Computer simulation of communication systems

Several considerations regarding the computer simulation of communication systems are presented. The following aspects will be discussed:

- Representing continuous-time signals by their samples
- Using computer simulation to determine (an estimate of) error probability and power spectral density
- The accuracy of computer simulations

Some notations and definitions:

- A continuous-time signal and its Fourier transform (FT) will be denoted by a lower-case letter (e.g., $x(t)$) and the corresponding upper-case letter (e.g., $X(f)$), respectively.
- The expectation μ_x and the autocorrelation function $R_x(u)$ of a stationary random process $x(t)$ are defined as $\mu_x = \mathbb{E}[x(t)]$ and $R_x(u) = \mathbb{E}[x(t+u)x^*(t)]$, where $\mathbb{E}[X]$ denotes the expectation (statistical average) of the random variable X. The variance of $x(t)$, which is defined as $\sigma_x^2 = \mathbb{E}[|x(t) - \mu_x|^2]$, is obtained as $\sigma_x^2 = R_x(0) - |\mu_x|^2$. When the stationary process $x(t)$ has zero mean (i.e., $\mu_x = 0$), we obtain $\sigma_x^2 = R_x(0)$.
- The notation δ_k stands for the Kronecker delta function, which is defined as

$$\delta_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

1 The Nyquist/Shannon sampling theorem

Sampling a signal $x(t)$ at a rate $1/T_\mathrm{s}$ yields the signal values $x(kT_\mathrm{s})$; T_s denotes the sampling interval. Obtaining from the samples $x(kT_\mathrm{s})$ (an estimate of) the signal $x(t)$ is referred to as interpolation. In general, $x(t)$ cannot be perfectly reconstructed because the samples $x(kT_\mathrm{s})$ do not contain the full information on $x(t)$; the interpolation becomes more accurate with increasing sampling rate $1/T_\mathrm{s}$. However, the Nyquist/Shannon sampling theorem states that a continuous-time signal $x(t)$ with *finite bandwidth* B (i.e., $X(f) = 0$ for $|f| > B$) can be perfectly reconstructed from the samples $x(kT_\mathrm{s})$, provided that the sampling rate satisfies $1/T_\mathrm{s} \geq 2B$. Under this condition, not only perfect interpolation can be achieved, but also the signals can be processed, stored and simulated based on only their samples.

Let us assume that $X(f) = 0$ for $|f| > B$. We want to reconstruct $x(t)$ from its samples $x(kT_\mathrm{s})$, with $1/T_\mathrm{s} \geq 2B$. To this aim, we construct the signal $x_\mathrm{int}(t)$ using an interpolation pulse $h_\mathrm{int}(t)$, yielding

$$x_\mathrm{int}(t) = \sum_{k=-\infty}^{+\infty} x(kT_\mathrm{s}) h_\mathrm{int}(t - kT_s) \tag{1}$$

We now investigate under which conditions we obtain $x_\mathrm{int}(t) = x(t)$.

The FT of $x_\mathrm{int}(t)$ is given by

$$X_\mathrm{int}(f) = H_\mathrm{int}(f) \left(\sum_{k=-\infty}^{+\infty} x(kT_\mathrm{s}) e^{-j2\pi f T_\mathrm{s}} \right) \tag{2}$$

The expression between parentheses is the discrete-time Fourier transform (DTFT) of the sequence $\{x(kT_\mathrm{s})\}$. This DTFT equals the periodic extension of $X(f)/T_\mathrm{s}$ with period $1/T_\mathrm{s}$ (see section 16):

$$\sum_{k=-\infty}^{+\infty} x(kT_\mathrm{s}) e^{-j2\pi f T_\mathrm{s}} = \frac{1}{T_\mathrm{s}} \sum_{n=-\infty}^{+\infty} X\left(f - \frac{n}{T_\mathrm{s}}\right) \tag{3}$$

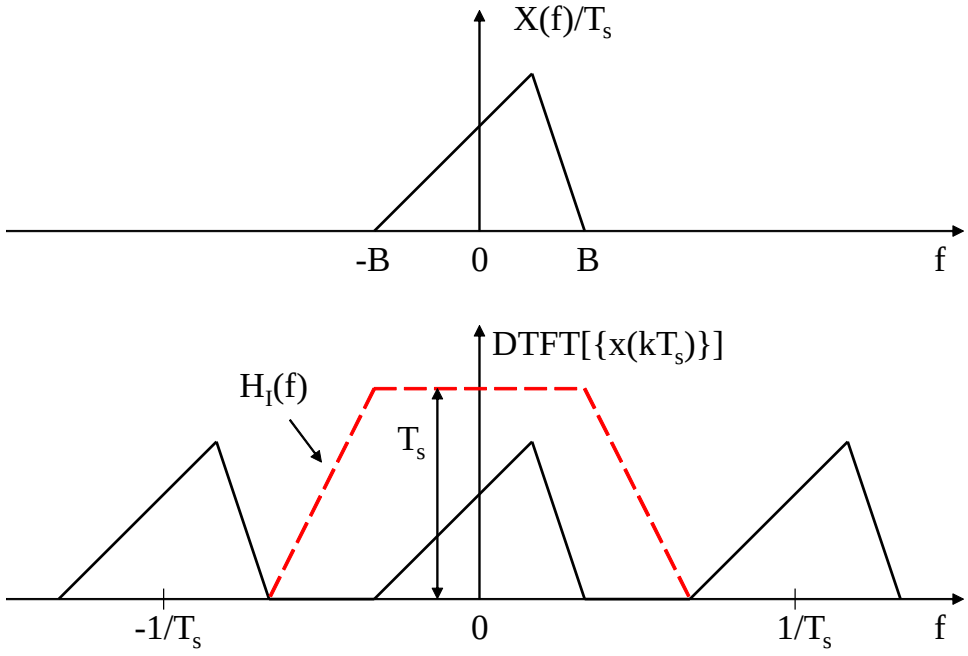

Figure 1: DTFT of $\{x(kT_s\}$

Substituting (3) into (2) yields

$$X_{\text{int}}(f) = \frac{H_{\text{int}}(f)}{T_{\text{s}}} \sum_{n=-\infty}^{+\infty} X\left(f - \frac{n}{T_{\text{s}}}\right) \tag{4}$$

Fig. 1 displays the DTFT of $\{x(kT_{\text{s}})\}$ as the periodic extension of $X(f)/T_{\text{s}}$, assuming $1/T_{\text{s}} \geq 2B$. Note that the different terms $X\left(f - \frac{n}{T_{\text{s}}}\right)$ in (4) do not overlap. Hence, we can achieve $X_{\text{int}}(f) = X(f)$ (or, equivalently, $x_{\text{int}}(t) = x(t)$) by selecting $H_{\text{int}}(f)$ such that

$$H_{\text{int}}(f) = \begin{cases} T_{\text{s}} & |f| < B \\ 0 & |f| > \frac{1}{T_{\text{s}}} - B \\ \text{arbitrary} & \text{elsewhere} \end{cases} \tag{5}$$

When $1/T_{\text{s}} = 2B$, (5) gives rise to only one solution for $H_{\text{int}}(f)$, i.e., $H_{\text{int}}(f) = T_{\text{s}}\text{rect}(fT_{\text{s}})$, where $\text{rect}(u) = 1$ for $|u| < 1/2$ and $\text{rect}(u) = 0$ otherwise. This is also a valid solution when $1/T_{\text{s}} > 2B$, but in this case that solution is not unique. Hence, for $1/T_{\text{s}} \geq 2B$, we have

$$x(t) = \sum_{k=-\infty}^{+\infty} x(kT_{\text{s}}) h_{\text{int}}(t - kT_s) \tag{6}$$

provided that the interpolation pulse satisfies (5). It follows from (6) that $x(t)$ for any t can be reconstructed as a linear combination of the samples $x(kT_{\text{s}})$, by taking the weight of the sample $x(kT_{\text{s}})$ equal to $h_{\text{int}}(t - kT_s)$. Selecting $H_{\text{int}}(f) = T_{\text{s}}\text{rect}(fT_{\text{s}})$, (6) reduces to

$$x(t) = \sum_{k=-\infty}^{+\infty} x(kT_{\text{s}}) \text{sinc}\left(\frac{t - kT_{\text{s}}}{T_{\text{s}}}\right) \tag{7}$$

where $\text{sinc}(u) = \sin(\pi u)/(\pi u)$; note that $\text{sinc}(k) = \delta_k$ for integer k.

When $1/T_{\text{s}} < 2B$, the terms $X\left(f - \frac{n}{T_{\text{s}}}\right)$ in (4) do overlap, yielding frequency-aliasing which inevitably causes $x_{\text{int}}(t)$ from (1) to be different from $x(t)$.

In the following we make use of the sampling theorem in the context of computer simulations to represent, without loss of information, continuous-time signals and operations on these signals.

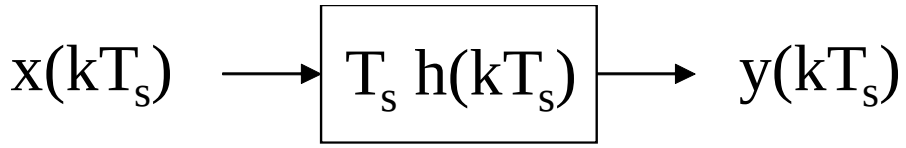

Figure 2: $\{y(kT_\mathrm{s}\}$ is the convolution of $\{x(kT_\mathrm{s})\}$ and $\{T_\mathrm{s}h(kT_\mathrm{s})\}$

2 Filtering of continuous-time signals

We apply a bandlimited signal $x(t)$, with $X(f) = 0$ for $|f| > B_x$, to a filter with bandlimited impulse response $h(t)$, i.e., $H(f) = 0$ for $|f| > B_h$. The resulting output signal $y(t)$ is given by the continuous-time convolution

$$y(t) = \int_{-\infty}^{+\infty} h(u)x(t-u)\mathrm{d}u = \int_{-\infty}^{+\infty} x(u)h(t-u)\mathrm{d}u \tag{8}$$

As $Y(f) = H(f)X(f)$, it follows that $Y(f) = 0$ for $|f| > B_\mathrm{min}$, where $B_\mathrm{min} = \min(B_x, B_h)$. To realize this filtering operation in a computer simulation, we need to represent the signals $x(t)$ and $y(t)$ and the impulse response $h(t)$ in (8) by their samples, which implies that the time variables t and u must be discretized.

When $1/T_\mathrm{in} \geq 2B_\mathrm{max}$, with $B_\mathrm{max} = \max(B_x, B_h)$, we can apply the sampling theorem to replace (8) by

$$y(t) = T_\mathrm{in} \sum_{m=-\infty}^{+\infty} h(mT_\mathrm{in})x(t-mT_\mathrm{in}) = T_\mathrm{in} \sum_{m=-\infty}^{+\infty} x(mT_\mathrm{in})h(t-mT_\mathrm{in}) \tag{9}$$

which formally corresponds to replacing u, $\mathrm{d}u$ and the integration over u by mT_in, T_in and the summation over m, respectively. Indeed, computing from (9) $Y(f)$ and applying a similar reasoning as in section 1 yields $Y(f) = H(f)X(f)$, which proves that (9) and (8) are equivalent.

Sampling $y(t)$ from (9) at a rate $1/T_\mathrm{out}$ gives rise to

$$\begin{aligned} y(kT_\mathrm{out}) &= T_\mathrm{in} \sum_{m=-\infty}^{+\infty} h(mT_\mathrm{in})x(kT_\mathrm{out} - mT_\mathrm{in}) \\ &= T_\mathrm{in} \sum_{m=-\infty}^{+\infty} x(mT_\mathrm{in})h(kT_\mathrm{out} - mT_\mathrm{in}) \end{aligned} \tag{10}$$

The samples $\{y(kT_\mathrm{out})\}$ contain all information about $y(t)$, provided that $1/T_\mathrm{out} \geq 2B_\mathrm{min}$. Considering the second line of (10), we refer to $1/T_\mathrm{in}$ and $1/T_\mathrm{out}$ as the sampling rates at the input and the output, respectively. One often can assume that $h(t)$ is significantly different from zero only over a finite interval; in this case the summation over m in (10) can be replaced by a finite summation.

2.1 Same sampling rates at input and output

Let us for simplicity take the same sampling frequencies at the input and the output of the filter: we select $1/T_\mathrm{in} = 1/T_\mathrm{out} = 1/T_\mathrm{s}$. To avoid aliasing, we need $1/T_\mathrm{s} \geq 2B_\mathrm{max}$. Eq. (10) simplifies to

$$y(kT_\mathrm{s}) = T_\mathrm{s} \sum_{m=-\infty}^{+\infty} h(mT_\mathrm{s})x(kT_s - mT_\mathrm{s}) = T_\mathrm{s} \sum_{m=-\infty}^{+\infty} x(mT_\mathrm{s})h(kT_\mathrm{s} - mT_\mathrm{s}) \tag{11}$$

which indicates that the samples $\{y(kT_\mathrm{s})\}$ are obtained as the discrete-time convolution of $\{x(kT_\mathrm{s})\}$ and $\{T_\mathrm{s}h(kT_\mathrm{s})\}$. This convolution is represented in Fig. 2.

2.2 Downsampling (decimation)

When $B_x \gg B_h$ or $B_h \gg B_x$, we obtain $B_\mathrm{min} \ll B_\mathrm{max}$; in this case, the sampling rate at the output can be taken much smaller than the sampling rate at the input, without causing aliasing at the output. We select $T_\mathrm{out} = N_\mathrm{d}T_\mathrm{in}$, where the integer N_d represents the downsampling factor or decimation factor, compared to the input sampling rate $1/T_\mathrm{in}$. To avoid aliasing, we simultaneously need $1/T_\mathrm{in} \geq 2B_\mathrm{max}$ and $1/T_\mathrm{out} \geq 2B_\mathrm{min}$.

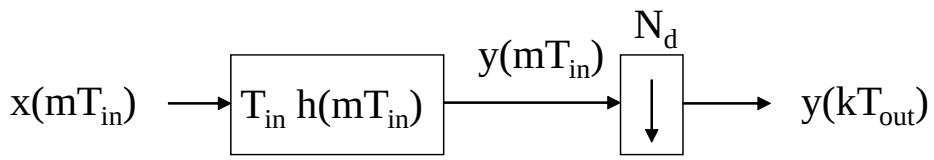

Figure 3: Convolution followed by decimation by factor N_d ($1/T_\mathrm{out} = 1/(N_\mathrm{d}T_\mathrm{in})$)

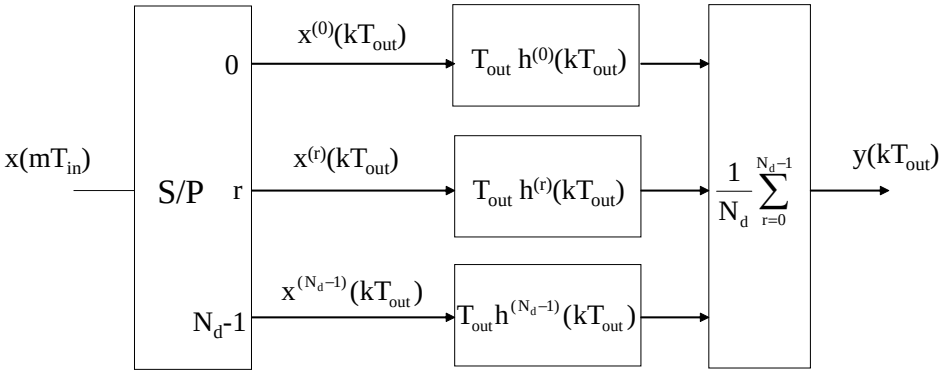

Figure 4: Efficient realization of convolution followed by decimation

Fig. 3 shows how the samples $\{y(kT_\mathrm{out})\}$ can be obtained: first the samples $\{y(mT_\mathrm{in})\}$are computed as the convolution of $\{x(mT_\mathrm{in})\}$ and $\{T_\mathrm{in}h(mT_\mathrm{in})\}$, and next a decimation by a factor N_d is applied, such that $y(kT_\mathrm{out}) = y(kN_\mathrm{d}T_\mathrm{in})$.

