Practice 2.1: Signal Analysis: ASK simulation in Matlab

Alemón Pérez Alejandro, Álvarez Zamora Óscar Eduardo, Gallegos Ruiz Diana Abigail, Rojas Gómez Ian

***Abstract*—The physical layer is the lowest layer. This layer provides mechanical, electrical and other functional aids available to enable or disable; they maintain and transmit bits about physical connections. The techniques used are called technical transmission processes. Its physical layer digital bit transfer is accomplished on a wireline or cable-free transmission path.**

I. Motivation

To have a better understanding of the physical layer by developing simulations of a communication signal through a medium, including the noise it may have, to predict its behavior at the receiver in a similar way to real-life measurements.

II. Objectives

- Implement an ASK (mod-demod) transmission system simulation with different SNR levels.
- Compare and discuss the graphs seen in MatLab from the recovered signals.

III. Introduction

Physical layer

As we have mentioned, at physical layer, digital bit transfer is accomplished on a wireline or cable-free transmission path. The sharing of a transmission medium can be carried out on this layer by static multiplexing and dynamic multiplexing. This requires not only the specifications of certain transmission media (for example, copper cable, fiber optic cable, power grid) and the definition of connectors further elements. Furthermore, it must be resolved at this level, in what way a single bit to be transmitted.

This means the following: In computer networks today information is generally transmitted in the form of bit or symbol sequences. In copper cables and radio transmission, however, are modulated high frequency electromagnetic waves, the information carrier, in the optical waveguide light waves of a certain wavelength or different. The information carrier know no bit strings, but can take a lot more different states than just 0 or 1. For each type of transmission must therefore encoding are defined. That is due to the specification of the physical layer of a network.

Services in this layer.

- Bit-by-bit or symbol-by-symbol delivery
- Modulation
- Circuit switching
- Line coding
- Multiplexing
- Physical network topology, like bus, ring, mesh or star network

ASK modulation

Amplitude Shift Keying Theory This type of modulation comes under Digital Modulation schemes. In ASK, it requires two input signals, First input is binary sequence signal and the second input is carrier signal.

Some amplitude shift keying applications are mentioned below.

- Low-frequency RF applications
- Home automation devices
- Industrial networks devices
- Wireless base stations

Thus, Ask (amplitude shift keying) is a digital modulation technique to increase the amplitude characteristics of the input binary signal. But its drawbacks make it so limited. And these drawbacks can be overcome by the other modulation technique which is FSK.

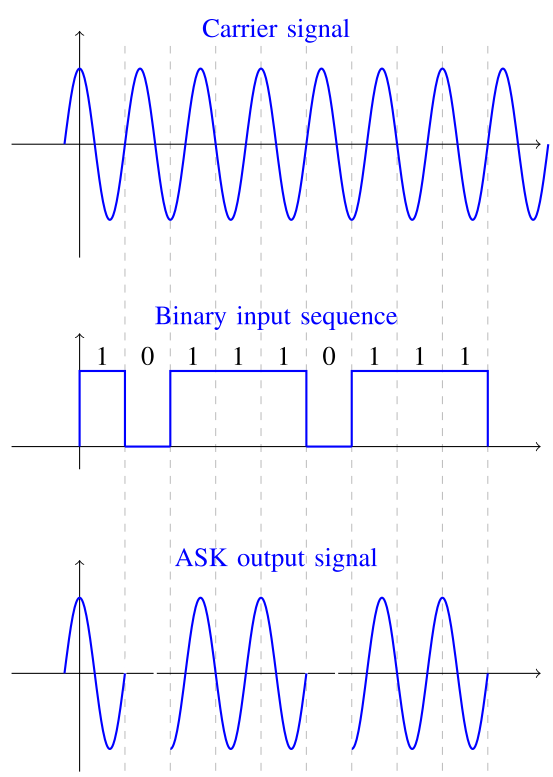


Fig. 1: ASK modulation

For digital signals, bit errors may be introduced, such that a binary 1 is transformed into a binary 0 or vice versa. In this practice, various impairments will be analyzed by how they may affect the information-carrying capacity of a communication link.

The most significant impairments are:

- Attenuation
- Delay
- Noise

Attenuation
The strength of a signal falls off with distance over any transmission medium. For guided media, the attenuation, is generally exponential and thus is typically expressed as a constant number of **decibels per unit distance**. For unguided media, attenuation is a more complex function of distance and the makeup of the atmosphere.
Delay
Delay distortion occurs because the velocity of propagation of a signal through a guided medium varies with frequency. Thus various frequency components of a signal will arrive at the receiver at different times, resulting in phase shifts between the different frequencies. This effect is referred to as delay distortion because the received signal is distorted due to varying delays experienced at its constituent frequencies.
Noise
For any data transmission event, the received signal will consist of the transmitted signal, modified by the various distortions imposed by the transmission system, plus additional unwanted signals, referred to as noise, that are inserted somewhere between transmission and reception. Noise is the major limiting factor in communications system performance.

