

4 Timing Synchronization

The timing problem is easily understood: The receiver samples the incoming signal with the known sampling rate $1/T_s$, but how does it know exactly *when* to sample? It does not know this, it can only sample the signal with a random time offset, and it is the task of the time synchronization unit to estimate this time offset and correct it.

4.1 Effect of a Timing Offset

The effect of the timing offset can be modelled by including the filter $\delta(t - \varepsilon T)$ into the effective channel. The parameter ε denotes the time offset, normalized by the symbol duration T . The matched filter output then becomes

$$z(kT_s) = \sum_{n=-\infty}^{\infty} a[n]g_{\text{RC}}(kT_s - nT - \varepsilon T) + w'(kT_s). \quad (4.1)$$

Take a look at Figure 4.1, where several RC pulses are plotted. Imagine that we want to detect the data symbol $a[0]$. If we sample the signal exactly at $t = 0$, all pulses except of the middle one are zero, and the received sample is equal to the data symbol plus noise. Now assume that we have a timing error of, say, $\varepsilon = 0.1$, which means that the signal arrives a bit too late, and hence our receiver samples the signal at $t = -0.1T$. At this time, however, the center pulse has

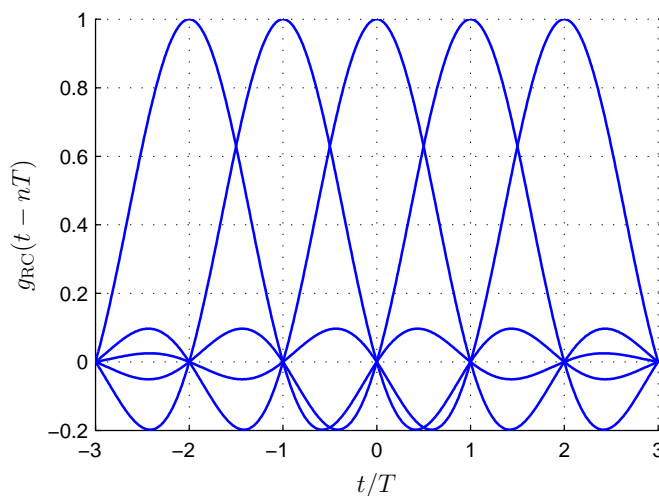


Figure 4.1: Raised cosine pulses with $\alpha = 0.22$

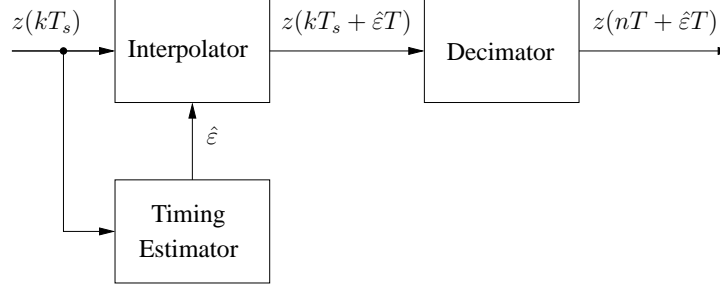


Figure 4.2: Time synchronization unit

not yet reached its maximum amplitude; furthermore, the contributions of the other pulses do not vanish, which leads to inter-symbol interference.

In order to attain ISI-free reception, we must attempt to generate symbols

$$z(nT + \hat{\epsilon}T) = \sum_{n'=-\infty}^{\infty} a[n']g_{\text{RC}}(nT - n'T - \epsilon T + \hat{\epsilon}T) + w'(nT + \hat{\epsilon}T) \quad (4.2)$$

where $\hat{\epsilon}$ is an estimate of the timing error. If $\hat{\epsilon} = \epsilon$, then $z(nT + \hat{\epsilon}T) = a[n] + w'(nT + \hat{\epsilon}T)$ and the data can be optimally recovered.

4.2 Synchronization via Interpolation

In principle, there are two possibilities to generate the desired symbols $z(nT + \hat{\epsilon}T)$. The first possibility is to adjust the time instant of A/D conversion. However, this results in an undesired feedback to the analog part of the system. We will therefore use a different method that works completely in the digital domain.

