# Project in SSY305 Kommunikationssystem

E. G. Ström, E. Steinmetz, M. Tavana, K. Hausmair, and Chouaib B.L

# Contents

# Introduction

This project deals with several aspects of communication systems. It is divided into three parts. The first part is focused on the physical layer of the protocol stack. It is supposed to introduce you to the different building blocks of the physical layer. After completion of the project you should understand how pulse amplitude modulation (PAM) and matched filter receivers work. The second part of the project is about automatic repeat request (ARQ) protocols, which are often used to implement reliable transmission over an unreliable service. This part of the project aims at getting you familiar with the most common types of ARQ protocols, and you are also supposed to implement a so-called *stop-and-wait ARQ* protocol. In the third and last part of the project you should study an information and communication technology (ICT) application and analyze which requirements this specific application poses on communication technology. Part one and two require you to produce deliverables in the form of reports. The reports are independent, but you are supposed to hand in one file containing both reports. For part three, an oral presentation has to be given. Detailed rules for the project can be found in the Course PM at Canvas.

# Part I - Pulse Amplitude Modulation (PAM)

## 1 Project Purpose

### 1.1 Aims

The main objective of this project is to give the student an understanding of the basic building blocks of the physical layer in a digital communication system. The student will also learn different performance measures for a digital communication system.

### 1.2 Learning outcomes

The communication system designed in this projects aims at demonstrating the basics in digital communication. After completion of this project you should be able to:

- Design a signal constellation
- Map information bits to symbols given a certain constellation
- Explain how pulse amplitude modulation works
- Describe how a matched filter receiver works
- Design the transmit and receive filters in order to maximize the throughput for a given bit error rate (BER)
- Determine decision boundaries for a minimum distance decoder given a certain constellation

## 2 Project Task

### 2.1 Development methodology and project Overview

When designing a communication system, it is common to use the following flow of activities.

1. Find a sufficiently accurate model of the channel
2. Make a system design based on the channel model
3. Test the design by simulations (using the channel model)
4. Test the design in practice (using the real channel)

The process is often iterative in practice. For instance, it is quite common that it is difficult or even impossible to make a perfect design at step 2. We then have to make an approximate design which may fail when tested at step 3 or step 4, and we are forced to redesign the system (step 2) or to refine our channel model (step 1).

In this project, we imagine that you are working at a company that is participating in a large development project for designing and implementing a satellite-to-satellite communication system. The development project is divided into the following work packages:

- WP1 - Project management
- WP2 - Identification of channel model
- WP3 - System design
- WP4 - Evaluation of system design by simulation
- WP5 - Hardware design

Your team (group) is involved in WP3 and WP4 hence the focus will be on step 2 and step 3 in the development process.

