Communication Systems report, supervised by

Prof. Dr. Volker Strumpen at Rhine-Waal University of Applied Sciences.

Winter Semester 2021/2022.

Participants:

Abdalrahman Elmessiry, Matriculation number: **24628**
email: Abdalrahman.Elmessiry@hsrw.org,
Communication and information engineering bachelor.

Konstantinov Denis, Matriculation number: **27567**
email: Denis.Konstantinov@hsrw.org,
Communication and information engineering bachelor.

Responsibilities: we have to state that both of us worked as a single team on every aspect of the project and equally contributed in every single step towards the successful finishing of the project, modules implementation, testing, preparing, finalizing and debugging.

Table of contents:

SRD review

Introduction

For the first step we made a short code to choose from different mysteries by typing 1, 2 or 3 we can then select a particular mystery to execute, All three signals have next properties: SRRC - Square-Root Raised Cosine, Signals length and rolloff parameter, T_t is a period of signal before modulation, f_if is an intermediate frequency in hertz and f_s is a sampling frequency that is hertz. By running the first section of the code we plot sampled passband signals and get the plotspect of the mysteries with the amplitude/seconds and magnitude/frequency spectrums which are presented on figures 1, 1.2 and 1.3. As we can see from plots all three signals are different, the first mystery spectrum magnitude is carrier frequency centered at around $2.5x10^5$ hz and $-2.5x10^5$ hz, the second mystery is centered at $1x10^5$ and $-1x10^5$ Hz, the third mystery's signal has a noise starting from $-4x10^5$ up to $-2x10^5$ and from $2x10^5$ to $4x10^5$ Hz which we will have to get rid of later on during the matched filtering application. After we receive the initial signal first we input required parameters for the filter and then we apply FIR filter on each mystery, then we pass the result to the zero phase digital filter, the signal we have is a DT signal therefore zero phase filter is a good solution because we have the full signal as a finite sequence and the filter is actually capable to sort it out fast with no delay. We save the filtered signal as fr for post processing.

