



University Library

Author/Filing Title ANDREASIS

Class Mark T

Please note that fines are charged on ALL
overdue items.

FOR REFERENCE ONLY

0402798430



DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING
LOUGHBOROUGH UNIVERSITY

**AN OPTIMAL NEPHELOMETRIC MODEL DESIGN
METHOD FOR PARTICLE CHARACTERISATION**


BY
APOSTOLOS ANDREADIS, DIPL.ENG.

A Doctoral Thesis
submitted in partial fulfilment of the requirements
for the award of Doctor of Philosophy of Loughborough University

23 April 2002

Supervisor: Professor Peter R. Smith, Ph.D.
Department of Electronic and Electrical Engineering

© BY APOSTOLOS ANDREADIS, 2002

 Loughborough University Pilkington Library	
Date	11/20/03
Class	
Acc No.	040249843

To the open-minded

ABSTRACT

Scattering nephelometry is a particle characterisation method applicable to fluid suspensions containing impurities. Solutions derived by the method feature particle classification by size (diameter), volume or texture as well as continuous on-line and in-situ monitoring. The replacement of turbidimeters with nephelometers in many existing turbidity applications could result in suppression of side effects caused by limitations and uncontrolled parameter drifts and satisfaction of problem-defined constraints at virtually no change in implementation cost.

A major issue of nephelometric model design is the selection of a mathematical tool suitable for the modelling of the data analysis system. The data fitting technique based on Multilayer Feedforward Artificial Neural Networks (MFANNs) is advantageous to any analytical approach for a number of reasons: accomplishment of functional approximation and classification tasks from databases of input-output data associations with no need of further problem or data knowledge, satisfaction of constraints imposed on inputs, outputs and internal complexity, distributed processing, low implementation cost and possibility of structural optimisation. MFANN optimisation is distinctively significant because it reduces the number of data analysis system inputs to the absolute minimum. As a matter of fact, the complexity of the nephelometric chamber is reduced and hence the sensing module is easier to design and less expensive to implement.

This study focuses on the optimisation aspect of the scattering nephelometric model design. The novel Second Order Sensitivity Analysis optimisation algorithm is theoretically developed and incorporated to an innovative MFANN optimisation scheme that gives excellent network pruning and generalisation results. The ISO 12103-1 Arizona Fine (AF) dust is considered representative of the particles found in natural suspensions and examined for its light scattering properties. Theoretical scattering profiles of water suspensions of filtered AF dust are calculated with the aid of an analytical nephelometric model derived from Mie scattering theory. Experimental scattering profiles of filtered AF dust samples dispersed in water are acquired by a commercial polar nephelometric device. Finally, two separate scattering nephelometric models are designed and optimised to classify the theoretical and experimental suspensions by AF particle volume and size respectively.

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to Loughborough University for the academic, financial, technical, medical and social support I was offered throughout my research course. Above all, I show appreciation to my supervisor and director of the Optical Engineering Group, Professor Peter Smith, for his trust in my research skills and his continuous guidance and assistance.

My special thanks go to Dr. Rambod Naimimohasses. His encouragement and communication skills enabled me to settle in the new environment very quickly. Furthermore, his scientific advice and constructive criticism were sources of enthusiasm and inspiration that formed the basis of this research work.

I would also like to thank Mr. Leonardo Ciaccheri, research student of the Istituto di Ricerca sulle Onde Elettromagnetiche in Italy, for performing the nephelometric experiments and supplying me with numerical data and documentation that were essential for the completion of my doctoral study.

Last but not least, I am grateful to my former colleagues Mrs. Olga Kusmartseva, Dr. Matthew Hayes and Messer Vincent Crabtree, Wai Hoong Lum and Daniel Chan for their partnership and assistance.

