Lidar Data Pre-Processing

Giuseppe D’Amico

giuseppe.damico@cnr.it

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System (D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU

Ministero dell’Università e della Ricerca

Italiadomani

Lidar Data Pre-Processing and Optical Processing

Processing of raw lidar data involve two main steps

- **Data Pre-Processing**
  - Corrections of instrumental effects to minimize systematic uncertainties
  - Corrections of other effects (for example removal of atmospheric background)
  - Smoothing/averaging to improve SNR
  - Cloud screening
- **Optical Processing**
  - Apply optical retrieval algorithms on pre-processing signals to get vertical profiles of aerosol properties (extinction, backscatter, lidar ratio,…)

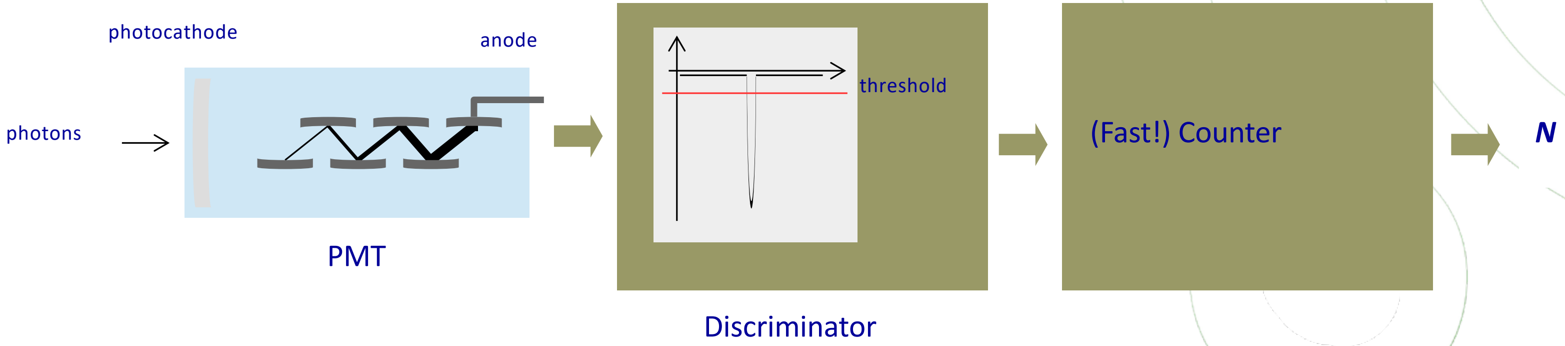
Lidar Data Pre-Processing

- **Dead-time correction**
  - Models for photon-counting systems
  - Dead time measurement
- **Trigger delay correction**
  - How to correct and measure trigger delay
- **Background subtraction**
  - Atmospheric
  - Electronic
- **Merging of lidar signals**
- **Error calculation**

Dead-time correction

ITINERIS

Typical photon-counting chain



- The number of counted pulses (***N***) is proportional to the incident intensity
- Very efficient in detecting weak signals (low count-rate)
- To handle with care in case of strong signals (high count-rate)
- ***N*** is equal to the number of incident photons only if:
  - the PMT output pulses width is negligible
  - all the electronics devices (amplifiers, discriminators, counters) are very fast

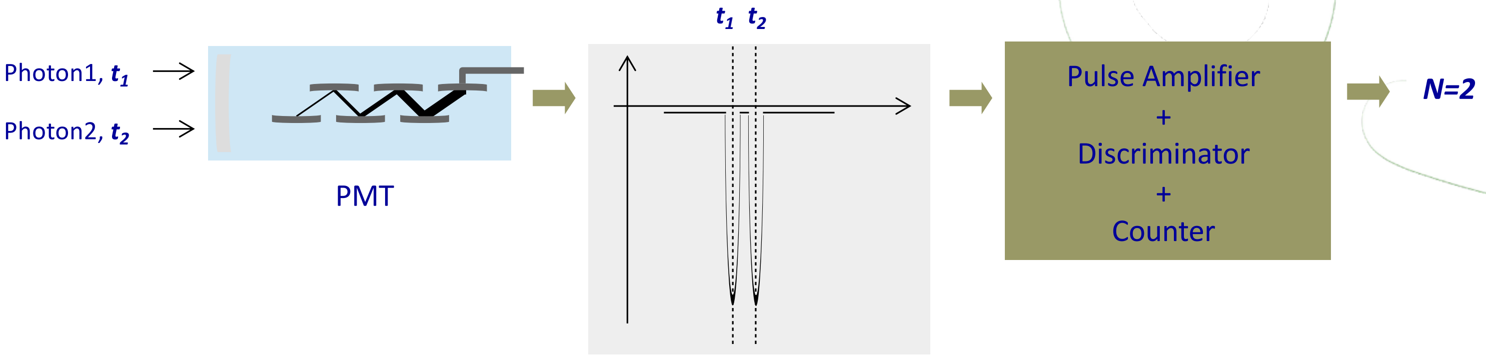
Dead-time correction

Photons count-rate: $C_{ph}=(t_2-t_1)^{-1}$

Electronic max. count-rate: C_{el}

Output pulse width: Δ_p

1st Case: Fast electronic $C_{el} > C_{ph}$ Narrow pulses $\Delta_p < (t_2-t_1)/2$



Photons counted correctly!

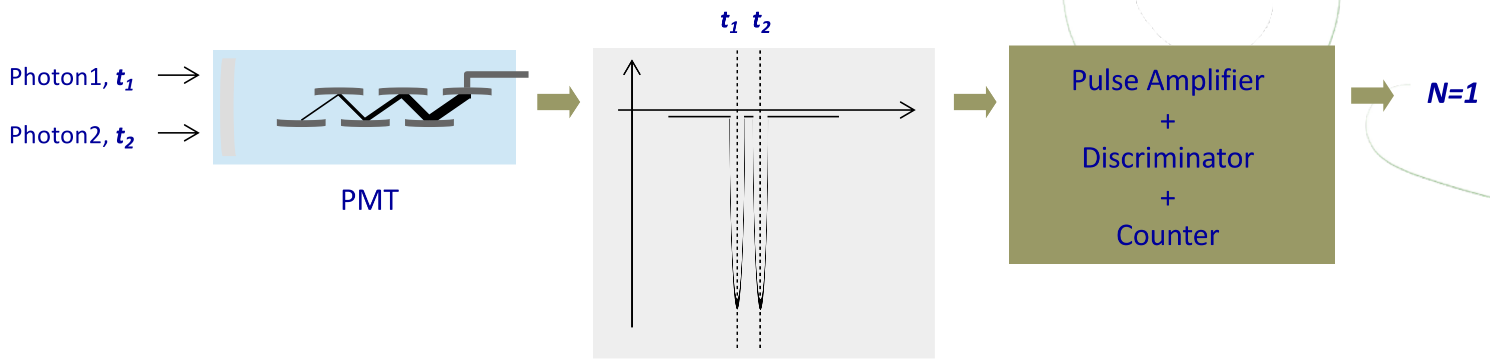
Dead-time correction

Photons count-rate: $C_{ph}=(t_2-t_1)^{-1}$

Electronic max. count-rate: C_{el}

Output pulse width: Δ_p

2nd Case: Slow electronic $C_{el} < C_{ph}$ Narrow pulses $\Delta_p < (t_2-t_1)/2$



Photons underestimated!

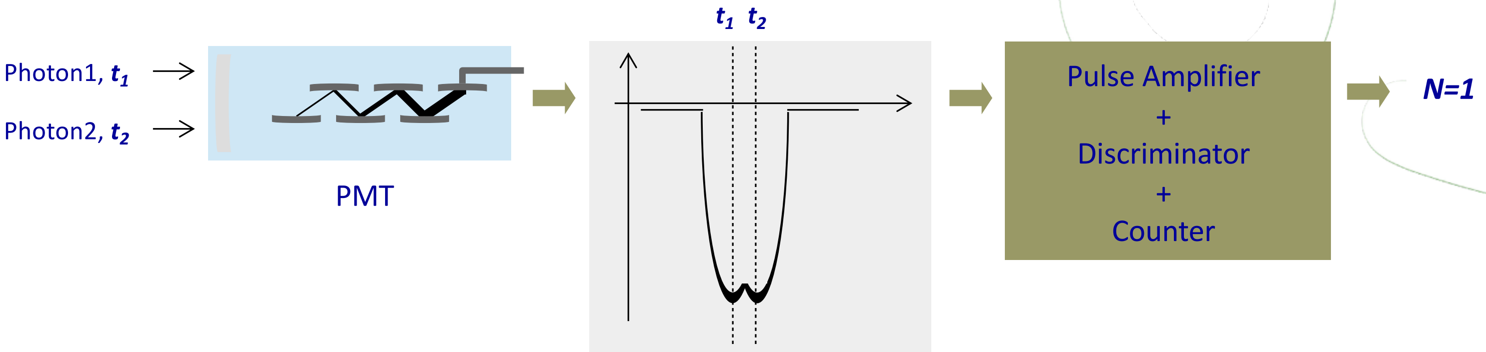
Dead-time correction

Photons count-rate: $C_{ph}=(t_2-t_1)^{-1}$

Electronic max. count-rate: C_{el}

Output pulse width: Δ_p

3rd Case: Fast electronic $C_{el} > C_{ph}$ Broad pulses $\Delta_p > (t_2-t_1)/2$



Photons underestimated!

Dead-time correction

Dead-time definition

Time interval in which two or more separate events (incident photons) cannot be counted individually

- Separate events within the dead time interval are always underestimated
- The amount of underestimation depends on incident count-rate (the more is the count-rate the more is the effect of dead time)
- In lidar signals the dead-time introduces a non-linearity between the number of counted pulses and the actual intensity impacting on the PMT
- Such non-linearity is more evident in the signal backscattered from the near-range where the count-rate is usually high

Dead-time correction

To correct for dead-time we need to model real counting systems

Option 1: Paralyzable systems

System unable to record a second output pulse unless there is a time interval of at least τ (dead-time) between two successive input pulses. If an additional pulse arrives during the response time the dead-time of the apparatus is further extended by τ

$$C_{obs} = C_{real} e^{-\tau C_{real}}$$

Dead-time correction

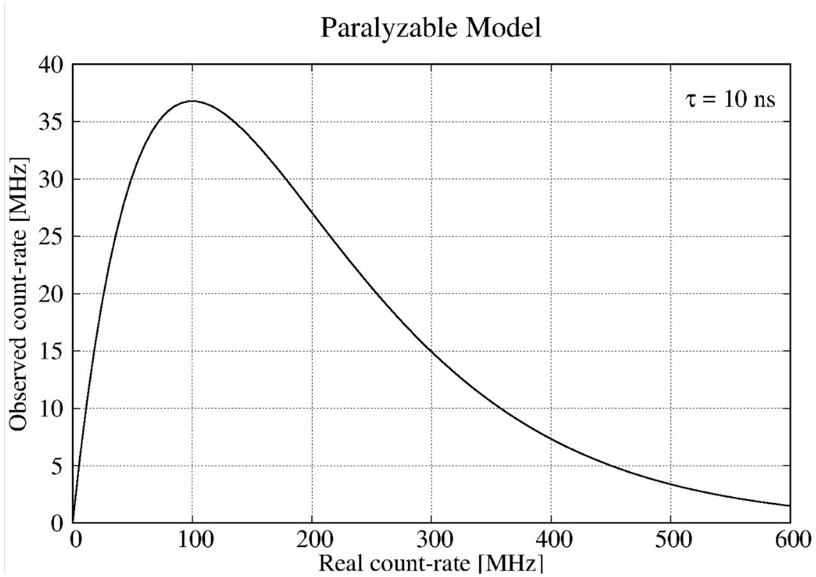
To correct for dead-time we need to model real counting systems

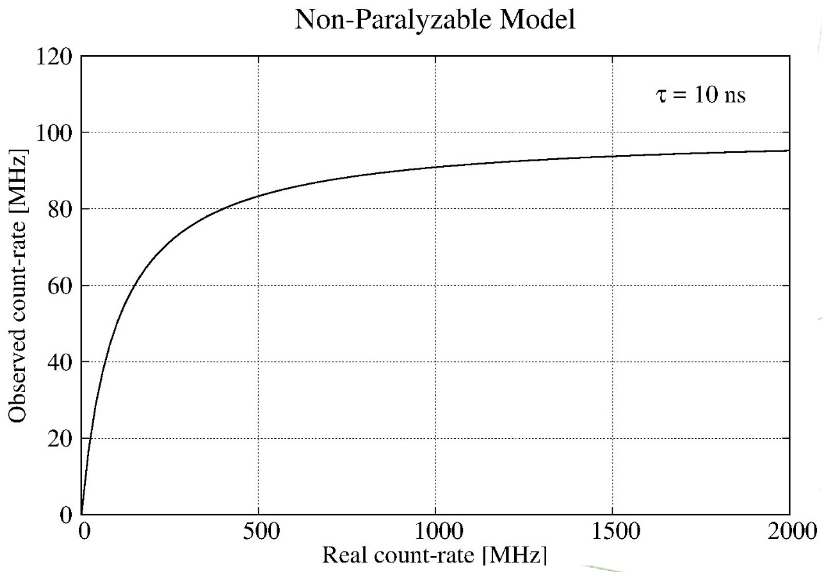
Option 2: Non-Paralyzable systems

System in which the response time τ is independent of the arrival of additional counts. It will asymptotically approach a maximum counting rate (inverse of dead-time value) as the actual count-rate increases

$$C_{obs} = \frac{C_{real}}{1 + \tau C_{real}}$$

Dead-time correction





$$C_{obs} = C_{real} e^{-\tau C_{real}}$$

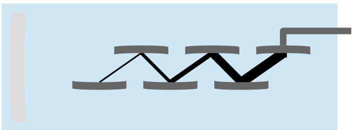
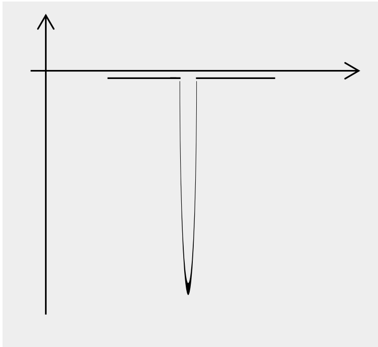
- $C_{obs} \rightarrow 0$ when $C_{real} \rightarrow \infty$
- Maximum at $C_{real} = 1/\tau$

$$C_{obs} = \frac{C_{real}}{1 + \tau C_{real}}$$

- $C_{obs} \rightarrow 1/t$ when $C_{real} \rightarrow \infty$
- $C_{obsl} < 1/\tau$

Dead-time correction

Which model should be used?

Photons →  →  → Pulse Amplifier + Discriminator + Counter → *N*

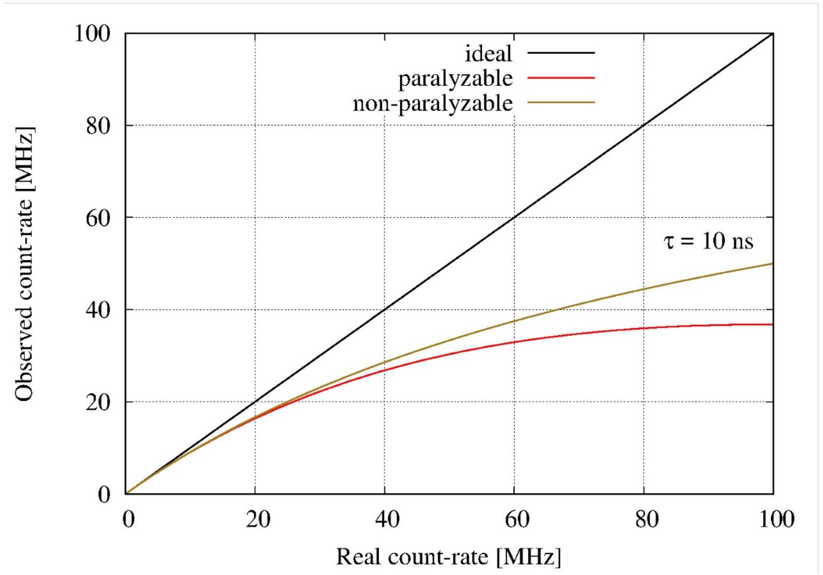
PMT

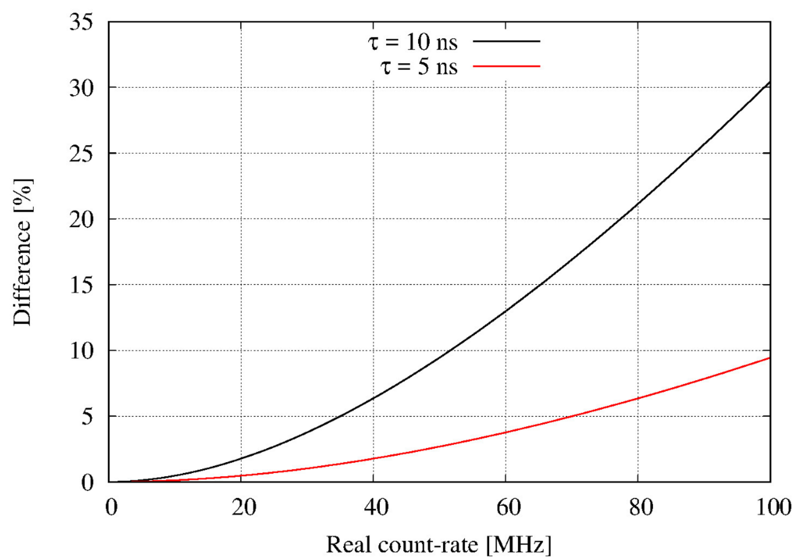
Comparable to a paralyzable system

Comparable to a non-paralyzable system

Response of a real system usually would lie somewhere in between the two models!!

Dead-time correction





- In general, the relative weight of the paralyzable and non-paralyzable contributions is unknown so it is recommended to keep the count-rate below a maximum value assuring a reasonable accuracy in correcting for dead-time.
- The maximum "effective" count-rate depends on dead-time value and can be defined by putting a threshold on the relative difference between paralyzable and non-paralyzable models.
- For example, to keep the accuracy of the dead time correction below 5% we should keep the count-rate below 35MHz for dead time of 10 ns and below 70MHz for dead time of 5 ns.

Dead-time correction

- The accuracy of dead-time correction depends on the accuracy of dead-time measurement
- The dead-time is not necessarily constant over the time
- Usually, the values reported in the specs are not accurate enough and typically they refer to specific part of photon-counting channels (for example PMT only) and not to the whole counting chain.

⬇

- Dead-time should be measured regularly for all photon-counting lidar channels
- How to measure it?

Dead-time measurement

For **ideal** systems (not affected by dead-time) the distribution of the counts is Poissonian:

$$p(n,T) = \frac{\bar{n}^n}{n!} e^{-\bar{n}}$$

where *p(n,T)* is the probability to have *n* counts in the sampling time *T*, and $\bar{n}$ is the mean counts within *T*.

For **real** systems we could estimate the value of dead-time from the deviations of the measured distribution from a pure Poissonian one.

It is possible to show that if the dead-time is negligible with respect to the sampling time *T* the probability *p(n,T)* is given (at first order) by:

$$p(n,T) = \frac{\bar{n}^n}{n!} e^{-\bar{n}} \left[1 + n(\bar{n} - n + 1)\frac{\tau}{T}\right]$$

$$F(n,T) = (n+1)\frac{p(n+1,T)}{p(n,T)} = -2\bar{n}\frac{\tau}{T}n + \left(\bar{n} + \bar{n}^2\frac{\tau}{T}\right)$$

Dead-time can be determined by making a linear fit of *F(n,T)*!