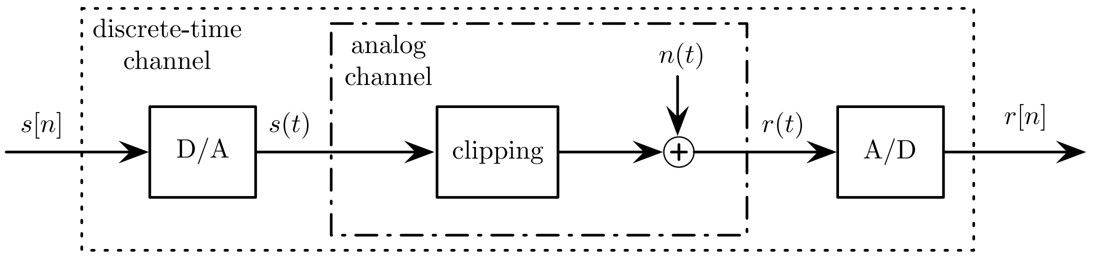


Figure 2.1: Block diagram of analog and discrete-time channels

## 2.2 Developed Channel Model

As we know that the physical channel is a satellite-to-satellite link, the transmitted signal will be subject to amplitude clipping (due to the finite output voltage range of the transmitter hardware), and additive noise (due to background noise and thermal noise in the receiver electronics). Two transmitter hardware platforms are considered. The more expensive platform, HW2, allows for a higher clipping threshold than the cheaper platform HW1. A discrete-time channel model that captures these effects and the analog-to-digital (A/D) conversion and digital-to-analog (D/A) conversion in the transmitter and receiver, respectively, has been developed in WP2, see Figure 2.1. For simplicity, we assume that the A/D and D/A are ideal, time-synchronized and operating at the sample frequency $f_s = 1/T_s$, where $T_s$ is the time between consecutive samples. The sampling theorem [1, Ch. 5] implies that, for *any* choice of $s[n]$, the bandwidth of the analog signal $s(t)$ will be no larger than $f_s/2$, and that $s(t)$ can be formed from its $T_s$-spaced samples $s[n] = s(nT_s)$ as

$$s(t) = \sum_{n=-\infty}^{\infty} s[n] \operatorname{sinc}\left(\frac{t - nT_s}{T_s}\right).$$

Moreover, $r[n]$ is $T_s$-spaced samples of the received signal $r(t)$ after lowpass-filtering with an ideal lowpass filter with cut-off frequency $f_s/2$. The noise signal $n(t)$ is white Gaussian noise with power spectral density $N_0/2$. We note that if there is no cliplping or noise, then $r[n] = s[n]$.

The discrete-time model has been implemented by WP2 in the MATLAB function `simchannel.m`, see the right-hand side of Figure 2.2. By comparing Figures 2.1 and 2.2, we see that $w[n]$ is $T_s$-spaced samples of the lowpass-filtered noise signal $n(t)$. The two hardware platforms result in two channel models: Channel 1 and Channel 2, which are modeled in `simchannel.m` (check Section 3.1 for details).

Hence, step 1 above is complete. Also parts of step 2 have been completed by another development team, and you will be provided with the two MATLAB functions `transmit.m` and `receive.m`. Your task is now to finalize the system design and test it using the channel model (step 3), this includes implementing the remaining parts of the `receive.m` and the `transmit.m` functions according to the block diagram in Figure 2.2.

## 2.3 Detailed description of project task

The remaining parts of the work in WP3 and WP4 for your team are to

a) specify appropriate signal constellations

b) convert bits to symbols

c) design the discrete-time pulse $g[n]$ which is used for pulse shaping in the transmitter, i.e., the transmit filter impulse response

d) perform PAM

e) design the receive filter impulse response $f[n]$

f) design and implement the down sampler

g) determine decision boundaries for the specified constellations and compute the received symbols

