Projects in Wireless Communication
Lecture 1

Fredrik Tufvesson
Department of Electrical and Information Technology

Lund University, Sweden

Lund, Fall 2024

Outline

- Introduction to the course
- Basics of digital communications
- Discrete-time implementations
- Carrier transmission

Introduction

Lecturer and course responsible: Fredrik Tufvesson, E:2361A
7 scheduled lectures

Teaching assistant:
Yingjie Xu, E:2364
Computer help sessions

Email: firstname.lastname@eit.lth.se

Introduction

Ultimate goals for PWC:
1) Two computers should communicate via speaker/microphones
2) Two computers should communicate using software defined radios

The projects should be performed in groups of ONE or TWO students

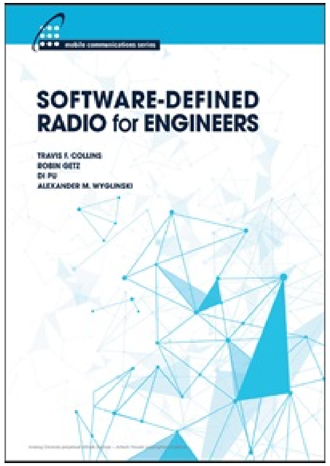
PWC System Simulation

In the first part of PWC we only work in software. For a passing grade you should solve three tasks:

1. A digital baseband BPSK system should be implemented in MATLAB and its performance should be measured and verified against theoretical results
$$P_e = \mathcal{Q}\left(\sqrt{d_{\min}^2 \frac{E_b}{N_0}}\right)$$
2. Later in PWC you will encounter physical passband signals at the input of the microphone. In the first part, we will provide each group with one such signal; the bits carried by the signals correspond to the ASCII code of a secret password. If you can decode the signals and provide me with the password, you have passed task 2.
3. Same as 2 but with OFDM transmission and convolutional code.

Recommended reading

Software-Defined Radio for Engineers,
by Travis F. Collins, Robin Getz, Di Pu, and Alexander M. Wyglinski, 2018, ISBN-13: 978-1-63081-457-1.
We will use some chapters in the second half of the course, and it covers many of the aspects in the first half as well.



mobile communications series

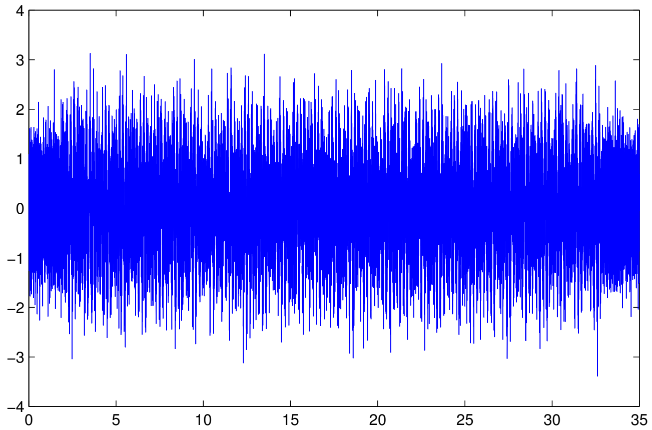
SOFTWARE-DEFINED
RADIO for ENGINEERS

TRAVIS F. COLLINS
ROBIN GETZ
DI PU
ALEXANDER M. WYGLINSKI

There is a free pdf of the book available, see
http://www.analog.com/en/education/education-library/software-defined-radio-for-engineers.html

Example

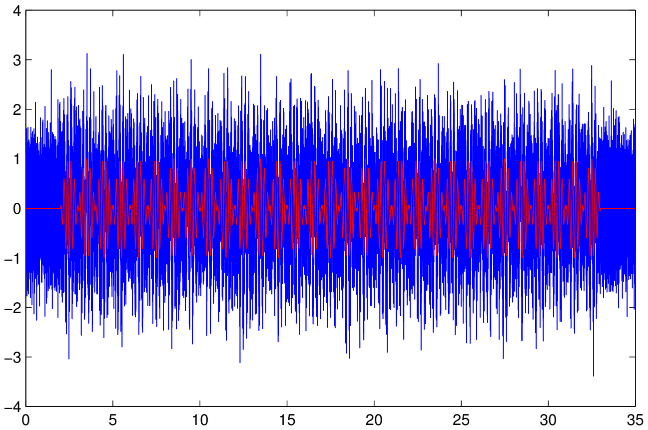
Assume that you receive the following noisy signal



You must remove the noise...

