Master International E3A

PRACTICAL REPORT

FOR

Ultra Wide Band (UWB) Signal Praticals with MATLAB

Radio Frequency for Connected Object (Phy4509)

BY:

ABODE DANIEL

SUBMITTED TO:

Prof. Muriel Muller

DATE: 09/06/2020

Abstract

This practical is divided into two parts. In the first part, we considered the modelling of a TR-UWB signal generator, which generates a modulated signal of pulses coding a binary data. The second part involves modelling the receiver that receives the signal generated in the first part, extracts, and display the information. The time plots and frequency response of the signals at every stages were plotted. The practical was carried out using MATLAB integrated development environment.

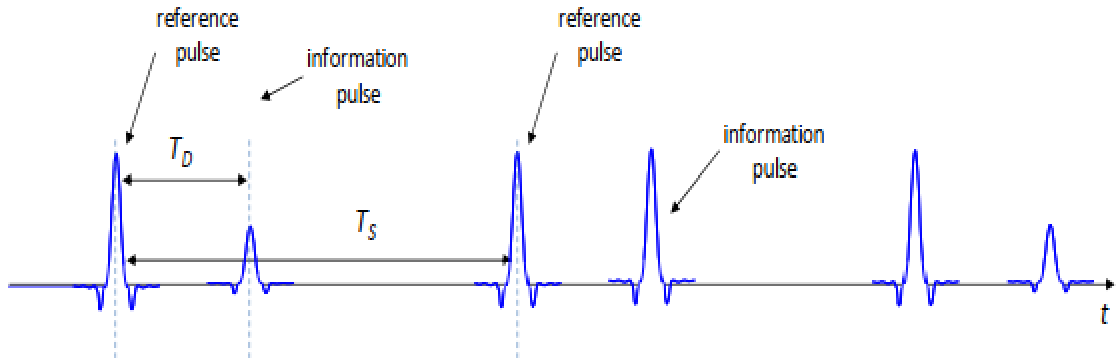
Table of Contents

First Part: Generation and Modulation of TR-UWB Signal

1.1 Introduction

This practical involved the generation and modulation of TR-UWB signal using MATLAB. UWB signal is a class of signal has a wide band in the frequency spectrum as oppose to narrow band signal, pulses with narrow time width generate it. In this practical, a MATLAB program was developed to generate a baseband TR-UWB signal with which we coded some binary data with the amplitude of the signal. For a "1" the pulse has the same amplitude as the reference pulse, and for a "0", the pulse has an amplitude that is one third of the reference pulse.

The pulses follows the Scholtz's monocycle with the width of the pulse Tp = 2 ns; the time delay between the reference pulse and the information pulse TD = 50 ns; and the time delay between the reference pulse TS = 300 ns as shown in figure below;



For the practical work, we generated a signal with the binary sequence [1 0 1 1 0 1 0]. We proceeded to modulate the baseband signal with a cosine carrier at frequency fp = 4.492GHz.

1.2 Objective

1. To learn how to generate a special TR-UWB baseband signal base on Scholtz's Monocycle using MATLAB program.
2. To modulate the baseband signal.
3. To plot and appreciate the time signal and frequency spectrum of the signal before and after modulation.

1.3 Practical Procedure

1.3.1 Generating the TR-UWB baseband

a) The variables were declared including TD, TS, tp

b) A single pulse was produced by implementing the second derivative of the formula shown below. The time signal of the pulse is shown in figure 1.1.

$$g_1(t) = A\left[\frac{t}{T_p}\right]e^{-2\pi(t/Tp)^2}$$

c) The binary sequence was coded with the signal and the reference pulse delayed as given by the specification using for-loops.

d) The graphs of the time signal and the corresponding frequency spectrum was made to scale as shown in figures 1.2 to 1.5.

1.3.2 Modulating the baseband signal

a) The baseband signal was modulated by with a carrier at frequency fp = 4.492 GHz by using the modulate() function.

b) The time signal of the modulated signal and its corresponding frequency spectrum was plot as shown in figure 1.6 to 1.8.

1.4 Results and Discussion

1. The sample of the pulse is as shown in figure 1.1. From the figure, it can be seen that the width of the pulse is $2x10^{-9}$ seconds i.e 2ns as required by the specification.
2. The pulse signal coded with the binary sequence is as shown below in figure 1.2, it can be seen that the width of the pulse is quite small compare with length of the signal, this is the reason the signals only appear as spikes. A zoomed view for the 1st binary data which is a One is shown in figure 1.3. It can be seen from the plot that the delay between the reference pulse and the information pulse is 50ns, which is in accordance with specification. It can also be seen that the information pulse which is a 1 has same amplitude with the reference signal. A zoomed view for the 2nd binary data which is a 0 is shown in figure 1.4. It can be seen from the plot that the delay between the reference pulse and the information pulse is 50ns, which is in accordance with specification. It

can also be seen that the information pulse which is a 0 has 1/3 the amplitude of the reference signal.

