UWB Transmission Simulation

Radio Frequency for Connected Object (PHY7509)

M1 Electrical Engineering for Communication and Information Processing

TELECOM SudParis – IP Paris

Author:

Areeb TARIQ

Rilwanu Ar Roiyyaan KASNO

Supervisor:

Prof. Muriel Muller

Paris Area, France

June 2021

Contents

1. Abstract

This work is segmented in two main parts as stated in problem statement. The first part mainly deliberates the process of Transmitted Reference of Ultra Wide Band (TR-UWB) signal. This baseband signal encompasses multiple Scholtz's Monocycle pulses representing the coded binary sequence of message, which is later modulated with a cosine carrier at a given frequency. The second part is meant for reception simulation in which the received signal is processed and decoded, as well as analysed by several parameter adjustments. At each of every step, the corresponding Matlab code is well commented with relevant information. The results are visually depicted in graphs along with figures in this report.

2. Introduction

UWB signal consists of short pulses which has a wide band and low spectrum density in the frequency spectrum. This lab is a practical demonstration of dealing with UWB signals using MATLAB tool. The pulse width of UWB, Tp is set to 2 ns.

TR-UWB signals consists of doublets. Each doublet has one reference pulse and one information pulse. There is a specific time separation with these two pulses within a single doublet represented by T_D. Moreover, there is a time separation between two doublets represented by T_S.

In the generation process of UWB signals, "special" TR-UWB baseband signal consisting of multiple doublets. In each doublet, the information binary value is coded by an amplitude modulation of information pulse. This is carried out in such a way that the information value of "1" is equivalent to the information pulse which has the same amplitude as the reference pulse and information value of "0" is equivalent to the one third of the reference pulse amplitude. This is demonstarted in figure-1.

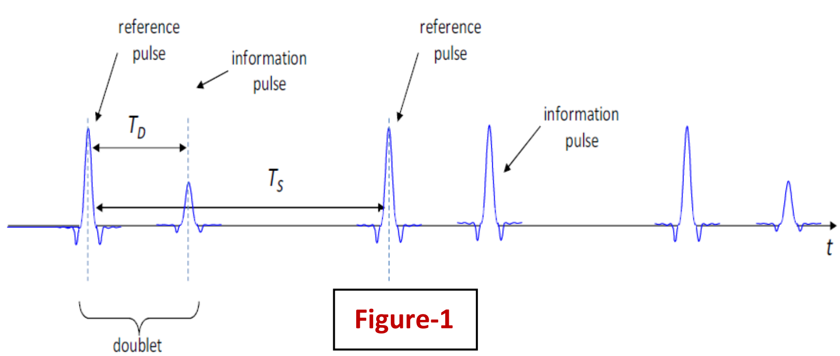

Figure-1

The pulses follows the Scholtz's monocycle with the width of the pulse Tp = 2 ns . The value of T_D is fixed to 50 ns, and T_S is set to 300 ns. The binary information message consists of 7 bits [1 0 1 1 0 1 0]. The baseband signal is modulated with with a cosine carrier at frequency fp = 4.492 GHz.

On the receiver side, the received modulated signal is first passed through low noise amplifier to remove the noise and then baseband filter to obtain the RF received signal.

Following this, the reception simulation is conducted by firstly squaring this filtered received signal. Afterward, the squared signal is integrated with considering parameter variations with the aim of gaining finer result. A considerable threshold is then determined for the decision block to classify or decode the integrated signal into binary informatio sequence.

The above-mentioned variations inlcuding integration parameters and threshold is investigated as a mean to comprehend the effect on the complete process of achieving expected binary signal. In this report, the plots for time and frequency domain for each stage are attached. Moreover, the MATLAB code file is also detailed commented explaining the implmentation methods.

3. Objectives

In this practical lab, there are following objectives:

a. TR-UWB generation based on Scholtz's Monocycle
b. Baseband generation with the specified T_p, T_d, T_s
c. Modulation of baseband signal with carrier frequency F_p
d. Performing signal squaring at the receiver side
e. Integration over the signal with different integration parameters
f. Implementation of decision block to classify into binary signal
g. Generation of time and frequency domain plots at every step.
h. N-point FFT implementation to obtain the frequency spectrum of different signals

4. Methods

In this section, the methods for each objective are briefly explained.

