

Hardware simulator for optical correlation spectroscopy with Gaussian statistics and arbitrary correlation functions

Matteo Molteni,¹ Udo M. Weigel,^{2,3} Francisco Remiro,^{2,3} Turgut Durduran,² and Fabio Ferri^{1*}

¹*Dipartimento di Scienza e Alta Tecnologia and To.Sca.Lab, Università dell'Insubria, Via Valleggio 11, 22100 Como, Italy*

²*ICFO- Institut de Ciències Fotòniques, Mediterranean Technology Park, 08860 Castelldefels (Barcelona), Spain*

³*Hemophotonics S.L., Mediterranean Technology Park, 08860 Castelldefels (Barcelona), Spain*

[*fabio.ferri@uninsubria.it](mailto:fabio.ferri@uninsubria.it)

Abstract: We present a new hardware simulator (HS) for characterization, testing and benchmarking of digital correlators used in various optical correlation spectroscopy experiments where the photon statistics is Gaussian and the corresponding time correlation function can have any arbitrary shape. Starting from the HS developed in [Rev. Sci. Instrum. **74**, 4273 (2003)], and using the same I/O board (PCI-6534 National Instrument) mounted on a modern PC (Intel Core i7-CPU, 3.07GHz, 12GB RAM), we have realized an instrument capable of delivering continuous streams of TTL pulses over two channels, with a time resolution of $\Delta t = 50ns$, up to a maximum count rate of $\langle I \rangle \sim 5MHz$. Pulse streams, typically detected in dynamic light scattering and diffuse correlation spectroscopy experiments were generated and measured with a commercial hardware correlator obtaining measured correlation functions that match accurately the expected ones.

© 2014 Optical Society of America

OCIS codes: (070.4550) Correlators; (290.0290) Scattering; (300.0300) Spectroscopy.

References and links

1. B.J. Berne and R. Pecora, *Dynamic Light Scattering*, (Wiley, NY, 1976).
2. D. J. Pine, D. A. Weitz, P. M. Chaikin, and E. Herbolzheimer, "Diffusing wave spectroscopy", Phys. Rev. Lett. **60**, 1134-1137 (1988).
3. D.A. Boas, L.E. Campbell, and A.G. Yodh, Scattering and imaging with diffusing temporal field correlations, Phys. Rev. Lett. **75**, 1855-1858 (1995).
4. T. Durduran, R. Choe, W.B. Baker and A.G. Yodh, Diffuse optics for tissue monitoring and tomography, Rep. Prog. Phys. **73**, 076701-076744 (2010).
5. K. Schatzel, "Single-photon correlation techniques", in *Dynamic Light Scattering*, edited by R.G.W. Brown (Clarendon, Oxford, 1993). Chapt. 2.
6. D. Magatti and F. Ferri, "Fast multi-tau real-time software correlator for dynamic light scattering", Appl. Opt. **40**, 4011-4021 (2001).
7. D. Magatti and F. Ferri "25 ns software correlator for photon and fluorescence correlation spectroscopy", Rev. Sci. Instrum. **74**, 1135-44 (2003).
8. M. Wahl, I. Gregor, M. Patting and J. Enderlein, "Fast calculation of fluorescence correlation data with asynchronous time-correlated single-photon counting", Opt. Express **11**, 3583-3591 (2003).

9. L.-L. Yang, H.-Y. Lee, M.-K. Wang, X.-Y. Lin, K.-H. Hsu, Y.-R. Chang, W. Fann, and J.D. White, "Real-time data acquisition incorporating high-speed software correlator for single-molecule spectroscopy", *J. Microscopy*, **234**, 302–310 (2009).
10. J. Buchholz, J. Wolfgang Krieger, G. Mocsar, B. Kreith, E. Charbon, G. Vamosi, U. Kebschull, and J. Langowski, "FPGA implementation of a 32x32 autocorrelator array for analysis of fast image series", *Opt. Express* **20**, 17767-17781 (2012).
11. E. Schaub, "F2Cor: fast 2-stage correlation algorithm for FCS and DLS", *Opt. Express* **20**, 2184–2195 (2012).
12. E. Schaub, "High countrate real-time FCS using F2Cor", *Opt. Express* **21**, 2184-2195 (2012).
13. F. Ferri and D. Magatti, "Hardware simulator for photon correlation spectroscopy", *Rev. Sci. Instrum.* **74**, 4273-4279 (2003).
14. J.W. Goodman, *Statistical Optics*, (Wiley & Sons Inc., New York, NY, 2000).
15. J.A. Williamson, "Statistical evaluation of dead time effects and pulse pileup in fast photon counting. Introduction of the sequential model", *Anal. Chem.* **60**, 2198-2203 (1988).
16. C. Zhou, G. Yu, F. Daisuke, J. H. Greenberg, A. G. Yodh, and T. Durduran, "Diffuse optical correlation tomography of cerebral blood flow during cortical spreading depression in rat brain", *Opt. Express*, **14** 1125-1144 (2006).
17. K. Schatzel, "Noise on photon correlation data: I. Autocorrelation function", *Quantum Opt.*, **2**, 287 305, (1990).
18. R. Peters, "Noise on photon correlation and its effects on data reduction algorithms", in *Dynamic Light Scattering* edited by W. Brown (Clarendon Press, Oxford, 1993) Chapt. 3.
19. W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, *Numerical Recipes in C* (Cambridge University Press, Cambridge, 1992).

1. Introduction

Optical correlation spectroscopy (OCS) is a powerful tool for studying many physical, biological and chemical systems that exhibit relaxation dynamics over a wide range of correlation times, from nano to milliseconds or greater. OCS works by measuring the time auto- or cross-correlation function of the intensity signal scattered by the sample under investigation. When the detected photons are singly scattered, the method is known with the classical names of Photon Correlation Spectroscopy (PCS) or Dynamic Light Scattering (DLS) [1] and has been applied for decades in the field of particle sizing, laser Doppler velocimetry and, in general, for the characterization of quasi-transparent complex fluids and soft matter systems. Conversely, in the opposite limit, when the sample is very turbid, the photons are scattered so strongly that they undergo diffusive propagation and the method, depending on the application, takes the original name of Diffusing Wave Spectroscopy (DWS) [2] or, as recently proposed in biomedical optics, the name of Diffuse Correlation Spectroscopy (DCS) [3, 4]. In all these cases, the statistics of the scattered photons is expected to be Gaussian and consequently the electric field correlation function can be recovered, via the well known Siegert relation [5] from the intensity correlation function which is the experimentally measured quantity.

In a typical OCS experiment, the intensity correlation function is computed by using a digital correlator, a device capable of computing the correlation function of the pulse stream out from the photodetector in real-time. Digital correlators can be either hardware-correlators that are commercially available [see, for example, the ones manufactured by ALV (Langen, Germany) or Brookhaven Instrument (Holtville, NY) or Correlator.com (Bridgewater, NJ)] or homemade software-correlators that have been proposed in literature over the last decade [6–12]. In both cases, the correlators need to be accurately tested and benchmarked to compare their performances.

For this reason, about a decade ago, some of us developed [13] a new tool known as a hardware simulator (HS). The HS is an instrument capable of delivering a continuous stream of transistor-transistor logic (TTL) pulses with the desired statistical properties over one or two channels. Its maximum time resolution was set by the board used (PCI-6534, National Instrument) and corresponded to a minimum temporal distance between pulses of $\Delta t = 50ns$. Conversely, the maximum count rates deliverable on one or two channels were determined

by the performances of the used personal computer (PC) and, in particular, by the speed at which the data could be read from the hard disk. In [13], where a *Pentium-4* PC and an *ATA-66, 7200rpm* hard disk were used, the maximum deliverable count rates were $\langle I \rangle \sim 350\text{KHz}$ and $\langle I \rangle \sim 80\text{KHz}$ on one or two channels, respectively.

