

일반논문 (Regular Paper)

방송공학회논문지 제24권 제3호, 2019년 5월 (JBE Vol. 24, No. 3, May 2019)

<https://doi.org/10.5909/JBE.2019.24.3.485>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

RTL-SDR을 이용한 스테레오 주파수 변조 방송의 실시간 수신기 구현

김 영 주^{a)‡}

Implementation of Real-time Stereo Frequency Demodulator Using RTL-SDR

Young-Ju Kim^{a)‡}

요 약

주파수 변조 방식의 방송 주파수에 동조되는 안테나와 Realtek 社의 RTL2832 칩을 이용하는 디지털 TV용 튜너와 아날로그-디지털 변환기로 구성되는 universal serial bus (USB) dongle을 이용하여 스테레오 주파수 변조 방송의 실시간 수신기를 컴퓨터의 소프트웨어로 구현한다. 아날로그 방송 신호가 USB dongle에서 디지털 신호로 변환되고 이진 데이터를 컴퓨터에서 매트랩 및 파이썬 프로그래밍 언어의 신호처리 기법을 이용하여 저역 통과 필터, 대역 통과 필터, 주파수 판별기, 양측파대 진폭 복조, 위상 고정 루프, 샘플링 변환, 디엠퍼시스 등의 기능 블록을 설계한다. 최종적으로 수신기의 실시간 구현을 위하여 파이썬 및 C++로 구성되는 그누라디오 (GNU Radio)를 이용하여 수신기 알고리즘을 소프트웨어로 구현한다.

Abstract

A software-driven real-time frequency de-modulator is implemented with the aid of universal-serial-bus (USB) type software defined radio dongle. An analog stereo frequency modulation (FM) broadcast signal is down-converted to the baseband analog signal then converted to digital bit streams in the USB dongle. Computer software such as Matlab, Python, and GNU Radio manipulates the incoming bit streams with the technique of digital signal processing. Low pass filtering, band pass filtering, decimation, frequency discriminator, double sideband amplitude demodulation, phase locked loop, and deemphasis function blocks are implemented using such computer languages. Especially, GNU Radion is employed to realize the real-time demodulator.

Keyword : SDR, GNU Radio, RTL-SDR, discriminator, deemphasis

a) 충북대학교 전자정보대학 정보통신공학부(College of Electrical and Computer Engineering, School of Information and Communication Engineering, Chungbuk National University)

‡ Corresponding Author : 김영주(Young-Ju Kim)

E-mail: yjkim@cbnu.ac.kr

Tel: +82-43-261-3375

ORCID: <https://orcid.org/0000-0002-5844-8612>

· Manuscript received December 4, 2018; Revised March 11, 2019; Accepted May 13, 2019.

I. 서론

라디오 주파수 (radio frequency, RF)의 송수신 신호를 아날로그 및 디지털 하드웨어가 아닌 소프트웨어로 처리하는 소프트웨어로 정의된 라디오 (software defined radio, SDR)라는 용어가 등장한 지 약 30년이 되었다^[1]. SDR은 통신 시스템의 물리 계층 (physical layer, PHY)과 라디오 주파수 회로의 일부를 이산 신호처리 (discrete signal processing, DSP) 기법을 활용하는 소프트웨어로 처리하는 것이다. 이상적인 SDR 수신기는 안테나에 연결된 튜너 회로와 아날로그-디지털 변환 (analog-to-digital converter, ADC) 부로 구성된다. 최근에는 그림 1과 같이 라파엘 마이크로 (Rafael Micro) 사의 R820T2 튜너 칩과 리얼테크 (Realtek) 사의 RTL2832U 아날로그-디지털 변환 반도체 칩을 사용하는 미화 20달러 내외의 RTL-SDR이라 불리는 소프트웨어 라디오가 출시되면서 SDR을 이용한 통신 시스템을 구성하는데 진입 장벽이 매우 낮아져 광범위한 분야에서 SDR이 사용되고 있다^[2]. 2010년 에릭 프라이 (Eric Fry)가 위의 두 개 칩에서 8 비트의 비부호 정수 (unsigned integer)의 동 위상 (inphase, I) 및 직교 위상 (quadrature, Q) 신호를 얻을 수 있음을 발견하면서 매우 저가의 SDR로 사용되게 되었다. 위 두 개 칩은 원래 유럽에서 주파수 변조 방송, 디지털 오디오 방송, 디지털 비디오 방송 수신용으로 개발된 칩셋으로 수신 주파수 대역은 20MHz에서 1900MHz에 이른다. 최대 샘플링율은 3.6Mps (symbol per second)로 매우 다양한 무선 신호를 수신할 수 있다. 본 논문에서는 SDR의 다양한 응용분야 중에서 가장 기초가 되는 주파수 변조 방

송 신호를 수신하고 수신기 알고리즘을 매트랩, 파이선으로 작성하여 각 기능 블록을 검증한다. 그리고 실시간 수신기 구현을 위하여 그누라디오로 수신기의 각 기능 블록을 구현한다.