h) convert symbols to bits

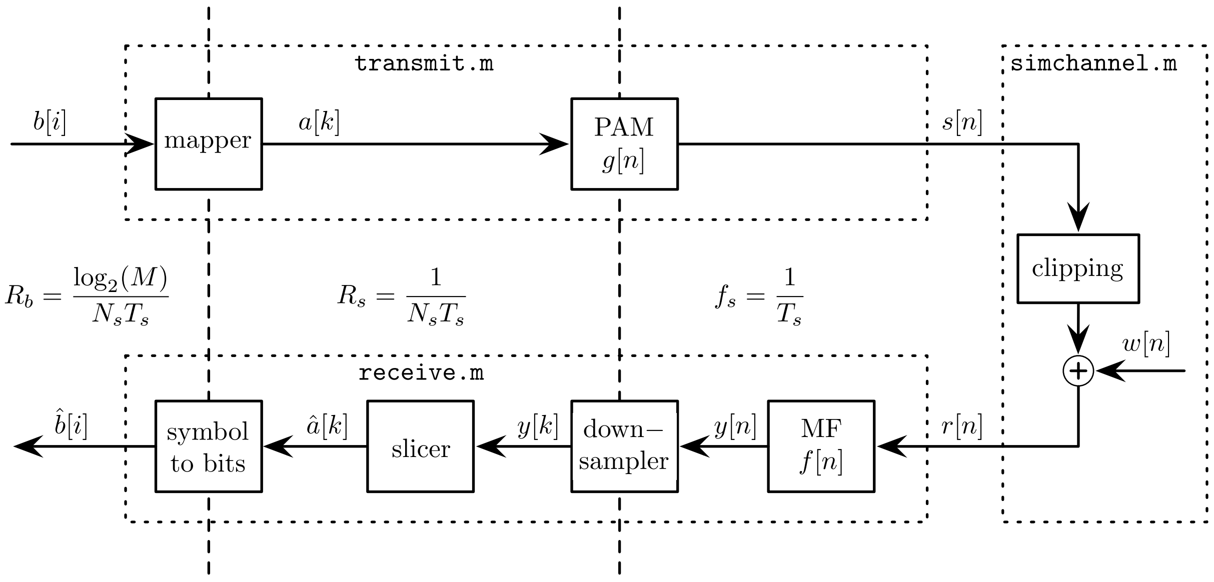


Figure 2.2: Block diagram of transmitter-receiver chain

The transmit and receive filters should be chosen such that the bit rate $R_b$ of the system is maximized and that the bit error rate is below the specified target bit error rate $P_{b,\text{target}} = 10^{-3}$. More specifically, the design of the filters includes specifying the length (in number of samples $N_s$) and shape of the impulse responses $g[n]$ and $f[n]$. When designing $g[n]$ you should work under the assumption of an *infinite bandwidth* and that $g[n]$ should be *restricted to a single symbol interval.* The system should also be able to support two different modulation formats binary PAM ($M = 2$) and 4-PAM ($M = 4$). Hence your task is to repeat the above procedure for both constellations, i.e., for $M = 2$ and $M = 4$.

Given the two available channel models you are supposed to do your investigation following these guidelines

1. Perform tasks a) to h) with respect to Channel 1 (i.e., the channel with the lower clipping threshold imposed by the inexpensive hardware platform HW1), and satisfying the requirements indicated above
2. After finishing your design and analysis based on Channel 1, you are requested to investigate the performance of the system over Channel 2 following the tasks
   - set $g[n]$ and $f[n]$ to the optimal values found during the analysis of 2-PAM over Channel 1
   - using $g[n]$ and $f[n]$ as defined in the previous point, modify the signal constellation for both the 2-PAM and 4-PAM case such that the bit error probability over Channel 2 is minimized (OBS: make sure to use the same pulse length for both modulations over Channel 2)
   - record, from the GUI, the bit error probability performance of the two modulation format over Channel 2

## 2.4 Deliverables

You are expected to deliver a project report containing a detailed description of all your design choices of the 2−PAM and 4−PAM systems, over the two channels. The report should include

- constellation diagrams with clearly marked decision boundaries
- plots of the transmit pulse (use matlab `stem` function), with clear and detailed motivation why you chose this specific pulse shape
- plots of the receiver filter shape
- plots of the receiver filter output where you clearly indicate how the down sampling is done
- comparison of the results for 2-PAM and 4-PAM over Channel 1 ($N_s$, resulting $R_b$, etc.), discuss and explain the differences!

- comparison of the results for 2-PAM and 4-PAM over Channel 2 (is $P_{b,\text{target}} = 10^{-3}$ satisfied, resulting $R_b$, etc)
- theoretical assessment* of bit error probability of the designed PAM systems over the two channels
- appendix with the MATLAB code for the finalized versions of `transmit.m` and `receive.m` (Only for $M = 4$), and any other files containing code implemented by you

Always explain and motivate why you chose certain settings, parameters etc.! Discuss trade-offs and draw comparisons to theory! Make sure to label all figure axes correctly!

*Hint: It can be shown that $w[n]$ is discrete-time white Gaussian noise with variance $\mathrm{E}\{|w[n]|^2\} = \sigma^2 = f_s N_0/2$. Moreover, using ideal D/A-conversion with sample frequency $f_s = 1/T_s$, the energy of an analog signal $g(t)$ and its sampled version $g[n] = g(nT_s)$ are related as

$$\int_{-\infty}^{\infty} |g(t)|^2 \, dt = T_s \sum_{n=-\infty}^{\infty} |g[n]|^2.$$

## 3 MATLAB listing of the project files and hints

This section lists the MATLAB code for the files used in this project. The file set includes the simulated channel `simchannel.m`, the user interface `Interface.m` with `Interface.fig`, the file `PlotSignals.m` that generates plots of the signals in the system, and outlines for the transmission and reception programs, named `transmit.m` and `receive.m`, respectively. The files can be downloaded from the Canvas course web page (`ProjectFilesPart1.rar`). Hint: MATLAB has a built-in debugger that can help resolving bugs that may occur in your code. (Search for "Debug a MATLAB Program" on `MathWorks.com` for details).