It is well known that a bandlimited signal is completely described by time-discrete samples if $1/T_s \geq 2f_{\text{max}}$, where f_{max} is the highest frequency that is contained in the signal. Due to the oversampling, this condition is fulfilled in our case. It is thus possible to calculate arbitrary intermediate values from the sample sequence by means of an interpolator.

Figure 4.2 shows a block diagram of the time synchronization unit. The timing estimator calculates an estimated timing offset $\hat{\epsilon}$ using the received samples $z(kT_s)$. The interpolator then generates the time-delayed sample stream $z(kT_s + \hat{\epsilon}T)$ which is then converted to symbol rate by discarding the $L - 1$ intermediate samples between two adjacent data symbols.

The interpolator has a strong impact on the implementation complexity. An ideal interpolator would be implemented as an si-filter, and its impulse response would have to be of infinite duration. Of course, from a practical point of view, the interpolator is only allowed to have a finite number of taps. A brief discussion of practical interpolators is given in Appendix 4.6.

4.3 The Estimation Algorithm

The remaining question is how to obtain the estimate $\hat{\epsilon}$. For the derivation of the estimator we use the sequence $x[k] = x(kT_s)$ which is defined as the instantaneous signal power

$$x(kT_s) = |z(kT_s)|^2. \quad (4.3)$$

Appendix 4.5 shows the following relation: if there is no time offset ($\varepsilon = 0$), then the expectation of $x(kT_s)$ is a real-valued and even signal. Accordingly, its spectrum (i.e., the expectation of $X(f)$) is then real-valued for all frequencies f . Now, if we introduce a nonzero offset εT in $z(kT_s)$ and, accordingly, in $x(kT_s)$, the original real spectrum is multiplied by $e^{-2\pi j \varepsilon T f}$. Evaluating the spectrum at the symbol rate $f = 1/T$ then results in a complex value whose phase is $-2\pi\varepsilon$. We can thus use

$$\hat{\varepsilon} = -\frac{1}{2\pi} \arg\{X(1/T)\} \quad (4.4)$$

as an estimator of the timing offset (see Appendix 4.5 for details).

The spectral component $X(1/T)$ can be calculated by performing a *discrete fourier transform* (DFT). In general, the spectrum $X(f)$ is given as

$$X(f) = \sum_{k=0}^{\infty} x(kT_s) e^{-2\pi j k T_s f}, \quad (4.5)$$

where we have assumed that the signal starts at $t = 0$. The spectral component at $f = 1/T$ is thus

$$\begin{aligned} X(1/T) &= \sum_{k=0}^{\infty} x(kT_s) e^{-2\pi j k \frac{T_s}{T}} \\ &= \sum_{k=0}^{\infty} x[k] e^{-\frac{\pi}{2} j k} \\ &= \sum_{k=0}^{\infty} x[k] (-j)^k. \end{aligned} \quad (4.6)$$

Due to the oversampling factor of $T/T_s = 4$, the implementation of the Fourier transform becomes very simple, as the exponential term only takes on the values $\{1, -j, -1, j\}$.

Of course, Equation (4.6) cannot be implemented in a real system, because we cannot calculate an infinite sum. We can, however, get an estimate $\hat{\varepsilon}[n]$ that is updated every symbol duration by taking all available samples into account:

$$\hat{\varepsilon}[n] = -\frac{1}{2\pi} \arg \left\{ \sum_{k=0}^{4n-1} x[k] (-j)^k \right\}. \quad (4.7)$$

The cumulative sum acts as an averager: During the reception, more samples become available, and the estimate becomes more and more reliable.

It is easily seen that the algorithm (4.7) only works if the true parameter ε is constant during the whole reception, which is the case in this lab. In practice, however, the timing offset will change over time, and the estimation accuracy would be severely degraded if we took all available samples—even old ones that were caused by a different ε than the current—into account.