II. RTL-SDR의 구조 및 동작

본 논문에서 사용하는 RTL-SDR의 동작을 더 잘 이해하기 위해 그림 2와 같은 동작 수준 (behavioral level) 모델을 보인다. 안테나에서 입력되는 신호 $s(t)$ 는 반송파 주파수 f_c 에 주파수 변조된 방송 신호가 저잡음 증폭기 (low noise amplifier)를 통과한 신호이다. R820T 튜너에서는 $f_0 = -f_c$ 의 국부 발진기 (local oscillator)와 수신 신호를 입력으로 하는 믹서에 의해 수신 신호의 중심 주파수가 f_0 만큼 이동

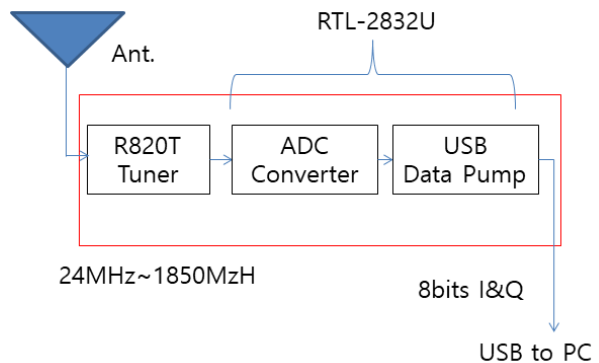


그림 1. 리얼테크 사 RTL-SDR 블록 다이어그램
Fig. 1. Block diagram of Realtek RTL-SDR

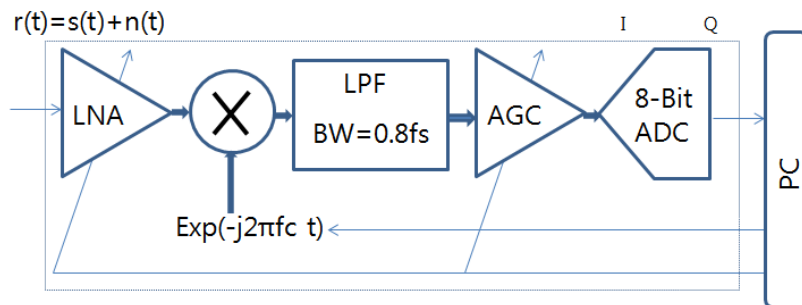


그림 2. RTL-SDR의 동작 흐름도
Fig. 2. A behavioral level model of the RTL-SDR

된다. 기저대역 신호로 이동된 신호는 저역통과 필터(low pass filter)를 통과하여 다른 방송 주파수 신호들을 걸러낸다. 샘플링 이론에 의하면 입력되는 신호의 최대 주파수는 샘플링을 f_s 의 1/2인 $f_s/2$ 보다 작아야 한다. 실제 구현되는 필터는 통과 대역과 비통과 대역 사이에 천이 (transition) 구간이 있으므로 가용 대역폭은 f_s 의 80%가 된다. 그림 2의 믹서에 $e^{j\theta} = \cos\theta + j\sin\theta$ 가 곱해지므로, 이후의 신호들은 복소수가 된다. 그림 2의 저역 통과 필터는 실수부와 허수부로 이루어진다. 실제의 하드웨어에서 이 저역 통과 필터는 두 개로 분리가 되어, R820T 튜너 쪽의 실수 통과 대역 필터와 RTL2382U 쪽의 I/Q 디지털 필터로 구성된다. 튜너와 저역통과 필터링된 아날로그 신호는 양자화기 (quantizer) $Q[\cdot]$ 를 지나 복소수 이산 신호 $r[n]$ 이 된다.

$$r[n] = G \cdot LPF[r(t)e^{-j2\pi f_c t}]_{t=nT=n/f_s} \quad (1)$$

이때 G 는 자동 이득 조정 (automatic gain control) 값이고, $LPF[\cdot]$ 은 저역 통과 필터링 함수이다.

RTL-SDR의 양자화기 $Q[\cdot]$ 의 출력은 샘플 당 실수부와 허수부 각각 8비트가 나온다. 컴퓨터에서 양자화기 출력 신호를 얻기 위해서는 소프트웨어 드라이버가 필요하다. 복조기를 매트랩 또는 파이썬을 이용하여 구현할 경우 이러한 드라이버 소프트웨어들이 필요하며 오스모콤 (Osmocom) 사이트에서 구할 수 있다. RTL-SDR USB 동글의 드라이버는 윈도우 응용 프로그램인 Zadig를 이용한다. 예를 들어 윈도우에서 매트랩을 이용하여 복조기를 구현할 경우 우선 Zadig 프로그램을 실행하여 SDR 설정을 하고 프로그램이 동작하는 디렉토리에 동적 링크 라이브러리 (dynamic link library, dll) 파일들과 실행 (execution, exe) 파일들을 가지고 있어야 한다. RTL-SDR의 데이터를 얻기 위해서는 rtl_sdr.exe 파일을 실행한다. 94.1MHz의 클래식 FM 방송을 수신하기 위해서는 다음의 명령어를 윈도우 명령 프롬프트에서 실행시킨다³⁾.

`c:\sdrsharp\rtl_sdr.exe -s 2.4e6 -f 94.1e6 -g 30 classicfm.bin`

위 명령어는 클래식 FM의 주파수인 94.1 MHz 방송 신

호를 초당 2.4 메가 샘플로 샘플링하고 튜너 및 기저대역의 신호이득을 30dB로 하여 classicfm.bin의 파일 이름으로 저장시킴을 의미한다. classicfm.bin 파일에는 이진수 값들이 저장되므로, 소프트웨어로 이산 신호 처리를 하기 위해서 매트랩에서 아래와 같은 함수를 만들어 복소수 값으로 변환을 시킨다.

```
function y = loadFile(filename)
% y = loadFile(filename)
% reads complex samples from fm941.bin
%
fid = fopen(filename,'rb');
y = fread(fid,'uint8=>double');
y = y-127;
y = y(1:2:end) + i*y(2:2:end);
```

맷랩에서 `x=loadFile('fm941.bin')` 명령을 실행시키면 변수 `x`에 RTL-SDR에서 컴퓨터로 입력되는 기저대역 신호가 복소수 값으로 변환된다.

