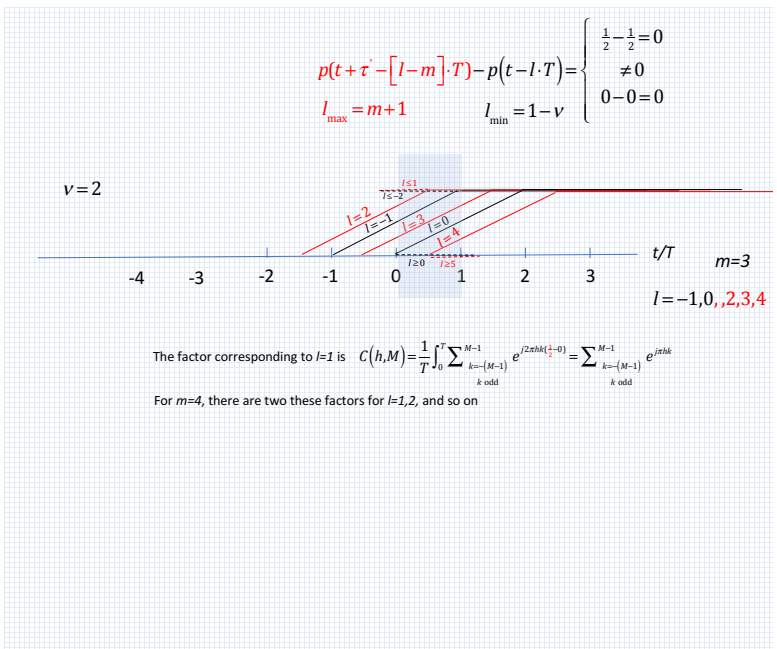


Figure 6.27: Illustration of phase-pulse-response differences that overlap the interval $[0,T]$ when $m > \nu$.

There is further observation possible from Figure 6.27 that shows more detail for the case where $\nu = 2$ and $m = 3 > \nu$. For values of $m > \nu$ in general, some of the product terms inside the integral range are effectively duplicated, although multiplied by a constant independent of τ. Some CPM autocorrelation functions $\tilde{r}_{bb,m}(\tau)$ are of infinite time extent, where the observation of the $m - \nu$ "repeated integration components" can be exploited. These repeated components occur when the red curves are always constant at 1/2 in the $[0,T]$ interval, but the black curves are constant at 0. In effect there is an extra repeated multiplicative factor of the term with $p(t+\tau'-[m-l]T) = 1/2$ and $p(t-lT) = 0$, which corresponds to the constant

$$C(h,M) \triangleq \sum_{\substack{i=-(M-1) \\ i \text{ odd}}}^{M-1} e^{\jmath 2\pi h \cdot i \cdot (\frac{1}{2}-0)} = \sum_{\substack{i=-(M-1) \\ i \text{ odd}}}^{M-1} e^{\jmath \pi h \cdot i} \quad . \tag{6.160}$$

The constant $C(h,M)$ is such that $|C| \le 1$. When $\nu > m$, the autocorrelation function then becomes

$$\tilde{r}_{bb,m+1}(\tau') = C(h,M) \cdot \tilde{r}_{bb,m}(\tau') = [C(h,M)]^{m-\nu}\, \tilde{r}_{bb,\nu}(\tau') \quad . \tag{6.161}$$

This defines essentially an exponential series for each value of τ' in the interval $[0, T]$ so that $\tilde{r}_{bb}(\tau)$ can be computed from its values over the internal $[0, (\nu+1)T]$ with the values for $[\nu T, (\nu+1)T]$ repeating (but for multiplication by an additional factor of C) for each additional interval of duration T increase in the value of τ.

The baseband spectrum is the Fourier transform of the baseband autocorrelation function. This development deviates from Aulin and Sundberg in that multiple integrals nested inside Matlab subroutine calls with corresponding nested integrals can be problematic (there are integrals for the phase-pulse-response as well as for the time averaging, and then creating a 3rd integration for the Fourier Transform). Instead of computing the continuous time Fourier Transform, the DFT is used instead with 10000 points, spaced at $T/100$. The function $\tilde{r}_{bb}(\tau)$ is thus sampled at $\tau = kT/100$ and passed through a DFT, after ensuring it is periodic over one interval of 10000 points. The sampling increases amplitude with respect to the continuous-frequency Fourier Transform of the autocorrelation by the factor $100/T$. Thus, if $T = 1$ as in the plots to come, then they have been reduced by 20 dB (factor of 100) from the FFT Matlab program output to get the correct amplitude for the continuous-frequency Fourier transform (which is the true continuous-frequency power spectral density in this case). The Matlab programs of Appendix E include this factor, as do the graphs that follow (except Figure 6.28, which normalizes all plots to have unit gain at $f = 0$).

Also of interest will be the frequency (normalized to $1/(T \cdot \log_2(M))$ or the Hz/bit (inverse of $2\bar{b}$) below which 99% of the energy is contained. This frequency will allow simplified comparison of spectral efficiency when the added white noise is not a factor (thus performance is dominated by adjacent band energy). In this section, then the Hz/bits/second or $0.5/\bar{b} \triangleq T_b$ will be provided for this 99% point.

6.6.4.1 REC Spectra

Figure 6.28 illustrates the simple MSK spectrum (for which $h = 1/2$ and $M = 2$). These were plotted using the Matlab programs in Appendix E for the REC pulses. However, the following example provides a useful closed form expression for the autocorrelation functions and for the MSK autocorrelation function and corresponding spectrum by directly evaluating (6.26) for this special case and then taking its Fourier Transform directly. For this spectra, $T_b = 0.6$, meaning that essentially for $M = 2$ that the bandwidth beyond the bare minimum necessary (which uses infinite-length synch function in time and up to frequency 0.5 is 20% with a very finite-length MSK implementation).

EXAMPLE 6.6.1 *[Closed form Expressions for MSK]* Equation (6.158) provides a general way to compute the autocorrelation function in time-lag segments of length T. For MSK, the constant $C(h = 1/2, M = 2) = 0$ so that the autocorrelation function is non-zero only for (positive) lags τ up to $2T$. Again for MSK, $h = 1/2$ and $M = 2$ (noting that $\cos x = \frac{1}{2}(e^{\jmath x} + e^{-\jmath x})$, Equation 6.158 can be evaluated for $m = 0$ and for $m = 1$ because this autocorrelation function has nonzero positive lag extent $(\nu+1)T = 2T$:

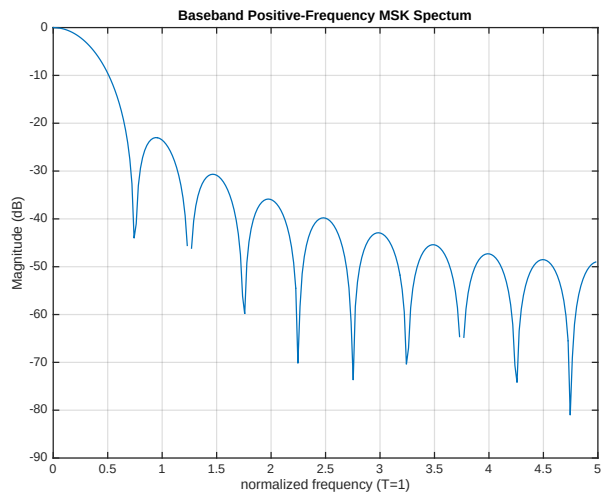


Figure 6.28: MSK Spectra ($h = 1/2$, $M = 2$, 1REC) - $f = 0$ is normalized to 0 dB.

$m = 0$, $h = 0.5$, $M = 2$, $\tau = \tau'$:

$$r_{bb,0}(\tau) = \frac{1}{T}\int_0^T \underbrace{\cos\left[\pi p(t+\tau) - \pi p(t)\right]}_{l=0} \cdot \underbrace{\cos\left[\pi p(t+\tau-T) - \pi p(t-T)\right]}_{l=1} dt \tag{6.162}$$

$$= \frac{1}{T}\int_0^{T-\tau} \underbrace{\cos\left[\frac{\pi(t+\tau)}{2T} - \frac{\pi t}{2T}\right]}_{l=0} \cdot \underbrace{\cos\left[0-0\right]}_{l=1} dt$$

$$+\frac{1}{T}\int_{T-\tau}^{T} \underbrace{\cos\left[\frac{\pi}{2} - \frac{\pi t}{2T}\right]}_{l=0} \cdot \underbrace{\cos\left[\frac{\pi(t+\tau-T)}{2T} - 0\right]}_{l=1} dt \tag{6.163}$$

$$= \frac{1}{T}\int_0^{T-\tau} \cos\left[\frac{\pi\tau}{2T}\right] dt + \frac{1}{T}\int_{T-\tau}^{T} \underbrace{\cos\left[\frac{\pi}{2} - \frac{\pi t}{2T}\right]}_{\sin(\pi t/2T)} \cdot \underbrace{\cos\left[\frac{\pi}{2T}(t+\tau-T)\right]}_{\sin[\pi(t+\tau)/2T]} dt$$

$$= (1-\tau/T)\cdot\cos\left[\frac{\pi\tau}{2T}\right] + \frac{1}{2T}\int_{T-\tau}^{T} \cos\left[\frac{\pi\tau}{2T}\right] - \cos\left[\frac{\pi t}{T} + \frac{\pi\tau}{2T}\right] dt \tag{6.164}$$

$$= (1-\tau/T)\cdot\cos\left[\frac{\pi\tau}{2T}\right] + \frac{\tau}{2T}\cos\left[\frac{\pi\tau}{2T}\right] - \frac{1}{2\pi}\left\{\sin\left[\pi + \frac{\pi\tau}{2T}\right] - \sin\left[\pi - \frac{\pi\tau}{T} + \frac{\pi\tau}{2T}\right]\right\}$$

$$= (1-\frac{\tau}{2T})\cdot\cos\left[\frac{\pi\tau}{2T}\right] + \frac{1}{2\pi}\left\{\sin\left[\frac{\pi\tau}{2T}\right] + \sin\left[\frac{\pi\tau}{2T}\right]\right\} \tag{6.165}$$

$$= (1-\frac{\tau}{2T})\cdot\cos\left[\frac{\pi\tau}{2T}\right] + \frac{1}{\pi}\sin\left[\frac{\pi\tau}{2T}\right] \tag{6.166}$$

$m = 1$, $h = 0.5$, $M = 2$ $\tau = T + \tau'$:

$$
\begin{aligned}
r_{bb,1}(\tau) &= \frac{1}{T}\int_0^T \underbrace{\cos\left[\pi p(t+\tau'+T) - \pi p(t)\right]}_{l=0} \cdot \underbrace{\cos\left[\pi p(t+\tau') - \pi p(t-T)\right]}_{l=1} \cdot \underbrace{\cos\left[\pi p(t+\tau'-T) - \pi p(t-2T)\right]}_{l=2} dt \\
&= \frac{1}{T}\int_0^{T-\tau'} \underbrace{\cos\left[\frac{\pi}{2} - \frac{\pi t}{2T}\right]}_{l=0} \cdot \underbrace{\cos\left[\frac{\pi(t+\tau')}{2T} - 0\right]}_{l=1} \cdot \underbrace{\cos\left[0-0\right]}_{l=2} dt \qquad (6.167) \\
&\quad + \frac{1}{T}\int_{T-\tau'}^{T} \underbrace{\cos\left[\frac{\pi}{2} - \frac{\pi t}{2T}\right]}_{l=0} \cdot \underbrace{\cos\left[\frac{\pi(t+\tau')}{2T} - 0\right]}_{l=1} \cdot \underbrace{\cos\left[0-0\right]}_{l=2} dt \qquad (6.168) \\
&= \frac{1}{T}\int_0^{T-\tau'} \sin\left[\frac{\pi t}{2T}\right] \cdot \cos\left[\frac{\pi t}{2T}\right] \cdot \cos\left[\frac{\pi\tau'}{2T}\right] - \sin\left[\frac{\pi t}{2T}\right] \cdot \sin\left[\frac{\pi\tau'}{2T}\right] dt \qquad (6.169) \\
&= \frac{1}{T}\cos\left[\frac{\pi\tau'}{2T}\right] \cdot \int_0^{T-\tau'} \underbrace{\sin\left[\frac{\pi t}{2T}\right] \cdot \cos\left[\frac{\pi t}{2T}\right]}_{\frac{1}{2}\sin\frac{\pi t}{2T}} dt - \frac{1}{T}\sin\left[\frac{\pi\tau'}{2T}\right] \cdot \int_0^{T-\tau'} \underbrace{\sin^2\left[\frac{\pi t}{2T}\right]}_{\frac{1}{2}(1-\cos(\pi t/T)} dt \qquad (6.170) \\
&= -\frac{1}{2T}\cos\left[\frac{\pi\tau'}{2T}\right] \cdot \left(\frac{T}{\pi}\right) \cdot \left\{\cos\left[\frac{\pi(T-\tau')}{T}\right] - \cos(0)\right\} - \frac{T-\tau'}{2T}\sin\left[\frac{\pi\tau'}{2T}\right] \\
&\quad + \frac{1}{2\pi}\sin\left[\frac{\pi\tau'}{2T}\right] \cdot \left\{\sin\left[\frac{\pi(T-\tau')}{T}\right] - 0\right\} \qquad (6.171) \\
&= -\frac{1}{2\pi}\cos\left[\frac{\pi\tau'}{2T}\right] \cdot \left\{\cos\left[\pi - \frac{\pi\tau')}{T}\right] - 1\right\} - \left(\frac{1}{2} - \frac{\tau'}{2T}\right) \cdot \sin\left[\frac{\pi\tau'}{2T}\right] \qquad (6.172) \\
&\quad + \frac{1}{2\pi}\sin\left[\frac{\pi\tau'}{2T}\right] \cdot \sin\left[\pi - \frac{\pi\tau')}{T}\right] \qquad (6.173) \\
&= -\frac{1}{2\pi}\cos\left[\frac{\pi\tau'}{2T}\right] \cdot \left\{\cos\left[\frac{\pi\tau')}{T}\right] - 1\right\} - \left(\frac{1}{2} - \frac{\tau'}{2T}\right) \cdot \sin\left[\frac{\pi\tau'}{2T}\right] \qquad (6.174) \\
&\quad + \frac{1}{2\pi}\sin\left[\frac{\pi\tau'}{2T}\right] \cdot \sin\left[\frac{\pi\tau')}{T}\right] \qquad (6.175) \\
&= \frac{1}{2\pi}\cos\left[\frac{\pi\tau'}{T} - \frac{\pi\tau'}{2T}\right] + \frac{1}{2\pi}\cos\left[\frac{\pi\tau'}{2T}\right] - \left(\frac{1}{2} - \frac{\tau'}{2T}\right) \cdot \sin\left[\frac{\pi\tau'}{2T}\right] \qquad (6.176) \\
&= \frac{1}{\pi}\cos\left[\frac{\pi\tau'}{2T}\right] - \left(\frac{1}{2} - \frac{\tau'}{2T}\right) \cdot \sin\left[\frac{\pi\tau'}{2T}\right] \qquad (6.177) \\
&= \frac{1}{\pi}\cos\left[\frac{\pi(\tau-T)}{2T}\right] - \left(\frac{1}{2} - \frac{\tau-T}{2T}\right) \cdot \sin\left[\frac{\pi(\tau-T)}{2T}\right] \qquad (6.178) \\
&= (1 - \frac{\tau}{2T}) \cdot \cos\left[\frac{\pi\tau}{2T}\right] + \frac{1}{\pi}\sin\left[\frac{\pi\tau}{2T}\right] \quad . \qquad (6.179)
\end{aligned}
$$

