Real Time Packet reception using SDR

Chenhong JI (4624238) & Vinay Balaji (4617363)
Kavya M Managundi(4616464)

Group KHV

March 29, 2017

1 Introduction

Frequency modulation and demodulation is widely used for radio transmissions for a wide variety of applications from broadcasting to general point to point communications. Frequency modulation, *FM* offers many advantages, particularly in mobile radio applications where its resistance to fading and interference is a great advantage. *FM* was invented and commercialized after Amplitude modulation(AM). Its main advantage is that it is more resistant to additive noise than *AM*. There are many methods for Frequency modulation and demodulation like zero crossing detection, slope detection etc. In this project we are using the SDR hardware platform in order to demodulate and receive the FM signal. Matlab is used as the algorithm development tool.

SDR is the software implementation of Radio components such as modulators, demodulators and amplifiers etc., which enables the radio signal processing and the production of different band radio signals. [1]Figure 1 shows the block diagram of SDR transmitter. In the block diagram we

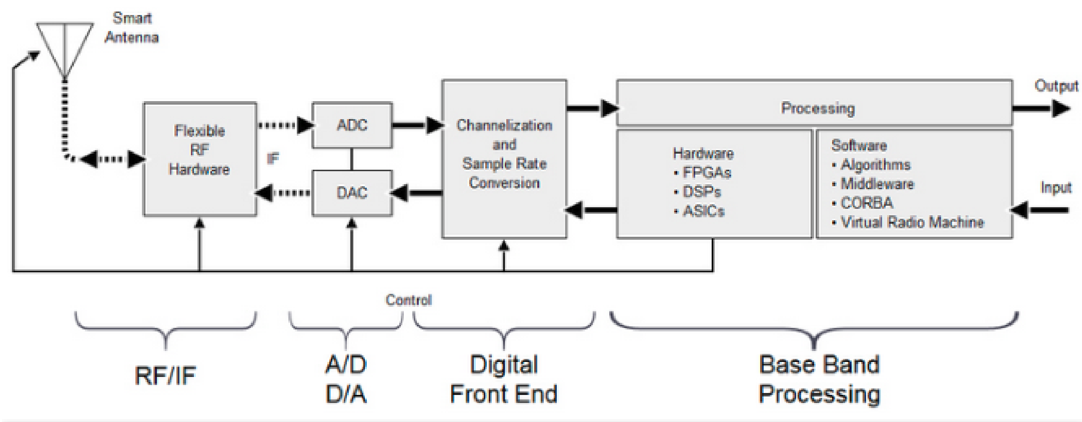

Figure 1: Block diagram of SDR transmitter

can see that in the receiver there is an *ADC* block which converts the analog signal into the digital signal this digitized signal is then filtered, demodulated and separated into individual signals. At the transmitter the signal is coded, modulated etc., and then converted from digital to analog in the *DAC* before transmitting the signal. As we can see that there is a *Flexible RF Hardware* block in between the *ADC*/*DAC* block that is because Practical RF/microwave circuit design makes it necessary to include that block. In this project the SDR hardware platform that we have chosen is the RTL-SDR[2]

[1]http://phonespeco.com/software-defined-radio-block-diagram
[2]http://www.rtl-sdr.com

2 FM Modulation

Frequency modulation is a kind of Angle modulation in which the instantaneous frequency of the carrier is varied according to a modulating signal. In *FM* the amplitude of the carrier is not varied hence it is also called as constant envelop method phase modulation. As shown in figure 2 it can be seen that as the modulating or base band signal voltage varies, so the frequency of the signal changes in line with it.

