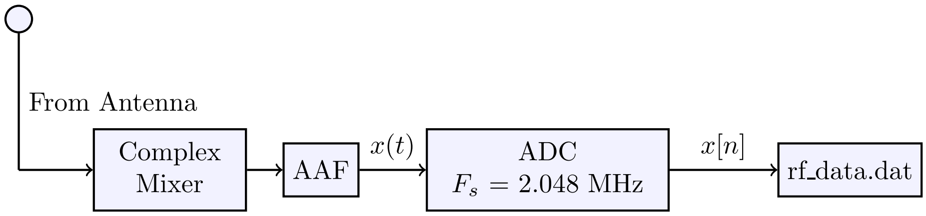
Complex Engineering Problem
An FM Demodulator

April 21, 2018

The lab assignment is due in the last week of the semester.

Task 1

You are given the output of the following system, which samples FM94.1 signal after downconversion to baseband:



File rf_data.dat is available on piazza. To read the file, you will need `loadFile.m` script which is also available on piazza.

RF signal received by your antenna passes through a complex mixer tuned to 94.1 MHz. The mixer downconverts the FM radio signal of bandwidth 200 kHz centered at 94.1 MHz to baseband. The downconverted signal passes through an antialiasing filter (AAF) and goes into a complex analog-to-digital converter (ADC) with sampling frequency $F_s = 2.048$ MHz. A complex ADC has two channels – one sampling the real part of the signal and the other sampling the imaginary one.

An FM signal is given by the following expression:

$$\phi_{FM}(t) = A(t)e^{j(2\pi f_c t + \int_{-\infty}^{t} m(\alpha)d\alpha)}.$$

Ideally $A(t)$ should be a constant, but the channel impairments make it a time-varying signal.

After downconversion, $f_c = 0$ and the signal $x(t)$ is given by:

$$x(t) = A(t)e^{j(\int_{-\infty}^{t} m(\alpha)d\alpha)}.$$

An FM demodulator task is to get message signal $m(t)$ out of this signal. We have sampled the signal $x(t)$ using a sampler of rate $F_s = 2.048MHz$. The task of a digital FM demodulator will be to reach $m[n]$, which is a sampled version of $m(t)$, from $x[n]$. Once we have $m[n]$, we can play it back to our speaker using digital-to-analog converter of a sound card.

FM demodulation consists of the following steps:

1. Normalize $x(t)$ that it has a constant amplitude $A(t)$ instead of a time-varying one.
2. If the sampling rate of 2.048 MHz is very large, decimate it down to a smaller rate. Note that the bandwidth of $m(t)$ is around 20 kHz for music signals.
3. We can get $m(t)$ by first differentiating the signal $a(t)$ and then multiplying it by its complex conjugate $a^*(t)$. For sampled signals, you can follow the same process of getting $m[n]$ from $a[n]$. You will need to design a discrete-time differentiator for this purpose.
4. Normailze $m[n]$ by its maximum value so that $m[n]$ values go from -1 to 1.
5. Decimate the rate of $m[n]$ to 16kHz so that we can play it back to the speakers of our sound card using `sound` command.

Questions

1. Draw a block diagram of your FM demodulator and implement it in MATLAB. Do you hear any voice?
2. What is approximate bandwidth of $x(t)$? Is $F_s = 2.048$ MHz adequate sampling frequency?
3. Draw frequency response (magnitude, phase and group delay) of your differentiator. Is your differentiator linear phase or not?
4. How many decimation stages did you use? Justify the positions of your decimators in your block diagram.
5. Describe which structure have you used to implement different filters and why?
6. How many multiplications and additions per second does your demodulator require?

You are required to design and implement all filters yourselves (no help from MATLAB filter design routines such as `fir1`, `butter`, etc. Implement all filters using least computational resources.