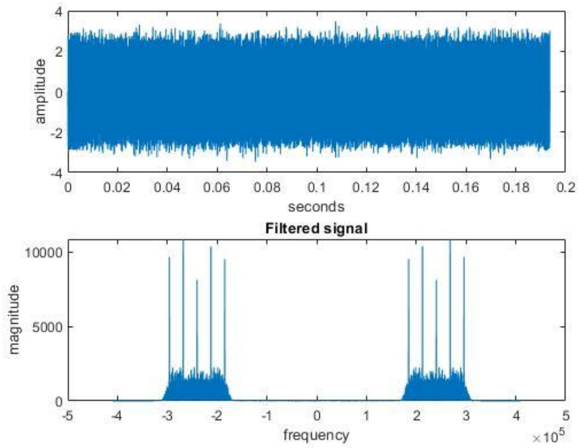

Figure 1. Mystery A. Amplitude, magnitude spectrum

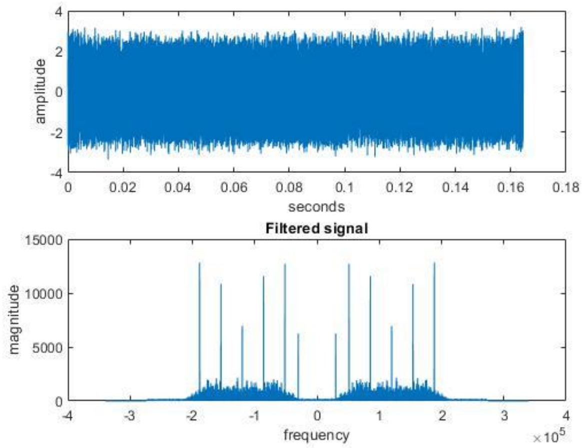

Figure 1.2. Mystery B, A.M. spectrum

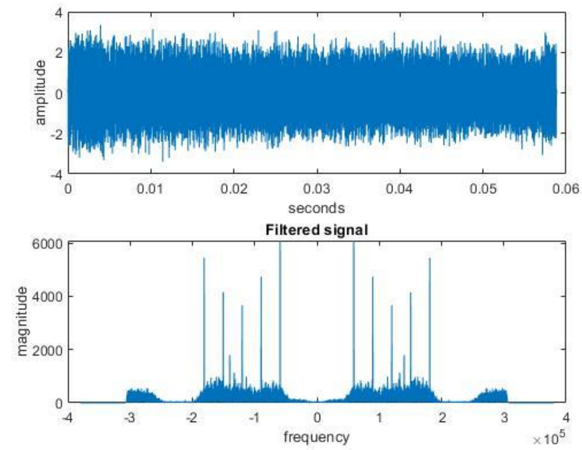

Figure 1.3. Mystery 3. A.M. spectrum

Carrier recovery

The successful demodulation means that our receiver can estimate the frequency offset and a phase offset of the modulated signal. By estimation of carrier frequency and adjusting gain the process of carrier recovery may be finished. SRD makes automatic adjustments for extra imprecisions by using adaptive parameters in order to estimate and extract the hidden phase offset between receiver and transmitter.

Costas loop is an algorithm that tracks the phase, in costas loop the first loop is tracking the frequency offset and the second loop adjusts the estimation of the phase, we used costas loop because it does not require a bandpass filter to be implemented as part of a downconverter on a receiver side. The code is presented on figure 2.1

First we initialized a two step sized algorithm with mu=0.07 and mu2=0.00018 which are called voodoo parameters, it took us some time to adjust them manually.Then we initialize the estimation vectors for theta and theta1 and buffers for LPS. As a result we have a new output for filters and the algorithm is now updating itself. By running the costas loop we have an output for different mysteries as shown on figures 2, 2.1 and 2.2. The output of theta almost converges on 0.12 for theta1 for the first mystery, on 0.12 for the second and on 0.06 for the third one. Therefore as a total output of the function we get frequency offset as for the first loop and phase offset as for the second loop for each mystery. After the carrier recovery step is done and we obtain the necessary phase and carrier offsets which then are added together to form an updated signal with cos(2*pi*f0*t) now we can continue further to demodulation. The costas loop code is based on costasloop.m and written an additional loop to figure out the phase offset.