III. 복조기 구조 및 주파수 판별기의 구현

주파수 변조된 반송파는 메시지 신호 $m(t)$ 를 반송파 신호 $x_c(t)$ 의 위상에 싣는다. 반송파 신호의 위상의 미분 값이 메시지 신호에 비례한다.

$$x_c(t) = A_c \cos[2\pi f_c t + \phi(t)] = A_c \cos\left[2\pi f_c t + 2\pi f_d \int^t m(\tau) d\tau\right] \quad (2)$$

이때, f_d 는 주파수 변조의 주파수 편이 상수이다. 주파수 변조 방송 신호를 복조하기 위해서는 우선 이상적인 주파수 판별기를 고려한다^{4)[5][10]}.

$$y_D = \frac{1}{2\pi} K_D \frac{d\phi(t)}{dt} \quad (3)$$

이때, K_D 는 주파수 판별기의 이득 상수이다. 주파수 변

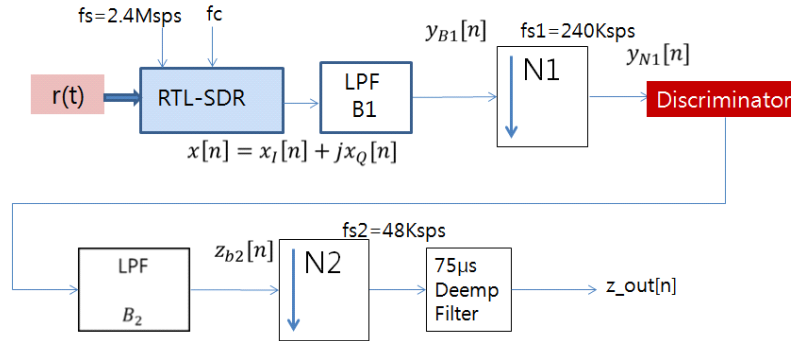


그림 3. 주파수 복조기 블록 다이어그램
Fig. 3. FM demodulator block diagram

조 신호에서 $\phi(t) = 2\pi f_d \int^t m(\tau) d\tau$ 이므로,

$$y_D = K_D \cdot f_D \cdot m(t) \quad (4)$$

이때, K_D 의 단위는 v/Hz , f_D 의 단위는 Hz/v , $m(t)$ 의 단위는 v (voltage)이다.

주파수 변조 신호를 복조하기 위하여 이산신호 구현에 편리한 복소수 기저대역 주파수 판별기 다른 용어로는 사상관기 (四相關機, quadri-correlator)를 이용한다. $x_c(t)$ 신호의 복소수 기저대역 신호 $\tilde{x}_c(t)$ 는 다음과 같다.

$$\begin{aligned} \tilde{x}_c(t) &= \cos [2\pi\Delta ft + \phi(t)] + j \sin [2\pi\Delta ft + \phi(t)] \\ &= \cos \theta(t) + j \sin \theta(t) = x_I(t) + jx_Q(t) \end{aligned} \quad (5)$$

이때, 작은 주파수 에러 Δf 가 있다고 가정한다. 그리고 주파수 판별기는 위 식에서 $d\theta(t)/dt$ 값을 구하는 것이므로, 다음의 $\theta(t)$ 에서

$$\theta(t) = \tan^{-1} \frac{x_Q(t)}{x_I(t)} \quad (6)$$

$\theta(t)$ 를 미분하여 구한다.

$$\frac{d\theta(t)}{dt} = \frac{x_I(t)x'_Q(t) - x'_I(t)x_Q(t)}{x_I^2(t) + x_Q^2(t)} \quad (7)$$

이산 신호 처리에서는 $x_I(t)$ 신호를 $x_I(nT) = x_I[n]$ 신호로, $x_Q(t)$ 신호를 $x_Q(nT) = x_Q[n]$ 신호로 바꾸어 처리한다. 이때, $1/T = f_s$ 값은 샘플링 율이다. $x_I(t)$ 및 $x_Q(t)$ 신호의 미분 값은 후향 차분 신호인 $x_I[n] - x_I[n-1]$ 및 $x_Q[n] - x_Q[n-1]$ 신호로 근사화하여 구현할 수 있다. 파이선 프로그램으로 작성한 기저대역 주파수 판별기는 아래와 같다.

```
import numpy as np
import scipy.signal as signal
def discrim(x):
    """
    disdata = discrim(x)
    where x is an angle modulated signal
    in complex baseband form.
    """
    X=np.real(x)
    Y=np.imag(x)
    b=np.array([1, -1])
    a=np.array([1, 0])
    derY=signal.lfilter(b,a,Y)
    derX=signal.lfilter(b,a,X)
    temp=(X*derY-Y*derX)
    disdata=temp/(X**2+Y**2)
    return disdata
```

매트랩으로 동일한 함수를 작성하면 다음과 같다.

```
function disdata = discrim(x)
% disdata = discrim(x)
% where x is an angle modulated signal in complex
% baseband form.
%
% Baseband discriminator
%
X = real(x); % X is the real part of the received signal
Y = imag(x); % Y is the imaginary part of the received
% signal
b = [1 -1]; % filter coefficient for discrete derivative
a = [1 0]; % filter coefficient for discrete derivative
derY = filter(b, a, Y); % derivative of Y
derX = filter(b, a, X); % derivative of X
disdata = (X .* derY - derX .* Y) ./ (X.^2 + Y.^2);
```

