



ULBS

Universitatea "Lucian Blaga" din Sibiu



Interdisciplinary Doctoral School

PhD Domain: Industrial Engineering

PhD Thesis

DEVELOPING AND IMPLEMENTING AN ALGORITHM AND
DEVICE FOR ASSESSING THE MICRO AND NANOPARTICLE
SIZE BASED ON DYNAMIC LIGHT SCATTERING PROCEDURE

- ABSTRACT -

PhD Student:

SILVIU-MIHAI REI

Scientific Advisor:

Prof. univ. dr. ing. DAN CHICEA

TABLE OF CONTENTS



TABLE OF CONTENTS.....	I
KEYWORDS.....	II
LIST OF PUBLISHED PAPERS AND OTHER CONTRIBUTIONS.....	III
ABSTRACT.....	VI
1. INTRODUCTION.....	VI
2. THEORETICAL FOUNDATION OF DYNAMIC LIGHT SCATTERING.....	IX
3. DEVELOPMENT OF A DATA ACQUISITION SYSTEM FOR DYNAMIC LIGHT SCATTERING MEASUREMENTS.....	XV
4. DEVELOPING THE ALGORITHM FOR DYNAMIC LIGHT SCATTERING DATA PROCESSING.....	XX
5. PRELIMINARY SIMULATION AND TESTING RESULTS.....	XXIII
6. EXPERIMENTAL AND PERFORMANCE RESULTS.....	XXVI
7. ALTERNATIVE DYNAMIC LIGHT SCATTERING ARTIFICIAL NEURAL NETWORK IMPLEMENTATION.....	XLVI
8. FURTHER DEVELOPMENT IDEAS.....	LI
9. CONCLUSION.....	LI
10. BIBLIOGRAPHY.....	LIV



KEYWORDS

Dynamic Light Scattering; artificial neural networks; particle sizing; nanoparticles; measurement; data acquisition; time series; signal processing.

LIST OF PUBLISHED PAPERS AND OTHER CONTRIBUTIONS

LIST OF PUBLISHED PAPERS

A. ISI

1. **Dan Chicea, Silviu Rei**, *A FAST ARTIFICIAL NEURAL NETWORK APPROACH FOR DYNAMIC LIGHT SCATTERING TIME SERIES PROCESSING*, Measurement Science and Technology, 29(2018) 105201 (15pp), IOP Publishing, <https://doi.org/10.1088/1361-6501/aad937> (Q2)

B. SCOPUS

1. **Silviu Rei**. *IMPLEMENTING A LOW COST DATA ACQUISITION SYSTEM FOR EDUCATION PROGRAMS IN UNIVERSITIES*. Balkan Region Conference on Engineering and Business Education 3 (1) (2017), 130-136.
2. **Dan Chicea, Silviu Rei**. *TIME SERIES SPACE PHASE QUALITATIVE ANALYSIS AND A POSSIBLE APPLICATION*. Academic Journal of Manufacturing Engineering . 2015, Vol. 13 Issue 2, p148-153. 6p. Paper also presented in the conference.

C. BDI

1. **Silviu Rei, Dan Chicea**. *USING DYNAMIC LIGHT SCATTERING EXPERIMENTAL SETUP AND NEURAL NETWORKS FOR PARTICLE SIZING*. ACTA Universitatis Cibiniensis 69 (1), 155-161. Paper also presented at the 3rd International Conference for Doctoral Students - IPC 2017 June 22-23, 2017.Braşov, Romania.
2. **Silviu Rei, Dan Chicea, Beriliu Ilie, Sorin Olaru**. *DYNAMIC LIGHT SCATTERING SIGNAL CONDITIONING FOR DATA PROCESSING*. ACTA Universitatis Cibiniensis 69 (1), 130-135. Paper also presented at the 3rd International Conference for Doctoral Students - IPC 2017 June 22-23, 2017.Braşov, Romania.
3. **Silviu Rei, Dan Chicea, Sorin Olaru**. *AUTOCORRELATION FUNCTION ANALYSIS IN PROCESSING STOCHASTIC TIME SERIES*. Annals of the University Dunarea de

Jos of Galati: Fascicle II, Mathematics, Physics, Theoretical Mechanics . 2016, Vol. 39 Issue 1, p57-60. 4p.

4. **Dan Chicea, Silviu Rei.** *USING THE FOURIER TRANSFORM AND THE POWER SPECTRAL DENSITY FUNCTIONS FOR PATTERN RECOGNITION IN DYNAMIC LIGHT SCATTERING TIME SERIES.* Proceedings of the Fourth International Conference: Modelling and Development of Intelligent Systems Sibiu, Romania. 2015. p33-39. Paper also presented in the conference.

D. CONFERENCE PROCEEDINGS

1. **Dan Chicea, Silviu Rei, Dan Leca.** *MONITORING YEAST CELLS SIZE DURING FERMENTATION USING DYNAMIC LIGHT SCATTERING.* 9th Central European Congress on Food (CEFood), 24-26 May 2018, Sibiu, Romania.
2. **Dan Chicea, Silviu Rei.** *A FAST ALGORITHM FOR DLS TIME SERIES PROCESSING FOR MONITORING THE SIZE OF THE SUSPENDED PARTICLES IN WASTEWATER.* 18th International Balkan Workshop on Applied Physics and Materials Science, Constanta, Romania 10-13 July 2018.
3. **Dan Chicea, Silviu Rei, Liana Chicea.** *A FAST ALGORITHM FOR PROFILING SUSPENSIONS IN NATURAL WATER USING DYNAMIC LIGHT SCATTERING AND ARTIFICIAL NEURAL NETWORK PROCEDURE.* Presented at AQUATIC BIODIVERSITY INTERNATIONAL CONFERENCE 2017, Sibiu.
4. **Dan Chicea, Silviu Rei.** *SIMPLE ALGORITHMS TO GENERATE DYNAMIC LIGHT SCATTERING TIME SERIA.* Proceedings of the 16th International Balkan Workshop on Applied Physics, Constanta, Romania, July 7-9, 2016.S2-P19, p96. Paper also presented in the conference.
5. **Dan Chicea, Silviu Rei.** *A NONCONVENTIONAL PROCEDURE FOR DLS TIME SERIES PROCESSING.* Proceedings of the 16th International Balkan Workshop on Applied Physics, Constanta, Romania, July 7-9, 2016.S2-P22, p97. Paper also presented in the conference.

OTHER CONTRIBUTIONS

1. **Important contribution to:** Dan Chicea. "Using neural networks for dynamic light scattering time series processing." *Measurement Science and Technology* 28.5 (2017): 055206.
2. **Minor contribution to:** Dan Chicea. "DYNAMIC LIGHT SCATTERING TIME SERIES GENERATION USING HARMONIC FUNCTIONS". Proceedings of 17th International Balkan Workshop on Applied Physics Constanța, Romania, July 11-14, 2017

RESEARCH GRANTS

- **JINR-RO 2018 04-4-1121-2015/2020 (JINR Dubna):** Advanced structural and magnetic investigations on FeCo/Al₂O₃ nanocomposites for novel soft magnetic materials for high frequency applications
- **JINR-RO 2018 04-4-1121-2015/2020 (JINR Dubna):** Co-based magnetic nanostructured material with potential space applications – synthesis and complex characterisation for novel soft magnetic materials for high frequency applications

ABSTRACT

The text of this chapter is based on the following publication:

Dan Chicea, Silviu Rei, A fast artificial neural network approach for dynamic light scattering time series processing, Measurement Science and Technology, 29(2018) 105201 (15pp), IOP Publishing, <https://doi.org/10.1088/1361-6501/aad937> [300].

1. INTRODUCTION

Dynamic Light Scattering (DLS) is an innovative experimental method for measurement of the properties of small particles (in ranges from 10^{-9} – 10^{-6} m) found in suspensions and colloids [1]. The method is typically used to determine particle hydrodynamic sizes but it could also be used for determining the speed of movement of the particles or to analyze the flow in fluids.

The technique involves shining an incident monochromatic coherent light beam onto a sample of fluid containing small particles which are of interest. Each particle absorbs and re-emits light and acts as a secondary light source. As the particles thermally move, the secondary light waves combine and create interference patterns. The scattering image which is observed in a far field is that of boiling speckles. The technique is measuring the intensity variation at a fixed point. The measurement is recorded into a time series which is used to extract the particle parameters of interest.

In the current work we will be mainly interested in extracting information regarding particle size by analyzing the time series obtained from a typical DLS experiment using a novel procedure using an artificial neural network. The method we considered as reference for DLS time series processing consists of fitting the analytical form of the Lorentzian line to the frequency spectrum of the recorded scattered light intensity. A set of artificial neural networks were designed, trained and tested. The training data consisted of a big set of autocorrelations of simulated time series for monodispersed spherical particles with diameter in the range 10 – 3000 μm . The artificial neural network output precision was tested both on simulated and on experimental time series recorded on fluids containing nanoparticles and microparticles

Particles, whether nanoparticles or microparticles, when suspended in a carrier fluid, have a complex motion consisting of both sedimentation and random, Brownian motion [17, 264]. When a light beam is incident on a fluid containing suspended particles, each particle scatters light and therefore becomes a secondary light source as elastic scattering occurs. We will name the particles scattering centers (SC). If the incident light beam is coherent, the scattered waves will be coherent as well, and, consequently, they will interfere in the far field. The consequence

of the complex motion of the SCs is the dynamic character of the far interference field, which appears as “boiling speckles”. Several articles report on the variation of several parameters, such as the average intensity, speckle size and speckle contrast, with the size and the concentration of the SCs, references [264, 265, 266] being some of them. But the physical method that exploits the correlation of the speckle dynamics with the Brownian motion is called Photon Correlation Spectroscopy (PCS) or Dynamic Light Scattering (DLS) and the theoretical basis of the method is explained in many studies, [1], [7], [33], [267], [268] being just some of them.

In the early days of DLS, photomultiplier tubes (PMT) were used as detectors. They had a chain of dynodes maintained at progressively bigger electric potential that produces a pulse of millions of electrons that is recorded. The PMTs have a fast response, linear over a broad range with a very good amplification and a narrow pulse with, in the range of nanoseconds [105]. Photodiodes (PD) were used later on with the improvement called avalanche photodiodes (APD). P-I-N diodes were introduced in DLS setup later on, as they improve the quantum efficiency of the device by replacing the depletion region with an intrinsic layer of semiconductor sandwiched between the P- and N- sides of the diode [105]. An autocorrelator was another typical part of the DLS setup, and contained the hardware to compute in real time the autocorrelation of the recorded signal.

As time passed, improvements in the DLS setup were done. A Laser diode can be used as coherent light source. A PD with a data acquisition system can be used to record the DLS time series on a PC, which can be used later on to process it, and examples of simple yet functional experimental setups can be found in papers like [105] and [269], among others. Certain improvement still can be done in time series processing, towards making the process faster and less computational intensive, aiming to turn DLS into an almost real time particle sizing procedure. Using artificial neural networks might be an alternative that can be used to accomplish this purpose.

An artificial neural network (ANN) is a computer implementation of a model that imitates the biological structure of a brain. The ANN can be described as being an object that has an output which depends on a set of input values. It uses artificial neurons which process information based on transfer functions. The neurons are interconnected and the connections are described using weights and biases. These complex interconnections enable the code of the

ANN to “learn”, that is to iteratively update the weights and the biases, using a large set of input values in such a manner to compute the expected output.

Once the ANN is trained, it can perform different tasks that emulate intelligent actions. Some of such actions are to compute the output of a function that has an unknown analytic form, based on the training of the ANN with sets of input data with known output. Pattern recognition, like faces and objects classification, image processing, sound filtering, general regression are some examples where ANNs are successfully used. The limits in using ANNs are imposed by the human imagination and skills in representing the problem correctly, within the limits of the universal approximation theorem [172, 173, 174]. This often leads to proper selecting of the type of ANN, of the training algorithm and training set of data. The ANN concept is not quite new and has been explained in detail in books like [178] and [270], among others.

ANNs have previously been considered for data processing in Physics and, in particular, in Optics. Reference [271] reports on using the polarized light signature in the shape of Mueller matrix as input for a ANN. The trained ANN was successfully used to detect amino acids and other solid organic compounds. Reference [272] reports on using ANN to retrieve the size and the refractive index angle-dependent light scattering measurements. Reference [273] presents a procedure for measuring the radius of spherical particles using the angular distribution of the

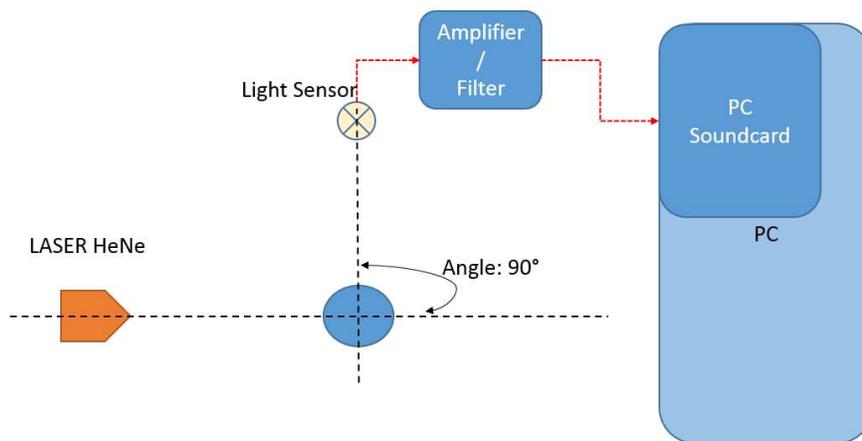


Figure 1. DLS Experimental Setup Example

scattered light and a three level ANN. In [274] the authors present a pattern recognition ANN approach in a flow cytometer type design together with the results in identifying the presence of hazardous fibers, like asbestos, in air. Reference [275] presents the results on assessing the

size and the refractive index of suspended particles from angle-dependent light scattering measurements using a radial function based ANN.

The work presented should be viewed as a step toward designing a miniature DLS particle sizer, stand alone type, that uses a very light computing platform, rather than a PC or a laptop computer.

The experimental setup used for testing the algorithm proposed, presented in Figure 1 consists of a He-Ne laser, 10 mW, working in continuous regime, a 5 ml diameter circular glass tube for sample, a detector, a preamplifier with a linear response in the audio frequency range and a PC.

2. THEORETICAL FOUNDATION OF DYNAMIC LIGHT SCATTERING

Due to Brownian motion the particles from the sample under investigation are constantly translating and rotating, which creates fluctuations in the dielectric constant of the medium. The suspended particles can be viewed as secondary light sources. If the incident light is coherent, the secondary light waves emitted by these sources are coherent, therefore they interfere, both in constructive and destructive way, leading to a “boiling speckles” aspect of the far interference field. The fluctuation of the intensity of scattered light in a point is the result of the change in phase of the scattered light. Basically, we can easily reason that the faster the particles move, the faster are the fluctuations of the intensity measured. Also, due to Brownian motion, the smaller particles move faster than the bigger particles, as it can easily be seen in the Einstein-Stokes relation [1].

$$D = \frac{k_B T}{3\pi\eta} \quad (1)$$

In (1) D is the diffusion coefficient which indicates how fast the particle diffuse, k_B is Boltzmann's constant, η the dynamic viscosity coefficient of the solvent, T the temperature of the sample and d the hydrodynamic diameter of the particle.

The approach on which we will rely as frame for our further ideas is described in detail in [1], [6], [104], [105]. The method uses the scattered intensity time series measured with a data acquisition system and then computes the power spectral density, which is the frequency spectrum of the experimentally recorded intensity of the scattered light. The power spectral

density is then fitted to the Lorentzian line, described by equation (2), using a nonlinear least square minimization procedure, and the parameters of the best fit are determined.

$$S(f) = a_0 \frac{a_1}{(2\pi f)^2 + a_1^2} \quad (2)$$

In this expression, f represents the frequency in the spectrum, and a_0 and a_1 are parameters. The parameter a_0 is scaling the shape of the function to reach the initial value. The parameter a_1 is directly related to the size of the particles, as of equation (3).

$$R = \frac{2\pi k_B T q^2}{6\pi\eta a_1} \quad (3)$$

In this equation (3), the variables have the following meaning: k_B – Boltzmann's constant, η – the dynamic viscosity coefficient of the solvent, T – temperature of the sample, q – the modulus of the scattering vector, as described by equation (4).

$$q = \frac{4\pi n}{\lambda} \sin\left(\frac{\theta}{2}\right) \quad (4)$$

In equation (4), n is the refractive index of the solvent, λ is the wavelength of the laser light in vacuum and θ is the scattering angle. The procedure involves finding the two parameters

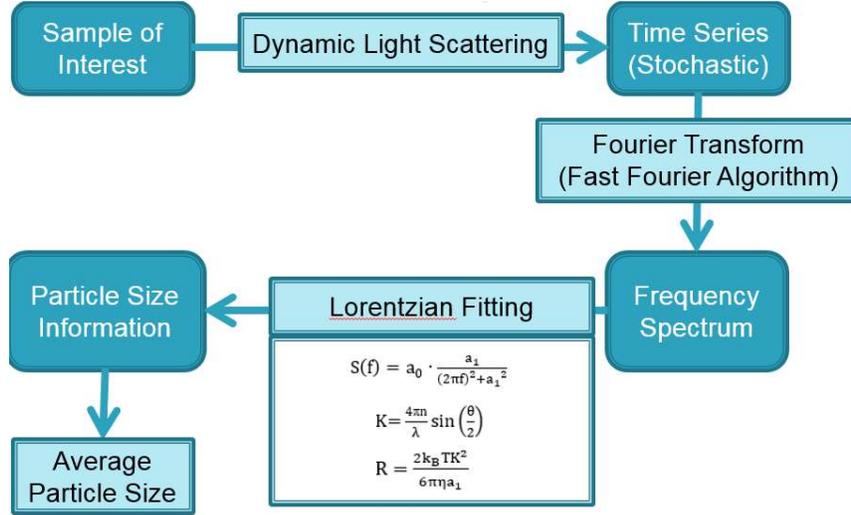


Figure 2. The flow chart of the reference DLS procedure

and then, based on parameter a_1 , estimating the average particle size using equation (3). This method was considered as reference in assessing the DLS diameters, hereafter reference DLS diameters. Figure 2 illustrates the flow chart of the reference DLS procedure.

The artificial neural network approach proposed by us involves computing the autocorrelation for the intensity of the scattered light intensity time series rather than the power spectrum. The method is based on recording the scattered light intensity $I(\theta, t)$, calculating the

autocorrelation for the recorded time series and then, using a numerical method for non-linear least squares fitting, determining the diffusion coefficient and therefore the particles average hydrodynamic diameter.

When we compare the intensity of the scattered light at two different moments, separated in time by τ , the values will most likely be different and uncorrelated. If the time τ , is comparable with the characteristic time of the fluctuations, the two values will be quite similar and most likely correlated. As this time is increased from zero to infinite, the correlation is decaying, from perfectly correlated to not correlated at all. As the fluctuations in the intensity are related to the particle sizes, namely, smaller particles generate faster fluctuations, the autocorrelation function will decay faster for smaller particles.

For a quantitative relation, there are various ways to extract this information from the time series. The method which we will consider as reference uses the time correlation function for the intensity, defined as:

$$G^{(2)}(\vec{q}, \tau) = \langle I_s(\vec{q}, t) I_s(\vec{q}, t + \tau) \rangle \equiv \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T I_s(\vec{q}, t) I_s(\vec{q}, t + \tau) dt \quad (5)$$

As we can see in the definition of the autocorrelation function, the intensity of light at time t , $I_s(\vec{q}, t)$, is compared with the same signal but delayed with time τ , $I_s(\vec{q}, t + \tau)$, for all time values t ranging from 0 to infinite. At zero time delay, the signal is perfectly correlated:

$$\lim_{\tau \rightarrow 0} \langle I_s(\vec{q}, 0) I_s(\vec{q}, \tau) \rangle = \langle I_s^2(\vec{q}) \rangle \quad (6)$$

At a delay time much bigger than the fluctuation characteristic time, $\tau \gg T_C$, the signal is not correlated at all

$$\lim_{\tau \rightarrow \infty} \langle I_s(\vec{q}, 0) I_s(\vec{q}, \tau) \rangle = \langle I_s(\vec{q}) \rangle^2 \quad (7)$$

This means that the autocorrelation decays from the mean-square value of the intensity to the square of the mean.

For practical reasons the normalized autocorrelation function is used:

$$g^{(2)}(q, \tau) \equiv \frac{\langle I_s(\vec{q}, 0) I_s(\vec{q}, \tau) \rangle}{\langle I_s(\vec{q}) \rangle^2} \quad (8)$$

A similar autocorrelation function can be calculated, but for the electrical field

$$g^{(1)}(q, \tau) \equiv \frac{\langle E_S(\vec{q}, 0) E_S(\vec{q}, \tau) \rangle}{\langle E_S(\vec{q}) \rangle^2} \quad (9)$$

The two autocorrelation functions are connected via the Siegert relation [278], which in its most general form can be expressed as:

$$g^{(2)}(q, \tau) = A \left(1 + \beta [g^{(1)}(q, \tau)]^2 \right) \quad (10)$$

Where β is a factor describing the spatial decoherence of the scattered light over a detector, determined by the ratio: speckle area – detector area, and having values between 0 and 1 and A determines the baseline. When the area of the detector matches the area of the speckle, the spacial decoherence factor is exactly 1.

