Chapter 10

Timing Recovery

From the eye patterns presented in Chapter 6, we recall that a demodulated digital communication signal has to be sampled at a proper timing phase before successful detection of the transmitted data. In systems where channel distortion is minimal and noise level is low, e.g., the cases shown in Figures 6.2 and 6.3, the timing phase, obviously, has to be at the point where the eye pattern has maximum opening. In such cases, visual identification of the best timing phase from an eye pattern is straightforward. Accordingly, one may think of and develop simple algorithms for timing recovery and tracking. On the other hand, when the distortion introduced by the channel is significant, e.g., the case shown in Figure 6.4, it is not obvious what criterion should be used for timing recovery.

As in any adaptive algorithm, a timing recovery algorithm is also developed based on a cost function whose optimization leads to the desired timing information. In this chapter, we discuss two classes of timing recovery methods: (i) non-data aided methods; and (ii) data aided methods.

In non-data aided timing recovery methods, the statistical characteristics of digital data signals are used in choosing a relevant cost functions. For instance, a detailed study of digital data signals reveals that their ensemble average power is a periodic function of time whose period equal to the incoming symbol interval. Moreover, the peak point of this function gives a timing phase which is optimum in a sense that is defined later, in this chapter. Hence, a PLL that locks to this periodic function can be used to generate a clock synchronous with the received signal. In addition, aligning the generated clock with the peaks of the proposed cost function allows one to design an optimum timing recovery loop.

Data aided methods are based on cost functions that depend on the transmitted data symbols. Since the transmitted data symbols are unknown to the receiver, tentative decisions at the receiver are replaced for the transmitted symbols. Such algorithms perform well as long as the decision errors are at an acceptably low level.

10.1 Non-Data Aided Timing Recovery Methods

10.1.1 Fundamental results

We recall from Chapter 3 that a communication channel, in general, is characterized by an equivalent complex-valued baseband impulse response $c_{\mathrm{BB}}(t)$. Accordingly, the demodulated received signal at the baseband (assuming that carrier recovery has already been established) is given by

$$y(t) = \sum_{n=-\infty}^{\infty} s[n] c_{\mathrm{BB}}(t - nT_b) \tag{10.1}$$

where $s[n]$s are data symbols and T_b is the baud/symbol interval. The data symbols are in general from a complex-valued constellation.

The key idea in development of non-data aided timing recovery stems from the following observation. If $s[n]$s are a set of independent and identically distributed symbols with mean of zero and variance of σ_s^2, we obtain the ensemble mean-square/power of $y(t)$ as

$$\rho(t) \triangleq E[|y(t)|^2] = \sigma_s^2 \sum_{n=-\infty}^{\infty} |c_{\mathrm{BB}}(t - nT_b)|^2. \tag{10.2}$$

By direct inspection, we note that $\rho(t)$ is a periodic signal with period of T_b. It thus can be expanded using Fourier series as

$$\rho(t) = \sum_{n=-\infty}^{\infty} \rho_n e^{j2\pi nt/T_b} \tag{10.3}$$

where ρ_ns are the Fourier series coefficients given by

$$\rho_n = \frac{1}{T_b} \int_0^{T_b} \rho(t) e^{-j2\pi nt/T_b} dt. \tag{10.4}$$

Substituting (10.2) in (10.4), we obtain

$$\begin{aligned} \rho_n &= = \frac{1}{T_b} \int_0^{T_b} \left(\sigma_s^2 \sum_{n=-\infty}^{\infty} |c_{\mathrm{BB}}(t - nT_b)|^2 \right) e^{-j2\pi nt/T_b} dt \\ &= \frac{\sigma_s^2}{T_b} \sum_{n=-\infty}^{\infty} \int_0^{T_b} |c_{\mathrm{BB}}(t - nT_b)|^2 e^{-j2\pi nt/T_b} dt \\ &= \frac{\sigma_s^2}{T_b} \int_{-\infty}^{\infty} |c_{\mathrm{BB}}(t)|^2 e^{-j2\pi nt/T_b} dt \end{aligned} \tag{10.5}$$

where the second line follows by changing the order of the integral and summation, and the third line follows by introducing the change of variable $t - nT_b$ to t and noting that the resulting integrals add up to a single integral over the range $-\infty < t < \infty$.

Next, we recall from Chapter 2 that, for any pair of functions $x(t)$ and $y(t)$,

$$\mathcal{F}[x(t)y(t)] = X(f) \star Y(f). \tag{10.6}$$

Also, we note that in (10.5), $\int_{-\infty}^{\infty} |c_{\rm BB}(t)|^2 e^{-j2\pi nt/T_b} dt$ is the Fourier transform of $|c_{\rm BB}(t)|^2 = c_{\rm BB}(t)c^*_{\rm BB}(t)$ at $f = n/T_b$,. Hence, substituting $x(t) = c_{\rm BB}(t)$ and $y(t) = c^*_{\rm BB}(t)$ in (10.6), and noting that $\mathcal{F}[c^*_{\rm BB}(t)] = C^*_{\rm BB}(-f)$, we get

$$\begin{aligned}\rho_n &= \frac{\sigma_s^2}{T_b} C_{\rm BB}(f) \star C^*_{\rm BB}(-f)|_{f=\frac{n}{T_b}} \\ &= \frac{\sigma_s^2}{T_b} \int_{-\infty}^{\infty} C_{\rm BB}(f) C^*_{\rm BB}\left(f - \frac{n}{T_b}\right) df. \end{aligned} \tag{10.7}$$

From (10.7), the following observations are made:

- $$\rho_0 = \frac{\sigma_s^2}{T_b} \int_{-\infty}^{\infty} |C_{\rm BB}(f)|^2 df \tag{10.8}$$

  is a real and positive number.

- $$\rho_1 = \frac{\sigma_s^2}{T_b} \int_{-\infty}^{\infty} C_{\rm BB}(f) C^*_{\rm BB}\left(f - \frac{1}{T_b}\right) df \tag{10.9}$$

  and

  $$\begin{aligned}\rho_{-1} &= \frac{\sigma_s^2}{T_b} \int_{-\infty}^{\infty} C_{\rm BB}(f) C^*_{\rm BB}\left(f + \frac{1}{T_b}\right) df \\ &= \frac{\sigma_s^2}{T_b} \int_{-\infty}^{\infty} C_{\rm BB}\left(f - \frac{1}{T_b}\right) C^*_{\rm BB}(f) df \\ &= \rho_1^*. \end{aligned} \tag{10.10}$$

- In almost all practical channels, the excess bandwidth of the transmit pulse shaping filter is less than 100%; e.g., when $p_{\rm T}(t)$ is a square-root raised-cosine pulse shape with role of factor $\alpha \leq 1$, the excess bandwidth is $100\alpha\%$. In such cases, $C_{\rm BB}(f) = 0$ for $|f| > 1/T_b$. Using this, one finds that $\rho_n = 0$, for $|n| > 1$.

From the above observations, we conclude that

$$\begin{aligned}\rho(t) &= \rho_0 + \rho_1 e^{j2\pi t/T_b} + \rho_1^* e^{-j2\pi t/T_b} \\ &= \rho_0 + 2|\rho_1| \cos\left(\frac{2\pi}{T_b} t + \angle\rho_1\right) \end{aligned} \tag{10.11}$$

where $|\rho_1|$ and $\angle\rho_1$ are the amplitude and phase of ρ_1, respectively.

10.1.2 The timing recovery cost function

Let us consider the case where the received signal $y(t)$ is sampled at the time instants $\tau + nT_b$, where τ is called *timing phase*. From the above results, one readily finds that $E[|y(\tau + nT_b)|^2] = E[|y(\tau)|^2] = \rho(\tau)$. Hence, we obtain

$$\rho(\tau) = \rho_0 + 2|\rho_1| \cos\left(\frac{2\pi}{T_b}\tau + \angle\rho_1\right). \tag{10.12}$$

We refer to $\rho(\tau)$ as the *timing recovery cost function*.

Figure 10.1 presents a plot of the timing recovery cost function $\rho(\tau)$ versus the timing phase τ. As seen, and, of course, could be understood by direct inspection of (10.12), $\rho(\tau)$ is a periodic function of τ with a period of T_b, the maximum value of $\rho_0 + 2|\rho_1|$, and the minimum value of $\rho_0 - 2|\rho_1|$. Moreover, either the minima or maxima points of $\rho(\tau)$ provide reference points where one may choose to lock the symbol clock rate to. This leads to algorithms that keep the receiver synchronized with the incoming symbols. In addition, a timing phase that results in the best performance of the receiver should be chosen. We refer to this as the *optimum timing phase*.

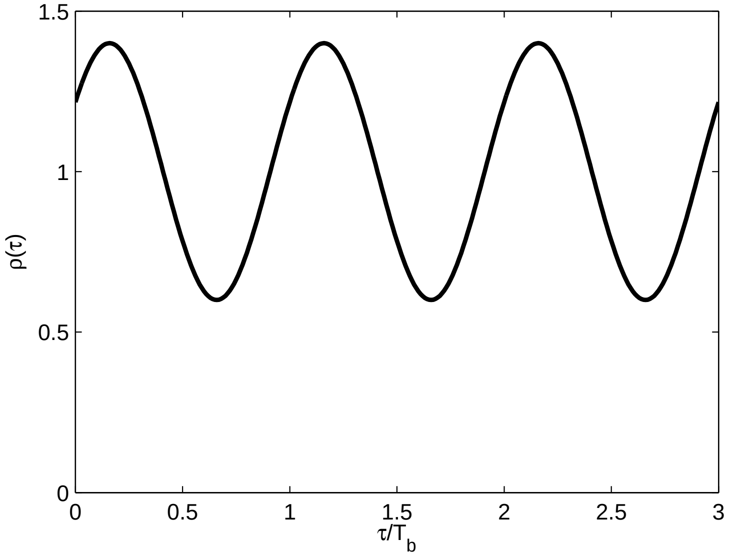

Figure 10.1: The timing recovery cost function $\rho(\tau)$.