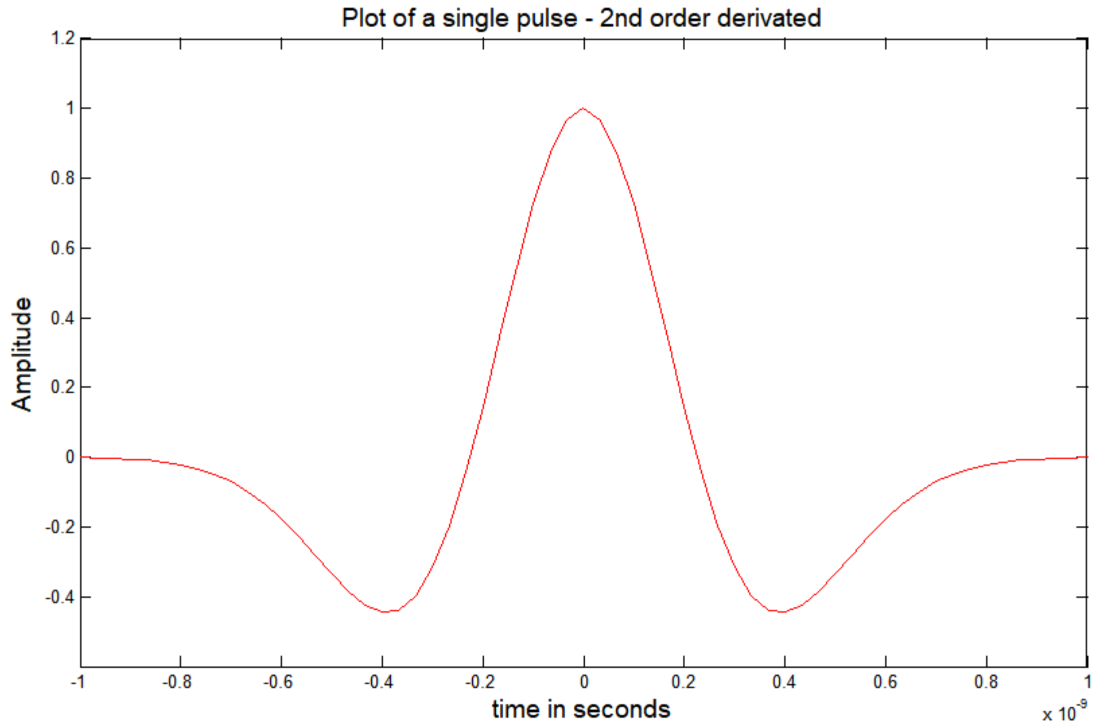


Figure 1.1 Plot of a single pulse generated

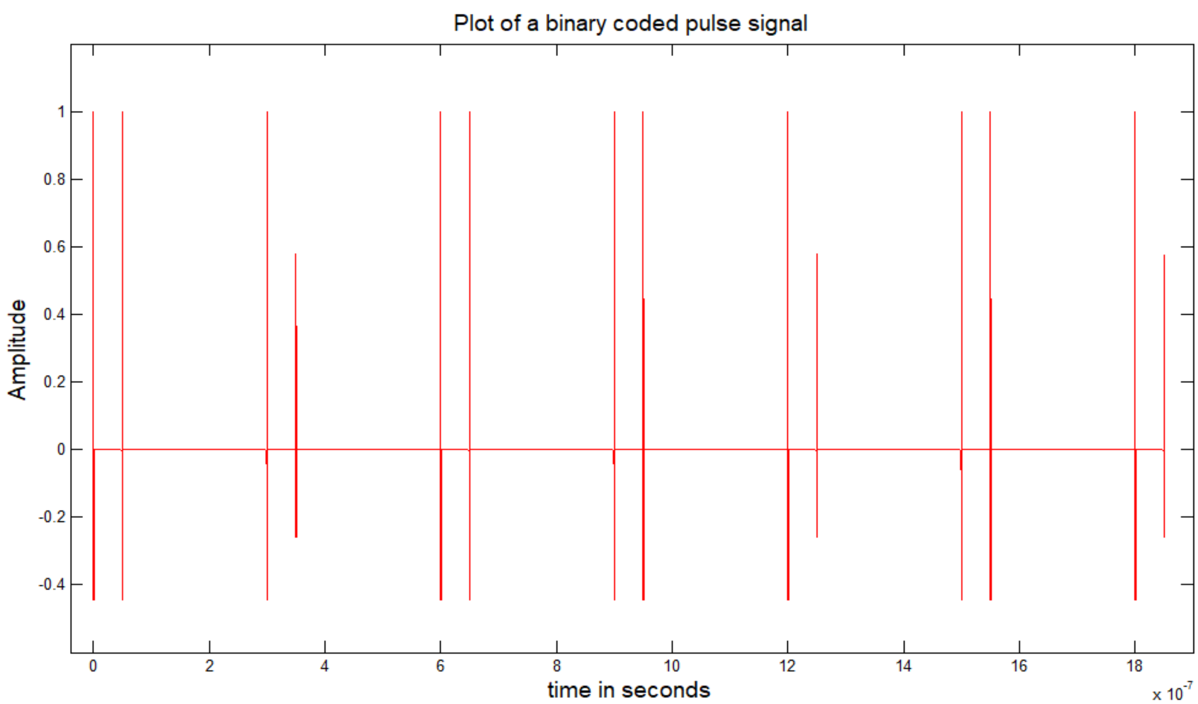


Figure 1.2 Plot of the baseband signal

3. Figure 1.5 shows the frequency spectrum of the signal, and it can be seen that the maximum power is less than -40dB/Hz.