To connect the above autocorrelation functions to the particle size, we need to consider not only the viscosity of the solvent and the temperature, but the diffusion coefficient, D , as well, which describes the Brownian motion of the particles. For a polydisperse system of particles, and considering the above, the electrical field autocorrelation can be derived as a distribution of exponentials

$$g^{(1)}(\tau) = \int_0^\infty G(\Gamma) e^{-\Gamma\tau} d\Gamma \quad (11)$$

Where $G(\Gamma)$ is the normalized distribution of exponentials and $\Gamma = q^2 D$, with q the scattering vector and D the diffusion coefficient. This last equation can also be expressed as a sum of exponentials, for a non-continuous distribution, as in equation (12):

$$g^{(1)}(\tau) = \sum_{n=1}^M A_n e^{-\Gamma_n \tau} \quad (12)$$

For true monodisperse systems, the autocorrelation of the electric field becomes a single exponential:

$$g^{(1)}(\tau) = e^{-\Gamma\tau} \quad (13)$$

In equation (12) the coefficients A_n describe the contributions of particles with different diffusion coefficients, hence different particle sizes, to the overall autocorrelation decay rate:

$$D_{0n} = \frac{\Gamma_n}{q^2} \quad (14)$$

As what we measure is intensity of the scattered light and not electrical field, we shall go back to the intensity autocorrelation, integrating all the above information. The intensity autocorrelation can be expressed as:

$$g^{(2)}(t) = 1 + \beta \sum_{i=1}^n a_i e^{-D_i q^2 t} \quad (15)$$

Or

$$g^{(2)}(t) = 1 + \beta \int_0^\infty A(D) e^{-D q^2 t} dD \quad (16)$$

For monodisperse solutions, the autocorrelation can be written as

$$g^{(2)}(t) = 1 + \beta e^{-Dq^2t} \quad (17)$$

We can extend the normalization procedure of the autocorrelation by subtracting 1 from the right part of equation (17) to simplify the shape and to use a simply decaying exponential function, which we actually did and used in this work.

The problem of evaluating the particle diameters can then be reduced to finding the parameters of the autocorrelation function, more precisely the diffusion coefficient D , which can then reveal the particle diameters using the Einstein Stokes. This can be done by a nonlinear least square minimization procedure, not used in this work.

The approach proposed relies on a much faster numerical method, which uses an ANN, which has as input the normalized autocorrelation function derived from the intensity signal and offers as output the average particle diameter, in one step.

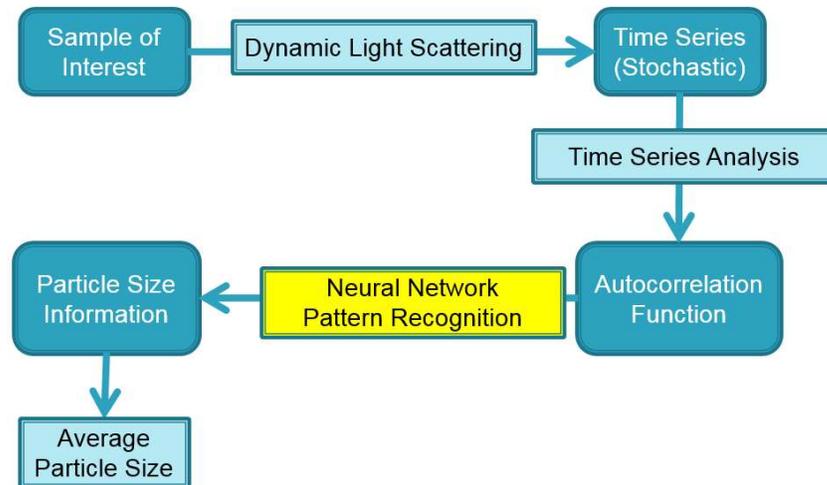


Figure 3. The flowchart of the ANN based DLS procedure

Figure 3 shows an overview of the novel procedure. The numerical fitting and the post fitting calculations were completely removed and replaced by an ANN, which does these actions in one step. One of the main reasons for such a replacement is to be able to miniaturize the computing system used for the calculations needed. The reference classical method requires numerical fitting and experience showed that the method needs human intervention, to adjust the fitting parameters, hence it is not fully automated, also it can be made so, at the expense of the computation time. The fitting itself generates a lot of floating point operations and even

more important, the fitting is required each time a diameter is evaluated. For these reasons the classical method requires a computational platform having enough power and memory, usually a PC. The ANN method requires indeed training of the ANN which also involves large number of computational operations, but this is done only once. As soon as the ANN reaches the mature level, as a result of training, it can be exported to a weaker computational platform, which is a key for miniaturization.

The ANN method is based on the universal approximation theorem [172], [173], [174]. This theorem states that a feed forward ANN with one hidden layer of finite size, can approximate continuous functions on compact subsets of R^n . If ϕ is a monotonic bounded non-constant continuous function, I_m the unit hypercube in m dimensions and $C(I_m)$ the space of continuous functions on I_m then, for any $\varepsilon > 0$ and any function $f \in C(I_m)$, there exists an integer N , constants $v_i, b_i \in R$ and vectors $w_i \in R^m$, with $i=1, \dots, N$ so that:

$$F(x) = \sum_{i=1}^N v_i \phi(w_i^T x + b_i) \quad (18)$$

is an approximation of the function f , where f is independent of ϕ , or:

$$|F(x) - f(x)| < \varepsilon \quad (19)$$

for all $x \in I_m$. This is true also when I_m is replaced with any compact subset or

This means that a ANN of the type described is an universal approximator. Cybenko [172], demonstrated that this theorem is true for sigmoid activation functions and Hornik [173], showed that the approximation power of the ANN is not related to the activation function but to the architecture of the ANN, which generalized the theorem from sigmoid activation functions to any activation functions.

3. DEVELOPMENT OF A DATA ACQUISITION SYSTEM FOR DYNAMIC LIGHT SCATTERING MEASUREMENTS

The problem of determining the particles size is equivalent to finding the correspondence between the set of ACRs and the correct average hydrodynamic diameter of the suspended

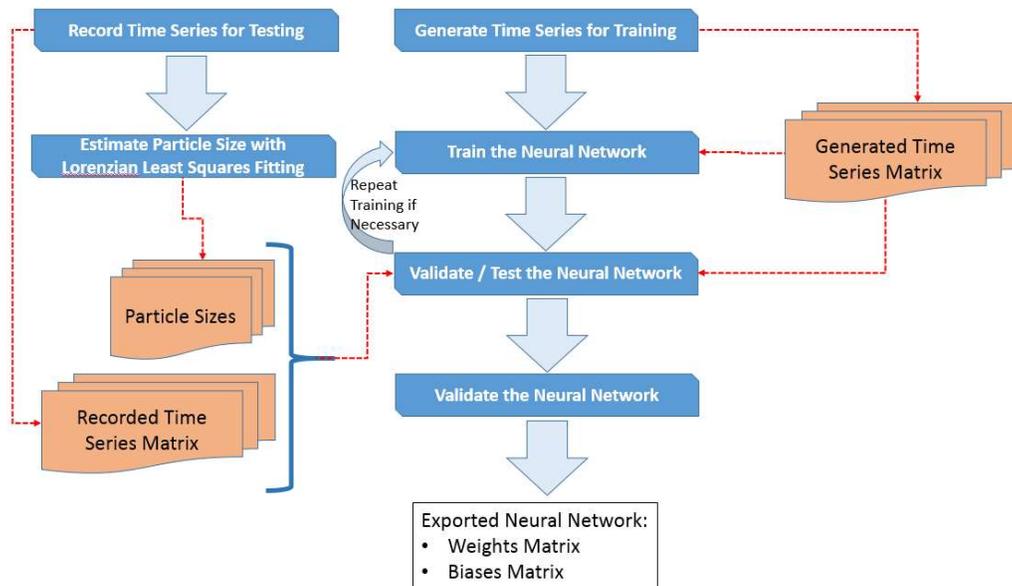


Figure 4. Neural Network Generation Procedure

particles. This is quite similar to fitting a mathematical function, which, we hypothesized that can be solved with a ANN with a proper architecture, transfer function and correctly prepared input and training vector. As with any problem that can be solved with NNs, the most important point is to correctly represent the problem in the ANN space.

As briefly mentioned before, the artificial ANN is based on a number of interconnected units called artificial neurons, structured in a certain architecture, usually in layers, and capable of communicating with some of the other artificial neurons. A ANN has several key components which together with its structure define the architecture of the ANN: a set of elementary units called neurons, connections between units, defined by connection weights and a set of functions for data processing. These functions can be a propagation function, an activation function and an output function. Also included in the ANN architecture is the learning strategy, used for adjusting the network to the problem

For finding the optimal ANN, we used an automated algorithm presented in Figure 4.

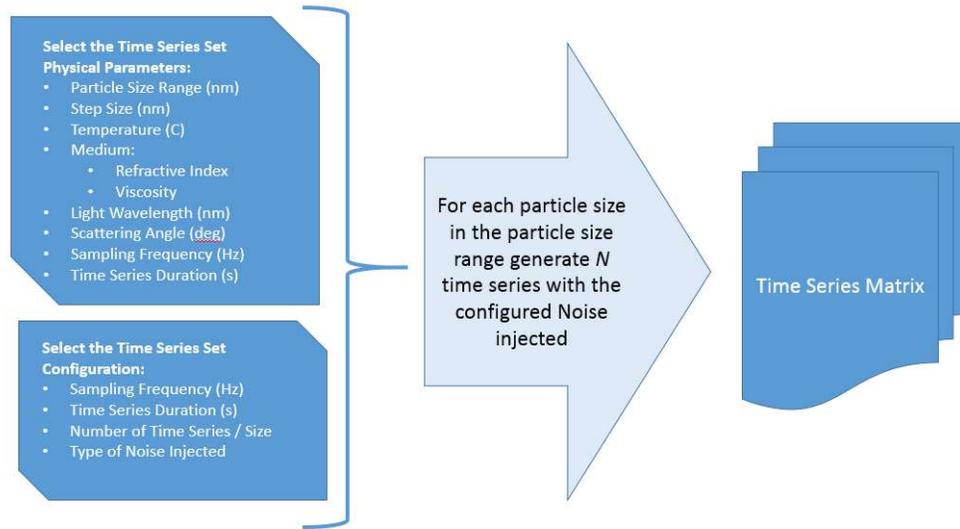


Figure 5. Time Series Matrix Generation Procedure Using the Harmonic Functions Method

The big set of autocorrelation sets used in training the ANN was computed on simulated time series that had both a 50 Hz and harmonics noise and a random noise added to it. This algorithm used in generating DLS time series, with noise, presented as an overview in Figure 5, is another novel aspect presented in this thesis being a simple procedure that uses deterministic functions (harmonic functions) to produce the output of a stochastic system.

Figure 6 presents the top view architecture of the miniaturized device mentioned.

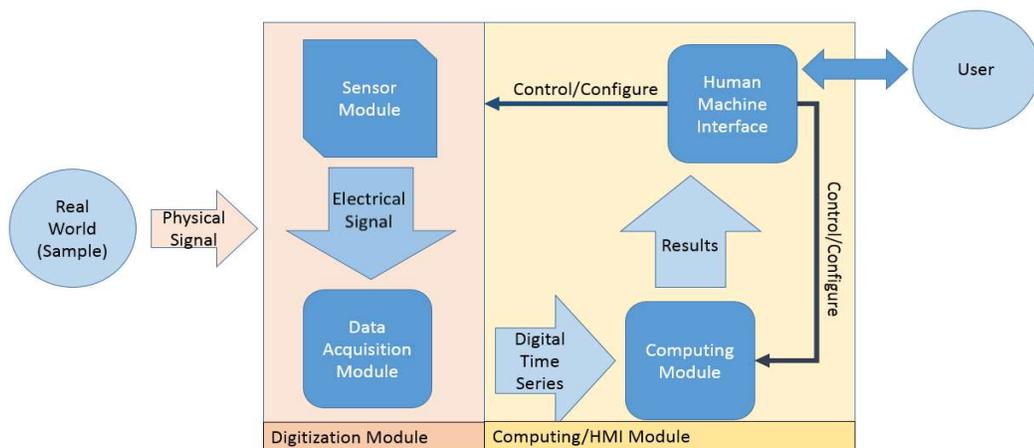


Figure 6. Abstract Top View of the System Architecture

The following modules are considered as part of the system:

- **The sensor module.** This module has the role of transforming the light intensity in a stable and accurate electrical signal, within the voltage ranges required by the next module.
- **The data acquisition module.** This module has the role of sampling and digitizing the electrical signal coming from the sensor module.
- **The computing module.** This module takes as input the digital time series and performs all the mathematical operations needed to obtain an estimation of particle size.
- **The user-machine interface.** This module allows the user to configure the system as well as to view the results of the estimation offered by the computing module.

Even though the Sensor Module and the Data Acquisition Module are independent, they will work together in what was named Digitization Module. Also, the Computing Module and the Human Machine Interface are not necessary one module, but they will work together in a module named Computing/HMI Module.

Going from the abstract architecture towards a more practical view, we shall have a look at the needed parts for the device to function correctly. A brief description of each module follows.

The sensor module is responsible for transforming the light intensity signal into an electrical signal. The output is an analog voltage. This module does a little bit more than just the measurement. It consists of a laser diode, a photo resistor, an amplifier and a cuvette to place the sample in. The laser diode, due to the objective of having the entire device miniaturized, is very small, similar to a laser pointer diode. The entire setup has to be placed in a dark chamber with black walls.

The sensor – data acquisition module shall be built in a way that would allow full flexibility of the device. Due to the complex interaction patterns of the parameters, as we can see in the analysis presented in detail in this thesis, the system shall be designed in a way that would allow the system to self-adapt and adjust to the various possible conditions. We implemented two layers of signal conditioning, one in the hardware of the system, implemented in the sensor module, and one in the software, more accurately called time series conditioning, presented in detail later.

The sensor module should have 4 sensing units, placed at different angles, and controllable from the control unit of the acquisition module. The sensor module should also have a laser and a cuvette for the sample. The design and geometry of the sensing unit is the result of the investigations in the previous chapters. Figure 7 shows the concept schematic of the entire discretization unit, sensor + data acquisition module.

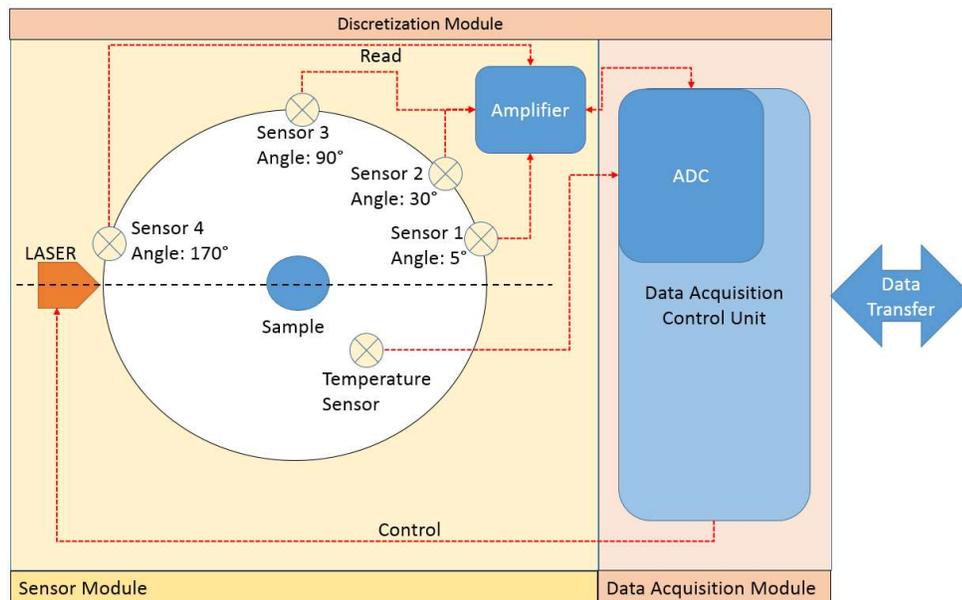


Figure 7. Concept Schematic for the Sensor - Data Acquisition Module

As the temperature plays a major role, affecting both the scattered light intensity and the roll-off frequency, we decided to design the system so that there is also a temperature sensor available. The sensor can provide the temperature in the dark room. The reading of the temperature can be done on request of the computing module. The computing module can use this value to perform some corrections on the estimated values for the particle size

The small room in which the sample is placed should be completely dark, to avoid any noise generated by environment light. In addition, the walls of the room should be painted in a dark, light absorbing paint, so that there are no parasitic reflections which reach the sensors.

The algorithm of the computing module will decide the optimum sensor based on several factors. If the first reading reveals a too low intensity, the system shall try again with a lower angle sensor. All the decisions are taken at the level of the computing platform. The data acquisition platform shall offer the software functionality of reading a selectable sensor for a configurable frequency and time series length.

What is important is that the signal is representing with high accuracy the light fluctuations resulting from the sample.

The input for this analog-to-digital module is the analog voltage signal. Using an analog-to-digital converter this module generates a series of numbers. These numbers are stored in the memory of the device or can be transferred to the computing module. For this process there are various possible options: integrating from the market a data acquisition module, implementing an own data acquisition module or adapting an existing system to work as a data acquisition module. We investigated the characteristics of a data acquisition module and we looked deeper into the option of implementing an own data such system.

The computing module processes the digital time series, performs various mathematical and computational operations and offers as output information about the particle sizes. The computing module also controls the data acquisition system. Here we also have several options which we can use. We investigated and implemented the following options:

- a Matlab implementation of a computing module, running on a PC,
- a Java implementation running on an Android device (smartphone, tablet),
- a Python implementation running on a small Linux computing platform (for example a Raspberry Pi with screen or headless)
- a web html/javascript implementation running in a browser, no matter if on the PC or on a smartphone.

4. DEVELOPING THE ALGORITHM FOR DYNAMIC LIGHT SCATTERING DATA PROCESSING

The algorithm proposed by this work allows a lot of flexibility which in the end leads to maximum portability of this solution.

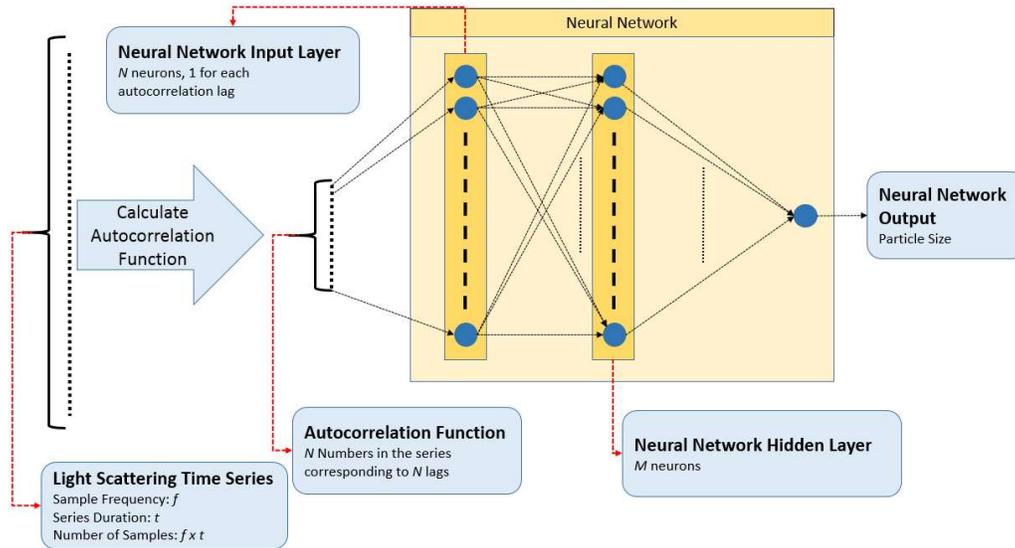


Figure 8. The Artificial Neural Network Approach Proposed for Estimating Particle Sizes

The core of the computing module is the ANN algorithm which will be extensively used by the end-user. Input is provided to the neural network in the form of a time series and then, based on the imported weights and biases, the network will estimate the particle size. The overview of the estimating engine is in this part, presented in Figure 8. In a first step the autocorrelation function is calculated numerically, with a preselected number of lags. Then each of the values of the autocorrelation lags is sent to one neuron from the input layer. The first lag to the first neuron, the second to the second and so on, but each time the same order of lag to the same neuron. The neural network will offer as output the average estimated hydrodynamic particle diameter.

This part is implemented in the computing module together with an additional support part which has the purpose to manage the data acquisition system. This support and acquisition management part provides the feature of software signal conditioning of the raw time series offered as input to the neural network, but also manages dynamically the behavior of the data acquisition system. The main idea of this part is to provide the right readings and a properly prepared time series to the right neural network. Because we target measurements of a wide range, both in air and in water, and as we have seen the scattered light varies widely in these

target ranges, both in intensity and in frequency spectrum, we need to reduce the variation of the input. Also, for certain combinations medium-particle size, different measurement angles need to be used. For example the measurements of particles in water at an angle of less than 30 degrees is not really possible with 633 nm light while on the other hand for air, we need angles

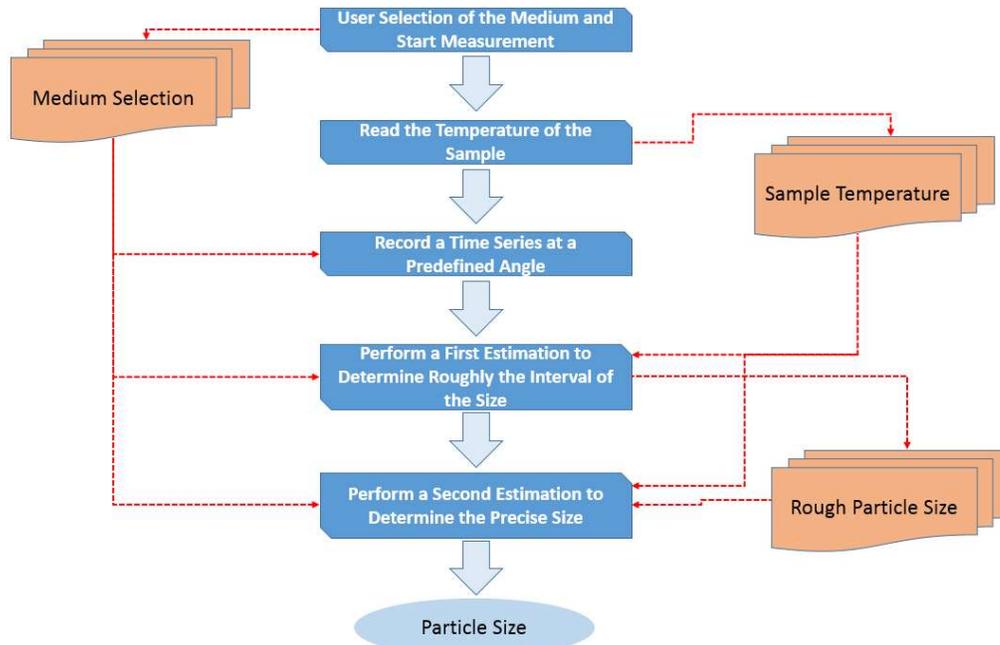


Figure 9. The Data Acquisition Management Algorithm

of less than 30 degrees to measure particles with the same light type. Also, even for the same medium, it might be favorable to select a lower or higher angle based on where in the particle range is the particle size placed. When we have a low angle we have higher intensities, allowing us to measure light scattered from smaller particles, but for particles in water for example, we also have the frequencies in the spectrum more close to each other for different sizes, making it harder to distinguish close particle sizes. Even more, we can implement a plausibility check by measuring at 2 different angles at the same time and sending the data to two parallel networks, cross-checking the answer to verify if the response is correct.

