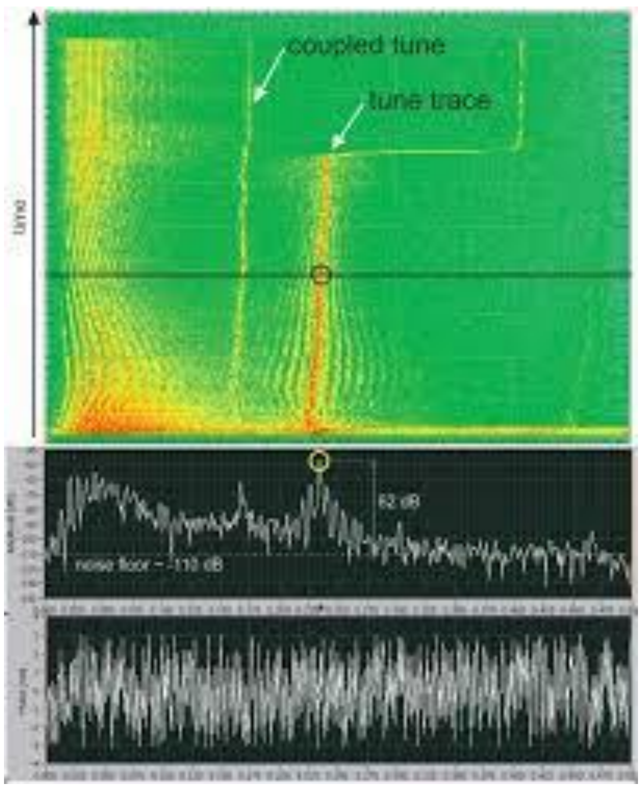
The CERN Accelerator School

General Introductory Course, Slovakia 2019

Time & Frequency Domain Measurements

H.Schmickler, CERN



Using several slides from:

M.Gasior (CERN)
R.Jones (CERN)
T.Lefevre (CERN)
H. Damerau (CERN)
S.Zorzetti (FNAL)

Part I

- Introduction: What Is time domain and frequency domain?

- Fourier synthesis and Fourier transform
- Time domain sampling of electrical signals (→ ADCs)
- Bunch signals in time and frequency domain

  a) single bunch single pass
  b) single bunch multi pass (circular accelerator)
  c) multi bunch multi pass (circular accelerator)

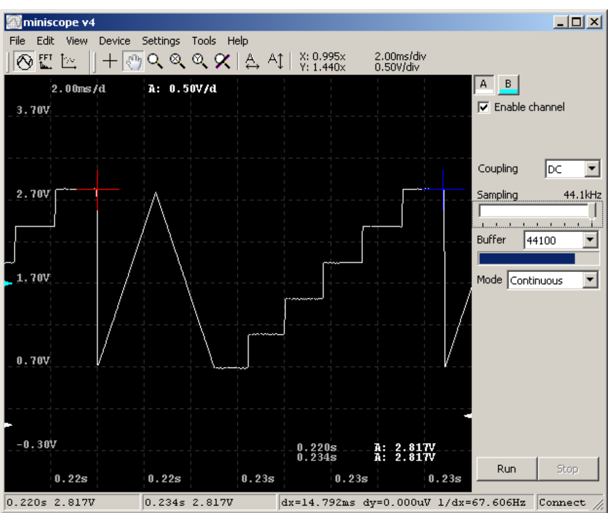
- Oscillations within the bunch (head-tail oscillations)
  → advanced course

Part II

- Fourier transform of time sampled signals
    - a) basics
    - b) aliasing
    - c) windowing
- Methods to improve the frequency resolution
    - a) interpolation
    - b) fitting (the NAFF algorithm)
    - c) influence of signal to noise ratio
    - d) special case: no spectral leakage + IQ sampling
- Analysis of non stationary spectra:
    - STFT (:= Short time Fourrier transform) (Gabor transform)
      also called: Sliding FFT, Spectogram
    - wavelet analysis
    - PLL tune tracking

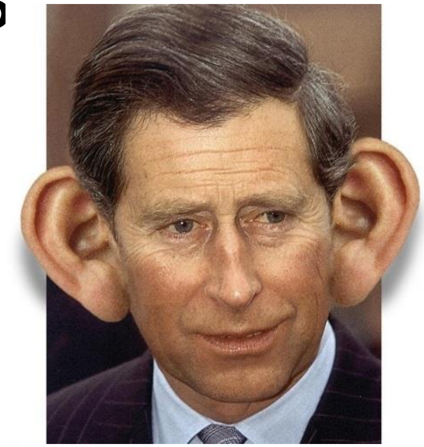
Introduction 1/3

- At first: everything happens in time domain, i.e. we exist in a 4D world, where 3D objects change or move as a function of time.
- And we have our own sensors, which can watch this time evolution: eyes → bandwidth limit: 1 Hz
- For faster or slow processes we develop instruments to capture events and look at them: oscilloscopes, stroboscopes, cameras…

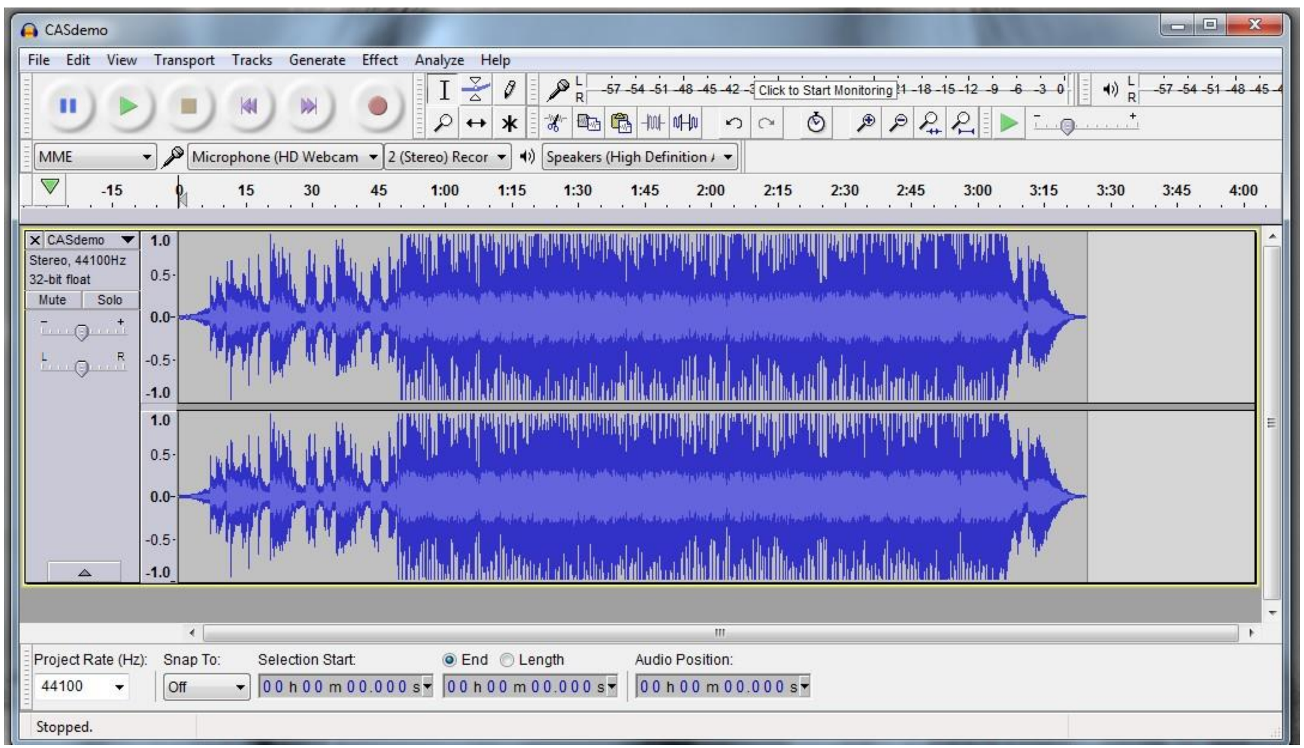


Introduction 2/3

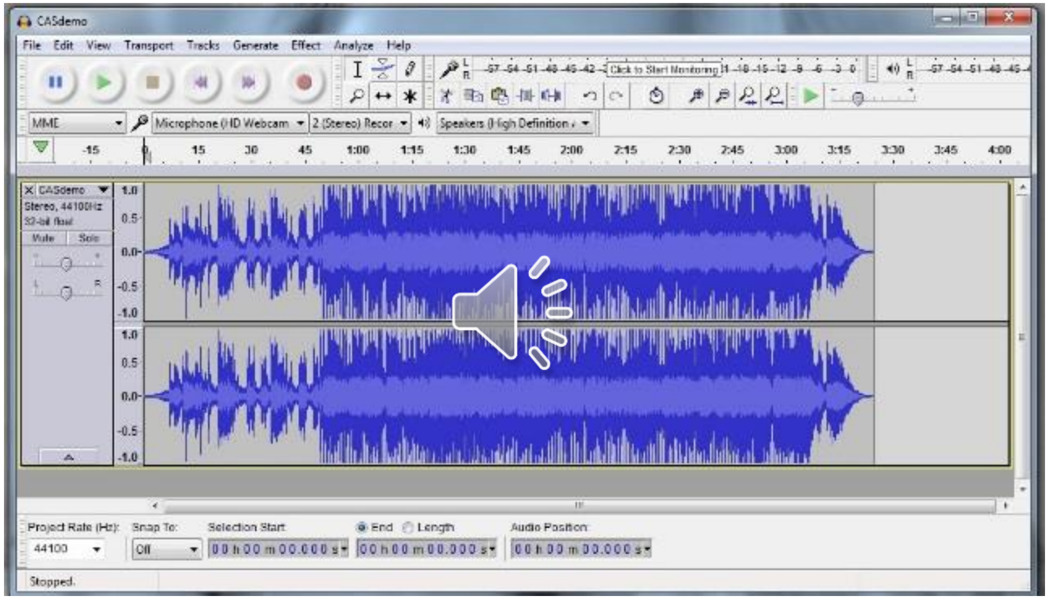
- But we have another sensor: ears



- What is this?



Introduction 3/3



- Once we perceive the material in frequency domain (our brain does this for us), we can better understand the material.

- **Essential:**

Non matter whether we describe a phenomenon in time domain or in frequency domain, we describe the same physical reality. But the proper choice of description improves our understanding!

Jean Baptiste Joseph Fourier (1768-1830)

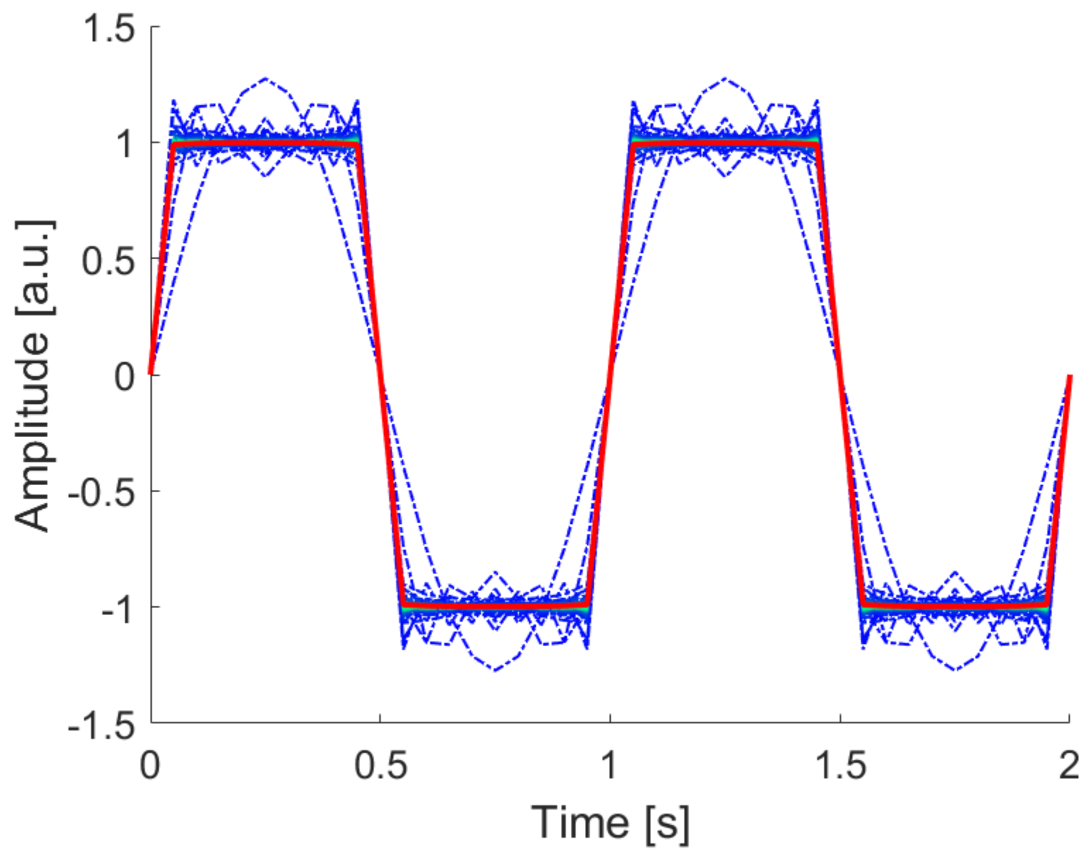
- Had crazy idea (1807):
- **Any** periodic function can be rewritten as a weighted sum of Sines and Cosines of different frequencies.
- Don’t believe it?
  - Neither did Lagrange, Laplace, Poisson and other big wigs
  - Not translated into English until 1878!
- But it’s true!
  - called Fourier Series
  - Possibly the greatest tool used in Engineering

Fourier Synthesis

A periodic function *f(x)* can be expressed as a series of harmonics, weighted by Fourier coefficients c_n

$$f(x) = \sum_{n=-\infty}^{n=+\infty} c_n e^{-2i\pi \frac{n}{T} x}$$

$$c_n = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} f(x) e^{-2i\pi \frac{n}{T} x} dx$$



Fourier Transform

- Also, defined as:

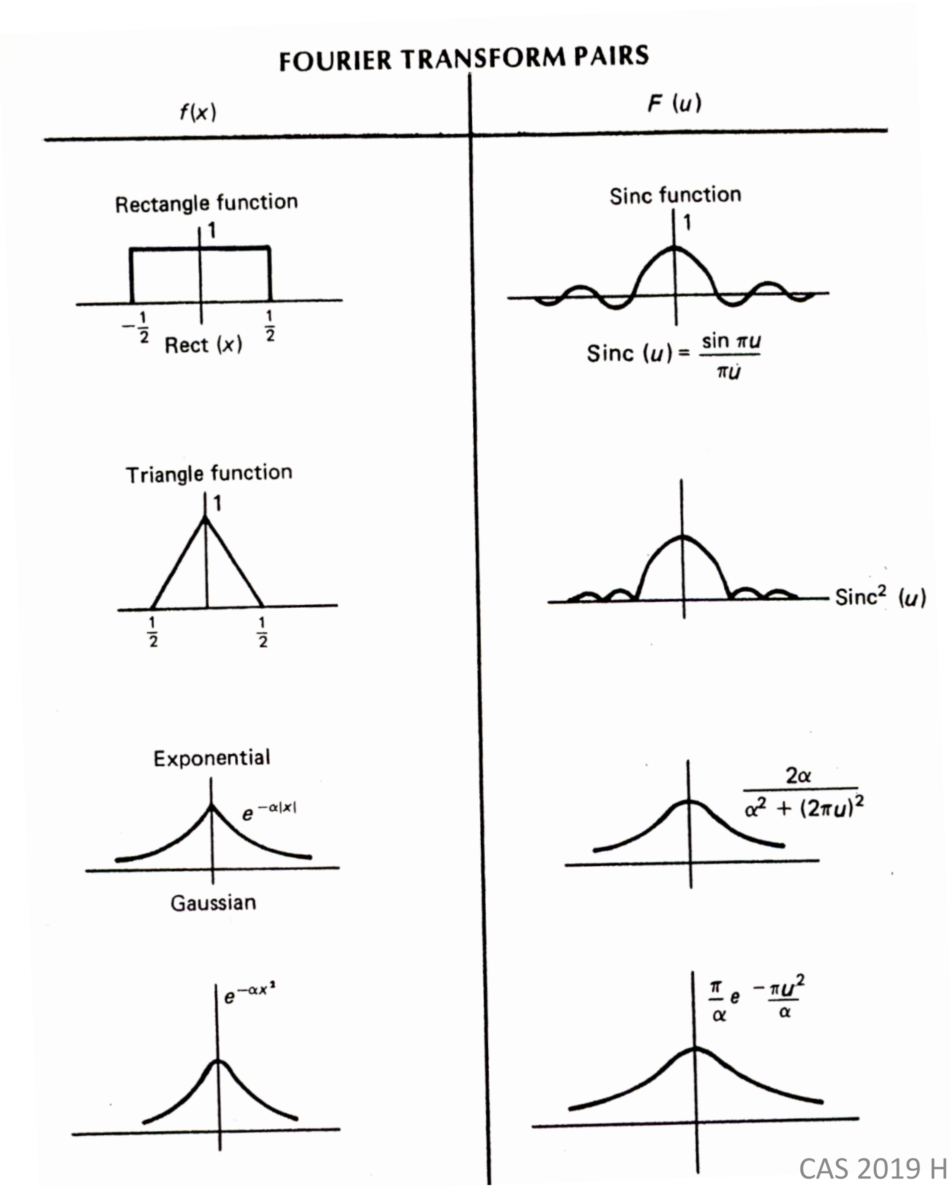
$$F(u)=\int_{-\infty}^{\infty} f(x)e^{-iux}dx$$

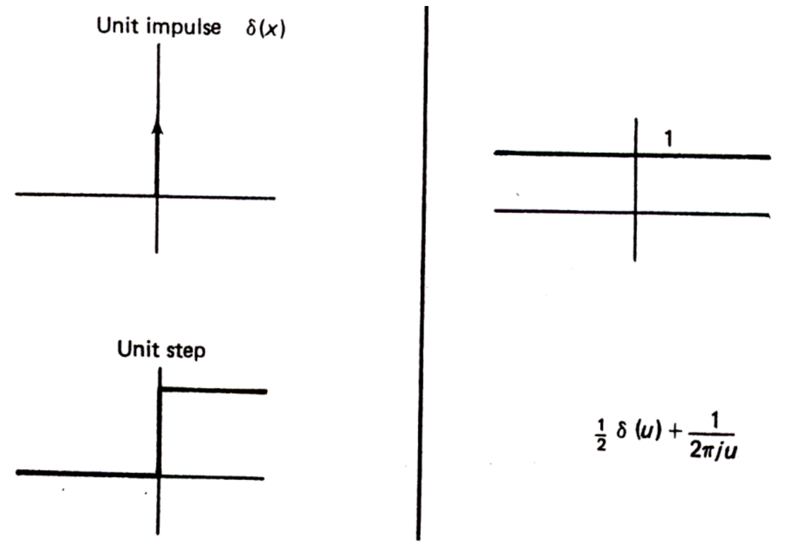
Note: $e^{ik}=\cos k+i\sin k \qquad i=\sqrt{-1}$

- Inverse Fourier Transform (IFT)

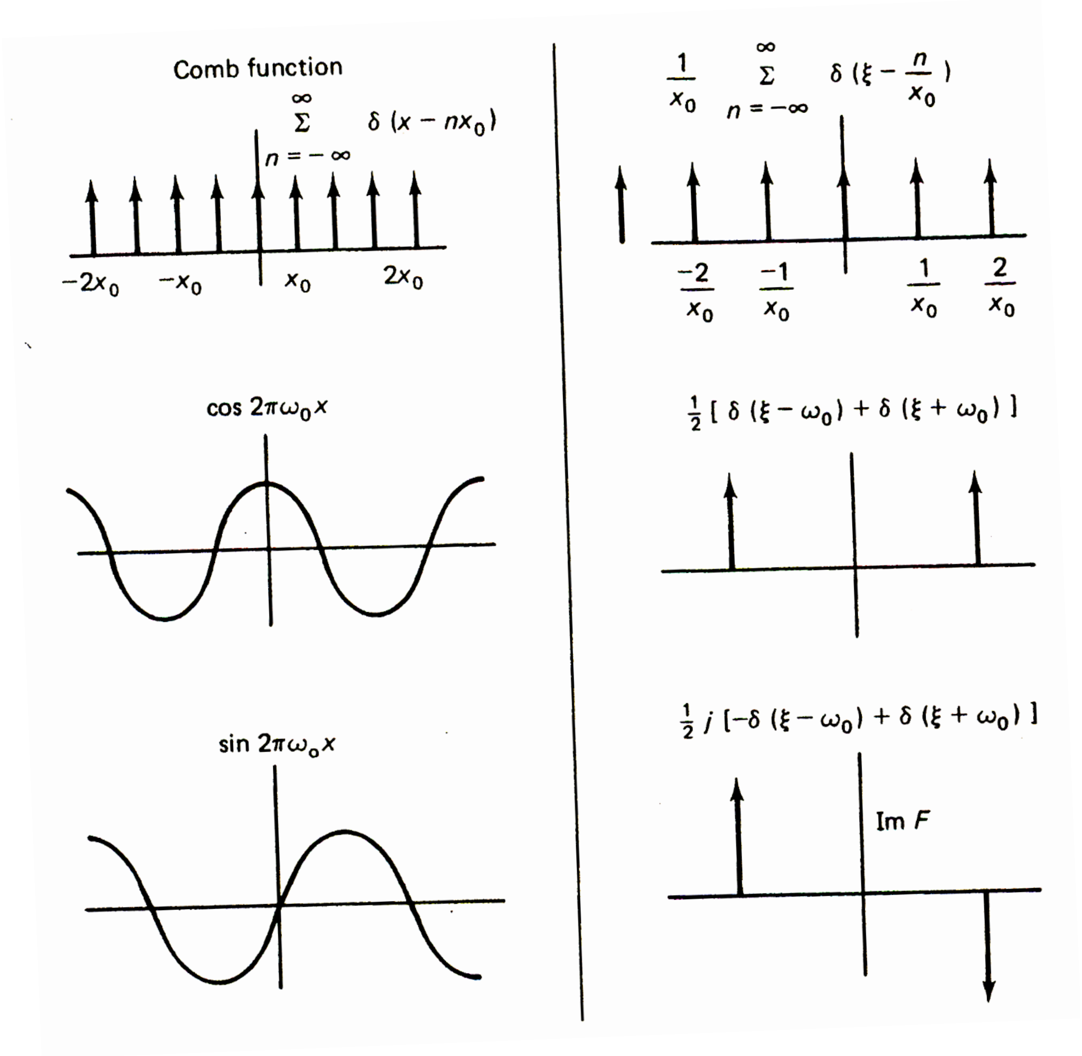
$$f(x)=\frac{1}{2\pi}\int_{-\infty}^{\infty} F(u)e^{iux}dx$$

Fourier Transform Pairs (I)

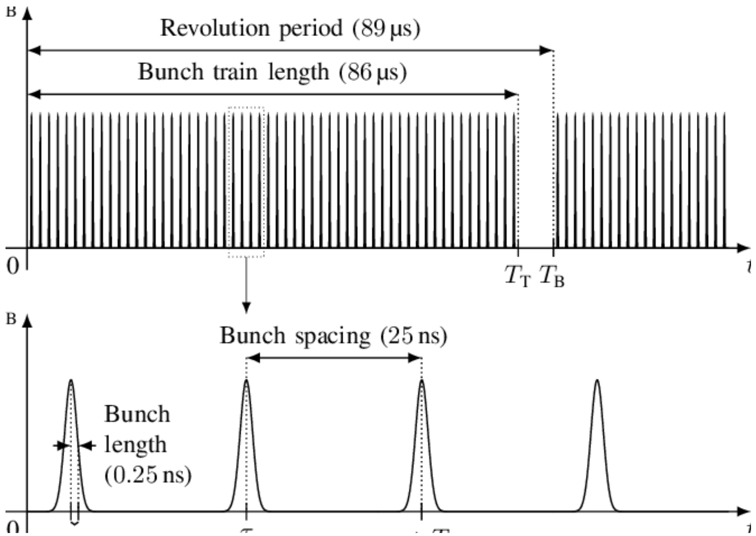




Fourier Transform Pairs (II)



Definitions



In real accelerators not all available RF-buckets are filled with particle bunches.
- a gap must be left for the injection/extraction kickers
- Physics experiments can impose a minimum bunch distance, which is larger than one RF period (i.e. LHC)

Revolution frequency: $\omega_{rev} = 2\pi\, f_{rev}$

RF frequency: $\omega_{RF} = 2\pi\, f_{RF} = h^*\, \omega_{rev}$ (h=harmonic number)

Bunch Repetition frequency: $\omega_{rep} = 2\pi\, f_{rep} = \omega_{rev}\,/n$ (n= number of RF buckets between bunches)
(f_{rep} = 1/bunch spacing)

Nominal LHC Filling Scheme

"Standard Filling Schemes for Various LHC Operation Modes", R. Bailey and P. Collier, ***LHC-Project-note-323*****.**

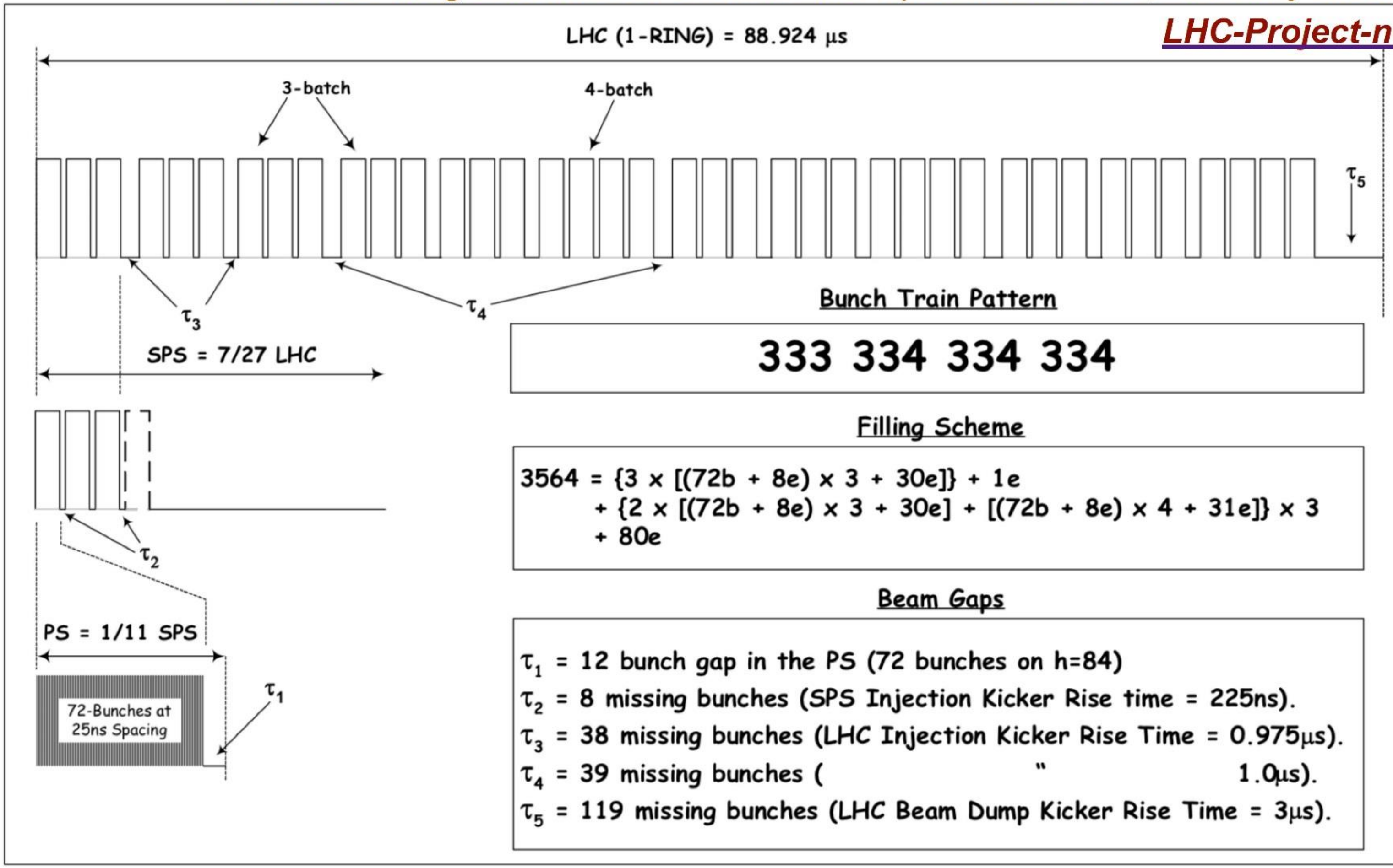


Figure 1: Schematic of the Bunch Disposition around an LHC Ring for the 25ns Filling Scheme

The CERN Accelerator School

Understanding beam signals in time and frequency domain

We start with:

Single bunch single pass

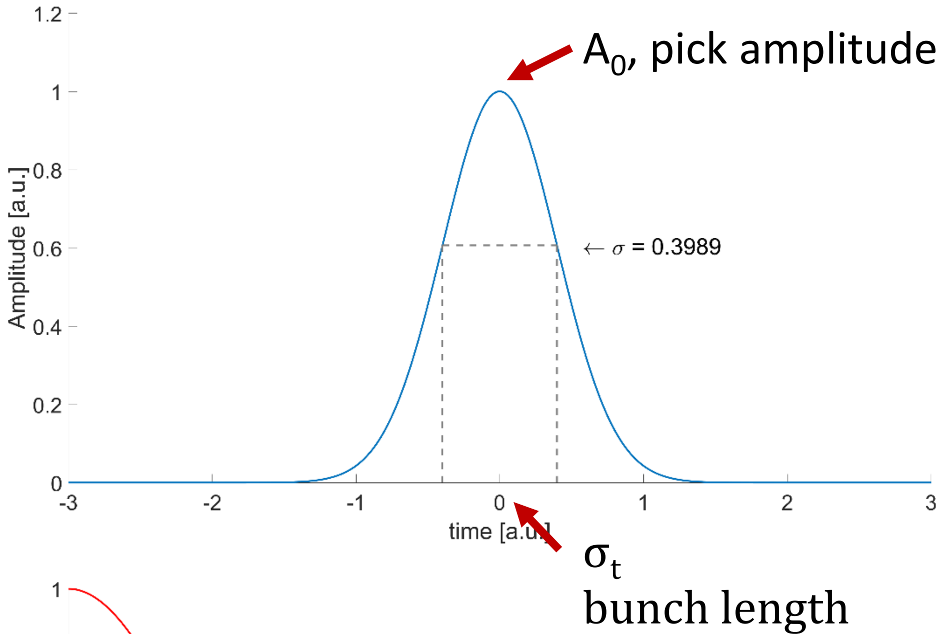
- Time and frequency domain description
- Measurement of bunch length in time domain
  → Sampling electrical signals with ADCs
- Measurement of bunch length in frequency domain

Particle beam with gaussian longitudinal distribution

Time domain

$$f(t) = A_0 \exp\left(-\frac{t^2}{2\sigma_t^2}\right)$$

$$area = \int_{-\infty}^{+\infty} f(t)dt = \sqrt{2\pi}A_0\sigma_t$$

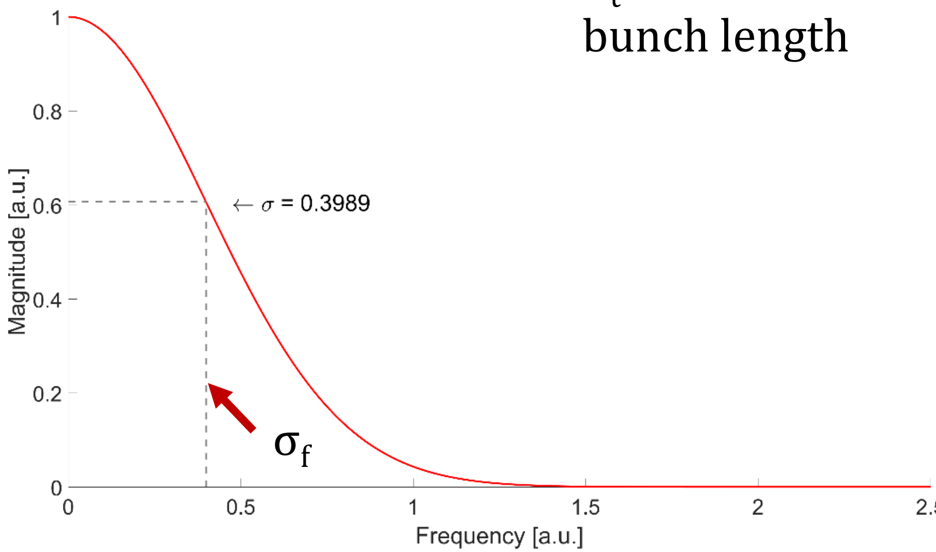


Frequency domain

$$F(k) = \frac{A_0}{\sqrt{2\pi}\sigma_f} \exp\left(-\frac{k^2}{2\sigma_f^2}\right)$$

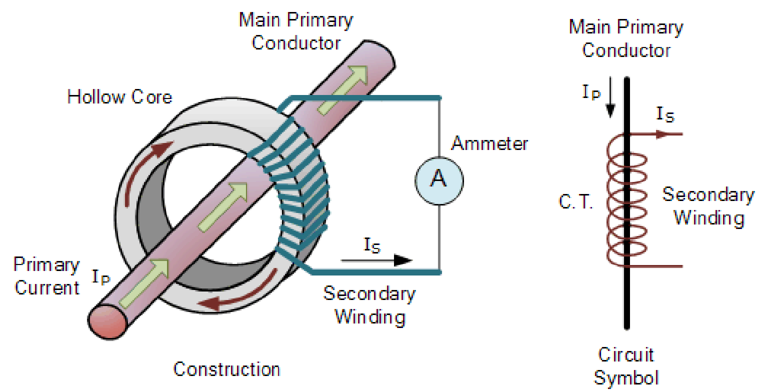
$$\sigma_f = \frac{1}{2\pi\sigma_t}$$

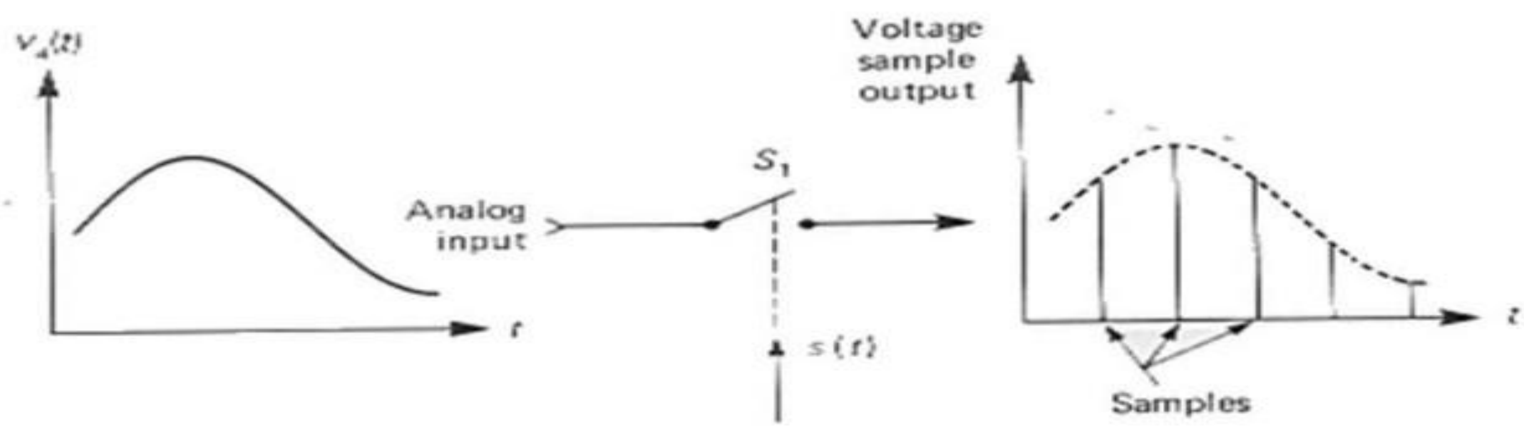
$$F(0) = area = \frac{A_0}{\sqrt{2\pi}\sigma_f} = \sqrt{2\pi}A_0\sigma_t$$



Time domain measurement of single bunch

- Sampling (=measurement) of an electrical signal in regular time intervals. The electrical signal is obtained from a monitor, which is sensitive to the particle intensity.