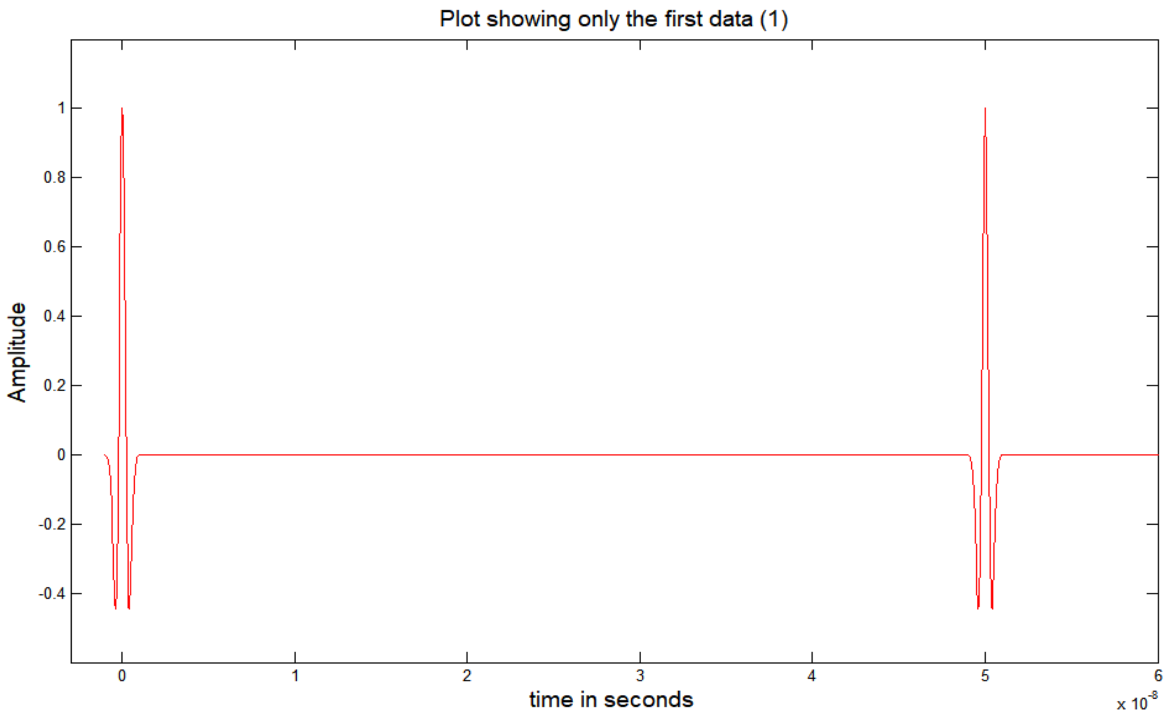


Figure 1.3 Plot of a part of the baseband signal showing 1

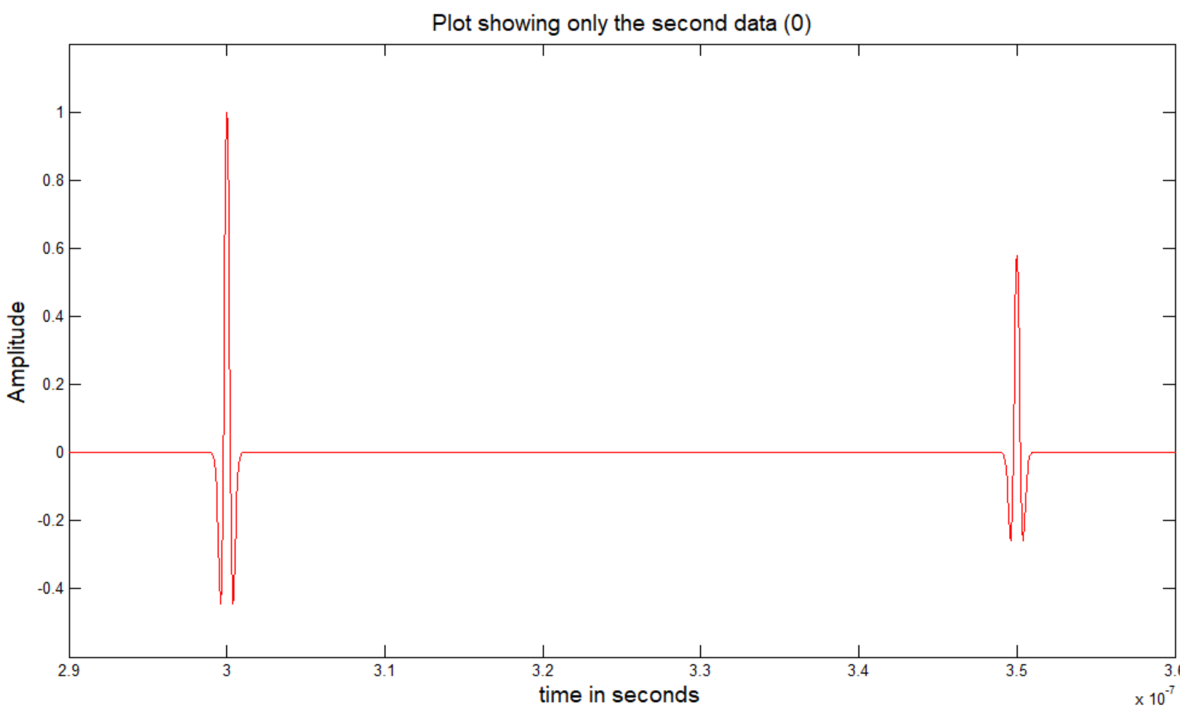


Figure 1.4 Plot of a part of the baseband signal showing 0

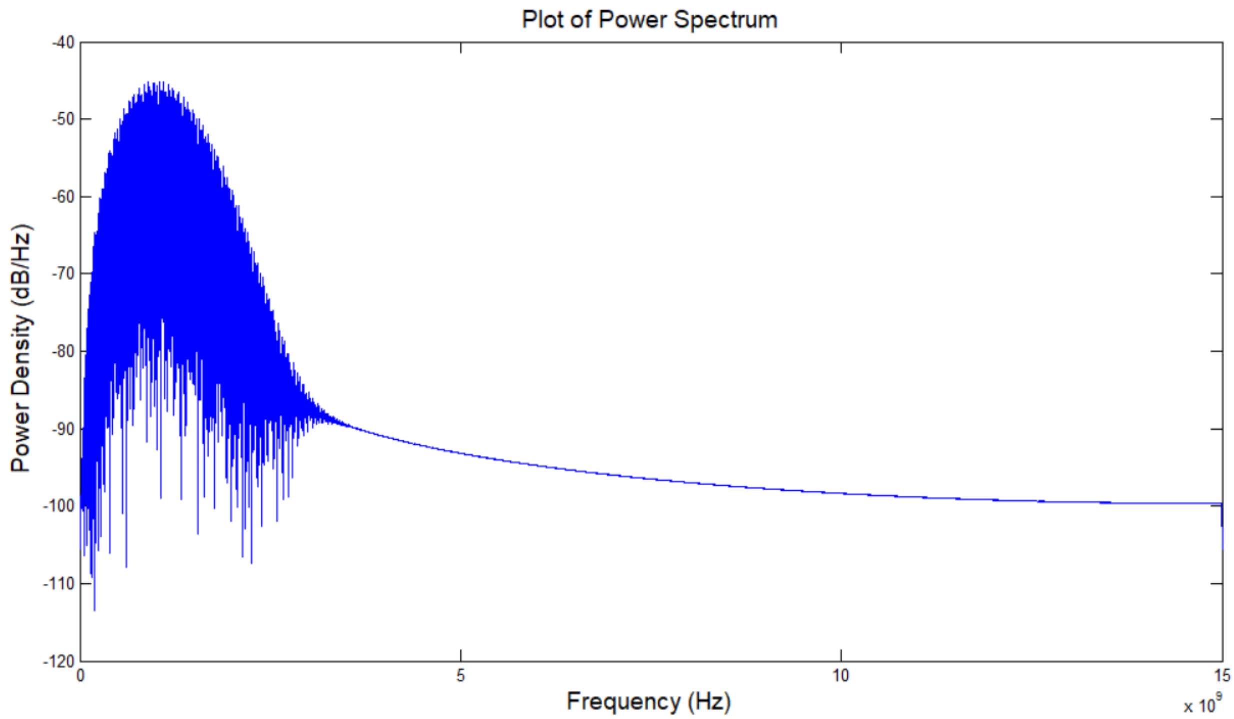


Figure 1.5 Plot of frequency spectrum of a the baseband signal

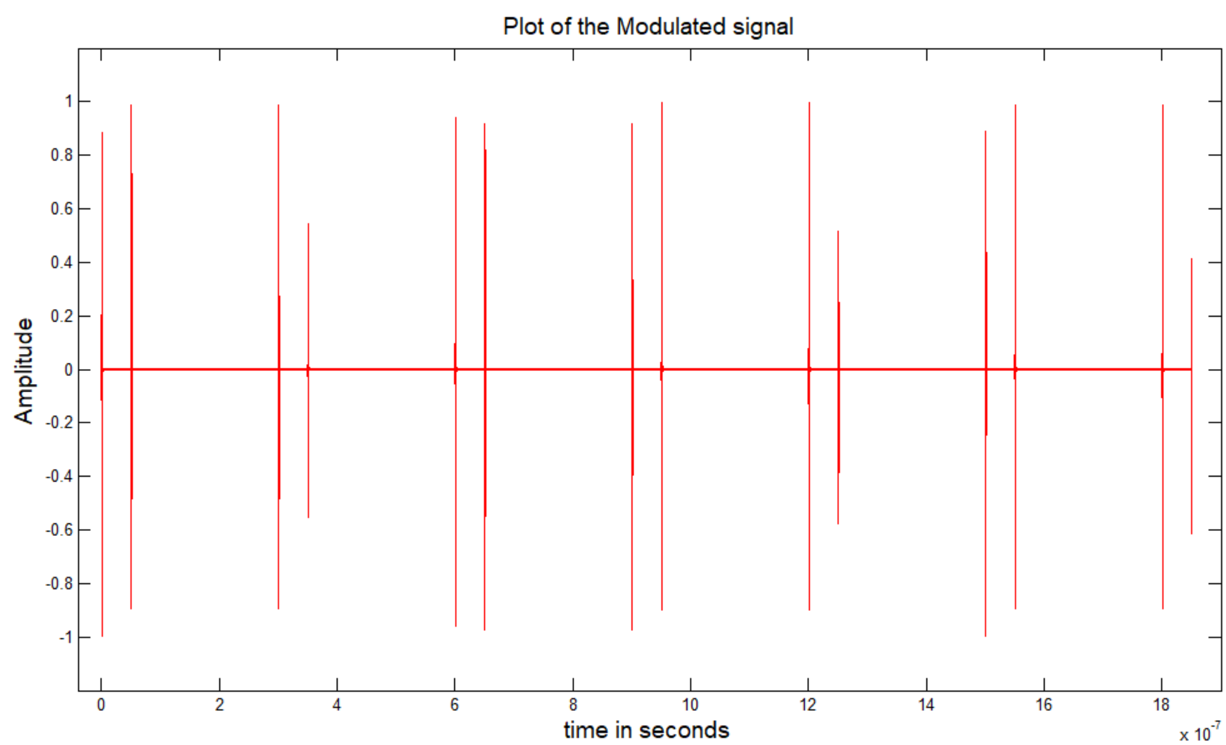


Figure 1.6 Plot of the Modulated signal

4. Figure 1.6 shows the time plot of the modulated signal and figure 1.7 shows just a part of the signal for clearer view of the cosines form of the signal. The power spectrum is

shown in figure 8. It can be seen as well, that the maximum power spectral density is less -40dB/Hz.

