Project Assignment 2
EQ1220 Signal Theory

Stockholm 2018

School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology

Background

Agent 007, James Bond, is visiting Stockholm on one of his missions. From a colleague at the CIA, Felix Leiter, Mr. Bond has received an encrypted picture of the enemy he is hunting.

You are working at MUST, the Swedish Intelligence Agency, and are assigned to assist Agent 007 in finding out who the suspect is. The picture has been encoded with an advanced encryption method to prevent unauthorized personnel from viewing it. Since you need the right decoding key to access the information, the encrypted picture has been sent over the Internet. The encoding key has however been sent to Agent 007's SpyPhone™, which is a digital radio communication unit. The SpyPhone™ contains the decoder that can take the encrypted picture and the decoding key and recreate the original picture. However, you need to design a detector that takes the received transmission on the SpyPhone™ and reveals the decoding key. Then you should use the key to decode the image and inform Mr. Bond who is the suspect (so he can take necessary actions).

To enable you to solve this, you are provided with the following model of a simple digital communication system. In the description, there are also some suggestions on how one can proceed to build an equalizer and a detector, as well as descriptions of the data files used in the project.

A digital communication system

Since the introduction of cellular/mobile telephony (like GSM and UMTS), most of you have been in contact with wireless digital communication systems. This type of systems has been examined carefully by researchers. It has been found that surprisingly simple models of how the signals are propagating over the radio channel are quite accurate.

Figure 1 presents a block diagram of the time-discrete model we are considering here. This model is a simplification of the more accurate (but still quite simple) models used in GSM, but will still give you some insight in the design of receivers. The different blocks in the figure will be explained below.

The input signal

The information that is to be transmitted, $s(k)$, is usually encoded in a binary form, $s(k) \in \{0, 1\}$. Before the information can be transmitted, we need to define a mapping from the binary information symbols to the channel space. One such mapping is pulse amplitude modulation (PAM), where the binary

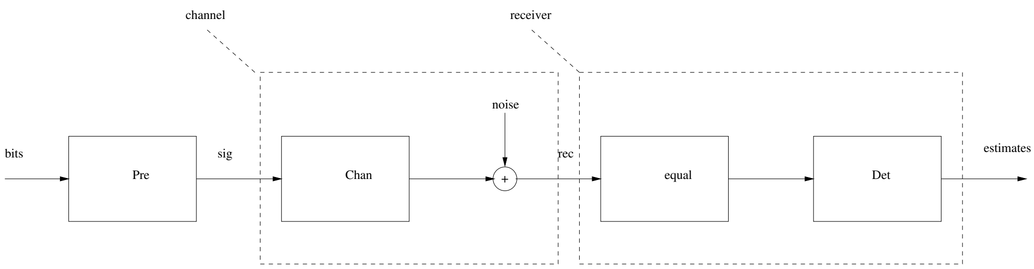

Figure 1: A block scheme of the communication system considered. The encrypted data bits, $s(k) \in \{0,1\}$, are initially mapped to symbols, $b(k) \in \{-1,1\}$, which are then transmitted. The channel consists of a distortion, h, as well as some additive noise, $n(k)$. The received signal, $r(k)$, is sent to an equalizer, w, and the intention of this filter is to remove noise and distortion from its output $z(k)$. The filtered signal is sent to a detector that tries to reconstruct $b(k)$.

signal is mapped to an antipodal signal in the following manner

$$b(k) = \begin{cases} -1 & s(k) = 0, \\ 1 & s(k) = 1. \end{cases} \tag{1}$$

The antipodal signal $b(k)$ is then used as the input signal to the communication channel. In our case $s(k)$ is the decoding key to the encrypted picture.

The communication channel

There exist a multitude of models for how a radio channel might look like. In GSM for instance, a number of different models have been standardized to enable realistic and repeatable simulations and evaluations (to enable comparison and development of new algorithms). Most channels introduce some kind of distortion to the transmitted signal. In this assignment, we will model the channel distortion as a time-invariant, time-discrete and unknown finite impulse response (FIR) filter, h, with four taps (order 3). Furthermore, as all practical systems are corrupted by noise, we will assume that the channel will add white noise to the distorted input signal. Given the input signal $b(k)$, we obtain

$$r(k) = \sum_{l=0}^{3} h(l)b(k-l) + n(k), \quad k = 1, \ldots, N, \tag{2}$$

as the channel output. The noise term, $n(k)$, is assumed to be additive white Gaussian noise (AWGN) with (the unknown) variance σ^2.