Example

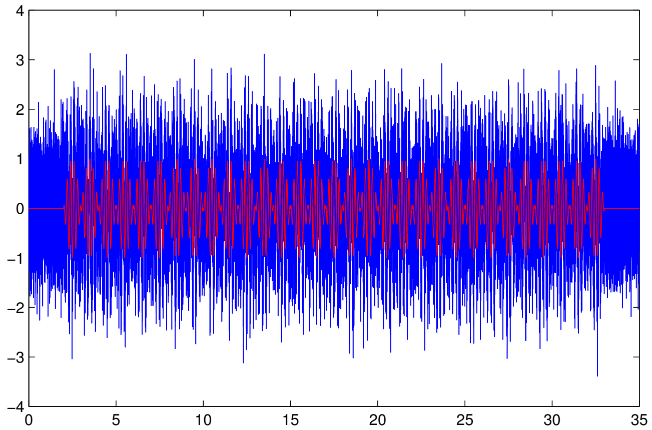
Assume that you receive the following noisy signal



You must remove the noise...Done!

Example

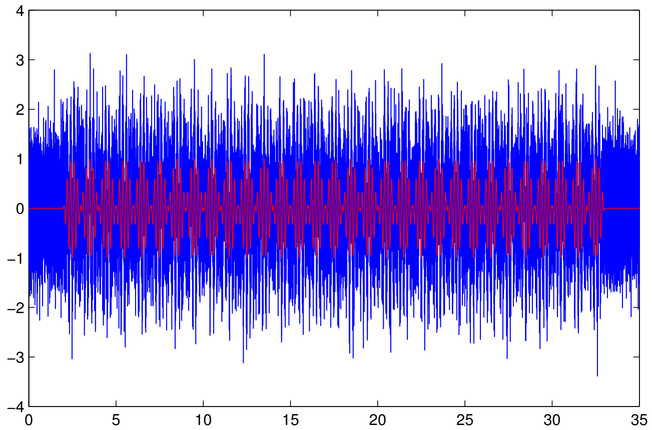
Assume that you receive the following noisy signal



You must remove the noise...Done!
Decode the bits:

Example

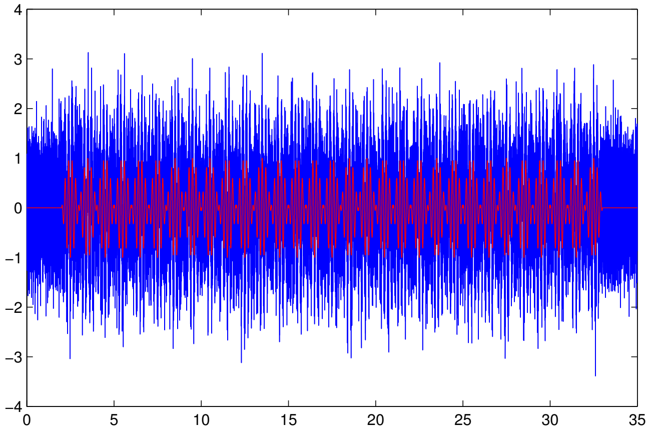
Assume that you receive the following noisy signal



You must remove the noise...Done!
Decode the bits: 1 1 1 0 0 0 0 0 1 1 0 1 0 0 1 1 1 0 1.....

Example

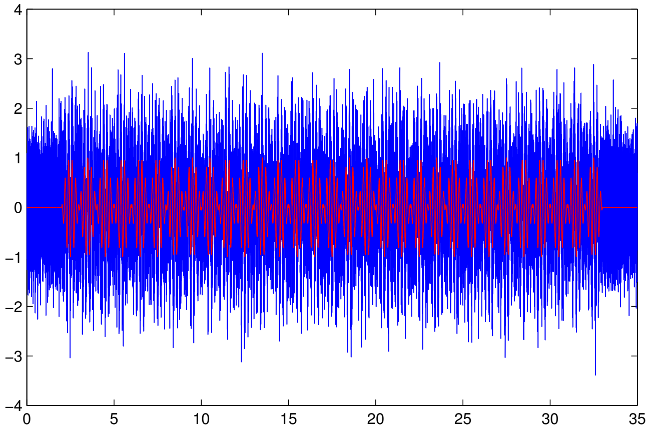
Assume that you receive the following noisy signal



