Project Report

Measuring the channel frequency and time response through simple channel sounding using software defined radio.

ECE-GY-6023 : Wireless Communications

Prof. Sundeep Rangan

Ankit Singh (as14671)

Table of Contents

Background

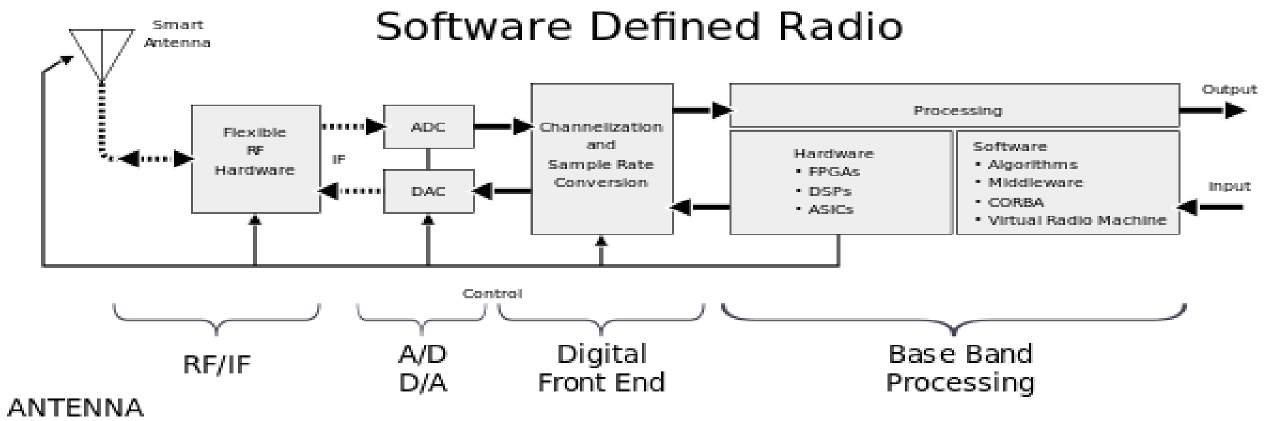
Introduction

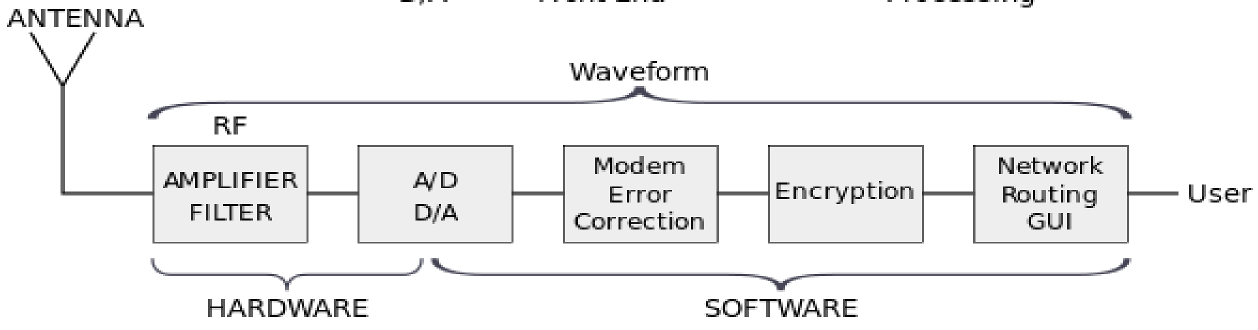
Channel sounding is a technique that evaluates the radio environment for wireless communication, especially MIMO systems. Because of the effect of terrain and obstacles, wireless signals propagate in multiple paths (the *multipath* effect). To minimize or use the multipath effect, engineers use channel sounding to process the multidimensional spatial-temporal signal and estimate channel characteristics. This helps simulate and design wireless systems.

Software Defined Radios

Software-defined radio (SDR) is a radio communication system where components that have been traditionally implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system.While the concept of SDR is not new, the rapidly evolving capabilities of digital electronics render practical many processes which were once only theoretically possible.

A simple block diagram given below explains the main components and the flow of the SDR.