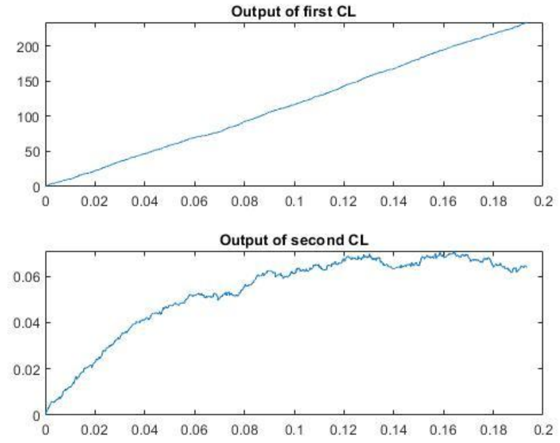

Figure 2. Mystery A

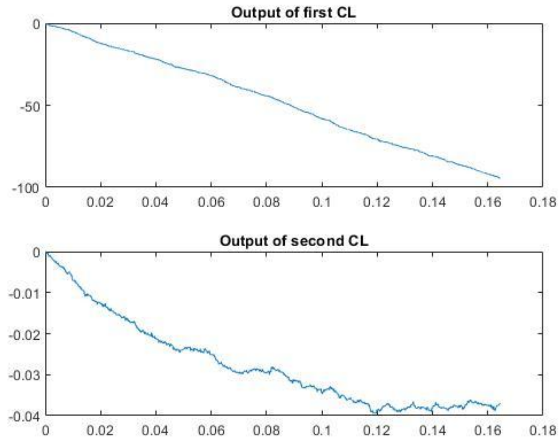

Figure 2.1. Mystery B

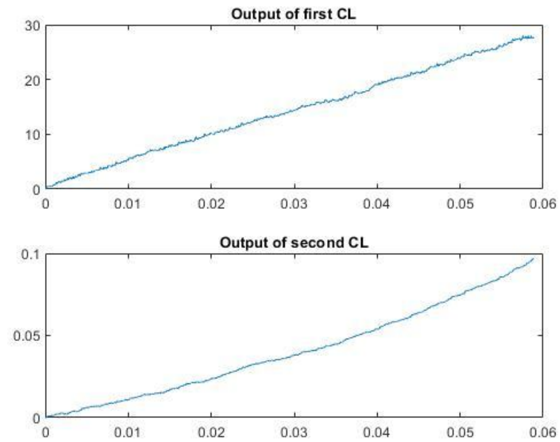

Figure 2.2.Mystery C

Demodulation

After we recover the signal with costas loop and make a frequency offset estimation and phase offset estimation then by a dot product of recovered frequencies and originally FIR filtered sequences we get the main recovered frequency/magnitude signals shown on the figures 3, 3.1 and 3.2 which are centered at frequency 0 now and initial signals or replicas are now shifted on sides. The main signal is now recovered, however we have to get rid of replicas in order to have a clear signal, therefore we can continue with a matched filter step.

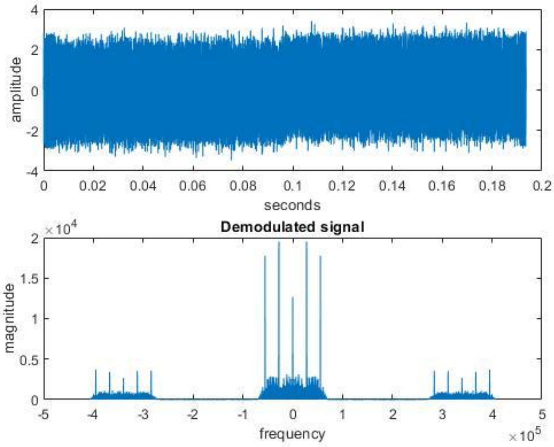

Figure 3 Mystery A

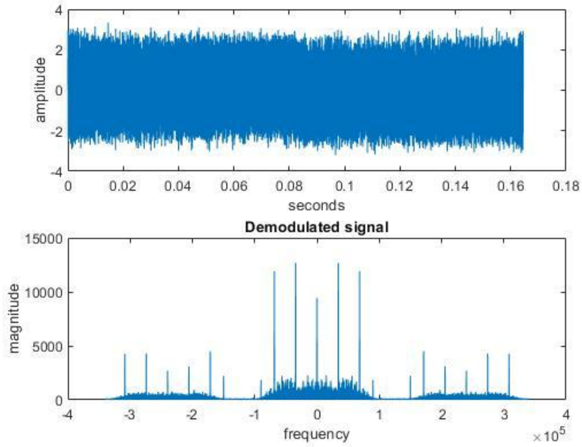

Figure 3.1 Mystery B

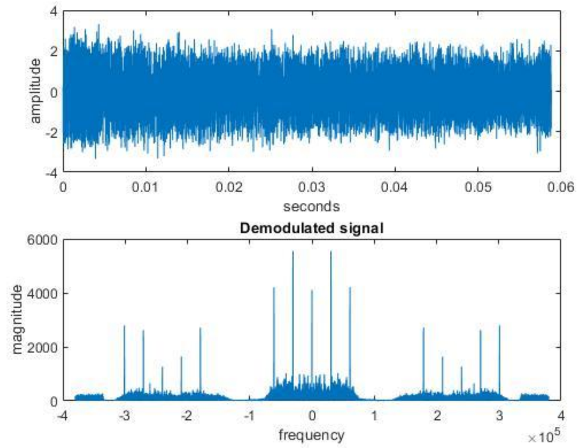

Figure 3.2 Mystery C

Matched filtering

First we sample intervals and time bases. After that we define a "time" vector. Next we demodulate received signals from the previous step. Then yet again we apply the LPF filter. After demodulation we filtered out extra replicas on sides shown on figures 4, 4.1 and 4.2. After applying a matched filtering, noise on the sides is now gone and we have a magnitude spectrum of signals that is now clear.