This is the reason why we designed the sensor module with 4 sensors and we need to implement an intelligent strategy for reading the sensors so that we can get the maximum precision from the neural network. The idea of the control algorithm is implemented as follows. First the user selects a medium, water or air then requests via the interface a measurement start. The computing module will request a temperature reading to the data acquisition. It will store the value received for later use. Then the computing module will request a time series recording from one of the sensors. The selection of the sensor, and therefore of the measurement angle,

is related to the medium selection. This first reading has the purpose to offer data for the system to evaluate roughly in which part of the size interval is the particle size placed. With this time series, a first neural network, different for different medium, will estimate the rough size. The temperature reading is also used as input.

Based on the particle size in the interval, the computing module will request an additional time series reading, for a more precise determination. Again, the most suited angle will be selected, based on the medium and particle size in the range. The time series recorded will be processed with another neural network, specialized for that specific angle and range. For better precision, we could have tens of neural networks implemented. For example for a size range of 1-1000 nm we could have 10 neural networks for each angle, each of them for a different range: 1-100, 100-200, 900-1000 nm. This would allow a more precise determination as the network is specialized for those conditions. The general neural network will be trained for the entire size range and function as a first assessor.

The Human Machine Interface depends on the selection of the computing module, running together with it. It can either be the Matlab standard interface (for the Matlab implementation of the computing platform) or it can be an Android application, a web page or a touchscreen interface (for the standalone computing platform Raspberry Pi).

The solution proposed can run in the following combinations:

- A PC running Matlab, with a soundcard (or an USB soundcard) used as data acquisition module and with the sensor module connected to the soundcard
- A PC running Matlab, with the sensor connected to the own implemented data acquisition system, connected to the USB port of the PC
- A small, standalone device, with touchscreen, containing all the modules in one
- A small, standalone device for data collecting and storing, which can later be connected to the USB port of the PC for downloading of data and analysis
- The sensor and the data acquisition system connected to the USB of the PC for computing in a web page
- The sensor and the data acquisition system connected via Bluetooth to either a smartphone, tablet or PC, each with a application allowing computation of results.

5. PRELIMINARY SIMULATION AND TESTING RESULTS

We performed a detailed analysis on how should the data acquisition module be constructed together with details about how the computing platform should be designed. As a result of this analysis, we can conclude that, from intensity of scattered light point of view, we can be able to measure particles in the range 5-1500 nm, with the correct choices in the configuration of the system. From the list of parameters listed in the previous paragraph, we can control: the scattering angle, the light intensity, the light wavelength and the distance between sensor and scattering volume, and sometimes we can control also the concentration of the sample. As we have seen, one specific major problem for a miniaturized system is to detect the small intensity variations for the low end range of sizes. To maximize the scattered intensity we have the following possibilities:

- Using a low scattering angle: this makes sense judging only from the intensity point of view, but extra-care should be taken here as the scattering angle influences also the roll-off frequency in the spectrum, as we will see in the next chapter. For particles in water, using a low scattering angle is not optimal in all situations.
- Using a small distance sensor-scattering volume
- Increasing the initial light intensity by using a high power laser: this should also be used with extra care, as the laser power affects directly the temperature of the sample. The higher the laser power, the higher the temperature increase induced in the sample. The temperature, as we have seen, affects both the light intensity through the refractive index, but will also affect the roll-off frequency, which will lead to very imprecise measurements. Ideally, the laser should be as low power as possible, so that the heating effects are reduced to a minimum. The problem of studying the heating effects of laser on nanoparticle solutions was already addressed in several papers such as [148] [149][150][151] and, as the topic is out of scope for us, we will not place details here.
- Using a low wavelength for the light: this parameter affects also the roll-off frequency and we, again, need to be careful when adjusting it. However, this parameter will not be adjustable in the final system, as that would mean to have various laser types, which is considered impractical.
- Increasing the concentration: this may or may not be possible, depending on the measurements conditions. For example for real-time measurements on the field, for

particles in air for example, it will be difficult to adjust the concentration. Also, most of the time, only dilution is possible.

- An additional idea to increase the size range is to use an analog sensor and to amplify the signal from the sensor with a performant amplifier. Of course this can work only if the sensor is still able to detect low light variations.

But, none of these rules can be applied blindly because, as we have seen when analyzing each of them, they all have influence on other key parameters than just the scattered intensity, and even the influence on the scattering intensity is quite complex.

We found that it is realistic to target the following ranges for our own designed system:

- Medium: air. Range: 5 - 1000 nm
- Medium: water. Range: 50 – 1500 nm

To achieve the proposed ranges, we need to use a smart combination of parameters, in a dynamical way, in order to be able to assess the selected range of particle sizes, both for water and air solvent. Figure 10 shows the general picture of the dependencies between physical parameters and data acquisition performance parameters.

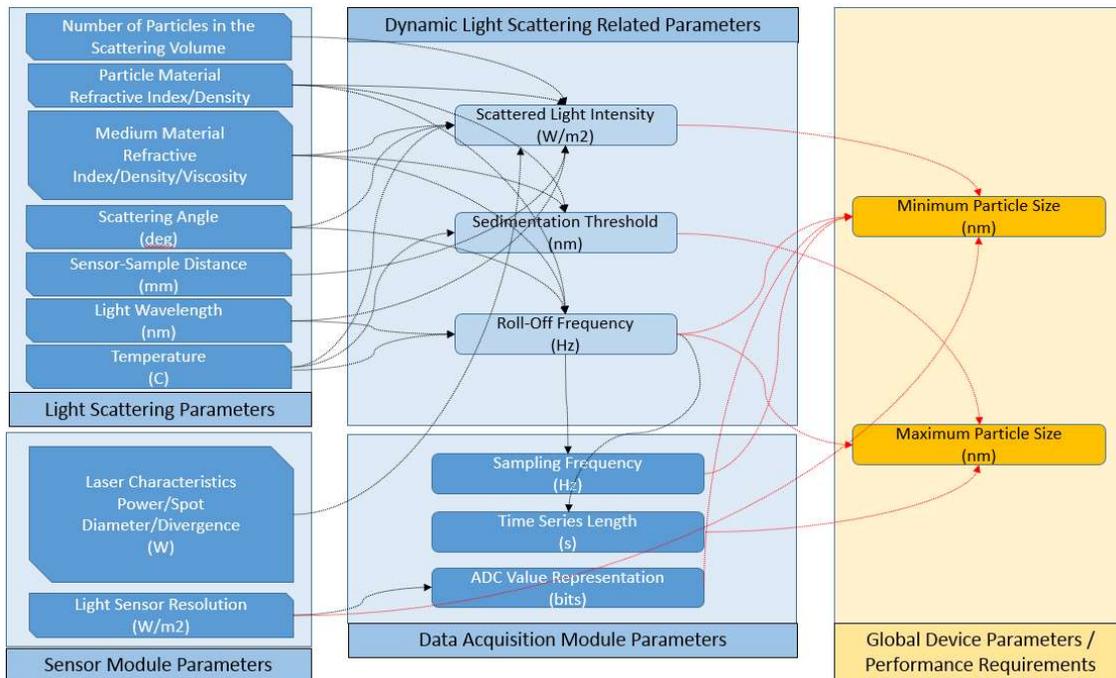


Figure 10. Light Scattering, Sensor, Data Acquisition and Global Device Parameters and Connections between Them

As the interactions are quite complex, we cannot state a clear value for all the parameters. We performed an extensive and detailed analysis on the impact of each of these parameters

on our system. One of the purposes of this exercise was to find out first if it is possible and realistic to design a system for the medium and size range targeted, and the answer is yes, we consider it is possible. The other purpose was to picture and realize the connections between variables.

We summarize the main parameters of the data acquisition system in Table 1.

Data Acquisition Parameters			Global Device Parameters		
Data Acquisition Frequency (Hz)	Time Series Maximum Length (seconds)	ADC Representation (bits)	Minimum Particle Size (nm)	Maximum Particle Size (nm)	Medium
16000	4	16	50	1500	Water
16000	4	16	5	1000	Air

Table 1. Device Parameters Overview

After a review of the objectives and of the theoretical foundation for signal processing, we identified the key physical parameters which influence the design of the system. We investigated each parameter in detail and evaluated what should be the target performance of our acquisition system. We reached the following performance requirements:

- Particle size range:
 - In water: 50 – 1500 nm
 - In air: 5 – 1000 nm
- Acquisition Frequency: 16000 Hz
- Acquisition Time: 2 seconds
- Acquisition Angle: 5, 30, 90, 175 degrees
- Laser Power: 0.05 – 10 W
- Sensor Resolution: 0.001 W/m²

We also concluded that there is no single universal solution, therefore we need to design an adaptable system, capable of switching between various configurations.

After all the executed tests and simulations for the ANN algorithm, we established that we will use a $m \times 100 - n - 1$ neural network with Fletcher-Power Conjugate Gradient training algorithm in the next steps of tests. However, these is just a rough indication on the configuration of the network, as the actual network configuration needs to be fine tuned to be optimal for the problem to be solved. For this configuration we will use the neural network hidden layer size tool for 2 main time series training:

- Particle size range: 1-1-1000 nm, scattering angle: 4° , wavelength: 532 nm, temperature: 20C, sampling frequency: 16 kHz, time series length: 2s
- Particle size range: 1-1-1500 nm, scattering angle: 90° , wavelength: 633 nm, temperature: 20C, sampling frequency: 16 kHz, time series length: 2s

6. EXPERIMENTAL AND PERFORMANCE RESULTS

We will describe in the next paragraphs in a first step the procedure we used to generate the time series used for training the ANN. In order to use a ANN for solving a particular problem, the ANN structure has to be trained for it [178], [270]. Training requires big amount of data, consisting of sets of inputs with known outputs. The input to a ANN cannot be the time series, which can have different lengths and look totally different from each other, yet being produced by the same SCs. The work in [276] used NNs trained with the frequency spectrum of the intensity of the scattered light, also known as the power spectrum and the details will not be repeated here. In the work described in this paper we will use the autocorrelation of the time series, which, as previously shown, has the same unique shape for SCs of a certain diameter, regardless the length of the time series, should this length be bigger than a minimum value and this aspect is presented in this subsection.

Using recorded experimental DLS time series is not a solution, because the typical latex balls have a certain error in manufacturing them for a particular diameter. Moreover, they cannot be manufactured with a diameter step of 1 nm with a good precision, which is the diameter step we used in generating the inputs.

The alternative was to simulate DLS time series with the desired diameter, frequency, scattering angle and number of data points. The time series generation algorithm is actually an improvement of the algorithm used in [276]. The fast Fourier transform algorithm [101], [280]

can be used to compute the absolute value of the frequency spectrum of the light intensity time series, also known as the power spectrum (PS). We can compare the spectrum calculated from the experimental data with the theoretically expected spectrum, which is described by the Lorentzian line $S(f)$ (2) [7], [281], and this is used in the reference DLS.

The challenge we met at this stage was to produce the outcome of a stochastic system, which is the time series produced by coherent light scattered by a suspension, using a totally deterministic procedure. Such a simple procedure consists of summing harmonic functions, like sine or cosine, having different frequencies and amplitudes, at different times, to compose the time series. Equation (20) describes the sum:

$$x(t) = \sum_{i=1}^{N_f} A(f_i) \cdot \sin(2\pi f_i t + \varphi_i) \quad (20)$$

In (20), $A(f_i)$ is the amplitude of the i -th component, f_i is the frequency of the i -th component, φ_i is the initial phase of the i -th component, t is the time when we compute that particular data in the DLS time series and N_f is the number of frequencies used in generating the time series. The a_0 is selected to be a fixed value, of the order of tens, the same for all series, while is calculated from equation (3) for each particle diameter, hence radius R . $A(f_i)$ was selected to be the square root of $S(f_i)$ computed with equation (2).

All the φ_i initial phases were generated using random numbers with uniform distribution in $[0, 2\pi]$. The f_i values were generated equally spaced in the interval $[0, f_s/2]$, where f_s is the sampling rate of the data acquisition system supposed to have recorded the time series, sampling rate hereafter. The reason for not using frequencies bigger than $f_s/2$ is explained by the Whittaker–Nyquist–Kotelnikov–Shannon theorem [58, 282, 283, 284, 285], which in Shanon’s version states: “If a function $x(t)$ contains no frequencies higher than f hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2f)$ seconds apart” [101, 280]. Moreover, having a time series recorded or generated with a sampling rate f_s , the fast Fourier transform will produce a frequency spectrum in the interval $[0, f_s]$ that is symmetrical to $f_s/2$ [101, 280]. The amplitude $A(f_i)$ of the i -th component is determined using the square root of $S(f_i)$.

For a best use of the computation resource during training data generation process, we investigated the influence of the number of frequencies N_f and of the number N_t of data contained by the time series, on the precision of assessing the diameter. We concluded that the best result for a generating a time series with $N_t=2^n$ data is obtained when the time series is generated using a minimum number of $N_f=2^{n-1} + 1$ frequencies. Also, the time series that can

be used in training the ANN should have a number of data $N_t=2^n$ where n should be minimum 13. For a time series having $N_t=2^n$ a number of frequencies $N_f > 2^{2n-1}+1$ should be used, in order to get an accurate diameter when computing it with the reference DLS algorithm.

For the purpose of solving the problem of determining the particle size with a ANN, we used a supervised training model. We will describe below in more detail the steps we performed for training the ANN.

First we generated the set of time series for selected diameters. We selected $n=15$, therefore each time series had 32768 data points and a number of 16385 frequencies were used in generating each of them. The range of diameters was 10 – 1200 nm and 20 time series were generated for each diameter. The step for increasing the diameter was 1 nm. Random numbers with uniform distribution in the $[0 - 2\pi]$ were used in generating the phase φ_i of each harmonic component, which assured that the series generated for the same diameter were not identical and corresponded to slightly different diameters when computed with the reference DLS procedure. The angle was chosen to be 90° and the data acquisition frequency 16000 Hz.

Moreover, as the electric power grid was operated at 50 Hz, we added noise to the generated time series, consisting of a sum of sine functions, as in equation (20), having frequencies of type $50 \cdot i$, i being a natural number in the range $1 - i_{max}$. i_{max} is the sampling frequency divided by 50, therefore it is the maximum frequency of the 50 Hz harmonic that is smaller than f_s . The amplitudes of the 50 Hz noise harmonics decreased exponentially with the number of the harmonic, as in equation (21), where A_{ts} is the amplitude of the time series, assessed as the difference between the maximum and the minimum of the time series and i is the harmonic number. The initial phases of the harmonics φ_i were generated using random numbers with uniform distribution in $[0, 2\pi]$. The power grid noise time series x_h was computed apart from the time series and added to it.

$$x_h(t) = \sum_{i=1}^{i_{max}} 0.03 \cdot A_{ts} \cdot \exp(-0.25 \cdot i) \cdot \sin(2\pi f_i t + \varphi_i) \quad (21)$$

In addition to the 50 Hz noise, random noise was also considered. The random noise time series, x_n , was computed apart from the time series using equation (22), where N_{rnd} is the number of frequencies f_i generated using random numbers with uniform distribution in the range $[1 - f_s]$. For the work reported here N_{rnd} was selected to be 300.

$$x_n(t) = \sum_{i=1}^{N_{rnd}} 0.01 \cdot A_{ts} \cdot \exp(-0.005 \cdot i) \cdot \sin(2\pi f_i t + \varphi_i) \quad (22)$$

After computing the series with the power grid noise and with the random noise, these series were added to the generated time series to produce the (relatively) noise time series. The

particular values of the coefficients in equations (21) and (22) were selected after a trial and error process of comparing the frequency spectrum of the generated time series with the frequency spectrum of the experimentally recorded DLS time series. We must mention at this stage that this feature of adding noise in data used for training NNs is crucial for producing a ANN that can handle experimental time series as input, as they contain noise.

The second step consists of computing the ACR of each time series. This was achieved

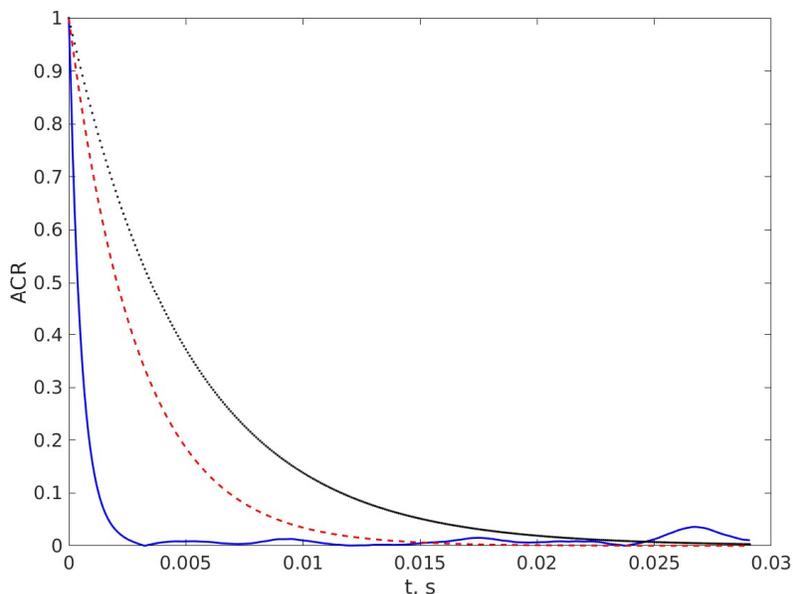


Figure 11. The ACR of three computed time series with noise added, for diameters of 100 nm the blue continuous line, 700 nm the red dashed line, and 1200 nm the black dots

using the Matlab function *autocorr*. The ACR of each time series was computed on 350 lags and loaded as a column in an array with 350 lines and as many columns as the time series. The targets were the diameters computed for each time series using the reference DLS procedure. Figure 11 illustrates the ACR of three computed time series with noise added, as described above, for diameters of 100, 700 and 1200 nm.

Examining Figure 11 we notice the effect of the added noise as a distortion from the ideal exponentially decaying form. The distortion is evident on the ACR of the time series generated for smaller diameters, as the ACR gets closer to the lags axis faster than the ACR of time series generated for bigger diameters.

The Matlab package was used in generating and training the ANN which was a ANN for fitting type, having three layers, as previously described. The first (input) layer had 350 neurons,

the hidden layer had 26 neurons and the third (output) layer had one neuron, as the output is the average diameter. The training algorithm was Levenberg-Marquard [286]. Training stopped after 58 iterations, as the regression coefficient R value reached 1. Training lasted for 38 minutes on a laptop having an Intel I7 7300 processor. 70% of the data sets was used for training, 15% for testing and 15% for validation.

A Matlab function was generated at the end of the training process, having the arrays with the biases and weights saved in the code, which makes the function portable to other platforms, like Octave and allows an easy translation to other programming languages.

In order to compute the diameters using this function, the ACR on 350 lags of each time series is computed first and placed as a column in an array. Running the function involves matrix multiplications and addition, which is less computing intensive than fitting a function to data using a nonlinear minimization procedure for the χ^2 , which involves a big number of function evaluations.

After implementation, the algorithm and approach proposed was tested in detail both on simulated and experimental data. We will briefly review some of the results here.

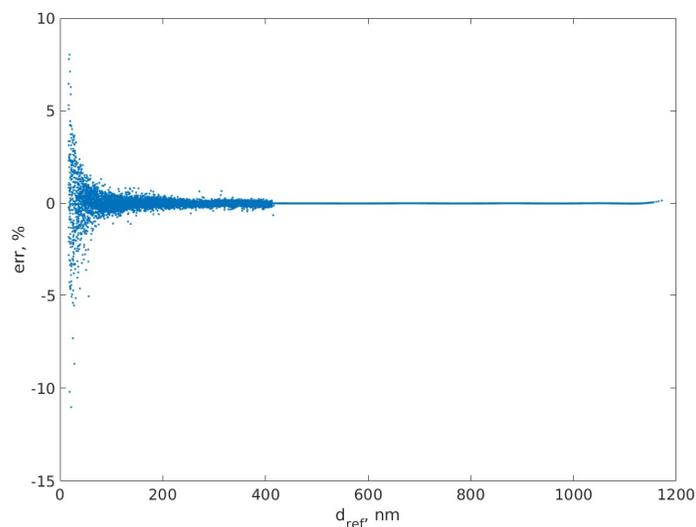


Figure 12. The relative errors versus the reference diameter

First, we tested the approach using simulated data. The ANN was tested on the simulated data that was used for training it, more precisely on the whole set of data, not only on the 70% of data chosen for training. The diameters that were used as targets were computed with the reference DLS method. We named these diameters reference diameters, d_{ref} . These reference diameters were compared with the diameters computed with the ANN, hereafter ANN

diameters, d_{NN} . Plotting the ANN diameters versus the reference diameters they will reveal a straight line, if observed by eye, therefore we will not present it here. More relevant is an analysis of the relative errors, err , expressed in %, defined by equation (23).

$$err = \frac{d_{NN} - d_{ref}}{d_{ref}} \cdot 100(\%) \quad (23)$$

A plot of the relative errors versus the reference diameter is presented in Figure 12.