Block diagram of SDR , Source : Wikipedia

Experiment Design and Methodology

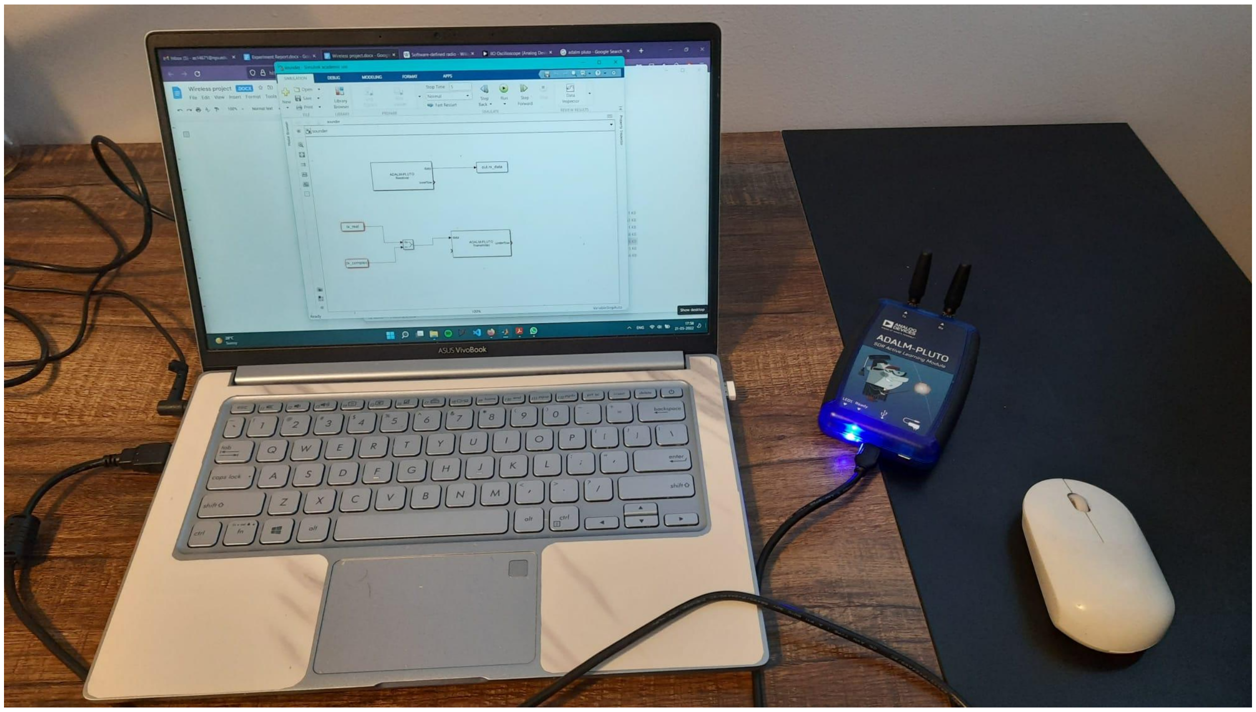
I used a software defined radio called ADALM-PLUTO which is a radio learning module from Analog Devices Inc.. Based on the AD9363, it offers one receive channel and one transmit channel which can be operated in full duplex, capable of generating or measuring RF analog signals from 325 to 3800 MHz, at up to 61.44 Mega Samples per Second (MSPS) with a 20 MHz bandwidth. The frequency range can be extended upto 6 GHz.

PLUTO SDR, Source : Analog Device Inc.

The SDR device was used with Matlab and Simulink to generate the signals, to send and receive the signals, and for pre and post processing . The device uses a toolbox especially provided to communicate ADALM PLUTO through MatLab and Simulink.