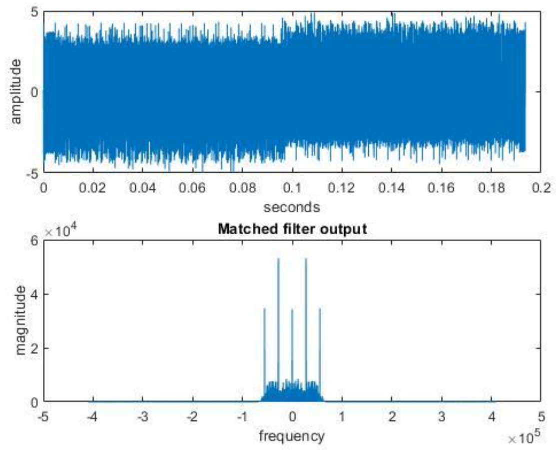

Figure 4. Mystery A. Match filter output

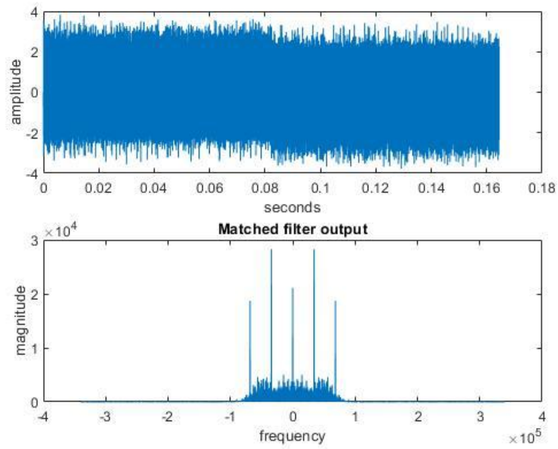

Figure 4.1 Mystery B. Match filter output

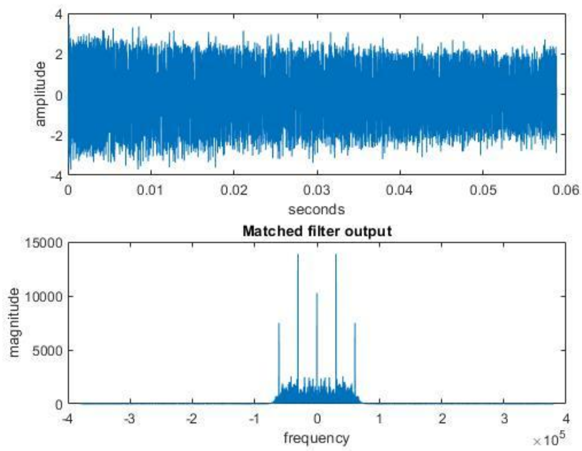

Figure 4.2 Mystery C. Match filter output

Time recovery

For the clock recovery, we used the Dual decision directed algorithm, dual because it acts similar to the Dual costas loop for the carrier recovery. The code is based on clockrecDD.m.

It is a procedure of finding the time offset using only the received signal. On the code we assign l to be its length and we define variable n as the rounded length of the signal after matched filtering over the oversampling rate.

The algorithm is initialized with an offset estimate of tau=0 and stepsize mu. The received signal is sampled at m times the symbol rate, and the while loop runs through the data, incrementing i once for each symbol (and incrementing tnow by m for each symbol). The offsets tau and tau1 are indistinguishable from the point of view of the algorithm.

The update term contains the interpolated value xs as well as two other interpolated values to the left and right that are used to approximate the derivative term. We aslo quantize xs to nearest 4-PAM symbol and we get values to the left and right by changing the sign of delta1 from plus to minus to then save and update both tau and tau1. As what we can see now on figures 5, 5.1 and 5.2 is the estimation of symbol values on 4-PAM scale.

The constellation diagrams for the second and third mysteries may show unpromising results the same as a timing converge estimate due to excessive divergence among symbol samples, the resulting output may be incorrect due to the fact that the 4 PAM constellation is too diverse. In order to fix that we have to apply an equalizer which is the next step.