### 3.1 The `simchannel.m` program

The simchannel m-file is used to simulate the real channel. This is convenient since we can develop the transmitter and receiver algorithms without access to the real channel. Furthermore, it allows for some more systematic testing of your algorithms. The channel will add discrete-time white Gaussian noise $w[n]$ with variance $\mathrm{E}\{|w[n]|^2\} = \sigma^2 = 15$ and clip the input signal to be in either the interval $[-5, +5]$ (for Channel 1), or $[-15, +15]$ Volts (for Channel 2). This selection can be made by choosing Channel 1 or Channel 2 from the GUI. The channel model is a discrete-time model, and we assume that the analog-to-digital (A/D) and digital-to-analog (D/A) conversions are performed with the sample rate $f_s$ Hz. You are not allowed to make any changes to `simchannel.m`.

### 3.2 The `transmit.m` and `receive.m` programs

Note that these programs are merely skeletons and need significant programming to work as intended. As a matter of fact, the main task of this project is to complete `transmit.m` and `receive.m` programs as well as their subroutines. The file `PlotSignals.m` is used in both `transmit.m` and `receive.m` to generate plots of the signals in the system. The programs are called by the GUI program `Interface.m`.

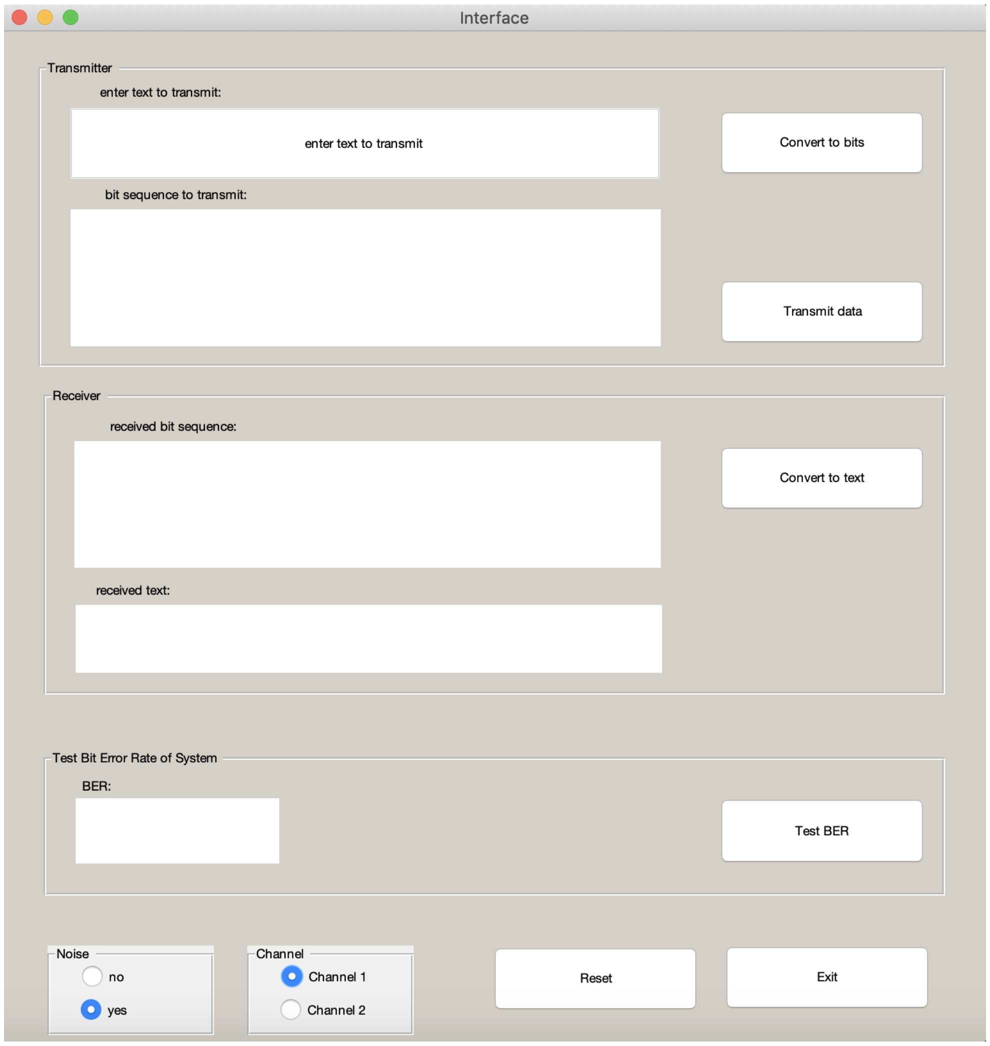
### 3.3 The graphical user interface `Interface.m` with `Interface.fig`