10.1.3 The optimum timing phase

It turns out that, in general, the optimum timing phase is obtained by maximizing the cost function $\rho(\tau)$. To provide a clear understanding of this interesting result, we first consider the case where the channel is ideal. In this case, $C_{\mathrm{BB}}(f) = P(f)$ is a Nyquist pulse. Also, we recall from Chapter 4 that $p(t) = \mathcal{F}^{-1}[P(f)]$ is a real-valued function of time. This implies that $P(f) = P^*(-f)$. Moreover, for convenience of discussion, we assume that $P(f)$ has zero phase. It is thus a real-valued function of f and, hence, $P(f) = P(-f)$. On the other hand, we recall the Nyquist criterion

$$\frac{1}{T_b}\sum_{n=-\infty}^{\infty} P\left(f-\frac{n}{T_b}\right) = 1. \tag{10.13}$$

We also recall from Chapter 4 that the Fourier transform of the sampled signal

$$p_s(t) = \sum_{n=-\infty}^{\infty} p(nT_b)\delta(t-nT_b) \tag{10.14}$$

is

$$P_s(f) = \frac{1}{T_b}\sum_{n=-\infty}^{\infty} P\left(f-\frac{n}{T_b}\right) = 1 \tag{10.15}$$

Figure 10.2 visualizes this result, for $0 \le f \le \frac{1}{T_b}$. We may also note that the graphs are given for the important case where $P(f)$ has an excess bandwidth of less than 100% and in this case, for $0 \le f \le 1/T_b$,

$$P_s(f) = \frac{1}{T_b}\left(P(f) + P\left(f-\frac{1}{T_b}\right)\right). \tag{10.16}$$

Next, consider the case where $p(t)$ is sampled at the time instants $\tau + nT_b$, i.e., at a timing phase τ. This results in the sampled signal

$$p_s(t,\tau) = \sum_{n=-\infty}^{\infty} p(\tau+nT_b)\delta(t-\tau-nT_b). \tag{10.17}$$

Applying the Fourier transform to both sides of (10.17), we get

$$\begin{aligned} P_s(f,\tau) &= \sum_{n=-\infty}^{\infty} p(\tau+nT_b)e^{-j2\pi f(\tau+nT_b)} \\ &= e^{-j2\pi f\tau}\sum_{n=-\infty}^{\infty} p(\tau+nT_b)e^{-j2\pi f nT_b} \\ &= \frac{1}{T_b}e^{-j2\pi f\tau}\left(P(f)e^{j2\pi f\tau} + P\left(f-\frac{1}{T_b}\right)e^{j2\pi(f-1/T_b)\tau}\right) \\ &= \frac{1}{T_b}\left(P(f) + P\left(f-\frac{1}{T_b}\right)e^{-j2\pi\tau/T_b}\right) \end{aligned} \tag{10.18}$$

where the third line is obtained by noting that $\sum_{n=-\infty}^{\infty} p(\tau+nT_b)e^{-j2\pi f nT_b}$ is the Fourier transform of the sampled version of $p(t+\tau)$ at the time instants nT_b, $\mathcal{F}[p(t+\tau)] = P(f)e^{j2\pi f\tau}$, and applying (10.16) to this case.

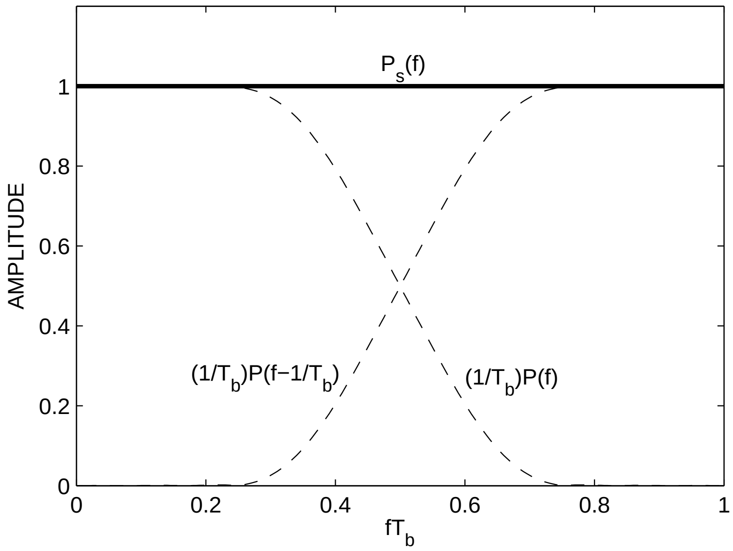

Figure 10.2: Visualization of (10.16).

The phase-shift introduced by the delay τ will put $P(f)$ and $P(f-1/T_b)$ out of phase and as a result over the portion of the frequency band where $P(f)$ and $P(f-1/T_b)$ overlap, $P_s(f,\tau)$ reduces in amplitude. In particular, when $\tau = 0.5T_b$, $e^{-j2\pi\tau/T_b} = -1$ and, hence, (10.18) reduces to

$$P_s(f,\tau) = \frac{1}{T_b}e^{-j2\pi f\tau}\left(P(f) - P\left(f - \frac{1}{T_b}\right)\right). \tag{10.19}$$

Figure 10.3 presents plots of $|P_s(f,\tau)|$ for three choices of $\tau = 0.2T_b$, $0.3T_b$ and $0.5T_b$. Here, $p(t)$ is a raised-cosine pulse with roll-off factor $\alpha = 0.5$. We note that as τ increases from 0 to $0.5T_b$, a notch is developed in the transfer function $P_s(f,\tau)$.

Another important and relevant point which needs our attention is that the data sequence $s[n]$ and the output sequence $y[n,\tau] = y(\tau + nT_b)$ are related through a discrete-time channel whose impulse response is the sequence $p(\tau + nT_b)$. The frequency response of this channel is given by the Fourier transform of the sequence $p(\tau + nT_b)$, viz.,

$$P(e^{j2\pi f},\tau) = \sum_{n=-\infty}^{\infty} p(\tau + nT_b)e^{-j2\pi nf}. \tag{10.20}$$

Also, using the results of Section 4.3.1, one may find that

$$P(e^{j2\pi f},\tau) = P_s\left(\frac{f}{T_b},\tau\right). \tag{10.21}$$

This result is obtained by direct application of (4.45), replacing $x(t)$ by $p(\tau + t)$ and T_s by T_b. This observation show that the channel, here, may be treated as

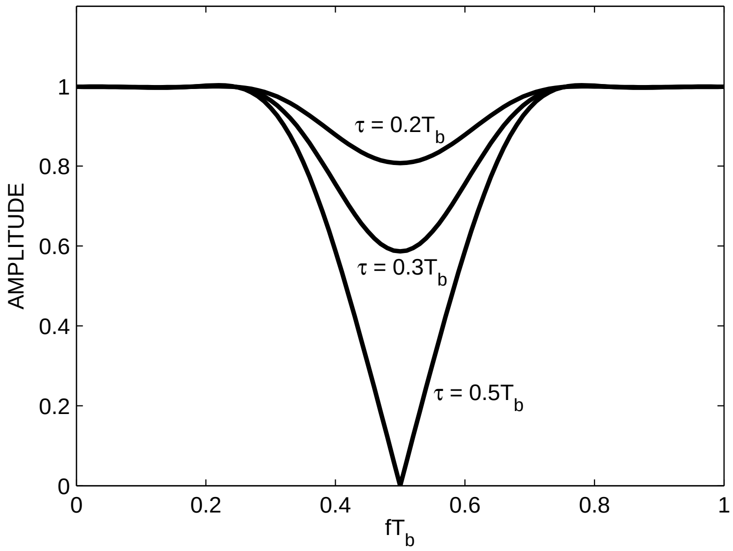

Figure 10.3: The impact of sampling phase on the amplitude of $P_s(f,\tau)$.

a discrete-time system with input $s[n]$ and output $y[n,\tau]$. Accordingly, the power spectral density of the output signal $y[n,\tau]$ is given by

$$\Phi_{yy}(e^{j2\pi f},\tau) = \Phi_{ss}(e^{j2\pi f})|P(e^{j2\pi f},\tau)|^2 = \Phi_{ss}(e^{j2\pi f})\left|P_s\left(\frac{f}{T_b},\tau\right)\right|^2 \tag{10.22}$$

where $\Phi_{ss}(e^{j2\pi f})$ is the power spectral density of the discrete-time sequence $s[n]$. Moreover, using the Rayleigh's relation (4.36), we obtain

$$E[|y[n,\tau]|^2] = \int_0^1 \Phi_{yy}(e^{j2\pi f},\tau)df = \int_0^1 \Phi_{ss}(e^{j2\pi f})\left|P_s\left(\frac{f}{T_b},\tau\right)\right|^2 df. \tag{10.23}$$

Finally, considering the plots of $|P_s(f,\tau)|$ in Figures 10.2 and 10.3, we find that $\tau = 0$ maximizes $E[|y[n,\tau]|^2]$. Also, $\tau = 0$ is the optimum timing phase since it maximizes the eye-opening. This follows from

$$p(0+nT_b) = p(nT_b) = \begin{cases} 1, & n=0 \\ 0, & \text{otherwise} \end{cases} \tag{10.24}$$

which implies $y[n,0] = s[n]$ and this corresponds to the case where there is no ISI. On the other hand, for any $\tau \neq 0$, in the range of $0 < \tau < T_b$, $p(\tau + nT_b)$ is non-zero for any n, hence, ISI will be present and the presence of ISI reduces the eye-opening.

The above discussion clearly shows that in a communication system with an ideal channel, the timing phase that maximizes the power of the signal samples is the optimum timing phase. We also note that this corresponds to the case where the aliased signal components are in phase and thus augment. Any deviation of the timing phase from the optimal phase results in attenuation of the signal spectra over the aliased band. Theoretically such distortion could be compensated through a channel equalizer which will be designed to amplify the signal over the bands that the spectra attenuation has occurred. However, such amplification also results in amplification/enhancement of the channel noise which clearly has a detrimental effect on the receiver performance.

When the channel is non-ideal, also, it is intuitively understandable that a timing phase that results in significant cancellation of aliased components reduces the power of the sampled signal. This clearly should correspond to a timing phase near the minimum of the timing recovery cost function $\rho(\tau)$. Moreover, equalization of sampled signal results in significant noise enhancement and thus poor performance of the receiver is expected. On the other hand, a timing phase that maximizes the power of the sampled signal corresponds to a case where the aliased components are mostly in phase and thus will augment. From this, one may argue that a timing phase that maximizes the cost function $\rho(\tau)$ leads to, at least, a near optimum timing phase.