The HS developed in [13] was specifically designed for testing digital correlators used in an *ideal* OCS experiment where the electric field correlation function $G_1(\tau) = \langle E(t)E^*(t+\tau) \rangle$ is a single exponential decay, i.e. $G_1(\tau) = \exp[-\tau/\tau_1]$, and the detection is acquired well inside a single coherence area so that the intensity correlation function, i.e. $G_2(\tau) = \langle I(t)I(t+\tau) \rangle$, according to the Siegert relation [5], is given by $G_2(\tau) = \langle I \rangle^2 [1 + |G_1(\tau)|^2 / \langle I \rangle^2]$. In this work we have removed the above restrictions, upgrading the HS with the possibility of generating pulse streams with an arbitrary correlation function and detecting the intensity over an (integer) number of coherence areas, $N_{ca} \geq 1$. While implementing these new features, we have also boosted up the overall performances of the HS and achieved, with a time resolution of $\Delta t = 50\text{ns}$, a maximum deliverable count rate higher than $\langle I \rangle \sim 5\text{MHz}$ on both channels which is similar to the maximum count-rates of common photon-counting detectors.

2. Theoretical background

In this section we will first describe how to synthesize a correlated stochastic signal $z(t)$ that is characterized by a Gaussian distribution and whose autocorrelation function can have any arbitrary shape $G_z(\tau)$. Secondly, we will show how to extract from $z(t)$ a stream of pulses that simulate the output of an ideal detector and are characterized by a given average intensity $\langle I \rangle$ and a autocorrelation function equal to $G_z(\tau)$.

As well-known [14], when a stochastic variable, $y(t)$, characterized by a probability Gaussian distribution with an average value, $\langle y \rangle$, standard deviation, σ_y , and a correlation function, $G_y(\tau) = \langle y(t)y(t+\tau) \rangle$, is fed through a time invariant linear filter with a normalized impulse response function, h ($\int_{-\infty}^{\infty} h(t)dt = 1$), the resulting stochastic variable $z(t)$ given by

$$\begin{aligned} z(t) &= \int_{-\infty}^{\infty} y(t-t')h(t')dt' \\ &= [y \otimes h](t), \end{aligned} \quad (1)$$

(symbol \otimes representing convolution) is still characterized by a Gaussian distribution, same average value $\langle z \rangle = \langle y \rangle$ and reduced (see below) standard deviation, σ_z . Its correlation function, $G_z(\tau) = \langle z(t)z(t+\tau) \rangle$, can be written in terms of $G_y(\tau)$ as [14]

$$G_z(\tau) = [G_y \otimes G_h](\tau) \quad (2)$$

where $G_h(\tau)$ is the symmetric [$G_h(\tau) = G_h(-\tau)$] correlation product of h given by

$$G_h(\tau) = \int_{-\infty}^{\infty} h(t)h(t+\tau)dt. \quad (3)$$

If the data $y(t)$ are zero-averaged ($\langle y \rangle = 0$) and δ -correlated ($G_y(\tau) = k\langle y^2 \rangle \delta(\tau)$, where k is a constant with the dimensions of time), it is straightforward to show that

$$G_z(\tau) = k\langle y^2 \rangle G_h(\tau) \quad (4)$$

demonstrating that $G_z(\tau)$ and $G_h(\tau)$ have the same shape or, equivalently, that their normalized versions are the same, i.e. $g_z(\tau) = g_h(\tau)$, where $g_x(\tau) = G_x(\tau)/G_x(0)$. Therefore, if we want to generate synthetic data $z(t)$, characterized by a given $g_z(\tau)$, we have to invert the equation

$$g_z(\tau) = \int_{-\infty}^{\infty} h'(t) h'(t + \tau) dt, \quad (5)$$

solve for $h'(t)$, find the normalized $h(t) = h'(t) / \int_{-\infty}^{\infty} h'(t) dt$, and finally apply Eq. (1) to the δ -correlated $y(t)$ data. Equation (5) can be solved in the Fourier domain

$$\widehat{g}_z(\omega) = \left| \widehat{h}'(\omega) \right|^2 \quad (6)$$

where the operator $\widehat{\cdot}$ denotes Fourier Transform and we have used the Fourier transform correlation theorem. Notice that in Eq. (6) $\widehat{g}_z(\omega)$ is real and symmetric, being the Fourier transform of the real and symmetric function $g_z(\tau)$. Square root of Eq. (6) gives

$$\sqrt{\widehat{g}_z(\omega)} e^{j\phi(\omega)} = \widehat{h}'(\omega) \quad (7)$$

where $\phi(\omega)$ is an arbitrary phase function. Among all the solutions satisfying Eq. (7), we can choose the one with $\phi(\omega) = 0$, implying that $\widehat{h}'(\omega)$ is real and symmetric (because $\widehat{g}_z(\omega)$ is real and symmetric). By inverse Fourier transforming Eq. (7) (with $\phi(\omega) = 0$) we get the solution

$$h'(t) = \sqrt{\widehat{g}_z(\omega)} \quad (8)$$

which is a real and symmetric function, i.e. $h'(-t) = h'(t)$.

In practice, we invert Eq. (5) by discretizing it on a time grid with a clock time Δt much smaller than the width of g_z , ($\Delta t \ll \delta_{g_z}$) and windowing the function g_z within a finite support $[-\tau_{max}, \tau_{max}]$, so that it can be sampled over a finite number of points. The value of τ_{max} can be found accordingly to the accuracy required for g_z , i.e. by zeroing all the τ -values for which $|g_z(|\tau| > \tau_{max})| \leq \varepsilon$, where ε is the required accuracy and we have used the property that $g_z(\tau \rightarrow \pm\infty) \rightarrow 0$. Thus, the approximate version of Eq. (5) evaluated at the discrete lag-times $\tau_k = k\Delta t$ with $k = 0, \pm 1, \pm 2, \dots, \pm k_{max} = \pm \tau_{max}/\Delta t$, is

$$\begin{cases} g_z(\tau_k) = \Delta t \sum_{j=-\infty}^{+\infty} h'_j h'_{j+k} & |\tau_k| \leq \tau_{max} \\ g_z(\tau_k) = 0 & |\tau_k| > \tau_{max} \end{cases} \quad (9)$$

Equation (9) can be solved numerically in the Fourier domain by computing, as outlined above, direct and inverse discrete Fourier Transforms. In this way we obtain the function h'_j that is discretized on the same times ($t_j = j\Delta t$) as g_z , is made of the same number of points $N_p = 2j_{max} + 1$ ($j_{max} = k_{max}$), and extends over the same finite interval $[-t_{max}, t_{max}] = [-\tau_{max}, \tau_{max}]$ (see Fig. 6 in appendix B).

The next step is to synthesize a continuous stochastic function $y(t)$ by using the random generator available on the PC and generating a stream of random, uncorrelated numbers y_i with a Gaussian distribution, average value $\langle y \rangle = 0$ and standard deviation σ_y . The function $y(t)$ is assumed to be constant between launches, i.e. $y(t) = y_i$ for any $(i - 1/2)\Delta t \leq t < (i + 1/2)\Delta t$ with $i = 0, 1, 2, \dots, \infty$. A useful equivalent representation of such a function is

$$y(t) = \sum_{i=0}^{\infty} y_i R\left(\frac{t-t_i}{\Delta t}\right) \quad (10)$$

where $R(x)$ is the rectangle function ($R(x) = 1$ for $-0.5 \leq x < 0.5$ and 0 elsewhere) and $t_i = i\Delta t$. Equation (10) represents the discretized version of a δ -correlated function that is expected

to remain correlated only for lag times $|\tau| < \Delta t$. Indeed, its correlation function reads (see appendix A)

$$G_y(\tau) = \langle y^2 \rangle \Lambda(\tau/\Delta t) \quad (11)$$

where the triangular function $\Lambda(x)$ is the correlation product of $R(x)$ and is defined as $\Lambda(x) = 1 - |x|$ for $|x| \leq 1$ and 0 elsewhere. As expected, Eq. (11) shows that $G_y(\tau) = 0$ as soon as $|\tau| \geq \Delta t$.