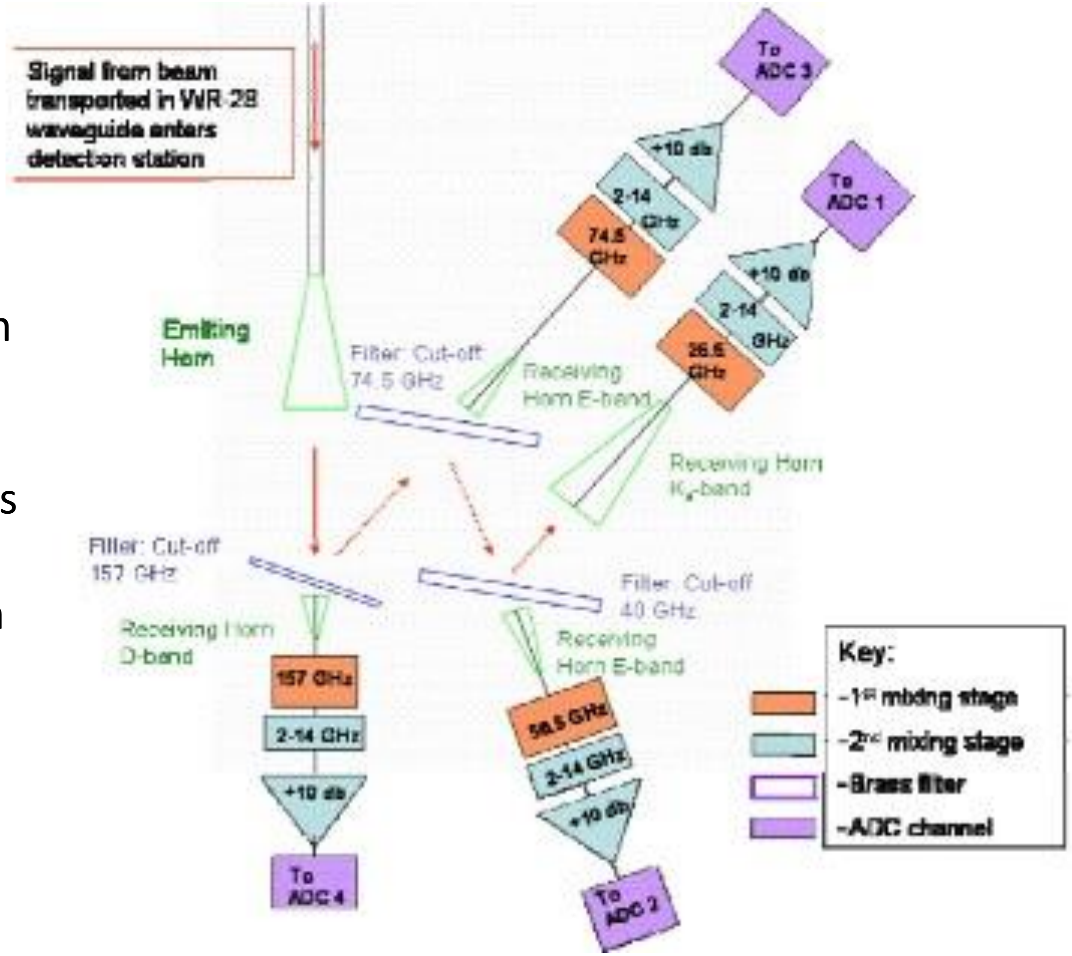
Frequency domain measurement of single bunch

Nice example from R&D work in CTF3 (CERN)
A.Dabrowski et al., Proc of PAC07, FRPMS045

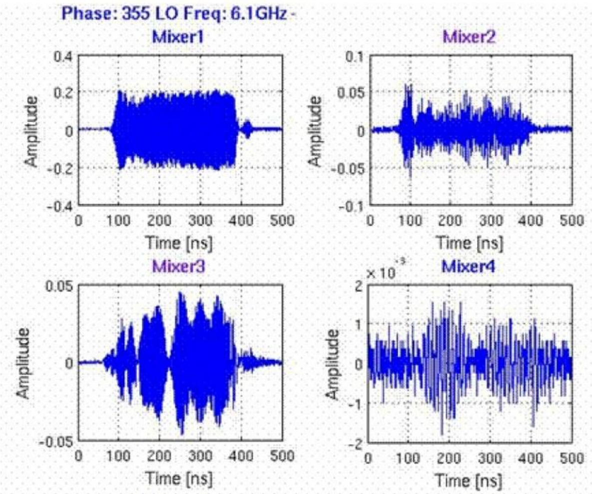
Primary signal is EM wave of beam extracted through a thin window

Subdivision into 4 frequency bands

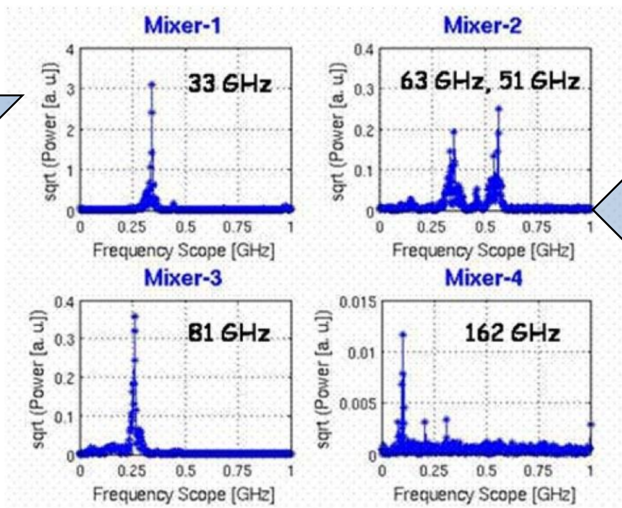
Measurement of rms amplitude in the 4 bands



CTF3 results



Time domain measurements of 4 bands



FFT of down-converted signals

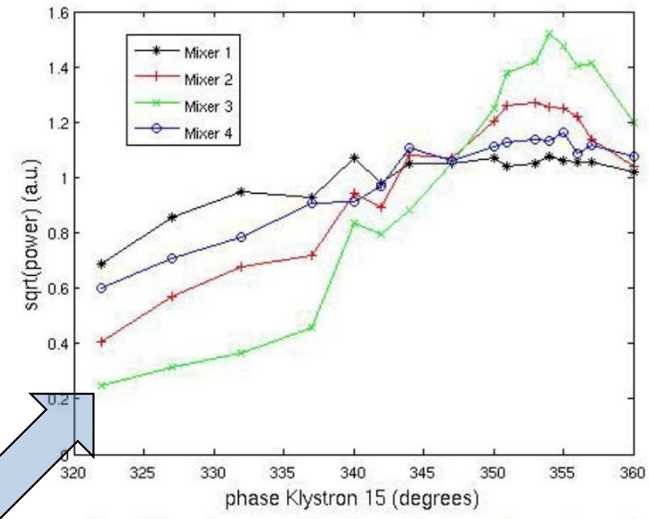


Figure 5: Signal amplitudes from the 4 selected frequencies as a function of the phase in Klystron 15.

fit

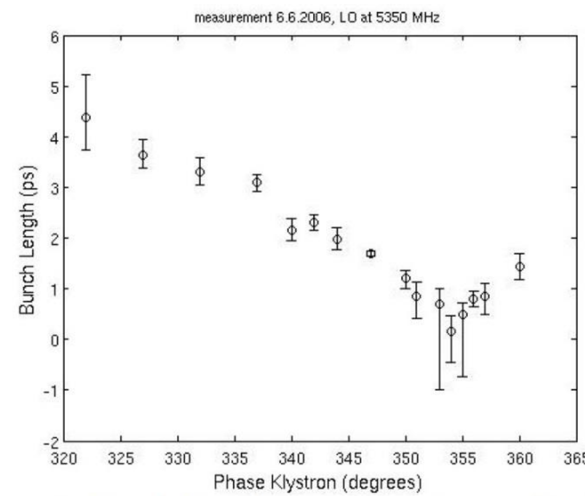
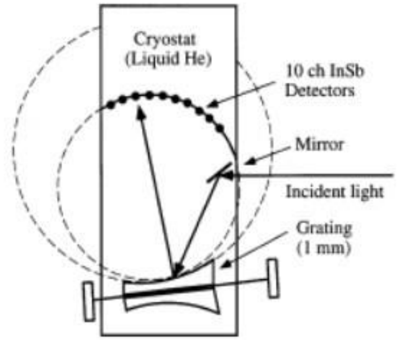


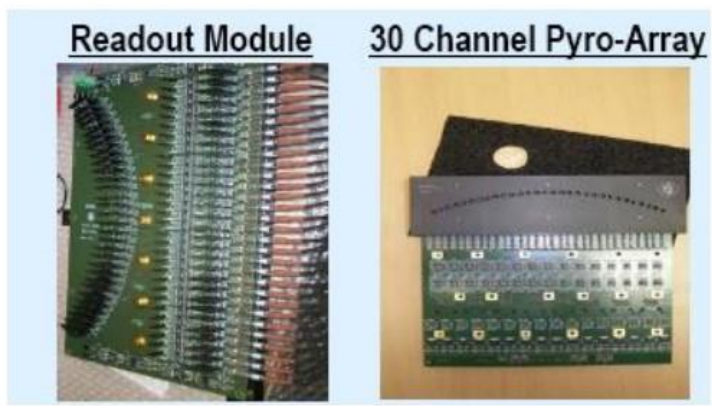
Figure 6: Bunch length measurements as a function of the phase of Klystron 15

Another example (DESY) based on optical radiation

1 *'The **polychromator** enables to get the spectrum directly by a single shot. The radiation is deflected by a grating and resolved by a multi-channels detector array'*



T. Wanatabe et al., NIM-A 480 (2002) 315-327
H. Delsim-Hashemiet al., Proc. FLS, Hamburg 2006, WG512

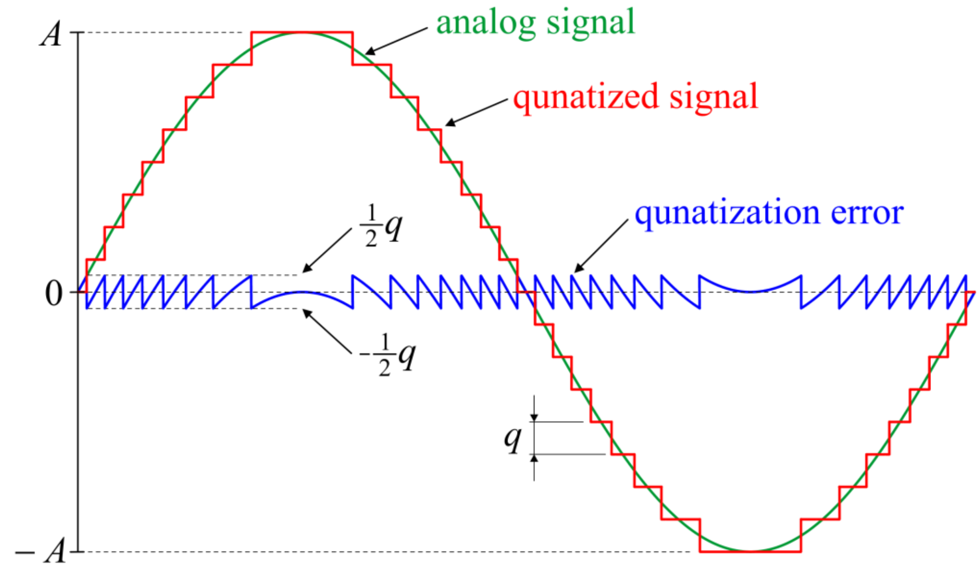


B. Schmidt, DESY

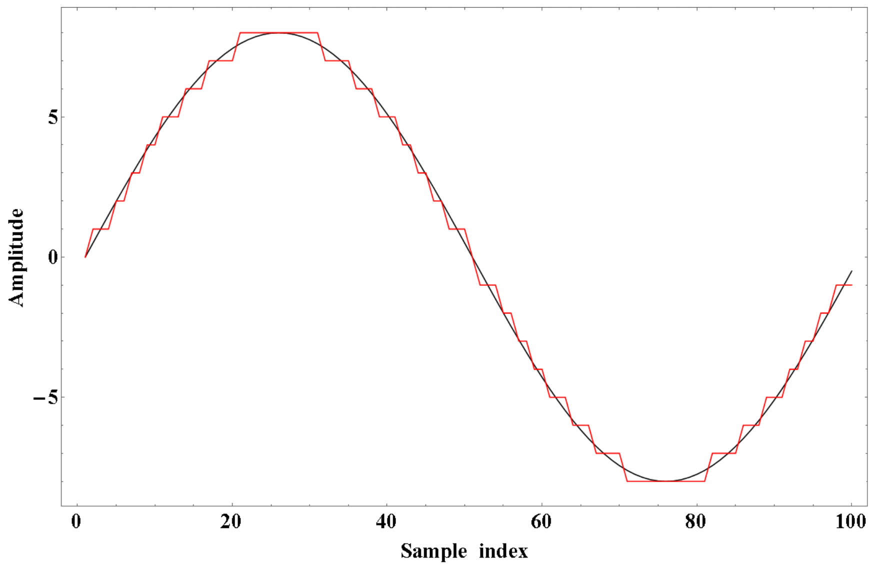
Slide taken from T.Lefevre (CERN)

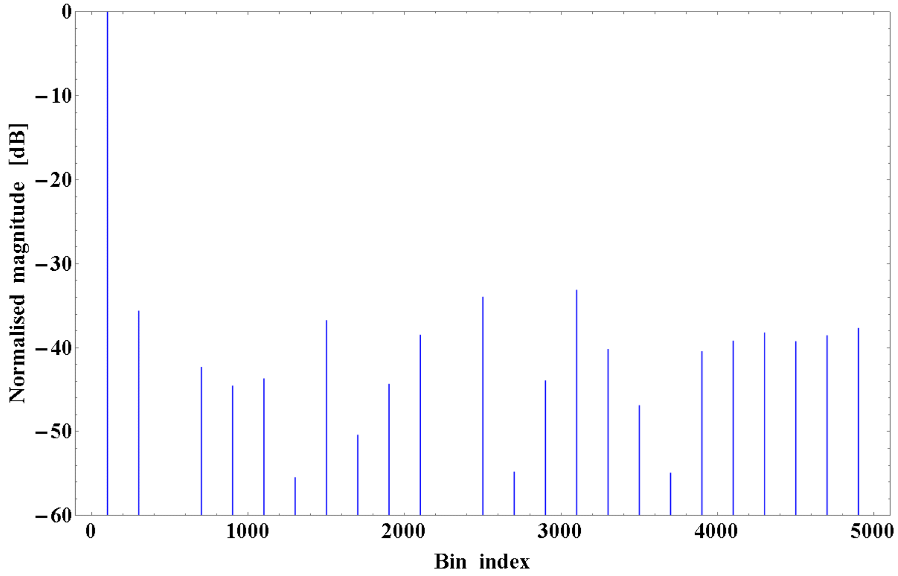
Quantization error

- Here we consider only an ideal quantization of a continuous signal (no sampling). Quantized signal is an approximation o the input signal; their difference is the quantization noise.
- Used quantities:
  - *A* – input signal amplitude
  - *n* – number of bits
  - *q* – one bit amplitude: $q = \frac{2A}{2^n}$
- Max quantization error: $e_m = \frac{\pm q}{2}$
- RMS amplitude of the input signal: $A_{RMS} = \frac{A}{\sqrt{2}}$
- Quantisation error RMS amplitude: $e_{RMS} = \frac{1}{\sqrt{12}} q \cong 0.289\, q$
- Signal to Noise Ratio: $SNR = \frac{A_{RMS}}{e_{RMS}} = \frac{\sqrt{6}}{2} 2^n \qquad SNR\,[\mathrm{dB}] = 20 \log_{10} \frac{\sqrt{6}}{2} 2^n \cong 1.76 + 6.02\, n$
- Effective Number of Bits: $ENOB = \frac{SNR\,[\mathrm{dB}] - 1.76}{6.02}$



Quantization error

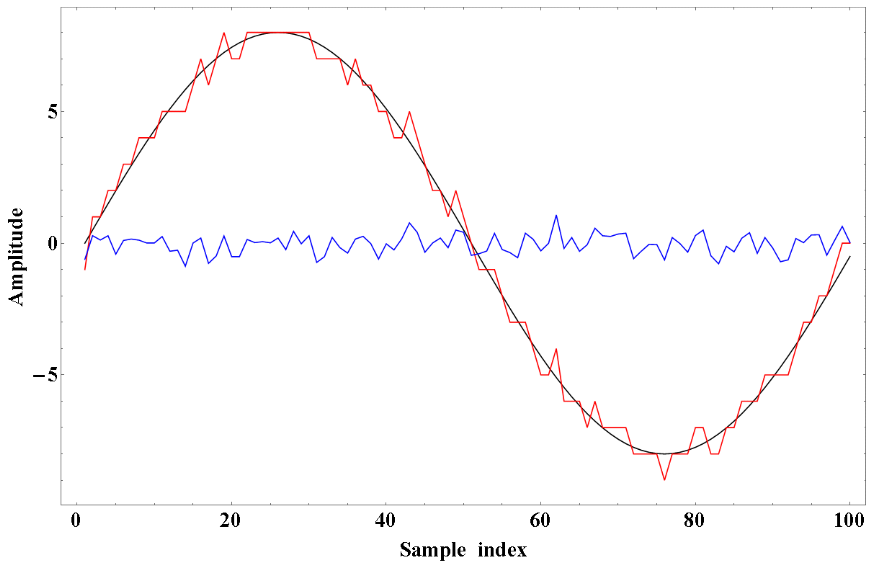


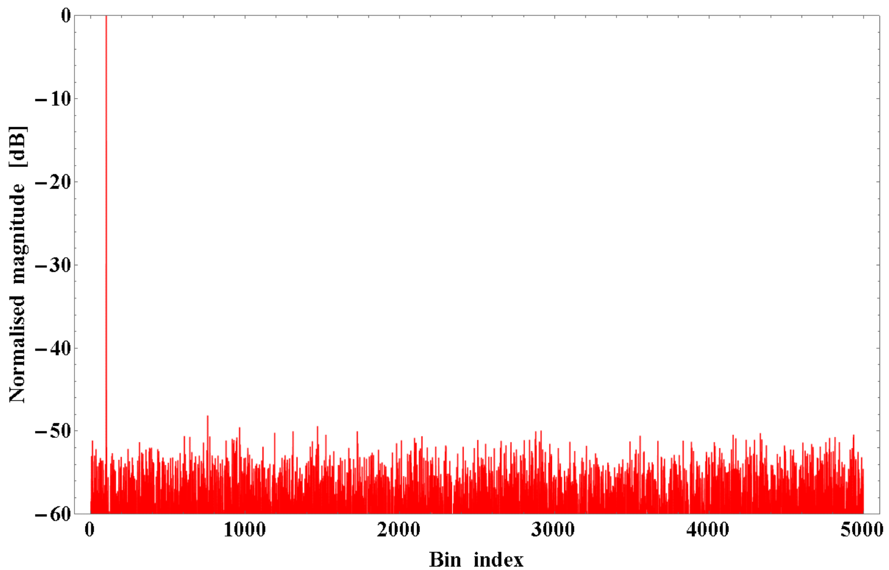


- N = 10000 samples of a full scale 4-bit sine
- $f_{in} = 0.01\ f_s$ (100 samples per s_{in} period)
- 100 samples of 1 period shown, corresponding to 1 % of the whole signal

- only one component expected, at N × f_{in} / f_s, that is 100th bin
- Other components have levels in the order of – 40 dB, that is about 1 % of the fundamental

Quantization error and dither





- As before, but added a small noise (blue) to the input signal, of RMS amplitude 0.4 q

- Now only one component seen at the expected location
- Noise floor seen at the level in the order of – 55 dB, that is about 0.18 % of the fundamental

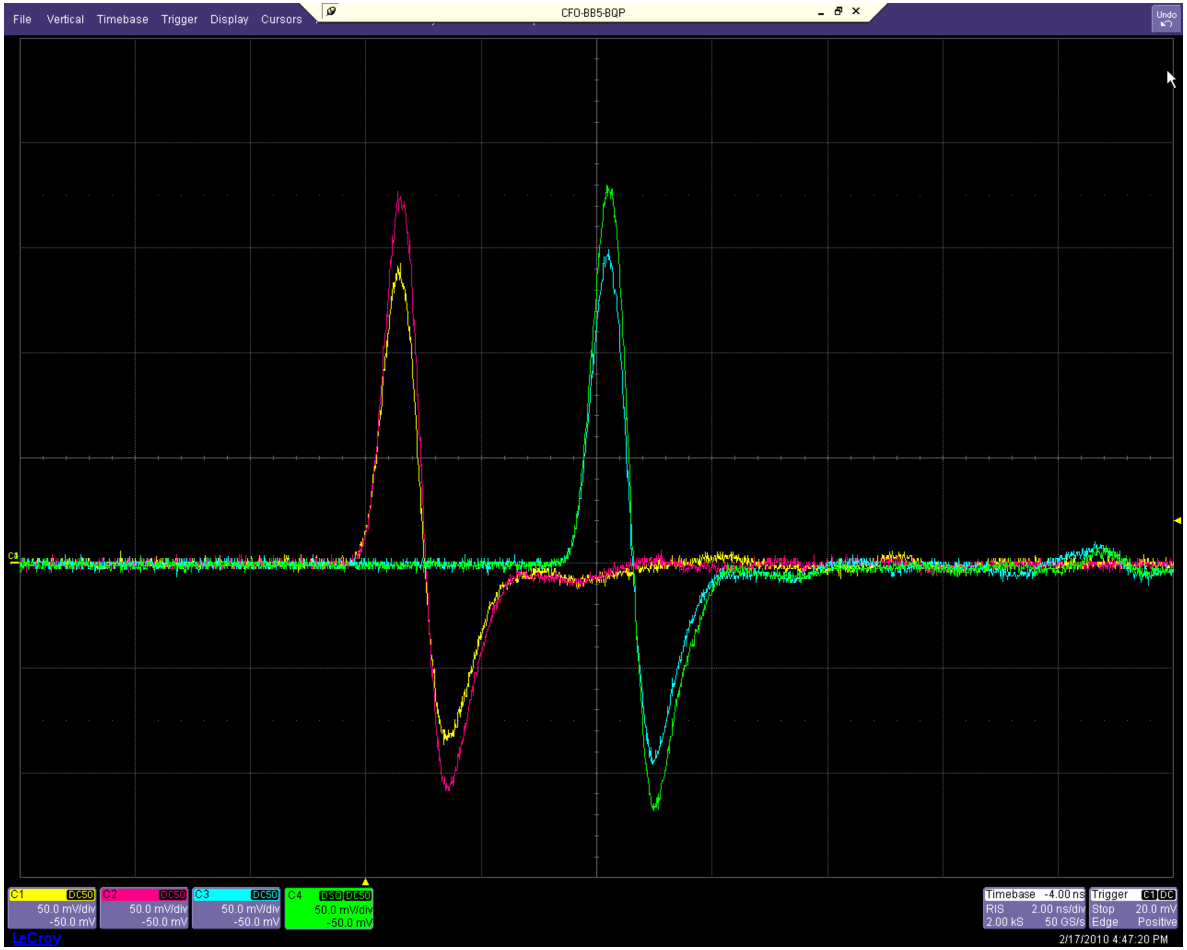
ADC's: Further considerations

- So far we looked at the digitization of continuous signals
- Beam signals are different:
  - usually short pulses
  - shape of pulses changes due to beam dynamics
  - good idea is to "look" at these signals with analogue means before using digitized information

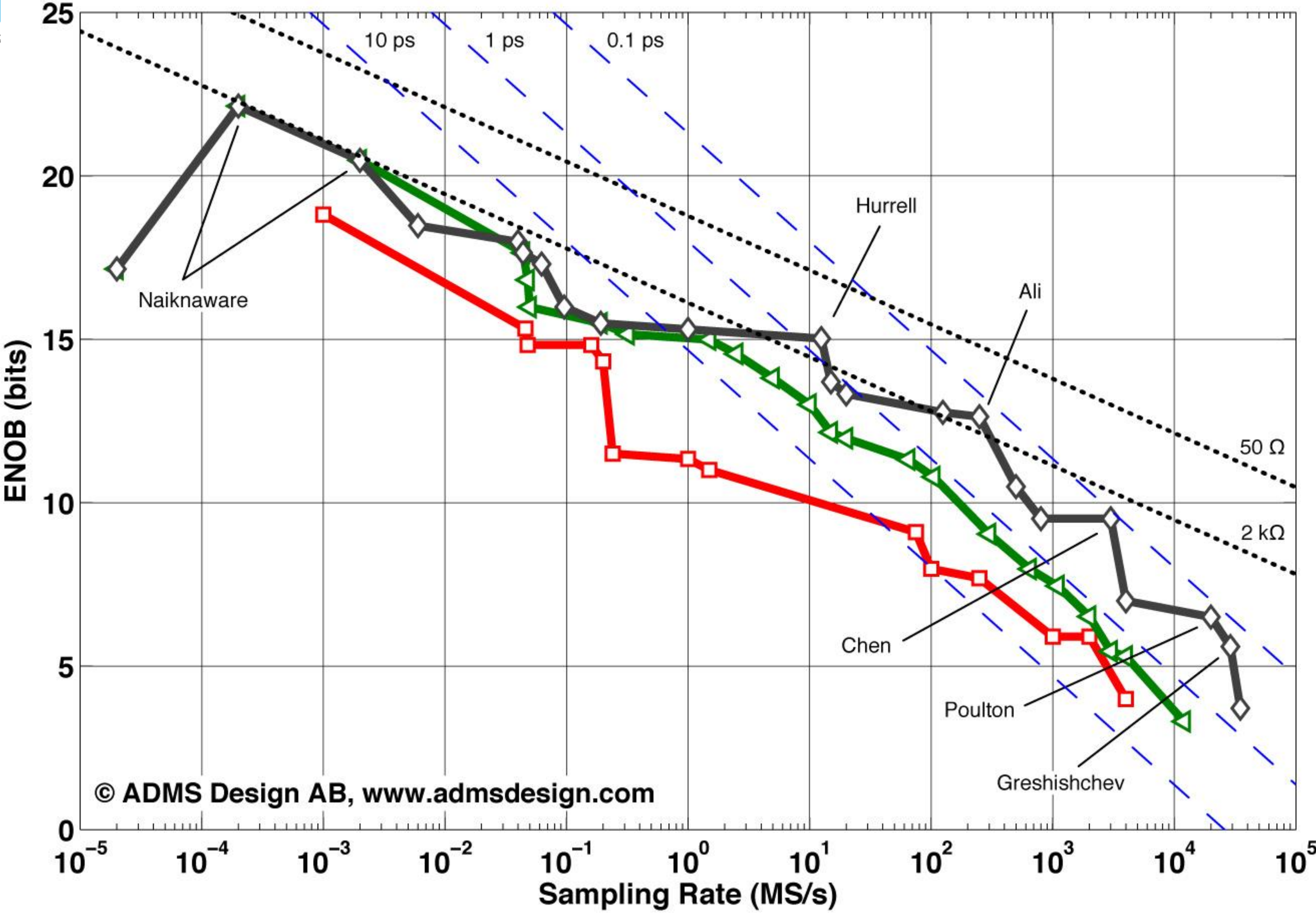
- Criteria/buzzwords to design an ADC system:
  - required resolution → number of bits
  - required bandwidth → sampling frequency
  - stability/synchronicity of ADC clock → clock jitter
  - signal level → use full scale of ADC
  - noise contribution: shielding, low impedance signals (low thermal noise)

The CERN Accelerator School

Sampling a pulse



- 50 mV/div, 2 ns/div
- SPS beam
- 2 pairs of 10 mm button electrodes
- Signals already "filtered" by quite long cables



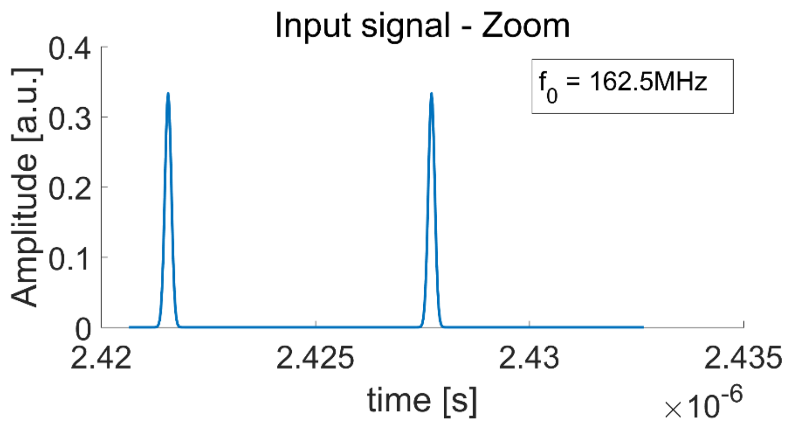
10 ps
1 ps
0.1 ps
Hurrell
Ali
Naiknaware
50 Ω
2 kΩ
Chen
Poulton
Greshishchev
© ADMS Design AB, www.admsdesign.com
ENOB (bits)
Sampling Rate (MS/s)

Single bunch multi pass (circular accelerator)

Time domain

$$f(t) = \sum_{n=1}^{N} A_0 \exp(-\frac{(t-nT)^2}{2\sigma_t^2})$$

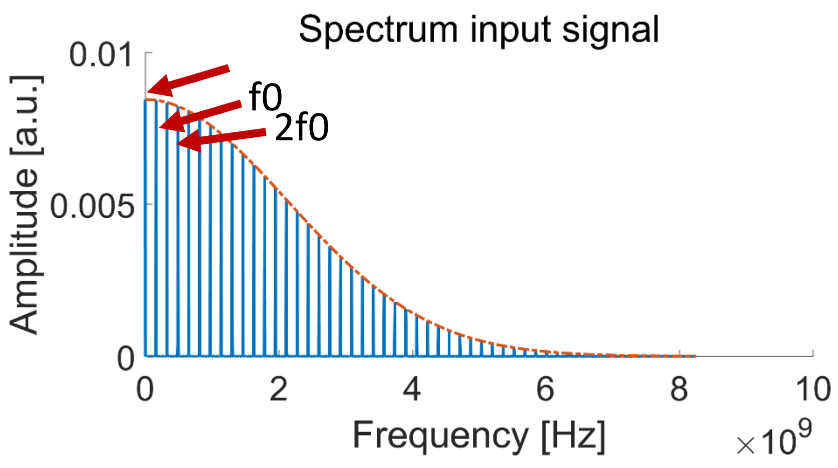
$$area = \int_{-\infty}^{+\infty} f(t)dt = N \times \sqrt{2\pi} A_0 \sigma_t$$



Frequency domain

$$F(k) = \sum_{i=1}^{N} F_c(ik_0) \exp\left(-\frac{(k-ik_0)^2}{2\sigma_f^2}\right),$$

$$\sigma_f = \frac{1}{2\pi\sigma_t}$$



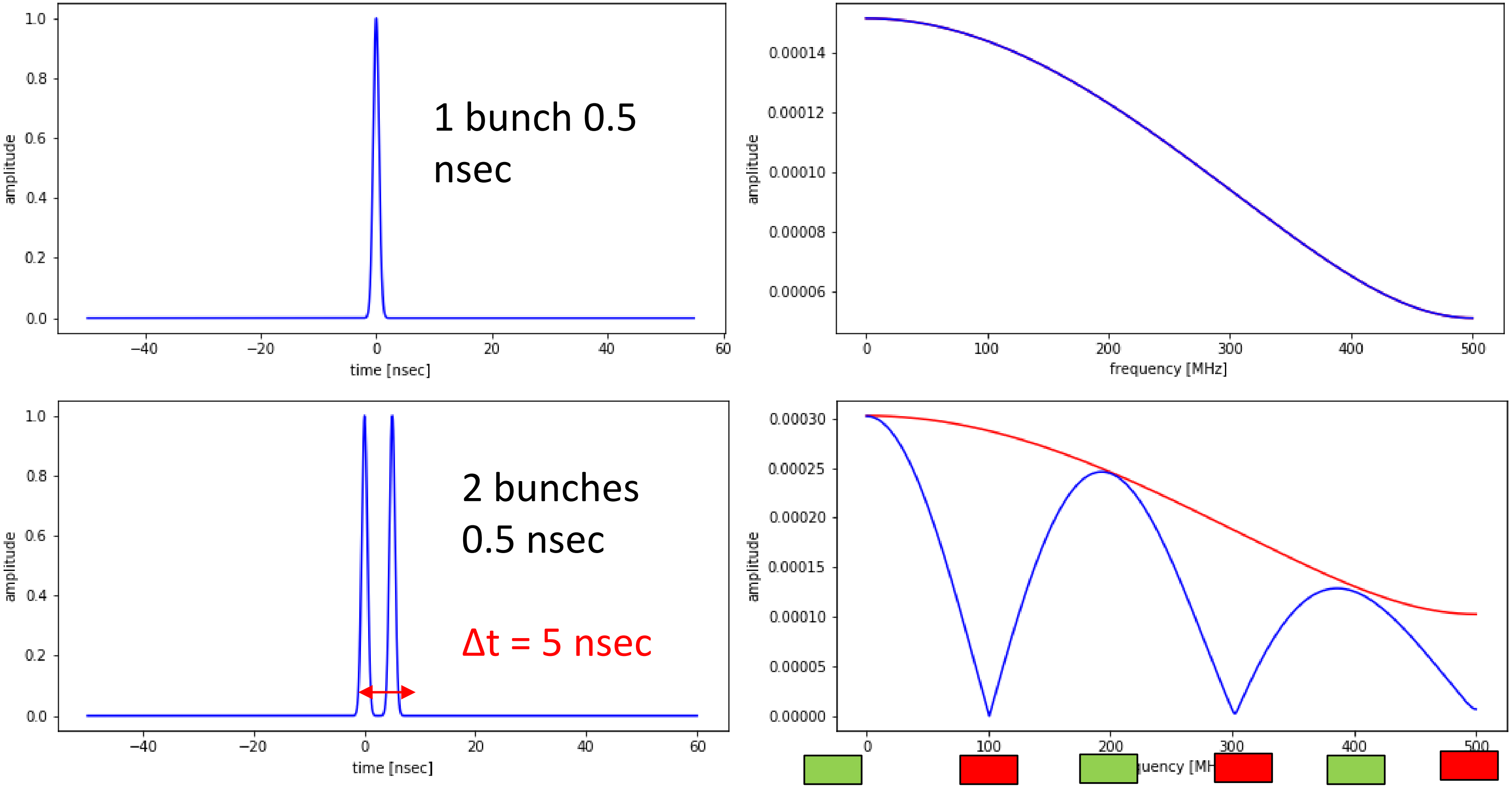
Understanding the revolution harmonics

- The continuous spectrum of a single bunch passage becomes a line spectrum.
- The line spacing is $f_{rev} = 1/T_{rev}$. (T_{rev} = revolution time)
- The amplitude envelope of the line spectrum is the "old" single pass frequency domain envelope of the single bunch.

