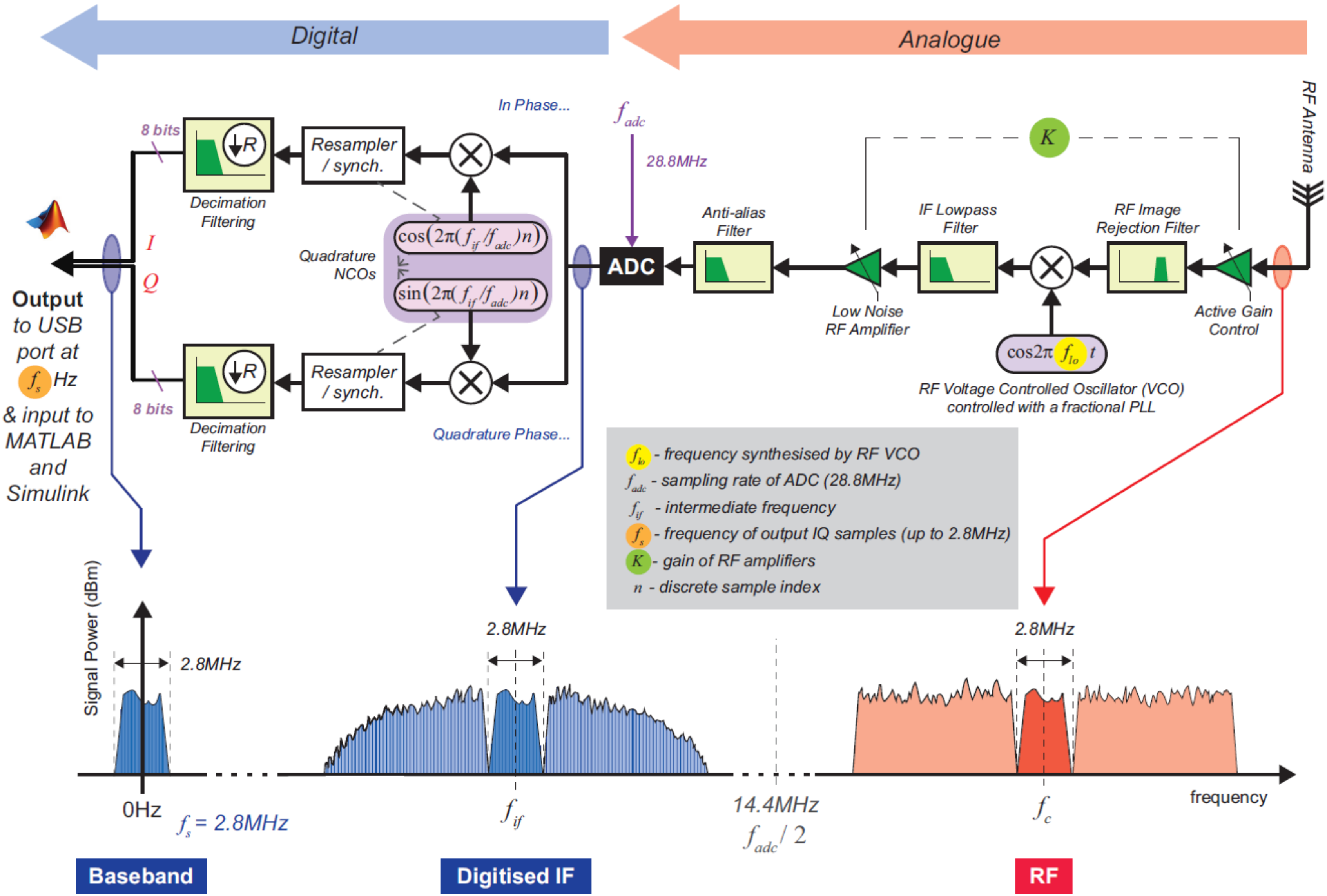