CONTENTS

ABSTRACT	V
ACKNOWLEDGEMENTS	VI
CONTENTS	VII
CHAPTER 1 INTRODUCTION	1
1.1 Chapter overview	2
1.2 Metrology	3
1.2.1 Microscopy	3
1.2.2 Sieving	4
1.2.3 Sedimentation	5
1.2.4 Stream scanning	6
1.2.5 Field scanning	7
1.3 Surrogates	9
1.3.1 Turbidity	9
1.3.1.1 Overview	9
1.3.1.2 Limitations	15
1.3.2 Colour	18
1.4 Particle characterisation	21
1.5 Thesis overview.....	25
1.6 Original contributions	26
CHAPTER 2 ARTIFICIAL NEURAL NETWORK MODELLING.....	28
2.1 The generic modelling procedure.....	29
2.2 Why Artificial Neural Networks?	30
2.3 Historical overview	33
2.4 The biological neuron and neural network.....	36
2.5 The artificial neuron model	38
2.6 ANN architectures.....	42
2.7 ANN learning algorithms	52
2.8 ANN input pre-processing	66
2.9 ANN optimisation algorithms	68
2.9.1 Overview.....	68

2.9.2 Second-Order Sensitivity Analysis Algorithm	73
2.9.3 Optimal Brain Surgeon Algorithm.....	85
2.9.4 Complete MFANN optimisation scheme.....	88
2.10 ANN Modelling Examples.....	91
2.10.1 Overview.....	91
2.10.2 Fourier Series Generator	91
2.10.3 Multi-dimensional function generator	97
2.11 Summary	100
CHAPTER 3 INTRODUCTION TO MIE SCATTERING THEORY.....	102
3.1 Introduction	103
3.2 Historical overview	103
3.3 Electromagnetic theory elements	105
3.4 Scattering by a single spherical particle	114
3.5 Nephelometric scattering and turbidity measurements	128
3.6 Scattering and turbidity of ISO Arizona Fine dust.....	135
3.7 The inverse scattering problem	152
3.8 Summary	154
CHAPTER 4 SCATTERING NEPHELOMETRY APPLICATIONS	155
4.1 Introduction	156
4.2 Binned AF sand volume fraction estimation problem.....	156
4.2.1 Theoretical sample preparation.....	156
4.2.2 Nephelometric model setup	157
4.2.3 Theoretical results.....	158
4.2.4 Volume fraction estimation model.....	163
4.3 Filtered AF sand type detection problem	167
4.3.1 Experimental sample preparation	168
4.3.2 Experimental setup.	173
4.3.3 Experimental procedure	177
4.3.4 Experimental results.....	179
4.3.5 Modelling the experiment.....	189
4.3.6 Filtered AF sand type detection results.....	198
4.3.6.1 First solution	203
4.3.6.2 Second solution.....	207
4.4 Summary	214

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS	216
5.1 Theoretical conclusions.....	217
5.2 Experimental conclusions.....	219
5.3 Recommended extensions	220
APPENDIX A REFERENCES	223
APPENDIX B LIST OF FIGURES AND TABLES	238
B.1 List of figures.....	239
B.2 List of tables	242
APPENDIX C NOTATION	243
C.1 Symbols	244
C.2 Acronyms.....	256
APPENDIX D SOFTWARE OVERVIEW	258
D.1 Introduction	259
D.2 Fourier series generator	259
D.3 Multi-dimensional function generator.....	262
D.4 Scattering and turbidity of ISO Arizona Fine dust.....	265
D.5 Binned AF sand volume fraction estimation problem.....	267
D.6 Filtered AF sand type detection problem	270
D.6.1 Experimental results.....	270
D.6.2 Filtered AF sand type detection results.....	273
D.6.2.1 First solution.....	273
D.6.2.2 Second solution.....	275
D.6.2.2.1 First MFANN	275
D.6.2.2.2 Second MFANN.....	278

CHAPTER 1

INTRODUCTION

♪ *Give me a reason...* ♪
(*The Corrs, In blue, 2000*)

1.1 Chapter overview

One of the aims of particle science and technology is the research and development of new and improved ways of continuously measuring mass, size (diameter), volume and other physical properties of solid impurities suspended in fluids. The results obtained find application in many diverse areas. For example, food, drink, potable water, drug, pigment, powder and other consumer products can be processed under more efficient control procedures that increase product throughput and quality and hence maximise manufacturing industry profits. Environmental pollution and climate change caused by industrial or land waste, chemicals, accidents and other human activities can be controlled more effectively or challenged faster and hence natural life and environment are less affected. Finally, contamination of fluids used in mechanical systems with unwanted solid particles may be kept to minimum so that machine damage is prevented.