4.1. TR-UWB generation based on scholtz's monocycle

This was generated in MATLAB using the formula mentioned in the course slide:

$$g_3(t) = A\left[1 - 4\pi\left(\frac{t}{T_p}\right)^2\right] e^{-2\pi(t/T_p)^2}$$

The time and and frequency plot for this UWB reference pulse are generated and are shown in next results section.

4.2. Baseband generation with the specified T_p, T_d, T_s

A loop in MATLAB with length of number of binary informations bits was added in MATLAB. The loop first generated the number of reference pulses equal to the number of information bits separated by T_S=300ns. Then, for each information bit, a information pulse is added with each reference pulse at the separation of T_D=50ns. The amplitude of the information pulse is equal to amplitude of refernce oulse if information bit is '1'. In second case, the amplitude of the information pulse is equal to $1/3^{rd}$ of the amplitude of refernce oulse if information bit is '0'.

The time and and frequency plot for this baseband signal are generated and are shown in next results section.

4.3. Modulation of baseband signal with carrier frequency F_p

In MATLAB, "modulate" function is used to modulate the generated baseband signal with the carrier with frequency F_P= 4.492 GHz. This function takes input of baseband signal, carrier frequency, sampling frequency, and type of modulation. We have implmeneted the default "Amplitude Single Sideband" modulation with other given parameters. The output of this function are two arrays of modulated signal and respective time values for this modulated signal.

The time and and frequency plot for this modulated signal are generated and are shown in next results section.

4.4. Performing signal squaring at the receiver side

In this step, the received modulated signal at the receiver side is simply squared using square "^" operation in MATLAB.

The time and and frequency plot for this squared received signal are generated and are shown in next results section.

4.5. Integration over the signal with different integration parameters

The goal here is to integarte the received squared signal for each doublet. It's obvious the integrated values for modulated '1' and modulated '0' are different. This is because the modulated '0' has the baseband signal with information pulse amplitude $1/3^{rd}$ of the amplitude of the information pulse present in modulated '1'.

As total signal time can be <u>approximately</u> estimated using this theoretical expression

$$0.5T_P + 7T_S + 1T_D + 0.5T_P = 1 + 6*300 + 50 + 1 = 1852\text{ns}.$$

However each doublet <u>approximately</u> equal to

$$T_D+T_P = 50+2 = 52\text{ns}$$

Total 7 doublets are approximately equal to the time of 7*52 = 364ns. This means that approximately 1852-364 = 1488ns are dead zones or the regions between doublets. In MATLAB, they are represented by zeros.

Hence, to integrate the doublet region, we made a safe block size of 100ns and then integrated total signal by dividing into blocks of this size. This resulted in discrete values for different blocks.

For integration in the MATLAB, we used the "cumtrapz" function. This performs Cumulative trapezoidal numerical integration using trapezoidal method with unit spacing. This gave definitive and consistent values for the '1' and '0', 45 and 25 respectively. This made the detection smooth in the next stage. The detection accuracy was 100%.

However, we tried to modify the spacing parameter for integration. This resulted in variation of integrated values and therefore inaccurate detection in next stage. We tried increasing and reducing step size of the integration, and observed the results.

CASE-A: When step size was 0.9 (less than 1), we observed the increase in maximum integrated values for both '1' and '0'. For example, for first '1' had value of 5.5 but next '1' had 10.29, and then 14.11 and so on. Similarly for first '0' had value of 4.11 and for next '0' had 10.75, and so on. Here, the noticable thing is that value of 2^{nd} '0' is higher than value of 2^{nd} '1', which is really undesirable and will lead to inaccurate results. The detection accuracy was 43%.

We also observed that reducing further the step size drastically decreases the integrated values, hence makes it more difficult to detect.

CASE-B: When step size was 1.1 (higher than 1), we observed the decrease in maximum integrated values for both '1' and '0'. For example, the first '1' had the value 300, but the fourth '1' had value of 72. Similarly, the first '0' had value of 126, and the second '0' had 53. Here, the noticable thing is that the value of 1^{st} '0' is higher than the value of 4^{th} '1', which is once again highly undesirable and will lead to inaccurate results. The detection accuracy was 57%

We also observed that increasing further the step size rapidly increases the integrated values, hence makes it more difficult to detect.