IV. 매트랩 및 파이선의 복조기 알고리즘

실시간 복조기에 사용되는 기능 블록들을 매트랩 및 파

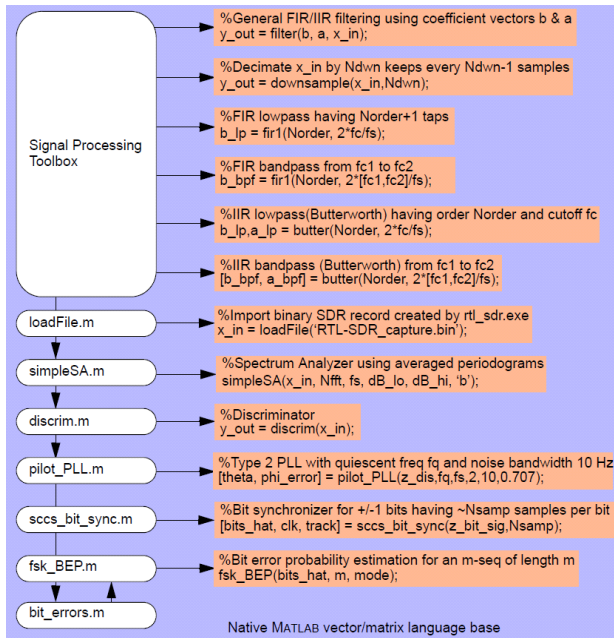


그림 4. 복조기에 사용되는 매트랩 함수
 Fig. 4. Matlab functions used to develop demodulator

이선 소프트웨어로 구현하기 위한 방법을 기술한다. 매트랩은 기본적으로 벡터 및 행렬 계산을 용이하게 하는 소프트웨어이다. 소프트웨어 라디오는 이산 신호 처리 (digital signal processing, DSP) 기법을 기반으로 하므로 매트랩의 기본 기능들 뿐만 아니라 DSP 툴박스 (Toolbox™)를 이용해야 한다⁶⁾. 3장에서 언급한 RTL-SDR에서 획득한 I 및 Q 신호를 복소수 벡터로 변환시키는 함수를 이용하여 매트랩에서 복조기를 구현한다. 복조된 오디오 신호를 컴퓨터에서 실행시키기 위해서 매트랩의 sound(z, Fs) 함수를 이용한다. 이때 Fs는 이산 오디오 신호의 샘플링 주파수이고, z 변수는 1행 또는 2행의 행렬의 이산 오디오 신호이다. 1행 행렬의 경우는 모노 오디오이고 2행 행렬의 경우는 스테레오 오디오로 실행된다. 오디오 신호의 왜곡을 방지하기 위해서 이산 오디오 신호가 -1에서 +1값을 가지도록 한다. 이와 같이 정규화하기 위해서 매트랩 함수인 max(abs(z)) 명령어를 이용할 수 있다. 그림 4에 함수들을 정리한다.

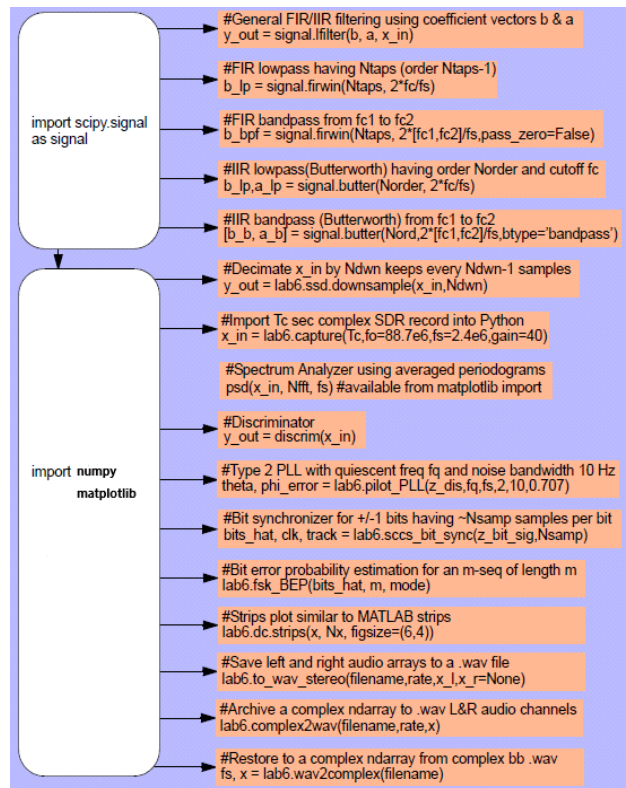


그림 5. 복조기에 사용되는 파이선 함수
 Fig. 5. Python functions used to develop demodulator

최근 광범위하게 사용되는 파이선과 같은 범용 소프트웨어로 복조기를 구현한다. IPython, Jupyter Notebook 등을 설치하기 위해 Anaconda 혹은 Enthought Canopy 등을 설치하여 개발 환경을 구축한다. 본인의 컴퓨터의 성능이 좋지 않을 경우에는 IBM의 Cognitive class Lab의 서버를 이용할 수 있다. 파이선에서 벡터 행렬의 연산을 수행하기 위해서 라이브러리들을 이용한다. 파이선에서 매트랩의 툴박스와 유사한 개발 환경을 만들기 위해서 행렬 계산을 위한 라이브러리 numpy, 2차원 (2 dimension, 2D) 플롯 라이브러리 matplotlib를 import한다. 이산 신호처리 라이브러리 scipy.signal (scientific Python)을 import하여 저역 통과 필터, 대역 통과 필터 등을 그림 5와 같이 구현할 수 있다.