By inserting Eq. (11) into Eq. (2) and using the property that the width of G_y is much smaller than the width of G_h ($\Delta t \ll \delta_G$), we obtain the correlation of the function $z(t)$ as

$$G_z(\tau) = \langle y^2 \rangle G_h(\tau) \Delta t \quad (12)$$

which is the discretized version of Eq. (4) when the time resolution of the system is Δt .

At this point the stream of correlated random numbers z_i can be obtained from the discretization of Eq. (1) that, after normalization of h' so that $h_j = h'_j/\Delta t \sum h'_j$, reads

$$z_i = \Delta t \sum_{j=-j_{max}}^{j_{max}} y_{i-j} h_j \quad (13)$$

where the convolution extends over $N_p = 2j_{max} + 1$ points. Equation (13) is the critical part of the procedure described so far for the synthesis of the z_i numbers because each new point z_i requires the computation and the sum of N_p products, a task that could become overwhelming when N_p is very large. We will describe how to cope with this problem at the end of next section and report a practical example in Appendix B.

An important parameter that will be used below for setting the amplitude of the signal to be synthesized is the variance σ_z^2 . This is given by the zero-value of Eq. (12), i.e.

$$\sigma_z^2 = G_z(0) = \sigma_y^2 G_h(0) \Delta t \quad (14)$$

where we have used $\langle y \rangle = \langle z \rangle = 0$. Equation (14) shows that σ_z^2 and σ_y^2 are related to each other by the factor $G_h(0) \Delta t$, which can be easily computed. Indeed, since $G_h(0) = \Delta t \sum h_j^2$ and, $\Delta t \sum h_j = 1$, we can write

$$\sigma_z^2 = \sigma_y^2 \frac{\sum h_j^2}{[\sum h_j]^2} \quad (15)$$

or, in terms of the first two moments of h , as

$$\sigma_z^2 = \sigma_y^2 \frac{\langle h^2 \rangle}{\langle h \rangle^2} \frac{1}{N_p} \sim \sigma_y^2 \frac{\Delta t}{\delta_h} \quad (16)$$

where $\langle h \rangle = (1/N_p) \sum h_j$, $\langle h^2 \rangle = (1/N_p) \sum h_j^2$ and in the right hand side term we have used the property $\langle h^2 \rangle / \langle h \rangle^2 \sim 1$ (which is a direct consequence of the normalization of h to 1). Equation (16) shows that the convolution filter dampers the squared fluctuations of the variable $y(t)$ by a factor proportional to $\Delta t/\delta_h \ll 1$.

We can now to use the above framework for generating synthetic intensity data similar to the ones collected in an OCS experiment carried out with polarized light. As mentioned above, in

a system governed by a gaussian statistics, the intensity, $G_2(\tau)$, and field, $G_1(\tau)$, correlation functions are related to each others by the Siegert relation [5]

$$G_2(\tau) = \langle I \rangle^2 \left[1 + \beta \frac{|G_1(\tau)|^2}{\langle I \rangle^2} \right] \quad (17)$$

where $\beta \leq 1$ is a geometrical factor (called coherence factor), which scales as the reciprocal of the number of coherence areas falling onto the detector, $\beta \sim 1/N_{ca}$ (for $N_{ca} \gg 1$). Geometrically, for each coherence area, the electric field scattered by the sample can be represented as a vector performing a “random walk” in two dimensions, in which the two components, E_x and E_y , are independent stochastic variables representing the *real* and *imaginary* parts of the polarized electric field [14]. E_x and E_y are characterized by the same Gaussian probability distribution with $\langle E_x \rangle = \langle E_y \rangle = 0$ and $\sigma_{E_x}^2 = \sigma_{E_y}^2 = \langle I \rangle / 2$, where $\langle I \rangle$ is the average intensity $\langle I \rangle = \langle E_x^2 \rangle + \langle E_y^2 \rangle$. Thus the intensity data stream collected within a single coherence area is

$$I_i = (E_x)_i^2 + (E_y)_i^2 \quad (18)$$

where $(E_x)_i$ and $(E_y)_i$ are generated by using Eq. (13), in which the y_i are random numbers drawn from a Gaussian distribution with $\langle y \rangle = 0$ and σ_y chosen according to Eq. (15) so that $\sigma_z^2 = \langle I \rangle / 2$.

When $N_{ca} > 1$, the procedure has to be repeated N_{ca} times, so that the k^{th} intensity data stream is $I_i^k = [(E_x^k)_i]^2 + [(E_y^k)_i]^2$ and the overall intensity is $I_i = \frac{1}{N_{ca}} \sum_{k=1}^{N_{ca}} I_i^k$. In this way single coherence area intensities I_i^k sum up incoherently, and the correlation function of the overall intensity is

$$G_2(\tau_k) = \langle I \rangle^2 \left[1 + \frac{1}{N_{ca}} \left(\frac{G_z(\tau_k)}{G_z(0)} \right)^2 \right]. \quad (19)$$

Equation (18) represents the generalization of Eq. (11) of [13] to an arbitrary correlation function with an arbitrary coherence factor $\beta = 1/N_{ca}$, being N_{ca} an integer number.

If we express $\langle I \rangle$ as the average number of photons scattered per unit time, the number R_i of photons falling within the i^{th} interval Δt is simply $R_i = I_i \Delta t$. To obtain from R_i an integer number N_i which represents the number of photocounts in the i^{th} interval Δt_i , it is necessary to pass R_i through a Poisson filter (see Fig. 1 and [13]). The Poisson filter simulates what happens in an ideal photodetector where the output is an integer random number N_i (the number of photoelectrons) sampled according to a Poisson distribution characterized by an average value equal to R_i . When $R_i \ll 1$, the integers N_i are mostly either zeros or ones. This condition is easily fulfilled because the probability of $P(N_i \geq 2)$ is equal (for an uncorrelated signal) to $\langle R \rangle^2 / 2$, where $\langle R \rangle$ is the probability of having one photon per time interval, given by $\langle R \rangle = \langle I \rangle \Delta t$. Thus for example, even at a count rate as high as $\langle I \rangle = 1MHz$, with $\Delta t = 50ns$, we have $\langle R \rangle = 5 \times 10^{-2}$ and $P(N_i \geq 2) \sim 10^{-3}$.

Once the photocounts, N_i , have been generated, the sequence of zeros and ones must be transformed in integrated arrival times and written onto the PC hard disk (see [13]). The space required on the hard disk is large, but, definitely, much smaller than standard capacities of modern hard disks $\sim 10^2 - 10^3 GByte$. For example, at a pulse count rate $\langle I \rangle = 1MHz$ for a measuring time of $T = 100s$, the overall number of photocounts would be $\sim \langle I \rangle T = 10^8$, and the corresponding file would be $\sim 1 GByte$ in size.

3. Simulator architecture

The simulator architecture is identical to the one reported in [13]. It uses the same I/O board (NI PCI-6534, National Instruments), but a much higher performance PC (Intel Core i7-CPU, 3.07GHz, 12GB RAM) running under *Windows-7 Enterprise-N, 64 bit*. The homemade software, originally developed under LabVIEW 6.1, was upgraded to LabVIEW 2010, optimized and developed according to the scheme reported in Fig. 2 of [13].

The two main advantages of this report with respect to the original work are: (a) the reading task, which was the major limiting factor for the performances of the simulator, is now much faster. By using a standard hard disk (SATA, 3.0 Gb/s, 7200 rpm) we attain, for the reading task, a value of $\delta t_1 \sim 0.02 \mu s/pulse$, which is a factor ~ 100 smaller than the original value; (b) the “data buffer”, which is limited by the RAM memory available on the host PC, is now much larger and set to a size of $\sim 8.8 \times 10^8$. The combination of these two features boosts up the performances of the hardware simulator to levels that, as matter of fact, are only limited by the time resolution of the PCI-6534 board.

