LSI Correlator

User's Manual

Version 3.2.1

January 2018

LS Instruments AG
Switzerland
www.lsinstruments.ch

Contents

1 LSI Correlator

The *LSI Correlator* by *LS Instruments* is among the most sophisticated instruments for data processing of photo-electron signals. Its core feature consists in the real-time computation of the auto-correlation or cross-correlation function originating, for example, from time-dependent scattered photon pulse trains.
The LSI Correlator implements both the multiple-tau and linear-tau digital correlation algorithms which allow spanning the largest lag-time range nowadays available. This makes our correlator suitable for DLS particle sizing of both fast and slow dynamics samples.
Our device is especially designed to offer the greatest flexibility, robustness, reliability, and platform portability. The hardware architecture of LSI Correlator makes it a powerful stand-alone device, and its modular design allows for a high degree of customization.
The following list details the main features offered by the LSI Correlator:

- **Multi-τ Auto- and Cross-Correlation:** $16/8$ multi-tau correlation scheme, covering lag-times spanning form 12.5 ns up to 3436 s by means of 322 channels, covering a lag time range of 54976 s, hence spanning 13 orders of magnitude.

- **Programmable fastest lag-time:** The LSI Correlator offers the possibility to choose the fastest lag-time of the averaged multi-τ correlation functions in order to reduce the statistical noise for signals that don't need the fastest channels. Initial selectable lags are: 12.5 ns, 200 ns, 400 ns, 800 ns, 3200 ns.

- **Normalization:** Both symmetric and compensated normalizations are selectable. To this end 296 monitor channels are implemented.

- **Linear Correlator:** User selectable sampling times ranging from 0.4 μs up to 25.6 μs. The channel layout is user-programmable over 1024 user selectable lags to be chosen among more than 10^6 available channels.

- **API:** The LSI Correlator can be accessed through an application programming interface (API) available for the MATLAB® and LabVIEW® environments.

- **Communication:** The LSI Correlator communicates either over USB 2.0 or Gigabit Ethernet in case a remote networked usage is required.

- **OS:** Windows and Linux support.

- **Software:**The device is accompanied by a user friendly acquisition software for device control and advanced data-treatment (CONTIN, Cumulant Analysis, Exponential Fitting).

- **Detector Overload Protection:** The LSI Correlator implements a fast gating output that is triggered when the instantaneous count rate exceeds a user programmable threshold over a user programmable time interval. This functionality is active even when not measuring and helps protecting expensive detectors from unexpected overloads.

- **Synchronization Clock Output:** An output clock is available in case the user wants to synchronize the correlation function computation with the operation of other devices present in their setup.

- **Upgradeable Firmware:** The firmware of the LSI Correlator can be upgraded in few minutes by the user. This allows the end user to stay up-to-date with the latest features developed by the LS Instruments team.

- **Maximum Count Rate:** 20 Mcps over 52 ms integration interval.

- **Standard count trace sampling time:** 52 ms
- **Size:** The correlator in housed in protective aluminum housing of $6x12x12$ cm.

2 Introduction

2.1 Photon Correlation Spectroscopy

Photon Correlation Spectroscopy (PCS) refers to an ensemble of optical non-destructive techniques used for analyzing dynamic properties and size distribution of a broad variety of physical, chemical, and biological systems composed of several interacting suspended constituents. Major examples of PCS include Dynamic Light Scattering (DLS) [1], Fluorescence Correlation Spectroscopy (FCS) [2] and Diffusing Wave Spectroscopy (DWS) [3].

PCS is based on the extraction of spectral information derived from time-dependent fluctuations of the light scattered from a sample volume. Specifically, when a suspension of colloidal particles or a solution of macromolecules is hit by a monochromatic coherent beam of light, generated scattered light waves spread out in all directions. Scattered waves interference in the far field region generates a net scattered light intensity $I(t)$, which displays stochastic fluctuations depending on whether the interference is constructive or destructive due to the random motion undergone by suspended particles.

More specifically, in a DLS experiment the suspended particles making up the colloidal dispersion under investigation undergo Brownian motion. This motion results in fluctuations of the distances between the particles and hence also in fluctuations of the phase relations of the scattered light. Additionally the number of particles within the scattering volume may vary in time. The end result is a fluctuating scattered intensity $I_s(q,t)$. The corresponding measured intensity correlation function is written as:

$$g_2(q,\tau) = \frac{\langle I_s(q,t) I_s(q,t+\tau)\rangle}{\langle |I_s(q,t)|^2\rangle} \tag{2.1.1}$$

with τ being the time lag and q being the scattering vector module. By means of the Siegert relation $g_2(q,\tau)$ can be related to the electric field correlation function $g_1(q,\tau)$:

$$\begin{aligned} g_2(q,\tau) &= 1+\beta|g_1(q,\tau)|^2 && (2.1.2)\\ g_1(q,\tau) &= \frac{\langle E_s(q,t) E_s^*(q,t+\tau)\rangle}{\langle |E_s(q,t)|^2\rangle} && (2.1.3)\end{aligned}$$

β being the so-called intercept. The field correlation function may be used to determine the diffusion coefficient D of the scatterers. In the simplest approach $g_1(q,\tau)$ is fitted to an exponential function

$$g_1(q,\tau) = \exp(-\Gamma\tau) \tag{2.1.4}$$

yielding the decay rate, Γ. From its definition

$$\Gamma = q^2 D \tag{2.1.5}$$

one obtains the diffusion coefficient. By using the Stokes-Einstein equation one can obtain the hydrodynamic radius as

$$R = \frac{kT}{6\pi\eta D} \tag{2.1.6}$$

with k the Boltzmann, T the temperature, η the viscosity of the suspending medium and R the hydrodynamic radius of the particle.

2.2 Digital Correlation

The technology underlying PCS relies on the digitalization of the scattered signal. This can be recorded with a suitable photon detector, such as a photomultiplier (PMT) or an avalanche photodiode (APD). Both the detectors will produce electronic current pulses with a probability proportional to $I(t)$. Output pulses are then converted to voltages by a wide-band amplifier and fed to a discriminator which compares voltage pulses with a reference threshold, by eliminating noise components and reducing the input light signal to a time-dependent train of random binary levels $n(t)$ according to the detection or the absence of pulses.

2.2.1 Linear Correlator

A discrete approximation of the unnormalized photon count rate autocorrelation function, reads:

$$G(\tau_k) = \frac{1}{N}\sum_{i=1}^{N} n(t_i)n(t_i - \tau_k) \tag{2.2.1}$$

where the time axis has been discretized over N samples and τ_k is the time lag between the counting rates $n(t_i)$ and $n(t_j)$. Computation of the previous quantity can be carried out through a digital correlator, whose simplest schematic is shown in Fig. 2.1. The basic operations performed by a digital correlator for N samples are:

1. Count photoelectron pulses falling within a sampling time Δt.
2. For each correlation channel k, delay the samples by Δt, such that $\tau_k = k\Delta t$.
3. For each channel, multiply the delayed samples by the current one.
4. For each channel, sum the products.

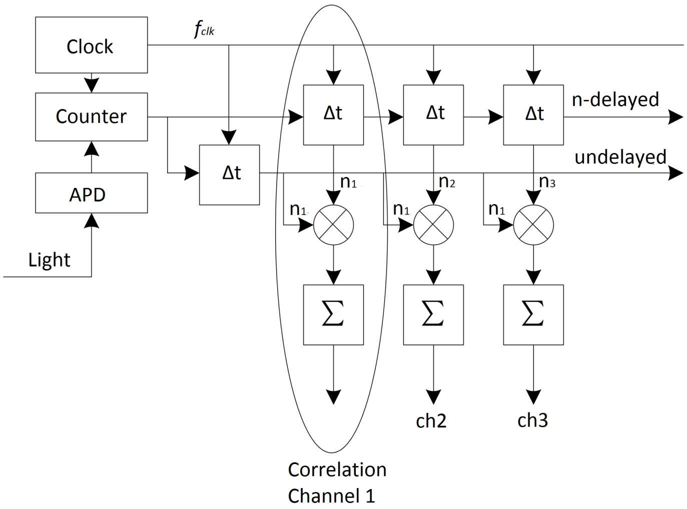


Figure 2.1: Schematic of a standard digital linear correlator for three correlation channels [4].

This scheme, known as *Linear correlator*, is characterized by a subsequent set of k equally spaced time operations for a fixed sampling time Δt. The extension of the lag time range is evidently determined by the maximum number of channels k available. The main drawback of this method lies in the the finite number of channel implementable. In order to obtain high measurement accuracy (i.e. avoid finite sampling systematic errors), a very short sampling time, usually of the order of 10 ns, is required. Under this assumption, a total measurement duration of even 1 s would require a dynamic range of 10^8 s and a corresponding number of channels of 10^8, which is technically unfeasible.

In addition, the inherent discretization of the time axis, and the resulting averaging of the photoelectron signal, sampled over Δt, yields to distortions of a decaying correlation function, increasingly evident when the lag time τ_k is considerably larger than Δt.

2.2.2 Multi-τ Correlation

In order to circumvent the disadvantages of the linear correlator, other schemes, based on non-linearly spaced time sampling have been proposed [5]. In particular, the multi-τ architecture is composed of a sequence of linear correlator blocks, each one characterized by a sampling time doubled compared to the correlator one level up in the hierarchical structure. This technique combines the fine timing resolution of the linear correlator with the advantage of a large dynamic range by establishing a quasi-logarithmic time grid. A general schematics of the LSI correlator main algorithm is depicted in Fig. 2.2. The photon pulse train emitted by the photomultiplier is firstly acquired and pulses are counted over an initial sampling time Δt. Each m-th correlator is clocked via its peculiar sampling time $2^m\Delta t$.

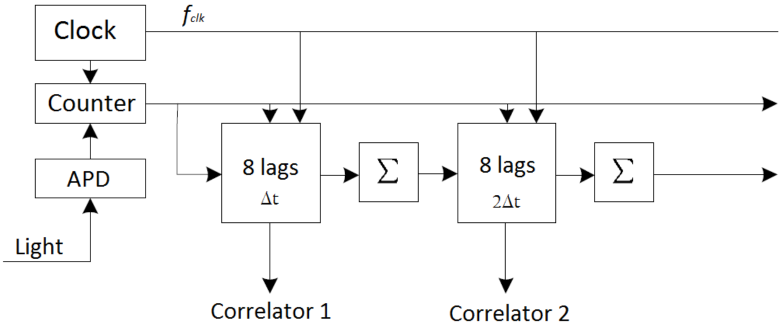


Figure 2.2: Schematic of a multi-τ correlator composed of two linear correlators, each of which composed of channels [6].

Within this architecture, constituted of a cascade of linear correlator blocks, when a new sample is ready (Fig. 2.2 form left to right), it is transferred to the first correlator that computes the correlation function for 8 correlation channels. Once two samples have been processed, they are accumulated and sent to the next correlator which in turn, repeats the same operation.

The multi-τ framework based on a semi-logarithmic spacing of the lag-time axis holds the ability to cover extremely large lag time ranges with a small number of channels without resulting in substantial sampling errors. Such errors can be kept negligibly small by using lag times larger than the sampling time. Specifically, a factor $\tau_k/\Delta t = 8$ suffices to reduce the absolute distortions below the typical statistical accuracy in PCS measurements [5].

2.2.3 Normalization

Normalization of the numerical correlation function consists in the baseline subtraction and division by the square of the count rate estimator, hence yielding the normalized intensity correlation function:

$$g_2(\tau_k) = \frac{G(\tau_k) - n_0^2}{n_0^2} \tag{2.2.2}$$

Where $n_0 = \sum_{i=1}^{N} n(t_i)/N$ is the count rate estimator. This procedure requires the addition of an extra monitor channel for computing the total photoelectron pulses detected, n_0.
In order to obtain better noise performance, especially at large lags and sample times, two additional normalization schemes have been proposed [3]. In particular, if $n_k = \sum_{i=1}^{N-k} n(t_i)/N$ is the partial sum of those samples crossing the correlator channel k, the *symmetric* and *compensated* normalizations read respectively:

$$g_2^{sym}(\tau_k) = \frac{G(\tau_k) - n_0 n_k}{n_0 n_k}, \tag{2.2.3}$$

and

$$g_2^{com}(\tau_k) = \frac{G(\tau_k) - n_0 n_k}{n_0^2}. \tag{2.2.4}$$

Both advanced schemes evidently require extra monitor channels, for each single correlator channel and for each sampling time $2^m \Delta t$.

3 Basic Operation

This section gives the user basic information about the steps required to perform a measurement with the *LSI Correlator*. Hardware and device maintenance are discussed here, while software usage is addressed in separate sections.