Figure 12 reveals that the relative errors are bigger in the diameter range smaller than 500 nm and are negligible for diameters bigger than this value. Looking deeper, we notice that the

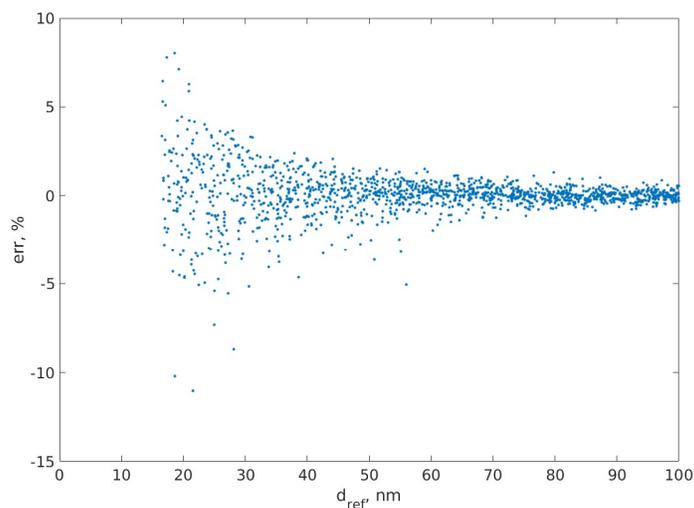


Figure 13. The relative errors versus the reference diameter – the small diameter range

errors are quite big in the very small diameter range. A possible explanation is related to the added noise in the generated data, which distorts the ACR of the time series generated for small diameters much more than the time series generated for bigger diameters. Zooming in Figure 12 in the small diameter range we find Figure 13. We notice that for diameters bigger than 40 nm the relative error is smaller than 3% and decreases fast with the diameter, therefore we can conclude that the ANN approach for DLS time series processing is accurate for diameters in the range [40 – 1200] nm.

The next set of tests we performed on experimental data. For the reference, non ANN procedure, we used a data set measured on samples containing yeast, for the ANN approach we used three data sets: clay, milk (proteins) and waste water.

The first experimental data set was using the reference procedure which evaluates the particle sizes by a non-linear least squares fitting and not the artificial neural network approach.

The purpose of this is to present an example which illustrates with experimental measurements the reference procedure.

To obtain the particles suspension, small amount of solid “baker’s yeast” (*Saccharomyces cerevisiae*) was dissolved into water. A concentrated brown sugar solution (15% weight in water) was created. A yeast suspension with volume of 0.2 cm³ was added to 3.5 cm³ of syrup and placed in a cuvette in the path of a laser beam (632 nm). The sample temperature was 20 degrees C. The intensity of light was monitored and recorded, creating time

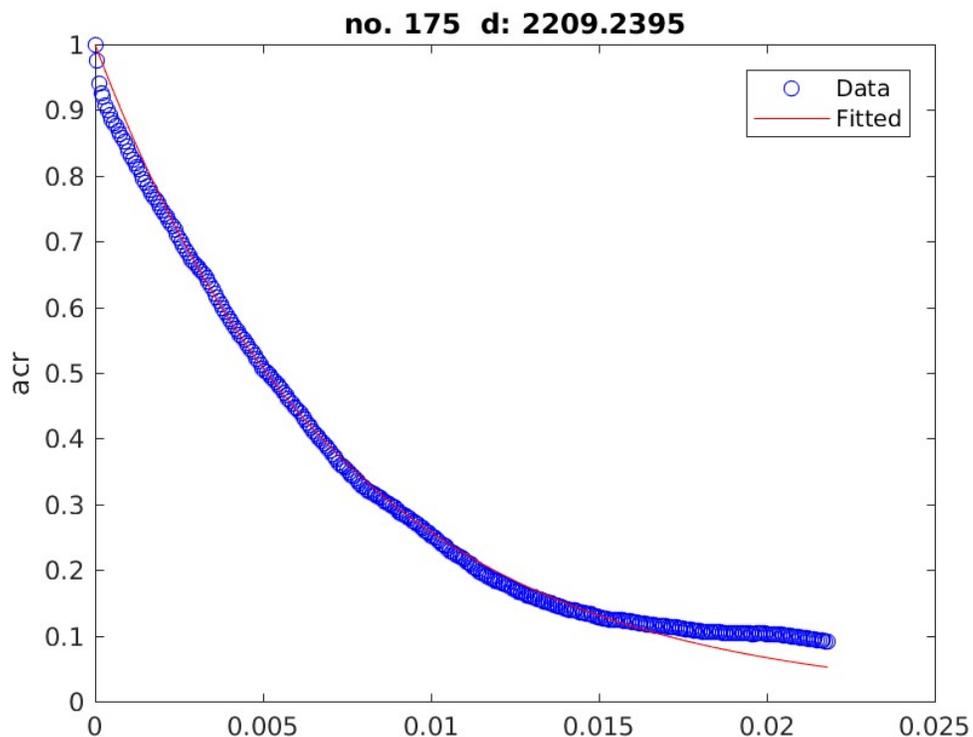


Figure 14. Autocorrelation function data (dotted) and non-linear exponential fitting for the 175th time series recorded (estimated particle size 2209.2 nm)

series of 8 seconds duration. Time series were recorded each 0.5 hours for about 160 hours, roughly 1 week.

The average size of the yeast cells was estimated using the reference Dynamic Light Scattering procedure. Figure 14 and Figure 15 show the autocorrelation decay and the result of the non-linear least squares fitting for two distinct measurements during the 1 week recording. We can observe in the examples that the experimentally observed decay is fitted quite well by the non-linear algorithm. We also can see in the examples the particle sizes estimated using the reference method.

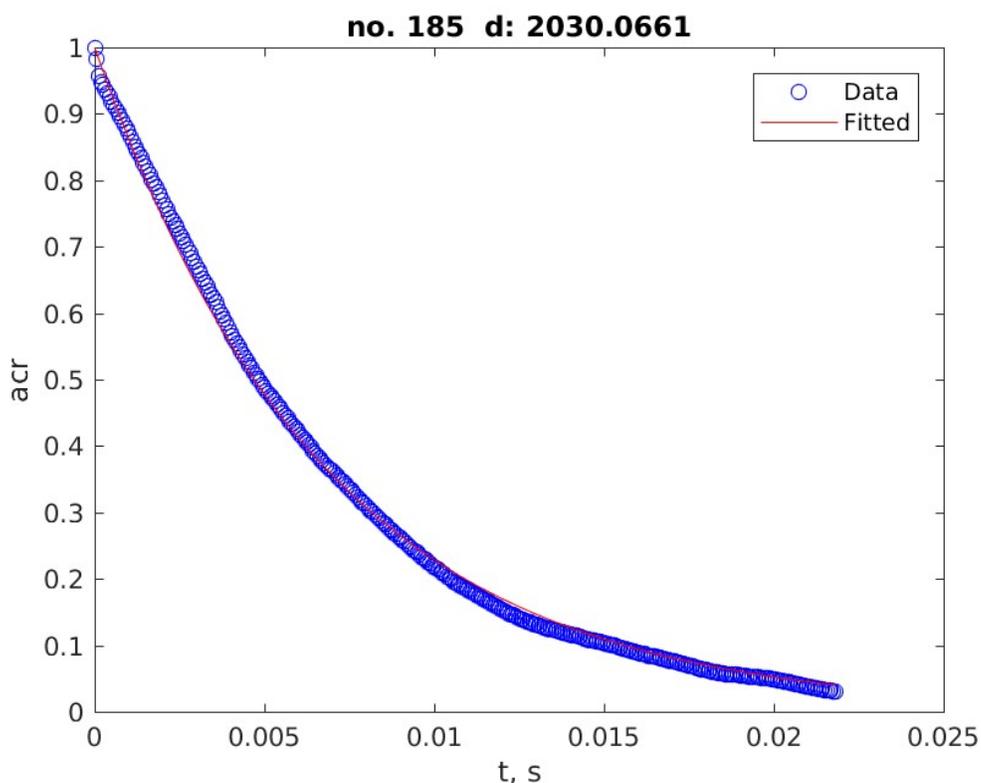


Figure 15. Autocorrelation function data (dotted) and non-linear exponential fitting for the 185th time series recorded (estimated particle size 2030.1 nm)

Figure 16 shows the evolution of the average yeast cell size during 1 week. We can observe that in the first days there is a growth of particle size, from around 600-700 nm to a maximum of around 2600-2700 nm. After that peak, the average yeast cell size starts to decrease slowly for the remaining period. The possible explanation of this evolution is the following: during the first 3 days, yeast cells are growing, as they can feed on the sugar solution in which they are placed. When the sugar solution gets depleted, the yeast cells stop growing but also start dying. The decrease in size is caused by the disintegration of the yeast cells and their death.

We can also notice that the yeast cell size estimated is consistent with the sizes for bacteria and yeast: 1000-5000 nm [291].

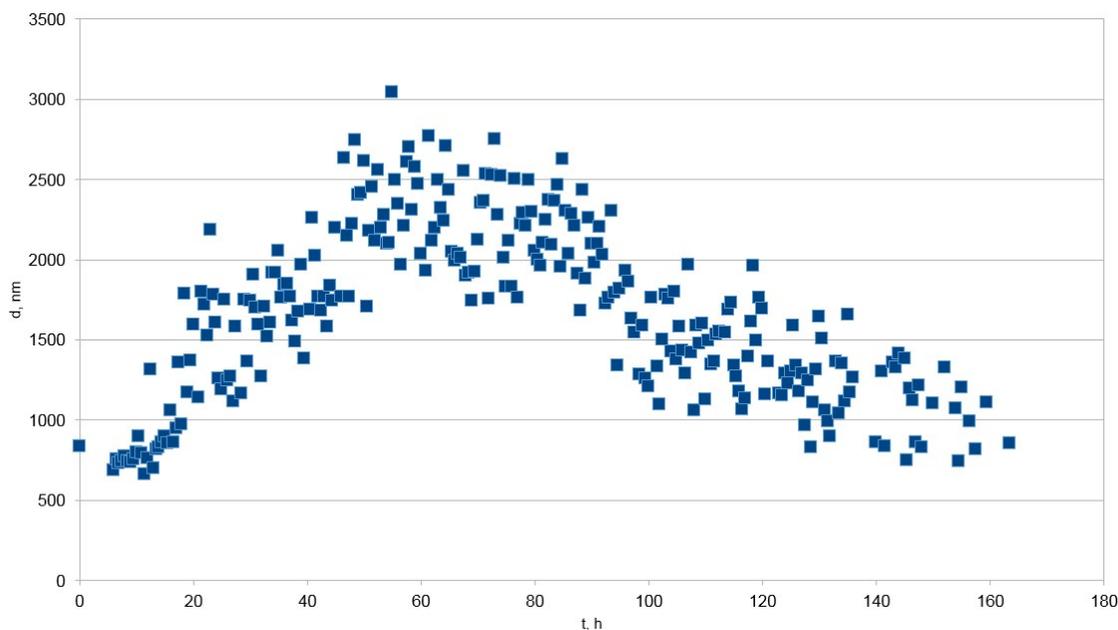


Figure 16. Yeast cells size estimated during the 1 week measurement

The results show that the method allows evaluation of biological cell sizes and monitoring of the dynamics of the process affecting the size, like cell growth, cell split, cell death, etc. Yeast cell size and dynamics of the fermentation process, as example, can be easily monitored, with this method. The method can be used in general to monitor the variation of the particle sizes for various suspensions.

After testing the ANN based DLS time series processing approach on simulated time series, that is time series computed for monodispersed SCs, we tested the ANN function on data recorded during DLS experiments. The first type of suspension used as SCs consisted of burned clay particles manually ground and placed in a transparent tube. Grinding in a mortar produces spherical particles with a wide diameter size distribution. Prior of starting the DLS time series recording, the suspension was allowed to sediment for 4 hours, to make sure that the remaining particles have the diameter in the range of hundreds of nanometers.

As the consequence of the null resultant of three forces: gravity, buoyant force and the viscous force in laminar flow regime (Stokes), the sedimentation motion of the SC carries on with a constant velocity, v_s , which, for a spherical shaped particle is:

$$v_s = \frac{2(\rho - \rho_0)g}{9\eta} \cdot r^2 \quad (24)$$

where r is the radius of the sphere, ρ is the density of the scattering centers material, ρ_0 is the density of the fluid, η is the dynamic viscosity coefficient of the fluid. More details on the role of the sedimentation and of the Brownian motion in the fluctuations of a DLS time series can be found in [104].

We notice from equation (24) that the sedimentation velocity increases with the square of the particle radius/diameter, which makes the bigger particles to sediment faster. Following this sedimentation procedure, by adjusting the distance from the upper part of the solution to the beam to be 1 mm, in a long lasting experiment, of the order of days, we can select suspensions that have suspended particles smaller than a certain value of the diameter, which can be as small as roughly 200 nm. The variation of the diameter of the biggest remaining particles versus the sedimentation time is presented in Fig. 1 of reference [276], not repeated here. We must mention at this point that this procedure is not precise regarding filtering of the particles by their diameter, simply because the 1 mm distance cannot be measured with very good accuracy, as the superficial forces create a concave surface at the upper part of the fluid in contact with air, but the same recorded time series is processed in both manners, using DLS as reference and ANN as an alternative method, therefore the precision of the sedimentation method is not crucial. The sedimentation method described here should be viewed just as a sample preparation procedure, which assures that the particles in the beam have diameters in a relatively small range, which is smaller than a certain diameter.

For the clay particles measurements, suspensions of different average diameters were prepared by manually milling burned clay in a mortar for different amounts of time. Milling was dry at the beginning and lasted for 10 minutes and continued for 25 more minutes after adding a small amount of deionized water. A small amount of the mix was extracted and diluted with deionized water to reach a good degree of transparency and was placed in a transparent cylindrical tube and sealed. The carrier fluid was water at 20 °C. The burned clay density was 1362 kg/m³, water density 1000 kg/m³, $\eta=1.002$ mPa·s.

The diameters computed with the ANN function are hereafter considered the ANN diameters. Figure 17 illustrates the variation of the diameters of the particles in the active area of the sample tube, which is in the laser beam, versus time (measured from the beginning of the recording).

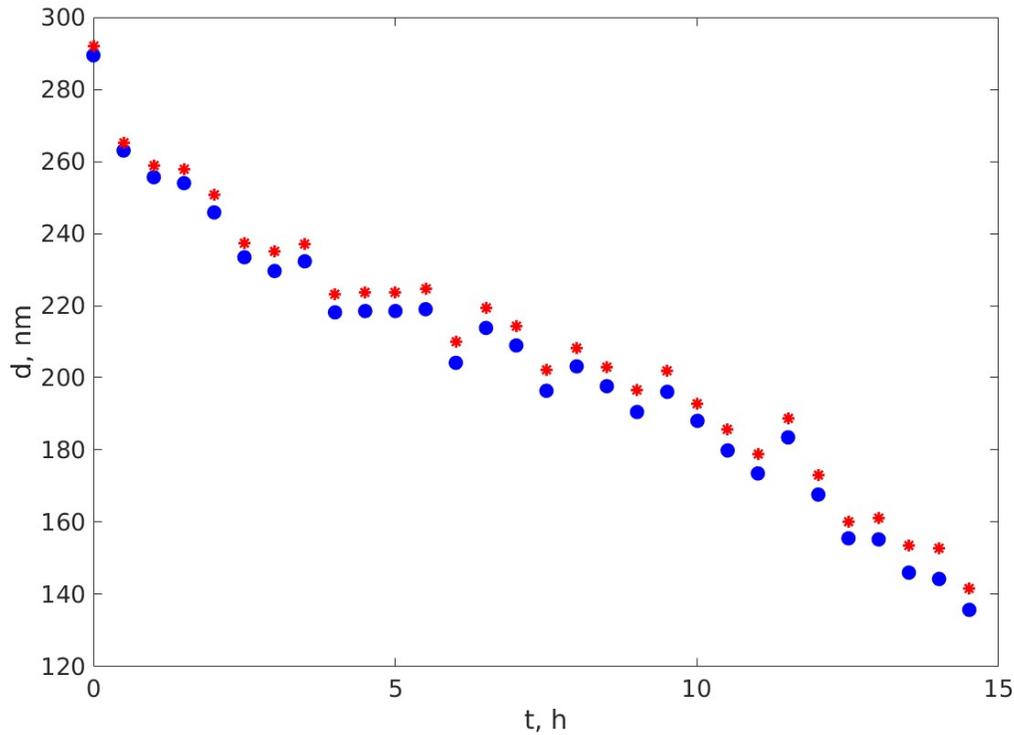


Figure 17. The reference diameters (circles) and the ANN diameters (stars) versus time

We notice that there is very good correlation of the diameters computed with the two methods. Actually the linear dependency $d_{NN}=a \cdot d_{ref}+b$ has a linear correlation coefficient [287, 288], $R=0.9998$ and we notice that the errors increase in time, as the diameter (average diameter actually) decreases, which is consistent with the conclusion of the previous sections describing the errors for simulated time series.

We can also notice that the diameter is decreasing, which is consistent with the physical process that takes place in the suspension, namely the sedimentation process.

The second suspension that was used in testing the method consisted of milk proteins. 30 ml of skimmed milk having 1.5% fat was centrifuged at 3000 rpm for 10 minutes and the protein content was extracted from the bottom of the vessel and stored using a plastic flask at 0.4 oC. Prior of the DLS experiment the sample was diluted in deionized water till the sample was transparent for the laser beam, slightly looking bluish in white light. For this purpose 0.05 ml of milk proteins separated as described above was diluted in 3.45 ml of deionized water. Milk proteins aggregation has been induced by Calcium lactate, in its most common form of pentahydrate $C_6H_{10}CaO_6 \cdot 5H_2O$. In order to prepare the aggregation inducer, 0.0333 g of Calcium lactate was dissolved in 20 ml of deionized water, thus making a solution of 0.1665 % (weight). A volume of 0.5 ml of such solution was injected in 3.5 ml of protein aqueous

suspension and induced protein aggregation. Time series each lasting for 8 s each were recorded with 172 s time delay between them and the average diameter was assessed both using the reference DLS time series processing and the ANN based time series processing procedures.

The sample was bovine milk, which contains about 30–35 grams of protein per liter. Most of the proteins (80%) are casein, which can be found in the form of casein micelles, which are aggregates of several thousand protein molecules. The size of the proteins suspended in bovine milk has been measured and data is reported in papers like [289, 290], and the reported results indicate that the most of the proteins have a diameter in the range 77 to 115 nm [289] and bigger in [290], but the preparation process was filtration, to remove fat particles, rather than centrifugation. We used centrifugation and the samples were extracted from the bottom of the recipient. We found a reference diameter of 118 nm, in very good agreement with [289] and [290].

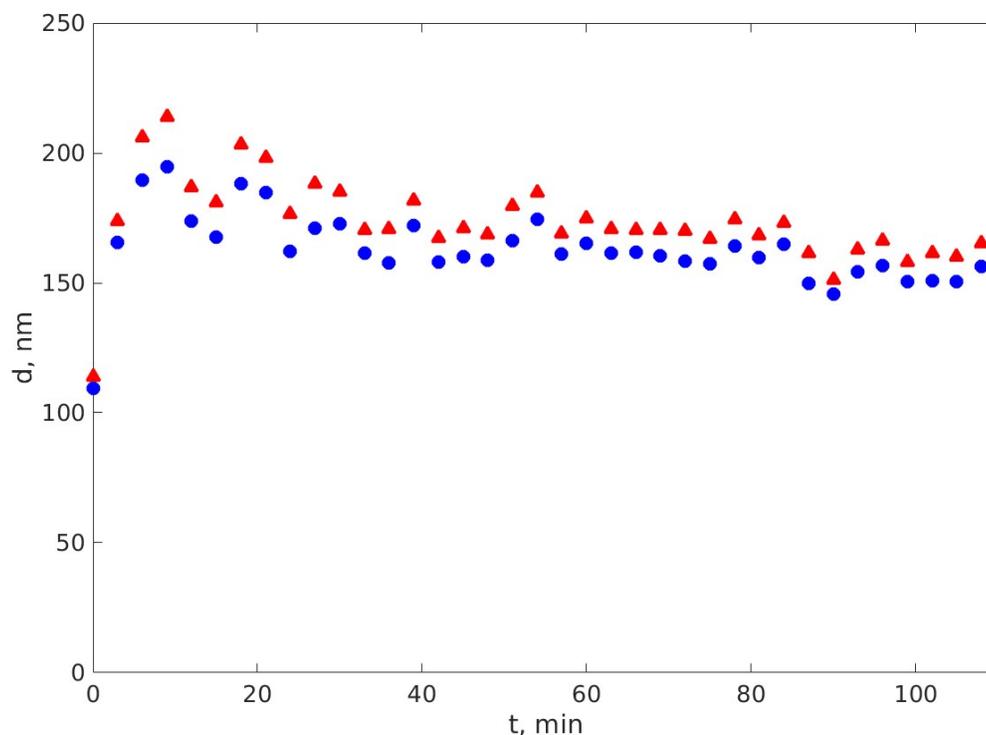


Figure 18. The variation of the proteins diameters during aggregation. Circles are the reference diameters and the triangles are the ANN diameters

Figure 18 illustrates the variation of the SCs diameters, computed both using the reference DLS and the ANN based DLS in time. The moment 0 was selected to be the moment prior of injecting the aggregation agent.

