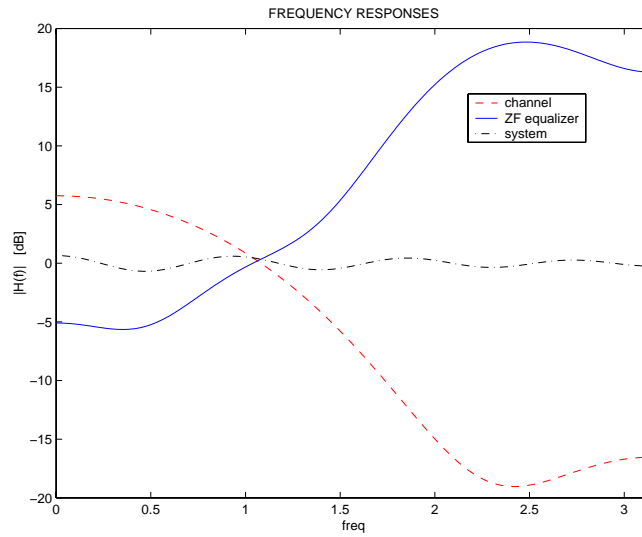


S-72.260

Laboratory Works in Radiocommunications

Laboratory Work #2

RECEIVER



Version 0.11

## Version history

Date	Version	Changes
1.9.2001/JSa	0.11	Some errors corrected

Prerequisites: S-72.232 (or S-88.211), knowledge of MATLAB.

See the course home page for more information.

Student laboratory is in the room E306.

Good luck!

# RECEIVER FOR BANDLIMITED CHANNEL

## 1 Introduction

In this laboratory work we concentrate on receiver for a bandlimited channel that causes intersymbol interference (ISI) to the communications signal. The purpose is to implement a receiver in MATLAB and detect data transmitted through such a channel.

The main blocks of the receiver are channel estimation and equalization units. All practical communication systems also use channel coding, but we exclude it in this laboratory work, partly because it is a theoretical topic that is not easily fitted into laboratory work format. Furthermore, MATLAB is not the ideal tool for processing tens of millions of symbols required in error-control coding simulations.

Channel estimation can be considered a part of system identification and/or estimation theory. Estimation theory is covered in detail in [Kay93], and partly in [Koi01] and [Paj00]. In this laboratory work we only need a tiny fragment of that material, to be found in [Puk00] and as a preliminary problem.

Channel equalization is also a wide-ranging topic. In this work we treat it as a linear filtering problem. Necessary background material can be found in [Häg01] or [Laa01].

The purpose of this introductory text is not to reproduce material that can be found in hundreds of books, scientific papers, and lecture notes. Instead, certain key points, important in this laboratory work, are discussed briefly and appropriate references are given.

## 2 A bandlimited communication system (abridged)

The simplest<sup>1</sup> version of a receiver for AWGN channel consists of a matched filter and a decision device. If the channel induces ISI to the signal, it has to be removed

---

<sup>1</sup> Assume phase synchronization, symbol timing etc are known.

before making symbol decisions. Thus, we introduce an equalizer to the receiver structure. The equalizer, however, needs to know the channel impulse response to be able to cancel its effect. Hence, we also need a channel estimator in our receiver. These blocks are briefly reviewed in the next two sections.

### 3 Channel estimation

Many types of channel estimators have been reported in literature. We can separate them in two categories: blind estimators and non-blind estimators.

Blind estimators - which are not considered in this lab work - extract the channel estimate from the received signal without any known symbols inserted in the transmitted signal. The obvious advantage is that bandwidth is saved. However, given a fixed number of basic computation operations their performance is, in general, worse than non-blind estimators'.

Non-blind estimators use *a priori* known symbols to compute an estimate of the channel IR. The known symbols are typically called training sequence, or pre/mid/postamble. For instance, in the GSM/GPRS system each transmitted burst contains 26 training symbols.

When channel noise is white Gaussian noise it can be shown that the minimum-variance unbiased estimator is linear, and the problem simplifies considerably. In this special case we can utilize a simple least squares (LS) type channel estimator for optimum performance. Reference [Puk00, pp. 3-5] and the preliminary problem P1 include the necessary background material you need about LS estimation in this laboratory work.

## 4 Zero-forcing equalizer, discrete-time case

### 4.1 General

Many types of equalizers can be used to cancel ISI of the received signal. The Viterbi equalizer<sup>2</sup> provides maximum-likelihood sequence out of all possible transmitted sequences, and it is thus optimal in that sense. A less-known algorithm, which minimizes the symbol-error probability, is the maximum-a-posteriori (MAP) algorithm [Pro95]. The Viterbi detector and the MAP detector are the optimal<sup>3</sup> detectors in error-minimizing sense but so far their computational complexities have proven to be prohibitive in applications with fast data rate.

Decision-feedback equalizer (DFE) minimizes mean-square error or peak distortion and provides reasonably good performance with less complexity than the optimal detectors. It is a nonlinear equalizer structure, and has better performance than a linear equalizer with the same optimizing criterion (MSE or ZF).