The graphical user interface (GUI) is used for interaction with the `transmit.m` and `receive.m` programs and to visualize the information flow between the transmitter and the receiver. The GUI is launched by typing `Interface` at the command line, or by opening `Interface.m` (not `Interface.fig`!) and pressing "Run". This will open a GUI window, and as can be seen in Figure 3.1 the GUI window contains a Transmitter panel, Receiver panel, Test Bit Error Rate of System panel and a Noise panel. You are not allowed to make any changes to `Interface.m` and `Interface.fig`.

A quick description of the GUI functionality is provided below:

- Text is entered in the transmitter text box "enter text to transmit" in the GUI window. By pressing the `Convert to bits` button the text is converted to a bit sequence, which will be shown in the second transmitter text box. Once the `Transmit data` button is pressed the bit sequence is passed to the transmitter program `transmit.m` (which you are supposed to complete). The transmitter program computes the samples of the transmitted waveform and returns the samples to the GUI routine. The GUI then sends the samples to the simulated channel (implemented in `simchannel.m`), and the output from the channel is passed to the receiver program `receive.m` (which you are supposed to complete). In the receiver program the data samples are demodulated and decoded into a bit stream which is returned to the GUI routine, and presented in the text box "received bit sequence". By pressing the `Convert to text` button the received bit sequence is converted back to text. Hint: To make it simpler start with entering only one letter in the text box "enter text to transmit".

- The bit error rate (BER) of the system can be evaluated by pressing the button `Test BER`. When this button is pressed the BER of the system is evaluated by sending a long bit sequence through the simulated channel using the transmitter and receiver programs that you have implemented. The result of the BER test is shown in the text box "BER".
- In the Channel panel you can select to simulate your communication system using either Channel 1 or Channel 2. These introduce a clipping of $[-5, +5]$ and $[-15, +15]$, respectively. The AWGN noise power is the same in both.
- In the Noise panel you can choose whether noise should be added or not when the data samples are passed through the channel. Hint: In the beginning of the design phase it could be useful to turn off the noise, but of course your transmitter and receiver should be designed to meet the target BER when the noise is turned on.
- The `Reset` button clears all text boxes in the GUI
- The `Exit` button closes the GUI



Noise: no [✓ yes]

Channel: [✓ Channel 1] Channel 2

Figure 3.1: MATLAB graphical user interface.

# Part II - Automatic Repeat Request (ARQ)

## 4 Project Purpose

### 4.1 Aims

The main objective of this project is to get the student acquainted with the automatic repeat request (ARQ) technique often used to ensure reliable transmission.

### 4.2 Learning outcomes

The aim of this project is to be able to implement a basic ARQ protocol. After completion of this project you should be able to:

- Briefly describe the three different ARQ protocols *stop-and-wait ARQ*, *Go-back-N ARQ* and *Selective-Repeat ARQ*
- Give a detailed explanation of the stop-and-wait ARQ protocol
- Explain what an acknowledgment is
- Describe what sequence numbers are and why they are needed
- Implement a basic parity check for error detection
- Determine appropriate time-out period for a transmitter using an Stop-and-Wait ARQ scheme

## 5 Project background

In order to provide reliable transmission on a peer-to-peer link (e.g., between a client and a server) so-called Automatic Repeat Request (ARQ) protocols are often implemented. The basic principle for an ARQ protocol is that it combines error detection with retransmission to ensure that data is delivered to the user despite possible errors that could occur during the transmission. The three basic types of ARQ protocols are

