Exercise 2
Decimation and FM receiver

Aalto University
School of Electrical
Engineering

ELEC-E5410 Signal processing for communications

General Guidelines

- **min 50% points of the exercises required in the scale 0-100%**
- **Groups of 1-2 persons**
  - Sign up in MyCourses
- **If several groups return the same code without declaring cooperation (or similar code, changing names of the variables etc. doesn't count), the max. number of points/group becomes 100%/ #similar submissions**
- **Approved Exercise 1 required to be able to get the RTL-SDR**
  - You can still use your own hardware if you like

General Guidelines

- **Download Matlab from download.aalto.fi**
- **Install RTL-SDR hardware support package from Add-Ons in the Matlab menu bar**
- **Return .m files**
- **Return the .pdf made by Matlab's publish() with the results**
  - Not .html + several .pngs or some other awkward format

Known issues in installation

- **Admin rights needed to install driver**
- **On Windows, installation of the toolbox changes the rights of Matlab to admin and the path to toolbox is hidden from regular users**
- **(Always use the same USB port for RTL-SDR on Windows)**
- **Kernel drive active on Linux**
  - sudo echo “blacklist dvb_usb_rtl28xxu” >/etc/modprobe.d/rtlsdr.conf and reboot
- **Mac requires Xcode, which takes 7GB of disk space**
  - Install command line tools (130MB) only: osxdaily.com/2014/02/12/install-command-line-tools-mac-os-x/

Some Matlab tips

- **Vectorize**
  - If **x** is a row vector and **y** is a column vector, do **x***y instead of "sum=0; for ii = 1:N, sum=sum+ x(ii)*y(ii); end"
  - For point-wise multiplication do **y**.***x** instead of "for ii = 1:N, x(ii)*y(ii); end"
- **Watch vector dimensions**
  - If **x** is a column vector and **y** is a row vector **x***y is an NxN matrix instead of a scalar. A vector **x** is forced to a column vector by x(:)

Matlab tips

- **All functions can be included in one file as local functions**
- **These local functions are not visible outside the file**
- **The functions can be made visible outside by defining anonymous functions in the same file. This is the technique used in rw_ini… files**
  - p_fun() is visible outside the file, fun() is not

```
p_fun = @(x) fun(x);
function y = fun(x)
  ..
end
```

- **The parameters inside the anonymous function freeze their values when the function is defined**
  - y=1; foo = @(x) x+y; y = 2; foo(1) = 2
  - So if fun() is changed p_fun() remain unchanged unless the definition of p_fun() is executed again. Therefore, this kind of coding style using anonymous functions is not recommended.

Tasks

Tasks

1. **Download the Matlab script template fm_rx.m and the p-code file rw_ini_fm.p from Aalto Git**
   - The p-code file is an obfuscated file and the task is to implement your own functions instead of calling rw_ini_fm.p. The p-code is provided so that the script works even without
   - Functions/scripts starting by “rw_” are the ones you have to write yourself
   - Estimate your dongle’s carrier frequency offset (CFO) from the spectrum plot and compensate it approximately before low-pass filtering

Tasks

2. Design a low-pass FIR filter using the windowing technique with the following design criteria

- At least 30 dB attenuation at 19 KHz to mitigate the pilot signal
- As short filter as possible (for efficiency) such that the passband is still wide enough to listen to the radio.

Tasks

3. Implement the simple FM discriminator

- Stereo or de-emphasis filter are not necessary
- Subtract mean (after taking the angle) to remove residual CFO in your receiver

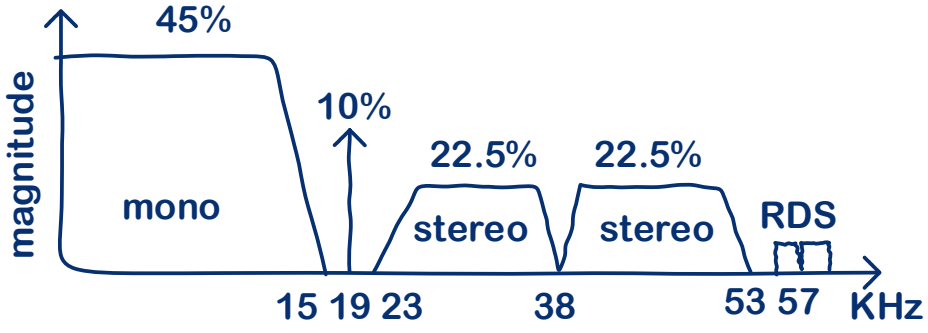
4. Filter, down-sample, demodulate FM (or filter, demodulate and down-sample) and listen to the output using audioWriteDevice(). Determine the front-end sampling rate and the decimation factor such that the output sounds reasonable

Report

- **Return your Matlab code (.m) and a pdf made by Matlab's publish function**
- **The document must contain the algorithms and the parameters**
- **In addition, describe the following:**
  - Which combination of the signal processing blocks (filter, decimation, discriminator) is the best and why
- **Deadline in the end of Period I**

Appendix

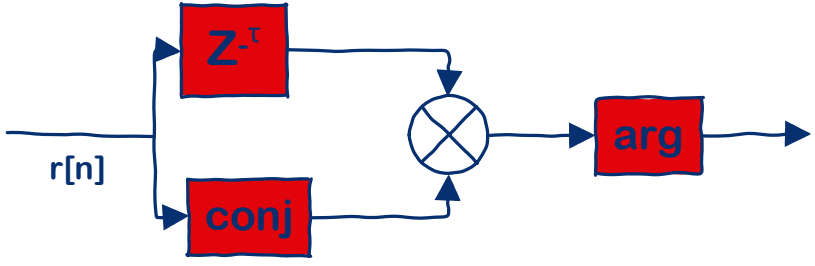
FM signal


45%
magnitude
mono
10%
22.5%
22.5%
stereo
stereo
RDS
15 19 23
38
53 57
KHz

Detection by complex discriminator

- The demodulated signal before analog-to-digital (ADC) conversion

$$r(t) = \frac{|h|}{2} \exp\left(j\psi + j\omega_\Delta t + j2\pi K \int_{-\infty}^{t} s_i(t)dt\right)$$



- First-order differentiator in digital baseband

$$y[n] = r^*[n]r[n-\tau] = \frac{|h|^2}{4}\exp(-j(\omega_\Delta n + \theta[n] + j(\omega_\Delta[n-\tau] + \theta[n-\tau])$$

- When τ is small, angle(y[n]) is approximately
- Non-coherent receiver

$$\frac{d}{dt}\omega_\Delta t + \frac{d}{dt}\theta(t) = -(\omega_\Delta + 2\pi K s_i(t))$$

CFO term