The time and and frequency plot for this integrated signal and different cases are generated and are shown in next results section.

4.6. Implementation of decision block to classify into binary signal

As we have 7 doublets separated by T_S=300ns, we compared the maximum value of each doublet (integared result obtained in last step) with the threshold. As it's shown in next results section, the maximum integrated value of doublet with information bit = '1' was approximately equal to 45. And for the doublet with information bit ='0', the maximum integrated value of doublet was approximately 25.

Therefore, we decided to make the safe thresold between these two values (45 and 25). In this exercise, we choose the thresold = 30. This safely distinguishes the information bit ‘1’ and ‘0’. If the maximum integarated value is greater than thresold (30), we mark it binary ‘1’ and if it’s less than the threshold, we mark it ‘0’. We achieved 100% correct detection with this safe threshold.

We tried changing the threshold parameter value to 20, and as expected obtained the invalid results. The detection accuracy was 57%.

The decoded binary sequence after this decision block is shown in next results section.

4.7. Generation of time and frequency domain plots at every step.

For each step, we plotted time and frequency results. These results are displayed in next section.

4.8. N-point FFT implementation to obtain the frequency spectrum of different signals

We implemented N-point FFT in MATLAB for the generation of the frequency spectrum of signals at different stages of this procedure. To calculate N, we rounded off the number of time domain signal points to the next multiple of 2, using “ceil” function. Then, we used the “fft” function in MATLAB to compute the frequency spectrum. This function takes time domain signal and number of FFT points (N) as inputs and outputs the frequency spectrum. Since FFT is symmetric and for our analysis we can use only side, we removed the repeated points and used one side for the results.

The frequency plots for all stages are generated and are shown in next results section.

Results

The time and frequency plot of TR-UWB reference signal are given below in figure-2.

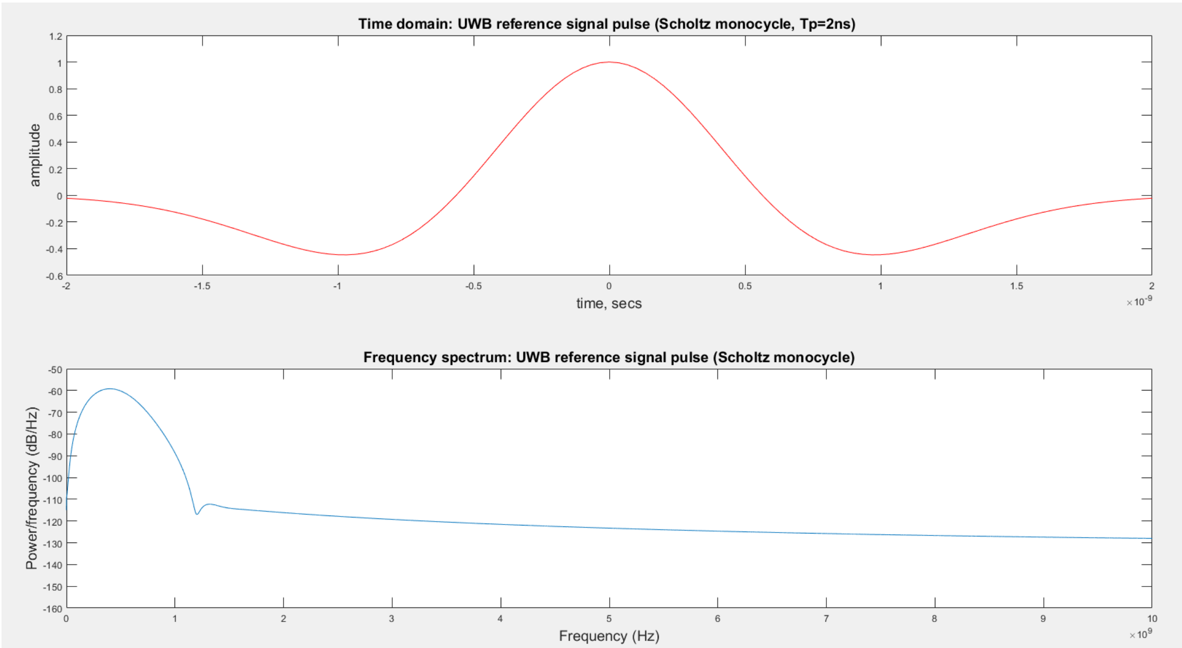

Figure-2: Time and Frequency plot of UWB reference pulse

The time and frequency plot of binary coded baseband signal are shown in below figure-3.