An important measure here is the *coherence time*, which is the duration over which a parameter can be assumed to be approximately constant. Let the coherence time for the timing offset be denoted as $T_{c;\varepsilon} = N_{c;\varepsilon} T$. Then the estimation algorithm can be implemented as a sliding sum over the $4N_{c;\varepsilon}$ most recently received samples:

$$\hat{\varepsilon}[n] = -\frac{1}{2\pi} \arg \left\{ \sum_{k=4(n-N_{c;\varepsilon})}^{4n-1} x[k] (-j)^k \right\}. \quad (4.8)$$

4.4 Your Tasks

1. Implement the timing synchronization unit and integrate it into the system provided on the Moodle (timingsync.m). At first, write and use a linear interpolator to decode the image (task4.mat).
2. Afterwards, adapt your system to do BER measurements. We provide a symbol sequence (pn_sequence.mat) and the corresponding bitstream (ber_pn_seq.mat) to do this.
3. Plot $\hat{\varepsilon}[n]$ over n . You will notice that the first estimates are very bad, because the averaging is over too few symbols. Can you imagine a way to improve these first estimates? (Think about the frame synchronizer!)
4. Now replace the interpolator by a cubic one (see Appendix 4.6). How big is the performance gain?

4.5 Appendix

In this section, we show that (4.4) is an *asymptotically unbiased* estimator of the timing offset ε . (An estimator is asymptotically unbiased if it becomes unbiased as the number of observations goes to infinity.)

4.5.1 Signal Model

We assume M -PSK signalling, which means that the amplitude of the data symbols $a[n]$ is constant. Without loss of generality, set $|a[n]| = 1$. The symbols are assumed to be uncorrelated, i.e. $E\{a[m]a^*[n]\} = \delta(m - n)$. The received signal after matched filtering is given as

$$z(t) = \sum_{n=-\infty}^{\infty} a[n]g_{\text{RC}}(t - nT - \varepsilon T) + w'(t), \quad (4.9)$$

where $g_{\text{RC}}(t)$ is the combined impulse response of the pulse shaping filter and the matched filter, and is therefore real-valued and even. In our lab, $g_{\text{RC}}(t)$ is a raised cosine pulse. Furthermore, $w'(t)$ is the AWGN with variance σ_w^2 after matched filtering.

For the digital realization, the signal is sampled at four times the symbol rate: $T/T_s = 4$.

4.5.2 Continuous Time Consideration

We start with examining the noise-free signal $\tilde{z}(t) = z(t) - w'(t)$. Let

$$\bar{x}(t) = E\{|\tilde{z}(t)|^2\} \quad (4.10)$$

denote the expected value of the instantaneous signal power. Since the data symbols are uncorrelated, $\bar{x}(t)$ becomes

$$\begin{aligned}
 \bar{x}(t) &= E \left\{ \left| \sum_{n=-\infty}^{\infty} a[n] g_{\text{RC}}(t - nT - \varepsilon T) \right|^2 \right\} \\
 &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} E\{a[m]a^*[n]\} g_{\text{RC}}(t - mT - \varepsilon T) g_{\text{RC}}^*(t - nT - \varepsilon T) \\
 &= \sum_{n=-\infty}^{\infty} |g_{\text{RC}}(t - nT - \varepsilon T)|^2.
 \end{aligned} \tag{4.11}$$

The function $\bar{x}(t)$ is periodic with period T

$$\begin{aligned}
 \bar{x}(t + T) &= \sum_{n=-\infty}^{\infty} |g_{\text{RC}}(t - (n - 1)T - \varepsilon T)|^2 \\
 &= \sum_{n'=-\infty}^{\infty} |g_{\text{RC}}(t - n'T - \varepsilon T)|^2 \quad \text{with } n' = n - 1 \\
 &= \bar{x}(t)
 \end{aligned} \tag{4.12}$$

and can therefore be expanded into a Fourier series

$$\bar{x}(t) = \sum_{k=-\infty}^{\infty} c_k e^{2\pi j k \frac{t}{T}}. \tag{4.13}$$

The Fourier coefficients are given as

$$c_k = \frac{1}{T} \int_0^T \bar{x}(t) e^{-2\pi j k \frac{t}{T}} dt. \tag{4.14}$$

The first Fourier coefficient, i.e. the spectral component $\bar{X}(1/T)$, is

$$\begin{aligned}
 c_1 &= \frac{1}{T} \int_0^T \bar{x}(t) e^{-2\pi j \frac{t}{T}} dt \\
 &= \frac{1}{T} \int_0^T \sum_{n=-\infty}^{\infty} |g_{\text{RC}}(t - nT - \varepsilon T)|^2 e^{-2\pi j \frac{t}{T}} dt \quad \text{substitute } u = t - \varepsilon T \\
 &= \frac{1}{T} \int_{-\varepsilon T}^{T-\varepsilon T} \underbrace{\sum_{n=-\infty}^{\infty} |g_{\text{RC}}(u - nT)|^2}_{\substack{\text{real and even} \\ := A \in \mathbb{R}}} e^{-2\pi j \frac{u}{T}} du e^{-2\pi j \varepsilon} \\
 &= A e^{-2\pi j \varepsilon}.
 \end{aligned} \tag{4.15}$$