Like many others, solid impurity measurement techniques are intended to satisfy two conflicting requirements: *monitoring time* and *accuracy*. The degree to which each of these goals is achieved determines the class the corresponding technique belongs to. Specifically, *metrological techniques* generally give the most accurate results but have poor sampling times, great complexity and high cost. As a matter of fact, these techniques are more suitable for off-line applications. *Surrogate techniques* attempt to extract solid particle information from appearance descriptors, such as turbidity, colour, fluorescence and odour. These techniques are suitable for on-line and in-situ monitoring because they are simple and inexpensive, but often fail to associate a surrogate measurement with a solid particle parameter of interest. On the other hand, *characterisation techniques* provide partial but reliable particle information from large collections of suspension parameters readily obtainable. Therefore particle characterisation is mainly suitable for applications that demand on-line and in-situ monitoring and control of a small number of solid particle parameters regarded as critical.

This chapter starts with a brief presentation of the main metrological and surrogate techniques existing nowadays. After that, the characterisation technique of scanning nephelometry, an extension to turbidity and alternative to field scanning, is described and proposed as ideal for a number of turbidity applications. Finally, an overview of this study is provided for reference purposes.

1.2 Metrology

The field of particle metrology encompasses a variety of techniques developed for the accurate determination of particle size (diameter), size distribution and morphology (shape and texture). The particles of interest are usually, though not necessarily, distinct inhomogeneities placed in an otherwise homogeneous medium. A non-exhaustive survey of metrological techniques follows.

1.2.1 Microscopy

Microscopy is based on the observation of visible light or electrons and radiation returned from the particles concerned upon illumination by a diffused visible light source or bombardment with electrons respectively. The former method is called *optical microscopy* while the latter is named *electron microscopy*.

Optical microscopy [1] is often used for examination of particles from about 3 μm to 150 μm . Electron microscopy usually takes one of the forms called *transmission electron microscopy* (TEM) [2] and *scanning electron microscopy* (SEM). TEM floods particles in the 2 nm to 1 μm size range with a high-energy electron beam and allows the analysis of bright field images formed on a fluorescent screen or a photographic plate by secondary electrons scattered elastically at forward angles. In SEM a fine beam of medium energy electrons is caused to scan across a sample of particles ranging from 20 nm to 1 mm in size in a series of parallel tracks. The elastically scattered secondary electrons and in-elastically backscattered electrons emitted due to interaction of the fine electron beam with the particles are displayed on CRT screens and examined. Other electron microscopy methods are the *scanning transmission electron microscopy* (STEM - see [3], page 148) and *scanning tunnelling electron microscopy* (STM) [4] ones.

The most severe limitation of optical microscopy is its small depth of focus so that, for a sample having a wide range of particle sizes only a few particles are in focus in any field of view. Therefore the examination of a large number of fields is essential in order to obtain a statistically significant particle size distribution. The tedious task is facilitated by the use of customised image processing systems. However, these systems cannot discriminate artefacts as readily as human operators. Most electron microscopy methods are unable to measure charged non-conducting particles correctly because the particle charge spreads the scanning electron beam around the par-

ticles concerned and makes sharp focusing impossible. Also, electron microscopy is sensitive to gradual accumulation of dirt on the particle sample and thus requires high vacuum instruments to reduce this effect.

Microscopy is often used as an absolute method of particle size analysis since it is the only technique in which individual particles are observed and measured. It is useful not only for particle size measurement but also for particle morphology (shape and texture) evaluation. Particle shape is defined either qualitatively (acicularity, roundness etc) or quantitatively, e.g., as shape factors obtained by comparison results of perpendicularly oriented diameters. Microscopy is sometimes performed on a prepared particle sample prior to use of another metrological technique in order to examine the quality of particle dispersion in that sample and thus the effectiveness of the particle dispersing procedure that was applied.

1.2.2 Sieving

Sieving has been used since early Egyptian times for the preparation of food-stuffs. The process is briefly described as follows. Firstly, a particle sample is placed on a sieve containing openings of a fixed size. Next, the sieve is agitated in such a manner that particles that can pass through the openings do so. Sieving is accelerated by the use of several sieves stacked on top of each other with the sieve containing the coarsest openings placed on top. The set of sieves is vibrated until the residue on each sieve contains particles that can pass through the upper sieve and cannot pass through the lower sieve.