Linear MMSE or ZF equalizers are the simplest types of equalizers. In the following we concentrate on the ZF equalizer.

### 4.2 ZF solution for the equalizer coefficients

Continuous-time ZF solution of the equalizer transfer function has been discussed in [Häg01] and [Laa01], see the material therein. We next focus on the equivalent discrete-time ZF equalizer, which is the one used in practice.

Assumption: Channel impulse response  $\mathbf{h} = [h_{-L} \ h_{-L+1} \ \cdots \ h_L]^T$ , or its estimate, is known and it has odd length. We want to find the odd-length equalizer impulse response  $\mathbf{c} = [c_{-N} \ c_{-N+1} \ \cdots \ c_{N-1} \ c_N]^T$  such that its convolution sum with  $\mathbf{h}$  produces an impulse. Thus  $\mathbf{c}$  must satisfy  $h(n) * c(n) = \delta(n)$ . Written out as a group of linear equations we get

---

<sup>2</sup> In fact, the Viterbi algorithm gives "only" a very good approximation of the ML sequence.

<sup>3</sup> It is assumed that noise is white with Gaussian pdf and channel impulse response is known perfectly.

$$q_m = \sum_{n=-N}^N c_n h_{m-n} = \begin{cases} 1, & m = 0 \\ 0, & m = \pm 1, \pm 2, \dots, \pm N. \end{cases} \quad (1)$$

In other words the impulse response of the cascade of the channel and the equalizer is forced to be an impulse, hence canceling the ISI induced by the channel. Equation (1) can be put into matrix form

$$\mathbf{H}\mathbf{c} = \mathbf{q} \quad (2)$$

from which the equalizer coefficients are solved by inverting the square matrix  $\mathbf{H}$ .

### 4.3 Example

Let  $\mathbf{h} = [0.5 \ 0.85 \ 0.2]^T$  and  $N = 2$ . Equation (2) becomes

$$\begin{bmatrix} 0.85 & 0.5 & 0 & 0 & 0 \\ 0.2 & 0.85 & 0.5 & 0 & 0 \\ 0 & 0.2 & 0.85 & 0.5 & 0 \\ 0 & 0 & 0.2 & 0.85 & 0.5 \\ 0 & 0 & 0 & 0.2 & 0.85 \end{bmatrix} \begin{bmatrix} c_{-2} \\ c_{-1} \\ c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}. \quad (3)$$

The solution is  $\mathbf{c} = [0.69 \ -1.18 \ 1.73 \ -0.47 \ 0.11]^T$ . Notice how zeros were appended on both sides of the vector  $\mathbf{h}$  to create a square matrix  $\mathbf{H}$ . The system impulse response is plotted in Figure 1. The two samples on both sides of the reference sample have been forced to zero, and reference sample has unit amplitude as dictated by the equation (3). However, some ISI remains in the samples outside the equalizer's length. Adopting a larger  $N$  may or may not alleviate the problem.

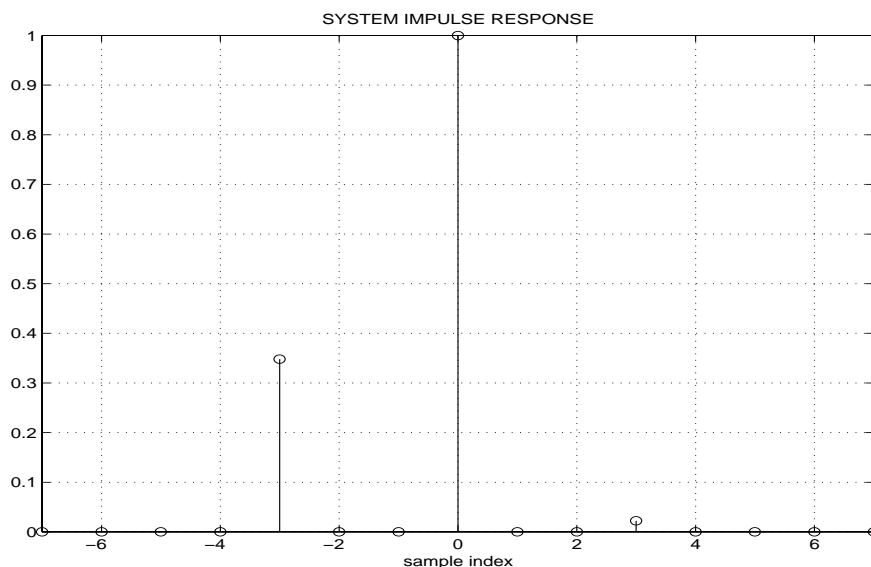


Figure 1. The impulse response of the cascade of the channel and the equalizer.