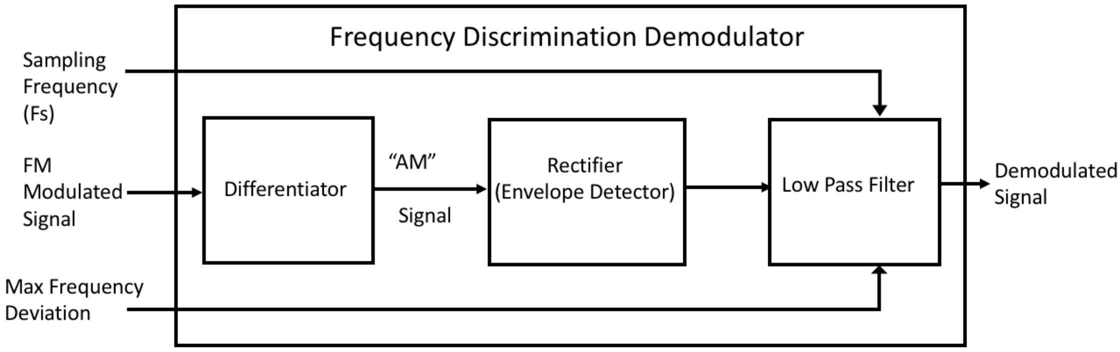

Figure 2: Frequency Modulation Waveform representation

2.1 Pre-Emphasis and De-Emphasis

In FM, the noise increases with modulation frequency. To compensate for this effect, FM communication systems have incorporated a noise-combating system of *pre-emphasis* and *de- emphasis*. *Pre-emphasis* provides increased amplitude to the higher modulating frequencies prior to modulation under a well-defined *pre-emphasis* (high-pass filter) curve. This added amplitude will serve to make the higher frequencies more immune to noise by increasing their index of modulation. *De-emphasis* is just the opposite operation (using a low-pass filter) and it is done at the receiver.

2.2 Wide band and Narrow band FM

There are a number of factors which govern the level of modulation. Such as bandwidth, applications, quality and many others. For example in broadcast *FM* transmissions the aim is to be able to transmit high quality audio and to achieve this, high levels of deviation are used and the bandwidth is wide. For communications purposes, quality is not the issue, but bandwidth is more important. Accordingly deviation levels are less and the bandwidth is much smaller. Depending on this *FM* is classified as *Narrow band FM* and *wide band FM.*

1. *Wide band FM*: Signals with high levels of deviation with a broadcasting deviation of +/- 75 kHz are called *Wide band FM.* Broadcast FM stations use *wide band FM* which enables them to transmit high quality audio, as well as other facilities like stereo, and other facilities like RDS, etc.

2. *Narrow band FM*: Signals with low levels of deviation are called *narrow band frequency modulation (NBFM)* with a broadcasting deviation of +/- 3 kHz . For NBFM the audio or data bandwidth is small, but this is acceptable for this type of communication, which generally is point to point communications.

2.3 Mathematical Analysis

The carrier signal is given by equation 1

$$v_c(t) = V_c cos(\omega t + \phi_c) \tag{1}$$

where $(\omega_c t + \phi_c)$ represents the angle of the carrier. We achieve *FM* by varying the frequency, ω_c. The message signal $m(t)$ controls the frequency f_c of the carrier.

Now the FM signal is given by equation 2

$$v_s(t) = V_c cos(2\pi(f_c + \Delta f)t) \tag{2}$$

where Δf is the frequency deviation which depends on $m(t)$.
For instantaneous carrier signal the equation is given by equation 3

$$v_c cos(\omega_i t) = V_c cos(2\pi(f_i t) = v_c cos(\phi_i) \tag{3}$$

where ϕ_i is the instantaneous angle which is $\phi_i = \omega_i t = 2\pi f_i t$ also f_i is the instantaneous frequency.

Since $\phi = 2\pi f_i t$ the $\frac{d\phi_i}{dt}$ is equal to $2\pi f_i$ or the instantaneous frequency $f_i = \frac{1}{2\pi}\frac{d\phi_i}{dt}$ i.e., frequency is proportional to the rate of change of angle.

If f_c is the unmodulated carrier and f_m is the modulating frequency, then we may deduce that

$$f_i = f_c + \Delta f_c cos(\omega_m t) = \frac{1}{2\pi}\frac{d\phi_i}{dt} \tag{4}$$

where Δf_c is the peak deviation of the carrier. Hence, we have

$$\frac{d\phi_i}{dt} = \frac{1}{2\pi} f_c + \frac{1}{2\pi}\Delta f_c cos(\omega_m t) \tag{5}$$

Therefore now

$$\phi_i = \int \frac{d\phi_i}{dt} = \int (\omega_c + \frac{1}{2\pi}\Delta f_c cos(\omega_m t))dt \tag{6}$$

$$\phi_i = \omega_c t + \frac{2\pi \Delta f_c sin(\omega_m t)}{\omega_m} \tag{7}$$

$$\phi_i = \omega_c t + \frac{\Delta f_c}{f_m} sin(\omega_m t) \tag{8}$$

Hence for the FM signal, given in equation 9

$$v_s t = V_c cos(\phi_i) \tag{9}$$

can be re-written as equation 10

$$v_s t = V_c cos(\omega_c t + \frac{\Delta f_c}{f_m} sin(\omega_m t)) \tag{10}$$

The *Modulation Index* β is the ratio of peak frequency deviation and the modulating frequency. Which is given by equation 11

$$\beta = \frac{\Delta f_c}{f_m} \tag{11}$$

We must note that $m(t)$ here s considered as a single modulating signal of the form given in equation 12

$$m(t) = V_m cos(\omega_m t)) \tag{12}$$

The equation 10 can be expressed in terms of a *Bessel function*.