F. A. Johnson et al, Dead-time corrections to photon counting distributions, Phy. Rev. Lett. 16, N. 13, pp. 589-592 (1966)

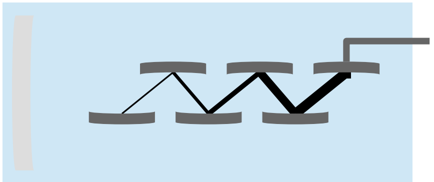
Dead-time measurement

ITINERIS

Experimental set-up (Potenza PEARL system)

Tungsten lamp

ND filters



PMT
Thorn EMI 9202QB

Pulse Amplifier
+
Discriminator

Philips 6908

Counter

Multichannels scales
EG&G ORTEC

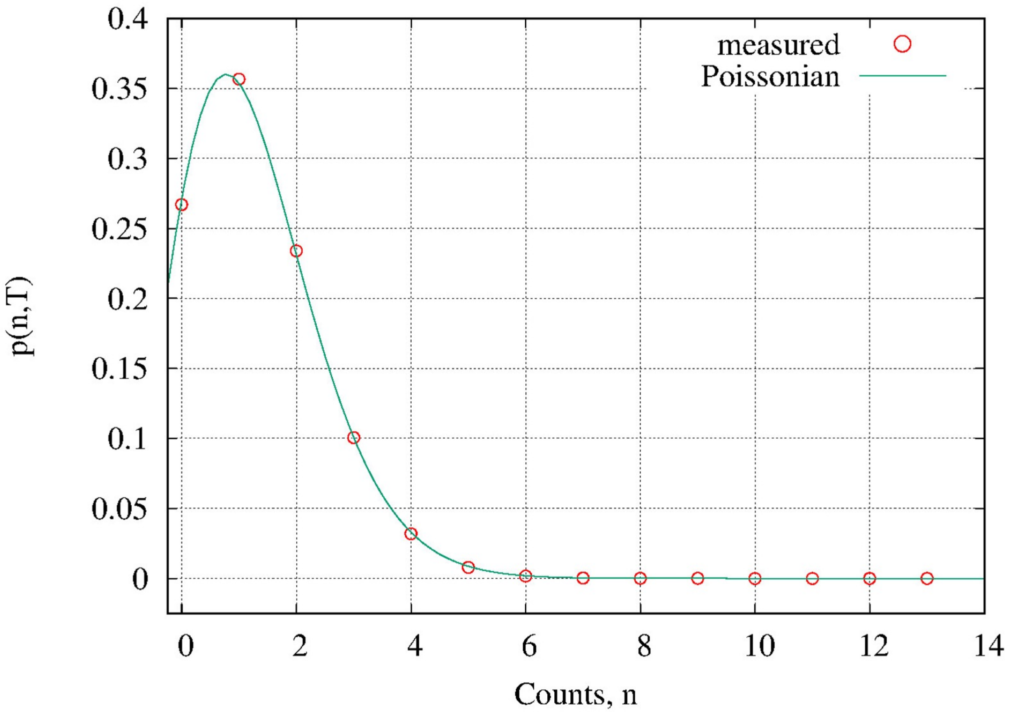
Settings:

Sampling time *T*	1 μS
Total number of samples	327352320
Observed Mean count within *T*	1.306

Dead-time measurement

Results (Potenza PEARL system)

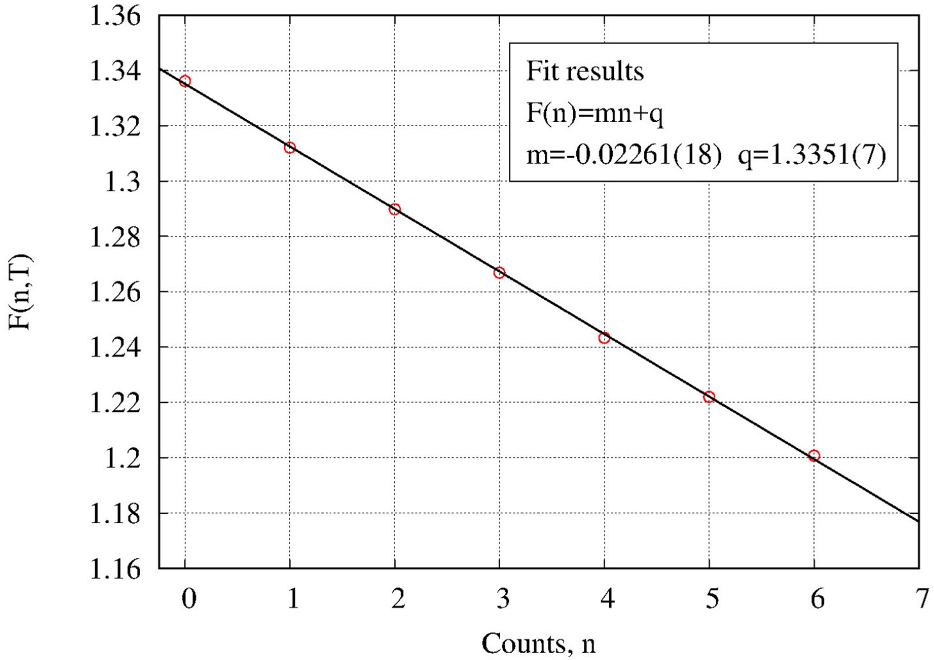
n	**Number of occurrences**	*p(n,T)*
0	87387336	0.26695
1	116765376	0.3567
2	76604416	0.23401
3	32935530	0.10061
4	10431103	0.03187
5	2593883	0.00792
6	528278	0.00161
7	90617	2.77E-4
8	13717	4.19E-5
9	1838	5.61E-6
10	204	6.23E-7
11	17	5.19E-8
12	4	1.22E-8
13	1	3.05E-9



Dead-time measurement

Results (Potenza PEARL system)

$$F(n,T) = (n+1)\frac{p(n+1,T)}{p(n,T)} = -2\bar{n}\frac{\tau}{T}n + \left(\bar{n} + \bar{n}^2\frac{\tau}{T}\right)$$



$$\frac{\tau}{T} = \frac{m(m-2)}{4q}$$

$$\bar{n} = \frac{2q}{2-m}$$

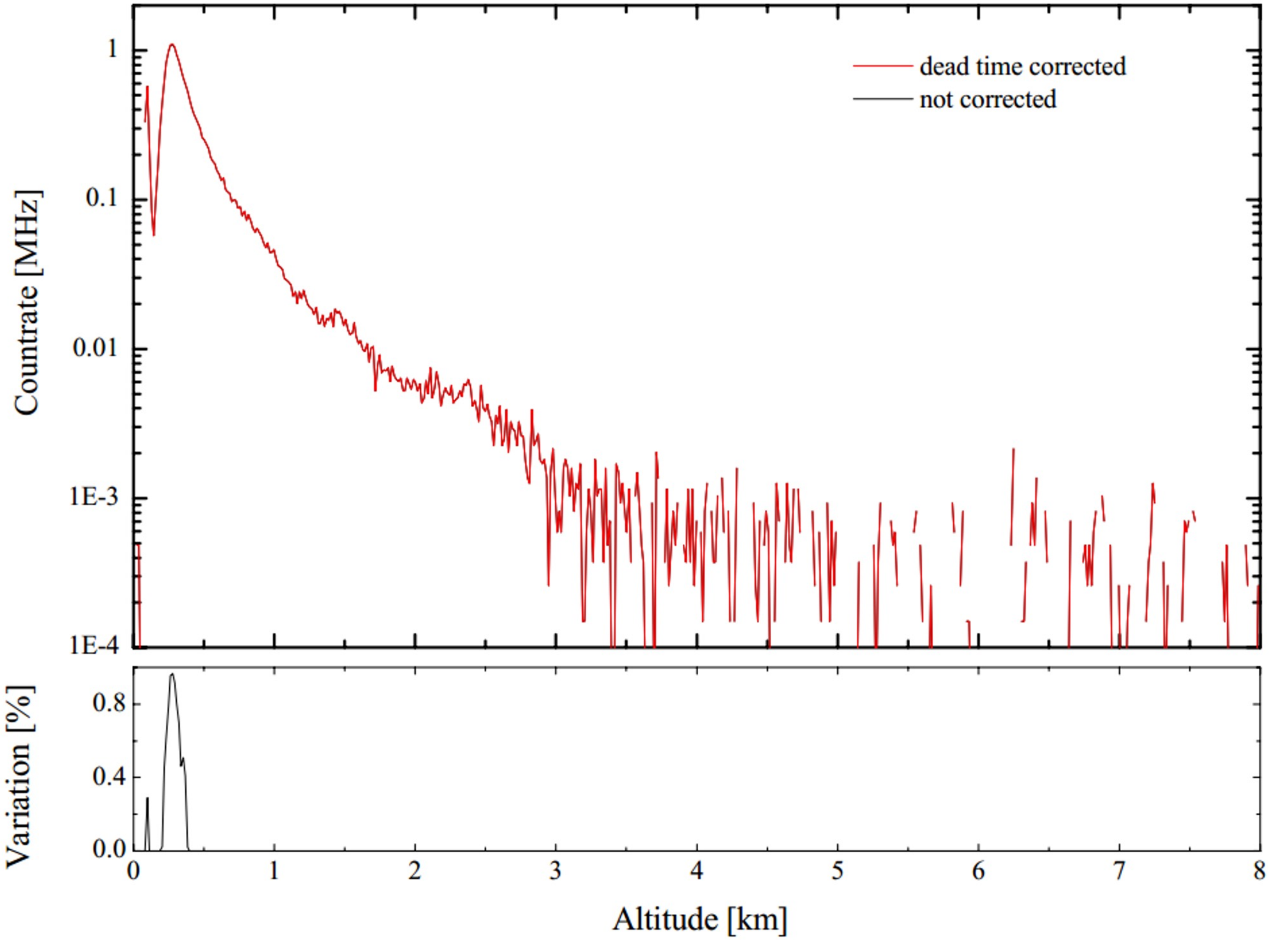
$$\tau = 8.56(7)\ ns$$

$$\bar{n} = 1.3202(7)$$

$$\bar{n}_{obs} = 1.306$$

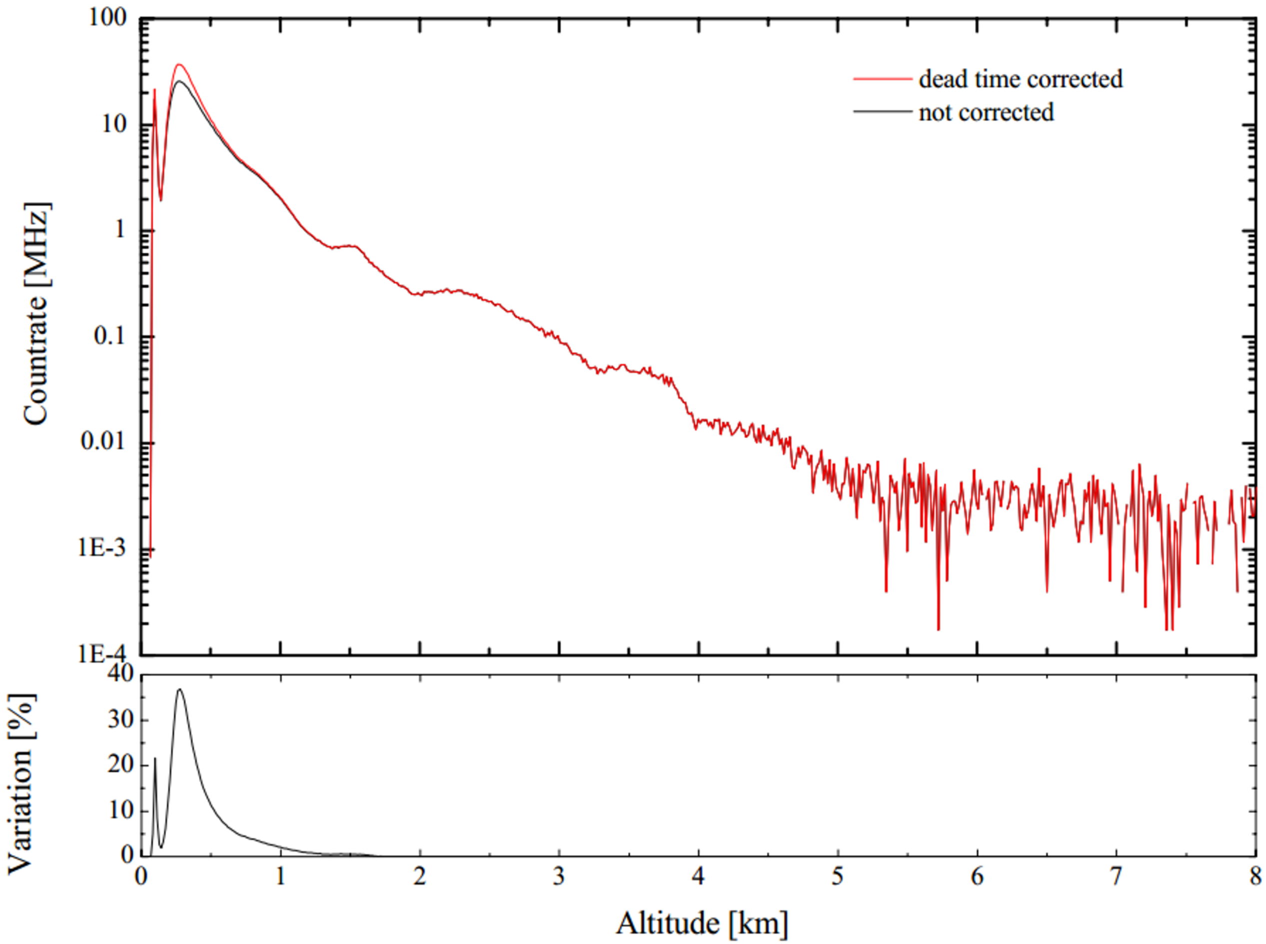
Dead-time correction: examples

Low count-rate



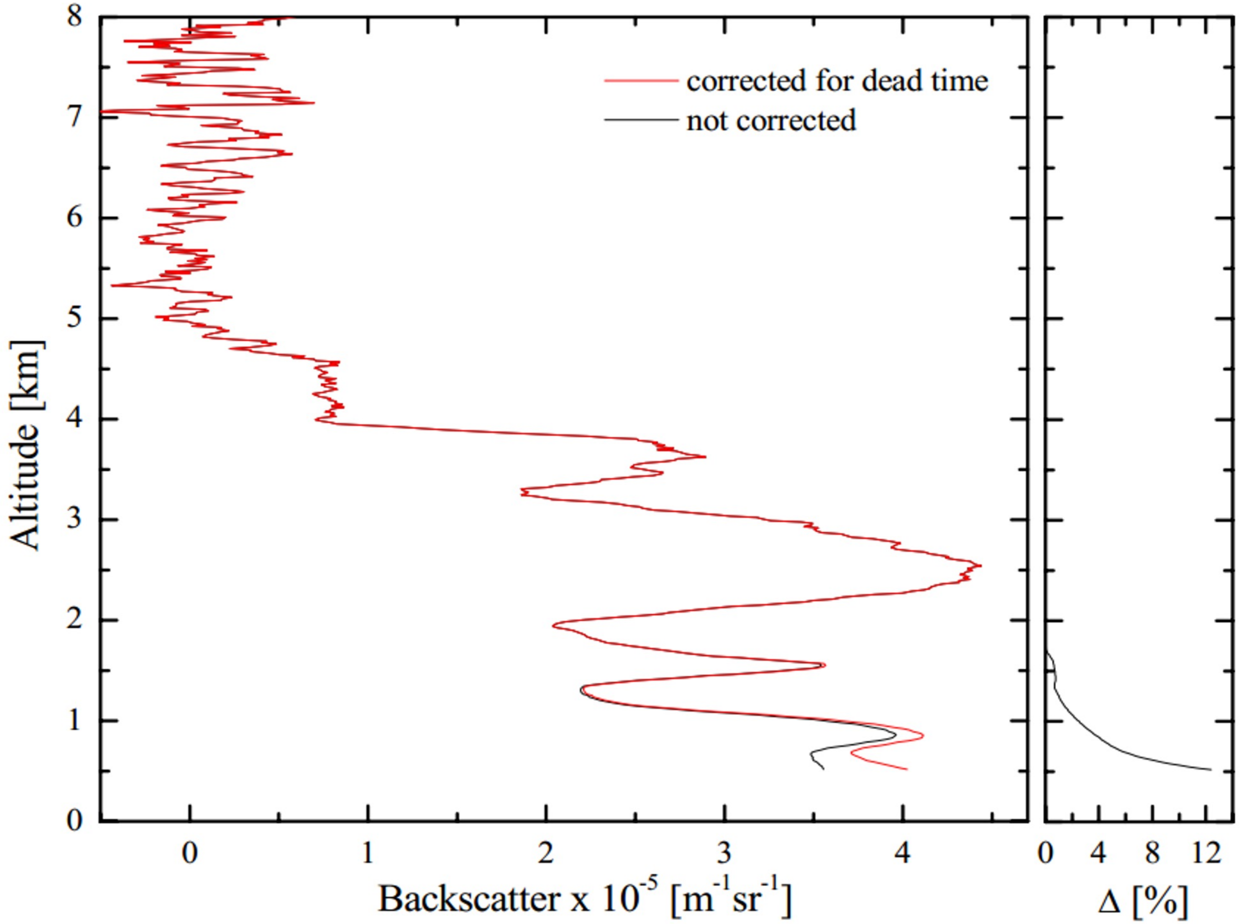
Dead-time correction: examples

High count-rate



Dead-time correction: examples

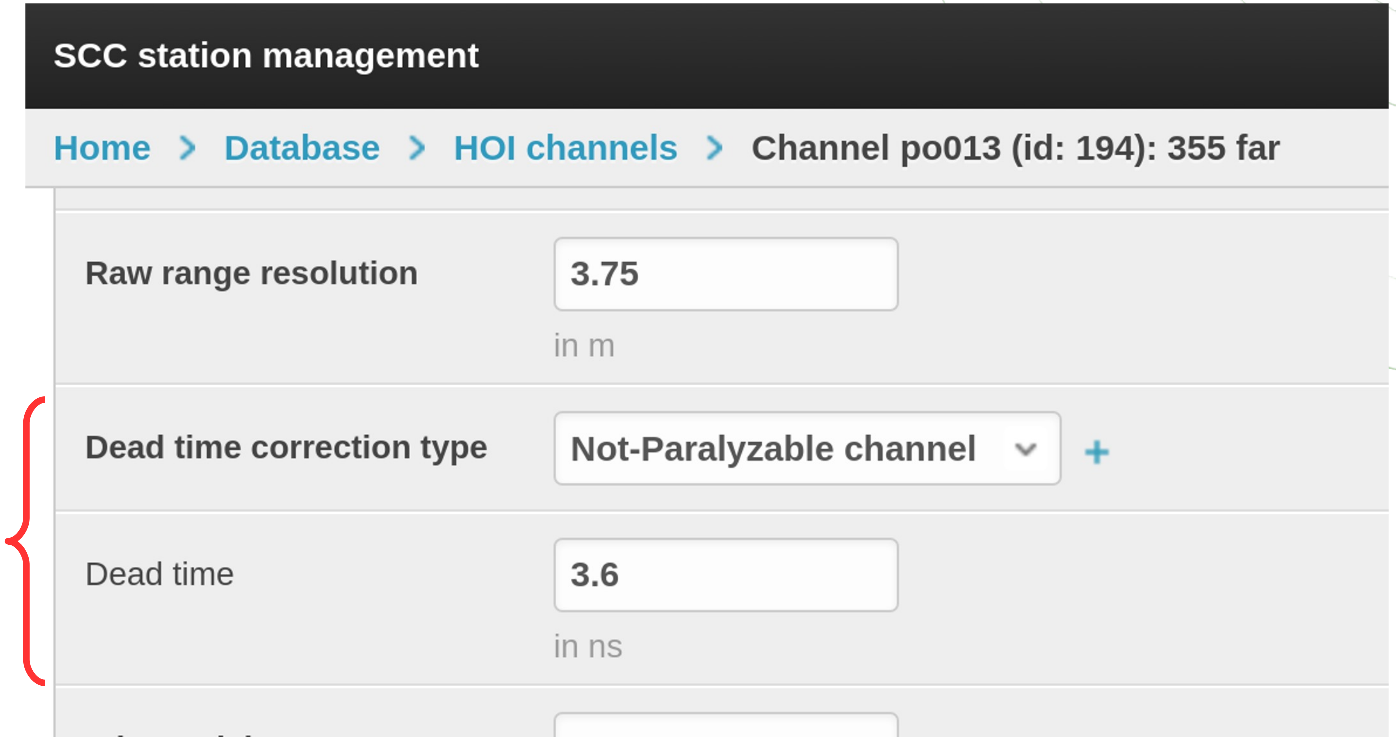
Effect on optical products



Dead-time correction

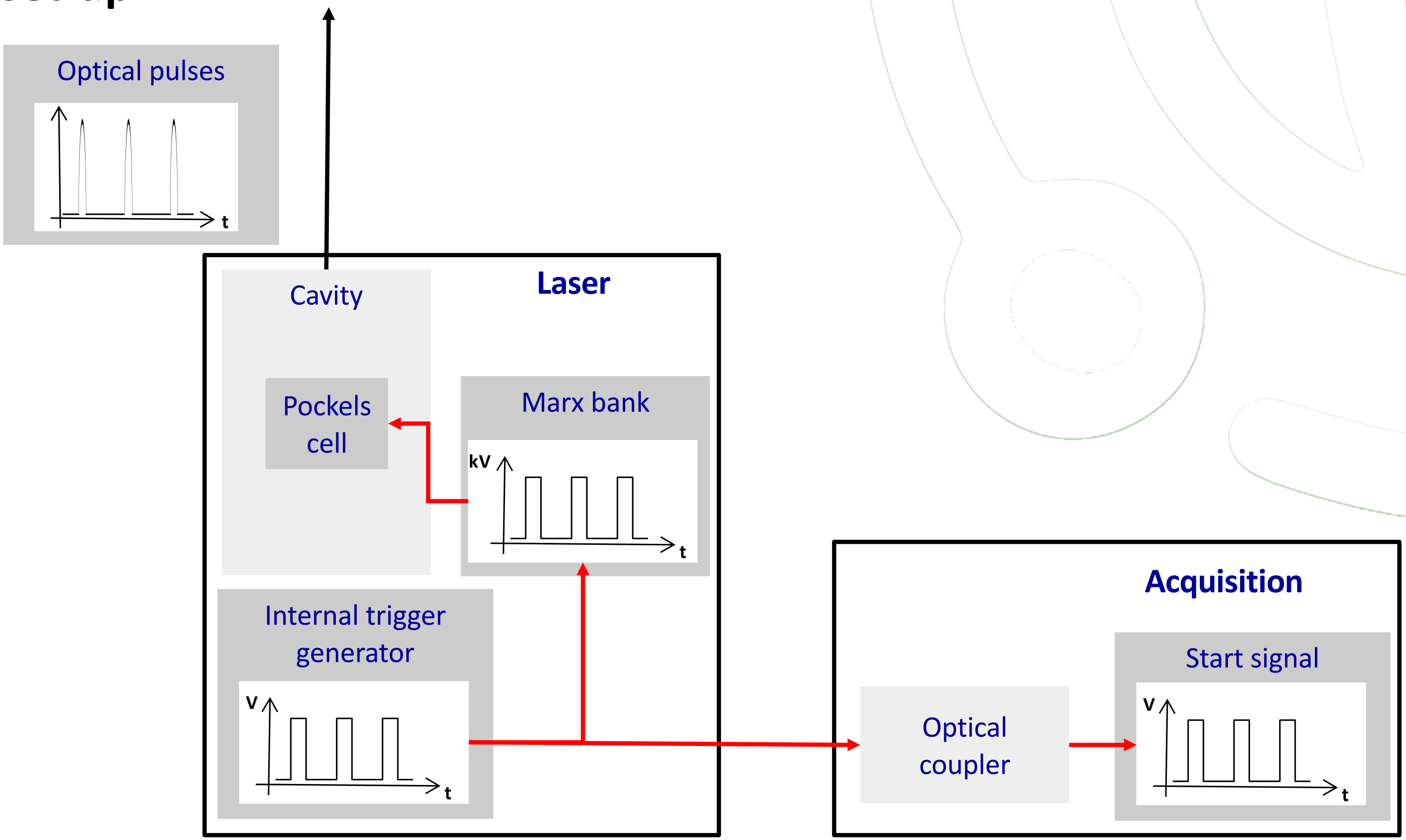
How to correct for dead-time using the SCC?