지금까지 설명된 복조기 알고리즘을 이용하여 소프트웨어로 복조기를 구현한다. 샘플링 율 $f_s = 2.4$ Msps, $N_1 = 10$, $N_2 = 5$ 등의 값으로 정한다. 이러한 값으로 중간 샘플링 율은 $f_{s_1} = 240$ Ksps 그리고 최종 샘플링 율은 $f_{s_2} = 48$ Ksps가 된다. 상기 파라미터를 적용하여 주파수 판별기, 다운샘플러, 디앰퍼시스 필터 출력을 얻을 수 있다. 디앰퍼시스는 시정수 (time constant) 값이 $75 \mu s$ 인 단일 폴 (single pole) 필터로 설계한다. 모노 수신기에 대한 기존 연구를 포함하여 스테레오 수신기 구현을 상세히 설명한다^[10].

(가) 우선 원하는 주파수 변조 방송 신호를 약 5초에서 10초 정도를 본 논문 2장 후반에 설명한 바와 같이 저장한다. 예를 들면 중심 주파수가 93.1 MHz이면 클래식 주파수 방송이다. 라디오 주파수 이득은 25 정도로 한다. 신호 레벨이 낮을 경우에는 이득을 높여도 된다. 너무 높으면 아날로그 신호 혹은 자동 이득 조정 회로 출력이 왜곡이 생길 수 있으므로 적절한 값을 찾아야 한다. 필요하다면 윈도우 응용 프로그램인 SDRSharp을 실행시켜 신호를 관찰할 수 있다.

(나) 저장된 복조수의 라디오 주파수 신호를 컴퓨터에서 주파수 영역 신호로 변환하여 관찰한다. 방송 주파수의 저대역 신호의 대역폭이 약 100 KHz임을 확인한다. 인접 FM 방송 신호가 있다면 최소 300 KHz 떨어져 있으므로 통과대역 신호의 대역폭은 ± 100 KHz 이기 때문이다.

(다) 그림 6의 블록도에서 컷오프 주파수 B_1 을 결정한다. 수신하는 방송국 신호 이외 타 방송 신호, 다른 주파수의 간섭신호와 잡음신호를 제거하기 위함이다. 이 저역 통

과 필터는 윈도우를 이용한 유한 임펄스 응답 필터 혹은 무한 임펄스 응답 필터로 구성할 수 있다. 유한 임펄스 응답 필터는 63차 이하 그리고 무한 임펄스 응답 필터는 6차 이하로 하면 충분히 다른 방송 신호를 제거할 수 있다.

(라) 첫 번째 저역 통과 대역 필터를 통과한 신호 $y_{B1}[n]$ 에서 샘플들을 N_1 마다 거르고 $y_{N1}[n]$ 을 얻는다.

(마) 주파수 판별기를 통과하고 $z_{dis}[n]$ 신호를 얻는다. 이 신호를 주파수 영역에서 플롯하면 그림 7과 같은 주파수 변조 방송 신호를 복조한 신호를 볼 수 있다. 주파수 영역에서 약 15 KHz 까지는 좌우 오디오 신호를 더한 신호를 볼 수 있고, 19 KHz에는 파일럿 신호가 있고, 다음에는 왼쪽 오디오 신호에서 오른쪽 오디오 신호를 뺀 신호가 있다. 마지막으로 57 KHz에는 대역폭이 약 4-5 KHz인 라디오 방송 데이터 서비스 (radio broadcast data service, RBDS) 신호가 있다.

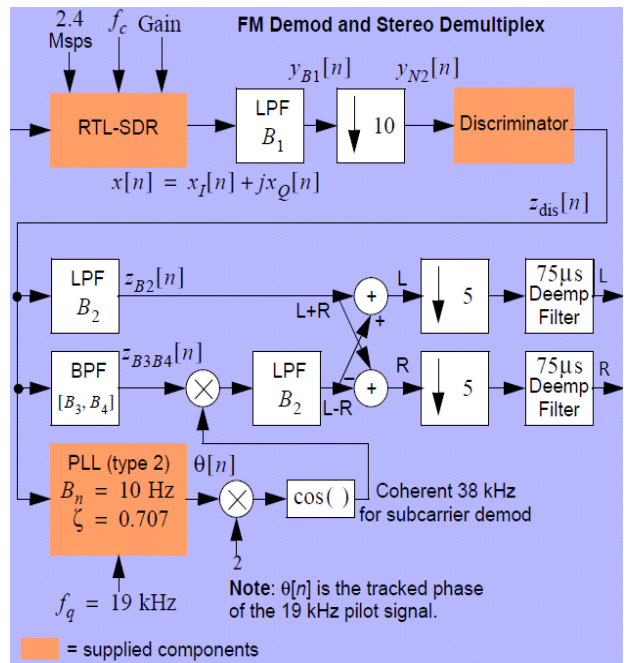


그림 6. 소프트웨어 라디오의 스테레오 디멀티플렉싱 블록도
Fig. 6. Stereo demultiplexing block diagram of software radio