$$v_s(t) = V_c \sum_{n=-\infty}^{\infty} J_n(\beta) cos(\omega_c + n\omega_m)t \tag{13}$$

Where $J_n(\beta)$ are the *Bessel functions* of the first kind. Expanding the equation 13 we get equation 14

$$\begin{aligned} v_s(t) = V_c J_0(\beta) cos(\omega_c)t + V_c J_1(\beta) cos(\omega_c + \omega_m)t \\ + V_c J_{-1}(\beta) cos(\omega_c - \omega_m)t + V_c J_2(\beta) cos(\omega_c + 2\omega_m)t \\ + V_c J_{-2}(\beta) cos(\omega_c - 2\omega_m)t + ... \end{aligned} \tag{14}$$

The figure 3 shows the FM Signal Spectrum. The amplitudes drawn are completely arbitrary, since we have not found any value for $J_n(\beta)$ - the figure 3 is only to illustrate the spectrum. The FM spectrum contains a carrier component and an infinite number of sidebands at frequencies $f_c \pm n f_m (n = 0, 1, 2, ..)$

The *Band width* approximation for an FM signal is given by $BW = 2(Maximum frequency deviation + highest modulated frequency)$ which is the **Carson's Rule**

$$Bandwidth = 2(\Delta f_c + f_m) \tag{15}$$

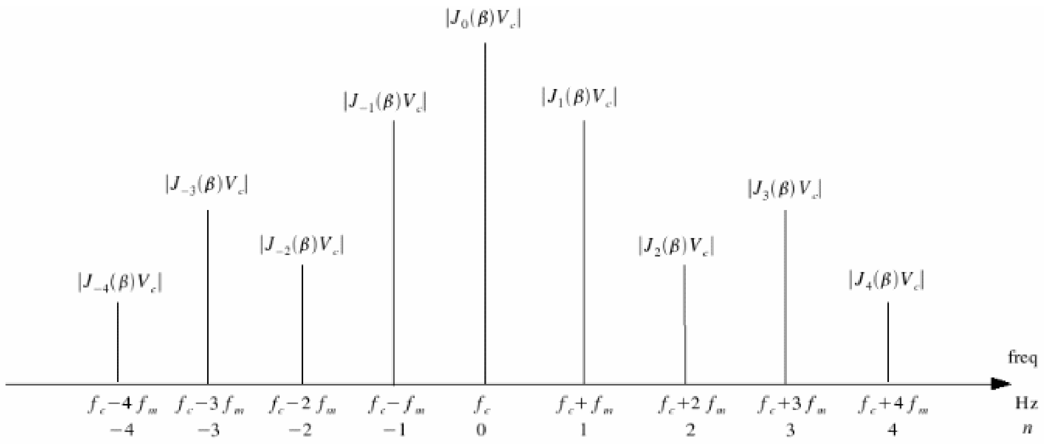

Figure 3: FM Signal Spectrum

3 Observations made on `SDR console version 2.3`

In order to get a better view of how the SDR radio works in real time and to play around with the vaious parametersof the radio, we downloaded SDR Console v2.3. This provided us with a graphical interace and we could see the available FM bands and analyse the spectrum of the signals.The figure 4 shows the SDR Console window.