Interestingly enough, the calculations for $m = 0$ and $m = 1$ produce the same function in (6.166) and (6.179). This need not be true for other values of h and in general for other CPM signals with $\nu = 1$, as is evident later in this example. In this case though, the Fourier

Transform expression integrates over the range $\tau \in [0, 2T]$.

$$S_{MSK}(f) = 2\Re\left\{\int_0^{2T} r_{bb}(\tau)e^{-\jmath 2\pi f\tau}d\tau\right\} \tag{6.180}$$

$$= \underbrace{2\int_0^{2T}\cos\left[\frac{\pi\tau}{2T}\right]\cdot\cos\left[2\pi f\tau\right]d\tau}_{A(f)} + \underbrace{2\int_0^{2T}\frac{\tau}{2T}\cdot\cos\left[\frac{\pi\tau}{2T}\right]\cdot\cos\left[2\pi f\tau\right]d\tau}_{B(f)}$$

$$+\underbrace{\frac{2}{\pi}\int_0^{2T}\sin\left[\frac{\pi\tau}{2T}\right]\cdot\cos\left[2\pi f\tau\right]d\tau}_{C(f)} \tag{6.181}$$

Each of the 3 integrals can be evaluated separated before adding them:

$$A(f) = \int_0^{2T}\cos\left[\frac{\pi\tau}{2T}+2\pi f\tau\right]+\cos\left[\frac{\pi\tau}{2T}-2\pi f\tau\right] \tag{6.182}$$

$$= \frac{1}{\frac{\pi}{2T}+2\pi f}\left\{\sin\left[\pi+4\pi fT\right]-\sin(0)\right\}+\frac{1}{\frac{\pi}{2T}-2\pi f}\left\{\sin\left[\pi-4\pi fT\right]-\sin(0)\right\}$$

$$= \frac{-2T}{\pi+4\pi fT}\cdot\sin\left[4\pi fT\right]+\frac{2T}{\pi-4\pi fT}\cdot\sin\left[4\pi fT\right] \tag{6.183}$$

$$= \left[\frac{-2T}{\pi+4\pi fT}+\frac{2T}{\pi-4\pi fT}\right]\cdot\sin\left[4\pi fT\right] \tag{6.184}$$

$$= \frac{8\pi fT}{\pi^2-16\pi^2f^2T^2}\cdot\sin\left[4\pi fT\right] \tag{6.185}$$

$$B(f) = -\underbrace{\int_0^{2T}\frac{\tau}{2T}\cos\left[\frac{\pi\tau}{2T}+2\pi f\tau\right]d\tau}_{B_+(f)} - \underbrace{\int_0^{2T}\frac{\tau}{2T}\cos\left[\frac{\pi\tau}{2T}-2\pi f\tau\right]d\tau}_{B_-(f)} \tag{6.186}$$

where (eventually using $\int x\cos x dx = x\sin x - \cos x$)

$$B_+ = \frac{\frac{-1}{2T}}{\frac{\pi}{2T}+2\pi f}\int_0^{2T}\left[\frac{\pi}{2T}+2\pi f\right]\tau\cdot\cos\left[(\frac{\pi}{2T}+2\pi f)\tau\right]d\tau \tag{6.187}$$

$$= \frac{-1}{\pi+4\pi fT}\cdot\frac{1}{\frac{\pi}{2T}+2\pi f}\cdot\left\{(\pi+4\pi fT)\cdot\sin\left[\pi+4\pi fT\right]-\cos\left[\pi+4\pi fT\right]+1\right\}$$

$$= \frac{-2T}{(\pi+4\pi fT)^2}\cdot\left\{(\pi+4\pi fT)\cdot\sin\left[\pi+4\pi fT\right]-\cos\left[\pi+4\pi fT\right]+1\right\} \quad . \tag{6.188}$$

Similarly

$$B_- = \frac{-2T}{(\pi-4\pi fT)^2}\cdot\left\{(\pi-4\pi fT)\cdot\sin\left[\pi-4\pi fT\right]-\cos\left[\pi-4\pi fT\right]+1\right\} \quad . \tag{6.189}$$

Then,

$$C(f) = -\frac{1}{\pi}\cdot\frac{2T}{\pi+4\pi fT}\cdot\left\{\cos\left[\pi+4\pi fT\right]-1\right\}-\frac{1}{\pi}\cdot\frac{2T}{\pi-4\pi fT}\cdot\left\{\cos\left[\pi-4\pi fT\right]-1\right\} \quad , \tag{6.190}$$

which has terms similar in form to the earlier two quantities. The $\sin\left[\pi-4\pi fT\right]$ terms in $A(f)$ are exactly equal to the $\sin\left[\pi-4\pi fT\right]$ terms in the sum $B_+(f)+B_-(f)$, but opposite

in sign. Thus, (using $1 - cos2x = 2\cos^2 x$ twice)

$$
\begin{aligned}
S_{MSK}(f) &= -2T \cdot \left[\frac{1}{(\pi + 4\pi fT)^2} + \frac{1}{(\pi - 4\pi fT)^2}\right] \cdot 2\cos^2(2\pi fT) \\
&\quad + \frac{2T}{\pi} \cdot \left[\frac{1}{(\pi + 4\pi fT)} + \frac{1}{(\pi - 4\pi fT)}\right] \cdot 2\cos^2(2\pi fT) && (6.191) \\
&= -4T\cos^2(2\pi fT) \cdot \left[\frac{2\pi^2 + 32\pi^2 f^2 T^2}{(\pi^2 - 16\pi^2 f^2 T^2)^2}\right] && (6.192) \\
&\quad + \frac{4T}{\pi}\cos^2(2\pi fT) \cdot \left[\frac{2\pi}{\pi^2 - 16\pi^2 f^2 T^2}\right] && (6.193) \\
&= \frac{16T}{\pi^2} \cdot \frac{\cos^2(2\pi fT)}{(1 - 16f^2T^2)^2} && (6.194)
\end{aligned}
$$

This same closed form expression occurs in many papers and texts for MSK. More generally, it can be shown that when

$$T_b \triangleq \frac{T}{\log_2(M)} \quad , \tag{6.195}$$

then the MSK power spectral density is

$$S_{CPFSK}(f) = \frac{16T_b}{\pi^2} \cdot \frac{\cos^2(2\pi fT_b)}{(1 - 16f^2T_b^2)^2} \quad . \tag{6.196}$$

$T/T_b = 2\bar{b}$ or the bits/second/Hz spectral-efficiency measure of Chapter 2. Thus, presuming $T = 1$ for normalization, then T_b is the Hz/bits-per-second.

Problem 6.13 follows a similar exercise to this example to compute the autocorrelation function of 1REC with $h = 1/4$ and $M = 2$ as

$$r_{bb,1REC} = \begin{cases} \left[1 - \frac{\tau}{2T}\right] \cdot \cos\left[\frac{\pi\tau}{4\pi}\right] + \frac{1}{\pi} \cdot \sin\left[\frac{\pi\tau}{4\pi}\right] & 0 \le \tau \le T \\ \left[\frac{3}{4} + \frac{1}{\pi} - \frac{\tau}{4T}\right] \cdot \cos\left[\frac{\pi\tau}{4\pi}\right] + \left[\frac{\tau}{4T} - \frac{1}{4} + \frac{1}{\pi}\right] \cdot \sin\left[\frac{\pi\tau}{4\pi}\right] & T \le \tau \le 2T \\ 0 & t \ge 2T \end{cases} \quad . \tag{6.197}$$

This autocorrelation is not the same function in both intervals, but is continuous at the $\tau = T$ boundaries. The power spectral density, already for this simple 1REC encouraging the program method developed earlier in this section as opposed to executing closed-form integration. Simon et al have published a report that does find a number of closed form CPM power spectra as products of certain derivable functions.

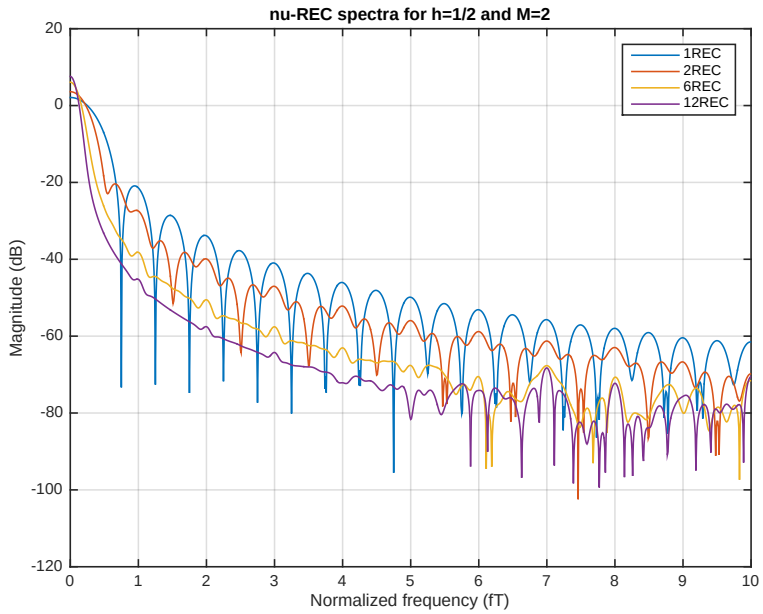


Figure 6.29: νREC Spectra ($h = 1/2$, $M = 2$).

Generally speaking, it is difficult to find closed-form expressions for the spectra, so the Matlab programs in Appendix E are instead provided here as a service to the reader who would like an easy way to find CPM spectra. Hopefully, Example 6.6.1 and Problem 6.13 illustrate this complexity for the simplest CPM form of 1REC. Figure 6.29 uses the REC programs to provide the spectra for CPFSK (including MSK). As the value of ν increases (as large as $\nu = 12$ is shown in the figure), the spectra sidelobes are lower, implying also of course greater complexity in transmitter and ML receiver.

Figure 6.30 shows the νREC spectra for $h = 1/4$. Again larger ν causes lower spectra values. A comparison of Figure 6.30 with Figure 6.29 also shows that lower values of the modulation index reduce the spectra; however, this also reduces the minimum distance between modulated signals so lowers performance.

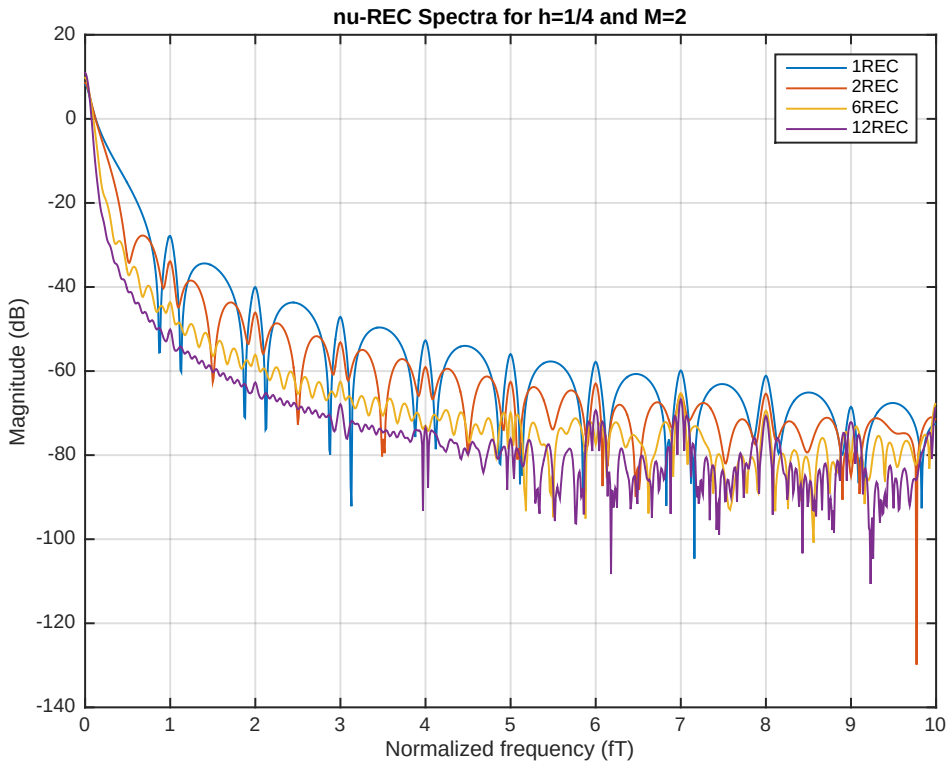


Figure 6.30: νREC Spectra ($h = 1/4$, $M = 2$).