- *Stop-and-Wait ARQ*
- *Go-Back-N ARQ*
- *Selective-Repeat ARQ*

## 6 Task

### 6.1 *Stop-and-Wait ARQ*, *Go-Back-N ARQ* and *Selective-Repeat ARQ* Understanding

Read the Chapter 23.2 on *Stop-And-Wait*, *Go-Back-N* and *Selective-Repeat ARQs* in the course book (B. Forouzan, *Data Communications and Networking,* 5th Edition, McGraw-Hill, 2013) and then answer the question below. Imagine that you are having a phone conversation with a friend. The connection you have is not very good, so sometimes you can not hear what your friend is trying to say. To make sure that you understand everything your friend says you can continue the conversation following one of the three procedures below:

1. you can say “what?” when you don’t understand something and then your friend repeats the whole sentence
2. you can let your friend say one word at a time and after each word you say yes until the conversation is over
3. if there is something you don’t hear you can ask your friend to repeat just that. For example if you would hear “lets meet in school at ... o’clock” your would respond “what time?” and your friend would repeat just the time.

<u>Question:</u> *Which ARQ protocol would each of those three procedures correspond to?*

## 6.2 Implementation of *Stop-and-Wait ARQ*

In this task you are supposed to implement a reliable transmission of data between a client and a server, i.e., you are supposed to make sure that the data transmitted from the client in Fig 6.1 is correctly received by the server[1]. You should do this by implementing a simple Stop-and-Wait ARQ protocol that makes the data transfer robust to bit errors and packet drops. To get you started you are provided with a set of MATLAB files (see Canvas `ProjectFilesPart2.rar`). In order to emulate two asynchronous processes such as a client and a server, two MATLAB sessions are used. By typing `ServerGui` at the command line in the first MATLAB session and `ClientGui` at the command line of the second MATLAB session, we will start the independent processes of the Client and Server GUI. Before you can start a transmission you need to press the button "Run Server" in the Server GUI. When the server is running you can transmit data from the client by pressing the "Transmit data" button. Note that in order to successfully establish a connection the server must be running before the client can start to transmit. In the Client GUI you can enter the desired frame length. You can also do several consecutive transmissions of the picture by changing the number of transmission attempts in the Client GUI. Also, note that the roundtrip time between the client and the server depend on the computer that is running the simulations.
Hint: if you think MATLAB is not responding correctly you can use Ctrl+C to terminate the current process.