You must remove the noise...Done!
Decode the bits: 1 1 1 0 0 0 0 0 1 1 0 1 0 0 1 1 1 0 1.....
Convert to ASCII:

Example

Assume that you receive the following noisy signal



You must remove the noise...Done!
Decode the bits: 1 1 1 0 0 0 0 0 1 1 0 1 0 0 1 1 1 0 1.....
Convert to ASCII: You have passed PWC1, congratulations......

Introduction

Formal descriptions of the tasks can be found online.

Basics of Digital Communications

This is a recall of baseband digital communications....

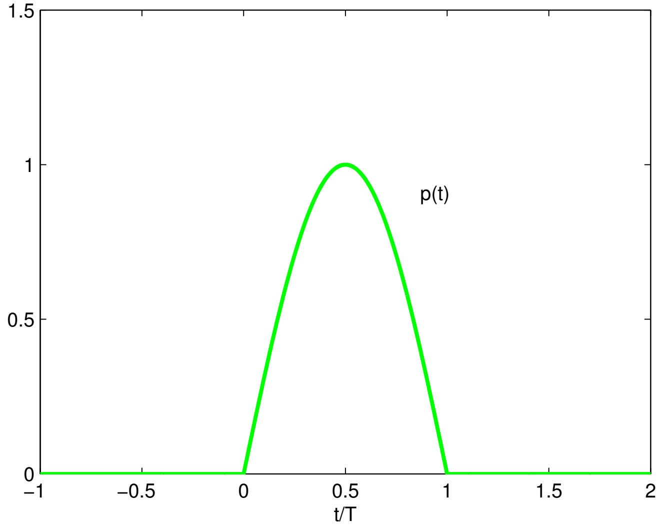
We need to transmit a bit sequence $\{u_k\} = 0111010.....$
Map to symbols $\{a_k\}$

$$\text{BPSK}: \quad a_k = \begin{cases} 1, & u_k = 0 \\ -1, & u_k = 1 \end{cases}$$

$$\text{QPSK}: \quad a_k = \begin{cases} 1, & u_{2k}u_{2k+1} = 00 \\ i, & u_{2k}u_{2k+1} = 01 \\ -1, & u_{2k}u_{2k+1} = 10 \\ -i, & u_{2k}u_{2k+1} = 11 \end{cases}$$

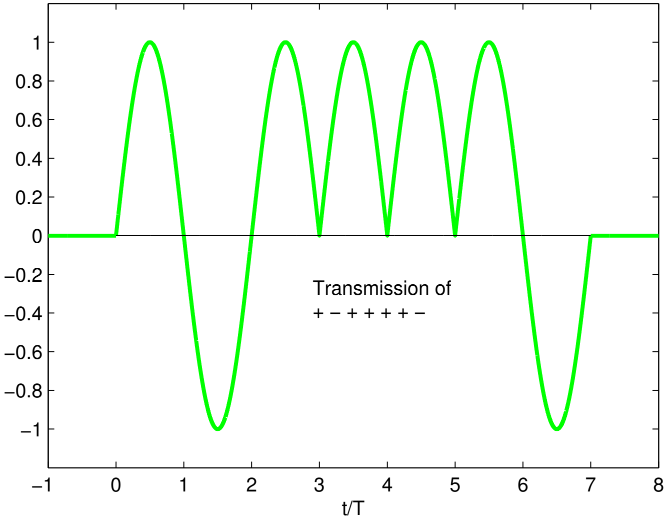
Basics of Digital Communications

Each symbol is carried by a base pulse $p(t)$ of length T, e.g. the half-cycle sinus