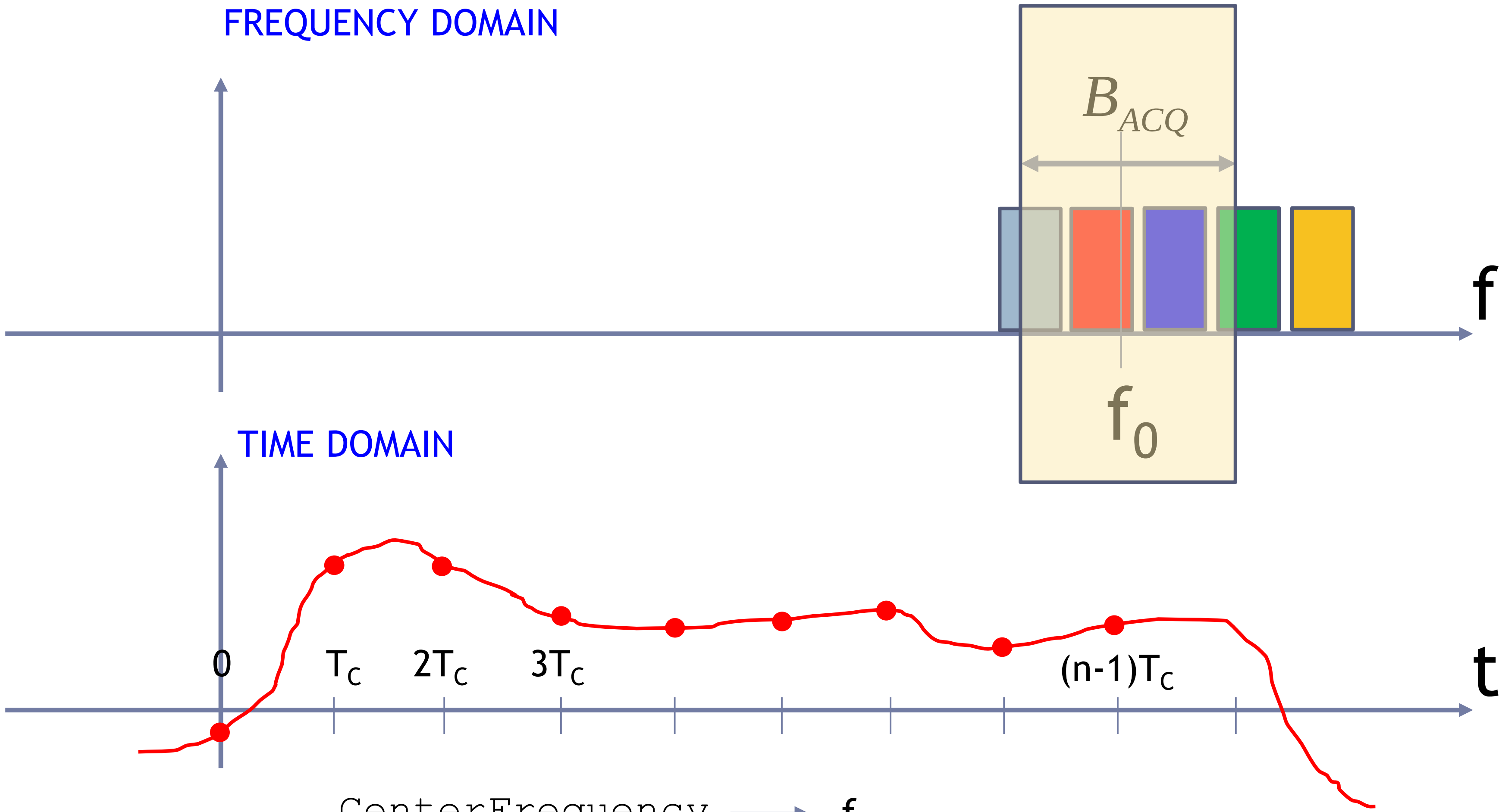
LAB #3.2 – SPECTRUM ANALYZER