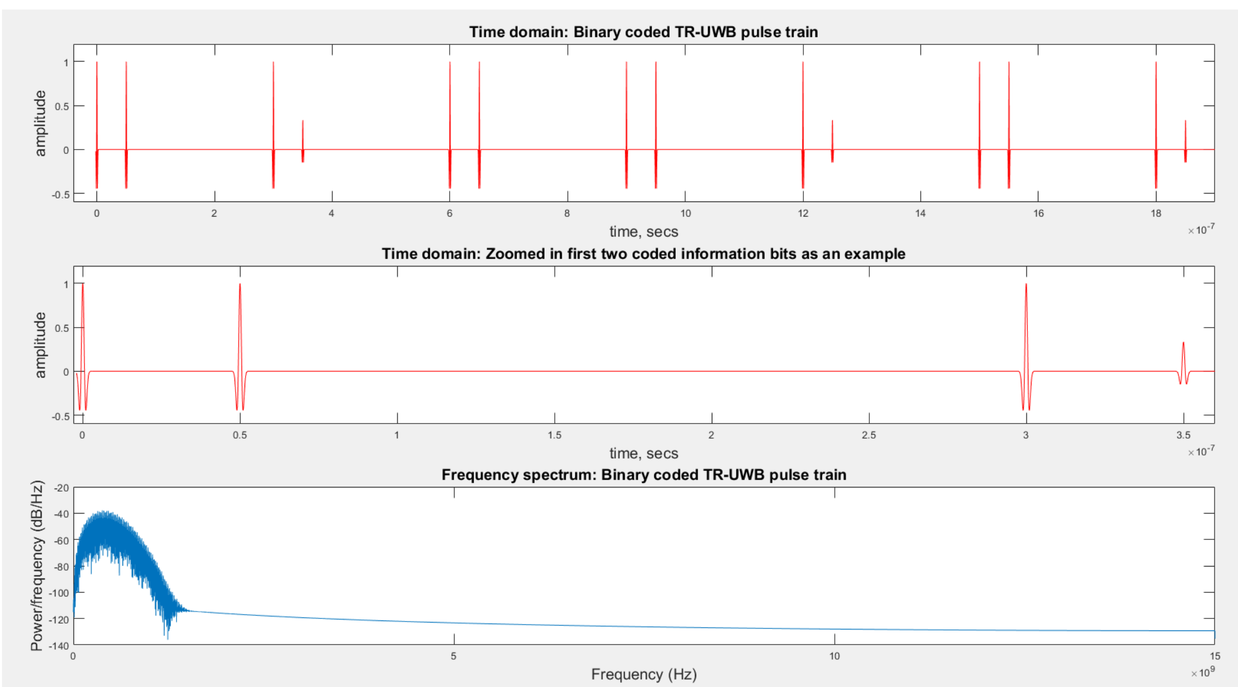

Figure-3: Time and Frequency plot of binary coded baseband signal. The middle plot is the zoomed version of the time plot to better observe the T_D and T_S.

The time and frequency plot of modulated UWB signal are shown in below figure-4.