Some important equations used in this practice:
Throughput probability

$$TP = \frac{Frames\ suc}{slots\ *\ TimeEachSlot}$$

Error probability (Pe) or BER

$$Pe = 1 - \left(\frac{FramesSuc}{TotalFrames}\right)$$

Outage probability

$$OP = \frac{FramesSuc - TotalFrames}{TotalFrames}$$

Age of Information (AoI)

(Current Message Arrival Time)
– (Previous Message Arrival Time)

Path Loss

$$L_{dB} = 10\log_{10}\frac{P_{in}}{P_{out}}$$

https://osi-model.com/physical-layer/
https://www.elprocus.com/amplitude-shift-keying-ask-working-and-applications/

IV. Development

To begin with, it was decided to treat the message as a .txt document since, using the dec2bin function, the characters read from the file are converted to an 8-bit vector.

```
close all;
clear all;
clc;

pathImages = 'SecondPractice/images/';
fileID = fopen('text.txt','r');%%Documento a leer
formatSpec = '%c';%%Formato del documento a leer
s = fscanf(fileID,formatSpec);%Texto leido
binary = dec2bin(s);%%Conversion binaria del texto
binary = reshape(dec2bin(s, 8).'-'0',1,[]);%Cambio
    a forma de vector
fclose(fileID);
```

The carrier frequency was set to 5 kHz, the pulse width to 1 ms and the sampling frequency as seen below.

```
fc = 5e3;
sequence = binary;
pulsesize = 1e-3;
ts = pulsesize/10;
fs = 1/ts;
tmax=length(sequence)*pulsesize;
t = 0:ts:tmax;

figurePlot = figure("Name", "Figura1");
tiledlayout('flow')
nexttile
plot(t,m)
axis([t(1) t(end) min(m)-0.1 1.1*max(m)]);
title("Message signal");
ylabel("m(t)");
xlabel("Time (t)");
grid on;
nexttile
plot(t,c)
axis([t(1) t(end) 1.1*min(c) 1.1*max(c)]);
title(" Carrier signal");
ylabel("c(t)");
xlabel("Time (t)");
grid on;
%saveas(figurePlot, [pathImages, 'Test1.png']);
```

The spectrum of the carrier looks as follows:

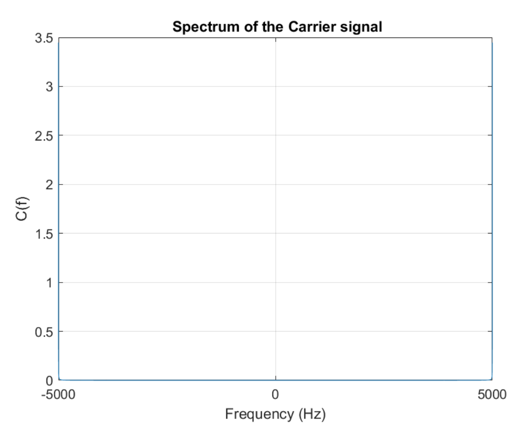


Fig. 2

The message signal and carrier are shown below.

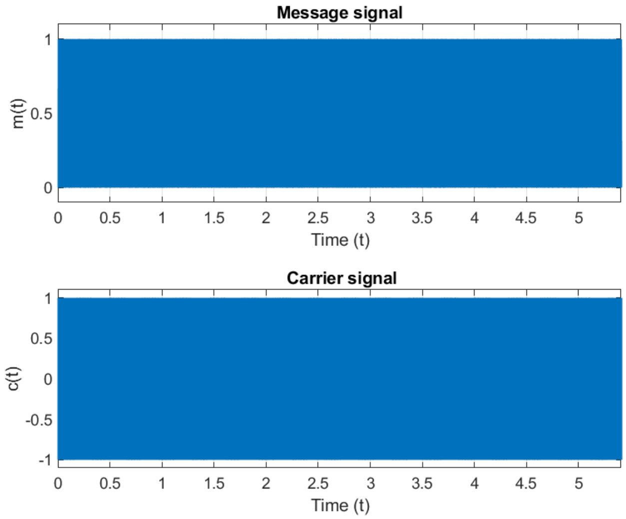


Fig. 3

For modulation:

```
%Modulation
ask = m.*c;
plot_image('Modulated signal', t, ask, [t(1) t(end) 1.1*min(ask) 1.1*max(ask)], 'Modulated signal', 'ASK(t)', 'Time (t)', pathImages, 'senial_modulada.png');

%Espectro
N = fs*t(end);
f = (-fs/2):(fs/N):(fs/2);
M = fftshift(fft(m))*ts;

plot_image('Frequency Spectrum of the Message signal', f, abs(M), [], 'Spectrum of the Message signal', 'M(f)', 'Frequency (Hz)', pathImages, 'senial_moduladora_espectro.png');

C = fftshift(fft(c))*ts;
plot_image('Frequency Spectrum of the Carrier signal', f, abs(C), [], ' Spectrum of the Carrier signal', 'C(f)', 'Frequency (Hz)', pathImages, 'senial_portadora_espectro.png');

asK = fftshift(fft(ask))*ts;
plot_image('Frequency Spectrum of the Modulated signal', f, abs(asK), [], 'Spectrum of the Modulated signal', 'asK(f)', 'Frequency (Hz)', pathImages, 'senial_modulada_espectro.png');
```

The spectrums of modulation looks as follows:

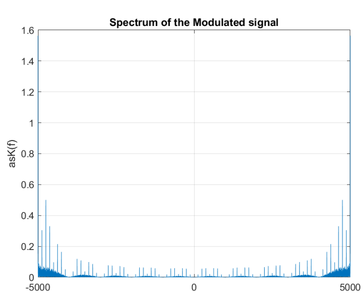


Fig. 4

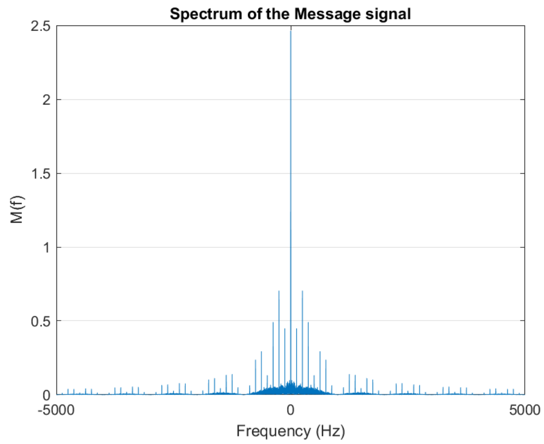


Fig. 5

The modulated signal looks as follows:

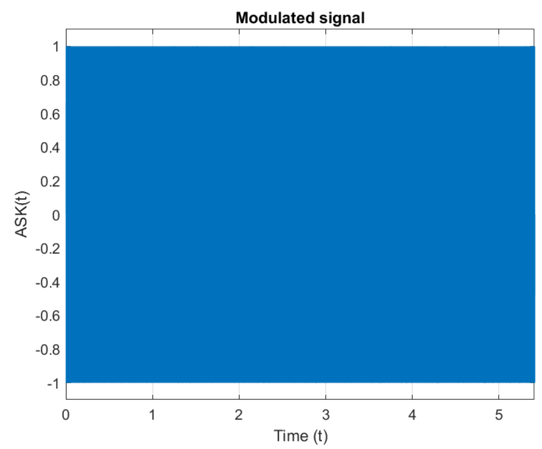


Fig. 6

A function was made in a .m for the coherent demodulation, in this case, it was decided to call it as "clean" because the signal did not contain any impairments. The function `demodulacion_no_coherente.m` can be seen on the appendix.

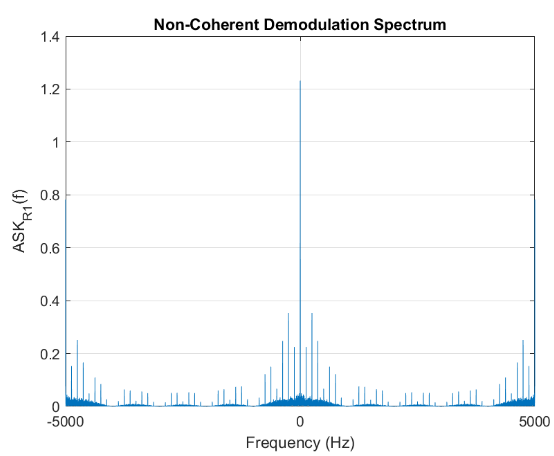


Fig. 7: Spectrum of the recovered signal

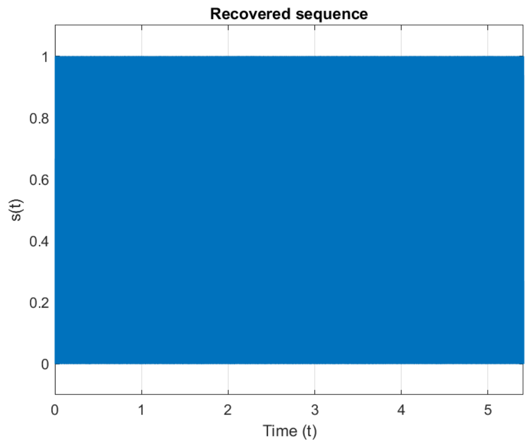


Fig. 8: Recovered sequence

The message recovered at the .txt is:

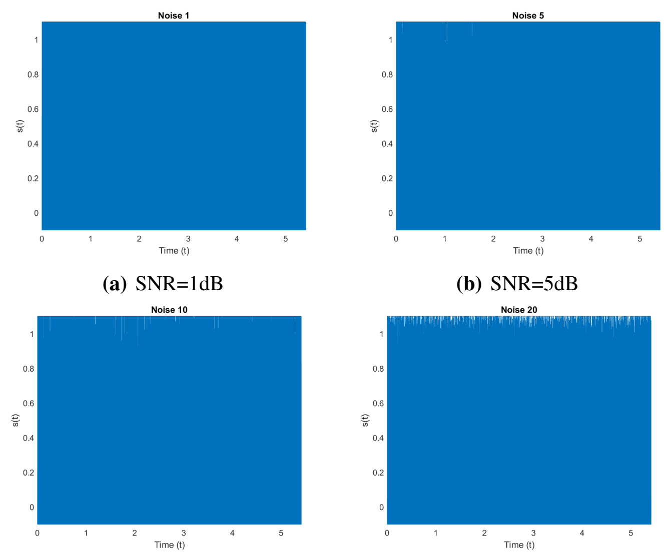
```
You have got to be kidding me", the bouncer said, folding his arms across his massive chest. He stared down at the boy in the red zip-up jacket and shook his shaved head. "You can not bring that thing in here". The fifty or so teenagers in line outside the Pandemonium Club leaned forward to eavesdrop. It was a long wait to get into the all ages club, especially on a Sunday, and not much generally happened in line. The bouncers were fierce and would come down instantly on anyone who looked like they were going to start trouble. Fifteen-year-old Clary Fray, standing in line with her best friend, Simon, leaned forward along with everyone else, hoping for some excitement.
```