Setting the fields ***Dead time*** and ***Dead time correction type*** for all the channels for which the correction is required



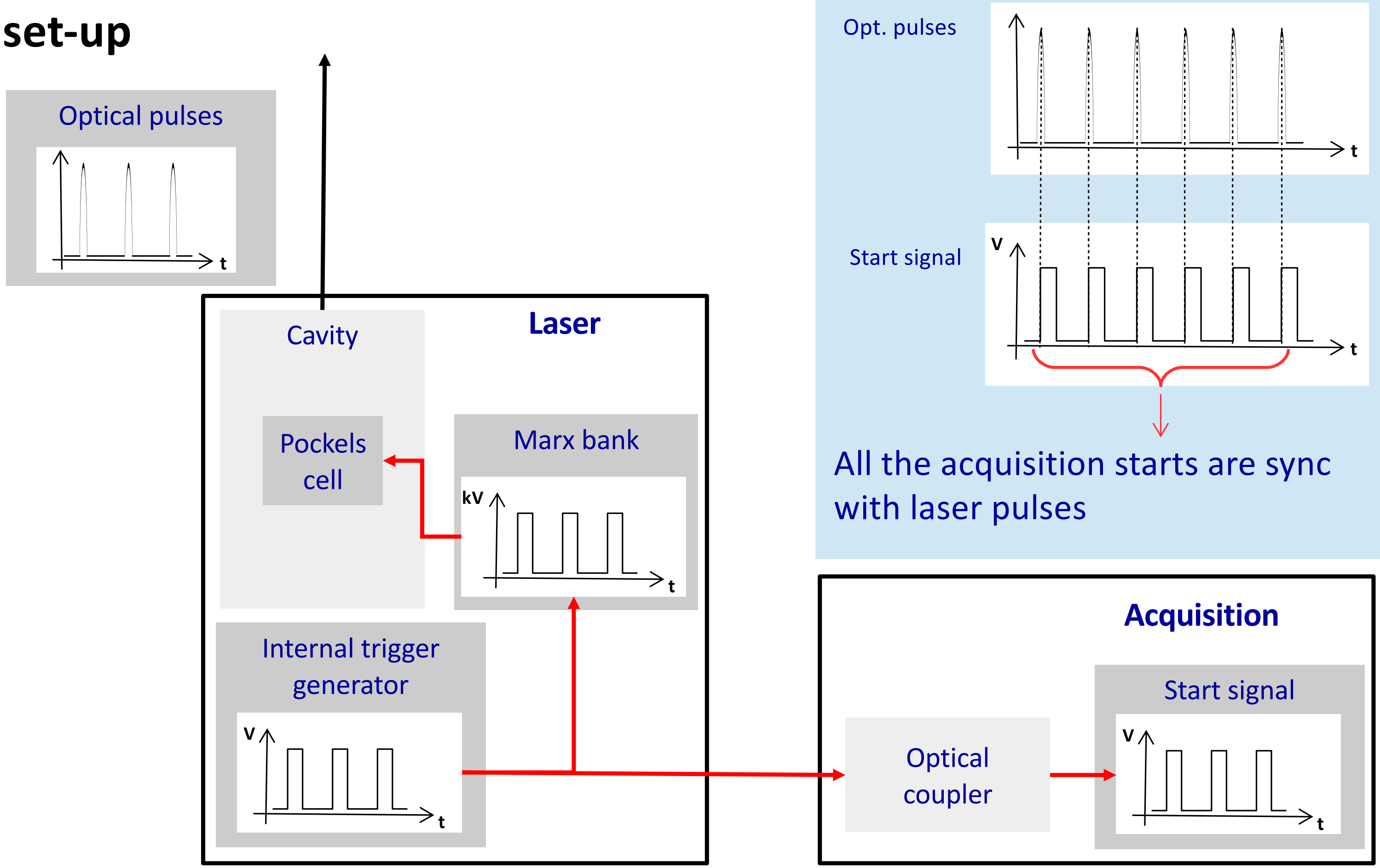
Trigger-delay correction

Typical lidar set-up



Trigger-delay correction

Typical lidar set-up



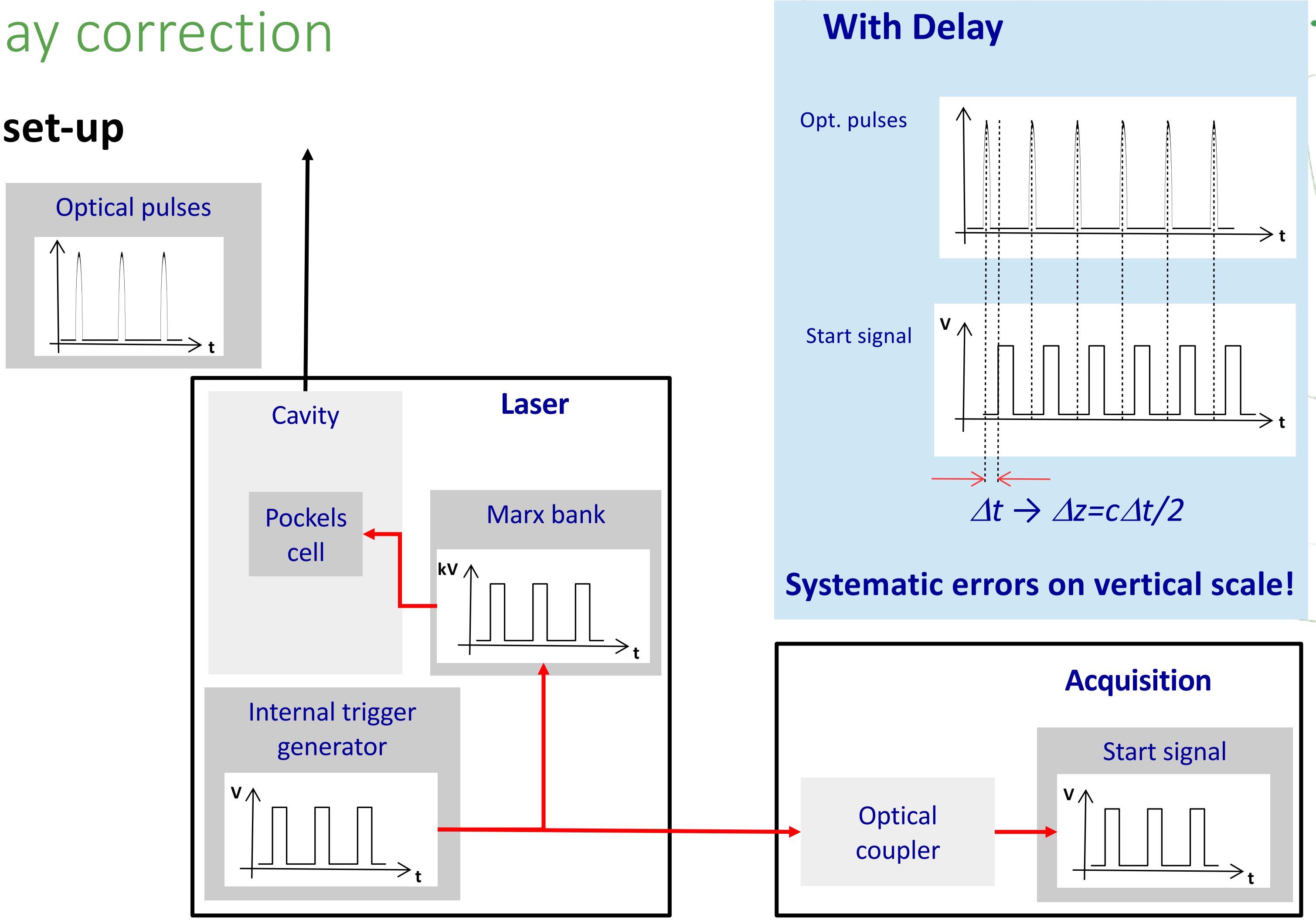
Without Delay
Opt. pulses
Start signal

All the acquisition starts are sync with laser pulses

Trigger-delay correction

TINERIS

Typical lidar set-up



Trigger-delay correction

How to correct?

$T_1=(t_1, t_2, ..., t_n)$ → acquisition system time scale sampling

Δt → trigger-delay of a particular lidar channel

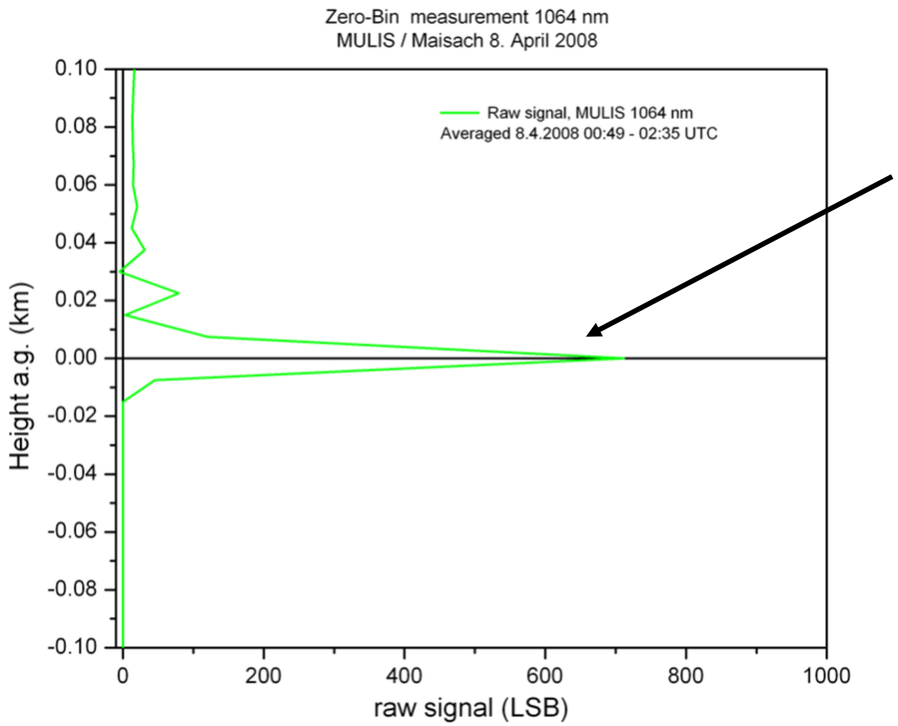
T_1 cannot be used as time scale for the lidar signals. The correct (delayed) time scale is:

$$T_2=(t_1+\Delta t, t_2+\Delta t, ..., t_n+\Delta t)$$

- As different lidar channels may have different trigger-delays, T_2 may change from channel to channel.
- Usually, it is needed to have all the lidar channels calculated on the same time scale. To do that an interpolation from the time scale T_2 to T_1 (which is channel independent) is needed.

Trigger-delay measurement

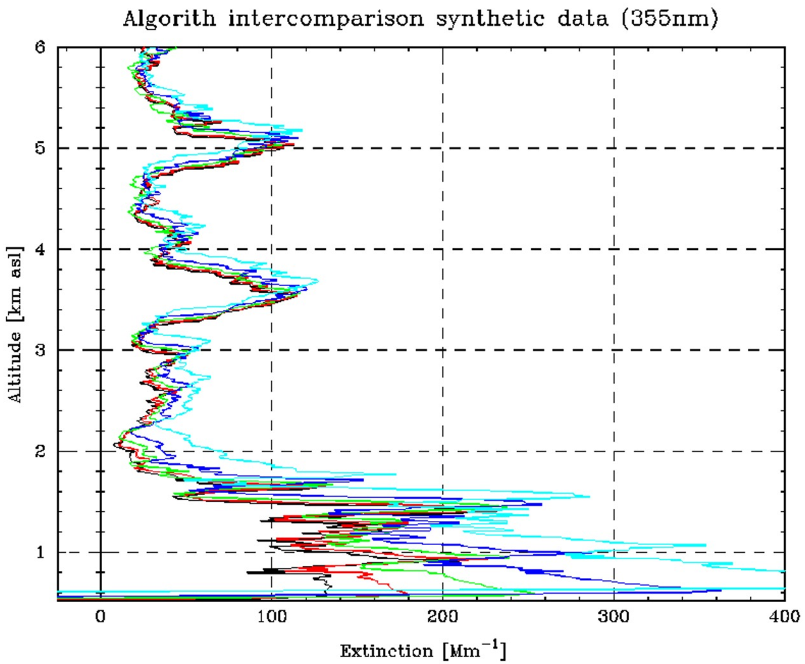
Measure the rangebin at which the peak corresponding to a properly attenuated reflection (or diffusion) of the laser beam by a near-range target (placed at well known distance) is located.

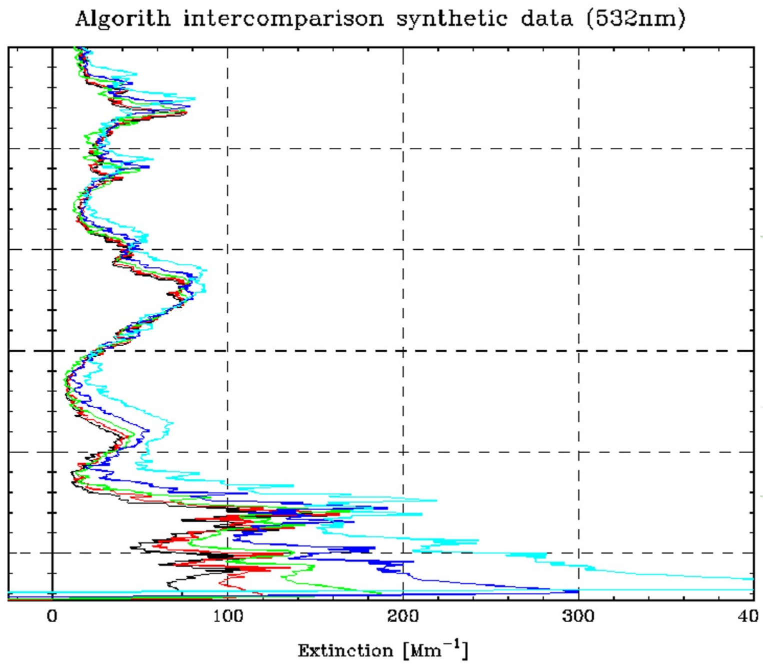


peak from diffuse reflections of the outgoing laser pulse from the roof window of the lidar lab

Trigger-delay correction

Effect on optical products: Extinction



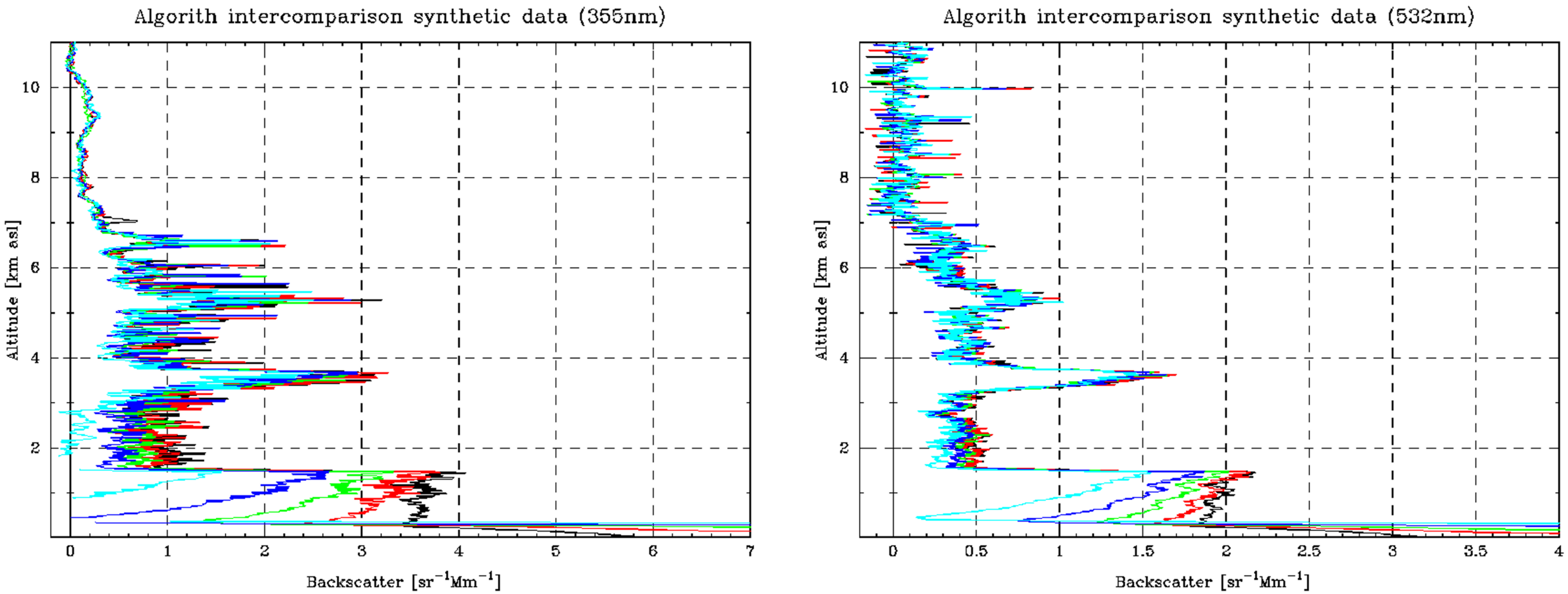


Trigger delay on Raman channel (raw resolution: 100ns)

0:black **100 ns:red** **250 ns:green** **500 ns:blue** **1000 ns: cyan**

Trigger-delay correction

Effect on optical products: Raman Backscatter



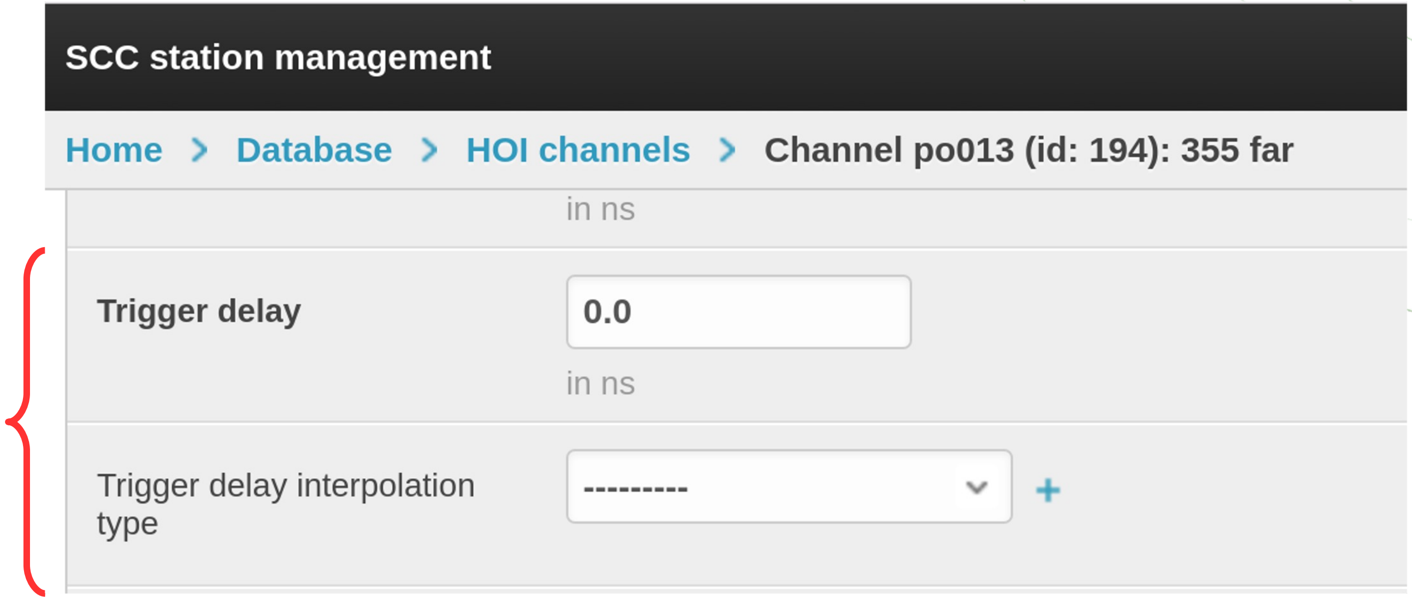
Trigger delay on Raman channel (raw resolution: 100ns)

0:black **100 ns:red** **250 ns:green** **500 ns:blue** **1000 ns: cyan**

Trigger-delay correction

How to correct for trigger-delay using the SCC?