## Appendices

1. General information on PROPSIM+ radio channel simulator
2. Technical data on PROPSIM+

## References

- Puk00 M. Pukkila, "Channel estimation modeling", seminar presentation, 19.12.2000. Available at <http://www.comlab.hut.fi/opetus/260/lab2.html>.
- Koi01 V. Koivunen, Lecture notes of S-88.200, fall 2001
- Pro95 J.G. Proakis, *Digital Communications*, 3<sup>rd</sup> edition, McGraw-Hill, Singapore, 1995
- Häg01 S.G. Häggman, Lecture notes of S-72.232, spring 2001
- Paj00 P. Pajunen, Lecture notes of Tik-61.238, fall 2000
- Kay93 S.M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*, Prentice-Hall, 1993.

Laa01 T.I. Laakso, Lecture notes of S-88.211, spring 2001



## Appendix 1: General information on PROPSIM+ radio channel simulator

The following excerpt is from the operation manual of PROPSIM+.

See also [www.elektrobit.fi](http://www.elektrobit.fi)

## **PROPSIM WIDEBAND RADIO CHANNEL SIMULATOR**

PROPSim represents the latest technology in multipath fading simulator systems. With features like 35 MHz bandwidth, 312 fading paths and 2.5 ns delay resolution it has the power to meet the channel simulation requirements of current and future mobile, WLL, and radio LAN systems as well as many other wireless applications.

PROPSim was developed with input from customers at the leading edge of wireless communication technology. This, coupled with high-performance RF/DSP hardware and enhanced channel simulation software, has led to a product of unparalleled performance and versatility. Consequently, PROPSim can be used to simulate a wide range of radio channels including GSM, DCS/PCS/PCN, DECT, IS-54, IS-95, TETRA, ISM-band, UMTS, FPLMTS, B-CDMA, W-CDMA, indoor and user defined / measured channels. PROPSim enables the use of both synthetically generated and real measured impulse responses, including real time changes of delay spread.

PROPSim has many powerful features based on very fast and flexible hardware and software. The incorporated digital signal processing technology enables accurate and reliable channel simulation, with calculations of up to 50 giga operations per second. With the provision of numerous academically proven synthetic channel models and an extendible selection of customisable features, PROPSim provides the user with the ultimate in state of the art radio channel simulation technology.



**(Note: Laptop PC is not included in standard PROPSim delivery.)**

## 1. GENERAL

This manual describes the usage of the PROPSim simulator. It contains a general overview of the product and examples concerning its use. In addition to examples, there are detailed descriptions regarding each User Interface item. Figure 1-1 shows the PROPSim, PropSoft and General-IR-Generator windows of the User Interface.

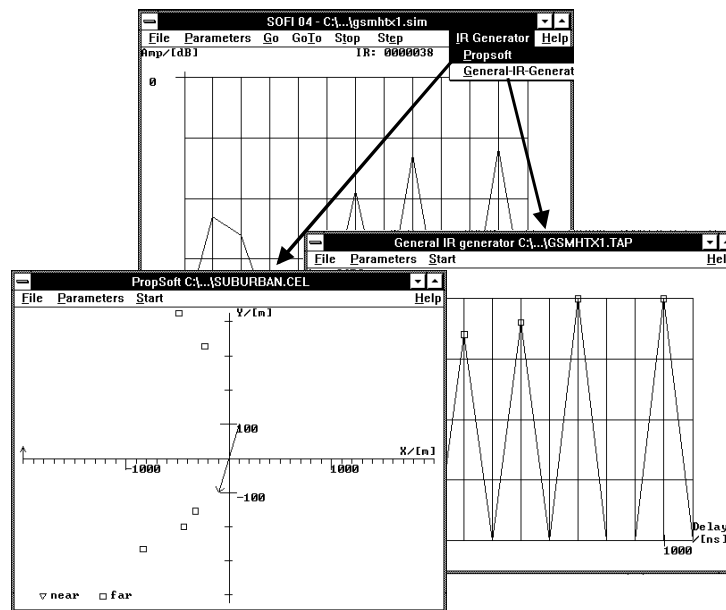


Figure 1-1. PROPSim, PropSoft and General IR generator windows.

The PROPSim system architecture is described in Figure 1-2. The user's RF input signal is down converted into analogue complex base band signals. These signals are filtered and then A/D converted. The multipath fading simulation and summing of taps is carried out in digital parts utilising DSP technology. After fading, the signal is D/A converted and up converted back to the original RF frequency.

