Projects in Wireless Communication, Part 1
Lecture 1

Fredrik Rusek

Department of Electrical and Information Technology
Lund University, Sweden

Lund, March 2012
Outline

- Introduction to the course
- Basics of digital communications
- Discrete-time implementations
- Carrier transmission
Lecturer and course responsible: Fredrik Rusek, E:2377
4-5 scheduled lectures

Teaching assistant: Meifang Zhu,
4 scheduled laboratory lessons, time-slots to be decided today.

Email: {fredrik.rusek,meifang.zhu}@eit.lth.se
Introduction

Ultimate goal for PWC1 + PWC2:
Two computers should communicate via speaker/microphones

We aim at a file-transfer and/or a conversation via the keyboards
Some form of advanced system should be implemented, e.g. MIMO, OFDM, Turbo coding etc

The projects should be performed in groups of TWO students (HARD LIMIT)
In PWC 1 we only work in software. For a passing grade you must solve three tasks:

1. A digital baseband BPSK system should be implemented in C++ and its performance should be measured and verified against theoretical results

   \[ P_e = Q \left( \sqrt{\frac{d_{\text{min}}^2}{N_0}} \frac{E_b}{N_0} \right) \]

2. In PWC2 you will encounter physical passband signals at the input of the microphone. In PWC1, 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.
Schedule according to LTH.

L1  L2  L3  L4

Easter  break

Exams
Proposed schedule
Lecture 1-3 deals with Tasks 1 and 2 (deadline 1), while lecture 4-5 deals with task 3 (OFDM, deadline 3).

In addition, Meifang will have help sessions in the lab.
Example

Assume that you receive the following noisy signal

![Graph of a 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

![Noisy Signal Graph]

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 (soon) be found online.
This is a recall of baseband digital communications....

We need to transmit a bit sequence \( \{u_k\} = 0111010 \).....

Map to symbols \( \{a_k\} \)

BPSK: \[ a_k = \begin{cases} 
1, & u_k = 0 \\
-1, & u_k = 1 
\end{cases} \]

QPSK: \[ a_k = \begin{cases} 
1, & u_{2k}u_{2k+1} = 00 \\
i, & u_{2k}u_{2k+1} = 01 \\
-1, & u_{2k}u_{2k+1} = 10 \\
-i, & u_{2k}u_{2k+1} = 11 
\end{cases} \]
Each symbol is carried by a base pulse $p(t)$ of length $T$, e.g. the half-cycle sinus.
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 base pulse $p(t)$. 
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 base pulse \( p(t) \).

How does \( T \) and \( T_s \) relate in this example?
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 base pulse \( p(t) \).

How does \( T \) and \( T_s \) relate in this example? \( T = T_s \)
To avoid intersymbol interference one can use $T < T_s$.

In this example we have $T = T_s/2$. 
The channel model assumed in this review is a pure AWGN channel

\[ y(t) \rightarrow + \rightarrow r(t) \]

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

\[ R(f) \]

\[ \frac{N_0}{2} \]

\[ f \]
What does WGN look like?
Can we show an example?
What does WGN look like?  
Can we show an example?

Consider the power of the process

\[ P = \int R(f) df \]
What does WGN look like?
Can we show an example?

Consider the power of the process

\[ P = \int R(f) \, df \]

\( n(t) \) has infinite power!

Thus, not possible to show an example of WGN
Explanation: Every signal we ever see in reality has been filtered by some low-pass filter.
Mathematically, in what way should the receiver process the received signal $r(t)$.

In other words

$\hat{a} = \ldots$?
Mathematically, in what way should the receiver process the received signal \( r(t) \).

Maximum-likelihood detection is the answer!

\[
\hat{a} = \arg \max_a \text{Prob}\{r(t)|a\}
\]
Mathematically, in what way should the receiver process the received signal $r(t)$.

Maximum-likelihood is equivalent to minimum Euclidean distance detection

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

$$x(t) = r(t) \ast p(t) = \sum_{k} a_k g(t - kT_s) + \eta(t)$$

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

$$x_k = E_p a_k + \eta_k$$

where $\eta_k$ is a complex Gaussian random variable with variance $E_p N_0$, that is $E_p N_0 / 2$ per dimension!
Energy computations and error probability:

The energy per transmitted symbol $E_s$ is given by:

$$E_s = \int_{-E_p}^{E_p} p^2(t) dt$$

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

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

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

$$P_e \approx Q \left( \sqrt{2 \frac{E_b}{N_0}} \right)$$
Discrete-Time Implementations

In a computer-based package such as Matlab 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 PWC2, $f_s = 44100$ samples/second

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

The symbol rate becomes

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

This is wrong!
Discrete-Time Implementations

Explanation: Plot two consecutive pulses.

There should only be one point.
Correct sampling!