Basics of Digital Communications

So the transmission of bits 0 1 0 0 0 0 1 generates the pulse train $y(t)$



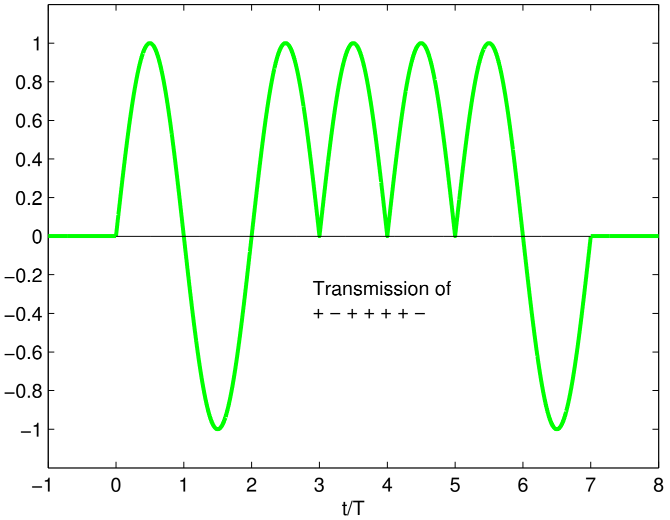
Mathematically we have

$$y(t) = \sum_k a_k p(t - kT_s)$$

Note that T_s is the symbol time while T is the duration of the pulse $p(t)$.

Basics of Digital Communications

So the transmission of bits 0 1 0 0 0 0 1 generates the pulse train $y(t)$



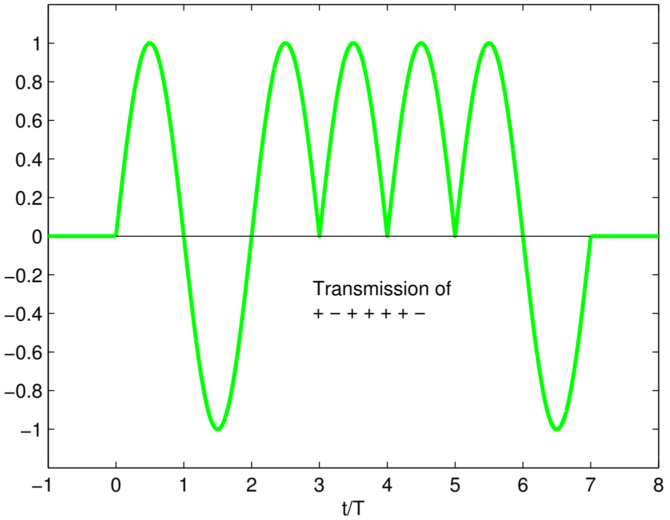
Mathematically we have

$$y(t) = \sum_k a_k p(t - kT_s)$$

Note that T_s is the symbol time while T is the duration of the pulse $p(t)$.
How does T and T_s relate in this example?

Basics of Digital Communications

So the transmission of bits 0 1 0 0 0 0 1 generates the pulse train $y(t)$



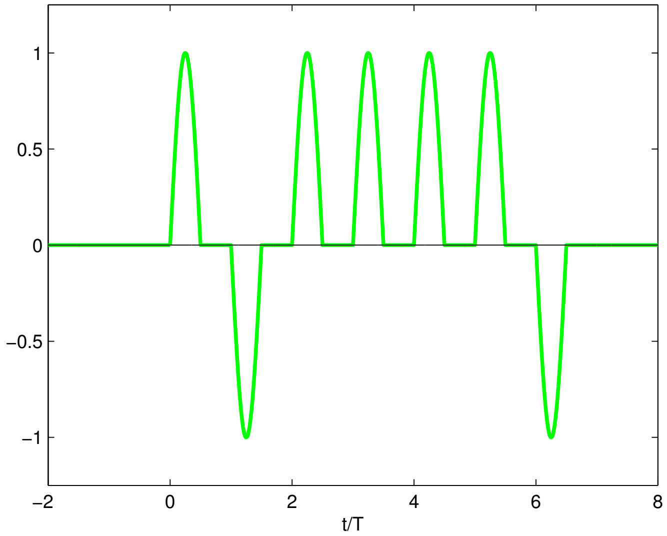
Mathematically we have

$$y(t) = \sum_k a_k p(t - kT_s)$$

Note that T_s is the symbol time while T is the duration of the pulse $p(t)$.
How does T and T_s relate in this example? $T = T_s$