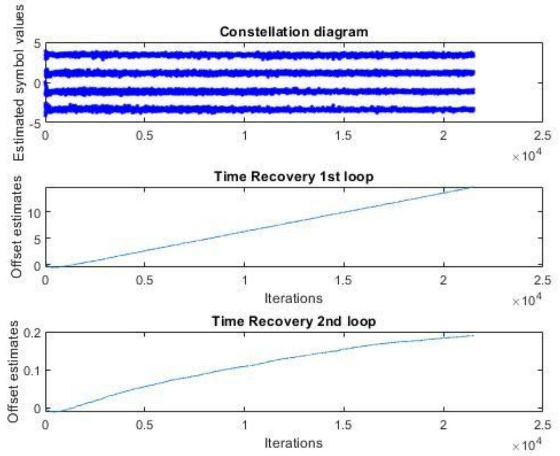

Figure 5. Mystery A

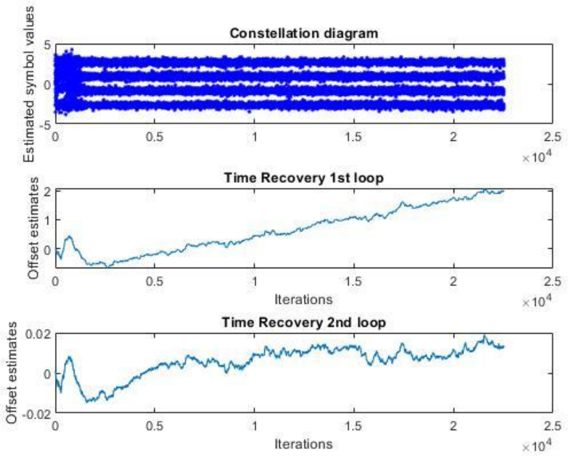

Figure 5.1. Mystery B

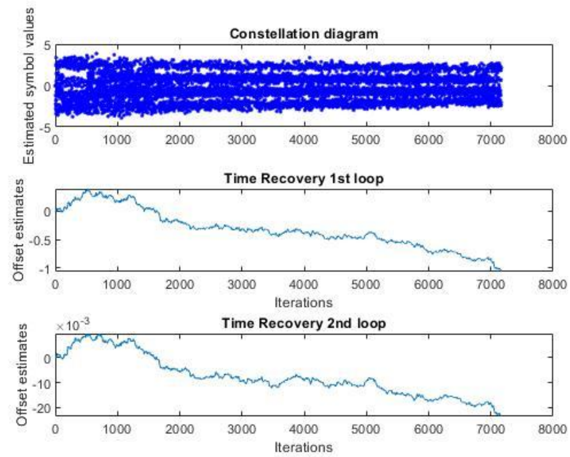

Figure 5.2. Mystery C

Correlation

Correlation deals with the time shifting frequencies of signals and the main purpose of which in communication systems is to synchronize signals in time. The process of correlation is to shift one signal sequence in time and check the difference in matching with another; in other words correlation plays a role of pattern matching algorithm. In order to do that we multiply signal points and sum them up at each shifting iterations, the function we're using is called xcorr and it does a cross correlation of a header and the signal we got from the time recovery module.

We save the location of the highest correlation of the header with the signal,

HeaderStartPosition = LengthOfSignal - CorrelationIndex + 1

then modulate it by 512 to get the position of the first header.
Afterwards the first frame is removed except for the last 2 letters '512-8' to fix a delay problem.

```
r4 =r3(headstart+512-8:end);
```

The next step here is to look for a phase offset of 180 degrees which has a negative value of max correlation so we then have to flip it by 180 degrees to remove the offset, which can be seen on figures 6.3, 6.4, 6.5

Then we're applying xcorr again to plot a new graph of the header correlation with data after fixing the 180 degree offset, show on figure 6, 6.1 and 6.2.

After the correlation is done and we now are able to distinguish where the header is and where the body of the message is we can proceed to the next and final step of decoding.

