Introduction to radar
Raw radar data processing

June 2021

Table des matières

Table des figures

1   Glossary

ADC : Analog to Digital Converter.

Beat Frequency Beat frequency is the difference between the original ramp signal and the echo reflection from the detected object.

CW : Continuous Wave.

FMCW : Frequency Modulated Continuous Wave.

EM : Electromagnetic.

FFT : Fast Fourier Transform.

FoV : Field of View.

2   Radar Theory

a RAdio Detection And Ranging or radar is an electromagnetic [EM] sensor, used to notice, track, locate, and identify various objects or targets.

When a radar signal is transmitted, it is reflected or re-radiated in one or more directions. Once received by the radar's receive antennas, the reflected signal or echo signal is amplified, down-converted to an intermediate frequency [IF], then processed to extract information such as the target's velocity.

Radar systems are classified into several types based on their functions, transmission rate, antenna geometry, and the illuminator they used : un-modulated continuous-wave radars, modulated continuous-wave radars, Doppler radars, and so on.

For this lab, we will use Infineon's sense2gol Pulse radar which is a Pulse-Doppler radar.

3   Radar data processing

3.1   Transmitted and received signals

The radar signal is sent at the operating frequency, f0, with a reference phase ϕ.

If we consider the reference phase of the transmitted signal to be null, the transmitted signal can be expressed as : $T_{signal} = cos(2 * \pi * f0 * t)$.

The echo signal returned by a moving target is then doppler shifted with an added phase ϕ_d and can be expressed as follows : $R_{signal} = cos(2 * \pi * (f0 + \mathbf{fd}) * t + \phi_d)$ where fd denotes the doppler frequency.

The echo signal is then split and mixed with cosine and sine signals from the demodulator's local oscillator before being low-pass filtered to remove the transmit frequency f0.

The resulting signal is either :

- a cosine wave-form or In-phase signal, $cos(2 * \pi * \mathbf{fd} * t + \phi_d)$

- a sine wave-form or Quadrature signal, $sin(2 * \pi * \mathbf{fd} * *t + \phi_d)$.

- or **both** (in most case)

The I (or In-phase) and Q (or Quadrature) signals have the same amplitude and frequency, but their phases are 90° apart.
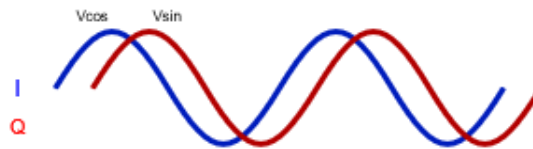
FIGURE 1 – I and Q signals

If the mixer only outputs real samples (I-only or Q-only), only the absolute frequency shift can be obtained, making it impossible to determine the target's direction of travel.

While if both **I & Q samples** (complex mixer) are outputted, the direction of travel of detected objects can be known by checking which of the signals is leading in phase.
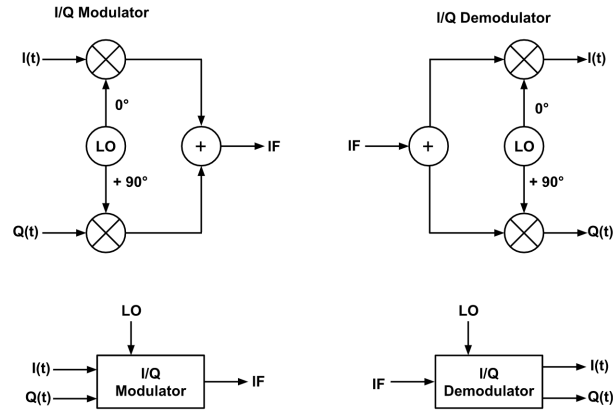
FIGURE 2 – IQ modulator and demodulator. [3]
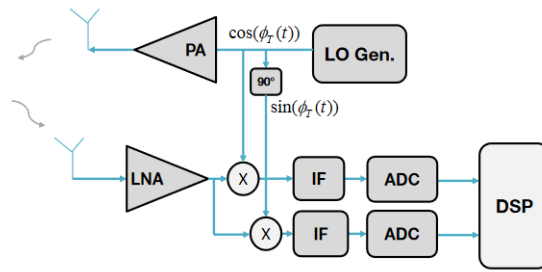
FIGURE 3 – Example of a complex-baseband architecture including an I & Q demodulator. [4]

For this lab, we will use the sense2GoL Pulse radar which has an operating frequency of 24 GHz and a FoV of 80°.

3.2   Sense2gol Pulse radar board

The raw I and Q data acquisition and processing for the sense2gol pulse radar is done for each frame, with a frame being a series of pulses.

The number of pulses is determined by the number of samples acquired by the Analog to Digital Converter (ADC) from the received continuous analog I and Q signals.

There can be up to 256 samples (pulses) for each frame, and each sample is made up of **an** I and/or a Q data value.
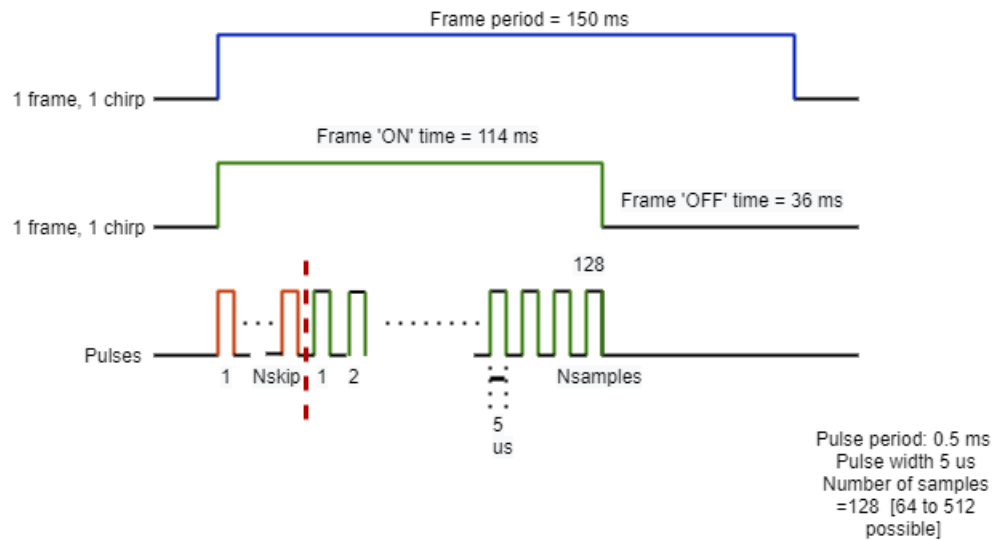
FIGURE 4 – Sense2gol Pulse Frame stucture

For this lab, we will set the number of I **and** Q values per frame to 128. As a result, for each frame, we will have, 128 values composing the digital In-phase cosine-wave **and** 128 values composing the digital Quadrature sine-wave.

A processing algorithm is then applied to each frame across the 128 raw I and 128 raw Q values to determine whether or not there is a target, the velocity of said target, and the direction of it's motion.

Note :

the sense2gol can't determine a target's range/ position.

Only when the radar, the target, or both are moving can a target be detected.
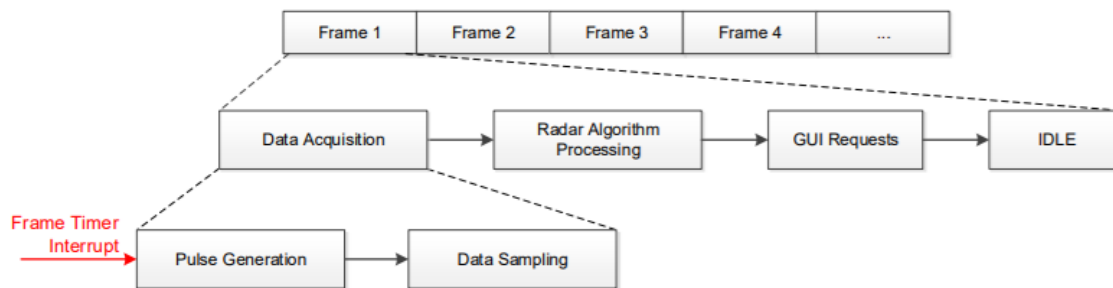
FIGURE 5 – Acquisition flow.Image source : [5]

3.3 Processing the raw data

Because the sense2gol pulse radar cannot change its operating frequency during measurements, key metrics like the range of detected objects can't be measured.

It can, however, estimate the velocity of a target thanks to the Doppler shift, which is a change in the frequency of an electromagnetic wave induced by motion of either the radar or the target.

Whenever the target is moving towards the radar, the signal's frequency will be **greater**, and the opposite is true if the target is traveling away from the radar.

