WIRELESS NETWORKING

BREATH DETECTION
Using a Doppler Radar Implementation on RTL- SDR

March 9, 2017

Daniël Petrarca
Charu Agrawal

INTRODUCTION

Radars exist in active and passive forms. An active Doppler radar consists of a transmitter that sends an EM signal at high frequency and at the same time one or more receivers receive, amongst other things, the reflected waveform from the object. We choose to design a radar using just 1 transmitting antenna and 1 receiving antenna.

Since we can't discern the directions from which reflections are coming, we need to make sure we don't receive too many auxiliary reflections, either by encasing our antenna making it directional, or by putting the radar close enough to the user so that the primary reflection has the largest magnitude.

Since the speed of EM waves is very large, there will only be a small frequency deviation that we must measure. Therefore, we need to use a high frequency transmitter (1Ghz or more) and it must be very narrow band. This rules out FM transmitters and other low frequency devices. A Wifi might be an option though. The minimum frequency of the transmitter can be calculated. Humans breath about max 32 times per minute, which is about 0.5 Hz. Children up to 60 times which is 1 hz.

$$Fdoppler = 2 * v * Ftransmitter/c \tag{1.1}$$

Where v is the relative speed between radar and reflected surface, max 1Hz*0.10m, and c is the speed of light (which is almost 3,00 *10^6).

1 THE ABSOLUTE MINIMUM TRANSMITTER FREQUENCY

Note that 1GHz is barely enough and we already need to do FFT with N=36120 plus more calculations including a FFT at least 10x per second and the laptop must handle this. Of course generating C code and compiling helps. Also GPU could do FFT up to some N.

The minimum frequency is given by:

10 updates/s minimum with device that does 2.4Msample/s - & gt; 240K samples per FFT window max frequency fastest breath (only children do this) = 120 per minute/ 60 = 2 Hz and the chest moves in the order of 5 cm. And the human breathes with max 2 Hz * 0,05 m (5 cm chest movement)

$$Fd = (2 * 2 * 0.05) * (ft/c) = 0,2 * ft/c$$
$$2.4M/Fresolution = Nfft$$

To discern 2 peaks we need at the very least 4 points per one sides of the baseband spectrum, where the spectrum ends at +- Fd, and one in the middle, so we need 10 points.

Fres = BW / 10 = 2*Fd / 10 = 0.2 * 0.2 ft/c = 0.04 ft/c
Nmin = Fsample / Fres = 2.4M / (0.04 ft/c) = 25*2.4M / (ft/c) = 60M / (ft/c) = 18 *(10^{16}) / ft
Nmin = (30M * c)/ ft = (60M * 300M)/ft = 18 MG / ft (Ft = 18 MG / N)

For example for 1Ghz we get:
Nmin = 18 M*G / 1 G = 18 M = 18 000 000 which is huge by the way And we get 2,4M / 18M = 0,13333333 updates per second

To visualize a quick breathing pattern, we either need to use a sliding window or we can only do.

However to see an animation rather than a frame that may or may not happen to taken sampled during the extrema of the breathing, we would say 10 frames gives us Pi/4 phases of the breathing, which is what we personally require to recognize is as breathing.

Note that requirement: Fupdate & gt; 10 Hz
Lets say we are designing for adults. Fbreath,max = 0.5 Hz
breathes with max 0.5 Hz * 0,05 m (5 cm chest movement)

$$Fd = (2 * 0.5 * 0.05) * (ft/c) = 0,05 * ft/c$$
$$Fres = 2 * Fd/10points = 0.05 * 0.2Fd = 0,01ft/c$$
$$For1Ghz; Fres = 0.08$$
$$Nmin = fs/fres = 2.4M/(0,01ft/c) = 2.4M/0.08 = 30M$$
$$Fd = 0.05Ft/c = 0.4Hz$$
$$Nmax = 2.4M/10updates/s = 24M$$

Since Nmin & gt; Nmax we shouldn't expect to reliably use 1Ghz for visualizing a quick breathing child. However this is still extremely choppy, and it requires tremendous amount of computing resources! Common DSP's prefer to take N = 4096.

$$Forft = 2.4GHz$$
$$Fd = 0.05ft/c = 0.05 * 8Hz = 0.4Hz$$
$$Fres = 2 * Fd/10 = 0.01ft/c = 0.08Hz$$
$$Nmin = 2.4M/Fres = 30M$$
$$10points * 10hz = Fs/Nmax; Nmax = Fs/100 = 24000$$

So it's still not possible

However if we are content with slightly less updates than we wanted but still above the absolute minimum, 2.4Ghz is our minimum frequency.