The technique can be applied for the determination of cumulative particle size distributions ranging from 5 μm up to 100 mm. Punched plate sieves are used for coarse size separation (1 – 100 mm) [5], while woven wire sieves classify medium-sized particles (20 μm – 1 mm) [6] and electroformed sieves screen fine particles (5 – 500 μm) [7]. Woven wire sieves generally have pseudo-square apertures, but punched plate and electroformed sieves are available with round and rectangular apertures.

Measuring particle size distributions by sieving is simple and inexpensive. However, the use of the method presents three major difficulties. Firstly, the aperture size of a typical sieve has considerable tolerance that worsens with size decrease. Secondly, sieves are easily damaged in use. Thirdly, the particles must be effectively presented to the sieve apertures; otherwise the finer particles may not pass through the sieve and block its openings. These problems are minimised when the sieves are cali-

brated prior to use, the sieving procedure is standardised to provide accurate and reproducible results (see [3], pages 169-175), and the sieves are checked on a regular basis for signs of wear and tear.

Sieving is particularly useful since particles are sorted into categories solely on the basis of size, independently of other properties (density, surface etc). The technique can be used to classify dry or wet powders and generates narrowly classified fractions.

1.2.3 Sedimentation

The sedimentation technique is based on the settling behaviour of a single sphere in a fluid of infinite extent under the gravitational (or centrifugal) force and the opposing drag force. Specifically, a unique relationship between the settling velocity of the sphere and its diameter has been shown by a multitude of experiments and summarised into the *Stokes equation* (see [3], pages 227 and 283). The equation also applies to a distribution of particles settling together in a fluid of finite extent provided that the particle concentration is small (typically less than 0.2% per volume).

The determination of particle size distribution by sedimentation is performed in practice by a variety of procedures that are classified according to the principles outlined in table 1-1. The particles may be homogeneously diluted in the fluid used by the measurement procedure (*homogeneous* suspension) or floating on top of it (*line-start* suspension) when sedimentation begins. The particle-settling rate can be determined by monitoring either the concentration change at the measurement zone of a glass container drawn at a fixed depth below the surface of the particle suspension (*incremental* measurement) or the weight of particles settled out of the suspension (*cumulative* measurement). Finally, the field that makes the particles settle may be either gravitational or centrifugal as mentioned above.

Gravitational sedimentation methods have limited worth for particles smaller than about a micron due to the long settling times required. In addition, most sedimentation devices suffer from the effects of convection, thermal diffusion and Brownian motion. These difficulties are reduced and the settling process is accelerated substantially by centrifuging the suspension. Particle size distributions ranging from 10 nm to 500 μm can be measured within a reasonable time by advanced centrifuge apparatus available nowadays (see [3], page 318); the wider the real particle size distribution, the less accurate the measured particle size distribution, however.

Table 1-1 – Principles of sedimentation techniques

Suspension Type	Measurement Principle	Force Field
Homogeneous	Incremental	Gravitational
Line start	Cumulative	Centrifugal

Sedimentation is widely used for particle size analysis since it provides particle size distributions that relate to unit operations (*particle classification*) and many end-use properties (the hiding power and gloss of pigments). The results obtained by the technique over a period of time may be of rather limited accuracy when viewed independently but they can definitely show size distribution changes when compared to each other.

1.2.4 Stream scanning

The stream scanning technique examines the interaction between each and every particle of a representative set and an external field in order to measure the size of that particle and hence the size distribution of the whole population.

Depending on the kind of external field used and parameters monitored for possible changes, the technique is met in one of the forms displayed in table 1-2 (see page 8). If the *Coulter* principle is employed, the interrogating field is electrical and the particle volume is proportional to the change in the electrical impedance as particles pass through the field (table 1-2a). Alternatively, the amount of light blocked as a particle passes through a light beam allows the measurement of the particle's projected area (table 1-2b). With a rotating or scanning beam, smaller in size than the particles, the pulse length is a measure of a random chord length (table 1-2c). Light scattered in the forward direction or at right angles by a particle is dependent on particle size; the former can be collected by a photo-detector and measured (table 1-2d and table 1-2f). The signal received by the detector is greatly enhanced with the aid of an elliptical mirror (table 1-2e). Scanning beams, in collaboration with back-scattering detectors, are also used for chord size determination (table 1-2g). *Interferometers* [8, 9] operate as a function of phase shift between a split laser beam, one passing through the particle and the other through the particle suspension (table 1-2h). The time it takes for accelerated particles to pass through two laser beams is a measure of their

aerodynamic size (table 1-2i) [10]. Finally, the *phase Doppler* method [11] can be employed to determine particle size from the interference pattern of a particle passing through the intersection of two laser beams (table 1-2j).