Setting the fields ***Trigger delay*** and (optionally) ***Trigger delay interpolation type*** for all the channels for which the correction is required



Background subtraction

A raw lidar signal can be expressed as

$$S(z,\lambda) = S_{par}(z,\lambda) + S_{mol}(z,\lambda) + S_{atm}(\lambda) + S_{el}$$

- S_{par} and S_{mol} the signal contributions backscattered by particles and molecules
- S_{atm} is the optical signal background from the atmosphere, i.e. the sky brightness, which is independent of range
- S_{el} is the electronic signal background (electronic effects of the signal detection and data acquisition). In general, it can be temporally constant or variable

It is fundamental to remove both S_{atm} and S_{el} from the measured lidar profiles before applying any optical retrieval algorithm.

Background subtraction

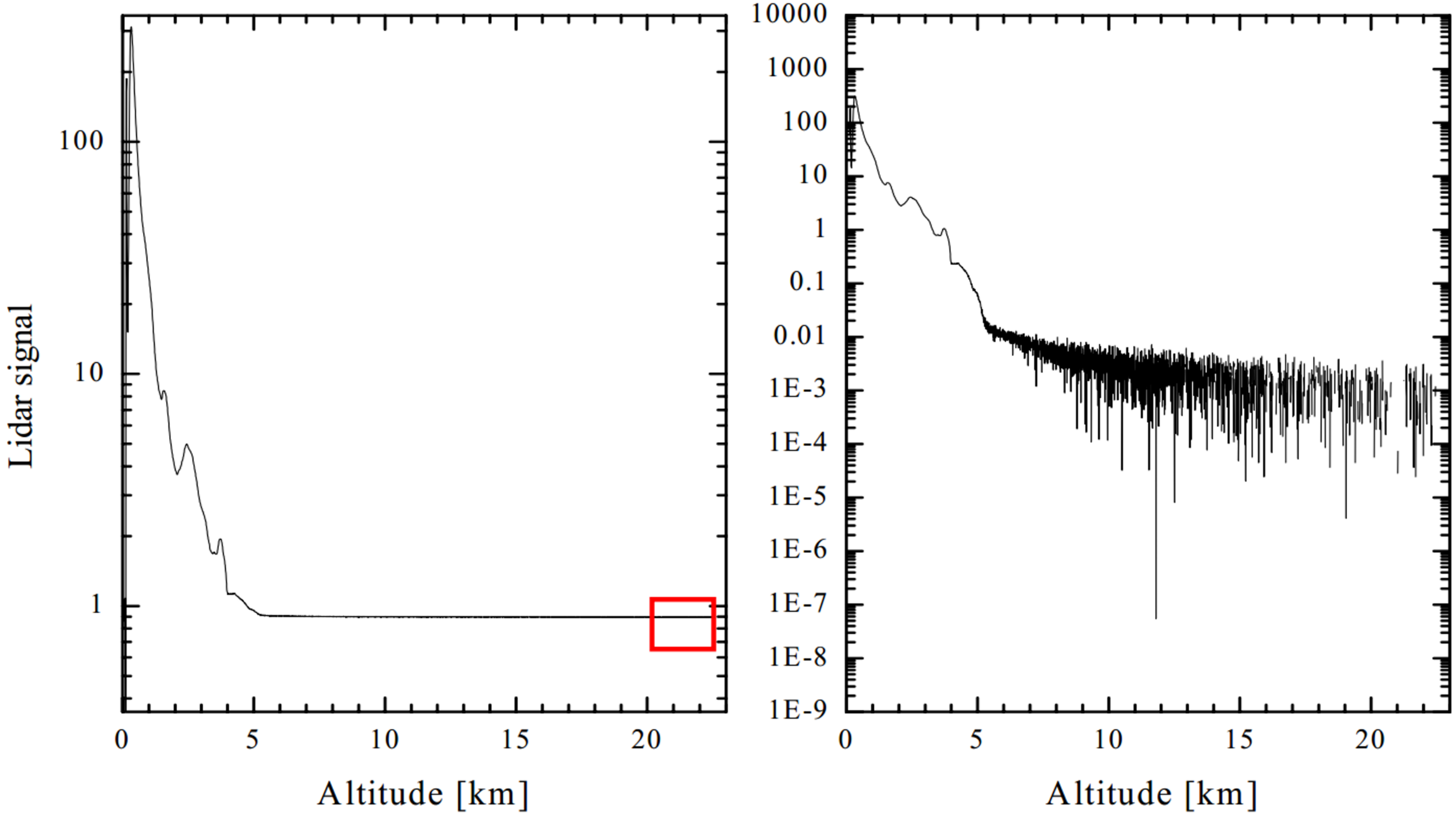
A raw lidar signal can be expressed as

$$S(z,\lambda) = S_{par}(z,\lambda) + S_{mol}(z,\lambda) + S_{atm}(\lambda) + S_{el}$$

- The **constant** background components $S_{atm} + S_{el}$ can be determined:
  - in the far range of the lidar signal where the expected contribution from atmospheric backscatter from both molecules and particles is negligible
  - in the pre-trigger range before the laser pulse, where the signal must be free of electronic distortions, which could influence the determination of the constant background
- In both cases the constant background value is calculated as mean value over signal ranges, which are large enough so that the residual standard error of the mean is negligible.

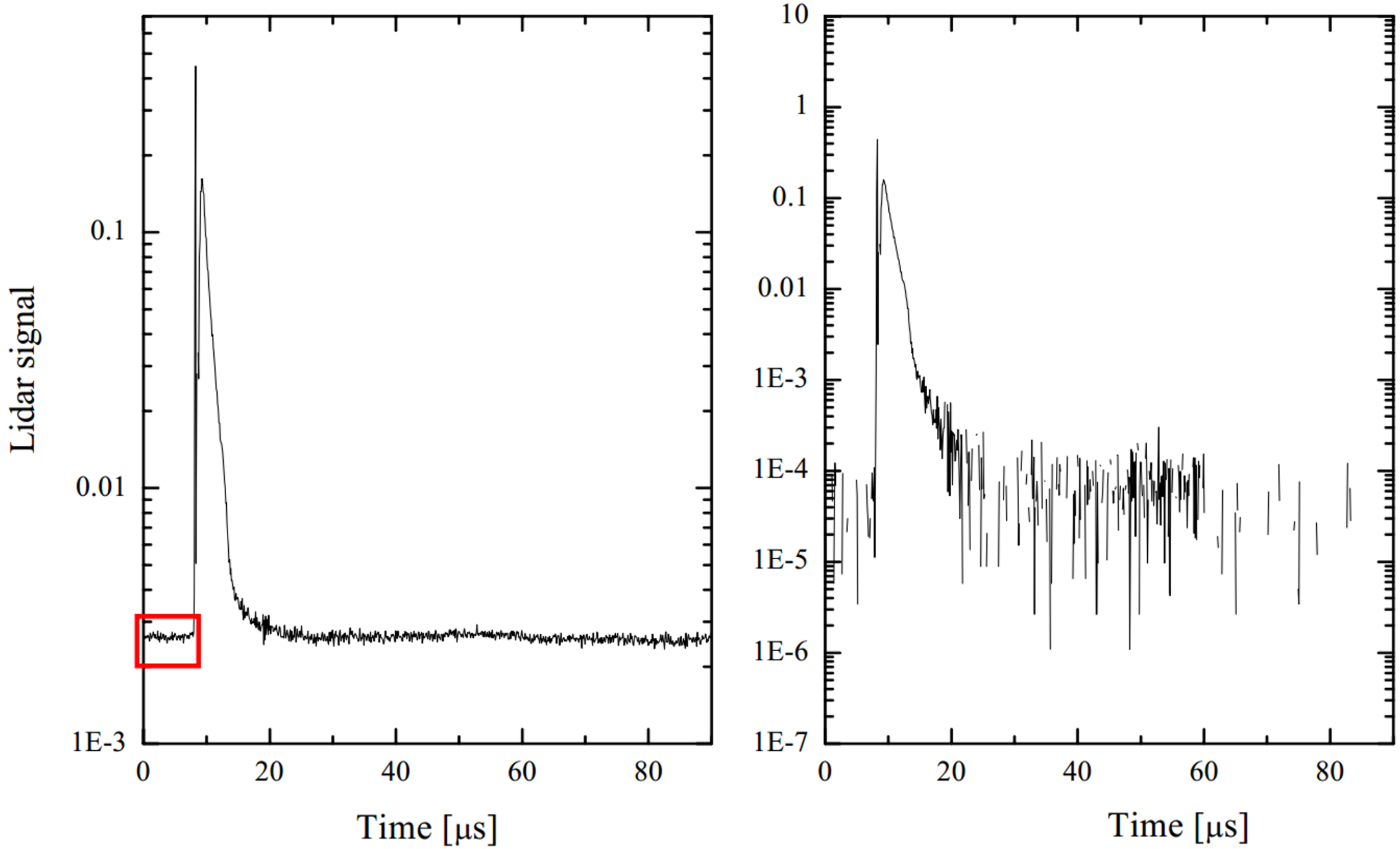
Background subtraction

Using far-range region



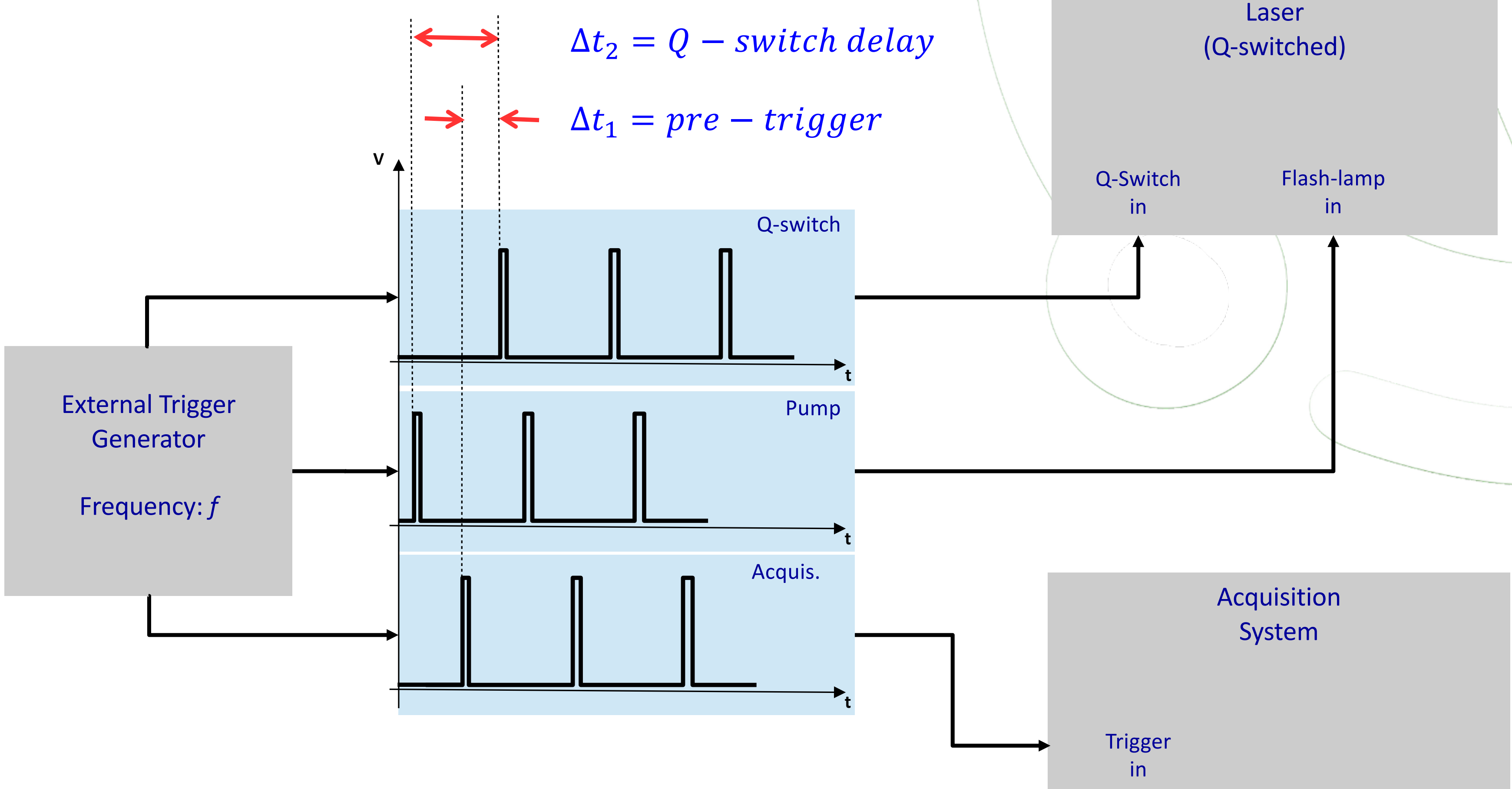
Background subtraction

Using pre-trigger region



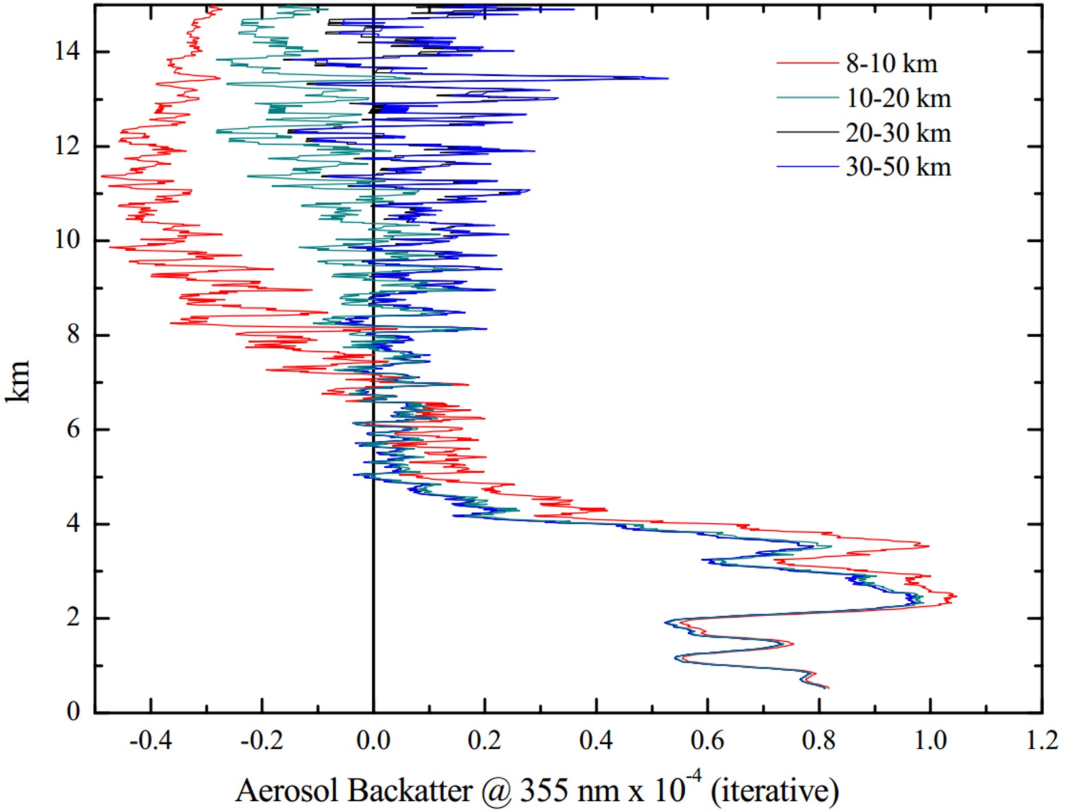
Background subtraction

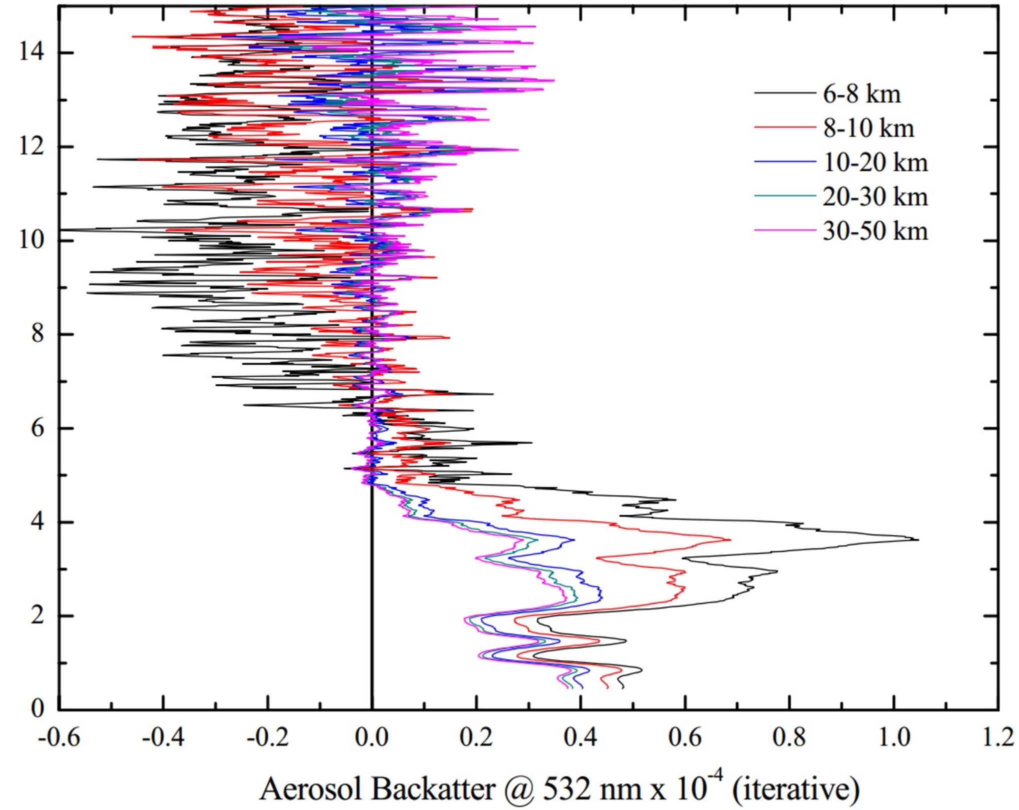
How to set-up a pre-trigger region?