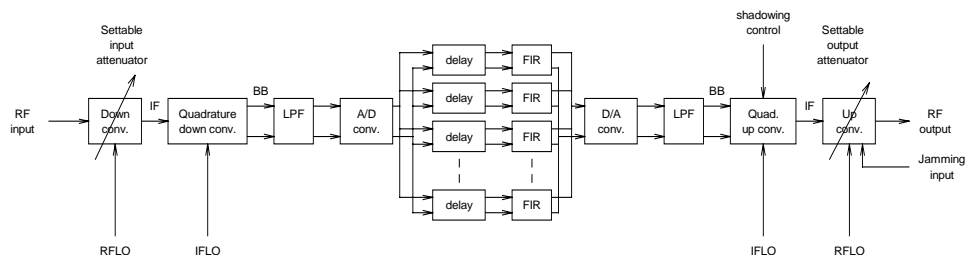


Figure 1-2. General block diagram

## 1.1 Main functions of external equipment

**Note:** The external equipment is not included in the standard PROPSim delivery

### Host PC

- User interface for setting the operating parameters
- Loading the impulse response files into the PROPSim
- Stores and displays the impulse response files
- Generates the synthetic impulse response files

### RF signal generator(s)

Provides the RF local oscillator (RFLO) signals for the LOCLK-unit

### Reference source

Reference signal to the LOCLK-unit for phase/frequency locking of the 500 MHz IF local oscillator and the 40 MHz system clock signals. The reference source is usually included in RF signal generators.

**NOTE:** An alternative reference signal is also provided internally by the LOCLK-unit.

## 1.2 Configurations

### Channels

- One or two channels

### Interfaces

- Baseband to baseband
- RF to RF

**Note:** Independently configurable for each channel.

### Fading paths

- 26 paths / taps per fading block
- Min. configuration is 26 paths / taps per channel
- Max. configuration 12\*26 paths / taps.

## Technical Data

From <http://www.elektrobit.fi/propsim/> [24.1.2001]

Maximum number of channels	2 <sup>1</sup>
Maximum number of fading taps	312
Simulation interfaces	RF, analog baseband
RF frequency range	100...400, 600...2650 MHz
Maximum RF bandwidth	35 MHz
Maximum RF peak input	-15...+10 dBm in 1 dB steps, adjustable
Typical RF output	-40...-15 dBm in 1 dB steps, adjustable
Delay resolution	any
Tap Spacing (TS)	25...3200 ns, adjustable
Tap to tap amplitude and delay error	negligible due to digital tap combining
Maximum propagation delay	100 µs with 25 ns TS 400 µs with 100 ns TS
Maximum Doppler shift	1250 Hz (SD <sup>2</sup> =12)
Maximum vehicle speed	613 km/h at 2.2 GHz center frequency (SD=12)
Pre-defined fading distributions	Constant, Classical (Rayleigh), Flat, Pure Doppler, Rice, Gaussian, Nakagami, Log-Normal, Suzuki
Pre-defined channel models	GSM, IS-54, IS-95, ITU 3G, TETRA
Channel modelling software <sup>3</sup>	General IR Generator, PropSoft+
Supported Windows versions	95, 98, NT 4.0
Power supply	115/230 V, 50/60 Hz, 800 W
Mechanical dimensions (h x w x d)	400 x 450 x 440 mm

1) Possible to synchronize multiple PROPSim™ units. 2) Sample Density. 3) Also possible to import user defined channel models and measured channel data. 4) Digital Baseband available in PROPSim DBB.

### Configurations

- Number of Channels (1 or 2)
- Number of Fading Blocks (1 ... 12)
  - 26 Taps in each Fading Block
  - Maximum Number of Taps 12 x 26 = 312
- Interfaces<sup>4</sup>:
  - RF and Analog Baseband
  - Analog Baseband only

## PRELIMINARY PROBLEMS

### P1 (Least squares channel estimation)

Suppose we model the unknown communication channel as a discrete-time FIR filter with tap coefficients  $\mathbf{h} = [h_0 \ h_1 \ \dots \ h_L]^T$ . Let  $a_0, a_1, \dots$  denote the transmitted symbols<sup>1</sup> at times  $n = 0, 1, \dots$ . The output of the channel  $y_L, y_{L+1}, \dots$  is a convolution sum (discard the first  $L$  transient samples):

$$\begin{aligned} y_L &= a_L h_0 + a_{L-1} h_1 + \dots + a_0 h_L + n_0 \\ y_{L+1} &= a_{L+1} h_0 + a_L h_1 + \dots + a_1 h_L + n_1 \\ &\vdots \\ y_{L+P-1} &= a_{L+P-1} h_0 + a_{L+P-2} h_1 + \dots + a_{P-1} h_L + n_{P-1} \end{aligned}$$

or in matrix form

$$\mathbf{y} = \mathbf{A}\mathbf{h} + \mathbf{n}, \tag{P1}$$

where  $\mathbf{n}$  is a noise vector.

In this preliminary problem we first derive the least squares solution  $\hat{\mathbf{h}}_{ls}$  of (P1) and then examine its properties.

a) Show that the  $\mathbf{h}$  that minimizes the squared error  $J(\mathbf{h}) = \|\mathbf{y} - \mathbf{A}\mathbf{h}\|^2$  is

$$\hat{\mathbf{h}}_{ls} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}. \tag{P2}$$

Recall that  $\nabla_{\mathbf{x}} \{\mathbf{c}^T \mathbf{x}\} = \mathbf{c}$  and  $\nabla_{\mathbf{x}} \{\mathbf{x}^T \mathbf{B} \mathbf{x}\} = 2\mathbf{B}\mathbf{x}$ .  $\mathbf{c}$  and  $\mathbf{x}$  are column vectors,  $\mathbf{B}$  is a symmetric matrix.

b) (recap on linear algebra) Consider the system of linear equations:

<sup>1</sup> These symbols are assumed known.