The strict execution of the operations from Fig. 3 is not computationally efficient, because only a fraction $1/N_\mathrm{d}$ of the samples $\{y(mT_\mathrm{in})\}$ is kept for further processing. To avoid unnecessary computations, one should compute the convolution of $\{x(mT_\mathrm{in})\}$ and $\{T_\mathrm{in}h(mT_\mathrm{in})\}$ only at the instants $kT_\mathrm{out} = kN_\mathrm{d}T_\mathrm{in}$ (rather that at all mT_in), which corresponds to shifting the samples $\{x(mT_\mathrm{in})\}$ by N_d steps (rather than a single step) in the filter memory before computing the next output sample. An equivalent approach is obtained by replacing in (10) T_in by $T_\mathrm{out}/N_\mathrm{d}$ and using the unique decomposition $m = qN_\mathrm{d} + r$, with $r \in \{0, 1, ...N_\mathrm{d} - 1\}$. This yields

$$\begin{aligned} y(kT_\mathrm{out}) &= \frac{1}{N_\mathrm{d}} \sum_{r=0}^{N_\mathrm{d}-1} \left(T_\mathrm{out} \sum_{q=-\infty}^{+\infty} x^{(r)}(kT_\mathrm{out} - qT_\mathrm{out}) h^{(r)}(qT_\mathrm{out}) \right) \\ &= \frac{1}{N_\mathrm{d}} \sum_{r=0}^{N_\mathrm{d}-1} \left(T_\mathrm{out} \sum_{q=-\infty}^{+\infty} h^{(r)}(kT_\mathrm{out} - qT_\mathrm{out}) x^{(r)}(qT_\mathrm{out}) \right) \end{aligned} \tag{12}$$

with $h^{(r)}(iT_\mathrm{out}) = h(iT_\mathrm{out} + rT_\mathrm{in})$ and $x^{(r)}(iT_\mathrm{out}) = x(iT_\mathrm{out} - rT_\mathrm{in})$. Fig. 4 illustrates the computations resulting from (12). The samples $x(mT_\mathrm{in})$, at a rate $1/T_\mathrm{in}$, are applied to a serial-to-parallel (S/P) converter (demultiplexer). The demultiplexer output with index r provides the sequence $\{x^{(r)}(kT_\mathrm{out})\}$, at a rate $1/T_\mathrm{out}$, which results from decimating (by a factor N_d) the samples $x(mT_\mathrm{in})$: $x^{(r)}(kT_\mathrm{out}) = x(kN_\mathrm{d}T_\mathrm{in} - rT_\mathrm{in})$. The decimated sequence $\{x^{(r)}(kT_\mathrm{out})\}$ is applied to a filter with impulse response $\{T_\mathrm{out}h^{(r)}(kT_\mathrm{out})\}$, which is obtained by decimating (by a factor N_d) the impulse response $\{T_\mathrm{out}h(mT_\mathrm{in})\}$: $h^{(r)}(kT_\mathrm{out}) = h(kN_\mathrm{d}T_\mathrm{in} + rT_\mathrm{in})$. Finally, $y(kT_\mathrm{out})$ is obtained as the arithmetical average of the filter outputs.

2.3 Upsampling

Suppose that we want the output sampling rate to be an integer multiple of the input sampling rate (for instance, to increase the time-domain resolution): $1/T_\mathrm{out} = N_\mathrm{u}/T_\mathrm{in}$, with N_u denoting the upsampling factor compared to $1/T_\mathrm{in}$. To avoid aliasing, we need $1/T_\mathrm{in} \geq 2B_\mathrm{max}$; then the condition $1/T_\mathrm{out} \geq 2B_\mathrm{min}$ is automatically satisfied because of the upsampling at the output. From (10) it follows that

$$y(kT_\mathrm{out}) = T_\mathrm{in} \sum_{m=-\infty}^{+\infty} x(mN_\mathrm{u}T_\mathrm{out}) h(kT_\mathrm{out} - mN_\mathrm{u}T_\mathrm{out})$$

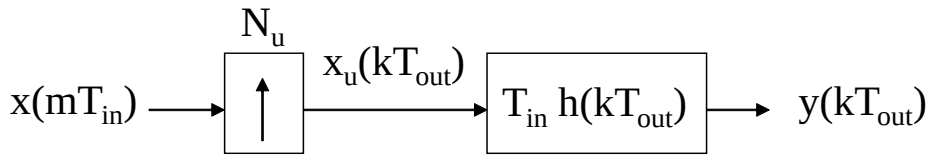

Figure 5: Upsampling (by factor N_u) followed by convolution ($1/T_\mathrm{out} = N_\mathrm{d}/T_\mathrm{in}$)

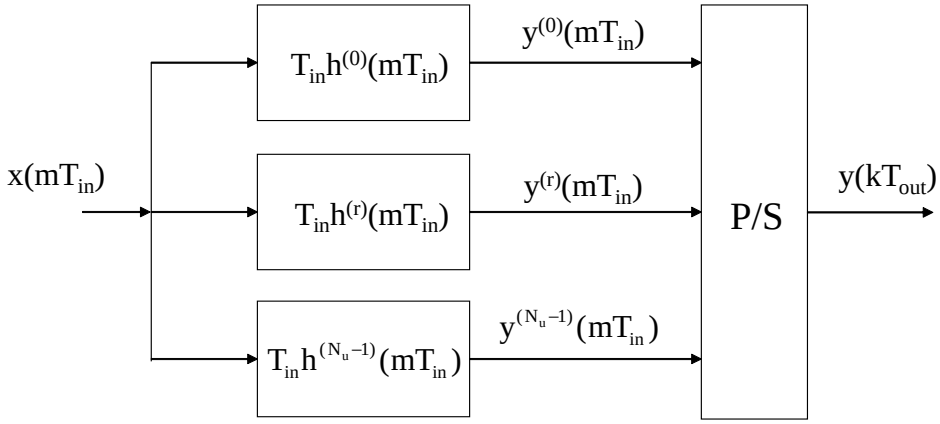

Figure 6: Efficient realization of upsampling followed by convolution

$$= T_\mathrm{in} \sum_{n=-\infty}^{+\infty} x_\mathrm{u}(nT_\mathrm{out}) h(kT_\mathrm{out} - nT_\mathrm{out}) = T_\mathrm{in} \sum_{n=-\infty}^{+\infty} h(nT_\mathrm{out}) x_\mathrm{u}(kT_\mathrm{out} - nT_\mathrm{out}) \quad (13)$$

where

$$x_\mathrm{u}(mN_\mathrm{u}T_\mathrm{out} + rT_\mathrm{out}) = \begin{cases} x(mT_\mathrm{in}) & r = 0 \\ 0 & r \in \{1, 2, ..., N_\mathrm{d} - 1\} \end{cases} \quad (14)$$

The sequence $\{x_\mathrm{u}(nT_\mathrm{out})\}$ is obtained by inserting $N_\mathrm{d} - 1$ zeroes between consecutive elements of $\{x(mT_\mathrm{in})\}$:

$$\{x_\mathrm{u}(nT_\mathrm{out})\} = \{\ldots 0, x(-T_\mathrm{in}), 0, \ldots 0, x(0), 0, \ldots, 0, x(T_\mathrm{in}), 0, \ldots\}$$

The operation (14) is referred to as upsampling the sequence $\{x(mT_\mathrm{in})\}$ by a factor N_u. Fig. 5 shows how the samples $y(kT_\mathrm{out})$ can be computed. First, $\{x_\mathrm{u}(kT_\mathrm{out})\}$ is obtained by upsampling $\{x(mT_\mathrm{in})\}$ by a factor N_u; next, $y(kT_\mathrm{out})$ results from convolving $\{x_\mathrm{u}(kT_\mathrm{out})\}$with $\{T_\mathrm{in} h(kT_\mathrm{out})\}$.

As a fraction $(N_\mathrm{u}-1)/N_\mathrm{u}$ of the elements of the sequence $\{x_\mathrm{u}(nT_\mathrm{out})\}$ is zero, it is not computationally efficient to compute the convolution as indicated in the last line of (13). The multiplications involving the zeroes of $\{x_\mathrm{u}(nT_\mathrm{out})\}$ can be avoided in the following way. Using in the first line of (13) the unique decomposition $k = qN_\mathrm{u} + r$, with $r \in \{0, 1, ...N_\mathrm{d} - 1\}$, we obtain

$$y^{(r)}(qT_\mathrm{in}) = T_\mathrm{in} \sum_{m=-\infty}^{+\infty} x(mT_\mathrm{in}) h^{(r)}(qT_\mathrm{in} - mT_\mathrm{in}) = T_\mathrm{in} \sum_{m=-\infty}^{+\infty} h^{(r)}(mT_\mathrm{in}) x(qT_\mathrm{in} - mT_\mathrm{in})$$

where $y^{(r)}(qT_\mathrm{in}) = y(qT_\mathrm{in} + rT_\mathrm{out})$ and $h^{(r)}(iT_\mathrm{in}) = h(iT_\mathrm{in} + rT_\mathrm{out})$. Fig. 6 shows that the samples $y(kT_\mathrm{out})$ are obtained at the output of a parallel-to-serial (P/S) converter (multiplexer); the P/S converter input with index r is the convolution of the input sequence $\{x(mT_\mathrm{in})\}$ with $\{T_\mathrm{in} h^{(r)}(mT_\mathrm{in})\}$.

2.4 Computation of impulse response from transfer function via IDFT

So far we have assumed that the sequence $\{h(mT_\mathrm{s})\}$ of impulse response samples is available. However, often we know the transfer function $H(f)$ rather than $h(t)$, in which case we need to numerically compute $\{h(mT_\mathrm{s})\}$ from $H(f)$.

To this aim, we define $\{x(k)\}$ and $\{X(n)\}$ as the samples of the periodic extensions of $h(t)$ and $H(f)/T_\mathrm{s}$, respectively:

$$x(k) = \sum_{i=-\infty}^{+\infty} h(kT_\mathrm{s} + iNT_\mathrm{s}) \quad (15)$$

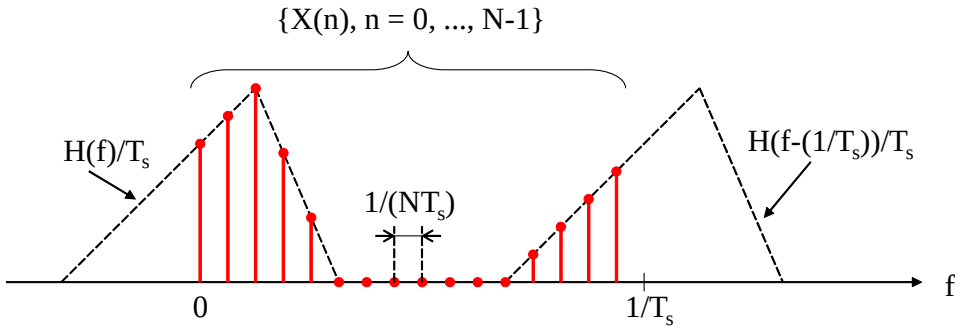

Figure 7: Relation between $H(f)$ and $\{X(n), n = 0, ..., N-1\}$

$$X(n) = \frac{1}{T_\mathrm{s}} \sum_{m=-\infty}^{+\infty} H\left(\frac{n}{NT_\mathrm{s}} + \frac{m}{T_\mathrm{s}}\right) \tag{16}$$

Note that $h(t)$ is extended with period NT_s and sampled at a rate $1/T_\mathrm{s}$, while $H(f)/T_\mathrm{s}$ is extended with period $1/T_\mathrm{s}$ and sampled with frequency spacing $1/(NT_\mathrm{s})$; N is a design parameter which will be specified later. We show in section 17 that $x(k)$ and $X(n)$, given by (15) and (16), respectively, are a discrete Fourier transform (DFT) pair[1]:

$$X(n) = \sum_{k=0}^{N-1} x(k) e^{-j2\pi \frac{kn}{N}} \tag{17}$$

$$x(k) = \frac{1}{N} \sum_{n=0}^{N-1} X(n) e^{j2\pi \frac{kn}{N}} \tag{18}$$

which expresses that $\{X(n), n = 0, ..., N-1\}$ is the N-point DFT of $\{x(k), k = 0, ..., N-1\}$, and $\{x(k), k = 0, ..., N-1\}$ is the N-point inverse DFT (IDFT) of $\{X(n), n = 0, ..., N-1\}$.

When $H(f) = 0$ for $|f| > B$ and $1/T_\mathrm{s} > 2B$, there is no overlap between the different shifts of $H(f)/T_\mathrm{s}$ in (16), which implies that $X(n)$ is simply obtained as (see Fig. 7)

$$X(n) = \begin{cases} \frac{1}{T_\mathrm{s}} H\left(\frac{n}{NT_\mathrm{s}}\right) & n = 0, ..., N/2 \\ \frac{1}{T_\mathrm{s}} H\left(\frac{n-N}{NT_\mathrm{s}}\right) & n = (N/2)+1, ..., N-1 \end{cases}$$

Suppose that $h(t)$ is essentially time-limited: $h(t) = 0$ for $|t| > T_h$. Selecting $N > 2T_h/T_\mathrm{s}$, it follows that there is no overlap between the different shifts of $h(t)$ in (15); hence, having obtained $x(k)$ as the IDFT of $X(n)$, we can easily retrieve $h(kT_\mathrm{s})$ as (see Fig. 8)

$$h(kT_\mathrm{s}) = \begin{cases} x(k+N) & k = (-N/2)+1, ..., -1 \\ x(k) & k = 0, ..., N/2 \end{cases}$$

From the obtained $\{h(kT_\mathrm{s}), k = (-N/2)+1, ..., N/2\}$, we keep only a subset $\{h(kT_\mathrm{s}), k \in [-K_1, K_2]\}$, which results from discarding samples that are negligibly small.

The size N of the IDFT depends on the duration of $h(t)$, which however is not known beforehand. The value of N must be determined experimentally: when $|h(NT_\mathrm{s}/2)|$ is not negligibly small compared to the maximum of $|h(kT_\mathrm{s})|$ over $k = (-N/2)+1, ..., N/2$, we must repeat the computation with a larger N.

[1] Attention: some software programs use a different factor in front of the summation or a different sign in the exponent when defining the DFT and IDFT.

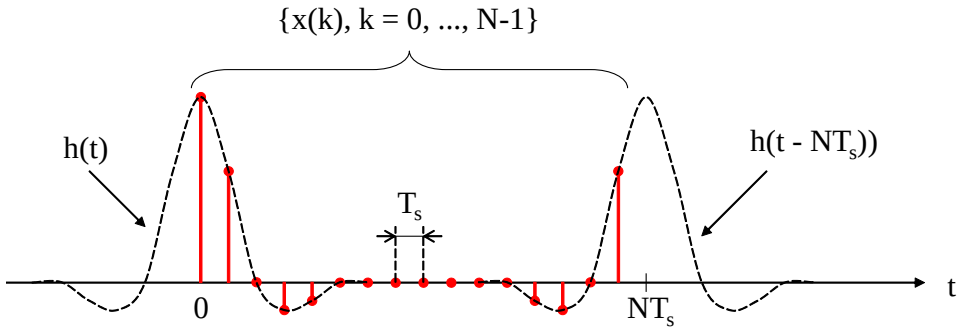

Figure 8: Relation between $h(t)$ and $\{x(k), k = 0, ..., N-1\}$

2.5 Computation of transfer function from impulse response via DFT

Here we consider the dual of the problem from section 2.4: we assume that the impulse response $h(t)$ is known, and we numerically compute the transfer function $H(f)$ at discrete values of f.

We assume that $H(f) = 0$ for $|f| > B$. We sample $h(t)$ at a rate $1/T_s$, with $1/T_s > 2B$. From the samples $\{h(kT_s\}$, we compute $\{x(k), k = 0, ..., N-1\}$ according to (15). Next, we determine $\{X(n), n = 0, ..., N-1\}$ as the DFT of $\{x(k), k = 0, ..., N-1\}$. As $1/T_s > 2B$, there is no overlap between the various shifts of $H(f)/T_s$ in (16); this yields (see Fig. 7)

$$H\left(\frac{n}{NT_s}\right) = \begin{cases} T_s X(n+N) & n = (-N/2)+1, ..., -1 \\ T_s X(n) & n = 0, ..., N/2 \end{cases}$$

This procedure yields $\{H(nF)\}$ with frequency step $F = 1/(NT_s)$. For given F, we must select N such that $1/T_s = NF$ exceeds $2B$. When B is not known, N must be obtained experimentally: when $\left|H\left(\frac{1}{2T_s}\right)\right|$ is not negligibly small compared to the maximum of $\left|H\left(\frac{n}{NT_s}\right)\right|$over $n = (-N/2)+1, ..., N/2$, the computation must be repeated with a larger N.

3 Linear digital modulation

Linear digital modulation involves the conversion of a sequence of data symbols $\{a(k)\}$ into a continuous-time signal $s(t)$, according to

$$s(t) = A\sum_m a(m)h_{\rm tr}(t - mT) \tag{19}$$

where $h_{\rm tr}(t)$ is a transmit pulse with bandwidth B (i.e., $H_{\rm tr}(f) = 0$ for $|f| > B$) and A is a scaling factor.