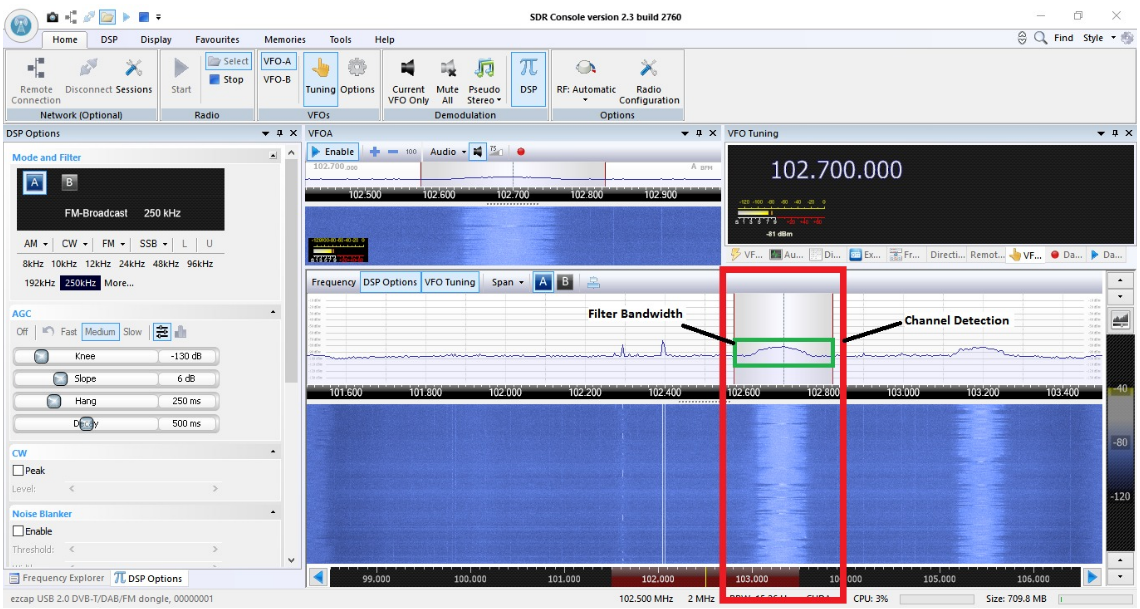

Figure 4: Console view of SDR

1. While running the demodulation module, the filter applied had a value of 250KHz. This could be seen as the frequency deviation from the center frequency as we reduce this value we experience a loss of information.
2. The FM receiver with noise filter and de emphasis is important as it suppresses the noise and spurious gains of the input signal. Without the de emphasis filter the output becomes loud and there is a large amount of noise present.
3. The SDR seems to have difficulty picking up signals in concrete buildings as the signals recieved were not very clear.

4 *FM* Demodulator implementation

As shown in figure 5, the FM demodulator consists of the differentiator, following by the envelope detector and the low pass filter. In this case the rectifier is used as a rectifier. The low pass filter is able to remove the noises outside the frequency deviation. Undersampling can be implemented after the demodulator for music output.

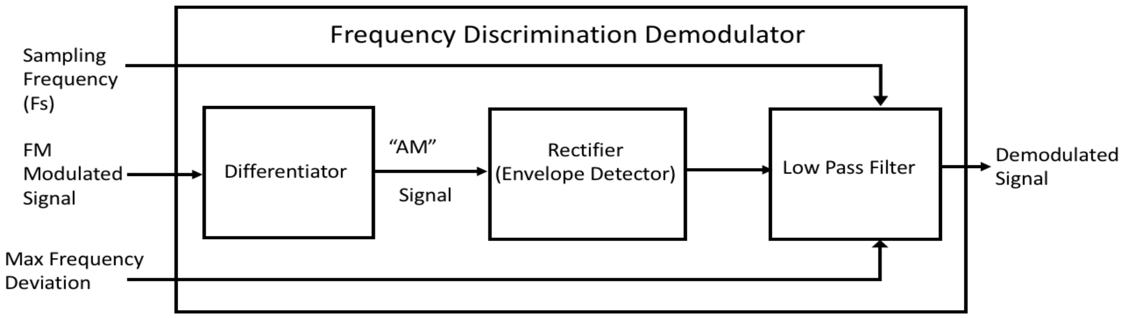

Figure 5: Block diagram of FM Demodulation

Figure 6 shows the block diagram of general demodulation. The signal output from the `RTL-SDR` is referred to as `x[n]` which contains at least one signal, receiver front-end noise and quantization noise. The signal flow makes use of *decimation* a *multirate* signal processing technique. From the sampling theory we know that the *sampling rate* should be twice the highest frequency of the sampling signal in order to avoid aliasing. If there is bandwidth reduction after passing this signal to the *low pass filter* it means that the effective sampling is greater than needed. The *downsampling* block(arrow pointing down followed by an integer factor) is added to keep every M^{th} sample and to discard the rest. Now the combination of *low pass filter* along with *downsampler* forms a *decimator* where the decimation takes palce by M, for input sampling rate f_s the output sampling rate becomes f_s/M.

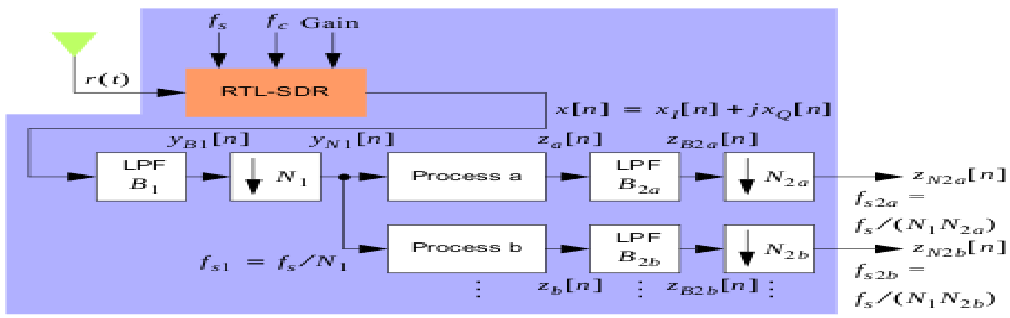

Figure 6: Implementation of basic FM Demodulation

Figure 7 shows the block diagram of the *FM Receiver*. The *Matlab* code for the implementation of the same is given in *Appendix A*. The code for the implementation of *Low pass filter* used in the *FM receiver* is also in *Appendix A*.