Represent the samples in a vector

\[ p = [0 \ 0.159 \ .309 \ldots] \]
Discrete-Time Implementations

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

\begin{center}
\includegraphics[width=\textwidth]{signal.png}
\end{center}

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

$$b = [a_1 \overbrace{00\ldots0}^{N-1} a_2 \overbrace{00\ldots0}^{N-1} a_3 \overbrace{00\ldots0}^{N-1} a_4 \ldots]$$

Then,

$$y_k = \sum_\ell b_\ell p_{k-\ell} \quad \text{or simply } y = b \ast p$$
**Discrete-Time Implementations**

**Convolutions in discrete-time:**

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

$$\int x(\tau)y(t - \tau)d\tau$$  (1)

Let $x$ and $y$ 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 $d\tau$ in (1).

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

$$E_p = \frac{1}{f_s} \sum_k p_k^2$$
Matched filters in discrete-time:

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

![Graph showing the matched filter output with a peak at sample 21.](image)

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

Fredrik Rusek: “PWC 1, Lecture 1” March 2012
If there is a guard band ($T < T_s$), then the pulse is not symmetric and we can not take $z = p$. We must then use

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

It is still true that the peaks occur at samples $1 + kN, \quad k = 1, 2, 3 \ldots$
Discrete-Time Implementations

**Implementation of discrete-time AWGN:**

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

We have that samples of

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

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

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

$$\eta_k = \frac{1}{f_s} \sum \ell n\ell z_k - \ell$$

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

Since

$$\sigma_n^2 \sum z_k^2 / f_s^2 = E_p \frac{N_0}{2}$$
Discrete-Time Implementations

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
Carrier Transmission

The transmitted signal is $y(t) = \sum_k a_k h(t - kT)$. What is the bandwidth? More generally, what is its Fourier transform?


### Table 2.3 Properties of the Fourier transform

<table>
<thead>
<tr>
<th>Property</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Linearity</td>
<td>$ax_1(t) + bx_2(t) \leftrightarrow aX_1(f) + bX_2(f)$</td>
</tr>
<tr>
<td>2. Inverse</td>
<td>$x(t) = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df$</td>
</tr>
<tr>
<td>3. Translation (time shift)</td>
<td>$x(t - t_0) \leftrightarrow X(f) e^{-j2\pi ft_0}$</td>
</tr>
<tr>
<td>4. Modulation (frequency shift)</td>
<td>$x(t) e^{j2\pi ft_0} \leftrightarrow X(f - f_0)$</td>
</tr>
<tr>
<td>5. Time scaling</td>
<td>$x(at) \leftrightarrow \frac{1}{</td>
</tr>
<tr>
<td>6. Differentiation in time</td>
<td>$\frac{d}{dt} x(t) \leftrightarrow j2\pi f X(f)$</td>
</tr>
<tr>
<td>7. Differentiation in frequency</td>
<td>$tx(t) \leftrightarrow \frac{1}{j2\pi f} X(f)$</td>
</tr>
<tr>
<td>8. Integration in time</td>
<td>$\int_{-\infty}^{\infty} x(\tau) d\tau \leftrightarrow \frac{1}{j\omega} X(f)$</td>
</tr>
<tr>
<td>9. Duality</td>
<td>$X(t) \leftrightarrow x(-f)$</td>
</tr>
<tr>
<td>10. Conjugate functions</td>
<td>$x^<em>(t) \leftrightarrow X^</em>(-f)$</td>
</tr>
<tr>
<td>11. Convolution in time</td>
<td>$x_1(t) * x_2(t) \leftrightarrow X_1(f)X_2(f)$</td>
</tr>
<tr>
<td>12. Multiplication in time</td>
<td>$x_1(t)x_2(t) \leftrightarrow X_1(f)X_2(f)$</td>
</tr>
<tr>
<td>13. Parseval’s formulas</td>
<td>$\int_{-\infty}^{\infty} x_1(t)x_2^<em>(t) dt = \int_{-\infty}^{\infty} X_1(f)X_2^</em>(f)f df$</td>
</tr>
<tr>
<td></td>
<td>or, when $x_1(t) = x_2(t)$,</td>
</tr>
<tr>
<td></td>
<td>$\int_{-\infty}^{\infty}</td>
</tr>
</tbody>
</table>
The baseband signal is $y(t) = \sum_k a_k h(t - kT)$. The power spectral density of the transmission is $\propto |H(f)|^2$.
Carrier Transmission

The baseband signal is \( y(t) = \sum_k a_k h(t - kT) \). The power spectral density of the transmission is \( \propto |H(f)|^2 \)

The carrier modulated signal is \( y_m(t) = y(t) \cos(2\pi t f_c) \)
The baseband signal is \( y(t) = \sum_k a_k h(t - kT) \). The power spectral density of the transmission is \( \propto |H(f)|^2 \).

The carrier modulated signal is \( y_m(t) = y(t) \cos(2\pi tf_c) \). But bandwidth gets twice as large!
Carrier Transmission

Where did the energy go?

Basic Fourier relations:

\[
\cos(2\pi f_c t) h(t) \leftrightarrow \frac{1}{2} H(f - f_c) + \frac{1}{2} H(f + f_c)
\]

\[
\sin(2\pi f_c t) h(t) \leftrightarrow \frac{1}{2} H(f - f_c) - \frac{1}{2} H(f + f_c)
\]
Carrier Transmission

Where did the energy go?