However this is only valid if there are no overlapping spectra. But we overlap with the transmitted signal's spectrum, because or transmitted signal is not necessarily a pure sine. For example for a sinc signal width primary lobe of BW 1/2 * Fd we need to double up to the 5Ghz range.

To be precise, if the band of the transmitter signal is wider than half the Fd we can't discern the signal anymore even at the extrema.

Let's see if we can **detect relaxed breathing of an adult** using 3.2 Msample/s (this mode can be set in SDR# but not in Matlab) using Wifi. A relaxed breathing pattern corresponds to 0.1Hz.

F slow breath = 12 per minute / 60 = 0.2 Hz

3 points in the [-Fd,+fd] range so we can see whether its In, Out, or no movement breathes with max 0.2 Hz * 0,05 m (5 cm chest movement) Doppler shift Fd = (2* 0.2 * 0.05)*(ft/c) = 0,02 * ft/c = 0.02 * 2.4G/300M = 0.16 Hz.

Required max resolution = Fresmax = 2*Fd/3 =0.10666666 FFT size: Nmin = fs / fresmax = 3.2M * 0.10666666 Hz = 30 M Updates/s required = 0.2 Hz breathing * 3 points = 0.6 updates/s Updates/s maximum = 3.2M / Nmin = Fd

Conclusion we require 0.1066666 & gt; 0.6 which it's not.
So it doesn't matter how fast we breathe.

However if we use a **sliding window** so we can take 30M samples per FFT it might be possible. Matlab can't handle such large buffers though.

If we are breathing fast, 2Hz we get Fd = 1.6 Hz.
Fresmax = 2 *Fd = 3.2 Hz (this covers the peaks so we can see on which side our reflected peak was. It's not great but if there is no interference it's doable)
Samples/fft = Nmin = 3.2M / 3.2 = 3M = 1 M.

Updates/s required minimum = 2 extrema per breath = 2*2 = 4Hz *if we happen to be in phase* or more if phase not aligned to breath.
Updates/sec gotten = 3.2M samples/sec / samples/update = 3.2M / (3.2M / 1M) = 1M
Updates/s maximum = 3.2M / Nmin = 3.2M / 1M = 3.2 And for 2.4Msample/s we would get 2.4 Which not more than 4 Hz but itâĂŹs getting very close.

If we say "it's ok if the display doesn't update much as long as it tells you from time to time if it's in or out" then it's possible at the bare minimum of visibility, and still we would need to perform FFT on 1M samples, or more to see proper peaks.

Luckily, wifi-g has thin peaks in it's spectrum, rather than one big block, so we can probably say that BWpeak < Fd.

Now if we use a **sliding window**. So, we can set FFT size to the desired 1M and get up to 2.4M updates per second. If we take frame step size = 1.2M we get 2 updates per second, if we take 0.6M we get 4 updates, and if we take 0.3M we get 8 updates. But lets take 100 updates. The frame step size becomes 2.4M/100 = 0.024M = 24K.

So all we have to do is take a sliding frame of minimum 1M size, that takes steps of size 24K, and feed it to a FFT with N=1M.

Then we take the magnitude plot (or power density plot) and subtract the one for the initial measurement of the plain transmission. We will now see the breathing as a peak moving back and forth up to +- Fd with breathing cycle time. Then we can simply see if the frequency of the detected peak is above or below 0Hz to say "breathing out, or breathing in".

2 Experimentation with WiFi Signal

We upload a file on the on Internet to check the frequency of operation of the signal which denoted by the uploading signal.