- Why?
  - short answer: Do the Fourier transform!
  - long answer:

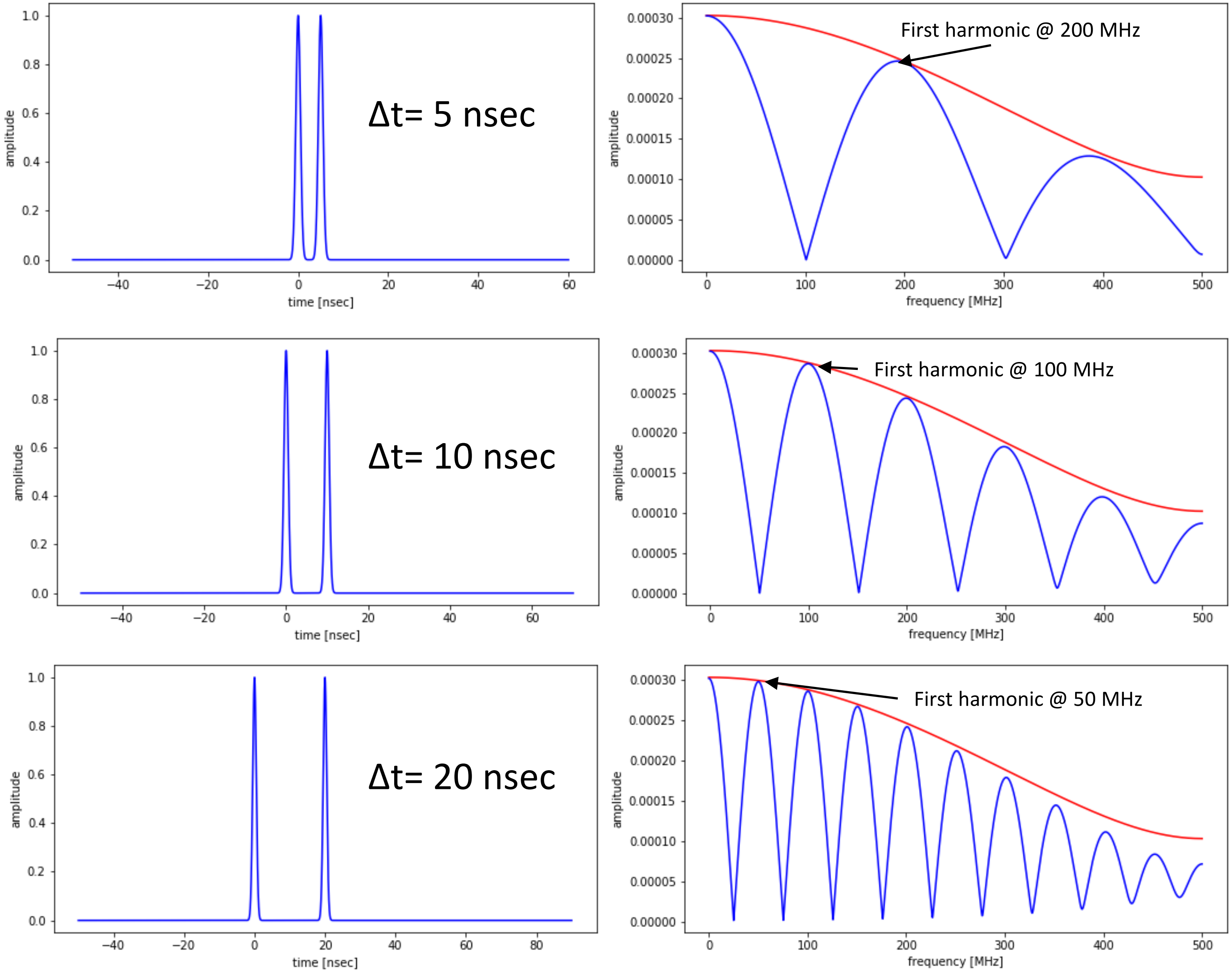
  Understand in more detail 2,3,4…N consecutive bunch passages in time and frequency domain (next slides)

Bunch pattern simulations (1/4)

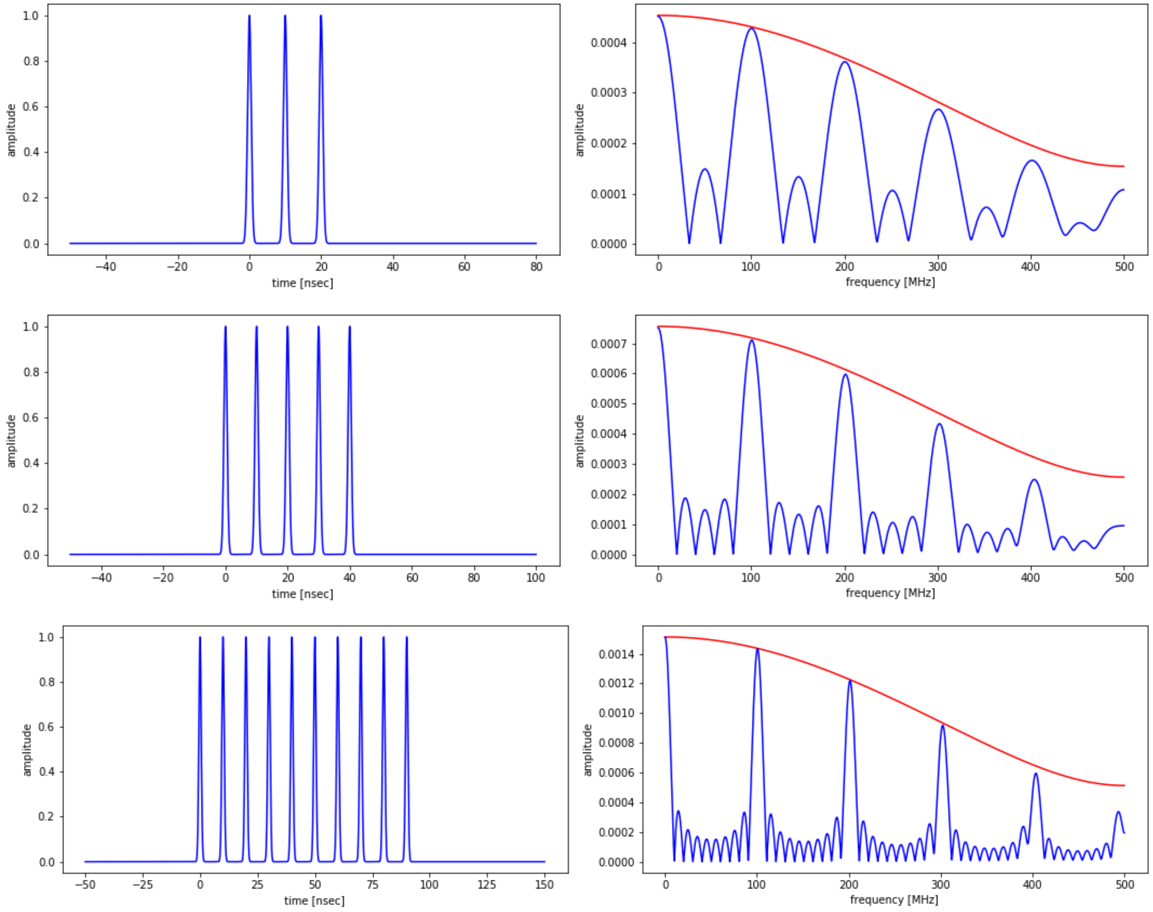


- Frequencies in this range make a constructive interference (no phase difference)
- Frequencies in this range cancel each other (180^0 phase difference)
- Other frequencies intermediate summation/cancelation

Bunch pattern simulations (2/4)

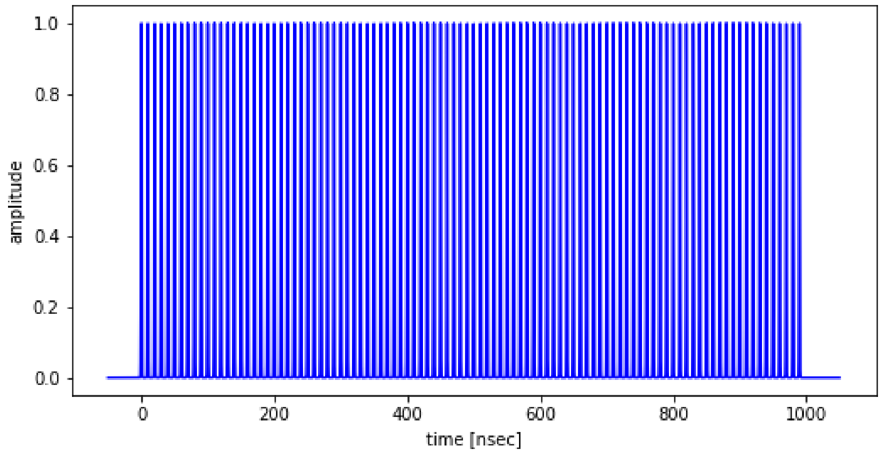


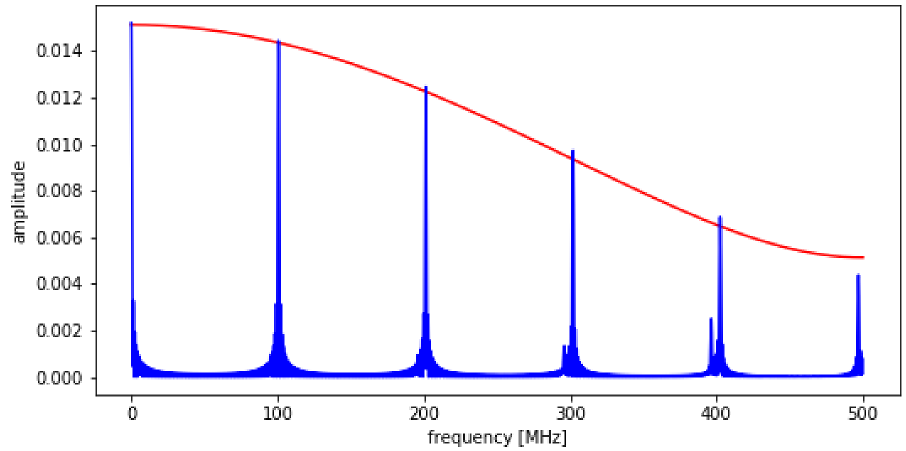
Bunch pattern simulations (3/4)



From top to bottom:
3, 5, 10 bunches (0.5nsec long, Δt = 10 nsec)

Last bunch pattern simulation





- 100 equidistant bunches (Δt = 10 nsec)
- Resulting spectrum is a line spectrum with the fundamental line given by the inverse of the bunch distance

A Measured Longitudinal beam spectrum

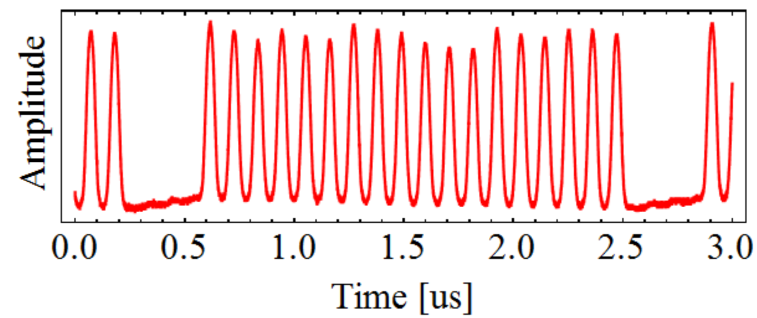
- **Circular accelerator**
  - → **Beam signal periodic with revolution frequency:** ω_{rev}

→ **Spectral components at:**

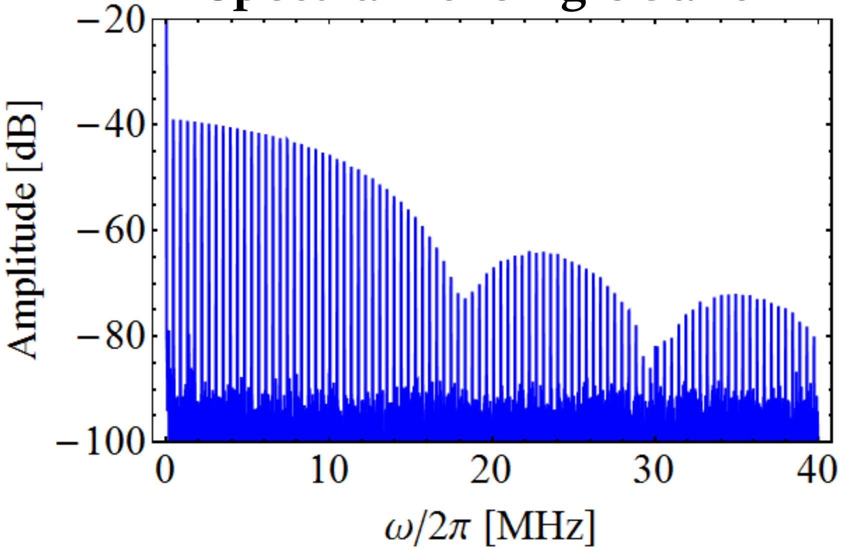
$$\omega = n\omega_{\text{rev}}$$

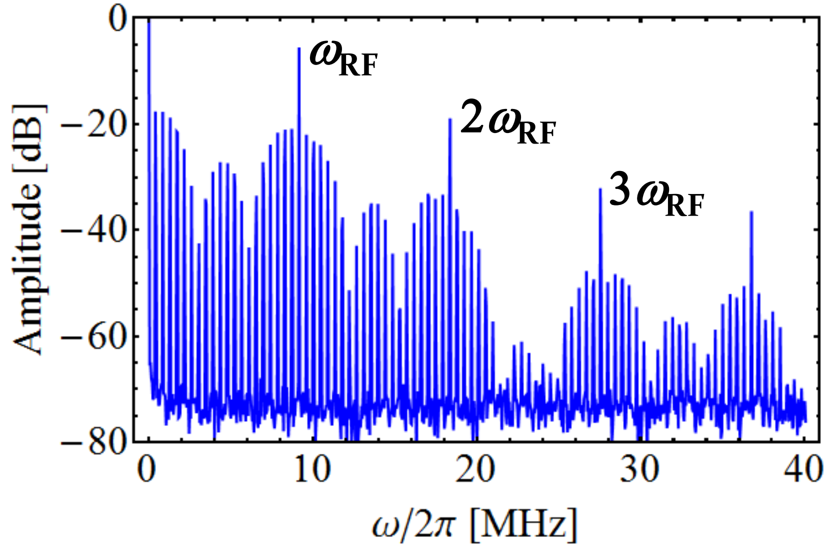
Bunch not Gaussian.
Somewhat between triangular and parabolic

Multi-bunch beam

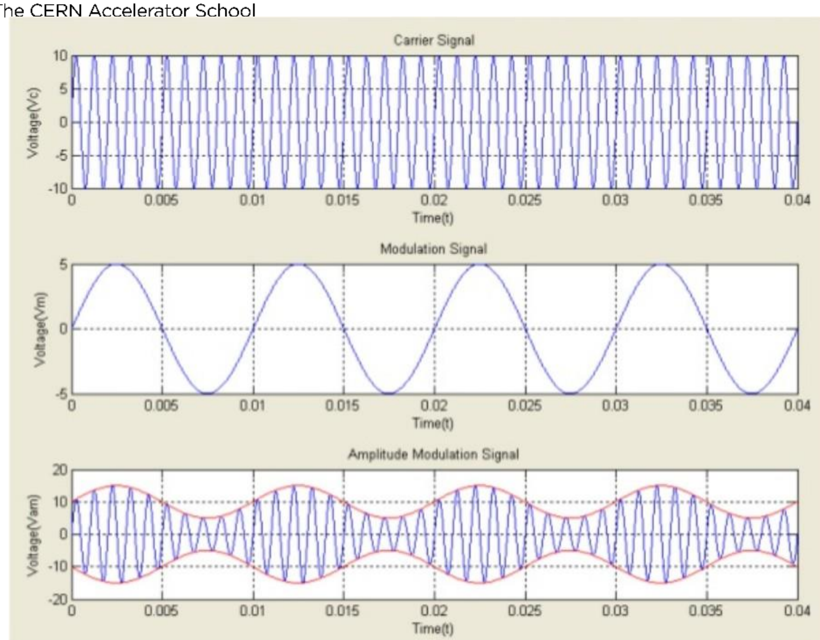


Spectrum of single bunch





Amplitude modulation



$$v = V_{env} \sin 2\pi f_c t$$
$$= V_c (1 + m \sin 2\pi f_m t) \bullet \sin 2\pi f_c t$$

m= modulation index 0…1 ($V_{env} = V_c$)

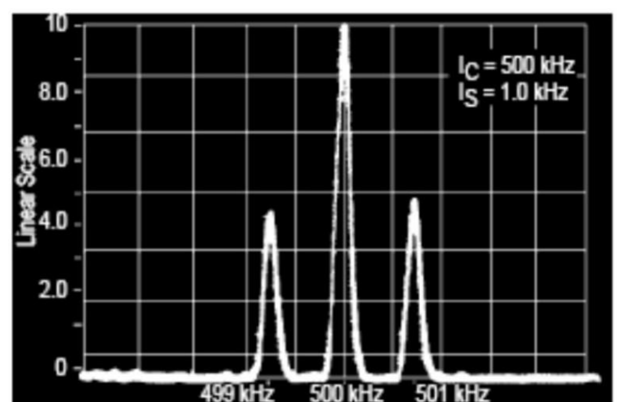
Using trigonometric identity:

$$(\sin a)(\sin b) = 1/2[\cos(a-b) - \cos(a+b)]$$

$$v = V_c \sin 2\pi f_c t$$
$$+ \frac{m}{2} V_c \cos 2\pi (f_c - f_m) t$$
$$- \frac{m}{2} V_c \cos 2\pi (f_c + f_m) t$$

$$v_{AM} = V_c \sin 2\pi f_c t + \frac{V_m}{2} \cos 2\pi (f_c - f_m) t - \frac{V_m}{2} \cos 2\pi (f_c + f_m) t$$

Carrier LSB USB



Relevant example of amplitude modulation: stimulated betatron oscillation(or: tune measurement)

taken from R.Jones,
proc. of BI-CAS 2018

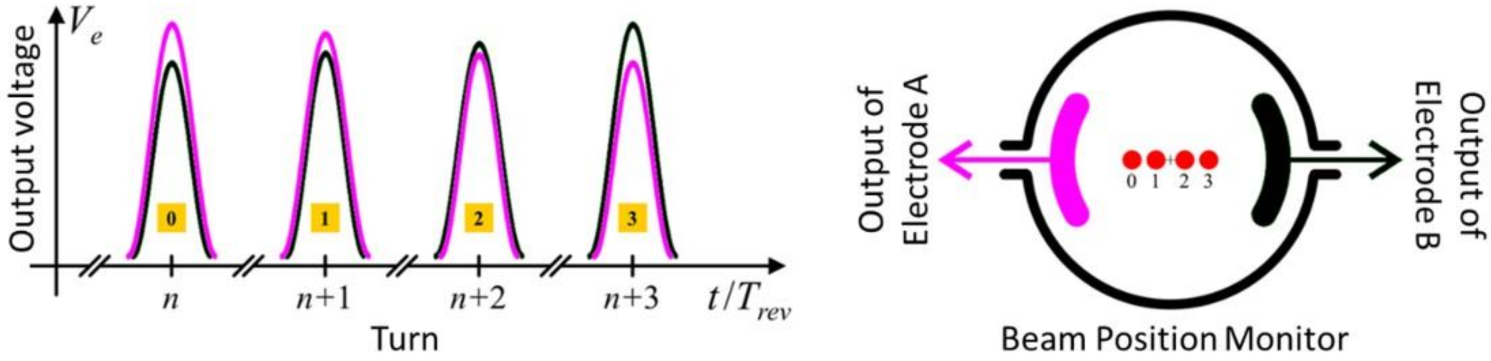


Fig. 4: Detecting oscillations using a beam position monitor. The oscillation information is superimposed as a small modulation on a large intensity signal.

Beam centre of charge makes small betatron oscillation around the closed orbit
(- stimulated by an exciter or by a beam instability)

Depending on the proximity to an EM sensor the measured signal amplitude varies.

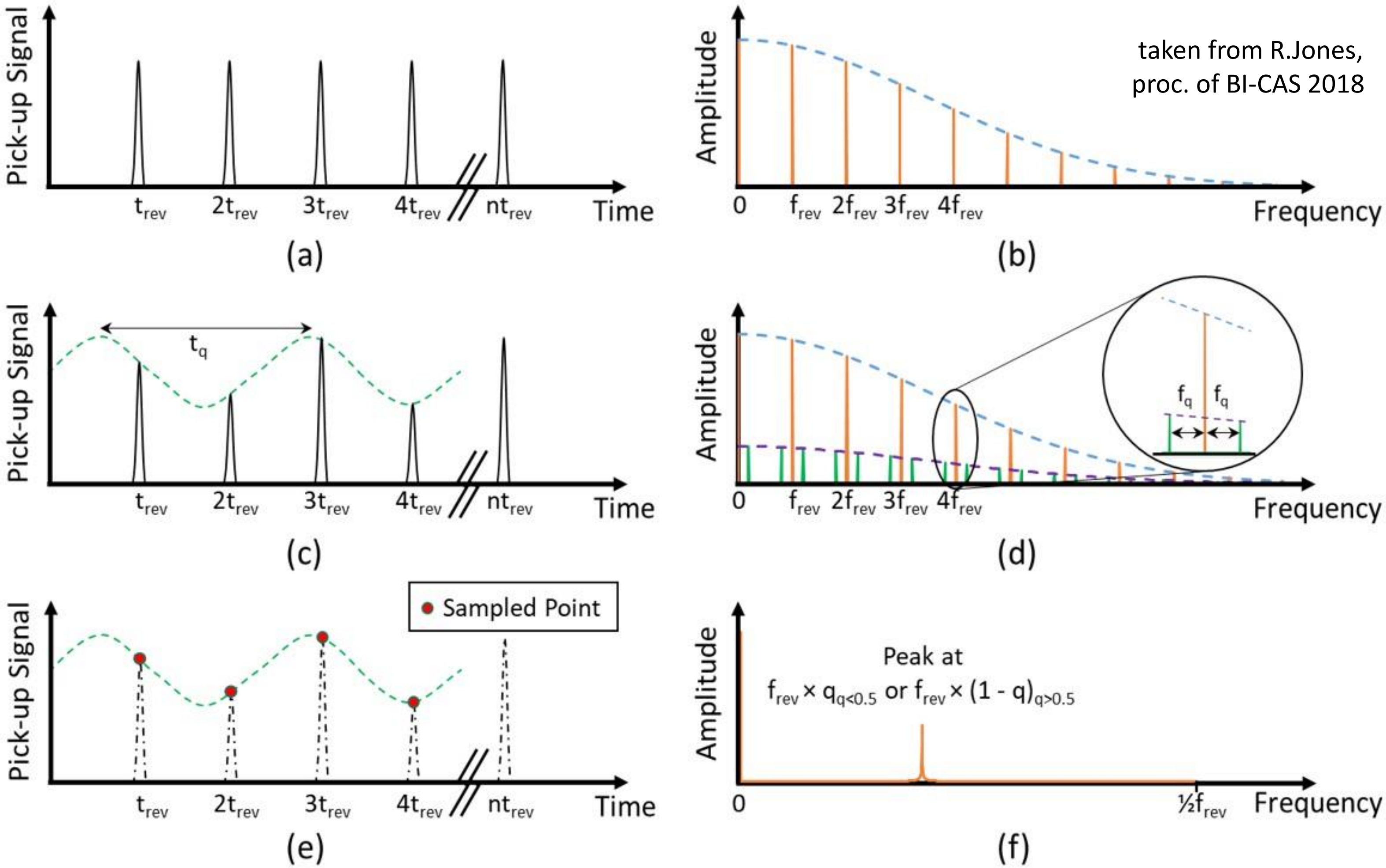
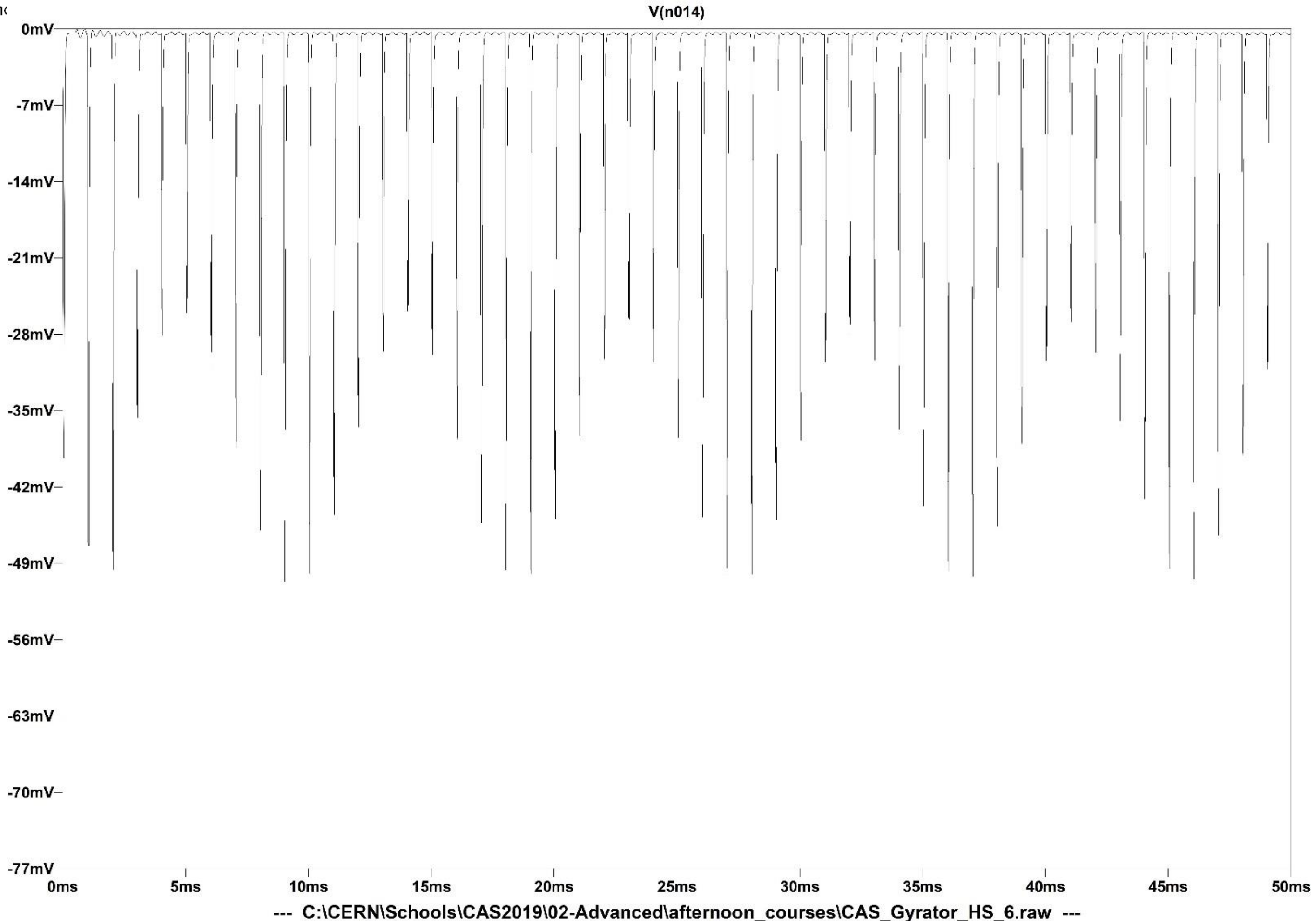


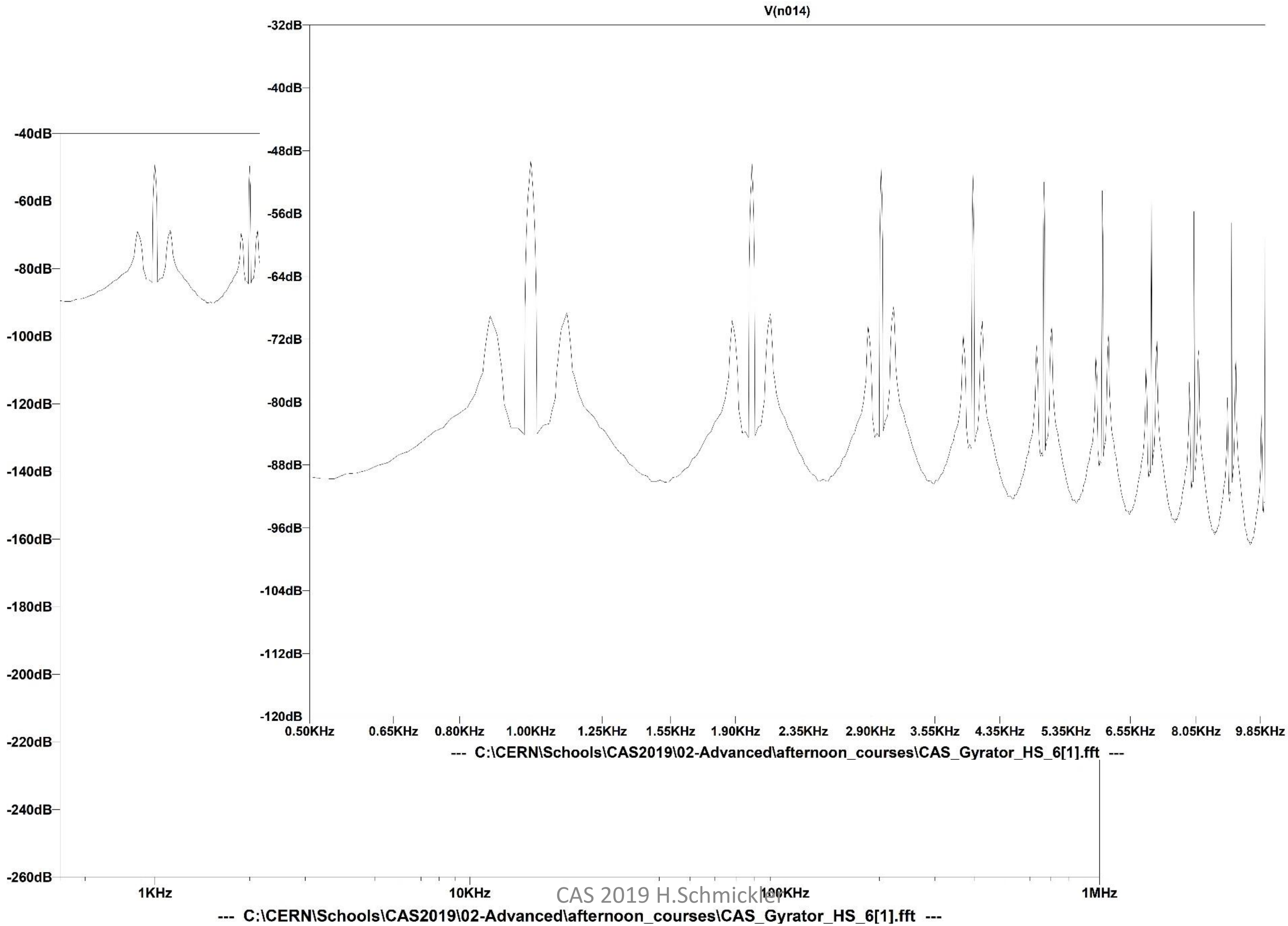
Fig. 2: Time and frequency domain representation for a bunch of particles observed at one single location on the circumference of the accelerator. (a & b) continuous measurement without betatron oscillation; (c & d) continuous measurement undergoing betatron oscillation (50% modulation); (e & f) sampled once per revolution.

