EE-342 Telecommunications Systems **Graded Lab**
Spring Semester
Telecommunications Circuits Laboratory
EPFL

Building an FM Stereo Receiver

In this lab, you will use everything you have learned in the previous practical sessions to build a functioning FM stereo receiver. You'll implement the receiver in GNU Radio and verify its well functioning using a recorded baseband RF signal as well as real signal received from a software-defined radio.

Organization

This lab is graded and should be done in groups of two (or three with some additional expectations).
You have to submit your code and a short report that answers the questions asked in this documents and provides screenshots of the requested signals. Please take screenshots using the **prtSc** key on the keyboard and not pictures of the screen!
One person per group should upload on Moodle one zip-file containing:

1. The report in PDF format.
2. The GNU Radio flowgraph of the mono FM receiver.
3. The GNU Radio flowgraph of the stereo FM receiver.

1 Introduction to FM Modulation

The FM modulation is a type of analog modulation that is used to transmit information over a carrier wave. The frequency of the carrier varies proportionally to the amplitude of the modulating signal, as illustrated in Fig. 1.1. The FM modulation can be mathematically described as

$$y(t) = \cos((2\pi f_c + k_f x(t))t), \tag{1.1}$$

where $y(t)$ is the modulated signal, f_c is the carrier frequency, and k_f is the modulation index. We can make a few observations from (1.1). First, to recover the original signal $x(t)$, we need to extract the frequency variations around our carrier frequency f_c. Second, the bandwidth used by the FM signal depends on the magnitude of $k_f \max(x(t))$. And third, the power of the FM signal is constant regardless of the modulated signal.

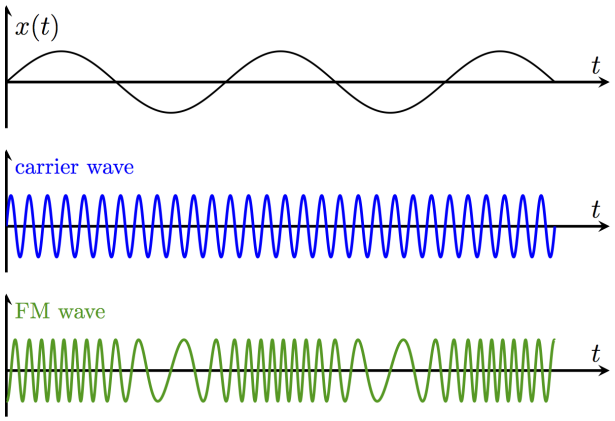

Figure 1.1: Illustration of FM modulation of a sine wave

Current FM radio stations do not only broadcast a simple mono audio signal, but a stereo one. To do so, they create two audio signals. One is the sum of the left and right audio channels (L+R), and the other is the difference between the left and right audio channels (L−R). The L−R channel is then up-converted to a higher frequency (i.e. $38\,\mathrm{kHz}$) and added to the L+R signal which creates a new signal which spectrum is shown on Fig. 1.2. In addition, a single tone at $19\,\mathrm{kHz}$ is also added to help create a reference $38\,\mathrm{kHz}$ carrier to down-convert the L−R signal at the receiver.
The resulting signal is used to modulate the carrier frequency. Typically, the passband bandwidth broadcasted by an FM station is around $200\,\mathrm{kHz}$.