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

or in matrix form  $\mathbf{Ax} = \mathbf{b}$ . Under what conditions on  $b_1, b_2, b_3$  is the system solvable? Find  $\mathbf{x}$  in that case. The linear combinations of columns of any matrix  $\mathbf{A}$  are called the column space of  $\mathbf{A}$ . Project  $\mathbf{b}$  onto the column space of  $\mathbf{A}$  by solving  $\mathbf{A}^T \mathbf{A} \hat{\mathbf{x}} = \mathbf{A}^T \mathbf{b}$  and  $\mathbf{p} = \mathbf{A} \hat{\mathbf{x}}$ . Find  $\mathbf{e} = \mathbf{b} - \mathbf{p}$ . Verify that it is perpendicular to  $\mathbf{A}$ .

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \\ 2 & 5 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix}$$

c) [Puk00] From the prequel we note that the matrix inversion  $(\mathbf{A}^T \mathbf{A})^{-1}$  might be computationally demanding. Since we are clever we choose the data  $a_1, a_2, \dots$  so that  $\mathbf{A}^T \mathbf{A}$  becomes diagonal. Show that (P2) reduces to

$$\hat{\mathbf{h}}_{ls} = \frac{1}{P} \mathbf{A}^T \mathbf{y} \tag{P3}$$

if the transmitted data vector is

$$\mathbf{a} = [-1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ -1 \ 1 \ -1 \ 1 \ 1 \ 1]$$

and  $L = 5, P = 16$ .

*Note:* The data sequence  $\mathbf{a}$  is one of the eight GSM training sequences and we shall use it throughout this laboratory work.

d) Let  $\mathbf{e} \triangleq \mathbf{h} - \hat{\mathbf{h}}_{ls}$ , where  $\hat{\mathbf{h}}_{ls}$  is given by (P3). Assume that noise is zero-mean Gaussian with covariance matrix  $\mathbf{C}_{nn} = E[\mathbf{nn}^T] = \sigma_n^2 \mathbf{I}$ . Show that the estimation error covariance matrix  $\mathbf{C}_{ee} = E[\mathbf{ee}^T] = \frac{1}{P} \sigma_n^2 \mathbf{I}$ . Show also that the estimate is unbiased, i.e.  $E[\hat{\mathbf{h}}_{ls}] = \mathbf{h}$ .  $E[\cdot]$  denotes the expectation operator.

*Note:* In part a you showed that  $\hat{\mathbf{h}}_{ls} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$  is the LS estimate of  $\mathbf{h}$ . In this part you actually showed that if  $\mathbf{n}$  is a zero-mean, white Gaussian vector

and  $\mathbf{A}$  is chosen as in part c,  $\hat{\mathbf{h}}_{ls}$  is also an efficient estimator of  $\mathbf{h}$ . In other words, in terms of variance, it is the best unbiased estimator of  $\mathbf{h}$ .

- e) Optional problem for estimation gurus: An efficient estimator is one whose covariance satisfies the Cramer-Rao lower bound with equality. Find the CRLB for an unbiased estimate of  $\mathbf{h}$  and verify that it equals  $\frac{1}{P} \sigma_n^2 \mathbf{I}$  thus showing that  $\hat{\mathbf{h}}_{ls}$  is in fact an efficient estimator.

## **P2 (MATLAB implementation of channel estimator)**

Implement a channel estimator using MATLAB. Your estimator should be able to handle a complex-valued QPSK modulated input signal and a complex-valued FIR channel. Channel noise is assumed to be complex WGN. Your estimator should be efficient in terms of variance and computation. Use the file `est.m` as a template for your MATLAB function. You can test your estimator function using the script `t_est.m`. Verify that the estimator is unbiased and its variance agrees with the result of problem P1e. Your estimator function will be used in the post-laboratory part.

## **P3 (Nyquist signaling, RC and RRC)**

- a) Create a BPSK modulated baseband signal using raised cosine (RC) signaling. Plot a segment of the signal and illustrate how intersymbol interference disappears if the signal is sampled at correct time instants. Produce another plot using root raised cosine signaling (RRC) signaling and illustrate that ISI does not disappear.
- b) Filter a BPSK/RRC signal with another identical RRC filter and plot a segment of the output. Illustrate how ISI disappears. Relate the group delay of the cascade of two filters to the length of the impulse response of a single RRC filter. Plot and compare the impulse responses of a cascade of two RRC filters and an RC filter.



- c) Produce scatter plots of sampled<sup>2</sup> BPSK/RC output using the correct sampling instant  $k$ , and sampling instants  $k \pm 1$ . Use four samples per symbol. Repeat for  $\alpha = \{0, 0.5, 1\}$ .  $\alpha$  is the roll-off factor of the RC filter. Comment on the results. In your opinion, what is the effect of roll-off factor to receiver synchronization?
- d) In practical communication systems RRC pulse shaping is always used instead of RC. Why?