To support the *SDR* algorithm rely on digital signal (DSP) we will be using the $DSPToolbox^{TM}$ in `MATLAB` as well as other custom functions too implement demodulators. The design parameters that we have used are `sample rate`, $f_s = 2.4MHz$ that is to capture the signal. `RF gain` as $35dB$. `Filter Bandwidth` as $250kHz$. This bandwidth is used because when experimented with *SDR* this is the bandwidth that gives the best reception. The *sampling rate* that is given as input in order to achieve the *FM* reception is $240kHz$.

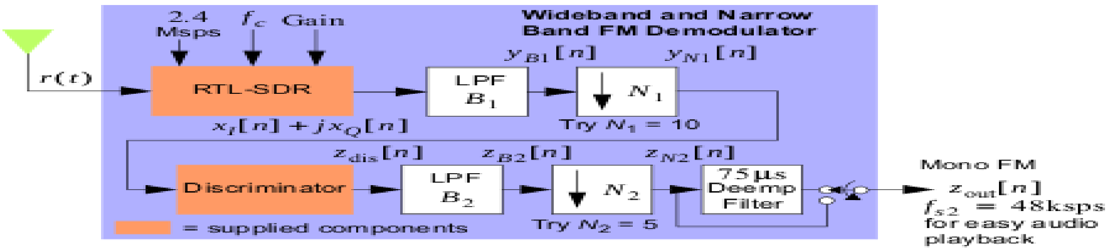

Figure 7: Block diagram for Implementation of FM Receiver

5 Radio Data System (RDS)

Now that the *FM radio reception* is done. Our next focus is to demodulate the *RDS*. Now We know that *FM* has it wide application in radio broadcast. To transmit stereo music, FM is enhanced by stereo multiplexing which carries both L and R audio channel content. The Radio Data System (RDS) enables FM to carry text information such as traffic, weather, and radio station information which can be displayed on the end-userâĂŹs device interface. Currently, growing number of mobile phones and consumer mobile devices will have an integrated FM receiver feature. The FM transmitter feature is also becoming popular for allowing users to transmit audio content from their mobile devices through their car radio. To make sure the FM-related functions work well, we have earlier tested the FM mono and FM stereo, now the FM RDS functions need to be tested in production.

In *FM stereo broadcasting* the system performs the multiplexing of two signals and further combines them into a complex baseband signal that modulates the FM carrier. The block diagram of which can be seen in figure 8.

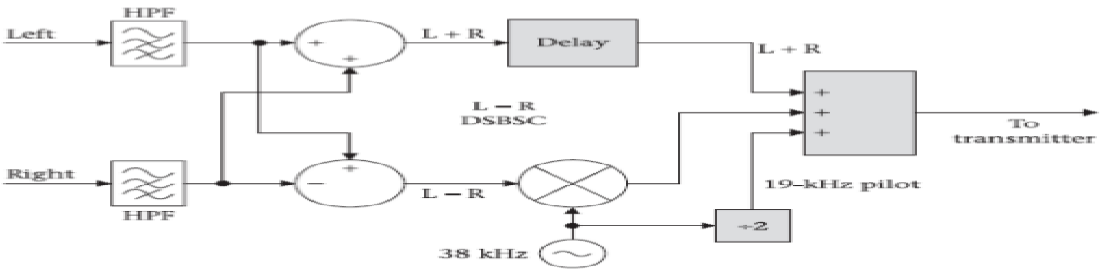

Figure 8: Block diagram of FM Stereo Multiplexing

From the figure 8 we can see that the `L+R` signal and `L-R` signals are fed to the balanced modulator along with additional 38-kHz signal, the output of which is the DSBSC AM signal centered at 38 kHz. The resulting stereo-generator output-signal spectrum is shown in Figure 9 Baseband