Stream scanning is generally limited to low concentration suspensions. Furthermore, the conversion of a number distribution returned by the technique to mass (volume) distribution can result in gross errors unless the width of the former is narrow or millions of small particles are measured. Therefore stream scanning is usually applied off-line so that both aforementioned limitations of the technique can be overcome. Stream scanning measures particle sizes spanning the 50 nm to 10 mm range.

1.2.5 Field scanning

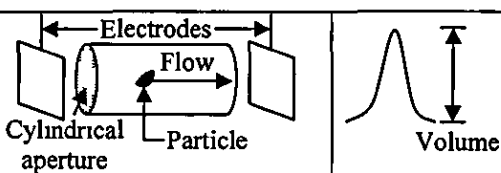
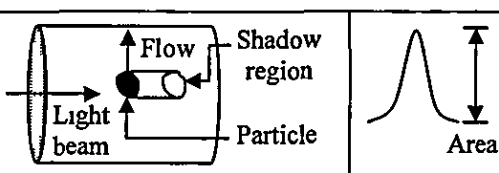
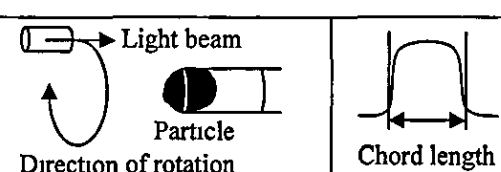
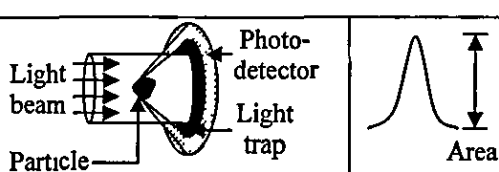
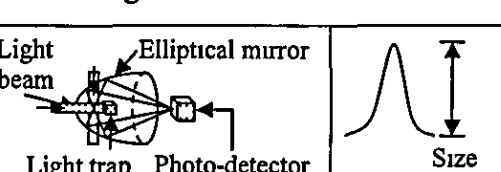
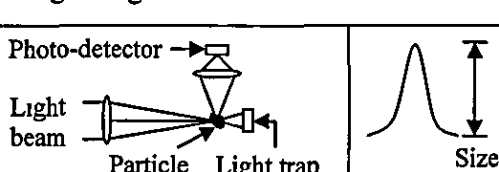
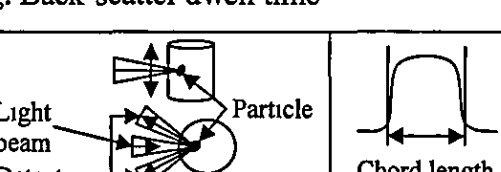
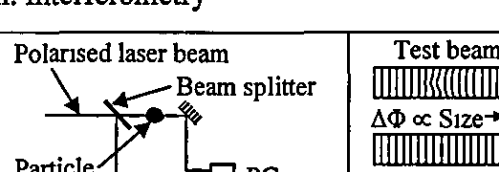
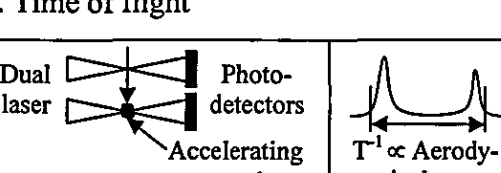
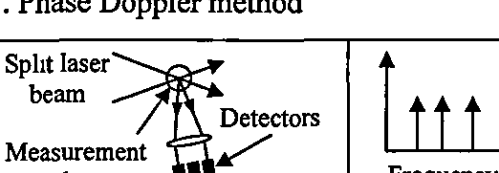
The field scanning technique infers the size distribution of an assembly of particles from the interaction between the assembly and an external field. The particle size range that can be detected by the technique extends from 1.5 nm to 3.5 mm.