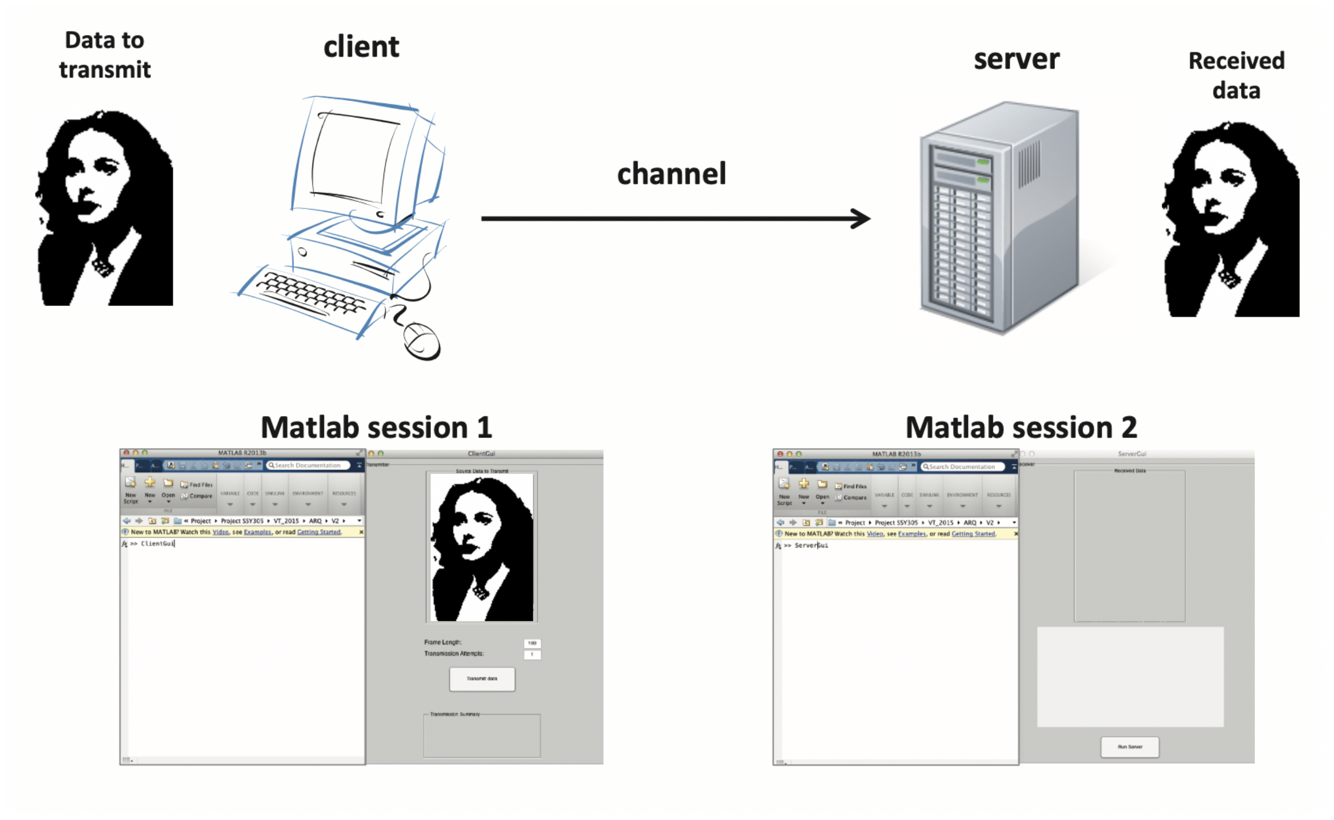


Figure 6.1: Data transfer between client and server emulated by two MATLAB sessions on the same computer.

When a connection is established, the client runs the program `Transmitter.m` and the server runs the program `Receiver.m`. You are supposed to complete these two programs by implementing both the transmitter and receiver side of the *Stop-and-Wait ARQ* protocol. This includes:

- embedding the data packets in a frame where header and trailer carry bits for error detection and sequence numbering. Implement this by completing `pkg2frame.m`
- implementation of two error detection methods (single parity bit, and another method of your choice). Implement error detection by completing `ErrorCheck.m`
- implementation of the time-out process in the transmitter

[1]The picture is of Hedy Lamarr, a co-inventor of an early Frequency Hoping Spread Spectrum (FHSS) system (patent granted in 1942). FHSS technique is used in modern communication systems like Bluetooth. `https://en.wikipedia.org/wiki/Hedy_Lamarr#Inventor`

  - implementing a mechanism to record round-trip-time (RTT) for successfully acknowledged packets.
  - choose an appropriate time-out period based on the statistics of the recorded RTT
  - test at least 3 different time-out periods
- sending acknowledgments (control frame) when the receiver correctly receives a frame
- choosing a suitable frame length
  - test at least 5 frame lengths for both error check methods
  - choose a frame length that minimizes the picture transmission time and results in no more than 3 bit errors per received picture

In the report, discuss what affects the probability of undetected errors, motivate your choice of frame length, and explain what trade-offs are involved.

As can be seen in the skeleton programs for `Transmitter.m` and `Receiver.m`, the function WriteToChannel and ReadFromChannel are used to send data over the channel, i.e., when the transmitter program wants to send data it makes the function call `WriteToChannel(Channel, frame)` and to receive data it makes the function call `Y = ReadFromChannel(Channel, ExpectedLengthOfFrame)`. This process is illustrated in Fig. 6.2