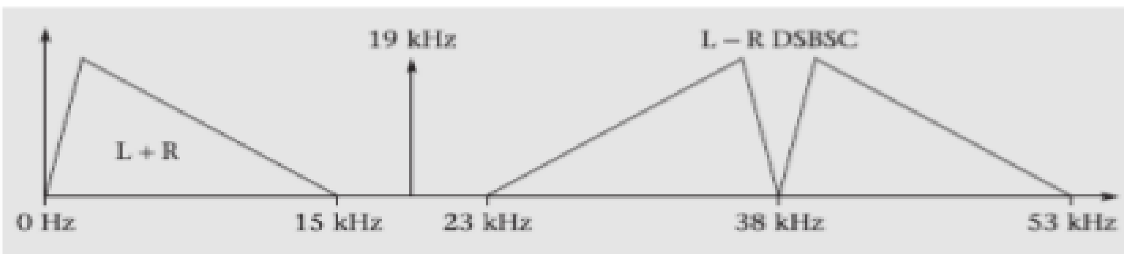

Figure 9: FM Stereo Multiplexing Spectrum

frequencies from near $0Hz$ to $15kHz$ in the spectrum are the base band frequencies. $19kHz$ is a pilot sub-carrier signal. $23kHz$ to $53kHz$ is the frequency range which is DSBSC amplitude modulated by a 38-kHz tone.

RDS baseband modulation is a kind of *differentially-coded BPSK*. The sub-carrier is amplitude-modulated by the shaped and binary-phase coded data signal. The sub-carrier is suppressed. This method of modulation may alternatively be thought of as a form of two-phase phase-shift-keying (PSK) with a phase deviation of Âś 90 degree. The RDS sub-carrier frequency is 57kHz which can be visualized in Figure 10 The reference RDS design is based on the block diagram in figure

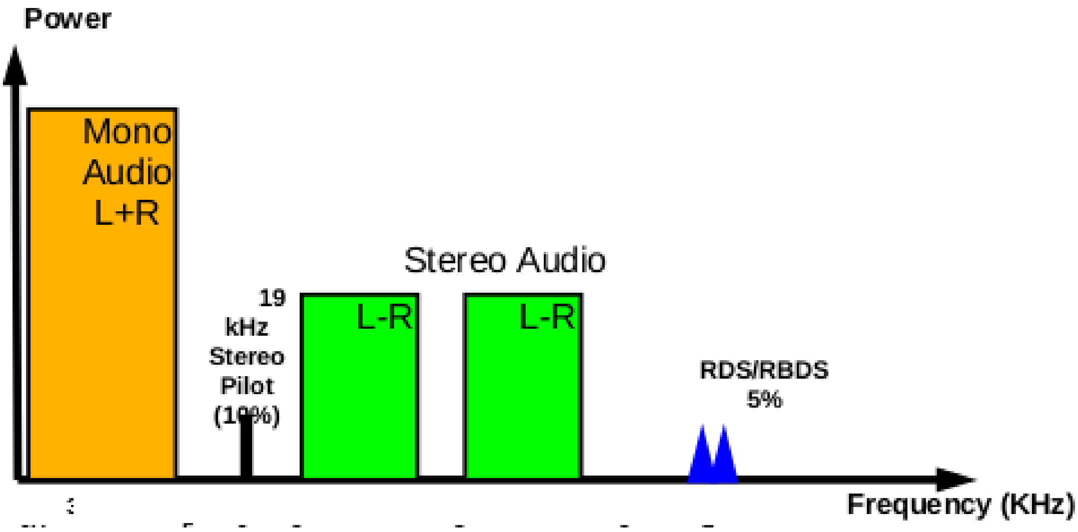

Figure 10: FM-Stereo-RDS Baseband spectrum

11 which is for demodulating stereo FM However we need to find the third harmonic, so our multiplication constant would be 3 instead of 2. We then pass the signal to the low pass filter and then sample it. There no need to for the steps after the LPF as we are not trying to acquire a stereo FM. The matlab code for the *RDS receiver* is in `Appendix A`. The output of which is