Figure 18 reveals that the milk proteins aggregation is quite fast, as in about 6 minutes the aggregates reached a maximum of the diameter. But relevant for this work is that the ANN diameters are closed to the reference diameters, yet slightly overestimated. This feature could be found in Figure 17 as well, depicting the diameters of a different type of suspension.

The third experimental data set selected was based on measurements on waste water. Waste water is water which results from domestic usage (toilet, bath, sinks) that is collected in the sewage system. The waste water is collected via pipes into a treatment center. The water treatment has typically three stages. The first stage is the screening phase, which has the purpose to remove large objects from the waste water. Large objects can be stones, sticks, sanitary items (diapers, cotton, wipes) but also bottles or clothes. These big objects can block and damage the treatment equipment for the following stages. In this same stage the grit is removed. The second stage consists of the primary treatment of water which involves separation of solid organic material (usually human waste) from the waste water. For this purpose the waste water is placed into large tanks to allow the solids to sink to the bottom. The solids which settle are called also 'sludge'. These tanks are usually circular and they have scrappers that scrape the floor of the tanks to move the sludge towards the center so that pumps can take the sludge away. Water is then moved to the next stage, the third one, where water undergoes secondary treatment. In this stage water is placed into aeration tanks, which are usually rectangular shaped. Aer is pumped into water and bacteria are stimulated to consume the organic material that remained after the first stages in the water. The fourth stage is where water is placed again into settlement tanks to allow sinking of new sludge formed as a result of the bacterial action. The process is then similar to the one used in stage 2. Water is allowed to flow over a wall and it is further filtered by sand to remove other harmful substances. The resulting water is then released into a stream in nature. Sometimes in the last phase water is also treated with chlorine or ultraviolet light to remove harmful bacteria.

Figure 19 shows a schematic for the water treatment station that was used for the measurements presented here. The figure also shows the places where the measurements were taken. At point 1, the untreated water was used for measurement. We consider this measurement as water input measurement. At point 2, water which went through the screen grit chamber and aeration tank active sludge was used for measurements. At point 3 the water which went through final sedimentation was used for measurements. Samples of water from the three mentioned

points were taken and measurements for short time (5 minutes) and long time (10 hours) were made. For the short duration, time series were recorded every 10 seconds. For the long time

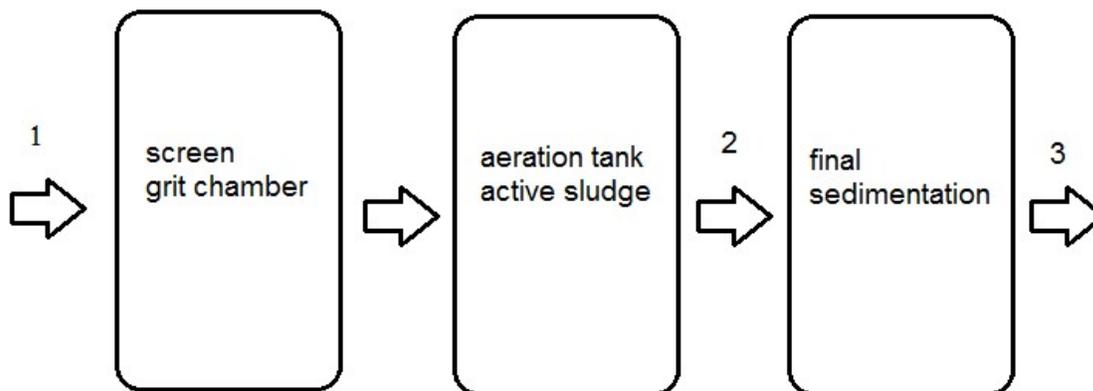


Figure 19. Schematic of the Water Treatment Station

duration, time series were recorded every 20 minutes.

Figure 20 shows some pictures taken at the actual wastewater treatment station used for taking the samples.

The ANN was slightly different due to the wider range of particle sizes that we expected



Figure 20. Various Water Treatment Station Stages

to find in the water. The ANN had an input layer of 350 neurons, corresponding to 350 lags. The hidden layer had 26 neurons and the output layer had 1 neuron, outputting the average diameter for the particles found in the wastewater. The training data was consisted of sets of simulated time series, for a diameter range of 10 to 3000 nm, generated with a step of 1 nm, with 20 sets per diameter.

Figure 21 shows the average particle size estimated during different stages of wastewater processing.

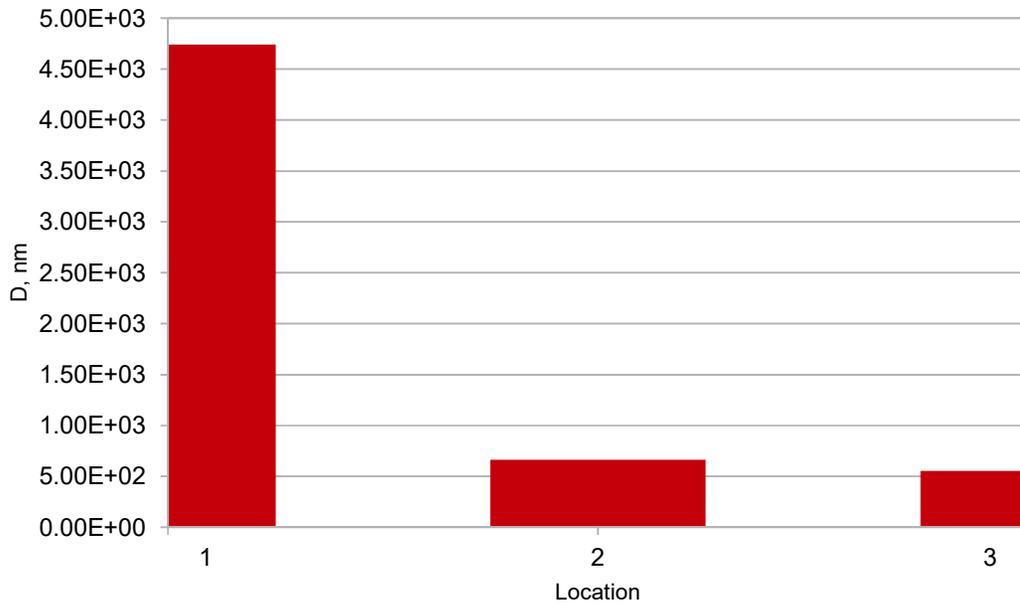


Figure 21. Average size of the suspended particles measured during different stages of wastewater processing

The wastewater at input, measurement at location 1, contains sand, silt and organic suspensions. The average diameter estimated was 4700 nm. It is worth to note that the input water is already pre-filtered so that the big particles are not in. After sedimentation, only the organic particles remained in the aeration tank. After about 7 – 8 hours average time spent in the aeration tank with aerobe bacteria, the size of the remaining particles was about 660 nm. This is visible in the measurement at location 2. At location 3, the water exits the sedimentation tank having small particles of 550 nm.

What can be observed and deduced from the observations is that with one type of physical measurement only, DLS, and monitoring the diameter variation in time, we can identify the type of suspended particles. We will present below some cases which we encountered also in our measurements.

When we have a decreasing trend in size, we have inorganic particles only, as this evolution is a characteristic in general of the sedimentation process, and sedimentation happens when the density of the particle material is higher than the density of water.

When the particles are organic, the density of the particles is comparable to the density of water and the particles do not sink. When we have an increasing size trend, we can deduce that we have organic particles, as the organic particles tend to grow by aggregation or by evolution and cellular growth.

However, when we have a decreasing size followed by a spurious output and then by constant big size, we have a mixture of inorganic and organic suspensions. This is the result of the fact that the size reported by the algorithm is the size of the predominant particles, with all the mentioned particularities of this phenomenon, for example bigger particles having bigger impact on the measurement, as seen in the theoretical foundations sections. The initial decreasing is caused by sedimentation of inorganic particles, but then the increase is caused by the fact that only organic big particles, originally lesser in number than the sediments, but now the only ones remaining floating, will impact the measurement.

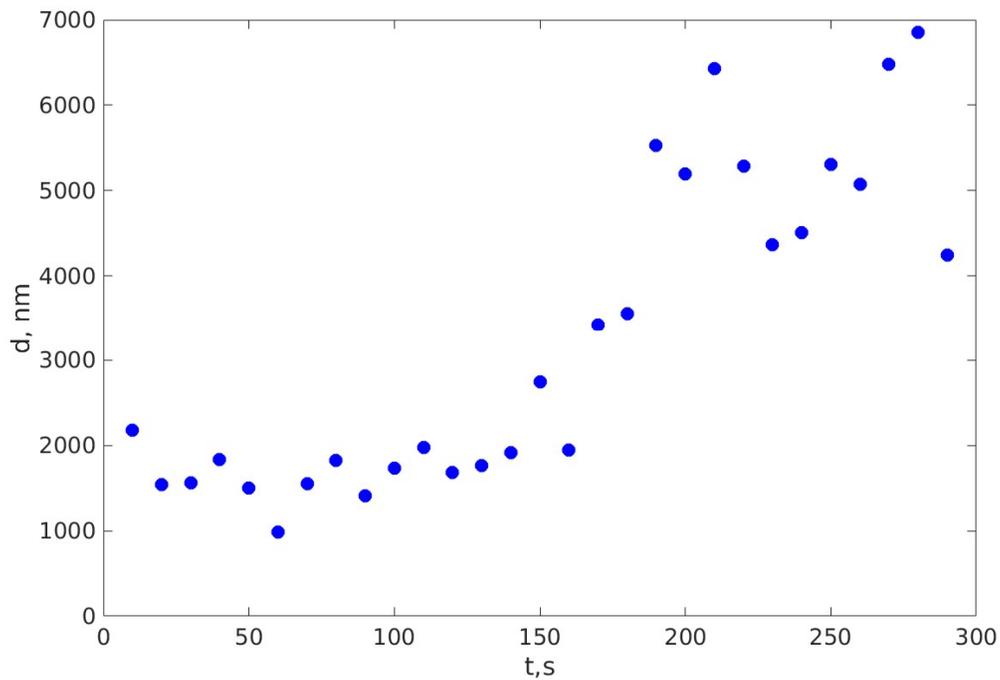


Figure 22. Size Monitoring at Location 1. Short Time Measurement.

Figure 22 shows the size monitoring at location 1, the input water, for a short time measurement of 300 seconds.

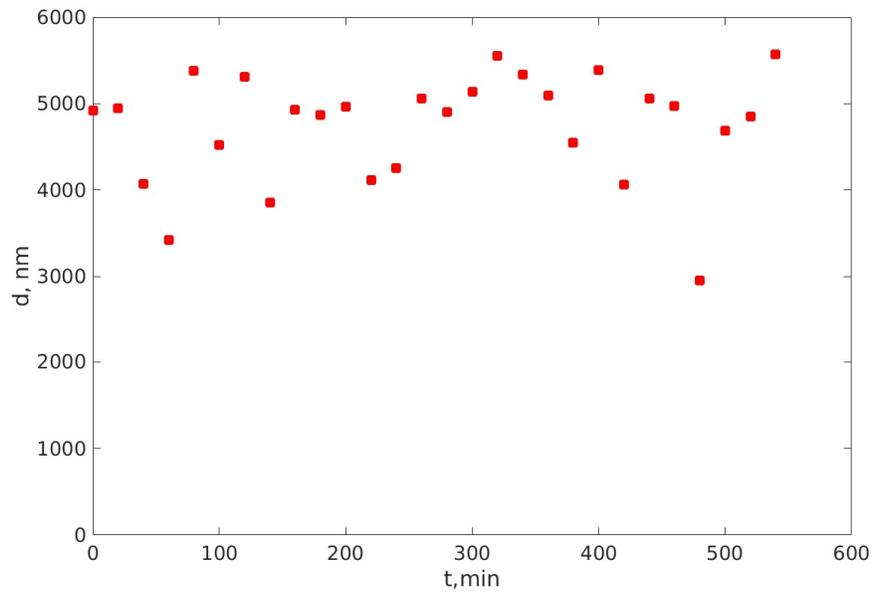


Figure 23. Size Monitoring at Location 1. Long Time Measurement.

Measurements for the same location but this time for a long period of 10 hours are presented in Figure 23.

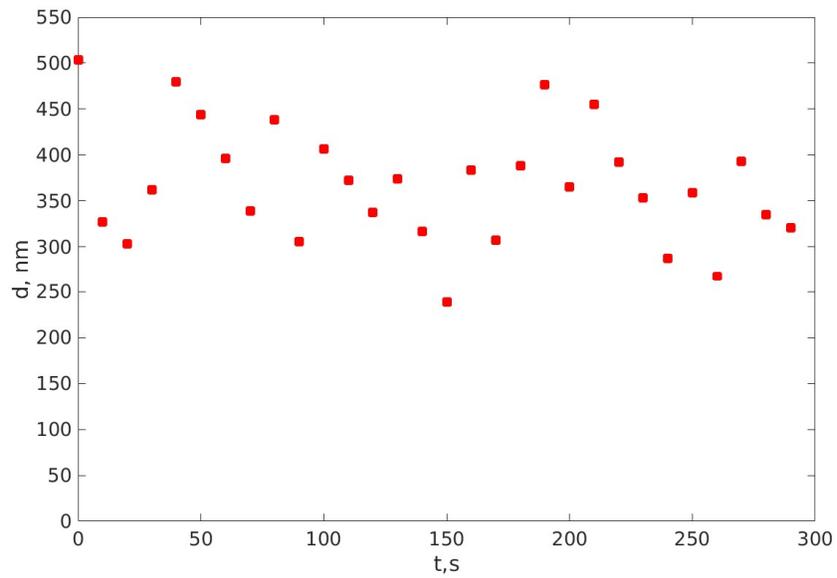


Figure 24. Size Monitoring at Location 2. Short Time Measurement.

Figure 25 presents the results of the size monitoring at location 2, the aeration tank. The figure

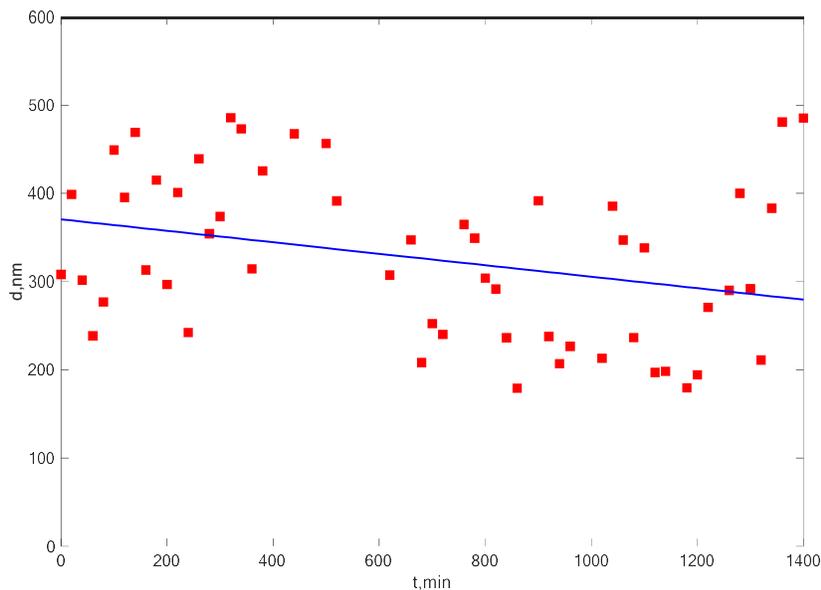


Figure 25. Size Monitoring at Location 2. Long Time Measurement.

also contains a linear fitting showing an overall size decrease trend, corresponding to the line described by the equation $y = -0.065x + 370$, $R^2 = 0.095$.

Figure 26 shows the size monitoring for the measurements at location 3, the exit location.

At location 1, the short time monitoring shows a fast sedimentation of the inorganic particles which initially dominate the measurement as they are in much higher number. As soon as their number decreases, the organic particles, which do not sink, as they have a density equal to the water density, start to dominate the measurement. This is the reason of the jump from low particle size value to a higher value after about 150 seconds. In the long time monitoring at location 1 we can observe that the organic particles remain in the water, which makes sense as the water did not undergo yet the stages of the treatment meant to reduce the organic content.

At location 2, the short time monitoring show that there are still inorganic particles that dominate the measurement, most likely particles which were not filtered out and were moved in the water due to movement through tanks. In the long time monitoring it can be seen that these particles sediment and the size tends to decrease slowly. This is consistent with the

treatment stage which involves mixing the water with active sludge which has small particles and bacteria in higher concentration.

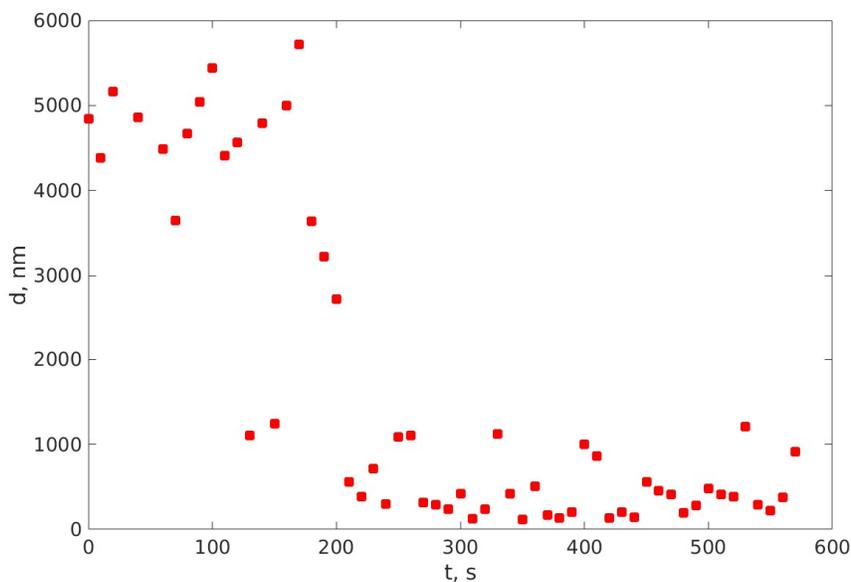


Figure 26. Size Monitoring at Location 3. Short Time Measurement.

At location 3, the exit, there is a smaller amount of particles but with a mixture of sizes in the low range. The evolution of sizes can be explained noting that the measurement takes place for the particles in the scattering volume, in the way of the laser beam. The result is consistent with particles which are organic but have a density slightly smaller than water. As time passed, particles drifted towards surface, the bigger the diameter, the bigger the velocity. Smaller particles drifted also but with a smaller velocity. In the end, some organic particles remain in wastewater after processing.

Interpreting the results for the clay and milk particles, we notice that while the sedimentation process acts as a filtering procedure, the protein aggregation does not have such a physical process once aggregation started, therefore the polydispersity is bigger and we selected this type of suspension to test the robustness of the ANN procedure. More insight though brings a plot of the relative errors of the ANN compared with the reference, defined as in eq. 9 for the two types of suspensions, prepared by sedimentation and aggregation, which is presented in Figure 27 .

Figure 27 reveals that the relative errors of the ANN based DLS procedure are bigger for protein aggregation samples than for sedimentation samples, yet reasonably small. The bigger errors are caused by the wider size distribution. Even so, the ANN based DLS procedure proved to be reasonably precise in computing the average diameter of the particles in

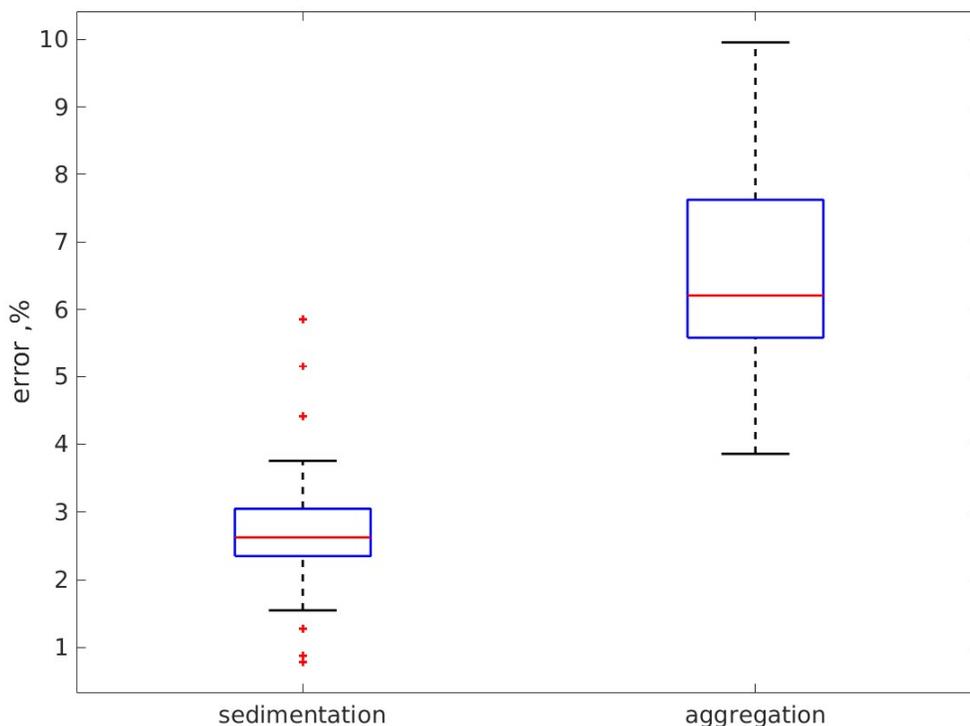


Figure 27. Plot of the relative errors of the ANN compared with the reference for the two types of suspensions, clay and milk, prepared by sedimentation and aggregation

suspension, despite being considerably faster than the reference DLS. The whole set of 30 time series recorded on the sedimentation clay sample were processed with the reference DLS and the procedure lasted for 0.12846 s, fitting only, after the PS had been computed. The ANN based DLS procedure lasted for 0.000019 s after the ACRs had been computed, which makes the ANN DLS procedure 6761 times faster. We should bear in mind that the time required for fitting a function to a set of data strongly depends on the start values selected for the parameters to be determined, therefore we can conclude that the ANN based DLS is thousands of times faster.