For impairments, we used the function `awgn()` from Matlab.Both attenuation and AWGN noise were contained in a function "for", for each of the SNR values.

```
for SNR = [1 5 10 20]
modulatedSignalASKnoise = awgn(ask,SNR,'measured');
plot_image('Recovered signal', t, modulatedSignalASKnoise, [t(1) t(end) -0.1 1.1], convertContainedStringsToChars(strcat(str, string(SNR))), 's(t)', 'Time (t)', pathImages2, convertContainedStringsToChars(strcat(string(SNR) ,str1)));
signalAten=exp(-alpha.*z).*exp(-i*Beta.*z).*modulatedSignalASKnoise;

signalAten = interference(signalAten, t); % over time
plot_image('Attenuation after noise and interference over time', t, signalAten, [], convertContainedStringsToChars(strcat(str2, string(SNR))), 'signalAten(t)', 'Time (t)', pathImages3, convertContainedStringsToChars(strcat(string(SNR),str3)));
plot(z,signalAten); % over distance
plot_image('Attenuation after noise and interference over distance', t, signalAten, [], convertContainedStringsToChars(strcat(str4, string(SNR))), 'signalAten(t)', 'Time (t)', pathImages3, convertContainedStringsToChars(strcat(string(SNR),str5)));
[ask_c, h] = filter_signals(tmax,ts,fc/2, signalAten);
demodulacion_no_coherente(ask_c, t, ts, pathImages, pulsesize, [num2str(SNR), 'snr'], tmax, fc, f);

end
```

The following shows how the modulated signal spectrum are modified according to a different value of SNR, and attenuation taking into account the air as the transmission medium.

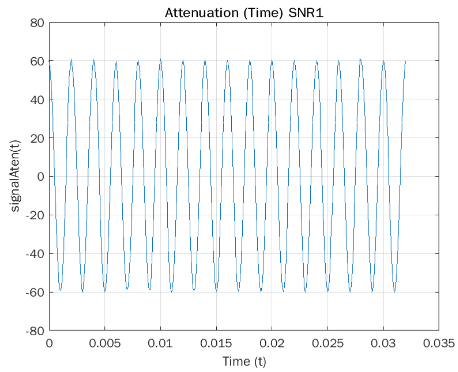
Noise



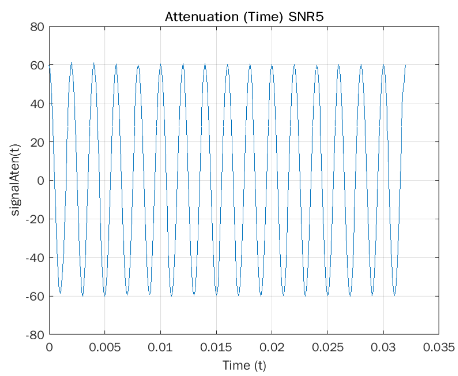
(a) SNR=1dB **(b)** SNR=5dB **(c)** SNR=10dB **(d)** SNR=20dB

Fig. 9: Noise

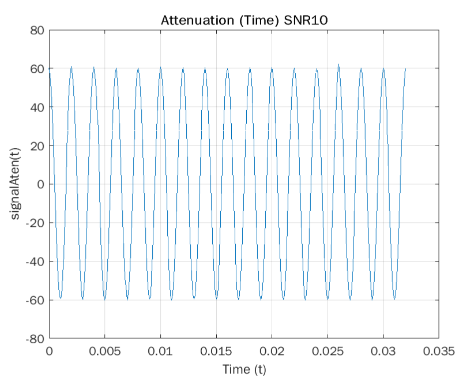
Attenuation after noise and interference over time