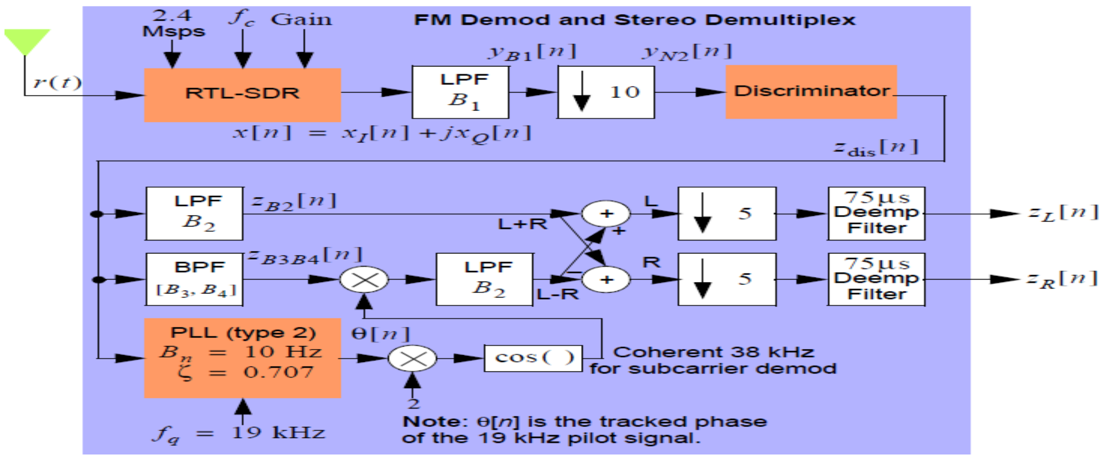

Figure 11: Demodulating Stereo FM

shown in the figure 13. The output isolates the RDS signal at 57khz. That is the The first power spectrum output after passing it through
texttttBPF. The final power spectrum obtained after multiplying with PLL signal is as shown in figure 12

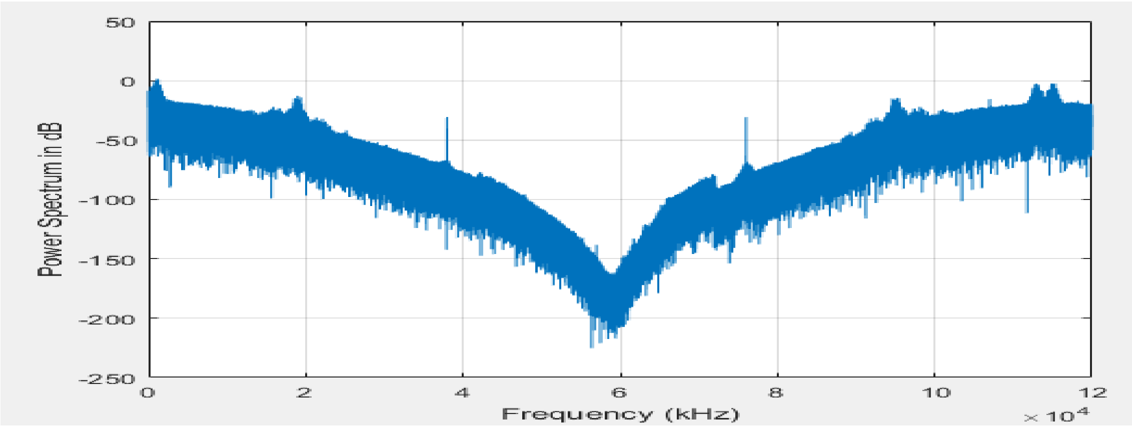

Figure 12: RDS final power spectrum

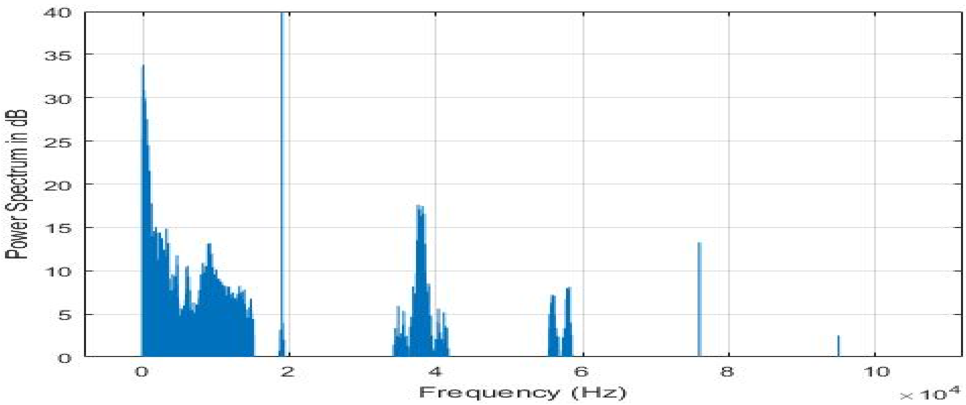

Figure 13: RDS first power spectrum output

A Appendix

A.1 Code for FIR filter implementation.

```
function h = FIR(fs ,fc ,M);
% 64-tap Low Pass filter design
% FIR filter design with Hanning Window
% fs: sampling frequency
% fc: passband frequency
clear all

beta = 0; %% Lowpass filter must be ideal and symmetric.
alpha = 0;  %% Using the case when the first sampling point is at zero frequency.
M = 64;

for i = 0:1:M-1
    if((i<(fc/fs*M-alpha))||(i>((1-fc/fs)*M-alpha)))
        G(i+1) = (-1)^i;
    else
        G(i+1) = 0;
    end
end

%%%%!!!!!!!!!!!!!!!!!!!!
U =  floor(fc/fs*M-alpha);
```

```
%% FREQUENCY RESPONSE
k = 0:M-1;
n = 0:M-1;


ht = zeros(M,1);
for i = 1:1:M % n
     for j = 2:1:U+1 % k
          ht (i) = ht(i)+G(j)*cos(2*pi*(j-1)/M*(i-1+0.5));
     end
     ht(i) = (G(1)+2*ht(i))/M;
end

% plot(n,ht);
% figure(2)
% plot(abs(fft(ht,4096)));
```