The DLS with ANN assisted time series processing is fast and can be implemented on light computing platforms. One single, simple physical method, DLS, can be used in monitoring the

size of the particles suspended in wastewater. The variation of the particle size on short and longer time can provide information on the type of suspended particles, without the use of a chemical procedure, of reagents or other type of consumables.

7. ALTERNATIVE DYNAMIC LIGHT SCATTERING ARTIFICIAL NEURAL NETWORK IMPLEMENTATION

After obtaining the results presented in the previous sections with the Matlab implementation, we aimed to optimize the approach even further and to test the limits of the precision which we can obtain with an ANN approach.

We considered that an own implementation of an ANN in a programming language would allow us more flexibility than the Matlab ANN toolset, which would then enhance our possibilities to optimize the approach. For implementing the ANN we selected the programming language Python [292], due to its inherent advantages, namely being easy to learn and work with it, excellent support and a multitude of already available libraries. For the ANN implementation we used the open source libraries TensorFlow [293], Theano[294] and Keras [295].

For the Python implementation we chose a different strategy than the Matlab used one. If on the Matlab implementation of the ANN we used a fixed architecture, which was chosen at the beginning and kept for the entire operation of the ANN, in the Python implementation we wanted to experiment with different architectures.

TensorFlow is a flexible architecture library which allows development of computational algorithms in a tensor paradigm. It is a library used in many scientific areas but it has a good level of support for machine learning and artificial intelligence, including neural networks [293].

Theano is a Python library used to operate multiple-dimensional arrays for numeric computations [294].

Keras is an application programming interface written in Python and interacting with TensorFlow and Theano. It is usually used for high level neural network programming [295].

Using these three libraries allows us to define and implement our own ANN, in a flexible way, controlling all the possible parameters of the network. In this way we were able to gradually evolve and modify our design until we reached the optimal performance. In the first

steps we searched for the optimal training strategy and then for the optimal neural network architecture, including optimum number of layers, neurons, activation functions and learning algorithm. After many trials we automated the process by implementing various Python scripts which varies in an intelligent way the key parameters of the network to search for the optimum.

As it can be seen in the section describing the ANN, the architecture can have a big impact on the quality of results obtained. In the first step we tried various architectures based on a structure with neurons ranging from 100 to 600 in the input layer, one hidden layer with neurons ranging from 5 to 100 and one output layer with one neurons. As activation functions we searched the optimal in the following functions in the hidden layer [296]: the rectified linear unit, the hyperbolic tangent, the sigmoid, the softmax, and the linear activation function. In our first tries we used the same activation function also in the output layer but during our research we found that it is optimal to use one of the above in the hidden layer but the linear activation function in the output layer.

For the training algorithm, we also searched the entire space offered by keras [297]. After extensive research, we found that the optimal algorithm for our problem is the Adam algorithm [298, 299].

In the next step we built Python code that varied the number of input and hidden neurons, but also varies the activation functions and the training algorithms and for each iteration trained the network and verified its performance on experimental data. All the networks having an average error higher than a threshold, defined in our research as 10% in a first step then 5% in a second step, were discarded, all the others saved.

In our search for the best ANN to solve the DLS problem we pushed the architecture one step forward and searched for optimum configuration when the neural network has two hidden layers. We automated the search in the space of activation functions, training algorithms and layers sizes and obtained an optimal configuration for the ANN.

The ANN configuration we considered as optimal, and which we used also to obtain the experimental results presented in the following section is the following (including various key parameters for the code):

- Input layer size (neurons): 350
- Hidden layer 1 size (neurons): 26
- Hidden layer 2 size (neurons): 10
- Output layer size (neurons): 1

- Number of training epochs: 40000
- Training algorithm (keras optimizer): adam
- Learning rate: 0.001
- Activation functions:
 - Hidden layer 1: softmax
 - Hidden layer 2: rectified linear unit
 - Output layer: linear
- Performance indicator monitored: mean absolute percentage error (MAPE)

For training the network, we used generated time series, following the same approach as described previously. We used time series for particles in range 50 – 400 nm, 20 time series per value, with induced noise of 50Hz and its harmonics.

After training, we used for validation of the network the experimental data for clay particles, the same dataset as the one used for testing the MATLAB implementation.

As a result of this set of trial runs we reached the following conclusions:

- A decent performance can be obtained on experimental data with this type of network. Best performance obtained was around 8.5% relative error but this was only for seed=7. For random seed, a performance of around 6% was observed in other runs.
- For an optimal experimental data performance, the overall performance of the network will not be optimal. This means that when the network will fit well on the experimental data, it will not fit good even experimental data for a larger interval. Just check the training fit plots.
- When the seed was randomized, the network performance was each time different.
- For generated data, obviously, the best performance is obtained when the training and testing set were identical
- For generated data, training with noisy series gives better results.

Further on, we tried various combinations of architecture and training data, on similar toy ANN models as in the previous trial set. For a specific choice of architecture/training data, a fixed number of neural networks were built. The lowest error network was recorded as the final result. Then the architecture/training data was changed, and results were compared.

In the next step we used the results obtained so far together with the architecture described. After 4409 training cycles, the ANN reached the optimal performance. The batch size for the training cycle was 7820. The performance of the ANN in training is described by the following indicators: error in training was 0.66% and error in testing (on simulated time series) was 2.12 %.

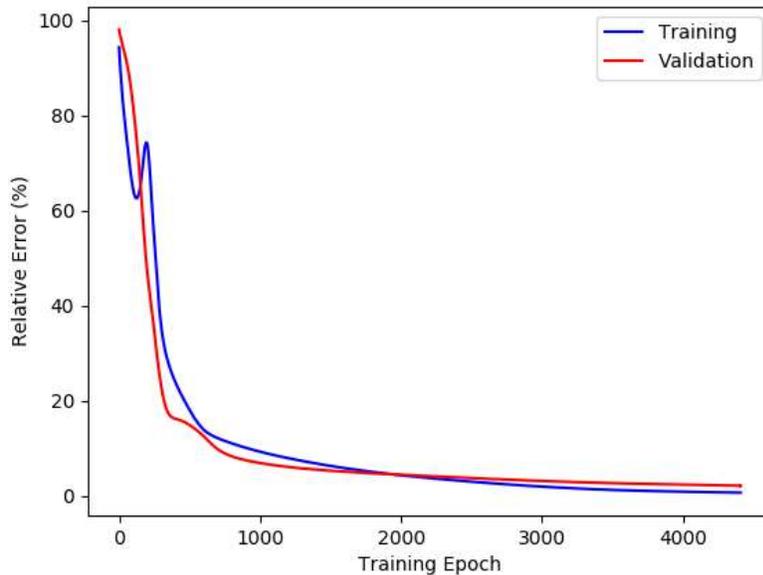


Figure 28. ANN Relative Error Evolution in Training

After the network was trained, we estimated the particle sizes for the clay particles data set. The error of the ANN approach relative to the reference approach was a very good value of 0.197 %.

Figure 28 shows the relative ANN error evolution during training. We can observe that the ANN converges towards the optimal performance and reaches the optimum after 4409

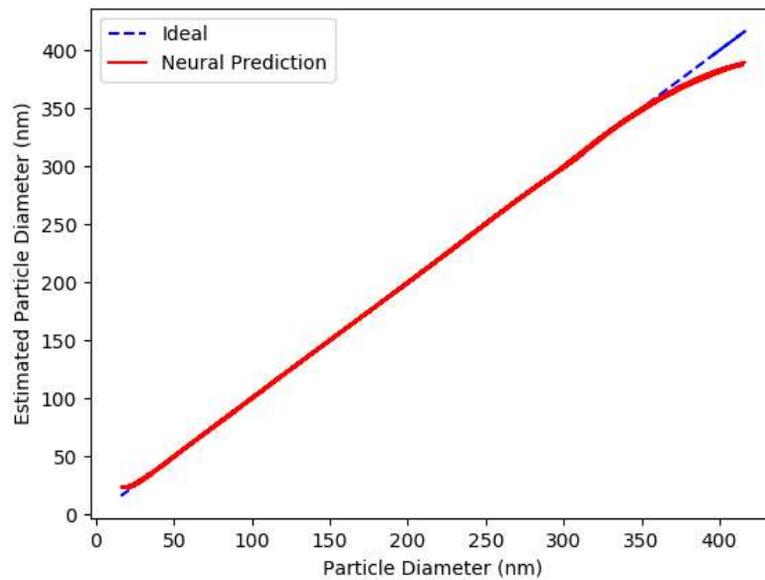


Figure 29. Estimated Particle Diameter for the Entire Generated Data Set Used for Training (range 20-400 nm)

cycles.

Figure 29 shows the estimated particle diameter of the trained ANN on the generated time series used for training. We can observe that in the range 50 – 350 nm the fit is perfect. The extremes, at around 20-30 nm and above 350 nm, the trained ANN does not make a perfect estimation, leading to the mentioned validation error of around 2.2%. However, if the generated range is covering the range in which the ANN is planned to be used, a trained ANN can function

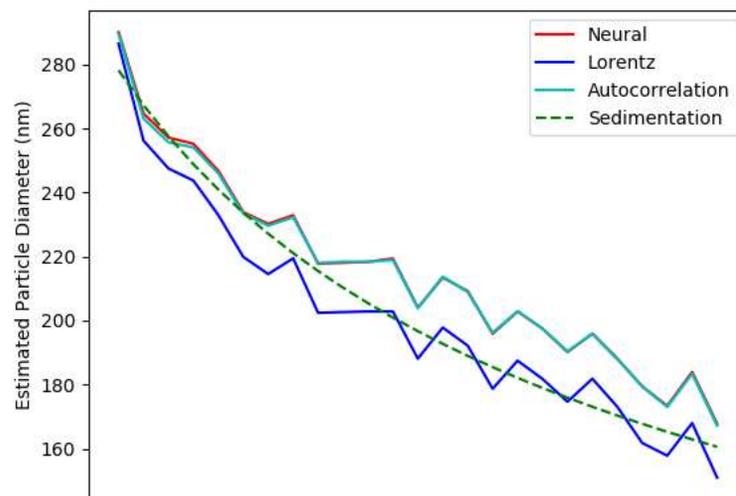


Figure 30. ANN Particle Size Estimation on Clay Experimental Data Set

close to perfect. This means that if we want to use an ANN for a certain interval, we shall generate the training time series for a bigger interval which includes the one in which the ANN is planned to operate.

Figure 30 presents the trained ANN size estimation for the clay particles data set. With green dashed line we represented the simulated expected particle size based on a sedimentation model. With blue we represented the size estimated with the Lorentz fitting approach. The light green line and the red line represent the reference and the ANN estimation respectively. We can observe first that all the methods follow a sedimentation pattern which is a sign that the measurements and estimations are correct qualitatively. We also can observe that the ANN and the reference estimations are very close to each other, fact that can also be seen in the low value of relative error of the ANN, namely 0.19 %. The ANN method is therefore a successful and precise way to estimate particle sizes, with all the advantages presented.

8. FURTHER DEVELOPMENT IDEAS

The approach presented here focuses on obtaining the average particle size. Most of the times, a suspension has a polydisperse system of particles. As we have seen in the experimental results sections, obtaining the average particle size is of great use, allowing an intelligent analysis of the suspension, but the next step would be to extend this work so that the output of the algorithm is the size distribution and not only the average size. Theoretical ideas and computer simulation results on how this could be achieved were already presented in this work but experimental and practical work is also required to put those ideas into practice. In addition, various other improvement ideas were explored and presented in the thesis.

9. CONCLUSION

In this work we investigated the possibility to improve the methods based on dynamic light scattering for assessing nano and micro particle sizes for particles found in liquid or gaseous fluids. The vision was that a miniaturized portable device capable of delivering these assessments can be implemented in future. The main objective was to implement a novel procedure for particle sizing using dynamic light scattering measurements assisted by an artificial neural network. The procedure should be simpler than the existing methods and shall allow smaller amount of computing time leading to the development of the mentioned vision,

namely the miniaturized equipment. As a secondary objective, we aimed to present ideas for a simple setup for achieving signal conditioning and conversion to a digital time series for a dynamic light scattering experiment but also to present the results of the investigations on how to integrate the novel procedure into a miniaturized device.

Having these objectives in focus we achieved the following in our present research:

- A brief but targeted review of the existing dynamic light scattering based methods and an analysis of the limitations they have
- A brief review and analysis of various theoretical and practical aspects of signal processing, data acquisition and artificial neural networks
- An investigation based on computer simulation results based on own developed code, on the connections and dependencies between the various physical parameters and the miniaturized device performance parameters
- A detailed design from architecture to hardware and software implementation of the various modules needed for the miniaturized device
- An implementation of a novel procedure for dynamic light scattering assisted by artificial neural networks
- An integration of the hardware and software as well as the algorithm proposed and an implementation of 5 constructive variants of the device
- A complete set of tests on simulated and experimental data for the procedure, revealing the performance and precision of the novel approach proposed
- An optimization of the proposed artificial neural network dynamic light scattering procedure, with an own implemented code based on Python
- A brief but valuable analysis of the further development steps

The ideas presented in this work can be further developed to implement a miniaturized device that can estimate particle sizes on the field. The advantages brought by the ANN approach would allow extremely fast computing times, practically the result can be obtained instantly, on the fly. The training of the ANN took less than one hour on an ultrabook laptop, which is optimized for power consumption and not for computing. Once the ANN is trained, it can be used to compute the average diameters of a set of input data. This computing is thousands of times faster than fitting a function by a nonlinear least squares procedure, because it resumes at matrix multiplication and addition rather than computing a function and computing numerically gradients many times. The precise time can vary depending on the size of the data set, of the disk access time and so on, but remains thousands of times faster.

The work presented stands as proof of concept in using NNs for DLS time series processing and a step toward designing a small, very low cost, portable DLS particle sizer, that can use a light computing platform, like a cell phone or a developing platform rather than a PC or laptop. It stands as a step towards designing a very fast DLS particle sizer, almost a real time sizer, for assessing the average diameter of the suspended particles. The application of such a device can be found in any domain in which there are micro and nano-particles of interest, both in knowing the sizes of them but also in studying the dynamics of the process ongoing, starting from environmental sciences, medical sciences, engineering and production processes, food industry, automotive and aerospace but also fundamental sciences such as physics and chemistry.

10. BIBLIOGRAPHY

1. B. Berne, R. Pecora – “Dynamic Light Scattering”, Dover Publications Inc., Mineola, New York, (2000).
2. D. Tong – “Electromagnetism”, University of Cambridge, Part IB and Part II of Mathematical Tripos, (2015).
3. http://www.edinformatics.com/math_science/electromagnetic_spectrum.htm
4. T.S. Evans – “Symmetry and Unification”, Theoretical Physics, Physics Department, Imperial College London, (2009)
5. D. Tong – “Quantum Field Theory”, University of Cambridge, Part III Mathematical Tripos, (2006).
6. B. J. Ackerson – “Selected Topics in Static and Dynamic Light Scattering”, Van't Hoff Laboratory, University of Utrecht, The Netherlands, (1986).
7. Clark, Lunacek, Benedek – “A Study of Brownian Motion Using Light Scattering”. American Journal of Physics. 38, 575, (1970).
8. D. T. Gillespie – “The Mathematics of Brownian Motion and Johnson Noise”. American Journal of Physics. 64, 225, (1996).
9. D. T. Gillespie – “Fluctuation and Dissipation in Brownian Motion”, American Journal of Physics. 61, 1077, (1993).
10. R. DeSerio, S. Hagen – “Brownian Motion”, University of Florida, Department of Physics, (2015).
11. Dan Chicea, "Using CHODIN to simulate the dynamics of coherent light scattering on nanofluids", Proc. SPIE 7469, ROMOPTO 2009: Ninth Conference on Optics: Micro- to Nanophotonics II, 74690B (17 May 2010); doi: 10.1117/12.866768; <https://doi.org/10.1117/12.866768>
12. D. Arzensek – “Dynamic Light Scattering and Application to Proteins in Solutions”, University of Ljubljana, Faculty of Mathematics and Physics, Ljubljana, (2010).
13. Y. Sun – “Different Particle Size Information Obtained from Static and Dynamic Laser Light Scattering”, Simon Fraser University, BC, Canada, (2004).
14. C. Urban – “Development of Fiber Optic Based Dynamic Light Scattering for a Characterisation of Turbid Suspensions”, Swiss Federal Institute of Technology Zuerich, PhD Dissertation, Zuerich (1999).
15. L. Oegendal – “Light Scattering Demystified – Theory and Practice”, University of Copenhagen, (2016).

16. K.-N. Liou – “A Complementary Theory of Light Scattering by Homogenous Spheres”, University of Utah, Dept. of Meteorology, Salt Lake City, Utah, (1977).
17. D. Chicea – “Coherent Light Scattering on Nanofluids: Computer Simulation Results”, Physics Department, ULBS, (2007).
18. D. E. Koppel – “Analysis of macromolecular polydispersity in intensity correlation spectroscopy: The method of cumulants.”, *The Journal of Chemical Physics*, 57:4814-4820, (1975).
19. T. Ursell – “Diffusion of Solid Particles Confined in a Viscous Fluid”, (2005).
20. <http://www.physics.nyu.edu/grierlab/methods/node11.html>
21. <https://www.dkfz.de/Macromol/research/dls.html>
22. B. J. Ackerson, N.A. Clark – “Dynamic Light Scattering At Low Rates of Shear”, *Journal de Physique* (1981), 929.
23. Zetasizer APS User Manual. December 2008. See <http://www.malvern.com>
24. J. H. Wen – “Dynamic Light Scattering: Principles, Measurements and Applications”, (2010).
25. Dynamic Light Scattering: An Introduction in 30 minutes. Technical Note (MRK656-01), (2010). See <http://www.malvern.com>
26. C. B. Bargeron – “Measurement of a Continuous Distribution of Spherical Particles by Intensity Correlation Spectroscopy: Analysis by Cumulants”, *J. Chem. Phys.*, 61:2134-2138, (1974).
27. J. C. Brown, P. N. Pusey, R. Dietz – “Photon Correlation Study of Polydisperse Sampels of Polystyrene in Cyclohexane”, *J. Chem. Phys.*, 62:1136-1144, (1975).
28. B. J. Frisken – “Revisiting the Method of Cumulants for the Analysis of Dynamic Light Scattering”, *Data. Appl. Opt.*, 40:4087-4091, (2001).
29. F. R. Hallet, T. Craig, J. Marsh, B. Nickel – “Particle Size Analysis: Number Distributions by Dynamic Light Scattering”, *Canadian Journal of Spectroscopy*, 34:63-70, (1989).
30. W. Brown – “Dynamic Light Scattering: The Method and Some Applications”. Clarendon Press, Oxford, (1993).
31. P. N. Pusey, W. van Megen – “Detection of Small Polydispersities by Photon Correlation Spectroscopy”, *J. Chem. Phys.*, 80:3515-3520, (1984).
32. B. Chu – “Laser Light Scattering: Basic Principles and Practice”, Academic Press, New York, (1991).

33. H. C. Van De Hulst – “Light Scattering by Small Particles”, Dover Publications, New York, (1981).
34. K. S. Schmitz – “An Introduction to Dynamic Light Scattering by Macromolecules”, Academic Press, San Diego, (1990).
35. H. R. Haller, C. Destor, D. S. Cannell – “Photometer for Quasielastic and Classical Light Scattering”, Rev. Sci. Instrum. 54: 973-983, (1983).
36. P. R. Bevington, D. K. Robinson – “Data Reduction and Error Analysis for the Physical Sciences”, McGraw-Hill, (1992).
37. G. Arutyunov – “Classical Field Theory”, Inst. For Theoretical Physics and Spinoza Institute, Utrecht University, (2011).
38. D. Fleisch – “A Student’s Guide to Maxwell’s Equations”, Cambridge University Press, (2008).
39. Bo Thide – “Electromagnetic Field Theory”, 2nd Edition, Upsala, Sweden, (2011).
40. T. Yu – “Lagrangian Formulation of the Electromagnetic Field”, (2012).
41. D. J. Griffiths – “Introduction to Electrodynamics”, Prentice Hall, New Jersey, (1999).
42. L. D. Landau, E. M. Lifshitz – „Electrodynamics of Continuous Media”, 2nd Edition, Pergamon Press, (1960).
43. L. D. Landau, E. M. Lifshitz – „The Classical Theory of Fields”, 3rd Edition, Pergamon Press, (1971).
44. A. S. Kompanejets – “Theoretical Physics”, Foreign Languages Publishing House, Moscow, (1961).
45. B. de Wit – “Introduction to Quantum Field Theory”, Institute for Theoretical Physics, Utrecht University, (2008).
46. B. U. Felderhoff – „On the Propagation and Scattering of Light in Fluids”, Phys. in press, (1974).
47. J. D. Jackson – „Classical Electrodynamics”, Wiley, New York, (1965).
48. R. Feynman – “The Feynman Lectures on Physics”, Addison Wesley Longman, (1970).
49. A. Einstein – “Investigation on the Theory of Brownian Movement”, Republication of the original 1926 translation, Dover Publications, (1956).
50. “The Collected Papers of Albert Einstein, Vol. 2, The Swiss Years: Writing, 1900 – 1909”, Princeton University Press, (1989).
51. R. Pecora – “Dippler Sifts in Light Scattering from Pure Liquids and Polymer Solutions”, Journal of Chemical Physics: 40:1604, (1964).