In a computer simulation we represent $s(t)$ by its samples taken at a rate $1/T_s$, which we select to be a multiple (N_s) of the symbol rate $1/T$ ($T = N_s T_s$):

$$s(kT_s) = A\sum_m a(m)h_{\rm tr}(kT_s - mN_sT_s) \tag{20}$$

To avoid aliasing, we require $1/T_s \geq 2B$. In the case of bandwidth-efficient modulation, we have $0.5 < BT < 1$, in which case $N_s = 2$ is sufficient to avoid aliasing. Considering that (20) has the same form as the first line of (13), with $a(m)$ standing for $x(mN_sT_s)$, we can compute $s(kT_s)$ using the methods from section 2.3: $\{a(m))$ is upsampled by a factor N_s, and the resulting sequence $\{a_u(k)\}$ is convolved with $\{Ah_{\rm tr}(kT_s)\}$, as indicated in Fig. 9; alternatively, the more computationally-efficient method as in Fig. 6 can be used.

4 White Gaussian noise

We consider a signal $r(t)$ which is the sum of a useful signal $s(t)$ and complex-valued white Gaussian noise $w(t) = w_{\rm R}(t) + jw_{\rm I}(t)$, yielding $r(t) = s(t) + w(t)$. The noise components $w_{\rm R}(t)$ and $w_{\rm I}(t)$ are statistically independent, and have a constant two-sided power spectral density (PSD) equal to $N_0/2$. The useful signal $s(t)$ has a bandwidth B_s, i.e., $S(f) = 0$ for $|f| > B_s$.

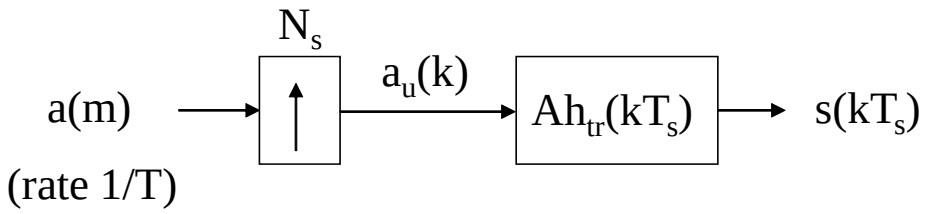

Figure 9: Simulation of digital modulation

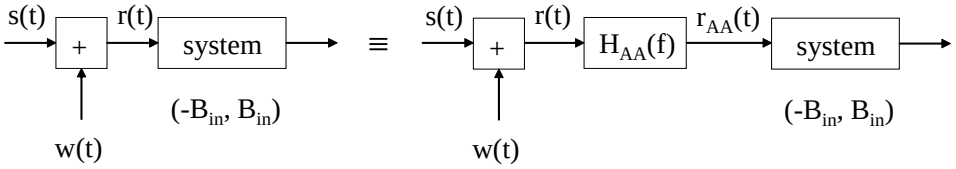

Figure 10: The output of the bandlimited system is not affected by inserting the filter $H_{\rm AA}(f)$.

In a computer simulation the time-variable t must be discretized. However, as the white noise $w(t)$ has an infinite bandwidth, we cannot represent $w(t)$ by its samples because of inevitable aliasing. Fortunately, in practical applications the signal $r(t)$ will be processed by a finite-bandwidth system, which does not respond to frequencies outside an interval $(-B_{\rm in}, B_{\rm in})$. Hence, the response of this finite-bandwidth will not be altered when we first apply $r(t)$ to a *fictitious* anti-aliasing filter $H_{\rm AA}(f)$, with $H_{\rm AA}(f) = 1$ for $|f| \leq B_{\rm s}$ and arbitrary $H_{\rm AA}(f)$ elsewhere, and apply the resulting signal $r_{\rm AA}(t)$ to the system with bandwidth $B_{\rm s}$, as indicated in Fig. 10. The introduction of the fictitious filter $H_{\rm AA}(f)$ allows making $r_{\rm AA}(t)$ bandlimited, so that $r_{\rm AA}(t)$ can be represented by its samples taken at a suitable rate, without the occurrence of aliasing.

We select $H_{\rm AA}(f)$ to be an ideal unit-gain lowpass filter with bandwidth $B_{\rm AA}$(see Fig. 11):

$$H_{\rm AA}(f) = \text{rect}\left(\frac{f}{2B_{\rm AA}}\right) \tag{21}$$

with $B_{\rm AA} > B_{\rm max}$ and $B_{\rm max} = \max(B_s, B_{\rm in})$. This selection of $B_{\rm AA}$ guarantees that $s(t)$ is not distorted by the anti-aliasing filter, and that the part of $r(t)$ contained in the frequency interval $(-B_{\rm in}, B_{\rm in})$ is also present in $r_{\rm AA}(t)$. This yields $r_{\rm AA}(t) = s(t) + w_{\rm AA}(t)$, where $w_{\rm AA}(t)$ is the response of the filter $H_{\rm AA}(f)$ to the white noise $w(t)$. Let us decompose $w_{\rm AA}(t)$ into its real and imaginary parts: $w_{\rm AA}(t) = w_{\rm AA,R}(t) + jw_{\rm AA,I}(t)$. Considering that the impulse response $h_{\rm AA}(t)$ of the filter $H_{\rm AA}(f)$ from (21) is real-valued (more specifically, we have $h_{\rm AA}(t) = 2B_{\rm AA}\text{sinc}(2B_{\rm AA}t)$), it follows that $w_{\rm AA,R}(t)$ and $w_{\rm AA,I}(t)$ are the responses of the filter $H_{\rm AA}(f)$ to $w_{\rm R}(t)$ and $w_{\rm I}(t)$, respectively. Hence, $w_{\rm AA,R}(t)$

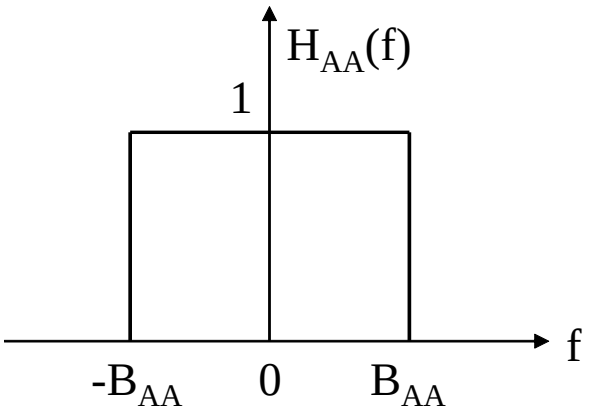

Figure 11: Ideal lowpass anti-aliasing filter

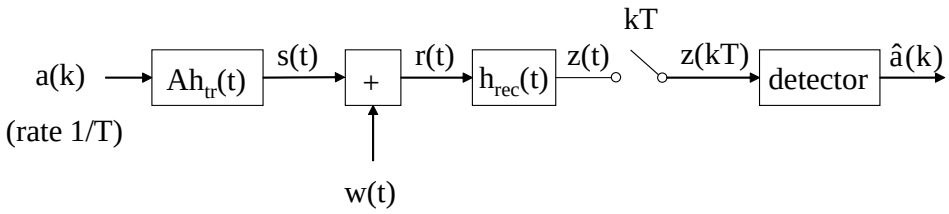

Figure 12: Digital communication system

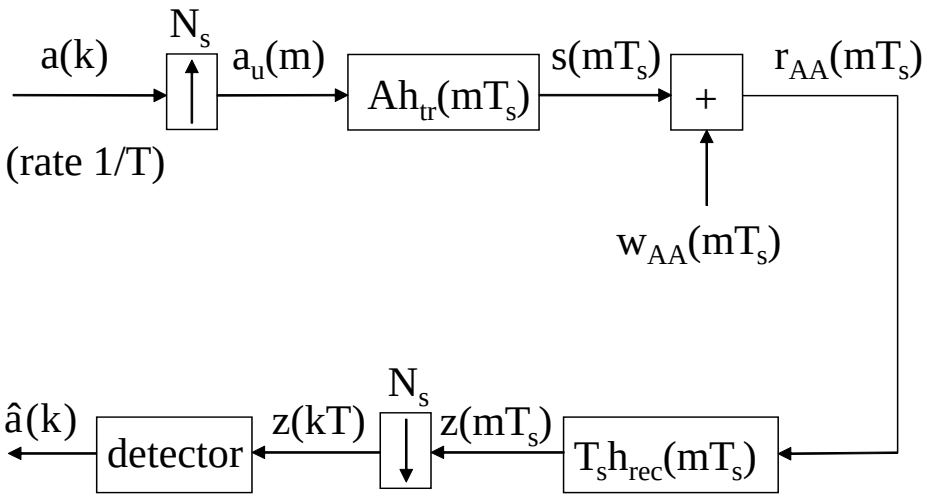

Figure 13: Simulation set-up for digital communication system

and $w_{\mathrm{AA,I}}(t)$ are statistically independent Gaussian processes, and have the same PSD $S_{\mathrm{AA}}(f)$, which is obtained as $S_{\mathrm{AA}}(f) = (N_0/2)|H_{\mathrm{AA}}(f)|^2$; their corresponding autocorrelation function $R_{\mathrm{AA}}(u)$ is the inverse FT (IFT) of $S_{\mathrm{AA}}(f)$, yielding $R_{\mathrm{AA}}(u) = N_0 B_{\mathrm{AA}} \mathrm{sinc}(2B_{\mathrm{AA}}u)$.

In the computer simulation, we represent $r_{\mathrm{AA}}(t)$ by its samples $r_{\mathrm{AA}}(kT_{\mathrm{s}})$; we select $1/T_{\mathrm{s}} = 2B_{\mathrm{AA}}$, so that no aliasing occurs. This yields

$$r_{\mathrm{AA}}(kT_{\mathrm{s}}) = s(kT_{\mathrm{s}}) + w_{\mathrm{AA}}(kT_{\mathrm{s}}) = s(kT_{\mathrm{s}}) + w_{\mathrm{AA,R}}(kT_{\mathrm{s}}) + jw_{\mathrm{AA,I}}(kT_{\mathrm{s}})$$

The noise sequences $\{w_{\mathrm{AA,R}}(kT_{\mathrm{s}})\}$ and $\{w_{\mathrm{AA,I}}(kT_{\mathrm{s}})\}$ have the same autocorrelation function $R_{\mathrm{AA}}(mT_{\mathrm{s}})$, which is obtained by sampling $R_{\mathrm{AA}}(u)$ at a rate $1/T_{\mathrm{s}}$. We have

$$R_{\mathrm{AA}}(mT_{\mathrm{s}}) = \frac{N_0}{2T_{\mathrm{s}}}\delta_m$$

which indicates that the elements of $\{w_{\mathrm{AA,R}}(kT_{\mathrm{s}})\}$(and, similarly, the elements of $\{w_{\mathrm{AA,I}}(kT_{\mathrm{s}})\}$) are statistically independent, with variance equal to $N_0/(2T_{\mathrm{s}})$. Hence, the generation of white Gaussian noise boils down to the generation of statistically independent Gaussian random variables with the proper variance, which depends not only on N_0 but also on the sampling rate $1/T_{\mathrm{s}}$.

5 Simulation of a digital communication system

We consider the simple digital communication system displayed in Fig. 12. The data symbol sequence $\{a(k)\}$ has a rate $1/T$. The transmit pulse $h_{tr}(t)$ and the impulse response $h_{\mathrm{rec}}(t)$ of the receive filter have a bandwidth B. Based on the sample $z(kT)$, a decision is made regarding the symbol $a(k)$ and the bits contained in $a(k)$.

The computer simulation of this system can be accomplished as indicated in Fig. 13:

- A sampling frequency $1/T_{\mathrm{s}} = N_{\mathrm{s}}/T$ is selected, with $N_{\mathrm{s}} > 2BT$ to avoid aliasing.

- The symbol sequence $\{a(k)\}$ is upsampled by a factor N_s. The upsampled sequence $\{a_\mathrm{u}(m)\}$ is convolved with $\{Ah_\mathrm{tr}(mT_s)\}$, yielding the samples $\{s(mT_s)\}$ of the transmit signal. These operations can be performed efficiently according to Fig. 6.

- The sequence $\{r_{AA}(mT_\mathrm{s})\}$ is obtained by adding to $\{s(mT_\mathrm{s})\}$ a discrete-time complex-valued white Gaussian noise sequence $\{w_{AA}(mT_\mathrm{s})\}$. The real and imaginary parts of $w_{AA}(mT_\mathrm{s})$ have a variance equal to $N_0/(2T_\mathrm{s})$.

- The sequence $z(kT)$ results from convolving $\{r_{AA}(mT_\mathrm{s})\}$ with $\{T_s h_\mathrm{rec}(mT_s)\}$, followed by downsampling by a factor N_s, so that only the result of the convolution at instants $kN_\mathrm{s}T_\mathrm{s}$ is kept. These operations can be performed efficiently according to Fig. 4.

- Application of the decision rule to $z(kT)$ yields the decision $\hat{a}(k)$. The corresponding bit decisions follow from the demapping of $\hat{a}(k)$.

In some cases it is useful to represent $z(t)$ by means of a large number of samples per symbol (e.g., to obtain the eye diagram of $z(t)$). This can be accomplished by upsampling $\{r_{AA}(mT_\mathrm{s})\}$ by a factor N_u, and convolving the resulting $\{r_{\mathrm{AA,u}}(nT_s/N_\mathrm{u})\}$ with $\{T_\mathrm{s}h(nT_s/N_\mathrm{u})\}$. This yields $\{z(nT_s/N_\mathrm{u})\}$, corresponding to $N_\mathrm{u}N_\mathrm{s}$ samples per symbol interval T. The corresponding operations can be performed efficiently according to Fig. 6.

6 Estimation of power and power spectral density

We consider a zero-mean stationary stochastic signal $x(t)$ with autocorrelation function $R_x(u) = \mathbb{E}[x(t+u)x^*(t)]$, where $\mathbb{E}[.]$ denotes statistical expectation. The corresponding two-sided PSD $S_x(f)$ equals the FT of $R_x(u)$. The average power P_x of $x(t)$ is given by $P_x = \mathbb{E}[|x(t)|^2] = R_x(0) = \int S_x(f)\mathrm{d}f$. We assume that $S_x(f) = 0$ for $|f| > B$. Samples $\{x(kT_\mathrm{s})\}$ with $1/T_\mathrm{s} > 2B$ are available for estimating the power and the PSD.

6.1 Power estimate

Considering that $P_x = \mathbb{E}[|x(t)|^2]$, an estimate $\hat{P}_x$ of the average power P_x can be obtained from a finite number of samples $\{x(kT_\mathrm{s}), k = 0, ..., K-1\}$ as

$$\hat{P}_x = \frac{1}{K}\sum_{k=0}^{K-1} |x(kT_\mathrm{s})|^2 \tag{22}$$

with K denoting the number of samples used. The estimate $\hat{P}_x$ deviates from P_x, because the statistical expectation is replaced by an arithmetical average over a finite number (K) of samples.