- when the target is approaching the radar, the doppler shift, fd is :

$$|f_d| = \frac{+2 * V_r}{\lambda}$$

- and when the target is receding from the radar,

$$|f_d| = \frac{-2 * V_r}{\lambda}$$

with :

- V_r, the radial velocity ($V_r > 0$ for approaching targets and $V_r < 0$ for departing targets)

-λ the signal's wavelength. $\lambda = \frac{Speed\ of\ light}{operating\ or\ transmission\ frequency}$
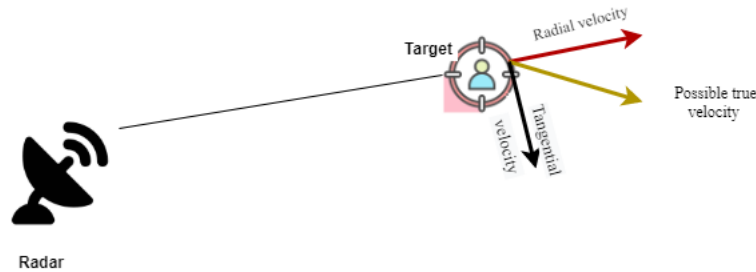
FIGURE 6 – Radial velocity of a target

To extract the velocity of a target, a doppler processing needs to be done.

Because this is an unmodulated radar (the transmit frequency does not change), all frequency shifts are caused by the Doppler shift. And since the Doppler shift spreads the signal power into both I and Q channels, we must rely on the signal's energy to retrieve the speed of the target.

The flow chart of the algorithm used to detect motion, compute the target's speed and determine its motion direction is given below.
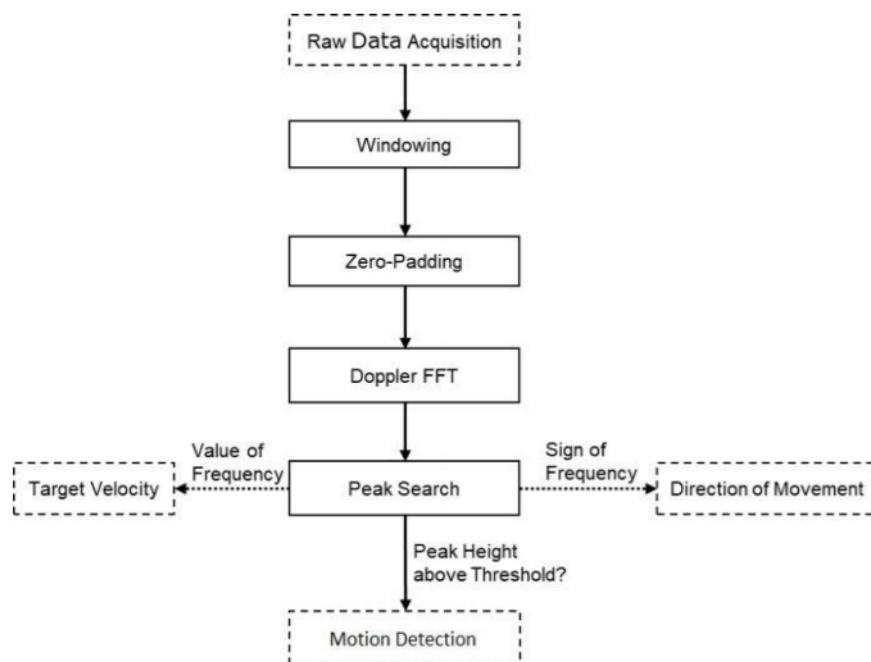
FIGURE 7 – Movement detection algorithm flow provided by Infineon. [5]

4 The lab

4.1 Data visualization

— **Raw I and Q data visualization**.

The **PlotDemo.m** matlab script can be used to visualize the I and Q waveforms in real time. One can experiment with moving their hands towards and away from the radar to see how it alters the waveforms.
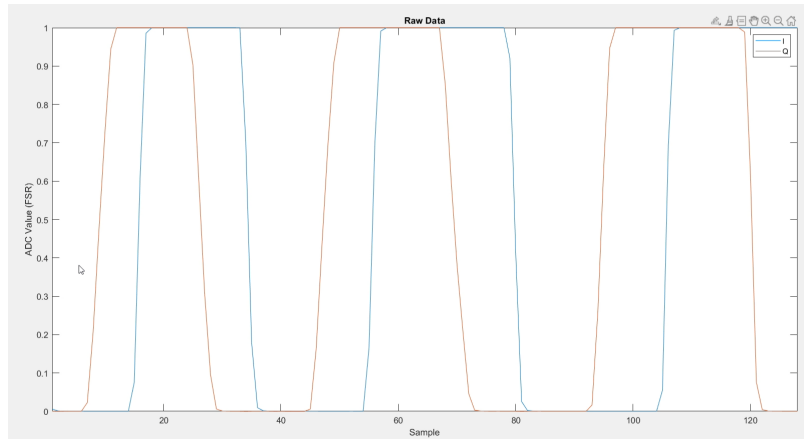
FIGURE 8 – raw I and Q data visualization

— **Live visualization of the radial velocity and direction of movement**.

The **Plot_processed_data.m** script is also provided for live visualization of the **radial** velocity.
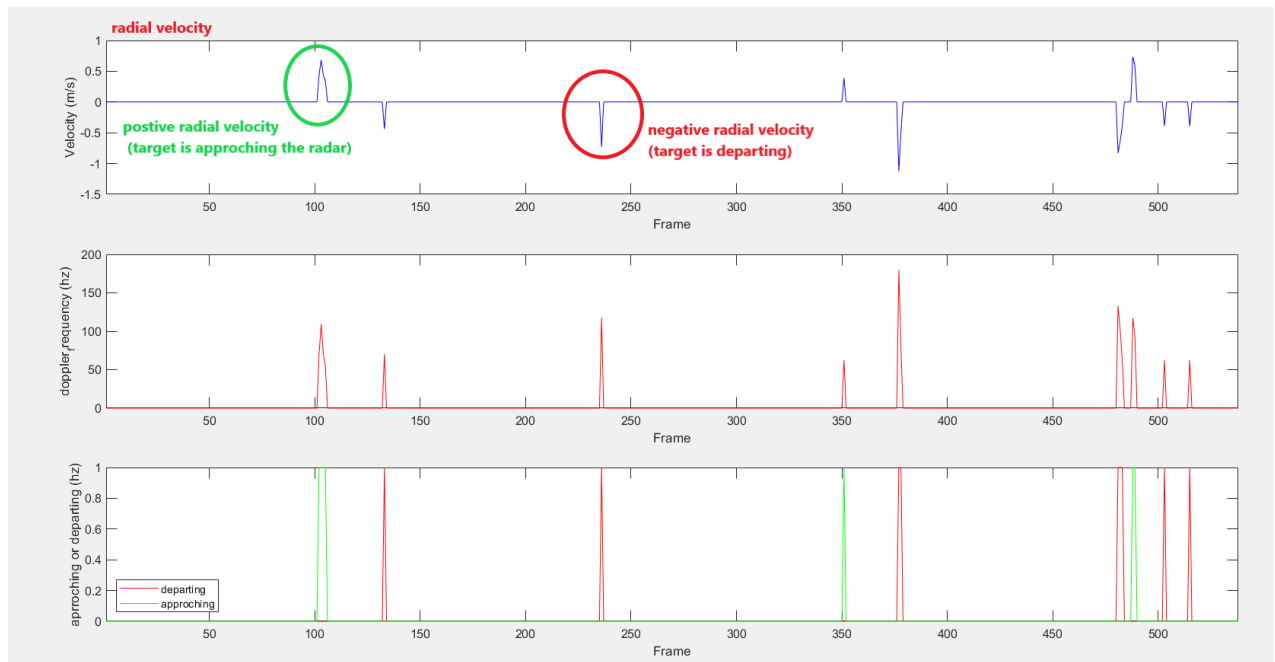
FIGURE 9 – live visualization of radial velocity and direction of movement

It should be noticeable that the velocity when moving towards the radar is positive while it's negative when backing off from the radar.

4.2   Data acquisition

— **Raw data and processing results acquisition**

Before processing the data, the raw I and Q radar data needs to be collected. Once acquired, We will process them to derive the velocity and direction of movement in order to compare our results to those of Infineon.

The following conduct for the data acquisition is recommended :

— Foresee approximately 2m of free space starting from the radar.
— Run the **extract_raw_data.m** script and wait for an empty figure to appear, which indicates that the radar has begun acquiring data.
— Begin walking away from the radar at a normal pace until the ~ 2 meters distance is reached.
— Stay still at that distance for a couple of seconds (~ 5 s) .
— Walk towards the radar at a normal pace and stop the matlab code from running by closing the matlab figure.