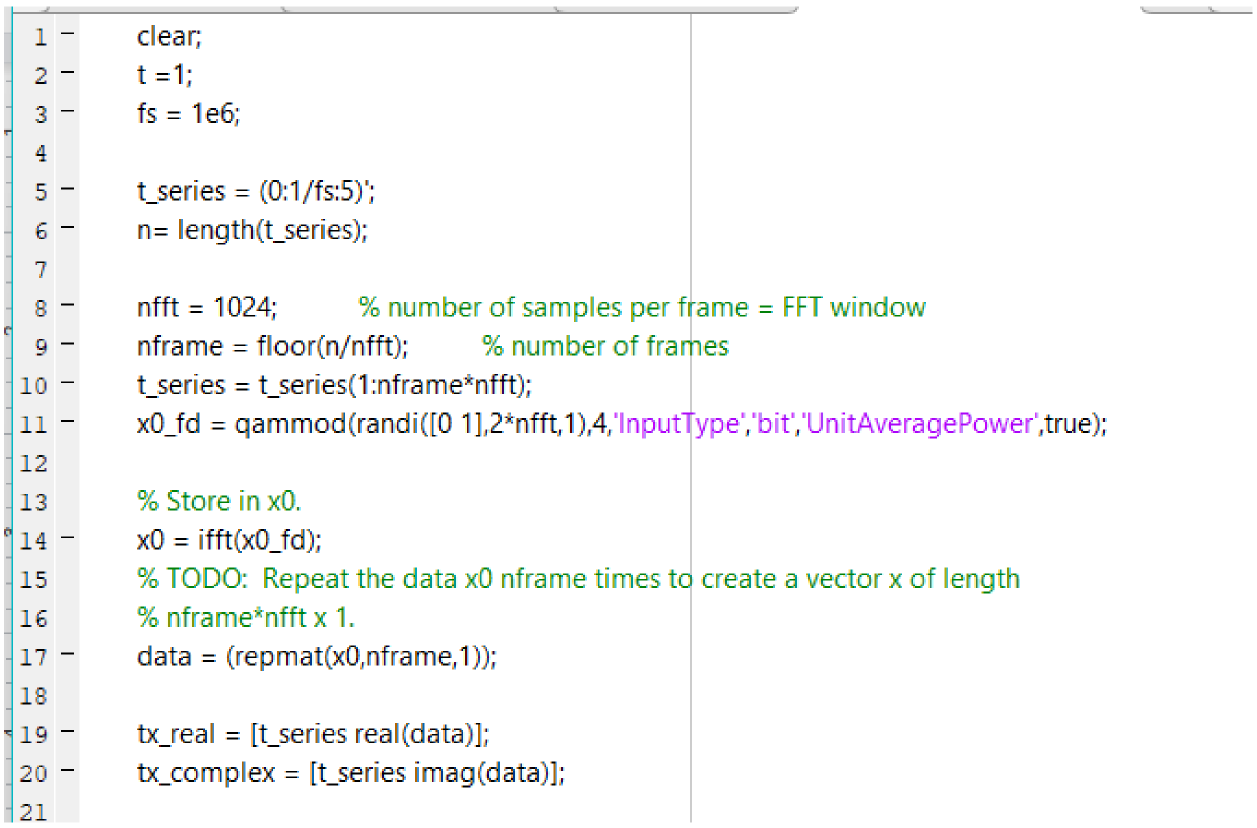
Setup

The setup for the receiver and sender modules were simply easy. The receiver and the transmitter were attached to the same PC with maximum distance between them. Attaching to the same PC allowed easy running of the scripts and avoided getting irrelevant data if the receiver kept listening even if the sender stopped sending .



Setup

The screenshot below shows the script to generate the data to be sent via the transmitter. The data is a simple QPSK modulated random bits of a fixed length. The transmitter sends the same data over and over again.