6.2 Power spectral density estimate

We present the method of *averaged modified periodograms* for estimating the PSD $S_x(f)$. According to this method, the sequence $\{x(kT_\mathrm{s})\}$ is first split into blocks of K samples. Assuming the number of blocks equals I, we denote the i-th block as $\{x^{(i)}(kT_\mathrm{s}), k = 0, ..., K-1\}$, where $x^{(i)}(kT_\mathrm{s}) = x(iKT_\mathrm{s} + kT_\mathrm{s})$, for $i = 0, ..., I-1$. A normalized window function $\{w(k), k = 0, ..., K-1\}$, i.e.,

$$\sum_{k=0}^{K-1} |w(k)|^2 = 1 \tag{23}$$

is applied to each block. This yields I windowed blocks $\{y^{(i)}(kT_\mathrm{s}), k = 0, ..., K-1\}$, with $y^{(i)}(kT_\mathrm{s}) = x^{(i)}(kT_\mathrm{s})w(k)$. Next, the N-point DFT of each windowed block is computed, with $N \geq K$; when $N > K$, $N-K$ zeroes are appended at the end of each windowed block before taking the DFT. The DFT of the i-th block is denoted $\{Y^{(i)}(n), n = 0, ..., N-1\}$. In section 18 we point out that, for $n = 0, ..., N-1$,

$$T_\mathrm{s}\mathbb{E}[|Y^{(i)}(n)|^2] = T_\mathrm{s}\int_{1/T_\mathrm{s}} \left(\sum_{l=-\infty}^{+\infty} S_x\left(\frac{n}{NT_s} - \frac{l}{T_s} - \nu\right)\right) |W(e^{j2\pi\nu T_s})|^2 \mathrm{d}\nu \tag{24}$$

where the summation over l represents the periodic extension (with period $1/T_s$) of the PSD, and

$$W(e^{j2\pi\nu T_s}) = \sum_{k=-\infty}^{+\infty} w(k)e^{-j2\pi k\nu T_s} \tag{25}$$

is the DTFT of the window $w(k)$; because of the finite window length, the summation in (25) can be limited to $k = 0, ..., K-1$. Eq. (24) indicates that $T_\mathrm{s}\mathbb{E}[|Y^{(i)}(n)|^2]$ equals the convolution (evaluated at frequency $n/(NT_\mathrm{s})$) of the periodic extension of $S_x(f)$ and the scaled squared magnitude $T_\mathrm{s}|W(e^{j2\pi fT_s})|^2$ of the window DTFT. With increasing K, $|W(e^{j2\pi\nu T_s})|^2$ becomes a narrow peak near $\nu = 0$, yielding the following approximation for large K:

$$\begin{aligned} T_\mathrm{s}\mathbb{E}[|Y^{(i)}(n)|^2] &\approx \left(\sum_{l=-\infty}^{+\infty} S_x\left(\frac{n}{NT_s} - \frac{l}{T_s}\right)\right) T_\mathrm{s}\int_{1/T_\mathrm{s}} |W(e^{j2\pi\nu T_s})|^2 \mathrm{d}\nu \\ &= \sum_{l=-\infty}^{+\infty} S_x\left(\frac{n}{NT_s} - \frac{l}{T_s}\right) \end{aligned}$$

where the last line follows from (23). Approximating in (24) the expectation by an arithmetical average over the blocks, we define

$$S_n = T_\mathrm{s}\frac{1}{I}\sum_{i=0}^{I-1} |Y^{(i)}(n)|^2$$

The sequence $\{S_n, n = 0, .., N-1\}$ is an estimate of the periodic extension (with period $1/T_\mathrm{s}$) of $S_x(f)$ at the frequencies $n/(NT_\mathrm{s})$ with $n = 0, .., N-1$. As $1/T_\mathrm{s} > 2B$, the different contributions $S_x(f - l/T_s)$ to the periodic extension do not overlap. Hence, the PSD estimate is obtained as

$$\hat{S}_x\left(\frac{n}{NT_\mathrm{s}}\right) = \begin{cases} S_{N+n} & n = -(N/2)+1, ..., -1 \\ S_n & n = 0, ..., N/2 \end{cases} \tag{26}$$

The PSD estimate (26) differs from the actual PSD for two reasons. First, the expectation in (24) is replaced by an arithmetical average over a finite (I) number of blocks. Secondly, because of the convolution in (24), the PSD estimate is a smeared version of the actual PSD. To limit the smearing effect, $|W(e^{j2\pi fT_s})|^2$ should be much more narrow than $S_x(f)$, which requires $KT_sB \gg 1$.

7 Error performance estimation

The error performance of a communication system is expressed as the "data-unit" error rate (DER); the DER is the ratio of the average number of data units received in error to the total number of data units transmitted. Depending on the specific context, a data unit can denote an information bit, a data symbol, a codeword, a packet, ..., in which case the error performance denotes the bit error rate (BER), the symbol error rate (SER), the word error rate (WER), the packet error rate (PER), ...

Two methods for estimating the error performance by means of a computer simulation will be considered. In the first method, a fixed number of data units is transmitted, and the corresponding number of erroneous data units is counted. In the second method, blocks containing a fixed number of data units are transmitted consecutively, until a given number of erroneous blocks is received; the number of erroneous data units in the erroneous blocks is counted.

Typically, the blocks are selected such that the number of data unit errors in the different blocks are statistically independent and identically distributed. This is illustrated using the simple additive white Gaussian noise (AWGN) channel model

$$y_n = a_n + w_n$$

where a_n denotes the n-th transmitted data symbol from a given M_c-point constellation, w_n is stationary additive white Gaussian noise, and y_n is the received signal. We make a distinction between coded and uncoded transmission.

- Uncoded transmission

  Because the noise is white and stationary, symbol errors occur independently and with the same probability. However, the $\log_2(M_\mathrm{c})$ bits contained in a same symbol in general have different error probabilities, and their errors do not occur independently. Exceptions are 2-PSK or 2-PAM (because they contain only one bit), and Gray-mapped 4-QAM or 4-PSK.

- When the error performance is expressed as the BER, a data unit corresponds to one bit. We select a block to be a data symbol (representing $\log_2(M_c)$ bits), because data symbol errors occur independently.
- When the error performance is expressed as the SER, a data unit corresponds to one data symbol. Because of the independent data symbol errors, we select a block to be a data symbol as well.

- Coded transmission

  Groups of information bits are encoded into codewords. The coded bits are mapped to data symbols $\{a_n\}$. We assume that a codeword contains an integer number of data symbols. The receiver observes the noisy data symbols that constitute a codeword, and performs the decoding: the decoder delivers the information bits for which the corresponding vector of coded symbols is the closest to the received noisy symbol vector. A decoding error occurs when at least one retrieved information bit is different from the information bit transmitted. Because the noise is white and stationary, decoding errors occur independently and with the same probability. However, the information bits contained in a same codeword in general have different error probabilities, and their errors do not occur independently.

  - When the error performance is expressed as the BER, a data unit corresponds to one information bit. We select a block to be a codeword, because codeword errors occur independently.
  - When the error performance is expressed as the WER, a data unit corresponds to one codeword. Because of the independent codeword errors, we select a block to be a codeword as well.

7.1 Transmission of a fixed number of data units

We consider the transmission of K data units as the transmission M blocks, each consisting of N data units. i.e, $K = MN$. We introduce the random variable E_k, with $E_k = 1$ when the k-th data unit is received in error, and $E_k = 0$ otherwise. The estimate of the DER is obtained as

$$\widehat{\mathrm{DER}} = \frac{1}{K}\sum_{k=1}^{K} E_k \tag{27}$$

with $\sum_{k=1}^{K} E_k$ denoting the number of erroneous data units. Alternatively, we rewrite the estimate (27) as

$$\widehat{\mathrm{DER}} = \frac{1}{MN}\sum_{m=1}^{M} J_m \tag{28}$$

where $J_m \in \{0, ..., N\}$ denotes the number of erroneous data units in the m-th block.

7.2 Transmission until a fixed number of erroneous blocks

Consider the transmission of blocks of N data units until M_e erroneous blocks occur. Denoting by m_i the index of the i-th erroneous block, the estimate of the DER is obtained as

$$\widehat{\mathrm{DER}} = \frac{1}{N}\cdot\frac{\sum_{i=1}^{M_\mathrm{e}} J_{\mathrm{e},i}}{m_{M_\mathrm{e}}} = \frac{1}{N}\cdot\frac{\sum_{i=1}^{M_\mathrm{e}} J_{\mathrm{e},i}}{\sum_{i=1}^{M_\mathrm{e}} L_i} \tag{29}$$

where $J_{\mathrm{e},i} = J_{m_i} \in \{1, ..., N\}$ is the number of data unit errors in the i-th *erroneous* block, and $L_i = m_i - m_{i-1}$ (with $m_0 = 0$) denotes the number of blocks transmitted since the $(i-1)$-th erroneous block until (and including) the i-th erroneous block.

8 Estimation accuracy

Several estimation problems can be cast within he following framework.

Assume we want to estimate a real-valued quantity U from a *stationary* sequence $\{u_k, k = 0, ..., K-1\}$ of random variables, with $\mathbb{E}[u_k] = U$. We consider an estimate $\hat{U}$ which is the arithmetical average of these random variables:

$$\hat{U} = \frac{1}{K}\sum_{k=0}^{K-1} u_k \tag{30}$$

It is easily verified that $\mathbb{E}[\hat{U}] = U$, i.e., the estimate is unbiased. Introducing the relative estimation error

$$\epsilon = \frac{\hat{U} - U}{U} \tag{31}$$

the estimation accuracy can be expressed by means of the root mean-square (rms) relative estimation error ϵ_{rms}, with

$$\epsilon_{\text{rms}}^2 = \mathbb{E}[\epsilon^2] = \frac{\mathbb{E}\left[\left(\hat{U} - U\right)^2\right]}{U^2} = \frac{\mathbb{E}\left[\hat{U}^2\right]}{U^2} - 1 \tag{32}$$

Hence, $\hat{U}$ has mean U and variance $\sigma_{\hat{U}}^2 = \epsilon_{\text{rms}}^2 U^2$. The quantity ϵ_{rms}^2 represents the mean-square relative estimation error.

Assuming that $\hat{U}$ has a Gaussian distribution, with mean U and variance $\sigma_{\hat{U}}^2$, it can be verified that $\Pr[|\hat{U} - U| < 1.96\sigma_{\hat{U}}] = 0.95$; equivalently, we have $\Pr[U \in \mathcal{I}(\hat{U})] = 0.95$, where $\mathcal{I}(\hat{U}) = \left(\hat{U} - 1.96\sigma_{\hat{U}}, \hat{U} + 1.96\sigma_{\hat{U}}\right)$ denotes the confidence interval (with a 95% confidence level) related to the estimate $\hat{U}$. This should be interpreted as follows. Consider I identically distributed sets $\{u_k^{(i)}, k = 0, ..., K-1\}$ with $i = 0, ...I-1$, and derive for each set the corresponding estimate $\hat{U}^{(i)}$, which is obtained by replacing u_k by $u_k^{(i)}$ in (30). For large I, 95 % of the estimates $\hat{U}^{(i)}$ satisfy the condition $|\hat{U}^{(i)} - U| < 1.96\sigma_{\hat{U}}$. Equivalently, the quantity U is contained in 95 % of the confidence intervals $\left\{\mathcal{I}(\hat{U}^{(i)}), i = 1, ..., I\right\}$.

In the following we evaluate ϵ_{rms}^2 based on the statistical properties of $\{u_k, k = 0, ..., K-1\}$. Using the decomposing $u_k = U + e_k$, from which we obtain $\mathbb{E}[e_k] = 0$, ϵ_{rms}^2 from (32) can be expressed as

$$\epsilon_{\text{rms}}^2 = \frac{1}{K^2 U^2} \sum_{k_1=0}^{K-1} \sum_{k_2=0}^{K-1} R_e(k_1 - k_2) \tag{33}$$

where $R_e(m) = \mathbb{E}[e_{k+m} e_m]$ is the autocorrelation function of the sequence $\{e_k\}$.

We separately consider the cases of uncorrelated and correlated variables $\{e_k\}$. We will show that ϵ_{rms}^2 is essentially inversely proportional to K (i.e., $\epsilon_{\text{rms}}^2 \approx C/K$) and explain how to obtain an estimate $\hat{C}$ of the constant of proportionality C. For given K, the mean-square relative estimation error is estimated as $\epsilon_{\text{rms}}^2 \approx \hat{C}/K$. Equivalently, the number K of variables u_k required to achieve a given ϵ_{rms}^2 is approximated as $K \approx \hat{C}/\epsilon_{\text{rms}}^2$.

8.1 Uncorrelated variables

Let us compute ϵ^2 assuming that the variables $\{e_k\}$ are uncorrelated: $R_e(m) = R_e(0)\delta_m$. In this case, (33) reduces to

$$\epsilon_{\text{rms}}^2 = \frac{C}{K} \tag{34}$$

with

$$C = \frac{R_e(0)}{U^2} = \frac{\mathbb{E}[u_k^2]}{U^2} - 1 \tag{35}$$

indicating that ϵ_{rms}^2 is inversely proportional to K.

However, as $\mathbb{E}[u_k^2]$ and U are not known beforehand, the value (35) of C to be used in (34) cannot be computed; instead, C but must be estimated from $\{u_k, k = 0, ..., K-1\}$. The following estimate can be used:

$$\hat{C} = \frac{\frac{1}{K}\sum_{k=0}^{K-1} u_k^2}{\left(\frac{1}{K}\sum_{k=0}^{K-1} u_k\right)^2} - 1 \tag{36}$$

which is obtained from (35) by substituting U by $\hat{U}$ and replacing the statistical expectation by an arithmetical average.

8.2 Correlated variables

When the variables $\{e_k\}$ are correlated, a simple approximation of ϵ^2 from (33) can be obtained when $R_e(m)$ becomes negligibly small for $|m| > L$; L is denoted the correlation length of $\{e_k\}$. For $K \gg L$, we obtain

$$\epsilon^2_{\text{rms}} \approx \frac{C}{K} \tag{37}$$

with

$$C = \frac{1}{U^2} \sum_{m=-\infty}^{+\infty} R_e(m) \tag{38}$$

It follows that ϵ^2 is (within the approximation) again inversely proportional to K, but in general the value of C is not the same as for uncorrelated variables as it also depends on $\{R_e(m), m \neq 0\}$. In the case of uncorrelated variables, C from (38) reduces to (35); hence, C from (38) can be used for both cases.

To obtain an estimate $\hat{C}$, we split the sequence $\{u_k\}$ into M blocks of N samples each, with $N \gg L$. We denote the m-th block as $\{u_n^{(m)}, n = 0, ..., N-1\}$, where $u_n^{(m)} = u_{mN+n}$, for $m = 0, ..., M-1$. From each block, we derive an estimate of U, based on N variables:

$$\hat{U}^{(m)} = \frac{1}{N} \sum_{n=0}^{N-1} u_n^{(m)} \quad m = 0, ..., M-1 \tag{39}$$

The estimate of U resulting from the entire sequence (MN samples) is given by the arithmetical average of the per-block estimates:

$$\hat{U} = \frac{1}{M} \sum_{m=0}^{M-1} \hat{U}^{(m)} = \frac{1}{MN} \sum_{k=0}^{MN-1} u_k \tag{40}$$

which is equivalent to (30) with $K = MN$. Considering that an estimate of U based on N variables yields a mean-square relative error given by $\epsilon^2 = C/N$, the estimate $\hat{C}$ is obtained as

$$\hat{C} = N \cdot \left(\frac{1}{\hat{U}^2} \left(\frac{1}{M} \sum_{m=0}^{M-1} (\hat{U}^{(m)})^2 \right) - 1 \right) \tag{41}$$

which results from approximating in (32) U by $\hat{U}$ and the statistical expectation by the arithmetical average over M terms. Taking $N = 1$ and $M = K$, the estimate (41) reduces to (36); note that taking $N = 1$ is not appropriate for correlated variables, as it violates the condition $N \gg L$.

9 Power estimation accuracy

In the case of estimating the power of a signal $x(t)$ with bandwidth B, U and $\hat{U}$ stand for P_x and $\hat{P}_x$, respectively. Taking $u_k = |x(kT_\text{s})|^2$, the estimate $\hat{U}$ from (30) coincides with the power estimate (22). As $\mathbb{E}[|x(kT_\text{s})|^2] = P_x$, the estimate $\hat{P}_x$ is unbiased. Usually the samples $\{x(kT_\text{s})\}$ are correlated, so that also the variables $\{|x(kT_\text{s})|^2\}$ are correlated; the order of magnitude of the correlation length L is $10BT_\text{s}$.

The power estimate based on K samples $\{x(kT_\text{s}), k = 0, ..., K-1\}$ gives rise to a mean-square relative estimation error ϵ^2_{rms} satisfying $\epsilon^2_{\text{rms}} \approx C/K$, where C can be estimated according to (41).

10 Error performance estimation accuracy

10.1 Transmission of a fixed number of data units

We consider the DER estimate (28), obtained from the transmission of M blocks of N data units each, i.e., transmission of $K = MN$ data units in total. We assume that the selected block size N is sufficiently large, so that the variables $\{J_m, m = 1, ..., M\}$ can be considered statistically independent and identically distributed. It is easily verified that $\mathbb{E}[J_m] = N \cdot \text{DER}$, so that $\widehat{\text{DER}}$ is an unbiased estimate. Using a similar reasoning as in section 8.1, we get

$$\epsilon^2_{\text{rms},1} = \frac{C}{M} \tag{42}$$

with

$$C = \frac{\mathbb{E}[J_m^2]}{(\mathbb{E}[J_m])^2} - 1 \tag{43}$$

An estimate of C is obtained as

$$\hat{C} = \frac{\frac{1}{M}\sum_{m=0}^{M-1} J_m^2}{\left(\frac{1}{M}\sum_{m=0}^{M-1} J_m\right)^2} - 1 \tag{44}$$

Special case: independent data unit errors

In the case of independent data unit errors, we can take $N = 1$ and $M = K$, in which case a block corresponds to a single data unit, and K such blocks are transmitted. This yields $J_m \in \{0, 1\}$, with $\Pr[J_m = 1] = \text{DER}$ and $\Pr[J_m = 0] = 1 - \text{DER}$, so that $\mathbb{E}[J_m] = \mathbb{E}[J_m^2] = \text{DER}$. Hence, (43) reduces to

$$C = \frac{1 - \text{DER}}{\text{DER}} \approx \frac{1}{\text{DER}}$$

where the approximation holds for $\text{DER} \ll 1$. Correspondingly, (42) becomes

$$\epsilon_{\text{rms},1}^2 \approx \frac{1}{K \cdot \text{DER}} \tag{45}$$

It follows from (45) that the mean-square relative estimation error is inversely proportional to K and DER. Hence, for given K the accuracy of the estimation deteriorates with decreasing DER; this problem can be circumvented by taking K inversely proportional to DER; for instance, to achieve $\epsilon_{\text{rms},1}^2 = 0.01$, we need $K \approx 100/\text{DER}$, so that on average $K \cdot \text{DER} = 100$ data unit errors occur.