We create a transmission signal in from the wifi antenna of the laptop to be received by the wifi router at the university. This signal is detected by SDR RTL antenna. We do this by uploading a file on drive from laptop. The following is the screenshot of the signal detected by the dongle.

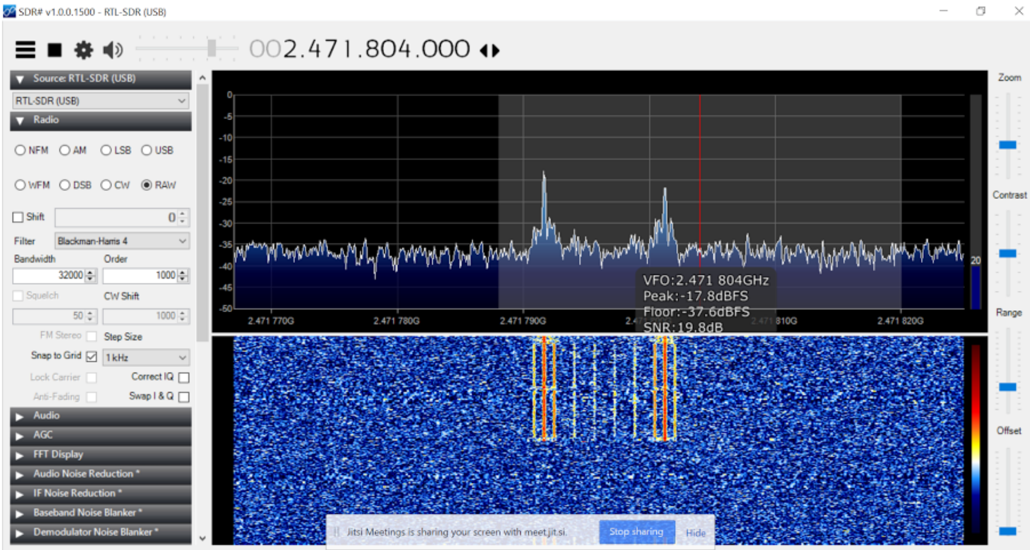

Figure 1.1

However at this frequency the RTL dongle adds noise to the signal making it inefficient to be used for the purpose at 2.41 GHz frequency range. So, it gives a practical proof that we need to optimize the operating frequency in the range of 24-1766 MHz being the efficient frequency range the dongle.

3 Matlab Simulation

Attached to the report is the simulink code which realizes signal processing to detect breath in and breath out. Signal is not taken from SDR RTL but is randomly generated.

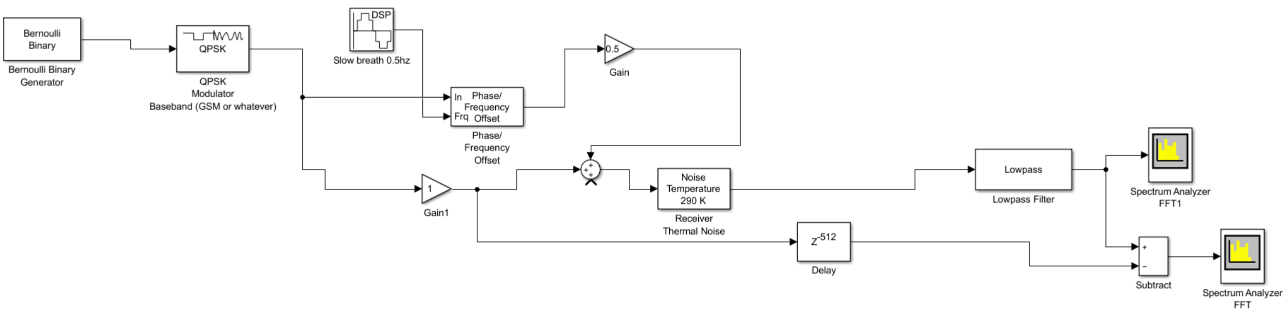

Figure 1.2

The simulation detects inhale and exhale.