RTL-SDR: how it works



SIGNAL ACQUISITION

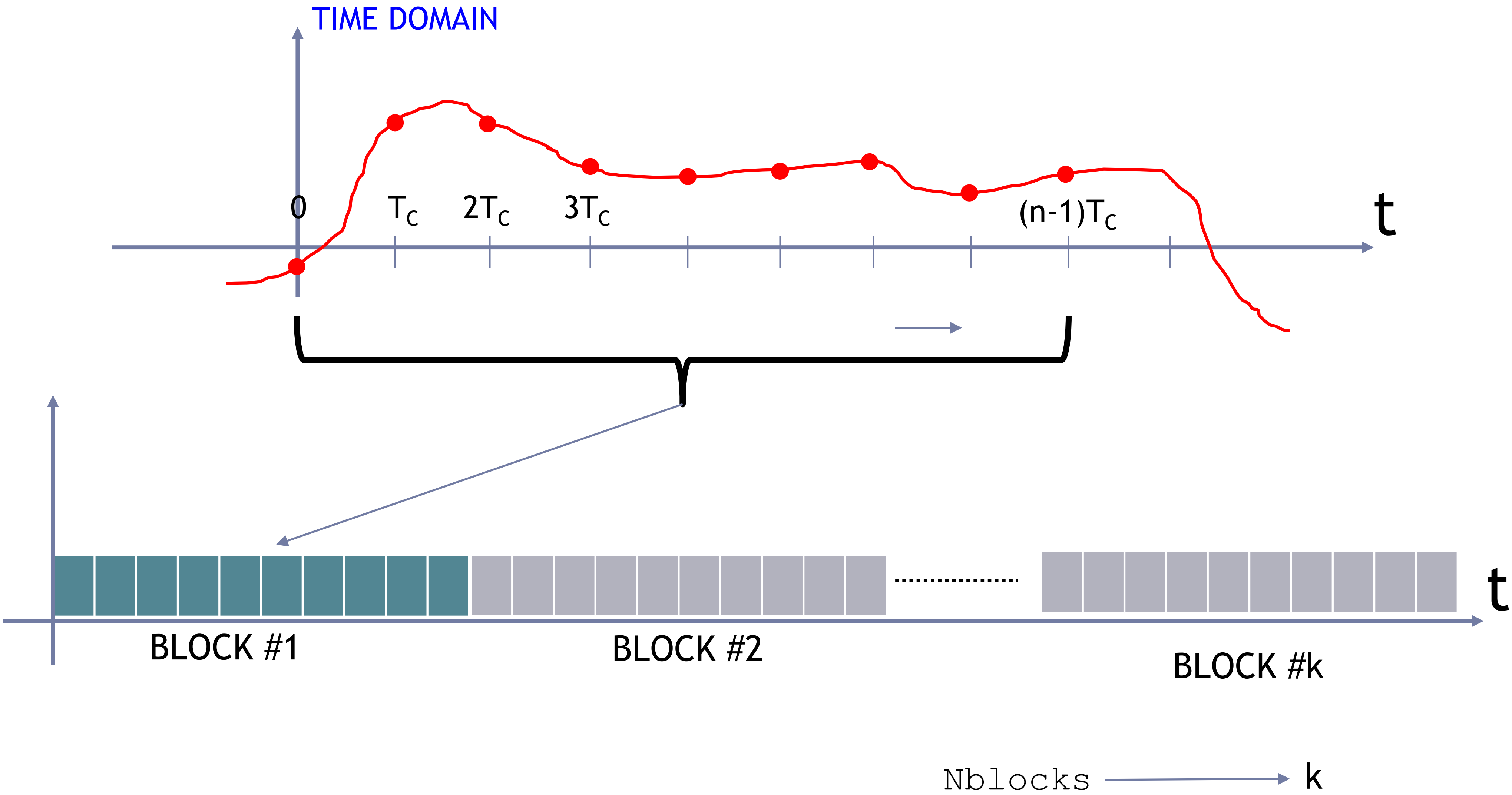


`CenterFrequency` → f_0

`SampleRate` → f_C → B_{ACQ} → T_C

`NsBlock` → n

SIGNAL ACQUISITION



SIGNAL TYPE

The SDR receiver down-shift a pass-band signal and it acquires the two QUADRATURES of the signal