A matlab figure that recapitulates the velocity of the detected motions should be displayed
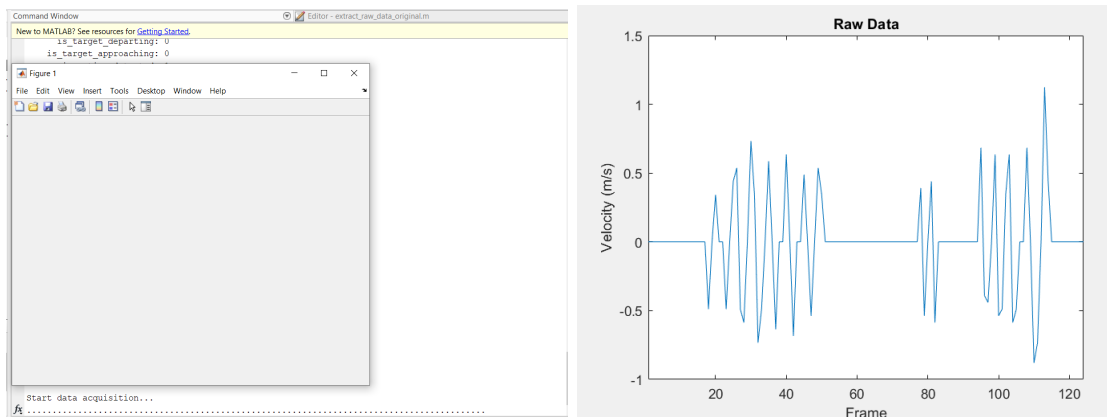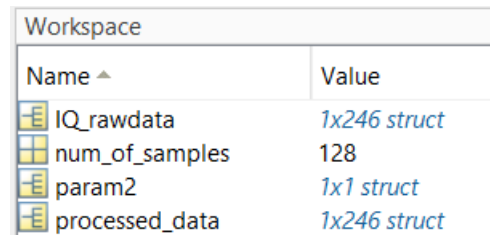
FIGURE 10 – **Left :** The empty figure to close when done acquiring data. **Right :** Summary of the velocity for detected movements

— **The acquired information**

The workspace should be saved so that the 'workspace'.mat file could be loaded in the matlab script for data processing. This will prevent having to acquire data again.
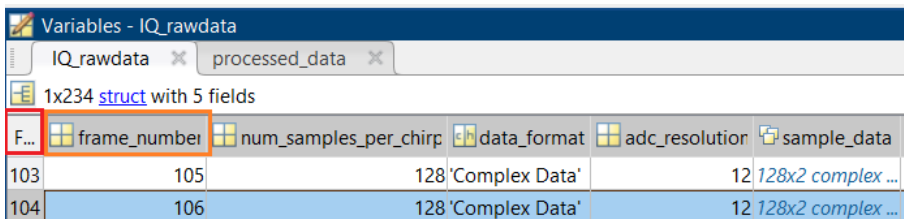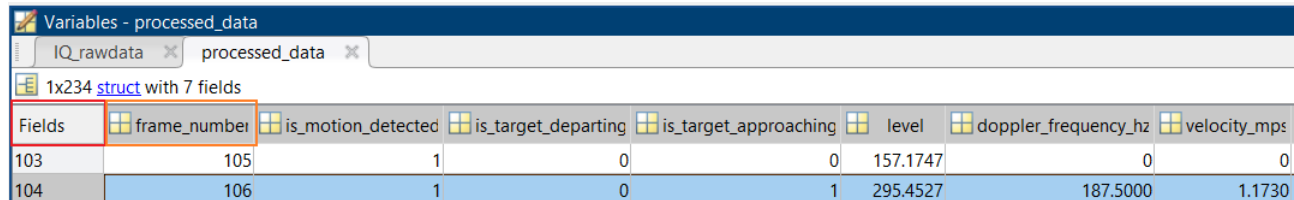
FIGURE 11 – Matlab workspace after running extract_raw_data.m

The **extract_raw_data.m** script gives for each frame the raw I and Q data (**IQ_rawdata**) and the results of Infineon's processing algorithm (**processed_data**).
Since the frame number is given, we can match the raw and processed data for each frame.

FIGURE 12 – **Left :** The empty figure to close when done acquiring data.

For instance, in the figure above, IQ_rawdata(field #104).sample_data(:,1) is the complex raw radar signal (I + JQ) of **frame** #106 and data(field #104).velocity_mps is the velocity estimation for said frame.

Note : The total number of frames (234 in the example above) depends on the acquired data but the number of samples for a given frame is set to 128.

4.3 Data processing using matlab

4.3.1 Prerequisites

The 'workspace'.mat used needs to be specified in the **IQ_data_Processing_TODO.m** script.

```
clc
close all;
clear all;

%% saved workspace data
load towards_128samples_workspace.mat
%load away_128samples_workspace.mat
```

FIGURE 13 – Loading the saved workspace

In the **IQ_data_Processing_TODO.m** script, some key parameters are defined :

— **The operating frequency** is the **fixed** base frequency of the transmitted and received radar signals . It therefore dictates the wavelength of those signals.

— **The sampling frequency** is the frequency at which the I and Q data are sampled by the ADC. See figure 3

— **The IF_scale** is the scaling factor by which we multiply the raw I and Q data to increase the raw signal power or strength.

Key parameters	
Parameters	Values
operating frequency, f0	24.05 GHz
wavelength m	C/f0 = 0.0125 m
sampling frequency	2000 Hz
number of samples	128
IF_scale	10

- C = Speed of light $\approx$ 300000000 m/s

Please double click on the the processed_data workplace variable. and select a field index of a frame which either has the "approaching" or "departing" flag set to 1.

Set the frame_index variable to the chosen frame's field index.

```
% Chose the frame
frame_index=21 ;
```

FIGURE 14 – Field index of the frame to process.

We can then retrieve the I, Q and complex I+jQ raw data from the selected frame and multiply the data by the scaling factor to enhance the power of the signal.

```
% I and Q values of the frame to process
I=real(IQ_rawdata(frame_index).sample_data(:,1))*IF_scale;
Q= imag(IQ_rawdata(frame_index).sample_data(:,1))*IF_scale;
%Raw complex signal = I + jQ
signal=IQ_rawdata(frame_index).sample_data(:,1)*IF_scale;
```

FIGURE 15 – retrieving I, Q and complex I+jQ raw data from chosen frame

A matlab figure for data visualization should appear when executing the IQ_data_Processing_TODO.m script.

Note : Different frame indexes can be chosen to see the wave-form differences for all three cases (approaching, backing off, no motion).
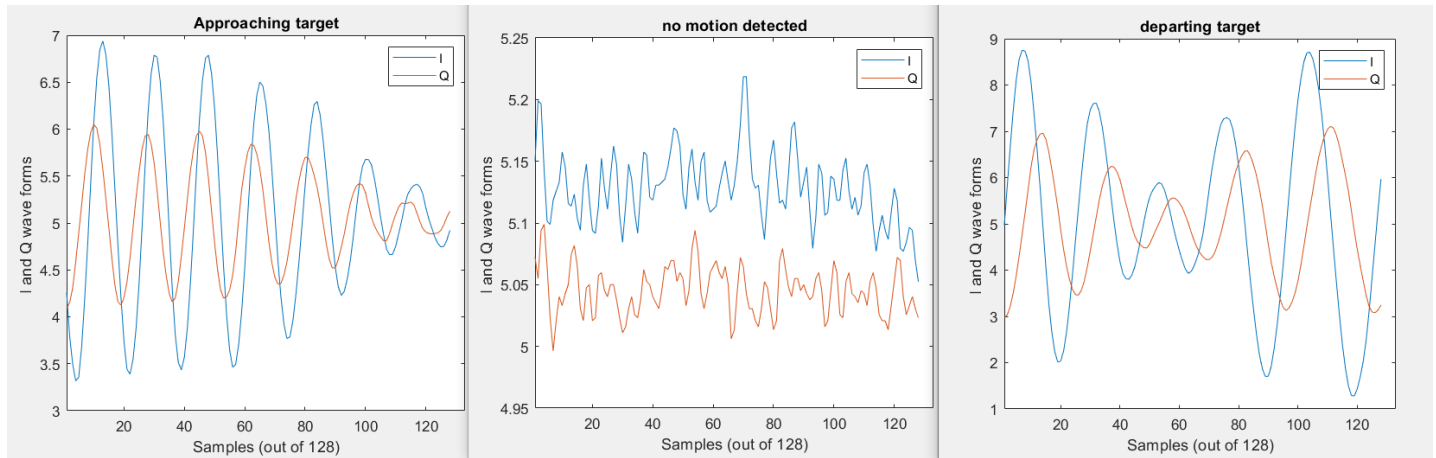
FIGURE 16 – example of I and Q wave-forms for direct cases

Remark : It's noticeable that when the target is backing off from the radar, the In-phase wave is leading in phase. Conversely, the Quadrature wave is ahead in phase when the target is moving towards the radar.