(바) 다음에는 두 번째 저역 통과 대역 필터에서 좌우 오디오 신호를 더한 모노 신호만을 통과시킨다. 이 필터는 30

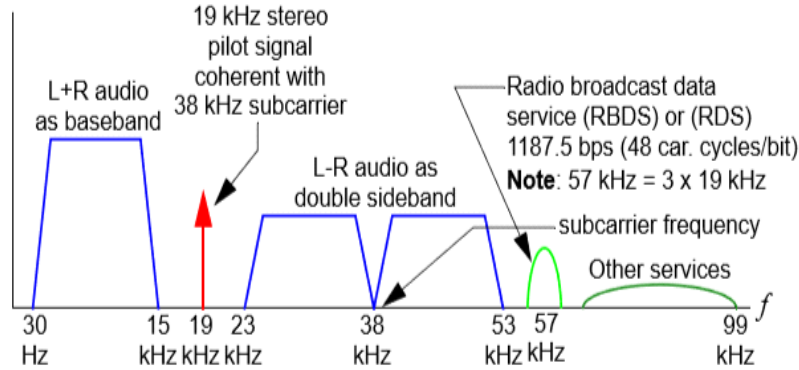


그림 7. 복조된 주파수 변조 방송 신호의 주파수 스펙트럼
 Fig. 7. The demodulated broadcast FM signal's frequency spectrum

Hz 이상 15 KHz 이하 신호를 통과시키는 통과 대역 필터가 아남에 주의한다. 약 15 KHz 이하의 신호를 통과시키고 $N_2 = 5$ 마다 샘플을 거르고 48 Ksps 신호를 얻는다. 이 신호를 .wav 파일 이름으로 저장하고 `sound(z, fs2)` 명령을 실행하면 수신한 방송 신호를 들을 수 있다. 아직은 디엠퍼시스 (deemphasis) 기능을 넣지 않았기 때문에 가청 주파수 대역에서 고주파 영역이 강조되어 들린다.

(사) 스테레오 오디오의 좌신호에서 우신호를 뺀 L-R 부반송파는 38KHz이다. 위상 고정 루프에서 부반송파를 복구한 후 대역 통과 필터에서 나오는 L-R 신호에 믹싱을 하여 양측파대 복조기를 구현한다. 이후 대역폭이 B2인 저역 통과 필터 출력에서 L-R 신호를 복조한다.

(아) 디엠퍼시스는 단일 폴 (single pole) 무한 임펄스 응답 (infinite impulse response) 저역 통과 필터로 구현하며 스테레오 오디오의 좌우 신호에 각각 적용한다. 아날로그 단일 폴 저역 통과 필터의 임펄스 응답은 다음과 같다^[5].

$$h(t) = \frac{1}{\tau} e^{-t/\tau} u(t) \quad (8)$$

이때, $\tau = RC$ 는 시정수이다. 주파수 변조 방송의 경우 $\tau = 75\mu s$ 로 정하고 있으므로 본 논문의 디엠퍼시스는 동일한 값으로 정한다. 디지털 필터의 임펄스 응답은 아래와 같다.

$$h[n] = h(nT) = e^{-nT/\tau} u[n] = (e^{-T/\tau})^n u[n] \quad (9)$$

이때, $a_1 = e^{-T/\tau} = e^{-2\pi f_s/f_c}$ 는 필터의 피드백 계수이고 $f_3 = 1/(2\pi\tau)$ 는 RC 저역 통과 필터의 진폭 값이 직류일 경우보다 -3dB되는 주파수이다. $x[n]$ 의 이득 계수는 $(1 - a_1)$ 으로 하여 필터의 직류에서의 이득이 1이 되도록 한다. 매트랩에서 디엠퍼시스를 구현하면 $a = [1, -a_1]$, $b = [1 - a_1]$ 으로 하고, $y = \text{filter}(b, a, x)$ 라고 한다. 파이선으로 구현할 경우에는 $y = \text{ifilter}(b, a, x)$ 를 실행시킨다. 디엠퍼시스를 동작시키고 소리를 들으면 고주파수 영역의 신호가 감쇄되어 들리게 된다.

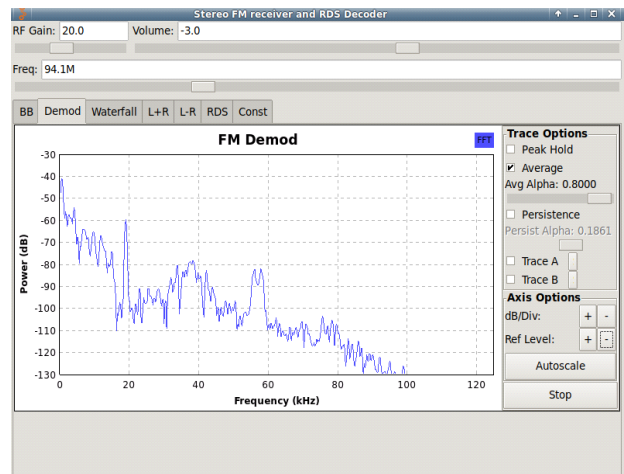


그림 8. 실시간으로 측정된 주파수 변조 방송의 주파수 스펙트럼
 Fig. 8. Frequency spectrum of real-time demodulated frequency demodulated broadcast signal