### P4 (QPSK transmitter/receiver for AWGN channel)

Simulate symbol error rate of QPSK modulation over an AWGN channel (Figure 1). Use RRC signaling with  $\alpha = 0.5$  and four samples per symbol. Use the script `t_ser.m` as a template. Compare your simulation results to theory, see e.g. pages 271-273 of [Pro95] or any communications theory text book. The script will be later expanded to a receiver for bandlimited channel.

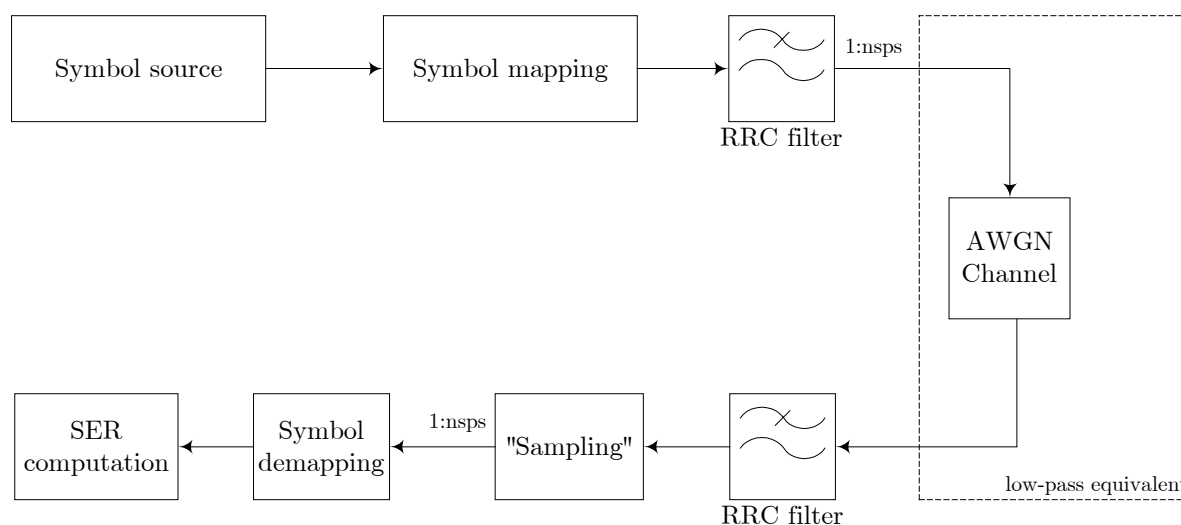


Figure 1. Communication system using RRC signaling.

<sup>2</sup> Since the data is already discrete-time a more accurate term would be decimation.

## P5 (ZF equalizer)

Implement a zero-forcing equalizer using MATLAB. Your equalizer should work with arbitrary channel length and equalizer length; both are assumed to be odd, though. (Later it will have to work also with complex channel and complex signal.) Use the file `equ.m` as a template for your equalizer function. You can test your equalizer using the script `t_equ.m`. The function will be used later in the post-laboratory part as the equalizer block of your receiver.

- a) Plot the impulse response of the cascade of the channel and the equalizer for both channel A and channel B given in the script `t_equ.m`. Compute ISI energy for both channels. Which channel is "worse" in terms of ISI canceling performance of the ZF equalizer?
- b) Plot the frequency response of the channel, equalizer, and the cascade for the two channels given.
- c) Consider the ZF equalizer of 11 taps' length used to equalize the channel B in the script `t_equ.m`. Compute the noise enhancement of this equalizer, i.e. the ratio of equalizer output SNR to input SNR. Give your answer in dB. Compare to noise enhancement of the ZF equalizer for channel A. Draw some conclusions on the noise enhancement and ISI cancellation properties of a ZF equalizer. Does the noise enhancement change if you increase the equalizer length, say... to 31 taps?

## LABORATORY PART

In the laboratory part we transmit a QPSK/RRC signal through a radio channel simulator and sample the simulator output using AD cards. The sampled data is transferred to MATLAB. You should design and implement a receiver that estimates the channel impulse response, removes intersymbol interference (ISI) from the signal, and demodulates the data.

There are two options in how to implement the equalizer. The default is to use the simple ZF equalizer designed in the preliminary part (option #1). If you have done the computer assignment of S-88.211 you can use the adaptive, linear MSE equalizer implemented therein (option #2). You may have to modify it to suit the purposes of this laboratory work.

### Step 1: Create the transmitted signal and download it to AWG

Assistant will assign you the message bits to be transmitted over the ISI channel. Write down the number of bits, since you will need it later. Map the bits to 4-ary symbols by using the function `bits2sym`. Add the training symbols to the beginning of the data block (Figure 1) and create the transmitted signal in MATLAB by using QPSK modulation and RRC signaling. Use eight samples per symbol. Save the real and imaginary part of the transmitted signal vector to separately under different names onto a floppy disk in ASCII form. Export both signals to AWG. Time interval between sampling points should be set to  $1 \mu\text{s}$ . "Preserve time" when exporting.