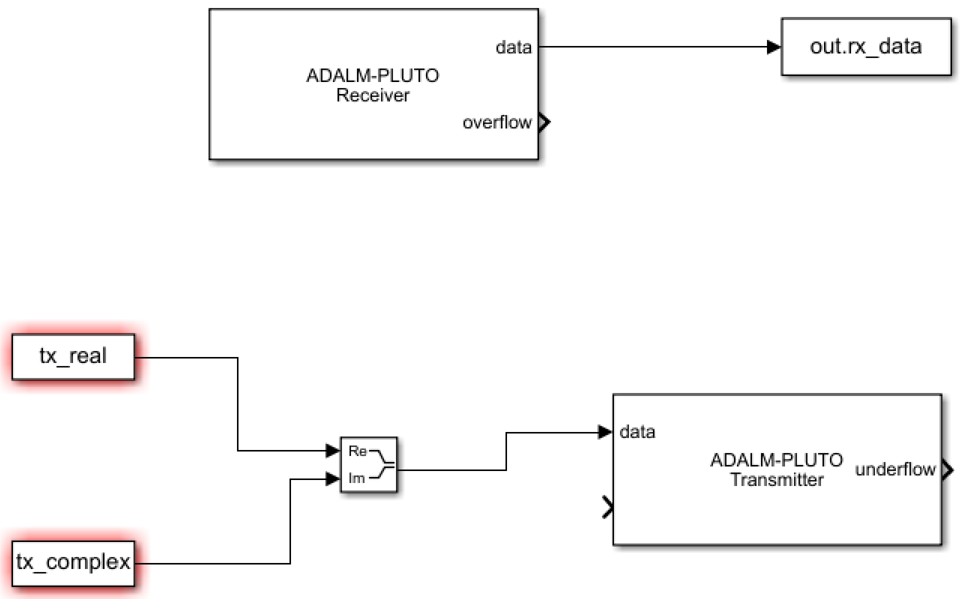


```
clear;
t =1;
fs = 1e6;

t_series = (0:1/fs:5)';
n= length(t_series);

nfft = 1024;         % number of samples per frame = FFT window
nframe = floor(n/nfft);         % number of frames
t_series = t_series(1:nframe*nfft);
x0_fd = qammod(randi([0 1],2*nfft,1),4,'InputType','bit','UnitAveragePower',true);

% Store in x0.
x0 = ifft(x0_fd);
% TODO:  Repeat the data x0 nframe times to create a vector x of length
% nframe*nfft x 1.
data = (repmat(x0,nframe,1));

tx_real = [t_series real(data)];
tx_complex = [t_series imag(data)];
```

Script producing the sounder signal

The actual interface to the PLUTO was done using the transmitter and receiver blocks of the SDR in Simulink. The signal data was fed from the workspace of the Matlab to the model in simulink. The received signal data was sent back to the workspace for the post processing.

The figure below shows the Simulink block model of the transceiver system employed. Since the transmitter block accepts a time series complex data , separate time series of real and imaginary parts were imported into the model and then fed to the transmitter block.

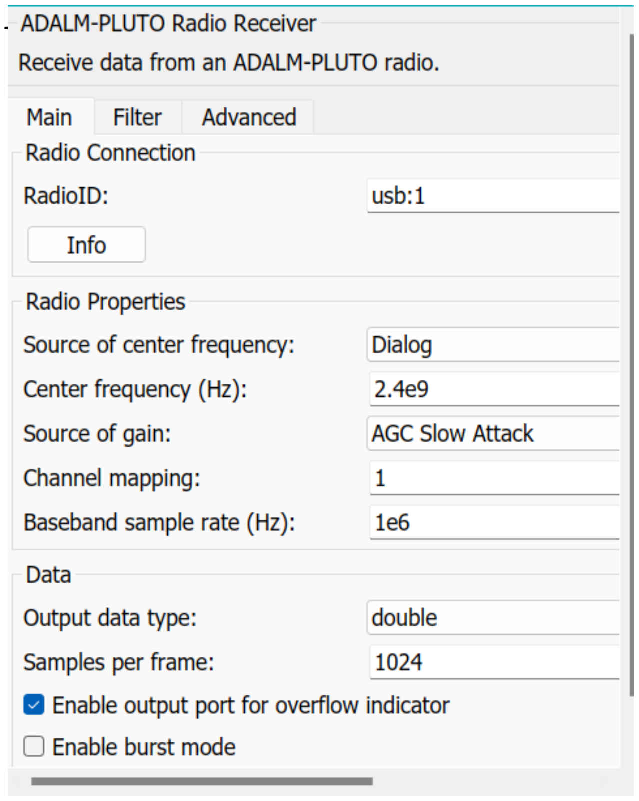


Block Diagram of the Simulink Model

The receiver and the transmitter blocks are highly configurable. I used the following parameters for the transmission and receiving.