10.2 Transmission until a fixed number of erroneous blocks occurs

We rewrite the DER estimate (29), obtained from the transmission of M blocks of N data units each, as

$$\widehat{\text{DER}} = \frac{1}{N} \cdot \frac{\bar{J}_\text{e}}{\bar{L}} \cdot \frac{1 + \epsilon_1}{1 + \epsilon_2} \approx \frac{1}{N} \cdot \frac{\bar{J}_\text{e}}{\bar{L}} \cdot (1 + \epsilon) \tag{46}$$

where $\bar{J}_\text{e} = \mathbb{E}[J_{\text{e},i}]$ is the expectation of the number of data unit errors in an *erroneous* block, $\bar{L} = \mathbb{E}[L_i]$,

$$\epsilon_1 = \frac{1}{M_\text{e}} \sum_{i=1}^{M_\text{e}} \frac{J_{\text{e},i} - \bar{J}_\text{e}}{\bar{J}_\text{e}}$$

$$\epsilon_2 = \frac{1}{M_\text{e}} \sum_{i=1}^{M_\text{e}} \frac{L_i - \bar{L}}{\bar{L}}$$

and $\epsilon = \epsilon_1 - \epsilon_2$. The approximation (46) is accurate when $\mathbb{E}[\epsilon_1^2] \ll 1$ and $\mathbb{E}[\epsilon_2^2] \ll 1$ (which is the case for large M_e). In the following, the approximation (46) is considered an equality.

We show that the estimate is (approximately) unbiased. Obviously, we have $\mathbb{E}[\epsilon_1] = \mathbb{E}[\epsilon_2] = 0$ which yields

$$\mathbb{E}[\widehat{\text{DER}}] = \frac{1}{N} \cdot \frac{\bar{J}_\text{e}}{\bar{L}}$$

The random variables L_i are geometrically distributed, i.e., for $l \geq 1$

$$\Pr[L_i = l] = (1 - p_\text{B})^{l-1} p_\text{B}$$

where p_B is the probability that a block of N data units contains at least one erroneous data unit. It follows from section 19 that $\bar{L} = 1/p_\text{B}$, yielding

$$\mathbb{E}[\widehat{\text{DER}}] = \frac{1}{N} \cdot \bar{J}_\text{e} p_\text{B}$$

In section 20 we show that $\bar{J}_\text{e} p_\text{B} = \mathbb{E}[J_m]$, which in turn equals $N \cdot \text{DER}$, the average number of erroneous data units in a block (averaged over all blocks, i.e., the correctly received blocks and the blocks containing erroneous data units). Hence, $\mathbb{E}[\widehat{\text{DER}}] = \text{DER}$, so the estimate is unbiased.

Let us compute the mean-square relative estimation error resulting from (46). As the relative estimation error is given by $\epsilon = \epsilon_1 - \epsilon_2$, we obtain

$$\begin{aligned}\epsilon^2_{\mathrm{rms},2} &= \frac{1}{M_\mathrm{e}}\left(\frac{\mathbb{E}[(J_{\mathrm{e},i}-\bar{J}_\mathrm{e})^2]}{\bar{J}_\mathrm{e}^2}+\frac{\mathbb{E}[(L_i-\bar{L})^2]}{\bar{L}^2}\right)\\ &= \frac{1}{M_\mathrm{e}}\left(\frac{\mathbb{E}[(J_{\mathrm{e},i}-\bar{J}_\mathrm{e})^2]}{\bar{J}_\mathrm{e}^2}+1-p_\mathrm{B}\right)\end{aligned} \tag{47}$$

where we made use of $\frac{\mathbb{E}[(L_i-\bar{L})^2]}{\bar{L}^2} = 1 - p_\mathrm{B}$, which follows from section 19.

Assuming $p_\mathrm{B} \ll 1$, the term $-p_\mathrm{B}$ in (47) can be safely neglected. In this case, $\epsilon^2_{\mathrm{rms},2}$ can be estimated as

$$\widehat{\epsilon^2_{\mathrm{rms},2}} = \frac{1}{M_\mathrm{e}} \cdot \frac{\frac{1}{M_\mathrm{e}}\sum_{m=1}^{M_\mathrm{e}-1} J^2_{\mathrm{e},i}}{\left(\frac{1}{M_\mathrm{e}}\sum_{m=1}^{M_\mathrm{e}-1} J_{\mathrm{e},i}\right)^2}$$

Special case: independent data unit errors

In the case of independent data unit errors, we can take $N = 1$ and $M = K$, in which case a block corresponds to a single data unit, and K such blocks are transmitted. It follows that $J_{\mathrm{e},i} = 1$ for all i, and $p_\mathrm{B} = \mathrm{DER}$. Consequently, (47) reduces to

$$\epsilon^2_{\mathrm{rms},2} = \frac{1-\mathrm{DER}}{M_\mathrm{e}} \approx \frac{1}{M_\mathrm{e}} \tag{48}$$

where the approximation holds for $\mathrm{DER} \ll 1$. Hence, the mean-square relative estimation error is essentially independent of DER; for instance, to achieve $\epsilon^2_{\mathrm{rms},2} = 0.01$, we need to simulate until $M_\mathrm{e} = 100$ data unit errors occur. This is in contrast with the case where a fixed number K of data units is transmitted, where according to (45) the accuracy gets worse with decreasing DER.

10.3 Comparison of error performance estimation accuracies

To facilitate the comparison of the mean-square relative estimation errors (42) and (47), which result from the transmission of a fixed number of data units and the transmission until a fixed number of block errors occurs, we make use of $\mathbb{E}[J_m] = \mathbb{E}[J_{\mathrm{e},i}]p_\mathrm{B}$ and $\mathbb{E}[J^2_m] = \mathbb{E}[J^2_{\mathrm{e},i}]p_\mathrm{B}$ (see section 20) to rewrite (42) as

$$\epsilon^2_{\mathrm{rms},1} = \frac{1}{M \cdot p_\mathrm{B}}\left(\frac{\mathbb{E}[(J_{\mathrm{e},i}-\bar{J}_\mathrm{e})^2]}{\bar{J}_\mathrm{e}^2}+1-p_\mathrm{B}\right) \tag{49}$$

Comparing (49) and (47), we conclude that both estimation methods yield the same accuracy, provided that the fixed number M of transmitted blocks is such that the expected number $M \cdot p_\mathrm{B}$ of block errors equals M_e. As p_B is not known beforehand, it is more practical to transmit until M_e erroneous blocks occur.

The mean-squared relative estimation errors (49) and (47) can be upper bounded as (see section 21)

$$\epsilon^2_{\mathrm{rms},1} \le \frac{1}{M \cdot p_\mathrm{B}}\left(\frac{(N+1)^2}{4N}-p_\mathrm{B}\right) \approx \frac{1}{M \cdot p_\mathrm{B}} \cdot \frac{(N+1)^2}{4N}$$

$$\epsilon^2_{\mathrm{rms},2} \le \frac{1}{M_\mathrm{e}}\left(\frac{(N+1)^2}{4N}-p_\mathrm{B}\right) \approx \frac{1}{M_\mathrm{e}} \cdot \frac{(N+1)^2}{4N}$$

where the approximation holds for $p_\mathrm{B} \ll 1$. The upper bound becomes an equality when $N = 1$ (in which case we have $J_{\mathrm{e},i} = 1$), or, for $N > 1$, when $J_{\mathrm{e},i}$ takes only the values 1 and N, with $\Pr[J_{\mathrm{e},i} = 1] = N/(N+1)$ and $\Pr[J_{\mathrm{e},i} = N] = 1/(N+1)$.

11 Some practical considerations

11.1 Transients

We consider the signal $y(t)$ from (8) which results from applying $x(t)$ to a filter with impulse response $h(t)$. We have available a sequence $\{x(kT_\mathrm{s})\}$of samples taken at a sufficiently high rate $1/T_\mathrm{s}$.

Samples of $y(t)$ at a rate $1/T_\mathrm{s}$ are obtained according to (11), which reduces to

$$y(kT_\mathrm{s}) = T_\mathrm{s} \sum_{m=-M_1}^{M_2} h(mT_\mathrm{s}) x(kT_\mathrm{s} - mT_\mathrm{s}) \tag{50}$$

when $h(t)$ has limited duration; when $h(t) = 0$ for $t \notin (-T_1, T_2)$, we have $M_1 = -\mathrm{ceil}\left(\frac{-T_1}{T_\mathrm{s}}\right) = \mathrm{floor}\left(\frac{T_1}{T_\mathrm{s}}\right)$ and $M_2 = \mathrm{floor}\left(\frac{T_1}{T_\mathrm{s}}\right)$. Hence, $y(kT_\mathrm{s})$ is a linear combination of only the samples $\{x(kT_\mathrm{s} - M_2T_\mathrm{s}), ..., x(kT_\mathrm{s} + M_1T_\mathrm{s})\}$. When only K samples $\{x(0), ..., x((K-1)T_\mathrm{s})\}$ are available, and we define $x(k) = 0$ for $k \notin \{0, ..., K-1\}$, the filter output $y(kT_\mathrm{s})$ exhibits the following behavior, for $K \geq M_1 + M_2 + 1$.

- We obtain $y(kT_\mathrm{s}) = 0$ for $k \leq -M_1 - 1$ and for $k \geq M_2 + K$, because all filter coefficients are multiplied with zeroes.
- A *transient* occurs for $-M_1 \leq k \leq M_2 - 1$ (which takes $M_1 + M_2$ samples) and for $K - M_1 \leq k \leq K - 1 + M_2$ (which takes $M_1 + M_2$ samples), because only part of the filter coefficients are multiplied with samples from $\{x(0), ..., x((K-1)T_\mathrm{s})\}$.
- The *steady state* occurs for $M_2 \leq k \leq K-1-M_1$ (i.e., the steady-state takes $K-M_1-M_2$ samples), during which all filter coefficients are multiplied with samples from $\{x(0), ..., x((K-1)T_\mathrm{s})\}$. During the steady state, the samples $y(kT_\mathrm{s})$ are the same as when an infinite sequence $\{x(kT_\mathrm{s})\}$ were applied to the filter.

Hence, the filter output consists of $K + M_1 + M_2$ samples (from $k = -M_1$ to $k = K - 1 + M_2$), of which the first $M_1 + M_2$ and the last $M_1 + M_2$ are transients, and the middle part of $K - M_1 - M_2$ samples (from $k = M_2$ to $k = K - 1 - M_1$) represents the steady-state.

When the output sampling rate equals N_s times the input sampling rate, we stack the N_s outputs $\left(y(qT_\mathrm{s}), ..., y\left(qT_\mathrm{s} + \frac{N_\mathrm{s}-1}{N_\mathrm{s}}T_\mathrm{s}\right)\right)$ into the vector $\mathbf{y}_q$, which can be expressed as

$$\mathbf{y}_q = \sum_{m=-M_1}^{M_2} \mathbf{h}_m x(qT_s - mT_s) \tag{51}$$

where $(\mathbf{h}_m)_r = T_\mathrm{s} h\left(mT_\mathrm{s} + \frac{r}{N_\mathrm{s}}T_s\right)$ for $r = 0, ..., N_\mathrm{s} - 1$. When the output sampling rate equals $1/N_\mathrm{s}$ times the input sampling rate, we obtain

$$y(kN_\mathrm{s}T_\mathrm{s}) = \sum_{q=-Q_1}^{Q_2} \mathbf{h}_q \mathbf{x}_{k-q} \tag{52}$$

with $(\mathbf{h}_q)_r = T_\mathrm{s} h(qN_\mathrm{s}T_\mathrm{s} + rT_\mathrm{s})$ and $(\mathbf{x}_k)_r = x(kN_\mathrm{s}T_\mathrm{s} - rT_\mathrm{s})$ for $r = 0, ..., N_\mathrm{s}-1$, where $Q_1 = -\mathrm{floor}\left(\frac{-T_1}{N_\mathrm{s}T_\mathrm{s}}\right) = \mathrm{ceil}\left(\frac{T_1}{N_\mathrm{s}T_\mathrm{s}}\right)$ and $Q_2 = \mathrm{floor}\left(\frac{T_2}{N_\mathrm{s}T_\mathrm{s}}\right)$. Considering that (51) and (52) have the same form as (50), the transient regions and the steady-state region can be identified in a similar way.

Note that a similar reasoning can be adopted in the case of a *nonlinear* operation with memory, i.e., when $y(t)$ is a *nonlinear* function of $x(t-u)$, with $u \in (-T_1, T_2)$. This occurs, for instance, in a system consisting of the cascade of a linear filter, a nonlinear memoryless amplifier, and a linear filter.

When determining the power, the power spectral density or the eye diagram of a signal, or the system error performance in terms of WER, SER, BER, ..., only the steady-state part of the signal should be taken into account.

11.2 Computation of convolution and (I)DFT

Software programs (such as MATLAB) contain efficient routines for computing the convolution of finite-length sequences, and the DTF and IDFT of a finite-length sequence. It is strongly recommended to make use of these built-in routines, rather than writing your own routines for performing these operations.

index	1	2	...	m	...
$\mathbf{a}_{\rm u}$	$a_{\rm u}(-K_1N_{\rm s})$	$a_{\rm u}(-K_1N_{\rm s}+1)$	...	$a_{\rm u}(-K_1N_{\rm s}+m-1)$	...
$\mathbf{h}_{\rm tr}$	$h_{\rm tr}(-L_1T_{\rm s})$	$h_{\rm tr}((-L_1+1)T_{\rm s})$	...	$h_{\rm tr}((-L_1+m-1)T_{\rm s})$	...
$\mathbf{h}_{\rm rec}$	$h_{\rm rec}(-M_1T_{\rm s})$	$h_{\rm rec}((-M_1+1)T_{\rm s})$	...	$h_{\rm rec}((-M_1+m-1)T_{\rm s})$	...
$\mathbf{s}$	$s((-K_1N_{\rm s}-L_1)T_{\rm s})$	$s((-K_1N_{\rm s}-L_1+1)T_{\rm s})$	...	$s((-K_1N_{\rm s}-L_1+m-1)T_{\rm s})$	...
$\mathbf{z}$	$z((-K_1N_{\rm s}-L_1-M_1)T_{\rm s})$	$z((-K_1N_{\rm s}-L_1-M_1+1)T_{\rm s})$	...	$z((-K_1N_{\rm s}-L_1-M_1+m-1)T_{\rm s})$	...

Table 1: storage of $\{a_{\rm u}(m)\}$, $\{h_{\rm tr}(mT_{\rm s})\}$, $\{h_{\rm rec}(mT_{\rm s})\}$, $\{s(mT_{\rm s})\}$ and $\{z(mT_{\rm s})\}$

11.3 Indices of matrix and vector elements

Several software programs (such as MATLAB) require the smallest index of elements in a matrix or vector to be equal to 1; the subsequent derivations can be easily adapted when this minimum value is 0 instead of 1. Hence, when storing a sequence $\{x(k), k=-K_1,...,K_2\}$ as a vector $\mathbf{x}$ with K_1+K_2+1 elements, we cannot just take $(\mathbf{x})_k = x(k)$. This problem is circumvented by storing a shifted version of $\{x(k)\}$, i.e., $(\mathbf{x})_k = x(k-K_1-1)$ for $k=1,...K_1+K_2+1$.

Let us consider $\{y(k)\}$ as the convolution of the sequences $\{x(k), k=-K_1,...,K_2\}$ and $\{h(k), k=-L_1,...,L_2\}$. Noting that $y(k)=0$ for $k \notin \{-K_1-L_1,...,K_2+L_2\}$, we need to compute only the elements $\{y(-K_1-L_1),...,y(K_2+L_2)\}$. The sequences $\{x(k)\}$, $\{h(k)\}$ and $\{y(k)\}$ are stored as vectors $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{y}$ given by

- $(\mathbf{x})_k = x(k-K_1-1)$ for $k=1,...K_1+K_2+1$
- $(\mathbf{h})_k = h(k-L_1-1)$ for $k=1,...L_1+L_2+1$
- $(\mathbf{y})_k = y(k-K_1-L_1-1)$ for $k=1,...K_1+L_1+K_2+L_2+1$ results from the convolution of $\mathbf{x}$ and $\mathbf{h}$

Hence, $y(k)$ is obtained as the $(k+K_1+L_1+1)$-th element from $\mathbf{y}$, for $k=-K_1-L_1,...,K_2+L_2$.