10.1.4 Improving the cost function

Figure 10.4 presents plots of the cost function $\rho(\tau)$ for an ideal channel where $C_{BB}(f) = P(f)$ and $P(f)$ is the Fourier transform of a raised-cosine pulse-shape. The plots are given for three values of the rolloff factor $\alpha = 0.25$, 0.5 and 1. An important point to note here is that the variation/gradient of $\rho(\tau)$ reduces with α. Also, in an adaptive setting, a stochastic gradient (similar to the one in the LMS algorithm) is used to search for the timing phase that maximizes $\rho(\tau)$. In addition, we note that the variance of a stochastic gradient is approximately proportional to the magnitude of the underlying signal power. Hence, noting that when the timing phase is near it optimal value, the signal power is almost independent of α (see Figures 10.2 and 10.3), one may argue that the stochastic gradients used for timing recovery become less reliable as α decreases.

To overcome the above problem, we proceed with an intuitive reasoning of why the gradient of $\rho(\tau)$ reduces with α and from there suggest a method of modifying the cost function $\rho(\tau)$ such that it will be less dependent on α. Referring to Figure 10.3, one finds that the variation of the received signal power as a function of timing phase, τ, is a direct consequence of augmentation or cancellation of the aliased signal components as τ varies. Moreover, if we note that the amount of aliased components reduces with α, it becomes obvious that the variation of $\rho(\tau)$ with τ reduces with α. Extending this argument, we suggest, to obtain a cost function which will be less dependent on α, one should only concentrate on the signal power over the band of the aliased components. This can be done easily by passing the

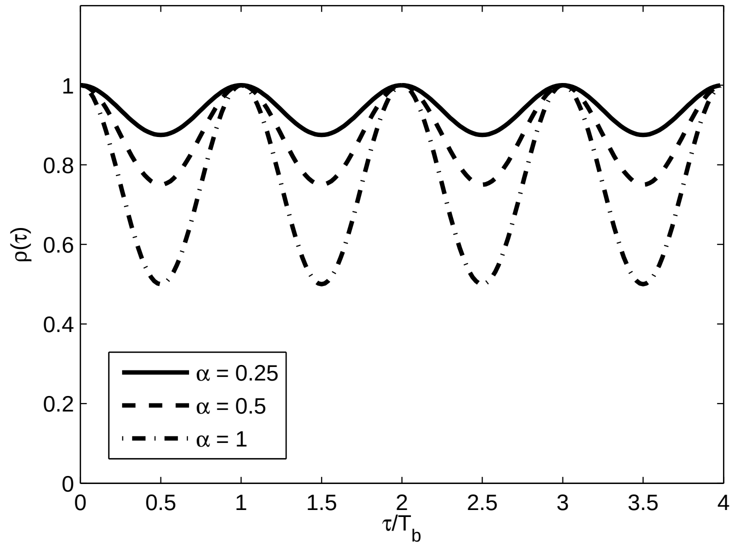

Figure 10.4: Plots of the timing recovery cost function $\rho(\tau)$ for an ideal channel and a raised-cosine pulse-shape with three values of rolloff factor α.

received signal through a bandpass filter that is centered around $1/2T_b$ and choosing the output power of this filter as the timing recovery cost function.

Figure 10.5 presents a set of plots of a modified cost function that is obtained by taking the T_b-spaced samples of the received signal and passing them through a single pole highpass filter with the transfer function

$$B(z) = \frac{\sqrt{1-\beta^2}}{1+\beta z^{-1}} \tag{10.25}$$

where $0 < \beta < 1$ determines the bandwidth of the filter and the factor $\sqrt{1-\beta^2}$ is to normalize the power gain of the filter, for a white input, to unity. We refer to this cost function as $\rho_\beta(\tau)$. We also note that $\rho(\tau)$ can be thought as a special case of $\rho_\beta(\tau)$ which is obtained by choosing $\beta = 0$. For the results presented in Figure 10.5, β is chosen equal to 0.95. As expected, unlike $\rho(\tau)$ which varies significantly with α, $\rho_\beta(\tau)$, for β close to one, remains nearly the same for all values of α.

10.2 Non-Data Aided Timing Recovery Algorithms

Application of the cost function $\rho(\tau)$, or its modified version $\rho_\beta(\tau)$, leads to a variety of timing recovery/tracking algorithms. Here, we emphasize and introduce algorithms that operate based on samples of $y(t)$ that are taken at a spacing of T_b/L, where L is an integer. This we have found convenient and relevant to the

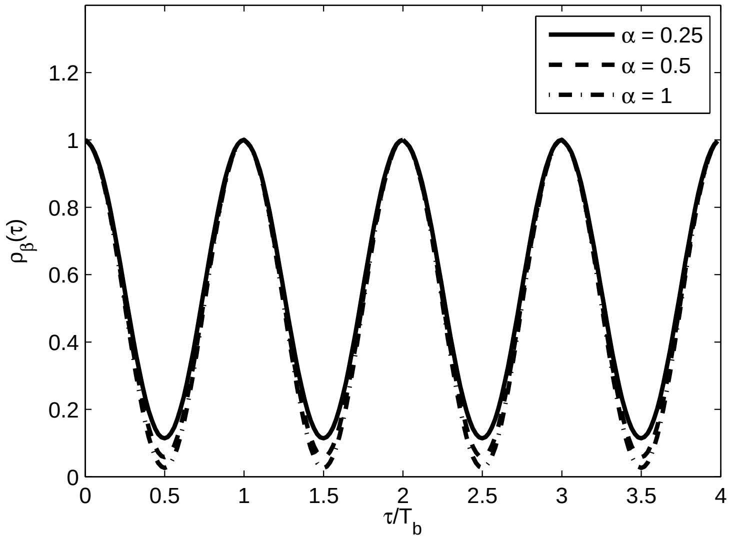

Figure 10.5: Plots of the improved timing recovery cost function for an ideal channel and a raised-cosine pulse-shape with three values of the rolloff factor α. The parameter β is set equal to 0.95.

content of this text since in a software radio setting, the demodulated baseband signal is always obtained by decimating a higher rate signal. For example, in the all digital receiver structure that was presented in Chapter 5, Section 5.8.3, the baseband signal is obtained by decimating a signal that originates from an IF signal whose sampling rate is usually an order of magnitude or more greater than the baud rate $f_b = 1/T_b$. Moreover, if the interpolation method mentioned in Section 5.8.1 is used, signal samples at a much higher rate will also be accessible and thus quantized values of the timing phase τ at any desired precision can be made available.

10.2.1 Early-late gate timing recovery

Early-late gate is one the most common methods of timing recovery and can be applied to the variety of cost functions. To develop an early-late gate timing recovery algorithm based on the cost function $\rho(\tau)$ we proceed as follows.

We recall that the goal of timing recovery is to choose a timing phase $\tau = \tau_{\rm opt}$ which maximizes $\rho(\tau)$ (or its modified version, $\rho_\beta(\tau)$). We also note that when $\tau = \tau_{\rm opt}$ and $\delta\tau$ is a timing phase deviation, $\rho(\tau + \delta\tau) - \rho(\tau - \delta\tau) = 0$. On the other hand, for a non-optimum timing phase τ and a small $\delta\tau$, we note that $\rho(\tau + \delta\tau) - \rho(\tau - \delta\tau) > 0$, when $\tau < \tau_{\rm opt}$, and $\rho(\tau + \delta\tau) - \rho(\tau - \delta\tau) < 0$, when $\tau > \tau_{\rm opt}$. The is demonstrated in Figure 10.6 for a case where $\tau < \tau_{\rm opt}$.

In the light of the above observation, one may propose the following update

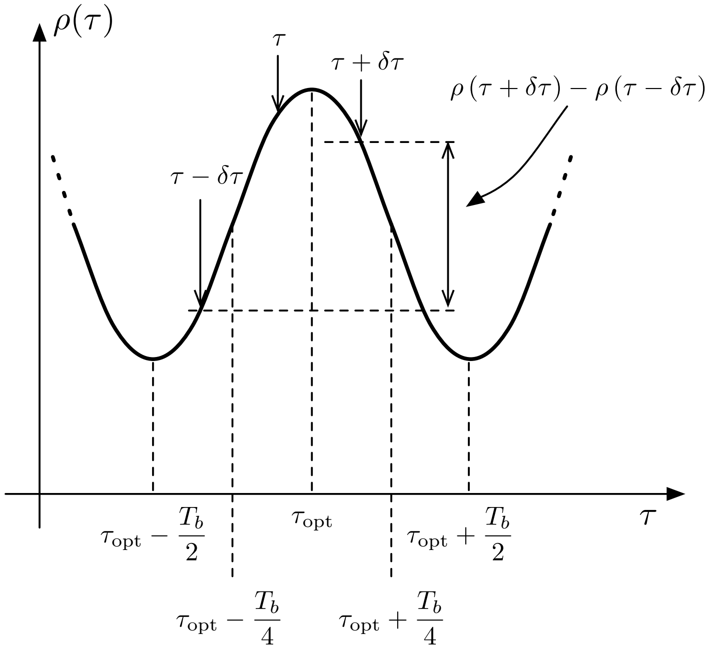
$\rho(\tau)$
τ
$\tau+\delta\tau$
$\rho(\tau+\delta\tau)-\rho(\tau-\delta\tau)$
$\tau-\delta\tau$
$\tau_{\rm opt}-\frac{T_b}{2}$
$\tau_{\rm opt}$
$\tau_{\rm opt}+\frac{T_b}{2}$
τ
$\tau_{\rm opt}-\frac{T_b}{4}$
$\tau_{\rm opt}+\frac{T_b}{4}$

Figure 10.6: A demonstration of the early-late timing recovery information.

equation for adaptive adjustment of the timing phase:

$$\tau[n+1]=\tau[n]+\mu\left(\rho\left(\tau[n]+\delta\tau\right)-\rho\left(\tau[n]-\delta\tau\right)\right) \tag{10.26}$$

where μ is a step-size parameter. Moreover, we note that in practice the cost function $\rho(\tau)$ is not available and only could be estimated based on the observed signal samples. Say, by taking the average of the squares of a few recent samples of $y(t)$. Or, we may follow the philosophy of the LMS algorithm and simply use $|y(\tau+nT_b)|^2$ as an estimate of $\rho(\tau)$. Applying such coarse estimates, we obtain the update equation

$$\tau[n+1]=\tau[n]+\mu\left(\left|y\left(\tau[n]+\delta\tau+nT_b\right)\right|^2-\left|y\left(\tau[n]-\delta\tau+nT_b\right)\right|^2\right). \tag{10.27}$$

The timing phase deviation $\delta\tau$ is a design parameter whose value (as long as it stays smaller than $T_b/4$) has very little effect on the algorithm performance; see the numerical results presented in Figure 10.7, below. Its value is usually dictated by the implementation consideration.