Center Frequency : 2.4 GHz
Sample Rate : 1 MHz
Samples per frame of the transmitter: 1024

These parameters are the same for the generated signals in the workspace. The model will run the same time as the data and the receiver will receive the data at the same sampling frequency and frame length.


ADALM-PLUTO Radio Receiver

Receive data from an ADALM-PLUTO radio.

Radio Connection

RadioID: usb:1

Radio Properties

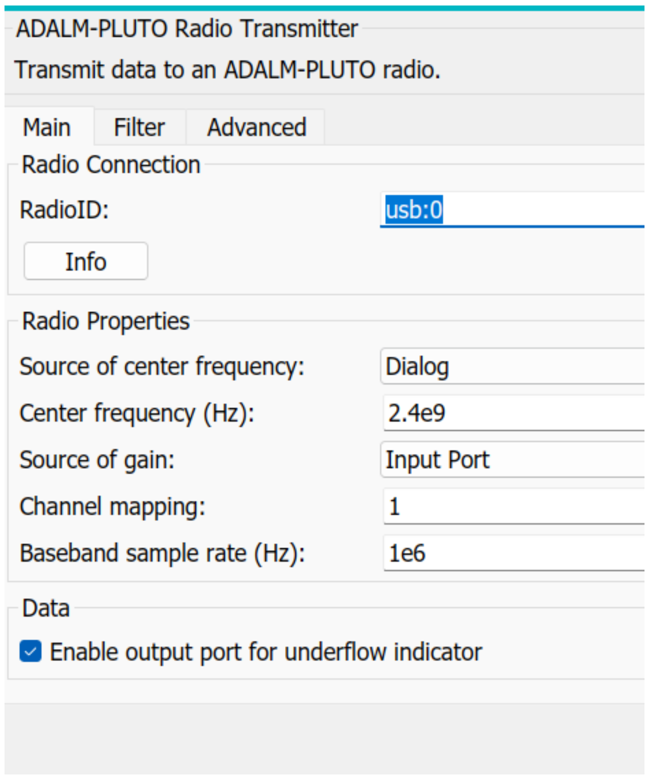
Source of center frequency:	Dialog
Center frequency (Hz):	2.4e9
Source of gain:	AGC Slow Attack
Channel mapping:	1
Baseband sample rate (Hz):	1e6

Data

Output data type:	double
Samples per frame:	1024

☑ Enable output port for overflow indicator

☐ Enable burst mode


ADALM-PLUTO Radio Transmitter

Transmit data to an ADALM-PLUTO radio.

Radio Connection

RadioID: usb:0

Radio Properties

Source of center frequency:	Dialog
Center frequency (Hz):	2.4e9
Source of gain:	Input Port
Channel mapping:	1
Baseband sample rate (Hz):	1e6