A measured signal as example



Time domain signal of one beam sensor during betatron oscillation

The same in frequency domain



Multi-bunch multi-pass

- If one has **N bunches** of equal intensity circulating in an accelerator with $\mathbf{T_{rev}}$ and those bunches only move coherently without any phase difference, then this is undistinguishable from an accelerator with $\mathbf{T_{rev}/N}$ as revolution time and **one bunch** in this accelerator
- The time domain and frequency domain signals get really mind-boggling, if the bunches start to do individual uncorrelated betatron oscillations
- The study of these oscillations is important in case of multi-bunch instabilities or in case of the design of transverse active feedback systems

Multi-bunch modes

Let us consider **M** bunches equally spaced around the ring

Any oscillation pattern of these bunches can be decomposed into a set of eigenmodes of oscillation, the so called multi-bunch modes.

Each multi-bunch mode is characterized by a bunch-to-bunch phase difference of:

$$\Delta\Phi = m\,\frac{2\pi}{M}$$

m = multi-bunch mode number (0, 1, .., M-1)

Each multi-bunch eigenmode is characterized by a set of frequencies:

$$\omega = p\,M\omega_{rev} \pm (m + \nu)\omega_{rev}$$

Where:

p is and integer number $-\infty < p < \infty$;   p=0 = baseband

ω_{rev} is the revolution frequency

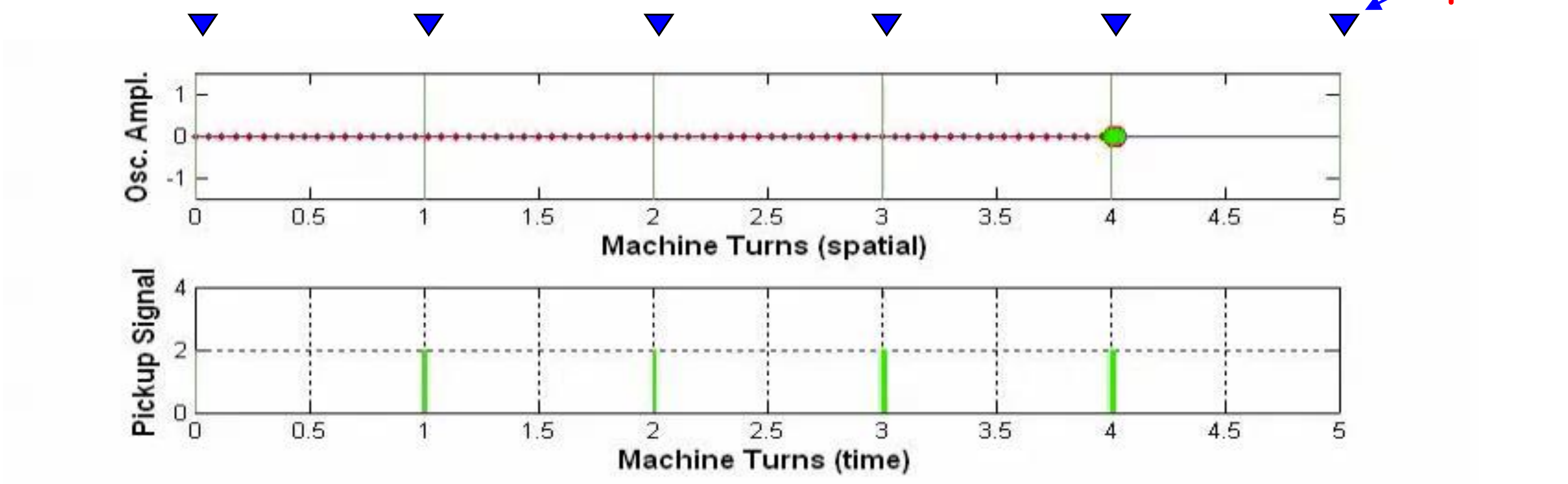
$M\omega_{rev} = \omega_{rep}$ is the bunch repetition frequency

ν is the tune

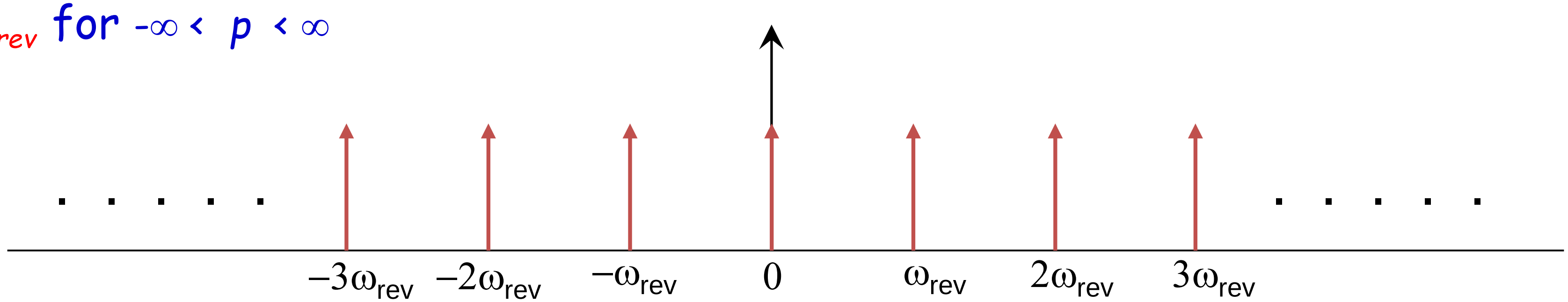
Hard to understand like this…needs some graphics

Multi-bunch modes: single stable bunch

transverse. One single stable bunch

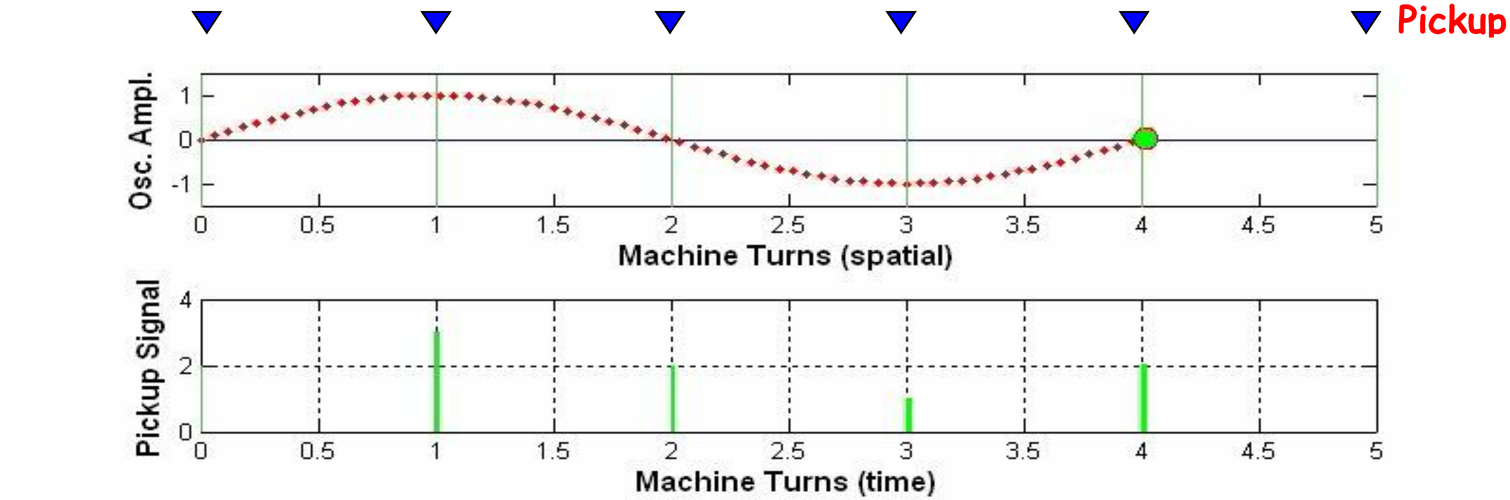


Every time the bunch passes through the pickup (▼) placed at coordinate 0, a pulse with constant amplitude is generated. If we think it as a Dirac impulse, the spectrum of the pickup signal is a repetition of frequency lines at multiple of the revolution frequency: $p\omega_{rev}$ for $-\infty < p < \infty$



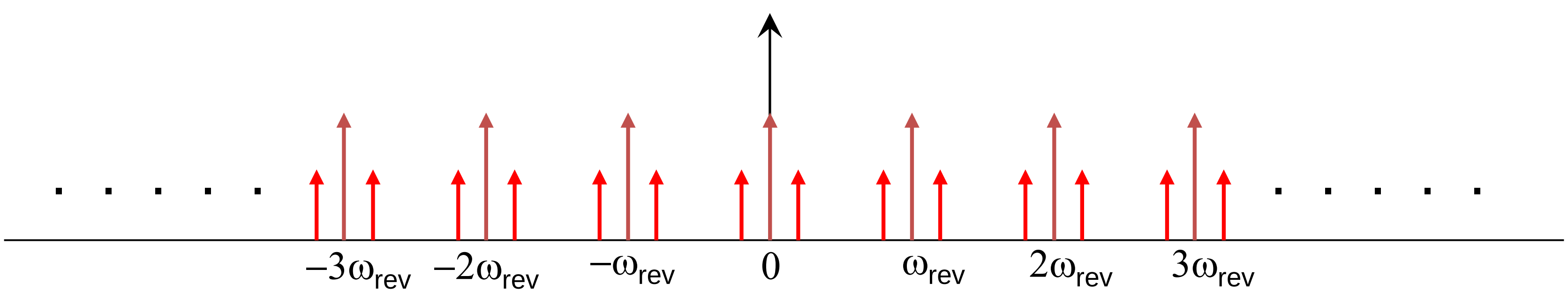
Multi-bunch modes: single oscillating bunch

One single unstable bunch oscillating at the tune frequency $\nu\omega_0$: for simplicity we consider a vertical tune $\nu < 1$, ex. $\nu = 0.25$. $M = 1 \rightarrow$ only mode #0 exists



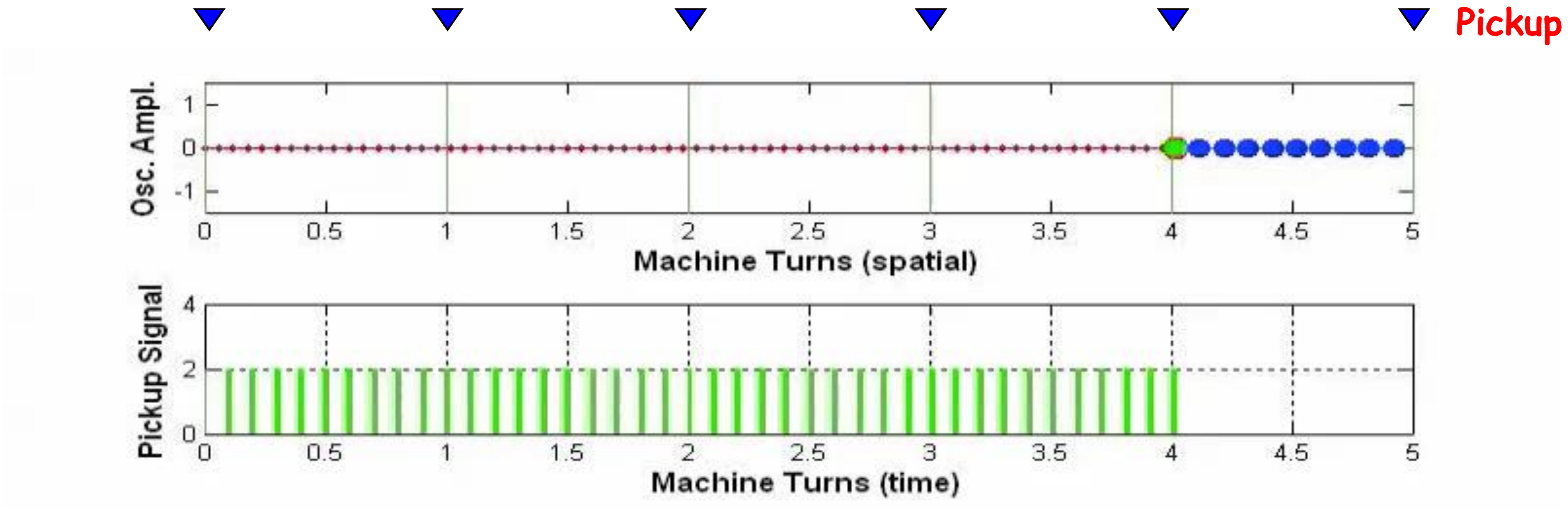
The pickup signal is a sequence of pulses modulated in amplitude with frequency $\nu\omega_0$

Two sidebands at $\pm\nu\omega_0$ appear at each of the revolution harmonics

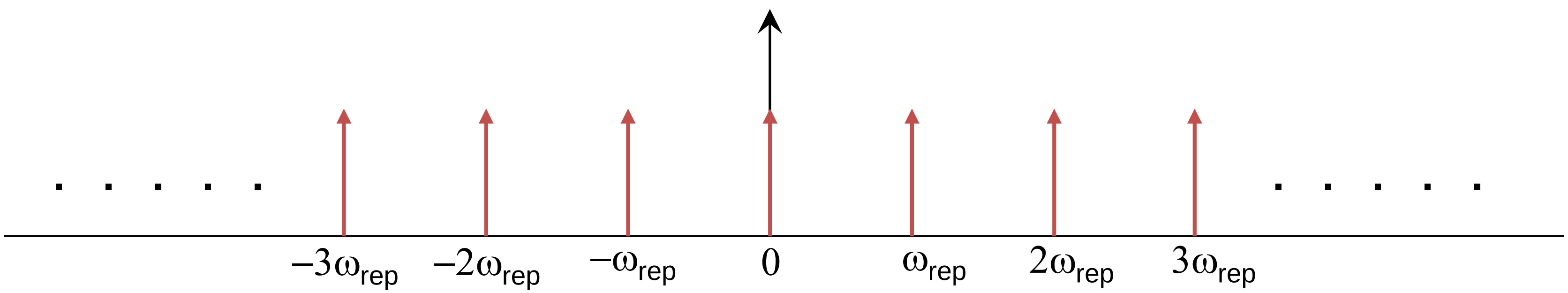


Multi-bunch modes: 10 stable bunches

Ten identical equally-spaced stable bunches (M = 10)



The spectrum is a repetition of frequency lines at multiples of the bunch repetition frequency: $\omega_{rep} = 10\ \omega_{rev}$ (bunch repetition frequency)



Multi-bunch modes: 10 unstable bunches (m=0)

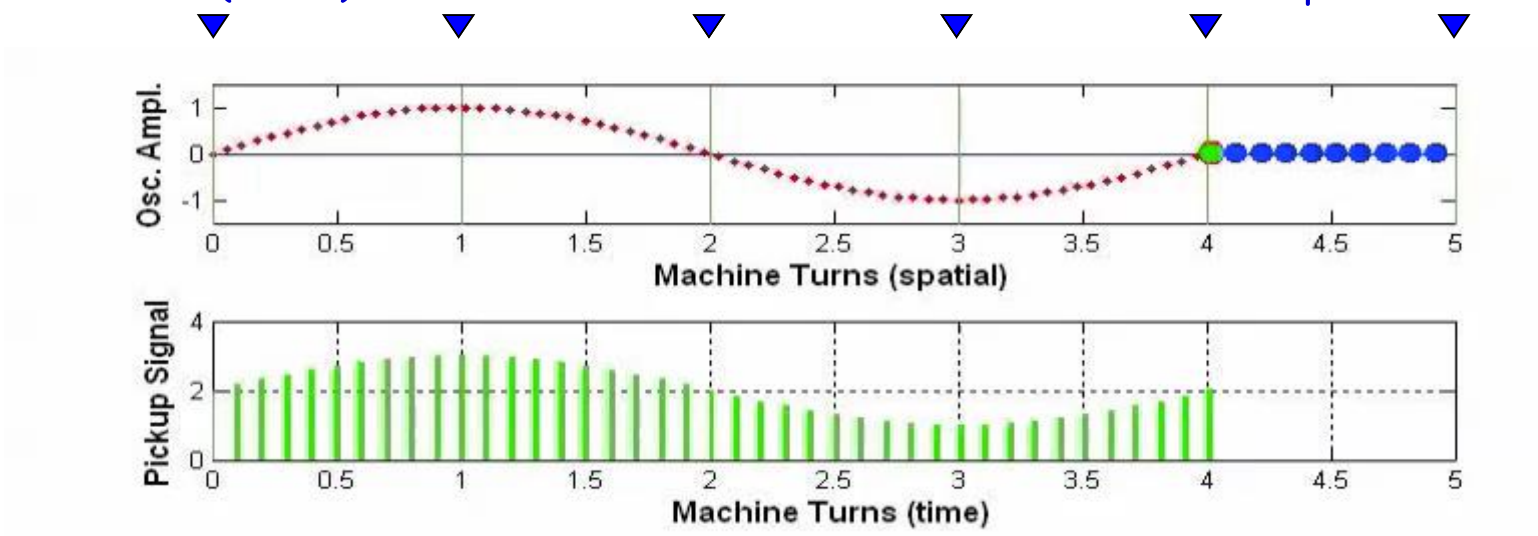
Ten identical equally-spaced unstable bunches oscillating at the tune frequency $\nu\omega_0$ (ν = 0.25)

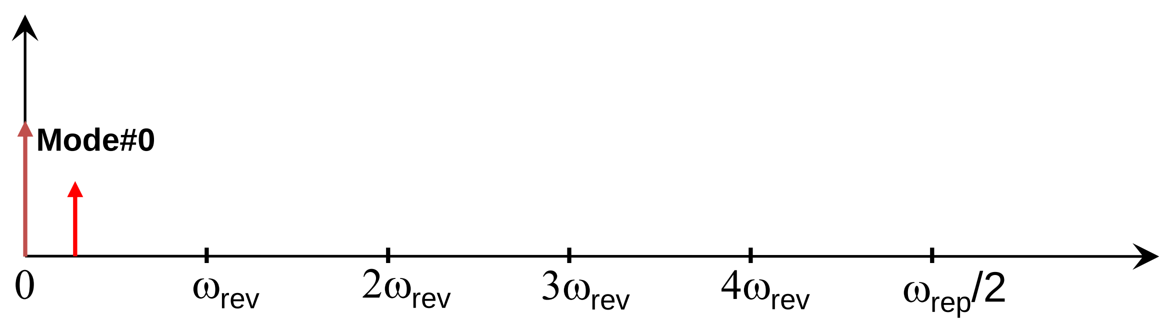
M = 10 → there are 10 possible modes of oscillation

$$\Delta\Phi = m\frac{2\pi}{M} \qquad m = 0, 1, .., M-1$$

Ex.: mode #0 (m = 0) $\Delta\Phi$=0 all bunches oscillate with the same phase

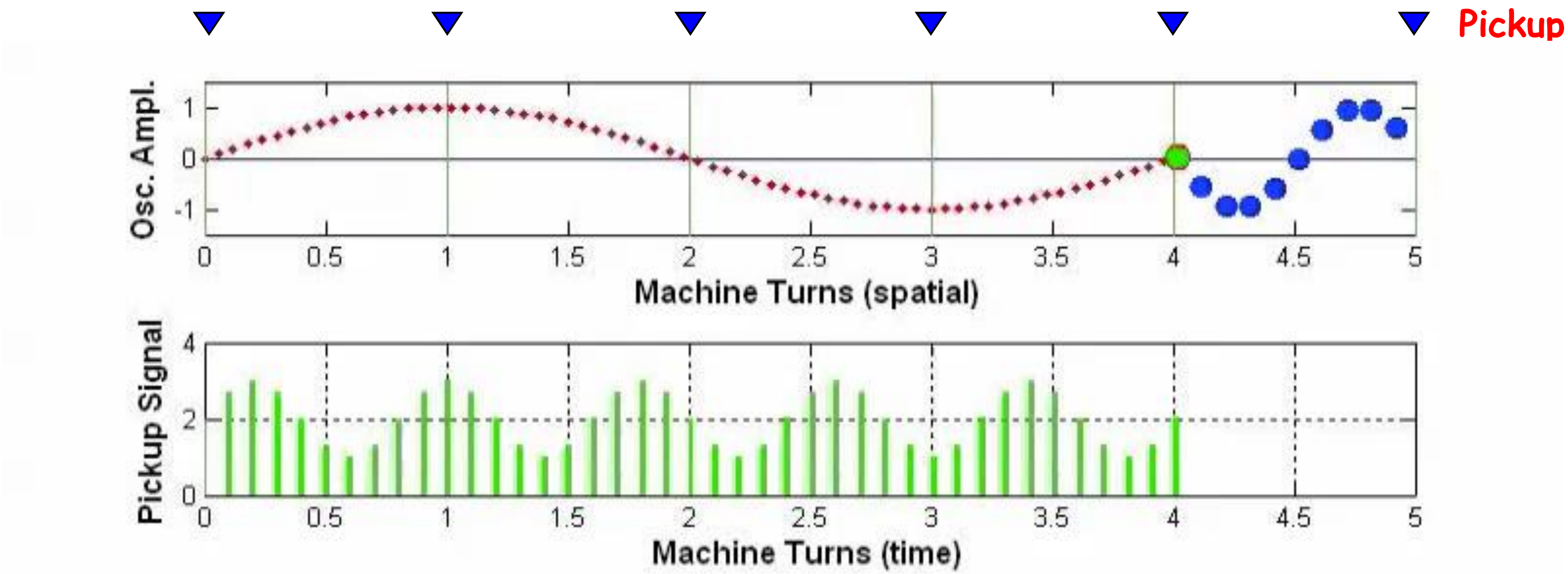
Pickup



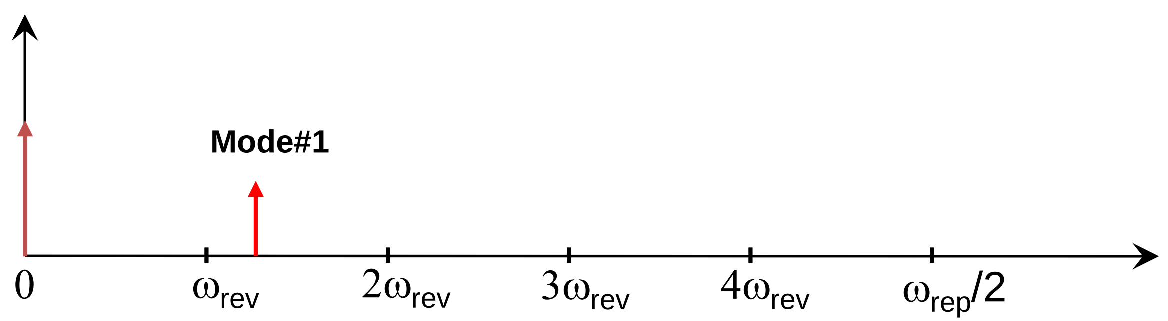


Multi-bunch modes: 10 unstable bunches (m=1)

Ex.: mode #1 (m = 1) $\Delta\Phi = 2\pi/10$ ($\nu = 0.25$)

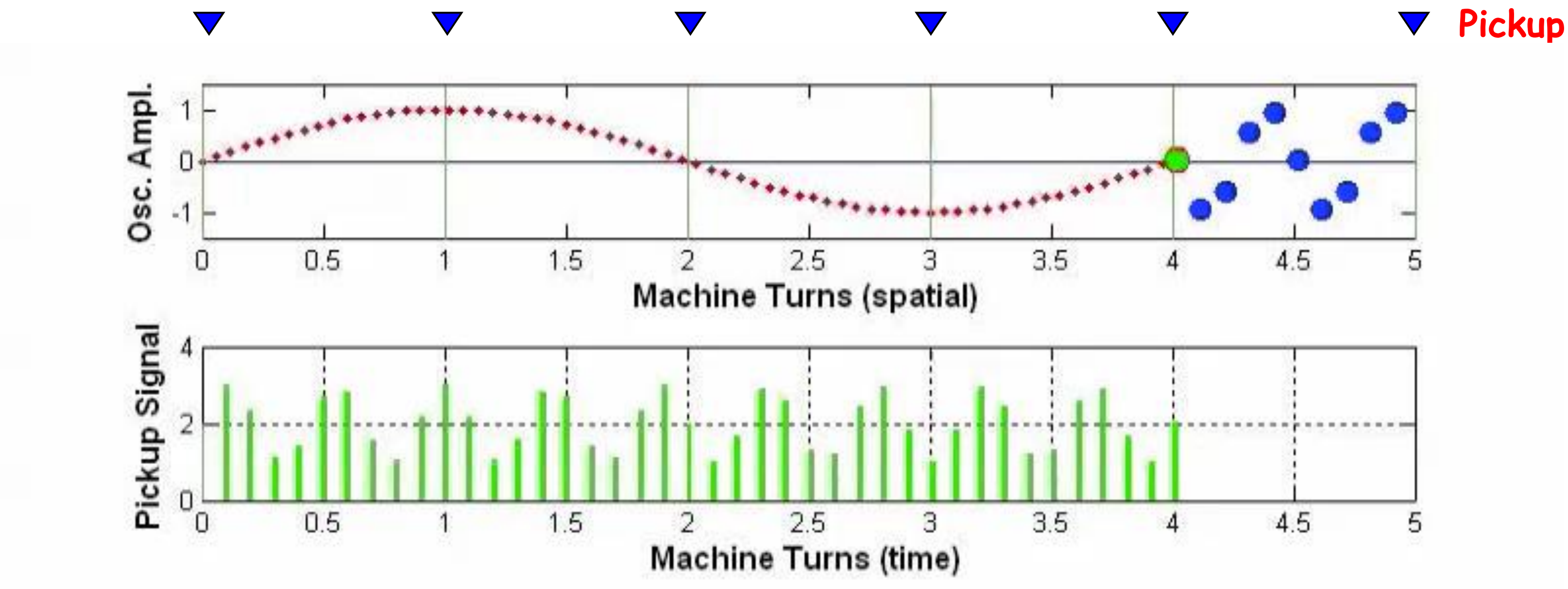


$$\omega = p\omega_{rep} \pm (\nu+1)\omega_{rev} \qquad -\infty < p < \infty$$

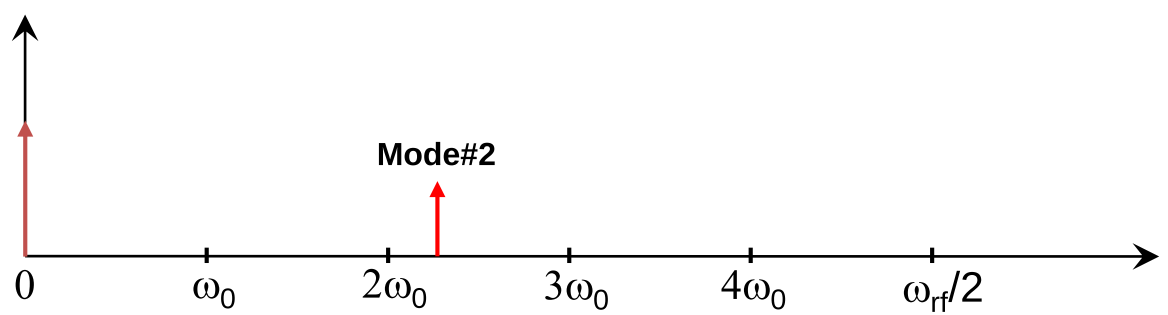


Multi-bunch modes: 10 unstable bunches (m=2)

Ex.: mode #2 (m = 2) $\Delta\Phi = 4\pi/10$ ($\nu = 0.25$)

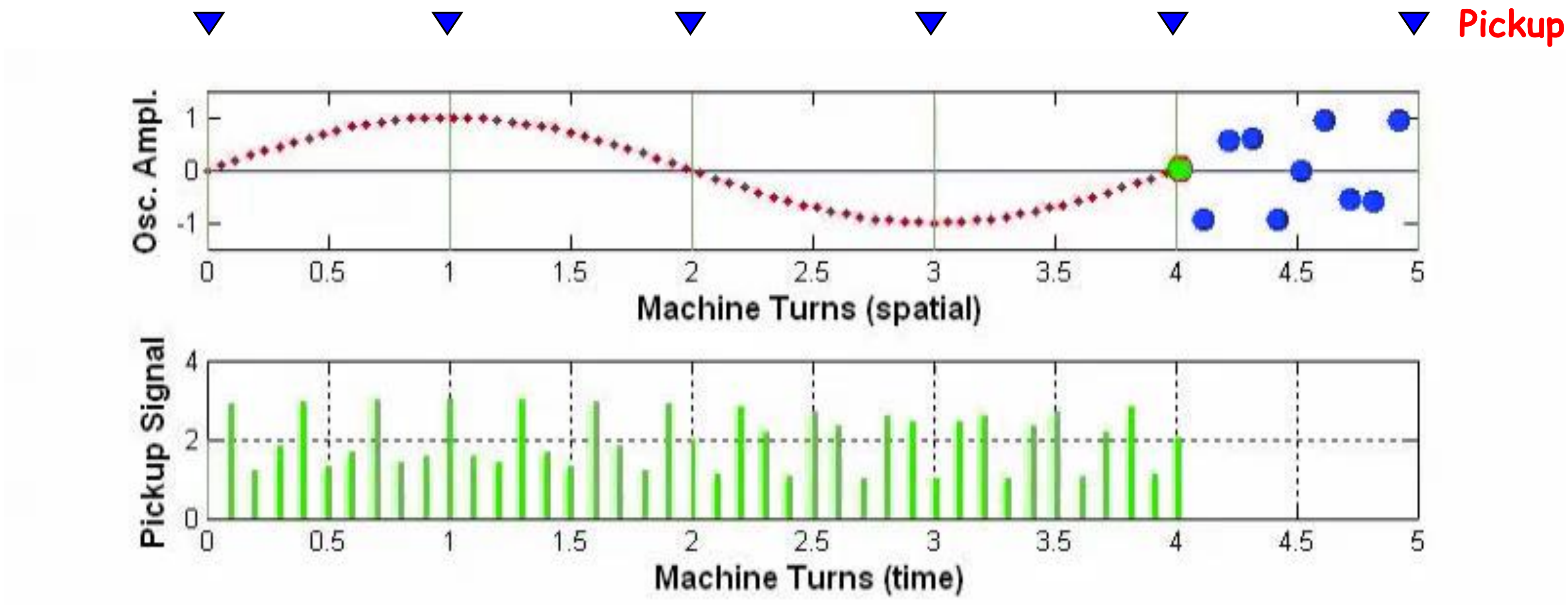


$\omega = p\omega_{rep} \pm (\nu+2)\omega_{rev}$ $\quad -\infty < p < \infty$

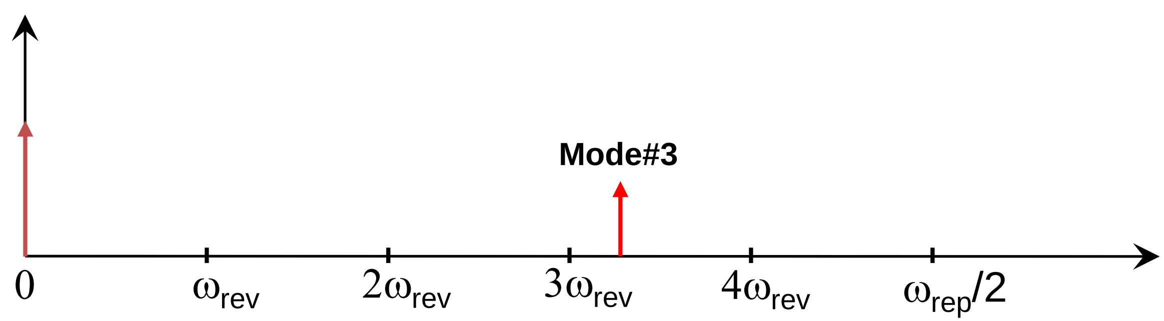


Multi-bunch modes: 10 unstable bunches (m=3)