Figure 1. The data format.

## Step 2: Acquiring the received signal to MATLAB

The measurement set-up is depicted in Figure 2. In AWG, select the real part of the signal to be transmitted from channel A, and the imaginary part from channel B. Set the triggering to external with one burst transmitted on each trigger instant. Set the triggering signal frequency to 10 Hz or lower.

Choose channel model *12tap\_1.sim* in the simulator; notice that the channel tap spacing is not symbol-spaced. Set the FIR update rate to 100 Hz (option #1) or 2000 Hz (option #2). What is the absolute delay spread of the channel? In theory, how many taps do you need in your channel estimate?

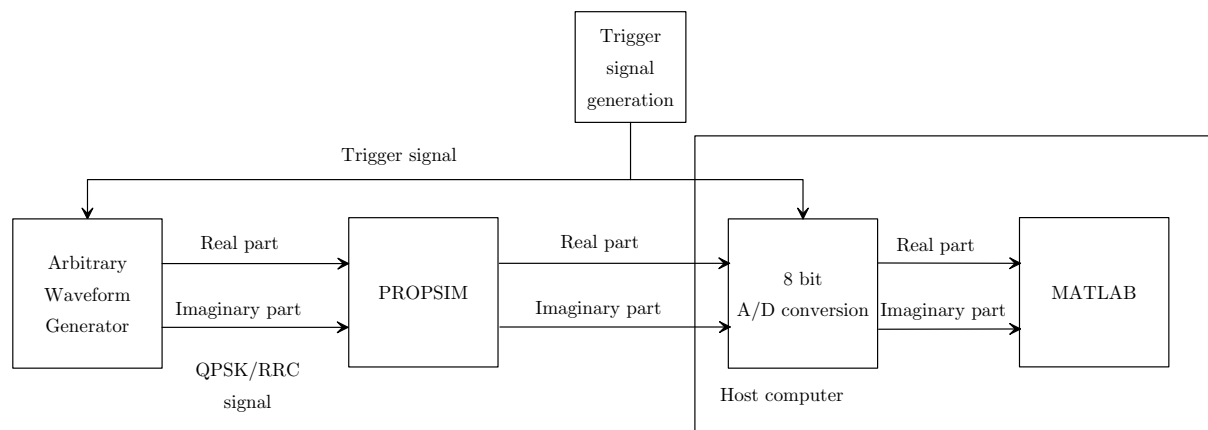


Figure 2. Measurement set-up.

Sampling cards are controlled with the script `lab2.m`. Running the script captures the complex channel output into two vectors that you should save for further processing.

Once you have obtained the received signal, you may proceed to the post-laboratory part. If you have time you can start implementing the receiver in the student laboratory, and ask questions from the assistant.

## Step 3: Implement the receiver and detect the transmitted symbols

Implement the receiver shown in Figure 3 and detect the transmitted symbols. You may expand the script `t_ser.m` you implemented in the preliminary problems.