3.1 Device Connections

A picture of the *LSI Correlator* device connector panel is presented in Fig. 3.1, where it's backside is shown. A basic set of I/O cable adapters are provided by *LSI Correlator*. Available ports and I/Os are listed below according to Fig. 3.1 numbering:

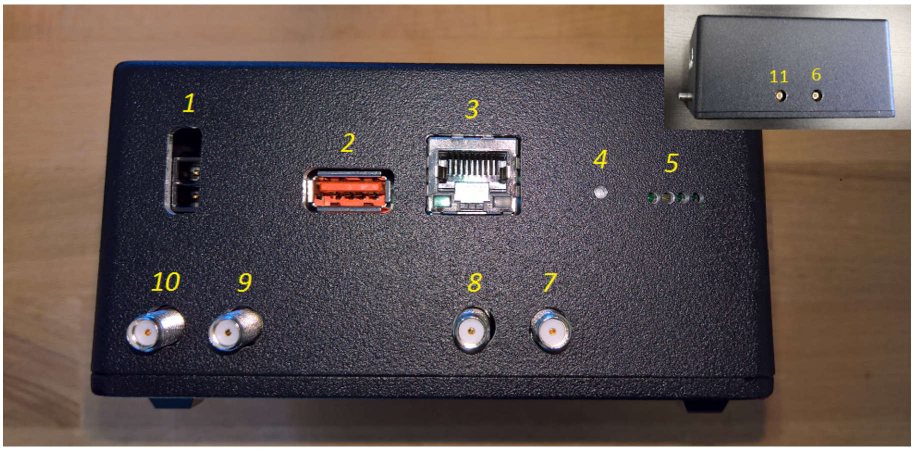


Figure 3.1: LSI Correlator connection panel

1. **Power Connector:** *LSI Correlator* power supply outputs 12 V-1.6 It delivers a maximum power of 18 W and is CE/FCC certified. The power supply is provided by *LS Instruments*.
2. **USB Port:** *LSI Correlator* is equipped with a USB 2.0 connectors sustaining a maximum data rate of 480 Mb/s, for data transmission to personal computers. Since data streaming is based on Ethernet IEEE 802.3 technology, a USB-Ethernet cable adapter is provided with the *LSI Correlator*. On the user-end side the device can either be connected to a USB port via a second Ethernet-USB adapter or directly to a Local Area Network. In both cases the *LSI Correlator* software is able to detect the device by retrieving its network IP address.
3. **Service Port:** this Ethernet socket is solely used by LS Instrument for service and debug. The user must not connect any Ethernet cable to this port.
4. **Reset Button:** reset and restart the *LSI Correlator*.
5. **LEDs** (From left to right) The first green LED indicates that the correlator is powered. The second orange LED indicates the status of the device (On Momentarily: Booting; Blinking: ann error occurred, Off: LS Correlator is in run mode). The third and fourth LEDs are off.
6. **External Clock (Inset in Fig. 3.1: furthest from the connection panel):** Output square wave TTL signal for external device synchronization. The SMB related connector is located on the right side of the housing. Signal characteristics are listed in table 3.2.

7. **APD overload protection Channel A**: Output TTL signal (table 3.3) for APD gating control. According to software parameter ”Overload Timed Interval” and ”Overload Limit” described in the next section, the *LSI Correlator* gates the APD input if the count rate has reached a threshold over the user selectable time interval. This feature prevents dangerous voltage overshoots and counting overflows.

8. **APD overload protection Channel B:** same functionality as above but for a second channel B.

9. **Channel A input:** Signals originating from a PMT or an APD are input via coaxial cable. The available built-in female SMA connector can be adapted to a BNC termination via specific connector adapter, provided with the *LSI Correlator*. Channel A input is implemented with a 50 Ω matching line impedance. Supported input signal characteristics are listed in table 3.1

10. **Channel B input:** read previous entry on Channel A input.

11. **Output Trigger** (Inset in Fig. 3.1: closest to the connection panel) LLVT signal designed for mastering external devices. When no measurement is running the signal in off (LOW). When a measurement starts this trigger turns on and holds the value HIGH until the end of the measurement. A delay between the user starting action and the actual correlator acquisition start can be adjusted in the configuration panel (see sec 4.2).

If the *LSI Correlator* is powered, it must be also connected to a computer for data transmission. Two connection modes are possible according to Fig. 3.2: (1) USB/Ethernet (left) allows to connect the LSI Correlator USB port to the Ethernet Network or directly to the computer Ethernet socket; (2) USB/USB connects the correlator URB port to the computer USB port.

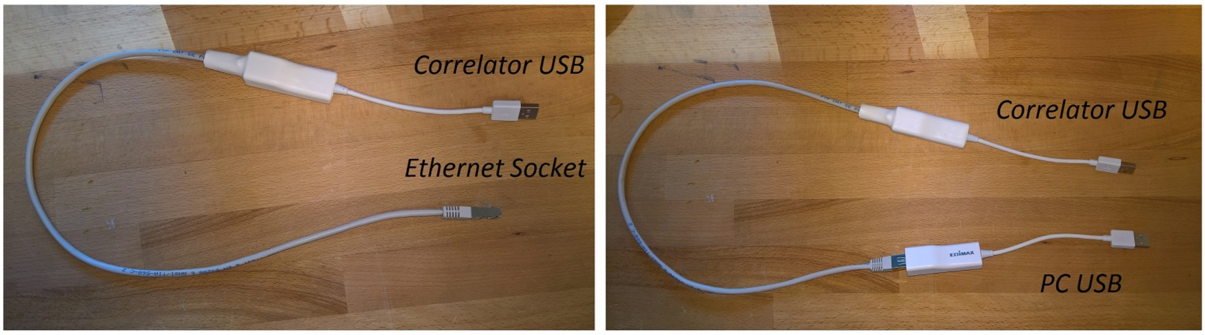


Figure 3.2: USB/Ethernet and USB/USB cable configurations

Note:

- In case no measurement is active and the device is powered and plugged correctly to the APDs the *LSI Correlator* constantly checks the presence of overload count rate and actively responds to an APDs overload accordingly. The same applies during a measurement is running. If an overload is detected the correlator will gate the photon detectors until a new measurement is started. It is the user responsibility to remove the cause of the overload before restarting a measurement.

- If for any reason the connection drops although the device is normally powered, the users should reset the correlator for restoring normal functioning.
- Output signal cables must be terminated with 50 Ω impedance for full LVTTL (TTL) compatibility.

3.2 Signal Characteristics

The following tables list the supported properties of the *LSI Correlator* digital input and output signals:

Input Signals	Specifications
Number of independent inputs	2
Supported input pulse voltage	2 x LVTTL (TTL tolerant)
Supported minimum input pulse width	$\sim$ 2.5 ns
Maximum Mcps Sustained	160 Mcps

Table 3.1: Input signals specification for Channel A and B

Note: The correlator is able to detect both TTL or LVTTL photoelectron pulses of width down to 2.5 ns separated by a dead time of about 10 ns. The supported average count rate in terms of computation before overflow is 20 Mc/s.

Output Clock	Specifications
Number of output	1
Delivered output voltage	LVTTL (TTL tolerant)
Accuracy	Rising edge: 2.5 ns, Peak Jitter: 250 ps
Clock period	200 ns, 400 ns, 800 ns, 1200 ns, 3200 ns (synchronized with the first lag time)

Table 3.2: Clock output specifications

Overload protection output	Specifications
Number of output	1
Delivered output voltage	LVTTL (TTL tolerant)
Response time after detection	12.5 ns

Table 3.3: Overload detection output specification for Channel A and B

3.3 Maintenance

In general we recommend that users never open the metal housing. This would void the device warranty. The *LSI Correlator* operates correctly at ambient temperatures from 0 °C to +60°C. Average power consumption is lower than 18 W. *LSI Correlator* must never be exposed to extreme temperatures, shocks, vibrations, etc..
In case of failures or hardware malfunctions, clean the *LSI Correlator* housing with a soft, nonmetallic brush. Make sure that the device is completely dry and free from contaminants before returning it to service. Please contact *LS Instruments* customer service to obtain replacement or service by a qualified technician.

4 LSI Correlator Acquisition Software

Version: 1.3.x.x
If an older version is installed on your system you might ask *LS Instruments* for an update.

This section explains in detail the structure and functionality of the *LSI Correlator Acquisition Software* provided with the device. The graphical user interface is organized into two separate tabs. The Acquisition tab, shown in Fig. 4.4, allows the user to configure the correlator and to perform data acquisition, while data analysis is performed in the Data Analysis tab shown in Fig. 4.6. Parameter settings as well as starting and measurement procedures are described below.

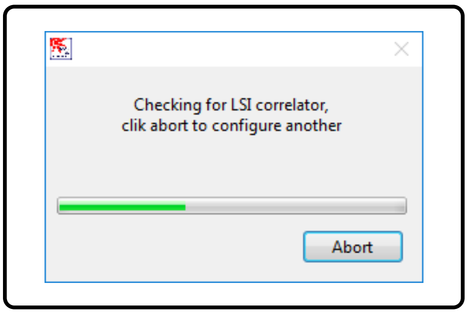
4.1 Requirements

The LSI Correlator Acquisition Software works on any PC with the following minimum requirements:

- Screen resolution $1680 \mathrm{x} 1050$
- Windows 7 or later
- English language locale settings
- USB 2.0 port and/or RJ45 Ethernet port

4.2 Configuration

The *LSI Correlator Acquisition Software* is able to recognize and connect automatically to the device once the latter has been powered and connected via USB/Ethernet cable and the correlator address set up in the configuration file. In this case the software user interface appears and measurements can be started. If the correlator is connected for the first time and/or the cable connection mode (Ethernet or USB) is changed, the device needs to be configured. In this case a first pop-up message window, as shown in Fig. 4.1, appears once the software is started. The software attempts to contact the hardware correlator for 30 seconds before launching a new configuration procedure as explained below.

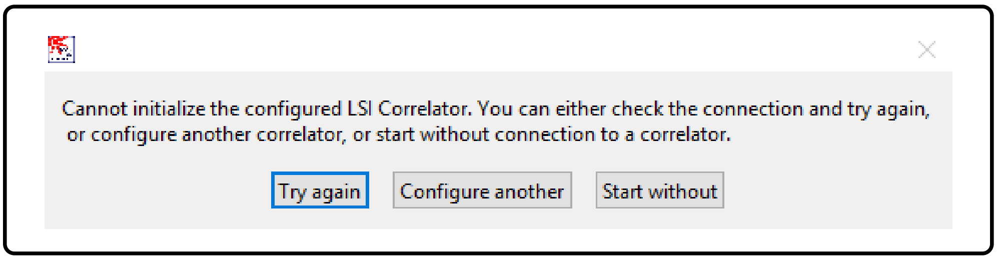


Checking for LSI correlator,
clik abort to configure another

Figure 4.1: Pop-up window of non-configured LSI Correlator Acquisition Software or booting hardware correlator.

Note: *If the correlator has been configured previously and it is properly connected, the pop-up windows in Fig. 4.1 appears because of the device booting. As soon as the booting completes, the pop-up closes and the LSI Correlator Acquisition Software starts automatically. The Abort button should not be pressed during booting, otherwise a new configuration is launched.*

The initial panel of the configuration procedure is displayed in Fig. 4.2.



Cannot initialize the configured LSI Correlator. You can either check the connection and try again, or configure another correlator, or start without connection to a correlator.

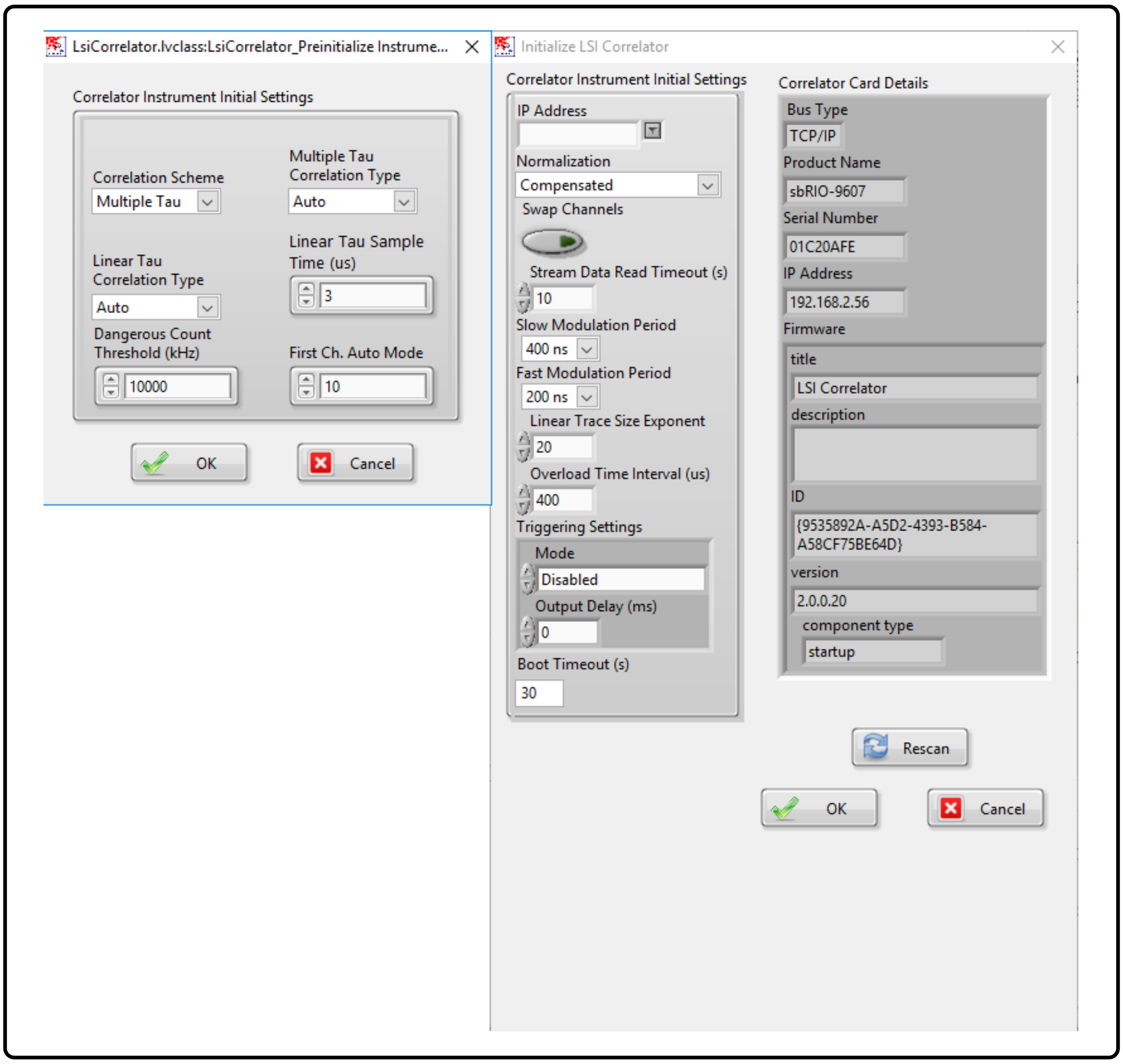
Figure 4.2: Pop-up window for non-configured LSI Correlator Acquisition Software.