Receiver

The receiver is the unit that decides what the received signals should be interpreted as. To describe how this can be done, we shall initially consider the channel distortion to be $h(k) = \delta(k)$ (where $\delta(k)$ denotes the Kronecker delta[1]). This has the effect that the received signal is the same as the input signal, except for the additive noise: $r(k) = b(k) + n(k)$. In this case, a natural choice of detector would be

$$\hat{b}(k) = \operatorname{sign}\{r(k)\} = \begin{cases} -1 & r(k) \leq 0, \\ 1 & r(k) > 0. \end{cases} \tag{3}$$

This simple detector is actually optimal (minimum number of bit errors) under the given premises, but how should one proceed when $h(k)$ also introduces a distortion?

It is possible to derive an optimal receiver also when the channel is distorting the signal, but such a receiver is non-linear and too complex for the problem at hand. We shall thus limit ourselves to linear receivers. A linear receiver is a time-discrete filter followed by the detector given in (3). To further simplify the detector, we shall only consider causal FIR filters. Let the FIR filter be denoted by w. This yields the receiver

$$\hat{b}(k) = \operatorname{sign}\Big\{ \sum_{l=0}^{L} w(l) r(k-l) \Big\}, \tag{4}$$

where L is the order of our equalizing filter w. Given an appropriate choice of filter order, and well-chosen filter coefficients $w(l)$ for $l = 0, \ldots, L$, this receiver can be found to work quite well in most cases. Note that the channel and the equalization filters can have (and generally will have) different filter orders. Also keep in mind that a filter of order L has $L + 1$ filter taps.

Design of the equalizer

To design the equalizer filter, one must in some way have knowledge of how the channel is distorting the transmitted signal. In order to assure good performance, the equalizer should be designed such that its output signal

$$z(k) = \sum_{l=0}^{L} w(l) r(k-l) \tag{5}$$

[1]Kronecker delta function: $\delta(0) = 1$ and $\delta(k) = 0$ for all $k \neq 0$.

is as close to the input signal $b(k)$ as possible (you should state clearly in the report how you measure the performance of the equalizer).

In order to determine appropriate values for the equalizer filter taps $w(l)$, for $l = 0, \ldots, L$, the distortion and noise introduced by the channel needs to be measured in some way. This can be done by starting the transmission with a training sequence of symbols that are known to the receiver. In practice, this is performed by always transmitting the same sequence in the beginning of each data transmission. If the receiver knows both the transmitted signal and the received signal, it can estimate the filter coefficients w.

SpyPhone™ uses a training sequence of 32 symbols; that is, the first 32 symbols of $b(k)$ are known to the receiver, even before anything has been transmitted. These symbols can be used to make sure that

$$\sum_{l=0}^{L} w(l)r(k-l) \approx b(k), \quad \text{for } k = L+1, \ldots, 32. \tag{6}$$

Observe in (6) that with an equalizer of order L, we can only measure the equalizer performance for $k = L+1, \ldots, 32$ (why can't we use $k = 1, \ldots, L$ without making an additional assumption on the transmission?). Thus, the choice of the order L is a consideration between having a detailed model (large order) and having enough equations in (6) to estimate the filter coefficients with good precision.

Description of the Matlab environment

The encoder and decoder of the decryption system will be available, and can be downloaded from the course homepage. The two functions are

```
[key,cPic] = encoder(pic)
```

and

```
dPic = decoder(key,cPic)
```

The input of the function `encoder(pic)` is a picture and the output is the encrypted picture and the decoding key that you need for the decryption. Thus, the vector `key` corresponds to a realization of the signal $b(k)$ (including the training sequence). The output of the encoder can be used as input to the decoder `decoder()`, which will return the original picture. In the problem at hand, only the encrypted picture `cPic` and a distorted version of `key` will be available. Hence, you will have to construct an equalizer and a detector that reconstructs the decoding key, before you can decrypt `cPic`. Observe that you should send the whole key (including the training sequence) as an input to `decoder()`.