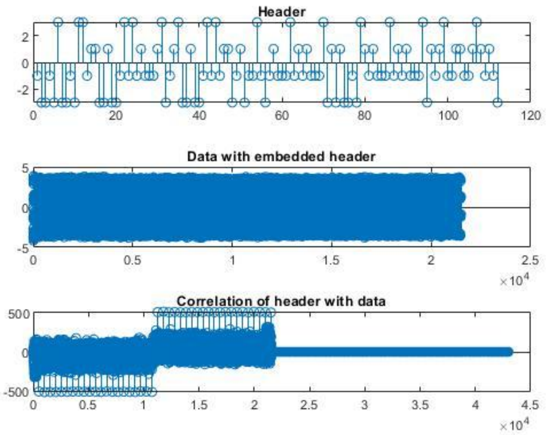

Figure 6 Mystery A

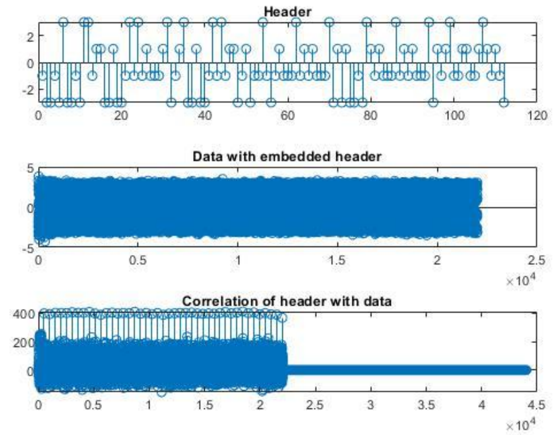

Figure 6.4 Mystery B After offset

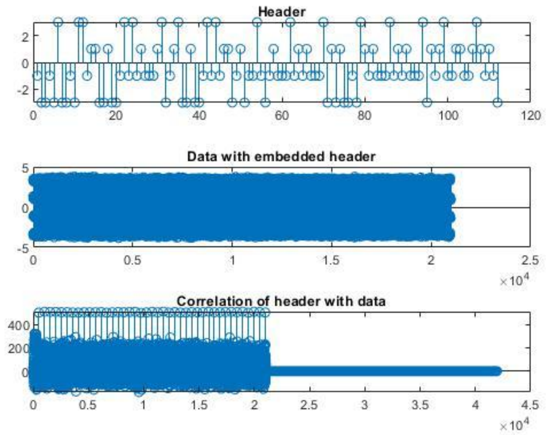

Figure 6.3 Mystery A after application of 180 degrees phase offset removal

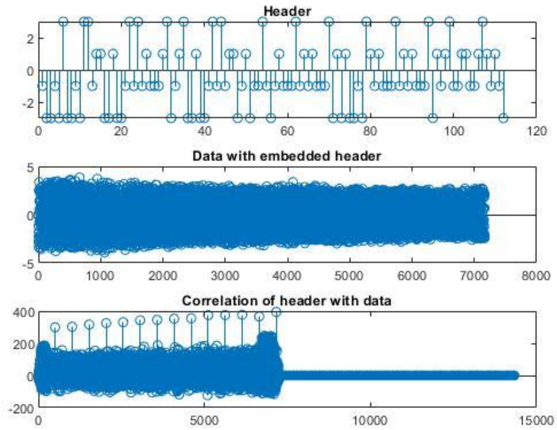

Figure 6.2 Mystery C

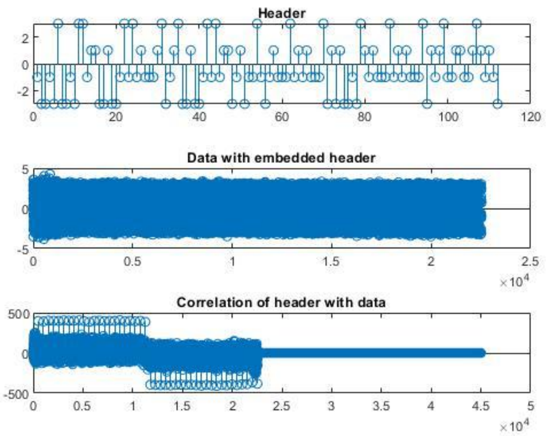

Figure 6.1 Mystery B

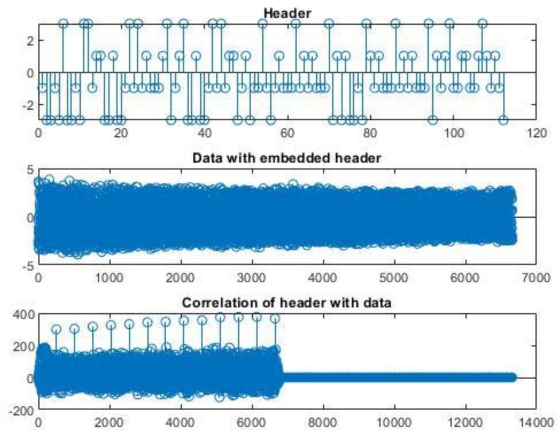

Figure 6.5 Mystery C after offset

Equalizer

On the basis of the equalization problem lies a problem of optimization. The main idea of an equalizer is to remove possible interferences and get rid of possible multipaths and distortions. For our project we decided to install an Blind Adaptive Decision-Directed Linear equalizer, because LMS was not working for us. DD equalizer adapts equalizer coefficients by making a difference of received signal and its output of the decision algorithm. The output of the equalizer is shown on the figures 7, 7.1 and 7.2.