Ex.: mode #3 (m = 3) $\Delta\Phi = 6\pi/10$ ($\nu = 0.25$)

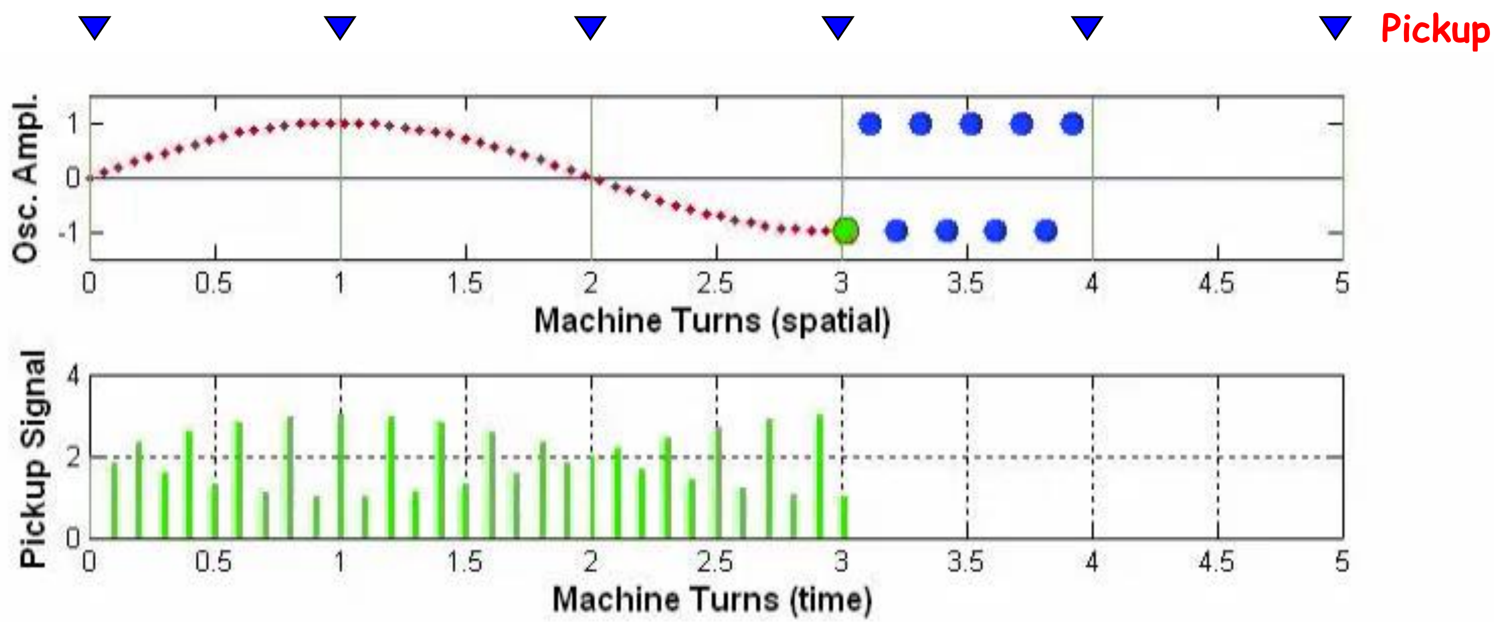


$\omega = p\omega_{rep} \pm (\nu+3)\omega_{rev}$ $-\infty < p < \infty$

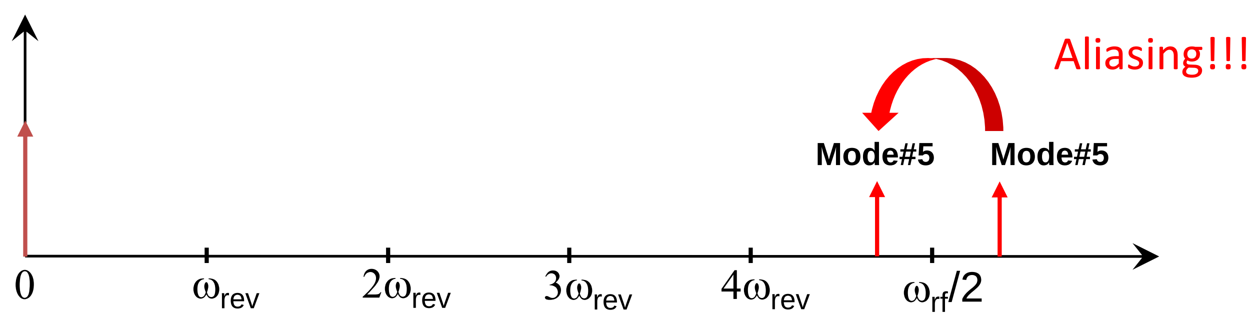


Multi-bunch modes: 10 unstable bunches (m=5)

Ex.: mode #5 (m = 5) $\Delta\Phi = \pi$ ($\nu = 0.25$)

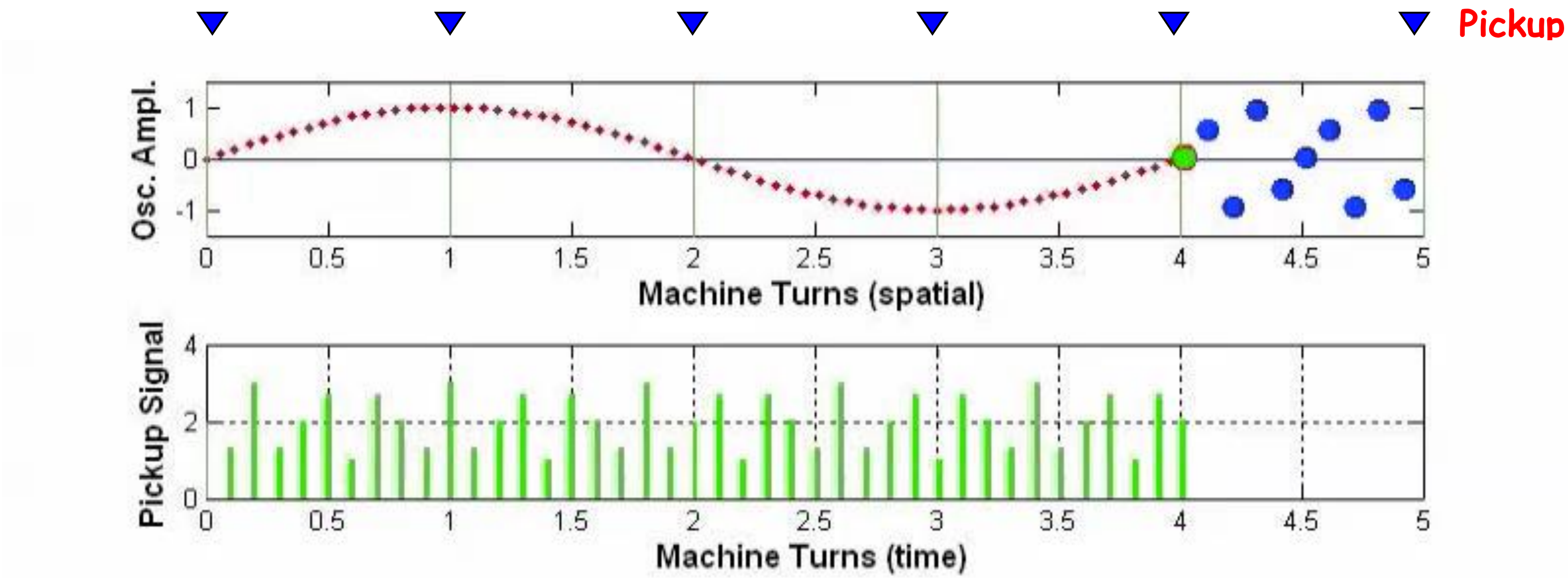


$\omega = p\omega_{rep} \pm (\nu+5)\omega_{rev}$ $\quad -\infty < p < \infty$

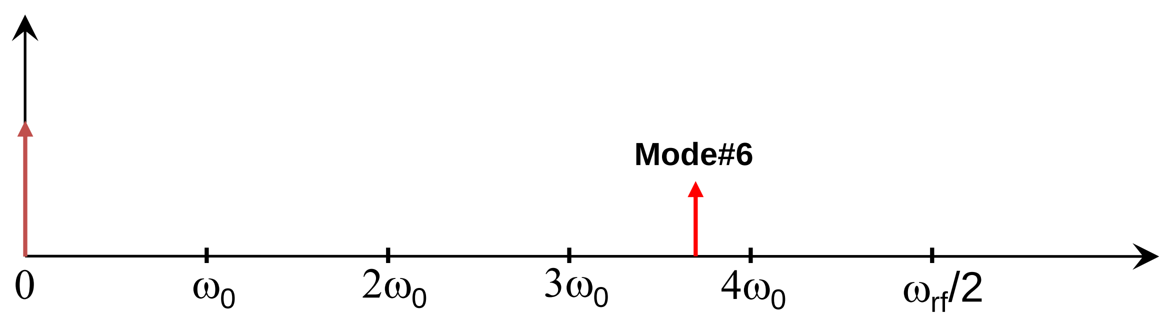


Multi-bunch modes: 10 unstable bunches (m=6)

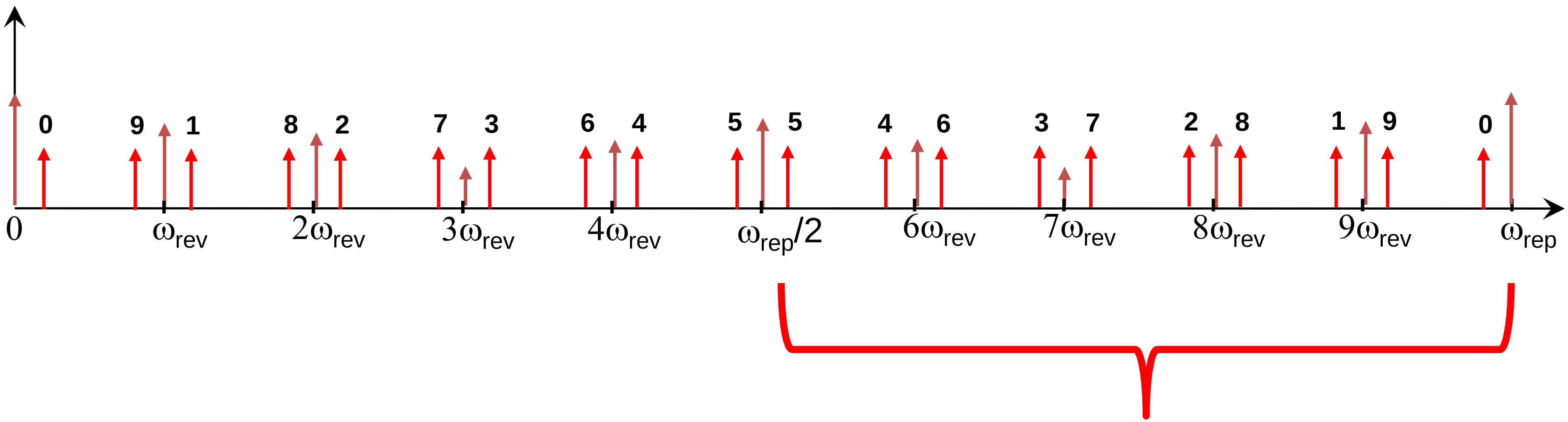
Ex.: mode #6 (m = 6) $\Delta\Phi = 12\pi/10$ ($\nu = 0.25$)



$$\omega = p\omega_{rf} \pm (\nu+6)\omega_0 \qquad -\infty < p < \infty$$



Multi-bunch modes: summary (10 bunches)



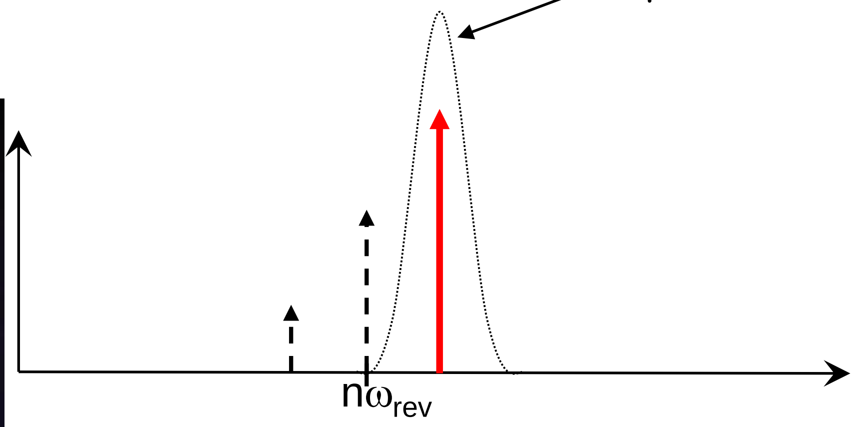
Lower sidebands of first revolution harmonics

$$\omega = p\,M\omega_{rev} \pm (m+q)\omega_{rev}$$

If the bunches have not the same charge, i.e. the buckets are not equally filled (uneven filling), the spectrum has frequency components also at the revolution harmonics (multiples of ω_{rev}). The amplitude of each revolution harmonic depends on the filling pattern of one machine turn

Multi-bunch modes: coupled-bunch instability

One multi-bunch mode can become unstable if one of its sidebands overlaps, for example, with the frequency response of a cavity high order mode (HOM). The HOM couples with the sideband giving rise to a coupled-bunch instability, with consequent increase of the sideband amplitude



Synchrotron Radiation Monitor showing the transverse beam shape

Effects of coupled-bunch instabilities:

- increase of the transverse beam dimensions
- increase of the effective emittance
- beam loss and max current limitation
- increase of lifetime due to decreased Touschek scattering (dilution of particles)

Real example of multi-bunch modes

The CERN Accelerator School

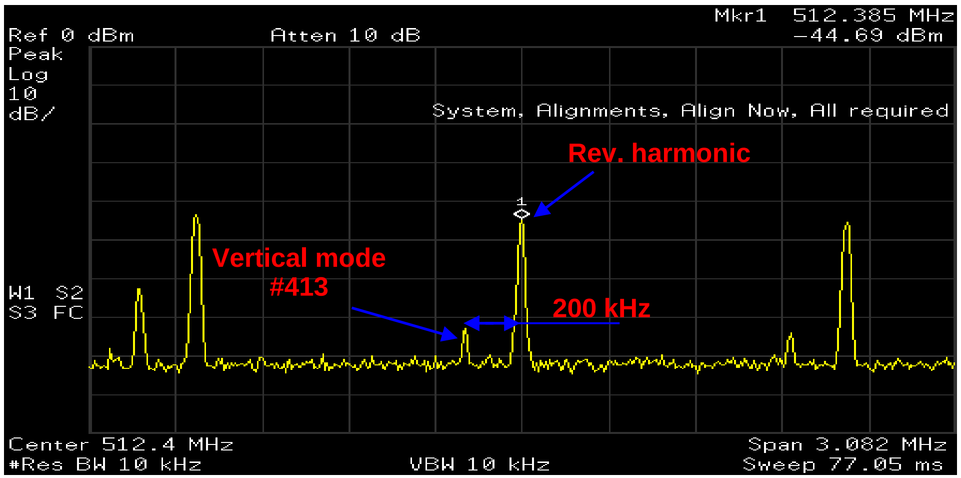
CERN

ELETTRA Synchrotron: f_{rf}=499.654 Mhz, bunch spacing≈2ns, 432 bunches, f_0 = 1.15 MHz

ν_{hor}= 12.30(fractional tune frequency=345kHz), ν_{vert}=8.17(fractional tune frequency=200kHz)

ν_{long} = 0.0076 (8.8 kHz)

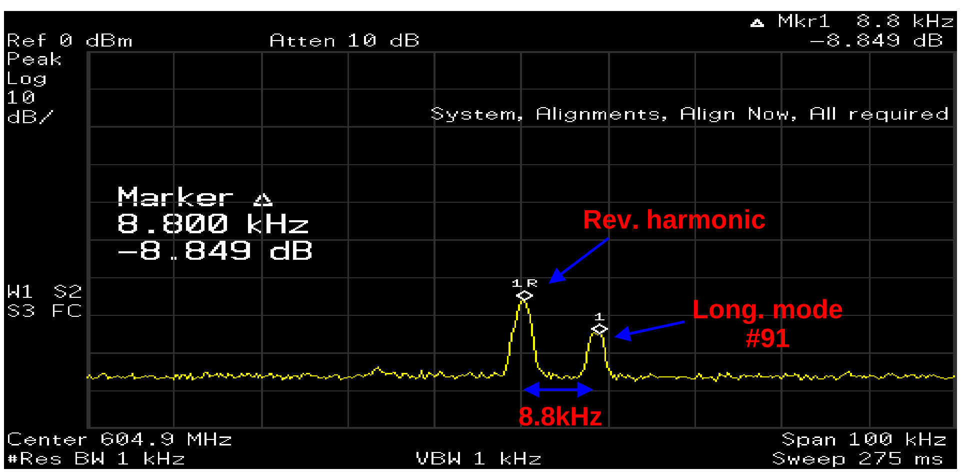
$$\omega = p M \omega_0 \pm (m+\nu)\omega_0$$



Spectral line at 512.185 MHz

Lower sideband of $2f_{rf}$, 200 kHz apart from the 443[rd] revolution harmonic

→ vertical mode #413



Spectral line at 604.914 MHz

Upper sideband of f_{rf}, 8.8kHz apart from the 523[rd] revolution harmonic

→ longitudinal mode #91

Part II

- Fourier transform of time sampled signals

  a) basics

  b) aliasing

  c) windowing

- Methods to improve the frequency resolution

  a) interpolation

  b) fitting (the NAFF algorithm)

  c) influence of signal to noise ratio

- Analysis of non stationary spectra:

  - STFT (:= Short time Fourrier transform) (Gabor transform)

    also called: Sliding FFT, Spectogram

  - wavelet analysis

  - PLL tune tracking

Discrete Fourier Transforms

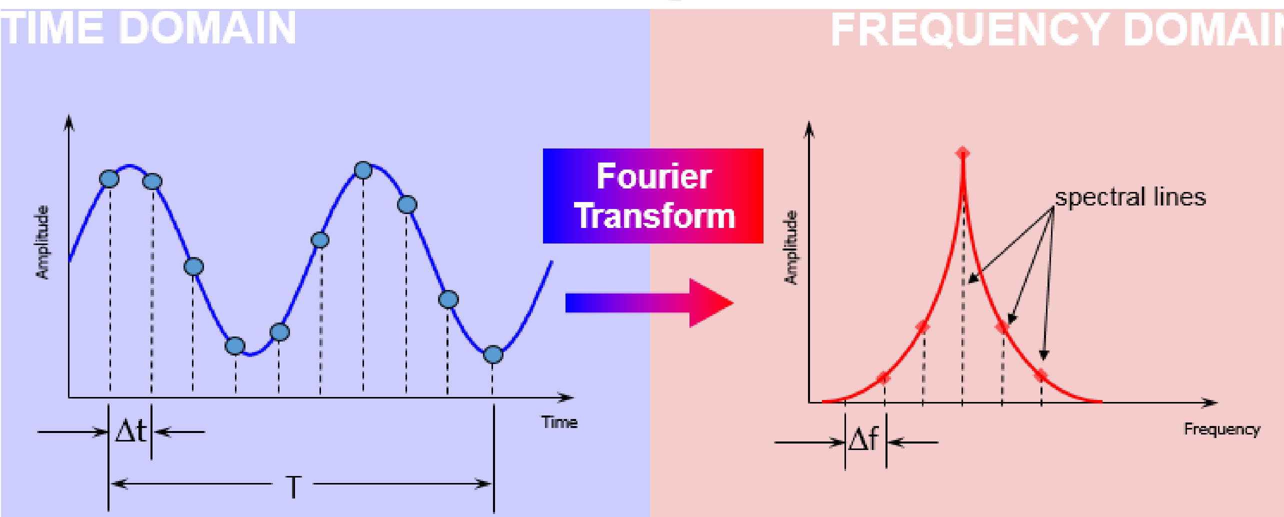
- Discrete Fourier Transform basics

In general:

We use DFTs of N equidistant time sampled signals;

A FFT (Fast Fourier transform) is a DFT with $N = 2^k$

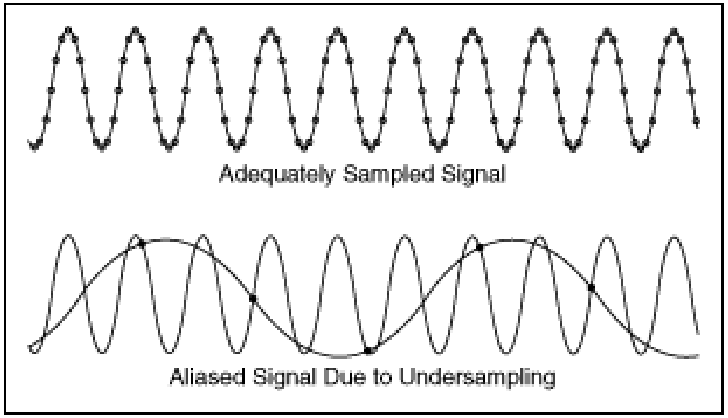
Time Duration		
Finite	Infinite	
Discrete FT (DFT) $X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\omega_k n}$ $k = 0, 1, \ldots, N-1$	Discrete Time FT (DTFT) $X(\omega) = \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n}$ $\omega \in (-\pi, +\pi)$	discr. time n
Fourier Series (FS) $X(k) = \int_0^P x(t) e^{-j\omega_k t} dt$ $k = -\infty, \ldots, +\infty$	Fourier Transform (FT) $X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$ $\omega \in (-\infty, +\infty)$	cont. time t
discrete freq. k	continuous freq. ω	



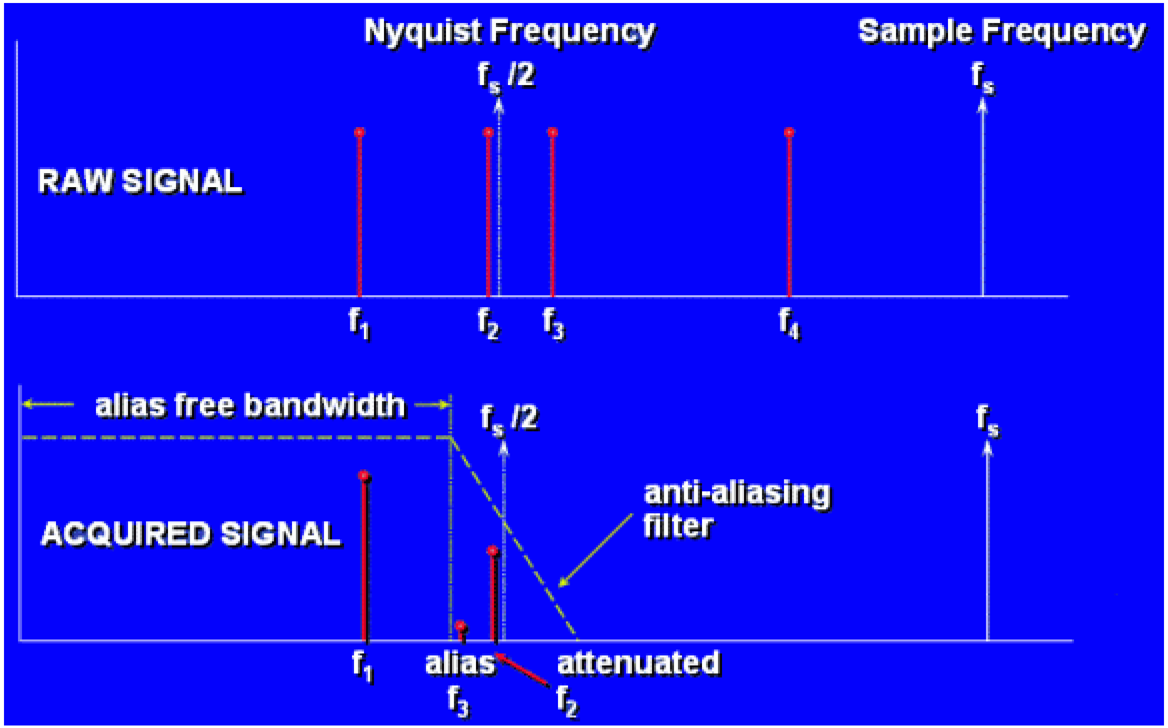
Sampling rate (samples/sec)	$F_s = 1/\Delta t$
Frame Size (seconds)	$T = N\,\Delta t$
Block Size (# samples)	N

Bandwidth or Max Freq (Hz)	$F_{max} = F_s/2$
Frequency Resolution (Hz)	$\Delta f = F_{max} / SL$
Spectral Lines (# samples)	$SL = N/2$

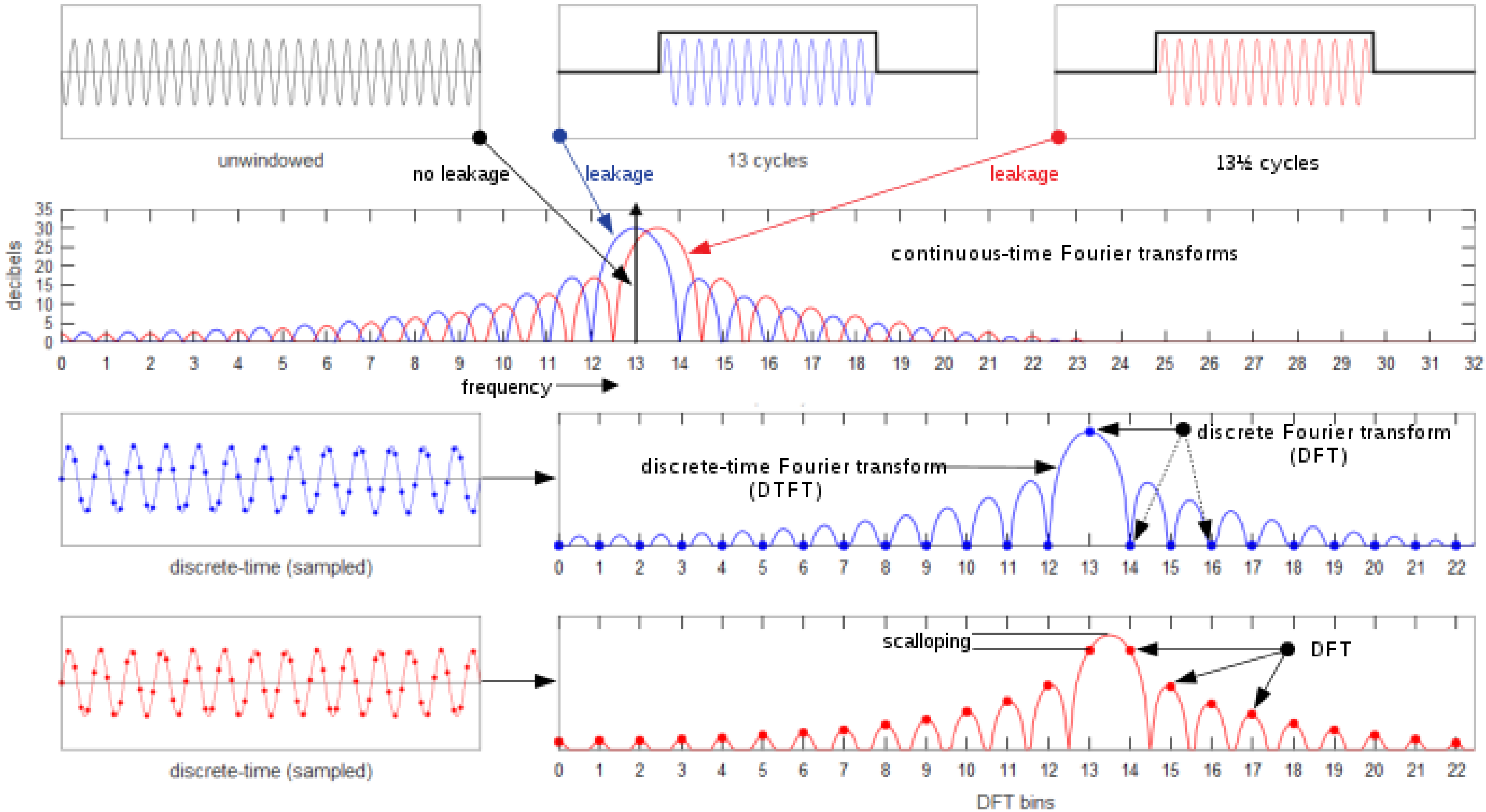
DFT - aliasing



- Periodic signals, which are sampled with at least 2 samples per period, can be unambiguously reconstructed from the frequency spectrum. (Nyquist-Shannon Theorem)
- In other words, with a DFT one only obtains useful information up to half the sampling frequency.
- Antialiasing filters before the sampling suppress usually unwanted higher spectral information.



Spectral leakage caused by windowing

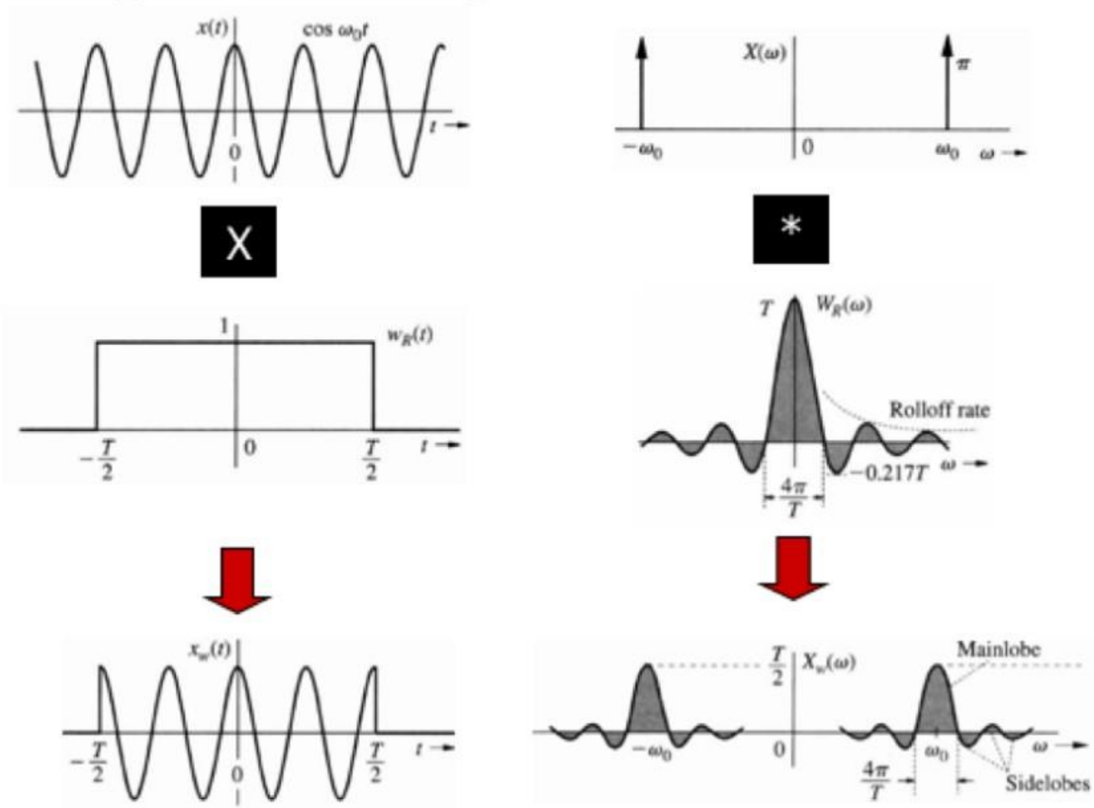


By measuring a continuous signal only over a finite length, we apply a “data window” to signal, which leads to spectral artefacts in frequency domain.

Windowing – Convolution of continuous signal with window function

- Recall: The Fourier transform of a product in time domain is the convolution of the individual Fourier transforms in Frequency domain

- Extracting a segment of a signal in time is the same as multiplying the signal with a rectangular window:



Spectral spreading

Energy spread out from ω0 to width of 2π/T – reduced spectral resolution.