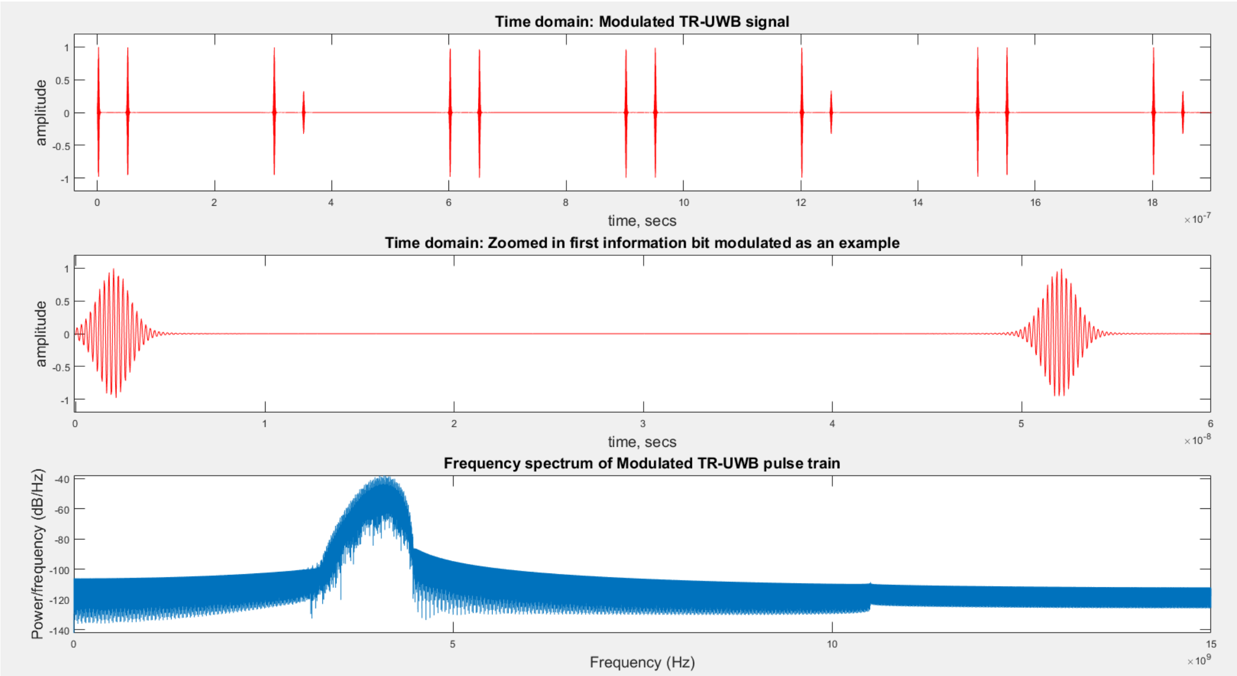

Figure-4: Time and Frequency plot of modulated UWB signal. The middle plot is the zoomed version of the time plot to better observe modulation.

The time and frequency plot of squared received UWB signal are shown in below figure-5.