In practice, we are now able to generate at the maximum time resolution of $\Delta t = 50 ns$ streams of pulses up to count rates $\sim 5 MHz$ on both channels. Note that a further increase of the count rate would be troublesome because pile up effects [15] would become dominant and affect both the count rate and the statistics of the pulse stream delivered by the board. An example of this effect is shown below.

As a final remark, we should mention the main drawback of the new HS. While in the original work the generation of the correlated data z_i was done iteratively and, for each new z_i only one single uncorrelated point y_i was needed (see Eq. (7) of [13]), in this work we have to use Eq. (13), which is a convolution equation. Thus, the generation of each new z_i requires the sum of a large number of terms, and the use of Eq. (13) can easily become unpractical because of extremely long computational times. We have overcome this problem by resampling both h_j and y_i on a pseudo-geometrical progression time grid (instead of the traditional linear grid) so to reach the same lag time interval $[-\tau_{max}, \tau_{max}]$, but with a much smaller number of points. We will describe in Appendix B how to practically implement such a non-linear time grid and provide estimates for the corresponding computational times.

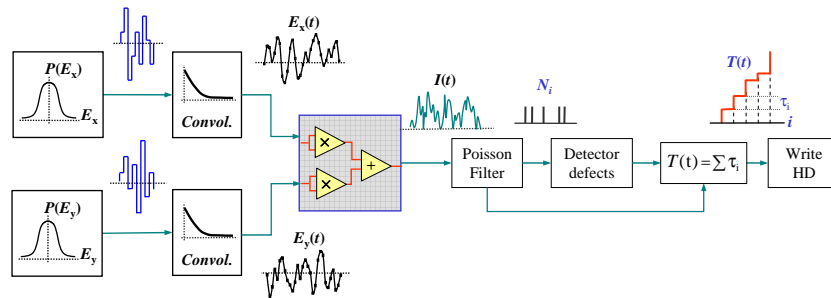


Fig. 1. (Adapted from Fig. 1 of [13]). Schematic diagram of the algorithm adopted for generating synthetic correlation data. $E_x(t)$ and $E_y(t)$ are the two components of the scattered electric field, $I(t)$ is the intensity, N_i the sequence of counts (0 or 1) after the Poisson filter and $T(t)$ the integrated photon arrival times that are stored on the hard disk (HD). Optionally, the presence of the detector defects can also be introduced.

4. Experimental results

The simulator was tested by processing its pulse stream with a commercial hardware correlator (model Flex2k 12x2 from Correlator.com) that adopts a multi-tau scheme [5] with a minimum delay time of $12.5ns$. Unless otherwise specified, all the tests were carried by generating the pulse stream at the highest time resolution of $\Delta t = 50ns$ with a count rate of $\langle I \rangle = 10^5 Hz$ for a measuring time of $T = 100s$.

The first test was aimed to show the ability of the simulator to generate correlation functions of different shapes such as the ones characterized by exponential decays with different stretched or compressed exponents α :

$$G_1(\tau) = \exp \left[- \left(\frac{\tau}{\tau_c} \right)^\alpha \right]. \quad (20)$$

In Fig. 2(a) we compare the full coherent ($N_{ca} = 1$), normalized intensity autocorrelation functions $g_2(\tau) = 1 + (G_1(\tau)/G_1(0))^2$ for three values of $\alpha = 0.5, 1.0, 1.5$ and same $\tau_c = 10^{-5}s$ (lines) with the corresponding correlation functions measured with the Flex correlator (symbols). The data corresponding to the first 32 channels of the Flex correlator (16 channels @ $12.5ns$, 16 channels @ $25.0ns$) were binned and averaged so to have lag times equal to multiples of the clock time $\Delta t = 50ns$ (see details in [13]). As one can appreciate, all the curves are accurately matched by the measured correlation functions, with only slight or non-systematic residuals [panel (b)], whose overall deviations (except for the first two points, see below) were less than 1% *rms*. The small systematic deviations occurring in the range $\sim 10^{-5} - 10^{-7}s$ are due to the finite accuracy used in Eqs. 9 and 10 ($\varepsilon = 5 \times 10^{-3}$). The first two points that exhibit much higher deviations are due to a malfunctioning of the Flex correlator. The same kind of deviations are present in all the test we have done, but do not show up when using other correlators such as the software correlator of [7] (data not shown).

The second test was based on a typical correlation function measured in a diffuse correlation

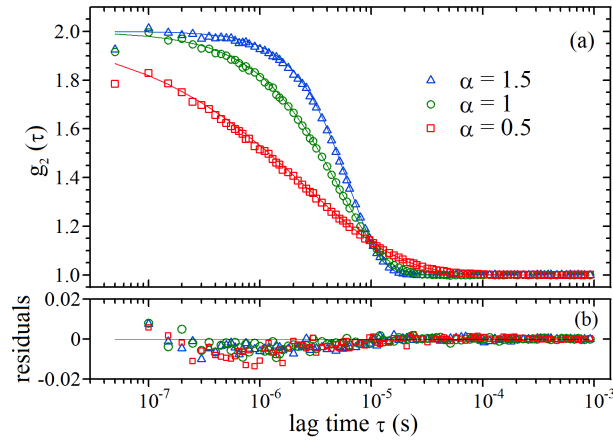


Fig. 2. (a): Normalized Intensity correlation functions (symbols) obtained by measuring with the Flex2k 12x2 correlator (by Correlator.com) a pulse stream characterized by the same decay time $\tau_c = 10^{-5}s$ and different stretched/compressed exponentials with $\alpha = 0.5, 1.0, 1.5$ (see Eq. (20)). The solid lines represent the expected behaviors. The pulses were delivered with a $\Delta t = 50ns$ clock, at a count rate $\langle I \rangle = 10^5 Hz$, for a measuring time $T = 100s$. (b): Relative deviations between data and expected curves.

spectroscopy (DCS) [4] experiment for the monitoring of blood flow in living tissues. A general expression for the electric field autocorrelation function in a reflection geometry with a point source illumination placed at a distance z from the tissue plane with semi-infinite slab boundary conditions and the detector placed at a transversal distance ρ from the source, is given by Eq. (40) of [4] as

$$G_1(\rho, z, \tau) = \frac{v}{4\pi D} \left[\frac{\exp[-K(\tau)r_1]}{r_1} - \frac{\exp[-K(\tau)r_b]}{r_b} \right] \quad (21)$$

where $K(\tau) = \sqrt{(\mu_a + \alpha\mu'_s k_0^2 \langle \Delta r^2(\tau) \rangle / 3) v / D}$, $r_1 = \sqrt{(z - \ell_{tr})^2 + \rho^2}$, $r_b = \sqrt{(z + 2z_b + \ell_{tr})^2 + \rho^2}$. In Eq. (21) v is the speed of light in tissue, D is the photon diffusion coefficient in tissue, μ_a is the tissue absorption coefficient, μ'_s is the tissue reduced scattering coefficient, α is the fraction of dynamic photon scattering events, $k_0 = 2\pi/\lambda$ is the wavenumber of the continuous-wave laser light at (medium) wavelength λ , $\langle \Delta r^2(\tau) \rangle$ is the mean-square displacement in time τ of the scattering particles, ℓ_{tr} is the photon transport mean free path ($\ell_{tr} = 1/\mu'_s$), and z_b is the boundary plane where the photon fluency is extrapolated to be zero (see Figure 5 and Table 1 of [4]).