4.3.2   FFT

— A windowed FFT with a Chebychev window will be performed on the data as it is the one used and re-commanded by Infineon. However, Hamming or Hanning windows could be used as well.

```
window = chebwin(128,60); % Chebychev window with 60 dB relative sidelobe attenuation
```

FIGURE 17 – Chebychev window of size 128 with 60 dB relative sidelobe attenuation

The FFT code is as it follows :

N = length(signal) ; | Default=128

Sfft = fftshift(fft(signal.*window)) ; | FFT

f = Fe*((N/2) : -1 : -((N/2)-1))/N ; | frequency vector for the FFT signal

| visualization of the complex I+jQ signal spectrum

figure(2)

plot(f,abs(Sfft))

ylabel('FFT magnitude') ;

xlabel('frequencies ') ;

title('showcase FFT of the complex I+jQ signal')
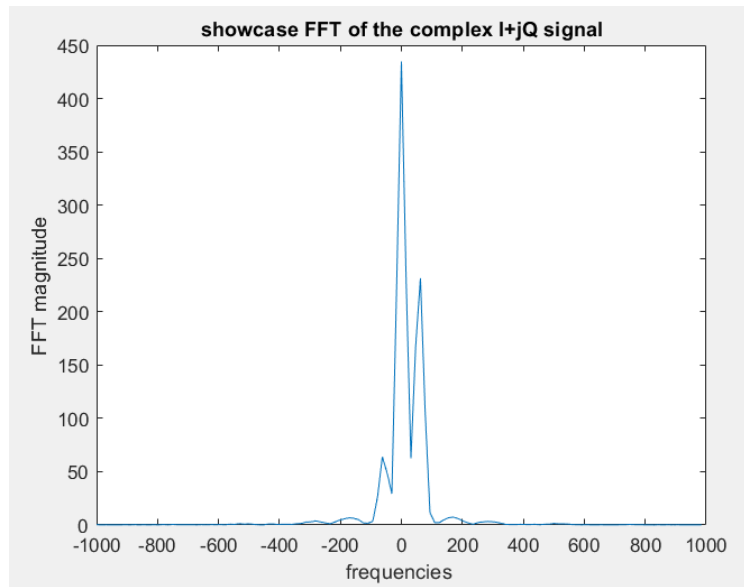
FIGURE 18 – the complex I+jQ signal's spectrum

The spectrum is peaking at the central frequency f0=0. That is because there is a DC component at that frequency which corresponds to the velocity 0m/s. Each frame (even no motion frames) has this DC frequency that conveys no useful information.

Rather, windowed FFT of a signal with DC offset will produce the shape of the FFT of the window function around DC offset bins, which in turn can mask the interesting signals at those bins.

We therefore have to **remove the DC component before performing FFT**. That can be done by simply removing the mean value from the data.

To further improve the signal's characteristics, the data will also be **zero padded** to reach 256 in size. (128 -> 256)

Indeed, increasing the size of the data increases the number of FFT samples (window size) which leads to a more accurate amplitude estimate.

Since **the sampling frequency is unchanged during the process**, more FFT samples will also mean a higher frequency resolution and therefore a more precise representation of the signal. Moreover, since only zeros are added, the data is not altered.

Please refer to Appendix A.2 for a more detailed explanation.

The FFT code of the windowed and zero padded data is as it follows :

Computation of mean values for the removal of the signal's DC component

I_mean=mean(I);

Q_mean=mean(Q);

S_mean=mean(signal);

Removal of the signal's DC component before windowing

I_prefft=((I- I_mean) .* window);

Q_prefft=((Q- Q_mean) .* window);

S_prefft=((signal-S_mean) .* window); |length of S_prefft= length of signal =128

Zero padding the data to 256

Note : 256 is the Infineon radar's maximum number of samples

FFT_size = 256; # Size of FFT = number of bins used for dividing the window into equal strips, or bins

if(num_of_samples < 256)

I_prefft((num_of_samples+1) :FFT_size)=0;

Q_prefft((num_of_samples+1) :FFT_size)=0;

S_prefft((num_of_samples+1) :FFT_size)=0;

end

FFT computing of the windowed and zero-padded raw data

Ifft= fftshift(fft(I_prefft,FFT_size));

Qfft= fftshift(fft(Q_prefft,FFT_size));

Sfft= fftshift(fft(S_prefft,FFT_size));

Frequency vector for the spectrum. (Fe = 2Khz = sampling frequency)

Please refer to Appendix A.2

frequency_resolution= Fe/FFT_SIZE;

f=[((FFT_SIZE/2)-1 : -1 : 0) (0 :-1 : - ((FFT_SIZE/2)-1))]* frequency_resolution;

The frequency vector is then :

All possible doppler frequencies									
Bin index	1	2	3	4	5	6	7	8	9	10
Bin number	127	126	125	124	123	122	121	120	119	118
frequency	992.1	984.3	976.5	968.7	960.9	953,1	945,3	937,5	929,6	921,8

··········

Bin index	124	125	126	127	128	129	130	131	132	133
Bin number	4	3	2	1	0	0	-1	-2	-3	-4
frequency	31.25	23.43	15.62	7.81	0	0	-7.81	-15.62	-23.43	-31.25

··········

Bin index	248	249	250	251	252	253	254	255	256
Bin number	-119	-120	-121	-122	-123	-124	-125	-126	-127
frequency	-929.6	-937.5	-945.3	-953.1	-960.9	-968.7	-976.5	-984.3	-992.1

The FFT of the windowed and zero-padded raw signals shows that there are two peaks with the same amplitude in the spectrum for the I-only and Q-only signals. These peaks appear on the doppler frequency -62.5 hz and +62.5 hz.

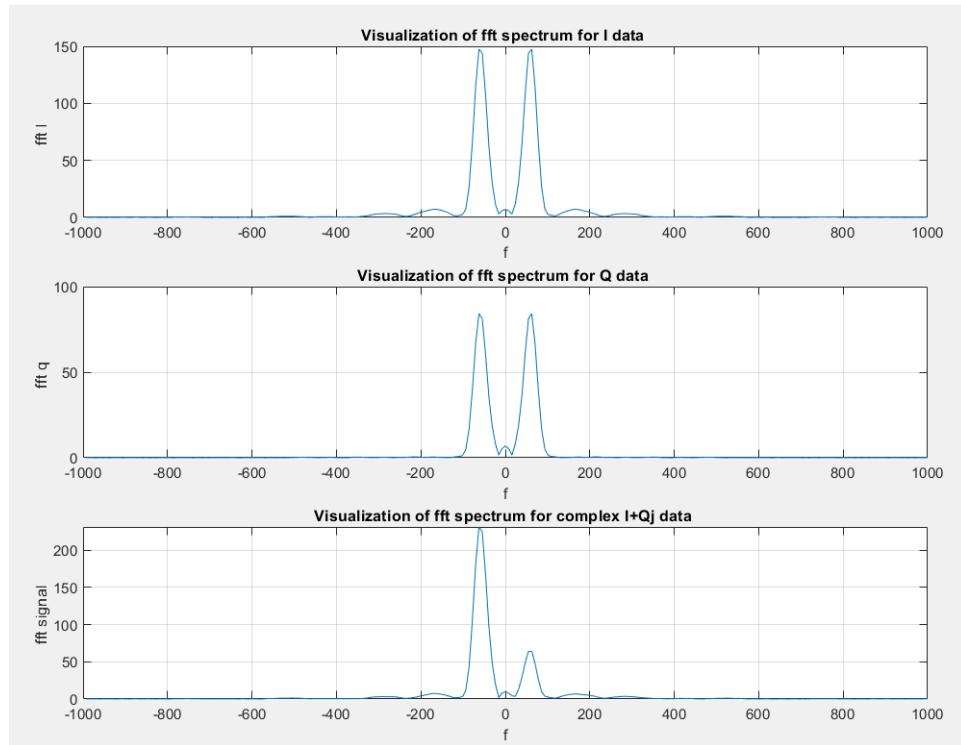However, there is only one peak at the -62.5 hz frequency in the the complex I+jQ signal spectrum.