Thus, we can calculate the timing offset perfectly from the noise-free and time-continuous received signal:

$$\varepsilon = -\frac{1}{2\pi} \arg\{\bar{X}(1/T)\}. \tag{4.16}$$

4.5.3 Discrete Time Realization

Now we consider the noisy and sampled signal

$$x[k] = |z(kT_s)|^2 \quad (4.17)$$

and show that

$$\hat{\varepsilon} = -\frac{1}{2\pi} \arg\{\hat{X}(1/T)\} \quad (4.18)$$

is an asymptotically unbiased estimate of ε .

We start by calculating the expected value of $x[k]$:

$$\begin{aligned} E\{x[k]\} &= E\{|\tilde{z}(kT_s) + w'(kT_s)|^2\} \\ &= \underbrace{E\{|\tilde{z}(kT_s)|^2\}}_{\bar{x}(kT_s)} + \underbrace{E\{\tilde{z}^*(kT_s)w'(kT_s)\}}_0 + \underbrace{E\{\tilde{z}(kT_s)w'^*(kT_s)\}}_0 + \underbrace{E\{|w'(kT_s)|^2\}}_{\sigma_w^2} \\ &= \bar{x}(kT_s) + \sigma_w^2 \end{aligned} \quad (4.19)$$

with the noise-free signal $\bar{x}(t)$ that we have examined in the previous section.

Assuming that the signal starts at $k = 0$, the corresponding spectrum is

$$E\{X(f)\} = \sum_{k=0}^{\infty} \bar{x}(kT_s) e^{-2\pi j k T_s f} + \frac{\sigma_w^2}{2\pi T_s} \sum_{k=-\infty}^{\infty} \delta(f - k/T_s). \quad (4.20)$$

Evaluating this for $f = 1/T$ yields

$$E\{X(1/T)\} = \sum_{k=0}^{\infty} \bar{x}(kT_s) e^{-2\pi j k/4}. \quad (4.21)$$

If we approximate $E\{X(1/T)\}$ at time $t = nT$ by using the $4N_{c;\varepsilon}$ most recently received samples (the $N_{c;\varepsilon}$ most recently received symbols), we get the estimate

$$E\{\hat{X}(1/T)\} = \sum_{k=4(n-N_{c;\varepsilon})}^{4n-1} \bar{x}(kT_s) e^{-2\pi j k/4}. \quad (4.22)$$

Since $\bar{x}(t)$ is bandlimited to $\pm 2/T$ we can calculate this from $\bar{x}(t)$, rather than from the discrete $\bar{x}(kT_s)$:

$$\begin{aligned} E\{\hat{X}(1/T)\} &= \sum_{k=4(n-N_{c;\varepsilon})}^{4n-1} \bar{x}(kT/4) e^{-2\pi j \frac{k}{4}} \\ &= \frac{4N_{c;\varepsilon}}{T} \int_0^T \bar{x}(t) e^{-2\pi j \frac{k}{4}} dt \\ &= 4N_{c;\varepsilon} c_1, \end{aligned} \quad (4.23)$$

with c_1 calculated in (4.15).

We can now calculate the expected value of $\hat{\varepsilon}$:

$$\begin{aligned}
 E\{\hat{\varepsilon}\} &= -\frac{1}{2\pi} E\{\arg \hat{X}(1/T)\} \\
 &\approx -\frac{1}{2\pi} \arg E\{\hat{X}(1/T)\} \\
 &= -\frac{1}{2\pi} \arg\{4N_{c;\varepsilon}c_1\} \\
 &= \varepsilon.
 \end{aligned} \tag{4.24}$$

Due to the “approximately equal” sign, the estimator is strictly speaking biased. However, if the number of samples is very large, the variance of $\hat{X}(1/T)$ becomes small, and the expectation and argument operations can be exchanged. Our estimator is therefore asymptotically unbiased.