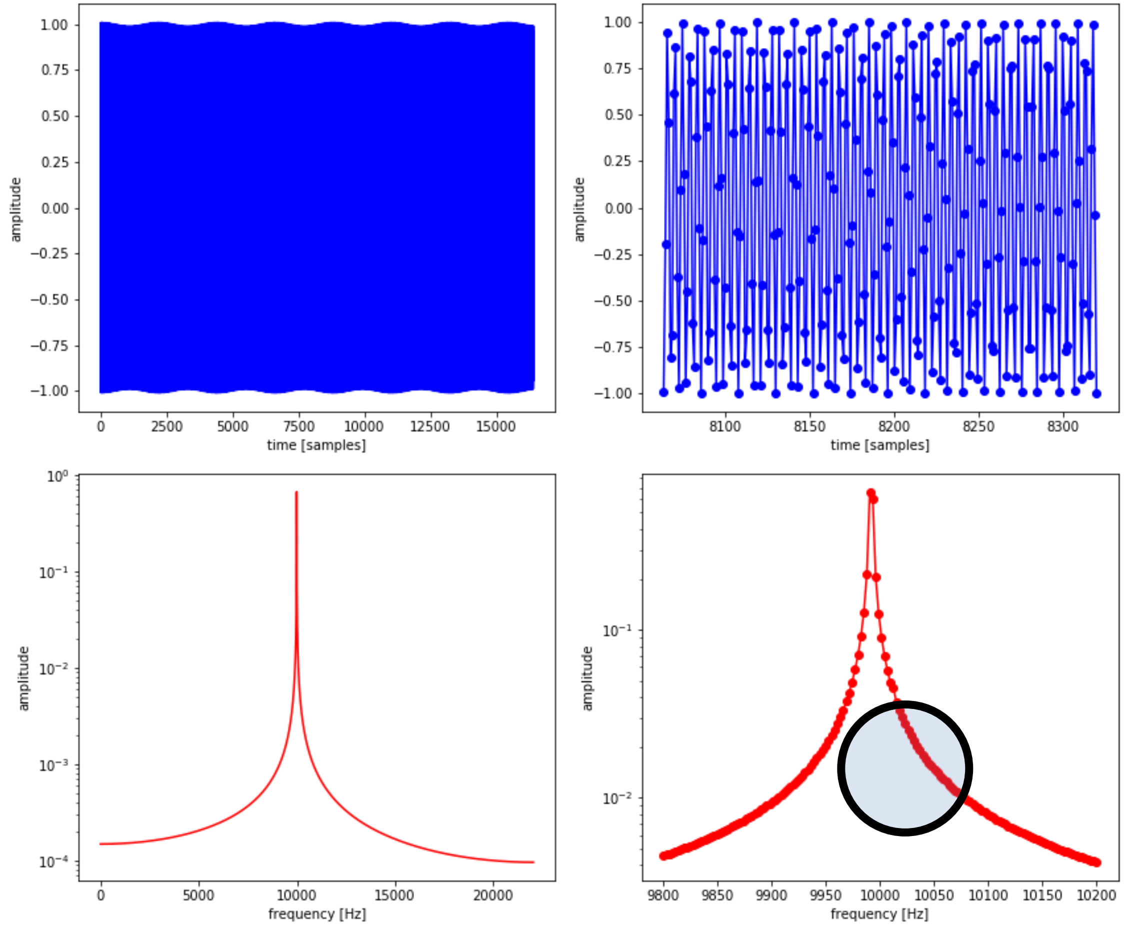
Leakage

Energy leaks out from the mainlobe to the sidelobes.

Rectangular window example

signal = amp1* sin ($2\pi\ \omega_1 t$) + amp2 * sin($2\pi\ \omega_2 t$)

FFT



ZOOM

amp1 =1
amp2=0.01

$\omega_1 = 2\pi * 9990$ Hz
$\omega_2 = 2\pi * 10010$ Hz

The small signal component is completely masked by the sidelobe of the large signal

Applying the Blackman-Harris window

signal = window * amp1* sin ($2\pi\, \omega_1 t$) + amp2 * sin($2\pi\, \omega_2 t$)

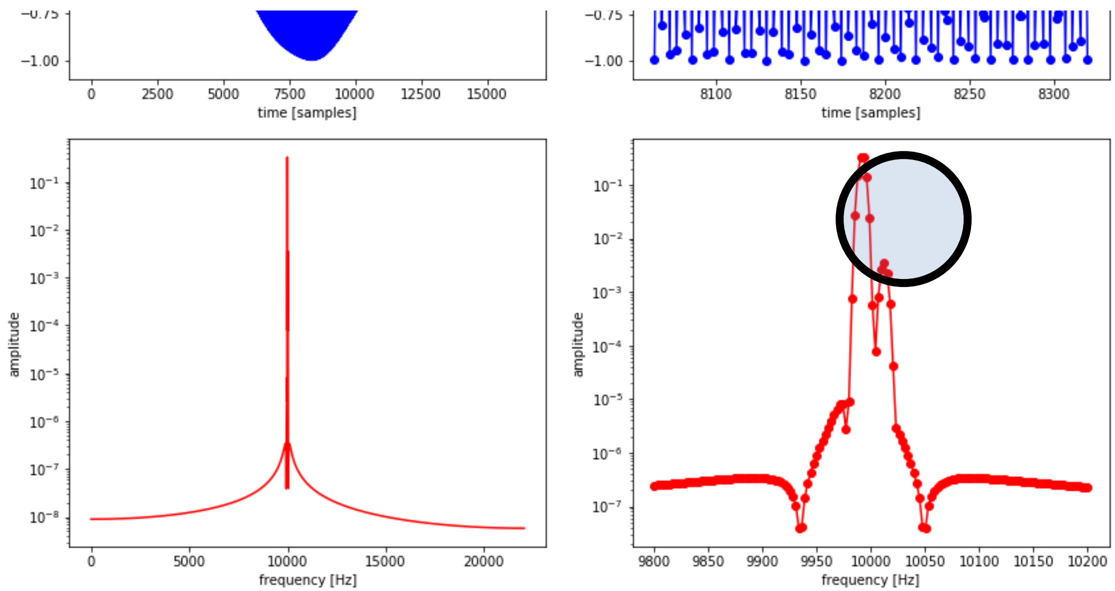
Blackman–Harris window

A generalization of the Hamming family, produced by adding more shifted sinc functions, meant to minimize side-lobe levels

$$w[n] = a_0 - a_1 \cos\left(\frac{2\pi n}{N}\right) + a_2 \cos\left(\frac{4\pi n}{N}\right) - a_3 \cos\left(\frac{6\pi n}{N}\right)$$

$$a_0 = 0.35875; \quad a_1 = 0.48829; \quad a_2 = 0.14128; \quad a_3 = 0.01168.$$

FFT

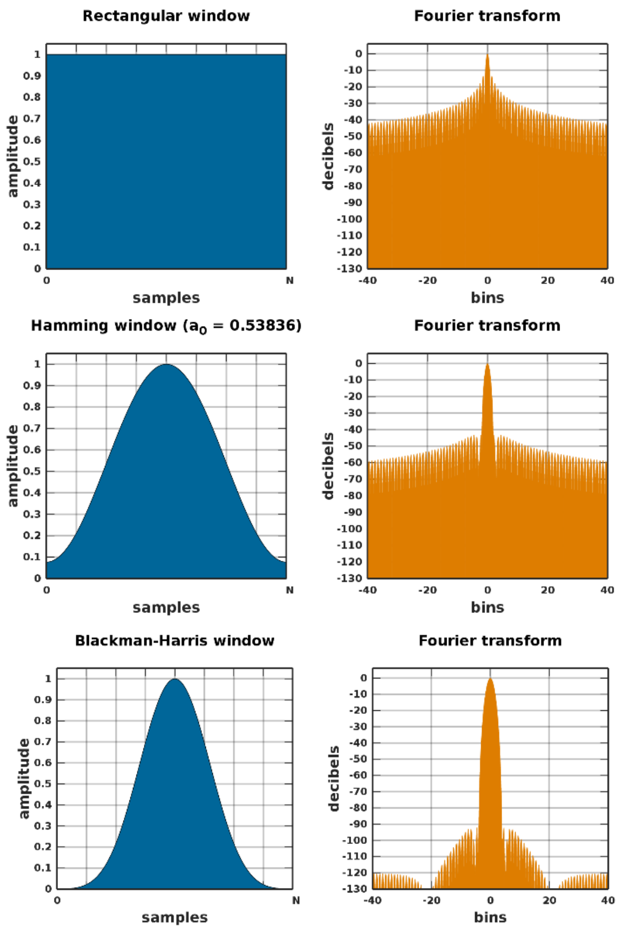


ZOOM

The small signal component is nicely resolved

Popular window functions

- The following link contains many frequently used window functions, their main features and application:
- https://en.wikipedia.org/wiki/Window_function



The actual choice of the window depends on:

- The signal composition
- The required dynamic range
- The signal to noise ration

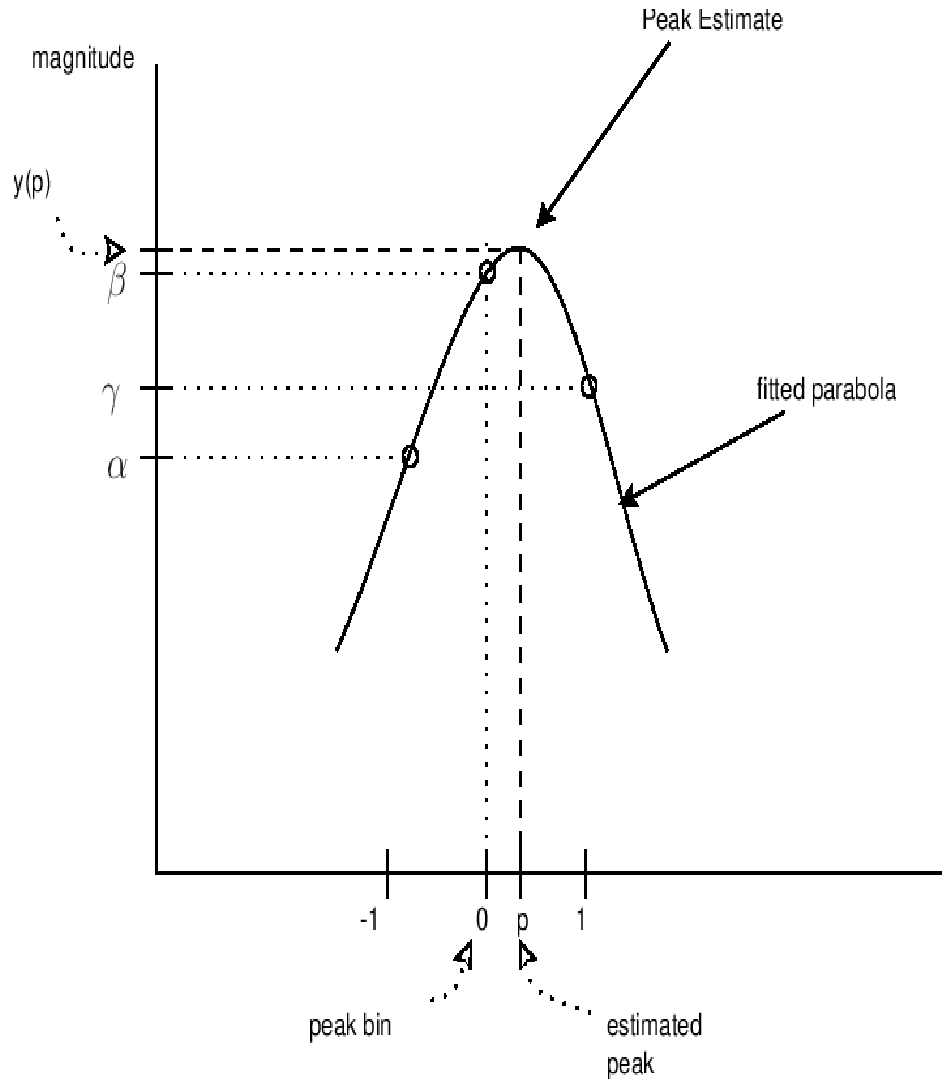
remark: every window except the rectangular window is linked to a loss in amplitude (we multiply many samples with almost “zero”)
→ reduced S/N up to 6 dB

Improving the frequency resolution of a DFT spectrum

- Recall: basic frequency resolution:
  $\Delta f = 2*f_{samp}/N_{samp}$
- We can interpolate between the frequency bin with maximum content and the left and right neighbouring bins
- We limit the discussion to “three point interpolation methods”
- The interpolation function is either:

  A) a parabola of the measurements
  (:= parabolic interpolation)

  B) a parabola of the log of the measurements
  (:= Gaussian interpolation)
- Can get up to $1/N^2$ resolution



Details: https://mgasior.web.cern.ch/mgasior/pap/FFT_resol_note.pdf

Improving the frequency resolution of a DFT spectrum

Table 1. Efficiency of the parabolic and Gaussian interpolation with different windowing methods. The windows are characterised by main lobe width, highest sidelobe level and sidelobe asymptotic fall-off. The maximum interpolation error is given as a percentage of the spectrum bin spacing Δ_f. The interpolation gain factor G is defined in (19). Some details concerning the windows and the interpolation errors are given in the Appendix.

Window	Main lobe width [bin]	Highest sidelobe [dB]	Sidelobe asymptotic fall-off [dB/oct]	**Parabolic interpolation**		**Gaussian interpolation**	
				Error max. [% of Δ_f]	**Gain factor G**	**Error max. [% of Δ_f]**	**Gain factor G**
Rectangular	2	-13.3	6	23.4	2.14	16.7	2.99
Triangular	4	-26.5	12	6.92	7.23	2.08	24.1
Hann	4	-31.5	18	5.28	9.47	1.60	31.2
Hamming	4	-44.0	6	6.80	7.35	1.60	31.2
Blackman	6	-68.2	6	4.66	10.7	0.578	86.5
Blackman-Harris	6.54	-74.4	6	4.18	12.0	0.476	105
Nuttall	8	-98.2	6	3.51	14.2	0.314	159
Blackman-Harris-Nuttall	8	-93.3	18	3.34	15.0	0.314	159
Gaussian $L = 6\ \sigma$	6.96	-57.2	6	4.95	10.1	0.240	208
Gaussian $L = 7\ \sigma$	10.46	-71.0	6	3.80	13.2	0.0516	970
Gaussian $L = 8\ \sigma$	11.41	-87.6	6	2.95	17.0	0.00869	5756

$$Gain\ factor\ G := \frac{\Delta_f}{2\ x\ Error\ max.}$$

from: https://mgasior.web.cern.ch/mgasior/pap/FFT_resol_note.pdf

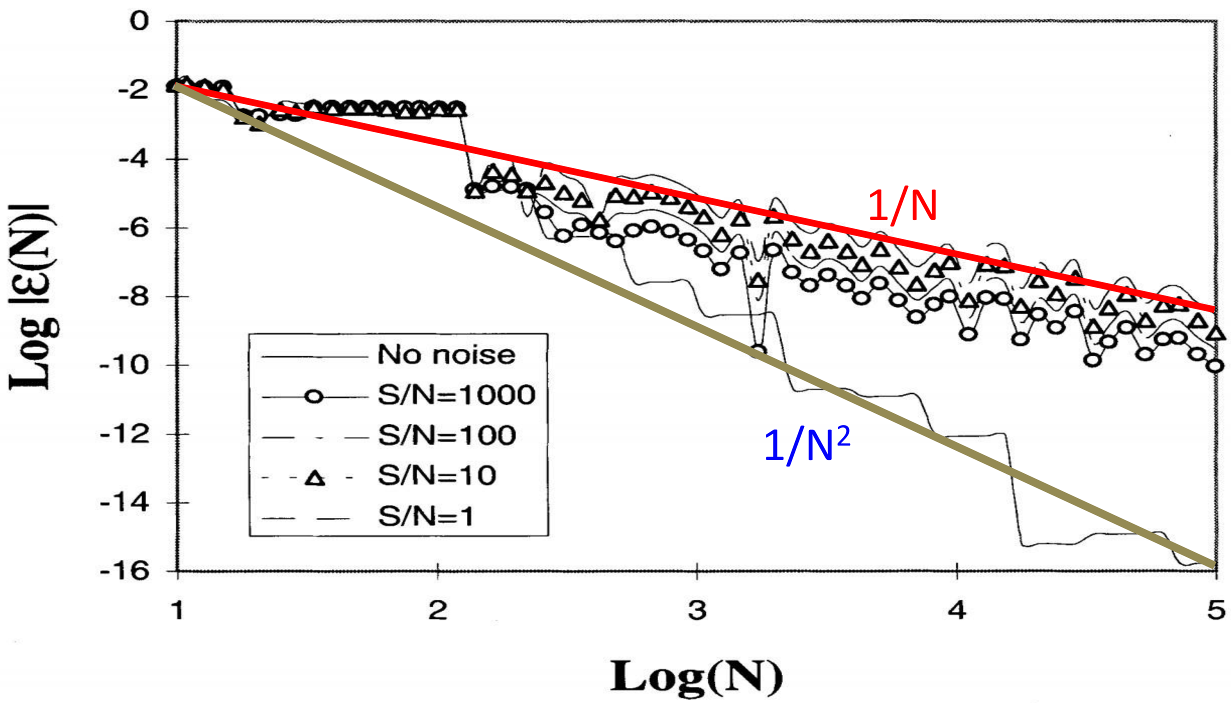
Even higher frequency resolution: fitting the data

1. Assume a model function for the data (sample $_{1 \ldots N}$)
(i.e. in the most simple case a monochromatic sin wave),
in general $\text{sample}_i = f(i * \Delta t)$
2. Get frequency and peak (or interpolated peak) from FFT:
f_{max} and a_{max}
3. Minimize:
$\Sigma = \sum_{i=0}^{N} (\text{sample}_i)^2 - (a_{max} * \sin(2\pi f_{max} * \Delta t))^2$
by varying a_{max} and f_{max}

(→NAFF algorithm:= Numerical Analysis of Fundamental Frequencies
→ NAFF algorithm can get up to **$1/N^4$** resolution)

4. Very good convergence for noise free data
(i.e. predominantly in simulations)

A little summary on frequency resolution



Taken from: R. Bartolini et al, Precise Measurement of the Betatron tune, Proceedings of PAC 1995, Vol. 55, pp 247-256

- Frequency measurement error ε(N) as function of log (N) for different S/N ratios
- Basic FFT resolution proportional to 1/N
- Plot shows result for interpolation using Hanning window.
- With interpolation and no noise proportional to $1/N^2$
- Data fitting (NAFF algorithm) also very sensitive to S/N

A recent development: MultiBPM analysis

- P.Zisopoulos et al, Phys. Rev. Acc.&Beams 22, 071002 (2019)

Refined betatron tune measurements by mixing BPM data

Basic idea: Create additional samples per turn by using data from neighbouring BPMs (up to 500 in the LHC) and transforming them from samples in space to samples in time.

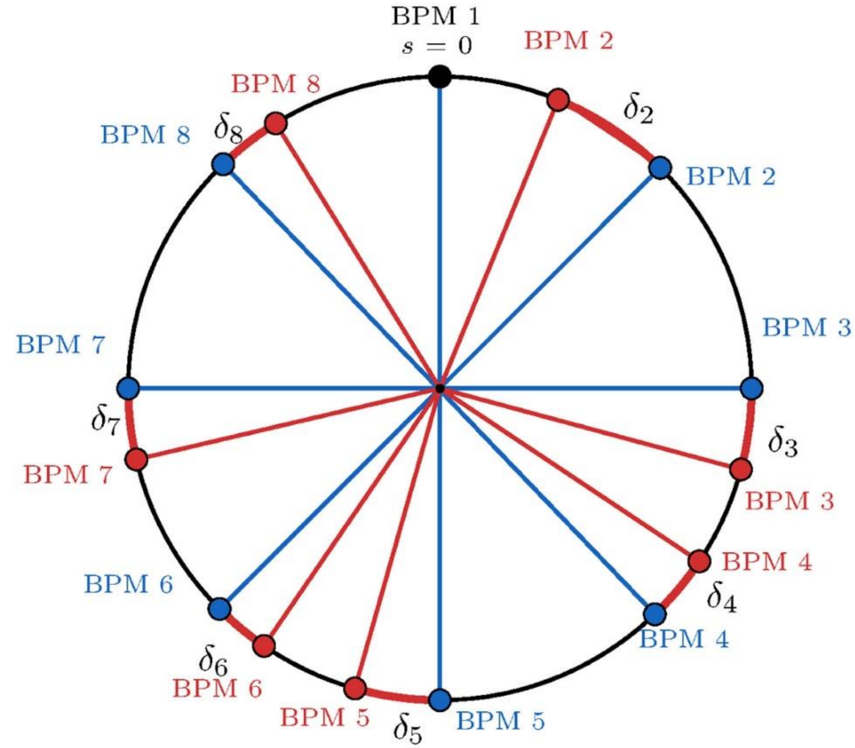


FIG. 1. A hypothetical ring with eight BPMs at longitudinal positions which are marked with red circles. When the mixed BPM method is employed, a sampling error δ_k is introduced, due to the deviation of the BPM positions from hypothetical locations that divide the circumference of the ring in exactly eight equal parts, marked with blue circles. BPM 1 is set as the reference point.

MultiBPM: Result for CERN-PS

- During the injection process into the CERN PS strong orbit deflectors are activated. In addition to the wanted orbit change this leads also to an unwanted tune change: Needs to be measured
- Single BPM measurements do not have enough time resolution at high frequency resolution

→ use several BPMs

With remarkable resolution for 40 turns

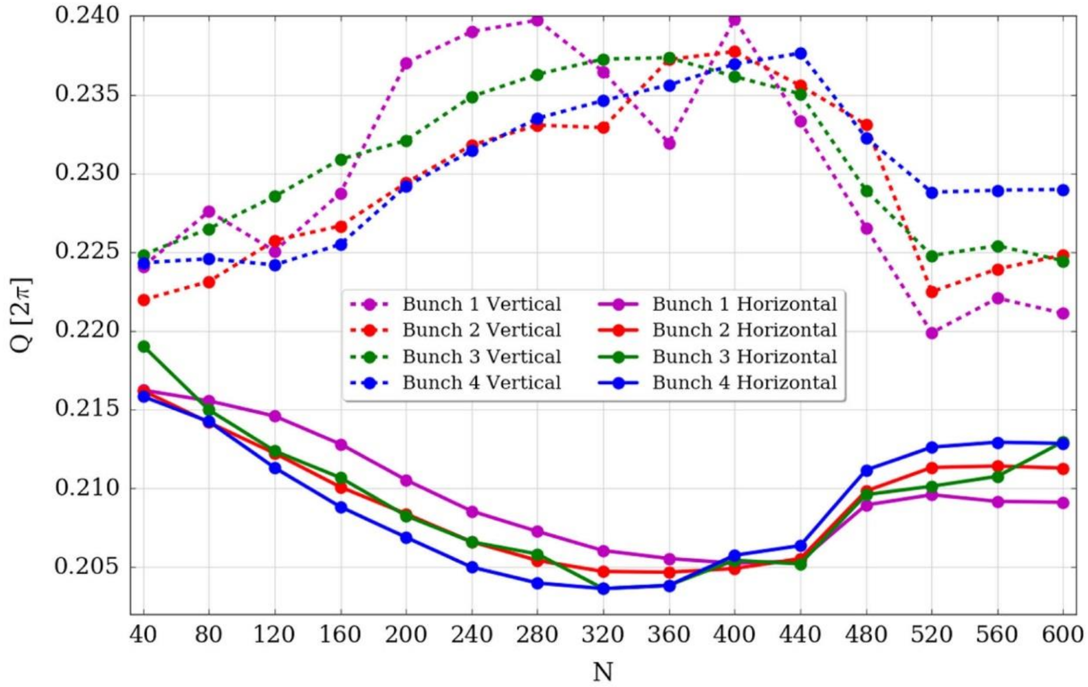


FIG. 20. Instantaneous betatron tune measurements with the mixed BPM method, during the injection process at the PS. The estimation of the horizontal tunes is shown in thick lines and of the vertical tunes in dashed lines. The analysis is performed for four bunches (bunch 1 in magenta, bunch 2 in red, bunch 3 in green, and bunch 4 in blue) by using a sliding window of 40 turns.

Special case: no spectral leakage

1. Introduction and outline

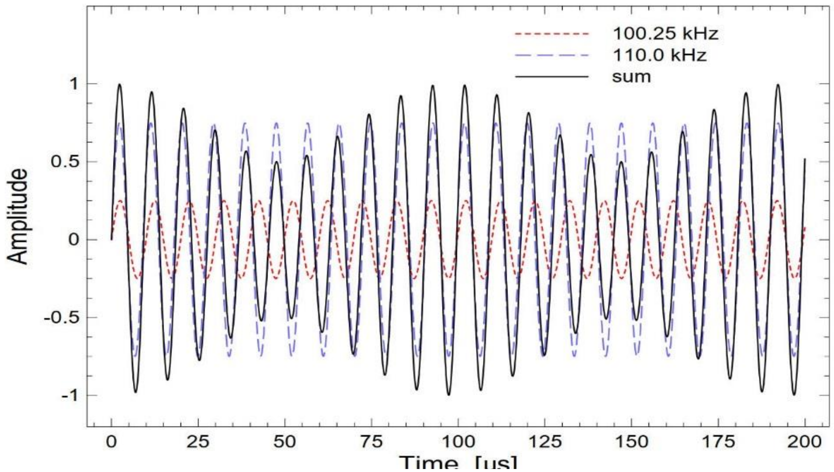


Fig. 1-1. Signal $s(t)$ (black solid line) of unitary amplitude contains two sinusoidal components: the component of interest, $s_{in}(t)$ (red dashed line), whose frequency is to be measured, and an undesirable component, $s_{bg}(t)$ (blue dashed line), considered as a simplified background. Component frequencies are $f_{in} = 100.25$ kHz and $f_{bg} = 110.0$ kHz, their amplitudes $A_{in} = 0.25$ and $A_{bg} = 0.75$.

All pictures: M.Gasior

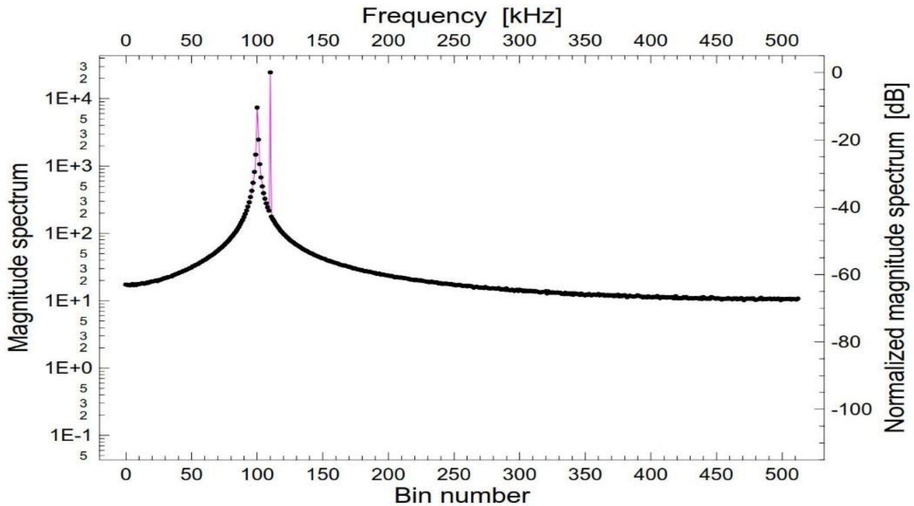


Fig. 1-3. The magnitude spectrum of $N = 1024$ samples of the example signal shown in Fig. 1-2. The smaller spectral peak corresponds to the input component and the bigger to the background one. The input component peak is biased by the spectral leakage effect, while the background peak is narrow and biased only by the spectral leakage of the input component. The right vertical axis is scaled in dB with respect to the highest peak.

The FFT of the so called background signal has no spectral leakage!!!!

Special case continued:

- In the shown example the following relation holds:

$$\frac{f_{background}}{f_{sampling}} = \frac{110}{1024} = \frac{M}{N} \text{ (ratio of rational numbers)}$$

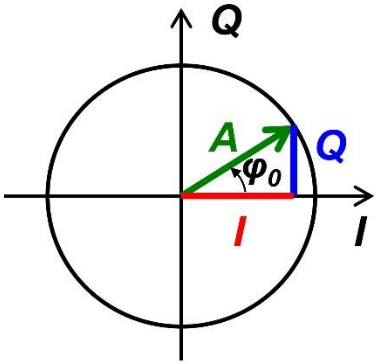
- This means that with the 1024 samples **exactly** 110 full periods of the background signals have been measured.

- The mathematical equivalent is that we have not applied a window function (no truncation), we get as result of the FFT the pure sine wave corresponding to the background frequency.

- In accelerators we often know the frequency of a signal for which we want to measure the amplitude (=multiple of RF frequency) → we can avoid spectral leakage.