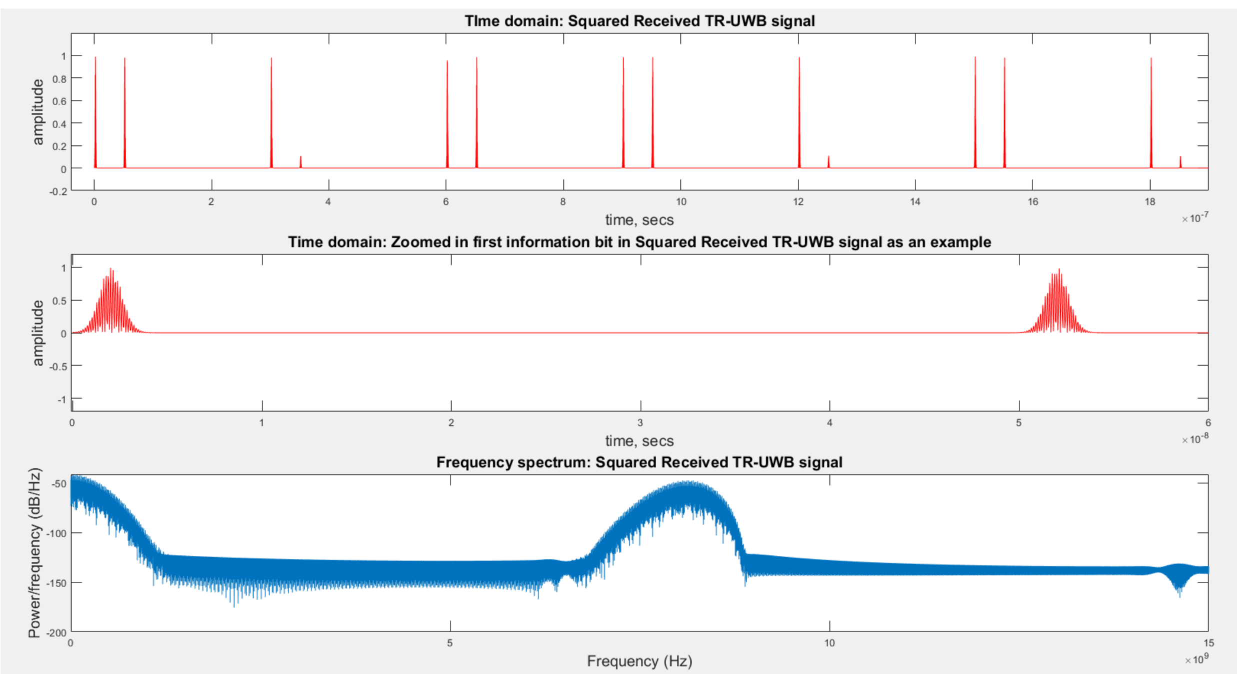

Figure-5: Time and Frequency plot of squared received signal. The middle plot is the zoomed version of the time plot to better observe the square operation.

The time plot of integrated received UWB signal and the decoded binary sequence are shown in below figure-6. **Conditions**: Integration Spacing = 1 (ideal), threshold = 30 (ideal), **detection accuracy = 100%** (ideal). This is our proposed solution and it has perfect results.

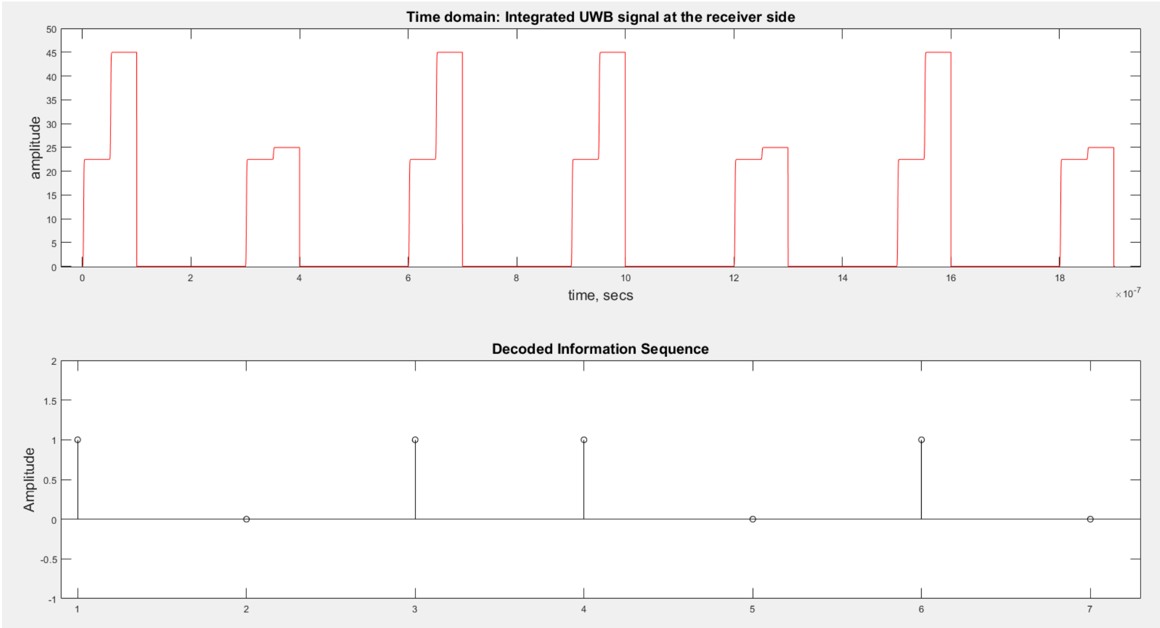

Figure-6: Time plot of integrated signal. The bottom is the decoded binary sequence.

Integration Spacing = 1, threshold = 30, detection accuracy = 100%

Now we are going to experiment with integration and threshold parameters and see results get affected.

The time plot of integrated received UWB signal and the decoded binary sequence are shown in below figure-7. **Conditions**: Integration Spacing = 0.9 (10% reduced), threshold = 30 (ideal), detection accuracy = 43%. It detects everything in this case as '0'