4.6 Appendix

4.6.1 Perfect Interpolator

Let $f(x)$ be a (possibly complex-valued) function of one real-valued parameter. Assume that we only know $f(x)$ for $x \in \mathbb{Z}$, i.e. a series of samples. If f is bandlimited with a maximum frequency of $1/2$, then the function is completely described by the samples. The problem of finding $f(x)$ for $x \notin \mathbb{Z}$ is called interpolation.

The ideal interpolator is a lowpass filter, i.e. the function can be calculated from its samples as

$$f(x) = \sum_{n=-\infty}^{\infty} f(n) \operatorname{si}(\pi(x - n)). \tag{4.25}$$

However, this interpolator is not realizable because it has infinitely many taps.

4.6.2 Linear Interpolator

The simplest possible interpolator is the linear one. Assume without loss of generality that we are interested in the intermediate value $f(x)$ with $x \in (0; 1)$. The linear interpolator then calculates $f(x)$ using the two closest samples $f(0)$ and $f(1)$ according to

$$f(x) = (1 - x)f(0) + xf(1). \tag{4.26}$$

The linear interpolator can be implemented very efficiently, but its accuracy can be problematic. Consider the situation illustrated in Figure 4.3. Given are the function $f(x) = \sin(\frac{\pi}{2}x + \frac{\pi}{4})$ and the samples $f(0) = f(1) = 1/\sqrt{2}$. Now, for all $x \in (0; 1)$, the linear interpolator returns $f(x) = 1/\sqrt{2}$, which is obviously a very bad approximation of the true value if x is around 0.5.

On the other hand, if we wanted to estimate a value with $x \in (-1; 0)$ using the samples $f(-1)$ and $f(0)$, the linear estimation would be quite good.

We see that the linear interpolator yields especially bad results if the function has a local maximum or minimum between the two considered samples.

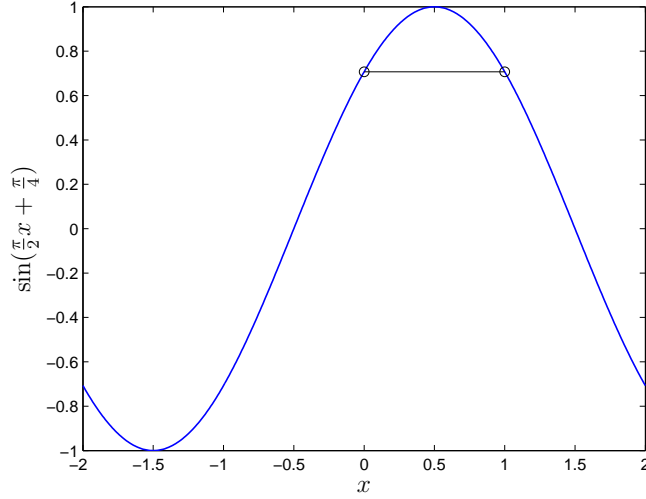


Figure 4.3: Linear interpolator for $f(x) = \sin(\frac{\pi}{2}x + \frac{\pi}{4})$

4.6.3 Cubic Interpolator

The interpolation can be improved by taking one additional sample on each side into account, which leads us to the cubic interpolator. Again we assume that we want to estimate $f(x)$ with $x \in (0; 1)$, this time using the samples $f(-1)$, $f(0)$, $f(1)$ and $f(2)$. The idea is to approximate $f(x)$ with a cubic polynomial $\tilde{f}(x)$ and then evaluate it at the desired point.

In general,

$$\tilde{f}(x) = ax^3 + bx^2 + cx + d, \quad (4.27)$$

where the coefficients a, b, c, d must satisfy

$$\begin{pmatrix} -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 8 & 4 & 2 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} f(-1) \\ f(0) \\ f(1) \\ f(2) \end{pmatrix}. \quad (4.28)$$

With a matrix inversion we get the result

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \frac{1}{6} \begin{pmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -2 & -3 & 6 & -1 \\ 0 & 6 & 0 & 0 \end{pmatrix} \begin{pmatrix} f(-1) \\ f(0) \\ f(1) \\ f(2) \end{pmatrix}. \quad (4.29)$$