-t

FIGURE 19 – I , Q and complex I+jQ signals spectra (departing target)

As previously stated (3.1), we can only retrieve the absolute doppler frequency and thus cannot determine the direction of motion using only the In-phase or the Quadrature signal.

Whereas the complex signal's spectrum peaks at the right doppler frequency (-62.5 hz) which indicates that the frame being processed captures a receding target.

4.3.3 Peak-finding in the spectrum and velocity computation

A peak search in the signal's spectrum is required to determine the doppler frequency which is the frequency with the highest amplitude.

The target's velocity is then calculated by multiplying the doppler frequency by the wavelength of the signal. However, because we have two-way propagation (from radar to target and from target back to radar's Rx antennas), it must be divided by two.

— The peak search code is as follows :

```
#  All peaks and their matching bin indexes are repertorized.
[peaks_values, frequency_shifts] = findpeaks(abs(Sfft)) ;

# Finding the peak with the highest amplitude and it's corresponding bin index
[peak_value, peak_Bin] = maxk(peaks_values,1) ;

maxBin= frequency_shifts(peak_Bin)

# Retrieving the frequency of the peak bin ( doppler frequency)
doppler_frequency =f(maxBin) ;
# Computing the velocity of the target
velocity= (doppler_frequency * wavelength)/2 ;
```

4.3.4 Direction of motion

Before we can determine the direction of motion, we must ensure that the peak value at the doppler frequency is greater than a predefined threshold. If not, the frame is interpreted as conveying either no motion or a motion with no discernible radial direction.

The sign of the doppler frequency then indicates the direction of motion. A target moving away from the radar's transmit antennas has a negative doppler frequency, whereas a target moving towards the radar has a positive doppler frequency.

The code for determining motion direction is as follows :

```
motion_direction_threshold= 20

if peak_value > motion_direction_threshold

        if doppler_frequency < 0
                disp('Target is departing')
                target_direction= - 1 ;

        elseif doppler_frequency > 0
                disp('Target is approaching')
                target_direction= 1 ;
        end
```

```
else
      target_direction= 0 ;
end
```

4.3.5   Comparison with Infineon's results

To compare the matlab based processing with Infineon's C language based algorithm, the **IQ_data_Processing_All_frames.m** script can be used to process every frame which has a spectrum peak amplitude higher than infineon's threshold.

Both results are put on the **processed_data** matlab structure for an easy, side by side comparison.

doppler_frequency_hz	velocity_mps	Matlab_dopplerFreq	Matlab_velocity
0	0	[]	[]
85.9375	0.5376	85.9375	0.5376
101.5625	0.6354	101.5625	0.6354
0	0	[]	[]
78.1250	0.4888	78.1250	0.4888
93.7500	0.5865	93.7500	0.5865
109.3750	0.6843	109.3750	0.6843
62.5000	0.3910	62.5000	0.3910
85.9375	0.5376	85.9375	0.5376
93.7500	0.5865	93.7500	0.5865
109.3750	0.6843	109.3750	0.6843
109.3750	0.6843	109.3750	0.6843
93.7500	0.5865	93.7500	0.5865
93.7500	0.5865	93.7500	0.5865
101.5625	0.6354	101.5625	0.6354
85.9375	0.5376	85.9375	0.5376
62.5000	0.3910	62.5000	0.3910
62.5000	0.3910	62.5000	0.3910
85.9375	0.5376	85.9375	0.5376
54.6875	0.3421	54.6875	0.3421

FIGURE 20 – Comparison between Infineon results and the matlab code.

A   Appendix

A.1   Sense2gol Pulse parameters

- Detection range is of $\sim$ 18 m for a human target

- The maximum detectable speed is 3 m/s or 10.8 km/h

- Number of TX antennas : 1

- Number of RX antennas : 1

- Minimum operating frequency : 24025000 Khz

- Minimum operating frequency : 24225000U Khz

- **Fixed** Base frequency for Doppler estimation : 24.05 Ghz

- Doppler algorithm sampling frequency 2000 hz

- Maximum number of chirps per frame : 1

- Minimum number of samples per frame : 32

- Minimum number of samples per frame : 256

- ADC resolution : 12bits

- Maximum ADC sampling frequency :1500000 Hz

- Maximum pulse width :10 usec - Minimum pulse width : 4 usec

- Maximum frame period : 2000msec

A.2   Frequency bins when zero padding

When performing an FFT on a signal, the size of the transform corresponds to the number of frequency bins with each bin representing the amount of energy present in the signal at that specific frequency.

The frequency resolution is the difference in frequency between each bin. Therefore, the results can only be as precise as the frequency resolution allows.

Sampling frequency, Fe = 2Khz

Possible bin numbers =[((FFT SIZE/2)-1 : -1 : 0) (0 :-1 : - ((FFT SIZE/2)-1))]

Frequency resolution= sampling frequency / FFT SIZE

Possible frequency bins =[((FFT SIZE/2)-1 : -1 : 0) (0 :-1 : - ((FFT SIZE/2)-1))]* frequency resolution

- Without zero-padding the 128 points signal, there are 128 possible bins and therefore 128 possible doppler frequencies.

Possible bin numbers = 63, 62 ... 1 0 0 -1 ... -62, -63

- While when we zero-pad the data to 256 of length, there will be 256 possible bins which helps have a more accurate frequency estimation.

Possible bin numbers = 127, 126 ... 1 0 0 -1 ... -126, -127

Note : the bins are arranged this way (two zeros at the center) so that there is 127 possible bins for negative frequencies (departing target) **and** 127 possible bins for positive frequencies (approaching target) .

Indeed, zero padding helps getting closer to the truer doppler frequency which leads to a more accurate velocity estimation.

For instance, if the true doppler frequency of the analog data is 990 hz, when processing the data, the estimated doppler frequency would mosty likely be 992.1 hz if we did zero-pad the data.

Conversely that true 990 hz doppler frequency would most likely fall into the 984.3 hz frequency bin when the data is not zero padded.

All possible doppler frequencies								
Bin index	1	2	3	4	5	6	7	8
Bin number (zero-padded)	128	127	126	125	124	123	122	121
frequency (zero-padded)	1000	992.1	984.3	976.5	968.7	960.9	953,1	945,3
Bin number (not zero-padded)	64	63	62	125	124	123	122	121
frequency (not zero-padded)	1000	984.3	968.7	953.1	937.5	921.8	906.2	890.6

- The true radial velocity of the target V_r= doppler frequency* wavelength = 990 hz *

0.0125m/2 = 6.187 m/s.

- The estimated velocity **without zero-padding** would then be 984.3* 0.0125 m /2

= 6.151 m/s.

- And the estimated velocity **with zero-padding** = 992.1* 0.0125m /2= 6.200 m/s.

Depending on the application, the difference could be negligible or **not**.

Note :

The frequency for a given bin, fbin=$Bin\ number * frequency\ resolution$

$$fbin = Bin\ number * \frac{Sampling\ Frequency\ (Fe)}{FFT\ size}$$

For instance, for Bin index 3,

- $Frequency\ of\ bin\ index\ 3 = 62 * \frac{2000}{128} = 968.75hz$ # **no zero padding**

- $Frequency\ of\ bin\ index\ 3 = 128 * \frac{2000}{256} = 983.3hz$ # **zero padding** to 256

A.3 Some radar equations

Given applications can have different requirements when it comes to velocity, the processor MIPs or the amount of usable data memory.

We will see in the following subsections that a change in the chirp configuration (frequency slope,sweep bandwidth,etc) can influence the radar's performances.

Therefore, understanding the relationships between the chirp configuration and the system's performance metrics is crucial in order to suitably configure the chirp parameters to meet the requirements.

A.3.1 Signal to noise ratio

The signal to noise ratio, SNR compares the level of a desired signal to the level of background noise. The higher the SNR is , the less noise there is in the signal, as the formula below shows :

$$\text{SNR} = 10 * \log \frac{s^2(t)}{n^2(t)}$$

$s(t)$: desired signal

$n(t)$: background noise

A.3.2 Maximum velocity and velocity resolution

The maximum measurable velocity in Fast FMCW radars depends on the chirp cycle time which in turn depends on how fast the frequency sweep can be performed and the minimum inter-chirp time allowed.

An object's velocity can be directly estimated from the measured phase difference of two peaks of chirp signals which corresponds to the spatial motion of the object :

$$\text{velocity} = \frac{\lambda * \Delta\phi}{4 * \pi * Tc} \text{ \& unambiguous max velocity} = \frac{\lambda}{4 * Tc}$$

$\Delta\phi$: The phase difference of two peaks of chirp signals

Tc : Total Chirp Time (chirptime + idletime)

λ : Wavelength of the signal

The maximum unambiguous velocity is the maximum range of radial velocity that can be observed without ambiguity by the radar. With the radial velocity being the motion along the radial (i.e directly toward or away from the radar) which is different from motion perpendicular to that radial or tangential velocity.