The following table summarizes the cut-off point (99% energy) for several νREC situations:

Table of νREC spectrum efficiency - $1/2b = T_b$						
	$M = 2$		$M = 4$		$M = 8$	
ν / h	1/2	1/4	1/2	1/4	1/2	1/4
1	.6	.45	.64	.39	.77	.45
2	.44	.29	.49	.265	.627	.34
3	.38	.22	.405	.22	.537	.280
6	.28	.15	.295	.155	.397	.203
12	.20	.11	.21	.11	.283	.147

A perfect "sinc" pulse in time would have minimum bandwidths of 0.5, 0.25, and 0.167 for $M = 2, 4, 8$ respectively so some interpretation is necessary. This would correspond to simple BPSK, QPSK, and 8PSK respectively. Also, h alters minimum distance in symbol-by-symbol detection. However, it is clear

that smaller h reduces bandwidth while larger ν also reduces bandwidth (more memory, more sharp reduction in spectra versus frequency). Indeed, it is possible to have "less" bandwidth than the perfect sinc function situation. However, this depends on the decoder to some degree. While both coherent and non-coherent symbol-by-symbol detection have constant distance ($d = 4h$), the assumption of correct previous decisions on phase will not be correct (indeed error propagation is much worse than with decision-feedback equalization). Thus, ML detection will be necessary when (as is often the case) error propagation is important. In that latter case, as will become shortly evident, the minimum distance for the different methods in the table also change so a strict comparison of bandwidth only may lead to false conclusions. Figure 6.31 plots the columns in the Table, from which all the trends are more evident. Increasing M at any value of h does not help this measure of spectral efficiency. An additional useful point may be for $M = 8$ and $h = 1/8$, for which the table entry would be .27.

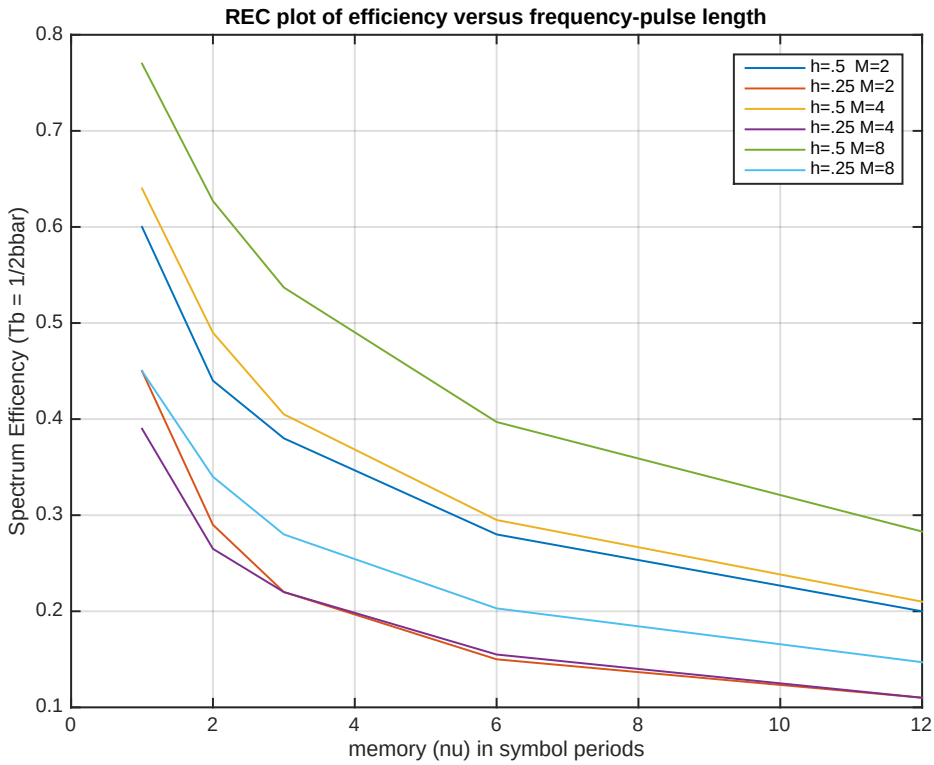


Figure 6.31: Plot of REC spectral efficiency T_b (lower is more efficient) versus memory length in symbol periods.

For the interested reader, the table above used the following short matlab routine that is self explanatory.

```
function edgefreq = bandedge(S,thresh,T,M)
\%    Bandedge computes the edge frequency of an input PSD S.
\%       S is presumed to be 1x10000, with the first 5001 points being the PSD
\%         of interest.
\%       thresh is the perecentage (.99 is 99%) at which energy = thresh*
\%         power
\%       T is symbol rate
\%       M is number of messages/symbol

temp=cumsum(abs(S(1:5001)));
total=sum(abs(S(1:5001)));
edge=0;
```

```
for index=1:5001
   if temp(index) < thresh*total
   edge=index+1;
   end
end
edgefreq=edge*.01*(1/(T*log2(M)));
```

6.6.4.2 RC Spectra

Figure 6.32 illustrates the Raised Cosine CPM spectra for a few values of ν. These CPM functions are more smooth than REC, and thus RC CPM systems have lower sidelobes, which is evident in comparing against the REC spectra. For instance 3RC has spectra as good as 6REC, and 6RC as good as 12REC for both $h = 1/2$ and $h = 1/4$.

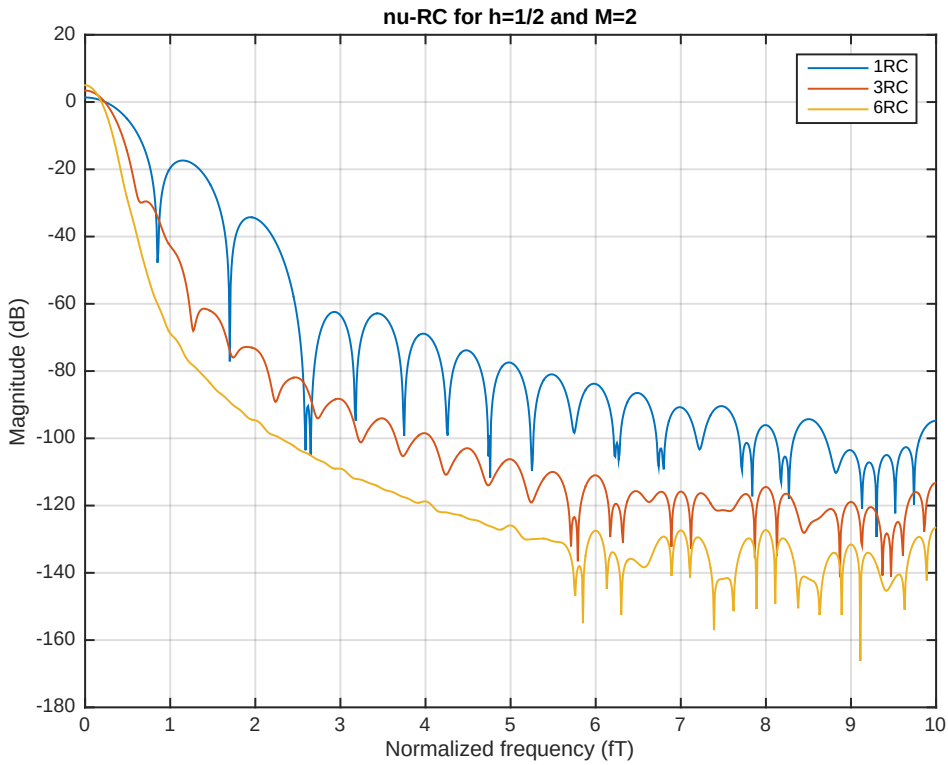


Figure 6.32: νRC Spectra ($h = 1/2$, $M = 2$).

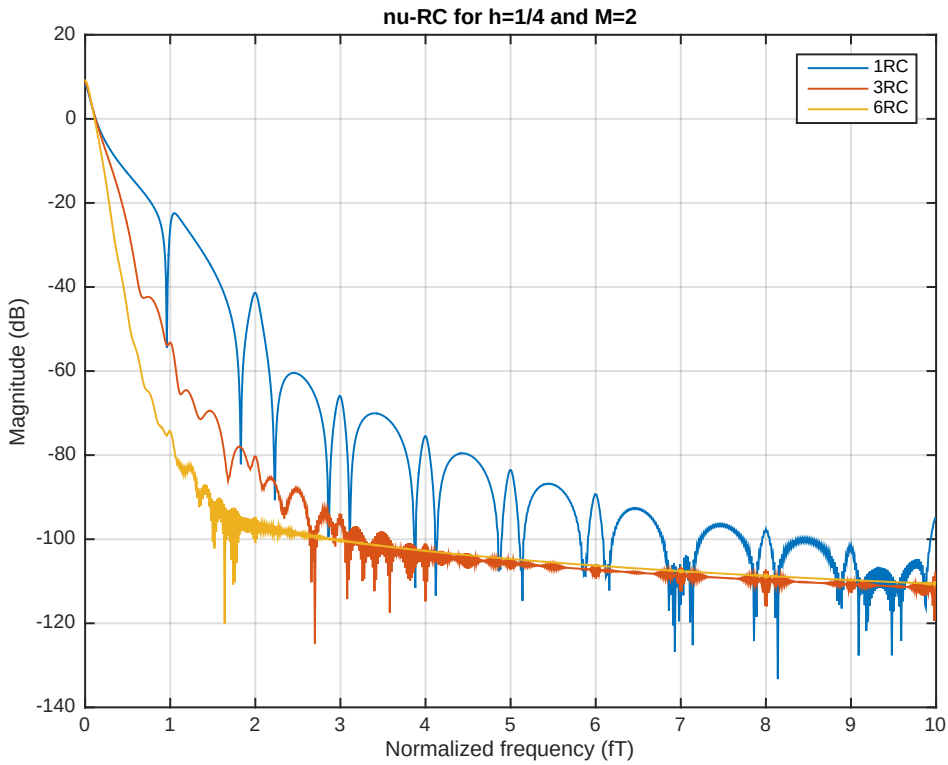


Figure 6.33: νRC Spectra ($h = 1/4$, $M = 2$).

Table of νRC spectrum efficiency - $1/2\bar{b} = T_b$						
	$M = 2$		$M = 4$		$M = 8$	
ν / h	1/2	1/4	1/2	1/4	1/2	1/4
1	1.11	.62	.97	.62	1.25	.673
3	.45	.31	.445	.26	.587	.31
6	.33	.2	.34	.185	.46	.237
9	.27	.16	.285	.15	.387	.197
12	.23	.13	.25	.13	.34	.173

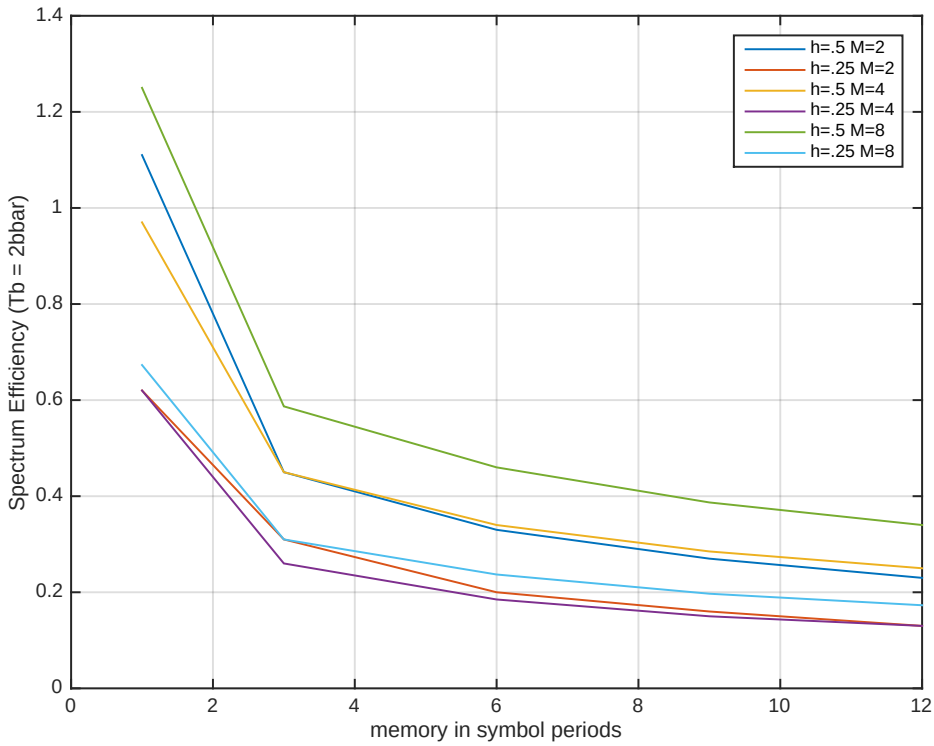


Figure 6.34: Plot of RC spectral efficiency T_b (lower is more efficient) versus memory length in symbol periods.

6.6.4.3 SRC Spectra

The SRC spectra are yet more smooth and have some of the best side-lobe properties. However, the ν values (relative to the n values) are large, and thus can complicate transmitters and receivers. The roll-off is 50% for the nSRC plots in Figure 6.35. The 3RC and 6RC spectra exhibit very high reduction in sidelobes, much better than REC or RC for $\nu = 3$ and $\nu = 6$.