A.2 Code for `FM receiver`

```
%% GABRIEL_V1: Demodulating FM signal from scratch

%% Read complex samples from the rtlsdr file
% SDR>rtl_sdr -s f_in_sps -f f_in_Hz -g gain_dB capture.bin
% Example: SDR>rtl_sdr -s 2400000 -f 70000000 -g 25 capture70R1k.bin

fid = fopen('capture.bin','rb');
y = fread(fid,'uint8=>double');
y = y-127;
y = y(1:2:end) + i*y(2:2:end);
z=length(y);

%% Low Pass filter Design

N=6;
cutoff=100000;
fs=240000;
%[b,a]=butter(N,cutoff/(fs/2),'low');
%lpf_1=filter(b,a,y);

h = FIR(fs,cutoff);
lpf = conv(h,y);
%% Decimating signal
x1=decimate(lpf,10);

%% Discriminator

X=real(x1);             % X is the real part of the received signal
Y=imag(x1);             % Y is the imaginary part of the received signal
N1=length(x1);           % N is the length of X and Y
b=[1 -1];               % filter coefficients for discrete derivative
a=[1 0];                %      "
derY=filter(b,a,Y);     % derivative of Y,
derX=filter(b,a,X);     %      "          X,

disdata=(X.*derY-Y.*derX)./(X.^2+Y.^2);

%% The Second low pass filter
```

```
N=6;
cutoff=5000;
fs=240000;
%[b1,a1]=butter(N,cutoff/(fs/2),'low');
%lpf1=filter(b1,a1,disdata);
h1 = FIR(fs,cutoff);
lpf1 = conv(h1,disdata);
%% Decimate the second time to the sound card

x2=decimate(lpf1,5);

%% Hear the signal

sound(x2,48000)
```

A.3 Code for RDS receiver

```
%% ARIEL_V1: Demodulating FM signal from scratch with RDS

%% Read complex samples from the rtlsdr file
% SDR>rtl_sdr -s f_in_sps -f f_in_Hz -g gain_dB capture.bin
% Example: SDR>rtl_sdr -s 2400000 -f 70000000 -g 25 capture70R1k.bin

fid = fopen('106_3Mhz.bin','rb');
y = fread(fid,'uint8=>double');
y = y-127;
y = y(1:2:end) + i*y(2:2:end);


%% Low Pass filter Design

N=6;
cutoff=100000;
fs=240000;
h = FIR(fs,cutoff);
lpf = conv(h,y);
%% Decimating signal
x1=decimate(lpf,10);

%% Discriminator

X=real(x1);             % X is the real part of the received signal
Y=imag(x1);             % Y is the imaginary part of the received signal
N1=length(x1);            % N is the length of X and Y
b=[1 -1];              % filter coefficients for discrete derivative
a=[1 0];               %    "
derY=filter(b,a,Y);    % derivative of Y,
derX=filter(b,a,X);    %    "          X,

disdata=(X.*derY-Y.*derX)./(X.^2+Y.^2);


%% The Second low pass filter
N=6;
cutoff=5000;
fs=240000;
h1 = FIR(fs,cutoff);
lpf1 = conv(h1,disdata);
```

```
%% Decimate the second time to the sound card

x2=decimate(lpf1 ,5);

%% Hear the signal

%sound(x2,48000)

%% Band Pass Filter for RDS


cutoff1=59000;
cutoff2=56000;
[b2,a2]=butter(N,cutoff2/(fs/2),'high');
bhpf=filter(b2,a2,disdata);
[b1,a1]=butter(N,cutoff1/(fs/2),'low');
blpf=filter(b1,a1,bhpf);



%% PLL loop with 19khz VCO
fs3=240000;
[theta, phi_error] = pilot_PLL(disdata,19000,fs3,2,10,0.707);
b=[3];
angles=theta*b;
angles2=cos(angles);
%c=[];
%for m=1:100:1000000
    %c=[c (m/fs3)];
%end
%plot(c,phi_error(1:100:1000000))
%% Multiply BPF output to PLL output
result=blpf.*angles2;
z=length(result);
simpleSA(result,z,240000);

%% Low Pass filter
cutoff3=22000;
fs3=350000;
[b3,a3]=butter(N,cutoff3/(fs3/2),'low');
fresult=filter(b3,a3,result);



%g=interp(fresult,20);

%z=length(g);
%simpleSA(g,z,17500,0,40);
```