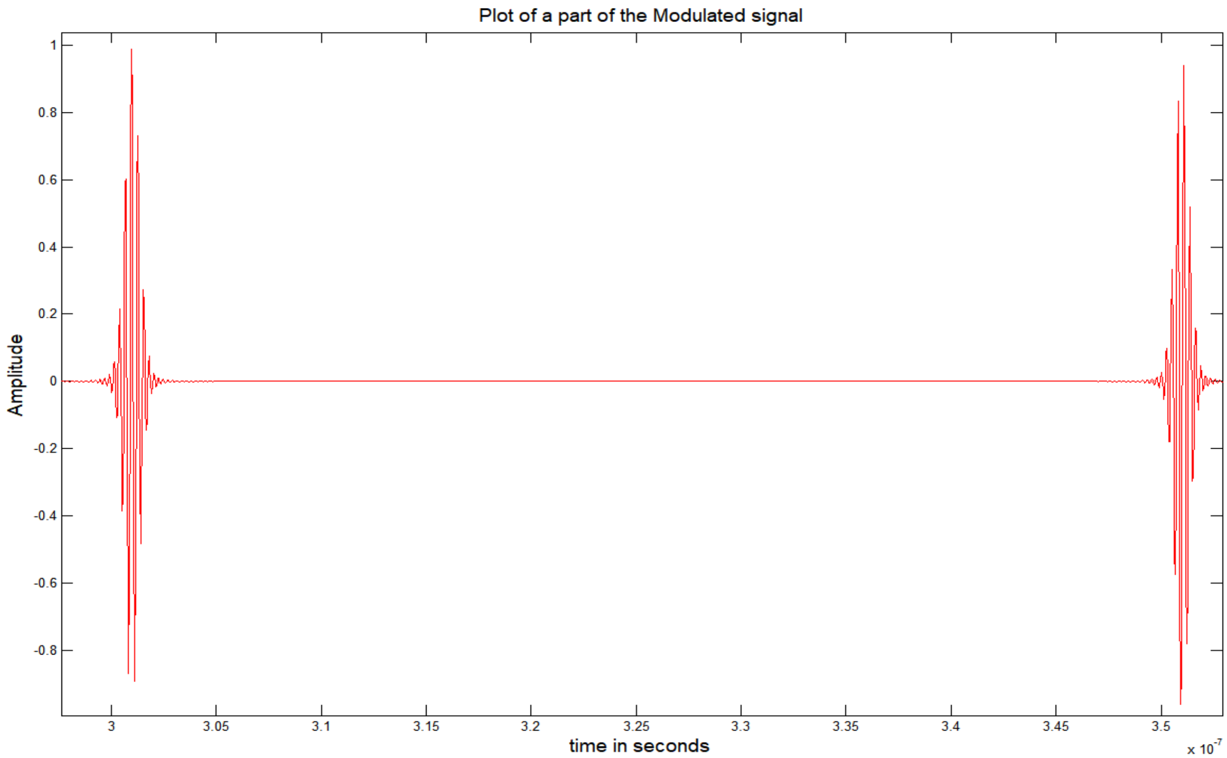


Figure 1.7 Plot of a part of the Modulated signal

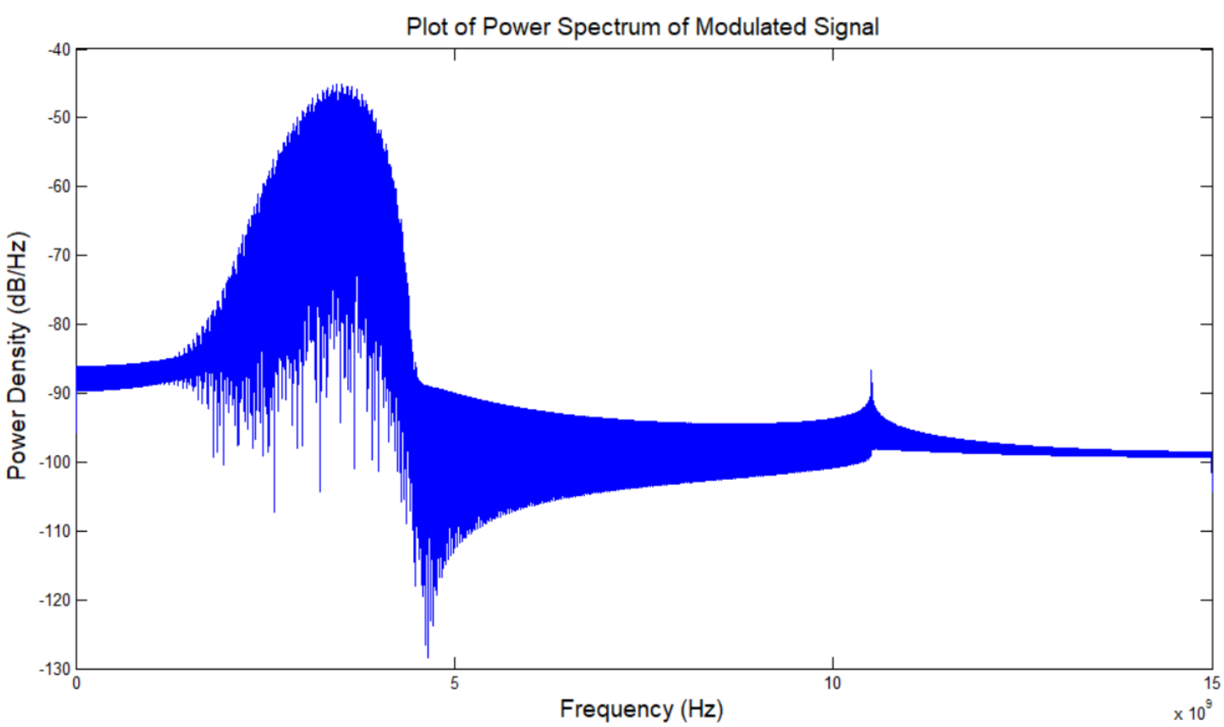


Figure 1.8 Plot of frequency spectrum of a the Modulated

1.5 Conclusion

Using MATLAB code, we were able to develop a signal generator for a special TR-UWB signal, we encoded a binary signal using the signal and we modulated it with cosine signal. This forms a complete UWB signal generated. Using the plot tools of MATLAB, we were able to visualize the signal generated, the modulated signal and their frequency spectrum.

Second Part: Reception and Detection of a TR-UWB signal

2.1 Introduction

At the receiver, the signal needs to be received by an antenna, passed through a Low Noise Amplifier, and then a pass band signal, after which it will be squared, integrated and pass through a decision level block to determine whether it is a 1 or a 0 as shown in figure 2.1.

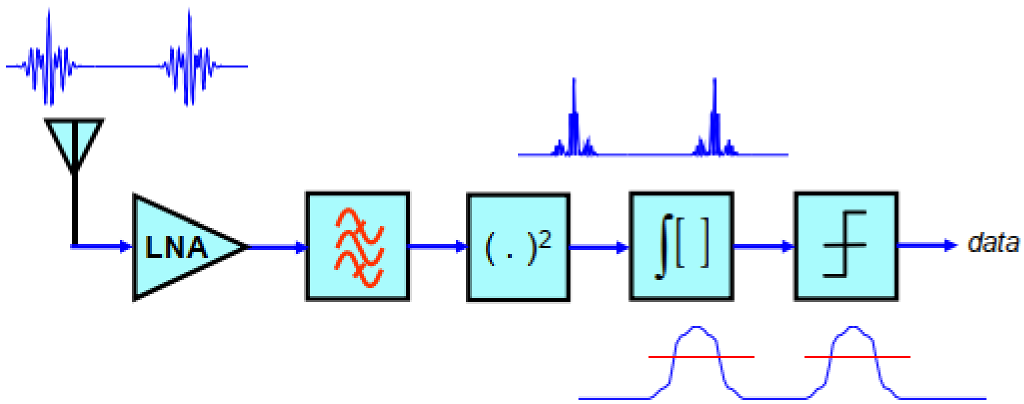


Figure 2.1 Block Diagram of TR-UWB signal Receiver

Using MATLAB, we develop a program, to square the received signal, an integrator that integrates the signal then pass it to a decision level block developed and set with the right threshold to determine if the received signal is a 1 or a 0.

2.2 Objectives

1. To develop a MATLAB program to model a TR-UWB receiver to receive the signal and extract the binary code.

2. To plot the signal in time and frequency domain at different stages of the receiver model.

2.3 Practical Procedure

1) Square the final received signal

a) The modulated signal from the first part was squared.

b) The time signal and corresponding frequency spectrum was produced using plot().

2) Passing through an Integrator

a) The signal was integrated using the cumtrapz() function of matlab, it was passed in every t = 100ns, different time t was tried but the best performance was for t = 100ns.

b) The graph of the time signal of the integrated signal was produced using plot().