The velocity resolution that estimates how well two objects with small velocity differences can be resolved is obtained with the following equation :

$$\text{velocity resolution } \Delta\text{v} = \frac{\lambda}{2 * Nc * Tc}$$

Nc : number of chirps in a frame

Frame duration= Nc * Tc

The velocity resolution depends highly on the number of chirps in a frame and thus, on the transmit frame duration. Having more chirps in a frame improves the velocity resolution.

A.3.3   Power of the reflected signal

As we stated before, a radar's basic operation is to transmit an EM signal and receive the echo signal reflected by the target. The power of the echo signal can computed thanks to the radar equation :

$$\text{Prx} = \frac{Ptx * Grx * Gtx * \lambda^2 * \sigma}{(4 * \pi)^3 * R^4 * L}$$

Pt : Transmit power

Grx : Reception Antenna's gain

Gtx : Transmission Antenna's gain

λ : wavelength

σ : Target's radar cross section (RCS)

R : Range from the radar to the target

L : Losses due to propagation or/and system

Note : A target's RCS indicates how easily an object can be detected. It depends, for example, on the material of which the target is made or it's size. High RCS targets are deemed easier to detect.

A.4   Infineon toolbox : start-up instructions

To be able to use infineon's radar softwares, the Infineon toolbox needs to be downloaded. All the step are described below :

A.4.1   Download and install the Infineon Toolbox

The Infineon Toolbox provides access to the Radar GUI,the Dave IDE as well as the XMC Flasher which helps program the radar board's memory.
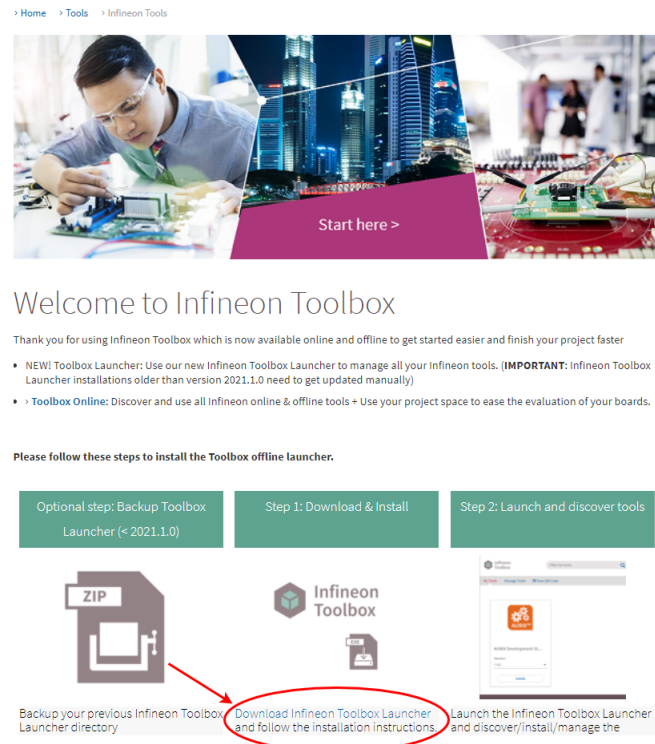
To download the Infineon Toolbox, you can use this link : `https://www.infineon.com/cms/en/tools/landing/infineontoolbox.html`

FIGURE 21 – Downloading the Infineon Toolbox.

A.4.2   Install the useful tools

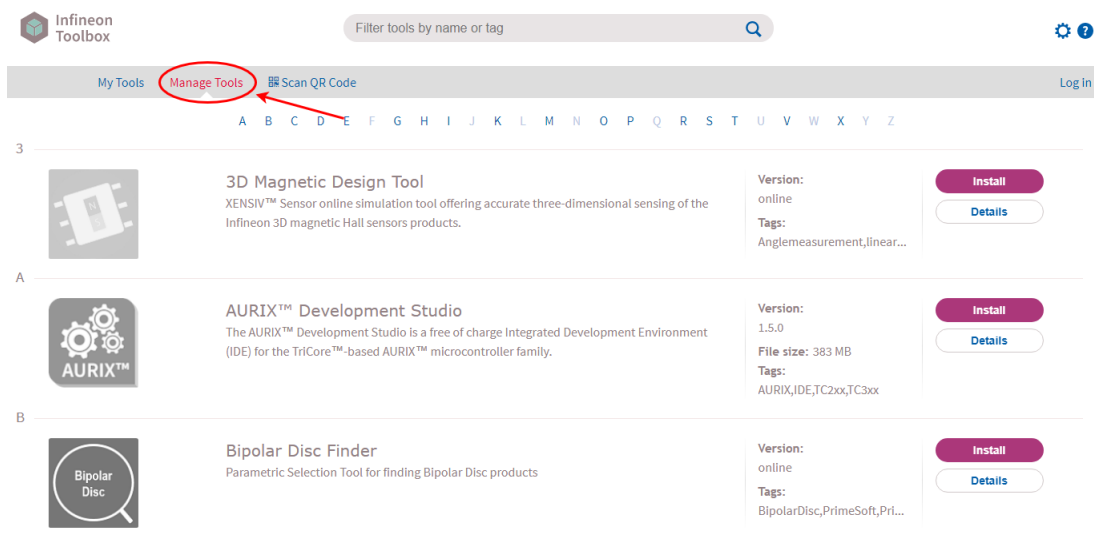Once the Infineon toolbox is successfully downloaded, you can launch it then click on "Manage Tools" :

FIGURE 22 – Manage Tools.

In the search bar, search for the useful tools that you need and install them :

- Sense2GoL Pulse kit : provides resources and demos for the board

- XMC flasher : helps program the board with an .hex file

- Radar gui : live visualization and parameter modification
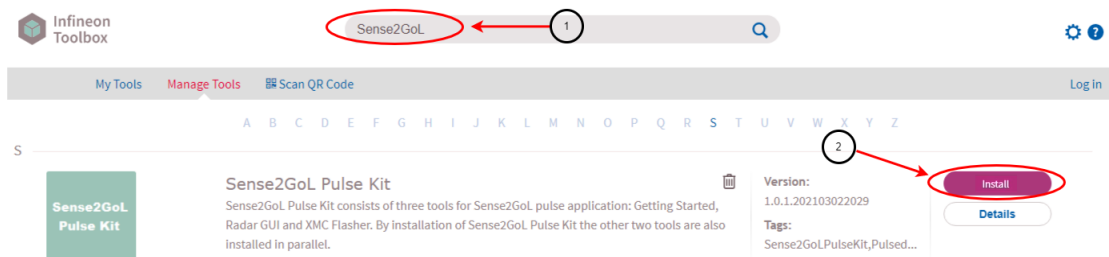
FIGURE 23 – Sense2GoL Pulse kit.
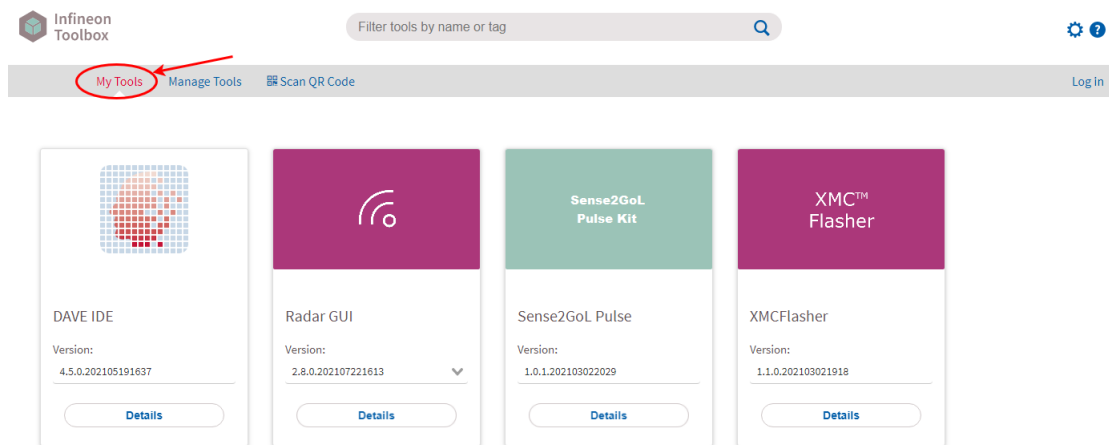
FIGURE 24 – All usefull tools.