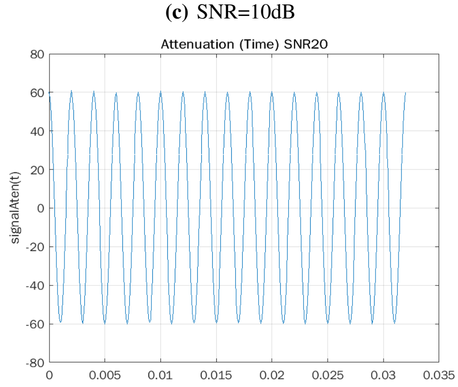
(a) SNR=1dB



(b) SNR=5dB



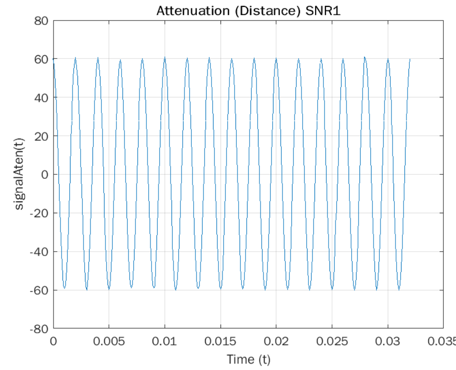
(c) SNR=10dB



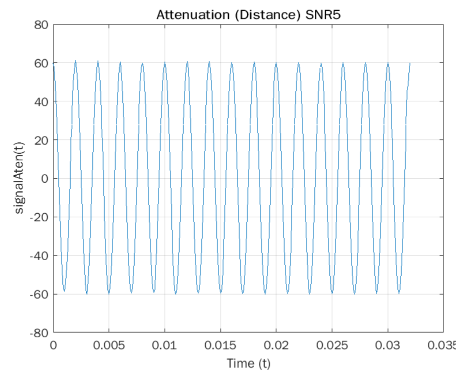
(d) SNR=20dB

Fig. 10: Attenuation after noise and interference over time

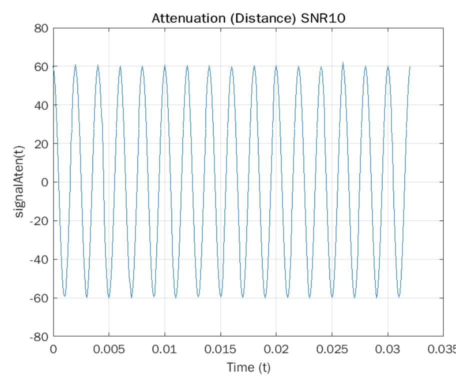
Attenuation after noise and interference over Distance



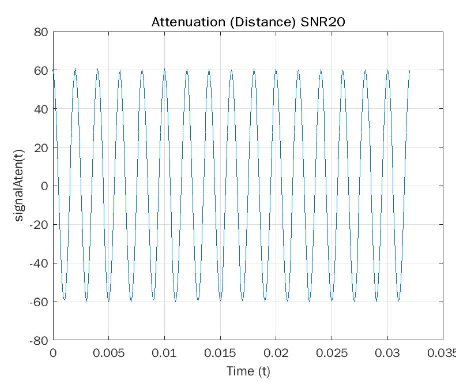
(a) SNR=1dB



(b) SNR=5dB



(c) SNR=10dB



(d) SNR=20dB

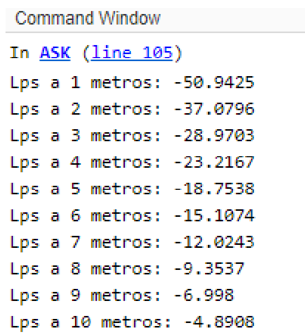
Fig. 11: Attenuation after noise and interference over distance

Due to the impairments , a band-pass filter was developed in the function `filter_signal` Which can also be reviewed in the appendix.

Path loss was calculated with the following function:

```
function Lps = pl(lambda, A)
d = 1:1:10; %distancia en metros

Pt= 10*log10(A/(1e-03)); %Esta en dbm
Gt = 3; %Esta en dbi
Gr = 3; %Esta en dbi

for i = d

Pr = Pt + Gt + Gr + 20*log10(lambda) - 20*log10(4*
    pi*i);
Lps = Pt - Pr;
display(['Lps a ', num2str(i), ' metros: ',
    num2str(Lps)]);
end

end
```

In the console it was obtained:



Separate functions were implemented for each of the parameters mentioned.

Attenuation.m

```
function signalAten = attenuation(s,t)
%%Atenuacion de la senial
taten=t;
distMax=10;
zStep=1*distMax/length(taten);
z=0:zStep:distMax;%%definimos los m para la
    formula de atenuacion
z=z(1:end-1);
alpha=0.2;%%aire
c=3e8;
lambda=c/fc;
Beta=2*pi/lambda;
signalAten=exp(-alpha.*z).*exp(-i*Beta.*z).*s;%
end
```

Age of information

```
function ageOfInformation = aoi(bits0,bits1, ts)
len = length(bits0);
succ = 0;
train = 1;
cmat = 0;
pmat = 0;
aoiAux = 0;

for i = 1:len

if bits0(i) == bits1(i)
succ = succ+1;
end

if i == 8*train

if(succ == 8)

if cmat ~= 0

pmat = cmat;

end

cmat = train;

aoiAux = (cmat-pmat)*ts*8;

if aoiAux > ageOfInformation
ageOfInformation = aoiAux;
end

end

succ = 0;
train = train + 1;

end
end

end
```

ep.m

```
function ep = ep(success,total)
ep = 1 - success/total;
end
```

op.m

```
function op = op(success,total)
op = (success-total)/total;
end
```

top.m

```
function top = top(success, total, ts)
top = (success*ts)/(total*ts);
end
```

V. Results

In the ideal case, the following was obtained:

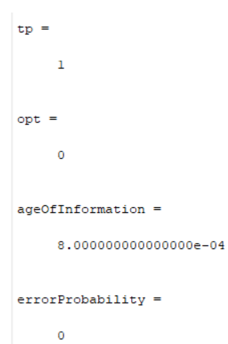


Fig. 12

For each SNR value, the above parameters were calculated and the results are shown in the console below.

```
tp =

     0


opt =

    -1


ageOfInformation =

   0.038400000000000


errorProbability =

     1
```

(a) SNR=1dB

```
tp =

     0


opt =

    -1


ageOfInformation =

   0.022400000000000


errorProbability =

     1
```

(b) SNR=5dB

```
tp =

     0


opt =

    -1


ageOfInformation =

   0.011200000000000


errorProbability =

     1
```