3) Designing the decision level block

a) The threshold was chosen as 14 base on the result of the integrated signal, it was chosen so that any value above it is considered as 1 and any value below it is considered as zero.

b) Using a while loop, we pass the integrated signal by batch of 100ns and with a delay of 200 ns, and took the max of the batch, if the max of the batch is greater than threshold we determine that the represented pulse is 1 and if it is lesser than the threshold we determine it to be 0.

c) The result was outputted and a time plot of it was made.

2.4 Results and Discussion

1. The plot of the squared received signal is shown below in figure 2.2, figure 2.3 shows only a part of the signal for clearer view.

2. The frequency spectrum after squaring is shown in figure 2.4. The maximum Power density is also less than -40dB/Hz.

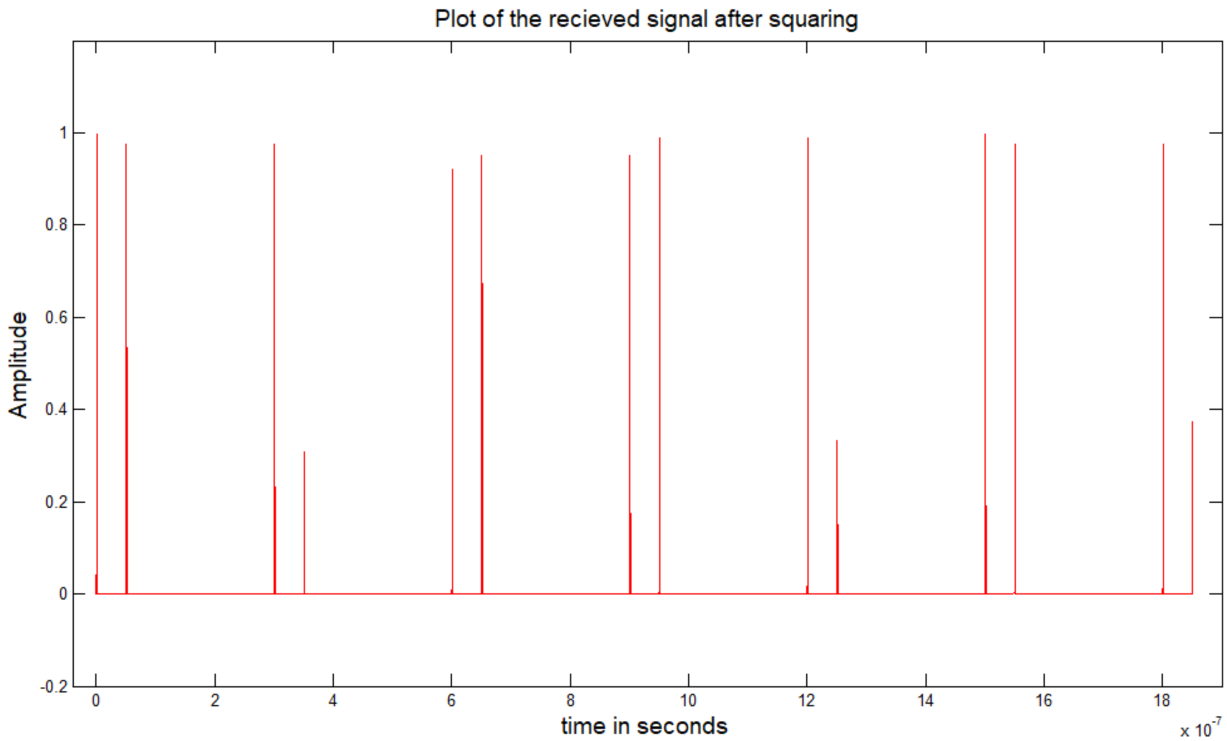


Figure 2.2 Plot of the received signal after squaring

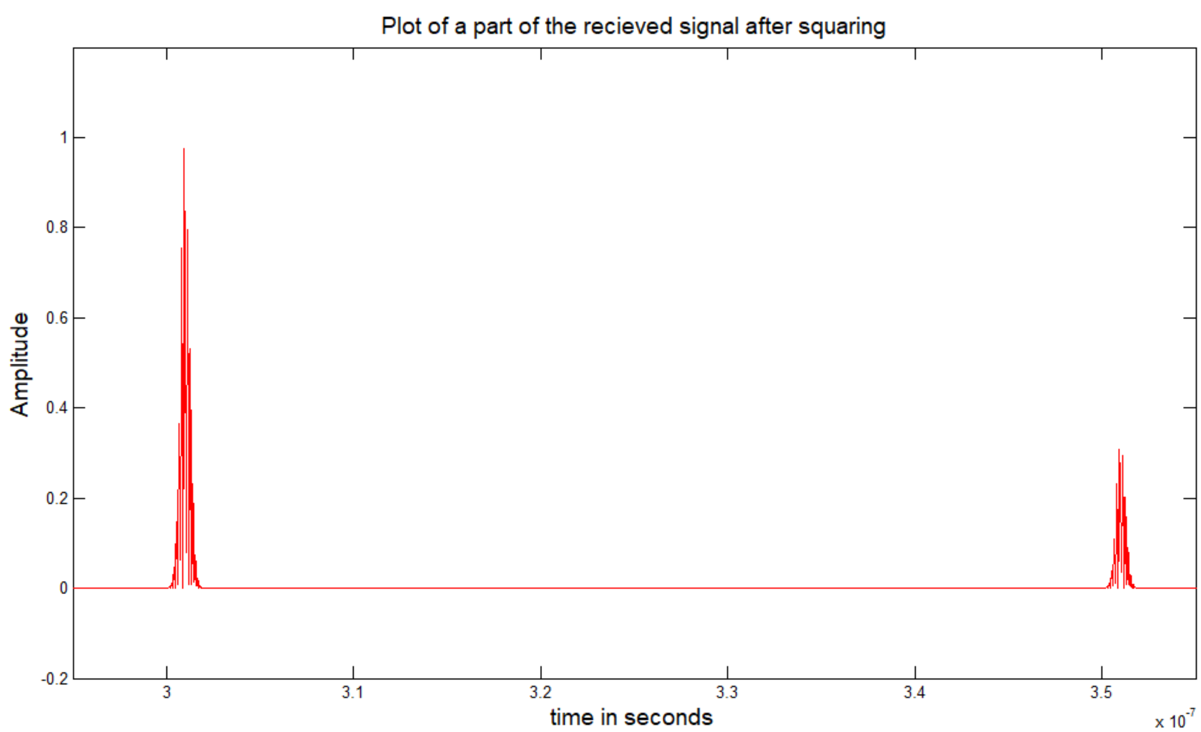


Figure 2.3 Plot of a part of the received signal after squaring

4. The plot of the integrated signal after the first integration that was done in batch of every 100ns is shown below in figure 2.5;