When the particles are subjected to Brownian motion their mean-square displacement is $\langle \Delta r^2(\tau) \rangle = 6D_b\tau$ (being D_b their diffusion coefficient), whereas when they undergo a random flow $\langle \Delta r^2(\tau) \rangle = \langle V^2 \rangle \tau^2$ (being $\langle V^2 \rangle$ the mean square particle velocity). Note that the dependence of Eq. (21) on the ratio v/D is only apparent because $D = v/3(\mu_a + \mu'_s)$. Fig. 3(a) shows, as solid lines, the behaviors of two full coherent ($N_{ca} = 1$), normalized intensity correlation functions $g_2(\tau) = 1 + |G_1(\rho, z, \tau)/G_1(\rho, z, 0)|^2$ for the same set of parameters ($z = 0$, $\rho = 2.5 \text{ cm}$, $\mu_a = 0.15 \text{ cm}^{-1}$, $\mu'_s = 15 \text{ cm}^{-1}$, $\alpha = 0.95$, $\lambda = 5.9 \times 10^{-5} \text{ cm}$), but different mean-square displacements for the particles motion. The green curve refers to a Brownian motion with $D_b = 1.4 \times 10^{-7} \text{ cm}^2/\text{s}$, whereas the blue curve corresponds to random flow with

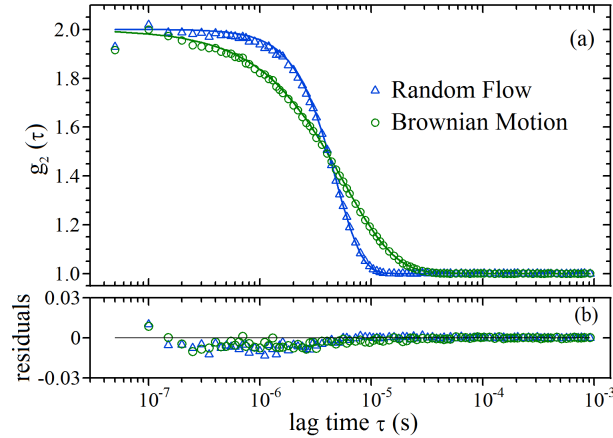


Fig. 3. (a): Normalized Intensity correlation functions (symbols) obtained by measuring with the Flex2k 12x2 correlator two pulse streams characterized by a field correlation function given by Eq. (21) (see text for parameters values). The solid lines represent the expected behaviors. The pulses were delivered with a $\Delta t = 50 \text{ ns}$ clock, at a count rate $\langle I \rangle = 10^5 \text{ Hz}$, for a time $T = 100 \text{ s}$. (b): Relative deviations between data and expected curves.

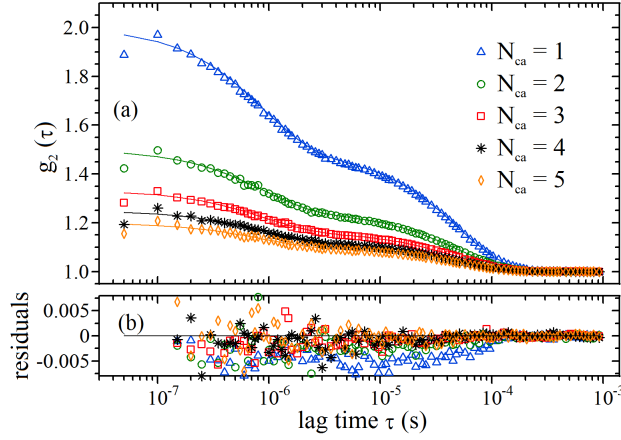


Fig. 4. (a): Normalized Intensity correlation functions (symbols) obtained by measuring with the Flex2k-12x2 correlator a pulse stream characterized by two single exponential decay times [see Eq. (22)] with $\tau_1 = 10^{-6}s$, $\tau_2 = 10^{-4}s$ and $A_1/(A_1 + A_2) = 0.30$. Different data sets refer to data being acquired within different coherence areas, from $N_{ca} = 1$ to 5. The solid lines represent the expected behaviors. The pulses were delivered with a $\Delta t = 50ns$ clock, at a count rate $\langle I \rangle = 10^5 Hz$, for a time $T = 100s$. (b): Relative deviations between data and expected curves.

$\langle V^2 \rangle = 0.2 cm^2/s^2$. On the same Figure, reported as open symbols, are shown the $g_2(\tau)$ data obtained with the Flex correlator by using the same working conditions of Fig. 2. As in the previous case, the two curves are accurately matched by the measured correlation functions, with slightly or non-systematic residuals, whose overall deviations (except for the first two points) were less than 1% rms.

The third test shows that the simulator is capable of delivering pulse streams characterized by two single exponential decay times and detection carried out by integrating the scattered intensity over a few coherence areas. Thus,

$$G_1(\tau) = A_1 \exp(-\tau/\tau_1) + A_2 \exp(-\tau/\tau_2). \quad (22)$$

and $g_2(\tau) = 1 + (1/N_{ca})(G_1(\tau)/G_1(0))^2$ with N_{ca} being the number of coherence areas. When τ_1 and τ_2 are sufficiently different and $A_1 \sim A_2$, the normalized intensity correlation function g_2 associated to Eq. (22) deviates significantly from a single exponential decay and assumes a sigmoidal shape, as shown by the five solid lines of Fig. 4(a). All these curves have been obtained with $\tau_1 = 10^{-6}s$, $\tau_2 = 10^{-4}s$, $A_1/(A_1 + A_2) = 0.30$ and N_{ca} ranging from 1 to 5 (top to bottom). The correlation function measured with the Flex correlator (open symbols) match quite accurately the expected curve, as evidenced by the residual plot reported in Fig. 4(b).

Finally, we discuss the main limitation of the hardware simulator, which is related to the maximum output count rate the simulator can deliver. As mentioned in Sect.3, the time resolution of the of the PCI-6534 board is $\Delta t = 50ns$, meaning that, if two or more pulses are closer than $50ns$ pile up effects [15] take places and only a single pulse is output on the port. Thus, if $\langle I \rangle$ is the true pulse count rate, the effective count rate delivered by the board is smaller and, for $\Delta t \langle I \rangle \ll 1$, can be approximated by $\langle I \rangle_{eff} \sim \langle I \rangle [1 - \Delta t \langle I \rangle]$ [15].

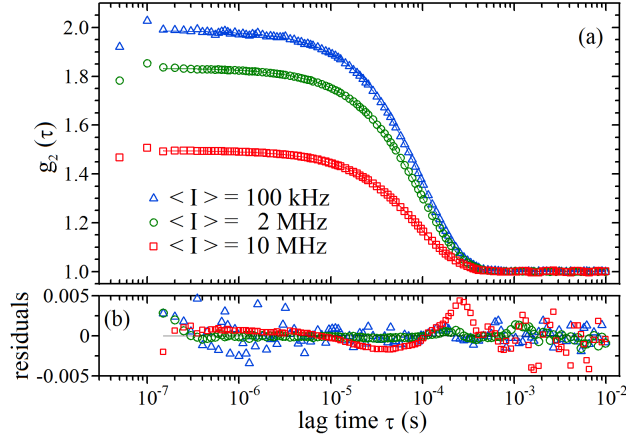


Fig. 5. (a): Normalized intensity correlation functions (symbols) obtained by measuring with the Flex2k 12x2 correlator pulse streams characterized by the same single exponential decay time $\tau_c = 10^{-4}s$ and different count rates from 100kHz to 10MHz. The solid lines represent the fitting to single exponential decay functions. The pulses were delivered with a $\Delta t = 50ns$ clock, for a time time of $T = 100s$ (blue and green symbols) and $T = 10s$ (red symbols). (b): Relative deviations between data and fitting curves.