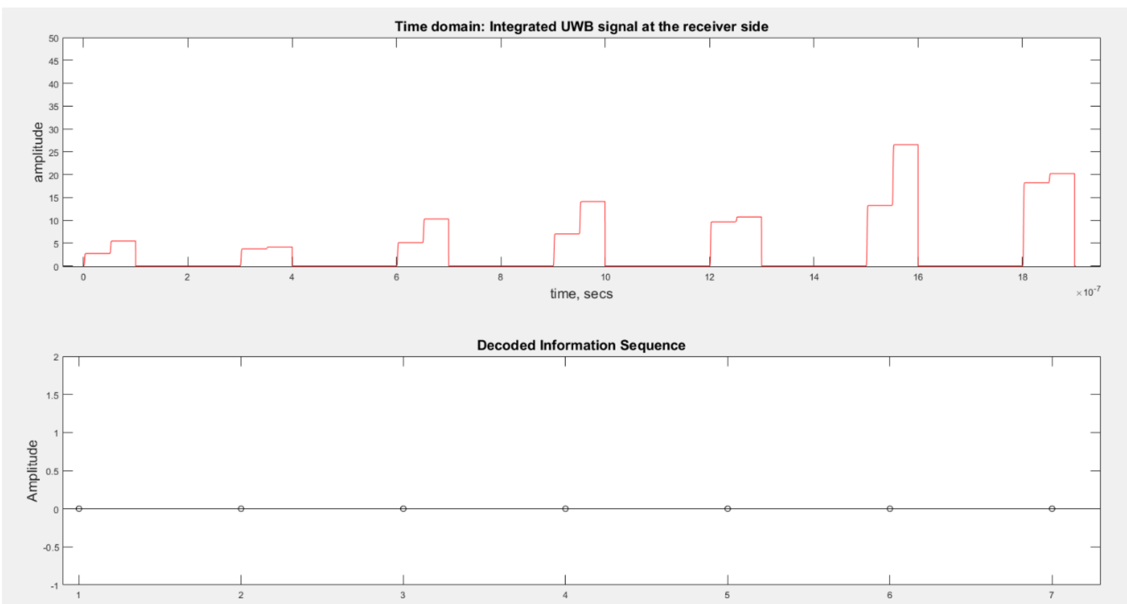

Figure-7: Time plot of integrated signal. The bottom is the decoded binary sequence.

Integration Spacing = 0.9, threshold = 30, detection accuracy = 43%

The time plot of integrated received UWB signal and the decoded binary sequence are shown in below figure-8. **Conditions**: Integration Spacing = 1.1 (10% increases), threshold = 30 (ideal), detection accuracy = 57%. It detects everything in this case as '1'