Basics of Digital Communications

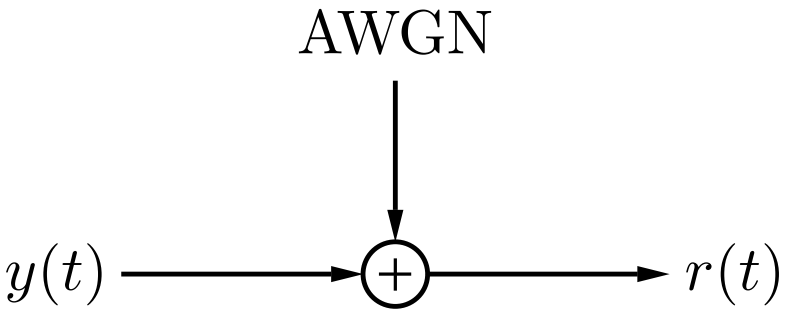
To avoid intersymbol interference one can use $T < T_s$



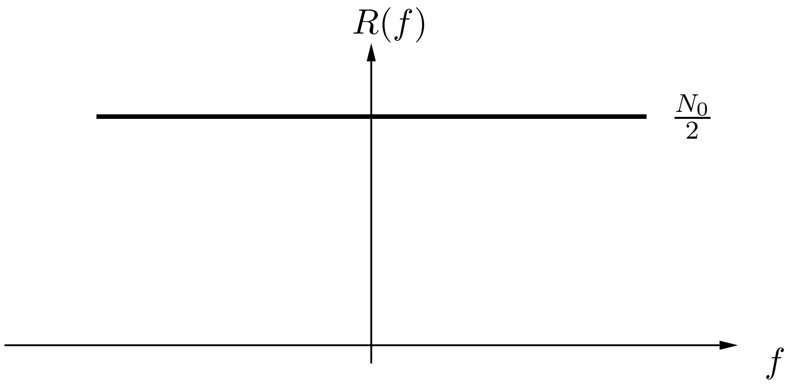
In this example we have $T = T_s/2$

Basics of Digital Communications

The channel model assumed in this review is a pure AWGN channel



Where the noise $n(t)$ satisfies $\mathcal{E}\{n^*(t)n(t+\tau)\} = \delta(\tau)N_0/2$; such a noise process must have power spectral density



Basics of Digital Communications

What does WGN look like?
Can we show an example?

Basics of Digital Communications

What does WGN look like?
Can we show an example?

Consider the power of the process

$$P = \int R(f) \mathrm{d}f$$

Basics of Digital Communications

What does WGN look like?
Can we show an example?

Consider the power of the process

$$P = \int R(f) \mathrm{d}f$$

$n(t)$ has infinite power!

Thus, not possible to show an example of WGN

Basics of Digital Communications

Explanation: Every signal we ever see in reality has been filtered by some low-pass filter.

Basics of Digital Communications

Mathematically, in what way should the receiver process the received signal $r(t)$.

In other words

$$\hat{\mathbf{a}} = ...?$$

Basics of Digital Communications

Mathematically, in what way should the receiver process the received signal $r(t)$.

Maximum-likelihood detection is the answer!

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} Prob\{r(t)|\mathbf{a}\}$$

Basics of Digital Communications

Mathematically, in what way should the receiver process the received signal $r(t)$.

Maximum-likelihood is equivalent to minimum Euclidean distance detection

$$\hat{\mathbf{a}} = \arg\min_{\mathbf{a}} \int_{-\infty}^{\infty} |r(t) - \sum_k a_k p(t - kT_s)|^2 dt$$

Basics of Digital Communications

To decode the (complex valued) signal $r(t)$, we pass $r(t)$ through a matched filter $z(t)$

$$z(t) = p(-t)$$

For symmetric pulses $p(t)$, we get

$$z(t) = p(t)$$

Let

$$\begin{aligned} x(t) &= r(t) \star p(t) \\ &= \sum_k a_k g(t - kT_s) + \eta(t) \end{aligned}$$

where $\eta(t)$ is $n(t) \star p(t)$ and $g(t) = p(t) \star z(t)$. Take samples every T_s seconds: $x_k = x(kT_s)$. Then

$$x_k = E_p a_k + \eta_k$$

where η_k is a complex Gaussian random variable with variance $E_p N_0$, that is $E_p N_0/2$ per dimension!