As a consequence, the ratio $\langle I \rangle_{eff} / \langle I \rangle$ becomes increasingly smaller as $\langle I \rangle$ gets higher. For example, we have $\langle I \rangle_{eff} / \langle I \rangle \sim .98$ for $\langle I \rangle = 5 \times 10^5 Hz$ and $\langle I \rangle_{eff} / \langle I \rangle \sim .75$ for $\langle I \rangle = 5 \times 10^6 Hz$. Pile-up effects cause also a reduction of the *rms* fluctuations of the signal, because high fluctuations are dampened and thresholded to a one count /clock. Fig. 5(a) shows an example of how the intensity correlation function change as the count rate is increased: the pulse streams, delivered at a fixed $\Delta t = 50ns$ clock, were characterized by the same single exponential decay [Eq. (20)] with $\alpha = 1$, $\tau_c = 10^{-4}s$ and with count rates varying from 100kHz to 10MHz. The open symbols represent the autocorrelation data taken with the Flex correlator for a measuring time of $T = 100s$ (blue and green symbols) and $T = 10s$ (red symbols). Surprisingly, whereas the correlation amplitude decreases remarkably with increasing the count rate, the effects on the shape of g_2 are much less pronounced. Indeed, when fitting the data to single exponential decay functions, the fittings appear rather accurate [residuals slightly systematic only for the 10MHz data (red symbols), see panel (b)] and the recovered correlation times appear to be quite close to the expected ones, i.e. $\tau_c = 1.00 \times 10^{-4} \pm 1.38 \times 10^{-7}s$, $0.99 \times 10^{-4} \pm 0.62 \times 10^{-7}s$, $0.93 \times 10^{-4} \pm 2.65 \times 10^{-7}s$ for the 100kHz, 2MHz and 10MHz count rates, respectively. Thus, we conclude that pile-up affects only moderately the relaxation dynamics of the system.

As a final comment, we would like to point out that pile-up effects would be greatly reduced if a faster board would be used. For example, with the board PCI-6562 (National instruments), which has a time resolution of $\Delta t = 5ns$, pile-up effects would be reduced by a factor 10 with respect to PCI-6534 board used in this work. Thus, even at a count rate $\langle I \rangle > 1MHz$, the correlation function of the generated pulses would present no appreciable distortions, exactly as the blue data of Fig. 5(a). Notably, common photon-counting avalanche photo diodes (e.g. SPCM series by Excelitas, Canada), that are utilized in methods such as DCS operate with a similar maximum count-rate.

5. Conclusions

In this work we have developed a new hardware simulator (HS) capable of delivering a continuous stream of TTL pulses with Gaussian statistics and an arbitrary correlation function. The HS is ideal for testing and benchmarking digital correlators used in various OCS experiments where the correlation function may be significantly different from that of a single exponential decay. Indeed, the latter one can be recovered only in an *ideal* OCS experiment, as for example, in the case of DLS measurements taken on a mono-disperse diluted particulate samples when the detection is acquired well inside a single coherence area. Conversely, in many cases these conditions are not met, such as for complex systems exhibiting sub-diffusive or stretched dynamics, or in the case of diffuse correlation spectroscopy experiments carried out in biomedical optics for monitoring blood flow in living tissues.

The HS developed in this work makes use of a commercial I/O board (National Instrument, PCI-6534), a modern PC (Intel Core i7-CPU, 3.07GHz, 12GB RAM), and a home made code developed under LabVIEW 2010 (National Instrument TM). It is capable of delivering pulses over one or two channels with a time resolution of $\Delta t = 50ns$ up to a maximum count rate of $\langle I \rangle \sim 5MHz$ for both channels. The two channels are synchronized with the same clock so that their outputs can be profitably used for testing the cross-correlation capability of the correlator and checking its efficiency in the removal of detectors afterpulses.

The proper functioning of the HS was ascertained by computing the autocorrelation function of the output pulses by using a commercial hardware correlator, namely the Flex2k-12x2 correlator (by Correlator.com). We synthesized pulse streams corresponding to various exponential and non-exponential decay correlation functions, at different count rates, with the detection being carried out over different coherence areas. In all the cases, the measured correlation functions were in excellent agreement with the expected ones.

Beside its use for testing and benchmarking digital correlators, the HS can be also exploited for investigating and optimizing their statistical performances, which are described by the variance-covariance matrix $C[g_2(\tau_k), g_2(\tau_m)]$ of the normalized intensity correlation function g_2 [17, 18]. This matrix, whose diagonal elements are directly related to the signal-to-noise ratio of g_2 , depends on many experimental factors, such as the measuring time T , the average count rate $\langle I \rangle_T$, and the number of collected coherence areas N_{ca} . However, $C[g_2(\tau_k), g_2(\tau_m)]$ depends also on (i) the features of the detection hardware components such as the photodetector defects and on (ii) the specific procedure used by the correlator for recovering g_2 from the raw data, such as the adopted normalization method and the multi-tau scheme [5]. In this respect, the HS offers a great opportunity because, knowing exactly the incoming intensity and corresponding incoming pulse sequence, it provides an easy way to select and/or design the specifications of the detection chain and optimize the correlator architecture. Various OCS based instruments could benefit from such optimization. For example, it was previously demonstrated that the signal-to-noise ratio of common DCS measurements is dependent on the correlator architecture and can be tuned for specific probe geometries and tissue conditions [16].

Finally, it is worth mentioning that, although the HS developed this work was designed for synthesizing only pulse streams characterized by Gaussian statistics (and arbitrary correlation functions), it can also be implemented for simulating pulse patterns with different statistics, such as the ones encountered in fluorescence correlation spectroscopy (FCS). The latter one is a quite popular and powerful technique, worldwide used for investigating the dynamics of many physical, chemical and biophysical systems, even at the level of single molecule dynamics. The implementation of the HS for the various applications of FCS is not straightforward, and we are currently working on this topic.

Appendix

A. Derivation of Eq. (11)

In this Appendix, we show how to work out Eq. (11) from Eq. (10). The autocorrelation function of $y(t)$, $G_y(\tau) = \langle y(t)y(t+\tau) \rangle$, can be worked out by computing the time and ensemble averages of the product $y(t)y(t+\tau)$. Thus

$$G_y(\tau) = \left\langle \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T y(t)y(t+\tau) dt \right\rangle \quad (\text{A.1})$$

where the symbol $\langle \cdot \rangle$ represents ensemble average. By inserting into Eq.(A.1) the expression for $y(t)$ given by Eq. (10), and interchanging ensemble average with the time average, we get

$$G_y(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T \sum_{m,k=0}^{\infty} \langle y_m y_k \rangle R\left(\frac{t-t_m}{\Delta t}\right) R\left(\frac{t+\tau-t_k}{\Delta t}\right) dt. \quad (\text{A.2})$$

By noting that the double sum in Eq. (A.2) reduces to a single sum because the y_m are δ -correlated ($\langle y_m y_k \rangle = \langle y^2 \rangle \delta_{mk}$), we obtain

$$G_y(\tau) = \lim_{T \rightarrow \infty} \langle y^2 \rangle \frac{1}{T} \int_0^T \sum_{m=0}^{\infty} R\left(\frac{t-t_m}{\Delta t}\right) R\left(\frac{t+\tau-t_m}{\Delta t}\right) dt. \quad (\text{A.3})$$

Upon interchanging again ensemble and time average, we obtain

$$G_y(\tau) = \lim_{N \rightarrow \infty} \langle y^2 \rangle \frac{1}{N\Delta t} \sum_{m=0}^N \int_0^T R\left(\frac{t-t_m}{\Delta t}\right) R\left(\frac{t+\tau-t_m}{\Delta t}\right) dt. \quad (\text{A.4})$$

where $T = N\Delta t$. Finally, by noting that for any $m > 0$, $\int_0^T R\left(\frac{t-t_m}{\Delta t}\right) R\left(\frac{t+\tau-t_m}{\Delta t}\right) dt = \int_{-m\Delta t}^{\infty} R(t/\Delta t)R(t+\tau/\Delta t) dt = \Delta t \Lambda(\tau/\Delta t)$, we have

$$G_y(\tau) = \langle y^2 \rangle \Lambda\left(\frac{\tau}{\Delta t}\right) \quad (\text{A.5})$$

which is identical to Eq. (11).