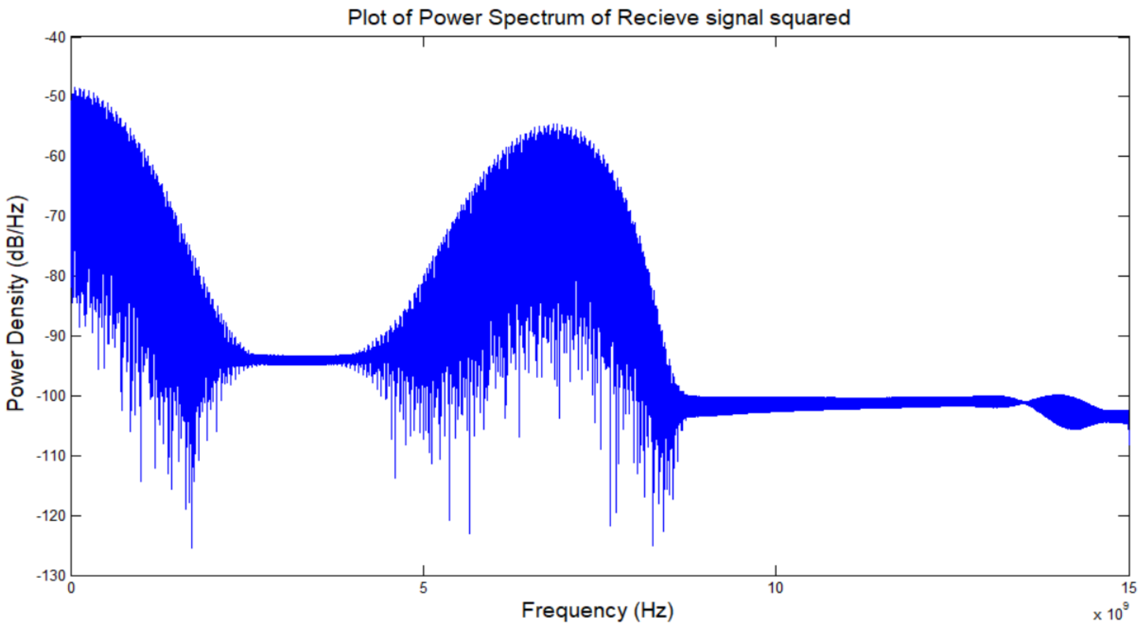


Figure 2.4 Plot of the frequency response of the received signal after squaring

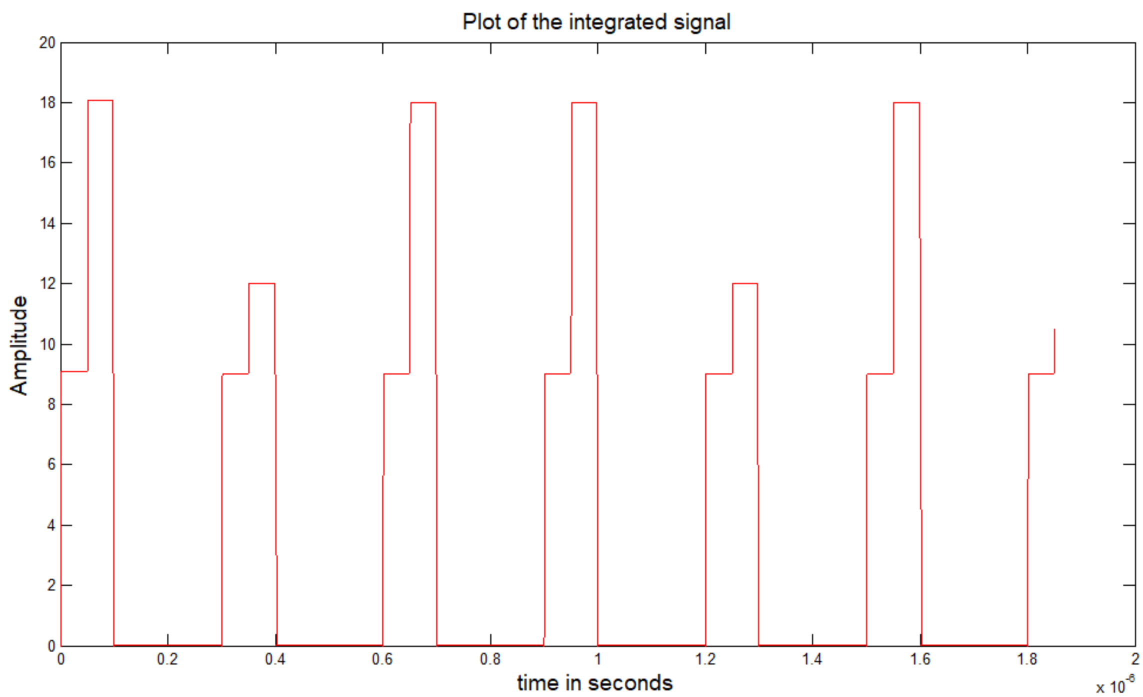


Figure 2.5 Plot of the integrated signal

5. We chose the threshold to be 14 as illustrated on figure 2.6. Choosing a lesser threshold like 8 will cause all result to be 1s and choosing a threshold of 20 will cause all result to be 0s, which is wrong, this means that determining the value of the threshold is a critical components of the decision level block.