The simplest way field scanning is applied is by classification of the particles concerned into two (or more) size bins and sensing of the mass percentage of every bin by static noise measurement, X-ray or β -ray [12] field attenuation, flow rate comparison [13, 14], or even air pressure changes (*single point analysers*). The *low angle laser light scattering* (LALLS) method [15] is a more sophisticated application of the field scanning principle which calculates the size distribution of an assembly of opaque or large (compared with the wavelength of light) particles by deconvolution of the diffraction pattern projected on a planar detector array as soon as the particles are illuminated by a collimated laser beam [16]. Alternatively, the *ultrasonic attenuation* method works out the size distribution and concentration of a particle assembly by deconvolution of the energy losses ultrasonic waves of different frequencies suffer upon interaction with the particles. Finally, the *photon correlation spectroscopy* (PCS) method [17] performs particle sizing of sub-micron particles by (a) deconvolution of the autocorrelation function of the radiation scattered at right angle from the assembly due to interaction with a collimated laser beam (*through dynamic light scattering*) [18] or (b) power spectrum analysis of the sum of a reference laser beam with its back-scattered version generated by the assembly (*controlled reference*) [19].

Field scanning instruments are ideally suited to on-line analysis because they provide fast results that can be updated quickly enough for monitoring purposes. However, particle concentration is a matter of concern for all optical field scanning

methods, as high concentrations result in multiple light scattering effects that make all reported particle size distributions biased towards the fine size end. Additionally, the outcomes reported by two instruments employing the same deconvolution-based method may differ significantly as a result of the numerical algorithm and initial assumptions selected to perform the deconvolution operation in each instrument [20]. Therefore any results obtained should always be interpreted or even corrected appropriately.

Table 1-2 – Stream scanning methods

a. Electrical resistance 		b. Light blockage 	
c. Dwell time 		d. Low-angle forward scatter 	
e. Solid angle scatter 		f. Right angle scatter 	
g. Back-scatter dwell time 		h. Interferometry 	
i. Time of flight 		j. Phase Doppler method 	

1.3 Surrogates

Surrogates are appearance descriptors of fluids often contaminated with solid particle impurities. Apart from their significance for the aesthetic perception of the suspension considered, surrogates are empirical indicators of suspended matter properties such as mass, volume or size distribution. The popularity of these descriptors is proved by the commercial availability of numerous instruments that are able to perform surrogate measurements under virtually any environmental condition and use empirical models to compute estimates of the suspended particle properties of interest. The main surrogates are turbidity and colour; more details about them are given in §1.3.1 and §1.3.2.

1.3.1 Turbidity

1.3.1.1 Overview

Turbidity, cloudiness in water, is the reduction of clarity or transparency of a fluid sample due to the presence of suspended and colloidal matter such as clay, silt, organic and inorganic matter and microscopic organisms. Turbidity should not be confused with colour (a darkly coloured liquid can still be clear and not turbid), however it is affected by intensity and hue of colour, direction of illumination, background, and optical path length.

Turbidity is optically based and hence not subject to non-optical parameters like temperature, viscosity, pH or conductivity. However, turbidity is not empirical, i.e., cannot be filtered out and measured separately from any particle suspension. Therefore turbidity is not absolute. Only similar instruments employing similar techniques can give comparable turbidity measurements. The main turbidity acquisition methods met in practice are the following:

- a) The *Jackson candle* method was the first ever used at waterworks and wastewater treatment plants for measuring turbidities of incoming raw waters and treated wastewater effluents. The associated equipment consisted of a long glass tube supported over a so-called standard candle. Water was added to or removed from the tube until the image of the candle flame became indistinct. The depth of the water in the tube was read off a calibrated scale etched into the side of the tube, and results were reported numerically as *Jackson Turbid-*

ity Units (JTU). The lowest turbidity that could be obtained with this method was 25 NTU (see [21] for NTU definition).