To have access to resources and demos for the board press the start button on the Sense2Gol pulse kit imageto start. You will be redirected in order to download the installing software.
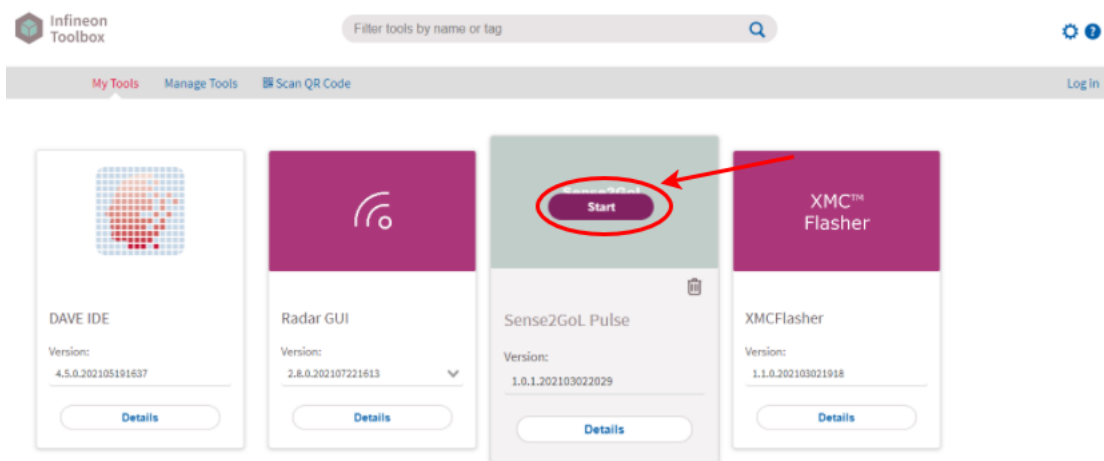
FIGURE 25 – Press the start button to download ressources for the board
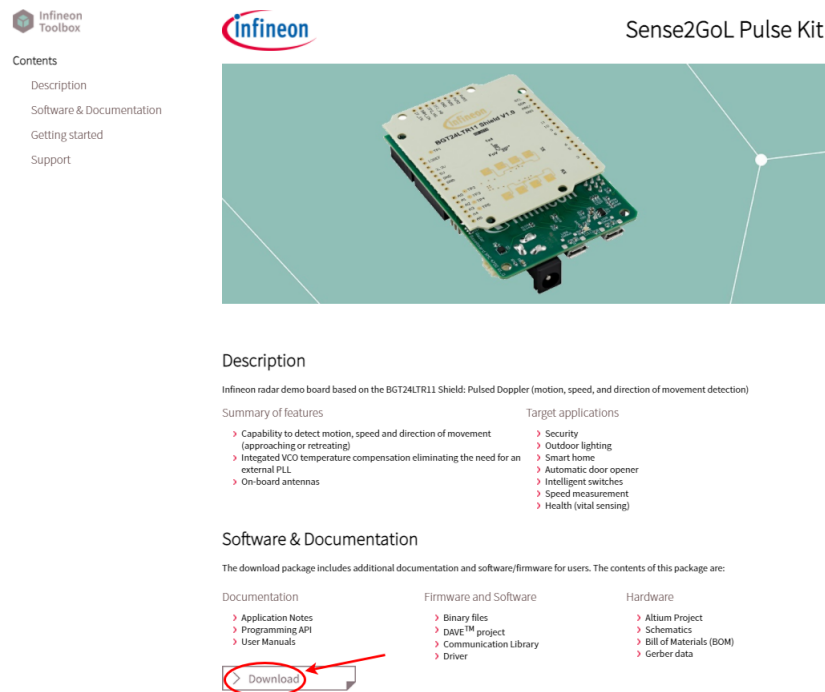
FIGURE 26 – Download the sofwtare demos and documentation.

Download these files into your work repertory. Thus, you will get a file entitled "IFX_S2GLP-HW-SW_Vx.x.x".

A.4.3   Radar GUI start-up

The next step is to use Infineon's live data visualization software, Radar GUI. On the home page of the toolbox software (→ My Tools) click on start for Radar GUI :
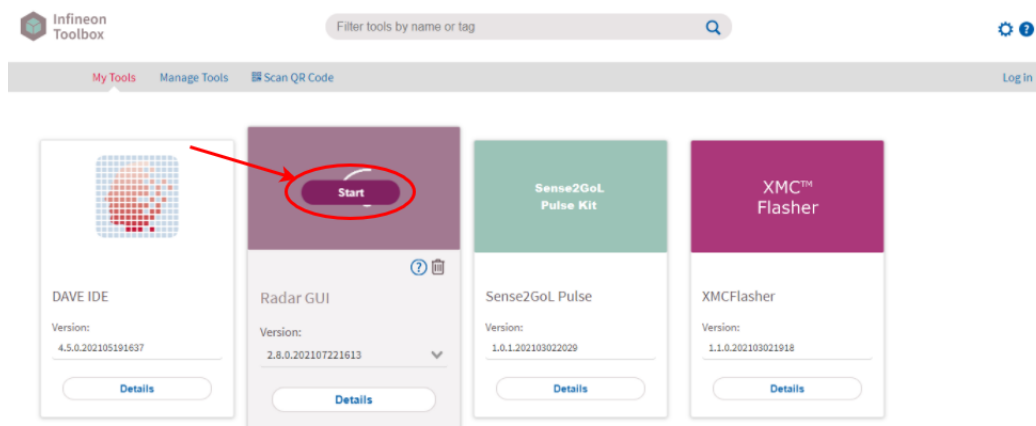
FIGURE 27 – Start of the Radar GUI.

Don't forget to plug in the Sense2gol pulse board using the **default** usb port.

FIGURE 28 – Connection with the USB port of the Sense2GoL Pulse board.

Notes : If the device is still not recognized, right-click on "My Computer" → Manage → Device manager → Other devices → Right-click on "Unknown device" → Update driver software → Browse → Firmware_Software → XMC_Serial_Driver to download the required driver.

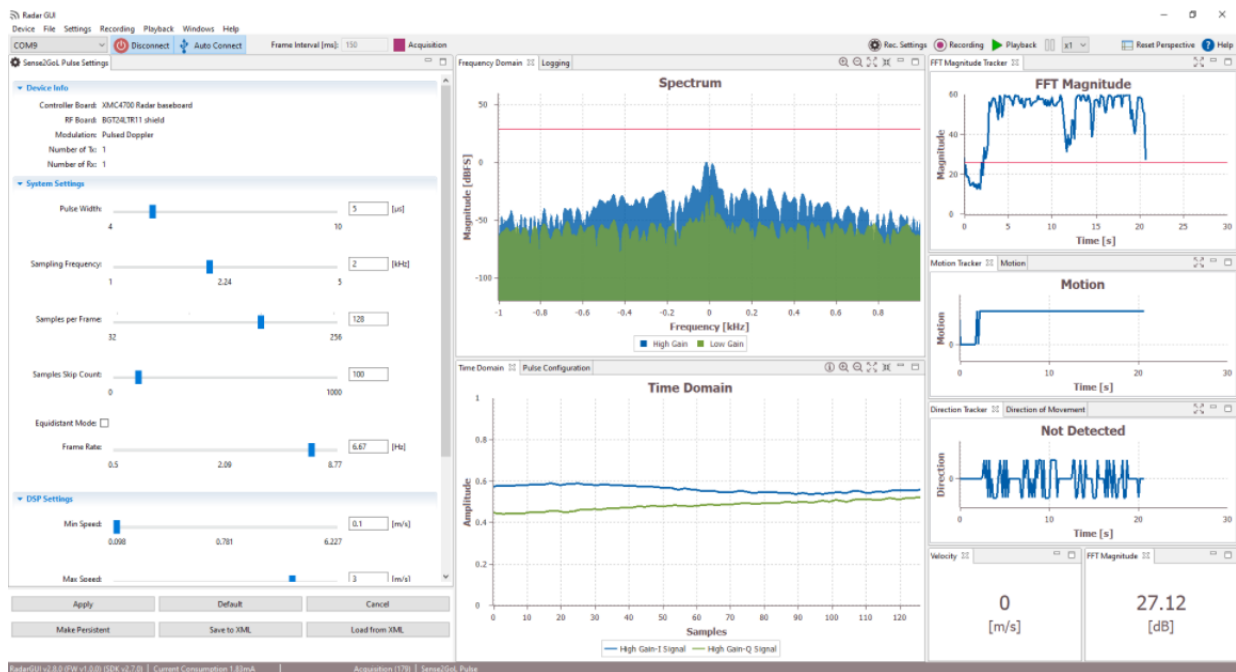Now the Radar GUI should be working and you should be able to see the real-time data :

FIGURE 29 – Radar GUI.