The resulting sample vector contains a COMPLEX SIGNAL

$$x_I(t)+jx_Q(t)$$

MATLAB CODE

```
CenterFrequency=92.1e6;
SampleRate=1e6;
NsBlock=1024;

Nblocks=1000;

hRadio = comm.SDRRTLReceiver('CenterFrequency', CenterFrequency, ...
   'SampleRate', SampleRate, ...
   'EnableTunerAGC', true, ...
   'SamplesPerFrame', NsBlock, ...
   'OutputDataType', 'single');

if ~isempty(sdrinfo(hRadio.RadioAddress))

    x_saved=NaN*ones(Nblocks*NsBlock,1);

    for Counter=1:Nblocks

        [x, len,lost(Counter)] = step(hRadio);

        x_saved((Counter-1)*NsBlock+1:(Counter-1)*NsBlock+NsBlock)=x;

    end

else
    warning('SDR Device not connected')
end
release(hRadio);
```

GENERAL PARAMETERS

SETUP SDR RECEIVER

CHECK SDR

INITIALIZE OUTPUT SIGNAL

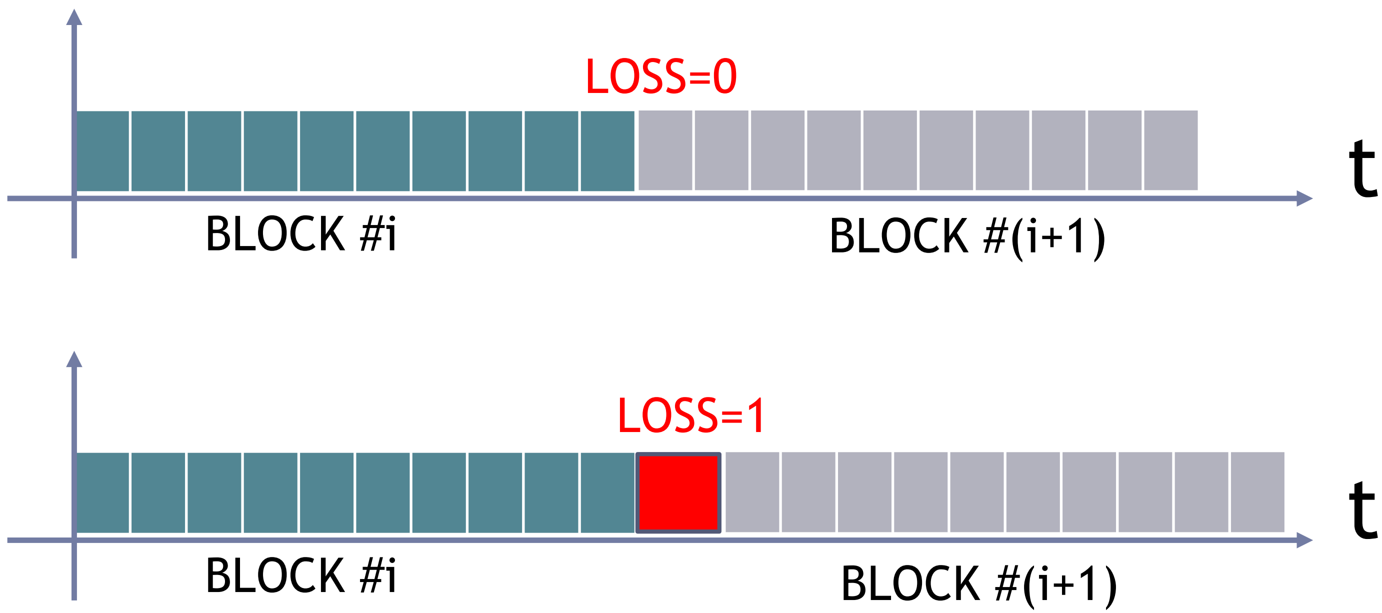
ACQISITION LOOP

SIGNAL ACQUISITION

SIGNAL STORAGE

LOSS OF SIGNAL

If blocks are not contiguous the LOSS flag is set to true



For spectral evaluation, even when averaging over multiple blocks, LOSS of signal continuity is not a problem

SPECTRUM EVALUATION

We can obtain the PSD, power spectral density, using Fourier transform: fft in matlab

We can apply the Fourier transform over the whole vector of samples

Let's set the frequency axis: which is our frequency resolution?

```
Df=SampleRate/(NsBlock*Nblocks)
f=[-SampleRate/2:Df: SampleRate/2-Df];
```