To be more specific, the equations used for realization of the early-late gate timing recovery with the modified cost function are summarized as follows:

$$y_1[n] \quad = \quad \sqrt{1-\beta^2}y\left(\tau[n]+\delta\tau+nT_b\right)-\beta y_1[n-1], \tag{10.28}$$

$$\begin{aligned} y_{-1}[n] &= \sqrt{1-\beta^2}\,y\left(\tau[n]-\delta\tau+nT_b\right)-\beta y_{-1}[n-1], & (10.29)\\ \tau[n+1] &= \tau[n]+\mu\left(|y_1[n]|^2-|y_{-1}[n]|^2\right). & (10.30)\end{aligned}$$

Note that $y_1[n]$ and $y_{-1}[n]$ are, respectively, sequences obtained by passing the signal samples $y\left(\tau[n]+\delta\tau+nT_b\right)$ and $y\left(\tau[n]-\delta\tau+nT_b\right)$ through the highpass filter $B(z)$.

To explore the performance of the timing recovery recursions presented above and also to show how they may be implemented in software, we present and use the MATLAB script ‘TR_ELG.m’. This script is presented below and is available on the accompanying CD. The reader is encouraged to use this program to gain better understanding of the behavior of the early-late gate timing recovery algorithm. ‘TR_ELG.m’ is similar to the script ’CRExp2.m’ that was presented in Chapter 9, with the last part of the program replaced by the timing recovery algorithm.

MATLAB Script TR_ELG.m: Early-late gate timing recovery

```
Tb=0.0001; L=100; M1=20; Ts=Tb/L; fs=1/Ts; fc=100000;
delta_c=0; N=8*L; phi_c=0.5; sigma_v=0; alpha=0.5; c=1;
b=sign(randn(10000,1));
M=input('QAM size (4, 16, 64, 256) =');
if M==4 s=b(1:2:end)+i*b(2:2:end);
elseif M==16 s=2*b(1:4:end)+b(2:4:end)+i*(2*b(3:4:end)+b(4:4:end));
elseif M==64 s=4*b(1:6:end)+2*b(2:6:end)+b(3:6:end)+...
          j*(4*b(4:6:end)+2*b(5:6:end)+b(6:6:end));
elseif M==256 s=8*b(1:8:end)+4*b(2:8:end)+2*b(3:8:end)+b(4:8:end)+...
          j*(8*b(5:8:end)+4*b(6:8:end)+2*b(7:8:end)+b(8:8:end));
else print('Error! M should be 4, 16, 64 or 256'); end
pT=sr_cos_p(N,L,alpha); xbbT=conv(expander(s,L),pT);
t=[0:length(xbbT)-1]'*Ts; xT=real(exp(i*2*pi*fc*t).*xbbT);
xR=conv(c,xT); xR=xR+sigma_v*randn(size(xR));
t=[0:length(xR)-1]'*Ts; y=exp(-i*(2*pi*(fc-delta_c)*t-phi_c)).*xR;
pR=pT; y=conv(y,pR);
%%%%%%%%%%%%%%%%%%%%%%%%
%   TIMING RECOVER: Early-late Gating   %
%%%%%%%%%%%%%%%%%%%%%%%%
beta=0; mu0=0.01; dtau=12; mu=mu0*(L/4)/dtau;
Ly=length(y); tau=0.3*ones(1,round(Ly/L)); kk=1; yp=0; ym=0; start=5*L+1
for k=start:L:length(tau)*L
   tauTb=round(tau(kk)*L);
   yp=sqrt(1-beta^2)*y(k+tauTb+dtau)-beta*yp;
   ym=sqrt(1-beta^2)*y(k+tauTb-dtau)-beta*ym;
   tau(kk+1)=tau(kk)+mu*(abs(yp)^2-abs(ym)^2); kk=kk+1;
end
figure, plot(tau(1:kk),'k')
xlabel('Iteration Number, n'), ylabel('tau[n]')
```

From Figure 10.6, we observe that the difference $\rho\left(\tau[n]+\delta\tau\right)-\rho\left(\tau[n]-\delta\tau\right)$, for $0<\delta\tau\leq L/4$ increases as $\delta\tau$ increases. This variation is almost linear with $\delta\tau$. Hence, μ should be given a value proportional the inverse of $\delta\tau$. This is taken care of in the script 'TR_ELG.m'.

Figure 10.7 presents a set of plots of τ generated by the MATLAB script 'TR_ELG.m'. The following parameters are used: $\beta=0$, $\mu_0=0.01$, 4QAM symbols, and four choices of $\delta\tau=25$ $(=L/4)$, 18, 12 and 1. From these results and further experiments that may be performed (using the MATLAB script 'TR_ELG.m'), one may observe that the convergence behavior of the early-late timing recovery algorithm remains independent of the parameter $\delta\tau$ over a relatively wide range within the interval 1 to $L/4$. Only when $\delta\tau$ approaches $L/4$, some drop in the convergence rate is observed. This observation can be explained, if we note that the ratio $\frac{\rho(\tau[n]+\delta\tau)-\rho(\tau[n]-\delta\tau)}{\delta\tau}$ reduces as $\delta\tau$ increases.

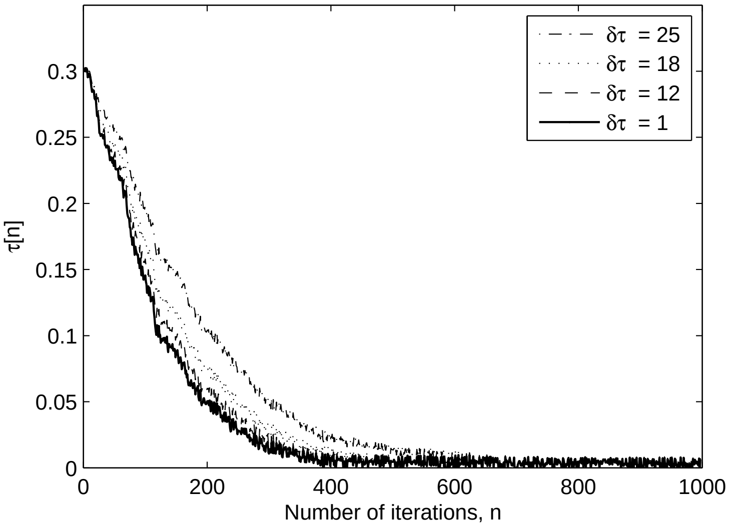

Figure 10.7: Plots of the timing phase update of the early-late gate timing recovery algorithm. The parameters used are : $\beta=0$, $\mu_0=0.01$, and four choices of $\delta\tau=25$ $(=L/4)$, 18, 12 and 1.

Figure 10.8 presents a plot of $\tau[n]$ as a function of n, when $\beta=0.9$, $\mu_0=0.005$, data symbols are from a 4QAM constellation, and $\delta\tau=12$. Comparing this result with those in Figure 10.7, we note that increasing β leads to an improved convergence rate, as predicted in Section 10.1.4. In addition, we observe a significant reduction in the timing phase jitter after convergence of the algorithm. This observation is also in line with the discussion in Section 10.1.4.

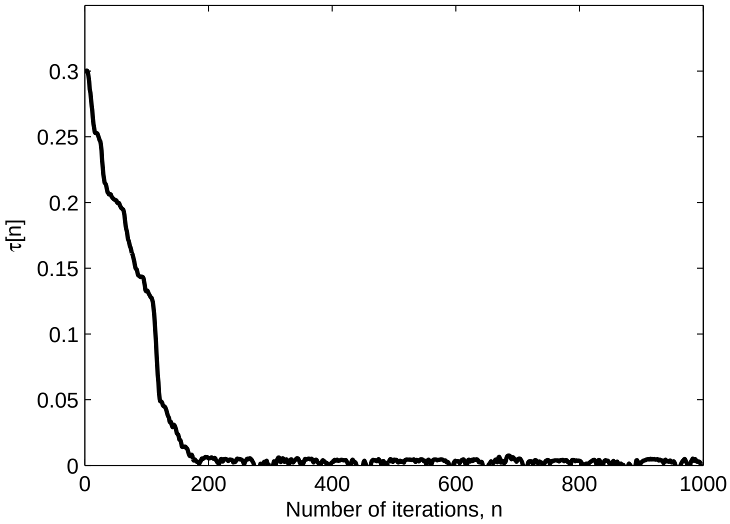

Figure 10.8: A plot of the timing phase update of the early-late gate timing recovery algorithm. The parameters used are : $\beta = 0.9$, $\mu_0 = 0.005$, and $\delta\tau = 12$.

10.2.2 Gradient-based algorithm

Using the gradient algorithm, the following recursion may be used to find the timing phase τ that maximizes the timing recovery cost function $\rho(\tau)$:

$$\tau[n+1] = \tau[n] + \mu \frac{\partial \rho(\tau)}{\partial \tau} \tag{10.31}$$

where μ is an step-size parameter.

The early-late gate timing recovery recursion (10.26) (and, thus, (10.27)) may also be thought as a gradient-based algorithm, where the approximation

$$\frac{\partial \rho(\tau)}{\partial \tau} \approx \frac{|y\left(\tau[n] + \delta\tau + nT_b\right)|^2 - |y\left(\tau[n] - \delta\tau + nT_b\right)|^2}{2\delta\tau} \tag{10.32}$$

is used in (10.31) and the ratio $\mu/(2\delta\tau)$ is redefined as the step size parameter. We also note that in the early-late gate timing recovery algorithm, each iteration requires computation of three samples of $y(t)$; the desired sample $y(nT_b + \tau)$ and the lagged samples $y\left(\tau[n] + \delta\tau + nT_b\right)$ and $y\left(\tau[n] - \delta\tau + nT_b\right)$, with $|\delta\tau| < T_b/4$. Here, we present a lower complexity timing recovery algorithm that operates based on only two samples $y(\tau[n] + nT_b)$ and $y(\tau[n] + nT_b + T_b/2)$ for each update of $\tau[n]$.