- b) *Comparative* methods are used in shallow water and determine turbidity by matching the turbidity of a water sample to a standard of known turbidity either with a target at the bottom of the tube or with a turbidity comparator.
- c) The *Secchi disk* method is often used to measure turbidity in the deeper waters of lakes, ponds, rivers and estuaries. The disk is either white or marked with black and white quadrants and its diameter is about 8 inches. The disk is lowered into the water on a calibrated line and the depth is noted where the disk just disappears from sight. The disk is then raised until it is visible. The average of these two distances is known as the *Secchi depth* and used as a distinct turbidity measure.
- d) The *optical turbidimeter* method is the most applied one nowadays. The associated devices being commercially available are divided in two categories regarding their principle of operation. *Attenuation turbidimeters* measure the loss in irradiance of a narrow parallel light beam passing through a known path length of the solid particle suspension considered. Turbidity is expressed in units of *percent transmittance* or *Parts Per Million (PPM)* of SiO_2 contained in a reference suspension called *Diatomaceous Earth (DE)*. *Nephelometric turbidimeters* have the detector aligned at right (or another) angle to the beam, and therefore measure scattered light. The scattered irradiance is often divided by the irradiance detected at line of sight to compensate for irradiance losses not attributed to suspended particles. Turbidity is often reported in *Nephelometric Turbidity Units (NTU)* in agreement with ISO 7027 standard [21]. The brewing industry prefers the *European Brewery Convention (EBU)* and *American Society of Brewing Chemists (ASBC)* turbidity units established by the European and American brewing societies respectively. However, all these units are derived from procedures based on standard formazine solutions [22]. As a matter of fact, constant rates between the three units can be established. These rates are taken from [23] and displayed in table 1-3.

Nephelometric turbidimeters are preferred when the impurities suspended in a fluid are of low concentration, while attenuation turbidimeters are employed at higher impurity concentrations. The reasoning behind this is simple. At low concentrations, the light scattered by the suspended particles has room to propagate freely, hence the

angular detector is more sensitive to particle concentration change and the signal returned by the nephelometric turbidimeter is proportional to that concentration (*single scattering regime*). In contrast, the light loss of the transmitted beam is limited due to the presence of only few scattering particles in the medium, thus hardly detectable by an attenuation turbidimeter. At higher impurity concentration, however, the situation changes abruptly. The suspended particles closer to the light source tend to screen off the incident light, and the scattered light can no longer propagate freely (*multiple scattering regime*). As a result, the scattered irradiance no longer increases in proportion to the concentration and nephelometric turbidimeters give meaningless turbidity values. On the other hand, the irradiance loss of transmitted light becomes considerable and hence detectable by attenuation turbidimeters. Attenuation turbidimeters apply the *Lambert-Beer law* [24] to convert turbidity measurements to suspended particle concentrations, while nephelometric turbidimeters perform the same conversion based on *Mie scattering theory* [25]. Typical detection ranges for nephelometric and attenuation turbidimeters are 0-10 NTU (electronic linearisation increases the upper limit to 2000 NTU) and 50-20000 NTU respectively [26].

Table 1-3 – Comparison of formazine-based turbidity units

	1 NTU	1 EBC	1 ASBC
NTU	1	4	0.057
EBC	0.25	1	0.014
ASBC	17.5	70	1

Turbidimeters are simple and inexpensive instruments that are able to measure turbidity continuously, on-line or off-line, in laboratory or process environments. As mentioned above, turbidity is proportional to suspended particle concentration under certain conditions. The surrogate can also be used for the estimation of mono-modal particle size distributions and refractive indices of suspended solids [27]. Therefore it comes to no surprise that turbidity has become so popular in the industrial, health and safety sectors, to name only a few. A representative set of applications employing turbidity nowadays follows.