- Simplest application IQ-sampling at 4*f (next slide)

I-Q Sampling



- **Vector representation of sinusoidal signals:**
  - **Phasor rotating counter-clockwise (pos. freq.)**

$$y(t) = A\sin(\omega t + \varphi_0)$$

$$y(t) = \underbrace{A\cos\varphi_0}_{=:I}\sin\omega t + \underbrace{A\sin\varphi_0}_{=:Q}\cos\omega t$$

I: in-phase component **Q: quadrature-phase component**

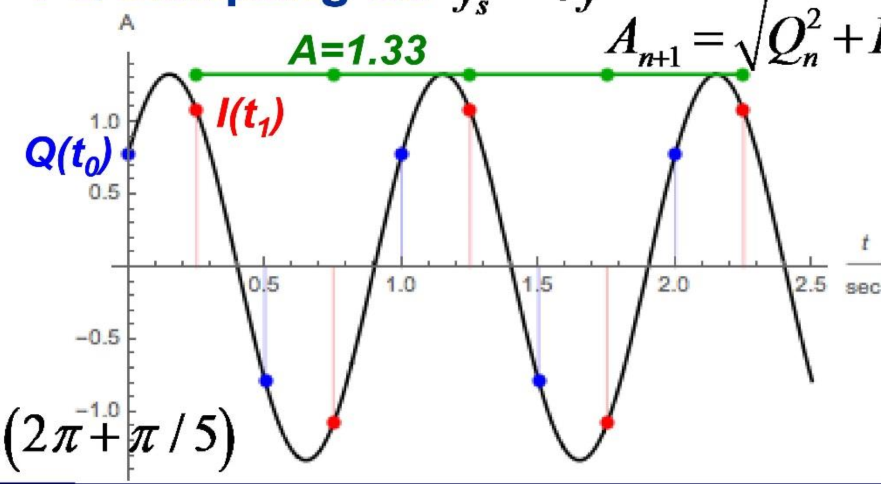
$$y(t) = I\sin\omega t + Q\cos\omega t$$

$$I = A\cos\varphi_0 \qquad A = \sqrt{I^2 + Q^2}$$

$$Q = A\sin\varphi_0 \qquad \varphi_0 = \arctan\left(\frac{Q}{I}\right)$$

- **I-Q sampling at:** $f_s = 4f$

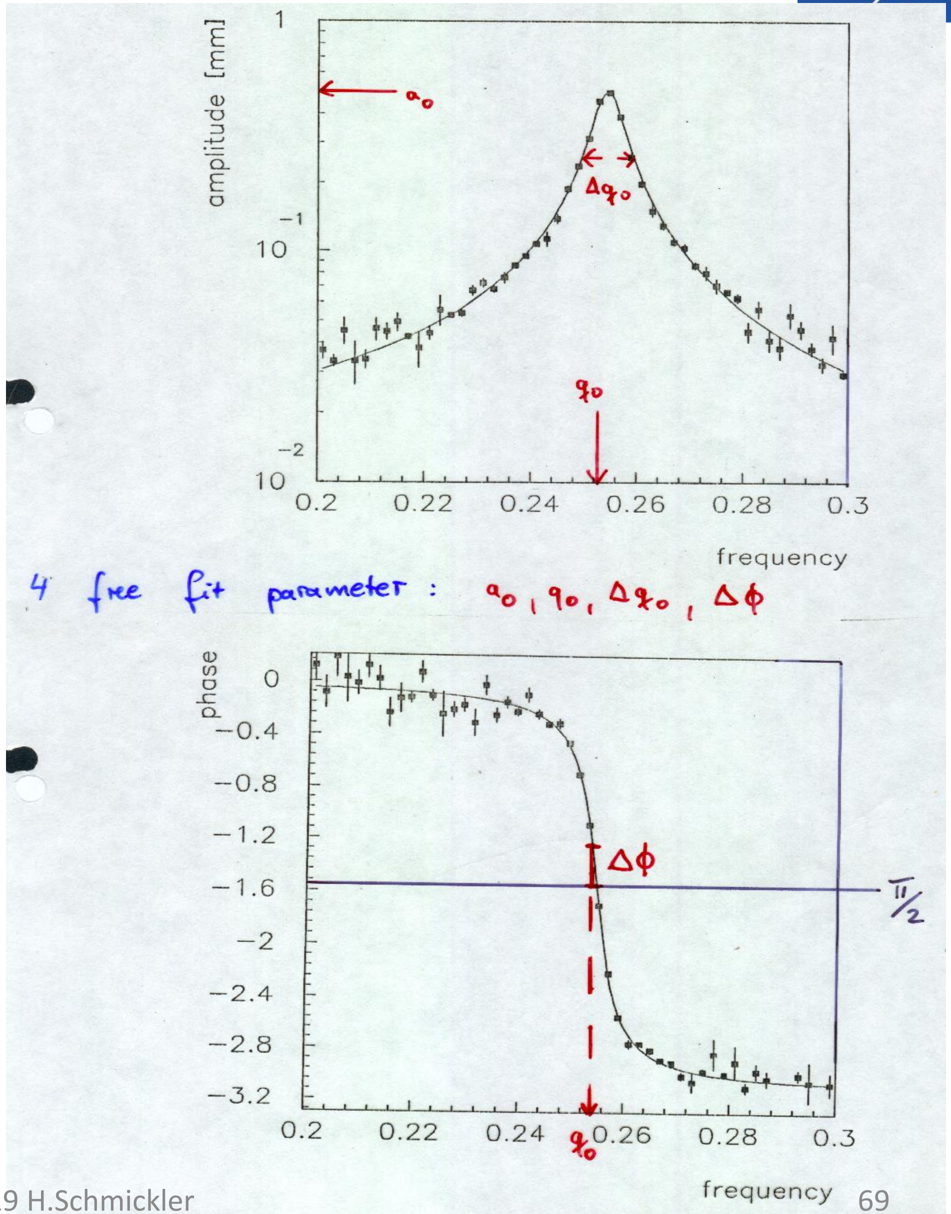
$$A_{n+1} = \sqrt{Q_n^2 + I_{n+1}^2}$$



$$y(t) = 1.33\sin(2\pi + \pi/5)$$

Other method: Network analysis

1. Excite beams with a sinusoidal carrier
2. Measure beam response
3. Sweep excitation frequency slowly through beam response



Analysis of non-stationary spectra

- Stationary Signal
  - Signals with frequency content unchanged in time
  - All frequency components exist at all times

    → ideal situation for Fourier transform (FT)
    (orthonormal base functions of Fourier transform are infinitely long, no time information when spectral component happens)

- Non-stationary Signal
  - Frequency composition changes in time

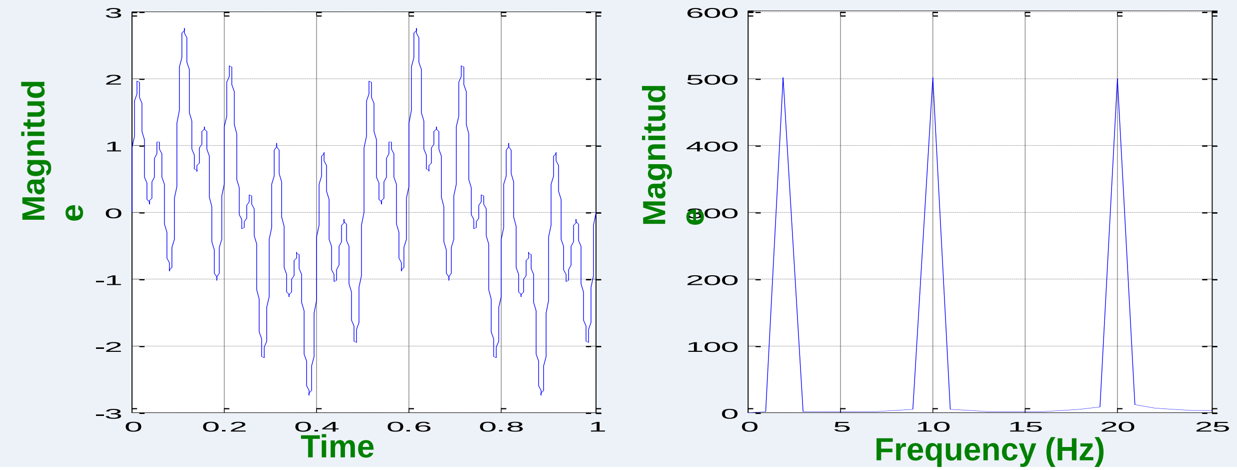
    → need different analysis tools

  - One example: the "Chirp Signal"

Example of simple stationary or non-stationary signals

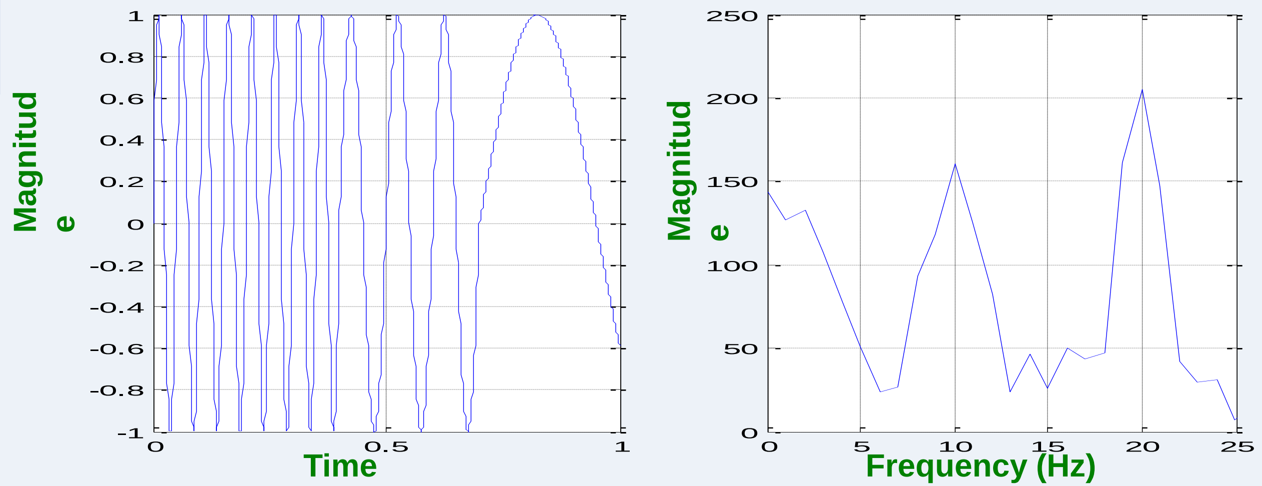
2 Hz + 10 Hz + 20Hz

Stationary



0.0-0.4: 20 Hz +
0.4-0.7: 10 Hz +
0.7-1.0: 2 Hz

Non-Stationary

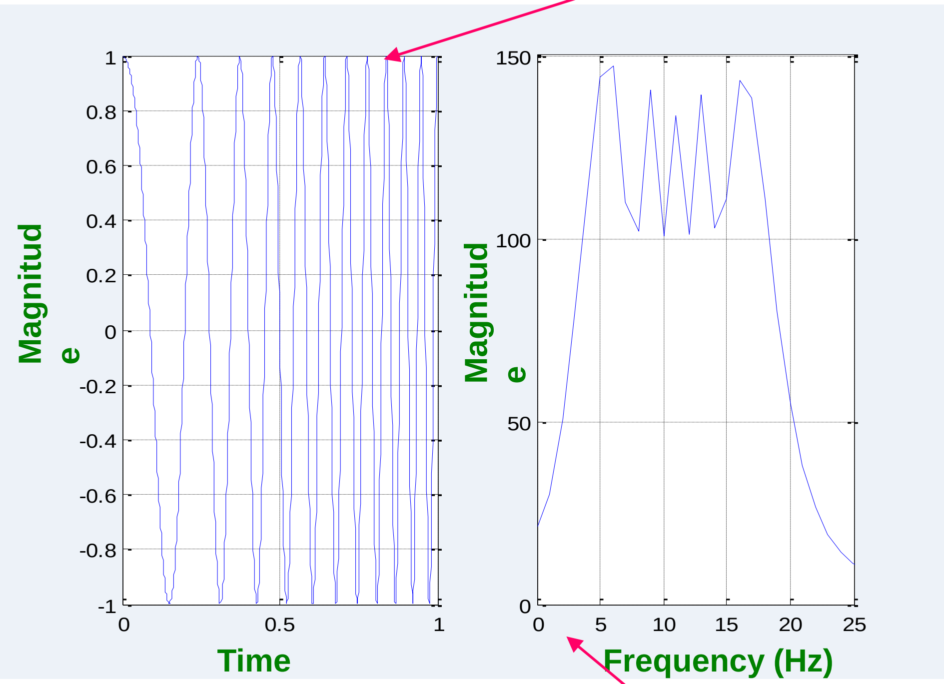


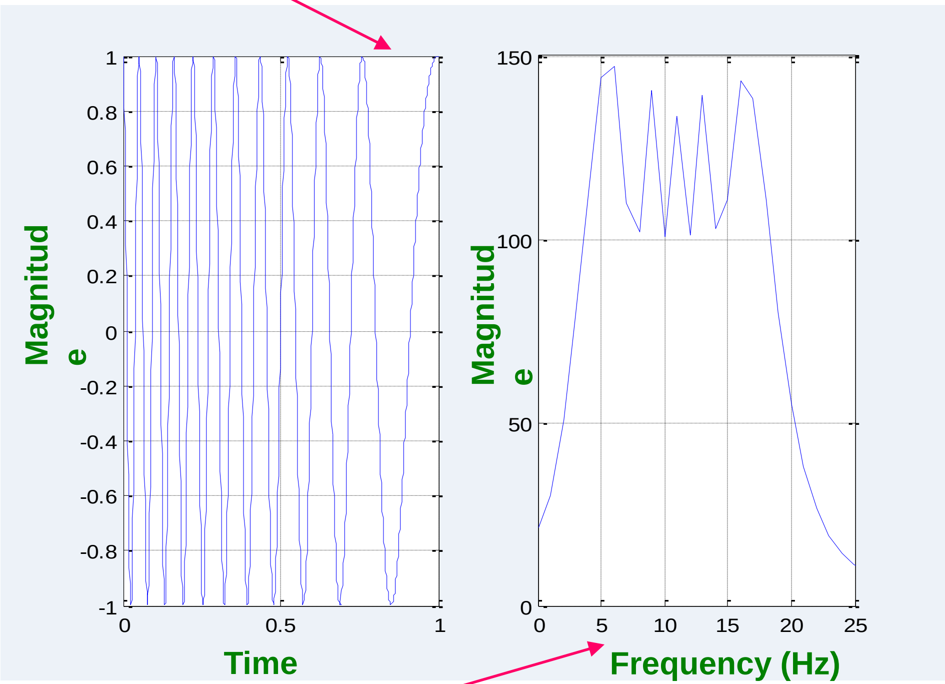
Upward or downward chirp

linear chirp: 2 Hz to 20 Hz

linear chirp: 20 Hz to 2 Hz

Different in Time Domain



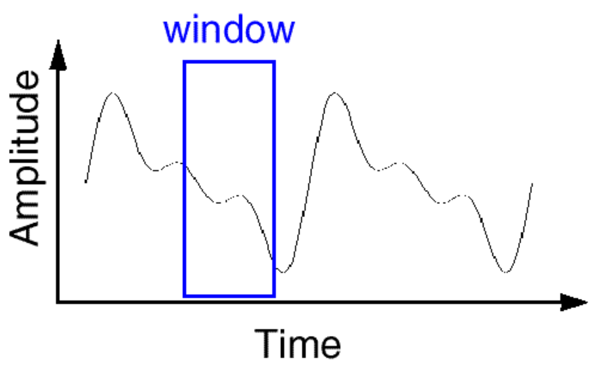


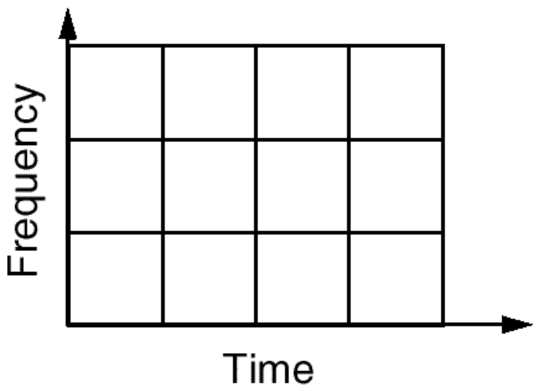
Same in Frequency Domain

At what time a frequency component occurs? FT can not tell!

Short Time Fourier Analysis

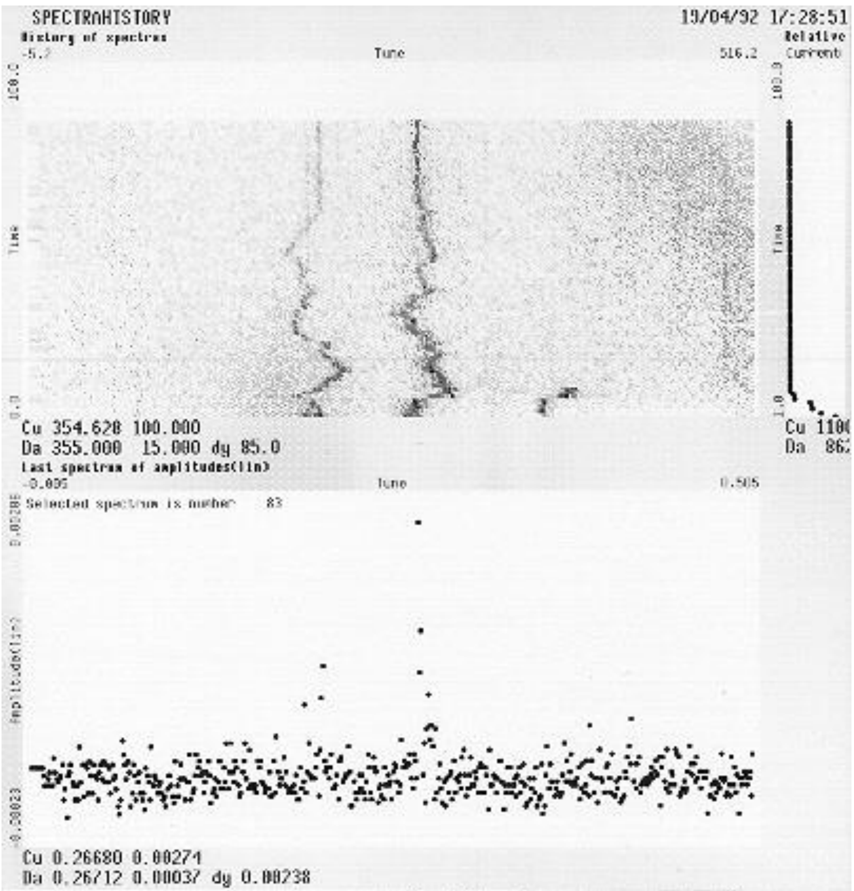
- In order to analyze small section of a signal, Denis Gabor (1946), developed a technique, based on the FT and using *windowing* :
Short Time Fourier Transform:= STFT

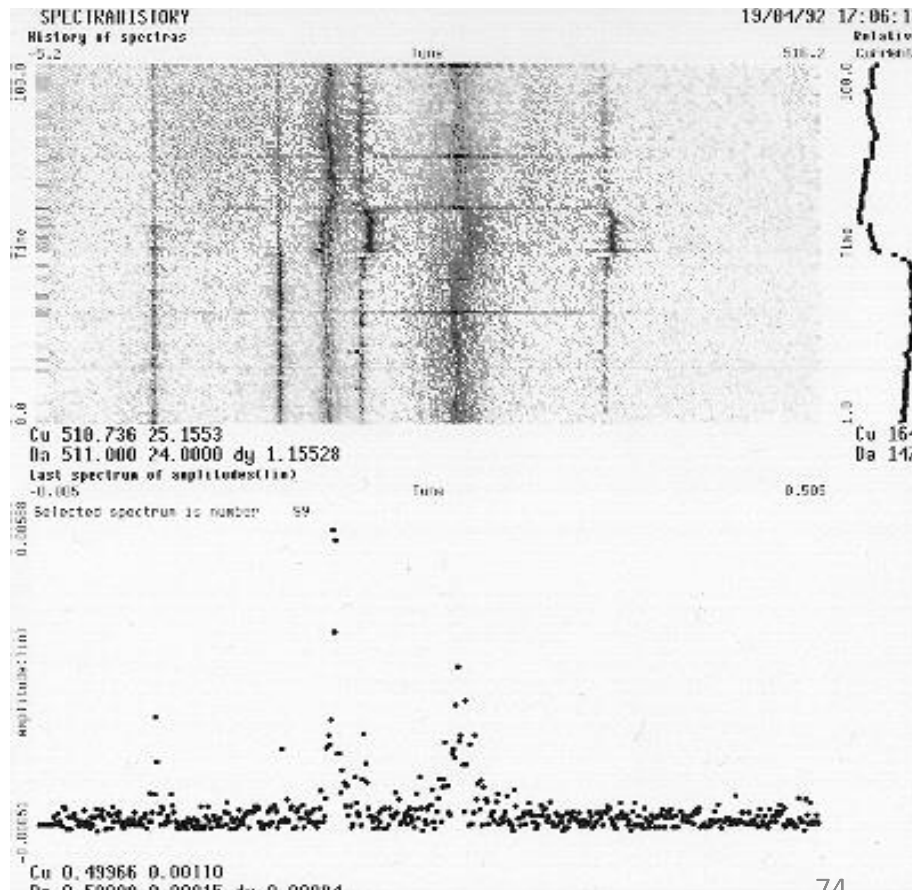




- A compromise between time-based and frequency-based views of a signal.
- both time and frequency are represented in limited precision.
- The precision is determined by the size of the window.
- Once you choose a particular size for the time window - it will be the same for all frequencies.

- To follow betatron tunes during machine transitions we need time resolved measurements. Simplest example:
  - repeated FFT spectra as before (spectrograms)

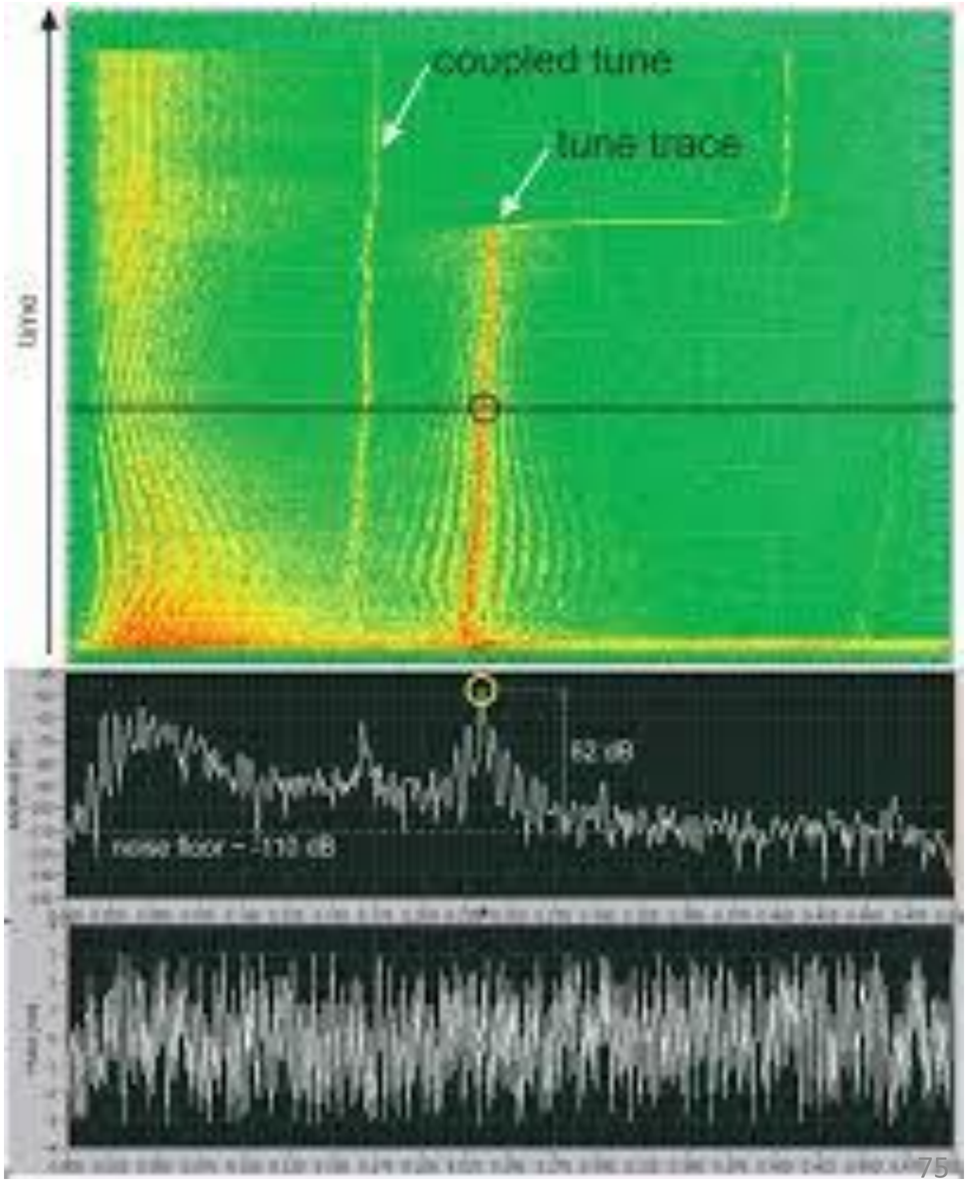




STFT display: Spectogram

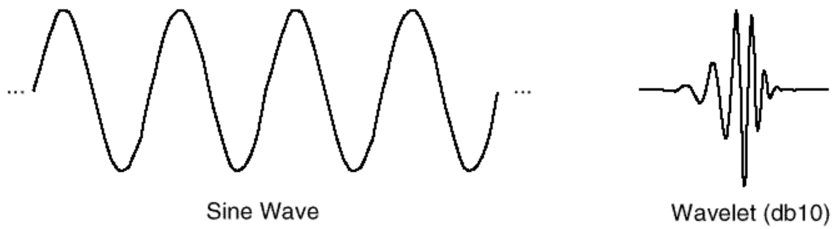
- A very useful form of displaying the result of a STFT is a spectrogram, i.e a 3D view of many consecutive Fourier transforms, which "slide" along the time series of data.

- **Not bad**, but often we wish a more flexible approach between time and frequencies: solution : wavelet analysis

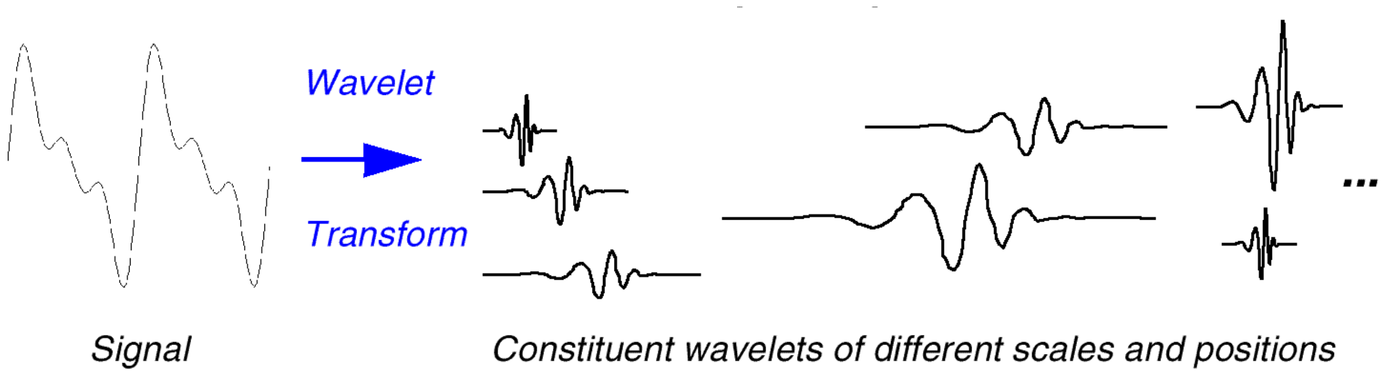


What is Wavelet Analysis ?

- And…what is a wavelet…?

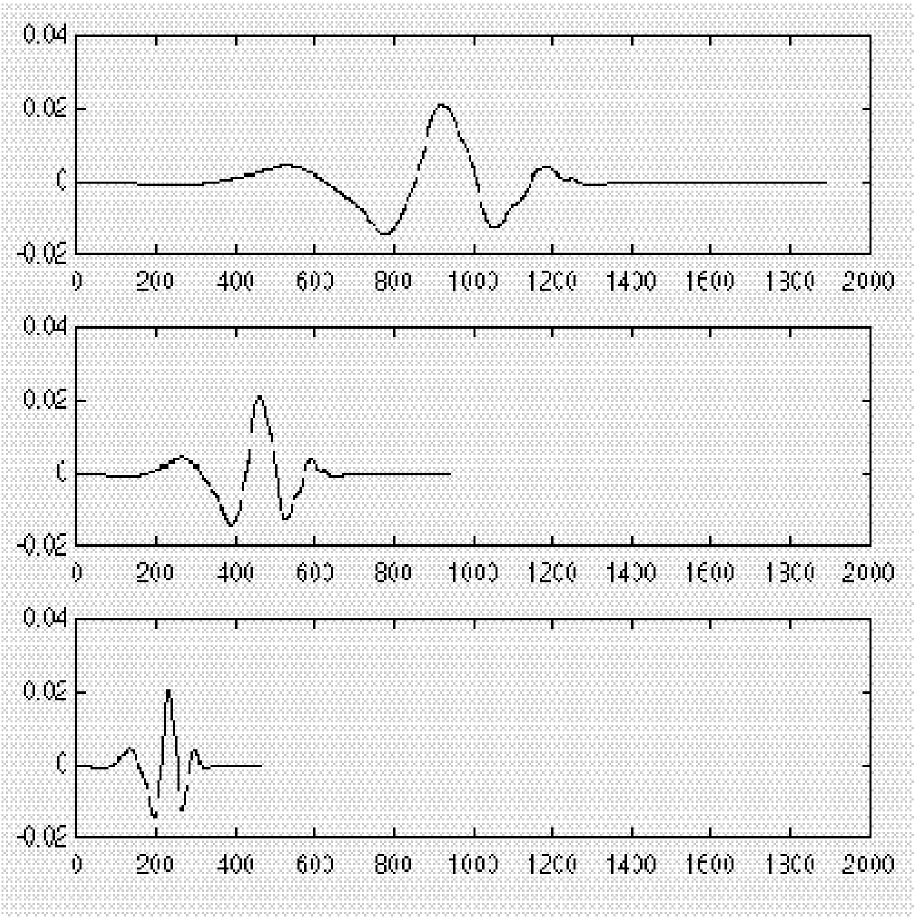


- A wavelet is a waveform of effectively limited duration that has an average value of zero.
- In a Fourier transform (FT) we represent the data by the weighted sum of infinite sine waves with different frequencies.
- In the continuous wavelet transform (CWT) we represent the data by the weighted sum of appropriately scaled and shifted wavelets.



Wavelet Scaling

- Time stretching or frequency scaling:



$$f_{(t)} = \Psi(t) \; ; \; a = 1$$

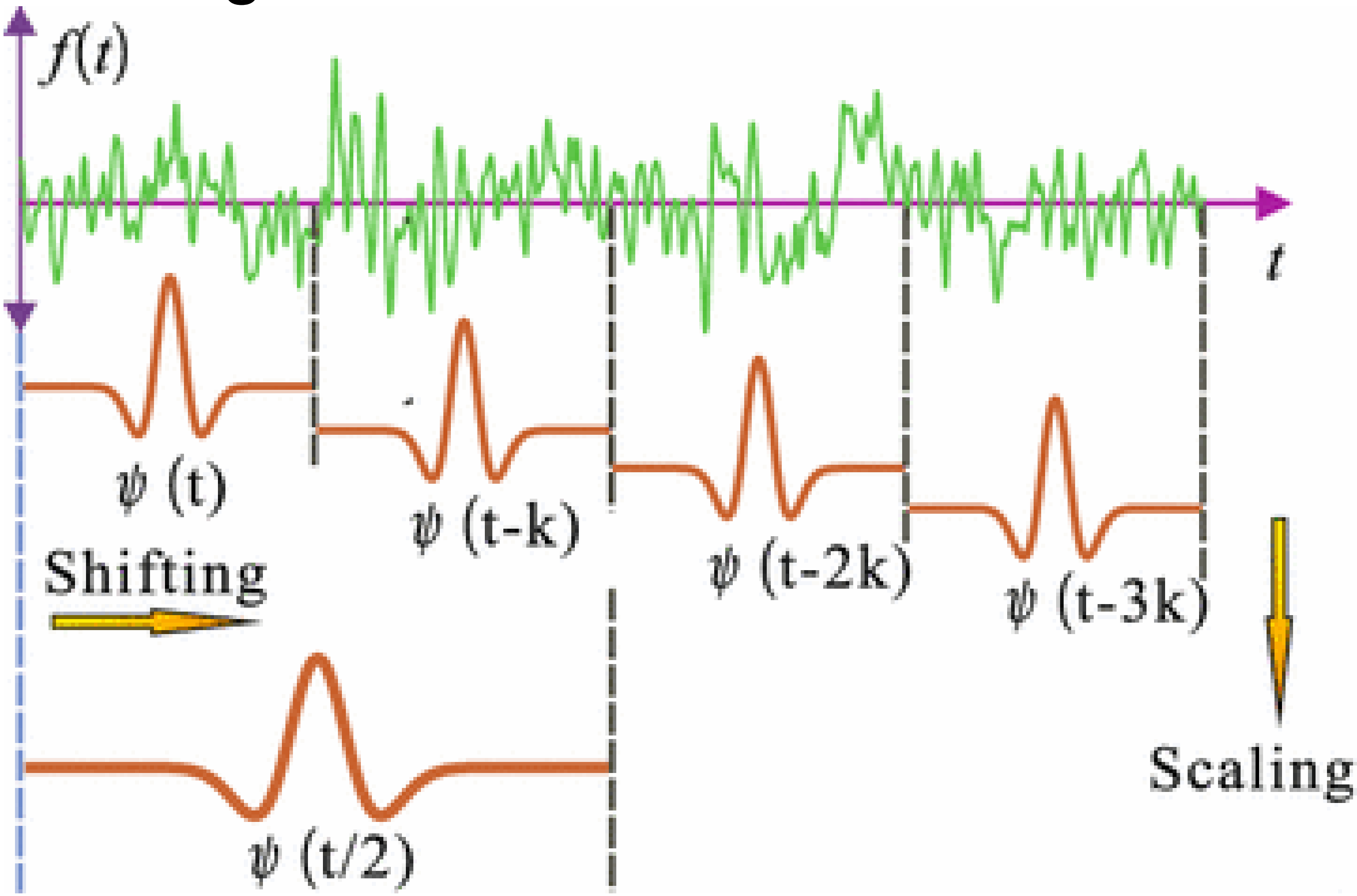
Replace a by s

$$f_{(t)} = \Psi(2t) \; ; \; a = \tfrac{1}{2}$$

$$f_{(t)} = \Psi(4t) \; ; \; a = \tfrac{1}{4}$$

Wavelet Shifting

- Moving the wavelet in time:



Historical Development of wavelet transforms (main contributors)

- Pre-1930
  - Joseph Fourier (1807) with his theories of frequency analysis
- The 1930^{s}
  - Using scale-varying basis functions; computing the energy of a function
- 1960-1980
  - Guido Weiss and Ronald R. Coifman; Grossman and Morlet
- Post-1980
  - Stephane Mallat; Y. Meyer; Ingrid Daubechies; wavelet applications today

CONTINUOUS WAVELET TRANSFORM (CWT)

$$\mathrm{CWT}_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \bullet \psi^*\left(\frac{t-\tau}{s}\right) dt$$

Translation

(The location of the window)

Scale

Mother Wavelet

Remember Fourier transform:

$$F(w) = \int_{-\infty}^{\infty} f(t) e^{-iwt} dt$$

- CWT can be considered as the two-dimensional equivalent to FT.
- The mother wavelets replace the sin/cos functions.
- The scaling of the mother wavelets gives the frequency resolution, the shifting the time resolution.
- There is a large number of different Mother wavelets with different properties.