The user has the choice here to either retry the connection to the pre-configured device by pressing the "*Try again*" button, or to discover and connect to another *LSI Correlator* by pressing the "*Configure Another*" button, or to start the software without a connection to a device by pressing the "*Start Without*" button. In the last case the user interface will appear but only data treatment will be allowed. If the device is connected for the first time the user will choose the "*Configure Another*" option.
In case this option is chosen two subsequent windows, shown in Fig. 4.3, will appear to guide the user through the initial configuration of the *LSI Correlator*. The window shown on the left side of Fig. 4.3 allows for the configuration of the following settings:

- *Correlation Scheme*: select Multi-τ or Linear-τ channel schemes.
- *Linear Tau Correlation Type:* select auto or cross-correlation mode for the linear-τ scheme. This value can be changed later on within the main software.
- *Multi Tau Correlation Type:* select auto or cross-correlation mode for the multi-τ scheme. This value can be changed later on within the main software.
- *Dangerous count Threshold:* Select the average photon count threshold (in kHz) beyond which the correlator shuts off of the APDs. An optimal value of 10000 kHz is advised. This value can also be changed later within the main software.
- *Linear tau Sample Time (us):* Select the sampling time (in μs) for the linear channel layout. The inserted integer will be rounded to the closest available sampling time values among 0.4, 0.8, 1.6, 3.2, 6.4, 12.8 and 25.6 μs. This value can also be changed later within the main software.
- *First Channel Auto Mode:* Exclude the first n points from the plot of the auto correlation function. A minimum value of 1 is advised.

The second window shown on the right side of Fig. 4.3 (right) and contains the remaining setting parameters as listed below (**non listed settings must remain unchanged**):

- *IP Address:* Select one among the available addresses listed in the drop-down menu. Multiple addresses are displayed if multiple correlators are detected. The software is able



LsiCorrelator.lvclass:LsiCorrelator_Preinitialize Instrume...

Correlator Instrument Initial Settings

Correlation Scheme: Multiple Tau

Multiple Tau Correlation Type: Auto

Linear Tau Correlation Type: Auto

Linear Tau Sample Time (us): 3

Dangerous Count Threshold (kHz): 10000

First Ch. Auto Mode: 10

Initialize LSI Correlator

Correlator Instrument Initial Settings

IP Address:

Normalization: Compensated

Swap Channels

Stream Data Read Timeout (s): 10

Slow Modulation Period: 400 ns

Fast Modulation Period: 200 ns

Linear Trace Size Exponent: 20

Overload Time Interval (us): 400

Triggering Settings

Mode: Disabled

Output Delay (ms): 0

Boot Timeout (s): 30

Correlator Card Details

Bus Type: TCP/IP

Product Name: sbRIO-9607

Serial Number: 01C20AFE

IP Address: 192.168.2.56

Firmware

title: LSI Correlator

description:

ID: {9535892A-A5D2-4393-B584-A58CF75BE64D}

version: 2.0.0.20

component type: startup

Figure 4.3: Configuration windows.

to retrieve all the addresses of the devices connected to the private network or locally if USB cabling is used. Once an IP address is selected, device information appear on the right section *Correlator Card Details*. If no IP address has been retrieved or the number of IP addresses doesn't correspond to the number of LSI Correlators connected, try to scan the network again (*Rescan* button), check if the device is connected or reset the device.

- *Normalization:* Select between Symmetric and Compensated normalization (see section 2.2.3 in chapter 2). This value can also be changed later within the main software.

- *Swap Channels:* If Correlation type is set to Auto-correlation mode, select the input channel for which the auto-correlation function is calculated ("off" for channel A, "on" for channel B). If Correlation type is in Cross-correlation mode, change the order of the input channel operands ("off" for ChA $\otimes$ ChB, "on" for ChB $\otimes$ ChA).

- *Overload Time Interval (us)*: Select the integration time interval (in microseconds) upon which the average photon count is integrated for the purpose of overload protection. This value can also be changed later within the main software.

- *Triggering Settings → Mode*: Select mode of operation the external triggering feature among the following options
  - "Disabled": The triggering feature is disabled.
  - "Output": When a measurement starts this trigger turns on and holds the the trigger output to the value HIGH until the end of the measurement. A delay between the user starting action and the actual correlator acquisition start timing can be set input field described in the following bullet point.
  - "Input Both Edges": Not yet implemented.
  - "Input Pulse': 'Not yet implemented.
- *Trigger Settings → Output Delay*: Set the delay time in ms between the external trigger rising edge, activated when the user presses the start button via software, and the actual measurement start, performed by the correlator.
- *Boot Timeout (s):* Set the timeout during which the LSI Correlator Software attempts to contact the hardware device before starting a new configuration procedure or hardware booting is completed. Default value is 30 seconds.

Once the *LSI Correlator Acquisition Software* is started, information about the correlator status is displayed in the bottom-left section ("*Correlator Management*") of the main software panel shown in Fig. 4.4. Further configuration can be accessed through "*Change Default*" button contained within.. This button allows for the change of the correlator configured by default in case the user wants to acquire data from another correlator.

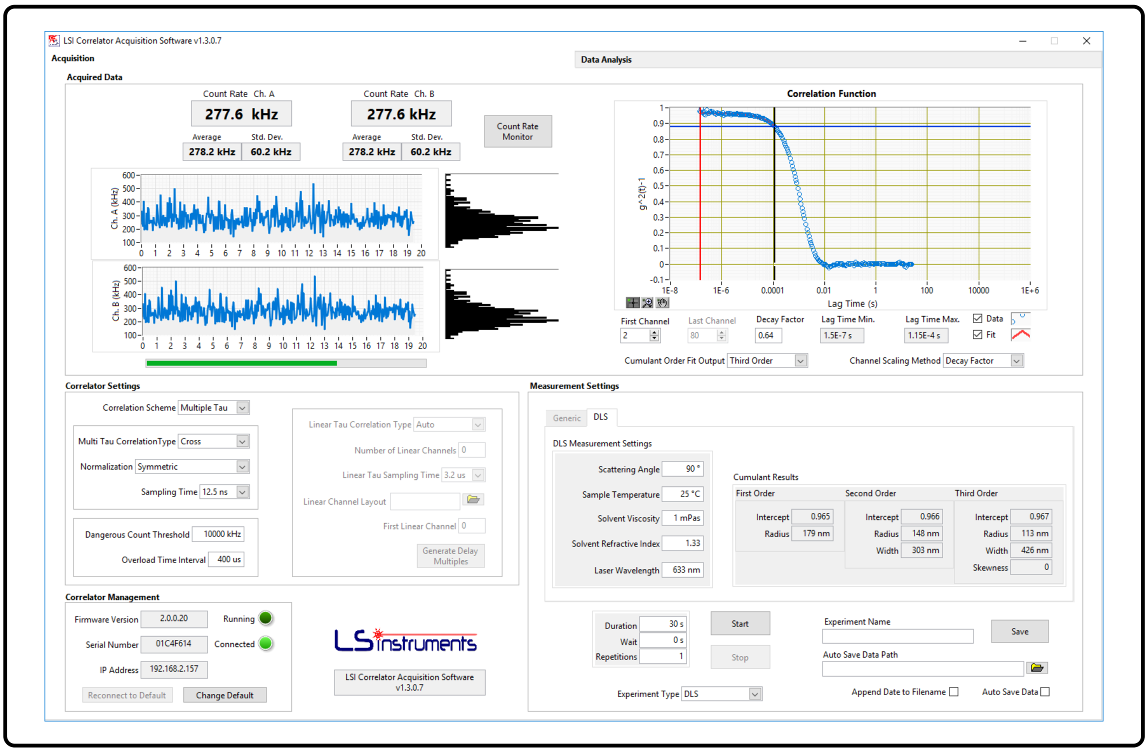
4.3 Acquisition

The acquisition tab shown in Fig. 4.4 is organized in 4 different sections described below.

The **Acquired Data** section shows all relevant information during acquisition after defining the parameters in the settings section and starting the measurement.

On the upper left side are the count rate traces for both (or one) detector. On the right side of this graph the distribution of the count rate is shown. This latter graph is useful for assessing the quality of a measurement. If, for example, a dust particle enters the scattering volume the distribution will clearly indicate this by displaying a long tail towards large intensities. Additional indicators show the current count rate (*Count Rate ChA/ChB*), the average count rate over the entire acquisition duration and the related standard deviations for both channels. In this section we have as well the button **Count Rate Monitor** whose main purpose is allowing to adjust the count rate before starting to measure. Often in fact, the experimental setup (e.g. scattering angle, laser intensity, etc.) needs to be calibrated in order to obtain a suitable count trace. The Count Rate Monitor as shown in Fig. 4.5 displays the photon trace while the measurement is still inactive and facilitates the overall calibration.
Finally, a progress bar, at the bottom of the section, shows the progress of the total measurement, by taking into account all the consecutive repetitions and possible delays set by the users.

The upper right graph displays in real-time the averaged correlation function. From the graph palette menu and by right-clicking on the graph itself, it is possible to zoom and control the graph appearance if *autoscale* option is unchecked.
The graph is equipped with movable red, blue, and black cursor lines appearing above of the correlation function plot. The two vertical lines specify the lowest and highest channels defining



LSI Correlator Acquisition Software v1.3.0.7

Acquisition

Data Analysis

Acquired Data

Count Rate Ch. A: 277.6 kHz
Average: 278.2 kHz Std. Dev.: 60.2 kHz

Count Rate Ch. B: 277.6 kHz
Average: 278.2 kHz Std. Dev.: 60.2 kHz

First Channel: 2
Last Channel: 80
Decay Factor: 0.64
Lag Time Min.: 1.5E-7 s
Lag Time Max.: 1.15E-4 s
☑ Data
☑ Fit
Cumulant Order Fit Output: Third Order
Channel Scaling Method: Decay Factor

Correlator Settings

Correlation Scheme: Multiple Tau
Multi Tau CorrelationType: Cross
Normalization: Symmetric
Sampling Time: 12.5 ns
Dangerous Count Threshold: 10000 kHz
Overload Time Interval: 400 us
Linear Tau Correlation Type: Auto
Number of Linear Channels: 0
Linear Tau Sampling Time: 3.2 us
Linear Channel Layout:
First Linear Channel: 0

Correlator Management

Firmware Version: 2.0.0.20
Serial Number: 01C4F614
IP Address: 192.168.2.157
Running ●
Connected ●

LSI Correlator Acquisition Software v1.3.0.7

Measurement Settings

DLS Measurement Settings

Scattering Angle: 90 °
Sample Temperature: 25 °C
Solvent Viscosity: 1 mPas
Solvent Refractive Index: 1.33
Laser Wavelength: 633 nm

Cumulant Results

First Order		Second Order		Third Order	
Intercept	0.965	Intercept	0.966	Intercept	0.967
Radius	179 nm	Radius	148 nm	Radius	113 nm
		Width	303 nm	Width	426 nm
				Skewness	0

Duration: 30 s
Wait: 0 s
Repetitions: 1
Experiment Type: DLS
Experiment Name:
Auto Save Data Path:
Append Date to Filename ☐
Auto Save Data ☐

Figure 4.4: Acquisition tab of the LSI Correlator Acquisition Software user interface.

the correlation function subset taken into account in the cumulant analysis fit performed in real-time during measurement. By dragging the red cursor line the user can select the lower limit. The upper limit data value of the correlation function is pinpointed by two orthogonal cursor lines, one vertical and one horizontal crossing on the last data-point of the correlation function included in the cumulant fit. Their operation changes depending on which *channel scaling* method is selected.
If ”No Scaling” is selected the vertical cursor line is blue and the horizontal black. In this case the user can drag the vertical blue line and change the last channel included in the cumulant fit, the horizontal black line will adapt its position depending on the position of the blue line and the value of the correlation function at that lag time. In this mode the value of the last channel can be also be input directly in the corresponding field, the cursor lines will adapt automatically If the ”Decay Factor” *channel scaling* method is selected, instead, the horizontal cursor line will become blue and the vertical black. The user is then able to drag in the vertical direction the former to select which fraction of the decay, called ”decay factor”, of the correlation function she/he wants to include in the fit. The software will adapt automatically the position of the cursor lines to ensure that their position corresponds always to the same decay factor even if the correlation function values change during the measurement. The decay factor can also be input directly in the corresponding field in the user interface.
The **Correlator Settings** section contains all the software controlled settings of the *LSI Correlator*:

- *Correlator Scheme*: select Multi-τ or Linear-τ correlation schemes.
- *Multi Tau Correlation Type:* select auto or cross-correlation mode for the multi-τ scheme.
- *Normalization:* select either Symmetric or Compensated normalization for the Multi-τ scheme (see section 2.2.3 in chapter 2).