Let us reconsider the simulation of the communication system from section 5. We take the following assumptions:

- The data sequence $\{a(k), k=-K_1,...,K_2\}$ is upsampled by a factor $N_{\rm s}$, giving rise to $\{a_u(m), m=-K_1N_{\rm s},...,K_2N_{\rm s}\}$.
- The transmit pulse $h_{\rm tr}(t)$ is zero for $t \notin (-L_1T_{\rm s}, L_2T_s)$. Sampling the pulse at rate $1/T_{\rm s}=N_{\rm s}/T$, we obtain $\{h_{\rm tr}(mT_{\rm s}), m=-L_1,...,L_2\}$.
- The impulse response $h_{\rm rec}(t)$ of the receive filter is zero for $t \notin (-M_1T_{\rm s}, M_2T_s)$. Sampling the impulse response at rate $1/T_{\rm s}=N_{\rm s}/T$, we obtain $\{h_{\rm rec}(mT_{\rm s}), m=-M_1,...,M_2\}$.

Table 1 shows how the samples $\{a_{\rm u}(m)\}$, $\{h_{\rm tr}(mT_{\rm s})\}$ and $\{h_{\rm rec}(mT_{\rm s})\}$, along with the samples $\{s(mT_{\rm s})\}$ and $\{z(mT_{\rm s})\}$ at the output of the transmit filter and the receiver filter, are stored as vectors $\mathbf{a}_{\rm u}$ (containing $K_1N_{\rm s}+K_2N_{\rm s}+1$ elements), $\mathbf{h}_{\rm tr}$ (containing L_1+L_2+1 elements), $\mathbf{h}_{\rm rec}$ (containing M_1+M_2+1 elements), $\mathbf{s}$ (containing $K_1N_{\rm s}+K_2N_{\rm s}+L_1+L_2+1$ elements, including transients) and $\mathbf{z}$ (containing $K_1N_{\rm s}+L_1+M_1+K_2N_{\rm s}+L_2+M_2+1$ elements, including transients). Considering that

$$(\mathbf{z})_m = z\left((m-1-K_1N_s-L_1-M_1)T_{\rm s}\right)$$

the sequence $\{z(kT)\} = \{z(kN_{\rm s}T_{\rm s})\}$, obtained by decimating $\{z(mT_{\rm s})\}$ at the instants that correspond to the transmission instants of the symbols $\{a(k), k=-K_1,...,K_2\}$, is obtained as $z(kT) = (\mathbf{z})_{1+K_1N_{\rm s}+L_1+M_1+kN_{\rm s}}$ for $k=-K_1,...,K_2$.

12 Example: numerical computation of impulse response and transfer function

We consider a square-root cosine rolloff pulse $h(t)$ with rolloff factor β; the corresponding transfer function $H(f)$ is given by $H(f)=\sqrt{G(f)}$, with

$$G(f) = \begin{cases} T & |fT| \leq \frac{1-\beta}{2} \\ \frac{T}{2}\left(1-\sin\left(\frac{\pi}{\beta}\left(|fT|-\frac{1}{2}\right)\right)\right) & \frac{1-\beta}{2} \leq |fT| \leq \frac{1+\beta}{2} \\ 0 & |fT| \geq \frac{1+\beta}{2} \end{cases} \quad (53)$$

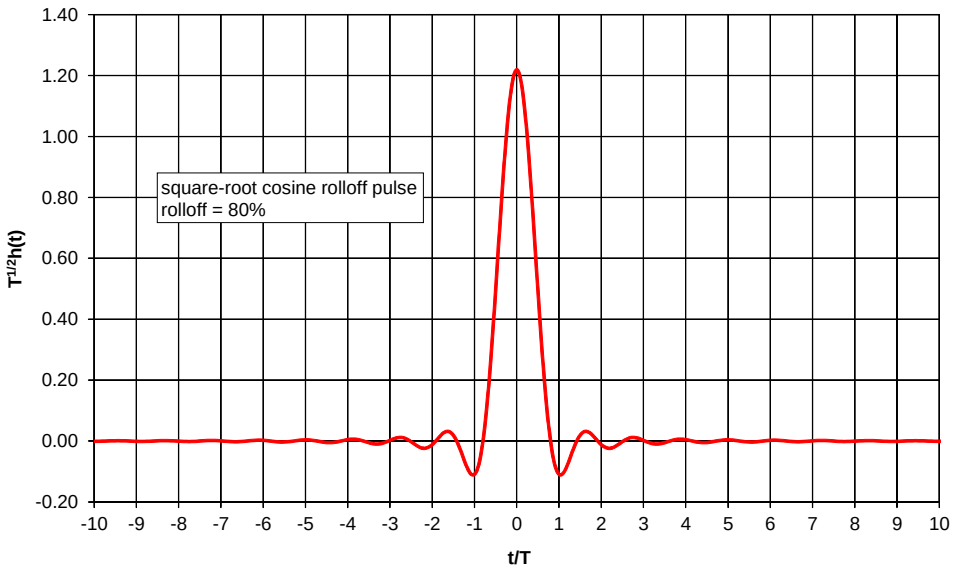

Figure 14: Pulse $h(t)$

It follows that $H(f) = 0$ for $|f| > B$, with $B = \frac{1+\beta}{2T}$.

We use an N-point IDFT to numerically compute, from the analytical expression for $H(f)$, N samples $h(kT_\mathrm{s})$ with $T_\mathrm{s} = T/N_\mathrm{s}$, according to the method outlined in section 2.4. The result is displayed in Fig. 14, for $N_\mathrm{s} = 16$, $N = 2048$ and $\beta = 0.8$, yielding $B = 0.9/T$; the resulting N samples cover $N/N_\mathrm{s} = 128$ symbol intervals (64 to the left and 64 to the right of the origin). It is clear from Fig. 14 that the effective duration of $h(t)$ is much less than $128T$. Hence, we will limit the duration of $h(t)$ to an interval $(-L_hT/2, L_hT/2)$, which corresponds to L_h symbol intervals; the corresponding truncated pulse $h_\mathrm{trunc}(t)$ is obtained as

$$h_\mathrm{trunc}(t) = \begin{cases} h(t) & |t| < L_hT/2 \\ 0 & \text{elsewhere} \end{cases} \tag{54}$$

To investigate the effect of this truncation, we use the method from section 2.5 to compute the transfer function $H_\mathrm{trunc}(f)$ by means of a DFT, using the L_hN_s samples $h_\mathrm{trunc}(kT/N_\mathrm{s}), k = -\frac{L_hN_\mathrm{s}}{2}, ..., \frac{L_hN_\mathrm{s}}{2} - 1$.

As $N_\mathrm{s} = 16$ and we again select $N = 2048$, we obtain $\{H_\mathrm{trunc}(nF)\}$ with a frequency step $F = N_\mathrm{s}/(NT)$, which corresponds to $N/N_\mathrm{s} = 128$ points per frequency interval of length $1/T$; hence, the N points cover the interval $(-8/T, 8/T)$. As $h_\mathrm{trunc}(kT/N_\mathrm{s})$ contains only L_hN_s samples, $N - L_hN_\mathrm{s}$ zeroes must be added to the truncated impulse response before the computation of the N-point DFT. Fig. 15 shows $|H_\mathrm{trunc}(f)|^2/T$ for $f > 0$ ($f = 0$ is a symmetry axis), along with $G(f)/T$ from (53); note that the scale on the ordinate is logarithmic (on a linear scale, $|H_\mathrm{trunc}(f)|^2/T$ and $G(f)/T$ virtually coincide). We observe that the truncation of $h(t)$ gives rise to frequency components extending beyond $(-B, B)$: whereas $G(f)$ consists of a single lobe extending from $-B$ to B, $|H_\mathrm{trunc}(f)|^2$ exhibits a main lobe which is slightly broader than the main lobe from $G(f)/T$, along with a large number of small sidelobes with a width of about $1/(L_hT)$; the width and magnitude of the sidelobes reduce with increasing L_h.

13 Example: power spectral density estimation

We consider a signal $x(t)$, given by

$$x(t) = \sum_m a(m)h(t - mT) \tag{55}$$

where $\{a(m)\}$ is a sequence of 2-PAM symbols with $a(m) \in \{-1, 1\}$, and $h(t)$ is a square-root cosine rolloff pulse with rolloff factor β: $H(f) = \sqrt{G(f)}$, with $G(f)$ given by (53); hence, $H(f) = 0$ for $|f| > B$, with $B = \frac{1+\beta}{2T}$. When simulating $x(t)$, we truncate $h(t)$ to an interval of L_h symbol intervals, to reduce the computational complexity: we replace in (55) $h(t)$ by $h_\mathrm{trunc}(t)$ from (54).

The PSD $S_x(f)$ of $x(t)$ from (55) can be obtained as

$$S_x(f) = \frac{1}{T}|H(f)|^2 \tag{56}$$

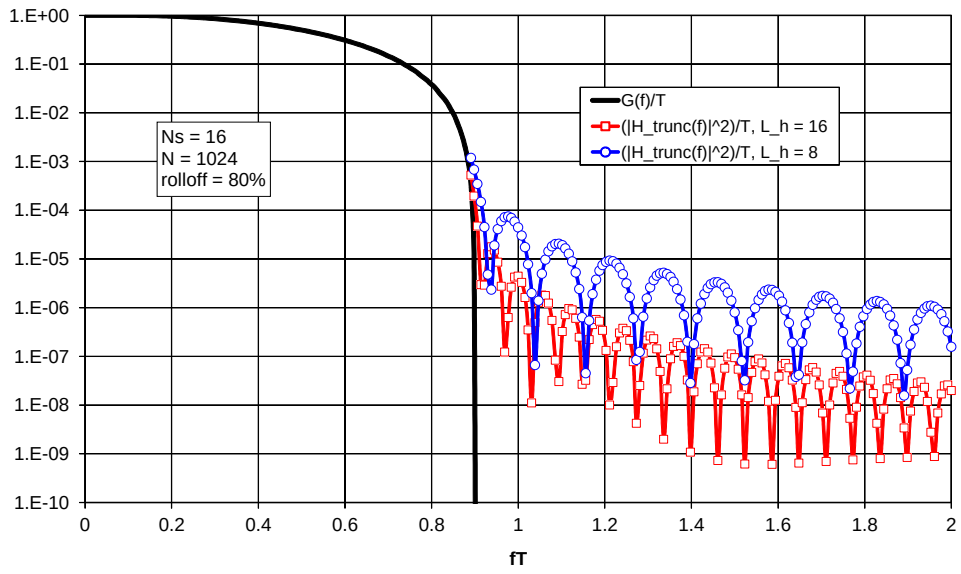

Figure 15: Comparison of $|H_{\text{trunc}}(f)|^2$ and $G(f)$

When $h(t)$ is truncated, the corresponding PSD is obtained by replacing in (56) $|H(f)|^2$ by $|H_{\text{trunc}}(f)|^2$.

We numerically compute (an estimate of) the PSD $S_x(f)$, using the method outlined in section 6.2. We consider two window functions, i.e., the rectangular window $w_{\text{rect}}(k)$ and the Hanning window $w_{\text{Hann}}(k)$, which are defined as

$$w_{\text{rect}}(k) = \begin{cases} C_{\text{rect}} & k = 0, ..., K-1 \\ 0 & \text{elsewhere} \end{cases}$$

$$w_{\text{Hann}}(k) = \begin{cases} C_{\text{Hann}} \left(1 - \cos\left(\frac{2\pi k}{K-1}\right)\right) & k = 0, ..., K-1 \\ 0 & \text{elsewhere} \end{cases}$$

with C_{rect} and C_{Hann} denoting the proportionality constants needed to ensure the normalization condition (23). The window functions and the corresponding squared magnitudes $|W(e^{j2\pi fT_s})|^2$ of their DTFT are displayed in Fig. 16 and Fig. 17, respectively. The Hanning window smoothly approaches zero near $k = 0$ and $k = K - 1$, which gives rise to a broader main lobe and much smaller sidelobes of $|W(e^{j2\pi fT_s})|^2$, compared to the rectangular window.

Fig. 18 shows the PSD estimate on a linear scale, for both the rectangular and the Hanning window, along with the exact PSD for the truncated pulse with $L_h = 16$. Both window function appear to provide similar PSD estimates, exhibiting some statistical fluctuation around the exact PSD. The statistical fluctuation could be reduced by increasing the number of blocks involved in the PSD estimation.

Fig. 19 and Fig. 20 display the exact PSD and the PSD estimates on a logarithmic scale, for $L_h = 128$ and $L_h = 16$, respectively; the part corresponding to $f < 0$ shows the exact PSD, while for $f > 0$ the PSD estimates are shown. These figures reveal the effect of window function of the PSD estimate:

- For $L_h = 128$, the exact PSD drops considerably with increasing $|fT|$ when $|fT|$ approaches $(1+\beta)/2 = 0.9$. Because of the smearing caused by the sidelobes of $|W(e^{j2\pi fT_s})|^2$, the PSD estimates drop more smoothly than the exact PSD . Because of its smaller sidelobes, the Hanning window produces a PSD estimate with stronger decay near $|fT| = 0.9$ than when a rectangular window is used.

- When the duration $h(t)$ is reduced to $L_p = 16$ symbol intervals, the exact PSD decays gradually to zero for $|fT| > 0.9$. The PSD estimate corresponding to the Hanning window follows this decay rather accurately, whereas the PSD estimate resulting from the rectangular window fails to follow the decay of the exact PSD.

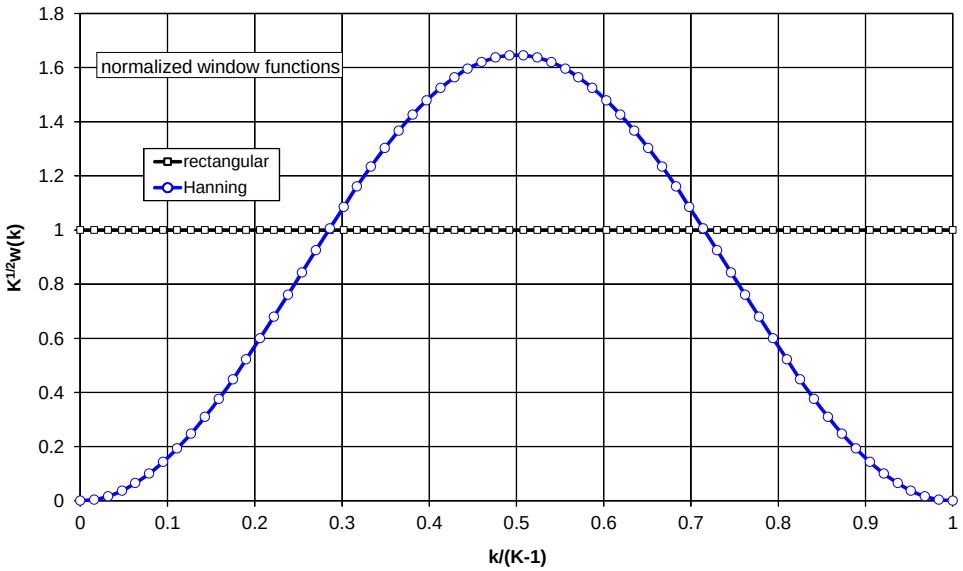

Figure 16: Rectangular window and Hanning window

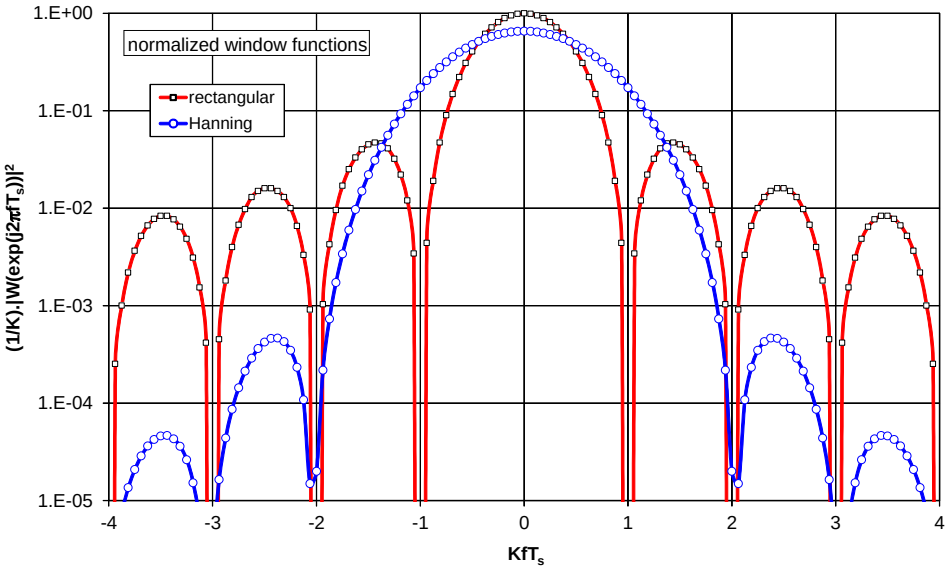

Figure 17: Comparison of $|W(e^{j2\pi fT_s})|^2$ for rectangular window and Hanning window