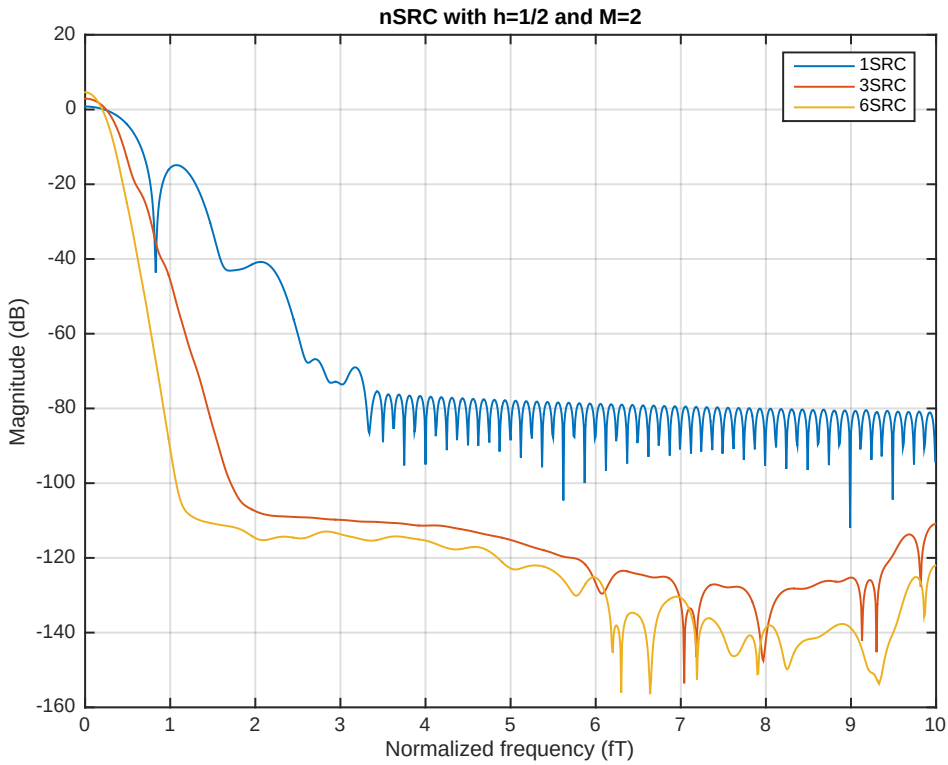


Figure 6.35: nSRC Spectra ($h = 1/2$, $M = 2$) with 50% excess bandwidth.

The excess-bandwidth parameter α is chosen as 100% for Figure 6.36, which appears to have higher side-lobe energy but of course this is with 100% roll-off. Thus, there is a trade-off between the excess bandwidth used for smoothness and the CPM signal's sidelobe bandwidth as evident in comparing this spectra with those of Figure 6.35. The reader may want to experiment themselves, using the program in Appendix E.

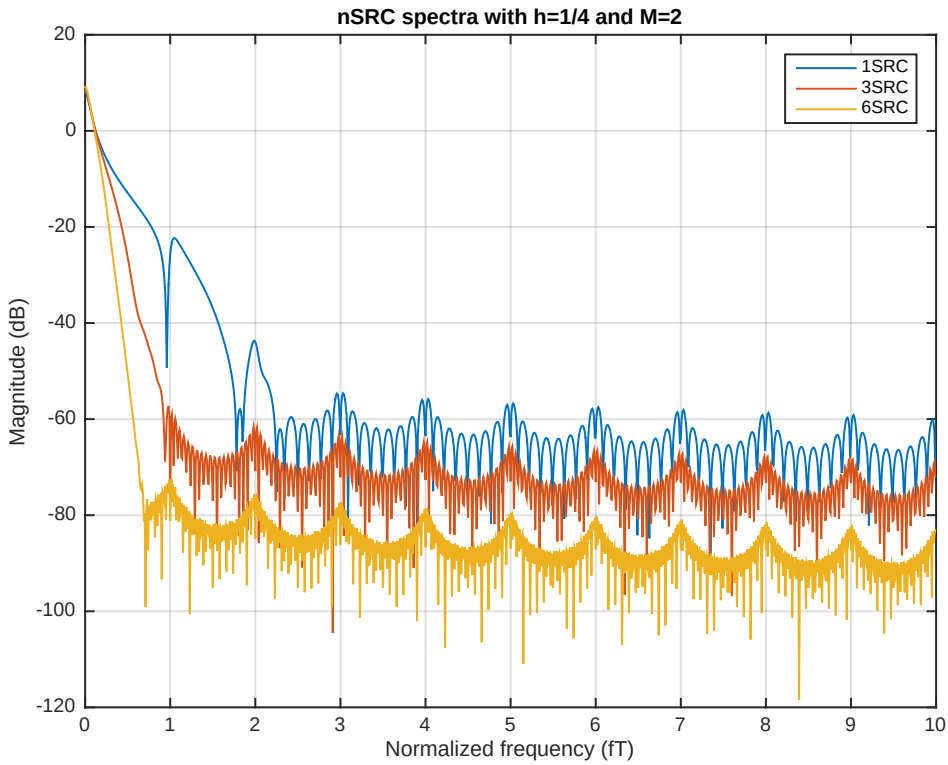


Figure 6.36: nSRC Spectra ($h = 1/4$, $M = 2$).

The following table summarizes the SRC spectral efficiency (with *alpha* at 100%) as with earlier CPM pulse shapes' earlier tables. Comparison of this table with the RC table earlier shows that spectral efficiency of RC and SRC are nearly the same when the n of SRC is compared to the RC at the value of ν equal to n. Any small deviations are attributed to truncation effects in the SRC spectrum calculation. It suggests a conclusion that the same level of smoothness in CPM spectrum is achieved whether the frequency pulse response is modeled in time or frequency, effectively a duality. The curious student might want to pursue this – the author is not aware that researchers in the field have noted this.

Table of nSRC spectrum efficiency - $1/2b = T_b$						
	$M = 2$		$M = 4$		$M = 8$	
n / h	1/2	1/4	1/2	1/4	1/2	1/4
1	1.11	.61	.97	.62	1.25	.673
3	.47	.31	.445	.26	.587	.31
6	.35	.2	.34	.185	.46	.237
9	.27	.16	.285	.15	.387	.197
12	.23	.13	.25	.13	.34	.173

6.6.4.4 GMSK Spectra

The GMSK spectra are similar or even better than the SRC spectra (although there is an implied higher ν), as in Figures 6.37 and 6.38. The effect of h appears minimal here (and indeed is not really MSK because that implies $h = 1/2$). The reader could experiment with different B values for $h = 1/4$ using the programs in Appendix E.

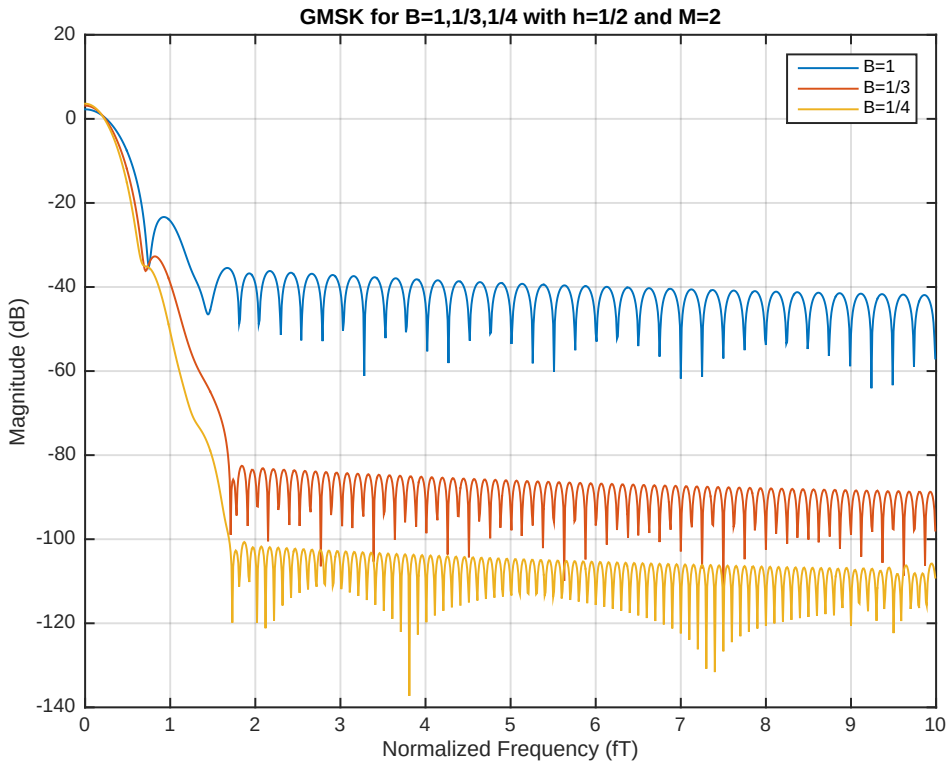


Figure 6.37: GMSK Spectra ($h = 1/2$, $M = 2$).

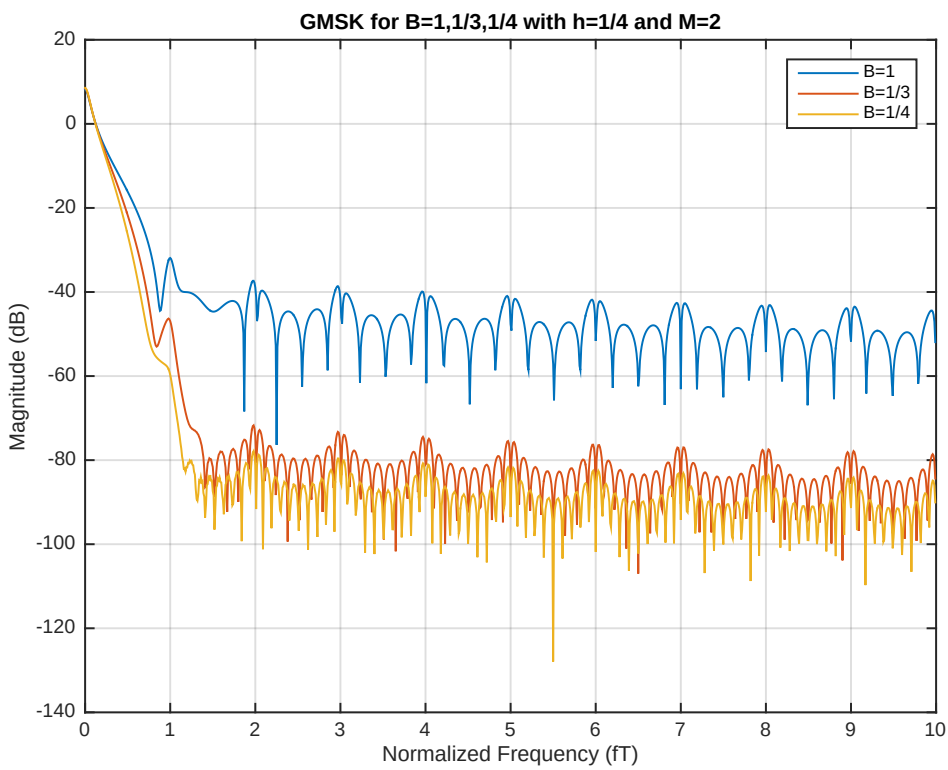


Figure 6.38: GMSK Spectra ($h = 1/4$, $M = 2$).

The GMSK efficiency is in the following table:

Table of GMSK spectrum efficiency - $1/2b = T_b$						
	$M = 2$		$M = 4$		$M = 8$	
$1/B$ / h	1/2	1/4	1/2	1/4	1/2	1/4
1	.59	.44	.59	.35	.71	.41
2	.52	.38	.51	.305	.647	.36
3	.48	.33	.465	.275	.60	.327
4	.44	.29	.43	.25	.56	.30

6.6.4.5 TFM Spectrum

Figure 6.39 shows the Tamed Frequency Modulation spectrum. It too is comparable to GMSK in terms of spectrum use.

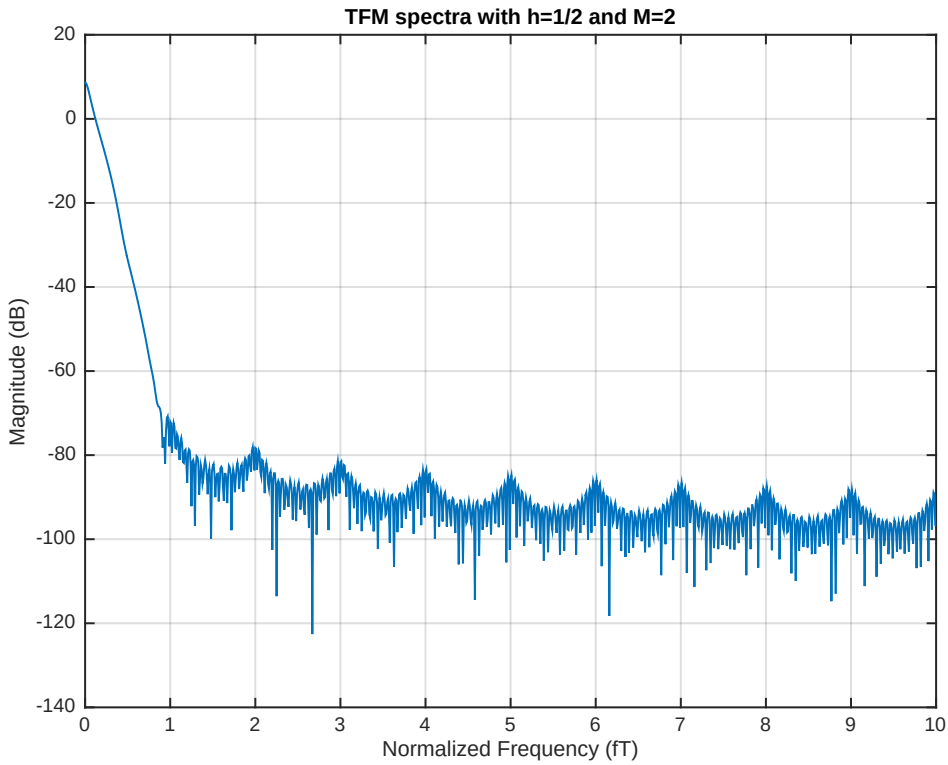


Figure 6.39: TFM Spectra ($h = 1/2$, $M = 2$).

Table of TFM spectrum efficiency - $1/2b = T_b$						
	$M = 2$		$M = 4$		$M = 8$	
- / h	1/2	1/4	1/2	1/4	1/2	1/4
-	.40	.26	.415	.23	.55	.287

6.6.4.6 Some other spectrum observations

Figure 6.40 plots 1REC for $M = 2$ and $M = 4$, which illustrates that $M = 4$ with twice the data rate does also occupy about twice the bandwidth. Some authors on CPM use a notation of T_b to represent the "bit" period, so that in this case $T_b = 1/2T$. Figure 6.41 re-plots this same figure to show the nominal bits/Hz normalized comparison. In terms of bits/Hz, the two are thus comparable.