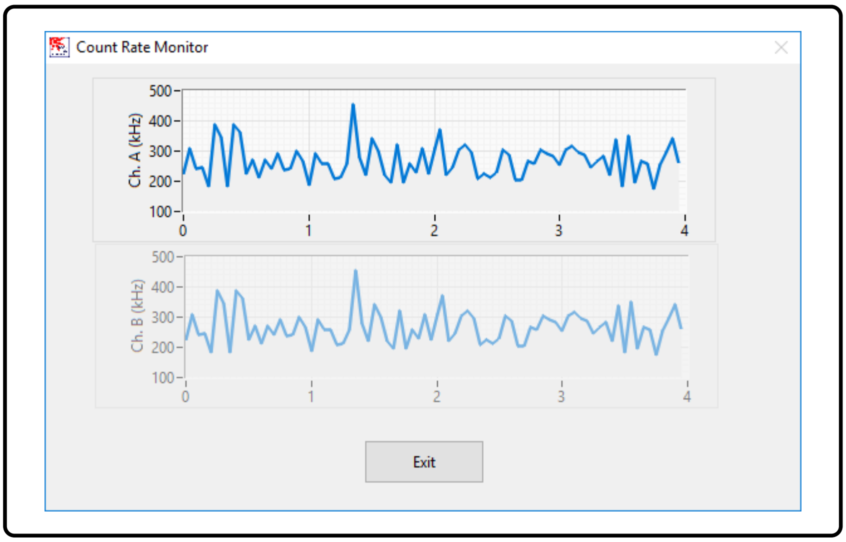


Figure 4.5: Count rate monitor window.

- *Sampling Time:* select the first lag time among 12.5, 200, 400, 800, 1600 and 3200 ns. This option allows to average the correlation function over larger sampling time compared to the default 12.5 ns in order to reduce potential noise that affects the quality of the fit.
- *Dangerous count Threshold:* select the average photon count threshold (in kHz) beyond which the correlator protects the detectors.
- *Overload Time Interval (μs)*: select the integration time interval (in microseconds) upon which the average photon count is averaged.
- *Linear Tau Correlation Type:* select auto or cross-correlation mode for the linear-τ scheme.
- *Linear Sample Time (μs):* select the sampling time (in µs) for the linear channel layout. Available sampling time values are 0.4, 0.8, 1.6, 3.2, 6.4, 12.8 and 25.6 µs.
- *Number of Linear Channels:* choose the number lag times composing the channel layout for the linear mode. Minimum and maximum allowed values are respectively 1 and 2048.
- *First Linear Channel:* choose the starting channel of the linear channel layout for the linear mode. Maximum allowed value, for a total number of linear channel m, is $2^{20} - m - 1$.
- *Linear Channel Layout:* Upload the channel layout form an external text file "filename.txt". The file must contain a column of integer numbers not necessarily consecutive, in ascending order e.g.
  1
  2
  3
  20
  100
  102
  The final lag-time layout will be given by the channel layout multiplied by the selected lag time.
- **Generate Delay Multiples:** if pressed this button creates a channel layout for the linear correlation mode. In particular it creates a layout that starts at "First linear channel" and

contains a number of consecutive channels defined by ”Number of Linear Channels”, e.g. if ”First Lag Time”=230 and ”Number of Linear Channel”=5, the resulting channel layout will be: 230, 231, 232, 233, 234.

The **Correlator Management** section shows the status of the correlator and allows to connect to the device according to the current default parameters (*Connect to Default* button) in case the software was started without connecting to the default device or configure a new device as explained above (*Change Default*). All configuration settings are stored into the ”LSI Correlator Acquisition Software Settings.ini” located in the main software folder. In the second column, the status of the correlator is shown by an green led (connected=on, disconnected=off). Another green led shows whether or not the correlator is actively measuring. If the correlator is disconnected, options which are not available remain grayed-out.

The **Measurement Settings** section is related to the control of the measurement procedure and storage of the acquired data. It also shows in real-time the result of the exponential fitting according to the user selected type of analysis.

- *Start and Stop:* The Start button runs the measurement according to the selected settings. If a measurement is composed of multiple repetitions, consecutive runs start automatically. The Stop button allows to terminate the measurement manually in any circumstance. In case of multiple repetitions the user interrupts the entire measurement by pressing this button.

- *Duration, Wait and Repetitions*

  The length of a measurement can be defined by two numbers. One is the *Duration* per repetition and the other is the number of *Repetitions* per measurement. The total duration of one measurement is thus the number of repetitions times its duration. The splitting of one measurement into several repetitions has the advantage of allowing to assess the stability of each measurement. A *Wait* parameter allows to add a time delay between the end and the beginning of two or more consecutive repetitions.

- *Save data*

  The *auto save data* option, if checked, allows automated saving of the last acquired data while the measurement is performed. The *append date to filename* option allows to concatenate the measurement time and date to the bottom of the file name according to the format *hh_mm_ss-dd_mm_yyyy* If a measurement is composed of consecutive repetitions, it will be saved entirely. Click on the folder icon and chose the folder path where you want your data to be saved and a file name. The measurement will be saved in a single file in XML format. If *Auto Save Data* is unchecked, pressing the Save button, allows in any case to save the last measurement after its completion.

- *Experiment Type* This option permits to specify the type of running experiment, by enabling the related data treatment during and after the measurement.

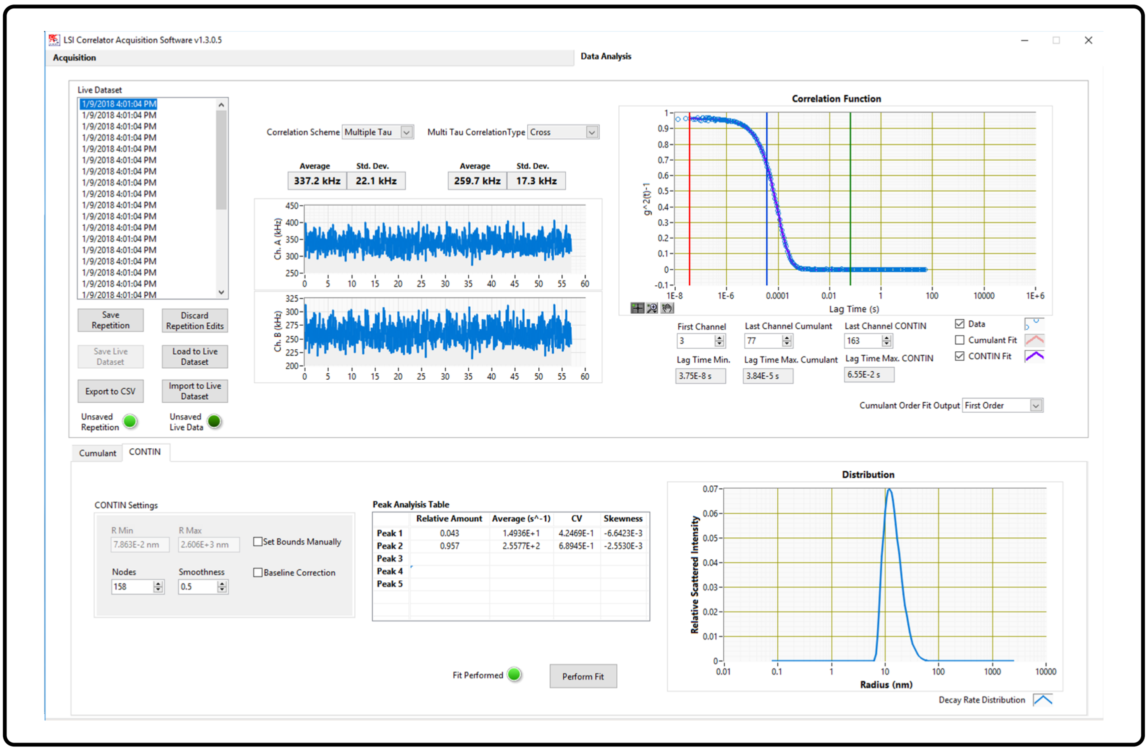
  – **Generic:** Extract pure cumulant factors (see section A, Ref. [1] and Ref. [7]) from real-time fitting without recovering any physical quantity. The associated panel *Generic* shows in time values of extracted parameter form zero to third order cumulant.

  – **DLS:** Enable DLS analysis. The above panel contains a list of sample experiment property fields (Scattering Angle, Sample Temperature, Solvent Viscosity, Solvent Refractive Index, Laser Wavelength) suitable for extracting the estimated Hydrodynamic radius, the width and the skewness of the particle size distribution according to the cumulant analysis. This option also enables the CONTIN analysis in the Data Analysis panel (see Sec. 4.4).

4.4 Data Analysis

The data analysis tab allows to visualize and perform data treatment for the last executed measurement. The software offers both cumulant and CONTIN analyses (refer to section A for a more detailed explanation of the analyses) of single DLS measurements as well as a generic cumulant analysis alone for non-DLS measurements. A theoretical overview of those analyses are given in the Appendix A. The CONTIN analysis is enabled if DLS option was previously selected in *Experiment Type* field in the "Acquired Data" panel.
The Data Analysis panel is divided into two sections. On the top are the list of acquired repetitions, the plot of the data, and other relevant additional information. The limits of the cumulant and the CONTIN analysis can be changed by dragging the corresponding vertical bars on the plot. On the bottom part, both the cumulant and CONTIN analysis are actually performed and final results are presented.

The field **Live Dataset** contains a list of the entire set of repetitions composing the last measurement. By selecting one of the items of the list, the corresponding data will be visualized in the plots on the right. Specifically, each repetition item includes the full set of data composed of the photon-trace and the correlation function. By default the cumulant fit is superimposed on the correlation function graph. By dragging the blue vertical line or by manually setting the numerical values in the corresponding fields below, the channel interval upon which the cumulant fit is performed can be changed. Accordingly, the parameters extracted from the cumulant analysis are updated in the related bottom section of the panel named *Cumulant*.



1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM
1/9/2018 4:01:04 PM

Correlation Scheme: Multiple Tau
Multi Tau CorrelationType: Cross

Average	Std. Dev.	Average	Std. Dev.
337.2 kHz	22.1 kHz	259.7 kHz	17.3 kHz

First Channel: 3
Last Channel Cumulant: 77
Last Channel CONTIN: 163
Lag Time Min.: 3.75E-8 s
Lag Time Max. Cumulant: 3.84E-5 s
Lag Time Max. CONTIN: 6.55E-2 s
☑ Data
☐ Cumulant Fit
☑ CONTIN Fit
Cumulant Order Fit Output: First Order

CONTIN Settings
R Min: 7.863E-2 nm
R Max: 2.606E+3 nm
☐ Set Bounds Manually
Nodes: 158
Smoothness: 0.5
☐ Baseline Correction

Peak Analysis Table

	Relative Amount	Average (s^-1)	CV	Skewness
Peak 1	0.043	1.4936E+1	4.2469E-1	-6.6423E-3
Peak 2	0.957	2.5577E+2	6.8945E-1	-2.5530E-3
Peak 3				
Peak 4				
Peak 5				

Figure 4.6: Data Analysis panel of the LSI Correlator software interface.

A **CONTIN** [8] analysis can be performed at the bottom section of the user interface. For this type of analysis, the type of boundary selection, and the number of nodes can be selected. Moreover, the degree of smoothness and the baseline correction can be activated to improve the quality of the result. The algorithm relies on the point scatter model which assumes the scatterers to be significantly smaller in size than the incident laser wavelength.