We begin with using (10.12) to obtain

$$\frac{\partial \rho(\tau)}{\partial \tau} = -\frac{4\pi}{T_b}|\rho_1| \sin\left(\frac{2\pi}{T_b}\tau + \angle\rho_1\right). \tag{10.33}$$

Also, it turns out that when β is close to, but smaller than, one,

$$E\left[\Re(y_0(\tau[n]+nT_b)y_1^*(\tau[n]+nT_b+T_b/2))\right] = -k\sin\left(\frac{2\pi}{T_b}\tau+\angle\rho_1\right) \tag{10.34}$$

where $y_0(\tau[n]+nT_b)$ and $y_1(\tau[n]+nT_b+T_b/2)$ signal sequences obtained by passing $y(\tau[n]+nT_b)$ and $y\left(\tau[n]+nT_b+T_b/2\right)$ through the transfer function $B(z)$ of (10.25), respectively, and k is a positive constant. Following the same approach as the one used in the LMS algorithm and, also, in (10.27), we use $\Re(y_0(\tau[n]+nT_b)y_1^*(\tau[n]+nT_b+T_b/2))$ as a stochastic estimate proportional to the gradient $\partial\rho(\tau)/\partial\tau$ in (10.31). This leads to the update equation

$$\tau[n+1] = \tau[n] + \mu\Re(y_0(\tau[n]+nT_b)y_1^*(\tau[n]+nT_b+T_b/2)). \tag{10.35}$$

The above algorithm can be implemented in software through a simple modification of the 'for loop' in the MATLAB script 'TR_ELG.m'. The modified 'for loop' may be written as:

Modified timing recovery loop for the realization of the recursion (10.35).

```
for k=start:L:length(tau)*L
    tauTb=round(tau(kk)*L);
    y0=sqrt(1-beta^2)*y(k+tauTb)-beta*y0;
    y1=sqrt(1-beta^2)*y(k+tauTb+L/2)-beta*y1;
    tau(kk+1)=tau(kk)+mu*real(y0*y1'); kk=kk+1;
end
```

Figure 10.9 presents a typical plot of $\tau[n]$ when the gradient-based algorithm is used. The simulation setup is similar to the one used to generate Figures 10.7 and 10.8 and the parameters used, here, are $\beta = 0.95$ and $\mu = 0.01$. From this result, we may observe that the gradient-based timing recovery algorithm proposed here is somewhat slower than the early-late gate algorithm proposed earlier. It also exhibits a higher level of timing jitter; compare Figures 10.8 and 10.9.

10.2.3 Tone extraction algorithm

The identity $\rho(t) = E[|y(t)|^2] = \rho_0 + 2|\rho_1|\cos(2\pi t/T_b + \angle\rho_1)$, i.e., (10.11), implies that $|y(t)|^2$ has a spectral line at $f_b = 1/T_b$. Moreover, from the results developed in this chapter, we infer that the optimum timing phase coincides with the maxima of $\rho(t)$. Following this observation, one may suggest the block diagram shown in Figure 10.10, for finding the optimum timing phase. In this figure $H(z)$ is a narrow-band filter tuned to the frequency $1/T_b$. Variety of techniques are available and may be used for realization of $H(z)$. Here, we introduce one realization that has some similarity (in structure) to the CIC filters that where introduced in Section 5.7.

Let us assume that the samples of $y[n]$ are available at a rate of $f_s = L/T_b$. Figure 10.11 presents an example of the power spectral density of $|y[n]|^2$ for the

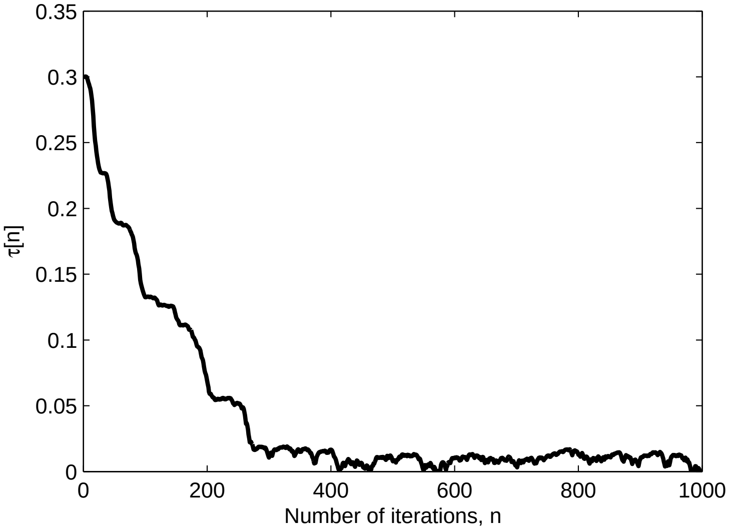

Figure 10.9: A plot of the timing phase update of the gradient-based timing recovery algorithm. The parameters used are : $\beta = 0.95$ and $\mu = 0.01$.

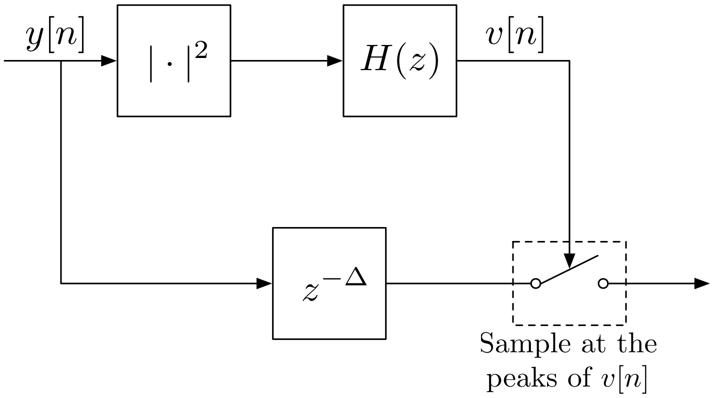

Figure 10.10: Block diagram of a timing recovery system based on tone extraction algorithm.

case where $L = 10$. Note that the frequency axis is normalized with respect to the sampling frequency f_s. As one would expect, there are three spectral lines; a strong spectral line at $f = 0$ and a pair of weaker spectral lines at $f = \pm\frac{1}{L} = \pm 0.1$. Our goal is to extract the latter pair.

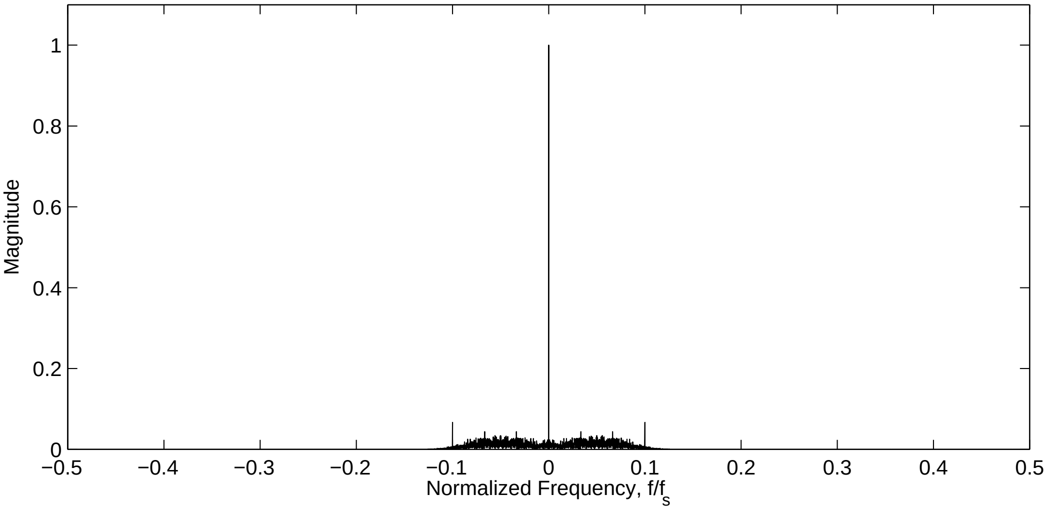

Figure 10.11: An example of the spectra of $|y[n]|^2$ for the case where $L = 10$.

To develop a low complexity structure for extracting the spectral lines at $f = \pm\frac{1}{L}$, we proceed as follows. First, we design a narrowband filter $G(z)$ whose passband is located at the normalized frequency $f = 0.5$. Then, each delay in $G(z)$ is replaced by $L/2$ delays. This results in a comb filter that one of its passbands selects the desired spectral lines at $f = \pm\frac{1}{L}$. A particular choice of $G(z)$ that results in a very low cost implementation is

$$\begin{aligned} G(z) &= 1 - z^{-1} + z^{-2} - \cdots - z^{-(K-1)} \\ &= \frac{1 - z^{-K}}{1 - z^{-1}} \end{aligned} \tag{10.36}$$

where K is the filter length. Here, we are interested in even values of K since such choices result in a null at $f = 0$ and thus removes the undesirable DC component ρ_0. Figure 10.12 presents an example of the power spectra density of the signal resulting from passing $|y[n]|^2$ through $G(z^{L/2})$ for the case where $L = 10$ and $K = 20$. Clearly, this filter has removed the DC component of the signal and also has greatly suppressed the background noise, around $f = 0$, that is seen in Figure 10.11.