NUMBER OF SAMPLE
ACQUIRED

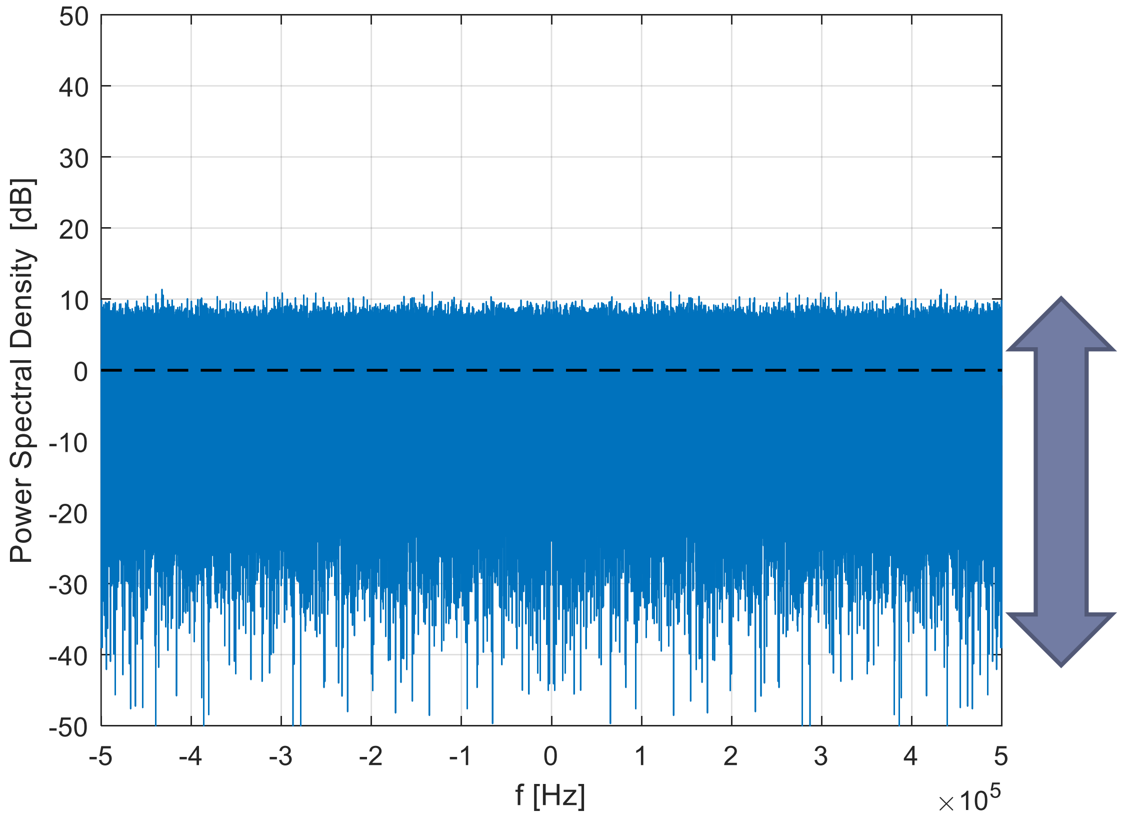
If Df is too small (because of an high number of samples), the PSD estimation is not very good

SPECTRUM EVALUATION: EXAMPLE

White Gaussian Noise – B_{SIM} = 1 MHz – $N_{samples}$=1e6

Level expected: 0 dB (black dashed line)