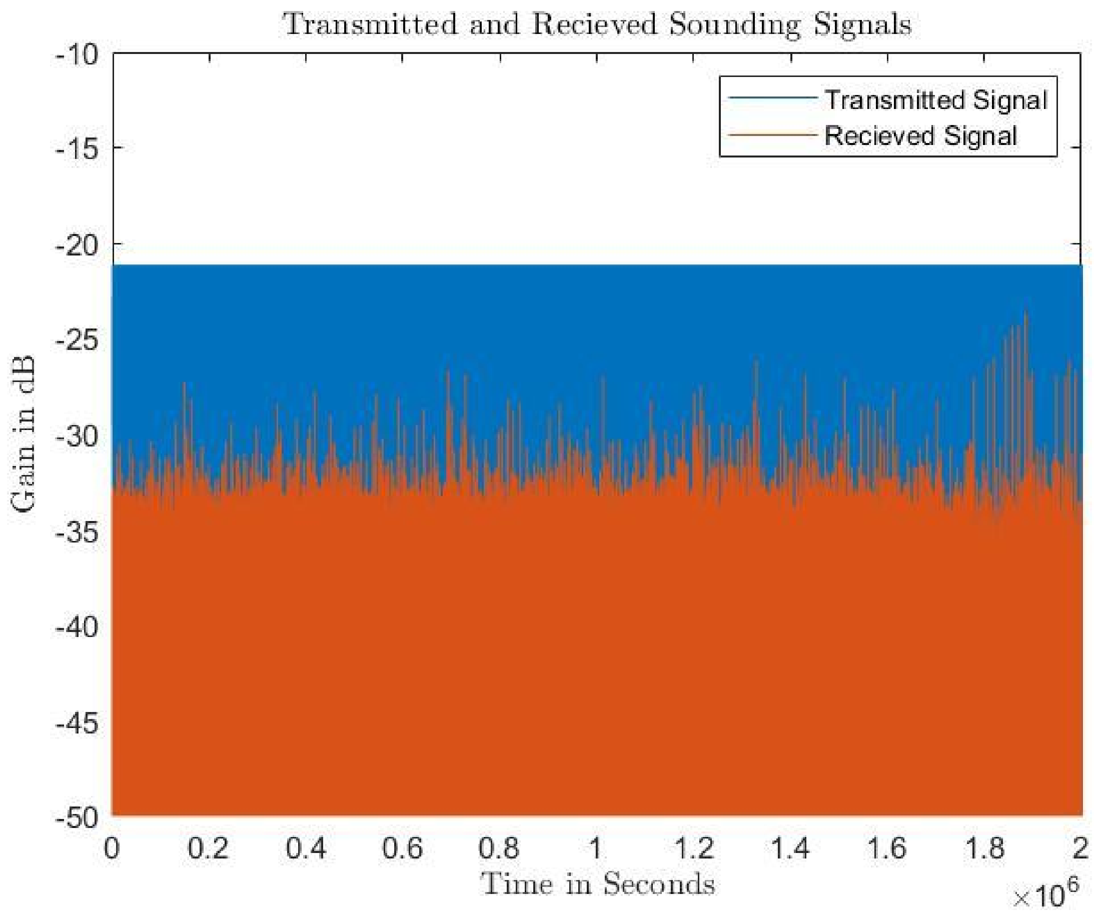
Data

☑ Enable output port for underflow indicator

Parameters of the Transmitter and Receiver Blocks

Results

The following graph shows the transmitted and received sounding signals in dB scale over a time period of 2 Seconds.



Here we can see the effect of the channel on the transmitted symbols. Now we can get the channel simply by dividing the received symbols in the frequency domain by the transmitted originals symbols.

This is done in the following script

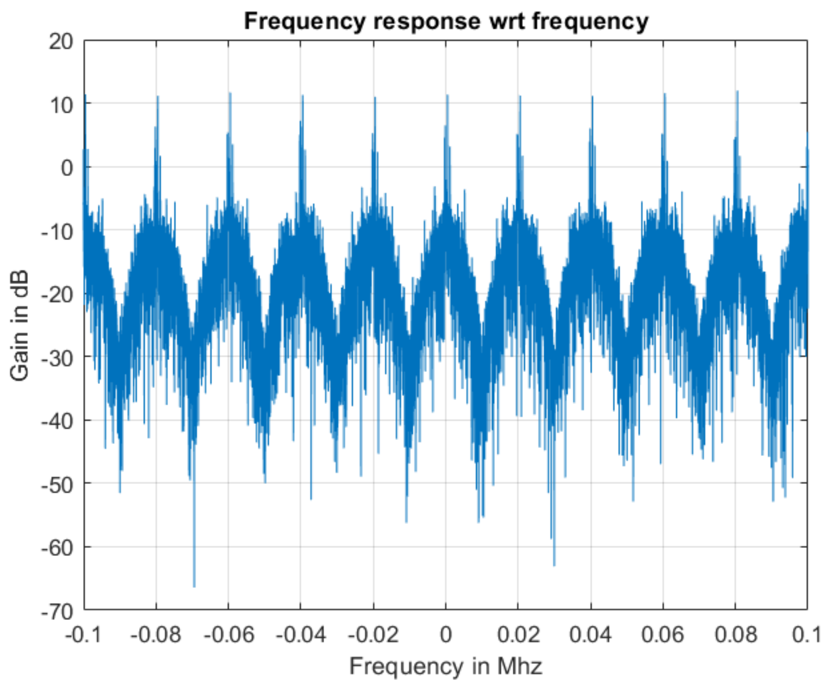
```
rxdata = out.rx_data;
n = 100
rxdat = rxdata(1:nfft*n);
```

```
yshape = reshape(rxdat,[nfft,n]);

yfd = fft(yshape,1024,1);
```