Background subtraction

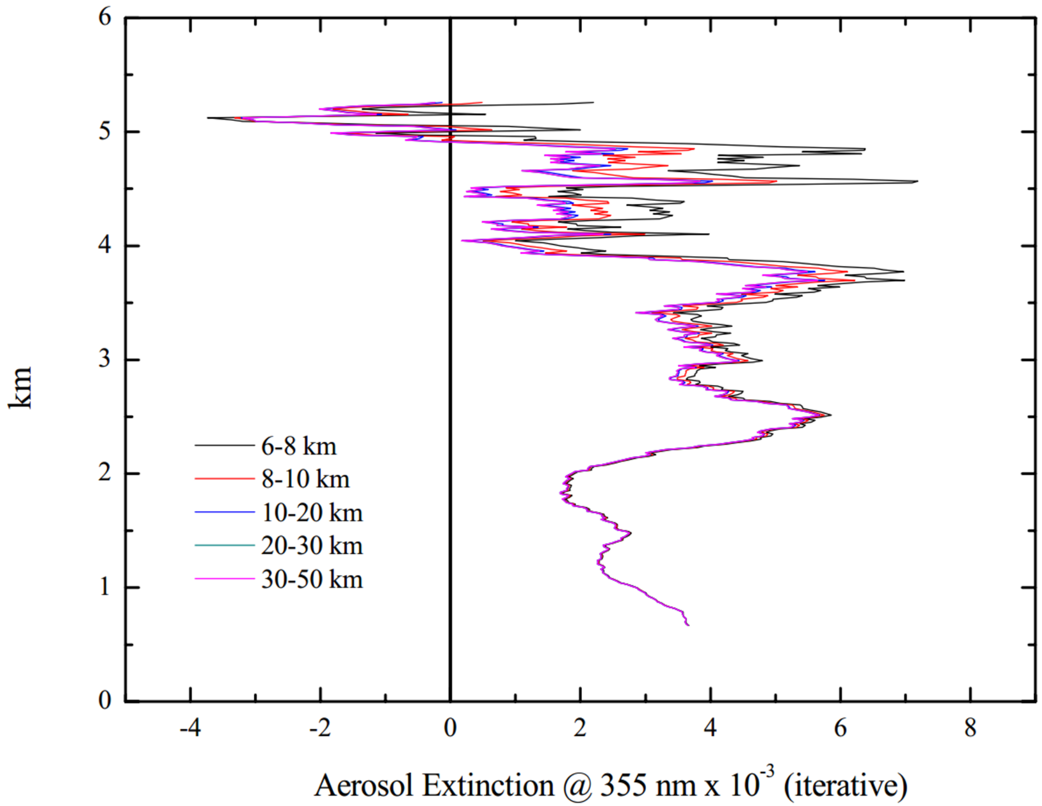
Effect on optical products: backscatter

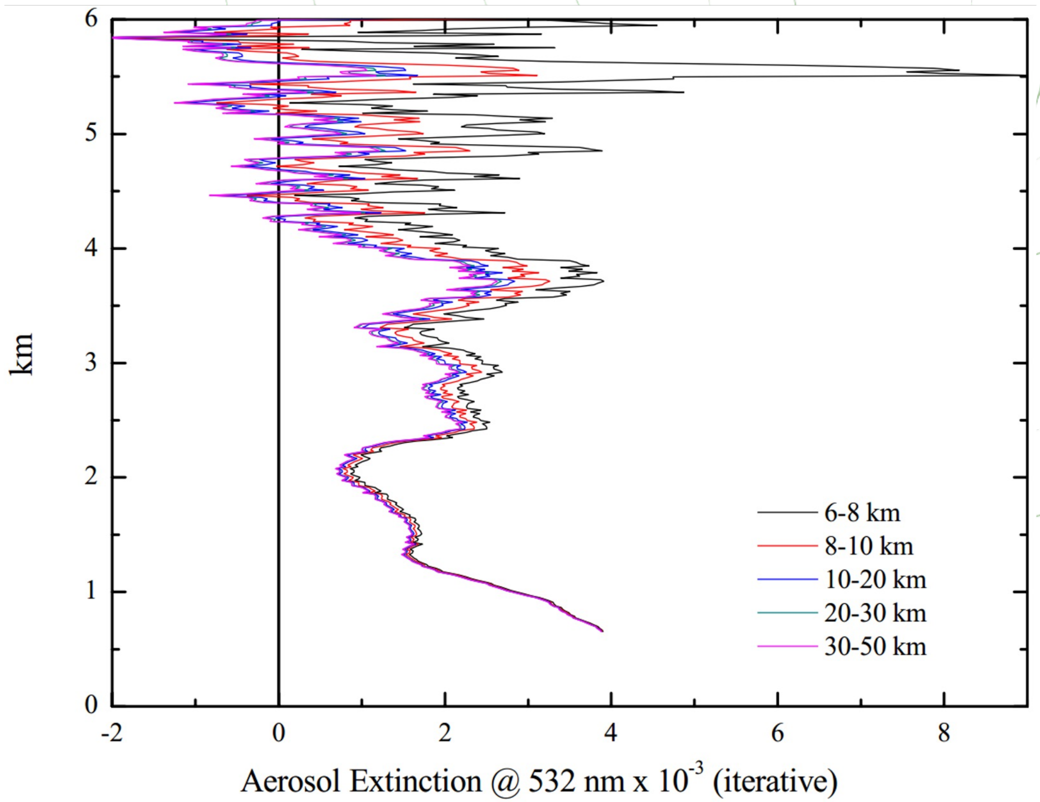




Background subtraction

Effect on optical products: extinction





Background subtraction

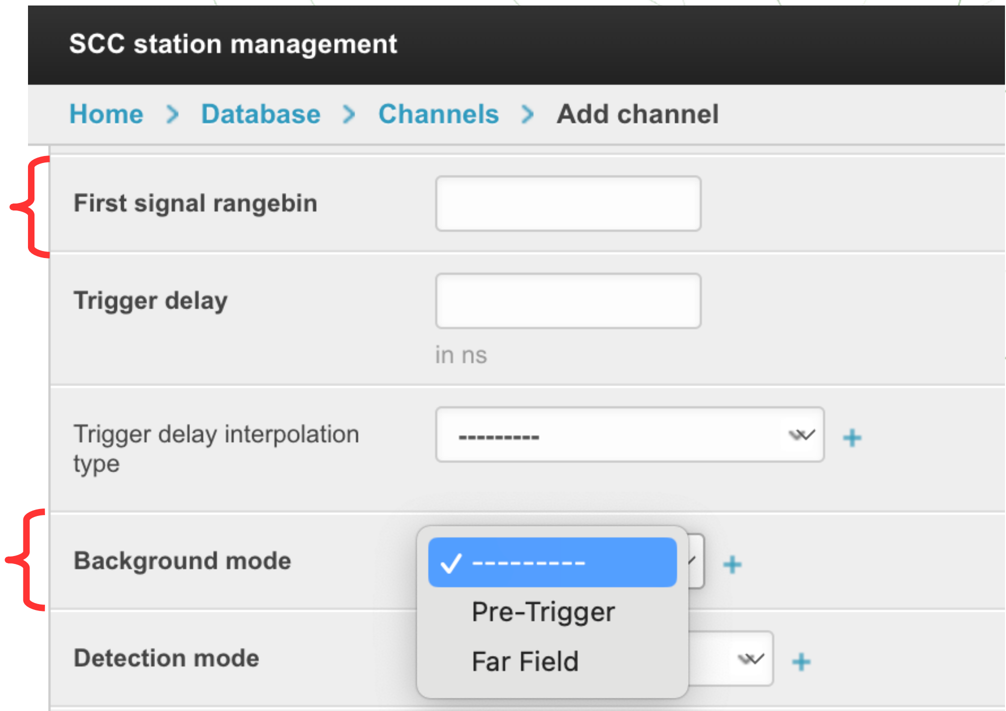
How to subtract constant background using the SCC?

1. Setting the fields ***Background mode*** and/or ***First signal rangebin*** for all the channels
2. Setting the variables (in the input NetCDF datafile)

   Background_Low

   Background_High

   First_Signal_Rangebin (optional)



Background subtraction

How to subtract constant background using the SCC?

Background_Low(channels)		
Minimum altitudes for atmospherical background calculation in meters for each channel. If pre-trigger is used as background subtraction mode for a particular channel, the corresponding value of this variable has to be set to the rangebin to be used as lower limit (within pre-trigger region) for background calculation.	`double`	Mandatory
Background_High(channels)		
Maximum altitude for atmospherical background calculation in meters for each channel. If pre-trigger is used as background subtraction mode for a particular channel, the corresponding value of this variable has to be set to the rangebin to be used as upper limit (within pre-trigger region) for background calculation. If the variable First_Signal_Rangebin is not given the first valid lidar rangebin will be the next after Background_High one.	`double`	Mandatory

Background subtraction

Background subtraction related variables in the NetCDF input file

double Background_Low(channels) ; **Mandatory**

double Background_High(channels) ; **Mandatory**

Minimum and maximum of the range to calculate atmospheric background for each channel.

Far-range subtraction mode

Minimum and maximum should be given as altitudes in meters a.g.l.

Pre-trigger subtraction mode

Minimum and maximum rangebins within the pre-trigger region

Background subtraction

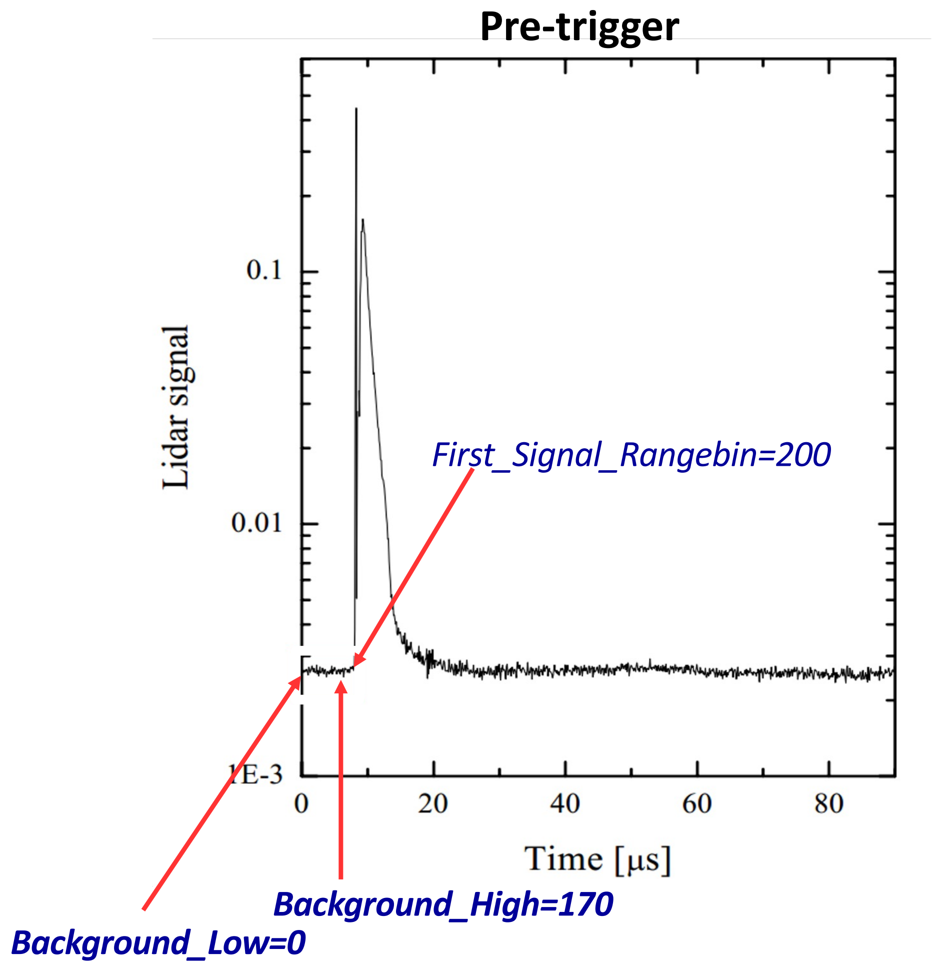
Background subtraction related variables in the NetCDF input file

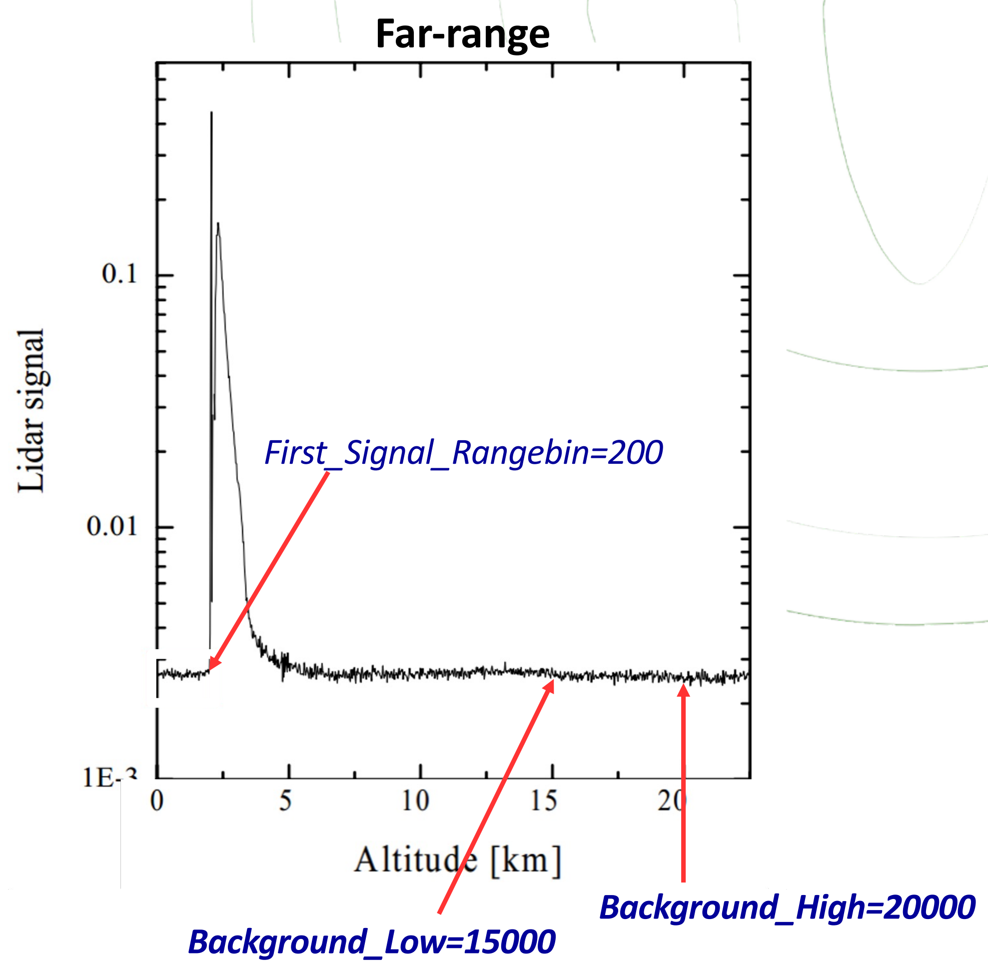
int First_Signal_Rangebin(channels); **Optional**

Rangebin at which lidar profile begins starting from 0.

If not given the first valid rangebin will be taken from the lidar configuration settings.

Background subtraction





Background subtraction

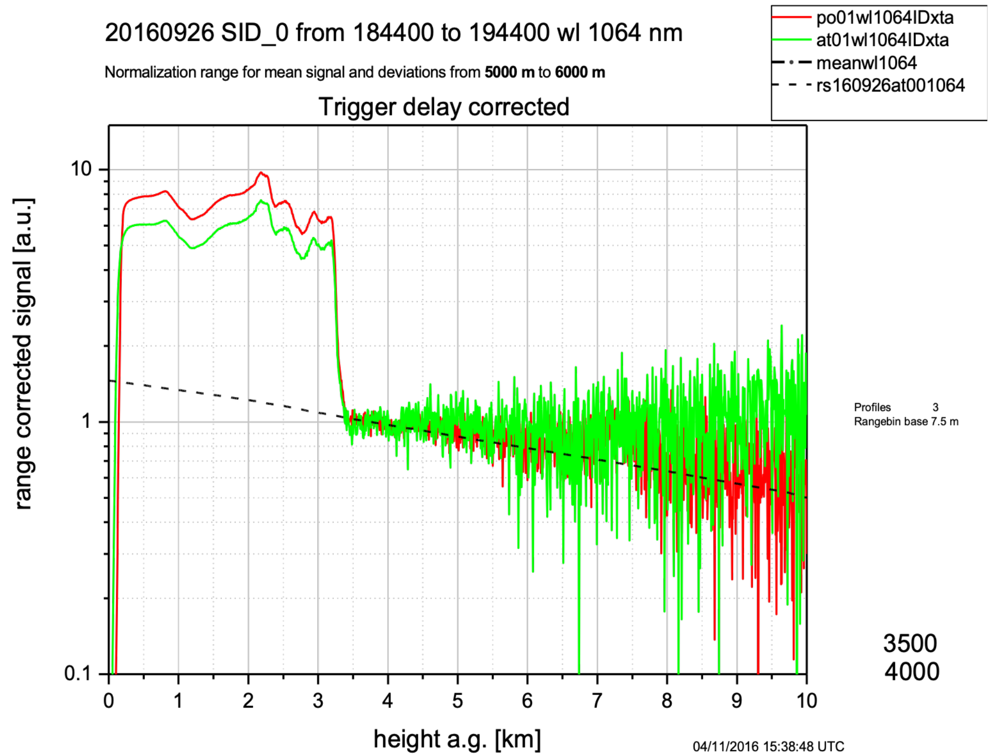
What if S_{el} is not constant?

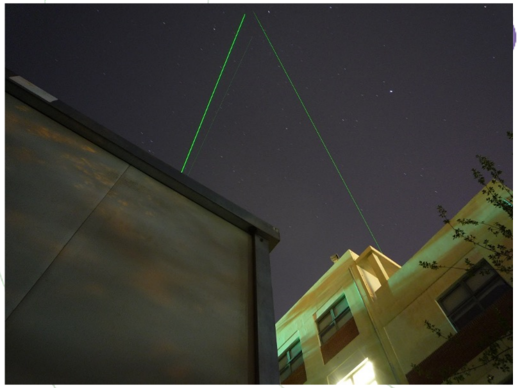
- typically, due to electronic distortions (mainly on analog lidar signals)
- we may have temporally random components and/or components synchronal with the repetition of the laser pulse.
- random components zero out in the average of many subsequent lidar signals
- the synchronal components do not! → significant distortions on lidar signals
- stationary synchronal components can be determined from dark signals, which are measured, for example, with the telescope fully obscured.
- dark signals have to be averaged over a long enough time to decrease the random contributions sufficiently.
- subtracting dark signals to the lidar signals the not constant contribution of S_{el} can be minimized (assuming stationary synchronal components which could be **NOT** always the case!).