The code is based on DDequalizer.m equalizer, except for quantalph was used instead of sign since we are using a blind adaptive DD equalizer.

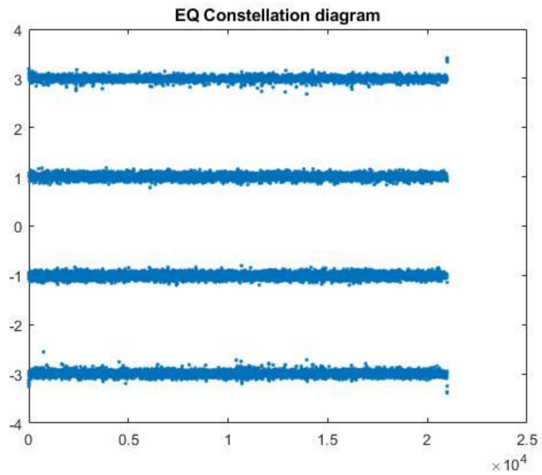

Figure 7 Mystery A

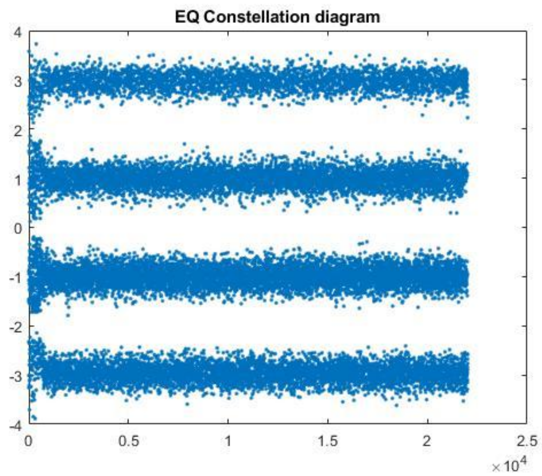

Figure 7.1 Mystery B

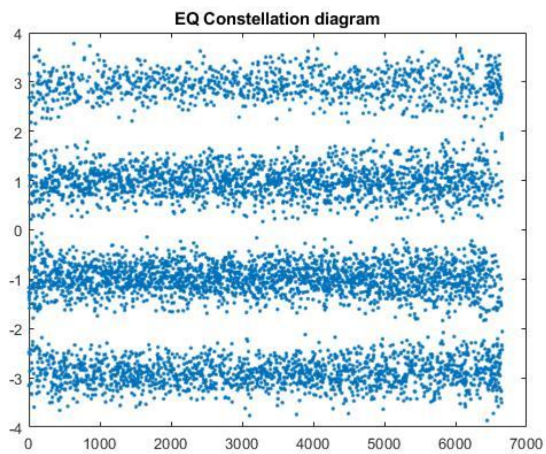

Figure 7.1 Mystery C

Decoding

Decoder converts the dt signal of symbols into text, where each symbol of ASCII table is 7 bits. We start by quantizing the signal using quantalph.m and then reconstructing from pam to letters using the pam2letters.m, and then the reconstructed message is ready to be printed. We also tried to use native2unicode instead of ASCII which we adjusted inside pam2letters.m however it gave no results.

Decoded result

We calculated that the current code is capable of decoding more than 90% of the signal for each mystery.

References

- Prof. Dr. Volker Strumpen, winter semester 2021 lectures.

- Software Receiver Design. Build your Own Digital Communication System in Five Easy Steps. AUTHORS:C. Richard Johnson, Jr, Cornell University, New YorkWilliam A. Sethares, University of Wisconsin, MadisonAndrew G. Klein, Worcester Polytechnic Institute, Massachusetts. ISBN 9781107007529

- Files for matlab were used from SRD package.