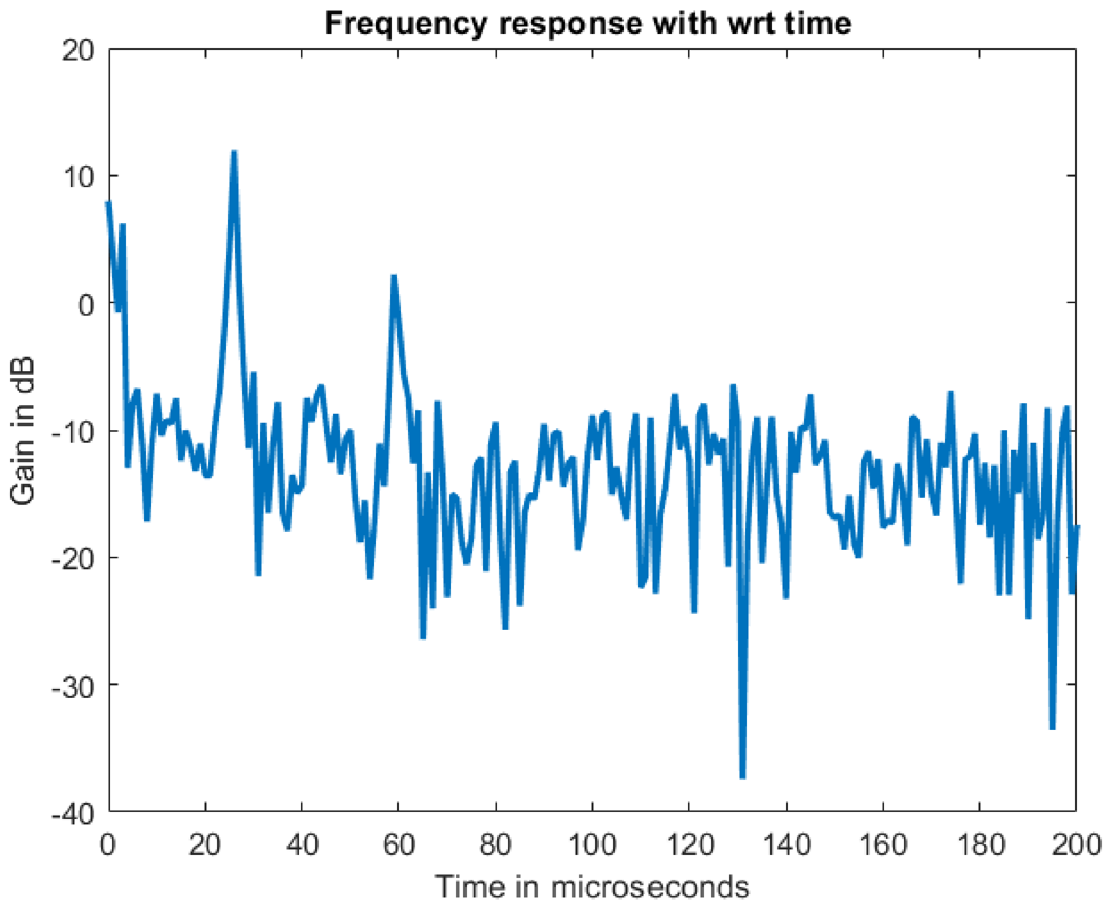
```
hestFd =zeros(nfft,n);


    for j = 1:n

        hestFd(:,j) = yfd(:,j)./x0_fd;
    end


fresp = reshape(hestFd,[nfft*n,1]);
```

Script to calculate the channel frequency response

We get the following frequency domain response of the channel .



The time domain response of the channel is :



We can see that the channel is rapidly changing in time.