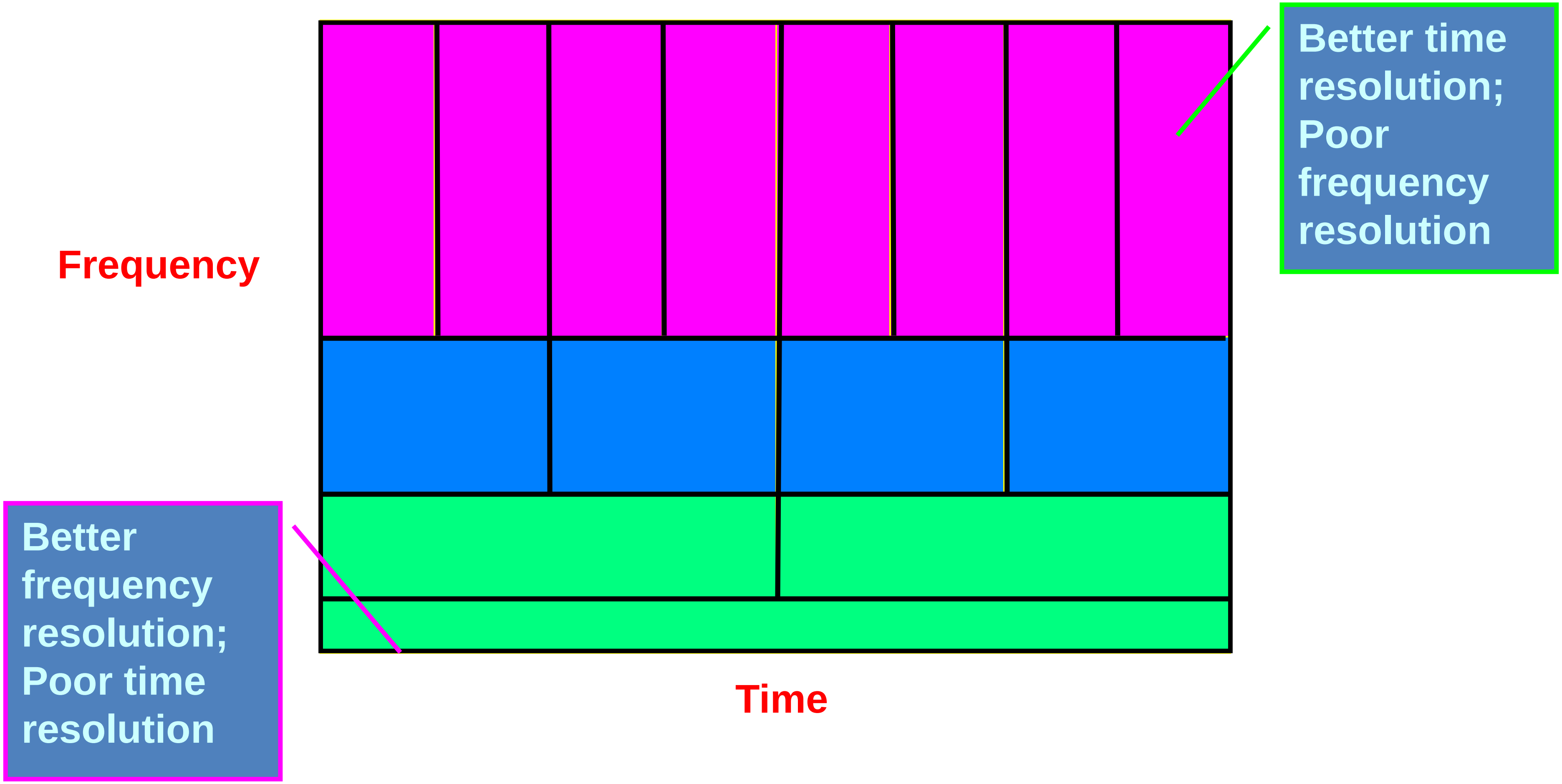
The CERN Accelerator School

Computation of CWT

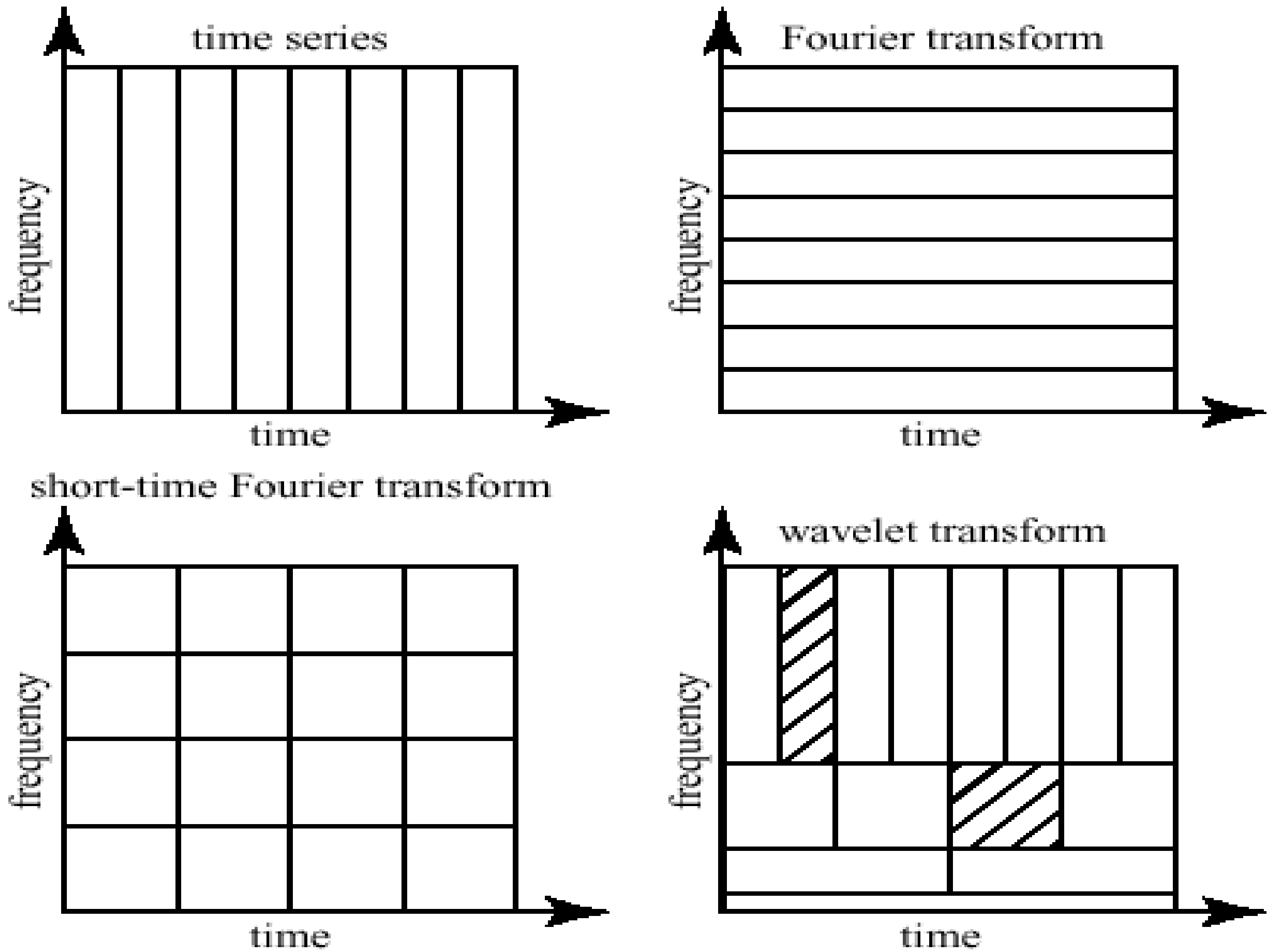
$$\mathrm{CWT}_x^{\psi}(\tau,s)=\Psi_x^{\psi}(\tau,s)=\frac{1}{\sqrt{|s|}}\int x(t)\bullet\psi^{*}\left(\frac{t-\tau}{s}\right)dt$$

Step 1: The wavelet is placed at the beginning of the signal, and set s=1 (the most compressed wavelet);
Step 2: The wavelet function at scale "1" is multiplied by the signal, and integrated over all times; then multiplied by ;
Step 3: Shift the wavelet to *t*= , and get the transform value at *t*= and *s*=1;
Step 4: Repeat the procedure until the wavelet reaches the end of the signal;
Step 5: Scale s is increased by a sufficiently small value, the above procedure is repeated for all *s*;
Step 6: Each computation for a given *s* fills the single row of the time-scale plane;
Step 7: CWT is obtained if all s are calculated.

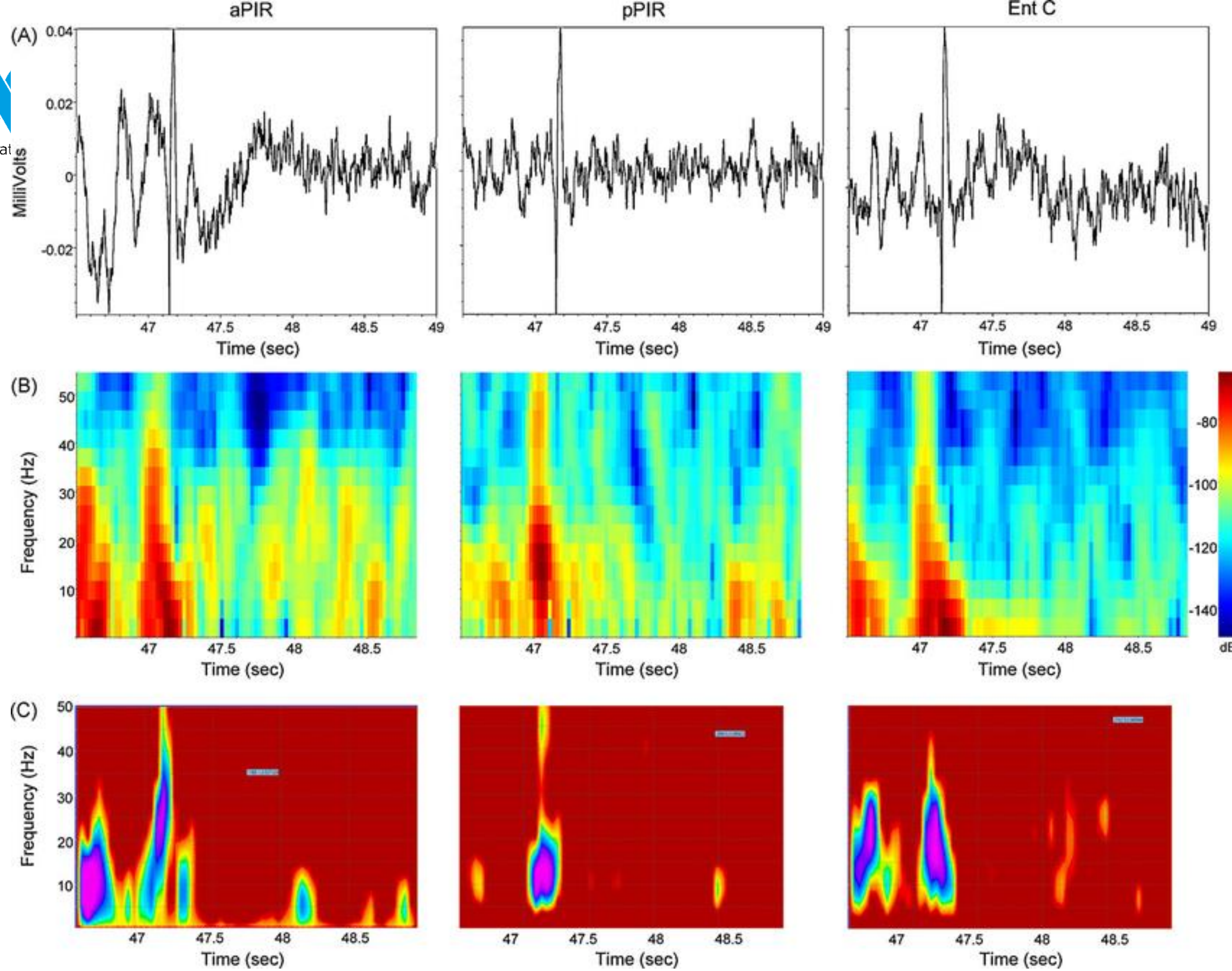
Time & Frequency Resolution of CWT



***COMPARSION** in terms of time and frequency resolution*



From http://www.cerm.unifi.it/EUcourse2001/Gunther_lecturenotes.pdf, p.10



Taken from: Linda Hemmer et al: A putatively novel form of spontaneous coordination
https://www.researchgate.net/publication/23937782, (concerns neural activities)

Which tool to use?

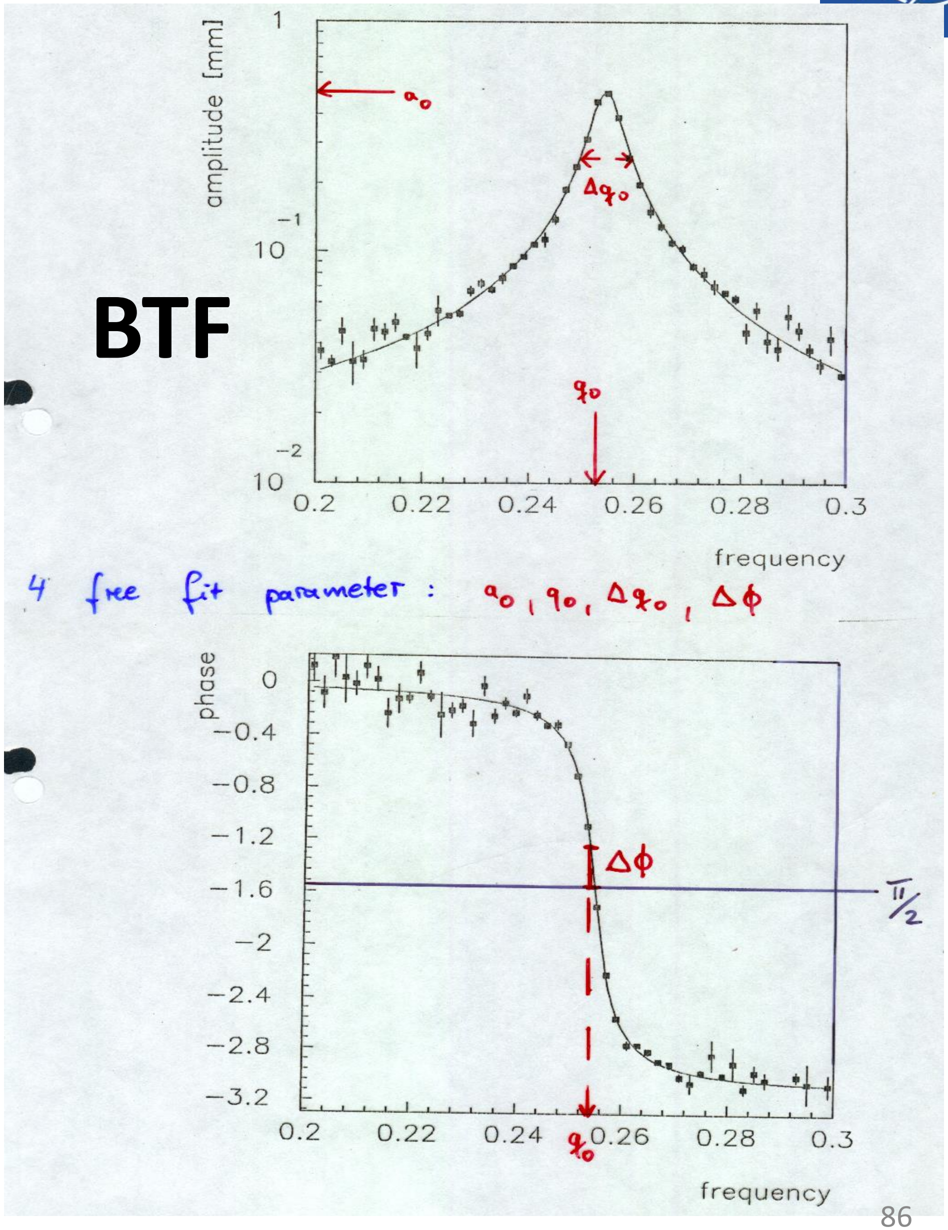
- Stationary Signals:
  windowed FFT with interpolation/fitting.

!!! Depending on the S/N the gain from very sophisticated methods needs to be evaluated!!!

- Time varying Signals:
  - Good S/N + lots of data: STFT (spectrograms)
    i.e. most of the accelerator applications
  - Small S/N + few data: wavelets
    possible case: instabilities at threshold

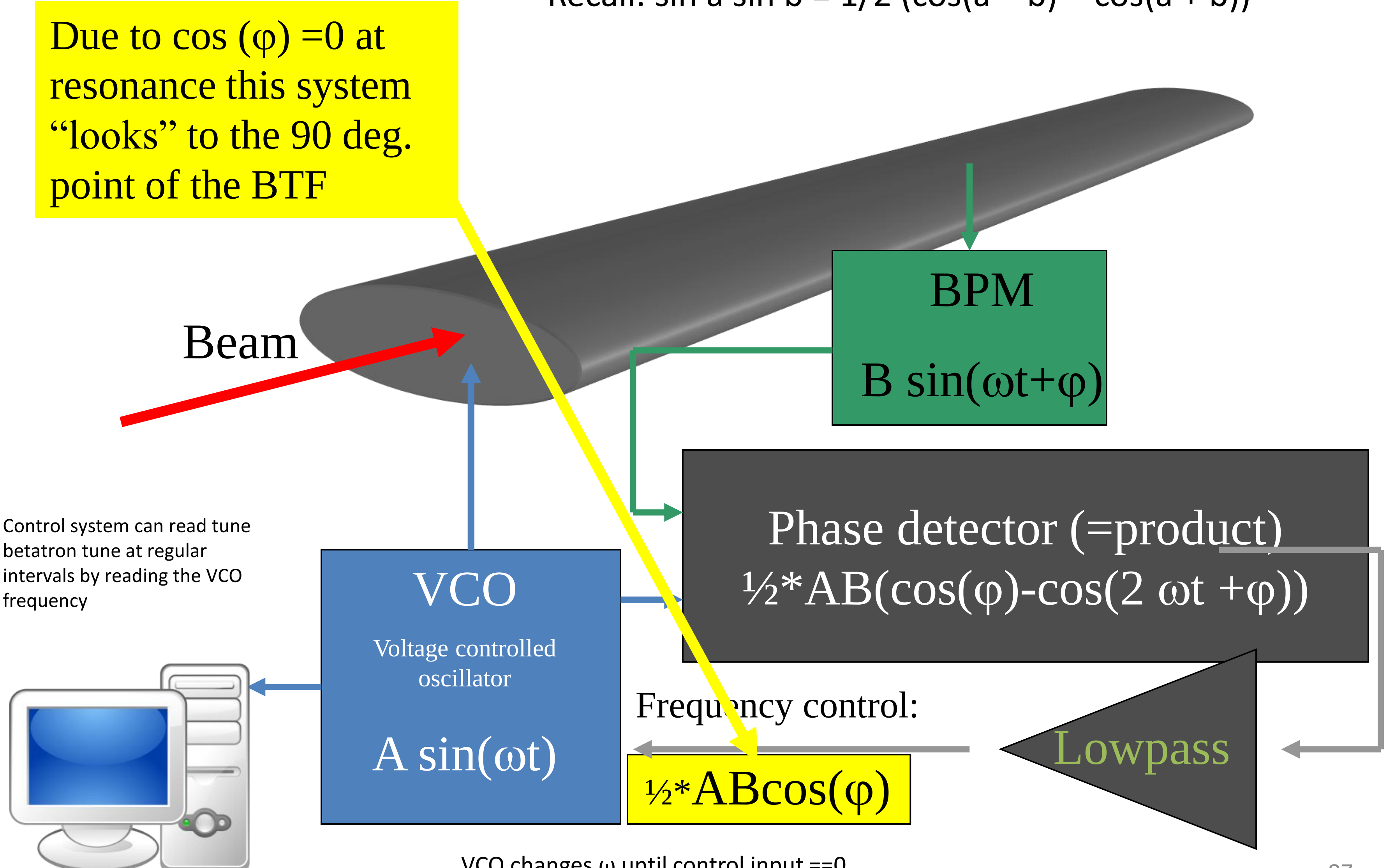
- Alternativly (if not complete spectral information is required): PLL tune tracking → next slides recall: Network analysis (BTF:=Beam transfer function)

1. So far all methods use exclusively the amplitude information (in the case of self excited oscillations this is the only way)
2. But if you drive through an external force for example a betatron oscillation, you can use the phase between the exciter and the beam response as observable



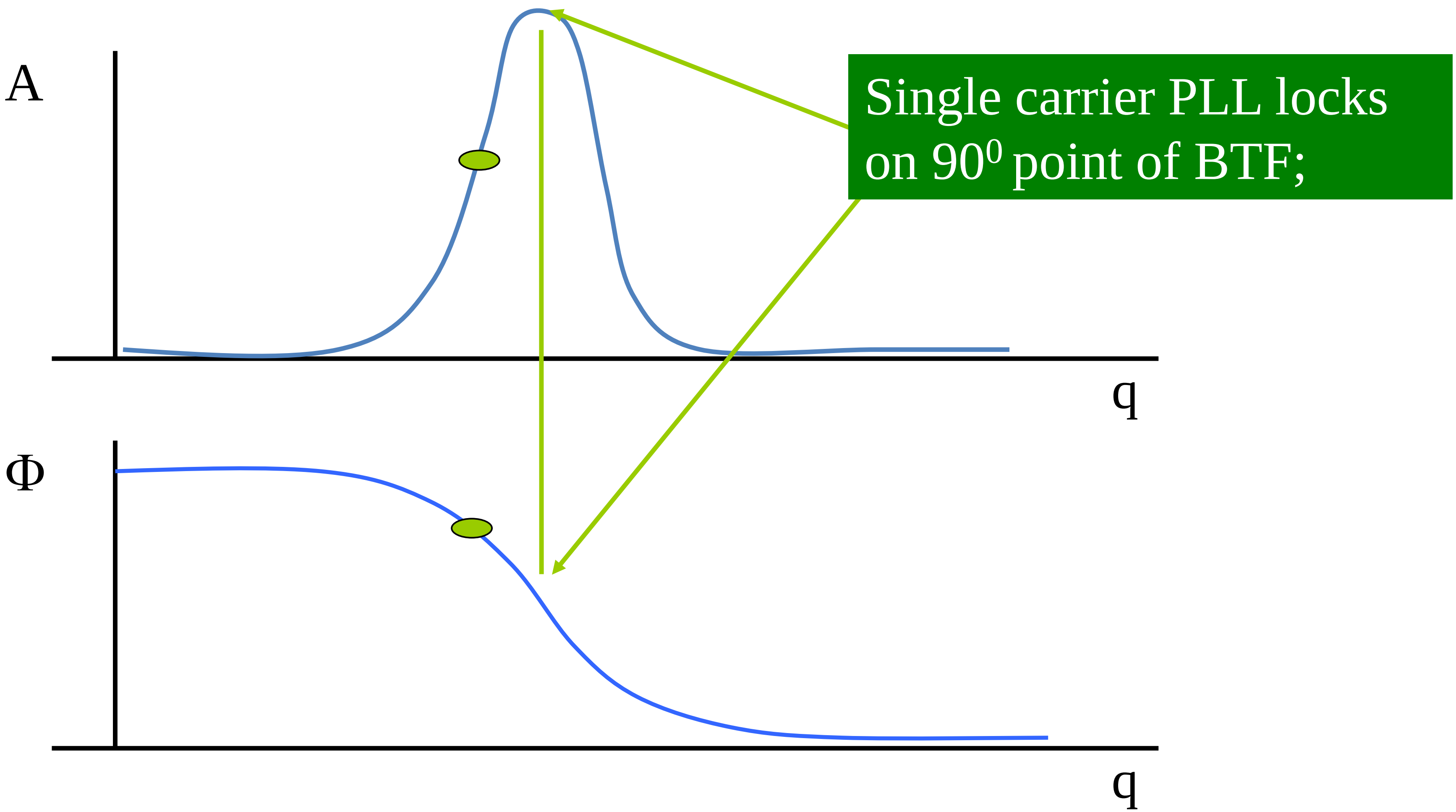
Principle of PLL tune measurements

Recall: sin a sin b = 1/2 (cos(a − b) − cos(a + b))

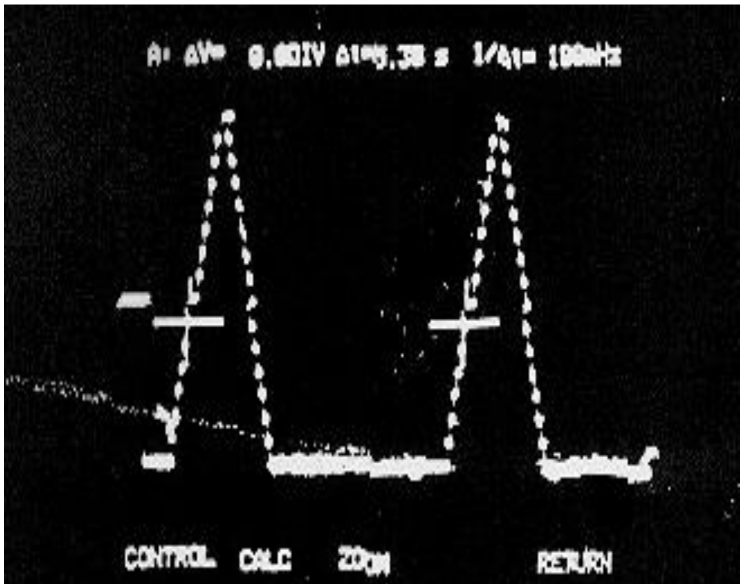


VCO changes ω until control input ==0

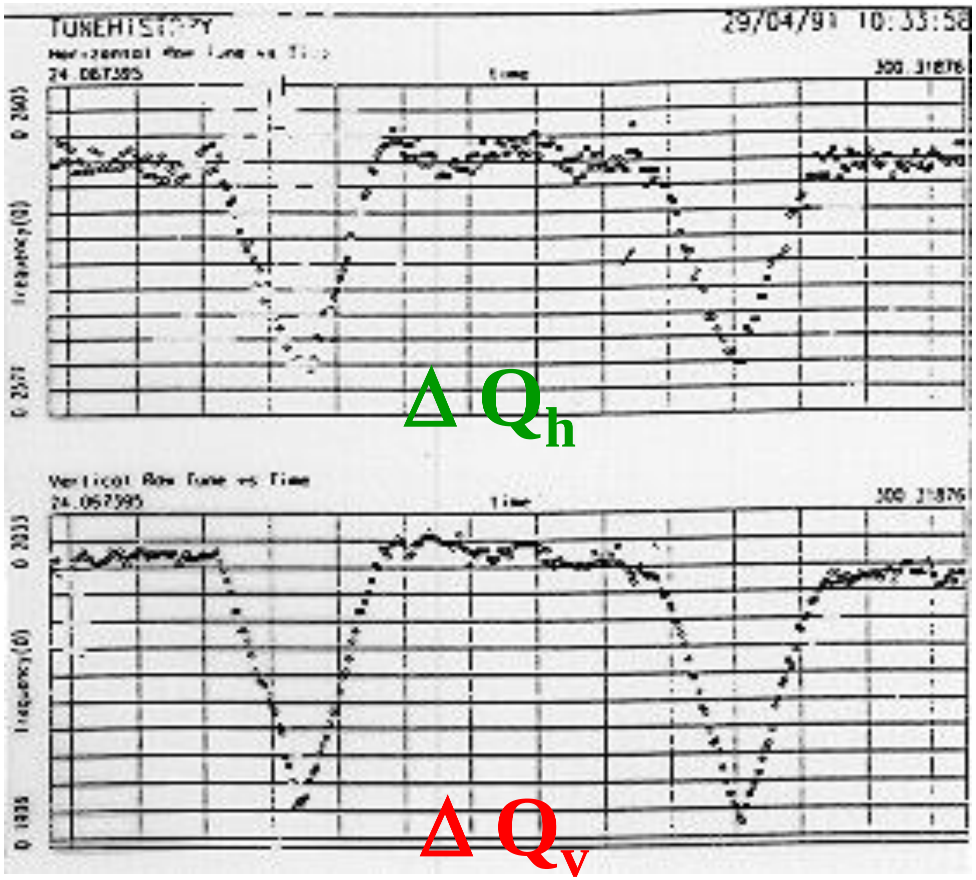
Illustration of PLL tune tracking



Q’ Measurement via RF-frequency modulation (momentum modulation)

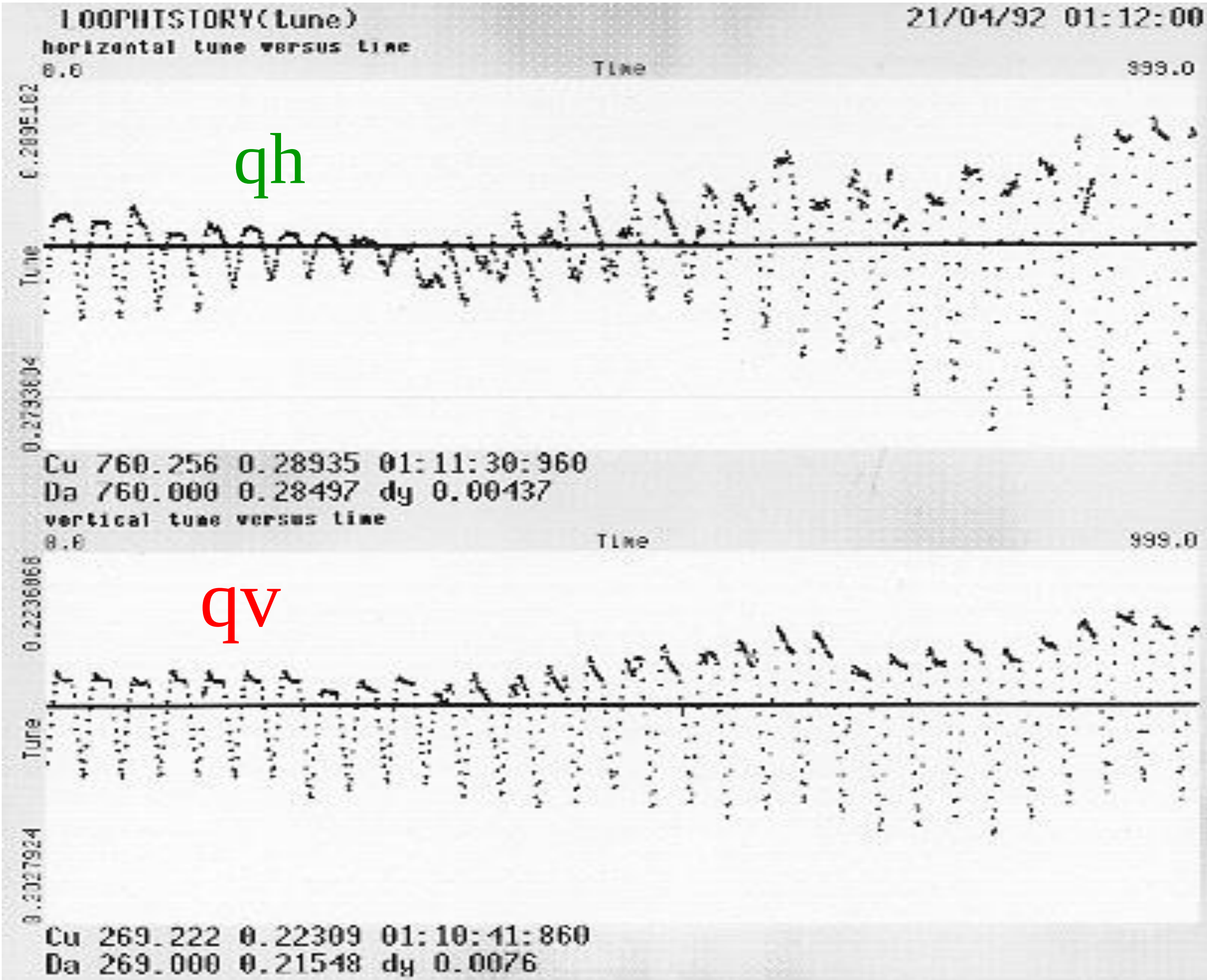


Applied Frequency Shift
Δ F (RF)



Amplitude & sign of chromaticity calculated from continuous tune plot

Measurement example during changes on very strong quadrupoles in the insertion: LEP β-squeeze



Summary

- Single beam passage in a detector produces a signal with a continuous frequency spectrum. The shorter the bunch, the higher the frequency content.
- Repetitive bunch passages produce a line spectrum. They are called revolution harmonics.
  Details of the bunch pattern, differences in bunch intensities etc. determine the final spectral distribution.
- Transverse or longitudinal oscillations of the bunch around the equilibrium produce sidebands around all revolution harmonics.
- These sidebands are used for the measurement of the betatron tunes or the synchrotron tune.

- The standard tool for obtaining spectral information is a Fourier transform (FFT) of the time sampled signals.
- Windowing and interpolation allow higher resolution measurements.
- Spectograms or STFTs are consecutive FFTs of larger datasets, which allow to follow time varying spectra.
- Wavelet diagrams are an alternative analysis tool.
- Phase locked loops can be used for continuous tune tracking.

Appendix I: Python Code for bunch pattern display

Appendix Ia: Python code for bunch pattern simulation 1st part

```
import numpy as np
from numpy import fft
import matplotlib.pyplot as plt

N=16384

NBUNCH=100
sigmax = 0.5


deltax=10

T=1/N

NLEFT=-50
NRIGHT=50

x1= np.linspace(NLEFT,N-NLEFT,N)
xtime=np.linspace(NLEFT,NBUNCH*deltax + NRIGHT,N)
IB=0
y=NBUNCH*np.exp(-(x1*x1)/(2*sigmax*sigmax))
ytime=NBUNCH*np.exp(-(xtime*xtime)/(2*sigmax*sigmax))
y1=0
y2=0
y3=0
ytime=0

while True:

  y1=y1+np.exp(-(x1-IB*deltax)*(x1-IB*deltax)/(2*sigmax*sigmax))
  ytime=ytime+np.exp(-(xtime-IB*deltax)*(xtime-IB*deltax)/(2*sigmax*sigmax))
  IB=IB+1
  if IB==NBUNCH:
    break
```

Appendix Ib: Python code for bunch pattern simulation 2nd part

- fft y=(fft.fft(y))
- ffty1=(fft.fft(y1))
- x2=np.linspace(0.0,500,N/2)
- y2=2.0*np.abs(ffty1[:N//2])/float(N)
- y3=2.0*np.abs(ffty[:N//2])/float(N)

- plt.rcParams["figure.figsize"] = [15,4]
- plt.subplot(1,2,1)

- plt.plot(xtime,ytime,'b-')

- plt.ylabel('amplitude')
- plt.xlabel('time [nsec]')

- plt.subplot (1,2,2)

- plt.plot (x2,y3,'r-')
- plt.plot (x2,y2,'b-')
- plt.ylabel('amplitude')
- plt.xlabel('frequency [MHz]')

- plt.tight_layout()
- plt.savefig ('whatever.png')

- plt.show()