52. J. Goodman – “Some Fundamental Properties of Speckle”, J. Opt. Soc. Am. 66: 1145-1150, (1976).
53. C. Urban, P.Schurtenberger – “Characterization of Turbid Colloidal Suspensions Using Light Scattering Techniques Combined with Cross – Correlation Methods”, J. Colloid Interface Sci. 207(1): 150 – 158, (1998).
54. I. Block, F. Scheffold – “Modulated 3D Cross – Correlation Light Scattering: Improving Turbid Sample Characterization”, Rev. Sci. Instruments. 81(12): 123107, (2010).
55. P. Hassan, S. Kulshreshtha – “Modification of the Cumulant Analysis of Polydispersity in Quasielastic Light Scattering Data”. Journal of colloid and interface science, 300(2): 744-8, (2006).
56. S. Provencher – “CONTIN: A General Purpose Constrained Regularization Program for Inverting Noisy Linear Algebraic and Integral Equations”, Computer Physics Communications, 27(3): 229, (1982).
57. S. R. Aragon, R. Pecora – “Theory of Dynamic Light Scattering from Polydisperse Systems”, The Journal of Chemical Physics, 64:2395, (1976).
58. Shannon, C. “A Mathematical Theory of Communication”. Bell System Technical Journal 27 (3): 379-423.
59. Schneider, T.D. “Information Theory Primer”. National Cancer Institute, 14.04.2007.
60. Carter, T. “An Introduction to Information Theory and Entropy”, Santa Fe. 2014.
61. Massey, J. “Guessing and Entropy”. Proc. IEEE International Symposium of Information Theory, 2013
62. Malone, David; Sullivan, Wayne. “Guesswork is not a Substitute for Entropy”. Proceedings of the Information Technology and Telecommunications Conference. 2005.
63. National Instruments Low-Cost Data Acquisition Family. <http://www.ni.com/low-cost-daq/>
64. Rei, S. “Implementing A Low Cost Data Acquisition System for Engineering Education Programs in Universities”. Proceedings of the BRCEBE – ICEBE 2017 Conference, Sibiu, Romania.
65. Enzo Mastinu, Bo Håkansson, Max Ortiz-Catalan, “Low-cost, open source bioelectric signal acquisition system”, (2017), Proc. Conference: 14th International Conference on Wearable and Implantable Body Sensor Networks, At High Tech Campus - Eindhoven, THE NETHERLANDS
66. Teensy USB Development Board: <https://www.pjrc.com/store/teensy36.html>

67. Teensy USB Virtual Serial Receive Speed:
https://www.pjrc.com/teensy/benchmark_usb_serial_receive.html
68. Adafruit Development Boards, Teensy 3.6: <https://www.adafruit.com/product/3266>
69. Teensy 3.6 Pinout: <https://www.pjrc.com/teensy/pinout.html>
70. Mihiu, P. “Procesarea Numerica a Semnalelor”, Editura Alma Mater, 2005.
71. Wikipedia: sine wave: https://en.wikipedia.org/wiki/Sine_wave
72. Euler, L. “Chapter 8: On transcending quantities arising from the circle of Introduction to the Analysis of the Infinite”, page 214, section 138 (translation by Ian Bruce, pdf link from 17 century maths). (1748).
73. Fourier, Collins English Dictionary - Complete & Unabridged 10th Edition, HarperCollins, accessed 5 May 2017
74. Mit Open Courseware: Fourier Analysis: <https://ocw.mit.edu/courses/mathematics/18-103-fourier-analysis-fall-2013/>
75. Weisstein, Eric W. "Fourier Series--Square Wave." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/FourierSeriesSquareWave.html>
76. Smith, S. “The Scientist and Engineer’s Guide to Digital Signal Processing”. Second Edition. California Tech Publishing. 1999.
77. Feng, J. “Digital Communications and Signal Processing – with Matlab Examples”. Department of Computer Science and Centre for Scientific Computing University of Warwick, UK. 2007
78. National Semiconductor Application Note 255. “Power Spectra Estimation”. 1980.
79. Wikipedia: Spectral Density: https://en.wikipedia.org/wiki/Spectral_density
80. Stoica, P. Moses, R.. "Spectral Analysis of Signals".(2005)
81. Stein, J. Y. Digital Signal Processing: A Computer Science Perspective. Wiley (2000)
82. Wikipedia: Fourier Transform: https://en.wikipedia.org/wiki/Fourier_transform
83. Wikipedia: Autocorrelation: <https://en.wikipedia.org/wiki/Autocorrelation>
84. Dunn, Patrick F. (2005). Measurement and Data Analysis for Engineering and Science. New York: McGraw–Hill. ISBN 0-07-282538-3.
85. Priestley, M. B. (1982). Spectral analysis and time series. London, New York: Academic Press. ISBN 0125649010.
86. Percival, Donald B.; Andrew T. Walden (1993). Spectral Analysis for Physical Applications: Multitaper and Conventional Univariate Techniques. Cambridge University Press. pp. 190–195. ISBN 0-521-43541-2.

87. Weisstein, Eric W. "Autocorrelation." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Autocorrelation.html>
88. Rei, S., Chicea, D., & Olaru, S. (2016). AUTOCORRELATION FUNCTION ANALYSIS IN PROCESSING STOCHASTIC TIME SERIES. Annals of the University Dunarea de Jos of Galati: Fascicle II, Mathematics, Physics, Theoretical Mechanics, 39(1).
89. Weisstein, Eric W. "Wiener-Khinchin Theorem." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/Wiener-KhinchinTheorem.html>
90. Forinash, K. "Sound. An Interactive eBook on the Physics of Sound": <https://soundphysics.ius.edu/>
91. All About Circuits: Introduction to Analog-to-Digital Converters: <https://www.allaboutcircuits.com/textbook/digital/chpt-13/digital-analog-conversion/>
92. Kuphaldt, T. "Lessons in Electric Circuits": <https://www.allaboutcircuits.com/textbook/>
93. Microchip Developer Help: Analog to Digital Converter Operation: <http://microchipdeveloper.com/pwr3101:analog-to-digital-converter-operation>
94. Dirac, Paul (1958), The Principles of Quantum Mechanics (4th ed.), Oxford at the Clarendon Press, ISBN 978-0-19-852011-5.
95. Wikipedia: Dirac Delta Function: https://en.wikipedia.org/wiki/Dirac_delta_function
96. Porat, B. (1994). Digital Processing of Random Signals: Theory & Methods. Prentice Hall. ISBN 0-13-063751-3.
97. AudioSculpt 3.0 User Manual: <http://support.ircam.fr/docs/AudioSculpt/3.0/co/AudioSculptguideWeb.html>
98. Wikipedia: Nyquist-Shannon Theorem: https://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem
99. Burg, Romney, Schwartz. "Digital Sound & Music: Concepts, Applications, and Science". Franklin, Beedle & Associates Inc (October 28, 2016)
100. An Introduction to Delta Sigma Converters: <http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html>
101. Cooley, James W.; Tukey, John W. (1965). "An algorithm for the machine calculation of complex Fourier series". Math. Comput. 19: 297–301. doi:10.2307/2003354
102. Englitz, B. "Dynamic Light Scattering". University of California San Diego. 2002.
103. Sartor, M. "Dynamic Light Scattering". UCSD. 2003.

104. D. Chicea, "Revealing Fe₃O₄ nanoparticles aggregation dynamics using dynamic light scattering, *Optoelectronics and Advanced Materials-Rapid Communications*, 3(12), 1299-1305, (2009).
105. D. Chicea, E. Indrea, C.M. Cretu, "Assesing Fe₃O₄ nanoparticle size by DLS, XRD and AFM", *Journal of Optoelectronics and Advanced Materials*, 14(5-6), 460-466, (2012).
106. Weisstein, Eric W. "Lorentzian Function." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/LorentzianFunction.html>
107. Chicea, D. (2013). Water Suspension Analysis by Modified Dynamic Light Scattering.
108. Chicea, D. (2014). A Simple Algorithm to Simulate Nanoparticles Motion in a Nanofluid. *UNIVERSITY POLITEHNICA OF BUCHAREST SCIENTIFIC BULLETIN-SERIES A-APPLIED MATHEMATICS AND PHYSICS*, 76(2), 199-206.
109. D.K. Lynch and W. Livingston, "Color and light in nature", Cambridge University Press, 1995 (1st ed.) and 2001 (2nd ed.), ISBN 0 521 77504 3 - see Table 4.3
110. D. Segelstein, "The Complex Refractive Index of Water", M.S. Thesis, University of Missouri, Kansas City (1981). A summary of the refractive index data can be downloaded from here or it is included as "Segelstein.txt" when downloading the MiePlot program.
111. IAPWS 5C: "Release on refractive index of ordinary water substance as a function of wavelength, temperature and pressure" (September 1997) published by International Association for the Properties of Water and Steam (IAPWS)
112. Humboldt State University. GSP216. Electromagnetic Spectrum. http://gsp.humboldt.edu/olm_2015/Courses/GSP_216_Online/lesson1-2/spectrum.html
113. Smith, J. Matlab Function: Air Refractive Index: <https://www.mathworks.com/matlabcentral/fileexchange/31240-air-refractive-index?focused=5186662&tab=function>
114. Phillip E. Ciddor, "Refractive index of air: new equations for the visible and near infrared," *Appl. Optics* 35, 1566-1573 (1996).
115. B. Edlén, "The refractive index of air," *Metrologia* 2, 71-80 (1966).
116. K.P. Birch and M.J. Downs, "An updated Edlén equation for the refractive index of air," *Metrologia* 30, 155-162 (1993).
117. K.P. Birch and M.J. Downs, "Correction to the updated Edlén equation for the refractive index of air," *Metrologia* 31, 315-316 (1994).
118. Langowski, J. Website: Biophysics of Macromolecules: <https://www.dkfz.de/Macromol/research/dls.html>

119. Danholt Company, ENVIRONMENT: Air filter - lab experiment - LIGHTAIR IonFlow gathers particles. Web: <http://www.danholt.de/english/3820/1727/1727/181001/design1.html>
120. Advanced Water Filters Particles Chart: <http://www.advancedwaterfilters.com/faq-particle-size-chart/>
121. Loschmidt, J. On the Size of the Air Molecules. *Journal of Chemical Education* 1995 72 (10), 870 DOI: 10.1021/ed072p870.2
122. D'Arrigo JS. Screening of membrane surface charges by divalent cations: an atomic representation. *Am J Physiol.* 1978 Sep235(3):C109-17 abstract, p.C109 right column bottom paragraph, p.C110 right column 2nd paragraph & p.C112 left column 2nd paragraph.
123. Engineering Toolbox: Particles Sizes: https://www.engineeringtoolbox.com/particle-sizes-d_934.html
124. Wikipedia: Rayleigh Scattering: https://en.wikipedia.org/wiki/Rayleigh_scattering
125. Barnett, C.E. (1942). "Some application of wavelength turbidimetry in the infrared". *J. Phys. Chem.* 46 (1): 69–75. doi:10.1021/j150415a009
126. Refractive Index Database: <https://refractiveindex.info/?shelf=main&book=Y3A15O12&page=Zelmon>
127. Cecie Starr (2005). *Biology: Concepts and Applications*. Thomson Brooks/Cole. ISBN 0-534-46226-X.
128. Nobbmann, U. Do refractive index and absorption matter for nanoparticles?: <http://www.materials-talks.com/blog/2014/08/05/faq-how-important-are-refractive-index-absorption-for-nanoparticles/>
129. George W. Mulholland, Raymond L. McKenzie, Egon Marx, and Robert A. Fletcher. Refractive index and evaporation rate of individual smoke droplets. *Langmuir* 1985 1 (3), 367-372 DOI: 10.1021/la00063a019.
130. Yamasoe, M. A., Y. J. Kaufman, O. Dubovik, L. A. Remer, B. N. Holben, and P. Artaxo (1998), Retrieval of the real part of the refractive index of smoke particles from Sun/sky measurements during SCAR-B, *J. Geophys. Res.*, 103(D24), 31893–31902, doi:10.1029/98JD01211.
131. Silicon Labs SI1133 Data Sheet: <https://www.silabs.com/documents/public/data-sheets/Si1133.pdf>

132. FX Solver: Rayleigh Scattering – Intensity of Light:
<https://www.fxsolver.com/browse/formulas/Rayleigh+Scattering+-+Intensity+of+Light>
133. Horiba: Choosing the Concentration Range for DLS Size Measurement:
<http://www.horiba.com/scientific/products/particle-characterization/education/sz-100/particle-size-by-dynamic-light-scattering-resources/choosing-concentration-for-dls-size-measurement/>
134. Almeida, R. Rayleigh and Mie Scattering, Colloidal Metals and Photo-elastic Properties. International Materials Institute for New Functionality in Glass, Lehigh University. 2005.
<https://www.lehigh.edu/imi/teched/OPG/lecture18.pdf>
135. Prah, S. Mie Scattering Calculator. Oregon Medical Laser Center.
http://omlc.ogi.edu/calc/mie_calc.html
136. Tyler J. Johnson, Jason S. Olfert, Ross Cabot, Conor Treacy, Caner U. Yurteri, Colin Dickens, John McAughey, Jonathan P.R. Symonds, Steady-state measurement of the effective particle density of cigarette smoke, In Journal of Aerosol Science, Volume 75, 2014, Pages 9-16, ISSN 0021-8502,
[https://doi.org/10.1016/j.jaerosci.2014.04.006.\(http://www.sciencedirect.com/science/article/pii/S0021850214000627\)](https://doi.org/10.1016/j.jaerosci.2014.04.006.(http://www.sciencedirect.com/science/article/pii/S0021850214000627))
137. Nobbmann, U. Protein Sizing by Light Scattering, Molecular Weight and Polydispersity. Malvern Instruments. Malvern, Worcestershire.
<http://www.nanoparticles.org/pdf/nobbmann.pdf>
138. Jimenez, J.L. Lecture Notes. Aerosol Scattering and Cloud Nucleation.
http://cires1.colorado.edu/jimenez/AtmChem/CHEM-5151_S05_L16.pdf
139. Hackley V.A., Clogston J.D. Measuring the Size of Nanoparticles in Aqueous Media Using Batch-Mode Dynamic Light Scattering. NIST Special Publication 1200-6. 2007, rev. 2015. <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1200-6.pdf>
140. Wikipedia: List of Refractive Indexes:
https://en.wikipedia.org/wiki/List_of_refractive_indices
141. Research Gate Forum: What is the lowest particle number or concentration for dynamic light scattering to detect? :
https://www.researchgate.net/post/What_is_the_lowest_particle_number_or_concentration_for_dynamic_light_scattering_to_detect
142. Lelli L. Lecture: Aerosol and Clouds. Lecture 1. Bremen University, 2014. www.iup.uni-bremen.de/~luca/?%20download=01_LL_VO.pdf

143. Shaw, R. Dynamic Light Scattering. Achieving reliable nano particle sizing. Malvern. <http://149.171.168.221/partcat/wp-content/uploads/Malvern-Zetasizer-LS.pdf>
144. Kondo K., Masaki S., Tabuchi T. Measurement of Particles in Liquid Materials Using the Light Scattering Method. 2013. http://www.rion.co.jp/english/product/docs/p_tech03.pdf
145. Malvern Technical Note: Dynamic Light Scattering, and Introduction in 30 Minutes. https://warwick.ac.uk/fac/cross_fac/sciencecity/programmes/internal/themes/am2/booking/particlesize/intro_to_dls.pdf
146. Physics. StackExchange Forum: Dependence of scattering on particles size and refractive index: <https://physics.stackexchange.com/questions/143438/dependence-of-scattering-on-particle-size-and-refractive-index>
147. Born M., Wolf E. Principles of Optics. Cambridge University Press; 7th edition (October 13, 1999)
148. Richardson, H. H., Carlson, M. T., Tandler, P. J., Hernandez, P., & Govorov, A. O. (2009). Experimental and theoretical studies of light-to-heat conversion and collective heating effects in metal nanoparticle solutions. Nano Letters, 9(3), 1139–1146. <http://doi.org/10.1021/nl8036905>.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2669497/>
149. Geints, Y. E., & Zemlyanov, A. A. (2017). Optimal conditions for laser-induced heating of a double-shell spherical nanocapsule. Journal of Applied Physics, 121(12), 123111.
150. Astafyeva, L. G., Voshchinnikov, N. V., & Waters, L. B. (2002). Heating of three-layer solid aerosol particles by laser radiation. Applied optics, 41(18), 3700-3705.
151. Pustovalov, V.K. Theoretical study of heating of spherical nanoparticle in media by short laser pulses. Department of Innovation Research, Belarussian Institute of System Analysis, Masherov Pr., 7, Minsk 220004, Belarus. Chem. Phys. 308 (2005) 103-108. Elsevier.
152. Texas Instruments: Selecting an A/D Converter. <http://www.ti.com/lit/an/sbaa004a/sbaa004a.pdf>
153. LaserPoints: Laser Buying Guide: <http://www.laserpoints.com/laser-buying-guide>
154. Value Amrita: Laser beam divergence and spot size: <http://vlab.amrita.edu/?sub=1&brch=189&sim=342&cnt=1>
155. Jacobs, G. Understanding Spot Size for Laser Scanning. Professional Surveyor Magazine. 2006. https://hds.leica-geosystems.com/hds/en/Prof_Surv_Spot_Size_Oct06.pdf
156. Koechner, Walter. Solid-State Laser Engineering. Berlin: Springer, 2006.

157. Ion, John. Laser Processing of Engineering Materials. Amsterdam: Elsevier/Butterworth-Heinemann, 2005.
158. Laser Surveying. London: Van Nostrand Reinhold (International), 1989.
159. Technical Data Sheet 1206 Package Phototransistor
<https://www.robofun.ro/docs/ELPT15-21C.pdf>
160. Karki, J. Understanding Operational Amplifier Specifications. Texas Instruments. 1998.
<http://www.ti.com/lit/an/sloa011/sloa011.pdf>
161. Texas Instruments NE5532x, SA5532x Dual Low-Noise Operational Amplifiers Data Sheet. <http://www.ti.com/lit/ds/symlink/ne5532.pdf>
162. Burr-Brown OPT 301 INTEGRATED PHOTODIODE AND AMPLIFIER Data Sheet.
<http://www.ti.com/lit/ds/sbbs001/sbbs001.pdf>
163. Electronics Tutorials: Frequency Response: <http://www.electronicstutorials.ws/amplifier/frequency-response.html>
164. Teensy 3.0, 3.1, 3.2, LC, 3.5, and 3.6 ADC implementation library created by Pedro Villanueva. <https://github.com/pedvide/ADC>
165. Rei, S. Data Acquisition System Based on ATmega328 Microcontroller. “Lucian Blaga” University of Sibiu, Master of Engineering Thesis, Sibiu, 2015.
166. Malvern Frequently Asked Questions: Calculating Volume Distributions from DLS Data: <http://www.materials-talks.com/wp-content/uploads/2015/08/FAQ-Mie-Equations.pdf>
167. Christopher M. Bishop. 1995. Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York, NY, USA.
http://cs.du.edu/~mitchell/mario_books/Neural_Networks_for_Pattern_Recognition_-_Christopher_Bishop.pdf
168. Simon Haykin. 1998. Neural Networks: A Comprehensive Foundation (2nd ed.). Prentice Hall PTR, Upper Saddle River, NJ, USA.
https://cdn.preterhuman.net/texts/science_and_technology/artificial_intelligence/Neural%20Networks%20-%20A%20Comprehensive%20Foundation%20-%20Simon%20Haykin.pdf
169. Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth. Matlab Neural Network Toolbox™ User's Guide. 2017 https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf
170. Martin T. Hagan, Howard B. Demuth, Mark H. Beale, Orlando De Jesus. Neural Network Design (2nd Edition). 2014. <http://hagan.okstate.edu/nnd.html>
171. Mathworks: Introducing Deep Learning with Matlab. 2017.