V. 그누라디오 (GNR Radio)를 이용한 실시간 복조기 구현

본 장에서는 실시간으로 복조기를 구현하기 위해 그누라디오를 이용한다⁷⁾. 그누라디오는 파이선과 C++로 작성된 강력한 이산 신호처리 응용 프로그램으로 리눅스 환경에서 개발되었으나 윈도우 버전으로도 제공된다. 매트랩의 시뮬링크와 유사하게 그래픽 유저 인터페이스 (GUI) 기반으로 원하는 신호의 주파수 대역에 맞는 안테나와 신호 처리 연산에 적합한 RTL-SDR과 같은 하드웨어만으로 다양한 통신 시스템을 구현할 수 있다⁸⁾⁹⁾. 매트랩 및 시뮬링크와 달리 무료로 배포되는 장점도 있다. 제 4장에서 검증된 복조기의 기능 블록들을 그누라디오로 구현함으로써 실시간으로 신호 처리를 하여 원하는 대역에 동조하고 방송 신호를 수신할 수 있다. 그누라디오의 옵션 (Options) 블록은 작성하는 소프트웨어 라디오의 아이디와 라디오 신호를 검증하는 계측기 기능을 하는 그래픽 유저 인터페이스의 여러 종류 중에서 하나를 선택할 수 있다. 기본 값은 각각 top_block과 QT GUI이다. 기본적인 변수로 샘플율을 의미하는 samp_rate을 정한다. RTL-SDR의 경우는 최대 샘플링율이 3.2 Msps이다. 그러나 보통 2.4Msps이하에서 수신 신호의 대역폭에 맞게 결정한다. 본 논문에서는 1MHz의 샘플링율을

사용한다. 수신에 필요한 주파수는, 주파수 오프셋, RF 이득 등의 변수들도 정의한다. 슬라이드 버튼을 이용하여 실시간 방송을 수신하면서 오디오 볼륨, RF 이득, 중심 주파수 등을 가변할 수 있도록 한다. 수신기의 주요 부분 즉, 주파수 판별기에서 복조된 신호, 기저대역 신호, 스테레오 신호, 디지털 정보 서비스 (radio digital service, RDS) 신호의 성상도 등을 마치 계측기를 가지고 실험하는 것과 같이 스코프 (scope)를 보이는 sink 블록들도 구현한다. 주파수 복조기는 RTL-SDR Source부터 시작한다. 주파수 판별기를 거치면 그림 8과 같은 주파수 방송 신호가 복조된 신호를 얻을 수 있다. 실시간으로 관찰된 신호는 그림 8과 같다. 그림 9는 그누라디오로 구현된 전체의 복조기이다. 그누라디오는 실제의 방송신호를 실시간으로 수신할 수 있으므로 가장 먼저 SDR 하드웨어에 적합한 샘플링율을 정한다. 본 논문에서는 RTL-SDR에서 최대로 구현할 수 있는 2.4Msps에 근접하는 2.0Msps를 적용한다. 옵션블록에서는 아이디를 입력하고 발생 옵션을 정한다. 그누라디오를 실행시키면 아이디 이름과 동일한 파이선 프로그램이 생성된다. 필터 블록, 광대역 주파수 복조 블록, 디엠퍼시스 블록, PSK 복조 블록, 오디오 블록 등의 파라미터들을 설정하고 서로 연결시키면서 복조기를 완성한다. 복조기의 신호들을 컴퓨터에서 볼 수 있도록 여러 계측기 블록을 활용한다. 시

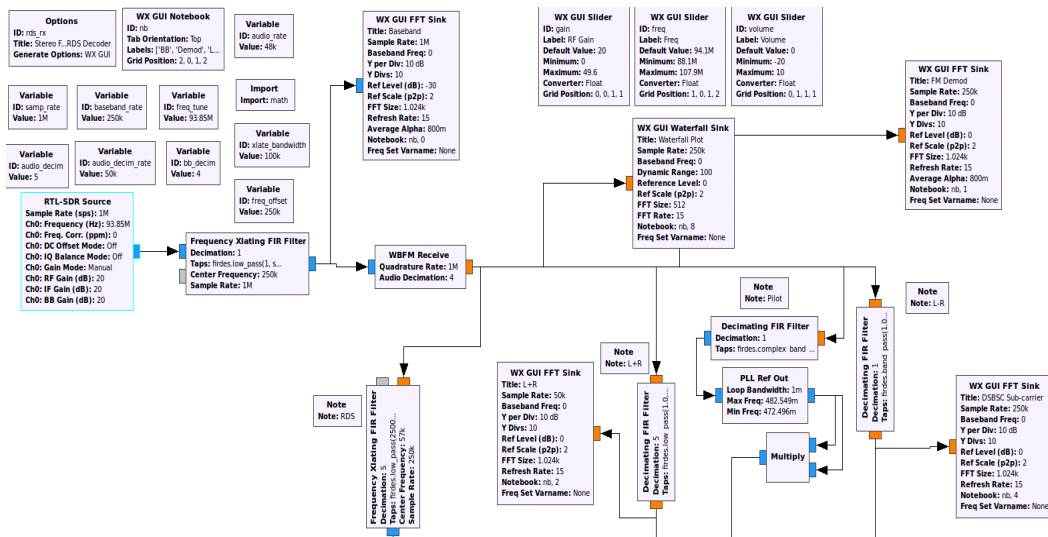


그림 9. 그누라디오로 구현된 주파수 복조기 상단
Fig. 9. Frequency demodulator upper part implemented by GNU Radio

간 축의 신호를 볼 수 있는 Time Sink 및 주파수 스펙트럼을 볼 수 있는 Frequency Sink를 이용한다. 그림 10과 같이 위상 고정 루프를 이용하여 파일럿 신호의 위상을 고정하여 스테레오 수신 및 디지털 데이터 수신에 활용한다. 그림 11에는 주파수 변조 방송의 디지털 데이터를 복조하여 성상도를 볼 수 있는 블록을 이용하여 클래식 FM 정보를 수신한다.