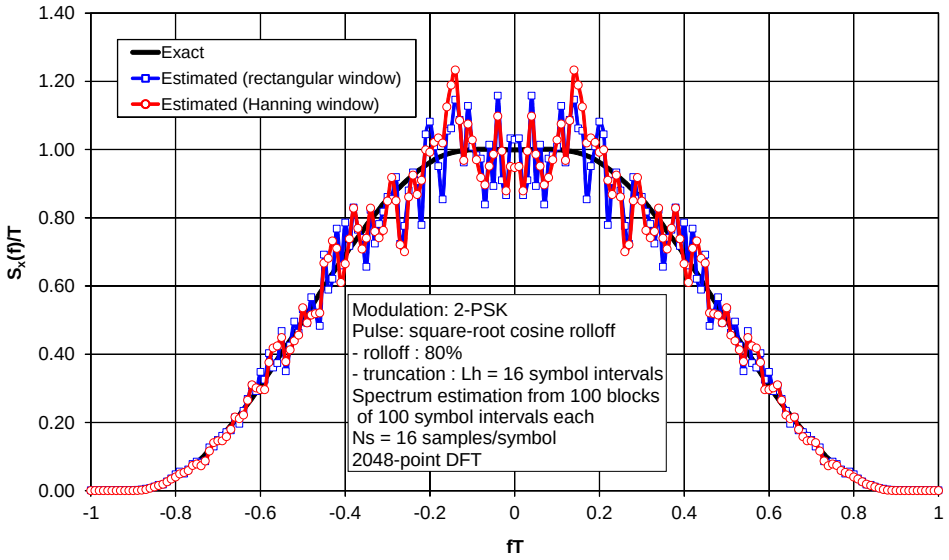

Figure 18: Exact PSD and PSD estimates (linear scale)

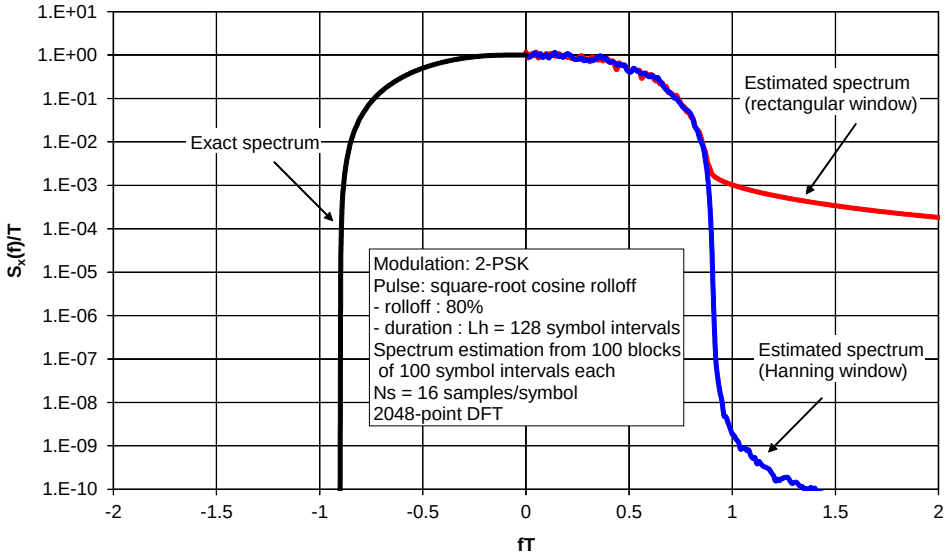

Figure 19: Exact PSD and PSD estimates for $L_h = 128$ (logarithmic scale)

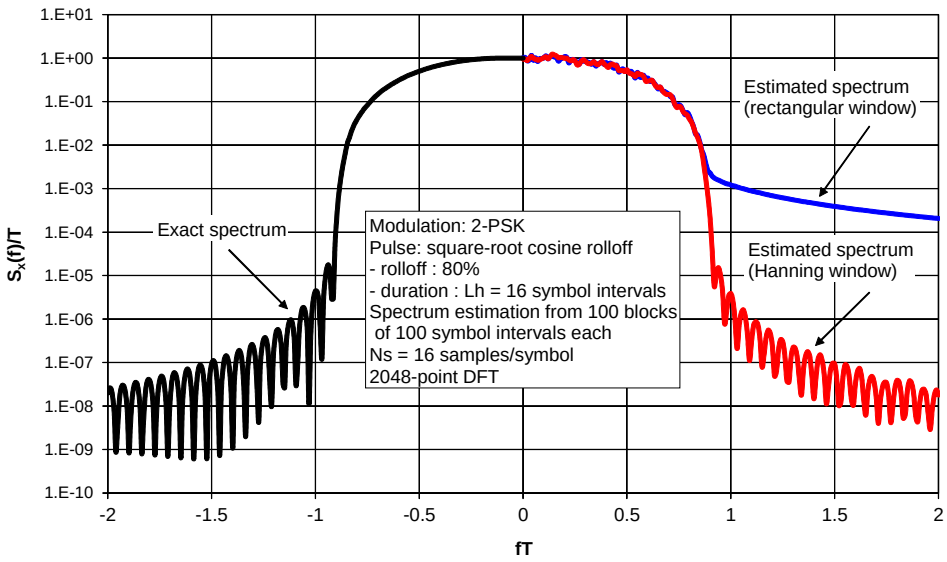

Figure 20: Exact PSD and PSD estimates for $L_h = 16$ (logarithmic scale)

14 Example: power estimation

We construct signal samples

$$x(kT_\mathrm{s}) = \frac{1}{\sqrt{L+1}} \sum_{l=0}^{L} v_{k-l} \tag{57}$$

where $\{v_k\}$ is a sequence of independent real-valued zero-mean Gaussian random variables with variance equal to 1. It is easily verified that

$$R_x(mT_\mathrm{s}) = \begin{cases} 1 - \frac{|m|}{L+1} & |m| \le L \\ 0 & |m| > L \end{cases} \tag{58}$$

which indicates that L is the correlation length of $\{x(kT_\mathrm{s})\}$; for $L = 0$, the samples $\{x(kT_\mathrm{s})\}$ are statistically independent. It follows from () that $P_x = R_x(0) = 1$, irrespective of L.

By means of a computer simulation, we generate K samples $\{x(kT_\mathrm{s}), k = 0, ..., K-1\}$ according to (57); next, we compute the power estimate $\hat{P}_x$ from (22) and the estimate $\hat{C}$ from (41).

- Fig. 21 shows C along with the estimate $\hat{C}$, as a function of the correlation length L. For real-valued zero-mean Gaussian $x(kT_\mathrm{s})$, it can be shown that the autocorrelation function of $e_k = |x(kT_\mathrm{s})|^2 - P_x$ is given by $R_e(mT_\mathrm{s}) = 2R_x^2(mT_\mathrm{s})$. Considering (38) and (58), it can be verified that

  $$C = \frac{4}{3}(L+1) + \frac{2}{3(L+1)}$$

  We observe that C increases with L, so that for given K the mean-square relative estimation error $\epsilon^2 \approx C/K$ increases with L. The estimate $\hat{C}$ from (41) with $(M, N) = (100, 100)$ exhibits some statistical fluctuation around C. The estimate $\hat{C}$ corresponding to $(M, N) = (10^4, 1)$ is the same as (36) for $K = 10^4$; this estimate does not depend on L, is accurate only for $L = 0$, and underestimates the mean-square relative estimation error $\epsilon^2 \approx C/K$.

- Fig. 22 shows $P_x = 1$, along with the power estimate $\hat{P}_x$ from (22) with $K = 10^4$. Also the 95% confidence interval, estimated using (41) with $(M, N) = (100, 100)$, is indicated. In agreement with Fig. 21, the length of the confidence interval tends to increase with L, indicating that the estimation accuracy for given K gets worse with growing L.

15 Example: error performance estimation

We consider the uncoded transmission of data symbols, belonging to a M_c-point constellation with Gray-mapping, over an AWGN channel. We estimate the BER performance by means of computer simulation.

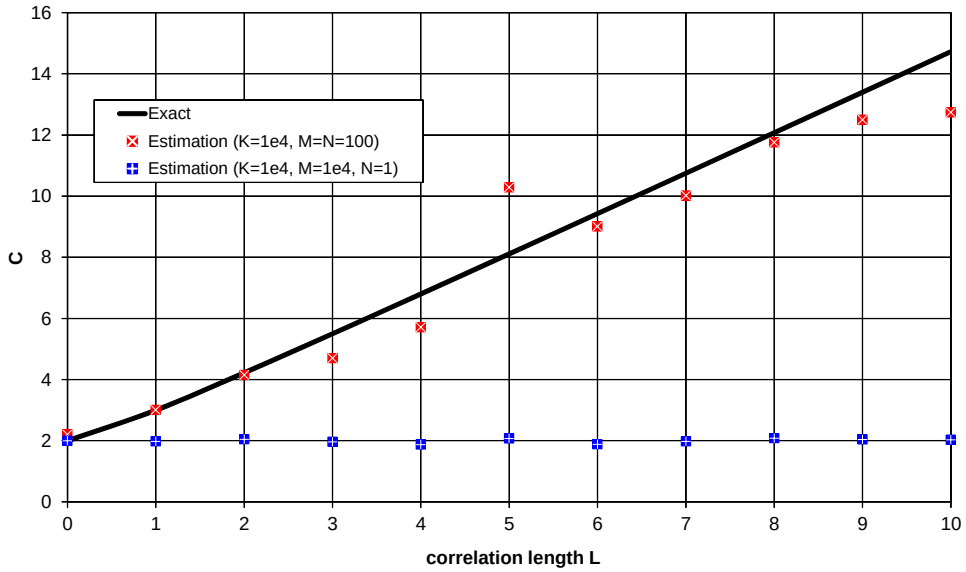

Figure 21: Estimation of C in the context of power estimation

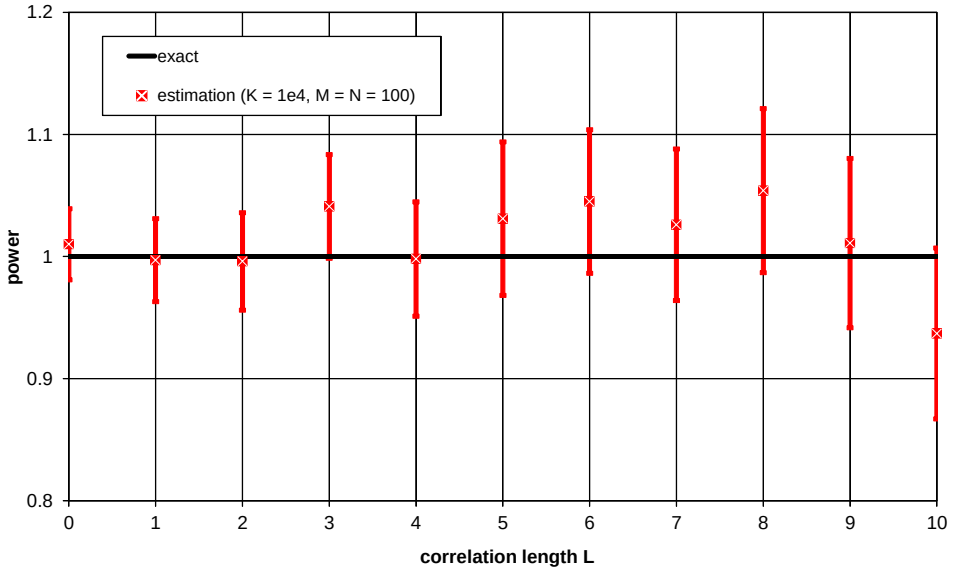

Figure 22: Power estimation

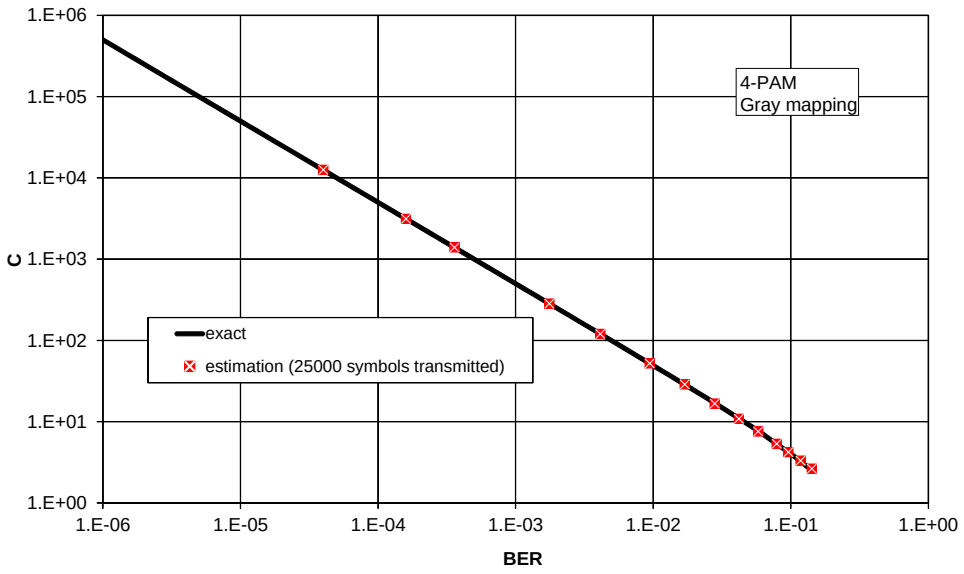

Figure 23: Exact and estimated values of C

Interpreting a bit and a symbol as a data unit and a block, respectively, we transmit a fixed number K of information bits, which corresponds to the transmission of $M = K/\log_2(M_c)$ symbols, each containing $N = \log_2(M_c)$ bits. The BER estimate $\widehat{\text{BER}}$ is given by (28), and the resulting estimation accuracy is determined by (42) and (43). Because of the Gray-mapping, we make the approximation (valid for large signal-to-noise ratio) that a detected symbol contains either no bit error (this occurs with probability equal to 1-SER) or a single bit error (this occurs with probability SER); the occurrence of multiple bit errors in the same symbol is neglected. This approximation gives rise to $\mathbb{E}[J_m^2] = \mathbb{E}[J_m] = \text{SER}$ and $\text{BER} = \text{SER}/\log_2(M_c)$; substitution in (43) yields

$$C = \frac{1}{\text{SER}} - 1 = \frac{1}{\log_2(M_c) \cdot \text{BER}} - 1 \tag{59}$$

In a computer simulation, C can be estimated according to (44); the corresponding (estimated) 95% confidence interval for given $\widehat{\text{BER}}$ and M is obtained as

$$\left(\widehat{\text{BER}} \left(1 - 1.96 \sqrt{\frac{\hat{C}}{M}} \right), \widehat{\text{BER}} \left(1 + 1.96 \sqrt{\frac{\hat{C}}{M}} \right) \right)$$

For 4-PAM with Gray-mapping, the theoretical error performance derived in section 22. With $M = 25{\cdot}10^3$ symbols transmitted, Fig. 23 shows that the estimate $\hat{C}$ and the expression (59) for C match very well. As indicated by (59), C is essentially inversely proportional to the BER for BER $\ll 1$; this implies that the estimation accuracy gets worse for decreasing BER. Fig. 24 displays the exact BER versus E_b/N_0, along with the simulation results for $M = 25 \cdot 10^3$ and the corresponding estimated 95% confidence intervals. Very high estimation accuracy (95% confidence intervals are small) is achieved for BER values down to about 10^{-3} (at BER $= 10^{-3}$, we have SER $= 2 \cdot 10^{-3}$, which yields on average 50 symbol errors when transmitting 25000 symbols, or, equivalently, 50 bit errors when transmitting 50000 bits), but the estimation quality deteriorates (95% confidence intervals get larger) for BER values below 10^{-3}.