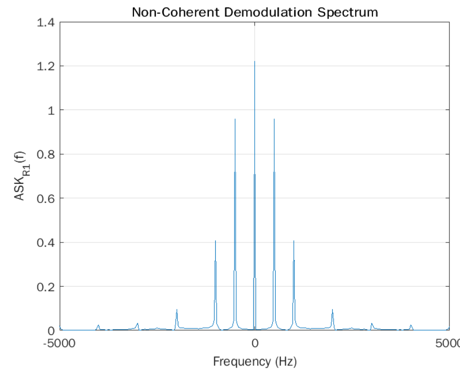
(c) SNR=10dB

```
tp =

     0


opt =

    -1


ageOfInformation =

   0.009600000000000


errorProbability =

     1
```

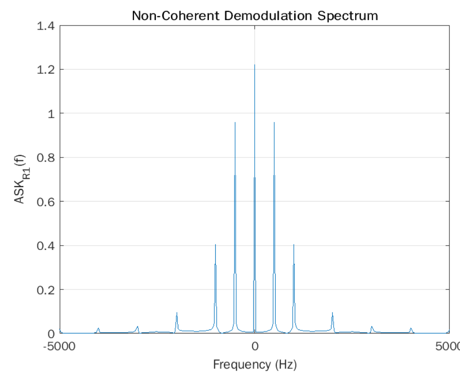
(d) SNR=20dB

Fig. 13: Parameters

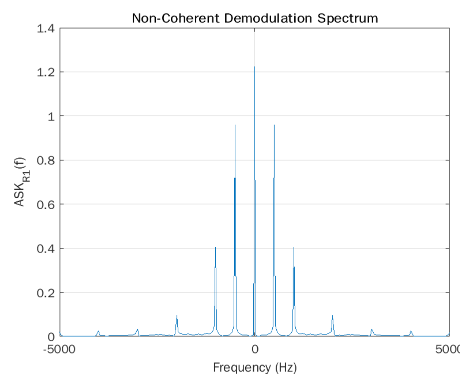
In the following figures we can see how the spectra and the recovered sequences are modified due to different impairment values.



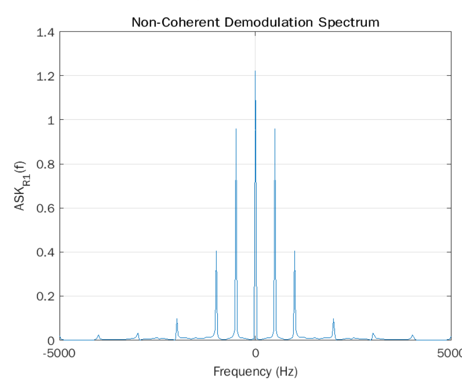
(a) SNR=1dB



(b) SNR=5dB

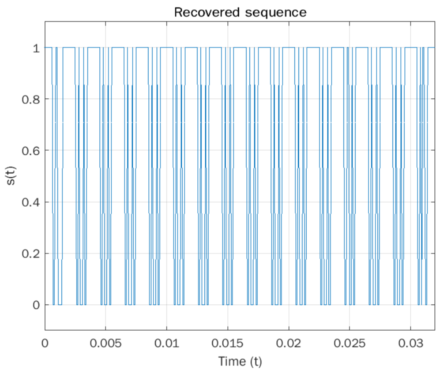


(c) SNR=10dB

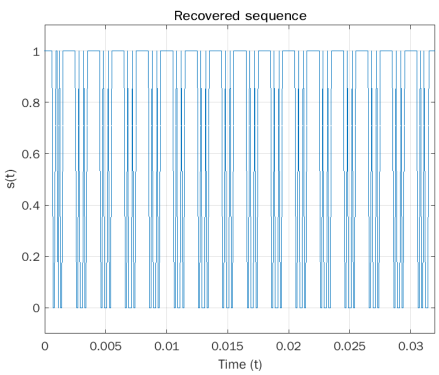


(d) SNR=20dB

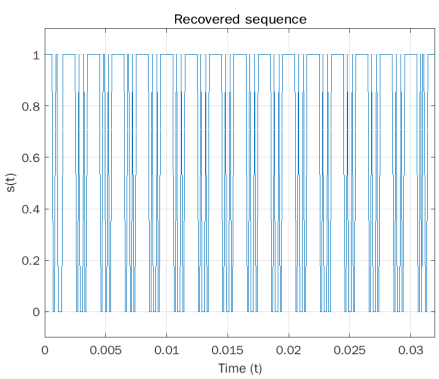
Fig. 14: Spectrum of the recovered signal



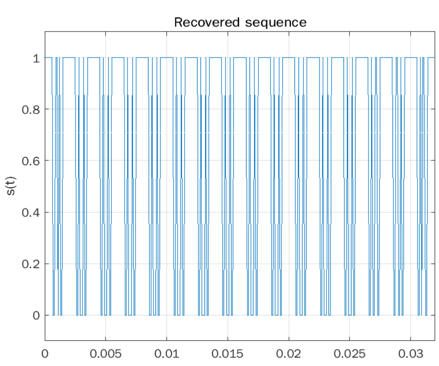
(a) SNR=1dB



(b) SNR=5dB



(c) SNR=10dB



(d) SNR=20dB

Fig. 15: Recovered sequences

Due to the `for` statement, the recovered signal, in this case the message, that has an attenuation over 10 meters with an SNR of 20 dB looks like the following:

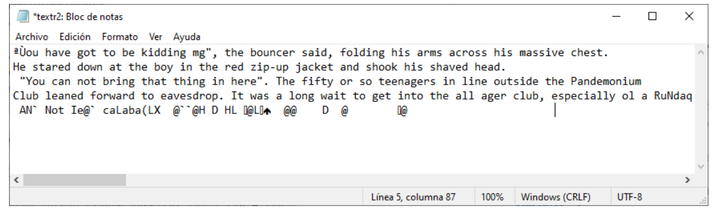


Fig. 16: Recovered message

VI. Conclusions

We could see throughout the practice the way the physical layer is the one in charge of the frames. As we had already pointed out in the previous practice, exactly this layer of the OSI model -and actually most layered models used for telecommunications- is the one who gets the information about the frames and bits. It gets the number of packets which arrived successfully and which were not.