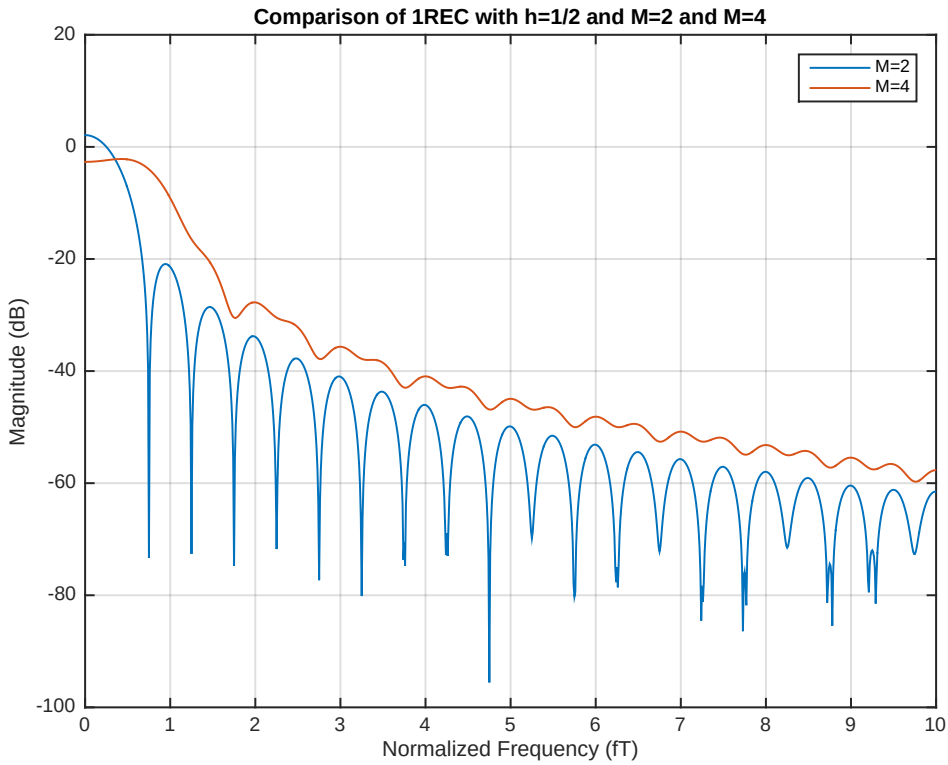


Figure 6.40: nREC Spectra ($h = 1/2$, $M = 2$ and $M = 4$).

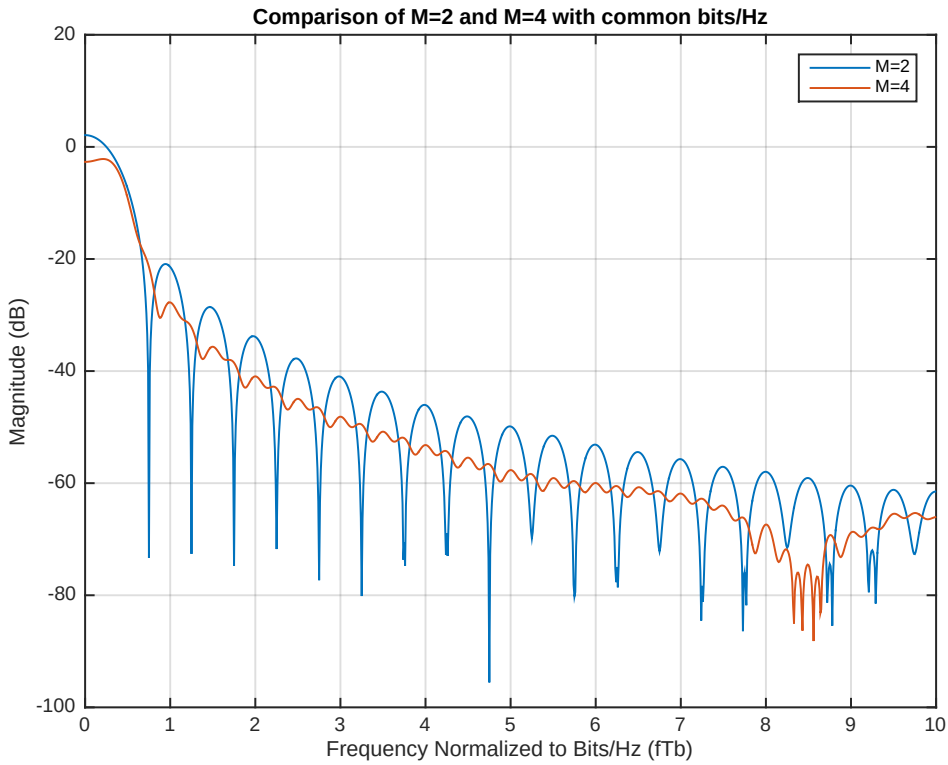


Figure 6.41: nREC Spectra ($h = 1/2$, $M = 2$ and $M = 4$) with normalized bits/Hz.

Figure 6.42 illustrates spectra for REC, RC, and SRC with $\nu = 2$, showing that the spectrum sidelobes of the SRC are considerably lower, but RC improves also on REC. Figure 6.43 shows this same observation holds true for $M = 4$.

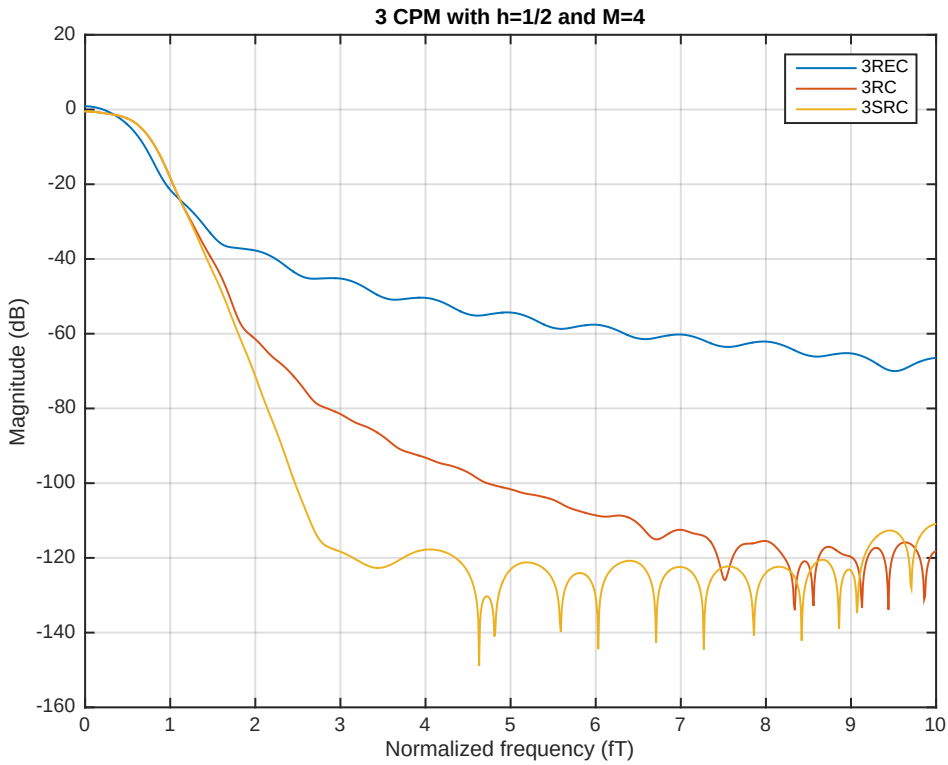


Figure 6.42: Comparison of CPM Spectra ($h = 1/2$, $M = 4$, and $\nu = 3$).

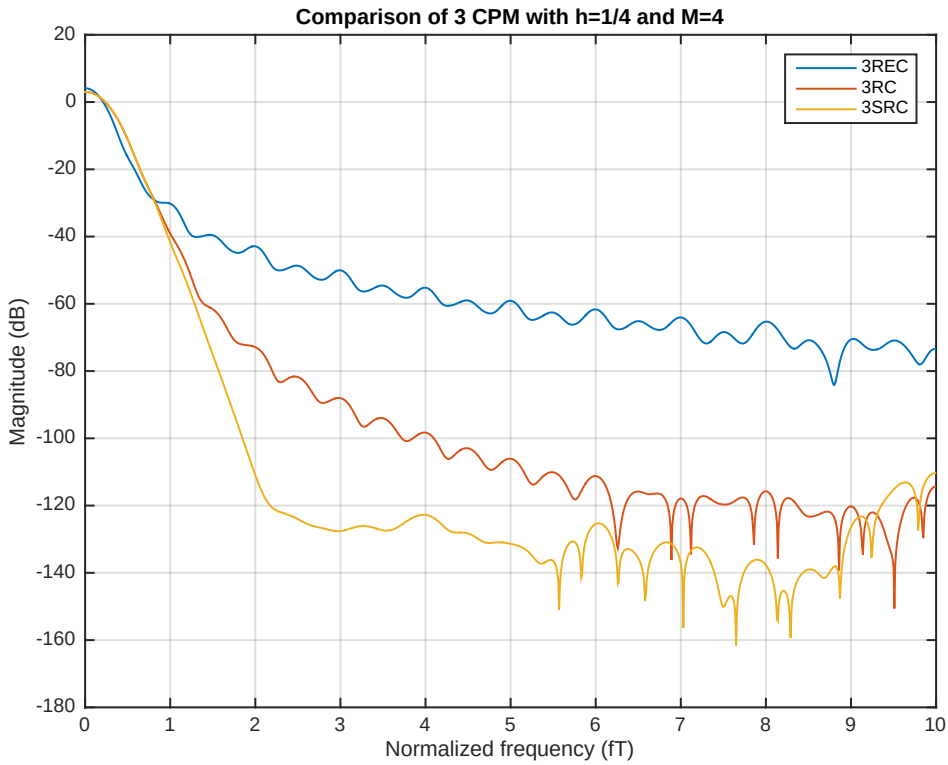


Figure 6.43: Comparison of CPM Spectra ($h = 1/4$, $M = 4$, and $\nu = 3$).

Table 6.1 now tabulates the 99% points (for $M = 2$ and $M = 4$) with normalized bits/Hz (using same T_b).

insert table here.

Table 6.1: Comparison of CPM types for CPM-DFE type receivers.

6.6.5 Coherent Maximum Likelihood Detection

A CPM transmitter implementation that might be associated with coherent detection is shown in Figure 6.44.

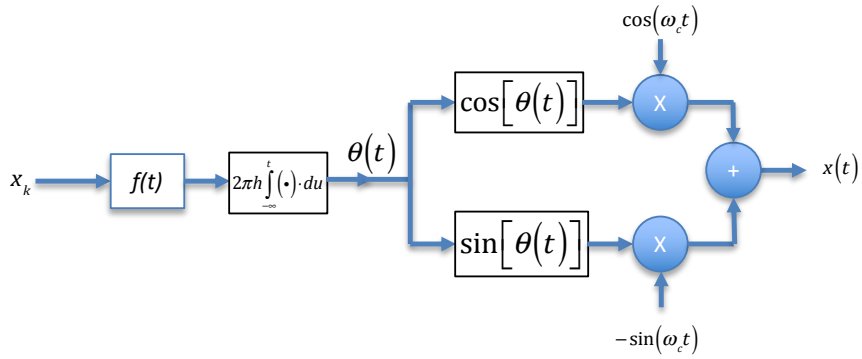


Figure 6.44: CPM Transmitter with traditional carrier modulation.

Coherent detection uses Chapter 2's scaling phase splitter to covert the channel-output signal to baseband. Basis-detectors have been heavily used in the first 5 Chapters of this text; however, basis detectors are not attractive for CPM. A basis detector requires a small number of filters matched to the orthonormal set of basis functions, and such a decomposition of signals into orthonormal filters is complex for CPM. While theoretically, a Gram Schmidt procedure could be executed on the set of all possible signals, there does not appear to be an advantage in general for CPM to do so.

For CPM instead, ML detection essentially uses a signal detector, which uses filters matched directly to the M^ν signal waveforms (see Chapter 1). The number M^ν needs some explanation: There are only a maximum of $|S| = q \cdot M^{\nu-1}$ possible states with q distant-past settings yielding signal waveforms at any time with CPM (given the constraint on the modulation index to be a rational fraction with even numerator and co-prime denominator integer q). There are also a factor of M possible extensions (for each state) at current time interval $t \in [(k-1)T, kT]$. This might suggest $M^\nu \cdot q$ filters. The factor of q really does not change the set of filters and so can be eliminated because the entire set of filters may be re-indexed from symbol time to symbol time, but that entire set is not affected by q, just reindexed by the M^ν values of x_k, x_{k-1}, ... $x_{k-\nu}$ that match with the various q distant-past possibilities. Figure 6.45 thus shows an ML detector implementation with M^ν sub-blocks each containing a complex-baseband signal-based matched filter.

For the AWGN channel at time t, an ML detector selects the input waveform $\hat{x}(t)$ (itself a function of $\{\hat{x}_l\}_{l=-\infty\ldots k}$) that is at minimum squared distance from the actual (noisy) channel output. The cost function is defined as

$$C_k^* = \underbrace{\min}_{\{x_l\}_{l=-\infty\ldots k}} \int_{-\infty}^{kT} |y(t) - \hat{x}(t)|^2 \, dt = \underbrace{\min}_{s_k \in S} \int_{-\infty}^{kT} |y(t) - \hat{x}(t)|^2 \, dt \quad , \tag{6.198}$$

where the second state-dependent expression recognizes that the finite number $|S|$ of all states at symbol time k represent all the possible countably infinite set of histories of the input symbol sequence. Since there are $|S|$ possible states at any symbol time, each with its own history and possible trellis paths into it, a state-dependent cost could be written also as $C_k(s)$ (and then the best overall cost C_k selected at that time from those $|S|$ possibilities), which admits the recursion

$$C_k(s) = \underbrace{\min}_{x_k, s' \in S} \left[C_{k-1}(s') + \Delta_k(s, s') \right] \quad , \tag{6.199}$$

where

$$\Delta_k(s, s') = \int_{(k-1)T}^{kT} |y(t) - \hat{x}(t)|^2 \, dt \quad . \tag{6.200}$$

As far as the minimization goes, the squared integrate terms could be rewritten in terms of inner products of $y(t)$ with the corresponding input waveforms, since the latter have all the same energy and the integral of $y^2(t)$ contributes nothing to the optimization. Accounting for a negative sign, the

minimization becomes selection of the largest cost function (now basically the sampled matched filter outputs) for each state, but the basic recursion remains. This is often called the **Viterbi Algorithm**, see Chapter 9. The decision is then the sequence of input symbols that correspond up to time k to the state that has the best cost function value at time k.