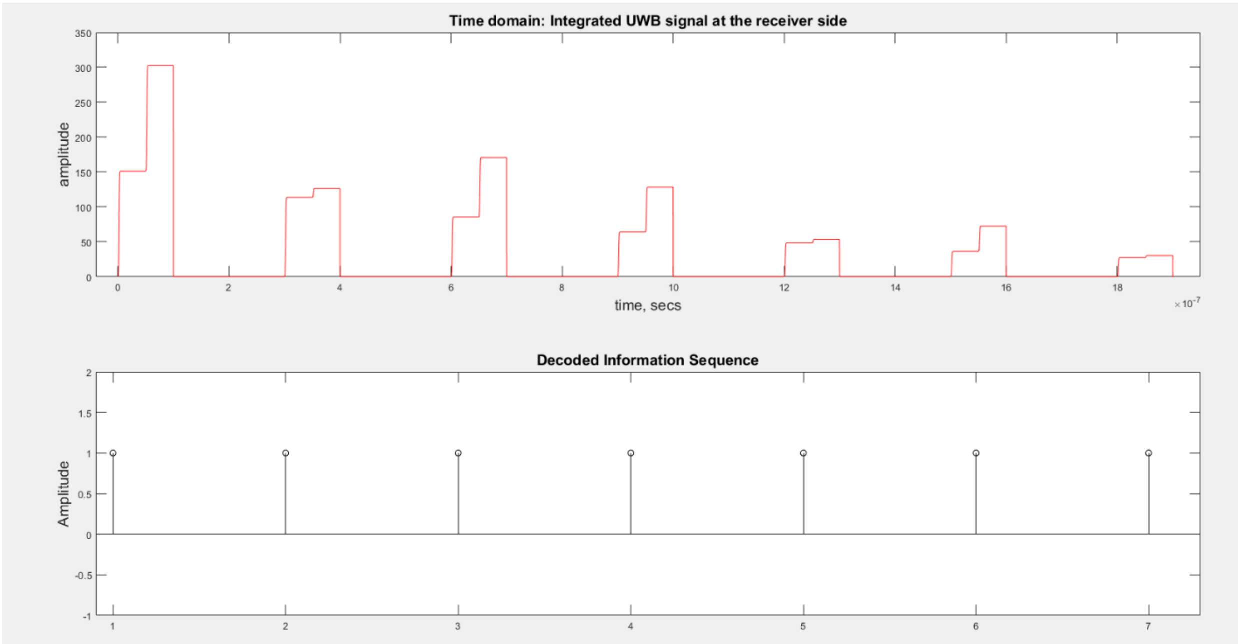

Figure-8: Time plot of integrated signal. The bottom is the decoded binary sequence. Integration Spacing = 1.1, threshold = 30, detection accuracy = 57%

The time plot of integrated received UWB signal and the decoded binary sequence are shown in below figure-9. **Conditions**: Integration Spacing = 1 (ideal), threshold = 20 (33% reduced), detection accuracy = 57%. It detects everything in this case as '1'

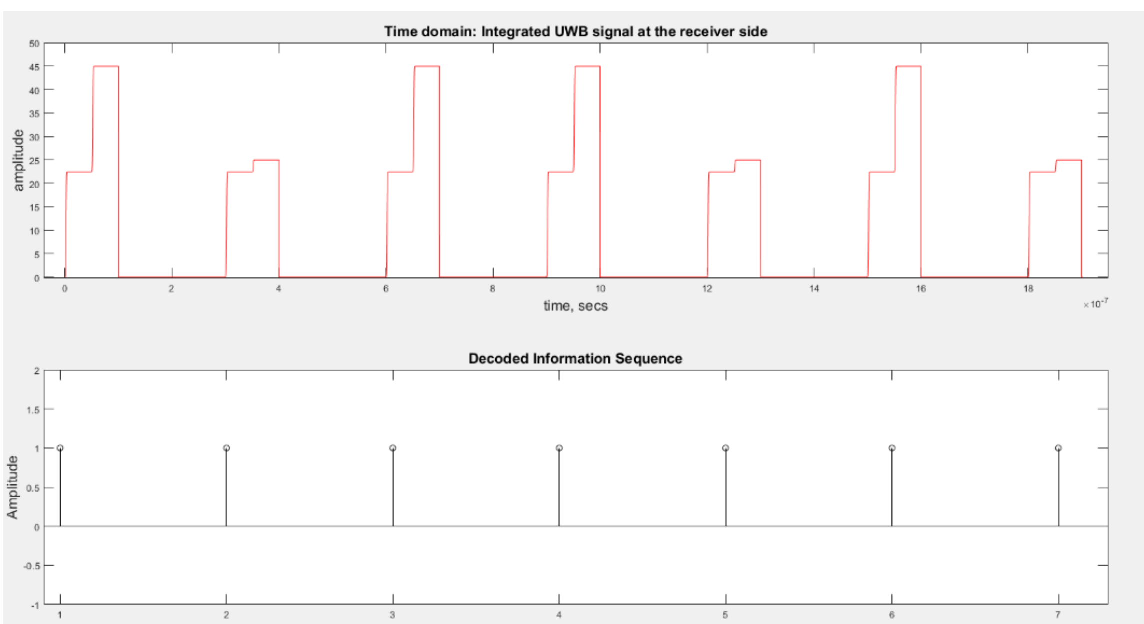

Figure-9: Time plot of integrated signal. The bottom is the decoded binary sequence. Integration Spacing = 1, threshold = 20, detection accuracy = 57%

Conclusion

To conclude our work, using MATLAB, we have managed to generate the designated Scholtz's Monocycle as reference pulses for the baseband signal with the indicated T_p. Then we generated the baseband signal using information binary sequence and specified T_d, and T_s. In next step, we carried out the modulation of the baseband signal with the specified carrier signal frequency. On the receiver side, we squared the received signal so that it can be integrated in next stage. In integration stage, we use Cumulative trapezoidal numerical integration in MATLAB. We also tried to modify the integration spacing parameters and observed the results. And finally, we implemented the decision block based on threshold. We also tried to modify the decision threshold and observed the results. For all the stages and scenarios mentioned, we generated the time and frequency plots and have shared in the results section of this report. The MATLAB code has been commented in detail and has a ReadMe file for usage.