You will also have access to two data files, `spydata.mat` and `training.mat`, which can be downloaded from the course homepage and loaded into Matlab using the command `load`. The file `spydata.mat` contains the variables

```
cPic     - encoded picture,
received - received signal.
```

Observe that the main part of this project will consist of reconstructing `key` from the received signal in `received` (that corresponds to $r(k)$).

The other file, `training.mat`, contains

```
training - training sequence,
```

which is a vector with the training sequence (i.e., $b(k)$ for $k = 1, \ldots, 32$).

Finally, it is important to point out that only the *key* has been transmitted over the communication channel. The encrypted picture in `cPic` is known perfectly (without distortion and noise)!

Preparative assignments (No solutions need to be provided)

In order to get to know the encryption system and the digital communication model, we suggest you go through the following preparative assignments before solving the final assignment.

- Construct a simulation environment where you first load an image (for example, `bond.jpg` or `kth.jpg` on the homepage) using the command

  ```
  pic = imread('kth.jpg');
  ```

  Then you encode the image using `encoder()` and decode it again using `decoder()`. Finally, you can look at the decoded picture with the command `image(dPic)`. If the image is shown with wrong proportions, you may write `axis square` to make it a square. When you get all of this to work, you have system that corresponds to $\sigma^2 = 0$ and

  $$h(k) = \begin{cases} 1, & k = 0, \\ 0, & \text{otherwise}. \end{cases}$$

- Next, you should try to distort the signal `key` by transmitting it over a noiseless channel with $\sigma^2 = 0$ and

$$h(k) = \begin{cases} 1, & k = 0, \\ 0.7, & k = 1, 2, \\ 0, & \text{otherwise.} \end{cases}$$

  This can be done using the commands `filter` or `conv`, but make sure that the output and input are vectors of the same size and that the output contain what it should. Next, you should design an equalizer that tries to remove the effect of the channel. Choose an appropriate measure of the equalization performance and try to find a good filter order L that minimizes it (for example, minimization of the sum of MSEs between the left and right hand side of (6) for all k). Then add a detector, like the one in (3) and (4). The equalizer and detector is working when the reconstructed key makes it possible to decode the picture with good quality.

Assignments

1. Use the data in `spydata.mat` and `training.mat` and try to reconstruct the key using an appropriate type of FIR equalizer and filter order. Motivate your choice of filter order by showing that it maximizes the performance (choose some appropriate quality measure, for example the number of detection errors). The motivation should contain both visual and numerical results, and should be presented in a way that you convince the reader that you have made a good choice of filter order.

2. Decode the image using the reconstructed key $r(k)$ (the output of the detector) with your choice of filter order. *What does the image show?*

3. Introduce random bit errors in the reconstructed key. Investigate the number of bit errors that the key can contain before the picture becomes "impossible" to decode.

Observe that the quality of the decoded picture can easily be measured visually. Since we are using a linear equalizer, instead of an optimal equalizer, we can not expect to reconstruct the key perfectly. Thus, a good equalizer will give a decoded picture that is clear, while a bad equalizer will give a picture with many errors.

Hint: There are many useful Matlab commands that can be used, for example `filter`, `sign`, `mldivide`, `convmtx`, `xcorr`, and `conv`.

Presentation

The project assignment should be presented in a written report (2-3 pages long). The written report should be organized as is done with technical publications and should be written in Swedish or English. Handwritten reports are not acceptable!

Do not just state your answers together with the questions but include the answers in the text. Explain in words how the problem was solved (especially in mathematical principles) and what the results were. MATLAB code is in many cases not necessary at all. The report should be written such that a fellow student can understand and reproduce your results without having access to this instruction. Support all your conclusions with either numerical and/or graphical results (the latter should be clear by just looking at the figure). In case you are using sophisticated in-built MATLAB functions provide an explanation of how they work in your own words.

The report will be given the grade Pass or Fail. The grading includes clarity of presentation, content and layout. If the report gets the grade Fail, it can be revised and submitted a second time. If the revised report also fails, you will have to redo the project next year. The assignment should be performed individually or in groups of two students (recommended!). It is allowed to discuss (orally, without pen and paper) the problem formulation and different methods of solving it with other groups, but you are not allowed to share neither calculations, Matlab code, nor images with other groups! The workload for this assignment is supposed to be about 25h (1 credit) per student.

Please hand in the report via Canvas.