FFT Estimation 1e6 samples: blue line

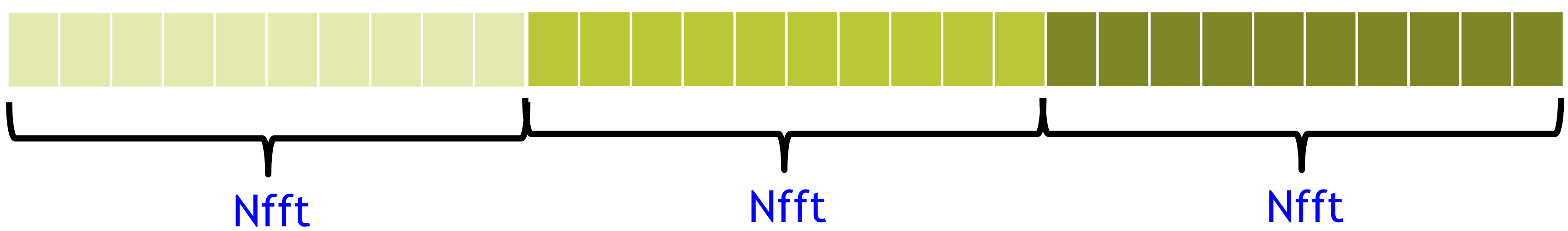


High resolution: 1 Hz
but
High uncertainity

SPECTRUM AVERAGING

We can improve PSD estimation by averaging over several fft evaluation

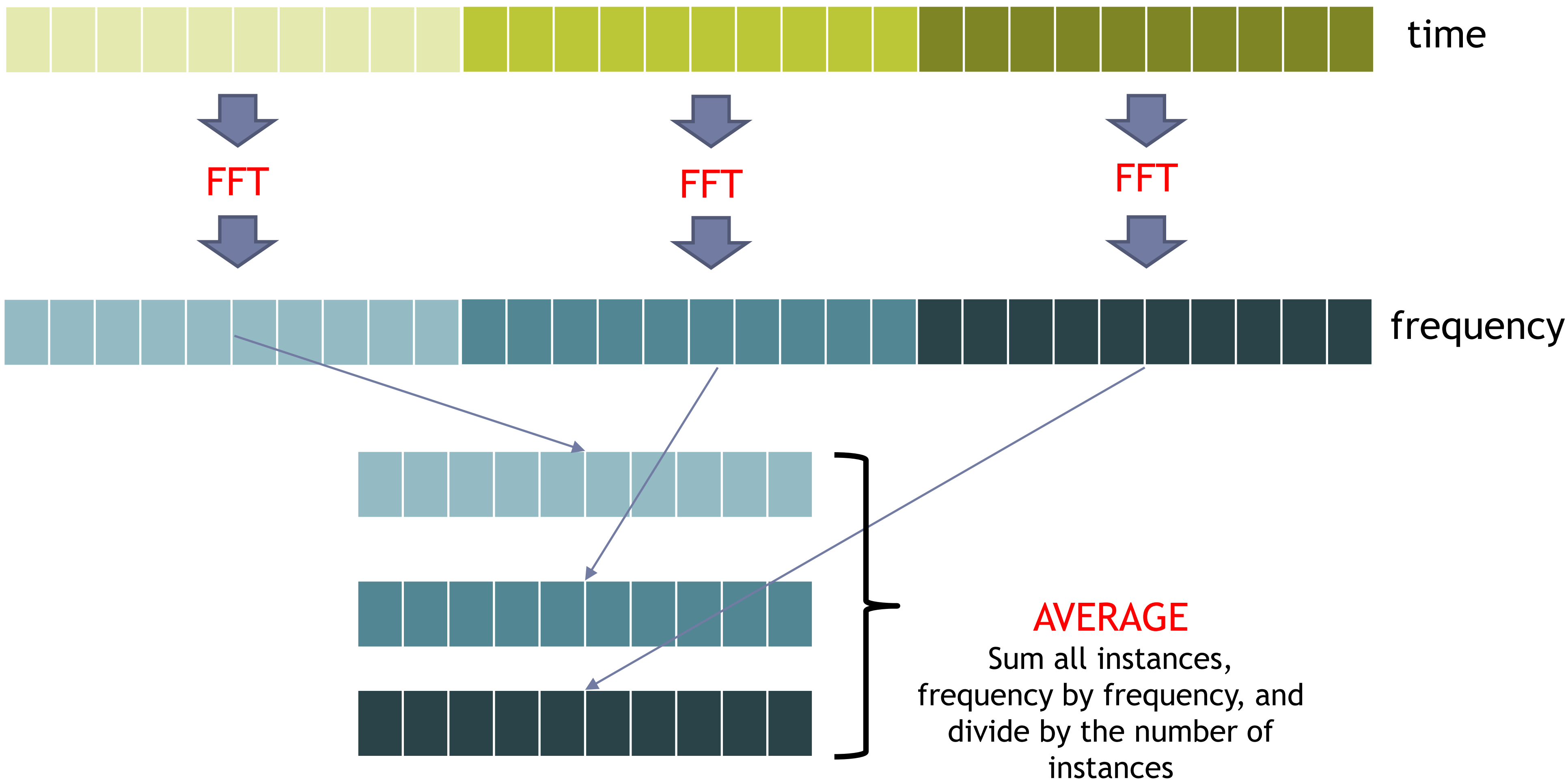
We divide the whole sample vector in short chunk (Nfft samples)




```
Df=SampleRate/Nfft
f=[-SampleRate/2:Df: SampleRate/2-Df];
```