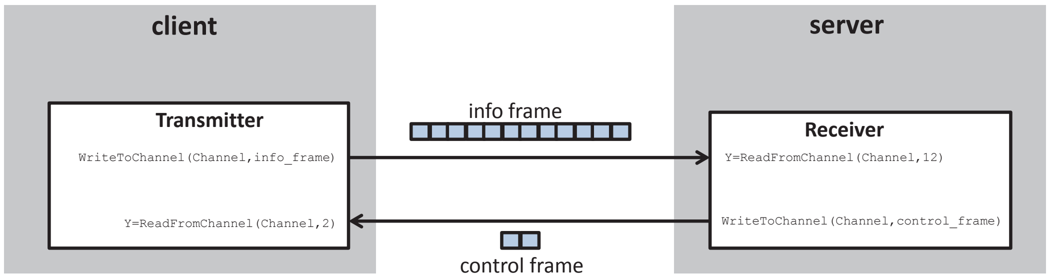

client
server
Transmitter
WriteToChannel(Channel,info_frame)
Y=ReadFromChannel(Channel,2)
info frame
Receiver
Y=ReadFromChannel(Channel,12)
WriteToChannel(Channel,control_frame)
control frame

Figure 6.2: Illustration of how to use the WriteToChannel and ReadFromChannel functions for an information frame length of 12 bits and a control frame length of 2 bits.

Note that you are allowed to make changes only in `Transmitter.m`, `Receiver.m`, `pkg2frame.m`, `ErrorCheck.m`, , but not in any other provided file. In order for any changes in the code to take effect the GUIs need to be restarted.

## 6.3 Deliverables

You are expected to deliver a project report with detailed documentation of your implementation of the *stop-and-wait ARQ* protocol. The report should contain:

- answers to the question in section 6.1 and
- description of the implemented error check methods
- documentation of (i) the average number of bit errors per picture transmission and (ii) the average picture transmit time, when testing the system with different frame lengths
- documentation of the average picture transmit time when using different time-out period
- the chosen frame length and time-out period
- appendix with the MATLAB code for the finalized versions of `Transmitter.m` and `Receiver.m`, and any other files containing code implemented by you.

All design choices have to be clearly described and motivated, for example which time-out period and frame length did you choose and why, are there any trade-offs, do your results agree with theory, etc.? All MATLAB code included in the report should be well commented.

# Part III - Requirements for ICT Application

## 7 Project Purpose

### 7.1 Aims

The main objective of this project is to give the student an understanding of what is important when choosing a communication technology for a specific application.

### 7.2 Learning outcomes

After completion of this project you should be able to:

- Give examples of different information and communication technology (ICT) applications
- Describe one ICT application in detail
- List typical requirements for communication applications (e.g., reliability, latency )
- Analyze an ICT application and determine which of the requirements are important for this particular application
- Suggest an appropriate communication technology

## 8 Project Task

### 8.1 Detailed description of project task

Communication systems are an enabler of ICT applications. For this project you have to choose one of the following ICT applications

- Traffic safety
- E-health
- Smart grid
- Process control

Register your project group for the chosen topic in Canvas! Note that each topic can only be chosen by a limited number of groups on a first-come first-serve basis. Your task is then to analyze which requirements this particular application poses on the communication technology. Think about things that are important to consider when designing a communication system, and which of the requirements are important for your application. For example, does your application need to be reliable, available everywhere, secure etc.? Why do you think some requirements are important or not important for your application (give examples)? Suggest which communication technology could be suitable for this particular application. Motivate why you think this is the best choice.

### 8.2 Deliverables

You are expected to give a presentation where each member of the group actively participates. The presentation should contain

- a short description of the ICT application
- an analysis of the communication requirements
- suggestions on suitable communication technologies for this application. In particular, a discussion about the suitability of 5G technology, should be included.

The presentation should be no longer than 15 minutes. All group members should speak during the presentation.

## 9 Grading Criteria

The third part of the project is graded according to the following guidelines

- Organization of presentation
  - Presentation is well structured
  - Appropriate amount of material is presented
  - Oral delivery is clear and suitable for the audience
  - Time keeping (15 minutes)
- Content of presentation
  - Description of the ICT application is thorough
  - Relevant communication requirements are well analyzed
  - Suggestions of suitable communication technologies are valid
  - Brief discussion about 5G's suitability is covered.
- Ability to answer questions and to ask relevant questions when in the audience .

## References

[1] Erik G. Ström. "A very brief review of signals and systems.", January 2016. Doc. no. SSY305/ext:02, rev. B. Available at the SSY305 Canvas course page.