B. Operative procedure for the generation of a pulse stream with a given average count rate and correlation function

In this Appendix, we describe how to generate a pulse stream corresponding to an average count rate $\langle I \rangle$ and characterized by a field correlation function $G_z(\tau)$. The steps necessary to synthesize such a signal are the following:

(1) compute the normalized correlation function $g_z(\tau) = G_z(\tau)/G_z(0)$ and sample it on a time grid with a clock time Δt much smaller than the width δ_{g_z} of g_z . Since g_z is symmetric [$g_z(\tau) = g_z(-\tau)$], normalized ($g_z(0) = 1$) and decays asymptotically to zero ($g_z(\tau \rightarrow \pm\infty) \rightarrow 0$), it is possible to set an accuracy ε and window g_z inside a finite interval $[-\tau_{max}, \tau_{max}]$ found so that $|g_z(\tau_k) - g_z(\tau)| \leq \varepsilon$. [see Fig. 6(a)]. In this way, $g_z(\tau_k)$ is sampled over a finite number of point equal to $N_p = 2k_{max} + 1$, where $k_{max} = \tau_{max}/\Delta t$.

(2) invert Eq. (9) in the Fourier domain by computing direct and inverse discrete Fourier Transforms, as described via Eqs.(6-8). The recovered function h'_j is discretized over the same times ($t_j = j\Delta t$) as g_z , is made of the same number of points $N_p = 2j_{max} + 1$ ($j_{max} = k_{max}$), and extends over the same finite interval $[-t_{max}, t_{max}] = [-\tau_{max}, \tau_{max}]$. From h'_j compute the

normalized response function as $h_j = h'_j / \Delta t \sum h'_j$. [solid line of Fig. 6(b)]

(3) to speed up the convolution procedure [Eq. (13)], resample h_j on a non-linear time-grid so to cover the same interval $[-t_{max}, t_{max}]$, but with a much smaller number of points. Since $h(t)$ is symmetrical [$h(t) = h(-t)$] and asymptotically decays to zero ($h(t \rightarrow \pm\infty) \rightarrow 0$), it is convenient to choose a time grid with symmetric intervals that increase progressively as they move away from $t = 0$. Let us indicate the new time grid with the times \tilde{t}_q ($q = 0, \pm 1, \pm 2, \dots, \pm q_{max}$) that are selected so that $\tilde{t}_0 = 0$ and $\tilde{t}_q = n(q)\Delta t$ where $n(q)$ is an integer number such that $n(q) = -n(-q)$. In this way, the time grid is symmetric around $t = 0$, all the \tilde{t}_q are multiples of Δt , and q_{max} is found so that $\tilde{t}_{q_{max}} = t_{max}$. A convenient way to define such a grid is to set the total number of points $\tilde{N}_p = 2q_{max} + 1$ to be used and let the modulus $|n(q)|$ increase according to a pseudo-geometrical progression that start with $|n(1)| = 1$. Thus the resampling will look like $[-t_{max}, \dots, \tilde{t}_{-1} = -\Delta t, \tilde{t}_0 = 0, \tilde{t}_{-1} = -\Delta t, \tilde{t}_{+1} = \Delta t, \dots, \tilde{t}_q = n(q)\Delta t, \dots, t_{max}]$ and the size of the intervals $\tilde{\Delta t}_q$ between adjacent times is $\tilde{\Delta t}_q = \tilde{t}_{|q|+1} - \tilde{t}_{|q|}$. We therefore define the resampled values \tilde{h}_q as the average values of h inside $\tilde{\Delta t}_q$, i.e.

$$\begin{cases} \tilde{h}_q = \frac{1}{\Delta n(q)} \sum_{j=n(q-1)+1}^{n(q)} h_j & q < 0 \\ \tilde{h}_q = h_0 & q = 0 \\ \tilde{h}_q = \frac{1}{\Delta n(q)} \sum_{j=n(q)}^{n(q+1)-1} h_j & q > 0 \end{cases} \quad (\text{B.1})$$

where $\Delta n(q) = n(|q| + 1) - n(|q|)$ is equal to the number of clocks Δt inside the interval $\tilde{\Delta t}_q$. A sketch of the new time grid, together with the first few values of \tilde{h}_q are reported in Fig. 7. The overall resampling of the function h corresponding to the g_z of Fig. 6(a) is shown in Fig. 6(b), where the number of points is $\tilde{N}_p = 100$.

(4) determine σ_y chosen according to Eq. (16), in which $\sigma_z^2 = \langle I \rangle / 2$.

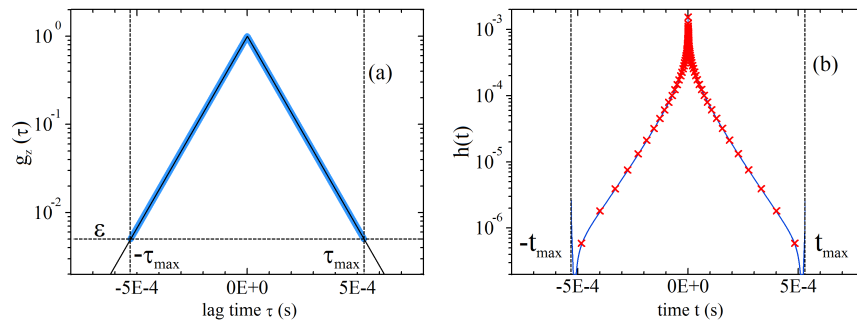


Fig. 6. (a): normalized field correlation function $g_z(\tau)$ corresponding to a single exponential decay with decay time $\tau_c = 10^{-5} \text{ s}$ (solid black line) and $g_z(\tau)$ windowed between $\pm \tau_{max} = \pm 5 \times 10^{-4} \text{ s}$, corresponding to an accuracy $\varepsilon = 5 \times 10^{-3}$. (b) normalized impulse response function $h(t)$ (solid blue line) and \tilde{h}_q resampled at the discrete times \tilde{t}_q (red crosses).

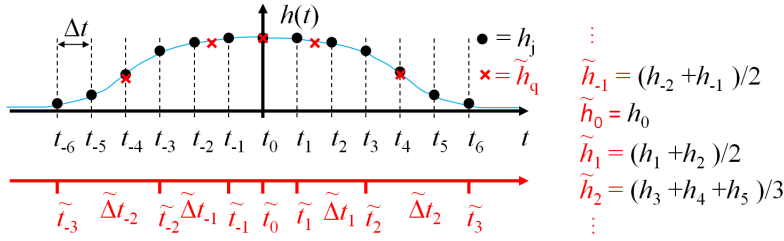


Fig. 7. (a): Sketch of the resampling procedure for the function $h(t)$. The original values h_j (black dots) sampled at the times t_j are replaced by the values \tilde{h}_q (red crosses) obtained by averaging h_j over increasingly wider intervals $\tilde{\Delta}t_q$ [Eq.(B.1)]. An example of the explicit expressions for the first few \tilde{h}_q values are reported on the right side of the figure.

(5) initialize the simulator by generating for each of the two components E_x and E_y of the electric field a number N_p of random y_i points drawn from a Gaussian distribution with standard deviation σ_y and average value $\langle y \rangle = 0$. Use these two data sets for loading two data buffers of N_p points each.

(6) for the component E_x (or E_y) of the field generate a new y_i random number (Gaussian distribution with σ_y and average value $\langle y \rangle = 0$) and update the buffer making it to work as a First-Input-First-Output (FIFO) buffer, where the new point is inserted into the first position, all the others are moved backwards by one position and the oldest point is discarded. Select the central point of the buffer and resample the y_i accordingly to the the same time grid defined at point 3. If $i_0 = i - j_{max}$ denotes the central index of the N_p points of the buffer, the resampled $\tilde{y}_{i_0,q}$ value is

$$\begin{cases} \tilde{y}_{i_0,q} = \frac{1}{\Delta n(q)} \sum_{j=i_0+n(q-1)+1}^{i_0+n(q)} y_j & q < 0 \\ \tilde{y}_{i_0,q} = y_{i_0} & q = 0 \\ \tilde{y}_{i_0,q} = \frac{1}{\Delta n(q)} \sum_{j=i_0+n(q)}^{i_0+n(q+1)-1} y_j & q > 0. \end{cases} \quad (\text{B.2})$$

A sketch of the resampling protocol for the points y_i is illustrated in Fig. 8.