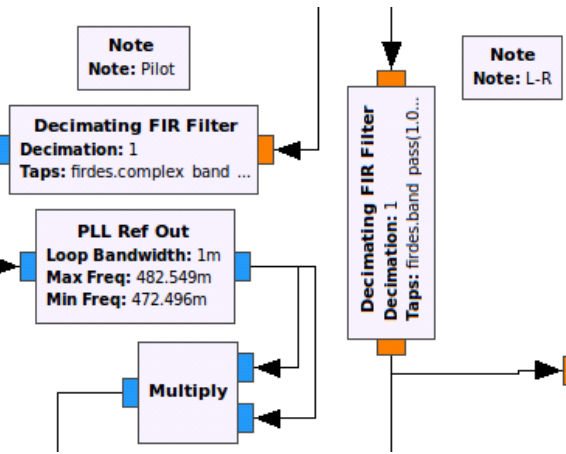


그림 10. 그누라디오로 구현된 위상고정루프
 Fig. 10. Phase locked loop part implemented by GNU Radio

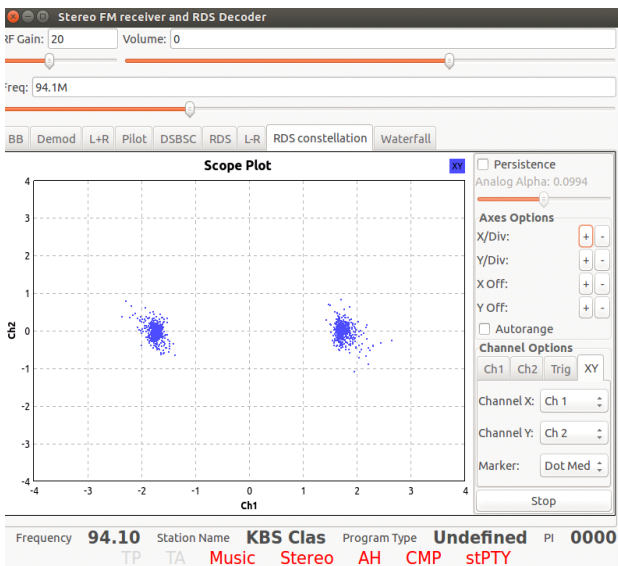


그림 11. 그누라디오로 구현된 디지털 데이터 성상도
 Fig. 11. Constellation of digital data implemented by GNU Radio

VI. 결론

주파수 변조 방송 신호를 미화 20달러 내외의 소프트웨어 라디오 및 그누라디오를 이용하여 실시간 수신하는 과정을 기술하고 복조 알고리즘을 소프트웨어로 구현하였다. 아날로그 소자인 튜너에서 원하는 주파수 신호를 기저대역으로 변환시키고 디지털 소자를 이용하여 아날로그 신호를 디지털로 변환시킨 후에는 컴퓨터를 이용하여 모든 복조 알고리즘을 구현하였다. 저역 통과 필터, 통과 대역 필터, 주파수 판별기를 거치면 좌우 오디오 신호, 19 KHz 파일럿 신호, 38 KHz에 진폭 변조된 스테레오 오디오 신호, 57 KHz에 변조된 방송 데이터 신호 (RBDS) 등으로 분리됨을 매트랩, 파이썬, 그리고 그누라디오를 이용하여 증명하였다. 어느 방법을 이용하든 이상적인 주파수 변조 방송 스펙트럼과 동일함을 알 수 있다. 특히 그누라디오라는 공개 소프트웨어를 이용하면 매트랩의 시뮬링크와 같이 기존에 작성된 여러 기능 블록들을 이용하여 보다 빠르게 수신기를 구현할 수 있고, 실시간 복조를 구현할 수 있다.

참고 문헌 (References)

- [1] J. Mitola, Software radios survey, critical evaluation and future directions. National Telesystems Conference, pp. 13-23, 1992.
- [2] T. Collins, R. Getz, D. Pu, A. Yuglinkski, Software Defined Radio for Engineers, Artecj Jpise. 2018.
- [3] <https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr>
- [4] <http://www.rtl-sdr.com/an-intro-to-rtl-sdr-technical-dsp-concepts-explained/>
- [5] A. Oppenheim, R. Schaffer, Discrete-time Signal Processing, Prentice Hall, 1989.
- [6] <http://www.eas.uccs.edu/~mwickert/>
- [7] <http://wiki.gnuradio.org>
- [8] <https://wiki.gnuradio.org/index.php/Tutorials>
- [9] http://files.ettus.com/tutorials/labs/Lab_1-5.pdf
- [10] Y. Kim, "Implementation of real-time FM de-modulator using software-defined radio," Journal of the Research Institute for Computer and Information Communication, Vol.26, No.1, pp.1-4, 2018.

저 자 소 개



김 영 주

- 1988년 ~ 1993년 : LG전자
- 1996년 ~ 1997년 : 동경공업대학 연구원
- 1994년 ~ 2001년 : 한국과학기술원 석박사
- 2011년 ~ 2012년 : 퍼듀대학 교환교수
- 2003년 ~ 현재 : 충북대학교 전자정보대학 정보통신공학부 교수
- ORCID : <https://orcid.org/0000-0002-5844-8612>
- 주관심분야 : 소프트웨어라디오, 이동통신