Being Nfft smaller than the total number of samples we loos resolution, but we can take advantage of averaging multiple PSD estimation

SPECTRUM AVERAGING



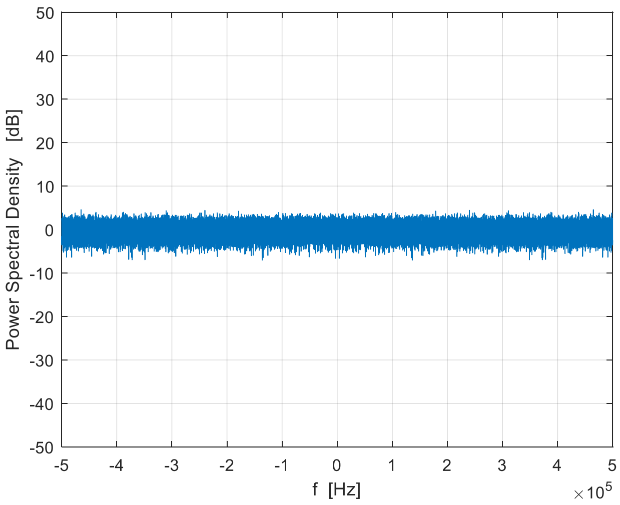
A simple choice fo Nfft could be the acquisition block length (NsBlock)

SPECTRUM AVERAGING: EXAMPLE

White Gaussian Noise – B_{SIM} = 1 MHz – $N_{samples}$=1e6

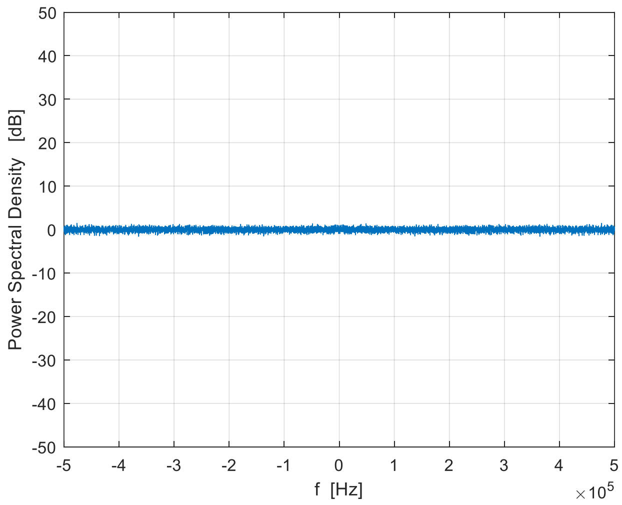
Level expected: 0 dB (black dashed line)

FFT Estimation over
1e5 samples
(Average over 10 blocks)



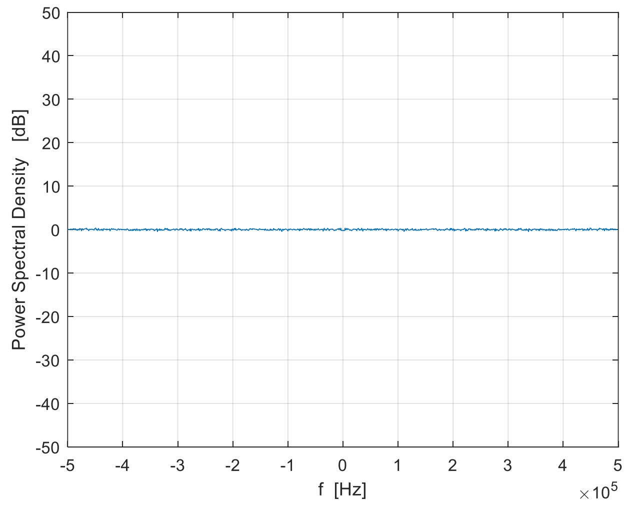
Resolution: 10 Hz

FFT Estimation over
1e4 samples
(Average over 100 blocks)