172. G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control Signals, and Systems* (1989) 2: 303-314. Springer – Verlag.
173. K. Hornik. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks*, Vol. 4, pp. 251-257, 1991.
174. Csasji, B.C. Approximation with Neural Networks. MSc Thesis. Faculty of Sciences. Eotvos Lorand University, Hungary. 2001.
175. M. Turhan. Neural Networks and Computation of Neural Network Weights and Biases by the Generalized Delta Rule and Backpropagation of Errors. *Rock Solid Images* 1995.
176. Anil K. Jain, Jianchang Mao. *Artificial Neural Networks: A Tutorial*. 1996.
177. Ben Kroese, Patrick von der Smagt. *An Introduction to Neural Networks*. University of Amsterdam. 1996.
178. Kevin Gurney. *An Introduction to Neural Networks*. University of Sheffield. 1997. UCL Press.
179. Dayan, Peter. L.F., Abbott. *Theoretical Neuroscience*. Massachusetts Institute of Technology. 2001.
180. David, Kriesel. *A Brief Introduction to Neural Networks*. 2007.
181. J. Roell. *From Fiction to Reality: A Beginner's Guide to Artificial Neural Networks*. <https://towardsdatascience.com/from-fiction-to-reality-a-beginners-guide-to-artificial-neural-networks-d0411777571b>
182. Alex Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Cambridge University Press 2008.
183. P. Harrington. *Machine Learning in Action*. Manning Publications. 2012.
184. Li Deng, Dong Yu. Deep Learning Methods and Applications. *Foundations and Trends in Signal Processing*. Vol.7, No. 3-4 (2013), 197-387.
185. F. Agostinelli, M Hoffman, P. Sadowski, P. Baldi. Learning Activation Functions to Improve Deep Neural Networks. ICLR 2015.
186. Mathworks Forum: Write CSV from Matlab: <https://www.mathworks.com/help/matlab/ref/csvwrite.html>
187. NIST/SEMATECH e-Handbook of Statistical Methods, <http://www.itl.nist.gov/div898/handbook/>
188. Stackoverflow Forum: Transposing a Matrix from a 2D Array: <http://stackoverflow.com/questions/26197466/transposing-a-matrix-from-a-2d-array>

189. Mathworks Forum: How to separate biases and weights - Matlab NN:
<https://www.mathworks.com/help/nnet/ref/separatwb.html>
190. Processing Forum: Matrix multiplication:
https://processing.org/discourse/beta/num_1206855932.html
191. Ario Muhammad, Java probabilistic functions:
<https://arioresearch.wordpress.com/2012/11/08/java-1-calculate-some-probabilistic-equations/>
192. Processing Forum: How to read/write numeric data in Processing/Java(See also code at the end of this): https://processing.org/discourse/beta/num_1263473746.html
193. Mathworks Forum: Matlab neural networks - getx function:
https://www.mathworks.com/matlabcentral/newsreader/view_thread/13060
194. Mathworks Forum: Matlab neural networks - getwb function:
<https://www.mathworks.com/help/nnet/ref/getwb.html>
195. Mathworks Forum: Matlab neural networks - how to show the weight and bias:
<https://www.mathworks.com/matlabcentral/answers/2356-how-to-show-the-weight-or-bias-in-a-neural-network>
196. Pyren Neural Network: How to load a Matlab NN in Python:
<http://pyrenn.readthedocs.io/en/latest/save.html>
197. Neural Network Blog: Neural Networks in VBA/Excel:<https://quantmacro.wordpress.com/2015/08/13/artificial-neural-network-with-backpropagation-training-in-vba/>
198. How to Implement XOR Function in Excel: <http://powerspreadsheets.com/xor-function-excel/>
199. Barber, S. Codeproject: Neural Networks for beginners:
<https://www.codeproject.com/Articles/16419/AI-Neural-Network-for-beginners-Part-of>
200. Excel Perceptron Example:
https://www.google.ro/url?sa=t&rct=j&q=&esrc=s&source=web&cd=4&cad=rja&uact=8&ved=0ahUKEwj_7oeXkLzRAhUZyFAKHTtBBmEQFgg4MAM&url=http%3A%2F%2Fwww.cs.nott.ac.uk%2F~pszgk%2Fcourses%2Fg5aiai%2F006neuralnetworks%2Fperceptron.xls&usq=AFQjCNG4NbnfwCaw4PTuGamEgYHDmVLrTg&sig2=mOO5OwjHpXmIvg24IgHgXA
201. Neural Network Blog: Simple VB example of NN:
<http://excelneuralnetwork.blogspot.ro/2014/09/simple-example-of-neural-networks.html>
202. Website: Computational Neuroscience in Excel: <http://toritris.weebly.com/>

203. Neural Network XOR Backpropagation Example in Excel:
https://www.google.ro/url?sa=t&rct=j&q=&esrc=s&source=web&cd=16&cad=rja&uact=8&ved=0ahUKEwi-6suDI7zRAhXChRoKHXYaA5Q4ChAWCDwwBQ&url=http%3A%2F%2Fmorphiles.com%2Fwp-content%2Fuploads%2F2013%2F05%2FXOR_BackPropagation_Example.xls&usg=AFQjCNHHKJ7suhX-82hnl-obVsUTrPimvg&sig2=3SQ56xsmmeItzW_CHex0Pg
204. Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015:
<http://neuralnetworksanddeeplearning.com/index.html>
205. Mathworks Forum: Hyperbolic tangent sigmoid transfer function (Matlab implementation):
<https://www.mathworks.com/help/nnet/ref/tansig.html?requestedDomain=www.mathworks.com>
206. Karpathy, A. Hacker`s Guide to Neural Networks: <http://karpathy.github.io/neuralnets/>
207. LinkedIn Forum: Neural Networks Implementation in Matlab:
<https://www.linkedin.com/pulse/neural-networks-implementation-matlab-randa-elanwar>
208. Stackoverflow Forum: Export a Neural Network trained with Matlab in other programming languages: <http://stackoverflow.com/questions/15526112/export-a-neural-network-trained-with-matlab-in-other-programming-languages>
209. Spencer-Harper, M. How to Build a Simple NN in Python in 9 Lines of Code. 2015.
<https://medium.com/technology-invention-and-more/how-to-build-a-simple-neural-network-in-9-lines-of-python-code-cc8f23647ca1#.rotvopz7g>
210. Spencer-Harper, M. How to Build a Multi Layered Neural Network in Python. 2015
<https://medium.com/technology-invention-and-more/how-to-build-a-multi-layered-neural-network-in-python-53ec3d1d326a>
211. Anguelov, B. Basic Neural Network Theory. 2008.
<https://takinginitiative.wordpress.com/2008/04/03/basic-neural-network-tutorial-theory/>
212. Anguelov, B. Basic Neural Network Implementation in C++. 2008.
<https://takinginitiative.wordpress.com/2008/04/23/basic-neural-network-tutorial-c-implementation-and-source-code/>
213. Tulleken, H. 15 Steps to Implement a Neural Network. 2009. <http://code-spot.co.za/2009/10/08/15-steps-to-implemented-a-neural-net/>

214. Shiffman, D. The Nature of Code: Simulating Natural Systems with Processing.
<http://natureofcode.com/book/chapter-10-neural-networks/>
215. Miller, S. How to Build a Neural Network. 2015. <http://stevenmiller888.github.io/mind-how-to-build-a-neural-network/>
216. Britz, D. Implementing a NN from Scratch. 2015.
<http://www.wildml.com/2015/09/implementing-a-neural-network-from-scratch/>
217. Scott C. Neural Network in Processing (2017).
<http://arduinoasics.blogspot.ro/2011/08/neural-network-part-1-connection.html>
218. Stanford University Lectures. CS231n Convolutional Neural Networks for Visual Recognition. <http://cs231n.github.io/>
219. Willems, K. Deep Learning in Python. 2017. Datacamp.
<https://www.datacamp.com/community/tutorials/deep-learning-python>
220. Freund, Y.; Schapire, R. E. (1999). "Large margin classification using the perceptron algorithm" (PDF). Machine Learning. 37 (3): 277–296. doi:10.1023/A:1007662407062.
221. H. Lohninger. Fundamentals of Statistics. Online Lecture Notes.
http://www.statistics4u.com/fundstat_eng/cc_classif_calib.html
222. M. Minsky and S. Papert, Perceptrons, Cambridge, MA. MIT Press, 1969.
223. Fjodor van Veen. A Mostly Complete Chart of Neural Networks.
<http://www.asimovinstitute.org/neural-network-zoo/>
224. Aytul Sofu, Fatma Yesim Ekinici. Estimation of Storage Time of Yogurt with Artificial Neural Network Modeling. August 2007. Journal of Dairy Science 90(7):3118-25. DOI 10.3168/jds.2006-591.
225. Mathworks: Supervised Learning: https://www.mathworks.com/discovery/supervised-learning.html?s_tid=srchtitle
226. Mathworks: Unsupervised Learning:
<https://www.mathworks.com/discovery/unsupervised-learning.html>
227. Dustin Stansbury. Topic in Computational Neuroscience and Machine Learning. Derivation: Error Backpropagation & Gradient Descent for Neural Networks. 2014.
<http://theclevermachine.wordpress.com/2014/09/06/derivation-error-backpropagation-gradient-descent-for-neural-networks/>
228. Alex Minnaar. Deep Learning Basics: Neural Networks, Backpropagation and Stochastic Gradient Descent. 2015. <http://alexminnaar.com/deep-learning-basics-neural-networks-backpropagation-and-stochastic-gradient-descent.html>
229. Wikipedia: Backpropagation. <https://en.wikipedia.org/wiki/Backpropagation>

230. ConvnetJS demo: toy 2d classification with 2-layer neural network.
<http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html>
231. TensorFlow Playground. <http://playground.tensorflow.org/>
232. Jeff Heaton. Introduction to Neural Networks for Java, 2nd Edition. Heaton Research, Inc.; 2 edition (October 1, 2008)
233. Stackexchange Forum: How to choose the number of hidden layers and nodes in a feedforward neural network? <https://stats.stackexchange.com/questions/181/how-to-choose-the-number-of-hidden-layers-and-nodes-in-a-feedforward-neural-netw>
234. Timothy Masters. Practical Neural Network Recipes in C++. Morgan Kaufmann; 1 edition (April 14, 1993)
235. Cireşan, Dan C., Ueli Meier, and Jürgen Schmidhuber. "Transfer learning for Latin and Chinese characters with deep neural networks." In The 2012 International Joint Conference on Neural Networks (IJCNN), pp. 1-6. IEEE, 2012.
https://scholar.google.com/scholar?cluster=7452424507909578812&hl=en&as_sdt=0,22 ;
http://people.idsia.ch/~cirestan/data/ijcnn2012_v9.pdf
236. Slav Ivanov. 37 Reasons why your Neural Network is not working. 2017.
<https://blog.slavv.com/37-reasons-why-your-neural-network-is-not-working-4020854bd607>
237. Neuralware Frequently Asked Questions.
<http://www.neuralware.com/index.php/frequently-asked-questions>
238. Dan CHICEA, Silviu REI. SIMPLE ALGORITHMS TO GENERATE DYNAMIC LIGHT SCATTERING TIME SERIA. 16th International Balkan Workshop on Applied Physics. Constanta, Romania, 2016. http://ibwap.univ-ovidius.ro/2016/uploads/template/Proceeding_IBWAP%202016.pdf
239. D. Chicea, CHODIN- a Computer Code to Simulate Coherent Light Scattering Dynamics on Biological Suspensions, Proceedings of SPIE Volume: 7007, DOI: 10.1117/12.801959, (2007).
240. D. Chicea, Computer Simulation Results of Light Scattered on Red Blood Cells, Romanian Journal of Physics, 51(3-4), 403-409, (2006).
241. REI Silviu, CHICEA Dan, ILIE Beriliu, OLARU Sorin. DYNAMIC LIGHT SCATTERING SIGNAL CONDITIONING FOR DATA PROCESSING. 3rd International Conference for Doctoral Students - IPC 2017 June 22-23, 2017 Braşov, Romania.

242. Liber Tecnologia. Raspberry Pi Zero W, il mini computer con Wi-Fi da 10 euro.
<https://tecnologia.libero.it/raspberry-pi-zero-w-mini-computer-10-euro-8833>
243. Raspberry Pi Foundation. <https://www.raspberrypi.org/>
244. Mathworks: Improve Neural Network Generalization and Avoid Overfitting.
<https://www.mathworks.com/help/nnet/ug/improve-neural-network-generalization-and-avoid-overfitting.html>
245. Mathworks: How to Save and Reuse a Trained Neural Network.
<https://www.mathworks.com/matlabcentral/answers/264160-how-to-save-and-reuse-a-trained-neural-network>
246. Mathworks: Analyze Shallow Neural Network Performance After Training:
<https://www.mathworks.com/help/nnet/ug/analyze-neural-network-performance-after-training.html>
247. Dan Chicea, Silviu Rei, Liana Chicea. A Fast Algorithm for Profiling Suspensions in Natural Water Using Dynamic Light Scattering and Artificial Neural Networks Procedure. AQUATIC BIODIVERSITY INTERNATIONAL CONFERENCE.25 - 28 September, 2019, Sibiu, Transylvania, Romania.
248. Silviu Rei. Implementing A Low Cost Data Acquisition System for Engineering Education Programs in Universities. Proceedings of the BRCEBE-ICEBE'17 Conference, Sibiu, Romania.
249. Silviu Rei, Dan Chicea. USING DYNAMIC LIGHT SCATTERING EXPERIMENTAL SETUP AND NEURAL NETWORKS FOR PARTICLE SIZING. 3rd International Conference for Doctoral Students - IPC 2017 June 22-23, 2017 Braşov, Romania.
250. Mathworks: Choose a Multilayer Neural Network Training Function:
<https://www.mathworks.com/help/nnet/ug/choose-a-multilayer-neural-network-training-function.html>
251. Particle Size Distribution Functions.
<http://www.eng.uc.edu/~beaucag/Classes/Nanopowders/ParticleSizeDistributml/ParticleSizeDistribu.htm>
252. John, V., Angelov, I., Öncül, A.A., Thévenin, D. Techniques for the reconstruction of a distribution from a finite number of its moments. Chemical Engineering Science, Volume 62, Issue 11, June 2007. Elsevier.
253. Robert M. Mnatsakanov¹, and Artak S. Hakobyan². Recovery of Distributions via Moments (2009), IMS Lecture Notes–Monograph Series Optimality: The Third Erich L.

- Lehmann Symposium Vol. 57 (2009) 252–265c Institute of Mathematical Statistics, 2009.
https://projecteuclid.org/download/pdfview_1/euclid.lnms/1249305333
254. G. Golub, B. Gustafsson, P. Milanfar, P. Putinar, J. Varah: Shape reconstruction from moments: theory, algorithms, and applications, SPIE Proceedings vol. 4116 (2000), Advanced Signal Processing, Algorithms, Architecture, and Implementations X (Franklin T.~Luk, ed.) pp. 406--416. <https://people.kth.se/~gbjorn/golub.pdf>
255. L.G.M. de Souza a, G. Janiga a, V. John b,c, D. The´venin a, Reconstruction of a distribution from a finite number of moments with an adaptive spline-based algorithm, Chemical Engineering Science 65 (2010) 2741–2750 https://www.wias-berlin.de/people/john/ELECTRONIC_PAPERS/SJTT10.CES.pdf
256. Parthapratim Biswas. Department of Physics and Astronomy, The University of Southern Mississippi, Hattiesburg, MS 39406, USA* Arun K. Bhattacharya Department of Physics, The University of Burdwan, Burdwan, WB 713104, India Function reconstruction as a classical moment problem: A maximum entropy approach. arXiv:1004.4928v1 [math-ph] 27 Apr 2010 <https://arxiv.org/pdf/1004.4928.pdf>
257. G. Cogoni and P. J. Frawley, Particle Size Distribution Reconstruction Using a Finite Number of Its Moments through Artificial Neural Networks: A Practical Application. Crystal Growth & Design 2015 15 (1), 239-246.DOI: 10.1021/cg501288z. <http://pubs.acs.org/doi/abs/10.1021/cg501288z>
258. V. M. Adamyan, J. Alcober, and I. M. Tkachenko, Reconstruction of Distributions by Their Moments and Local Constraints, AMRX Applied Mathematics Research eXpress 2003, No. 2. https://oup.silverchair-cdn.com/oup/backfile/Content_public/Journal/amrx/2003/2/10.1155_S1687120003212028/3/2003-2-33.pdf?Expires=1498815120&Signature=agUYSGscJj-XM5tgyfcFY8HVBZmGloAVacO9ywsJPVtYi6C9jH8UeT1wocJ~dqHQCGNEIUbt3losSuDjP1HvpxQc2HyGTmRez7RSSL3c-wET8hYYxJH4Q3Gm41VoWKjh-CD3Lc4bal0SSJDvwWoKKkZ18tEXUKince3QXHenYhmszBtHgw4GkMpWStRgpDKrtgpA8yt4qe4o1vhhJRviIpt-WkfN-QdQp~zXu-UTnL44dQL-310VEx7IM2yFrMOfrvo4~cF9VQzNIPwoAkJe8p9eh9ndWn2NRCZyN~f6TfHEESuqBwI-GEhovkqA2n7tBDKmyjt~J5-3DnExyxQ_&Key-Pair-Id=APKAIUCZBIA4LVPAVW3Q

259. Tony Saad, University of Utah, Salt Lake City, UT 84112, USA, The Maximum Entropy Method for Reconstructing Density Distributions. <http://www.tonyasaad.net/docs/tsaad-maximum-entropy-method.pdf>
260. John, V., Univ. des Saarlandes Angelov, I., Max Planck Institute for Dynamics of Complex Technical Systems Öncül, A. A., Otto-von-Guericke-Univ. Magdeburg (287p) Towards the Optimal Reconstruction of a Distribution from Its Moments, Conference: AIChE Annual Meeting Year: 2005 <https://www.aiche.org/conferences/aiche-annual-meeting/2005/proceeding/paper/287p-towards-optimal-reconstruction-distribution-its-moments-0>
261. G Talenti Recovering a function from a finite number of moments <http://iopscience.iop.org/article/10.1088/0266-5611/3/3/016/meta>
262. Weisstein, Eric W. "Raw Moment." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/RawMoment.html>
263. Malvern. Frequently Asked Questions. <http://www.materials-talks.com/wp-content/uploads/2015/08/FAQ-Mie-Equations.pdf>
264. Piederrière Y, Cariou J, Guern Y, Le Jeune B, Le Brun G, Lotrian J 2004 Scattering through fluids: speckle size measurement and Monte Carlo simulations close to and into the multiple scattering, Optics Express 12, 176-188,
265. Piederrière Y, Le Meur J, Cariou J, Abgrall JF, Blouch MT, 2004 Particle aggregation monitoring by speckle size measurement; application to blood platelets aggregation, Optics Express 12, 4596-4601.
266. Chicea D 2007 Speckle size, intensity and contrast measurement application in micron-size particle concentration assessment, European Physical Journal Applied Physics 40, 305-310, DOI: 10.1051/epjap:2007163
267. Goodman JW 1984 Laser speckle and related phenomena, Vol.9 in series Topics in Applied Physics, J.C. Dainty, Ed., Springer-Verlag, Berlin, Heidelberg, New York, Tokyo.
268. Xu R Particle Characterization: Light Scattering Methods 2002 Kluwer Academic Publishers, New York, Boston, Dordrecht, London, Moscow.
269. Chicea D 2012 A Study of Nanoparticle Aggregation by Coherent Light Scattering, Current Nanoscience 8(2), 259-265.
270. Haykin, S 1999 Artificial neural networks: A Comprehensive Foundation, Prentice Hall, ISBN 0-13-273350-1.
271. Carrieri AH 1999 Artificial neural network pattern recognition by means of differential absorption Mueller matrix spectroscopy, Applied Optics 38(17) 3759-3766.

272. Berdnik VV, Loiko VA 2009 Retrieval of size and refractive index of spherical particles by multiangle light scattering: artificial neural network method application, *Applied Optics* 48(32) 6178-6187.
273. Berdnik VV Mukhamedjarov RD, Loiko VA, 2004 Characterization of optically soft spheroidal particles by multiangle light-scattering data by use of the neural-networks method, *Optics Letters* 29(9) 1019-1021.
274. Kaye P, Hirst E, Wang-Thomas Z, 1997 Neural-network-based spatial light-scattering instrument for hazardous airborne fiber detection, *Applied Optics* 36(24) 6149-6156.
275. Ulanowski Z, Wang Z, Kaye P, Ludlow IK 1998 Application of artificial neural networks to the inverse light scattering problem for spheres, *Applied Optics* 37(18) 4027-4033.
276. Chicea D 2017 Using artificial neural networks for dynamic light scattering time series processing, *Measurement Science and Technology* 28(5) 055206.
277. Einstein, A. (1905). "Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen". *Annalen der Physik* (in German). 322 (8): 549–560. Bibcode:1905AnP...322..549E. doi:10.1002/andp.19053220806
278. Stetefeld J, McKenna SA, Patel TR. 2016 Dynamic light scattering: a practical guide and applications in biomedical sciences, *Biophysical Reviews*.;8(4):409-427, doi:10.1007/s12551-016-0218-6.
279. Mark Hudson Beale, Martin T. Hagan, Howard B. Demuth. *Matlab Artificial neural network Toolbox™ User's Guide*. 2017 https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf
280. Duhamel P., Vetterli M., *Fast Fourier Transforms: A Tutorial Review and a State of the Art*, *Signal Processing*, Vol. 19, 259-299, (1990).
281. Dubin S.B. , Lunacek J.H., Benedek G.B., *Observation of the Spectrum of Light Scattered by Solutions of Biological Macromolecules*, *PNAS*, 57(5), 1164-1171 (1967)
282. Shannon C.E., *A Mathematical Theory of Communication*, *Bell System Technical Journal*, 27(4): 623–666, (1948), doi:10.1002/j.1538-7305.1948.tb00917.x.
283. Whittaker E.T. , *On the Functions Which are Represented by the Expansions of the Interpolation Theory*, *Proc. Royal Soc. Edinburgh*. 35: 181–194, doi:10.1017/s0370164600017806, (1915).
284. Nyquist H., *Certain topics in telegraph transmission theory*, *Trans. AIEE*, 47, 363- 390, (1928).

285. Kotelnikov V.A. , On the transmission capacity of the 'ether' and of cables in electrical communications, Proceedings of the first All-Union Conference on the technological reconstruction of the communications sector and the development of low-current engineering, Moscow, (1933).
286. Levenberg K, A Method for the Solution of Certain Problems in Least Squares, *Quart. Appl. Math.*, 2, 164-168, (1944).
287. Karl Pearson (1895) "Notes on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, 58 : 240–242.
288. Stigler, S. M. (1989). "Francis Galton's Account of the Invention of Correlation". *Statistical Science*. 4 (2): 73–79. doi:10.1214/ss/1177012580.
289. de Kruif C. G. (Kees), Huppertz T., 2012, Casein Micelles: Size Distribution in Milks from Individual Cows, *J. Agric. Food Chem.*, 60 (18), pp 4649–4655, DOI: 10.1021/jf301397w.
290. Mootse H., Pisponen A., Pajumägi S., Polikarpus A., Tatar V, Sats A. and Poikalainen V., 2014, Investigation of Casein Micelle Particle Size Distribution in Raw Milk of Estonian Holstein Dairy Cows, *Agronomy Research* 12(3), 753–758.
291. BioNumbers Database: Milo et al. *Nucl. Acids Res.* (2010) 38 (suppl 1): D750-D753.
292. Python Website: <https://www.python.org/>
293. Tensorflow Website: <https://www.tensorflow.org/>
294. Theano Website: <http://deeplearning.net/software/theano/>
295. Keras Website: <https://keras.io/>
296. Keras Activation Functions: <https://keras.io/activations/#available-activations>
297. Keras Optimizers: <https://keras.io/optimizers/>
298. Diederik Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimization. Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015. arXiv:1412.6980
299. Sashank J. Reddi, Satyen Kale, Sanjiv Kumar, On the Convergence of Adam and Beyond. 15 Feb 2018 (modified: 27 Mar 2018)ICLR 2018 Conference Blind Submission.
300. Dan Chicea, Silviu Rei, A fast artificial neural network approach for dynamic light scattering time series processing, *Measurement Science and Technology*, 29(2018) 105201 (15pp), IOP Publishing, <https://doi.org/10.1088/1361-6501/aad937>