16 Appendix: Proof of eq. (3)

The right-hand side of (3) is a periodic in f with period $1/T_s$and can therefore be expressed as a Fourier series:

$$\frac{1}{T_s} \sum_{n=-\infty}^{+\infty} X\left(f - \frac{n}{T_s}\right) = \sum_{l=-\infty}^{+\infty} C_l e^{j2\pi l f T_s}$$

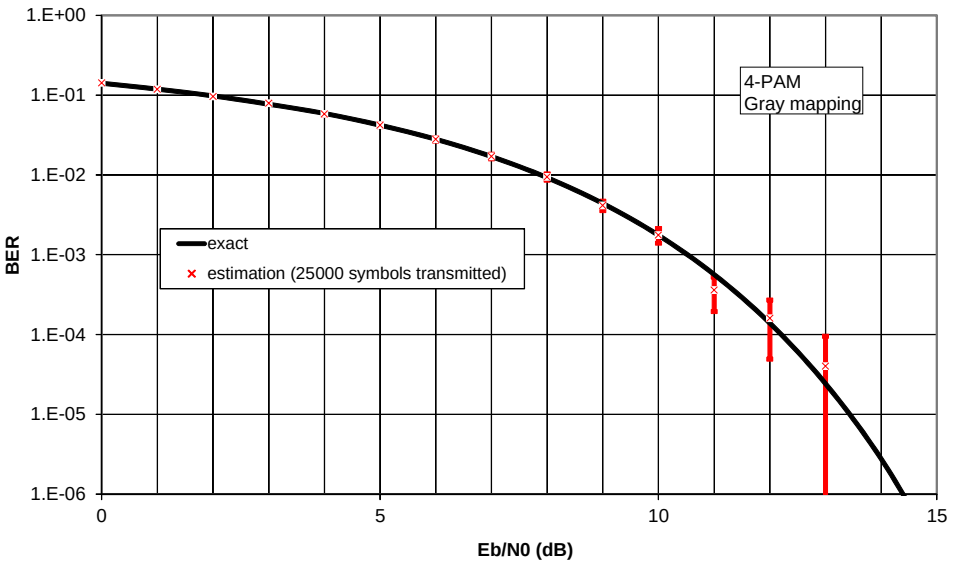

Figure 24: Exact and estimated values of BER

with

$$\begin{aligned} C_l &= T_\mathrm{s} \int_{-\frac{1}{2T_\mathrm{s}}}^{\frac{1}{2T_\mathrm{s}}} \left(\frac{1}{T_\mathrm{s}} \sum_{n=-\infty}^{+\infty} X\left(f - \frac{n}{T_\mathrm{s}}\right) \right) e^{-j2\pi l f T_\mathrm{s}} \mathrm{d}f \\ &= \sum_{n=-\infty}^{+\infty} \int_{-\frac{n}{T_\mathrm{s}}-\frac{1}{2T_\mathrm{s}}}^{-\frac{n}{T_\mathrm{s}}+\frac{1}{2T_\mathrm{s}}} X(\nu) e^{-j2\pi l \nu T_\mathrm{s}} \mathrm{d}f \\ &= \int_{-\infty}^{+\infty} X(\nu) e^{-j2\pi l \nu T_\mathrm{s}} \mathrm{d}f \\ &= x(-lT_s) \end{aligned}$$

This yields

$$\frac{1}{T_\mathrm{s}} \sum_{n=-\infty}^{+\infty} X\left(f - \frac{n}{T_\mathrm{s}}\right) = \sum_{l=-\infty}^{+\infty} x(-lT_s) e^{j2\pi l f T_\mathrm{s}}$$

Substituting in the above the summation index l by $-k$ yields (3).

17 Appendix: Proof that eqs. (15) and (16) are a DFT pair

Using (3) in (16), with $H(f)$ and $h(t)$ replacing $X(f)$ and $x(t)$, respectively, we obtain

$$\begin{aligned} X(n) &= \frac{1}{T_\mathrm{s}} \sum_{m=-\infty}^{+\infty} H\left(\frac{n}{NT_\mathrm{s}} + \frac{m}{T_\mathrm{s}}\right) \\ &= \sum_{k=-\infty}^{+\infty} h(kT_\mathrm{s}) e^{-j2\pi \frac{kn}{N}} \end{aligned}$$

Using the unique decomposition $k = qN + r$ with $r \in \{0, ..., N-1\}$ and taking (15) into account, we obtain

$$\begin{aligned} X(n) &= \sum_{q=-\infty}^{+\infty} \sum_{r=0}^{N-1} h(rT_\mathrm{s} + qNT_\mathrm{s}) e^{-j2\pi \frac{rn}{N}} \\ &= \sum_{r=0}^{N-1} x(r) e^{-j2\pi \frac{rn}{N}} \end{aligned}$$

Hence, $\{X(n), n = 0, ..., N-1\}$ is the DTFT of $\{x(k), k = 0, ..., N-1\}$. Therefore, $\{x(k), k = 0, ..., N-1\}$ is the IDTFT of $\{X(n), n = 0, ..., N-1\}$.

18 Appendix: Proof of eq. (24)

Considering that $w(k) = 0$ for $k \notin \{0, ..., K-1\}$ the DFT $Y^{(i)}(n)$ can be expressed as an infinite (rather than the usual finite) sum:

$$Y^{(i)}(n) = \sum_{k=-\infty}^{+\infty} x(kT_\mathrm{s} + iKT_\mathrm{s}) w(k) e^{-j2\pi \frac{kn}{N}} \tag{60}$$

This yields

$$T_\mathrm{s}\mathbb{E}\left[|Y^{(i)}(n)|^2\right] = T_\mathrm{s} \sum_{k_1,k_2=-\infty}^{+\infty} \mathbb{E}\left[x(k_1T_\mathrm{s} + iKT_\mathrm{s}) x^*(k_2T_\mathrm{s} + iKT_\mathrm{s})\right] w(k_1) w^*(k_2) e^{-j2\pi \frac{(k_1-k_2)n}{N}}$$

Introducing the autocorrelation function $R_x(u) = \mathbb{E}[x(t+u)x^*(t)]$ and setting $k_1 = m + k_2$, we obtain

$$T_\mathrm{s}\mathbb{E}\left[|Y^{(i)}(n)|^2\right] = T_\mathrm{s} \sum_{m=-\infty}^{+\infty} R_x(mT_\mathrm{s}) \rho_w(m) e^{-j2\pi \frac{mn}{N}} \tag{61}$$

where

$$\rho_w(m) = \sum_{k=-\infty}^{+\infty} w(k+m) w^*(k) \tag{62}$$

It follows from (62) that the DTFT of $\rho_w(m)$ is given by

$$\sum_{m=-\infty}^{+\infty} \rho_w(m) e^{-j2\pi m\nu T_\mathrm{s}} = |W(e^{j2\pi\nu T_s})|^2$$

Expressing in (61) $\rho_w(m)$ in terms of its DTFT, we have

$$T_\mathrm{s}\mathbb{E}\left[|Y^{(i)}(n)|^2\right] = T_\mathrm{s} \int_{1/T_\mathrm{s}} \left(T_\mathrm{s} \sum_{m=-\infty}^{+\infty} R_x(mT_\mathrm{s}) e^{-j2\pi m T_\mathrm{s}\left(\frac{n}{T_s} - \nu\right)} \right) |W(e^{j2\pi\nu T_s})|^2 \mathrm{d}\nu$$

Taking into account that $S_x(f)$ is the FT of $R_x(u)$, the expression between parentheses can be replaced by the periodic extension (with period $1/T_\mathrm{s}$) of $S_x(f)$, evaluated at $f = \frac{n}{T_s} - \nu$. This substitution gives rise to (24).

19 Appendix: Moments of the geometrical distribution

Consider a geometrically distributed random variable L with parameter p; L denotes the number of independent trials required to achieve a success, with p denoting the probability that a trial is successful. The probability mass function (pmf) of L is given by

$$\Pr[L = l] = \begin{cases} (1-p)^{l-1} p & l \geq 1 \\ 0 & \text{elsewhere} \end{cases} \tag{63}$$

The first moment $\mathbb{E}[L]$ is obtained as

$$\begin{aligned} \mathbb{E}[L] &= \sum_{l=1}^{\infty} l(1-p)^{l-1} p \\ &= p \frac{\mathrm{d}}{\mathrm{d}x} \left(\sum_{l=0}^{\infty} x^l \right)_{x=1-p} \\ &= p \frac{\mathrm{d}}{\mathrm{d}x} \left(\frac{1}{1-x} \right)_{x=1-p} \end{aligned}$$

$$= p \cdot \frac{1}{(1-x)^2}\bigg|_{x=1-p}$$
$$= \frac{1}{p}$$

The second moment $\mathbb{E}[L^2]$ is decomposed as $\mathbb{E}[L^2] = \mathbb{E}[L] + \mathbb{E}[L(L-1)]$. The first term is $\mathbb{E}[L] = 1/p$. The second term is

$$\begin{aligned}
\mathbb{E}[L(L-1)] &= \sum_{l=1}^{\infty} l(l-1)(1-p)^{l-1}p \\
&= (1-p)p\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(\sum_{l=0}^{\infty} x^l\right)_{x=1-p} \\
&= (1-p)p\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(\frac{1}{1-x}\right)_{x=1-p} \\
&= (1-p)p \cdot \frac{2}{(1-x)^3}\bigg|_{x=1-p} \\
&= \frac{2(1-p)}{p^2}
\end{aligned}$$

This yields

$$\mathbb{E}[L^2] = \frac{1}{p} + \frac{2(1-p)}{p^2} = \frac{2-p}{p^2}$$

The variance of L is obtained as

$$\mathbb{E}[L^2] - (\mathbb{E}[L])^2 = \frac{2-p}{p^2} - \frac{1}{p^2} = \frac{1-p}{p^2}$$

20 Appendix: Moments of number of errors in a block

Let J denote the number of errors in a block containing N data units. We denote by $P_j = \Pr[J = j]$ the corresponding pmf, with $j \in \{0, ..., N\}$. Let J_e denote the number of errors in an *erroneous* block. The corresponding pmf is denoted by $P_{\mathrm{e},j} = \Pr[J_\mathrm{e} = j]$, with $j \in \{1, ..., N\}$. Considering that a block is correct when $J = 0$ and erroneous when $J > 0$, the probability p_B that a block is erroneous is given by $p_\mathrm{B} = 1 - P_0$. Conditioning on whether or not a block is correct, we obtain

$$\begin{aligned}
P_j &= \Pr[J = j|J = 0]\Pr[J = 0] + \Pr[J = j|J > 0]\Pr[J > 0] \\
&= \begin{cases} 1 - p_\mathrm{B} & j = 0 \\ P_{\mathrm{e},j}p_\mathrm{B} & j > 0 \end{cases}
\end{aligned}$$

where we have made use of $\Pr[J = j|J = 0] = 0$ for $j > 0$, $\Pr[J = j|J > 0] = 0$ for $j = 0$, and $P_{\mathrm{e},j} = \Pr[J = j|J > 0]$ for $j > 0$.

Now we compute the expectation $\mathbb{E}[f(J)]$ of a function $f(J)$ of J. We get

$$\begin{aligned}
\mathbb{E}[f(J)] &= \sum_{j=0}^{\infty} f(j)P_j \\
&= (1-p_\mathrm{B})f(0) + p_\mathrm{B}\sum_{j=1}^{\infty} f(j)P_{\mathrm{e},j} \\
&= (1-p_\mathrm{B})f(0) + p_\mathrm{B}\mathbb{E}[f(J_\mathrm{e})]
\end{aligned}$$

Taking $f(J) = J$ and $f(J) = J^2$ (both for which $f(0) = 0$) yields $\mathbb{E}[J] = p_\mathrm{B}\mathbb{E}[J_\mathrm{e}]$ and $\mathbb{E}[J^2] = p_\mathrm{B}\mathbb{E}[J_\mathrm{e}^2]$.

21 Appendix: Upper bound on mean-square relative DER estimation error

Consider ϵ^2_rms from (47). For given $\bar{J}_\mathrm{e}$, $\mathbb{E}[(J_{\mathrm{e},i} - \bar{J}_\mathrm{e})^2]$ becomes maximum when $J_{\mathrm{e},i}$ takes only the extreme values 1 and N. Setting $\Pr[J_{\mathrm{e},i} = 1] = 1 - \alpha$ and $\Pr[J_{\mathrm{e},i} = N] = \alpha$, we have

$$\bar{J}_\mathrm{e} = 1 - \alpha + \alpha N$$

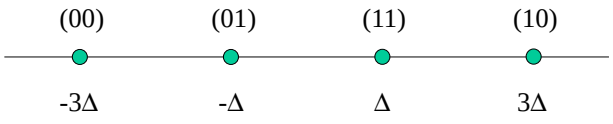

Figure 25: Gray-mapped 4-PAM constellation

which yields $\alpha = (\bar{J}_\mathrm{e} - 1)/(N-1)$. Hence,

$$\begin{aligned}\mathbb{E}[(J_{\mathrm{e},i} - \bar{J}_\mathrm{e})^2] &\leq (1 - \alpha + \alpha N^2) - \bar{J}_\mathrm{e}^2 \\ &= (\bar{J}_\mathrm{e} - 1)(N - \bar{J}_\mathrm{e})\end{aligned}$$

Dividing both sides by $\bar{J}_\mathrm{e}^2$ yields

$$\frac{\mathbb{E}[(J_{\mathrm{e},i} - \bar{J}_\mathrm{e})^2]}{\bar{J}_\mathrm{e}^2} \leq \frac{(\bar{J}_\mathrm{e} - 1)(N - \bar{J}_\mathrm{e})}{\bar{J}_\mathrm{e}^2}$$

The above upper bound is a quadratic function of $1/\bar{J}_\mathrm{e}$, which becomes maximum for $\bar{J}_\mathrm{e} = 2N/(N+1)$. The corresponding upper bound becomes

$$\frac{\mathbb{E}[(J_{\mathrm{e},i} - \bar{J}_\mathrm{e})^2]}{\bar{J}_\mathrm{e}^2} \leq \frac{(N-1)^2}{4N}$$

Substitution in (47) yields

$$\epsilon_\mathrm{rms}^2 \leq \frac{1}{M_\mathrm{e}} \left(\frac{(N+1)^2}{4N} - p_\mathrm{B} \right)$$

22 Appendix: Error performance of uncoded Gray-mapped 4-PAM transmission

We consider the transmission of a 4-PAM symbol a in additive white Gaussian noise w. The received signal is given by $y = a + w$, where $a \in \{-3\Delta, -\Delta, \Delta, 3\Delta\}$ with $E_\mathrm{s} = 5\Delta^2$ denoting the energy per symbol (the energy per bit is obtained as $E_b = E_s/2$), and w is real-valued zero-mean Gaussian with $\mathbb{E}[w^2] = \sigma^2 = N_0/2$. The two bits contained in a are Gray-mapped according to Fig. 25. Considering a real-valued zero-mean Gaussian random variable X with $\mathbb{E}[X^2] = 1$, we define $Q(x) = \Pr[X > x]$ as the complement of the cumulative distribution function of X. We introduce $Q_i = Q\left(i\frac{\Delta}{\sigma}\right)$. The following probabilities are obtained:

$$\Pr[\text{1 bit error}] = \frac{1}{2}\left(3Q_1 - 2Q_3 + Q_5\right) \approx \frac{3}{2}Q_1$$

$$\Pr[\text{2 bit errors}] = \frac{1}{2}\left(2Q_3 - Q_5\right) \approx Q_3$$

$$\mathrm{SER} = \Pr[\text{symbol error}] = \frac{3}{2}Q_1$$

$$\mathrm{BER} = \frac{1}{4}\left(3\mathrm{Q}_1 + 2\mathrm{Q}_3 - \mathrm{Q}_5\right) \approx \frac{3}{4}\mathrm{Q}_1$$

$$\begin{aligned}\Pr[\text{1 bit error}|\text{symbol error}] &= \frac{\Pr[\text{1 bit error}]}{\Pr[\text{symbol error}]} \\ &= 1 - \frac{2Q_3 - Q_5}{3Q_1} \approx 1\end{aligned}$$

$$\begin{aligned}\Pr[\text{2 bit errors}|\text{symbol error}] &= \frac{\Pr[\text{2 bit errors}]}{\Pr[\text{symbol error}]} \\ &= \frac{2Q_3 - Q_5}{3Q_1} \ll 1\end{aligned}$$

The above approximations result from $Q_5 \ll Q_3 \ll Q_1 \ll 1$, which holds for large $\frac{\Delta}{\sigma}$.