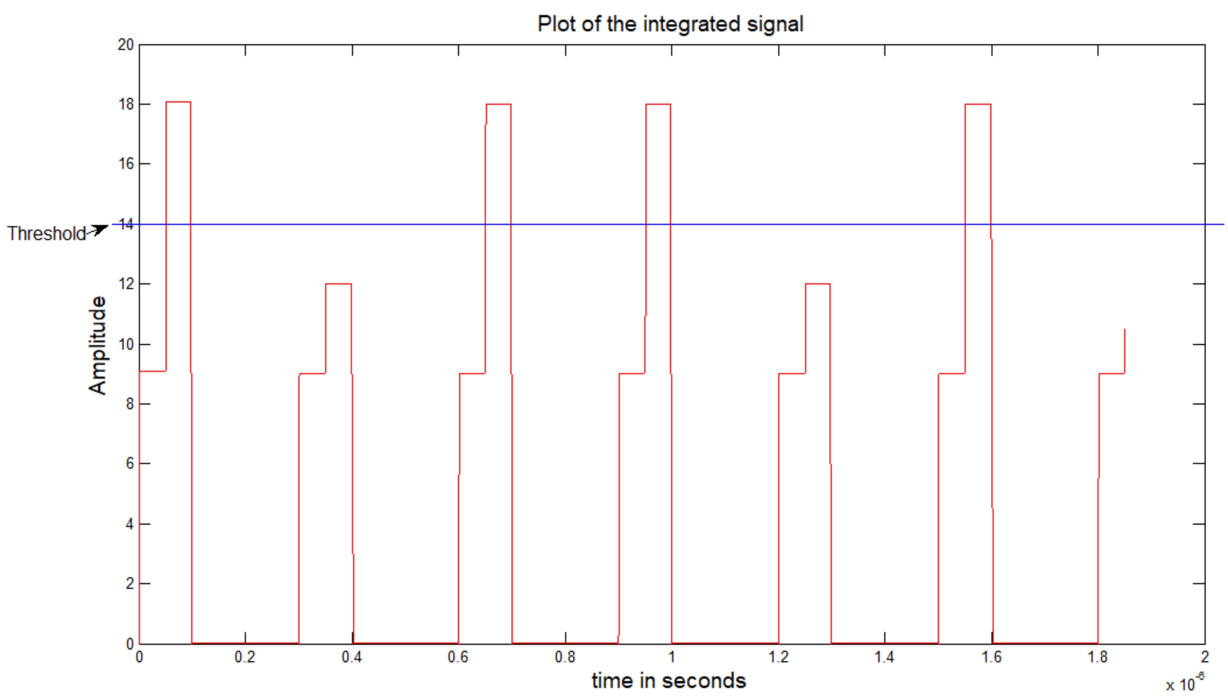


Figure 2.6 Plot of the integrated signal showing the threshold chosen

6. The recovered data was displayed as shown below and plotted in figure 2.7

```
recovered_data :
     1     0     1     1     0     1     0
```

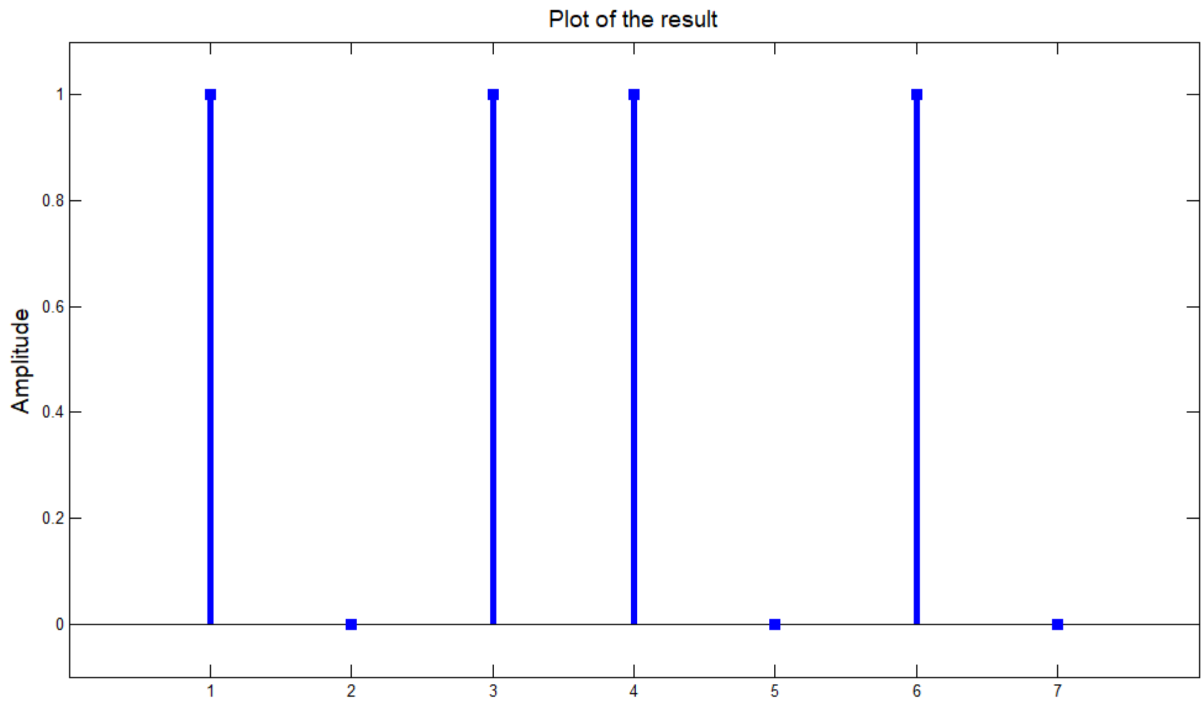


Figure 2.7 Plot of the decoded output

2.5 Conclusion

In the second part of the practical report, we modelled a receiver to decode the signal from the first part of the practical to realize our message (binary code) from the signal. The signal was first squared, after which an integrator was modelled to integrate the signal in real time every 100ns, the value of the integrator matters, 100ns was chosen after it was experimented to give the best result that distinctively show difference in amplitudes between the 1 and the 0 information. The result from the integrator was passed to a decision level block, the performance depends highly on the value of the threshold. We chose 14 as our threshold after viewing the time plot of the integrated signal. 14 was chosen because it was more obvious that values greater than 14 shows a 1 and value below 14 shows a 0. Choosing a wrong value like 8 will output all 1s and choosing 18 will output all 0. Hence, the choice of the threshold value is a major factor in designing a decision level circuit.