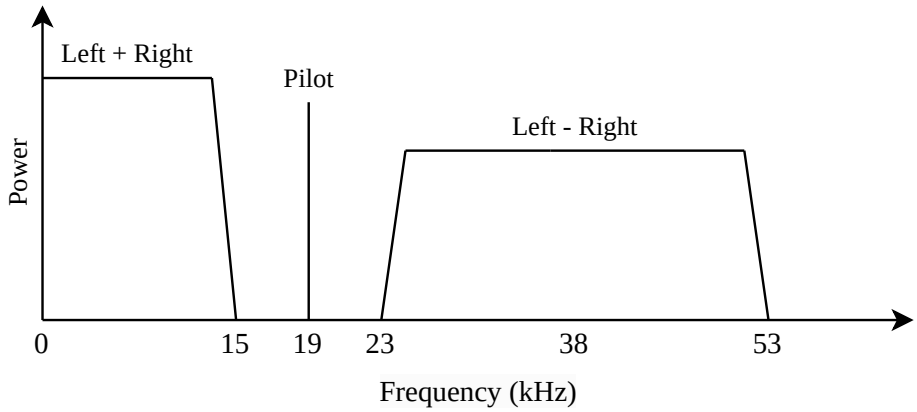

Figure 1.2: Spectrum of an FM station broadcast

2 Starting Point

You can find a template flowgraph on Moodle that you should use as a starting point.
The flowgraph already contains a few blocks:

- A *file source* block that reads a recorded signal from a file. The file contains complex baseband samples, which means that the carrier frequency has already been removed by the SDR that recorded the signal.

- A *Soapy RTLSDR Source* and *Rational Resampler* block that you can use to receive a real signal from a software-defined radio. Those block are disabled as they are only required if you are using a radio. If they do not appear in your flowgraph, press CTRL+D to show disabled blocks.
- An *audio sink* block that will play the recovered audio signal. It has two inputs, corresponding to the left and right audio output.
- Two *Rail* block that will clip the signal to the range $[-1, 1]$ prior to forwarding it to the audio sink. The clipping operation avoid to saturate the audio output.
- A *QT GUI Time Sink* block that you can use to display recovered audio signals in time domain.
- A *QT GUI Frequency Sink* that you can use to display the spectrum of the signal at different key points in the flowgraph.

Throughout this lab, you will need to filter the signal at different points in the flowgraph. For convenience, we have already created the taps of the four different filters that you can use. You can use the generated taps to create a filter using the *Interpolating FIR Filter* block and by setting the *Taps* parameter to the ID of one of the four filter taps already created for you.

Feel free to duplicate the *Virtual Source* block to access the signal connected to the corresponding *Virtual Sink* to keep your flowgraph clean and readable.

3 FM Demodulation

Your first task is to demodulate the FM signal, i.e estimate the instantaneous frequency of the signal. As you should know, the angular frequency corresponds to the derivative of the phase with respect to time as

$$\omega(t) = \frac{d\phi(t)}{dt}. \tag{3.1}$$

Therefore, a simple way to estimate the instantaneous frequency of a digital signal is to compute numerical derivate of the phase between two consecutive samples as

$$\omega[n] = \angle \left(y[n] \cdot y^*[n-1] \right), \tag{3.2}$$

where $\cdot^*$ denotes the complex conjugate, and $\angle(\cdot)$ is the argument of the complex number.

3.1 FM Demodulation

Implement the FM demodulation using (3.2) and feed to output to the *Virtual Sink* with ID "demod_signal".
Provide screenshots of the flowgraph you built and the spectrum of the demodulated signal.
On the screenshot of the spectrum, annotate the different parts shown in Fig. 1.2.
Are the frequencies of the different parts as expected?

On the Figure 1.2, why does the L-R signal appears to use a bandwidth twice as large as the L+R signal?

4 Mono Audio Recovery

You will now recover the mono audio signal from the demodulated signal and listen to it.
To listen to the audio signal, you can either use a speaker received from the TAs or a pair of personal headphones.

4.1 Mono Signal Extraction

Filter the demodulated signal to keep only the L+R channel and connect the resulting signal to the *Virtual Sink* block "L+R_channel".
Provide a screenshot of the spectrum of the L+R channel.
Which of the provided filter taps did you use and why?

4.2 Resampling

The original sampling rate of the FM signal is 250 kHz, however, the audio signal has to be resampled to 48 kHz to be usable by the audio driver of the computer.
Resample the signal to 48 kHz and connect it to the clipping rails.
You should be able to listen to the audio signal.
Provide a screenshot of the time domain Audio signal and explain how you resampled the signal.