The single (complex baseband) filter for each state s could be denoted $x_{s,bb}(-t)$, which has real and imaginary parts given by

$$x_{s,I}(t) = \Re\{x_{s,bb}(t)\} = \cos\left[\sum_{x_l \in \boldsymbol{x}_s} x_l \cdot p(t - lT)\right] \tag{6.201}$$

$$x_{s,Q}(t) = \Im\{x_{s,bb}(t)\} = \sin\left[\sum_{x_l \in \boldsymbol{x}_s} x_l \cdot p(t - lT)\right] . \tag{6.202}$$

These filters are linear time-invariant and can each be sampled at the symbol rate $1/T$ to generate the successive outputs for each successive symbol instant. The set of all matched signal filters is a linear transformation away (think Gram Schmidt) from a set of orthogonal filters for which Chapter 3 established the successive sample-time values (in this case for the entire set of M^ν filters is a sufficient statistic to preserve optimum detection by acting only on the samples).

The scaled-phase-splitter/carrier-demodulator output in Figure 6.45 $y_{bb}(t)$ is filtered by a bank of such filters in parallel. Their outputs are sampled at the symbol rate. For each state at time kT there is a smallest-squared-distance path (which corresponds to maximum signal-detector matched-filter output sample for that state) that determines the surviving path into each state. At any point in time, the path with the best metric value (largest matched-filter-output sample) across all the states can be traced in reverse to make a maximum-likelihood estimate (which is the input symbol sequence xZ_k along that best-surviving path) subject to the limitation of the detector becomes optimum only as $k \to \infty$. However, as transients abate, the distant past x_k values on the best survivor path will closely approximate the ML detector decisions at infinite length. For CPM (without outer coding, see Chapter 10) will not be catastrophic (meaning two different infinite-length sequences of inputs with finite squared distance between them will only differ in a finite number of positions) so any chance of an infinite number of errors goes to zero with CPM (this is not true of all trellis diagrams - see Chapters 9 and 10).

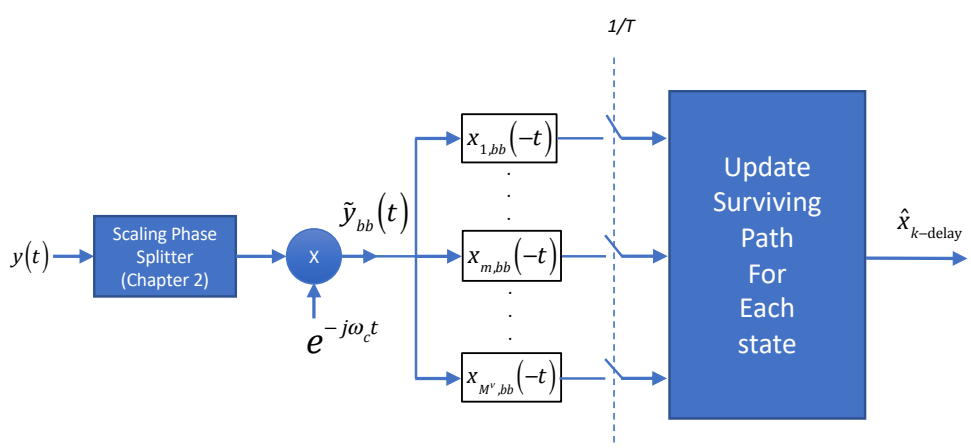


Figure 6.45: Calculation of minimum distance between any two different CPM signals.

6.6.6 Performance of Coherent Detection

6.6.6.1 Symbol-by-Symbol Case

Coherent symbol-by-symbol detection collapses the structure of Figure 6.45 to one state. Unlike non-coherent detection, the carrier is presumed known and the baseband signal is generated as on the

left in Figure 6.45. After a single (baseband complex) lowpass filter, the baseband signal is sampled at the symbol rate $1/T$ and will correspond to a point on the circle plus noise, call it $\tilde{y}_{bb}(kT)$. All previous symbol decisions are known and presumed correct, and the previous noise-free channel output sample then has a known phase that can be constructed from the previous decisions $\hat{\boldsymbol{x}}_{k-1:k-\nu}$ and the corresponding (one of q possible) value of phase prior to sampling time $(k-\nu)T$ essentially equivalent to a single path already assumed correct in the MLSD decoder described above (so no other potential survivors are maintained). This single corresponding state index s allows the contributions of previous inputs to phase to be computed at time kT and removed from $\tilde{y}_{bb}(kT)$'s (noisy) phase and then a decision directly made on the residual phase for $\hat{x}_k$.

The performance of such a system (presuming correct decisions in the past) is simply the same as BPSK for $M = 2$, QPSK for $M = 4$, 8PSK for $M = 8$ and so on., so from Chapter 1:

$$P_e \leq (M-1) \cdot Q\left[\frac{\sqrt{\mathcal{E}_{\boldsymbol{x}}} \cdot \sin(\pi/M)}{\sigma}\right] \quad . \tag{6.203}$$

The minimum distance for BPSK is thus $d_{\min} = 2\sqrt{P_x T}$. This performance is equivalent to a matched-filter-bound for uncoded transmission. It can be exceeded only because CPM methods effectively are codes that can increase the minimum distance. However, Equation (6.203) will often lower bound loosely performance and could actually lower bound ML detection so may be inaccurate because previous decisions are not correct sufficiently often in many CPM applications.

6.6.6.2 Maximum Likelihood Case

The performance of an ML detector is dominated by the minimum distance between the closest two symbols (which are symbol sequences in this case), as well as the number of nearest neighbors to a second-order degree and $P_e \approx N_e \cdot Q(d_{\min}/2\sigma)$ according to the NNUB of Chapter 1. Computation of minimum distance (and nearest neighbors) in a trellis can be complex and Chapter 9 addresses it. Chapter 9 provides a minimum-distance computing Matlab program is provided that works for any trellis description. However, for CPM, the minimum distance calculation can be simplified as follows: First, Figure 6.46 simplifies using basic geometry the distance between any two different CPM signals viewed in baseband as complex vectors on the circle with angles θ_1 and θ_2. Using basic geometry and the half-angle formula in Figure 6.46, the minimum distance squared between any two equal-amplitude phasors is

$$d^2_{\min} = \min_{\boldsymbol{x}_1 \neq \boldsymbol{x}_2} \frac{2\bar{\mathcal{E}}_{\boldsymbol{x}}}{T} \int_{-\infty}^{t} \left[1 - \cos(\theta(\boldsymbol{x}_1) - \theta(\boldsymbol{x}_2)\right] \, dt \quad . \tag{6.204}$$

The integral limits can be simplified to any finite length over which the two sequences (and thus angles differ).

The calculation of minimum distance for CPM is specifically performed at the public website

`https://people.eecs.ku.edu/~perrins/doku/doku.php?id=cpmdistance`

by Professor Erik Perrins of Kansas University, which has there a useful higher-level interface shown in Figure 6.47.

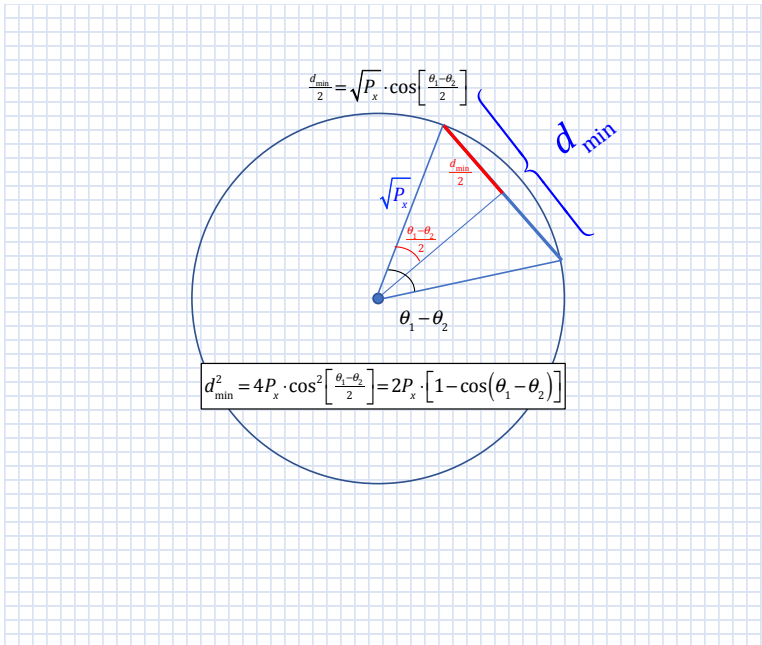
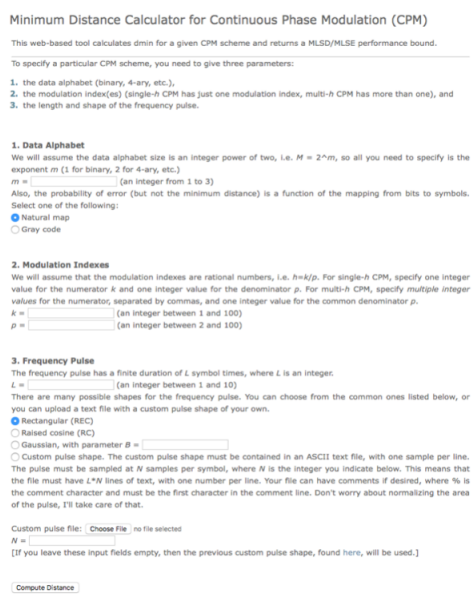


Figure 6.46: Calculation of minimum distance between any two different CPM signals.



Minimum Distance Calculator for Continuous Phase Modulation (CPM)

This web-based tool calculates dmin for a given CPM scheme and returns a MLSD/MLSE performance bound.

To specify a particular CPM scheme, you need to give three parameters:

1. the data alphabet (binary, 4-ary, etc.),
2. the modulation index(es) (single-*h* CPM has just one modulation index, multi-*h* CPM has more than one), and
3. the length and shape of the frequency pulse.

1. Data Alphabet

We will assume the data alphabet size is an integer power of two, i.e. *M* = 2^*m*, so all you need to specify is the exponent *m* (1 for binary, 2 for 4-ary, etc.)

m = (an integer from 1 to 3)

Also, the probability of error (but not the minimum distance) is a function of the mapping from bits to symbols. Select one of the following:

- (•) Natural map
- ( ) Gray code

2. Modulation Indexes

We will assume that the modulation indexes are rational numbers, i.e. *h*=*k*/*p*. For single-*h* CPM, specify one integer value for the numerator *k* and one integer value for the denominator *p*. For multi-*h* CPM, specify *multiple integer values* for the numerator, separated by commas, and one integer value for the common denominator *p*.

k = (an integer between 1 and 100)

p = (an integer between 2 and 100)

3. Frequency Pulse

The frequency pulse has a finite duration of *L* symbol times, where *L* is an integer.

L = (an integer between 1 and 10)

There are many possible shapes for the frequency pulse. You can choose from the common ones listed below, or you can upload a text file with a custom pulse shape of your own.

- (•) Rectangular (REC)
- ( ) Raised cosine (RC)
- ( ) Gaussian, with parameter *B* =
- ( ) Custom pulse shape. The custom pulse shape must be contained in an ASCII text file, with one sample per line. The pulse must be sampled at *N* samples per symbol, where *N* is the integer you indicate below. This means that the file must have *L**N lines of text, with one number per line. Your file can have comments if desired, where % is the comment character and must be the first character in the comment line. Don't worry about normalizing the area of the pulse, I'll take care of that.

Custom pulse file: Choose File no file selected

N =

[If you leave these input fields empty, then the previous custom pulse shape, found here, will be used.]

Compute Distance

Figure 6.47: Perrins' minimum-distance tool.

An example use of this tool produces the result in Figure 6.48 for MSK.

```
m = 1
SymbMap = 1
k = 1
p = 2
L = 1
PulseID = 1
PRECISION = high

The minimum distance is 2.000000.

The first two terms in the union bound approximation of the probability of bit error are:

1.000000*Q(sqrt(2.000000*Eb/No)) + 1.000000*Q(sqrt(2.000000*Eb/No))
```

Figure 6.48: Use Example of Perrins' minimum-distance tool.

This MSK example use products a minimum distance of 2 (so squared minimum distance 4) and thus corresponds to $P_x = 1$. Clearly ML-detected MSK and symbol-by-symbol-dected BPSK have the same performance when used on an AWGN channel where the noise dominates any adjacent-channel interference effects. Clearly, if the adjacent-channel interference is larger than the noise, then MSK (with its more complicated detector) will perform better since its spectrum decays more rapidly than does that of BPSK.

The table below illustrates a number of runs of Perrin's calculator, and converted to dB gain relative to BPSK and MSK's nominal minimum distance of 2 (for $P_x = 1$). Of interest and as expected, larger h nominally increases distance. The number of messages per symbol tends to provide best distances at intermediate values of ν for $M = 4$ and $M = 8$. Both of these increases (h or M) also tend to increase the 99% bandwidth point of the corresponding modulation type.

6.7 Frame Synchronization in Data Transmission

While carrier and symbol clock may have been well established in a data transmission system, the boundary of long symbols (say in the use of the DMT systems of Chapter 4 or the GDFE systems of Chapter 5) or packets may not be known to a receiver. Such synchronization requires searching for some known pattern or known characteristic of the transmitted waveform to derive a phase error (which is typically measured in the number of sample periods or dimensions of offset from best/desired exact time). Such synchronization, once established, is only lost if some kind of catastrophic failure of other mechanisms has occured (or dramatic sudden change in the channel) because it essentially involves counting the number of samples from the recovered timing clock.