Figure 10.13(a) presents the block diagram of a timing recovery system that uses $G(z)$ for tone extraction and accordingly for adjustment of the timing phase of a receiver. This system operates based on the decimated output samples $p[n]$

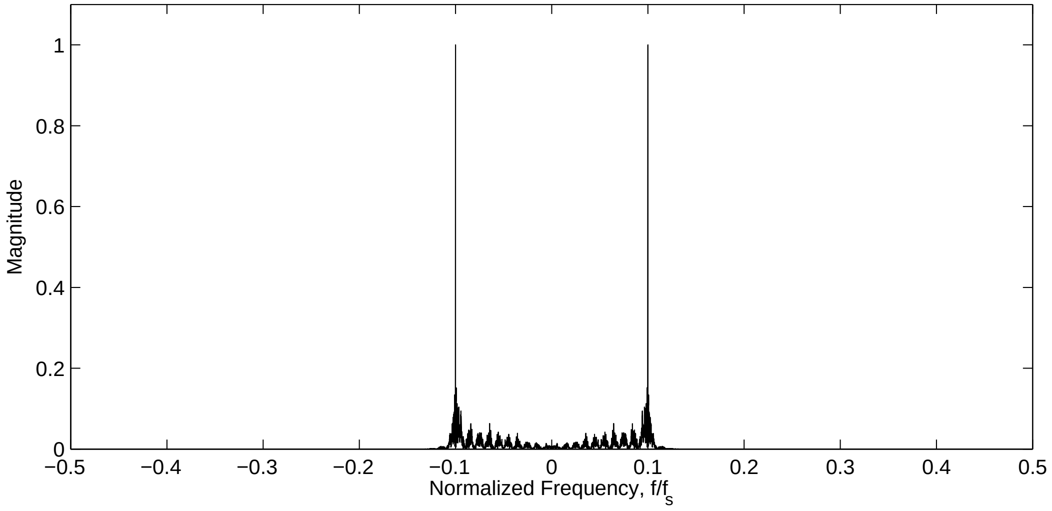

Figure 10.12: An example of the spectra of the signal resulting from passing $|y[n]|^2$ through $G(z^{L/2})$ for the case where $L = 10$ and $K = 20$.

and adjusts the time delay δ such that the extracted output samples coincide with negative-crossings of $p[n]$. This corresponds to the midpoint between maxima and minima points of the cost function $\rho(t)$. Hence, the timing phase that coincides with the peak point of signal power will be $\Delta = \delta + \frac{N}{4}$. To adjust δ, the following update equation may be used

$$\delta_c[n+1] = \delta_c[n] - \mu p[n] \tag{10.37}$$

where μ is a step-size parameter and δ_c is a continuous-time variable. The integer delay δ is obtained through rounding of δ_c, i.e., $\delta[n] = \text{round}(\delta_c[n])$.

We also note that the block diagram of Figure 10.13(a) can be simplified by using the first noble identity (of Chapter 5) to move part of the decimator ↓ L before $G(z^{L/2})$. This simplification is presented in Figure 10.13(b).

10.3 Data Aided Timing Recovery Methods

The timing recovery algorithms that have been developed so far operate based on the statical properties of the received signal samples. In this section, we present two alternative algorithms that make use of the detected data for tracking the timing phase. These algorithms performs well only when the receiver components, including the equalizer, and carrier and timing recovery blocks have converged and thus the data symbols are detected correctly with a low probability of error.

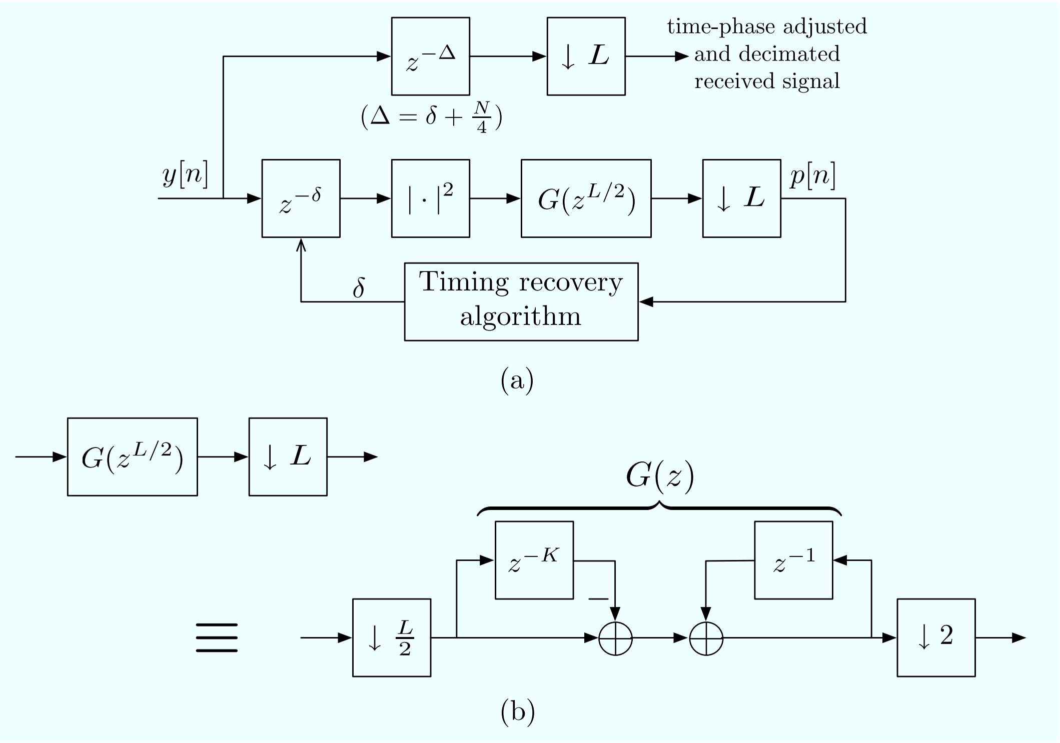

Figure 10.13: Block diagram of a timing recovery system based on tone extraction. (a) The complete system. (b) Detail of the comb filter and decimator blocks.

10.3.1 Muller and Muller's method

In the Muller and Muller's method, the goal is set to minimize a cost function which is a linear sum of the channel impulse response samples taken at T_b intervals. One particular choice of the cost function that is commonly used in practice is

$$\eta(\tau) = |c_{\mathrm{BB}}(\tau + T_b) - c_{\mathrm{BB}}(\tau - T_b)|. \tag{10.38}$$

Figure 10.14 presents an example of the channel impulse response $c_{\mathrm{BB}}(t)$ and its relevant samples. Since the optimum timing phase, $\tau = \tau_{\mathrm{opt}}$, minimizes the cost function $\eta(\tau)$, this corresponds to the case where $c_{\mathrm{BB}}(\tau+T_b) \approx c_{\mathrm{BB}}(\tau-T_b)$. It is also intuitively understood (by visual inspection of Figure 10.14) that for an equalized channel this condition positions $c(\tau)$ to around the peak of $c_{\mathrm{BB}}(t)$ and the rest of the samples near the zero-crossing points of $c_{\mathrm{BB}}(t)$. This, of course, is desirable as it leads to a situation where the sampled signal suffers from a small level of intersymbol interference.

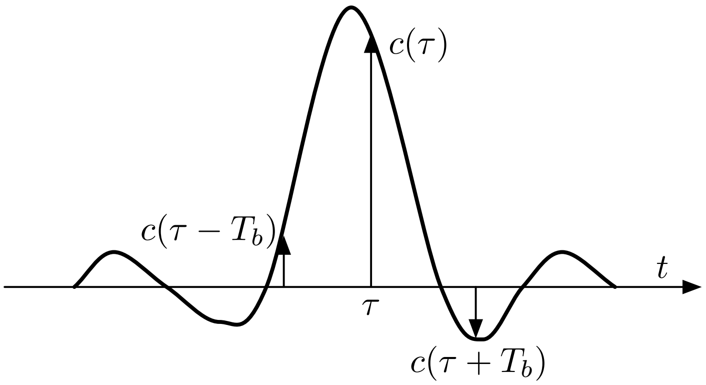

Figure 10.14: A typical impulse response, $c_{\mathrm{BB}}(t)$, of a channel and a set of relevant samples pertinent to the Muller and Muller's timing recovery method.

In a more general case, when $c_{\mathrm{BB}}(t)$ is a complex function of time, the real part of $c_{\mathrm{BB}}(t)$ is a pulse similar to the one presented in Figure 10.14 and its imaginary part is a low amplitude signal (ideally, zero). In that case, the cost function (10.38) is modified as

$$\eta(\tau) = |\Re\{c_{\mathrm{BB}}(\tau + T_b) - c_{\mathrm{BB}}(\tau - T_b)\}|. \tag{10.39}$$

To develop an adaptive algorithm for minimization of $\eta(\tau)$, we first note that for $\tau > \tau_{\mathrm{opt}}$, $\Re\{c_{\mathrm{BB}}(\tau + T_b) - c_{\mathrm{BB}}(\tau - T_b)\} < 0$, and for $\tau < \tau_{\mathrm{opt}}$, $\Re\{c_{\mathrm{BB}}(\tau + T_b) - c_{\mathrm{BB}}(\tau - T_b)\} > 0$. Hence, we suggest the update equation

$$\tau[n+1] = \tau[n] + \mu\Re\{\hat{c}_{\mathrm{BB}}(\tau + T_b) - \hat{c}_{\mathrm{BB}}(\tau - T_b)\} \tag{10.40}$$

where $\hat{c}_{\mathrm{BB}}(\tau+T_b)$ and $\hat{c}_{\mathrm{BB}}(\tau-T_b)$ are the estimates of $c_{\mathrm{BB}}(\tau+T_b)$ and $c_{\mathrm{BB}}(\tau-T_b)$, respectively. On the other hand, we recall that

$$y(t) = \sum_{k=-\infty}^{\infty} s[k]c_{\mathrm{BB}}(t-kT_b) \tag{10.41}$$

where $s[k]$ is the sequence of the transmitted data symbols. Using (10.41) and assuming that

$$E[s[n]s^*[m]] = \begin{cases} 1, & m = n \\ 0, & m \neq n \end{cases} \tag{10.42}$$

one finds that

$$E[y(nT_b+\tau)s^*[n-1]] = c_{\mathrm{BB}}(\tau+T_b) \tag{10.43}$$

and

$$E[y((n-1)T_b+\tau)s^*[n]] = c_{\mathrm{BB}}(\tau-T_b). \tag{10.44}$$

Finally, following the philosophy of the LMS algorithm, we use the coarse estimates $\hat{c}_{\mathrm{BB}}(\tau+T_b) = y(nT_b+\tau)\hat{s}^*[n-1]$ and $\hat{c}_{\mathrm{BB}}(\tau-T_b) = y((n-1)T_b+\tau)\hat{s}^*[n]$, where $\hat{s}[n-1]$ and $\hat{s}[n]$ are the detected values of $s[n-1]$ and $s[n]$, respectively, in (10.40) to obtain the update equation

$$\tau[n+1] = \tau[n] + \mu\Re\left\{y(nT_b+\tau[n])\hat{s}^*[n-1] - y((n-1)T_b+\tau[n])\hat{s}^*[n]\right\}. \tag{10.45}$$

The MATLAB script 'TR_MM.m' on the accompanying CD allows the reader to examine the performance of the Muller and Muller's method for various QAM modulation sizes. The beginning part of this code is similar to 'TR_ELG.m'. The timing recovery part of the codes is replaced by the lines shown in the script presented in the next page.