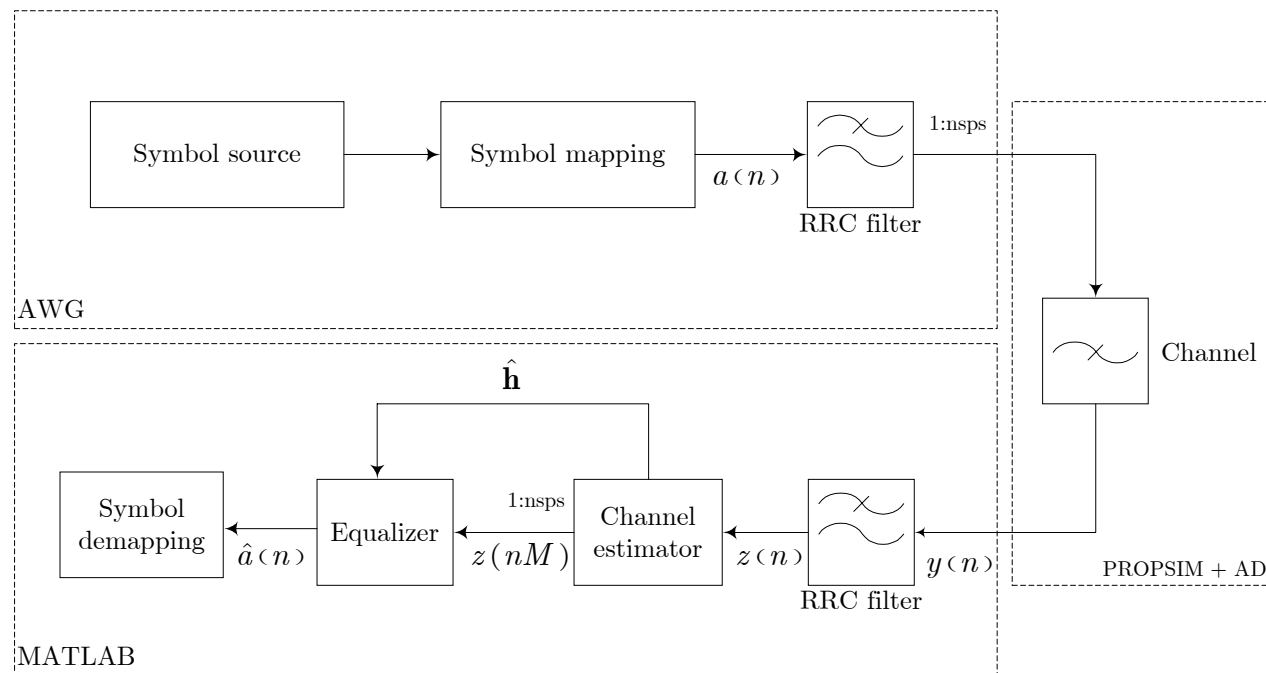


Figure 3. Block diagram of the communication chain.

Notes:

- RRC filter: Nothing strange here.
- Channel estimator: This block has two inter-related purposes: it has to supply a channel estimate  $\hat{\mathbf{h}}$  to the equalizer, and it has to reduce the sampling rate of  $z(n)$  to a single sample per symbol, since the equalizer works on that sampling rate. A heuristic procedure for accomplishing this is suggested below. You may figure out a better way of doing this.

Assume that  $\text{nsps} = M$ . Demultiplex the signal  $z(n)$  to  $M$  sequences with  $\text{nsps}=1$  in each. You can now use the estimator function you implemented in the preliminary part to each of the  $M$  subsequences separately. You will end up with  $M$  estimates  $\hat{\mathbf{h}}_i$ ,  $i = 1, 2, \dots, M$ . Choose the one with minimum squared error:  $\arg \min_i \|\mathbf{y} - \mathbf{A}^H \hat{\mathbf{h}}_i\|^2$ . Another method (and perhaps a better one...) is to choose the estimate that produces the minimum ISI energy system impulse response. (Of course this means that you will have to compute  $M$  sets of equalizer coefficients and convolve them with the channel estimate to find the minimum ISI energy of system IR; you would not want to do this in a practical receiver implementation...)

Due to sampling card triggering, AD/DA conversions, and fractional-delay channel tap spacing there is also uncertainty in delay to be used by the estimator. See hints for P2.

The estimate  $\hat{\mathbf{h}}$  and the signal  $z(nM)$  passed to the equalizer should be the one that has minimum square error both in time domain, and "subsequence" domain.

Note that the estimate includes the impulse responses of transmit filter, channel and receiver filter.

- Equalizer: If you use option #2, use the estimate provided by the channel estimator as the initial value of channel taps from which you can derive the initial coefficient values for the LMS equalizer. You can use the ZF equalizer implemented in preliminary problems for computing the initial coefficients. You have to use the LMS algorithm in decision-directed mode.
- Symbol demapping: You can map the symbols back to bits with `sym2bits`. At some point in the receiver chain you will have to remove the training sequence from the signal. You may do this after or before symbol demapping.

### Questions, option #1:

- a) Plot the system impulse response, i.e. the convolution of the channel response and the equalizer response. Repeat for a few different equalizer lengths and two different channel estimate lengths. Compute the ISI energy in each case. What is the optimal combination of channel estimate length and equalizer length?
- b) Compute the ISI energy for each sampling instant using the optimum length equalizer and channel estimate.
- c) Plot the frequency response of the channel, equalizer and the system for the optimum equalizer length, channel estimate length and sampling instant.
- d) Convert the detected data bits to ASCII text by using the script `bits2asc.m`. Include the text in the appendix of your report. Any errors in the text?

## Questions, option #2:

- a) Try different LMS step sizes, including zero. Compute and plot the square error of the signal in each case and estimate MSE. Comment on the results.
- b) Convert the detected data bits to ASCII text by using the script `bits2asc.m`. Include the text in the appendix of your report. Any errors in the text?
- c) What happens to the text if the LMS step size is zero? Why?

## Instructions for the lab report

Do not forget to include your email addresses to your lab report (for sending boomerang info). You should write the laboratory report with a word processor. There are no strict requirements on the format of the report, except that it must be readable and understandable. Anyone reading your report should be able to follow your line of reasoning and, if necessary, be able to repeat your actions and verify the results. To make the report more readable you should embed figures, tables etc to your report (not everything to appendices). Long MATLAB scripts and functions should be put into appendices.

For plotting figures, excellent black&white printing quality is obtained by using the command `print picname -deps2 -tiff`. The EPS picture file can be inserted into the document and resized according to one's needs without degradation in quality. Use `subplot` when applicable.

The report does not have to be long but you should include at least:

- A summary of your receiver implementation, including a short description of the MATLAB files.
- A section on what were the most difficult parts, and how did you solve them.
- Answers to the questions presented above along with the appropriate figures.