As seen during classes, most times when using ASK frames are not getting entirely correctly, as it is not the most reliable modulation compared to some that are way more advanced and complex, that is because it is susceptible to the effects of the channel used for the transmission. There are certain affectations of the data that makes frames get errored. Luckilly, it is not only the physical layer's job to get those frames and the information contained, as the data link layer gets involved in this as well.

We have not seen enough of it, but there are certain techniques in order to get back the information contained within in the best way possible, eventhough those come with a cost that could be in bandwidth, speed or any other way. As we know there is no perfect way to make a conection and a transmission, so it is up to the question of whether it is necessary to get all the information correctly the first way around or we can afford losing some bits as those might not be as important or relevant.

VII. Appendix

Additional Functions used in this practice:

demodulacion_no_coherente.m

```
function s = demodulacion_no_coherente(ask, t, ts, pathImages, pulsesize, nameFile, tmax, fc, f, SNR)

coe = find(ask < 0);
ask_r1 = ask;
ask_r1(coe) = 0;

%figure("Name"," Non-Coherent Demodulation")
%plot(t,ask_r1);

plot_image(' Non-Coherent Demodulation', t, ask_r1, [], ' Non-Coherent Demodulation', 'ask_{r1}(t)', 'Time (t)', pathImages, ['Espectro_recuperada_no_coherente', nameFile,'.png']);

ASK_R1 = fftshift(fft(ask_r1))*ts;
[ask_f1, h] = filter_signals(tmax,ts,fc/2,ask_r1);
plot_image('Recovered signal', t, ask_f1, [], 'Recovered signal', 'ask_{f1}(t)', 'Tiempo (t)', pathImages, ['Espectro_recuperada_no_coherente', nameFile,'.png']);


plot_image('Non-Coherent Demodulation Spectrum', f, abs(ASK_R1), [], 'Non-Coherent Demodulation Spectrum', 'ASK_{R1}(f)', 'Frequency (Hz)', pathImages, ['Espectro_recuperada_no_coherente', nameFile,'.png']);

seq1 = seq(ask_f1);
seqrecu = seq2(seq1,pulsesize,ts);
str = char(bin2dec(reshape(char(seqrecu+'0'), 8,[]).'));
fileID = fopen('textr2.txt','w');
s = fwrite(fileID,str);
plot_image('Recovered signal', t, seq1, [t(1) t(end) -0.1 1.1], 'Recovered sequence', 's(t)', 'Time (t)', pathImages, ['senial_secuencia_recuperada_no_coherente', nameFile,'.png']);

end
```

filter_clean_snr.m

```
function [x_r, h]= filter_clean_snr(tmax,ts,fc,x)
th = -tmax/2:ts:tmax/2;
h = 4*fc*sinc(2*fc*th)*cos(2*pi*fc*th);
x_r = conv(x,h,'same')*ts;
end
```

filter_signals.m

```
function [x_r, h]= filter_signals(tmax,ts,fc,x)
th = -tmax/2:ts:tmax/2;
h = 4*fc*sinc(2*fc*th);
x_r = conv(x,h,'same')*ts;
end
```

interference.m

```
function [x_r, h]= filter_signals(tmax,ts,fc,x)
th = -tmax/2:ts:tmax/2;
h = 4*fc*sinc(2*fc*th);
x_r = conv(x,h,'same')*ts;
end
```

modulate.m

```
function m = modulate(t, sequence, pulsesize)
m = zeros(size(t));
for k = 1:length(sequence)
m = m + sequence(k)*(((k-1)*pulsesize) <= t & t < (k*pulsesize));
end
end
```

plot_image.m

```
function [] = plot_image(figureName, t, m, axisDim, titlePlot, ylabelString, xlabelString, pathImages, imageName)

figurePlot = figure("Name", figureName);
plot(t,m)
axis(axisDim);
title(titlePlot);
ylabel(ylabelString);
xlabel(xlabelString);
grid on;
saveas(figurePlot, [pathImages, imageName]);
end
```

seq.m

```
function seq = seq(recu)
seq = zeros(size(recu));
for k = 1:length(recu)
if(recu(k) > 0.6)
seq(k) = 1;
end
end
end
```

seq2.m

```
function seq2 = seq2(seq, pulsesize, ts)
tm = pulsesize/2;
t = tm/ts;
cont = 1;
for k = t+1:10:(length(seq)-1)
seq2(cont) = seq(k);
cont = cont+1;
end
end
```

succs.m

```
function succs = succesful(seqi,seqo)
wrong = 0;
succs = 0;
for k = 1:8:length(seqi)
for i = 1:8
if(seqo(i) ~= seqi(i))
wrong = 1;
end
end
if(wrong == 0)
succs = succs + 1;
end
end
end
```