Muller and Muller's timing recovery method.

```
mu=0.01; Ly=length(y); kk=1; yp=0; ym=0; start=5*L+1;
tau=0.3*ones(1,floor((Ly-start)/L)); x=tau;
for k=start:L:length(tau)*L-L
    tauTb=round(tau(kk)*L);
    sk=sign(real(y(k+tauTb)))+i*sign(imag(y(k+tauTb)));
    skm1=sign(real(y(k+tauTb-L)))+i*sign(imag(y(k+tauTb-L)));
    tau(kk+1)=tau(kk)+mu*real(y(k+tauTb)*skm1'-y(k+tauTb-L)*sk');
    kk=kk+1;
end
figure, plot(tau(1:kk-1))
xlabel('Iteration Number, n'), ylabel('tau[n]')
```

Figure 10.15 presents an example of the learning curve of the Muller and Muller's method. The parameters use are $M = 4$ and $\mu = 0.01$. As seen, the Muller and

Muller's method has relatively fast convergence and a very low jitter after convergence. There will be some performance degradation as the constellation size M increases. However, still compared to other methods that have been discussed, so far, in this chapter, the Muller and Muller's method, usually, has a better performance. However, it has the limitation that can only operate with carrier compensated and equalized signals. Detail examination of the various timing recovery algorithms that have been introduced in this chapter are left as exercise problems at the end of the chapter.

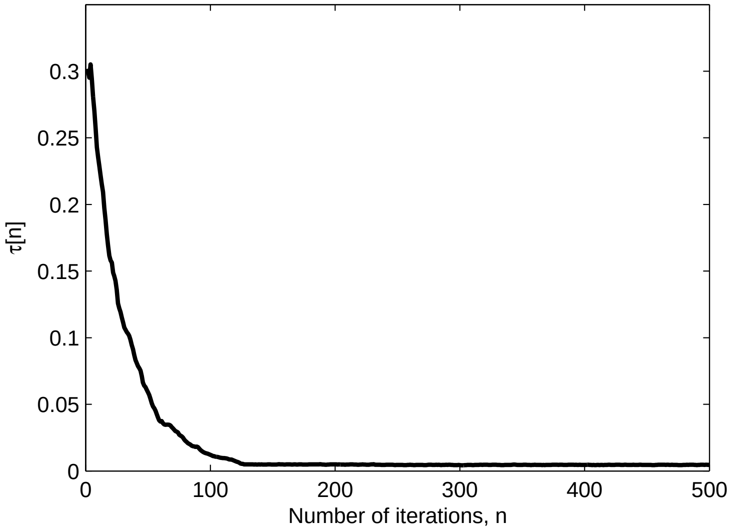

Figure 10.15: A plot of timing phase update of the Muller and Muller's timing recovery method .

10.3.2 Decision directed method

In decision directed method, the timing phase τ is adjust such that the mean-square error (MSE)

$$\xi = E[|e[n]|^2], \tag{10.46}$$

is minimized. In (10.46), $e[n] = s[n] - y(nT_b + \tau)$ where $s[n]$ is the transmitted data symbol. In practice, where the detected symbols $\hat{s}[n]$, with a high probability, are equal to the transmitted symbols, $s[n]$ is replaced by $\hat{s}[n]$. Also, following the same philosophy as the one used in the development of the LMS algorithm, we use the noisy estimate $\hat{\xi} = |e[n]|^2 = e[n]e^*[n]$ of ξ and note that

$$\frac{\partial \hat{\xi}}{\partial \tau} = e[n]\frac{\partial e^*[n]}{\partial \tau} + e^*[n]\frac{\partial e[n]}{\partial \tau}$$

$$
\begin{aligned}
&= 2\Re\left\{e^*[n]\frac{\partial e[n]}{\partial \tau}\right\} \\
&= -2\Re\left\{e^*[n]\frac{\partial y(nT_b+\tau)}{\partial \tau}\right\}. \qquad (10.47)
\end{aligned}
$$

Moreover, we use the approximation

$$\frac{\partial y[n]}{\partial \tau} = \frac{y(nT_b+\tau+\delta\tau) - y(nT_b+\tau-\delta\tau)}{2\delta\tau}. \qquad (10.48)$$

Substituting (10.48) in (10.47), the result in the update equation

$$\tau[n+1] = \tau[n] - \mu\frac{\partial\hat{\xi}}{\partial\tau} \qquad (10.49)$$

and redefining $\mu/\delta\tau$ as a new step-size μ, we obtain

$$\tau[n+1] = \tau[n] + \mu\Re\left\{e^*[n](y(nT_b+\tau[n]+\delta\tau) - y(nT_b+\tau[n]-\delta\tau))\right\}. \qquad (10.50)$$

The MATLAB script ’TR_DD.m’ on the accompanying CD allows the reader to examine the performance of the decision directed method for various QAM modulation sizes. The beginning part of this code is similar to ’TR_ELG.m’. The timing recovery part of the codes is replaced by the following lines. Here, for brevity, we have assumed that the transmitted symbols belong to a QPSK constellation, i.e., QAM with $M = 4$.

Decision directed timing recovery method.

```
mu=0.05; Ly=length(y); kk=1; start=5*L+1;
tau=0.3*ones(1,floor((Ly-start)/L));
for k=start:L:length(tau)*L-L
    tauTb=round(tau(kk)*L);
    sk=sign(real(y(k+tauTb)))+i*sign(imag(y(k+tauTb)));
    skm1=sign(real(y(k+tauTb-L)))+i*sign(imag(y(k+tauTb-L)));
    tau(kk+1)=tau(kk)+mu*real((sk-y(k+tauTb))*(y(k+tauTb+dtau)-y(k+tauTb-dtau))');
    kk=kk+1;
end
figure, plot(tau(1:kk-1))
xlabel('Iteration Number, n'), ylabel('tau[n]')
```

Figure 10.16 presents an example of the learning curve of the decision directed timing recovery method, when $\mu = 0.05$. As seen, this learning curve is similar to its counterpart from the Muller and Muller’s method. Further study of this algorithm that is left as exercises for the reader, reveals that this observation is general an applicable to all constellation sizes.

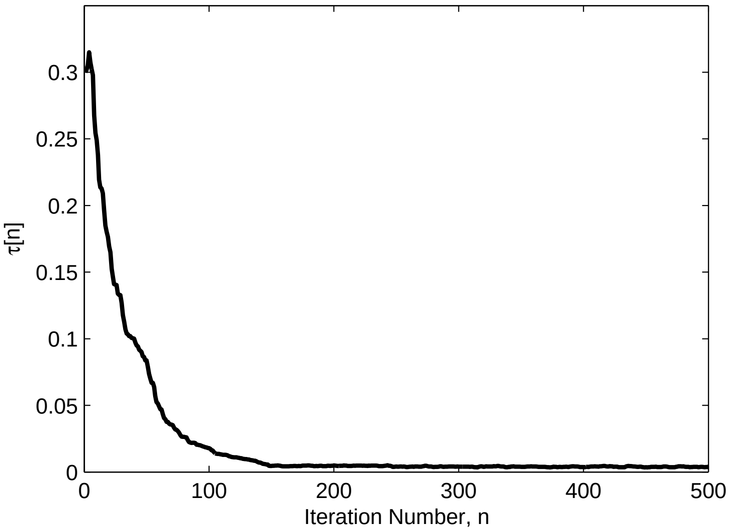

Figure 10.16: A plot of timing phase update of the decisiondirected timing recovery method .

Problems

1. In this problem, we introduce a numerical method of evaluating the timing recovery cost function $\rho(\tau)$ from the samples of the equivalent baseband impulse response $c_{\rm BB}(t)$.

   Substituting $t = nT_b + \tau$ in (10.41), we obtain

$$y(nT_b+\tau) = \sum_{n=-\infty}^{\infty} s[n]c_{\rm BB}((n-k)T_b+\tau). \tag{10.51}$$

   (a) Using (10.51), and assuming that the data symbols $s[n]$ are independent of one another and $E[|s[n]|^2] = \sigma_s^2$, show that

$$\rho(\tau) = E[|y(nT_b+\tau)|^2] = \sigma_s^2 \sum_{n=-\infty}^{\infty} |c_{\rm BB}(nT_b+\tau)|^2.$$

   (b) By letting $c_{\rm BB}(t) = p_{\rm rc}(t)$, where $p_{\rm rc}(t)$ is a raised-cosine pulse-shape with the role-off factor α, examine $\rho(\tau)$ for $0 \leq \tau \leq 4T_b$ and the choices of $\alpha = 0.25$, 0.5 and 1. Plot the results for $\sigma_s^2 = 1$ and compare them with those in Figure 10.4. Comment on your observation.

   (c) For $\alpha = 0.5$, obtain and present plots of $\rho(\tau)$ when the link between the transmitter and receiver is a multipath channel with the impulse response:

i. $c(t) = 0.3\delta(t - 0.5T_b) + \delta(t - T_b) - 0.2\delta(t - 2.3T_b)$.

ii. $c(t) = 0.45\delta(t - 0.7T_b) + \delta(t - T_b) - 0.53\delta(t - 2.7T_b)$.

2. A naive pulse-shape that satisfies the Nyquist condition is $p(t) = p_{\mathrm{T}}(t) \star p_{\mathrm{R}}(t) = \Lambda(t/T_b)$; see Chapter 2 for the definition of $\Lambda(t/T_b)$. For this choice of $p(t)$ and an ideal channel $c(t) = \delta(t)$, consider the received signal

$$y(t) = \sum_{n=-\infty}^{\infty} s[n]p(t - nT_b).$$

(a) Assuming that the data symbols $s[n]$ are independent of one another and $E[|s[n]|^2] = \sigma_s^2$, evaluate and obtain an expression for

$$\rho(\tau) = E[|y(nT_b + \tau)|^2]$$

for $0 \leq \tau \leq T_b$.

(b) Present a plot of $\rho(\tau)$. You should find that in contrast to the fundamental results presented in Section 10.1.1, where $\rho(\tau)$ was a biased sine-wave, here, $\rho(\tau)$ has a different form. Explain, what the source of this discrepancy is.

(c) What is the value of τ that maximizes $\rho(\tau)$. For this choice of τ, find samples of $y(t)$ at the sampling times $nT_b + \tau$ and show such choice results in zero ISI. It, thus, is the optimum timing phase.