Appropriate parameters can be set in the CONTIN Settings pane. In particular, the number of abscissa points of the intensity distribution can be specified in the box **Nodes**. This number defines the grid of the calculation. The more points are taken for the grid the higher is the resolution of the result. Finer grids result, however, in longer calculation times.

It is also important to choose the grid size according to the quality of the measurement. If **Set Bounds Manually** is not ticked, the boundaries of the radius are defined automatically according to cumulant analysis radius. Pay attention to carefully perform a cumulant analysis by properly selecting a meaningful portion of the correlation function to be fitted, lest to risk of obtaining a wrong CONTIN fit boundary estimation. By ticking the box you might define the smallest and the largest radius or decay rates (depending on whether a DLS or Generic experiment type is set) of the simulation manually.

If the **Baseline correction** box is ticked, the CONTIN attempts to improve the fit by taking into account possible drifting of the baseline away from its zero level.

Finally, the parameter **Smoothness** allows to adjust the shape of the intensity distribution by changing the degree of smoothing of the regularized least square inverse problem as defined in [8].

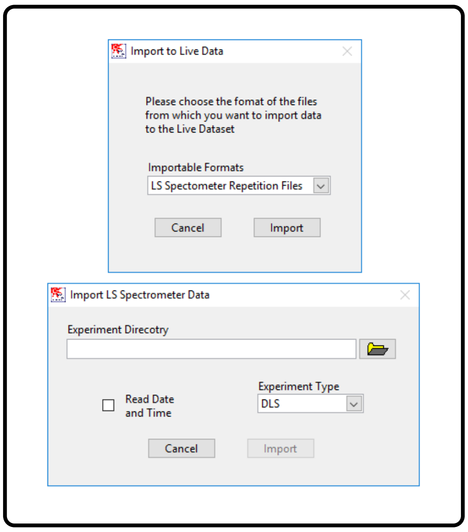
The results of the CONTIN analysis are shown in a table and a function plot. The former list the number of peaks detected with the related statistical estimated moments of the size distribution; the latter shows a plot of the computed particle size distribution for the selected repetition. Even for CONTIN, the range of the correlation function on which the analysis is performed can be chosen graphically by dragging the green vertical cursor line on the correlation function plot. In this case however the button *Perform Fit* must be pressed to activate the analysis itself each time a setting parameter is modified.

4.4.1 Data Saving and Loading

When the data acquisition is complete and the user switches to the Data Analysis tab, the acquired repetitions constituting the ”Live Dataset” are loaded into the *Live Dataset* list. A live dataset includes the raw data and the results of the cumulant analysis as obtained by the channel selection chosen in the Acquisition tab. Whenever the user either performs a CONTIN analysis or changes any parameter that result in a change of any value of either the CONTIN or the cumulant analysis the live repetition data can be either saved to the Live Dataset by pressing the **Save Repetition** button or discarded by pressing the **Discard Repetition Edits** button. A green led indicates whether the last data treatment result has been stored or not. For instance, if the user drags the blue vertical line on the correlation function plot, the led *Unsaved Repetition* will appear ”on” since a new cumulant fit has been computed over a new time range. If after an analysis the user select another repetition item from the list, a message will appear in case data analysis are not already saved.

The overall live dataset, which includes both numerical and post-processing analysis data, is overwritten via **Save Live Dataset** button onto the XML formatted text file saved after the acquisition or uploaded. Previously saved data files can be uploaded and visualized through **Load to Live Dataset**. Every time a repetition is edited and the corresponding edits are stored according to the procedure detailed above, the led called ”Unsaved Live Data” will indicate the need to save the dataset. Data are stored in XML formatted text files with *.lsicorr* extension.

Data files derived from DLS measurement performed through a *LS Spectrometer* (*LS*



Import to Live Data

Please choose the fomat of the files from which you want to import data to the Live Dataset

Importable Formats: LS Spectometer Repetition Files

Import LS Spectrometer Data

Experiment Direcotry:

□ Read Date and Time

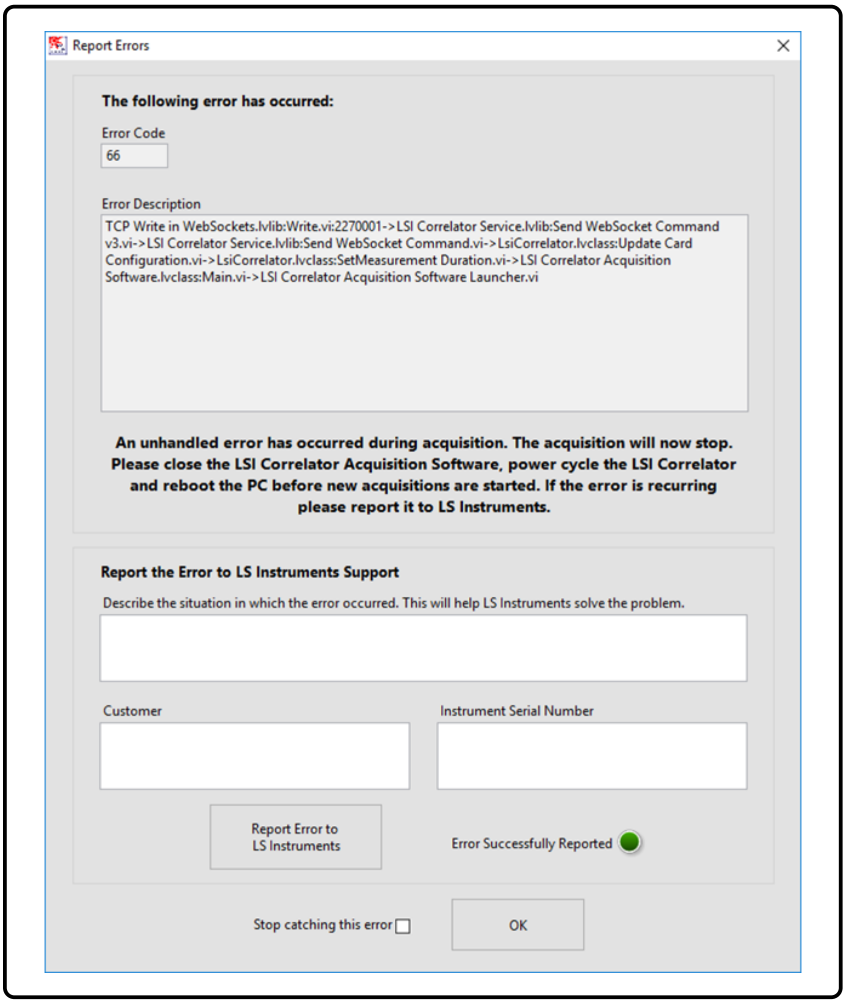
Experiment Type: DLS

Figure 4.7: Import dialogue pop-up windows for the Import to Live Dataset button.

Instruments AG) or formatted accordingly may also be uploaded. Specifically, the button **Import to Live Dataset** allows to select an entire experiment directory as shown in Fig. 4.7 and create an Live Dataset into the *LSI Correlator Software* for further analysis.
Finally the **Export to CSV** button allows data to be saved in simple comma separated spreadsheet format. In particular, a Live Dataset is stored into a user-selected repository folder containing a measurements summary (*Summary.csv*) and a subset of folders, one per each repetition. Each folder, named and numbered after the experiment, contains respectively the files *Raw Data.csv*, *Cumulant Result.csv* and, if CONTIN analysis has been performed, *Contin Result.csv*. All the CSV files can be opened in Excel, if the file extension .csv is associated to Excel (which is the most common situation) just double clicking the CSV file will open it in Excel.

4.5 Error Handling

Software errors and exceptions are caught through the error window shown in Fig. 4.8 displaying automatically when an unexpected event occurs during both acquisition or data analysis. Although the *LSI Correlator Acquisition Software* is continuously upgraded and optimized to avoid errors, unexpected behavior may rarely happen. The error report pop-up window is conceived to help us to improve the *LSI Correlator Acquisition Software* through the user feedback. As shown in Fig. 4.8 an exception is composed of an error code and a related description. When an error occurs, the user is warmly invited to provide a detailed description of the conditions or events that led to the unexpected fault, in the apposite field. Additional valuable information such as the user name and the device serial number can be inserted in the related text box. The button **Report Error to LS Instruments**, allows to immediately contact us and report the full error description. Finally, recurring warnings or mild errors that do not alter the normal execution of the software can be untracked by checking the **Stop catching this error box** and pressing the **OK** button to close the error window.



Report Errors

The following error has occurred:

Error Code
66

Error Description
TCP Write in WebSockets.lvlib:Write.vi:2270001->LSI Correlator Service.lvlib:Send WebSocket Command v3.vi->LSI Correlator Service.lvlib:Send WebSocket Command.vi->LsiCorrelator.lvclass:Update Card Configuration.vi->LsiCorrelator.lvclass:SetMeasurement Duration.vi->LSI Correlator Acquisition Software.lvclass:Main.vi->LSI Correlator Acquisition Software Launcher.vi

An unhandled error has occurred during acquisition. The acquisition will now stop. Please close the LSI Correlator Acquisition Software, power cycle the LSI Correlator and reboot the PC before new acquisitions are started. If the error is recurring please report it to LS Instruments.

Report the Error to LS Instruments Support

Describe the situation in which the error occurred. This will help LS Instruments solve the problem.

Customer

Instrument Serial Number

Report Error to LS Instruments

Error Successfully Reported ●

Stop catching this error □

OK

Figure 4.8: Error pop-up window.

5 MATLAB API

The LSI Correlator is designed for full stand-alone usage. Data processed by the correlator are transmitted via Ethernet or Ethernet adapted USB cabling via standard WebSocket protocol [9]. This allows the user to retrieve both photon electron traces and computed correlation function for custom data treatment.

This section contains information for advanced users in relation to custom usage of the LSI Correlator. Specifically it includes a comprehensive documentation for the LSI Correlator's MATLAB Application Programming Interface (API). The latter is based on an underlying Java implementation of WebSocket protocol and is especially suitable for Event Object Oriented programming model [10]. The MATLAB API has been tested in the following MATLAB versions

- R2015a with Java 1.7.0 60-b19
- R2015b with Java 1.7.0 60-b19
- R2016b (64 bits) with Java 1.7.0 60-b19.

In case of problems in making the MATLAB API work with other MATLAB and Java versions, the user can contact LS Instruments support.

5.1 Function Prototypes

- `LSI Correlator obj = LSICorrelator(correlator_IP, Version):` Open a Websocket connection with the *LSI Correlator* toward IP address $correlator\_IP$ and instantiate a correlator object obj through which parameters and commands can be accessed. Version is a string of the current firmware version (the latter can be found by starting and configuring the *LSI Correlator* Software at the left bottom of the main panel as shown in Fig. 4.4)

- `obj.configure():` configure the correlator once a connection has been established. This command transfers the entire set of parameters to the correlator. It must be called every time a parameter has been changed in order to set the correlator accordingly.

- `obj.start():` start a measurement. This command opens a new parallel WebSocket connection toward the correlator and runs a background process that receives data streams from the correlator.

- `obj.update(figure_handle_h):` during an active measurement, update the object properties $obj.TraceChA$, $obj.TraceChB$ and $obj.Correlation$ described below, with the last value received by the correlator. If a MATLAB figure handle is passed to this method, a new figure, plotting the total count rate traces and the last updated correlation function, opens automatically.

- `bool on = obj.MeasurementOn():` check whether or not a measurement is active.

- `obj.stop():` stop the current measurement process. If the user has previously set a measurement time duration, this command is unnecessary as long as the timer is expired.

- `obj.close():` close the correlator connection and destroy the $LSICorrelatorobj$.

5.2 Settings

Correlator's configuration parameters are accessible and modifiable through standard get and set methods i.e. `obj.get("Param_Name")$ or $obj.set("Param_Name")`. When a correlator parameter is changed through the set method, a command `obj.configure()` must be called whenever no measurement is active, in order to effectively update the internal device settings. Some parameter values are defined as `LSICorrelator` object's attribute and accessible upon importing LSI library within the user's script e.g. `LSI.Param.ParanName.ParamValue`. A detailed explanation of the accessible setting parameters is given below:

- *CorrelationType*: select auto or cross correlation.

  Values: $AUTO$, $CROSS$.

  Default: $AUTO$.

  Example: `obj.set("CorrelationType", LSI.Param.CorrelationType.AUTO).`

- *SamplingTimeMultiT:* select whether the correlation function is computed over the full lag time range starting at 12.5 ns ($NORMAL$) or an select an initial lag time among 200, 400, 800, 1600 and 3200 ns. ($AVERAGED$) according to the parameter $AveragingPeriod$.

  Values: $ns12\_5$, $ns200$, $ns400$, $ns800$, $ns1600$, $ns3600$.

  Default: $ns12\_5$.

  Example: `obj.set("SamplingTimeMultiT", LSI.Param.Mode.Normal).`

- *SwapChannels:* If $Correlationtype$ is set to $AUTO$, select the input channel upon which the auto-correlation function is calculated (ChA_ChB for channel A, ChB_ChA for channel B). If $Correlationtype$ is set to $CROSS$, change the order of the input channel operands (ChA_ChB computes ChA $\otimes$ ChB, ChB_ChA computes ChB $\otimes$ ChA).