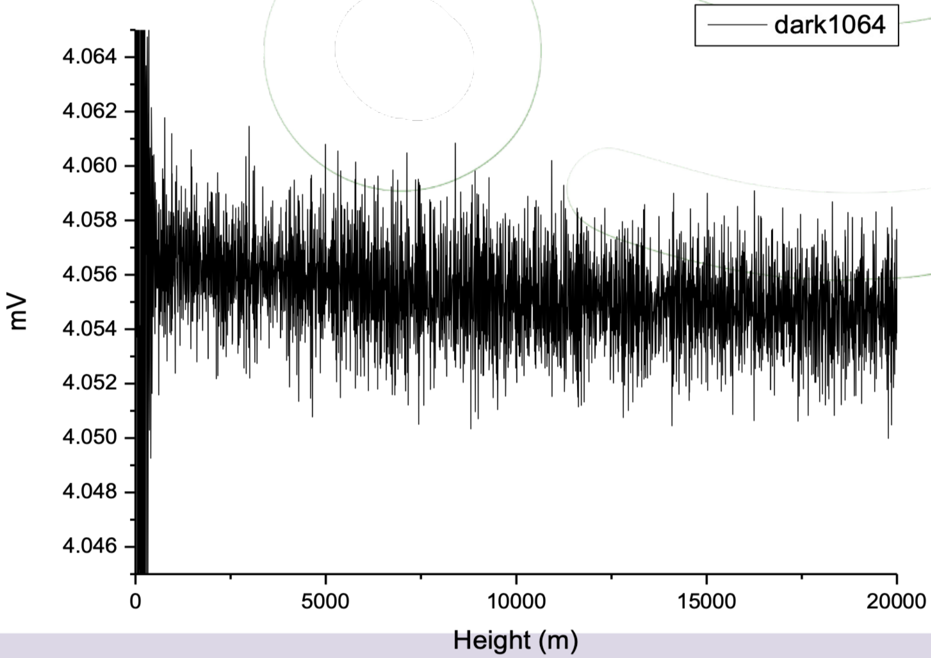
Background subtraction

Dark signal subtraction improvement

An Example from ATHL16



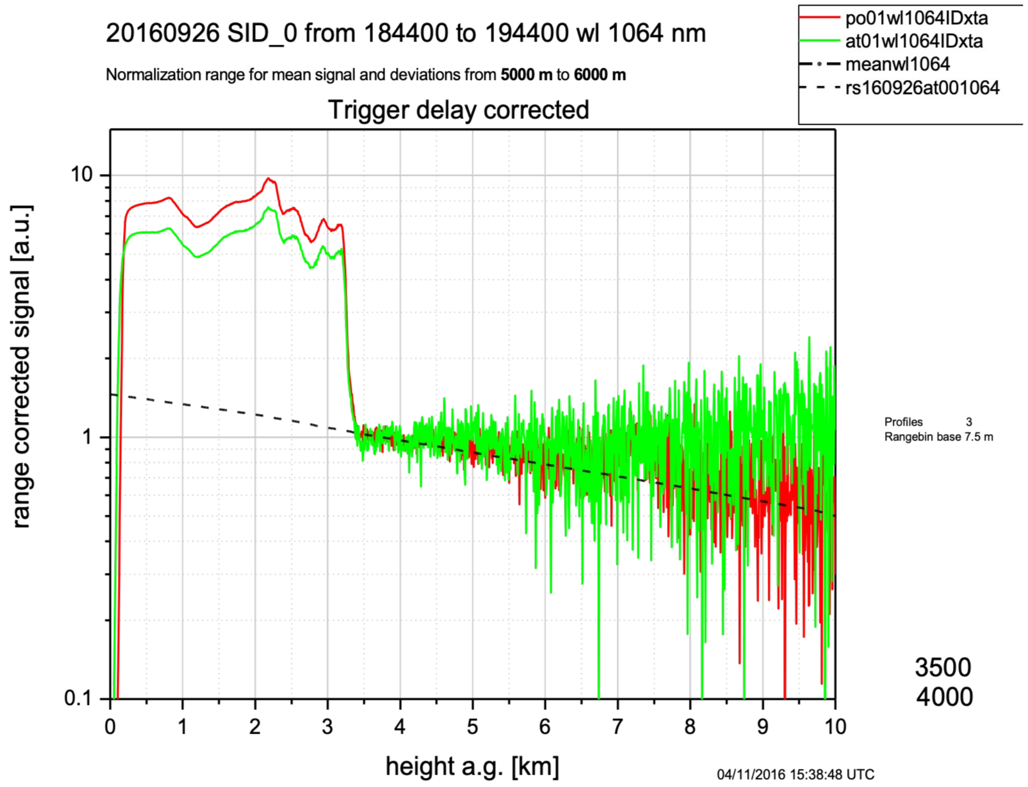


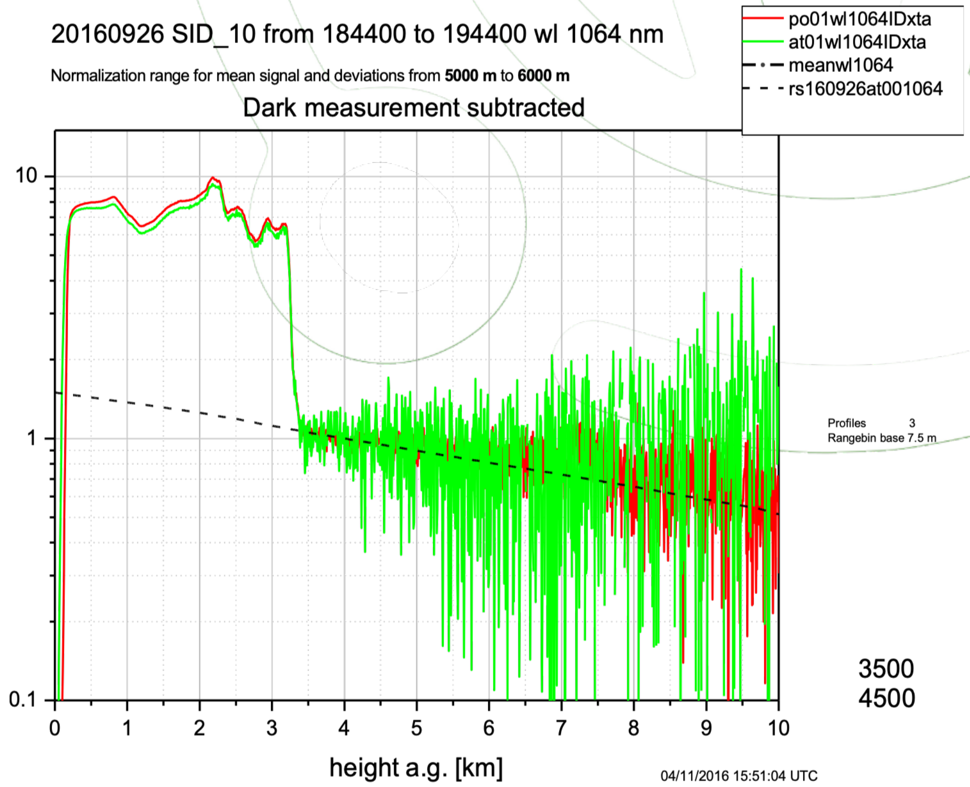


Background subtraction

Dark signal subtraction improvement

An Example from ATHL16





Background subtraction

Dark signal related variables in SCC NetCDF input file

double Raw_Bck_Start_Time(time_bck,nb_of_time_scales);

double Raw_Bck_Stop_Time(time_bck,nb_of_time_scales);

Start and stop time of each dark signal in second since *RawBck_Start_Time_UT* (global attribute)

double Background_Profile(time_bck,channels, points);

Dark signal profiles. These profiles will be averaged, and they subtracted to lidar signals

Signals glueing

Lidar signals can cover a quite large dynamic range

- backscattered signal from the near range is usually several orders of magnitudes higher than the one backscattered from the rather clean troposphere
- in general, it is demanding to cover this large dynamic range with one data acquisition channel with linear response, several approaches are used to overcome this problem.

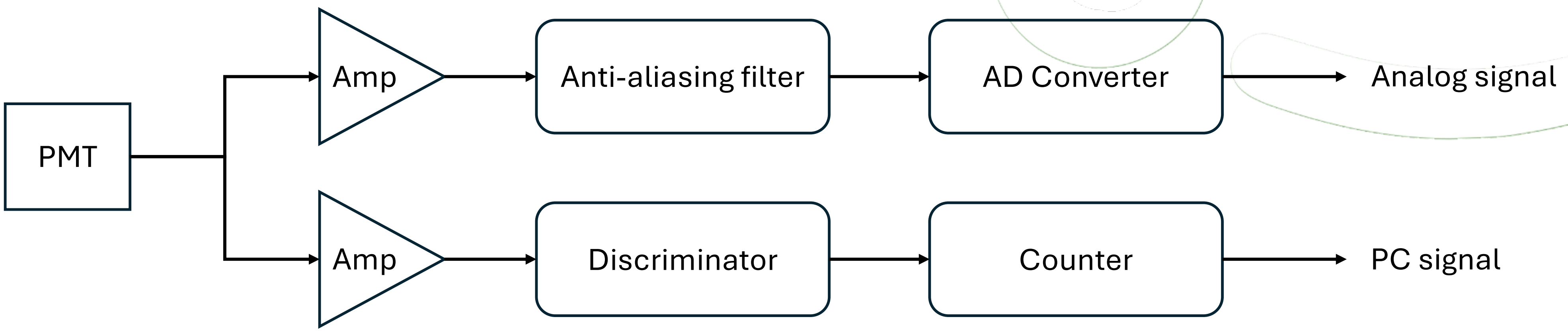
How to improve dynamic range as much as possible?

Signals glueing

Improve detection dynamic range

Option 1:

split the signal output from a single photo-multiplier in two signals and record one signal (low frequency component) using analog detection mode and the other (high frequency component) with photon-counting technique

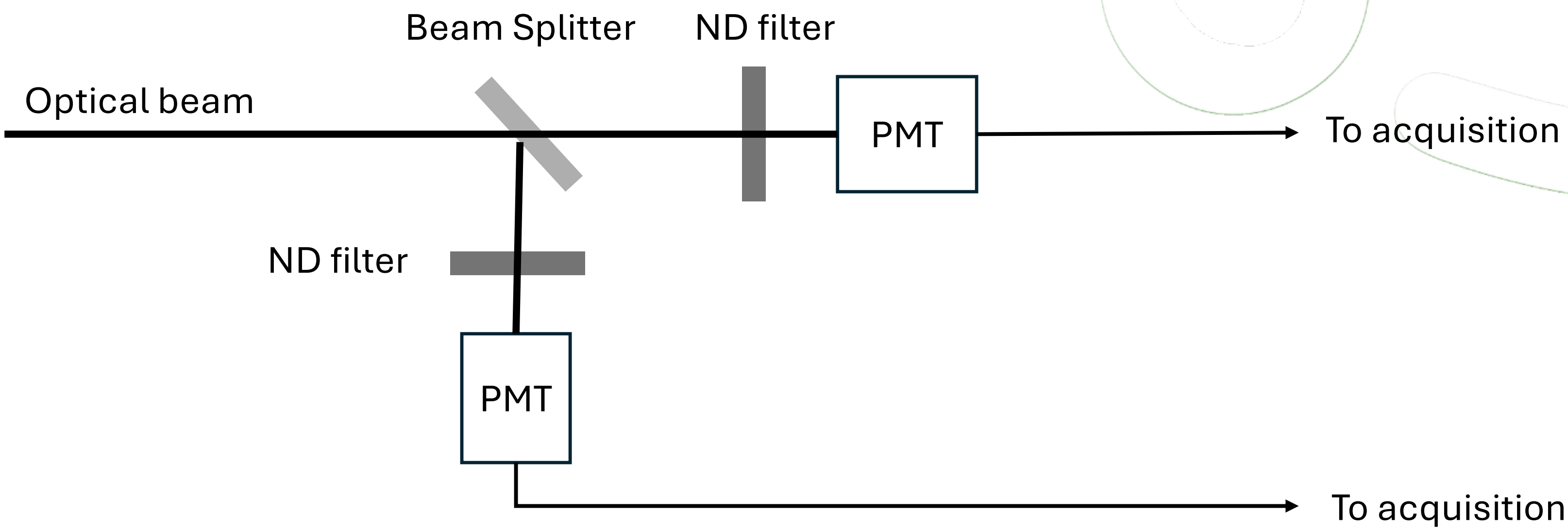


Signals glueing

Improve detection dynamic range

Option 2:

split the lidar signal optically using a beam splitter and detect the split components with two detectors and subsequent data acquisitions. Both signals are attenuated, if necessary, with neutral density filters.



Signals glueing

Improve detection dynamic range

Option 3:

to use two (or more) telescopes with separate detection electronics, i.e. one small telescope designed to detect the near-range signal, and another larger telescope optimized to measure the weak far-range signal.

Signals glueing

Improve detection dynamic range

Whatever will the option, the complementary signals need to be merged to get a single "extended" lidar signal for the analysis.

What to merge?

Signals glueing

Signals merging

Pros

often the near-range signals show a quite low SNR (or distortions) in the far-range making quite challenging (and sometime not possible at all!) the calculation of reliable optical products out of near-range signals only. Acquisition based on Licel TR are good example of this category

Cons

signals are not calibrated so the gain ratio of two signals need to be determined before to merge

Signals glueing

Optical products merging

Pros

the merging is made on calibrated quantities, so in principle we just need to merge

Cons

not always possible. Near range product may not be retrieved with enough accuracy due to the low SNR of the near range signal

Signals glueing

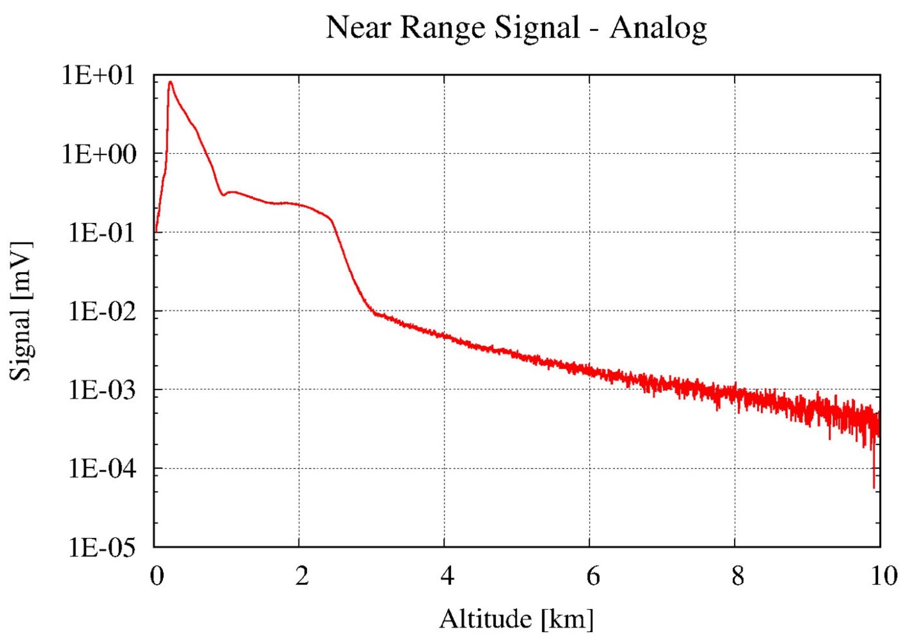
Merging steps

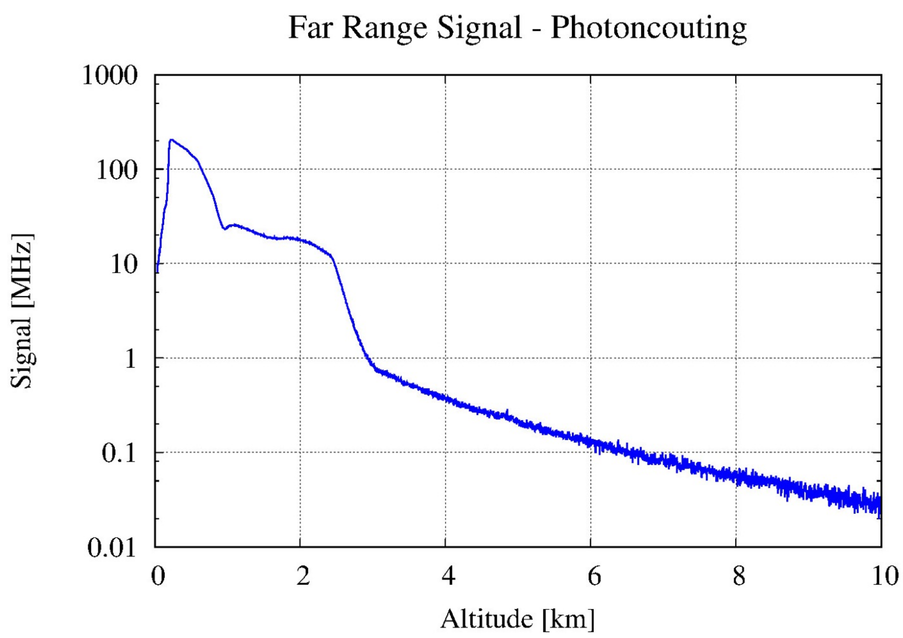
1. the near-range and the far-range signals need to be screened for low-level clouds, corrected for instrumental effects like dead-time, trigger-delay, etc., and the backgrounds have to be subtracted

2. normalization of the near-range signal on the far-range one within a proper glueing region

3. glue the normalized near-range signal and the far-rage one within the glueing region

Signals glueing





- Near range signal needs to be normalized on far range one
- Linear fit within a proper glueing region:

$$S_{pc}(z) = KS_{an}(z) \qquad z_{min} \leq z \leq z_{max}$$

How to find z_{min} and z_{max}?

Signals glueing

- z_{min}
  - upper threshold on the count-rate of far-range PC signal.
  - should be set assuring a reliable dead-time correction. Typical values used for that are 10-30 MHz (in general dependent from dead-time value)
- z_{max}
  - determined from the near-range signal, which can be an analog or a pc signal.
  - analog signals in general have a minimum level below which signal distortions and/or the signal noise become significant.

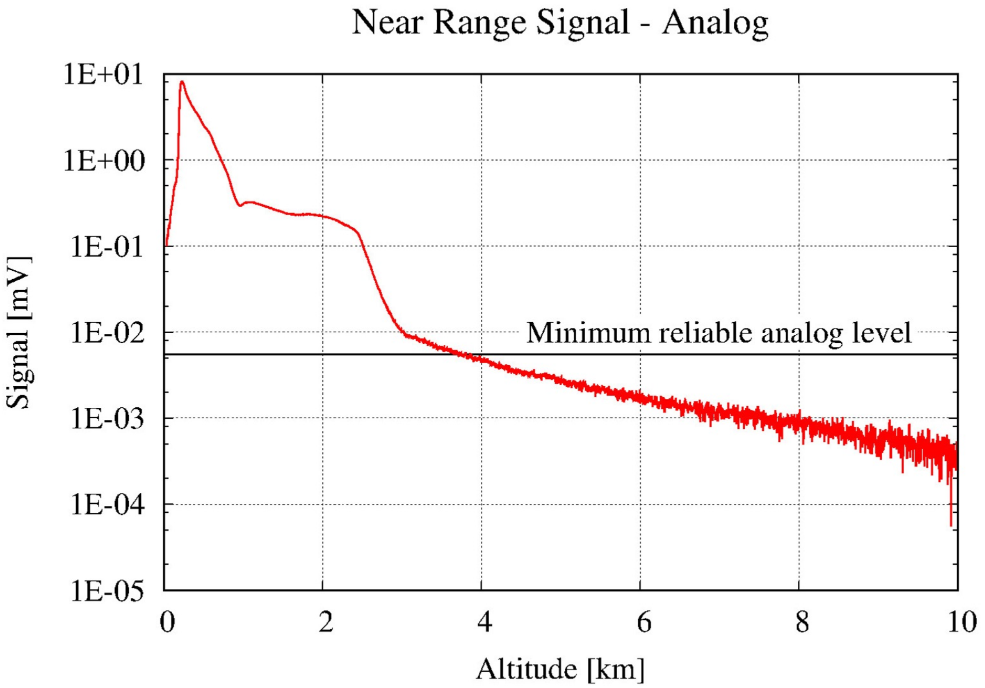
$$S_{min} = \frac{S_{max}}{F}$$

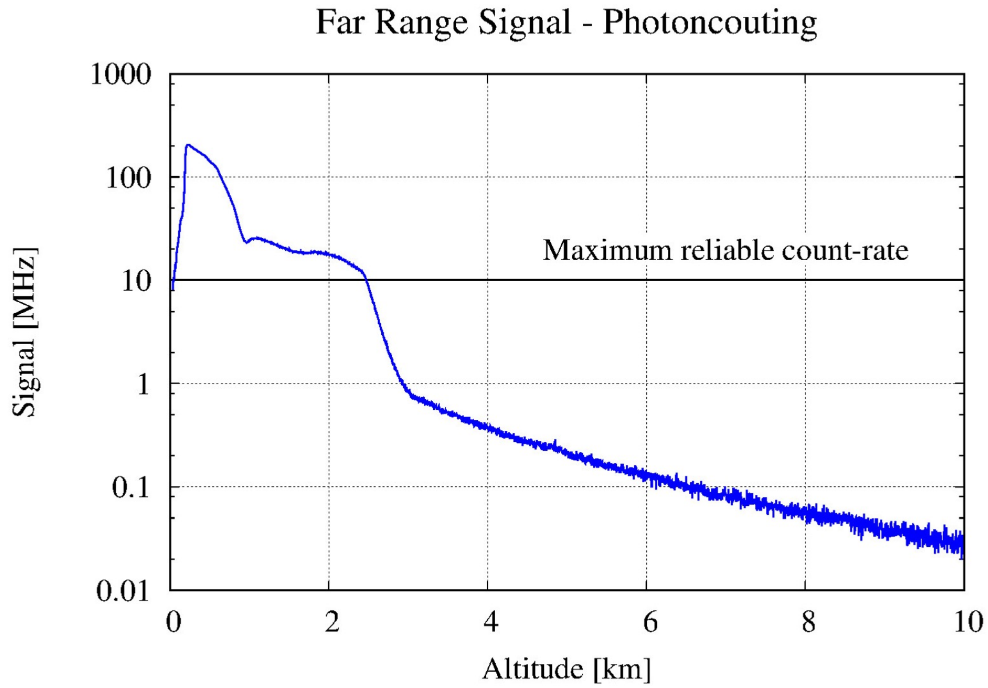
S_{max} ← maximum detectable input signal level

F ← parameter defining the “goodness” of the ADC converter

  - In case the near-range signal is detected in PC mode, z_{max} is determined by setting a lower threshold on SNR.