6.7.1 Autocorrelation Methods

Autocorrelation methods form the discrete channel output autocorrelation

$$\boldsymbol{R}_{\boldsymbol{yy}}(l) = E\left[y_k \cdot y_{k-l}^*\right] = h_l * h_{-l}^* * \boldsymbol{R}_{\boldsymbol{xx}}(l) + \boldsymbol{R}_{\boldsymbol{nn}}(l) \tag{6.205}$$

from channel output samples by time averaging over some interval of M samples

$$\hat{R}_{\boldsymbol{yy}}(l) = \frac{1}{M} \cdot \sum_{m=1}^{M} y_m \cdot y_{m-l}^* \quad . \tag{6.206}$$

Formation of such a sum implies that the receiver knows the exact sampling times $1, ..., M$. If those times are not coincident with the corresponding (including channel delay) positions of the packet in the transmitter, then the autocorrelation function will look shifted. For instance, if a peak was expected at position l and occurs instead at position l', then the difference is an indication of a packet-timing error. The phase error used to drive a PLL is that difference. Typically, the VCO in this case supplies only discrete integer-number-of-sample steps. If noise is relatively small and h_l is known, then one such error sample may be sufficient to find the packet boundary. If not, many repeated estimates of $\hat{R}_{\boldsymbol{yy}}$ at different phase adjustments may be tried.

There are 3 fundamental issues that affect the performance of such a scheme:

1. the knowledge of, or choice of, the input sequence x_k,
2. the channel response's (h_k's) effect upon the autocorrelation of the input, and
3. the noise autocorrelation.

6.7.1.1 Synchronization Patterns

A synchronization pattern or "synch sequence" is a known transmitted signal, typically with properties that create a large peak in the autocorrelation function at the channel output at a known lag l. With small (or eliminated) ISI, such a sequence corresponds to a "white" symbol sequence ($\bar{R}_{xx} = \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot I$ or $\bar{r}_{xx,k} = \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \delta_k$). Simple random data selection may often not be sufficient to guarantee a high autocorrelation peak for a short averaging period M. Thus, special sequences are often selected that have a such a short-term peaked time-averaged autocorrelation. A variety of sequences exist.

The most common types are between 1 and 2 cycles of a known periodic pattern. **Pseudorandom binary sequences (PRBS's)** of degree p have a period of $2^p - 1$ samples. These PRBS are specified by their basic property that the set of all the sequence's single-period cyclic shifts will contain each length-p binary pattern (except all 0's) once and only once. These sequences are sometimes used in the spreading patterns from CDMA methods in Chapter 5 also, or in random interleaving in Chapter 11 (where a table of primitive polynomials used to generate PRBS's appears). Chapter 12 shows how to implement a specific PRBS with binary logic and flip flops from the primitive poynomial. Here in Chapter 6, the important property is that the autocorrelation (with binary antipodal modulation and $M = 2^p - 1$ in Equation (6.206)) of such sequences is

$$\hat{R}_{\boldsymbol{xx}}(l) = \begin{cases} 1 & l = 0 \\ -\frac{1}{2^p-1} & l = 1, ..., 2^p - 1 \end{cases} \quad . \tag{6.207}$$

For reasonably long p, the peak is easily recognizable if the channel output has little or no distortion. In fact, the entire recognition of the lag can be positioned as a basic Chapter 1 detection problem, where each of the possible input shifts of the same known training sequence is one of the signal-set choices for a detector. Because all these patterns have the same energy and they're almost orthogonal, a simple largest matched-filter ouptut (the matched-filter outputs implement the computation of the autocorrelation at different time lags, and is thus optimum). The probability of error is well understood and addressed in Chapter 1.

Another synchronization pattern that can have appeal is the so-called **chirp sequence**

$$x_k = e^{\jmath 2\pi k^2/M} \tag{6.208}$$

which has period M. It also has a single peak at time zero of size 1 and zero autocorrelation at other time lags. The chirp sequence is harder to generate and requires a complex QAM baseband system, but in some sense is a perfect synch pattern.

A third alternative are the so-called "Barker codes" that are very short and not repeated and designed to still have peakiness in the face of unknown preceding and succeeding data surrounding the pattern. Such patterns may be periodically inserted in a transmission stream so that if the receiver for any reason lost packet synchronization (or a new receiver joined after a first had already acquired the signal – if multiple receivers scan or use the same channels), then the calculation of autocorrelation would immediately commence until the peak was found. An example of a good short synchronization pattern is the 7-symbol Barker code, which is forinstance used with 2B1Q transmission in some symmetric DSL (HDSL or SDSL) systems that employ PAM modulation. This code transmits at maximum amplitude the binary pattern

$$+++--+- \tag{6.209}$$

(or its time reverse on loop 2 of HDSL systems) so $M = 7$ and only binary PAM is used when this sequence is inserted (which means the $d_{\min}$ increases to 7 dB for a detection problem, simply because 2 levels replace 4 already). This pattern has a maximum $M \cdot \hat{R}_{\boldsymbol{xx}}(l)$ value of 7 units when aligned (the time offset to redefine $l = 0$). The 13 place autocorrelation function is

$$1010107010101 \quad . \tag{6.210}$$

Because there are adjacent transmissions, the minimum distance of 7 units (may be reduced to 6 or 5, as long as the searching receiver has some idea when to look) multiplies a distance that is already 7 dB better than 4 level transmission. Thus, this pattern can be recovered quickly, even with high noise.

Creative engineers can design patterns that will be easy to find in any given application. One set of sequences that can be used for timing also allows separation of different user signals when they share a common channel, and is known as a Walsh Code. These are used in 3G wireless CDMA systems and discussed more in the Chapter 5 on these Walsh codes, but they can be used for frame recovery also.

6.7.1.2 The channel response

Severe ISI can cause a loss of the desired autocorrelation properties at the channel output. The usual situation is that ISI is not so severe because other systems are already in use to reduce it by the time that synchronization occurs. An exception would be when a receiver is looking for a training sequence to adapt its equalizers (see Chapter 7) initially. In such a case, the cross-correlation of the channel output with the various delayed versions of the known training pattern can produce an estimate of h_l,

$$\frac{1}{M}\sum_{m=1}^{M} y_k \cdot x_{k-l}^* \approx h_l \tag{6.211}$$

so that the channel can be identified. Then the resultant channel can be used in a maximum likelihood search over the delayed patterns in the synchronization symbol to estimate best initial packet boundary.

6.7.1.3 The noise

Large or strongly correlated noise can cause the estimated autocorreleation to be poor. The entire channel output y_k should be whitened (as in Chapter 5) with a linear predictor. In this case, again the cross-correlation estimate of Equation (6.211) can be used to estimate the equivalent channel after oveall whitening and the procedure of the previous section for ISI followed.

6.8 Pointers and Add/Delete Methods

This final section of Chapter 6 addresses the question, "well fine, but how did the source message clock get synchronized to the symbol clock?" The very question itself recognizes and presumes that a bit stream (or message stream) from some source probably does not also have a clock associated with it (some do, some don't). The receiver of the message would also in some cases like to have that source clock and in addition to the receiver's estimates of the transmitter's symbol and carrier clocks. Typically, the source cannot be relied upon to supply a clock, and even if it does, the stability of that clock may not match the requiremens for the heavily modulated and coded transmission system of Chapters 1-5.

Generally, the source clock can be monitored and essentially passed through the monitoring of interface buffers that collect (or distribute) messages. Figure 6.49 illustrates the situation where certain tolerances on the input clock of the source may be known but the symbol clock is not synchronized and only knows the approximate bit rate it must transmit. For such situations, the transmit symbol clock must always be the higher clock, sufficiently so that even if at its lowest end of the range of speed, it still exceeds the transmit clock. To match the two clocks, regularly inserted "dummy" messages or bits are sometimes sent (or not sent) to reduce the queue depth awaiting transfer. These methods are known as "rob/stuff" or "add/delete" timing and sychronization methods. The receiver will need an indication (embedded as a control sequence in the transmission data) to know if/when to stuff/rob.

The principle is simple. The buffer depth is monitored by subtracting the addresses of the next-to-transmit and last-in bits. If this depth is reducing or steady while above a safety threshold, then dummy bits are used and transmitted. If the depth starts to increase, then dummy bits are not transmitted until the queue depth exceeds a threshold.

The same process occurs at the receiver interface to the "sink" of the messages. The sink must be able to receive at least the data rate of the link and typically robs or stuffs are used to accelerate or slow the synch clock. A buffer is used in the same way. The sink's lowest clock frequency must be within the range of the highest output clock of the receiver in the transmission system.

The size of the queue in messages (or bits) needs to be at least as large as the maximum expected difference in source and transmit bit-rate (or message-rate) speeds times the the time period between potential dummy deletions, which are known as "robs." ('Stuffs are the usual dummies transmitted). The maximum speed of the transmission system with all robs must exceed the maximum source supply of messages or bits.

Add/drop or Add/delete methods are essentially the same and typically used in what are known as asynchronous networks (like ATM for Asynchronous Transfer Mode, which is really not asynchronous at all at the physical layer but does have gaps between packets that carry dummy data that is ignored and not passed through the entire system. ATM networks typically do pass an 8 kHz network clock reference throughout the entire system. This is done with a "pointer." The pointer typically is a position in a high-speed stream of ATM packets that says "the difference in time between this pointer and the last one you received is exactly 125 μs of the desired network clock. Various points along the entire path can then use the eventual received clock time at which that pointer was received to synchronize to the network clock. If the pointer arrives sooner than expected, the locally generated network reference is too slow and the phase error of the PLL is negative; if arrival is late, the phase error is positive.

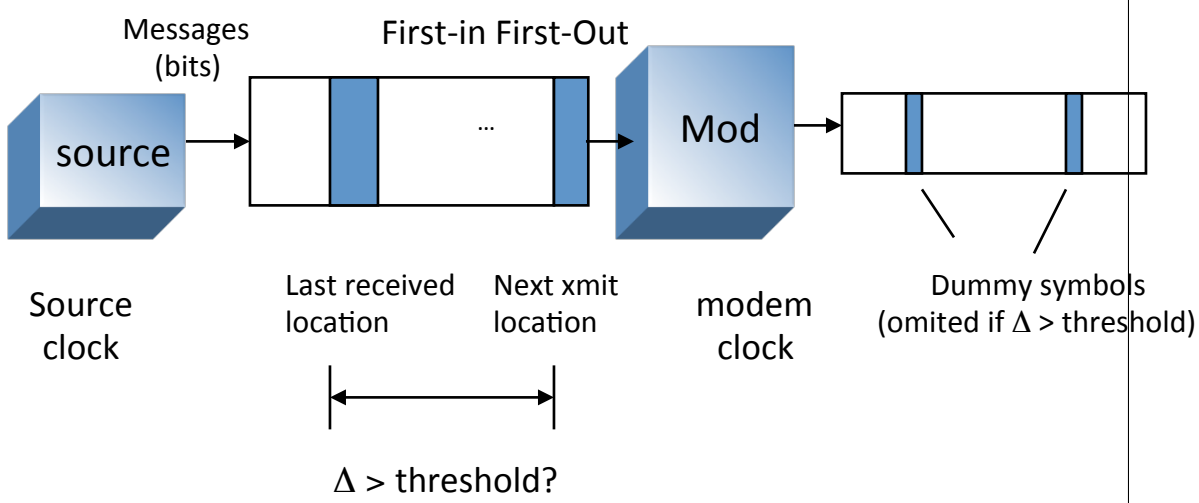


Figure 6.49: Rob-Stuff Timing.

Exercises - Chapter 6

6.1 *Phase-locked-loop error induction - 7 pts*

a. (1 pt) Use the first -order PLL transfer function

$$\frac{\hat{\Theta}(D)}{\Theta(D)} = \frac{\alpha \cdot D}{1-(1-\alpha)\cdot D}$$

to show that

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha \cdot \phi_k$$

where $\phi_k = \theta_k - \hat{\theta}_k$.

b. (3 pts) Use Part a's result to show by induction that when $\theta_k = \theta_0$ and $\hat{\theta}_0 = 0$, then:

$$\phi_k = (1-\alpha_k)^k \cdot \theta_0.$$

This was shown earlier in another way in Section 6.2.

c. (3 pts) Use the result of part (a) to show (again by induction) that when $\theta_k = \theta_0 + k\Delta$ and $\hat{\theta}_0 = 0$, then:

$$\phi_k = (1-\alpha)^k \cdot \theta_0 + \Delta \cdot \sum_{n=0}^{k-1} (1-\alpha)^n.$$

Also show that this confirms the result of Section 6.2 that ϕ_k converges to Δ/α.