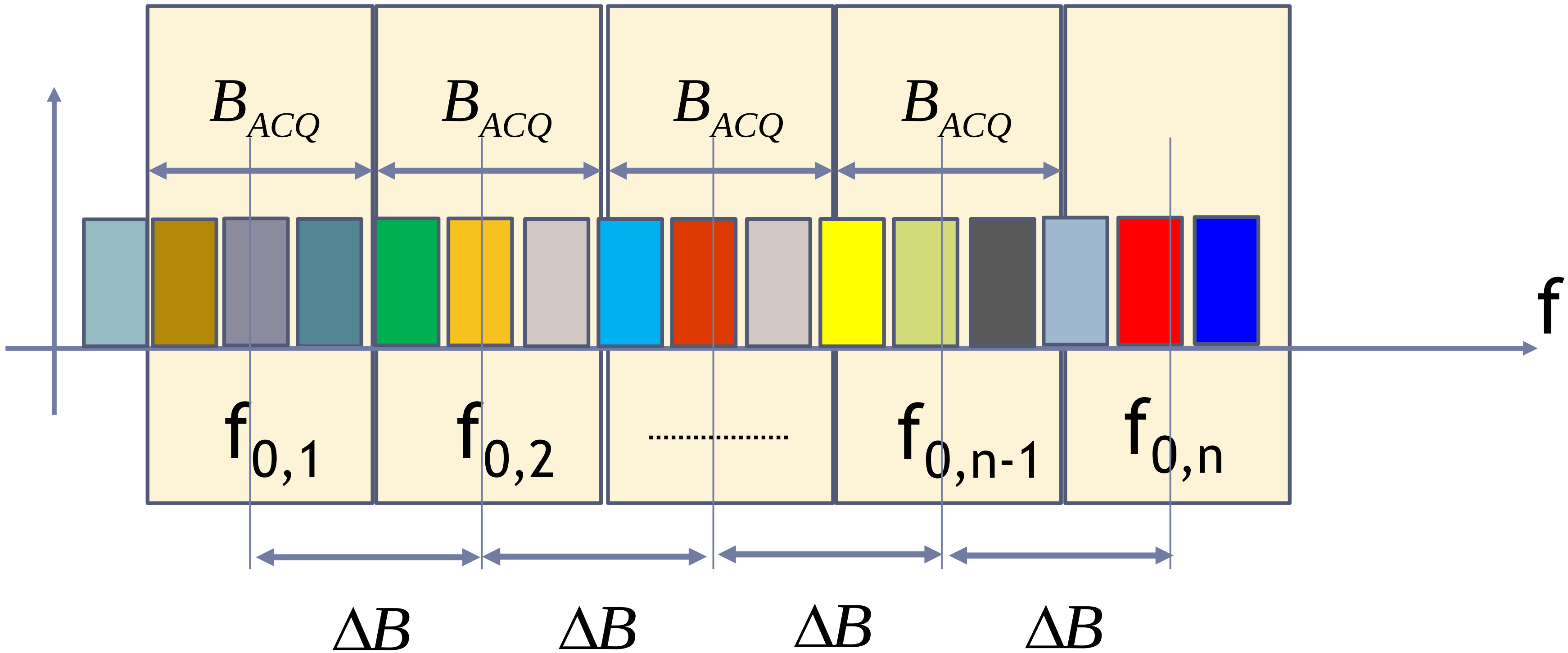
Resolution: 100 Hz

FFT Estimation over
1e3 samples
(Average over 1000 blocks)



Resolution: 1000 Hz

FREQUENCY SWEEPING



The simplest choice to completely capture a given range of frequency is to have $\Delta B = B_{ACQ}$

SDR INSTALLATION

1. Read instructions can be found at

http://it.mathworks.com/help/supportpkg/rtlsdrradio/ug/support-package-hardware-setup.html

2. Launch `supportPackageInstaller`

3. Look for the “Communications System Toolbox Support Package for RTL-SDR Radio” (filter using SDR)

4. Follow CAREFULLY instructions