  Values: ChA_ChB, ChB_ChA.

  Default: ChA_ChB.

  Example: `obj.set("SwapChannels", LSI.Param.SwapChannels.ChA_ChB) .`

- *Normalization:* Choose between Symmetric and Compensated normalization (see section 2.2.3 in chapter 2).

  Values: $SYMMETRIC$, $COMPENSATED$.

  Default: $COMPENSATED$.

  Example: `obj.set("Normalization",LSI.Param.Normalization.COMPENSATED).`

- *MeasurementDuration*: select the duration of the measurement.

  If $MeasurementDuration = 0$ the duration is undetermined and the running measurement must be stopped through the stop command $obj.stop()$.

  Values: Positive real number (double representation).

  Default: 0.

  Example: `obj.set("MeasurementDuration",0).`

- *TransferRate:* select the transfer time interval (in milliseconds) between two data packets delivered by the correlator through via the Ethernet. The number of samples processed for each packet is proportional to the selected transfer rate.

  Values: $ms100$, $ms150$, $ms200$, $ms250$, $ms300$, $ms400$,$ms500$,$ms600$,$ms700$.

  Default: $ms500$.

  Example: `obj.set("TransferRate",LSI.Param.TransferRate.ns200).`

- *OverloadLimit:* Select the average photon count threshold (in Mcps) beyond which the correlator shuts off of the APDs.

  Values: Positive real number (double representation).

  Default: 20 (Mcps).

  Example: `obj.set("OverloadLimit",20) .`

- *OverloadTimeInterval*: Select the integration time interval (in microseconds) upon which the average photon count is integrated.

  Values: Positive real number (double representation).

  Default: 400 (us).

  Example: `obj.set("OverloadTimeInterval",400).`

- *OutputTriggerDelay*: Choose the delay time (in milliseconds) between the start driven by the user and the actual data acquisition start.

  Values: Positive real number (double representation).

  Default: 0 (ms).

  Example: `obj.set("OutputTriggerDelay",20).`

5.3 Output Variables

LSI Correlator object, contains few attribute arrays that allow the user to read and process data acquired during measurements. Before accessing an updated array during a measurement, `obj.update()` must be called. Output variables are accessible through calling `ob.OutputVar`.

- **TraceChA**: contains the photon count trace for channel A over a sampling time of 52.4288 ms. After each update, the current trace is given by the total photon count stream received. Hence the size of TraceChA grows continuously as long as the measurement runs.

- **TraceChB**: contains the photon count trace for channel B with the same specifications ad for channel A.

- **Lags**: contains the lag time array.

- **Correlation**: contains the last computed value of the normalized correlation function, according to the user selected settings.

5.4 Installation and Example

The Matlab API folder is located into the LS Correlator software installation folder. For example in Windows 7 the full path for the Matlab API folder is `C:\Program Files (x86)\LS Instruments\LSI Correlator\LSI MATLAB API`.
To install the library, move the file `javaclasspath.txt`, stored into the Matlab API folder, to your Matlab preferences folder. To view the location of the preferences folder, type: $prefdir$ on the Matlab command prompt. Finally restart Matlab.
Any custom script calling the Correlator Matlab API's methods and classes must link the API. Specifically the following two lines must be added on the top of the script:

```
path(path,'YOUR PATH\LS Instruments\LSI Correlator\LSI MATLAB API')
import ('YOUR PATH\LS Instruments\LSI Correlator\LSI MATLAB API\LSI.*')
```

The next simple code segment shows a small script that creates a `LSICorrelator` object and runs 3 measurements of 5 seconds each.

```
%% SETUP LSI CORRELATOR OBJECT WITH DESIRED PARAMETERS
correlator_ip = '192.168.2.161';


obj = LSICorrelator(correlator_ip, '3.0.0.1');


% obj.get('CorrelationType')
% obj.get('Mode')
% obj.get('SwapChannels')
% obj.get('Normalization')
% obj.get('MeasurementDuration')
% obj.get('TransferRate')
% obj.get('OverloadLimit')
% obj.get('OverloadTimeInterval')


obj.set('SamplingTimeMultiT',LSI.Param.SamplingTimeMultiT.ns200)
obj.set('CorrelationType',LSI.Param.CorrelationType.AUTO)
obj.set('Normalization',LSI.Param.Normalization.COMPENSATED)
obj.set('SwapChannels',LSI.Param.SwapChannels.ChA_ChB)
obj.set('MeasurementDuration',5)
obj.set('TransferRate',LSI.Param.TransferRate.ms300)

obj.set('OverloadLimit',500)
obj.set('OverloadTimeInterval',500)


obj.set('OutputTriggerDelay',9)

%% SEND CONFIGURATION TO LSI CORRELATOR

obj.configure()

pause(0.05)

%% SETUP SOME MEASUREMENTS AND MONITOR THEM
totalCountRate = 0;
```

```
h = figure(1);

% perform three measurements of 5 sec each
nrun = 3;

for i=1:nrun

obj.start()% start measurement

while (obj.MeasurementOn())

pause(0.5) % pause between two updates if the MATLAB process
% is too fast compared to the
% tranfer speed set through "TransferRate"
obj.update(h);% update the LSI CORRELATOR object properties

totalCountRate = totalCountRate + sum(obj.TraceChA)/ length(obj.TraceChA);% calculate t

end

disp(['Iteration = ', num2str(i)]);

end


%update last data packet
obj.update();
obj.TraceChA;

% totalCountRate=sum(obj.TraceChA)/ length(obj.TraceChA)

%% CLOSE CONNECTION TO THE LSI CORRELATOR
obj.close()

%% CLEAR THE LSI CORRELATOR OBJECT
clear obj
```

5.5 XML Conversion and Import

Data file produced by the *LSI Correlator* software are stored in XML format. To import and parse these file in Matlab, additional script are provided. The function `LSICorr_XMT_to_Matlab(filename)`, included into the *LSI Correlator Acquisition Software* folder, produces an object containing all data and info of the saved experiment organized into Matlab cell structures. Fields are intuitively accessible via automatic completion, e.g. extraction of the correlation function array for the first repetition of a saved measurement would read as follow (the example holds also if the experiment is composed of one repetition only):

```
obj=LSICorr_XML_to_Matlab('C:\data.lsicorr')
obj.Raw_Data{1}.Correlation_Function
```

6 LabVIEW API

This section contains instructions and documentation of the LabVIEW API. This library allows to develop a custom software interface for connecting to the *LSI Correlator* and retrieving data from the device.

6.1 VIs documentation

The following VIs constitute the interface for using the *LSI Correlator*. They must be composed sequentially as shown in Fig.6.2 in order to connect, control and terminate the connection with the device. All the following listed VIs can be located within the library folders by opening them through the included example VIs and by selecting *View − > This VI in Project* on their respective Front Panels.

- *LsiCorrelator_Factory.vi:* Constructor.

  Connect to *LsiCorrelator*, start a connection and create an LSI Correlator abstract object.

  If no control is connected to the "Configuration File" input, this VI scans the network for retrieving the *LsiCorrelator*, starts a connection and create an LSI Correlator abstract object. When this VI is called, two additional panels appear. If the LS Correlator is properly connected, the upper left field "IP Address" on the second panel displays the address of the IP correlator. The right side of the panel displays info about the LS Correlator. For configuring the correlator via configuration panels, refer to section 4.2.

  When a "Configuration File" control is connected, the initial configuration is imported from the ini file (see section 6.3) accessible via the selector folder control shown in Fig. 6.1. In this case the network is not scanned since the IP address is explicitly recorded into the ini file. A wrong IP address produces an error. The *LsiCorrelator* ini file **LSI Correlator Acquisition Software Settings.ini** is stored into the installation folder. To activate the initialization form file, the "Configure File" control must be configured as shown in Fig. 6.1, where "Config File Path" is the location of the ini file and "Write on Destroy" allows to write the ini file after the program has exited, in case some configuration parameters have been changed.

  Inputs: Correlator Reference, Error, Configuration File.

  Output: Correlator Reference, Error.

- *Start.vi:* Start a measurement. Establish a new data connection with the *LsiCorrelator* and initialize the device according to the configure properties.

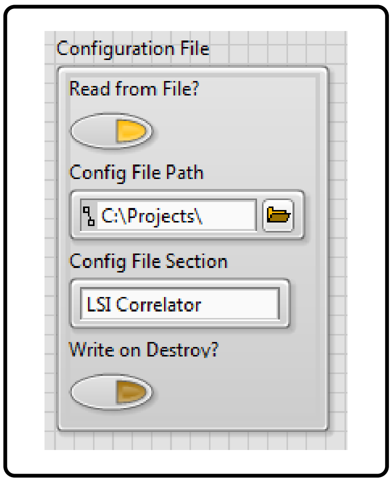
  Inputs: Correlator Reference, Error.

  Output: Correlator Reference, Error.

- *Grab.vi:* Retrieve data from the *LsiCorrelator*. This VI should be inserted into a loop during the duration of the measurement. When a new data transmission is started, a background process handles the communication and waits for the data packet transmitted by the correlator with a selected transfer rate. The Grab rate should be larger than the transfer rate in order not to retrieve the same data packet more the once. Therefore the loop should be slowed down by a wait VI. The otput parameter "Correlation Data" is a cluster containing the data produced by the device. "Grab Time" indicates the actual measurement time. The loop can be stopped though the "Measurement Complete" boolean property.

  Inputs: Correlator Reference, Error.

  Output: Correlator Reference, Error, Correlation Data, Grab Time, Measurement Complete.



Configuration File

Read from File?

Config File Path

C:\Projects\

Config File Section

LSI Correlator

Write on Destroy?

Figure 6.1: Example of "Configuration File" control

- *Stop.vi:* Stop the running measurement. If "measurement duration" has been set previously ("measurement duration" must be set equal to zero for unlimited duration) this VI is unnecessary. If the measurement is stopped manually this VI closes the data connection.

  Inputs: Correlator Reference, Error.

  Output: Correlator Reference, Error.

- *Destroy.vi:* Destroy the *LsiCorrelator* object and release the connection.

  Inputs: Correlator Reference, Error.

  Output: Correlator Reference, Error.

6.2 Properties

Configurable properties are accessible via Property Node as shown in Fig.6.2: The following properties are available (additional available properties not listed below, should not be accessed for a correct usage of the *LSI Correlator*):

- *New Multi Tau API:* Activate MultiTau features for LSI Correlator API . Input parameter "New Multi Tau API" must be set to TRUE.

  Values: Boolean.

  Default: True.

- *Correlation Scheme*: select multiple-tau or linear tau scheme.

  Values: $LinearTau$, $MultipleTau$.

  Default: $MultipleTau$.

- *Multi Tau Correlation Type*: select cross or auto correlation for the Multi-Tau.

  Values: $Auto$, $Cross$.

  Default: $Auto$.

- *Linear Tau Correlation Type*: select cross or auto correlation for the Linear-Tau scheme.

  Values: *Auto*, *Cross*.

  Default: *Auto*.

- *Sampling Time:* select whether the correlation function is computed over the full lag time range starting at 12.5 ns or the initial lag-time is changed according to the available parameter values.

  Values: 12.5 ns, 200 ns, 400 ns, 800 ns, 1600 ns, 3200 ns.

  Default: 12.5 ns.

- *Normalization:* Choose between Symmetric and Compensated normalization (see section 2.2.3 in chapter 2).

  Values: *Symmetric*, *Compensated*.

  Default: *Compensated*.

- *Measurement Duration*: select the duration of the measurement.

  If Measurement Duration=0 the duration is undetermined and the running measurement must be stopped through manual stop.

  Values: Positive real number (double representation).

  Default: 0.

- *TransferRate:* select the transfer time interval (in milliseconds) between two data packets delivered by the correlator through the Ethernet. The number of samples processed for each packet is proportional to the selected transfer rate.

  Values: 100 ms, 150 ms, 200 ms, 250 ms, 300 ms, 400 ms, 500 ms, 600 ms, 700 ms.

  Default: 500 ms.

- *Overload Time Interval (µs):* Select the average photon count threshold (in Mcps) beyond which the correlator shuts off of the APDs.

  Values: Positive real number (double representation).

  Default: 10 (Mcps).

- *Overload Time Interval*: Select the integration time interval (in microseconds) upon which the average photon count is integrated.

  Values: Positive real number (double representation).

  Default: 400 (us).

- *Linear Tau Sampling Time (µs):* select the sampling time (in µs) for the linear channel layout. Available sampling time values are 0.4, 0.8, 1.6, 3.2, 6.4, 12.8 and 25.6 µs.

  Values: Positive real number (double representation).

Default: 3.2 µs

- *Delay Time Multiple:* Create a new channel Layout. Channels can be non-consecutive. (Ex: 0,1,2, 201, 202, 203...) Minimum and maximum allowed number of linear channels are 1 and 2048. Minimum and maximum allowed values, for a total number of linear channel m, are respectively 0 and $2^{20} - m - 1$.