Complete ASK.m file

```
close all;
clear all;
clc;

%LaTeX folders
pathImages = 'SecondPractice/images/';
pathImages2 = 'SecondPractice/images/Noise/';
pathImages3 = 'SecondPractice/images/Attenuation/';
pathImages4 = 'SecondPractice/images/Demodulation/';



fileID = fopen('text.txt','r');
formatSpec = '%c';%%Formato del documento a leer
```

```
s = fscanf(fileID,formatSpec);%Texto leido
binary = dec2bin(s);%%Conversion binaria del texto
binary = reshape(dec2bin(s, 8).'-'0',1,[]);%Cambio a forma de vector
fclose(fileID);


fc = 5e3;
A = 1; %Verificar este parametro%
sequence = binary;
pulsesize = 1e-3;
ts = pulsesize/10;
fs = 1/ts;
tmax=length(sequence)*pulsesize;
t = 0:ts:tmax;

%Portadora
c = A*cos(2*pi*fc*t);
m = modulate(t,sequence,pulsesize);

figurePlot = figure("Name", "Figura1");
tiledlayout('flow')
nexttile
plot(t,m)
axis([t(1) t(end) min(m)-0.1 1.1*max(m)]);
title("Message signal");
ylabel("m(t)");
xlabel("Time (t)");
grid on;
nexttile
plot(t,c)
axis([t(1) t(end) 1.1*min(c) 1.1*max(c)]);
title(" Carrier signal");
ylabel("c(t)");
xlabel("Time (t)");
grid on;
saveas(figurePlot, [pathImages, 'Test1.png']);
ask = m.*c;
plot_image('Modulated signal', t, ask, [t(1) t(end) 1.1*min(ask) 1.1*max(ask)], 'Modulated signal', 'ASK(t)', 'Time (t)', pathImages, 'senial_modulada.png');

%Espectro
N = fs*t(end);
f = (-fs/2):(fs/N):(fs/2);
M = fftshift(fft(m))*ts;

plot_image('Frequency Spectrum of the Message signal', f, abs(M), [], 'Spectrum of the Message signal', 'M(f)', 'Frequency (Hz)', pathImages, 'senial_moduladora_espectro.png');

C = fftshift(fft(c))*ts;
plot_image('Frequency Spectrum of the Carrier signal', f, abs(C), [], ' Spectrum of the Carrier signal', 'C(f)', 'Frequency (Hz)', pathImages, 'senial_portadora_espectro.png');

asK = fftshift(fft(ask))*ts;
plot_image('Frequency Spectrum of the Modulated signal', f, abs(asK), [], 'Spectrum of the Modulated signal', 'asK(f)', 'Frequency (Hz)', pathImages, 'senial_modulada_espectro.png');


str="Noise ";
str1="Noise.png";
str2="Attenuation (Time) SNR";
str3="Attenuation_Time.png";
str4="Attenuation (Distance) SNR";
str5="Attenuation_Distance.png";
str6="Demodulation.png";


demodulacion_no_coherente(ask, t, ts, pathImages, pulsesize, 'clean', tmax, fc, f, sequence);

%%%ADDING NOISE AWGN  & ATTENUATION

taten=t;
distMax=10;
zStep=1*distMax/length(taten);
z=0:zStep:distMax;%%definimos los m para la formula de atenuacion
z=z(1:end-1);
alpha=0.2;%%aire
c=3e8;
lambda=c/fc;
Beta=2*pi/lambda;

for SNR = [1 5 10 20]
modulatedSignalASKnoise = awgn(ask,SNR,'measured');
plot_image('Recovered signal', t, modulatedSignalASKnoise, [t(1) t(end) -0.1 1.1], convertContainedStringsToChars(strcat(str, string(SNR))), 's(t)', 'Time (t)', pathImages2, convertContainedStringsToChars(strcat(string(SNR) ,str1)));
signalAten=exp(-alpha.*z).*exp(-i*Beta.*z).*modulatedSignalASKnoise;%
signalAten = interference(signalAten, t); % over time
plot_image('Attenuation after noise and interference over time', t, signalAten, [], convertContainedStringsToChars(strcat(str2, string(SNR))), 'signalAten(t)', 'Time (t)', pathImages3, convertContainedStringsToChars(strcat(string(SNR),str3)));
plot(z,signalAten); % over distance
plot_image('Attenuation after noise and interference over distance', t, signalAten, [], convertContainedStringsToChars(strcat(str4, string(SNR))), 'signalAten(t)', 'Time (t)', pathImages3, convertContainedStringsToChars(strcat(string(SNR),str5)));
[ask_c, h] = filter_clean_snr(tmax,ts,fc/2, signalAten);
demodulacion_no_coherente(ask_c, t, ts, pathImages, pulsesize, [num2str(SNR), 'snr'], tmax, fc, f, sequence);
end

pl(lambda, A); %Path Loss
```

VIII. REFERENCES

- Physical Layer | Layer 1. (s. f.). The OSI-Model. https://osi-model.com/physical-layer/
- Agarwal, T. (2019, 11 septiembre). Amplitude Shift Keying : Circuit Diagram, Working and Its Applications. ElProCus - Electronic Projects for Engineering Students. https://www.elprocus.com/amplitude-shift-keying-ask-working-and-applications/