Signals glueing

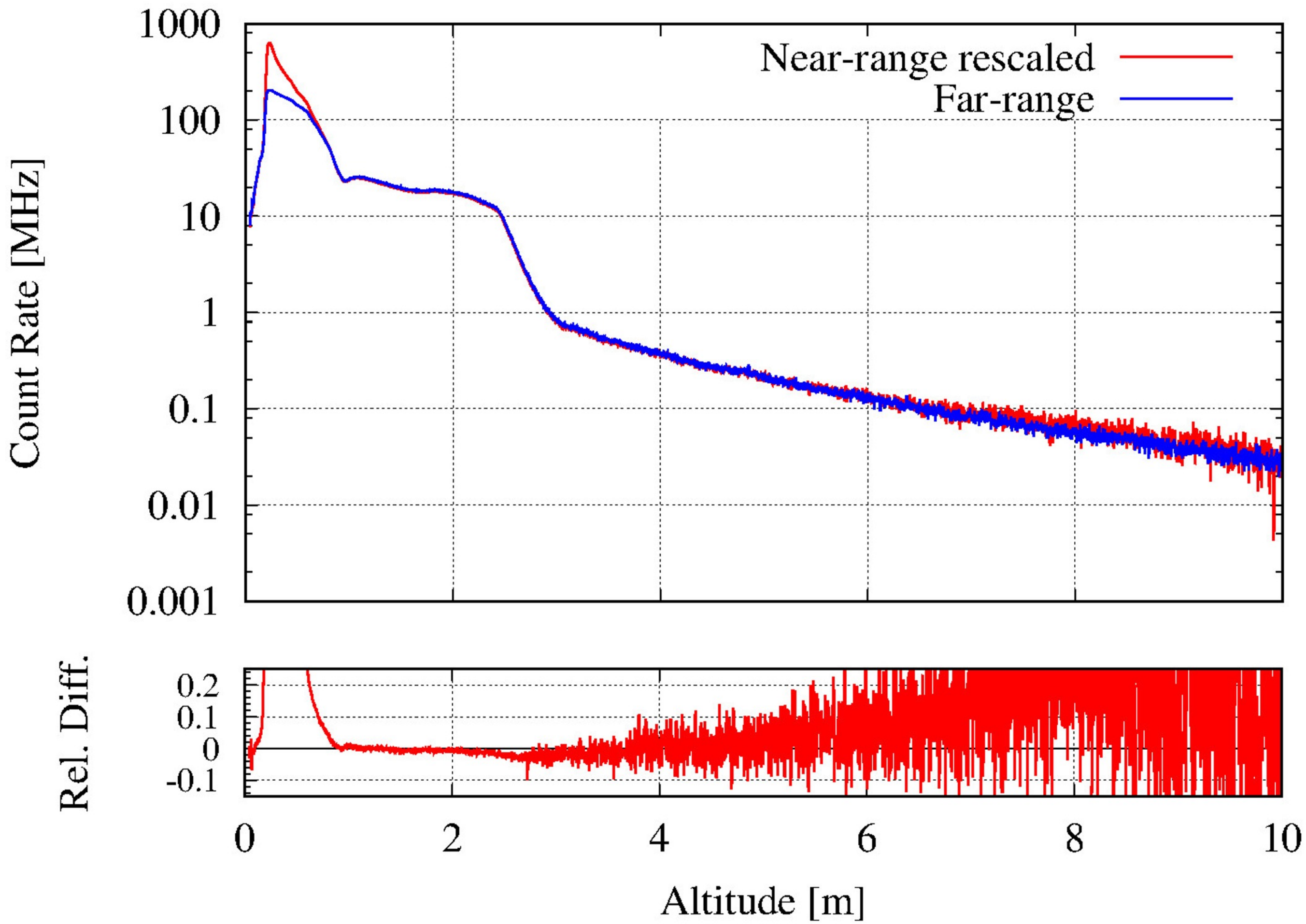




$z_{max} \simeq 3.9\text{km}$

$z_{min} \simeq 2.5\text{km}$

Signals glueing



Signals glueing

How to further improve the glueing (for low resolution pre-processing) ?

Slope test

Calculate residuals $R(z) = KS_{an}(z) - S_{pc}(z) = kz$

S_{an} and S_{pc} statistically "parallel" in the gluing region $\rightarrow$ $k \sim 0$

$$|k| < m\Delta k \qquad m: integer$$

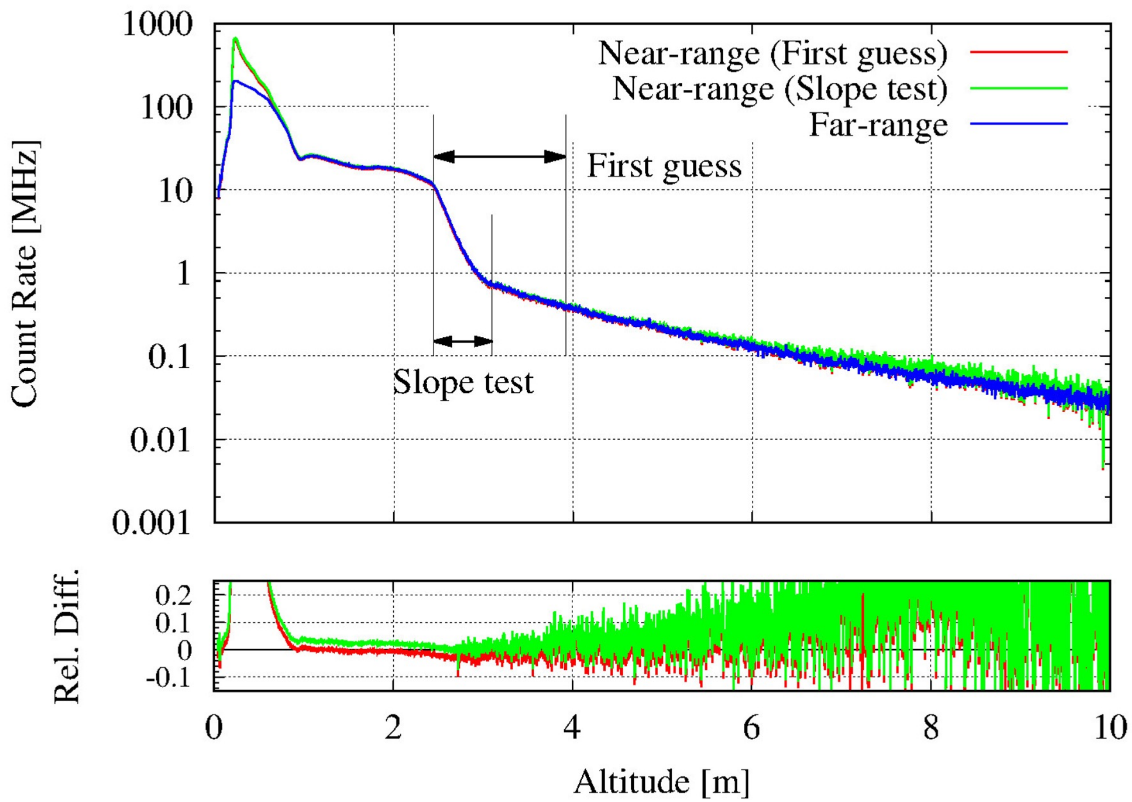
If the gluing range is large this condition may be too strict. In this case we introduce a constraint on the absolute value of the curvature C of the residuals, estimated from the difference of the slopes of the residuals in the first and the second half of the gluing range

$$C < m\Delta C \qquad |k_1 - k_2| < m\sqrt{\Delta k_1^2 + \Delta k_2^2}$$

G. D'Amico et al, EARLINET Single Calculus Chain -- technical Part 1: Pre-processing of raw lidar data, AMT 9, N. 2, 2016

Signals glueing

How to apply slope test?



Reduce the first guess of the glueing region until the slope test condition is satisfied

G. D'Amico et al, EARLINET Single Calculus Chain -- technical Part 1: Pre-processing of raw lidar data, AMT 9, N. 2, 2016

Signals glueing

Stability test (further test!)

The region, which has passed the slope test, is divided into two equal subregions, where the near-range signal is normalized on far-range one.

$$S_1(z) = K_1 S_{an}(z)$$

$$S_2(z) = K_2 S_{an}(z)$$

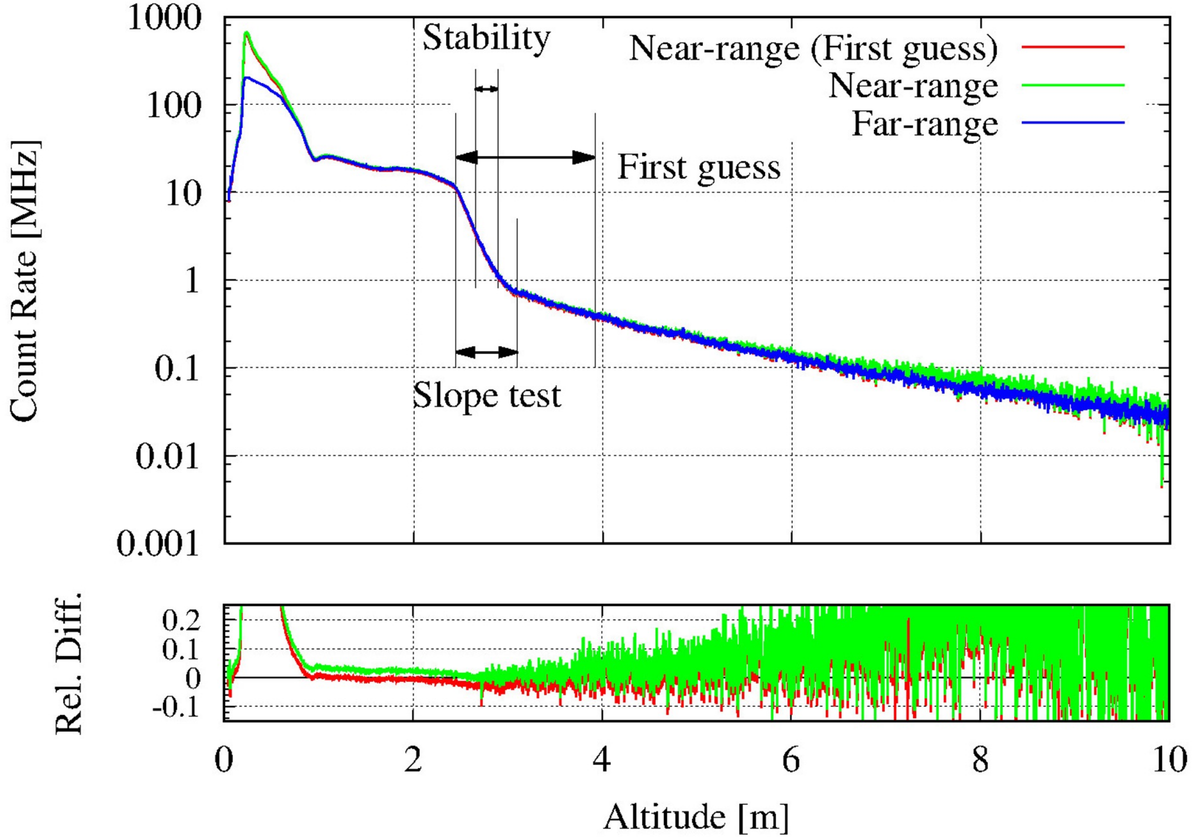
If the gluing region is chosen in a proper way, S_1 and S_2 should be indistinguishable taking into account the corresponding signal uncertainties.

$$|K_1 - K_2| < m\sqrt{\Delta K_1^2 + \Delta K_2^2}$$

G. D'Amico et al, EARLINET Single Calculus Chain -- technical Part 1: Pre-processing of raw lidar data, AMT 9, N. 2, 2016

Signals glueing

How to apply stability test?



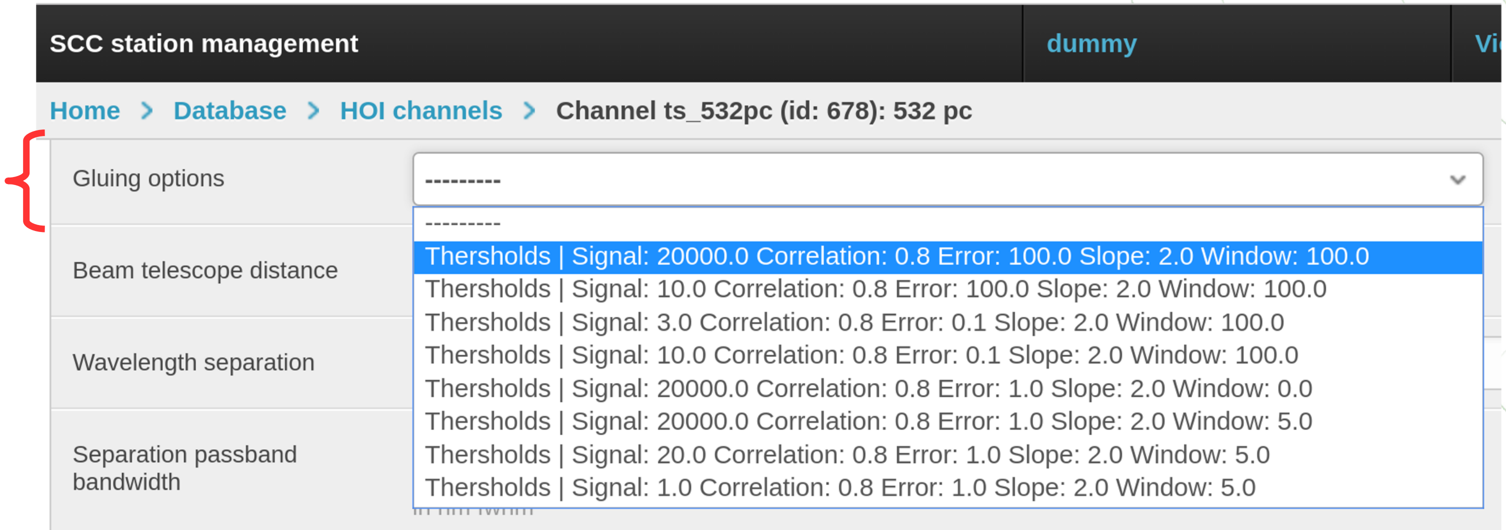
Reduce the glueing region until the stability test condition is satisfied.

G. D'Amico et al, EARLINET Single Calculus Chain -- technical Part 1: Pre-processing of raw lidar data, AMT 9, N. 2, 2016

Signals glueing

How to merge signals using SCC?

Select appropriate gluing options for each channel

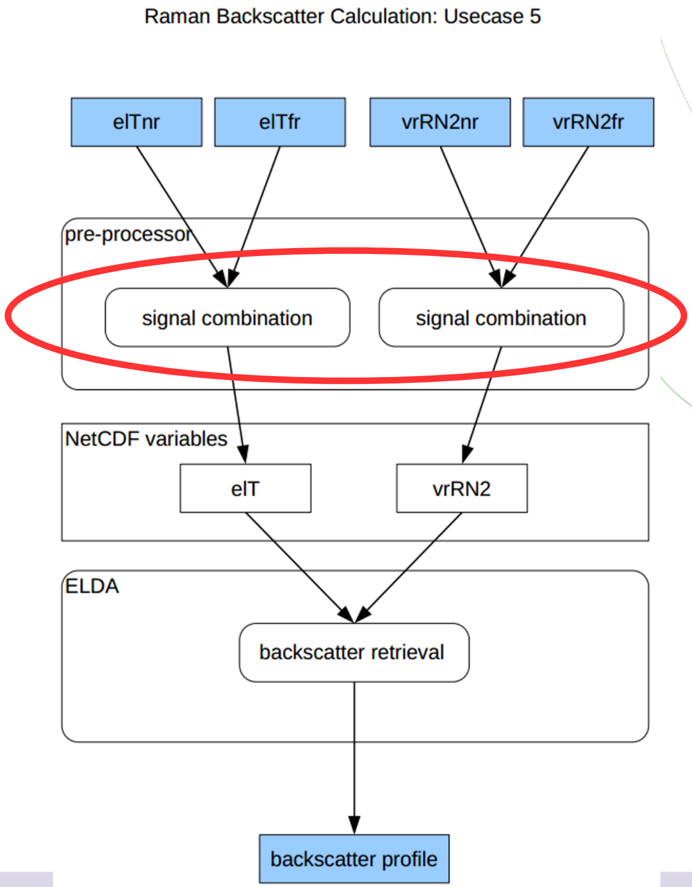


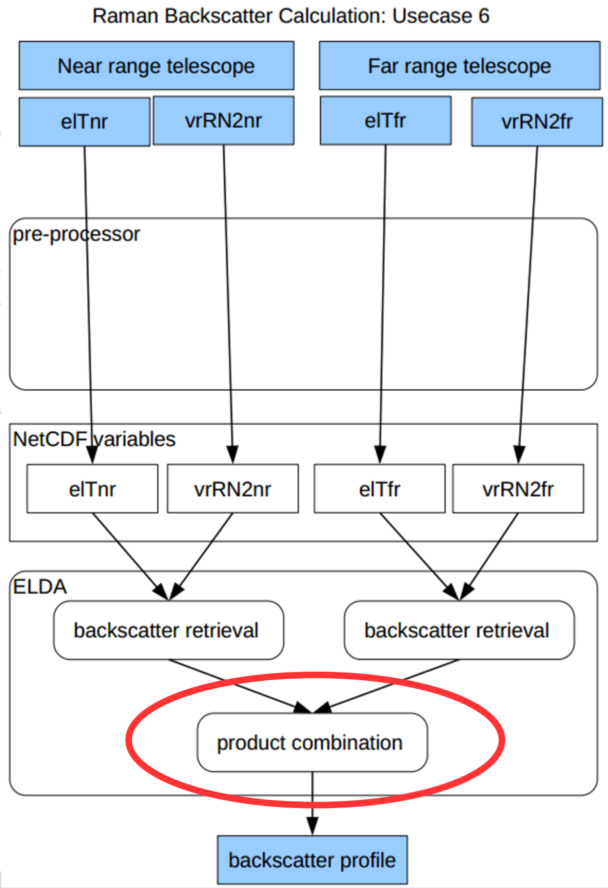
Signal Threshold	→	Maximum Count-Rate [MHz] for pc signals (default: 10) F factor for analog signals (default: 5000)
Correlation	→	Minimum correlation between nr and fr signals (default: 0.8)
Error Threshold	→	Minimum relative error for the data point (default: 1)
Slope	→	Number of st. dev. to evaluate slope test (m) (default: 2)
Window	→	window reducing step [m] (default: 0)

Signals glueing

How to merge signals using SCC?

Select appropriate usecase





Signals glueing

How to merge high resolution signals using SCC?

Multi-wave products → too much usecases needed to implement all options!