All the information about graphs and parameters are given in the following paragraph.

A.5   Adjustable Parameters

Some parameters such as the number of samples per frame can be modified. We recommend changing each parameter once at the time to see the influence of each parameter on the radar's performances. The following parameters are adjustable :

— **Pulse width :** duration of a single pulse.

— **Sampling frequency (or sampling rate) :** the number of occurrences within a given time period ;

— **Samples per frame :** the total number of pulses during the frame time

— **Sample skip count :** the number of samples to be skipped in each frame to get ride of the direct current offset in the I and Q signals.

— **Frame rate :** the number of frames per second.

— **minimum and maximum speed** .

— **motion and direction sensibility**.

Use the system settings and the DSP settings to test different parameter settings.
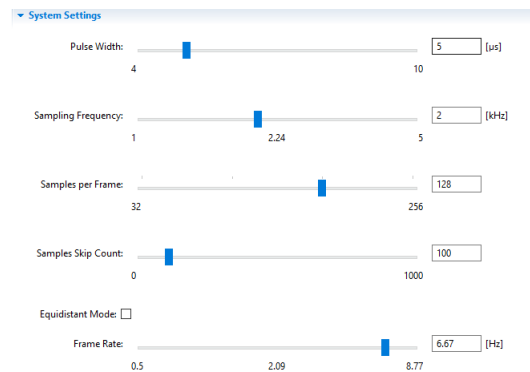
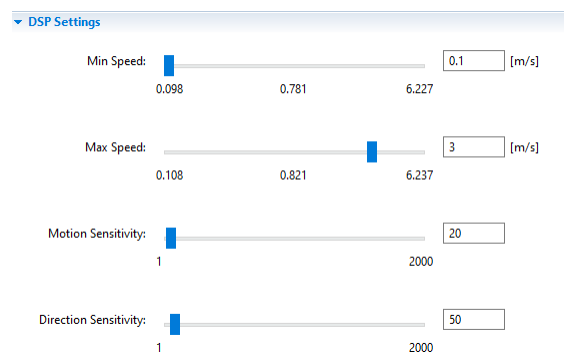FIGURE 30 – Adjustable setting parameters.

FIGURE 31 – DSP settings.

A.6   Graphs provided by the GUI
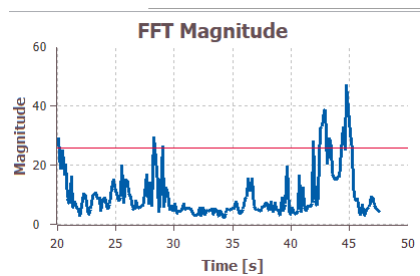
the signal's FFT magnitude in time domain

FIGURE 32 – FFT magnitude in time domain and threshold.

Note Changing the sensibility of the radar (motion or direction sensibility) changes the FFT magnitude threshold (red line) which is the motion detection threshold.

the signal's FFT magnitude in frequency domain
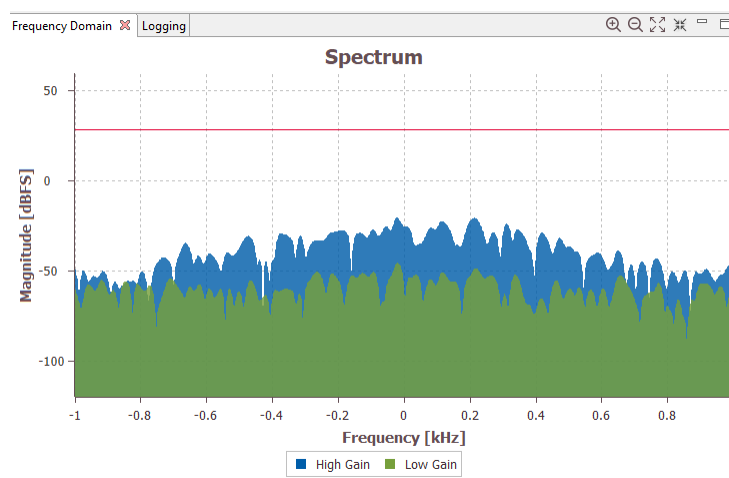

FIGURE 33 – FFT magnitude in frequency domain and threshold.

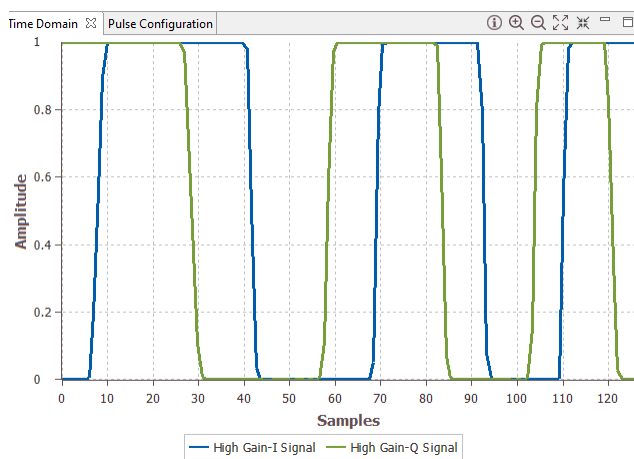raw I and Q data in phase quadrature.

FIGURE 34 – Amplitude in the time delay.
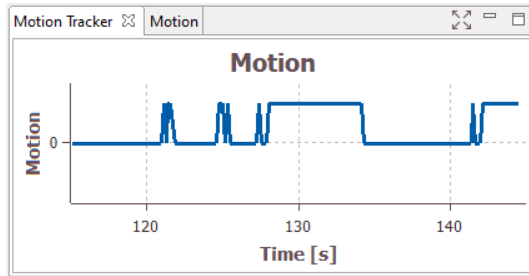
Motion and direction tracker
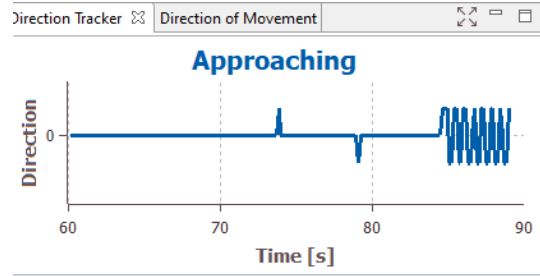

FIGURE 35 – Motion Tracker.
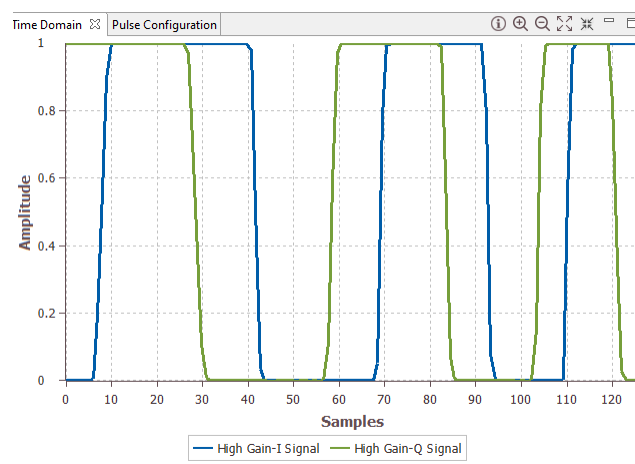

FIGURE 36 – Direction Tracker.

FIGURE 37 – Amplitude in the time delay.

Notes : There are no jumpers on the v2 model sense2gol pulse board that we use. So if you follow the getting started tutorial provided by Infineon don't be mindfull of the steps including Jumpers modification.
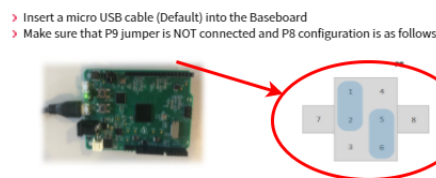

FIGURE 38 – Theoretical modification of jumpers.

B   Bibliography

[1] Doppler Estimation - MATLAB & Simulink - MathWorks Nordic.

[2] Ozdemir, Caner. Inverse Synthetic Aperture Radar Imaging With MATLAB Algorithms.
John Wiley & Sons,2021.

[3] « In-Phase and Quadrature Components ». Wikipedia, 14 juin 2021. Wikipedia,

[4] Using a complex-baseband architecture in FMCW radar systems. Texas Instruments

[5] Sense2GoL Pulse (Pulsed Doppler radar) software user manual. Infineon

[6] Electrical Engineering Stack Exchange.