Basics of Digital Communications

Energy computations and error probability:

The energy per transmitted symbol E_s is given by: $E_s = \underbrace{\int p^2(t)\mathrm{d}t}_{E_p}$ while

the energy per transmitted bit is

$$E_b = \begin{cases} E_s, & \text{BPSK} \\ E_s/2, & \text{QPSK} \end{cases}$$

The physical minimum Euclidean distance is

$$D^2_{\min} = \begin{cases} 4E_p, & \text{BPSK} \\ 2E_p, & \text{QPSK} \end{cases}$$

In both cases we end up with a normalized distance $d^2_{\min} = 2$. The error probability is given by

$$P_e \approx \mathcal{Q}\left(\sqrt{2\frac{E_b}{N_0}}\right)$$

Discrete-Time Implementations

In a computer-based package such as Matlab, Python or C/C++, we cannot represent the signals $y(t)$ as continuous time signals. Hence we must work with sampled versions.

Let f_s be the sample rate in samples/second and N be the number of samples per symbol.

In PWC part 2, $f_s = 44100$ samples/second

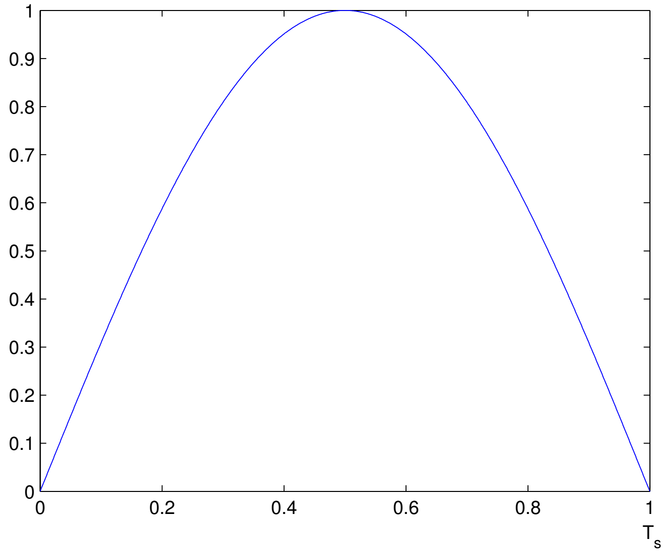
We get that $T_s = \frac{N}{f_s}$

The symbol rate becomes

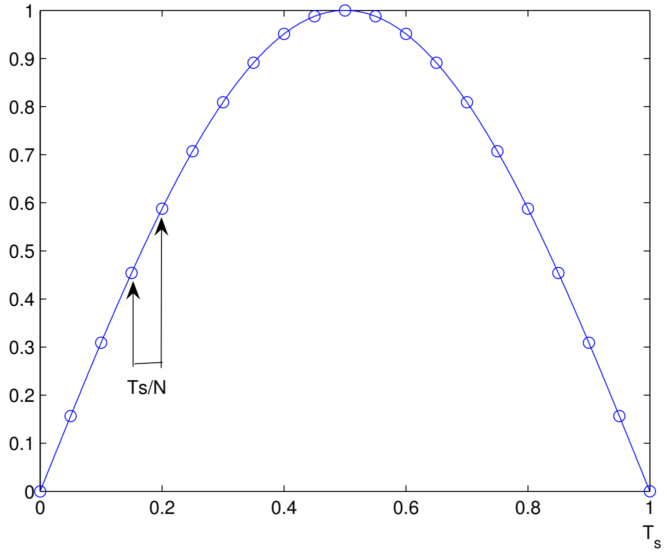
$$R_s = \frac{f_s}{N}$$

Discrete-Time Implementations

We must sample the base pulse $p(t)$.