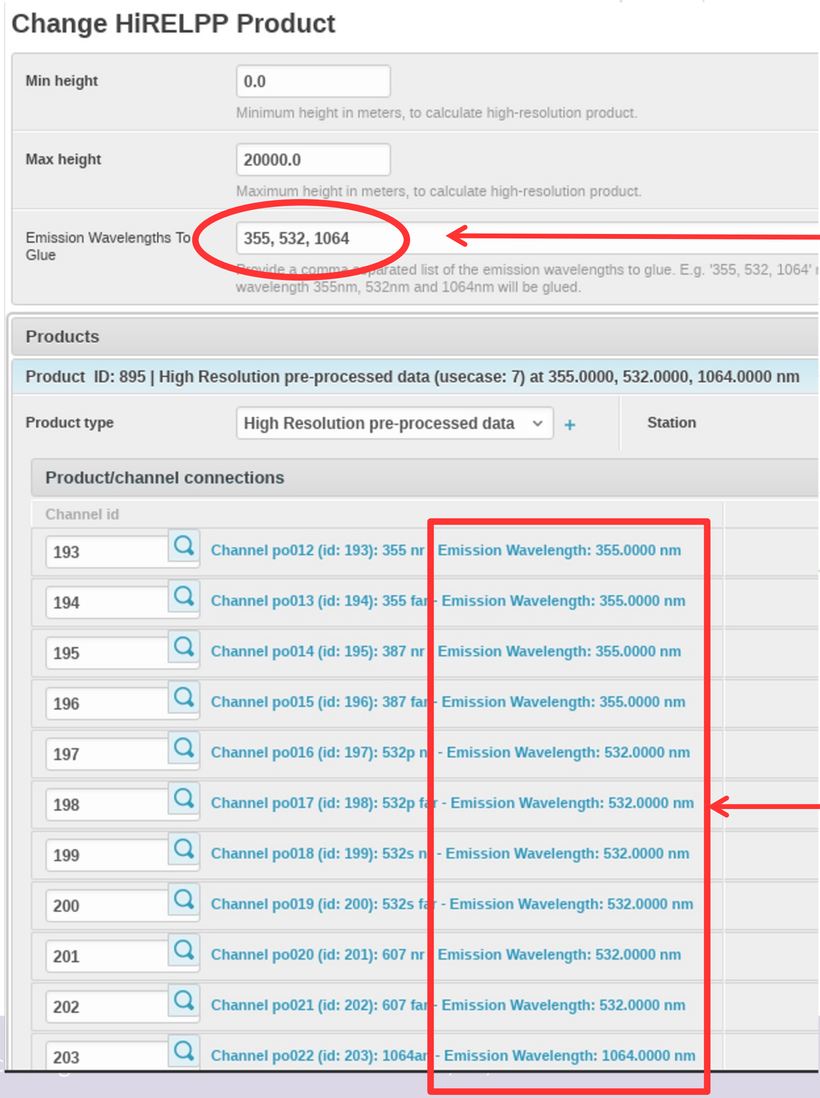
Easier solution:

- Group the channels on the base of emission wavelengths
- Unique glueing option for all the channels sharing the same emission wavelength
- For example, for a 3+2 system :

  Emission wave : 355nm → elTnr (355), elTfr (355), vrRN2nr (387), vrRN2fr (387)
  Emission wave : 532nm → elTnr (532), elTfr (532), vrRN2nr (607), vrRN2fr (607)
  Emission wave : 1064nm → elTnr (1064), elTfr (1064)

- If emission wavelength 532 is selected for glueing both the elastic and Raman channels in the green will be glued
- Not possible to glue only 532 and to keep 607 separate or to glue 387 and not 355 channels.

Signals glueing



Select emission wavelengths to glue

Uncertainties propagation

Standard uncertainties propagation

$$y = f(x_i)$$

assuming Gaussian independent variables (statistical error)

$$\Delta y = \sqrt{\sum_i \left(\frac{\partial f}{\partial x_i}\right)^2 \Delta x_i^2}$$

assuming uniform independent variables (systematic error)

$$\Delta y = \sum_i \left|\frac{\partial f}{\partial x_i}\right| \Delta x_i$$

Uncertainties propagation

Monte Carlo simulation

Numerical approach used when standard error propagation is not possible or too complex (for example, if interpolation or smoothing routines are applied).

If x_i are statistical variables with a known distribution (for example Gaussian, Possonian,...) the uncertainty on $y=f(x_i)$ can be estimated by the following steps:

- get “new” values of all x_i (let's call them x'_i) by randomly varying x_i according to their statistical distribution and compute

$$y_1 = f(x'_i)$$

- repeating the above step a statistically meaningful number of times (N), the error on y can be estimated calculating the standard deviation:

$$\Delta y = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(y_i - y)^2}$$

Uncertainties propagation

Photon-counting signals

- assumed to obey to Poisson statistic
- the statistical error can be evaluated for each photon-counting raw signal rangebin as the square root of the corresponding count
- the uncertainty of photon-counting signals can be propagated from the beginning to the end of the analysis chain

Uncertainties propagation

Analog signals

- assumed to obey to Gaussian statistic
- the standard deviation cannot be inferred from the mean value like for the Poissonian case!
- provide the raw analog signal time series and the corresponding statistical error time series (applicable only for systems which are able to measure such kind of values e.g. by storing not only the mean values but also the sum of the square values)
- statistical error estimator (especially for high resolution)
- calculates the statistical errors of analog signals only after a time or space averaging as the standard error of the mean (in all the operations made before that the uncertainty on analog signals is not propagated)

Estimation of statistical uncertainties on raw signals

Photon-counting

square root of the raw counts (after dead time correction)

Analog

standard deviation calculated using sliding average

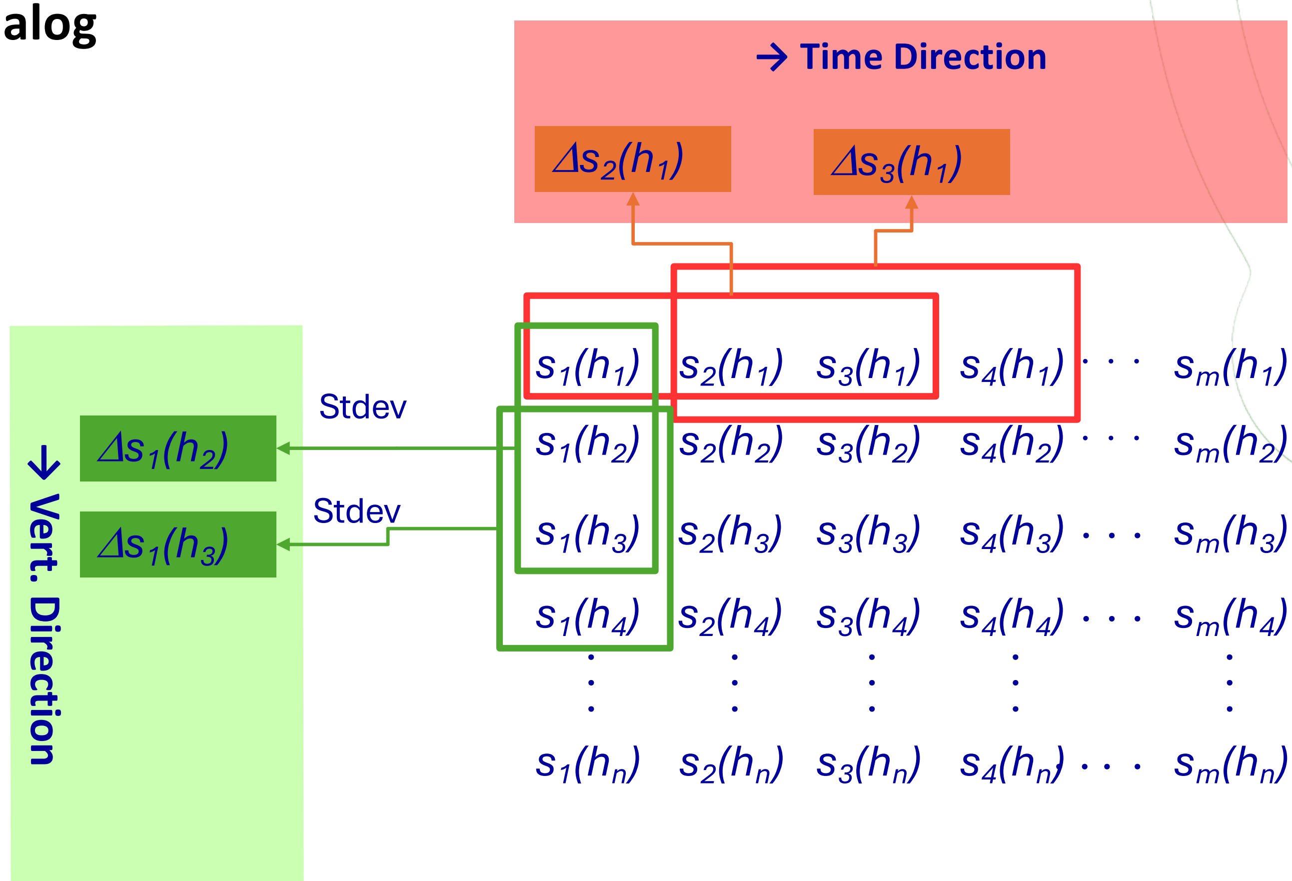
both vertical and horizontal average are available as options :

ANALOG_ERROR_ESTIMATION_BASED_ON_SPACE ← currently used (7 bins)

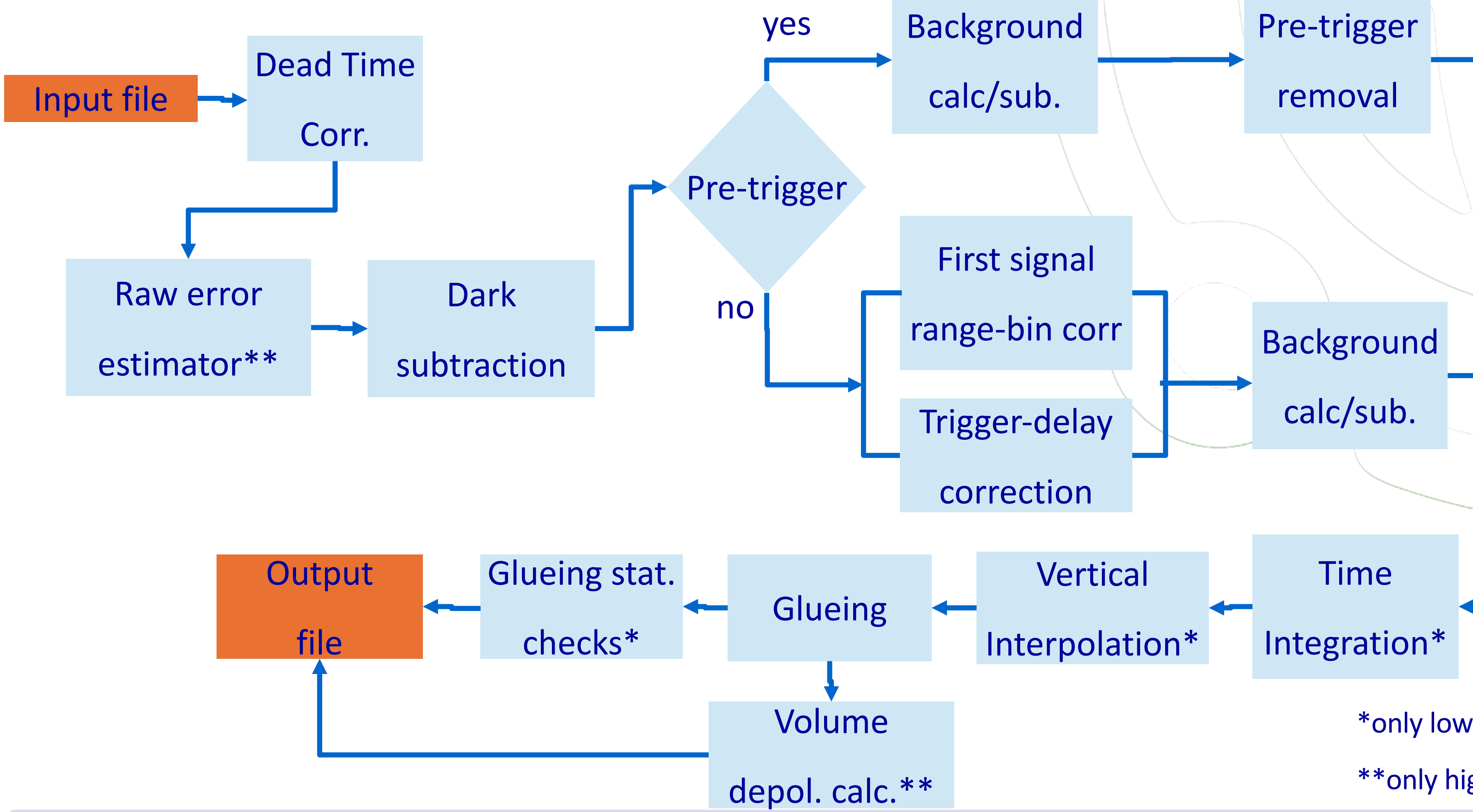
ANALOG_ERROR_ESTIMATION_BASED_ON_TIME

Estimation of statistical uncertainties on raw signals

Analog



Pre-processing Workflow



*only low-resolution products

**only high-resolution products

Molecular calculation

- During the pre-processing also the molecular extinction, backscatter and depolarization ratio is calculated
- Same vertical resolution of pre-processed signals
- Used during the optical processing as reference signals
- Options for the calculation of molecular density (to be specified in the input NetCDF raw datafile):

  Molecular_Calc=0 → Automatic*

  Molecular_Calc=1 → Correlative Sounding

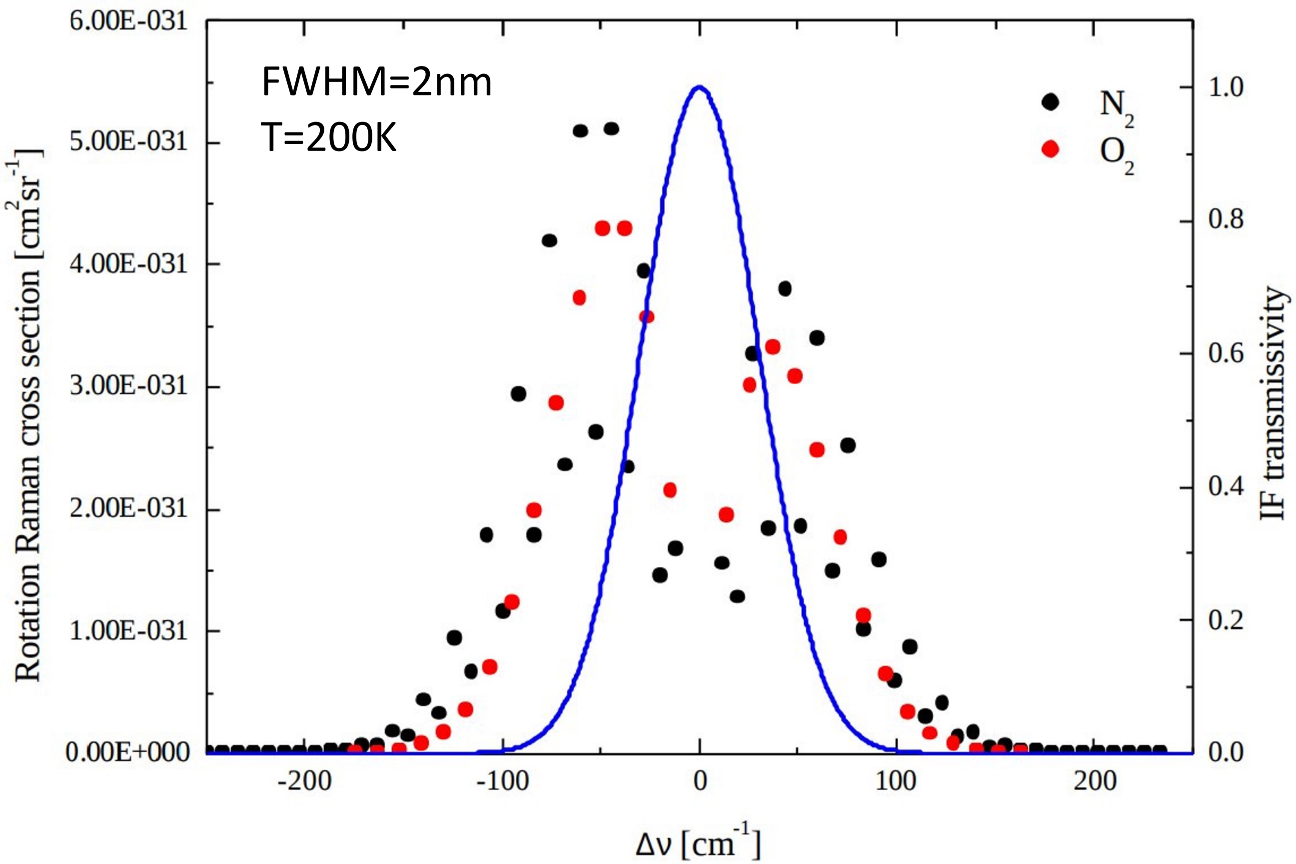
  Molecular_Calc=2 → Model data (from CloudNet not available for all the stations)

  Molecular_Calc=3 → Standard atmosphere

*first check model data on CloudNet if not found standard atmosphere is used

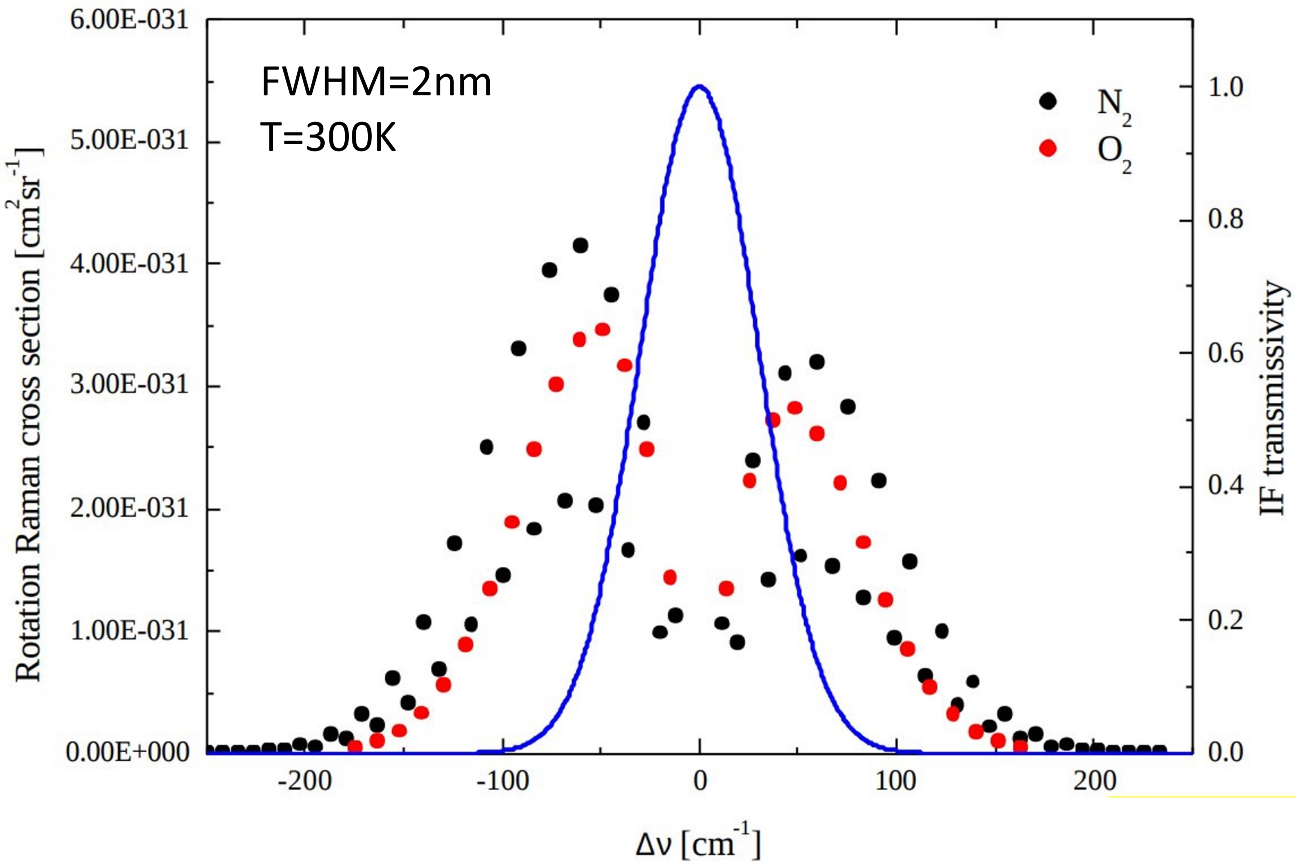
Molecular calculation

- Interferential filter center, bandwidth and actual atmospheric temperature are considered



Molecular calculation

- Interferential filter center, bandwidth and actual atmospheric temperature are considered



ITINERIS

THANKS!

IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-Mission 4 “Education and Research” - Component 2: “From research to business” - Investment 3.1: “Fund for the realisation of an integrated system of research and innovation infrastructures”

Finanziato dall'Unione europea NextGenerationEU

Ministero dell'Università e della Ricerca

Italiadomani