  Values: positive int array (integer representation).

  Default: empty array.

6.3 Init File

The initialization file **LSI Correlator Acquisition Software Settings.ini** is located within the LabVIEW API folder. It the device is connected properly and the ini file is configured with the correct IP address, the manual configuration procedure is skipped and the software starts automatically without scanning the network. An example of ini file is shown below. Although the manual editing of the ini file is deprecated, an experienced user can manually change the value of each configuration parameter below, unless explicitly written. Uncommented lines are self explanatory.

```
[LSI Correlator]

# IP address of the correlator
IP Address = "169.254.6.84"

# "Symmetric", "Compensated"
Normalization = "Symmetric"

# "TRUE","FALSE"
Swap Channels = "TRUE"

# Timeout on communication with card, increase for slow connections or to minimize
# risks of conncetion losses upon disconnection (do not change)
Stream Data Read Timeout (s) = "10"

# Default setting (do not change)
Slow Modulation Period = "400 ns"

# Default setting (do not change)
Fast Modulation Period = "200 ns"

# Exponent that dictates the size of the photon buffer (do not change)
Linear Trace Size Exponent = "20"

# Time interval over which the count rate is calculated and
# then compared to the dangerous count threshold to swithc off detectors
Overload Time Interval (us) = "400"

# Default setting (do not change)
Modulation Settings.Flex Position Ch. A (ns) = "0.000000"

# Default setting (do not change)
Modulation Settings.Flex Position Ch. B (ns) = "0.000000"

```

```
# Default setting (do not change)
Modulation Settings.Fitted Rise Time (ns) = "0.000000"

# Default setting (do not change)
Modulation Settings.Cutoff Factor = "0.000000"

# Default setting (do not change)
Modulation Settings.Ascending Edge Channel = "Ch. A"

# Default setting (do not change)
Modulation Settings.AOM B Clock Offset (ns) = "0"

# specifies the type of correlator (do not change)
Correlator Type = "LsiCorrelator"

# Multiple Tau, Linear Tau
Correlation Scheme = "Linear Tau"

# Auto, Cross.
Multiple Tau Correlation Type = "Auto"

# Auto, Cross
Linear Tau Correlation Type = "Auto"

# Count Threshoold over which count rate overload is detected
Dangerous Count Threshold (kHz) = "10000.000000"

# Select the first discarded lag-times for the Auto Correlation running mode
First Ch. Auto Mode = "3"

# sampling time in linear mode
Linear Tau Sample Time (us) = "12.000000"

# Exponent that dictates the size of the photon buffer (don't change)
Linear Trace Size Exponent = "20"

# Time interval over which the count rate is calculated and
# then compared to the dangerous count threshold to swithc off detectors
Overload Time Interval (us) = "400"

# Version of the init file syntax (don't change)
ini Version = 3

# For the mode we can choose "Disabled", "Output", "Input Both Edges",
# "Input Pulse"
Triggering Settings.Mode = "Disabled"

# If we set the Mode to "Output" this is the delay in milliseconds that has to
# elapse form the moment the output trigger has been given
# until the correlation function acquisition starts
Triggering Settings.Output Delay (ms) = "0"

# Time in seconds we wait for the correlator to boot up
Timeout (s) = 30
```

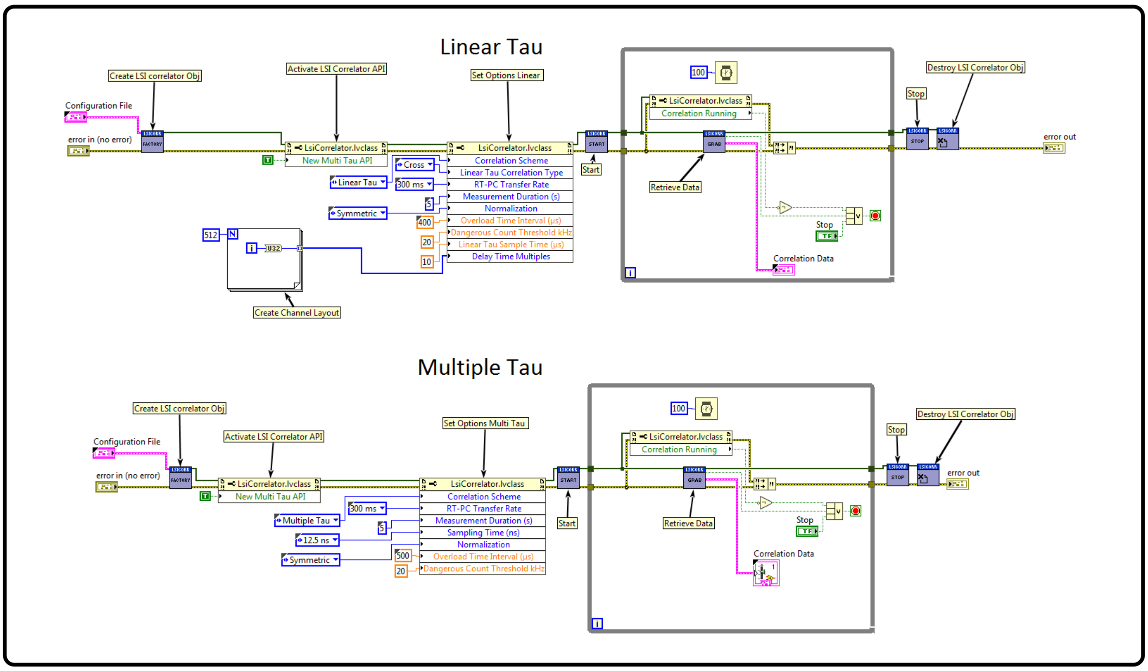


Figure 6.2: Example block diagrams for the LSI Correlator LABVIEW API.

6.4 Installation and Example

The LabView API is stored into the *Lsi Correlator* installation folder (path `C:\Program Files (x86)\LS Instruments\LSI Correlator\LABVIEW API`). For installing it, create a new project and import the library or simply drag the library into the project. The API is fully compatible with Labview 2016 and requires the installation of the following add-ons:

- G# Framework v1.6.1.4
- OpenG Error Library v4.2.0.23
- OpenG LabVIEW Data Library v4.2.0.21
- OpenG String Library v4.1.0.12
- OpenG Variant Configuration File Library v4.0.0.5
- GPower Numeric v2016.2.0.14

LabView Add-ons can be retrieved and automatically installed though the software VI Package Manager available online.
Fig. 6.2 shows two block diagrams for a measurement of five seconds in Linear mode (above) and MultiTau Mode (bottom). Both example VIs are included into the LABVIEW API.

7 Python API

This section contains information for advanced users in relation to OEM usage and custom integration of the *LSI Correlator*. Data produced by the *LSI Correlator* in particular, can be retrieved through the Python API described below. This allows the user to manipulate both photon electron traces and correlation function and build a custom software that integrates the device.
The API is based on the Autobahn Python Implementation of the WebSocket protocol [11] and is especially suitable for Event Object Oriented programming model. The latter is provided by the underlying event-driven networking engine Twisted [12]. The Python API has been compiled with the following Python versions

- Python 2.7.10 32bit (Autobahn 17.6.2 and Twisted 17.5.0)
- Python 3.6.1 32bit (Autobahn 17.7.1 and Twisted 17.5.0)

In case of issues in running the Python API with other Python versions, the user can contact LS Instruments support.

7.1 Function Prototypes

- LSI Correlator obj = LSICorrelator(correlator_IP, Version): Open a Websocket connection with the *LSI Correlator* toward IP address $correlator\_IP$ and instantiate a correlator object obj through which parameters and commands can be accessed. Version is a string of the current firmware version (the latter can be found by starting and configuring the *LSI Correlator* Software at the left bottom of the main panel as shown in Fig. 4.4)
- obj.configure(): configure the correlator once a connection has been established. This command transfers the entire set of parameters to the correlator. It must be called every time a parameter has been changed in order to set the correlator accordingly.
- obj.start(): start a measurement. This command opens a new parallel WebSocket connection toward the correlator and runs a background process that receives data streams from the correlator.
- obj.update(): during an active measurement, update the object properties $obj.TraceChA$, $obj.TraceChB$ and $obj.Correlation$ described below, with the last value received by the correlator.
- bool on = obj.MeasurementOn(): check whether or not a measurement is active.
- obj.stop(): stop the current measurement process. If the user has previously set a measurement time duration, this command is unnecessary as long as the timer is expired.
- obj.close(): close the command correlator connection established by the $LSICorrelator$ object.

7.2 Settings

Correlator's configuration parameters are modifiable through related set methods. When a correlator parameter is changed through the set method, a command `obj.configure()` must be called whenever no measurement is active, in order to effectively update the internal device settings. Some parameter values are defined as `LSICorrelator` object's attribute and accessible upon importing LSI library within the user's script e.g. `import LSI.LSI_Param as pr` and `pr.ParanName.ParamValue`. A detailed explanation of the accessible setting parameters is given below:

- *CorrelationType*: select auto or cross correlation.

  Values: $AUTO$, $CROSS$.

  Default: $AUTO$.

  Example: obj.setCorrelationType(pr.CorrelationType.AUTO).

- *SamplingTimeMultiT:* select whether the correlation function is computed over the full lag time range starting at 12.5 ns or an select an initial lag time among 200, 400, 800, 1600 and 3200 ns.

  Values: $ns12\_5$, $ns200$, $ns400$, $ns800$, $ns1600$, $ns3600$.

  Default: $ns12\_5$.

  Example: obj.setSamplingTimeMultiT(pr.SamplingTimeMultiT.ns12_5).

- *SwapChannels:* If $Correlationtype$ is set to $AUTO$, select the input channel upon which the auto-correlation function is calculated (ChA_ChB for channel A, ChB_ChA for channel B). If $Correlationtype$ is set to $CROSS$, change the order of the input channel operands (ChA_ChB computes ChA $\otimes$ ChB, ChB_ChA computes ChB $\otimes$ ChA).

  Values: ChA_ChB, ChB_ChA.

  Default: ChA_ChB.

  Example: obj.setSwapChannels(pr.SwapChannels.ChA_ChB) .

- *Normalization:* Choose between Symmetric and Compensated normalization (see section 2.2.3 in chapter 2).

  Values: $SYMMETRIC$, $COMPENSATED$.

  Default: $COMPENSATED$.

  Example: obj.setNormalization(pr.Normalization.SYMMETRIC).

- *MeasurementDuration*: select the duration of the measurement.

  If $MeasurementDuration = 0$ the duration is undetermined and the running measurement must be stopped through the stop command $obj.stop()$.

  Values: Positive real number (integer representation).

  Default: 5.

  Example: obj.setMeasurementDuration(5) .

- *TransferRate:* select the transfer time interval (in milliseconds) between two data packets delivered by the correlator through via the Ethernet. The number of samples processed for each packet is proportional to the selected transfer rate.

  Values: $ms100$, $ms150$, $ms200$, $ms250$, $ms300$, $ms400$,$ms500$,$ms600$,$ms700$.

  Default: $ms500$.

  Example: obj.setTransferRate(pr.TransferRate.ms150).

- *OverloadLimit:* Select the average photon count threshold (in Mcps) beyond which the correlator shuts off of the APDs.

  Values: Positive real number (double representation).

  Default: 20 (Mcps).

  Example: obj.setOverloadLimit(20) .

- *OverloadTimeInterval*: Select the integration time interval (in microseconds) upon which the average photon count is integrated.

  Values: Positive real number (double representation).

  Default: 400 (us).

  Example: obj.setOverloadTimeInterval(450).

- *OutputTriggerDelay*: Choose the delay time (in milliseconds) between the start driven by the user and the actual data acquisition start. This attribute is available with firmware version $3.0.0.0$ or later

  Values: Positive real number (double representation).

  Default: 0 (ms).

  Example: obj.setOutputTriggerDelay(10).

7.3 Output Variables

LSI Correlator object, contains few attribute arrays that allow the user to read and process data acquired during the measurements. Before accessing an updated array during a measurement, `obj.update()` must be called. Output variables are accessible through calling `obj.OutputVar`.

- **TraceChA**: contains the photon count trace for channel A over a sampling time of 52.4288 ms. After each update, the current trace is given by the total photon count stream received. Hence the size of TraceChA grows continuously as long as the measurement runs.

- **TraceChB**: contains the photon count trace for channel B with the same specifications ad for channel A.

- **Lags**: contains the lag time array.

- **Correlation**: contains the last computed value of the normalized correlation function, according to the user selected settings.