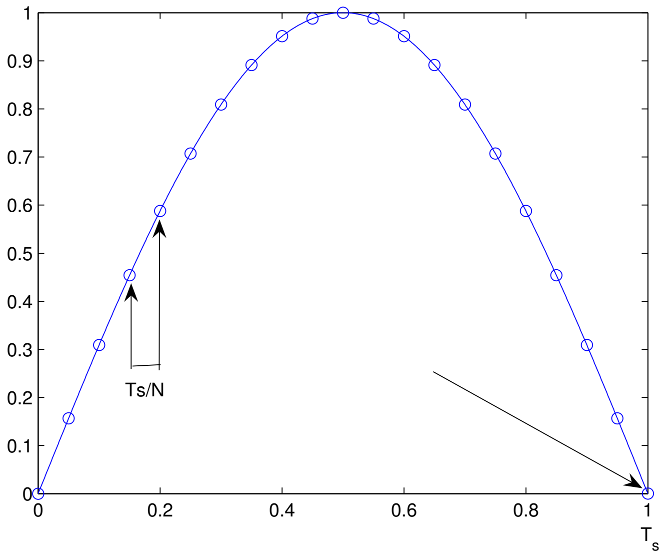


We must sample the base pulse $p(t)$. Assume a sample interval of T_s/N seconds



Discrete-Time Implementations

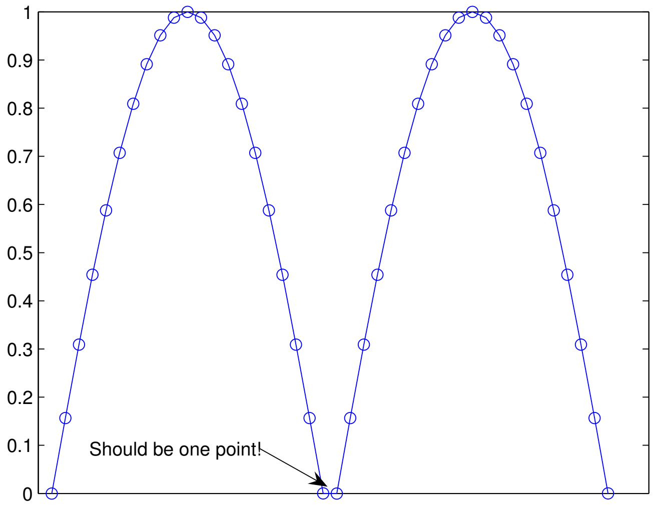
We must sample the base pulse $p(t)$. N+1 samples per symbol implies sample interval of T_s/N seconds



This is wrong!

Discrete-Time Implementations

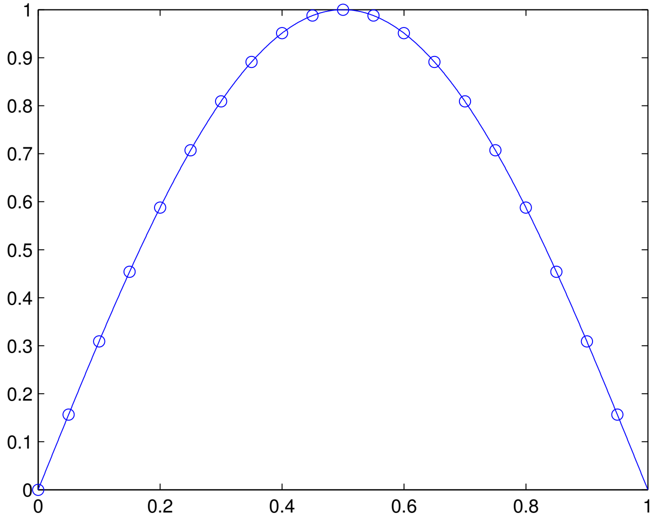
Explanation: Plot two consecutive pulses.



There should only be one point.

Discrete-Time Implementations

Correct sampling!

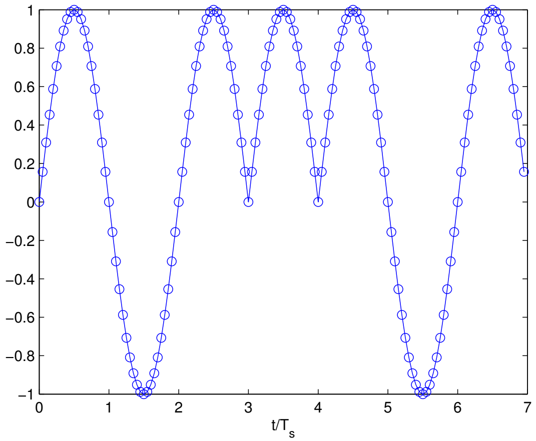


Represent the samples in a vector

$$\boldsymbol{p} = [0\ 0.159\ .309\ \ldots]$$

Discrete-Time Implementations

A sampled transmission signal of $+ - + + + - +$



Slightly harder mathematical representation. Let $\{b_k\}$ be a zero-padded version of $\{a_k\}$

$$\boldsymbol{b} = [a_1 \ \underbrace{00\ldots0}_{N-1} \ a_2 \ \underbrace{00\ldots0}_{N-1} \ a_3 \ \underbrace{00\ldots0}_{N-1} \ a_4 \ldots]$$