4.3 Clipping Prevention

The audio signal magnitude is sometime too high (> 1) and saturates the audio output.
Add an operation before the clipping rails to fix this issue.
Provide a screenshot of your flowgraph and explain what you added to fix the problem.

4.4 FM Station Reception

If you want to use your personal computer, you may have to install the adapted drivers for the RTL-SDR v4. You can find the instructions on the RTL-SDR website here.

To verify if your receiver is able to receive a real FM station, enable the *Soapy RTLSDR Source* block and the *Rational Resampler* block by selecting and pressing E. Disable the *File Source* as you will receive real world signals by selecting it and pressing D.

By changing the center frequency of the receiver, are you able to listen to a radio station? If yes, what is the station frequency?

Provide a screenshot of the audio wave and spectrum.

5 Stereo Audio Recovery

BEFORE CONTINUING: Save a copy of your mono FM receiver flowgraph as a separate file that has to be uploaded alongside you report.
You will now recover the stereo audio signal from the demodulated signal.
Re-enable the *File Source* block and disable the *Soapy RTLSDR Source* and *Rational Resampler* block.
To recover the left and right audio channels, you will need to down-convert the L−R signal back to baseband. Once you have both the L+R and L−R in baseband, you can either add or subtract them to recover the left and right audio channels individually.

5.1 Pilot Reconstruction

The pilot signal can be used to create the reference carrier to down-convert the L−R signal. Since we have a 19 kHz tone in the signal but require a 38 kHz one for the down-conversion, we need to double its frequency. This can be achieved in three steps:

1. Filter the signal to only keep the pilot.
2. Double the frequency of the pilot signal.
3. Filter again the signal to remove any images created by the operations used to double the pilot frequency to keep only the 38 kHz tone.

Explain which method you used to double the frequency of the pilot (hint: trigonometric identities might be useful).
Connect the resulting signal to the *Virtual Sink* with ID "38kHz_ref" and provide a screenshot of the part of your flowgraph handling the pilot reconstruction. In addition, provide a screenshot of the spectrum of the reconstructed pilot.

5.2 L-R Signal Down-conversion

Down-convert the L−R signal to baseband using the 38 kHz reference extracted in the previous task and keep only the signal of interest.
Connect the resulting signal to the *Virtual Sink* with ID "L-R_channel" and provide a screenshot of the spectrum of the signal.
How many filters do you need and explain which ones you used and why?
Why are we using the pilot to down-convert the L-R signal instead of generating a local 38 kHz cosine?

5.3 Left and Right Channels Extraction

Combine the L+R and L−R signals to recover the left and right audio channels.
Finally, connect both channels to the audio sink after resampling (Similarly to what you did for the tasks 4.2 and 4.3).
Provide a screenshot of the time domain of the left and right audio channels and the flowgraph part handling the combining and resampling.
Sketch the demodulated spectrum that we would receive if the radio station was broadcasting a song recorded in mono.

5.4 SDR Reception

You can now try to receive a real FM station in stereo using the SDR.
Similarly to 4.4, enable the *Soapy RTLSDR Source* and the *Rational Resampler* block and disable the *File Source*.

Can you still receive the same FM station? If yes, provide a screenshot of the audio signal and spectrum.

6 Extra Task for Groups of Three

If you are a group of three students, you have to implement a second FM demodulation method. Go back to your mono FM receiver and replace the FM demodulation based on the instantaneous derivative with **one** of the following methods:

- Zero-Crossing Demodulation
- Arctan-less Demodulation

The block diagram and intermediate signals of a zero-crossing detector are described in the patent available on Moodle, with some extra annotations specific to our case.
A demodulation method which does not rely on an Arctan computation is described in the section 13.22 of the book *Understanding Digital Signal Processing*, available on moodle.
Provide a screenshot of the flowgraph you built and the spectrum of the demodulated signal for the demodulation method you chose.