Basic Fourier relations:

\[ \cos(2\pi f_c t)h(t) \leftrightarrow \frac{1}{2}H(f - f_c) + \frac{1}{2}H(f + f_c) \]

\[ \sin(2\pi f_c t)h(t) \leftrightarrow \frac{1}{2}H(f - f_c) - \frac{1}{2}H(f + f_c) \]

The 1/2 factor corresponds to a 1/4 of the energy. Since there are two terms, 1/2 of the energy is preserved.
Carrier Transmission

Where did the energy go?

Basic Fourier relations:

\[
\cos(2\pi f_c t) h(t) \longleftrightarrow \frac{1}{2} H(f - f_c) + \frac{1}{2} H(f + f_c)
\]

\[
\sin(2\pi f_c t) h(t) \longleftrightarrow \frac{1}{2} H(f - f_c) - \frac{1}{2} H(f + f_c)
\]

The \(1/2\) factor corresponds to a \(1/4\) of the energy. Since there are two terms, \(1/2\) of the energy is preserved.

What about the increased bandwidth? Important
Assume two independent baseband transmissions
Carrier Transmission

Assume two independent baseband transmissions
After modulation with $\cos(2\pi tf_c)$ and $\sin(2\pi tf_c)$ we get

![Graph](image-url)
Assume two independent baseband transmissions. After demodulation with $\cos(2\pi f_c t)$ we get

The red spectras cancel out, thus, we can detect the blue independently from the red.

Similar for demodulation with $\sin(2\pi f_c t)$.  

Fredrik Rusek: “PWC 1, Lecture 1” March 2012
Carrier Transmission

The block diagram of the transmitter is

\[ y(t) = y_I(t) \cos(2\pi f_c t) - y_Q(t) \sin(2\pi f_c t) \]
The block diagram of the receiver is

\[
\begin{align*}
\cos(2\pi f_c t) & \rightarrow \text{LPF} \rightarrow y_i(t) + n_i(t) \\
-sin(2\pi f_c t) & \rightarrow \text{LPF} \rightarrow y_q(t) + n_q(t)
\end{align*}
\]

The in-phase and the quadrature components can be independently detected!
The LPF (low pass filters) can be taken as a matched filter to \( h(t) \)
The signals at both rails are baseband signals, and conventional processing follows: matched filter → sampling every $T_s$ second → decision unit

\[ r_I[k] = a_I[k] + n_I[k] \]
\[ r_Q[k] = a_Q[k] + n_Q[k] \]
What is a complex-valued symbol $1 + i$?

In QPSK, we transmit complex valued symbols. In one symbol interval, we have

$$y(t) = h(t) \cos(2\pi f_c t) - h(t) \sin(2\pi f_c t)$$
What is a complex-valued symbol $1 + i$?

In QPSK, we transmit complex valued symbols. In one symbol interval, we have

$$y(t) = h(t) \cos(2\pi f_c t) - h(t) \sin(2\pi f_c t)$$

Real part goes here and imaginary here
We can alternatively express the signal \( y(t) \) as

\[
y(t) = y_I(t) \cos(2\pi f_c t) - y_Q(t) \sin(2\pi f_c t)
\]
\[
= e(t) \cos(2\pi f_c t + \theta(t))
\]

where \( e(t) \) is the envelope and \( \theta(t) \) is the phase.

For QPSK, \( e(t) = \sqrt{2} h(t) \) and \( \theta(t) \in \{0, \pi/2, \pi, 2\pi/2\} \)

We can further manipulate \( y(t) \) into

\[
y(t) = \text{Re}\{(y_I(t) + iy_Q(t))e^{2\pi f_c t}\}
\]
\[
= \text{Re}\{\tilde{y}(t)e^{i2\pi f_c t}\}
\]

where

\[
\tilde{y}(t) = y_I(t) + iy_Q(t)
\]
Carrier Transmission

Example

Assume that we have two bits to transmit, say $+1$ and $-1$. 
Example

Assume that we have two bits to transmit, say +1 and -1.

We can either do this as

\[ y(t) = h(t) \cos(2\pi f_c t) - (-h(t)) \sin(2\pi f_c t) \]
Example

Assume that we have two bits to transmit, say $+1$ and $-1$.

We can either do this as

$$y(t) = h(t) \cos(2\pi f_c t) - (-h(t)) \sin(2\pi f_c t)$$

or as

$$y(t) = \sqrt{2} h(t) \cos(2\pi f_c t + 3\pi/2)$$
Example

Assume that we have two bits to transmit, say +1 and -1.

We can either do this as

\[ y(t) = h(t) \cos(2\pi f_c t) - (-h(t)) \sin(2\pi f_c t) \]

or as

\[ y(t) = \sqrt{2} h(t) \cos(2\pi f_c t + 3\pi/2) \]

or as

\[ y(t) = \sqrt{2} \text{Re}\{(1 - i)h(t)e^{2\pi f_c t}\} \]
In the last representation, we can change the receiver processing into

\[ r_I[k] = a_I[k] + n_I[k] \]
\[ r_Q[k] = a_Q[k] + n_Q[k] \]