Then,

$$y_k = \sum_{\ell} b_\ell p_{k-\ell} \qquad \text{or simply } \boldsymbol{y} = \boldsymbol{b} \star \boldsymbol{p}$$

Discrete-Time Implementations

Convolutions in discrete-time:

A convolution of $x(t)$ and $y(t)$ in continuous time is carried out as

$$\int x(\tau)y(t-\tau)\mathrm{d}\tau \tag{1}$$

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be sampled version of $x(t)$ and $y(t)$; the sampling rate is f_s. The discrete time version of (1) is

$$\frac{1}{f_s}\sum_{\ell} x_\ell y_{k-\ell}$$

The discrete time convolution must be scaled by the sampling rate!. $1/f_s$ works as $\mathrm{d}\tau$ in (1).

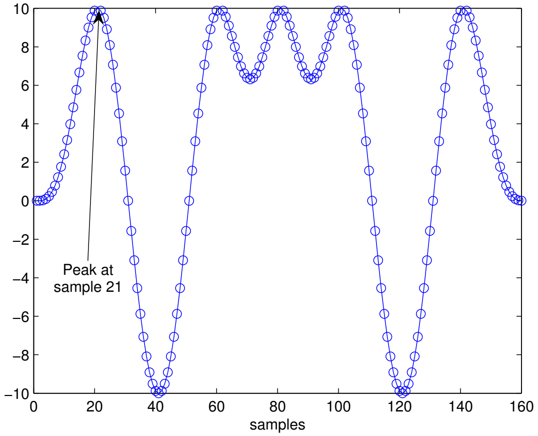
The energy of the pulse $p(t)$ must be approximated as

$$E_p = \frac{1}{f_s}\sum_k p_k^2$$

Discrete-Time Implementations

Matched filters in discrete-time:

The pulse train $\boldsymbol{p}$ should be filtered by a discrete-time matched filter. For symmetric pulses, we can take this mathched filter as $\boldsymbol{z} = \boldsymbol{p}$ where $\boldsymbol{p}$ includes the last sample!, i.e. the length of $\boldsymbol{p}$ is $N+1$. (This is however not crucial.) Then the output of the matched filter is ($N = 20$)



The number of samples in $\boldsymbol{y}$ is $N \times \text{number of symbols}$ and the length of the filter output is $N + N \times \text{number of symbols}$. The peak occurs at samples $1 + kN$, $k = 1, 2, 3 \ldots$.

Discrete-Time Implementations

If there is a guard band ($T < T_s$), then the pulse is not symmetric and we can not take $\boldsymbol{z} = \boldsymbol{p}$. We must then use

$$z_k = p_{N+2-k}, \quad k = 1 ... N + 1$$

It is still true that the peaks occur at samples $1 + kN$, $k = 1, 2, 3 ...$.

Discrete-Time Implementations

Implementation of discrete-time AWGN:

Until now we have constructed a modulation signal $\boldsymbol{y}$ in discrete time. We now seek a noise vector $\boldsymbol{n}$ to be added to $\boldsymbol{y}$ that represents continuous time AWGN (that has $\inf$ power).

We have that both the real and the imaginary parts of the samples of

$$\eta(t) = n(t) \star z(t)$$

are zero-mean and have variance $E_p N_0/2$.

In discrete-time, a sample of the filtered noise process is given by

$$\eta_k = \frac{1}{f_s} \sum_{\ell} n_\ell z_{k-\ell}$$

Discrete-Time Implementations

Assume that the variance of each n_k is σ_n^2. From probability theory it follows that η_k has variance $\sigma_n^2 \sum z_k^2/f_s^2$.

Since

$$\sigma_n^2 \sum_k z_k^2/f_s^2 = E_p \frac{N_0}{2}$$

we get that

$$\sigma^2 = E_p \frac{N_0}{2} \frac{f_s^2}{\sum_k z_k^2} = \frac{N_0}{2} f_s$$

Thus, The sampling rate affects the variance of the discrete time representation of continuous AWGN