7.4 Installation and Example

The Python API folder is located into the LS Correlator software installation folder. For example in Windows 7 the full path for the Python API folder is `C:\Program Files (x86) \LS Instruments\LSI Correlator\LSI Python API`.
Any custom script calling the Correlator Python API's methods and classes must link the API. Specifically the following two lines must be added on the top of the script:

```
import sys
sys.path.append('YOUR PATH\\LS Instruments\\LSI Correlator\\LSI Python API')
```

The next simple code segment shows a small script that creates a `LSICorrelator` object and runs 3 measurements of 5 seconds each.

```python


import LSI.LSI_Param as pr
from  LSICorrelator import LSICorrelator
import time
import matplotlib.pyplot as plt
from setPlot import setPlot
import numpy as np

# connect to LSI correlator
obj=LSICorrelator("192.168.2.171","3.0.0.1")

# example of plot setting: setPlot.py is included into the Python API folder
pl_obj=setPlot()

# set Configuration Parameters
obj.setCorrelationType(pr.CorrelationType.AUTO)
obj.setNormalization(pr.Normalization.SYMMETRIC)
obj.setMeasurementDuration(5)
obj.setSwapChannels(pr.SwapChannels.ChA_ChB)
obj.setSamplingTimeMultiT(pr.SamplingTimeMultiT.ns12_5)
obj.setTransferRate(pr.TransferRate.ms150)
obj.setOverloadLimit(20)
obj.setOverloadTimeInterval(450)

deltat=0.0524288;

# configure LSI Correlator
obj.configure()


# run measurements
for i in range(3):
    print "Measurement "+str(i+1)
    obj.start()

    while obj.MeasurementOn():

        time.sleep(0.5)
        obj.update()

        timeTr=np.arange(len(obj.TraceChA))*deltat

        # normalize the count trace Arrays
        TrA= np.asarray(obj.TraceChA)/(1000*deltat)
        TrB= np.asarray(obj.TraceChB)/(1000*deltat)
        LagTimes=np.asarray(obj.Lags)
        Correlation =np.asarray(obj.Correlation)


        pl_obj.myplot(timeTr, TrA , TrB, LagTimes, Correlation, 2)


```

```
# End connection
obj.close()
```

8 F.A.Q.

- **Is the LSI Correlator compatible with any single photon detector delivering a TTL signal?**: The LSI correlator is compatible with basically any single photon detector delivering a TTL or LVTTL signals. Moreover, it is able to detect photo-electron pulses of width down to 2.5 ns plus dead time of about 10 ns. These requirements are met by most of the currently available photo-electron transducers on the market such as Execelitas APDs and Hamamatsu PMT modules.

- **What is the minimum screen resolution for displaying the LSI acquisition software interface?**: In our specifications we state that the minimum resolution for our software is 1680x1050. Most relatively recent computers normally support resolutions higher than this.

- **Can an analog signal be input into the correlator?**: The LSI Correlator is a digital correlator hence it only supports TTL/LVTTL logic signals. Analog signals, if the detector is fast enough, must to be analyzed by a spectrum analyzer or converted into digital signal by a discriminator circuit.

- **Can the software be used for device control and data analysis for a FCS setup?**: The software, as well as the hardware correlator, can be used for both DLS and FCS measurements. Data treatment of the normalized correlation function is meant for DLS data analysis. Specifically, it includes both cumulant analysis and the more advanced regularized CONTIN algorithm for computing the correlation function inverse real Laplace transform suitable to obtain the full particle size distribution.

- **Is the fastest lag time programmable only through the available APIs?**: The multi-tau layout is programmable through both the acquisition software and the APIs. In multi-τ mode the fastest lag time can be set to 12.5 ns in both cases. On the other hand the Linear Correlator is at the moment only available through the software interface.

- **Is there a way to manually set the IP address of the LSI correlator?**: The correlator IP cannot be set manually: it is assigned either via Ethernet DHCP protocol if the device is connected to the network or link locally by the Ethernet-USB adapter if it is connected to the USB port. On the other hand, the IP address needs to be entered manually to allow the MATLAB or Python WebSocket client to connect. The address can be retrieved by opening the LSI acquisition software via the starting configuration panel. Alternatively, once the software has been set up for the first time and assuming the connection mode has not changed, the address can be retrieved from the ini file ”LSI Correlator Acquisition Software Settings” stored into installation folder

- **Is it possible to save the un-normalized correlation function?** The un-normalized correlation function cannot be saved. The advanced normalizations implemented in the correlator makes use of a sequence of counters, one per each correlator channel. We chose to normalize the CF within the board to prevent overload in the data transmission and to ensure a true stand-alone operation.

- **By looking at the auto-correlation function in multi-τ mode I find that: 1) the very first value is much larger than 1 and 2) the second value is negative and close to -1. Is there any issue?** 1) The behavior for the ACF is typical and well expected. Specifically, the first peak is the usual delta peak at zero lag, peculiar of every stochastic noise, and whose height is the normalized variance of the signal itself. 2) The negative point is the mark of the APD dead time after the emission of a pulse. This distortion is usually visible at lags of about $15 - 20$ ns.

A couple of ways to circumvent the issue could be:

1. Use two detectors/APDs for the same experiment and compute the Cross CF. This is what is commonly known as pseudo-cross-correlation. This function could show an intercept a bit lower than the ACF, but it is not affected by both dead-time and after-pulse effects since they are uncorrelated between two different signals.
2. Use the correlator feature that allows to choose a different (larger) initial lag time. This is what the related feature is meant for. Setting an initial lag time of 200 ns would average out all the details of the ACF within 200 ns lag, by hiding properly those effects and reducing the shot noise oscillation too. The final fit will benefit from this trick because the final ACF will appear smooth and less noisy.

- **The LSI Corelator is not retrieved by the PC or connection drops are frequent. What is the problem?**: The inability to detect the correlator has usually two possible causes:

  1. the attempt to detect the device before its full activation. Usually in fact, when the correlator is powered up or restarted, its full activation takes few tens of seconds. We suggest to wait about 1 minute or until the green LED on the front panel of the correlator housing is switched on and stable, before starting the LSI Acquisition software.
  2. a malfunction of one or both USB Ethernet adapters. This is a hardware problem upon which LS Instruments has very little control, because it depends on the lack of robustness of some adapter units. The adapters are cheap devices available everywhere and can be substituted easily. Moreover, testing the adapters is pretty straightforward: It is sufficient to check by eye whether or not the adapter LEDs blink properly once connected. Note that the head adapter is enclosed in a plastic membrane, but the LEDs are still visible. Alternatively the USB/Ethernet connection configuration (see Fig.3.2) can be used to test whether or not the problem persists. In this case, please contact LS instruments.

A DLS analysis

In this Appendix a short description of the cumulant and the CONTIN [8] analysis is given. The measured quantity in a DLS measurement is the so-called normalized intensity correlation function, $g_2(\tau, q)$:

$$g_2(\tau, q) = \frac{\langle I(0, q)I(\tau, q)\rangle}{\langle I(0, q)^2\rangle}. \tag{0.0.1}$$

The physically interesting quantity in terms of data analysis is however the normalized electric field correlation function, $g_1(\tau, q)$:

$$g_1(\tau, q) = \frac{\langle E(0, q)E^*(\tau, q)\rangle}{\langle I(0, q)\rangle}. \tag{0.0.2}$$

It allows us to extract useful information about the physical system at hand. Indeed, we can express $g_1(\tau, q)$ as follows:

$$g_1(\tau, q) = \frac{\int d\Gamma f(\Gamma)I(\Gamma, q)\exp(-\Gamma\tau)}{\int d\Gamma f(\Gamma)I(\Gamma, q)}. \tag{0.0.3}$$

where $\Gamma = q^2 D$ is the decay rate, $f(\Gamma)$ the decay rate distribution normalized to 1, and $I(\Gamma, q)$ the intensity scattered by the class of particles whose decay rate is Γ. In order to obtain $g_1(\tau, q)$ from the measured $g_2(\tau, q)$, we make use of the so-called Siegert relationship:

$$g_2(\tau, q) = 1 + \beta\left|g^{(1)}(\tau, q)\right|^2. \tag{0.0.4}$$

1 Cumulant

The cumulant analysis [1], [7], is an expansion of the normalized electric field correlation function, $g_1(\tau, q)$:

$$ln\left[g_1(\tau, q)\right] = -\bar{D}_0 q^2\tau + \frac{c_v}{2}(\bar{D}_0 q^2\tau)^2 + ... \tag{1.0.1}$$

$\bar{D}_0$ is the intensity weighted diffusion coefficient. Using the Stokes-Einstein equation and the viscosity of the sourrounding medium the radius can be calculated:

$$\bar{R} = \frac{kT}{6\pi\eta\bar{D}_0}. \tag{1.0.2}$$

2 CONTIN

Once $g_1(\tau, q)$ is obtained the purpose of the CONTIN [8] analysis is that of extracting $f(\Gamma)$ from it by means of Eq.0.0.3. The first step to be taken is the introduction of a numerical discretization for the integral which is accompanied by introducing a discrete set of Γ, $\Gamma = \{\Gamma_i\}$ values for which the corresponding discretized values of $f(\Gamma_i)$, $f = \{f_i\}$ are calculated. Taking into account the discretization and understanding that the denominator at right hand side of Eq.

0.0.3 is included in $f(\Gamma)$ itself, the correlation function measured at the delay time τ_j belonging to the discrete set of delay times $\tau = \{\tau_j\}$ where experimental data are available is expressed as

$$g_j = g_2(\tau_j, q) = \sum_i A_{ij} f_i \tag{2.0.1}$$

where A_{ji} arises from the numerical discretization scheme adopted for the integral. In matrix notation the previous equation takes the following form:

$$\mathbf{g} = A\mathbf{f}. \tag{2.0.2}$$

The simpler way to obtain the desired result is to simply invert the previous matrix equation as:

$$\mathbf{f} = A^{-1}\mathbf{g}. \tag{2.0.3}$$

The result of this is not well defined as Eq. 0.0.3 is said to be ill-posed; whichever the discretization we adopt and whatever the experimental accuracy we achieve, the discretized matrix equation will have a poor condition number and its inversion will not give a reasonable result. Since many f fit equally well our experimental data.
The CONTIN algorithm is an optimal smoothing strategy where among all the $\mathbf{f}$ fitting our experimental data the most suitable is chosen. This smoothing algorithm is based on the following: First of all Eq. 2.0.3 can be recasted in the equivalent least square form:

$$\mathbf{f} = \min_{f'} ||\mathbf{g} - A\mathbf{f}'||^2. \tag{2.0.4}$$

The equivalence can be proven by substituting $\mathbf{f} = A^{-1}\mathbf{g}$ into the previous equation. Once in this form, we can add to Eq. 2.0.4 a term that is inversely proportional to a measure of smoothness of $\mathbf{f}$:

$$\mathbf{f} = \min_{f'} \left[||\mathbf{g} - A\mathbf{f}'||^2 + \lambda B\mathbf{f}' \right]. \tag{2.0.5}$$

could be, for example, the result of the discretization of the following curvature modulus:

$$\int \left[f''(\Gamma) \right]^2 d\Gamma. \tag{2.0.6}$$

.
Clearly in this way noisy solutions are penalized and thus we regularize the problem. However the amount of regularization depends on the so-called regularization parameter λ. To close the problem we have to state an algorithm for the choice of λ, which leads to the optimal value that retains a parsimonious amount of information about the physical system in question and compatible with the noise present in the experimental data. This algorithm is the gist of the CONTIN procedure, however we cannot go into further detail due to its complexity. Please refer to [8] for more information.

B Troubleshooting

Please refer to the LS Instruments online forum for further useful information on the *LS Spectrometer* and light scattering topics:
http://www.lsinstruments.ch/services/online_forum

For technical support please visit:
http://www.lsinstruments.ch/support/
The customer support center allows you to review the status of your request and start new requests by opening a new ticket.

Optionally you may send an email to support@lsinstruments.ch

C Certificates

The LS Correlator is conform to the CE regulations.

Bibliography

[1] B.J. Berne and R. Pecora. *Dynamic light scattering*. Dover Publications, Mineola, 2000.

[2] Elson E. S. Rigler R. *Fluorescence Correlation Spectroscopy*. Springer, 2001.

[3] F. Scheffold. Particle sizing with diffusing wave spectroscopy. *Journal of Dispersion Science and technology*, 23:591–599, 2002.

[4] W. Liu, J. Shen, and X. Sun. Design of multiple tau photon correlation system implemented by fpga. *Embedded Software and Systems, 2008. ICESS 08.*, 2008.

[5] Schatzel K. *Dynamic Light Scattering: The Method and some Applications*. Ed. Clarendon Press, Oxford University Press, 1993.

[6] Ian D Block and Frank Scheffold. Modulated 3D cross-correlation light scattering: improving turbid sample characterization. *The Review of scientific instruments*, 81(12):123107, December 2010.

[7] Barbara J. Frisken. Revisiting the method of cumulants for the analysis of dynamic light scattering data. *Applied Optics*, 40(24):4087–91, 2001.

[8] Stephen W. Provencher. A constrained regularization method for inverting data represented by linear algebraic or integral equations. *Computer Physics Communications*, 27(3):213–227, September 1982.

[9] Websocket protocol. https://tools.ietf.org/html/rfc6455.

[10] Matlab event-listener. https://www.mathworks.com/help/matlab/matlab_oop/events-and-listeners--concepts.html.

[11] Python autobahn. https://autobahn.readthedocs.io/en/latest/contents.html.

[12] Python twisted. http://twistedmatrix.com/trac/.