6.2 *Second-order PLL update equations. (7 pts)*

This problem derives equations (6.63) - (6.65) as well as (6.70)-(6.73) in Section 6.2. The derivation starts from the transfer function for the phase prediction and then derives the second-order phase-locked-loop update equations.

a. (2 pts) Using only the transfer function from Equation (6.65)

$$\frac{\hat{\Theta}(D)}{\Theta(D)} = \frac{(\alpha+\beta)D - \alpha D^2}{1-(2-\alpha-\beta)D + (1-\alpha)D^2}$$

show that

$$\frac{\Phi(D)}{\hat{\Theta}(D)} = \frac{(1-D)^2}{(\alpha+\beta)\cdot D - \alpha \cdot D^2} \quad ,$$

recalling that $\phi_k = \theta_k - \hat{\theta}_k$.

b. (2 pts) Use the Part a's given and results to show that

$$\hat{\Theta}(D) = D \cdot \hat{\Theta}(D) + \alpha \cdot D \cdot \Phi(D) + \frac{\beta \cdot \Phi(D)}{1-D}.$$

and show that this is equivalent to the equation

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha \cdot \phi_k + \beta \cdot \sum_{l=0}^{k} \phi_l \quad .$$

c. (3 pts)

Now, defining as in Equation (6.63)

$$\hat{\Delta}(D) = \frac{\beta \cdot \Phi(D)}{1-D}$$

show that

$$\hat{\Delta}_k = \hat{\Delta}_{k-1} + \beta \cdot \phi_k \tag{6.212}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \alpha \cdot \phi_k + \hat{\Delta}_k \quad . \tag{6.213}$$

6.3 *PLL Frequency Offset*

A discrete-time PLL updates at 10 MHz and has a local oscillator frequency of 1 MHz. This PLL must try to phase lock to an almost 1 MHz sinusoid that may differ by 100 ppm (parts per million - so equivlaently a .01% difference in frequency). A first-order filter is used with α as given in this chapter. The PLL's steady-state phase error should be no more than .5% ($.01\pi$).

a. Find the value of α that ensures $|\phi| < .01(\pi)$. (2 pts)

b. Is this PLL stable if the 1 MHz is changed to 10 GHz? (1 pt)

c. How could the potential instability of this PLL be addressed without using a 2nd-order filter? (2 pts)

6.4 *First-order PLL - 7 pts*

A receiver samples a channel output $y(t)$ at rate $2/T$, where T is the transmitter symbol period. This receiver only needs the clock frequency (not phase) exactly equal to $2/T$, so a first-order PLL with constant phase error would suffice for timing recovery. (The update rate of the PLL is 400 kHz)

The transmit symbol rate is 400 kHz $\pm$50 ppm. The receiver VCXO has its crystal frequency at 800 kHz $\pm$50 ppm.

a. Find the greatest possible frequency difference between the transmitter's clock phase and frequency at $2/T$ and the receiver's clock phase and frequency at $2/T$. (2 pts)

b. Find the smallest value of the first-order PLL α that prevents cycle slipping or say that cycle-slip aversion is not possible. (2 pts)

c. Repeat part b if the transmit clock accuracy and VCO accuracy are 1%, instead of 50 ppm. (2 pts)

d. Suggest a potential fix for the cycle slipping of Parts b and c. (1 pt)

6.5 *Phase Hits*

A phase hit is a sudden change in the phase of a modulated signal, which can be modeled as a step-phase input into a phase-lock loop, without change in the frequency before or after the step in phase. A first-order discrete-time PLL operates with a sampling rate of 1 kHz (i.e., correct phase errors are correctly supplied somehow 1000 times per second) on a sinusoid at frequency approximately 640 Hz, which may have phase hits. Quick reaction to a phase hit unfortunately also forces the PLL to magnify random noise, so there is a trade-off between speed of reaction and noise immunity. The answer may assume sufficient inter-phase-hit time for a PLL to converge.

a. What is the maximum phase-hit magnitude (in radians) with which the designer must be concerned? [hint: think of cycle clips] (1 pt)

b. What value of the PLL parameter α will allow convergence to a 1% phase error (this means the phase offset on the 640 Hz signal is less than $.02\pi$) in just less than 1 s? Assume the value you determine is just enough to keep over-reaction to noise under control with this PLL. (2 pts)

c. For the α in part b, find the steady-state maximum phase error if the both the local oscillator and the transmitted clock each individually have 50 ppm accuracy (hint: the analysis for this part is independent of the phase hit)? What percent of the 640 Hz sinusoid's period is this steady-state phase error? (2 pts)

d. How might you improve the overall design of the phase lock loop in this problem so that the 1% phase error (or better) is maintained for both hits (after 1s) and for steady-state phase error, without increasing noise over-reaction? (1 pt)

6.6 *Square-Law Jitter Estimation - 9 pts*

A square-law timing recovery system with 50% excess bandwidth and symbol rate $1/T = 1$ tracks a sinusoid with a jittery or noisy phase of $\theta_k = \Delta_0 \cdot k + \theta_0 + n_k$ where n_k is zero-mean AWGN with variance σ_θ^2.

a. If a first-order PLL with parameter α is used, what is a formula for $E[\phi_\infty]$? (1 pt)

b. What is a formula for the mean-square steady-state value of this first-order PLL phase error? (1 pt)

c. Another way of viewing the effect of jitter more simply may derive by viewing the symbol-time sampling error ϵ as small compared to a symbol interval so that $e^{\jmath 2\pi f \epsilon T} \approx 1 + \jmath 2\pi f \epsilon T$ for all frequencies of interest. In this case, what largest value of ϵ in percentage of jitter that would insure less than .1 dB degradation in an equalizer/receiver output error of 13.5 dB? (3 pts)

d. With $\Delta = .1$ and $\sigma_\theta^2 = (.005 \cdot 2\pi)^2$, is there a first-order PLL α value for which the SNR is satisfied using the formula in Part b? If so, what is it? (1 pt)

e. Propose a better design. Try to estimate the new phase-error standard deviation of your design. (3 pts)

The next 4 problems are based on some provided by former EE379A student Jason Allen.

6.7 *Frequency-Offset Study- 12 pts*
A 2B1Q data signal

$$x(t) = \sum_k x_k \cdot p(t - kT)$$

is multiplied by a sinusoidal carrier with a frequency f_c to be transmitted over a wireless communications channel. At the receiver this multiplication must be reversed to recover the baseband data signal. One simple carrier demodulation might be accomplished by multiplying the received data signal by a locally generated sinusoid whose frequency is also f_c Hz, and whose phase is identical to that of the carrier signal used at the transmitter. The original signal is then recovered by lowpass filtering as in Section 6.3. Since the correct phase is likely unknown a priori, a PLL is necessary. The receiver local oscillator frequency is instead

$$f_{LO,Rx} = f_{LO,Tx} + \Delta f$$

where Δf represents the frequency offset between the receiver's PLL center frequency $f_{LO,Rx}$ and the transmitter's frequency $f_{LO,Tx}$. Ignoring the phase offset, find:

a. f_{sum}, the frequency of the sum-frequency term (sometimes called the "sum image"). (1 pt)

b. f_{diff}, the frequency of the difference image. (1 pt)

c. What is the result after lowpass filtering? (2 pts)

d. Is the filtered output signal still at baseband after multiplying by the carrier (and lowpass filtering)? (1 pt)

e. Denoting the transmitted signal as $x(t)$ with an equivalent Fourier transform of $X(f)$, the "near-baseband" signal can be denoted as $X(f - \Delta f)$. What is the time domain representation of the "near-baseband" signal? (2 pts)

f. What effect will this have on the data signal's constellation? Will this effect on the data signal constellation also effect the best symbol-by-symbol decision device (slicer)? (3 pts)

g. How might the ambiguity in Part f be resolved? (2 pts)

6.8 *Quadrature Detector- 5 pts*

a. Let $\hat{x}(t)$ be a quadrature input sinusoid given by

$$x(t) = cos(\theta_x(t)) + jsin(\theta_x(t))$$

and $y(t)$ be the locally generated sinusoid given by

$$y(t) = cos(\theta_y(t)) - jsin(\theta_y(t))$$

A quadrature phase detector generates its error by multiplying the two signals together and taking the imaginary part of the output. Find the error signal for the given signals.

6.9 *3rd Order Loops - 10 pts*

Using the transfer function given in Equation 6.38 for the phase error and input phase, this problem explores the outcomes of the following situations (assume updating occurs at the sampling rate).

a. Given a phase lock loop that has converged, how will a first-order PLL respond to a phase step $\theta_k = \theta_0 \cdot u_k$? From how large of a phase step can a PLL recover? (2 pts)

b. Given a phase lock loop that has converged, how will a second-order PLL respond to a frequency step so that the change in frequency starting at some time zero is the original phase plus $\theta_k = \Delta_0 \cdot k \cdot u_k$? (2 pts)

c. In situations where the transmitter or receiver (or both) are moving with respect to each other, a Doppler Effect is seen by the PLL as a changing frequency per time, or as a frequency ramp, which is modeled as the phase including an additive extra term of $\theta_k = \frac{\nu_0}{2} \cdot k \cdot (k-1) \cdot u_k$. What is the effect of this time-varying frequency change to the input signal on a converged second-order PLL? [hint: $k \cdot f_k \rightarrow D \cdot \frac{d}{dD} F(D)$] (2 pts)

d. Design a Loop Filter for a 3rd-Order PLL and show what effect it has in mitigating the error found in part c. (4 pts)

6.10 *Blue Tooth PLL - 8 pts*

"Bluetooth" communication supports short-range peer-to-peer voice and data transfer between portable devices. In the United States, Bluetooth utilizes the Industrial Scientific Medicine (ISM) band between 2.4 - 2.4835 GHz split into channels 0 - 79 according to

$$f_c = 2402 + i \cdot 1 \text{ MHz} \ .$$

There is a 2 MHz guard band at the low end and 3.5 MHz guard band at the high end. Bluetooth actually has a known frequency-hopping pattern for the index i that both the transmitter and receiver know, so the carrier frequency changes from one of the channels to another 1600 times per second, or 1600 hops/sec. This hopping allows robustness against frequency-selective fading in that a code is used on top of the system, similar to coded OFDM in Chapter 4 for Wi-Fi or LTE. Typical codes are rate 1/3 or 2/3 convolutional, see Chapter 10, which allow recovery of the bits lost to a fade on a hop. The VCO is used as a modulator for Bluetooth, according to output sinusoid

$$x(t) = A \cdot \cos\left(\omega_c t + 2\pi \cdot h \cdot \sum_{n=1}^{k} a_n \cdot p(t - nT)\right)$$

As in Section 6.6, $0 < h \leq 1$, and Bluetooth uses $h = 0.32$. $p(t)$ is the integral of a Gaussian pulse,

$$p(t) = \int_{-\infty}^{t} g(u) \cdot du$$

where $g(t) = \frac{1}{2} \cdot \frac{1}{2\pi} e^{-\frac{t^2}{2}}$ so that the integral over all time is equal to 1/2.

a. Write an expression for the VCO output frequency, calling the hop frequency $\omega_{c,hop}$. (1 pts)

b. Assuming the VCO has an operating voltage of 0 - 3.3 volts, what tuning slope is required to cover the stated ISM band range? (2 pts)

c. Design a symbol-by-symbol receiver for recovering the phase sequence $\sum_{n=1}^{k} a_n \cdot p(t-nT)$. (2 pts)

d. How quick should settling time of the VCO be if the input symbols are binary? What happens if they are 4 level? (2 pts)

e. Extra Credit: Where does the name Bluetooth come from? (1 pt)

6.11 *Half-Cycle Sinusoid - 12 pts*

The HCS frequency pulse response appears in Equation (6.141) . This problem investigates this form of CPM.

a. Find the corresponding phase pulse response in closed form. (2 pts)

b. How many continuous derivates does $p(t)$ have? What is n and how does the HCS spectra fall asymptotically in f? (2 pts)

c. Find a closed-form expression for the HCS power spectra density when $\nu = 1$. (2 pts)

d. Design a symbol-by-symbol receiver for recovering the phase sequence $\sum_{n=1}^{k} a_n \cdot p(t-nT)$. (2 pts)

e. Find the minimum distance for a DFE-like detector for HCS. (2 pts)

f. What is the minimum distance for ML detection with HCS? (2 pts)

g. (Extra Credit for 10 pts) Write programs like those of Appendix G.6 that will compute the PSD for $\nu \leq 10$ and plot the spectrum for 6HCS and compare it to RS.

6.12 *Some limiting values in MSK and Tamed Frequency Modulation central value - 6 pts*

a. Show that the TFM function $f_0(t)$ has value $f_0(0) = 1 + \pi^2/72$ using L'Hopital's Rule (maybe a few times). (3 pts)

b. What is the value for the MSK spectrum in Equation (6.194) frequency $f = 0.25T$? (3 pts)

6.13 *Spectrum for 1REC with h=1/4 - 15 pts*

The 1REC spectrum for $M = 2$ is developed in Example 6.6.1, specifically for $h = 1/2$ and thus MSK. A similar process is followed in this problem for $h = 1/4$.

a. For 1REC with $h = 0.25$ and $M = 2$, how long is the (nonzero portion of the) frequency pulse response in symbol periods? How long then is $r_{bb,1REC}(\tau)$? For positive lags τ? Why does the autocorrelation response for positive time lags have a length equal to the non-constant portion of the phase pulse response? (2 pts)

b. How many continuous derivates does $p(t)$ have? What is n and how does the 1REC spectra fall asymptotically in f? (1 pt)

c. By following Example 6.6.1, show that $r_{bb,1REC}(\tau)$ for $M = 2$ and $h = 1/4$ is as given in Equation (6.197). (5 pts)

d. Plot $r_{bb,1REC}(\tau)$ for $[0, 20T]$ and compare it to the same result as the R_nREC program in Appendix G.6. Explain any differences. (4 pts)

e. For your answers above, plot the spectrum for normalized frequencies over the range $[0, 10/T]$ [hint: think of the eaiest way to do this] (3 pts)

Bibliography

[1] John M. Cioffi and G. David Forney, Jr. *"Generalized Decision-Feedback Equalization for Packet Transmission with ISI and Gaussian Noise"*. Communications, Computation, Control, and Signal Processing, a tribute to Thomas Kailath. Springer, Boston, MA ISBN 978-1-4615-6281-8. Editors: Paulraj A., Roychowdhury V., Schaper C.D. pp. 79-127.

[2] Brian H. Marcus and Ron M. Roth and Paul H. Siegel. *"An Introduction to Codiing for Constrained Systems"*. 5th Edition IBM Research, 2001

Index

carrier frequency, 1033
carrier recovery, 1028
    decision-directed, 1029
    open-loop, 1028
continuous frequency modulation, 1037
continuous-phase modulation, 1033

digital frequency modulation, 1034

frequency
    offset, 1014
frequency modulation
    tamed, 1045

jitter, 1007, 1011
    phase, 1007, 1019
    timing, 1007

modulation
    index, 1033
MSK, 1037

phase
    tree, 1036
phase detector, 1007
    binary, 1009
    demodulating, 1008
    ideal, 1007
    modulo-2, 1007
phase error, 1005
phase locking, 1005
phase-detector
    sampled, 1009
PLL, 1013
    discrete-time, 1015
    first-order, 1013, 1016
    second-order, 1018, 1019
pseudorandom sequence, 1080
pulse response
    frequency, 1034
    phase, 1033

raised cosine, 1039

sample and hold, 1009
SK
    Gaussain, 1043

timing recovery, 1021
    bandedge, 1023
    decision-aided, 1021
    decision-directed, 1024
    envelope, 1022
    open-loop, 1021
    pilot-based, 1025

VCO, 1010
VCXO, 1011
Viterbi Algorithm, 1039