3. The plots presented in Figure 10.3 can be produced in two ways: (i) direct evaluation of (10.18), and (ii) by taking the DFT of T_b-spaced samples of $p(t)$ starting with different timing phases. Develop a MATAB program for generation of Figure 10.3 through these methods and confirm that both methods give the same results.

4. Recall that the results presented in Figure 10.7 are for the case where data symbols are from a 4-QAM constellation. Using the MATLAB script 'TR_ELG.m' on the accompanying CD:

   Generate and present a set of plots similar to those in Figure 10.7. Note: you should use the same received signal and run the timing recovery loop for different choices of $\delta\tau$. By running the experiment a few times, comment on variation of the learning curves (plots of $\tau[n]$) as data symbols vary, but still are from a 4-QAM constellation.

5. By running the MATLAB script 'TR_ELG.m' for the cases where data symbols are from 16-QAM, 64-QAM, and 256-QAM check whether the early-late gate timing recovery algorithm works for these constellations as well. You may note that as the constellation size increases, to reduce the jitter of $\tau[n]$,

after convergence, you need to reduce the step-size parameter μ proportional to the inverse of the received signal power. Add such a step normalization to 'TR_ELG.m' and examine the amended program. By presenting proper results, comment on your observation.

Hint: One possible method of step normalization is to evaluate the variance of the received signal ('y' in the script 'TR_ELG.m'), say, σ_y^2, and set $\mu = \mu_1/\sigma_y^2$, where μ_1 is a fixed step-size value.

6. Repeat Problem 4 for the case of the modified early-late gate timing recovery algorithm, i.e., reproduce Figure 10.8.

7. Repeat Problem 5 for the case of the modified early-late gate timing recovery algorithm. Set $\beta = 0.9$ for all the experiments that you perform.

8. Run and examine the results of the MATLAB script 'TR_ELG.m' for the following paramters. For each case comment on your observation. Also, compare the results in Parts (a) and (b) and comment.

   (a) Constellation size $M = 4$, three values of $\mu = 0.01$, 0.005 and 0.002, four values of $\beta = 0$, 0.8. 0.9, and 0.95, and the roll-off factor $\alpha = 0.5$.

   (b) Constellation size $M = 4$, three values of $\mu = 0.01$, 0.005 and 0.002, four values of $\beta = 0$, 0.8. 0.9, and 0.95, and the roll-off factor $\alpha = 0.25$.

9. This problem attempts to provide an in depth understanding of the gradient based timing recovery algorithm that was introduced in Section 10.2.2.

   Following the discussion in Section 10.2.2, one may draw the block diagram shown in Figure 10.17(a) for generation of the sampled signals $y_0(\tau + nT_b)$ and $y_1(\tau + nT_b + T_b/2)$. Note that since both input and outputs are discrete-time signals, the channel is also replaced by a discrete-time equivalent. Accordingly, $H(z,\tau)$ and $H(z,\tau + T_b/2)$ are the z-transforms of the equivalent channel impulse responses sampled at the time instants $\tau + nT_b$ and $\tau + nT_b + T_b/2$, respectively.

   (a) Following similar equations to those in Section 10.1.3, and replacing $P(\cdot)$ by $H(\cdot)$, show that

   $$H(e^{j2\pi f},\tau) = \frac{1}{T_b}\left(H\left(\frac{f}{T_b}\right) + H\left(\frac{f}{T_b} - \frac{1}{T_b}\right)e^{-j2\pi\tau/T_b}\right) \tag{10.52}$$

   where f is the frequency normalized to the baud rate $f_b = 1/T_b$. Note that while the function $H(\cdot)$ on the left-hand side of the above equation refers to the transfer function of the discrete-time channel, the function $H(\cdot)$ on the right-hand side refers to the transfer function of the continuous-time channel.

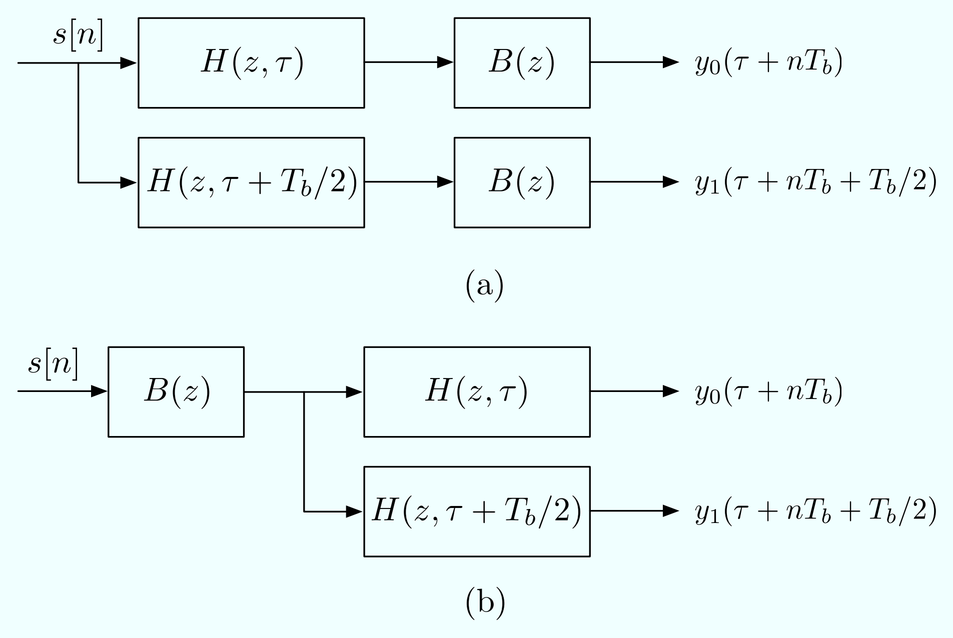

Figure 10.17: A plot of timing phase update of the decisiondirected timing recovery method .

(b) The goal of the timing recovery algorithm is to find the value of τ that maximizes $|H(e^{j2\pi f}, \tau)|^2$ at the normalized frequency $f = 0.5$, i.e., the goal is to maximize $\rho(\tau) = |H(e^{j\pi}, \tau)|^2$. Show that

$$H(e^{j\pi}, \tau) = \frac{1}{T_b}\left(H\left(\frac{1}{2T_b}\right) + H\left(-\frac{1}{2T_b}\right) e^{-j2\pi\tau/T_b} \right) \tag{10.53}$$

(c) Let $H(\frac{1}{2T_b}) = H_+ e^{j\theta_+}$ and $H(-\frac{1}{2T_b}) = H_- e^{j\theta_-}$, and show that

$$\rho(\tau) = H_+^2 + H_-^2 + 2H_+H_- \cos\left(\frac{2\pi\tau}{T_b} - (\theta_+ - \theta_-)\right). \tag{10.54}$$

(d) Using the result of Part (c), show that value of τ that maximizes $\rho(\tau)$ is

$$\tau_{\text{opt}} = \frac{\theta_+ - \theta_-}{2\pi} T_b. \tag{10.55}$$

(e) Show that the timing phase $\tau = \tau_{\text{opt}}$ is the one that phase-align the two terms on the right-hand side of (10.53).

(f) Show that

$$\frac{\partial \rho(\tau)}{\partial \tau} = -\frac{4\pi}{T_b} H_+H_- \sin\left(\frac{2\pi\tau}{T_b} - (\theta_+ - \theta_-)\right). \tag{10.56}$$

(g) Show that

$$\Re\left\{H(e^{j\pi},\tau)H^*(e^{j\pi},\tau+T_b/2)\right\} = -2H_+H_-\sin\left(\frac{2\pi\tau}{T_b}-(\theta_+-\theta_-)\right) \tag{10.57}$$

where $\Re\{\cdot\}$ denotes the real part of.

(h) Next, we note that the block diagram shown in Figure 10.17(a) can be arranged as in Figure 10.17(b). Also, assuming that $B(z)$ is a narrowband filter, its output will be a narrowband signal centered around the center frequency of the passband of $B(z)$. To get better understanding of this let $s[n]$ be a random binary sequence. Develop a MATAB code that takes $s[n]$ and pass it through the filter $B(z)$ defined in (10.25). Observe the output of the filter for values of $\beta = 0.95$, 0.99 and 0.999. You should observe that the output of $B(z)$ is a sequence of the form $+a$, $-a$, $+a$, $-a$, $\cdots$, where a is a slowly time-varing number whose rate of change is determined by the size of β. Comment on the relationship between the size of β and the rate of variation of a.

(i) Let us ignore the variation of the a, assume $a = 1$, and, thus, consider the case where the input to the filters $H(z,\tau)$ and $H(z,\tau+T_b/2)$ is the sequence $+1$, -1, $+1$, -1, $\cdots$. This sequence may be interpreted as sine-wave with the amplitude 1 and the normalized frequency $f = 0.5$. By giving clear explanation, show that when such sequence is passed through the channels $H(z,\tau)$ and $H(z,\tau+T_b/2)$, the resulting outputs are, respectively,

$$y_0(\tau+nT_b) = (-1)^n H(e^{j\pi},\tau)$$

and

$$y_1(\tau+nT_b+T_b/2) = (-1)^n H(e^{j\pi},\tau+T_b/2).$$

(j) Combining the results of Parts (f), (g) and (i), obtain the update equation (10.35).

10. Run and examine the results of the MATLAB script 'TR_GB.m', on the accompanying CD, for the following paramters. For each case comment on your observation.

    (a) Constellation size $M = 4$, two values of $\mu = 0.01$ and 0.005, four values of $\beta = 0.7$, 0.9, 0.95 and 0.99, and the roll-off factor $\alpha = 0.5$.

    (b) Constellation size $M = 4$, two values of $\mu = 0.01$ and 0.005, four values of $\beta = 0.7$, 0.9, 0.95 and 0.99, and the roll-off factor $\alpha = 0.25$.

11. By running the MATLAB script 'TR_GB.m' for the cases where data symbols are from 16-QAM, 64-QAM, and 256-QAM constellation check whether the

gradient based timing recovery algorithm works for these constellations as well. You may note that as the constellation size increases, to reduce the jitter of $\tau[n]$, after convergence, you need to reduce the step-size parameter μ proportional to the inverse of the received signal power. Add such a step normalization to 'TR_GB.m' and examine the amended program. The decision to which values of the parameters *alpha*, β and μ to use is left to you. By presenting proper results, comment on your observation.

12. Tone extraction

13. M & M

14. Decision directed