(7) compute $(E_x)_i$ or $(E_y)_i$ as the convolution between y and h by using

$$z_i = \sum_{q=-q_{max}}^{q_{max}} \tilde{y}_{i_0,q} \tilde{h}_q \tilde{\Delta}t_q \quad (\text{B.3})$$

where z_i stands for $(E_x)_i$ or $(E_y)_i$. Equation (B.3) is the generalization of Eq. (13) carried out on the non-linear time-grid $\tilde{\Delta}t_q$. Notice that in Eq. (B.3) the sum extends over $\tilde{N}_p = 2q_{max} + 1$ points, a figure that is usually $\ll N_p$. Thus the use of Eq.(B.3) instead of Eq. (13) is rather convenient as far as concerns computational times (see discussion at the end of this Appendix).

(8) repeat steps 6 and 7 also for the E_y component of the electric field.

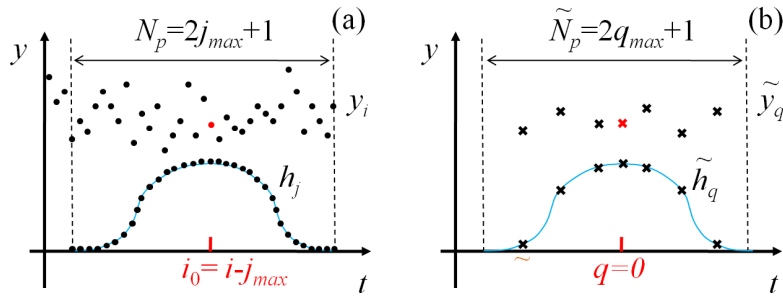


Fig. 8. Comparison between the convolution carried out by using Eq. (13) [panel(a)] and Eq.(B.3) [panel (b)]. Equation (13) uses a linear time grid and the generation of each new point z_i requires the computation of a very large number of terms, namely the the sum of $N_p = 2j_{max} + 1$ products. Conversely, Eq.(B.3) uses a pseudo-geometrical progression time-grid where the last N_p y_i points of panel (a) are resampled on the same time grid used for h (crosses), with y_{i_0} (red cross) being the central point with index $q = 0$. In this case the number of terms to be multiplied and summed is $\tilde{N}_p = 2q_{max} + 1$. Since typically $\tilde{N}_p = 100$, while N_p might be very large ($N_p \sim 10^4 - 10^5$), the use of Eq.(B.3) with respect to Eq. (13) is very convenient.

(9) calculate I_i according to Eq. (18) i.e. $I_i = (E_x^2)_i + (E_y^2)_i$.

(10) calculate $R_i = I_i \Delta t$ and pass R_i through a Poisson with a average count equal to R_i . For realizing a Poisson filter see, for example, [19]. The output of the Poisson filter is a integer count N_i (0, 1, 2, ...).

(11) if desired, introduce photodetector defects such as after pulse, dead time, dark count.

(12) if $N_i = 0$, no pulse is present. Return to point 6 for the next step.

(13) if $N_i = 1$, one pulse is present. Its arrival time is computed by counting the number of steps elapsed since the occurrence of the last pulse. The corresponding integrated arrival time, equal to the number of steps passed from the first one is recorded on the hard disk.

(14) if $N_i \geq 2$, two or more pulse are present (pile up effects). This case is handled exactly as point (13) and all the extra pulses are discarded.

(15) return to point (6) for the next step.

The accuracy of the overall procedure has been tested by comparing, for the same data of Fig. 6(a), the correlation functions of the intensity signal I_i the and corresponding pulse stream N_i with the theoretical curve $g_2(\tau) = 1 + |g_z(\tau)|^2$. In order to avoid spurious effects in the generation of the N_i 's, we did not introduce any photodetector defects and kept the count rate low enough ($\langle I \rangle = 10^5 \text{ Hz}$), which implies that pile-up effects are negligible at the clock $\Delta t = 50 \text{ ns}$. The averaging times were $T = 20 \text{ s}$ and $T = 100 \text{ s}$ for the I_i and N_i signals, respectively. The results, reported in Fig. 9(a), show an excellent matching between the two recovered correlations and the theoretical one. As expected, due to the shot noise introduced by the Poisson filter, the N_i -correlation is somewhat noisier than the I_i -correlation, but with no

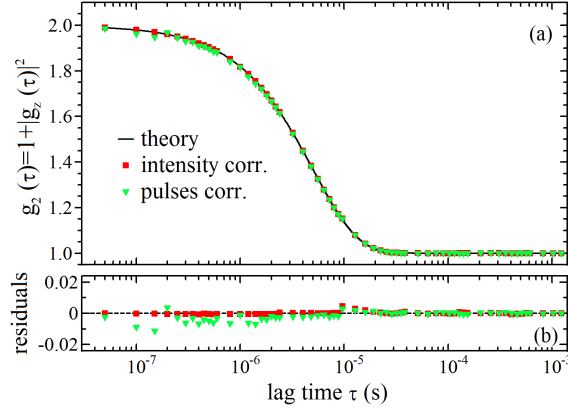


Fig. 9. Comparison between the theoretical intensity correlation $g_2(\tau) = 1 + |g_z(\tau)|^2$ (solid line) and the two correlations computed by correlating the intensity signal I_i and the corresponding pulse stream N_i . The theoretical $g_z(\tau)$ is the same as the one of Fig. 6(a). The averaging times were $T = 20s$ and $T = 100s$ for I_i and N_i signals, respectively. Both curves match excellently the theoretical one, with non systematic residuals that, because of the shot noise introduced by the Poisson filter, are higher in the case of the N_i -correlation.

systematic residuals, as shown in Fig. 9(b).

As final remark, we would like to discuss and emphasize the gain of computational times that Eq.(B.3) provides with respect to Eq. (13). This is particularly relevant when increasingly broader $g_z(t)$ functions are considered and smaller ε required. Indeed, the number of points required in Eq. (13) scales as $N_p \sim \alpha(\varepsilon)(\delta_{g_z}/\Delta t)$, where $\alpha(\varepsilon)$ is a decreasing function of ε that depends on the shape of the $g_z(t)$. For example, in the case of a g_z decaying as a single exponential with $\tau_c = 10^{-5}s$, with $\Delta t = 50ns$, we need $N_p \sim 2 \times 10^3$ points for attaining an accuracy of $\varepsilon = 5 \times 10^{-3}$. And as soon as $\tau_c = 10^{-4}$ or $10^{-3}s$, the number of points grows to $N_p \sim 2 \times 10^4$ or $\sim 2 \times 10^5$, figures that make the use of Eq. (13) fairly impractical. Conversely, with the use of Eq.(B.3), the number of points $\tilde{N}_p = 2q_{max} + 1$ can be lowered by orders of magnitudes ($\tilde{N}_p/N_p \leq 10^{-2}$) and the computational times correspondingly reduced, even though by not the same factor because of the time required for the resampling of y_i data (Eq.(B.2)). For example by using Eq.(B.3) (with $\tilde{N}_p = 100$, $\Delta t = 50ns$, $\varepsilon = 5 \times 10^{-3}$) pulse streams with a duration of $T = 1s$, characterized by single exponential decay with times $\tau_c = 10^{-4}s$ and $\tau_c = 10^{-3}s$, would take on our PC $\sim 0.4h$ and $\sim 4h$, respectively. Conversely, by using Eq. (13), they would take $\sim 1.1h$ and $\sim 19h$, increasing computational times by factors of ~ 3 and ~ 5 .

Acknowledgments

The authors, MM and FF, thank I. Guarneri for clarifying discussions about the theory behind this work. The research leading to these results has received access from LASERLAB-EUROPE (grant agreement no. 284464, EC's Seventh Framework Programme) and Fundacio CELLEX Barcelona.