- a) The potable water treatment industry uses turbidity for raw water quality monitoring, raw water flocculation control and sand filtration monitoring of treated water [28]. When raw water comes from the surface of earth, the turbidity value obtained is used to control the amount of flocculent added at the next processing stage. If spring water is tapped, natural phenomena like snow break and thunderstorms can render the spring temporarily unusable. In this case, turbidity allows real-time decisions to be taken whether raw water will be accepted for treatment or discarded [29]. Other turbidimeters placed between the settling tank and sand filters monitor the efficiency of a crucial treatment step for solid impurity elimination, i.e., the flocculation process. Finally, turbidity measurement immediately after the sand filters is performed for two purposes: monitoring of the actual filtration for breakthroughs, and optimisation of the sand purging (backwash) process [30]. Sand filtration is essential not only because it improves the efficiency of the previous flocculation step, but also precipitates iron salts dissolved in *mineral* raw waters [31]. If present, these salts can oxidise and precipitate as hydroxide to form unattractive sediments in water bottles and blocking sediments in water pipes.
- b) Electricity producers that use fossil fuels (oil, gas, coal) or nuclear fuels as energy source employ turbidity in their plants for a number of different purposes [32]. Firstly, the treatment equipment that feeds the water and steam circuits includes the same treatment steps before the ion exchangers that are found in potable water treatment, and hence incorporates the same detection systems for turbidity. Secondly, the iron concentration of the water flowing in the water circuit is constantly monitored via turbidimeters to detect early signs of pipe corrosion. Finally, the efficiency of filtration equipment built to prevent pollutants from entering the atmosphere is determined by turbidity monitoring of the soot (solid) [33] and sulphuric acid (droplet) [34] concentration in the flue gas that leaves the combustion chambers.
- c) One of the interests of oceanography is the concentration measurement of substances such as inorganic matter from land drainage, plankton, disintegrated cells and humic organics that are found suspended in seawaters [35]. The effect of these impurities on aqueous life is critical for many reasons. Suspended particles absorb light and emit thermal energy, thus increase the overall seawater temperature. The reduction of light penetration efficiency due to in-

creased light absorption in the sea surface causes a decrease in the rate of photosynthesis and a decrease in the amount of oxygen dissolved in the seawater. As suspended particles settle, they can impair the habitat needed for fish spawning and aquatic macro-invertebrates. They can also clog the gills of fish and the breathing apparatus of invertebrates. Miniature *backscatter turbidimeters* are available for *ocean profiling*, i.e. estimation of plankton density and total suspended solid particle concentration, at sea depths up to 6 km [36, 37].

- d) A common objective of hydrological and environmental sciences is to determine the quality of water found in rivers and lakes [38]. Natural water quality is affected by a number of factors such as temperature, microbes, agricultural chemicals, natural dissolved substances, industrial effluents, suspended sediment derived from the land bordering the river (lake) or the river (lake) bed, and other suspended material coming from natural or anthropogenic sources. As far as the suspended sediment is concerned, its total mass and size distribution are the two important factors. Increased sediment load transported in rivers means enhanced land erosion with major consequences not only in agriculture and forestry (both directly through loss of soil and indirectly through problems caused by reduction of the capacity of irrigation reservoirs), but also in navigation in both rivers and harbours, in hydroelectric power generation (through siltation of reservoirs), and in fisheries. Sediment transport also affects the efficiency of river channels and hence the chances of flooding. On the other hand, the particle size distribution of suspended sediment affects the total particle surface area being able to absorb toxic chemicals and pollutants such as phosphate, pesticides, organic industrial effluents, heavy metals and radio-chemicals. Particles smaller than 63 μm are more effective in absorbing these pollutants and transporting them over long distances, and thus seriously deteriorate water quality and endanger aqueous life. A *currentometer* monitors the velocity of natural water in-situ and a turbidimeter placed next to currentometers estimates the concentration of transported sediment in the water. The two outcomes are multiplied and the product is integrated over time in order to obtain the total sediment mass transported at a given time period. Natural water samples are gathered occasionally by an automatic sampler and analysed in the laboratory at a later time to find the sediment size distribution.

- e) Alcoholic beverages such as wine, whiskey and cognac contain small amounts of fatty acids, which tend to precipitate at low temperatures (during transport, in the refrigerator) and cause turbidity [39]. To prevent this undesired event, the beverage industry cools and filters the spirits prior to bottling. A turbidimeter is used to monitor the filtration step continuously and thus ensure both product quality and optimal filter utilisation.
- f) Breweries employ turbidity in the sectors of brewing water treatment, wastewater treatment and beer production [40]. Depending on the quality of the available raw water, treatment of the brewing water may require the same turbidimeter-controlled techniques as potable water treatment (flocculation, sand filtration). In wastewater treatment, turbidity measurement is used before and after the physical and biological treatment steps. Specifically, turbidimeters monitor the solid particle concentration in the raw wastewater and the turbidity of the effluent before it is released to the sewer. In the actual production process, turbidity is used to check the wort lautering, the yeast addition, and the beer filtration steps. Wort lautering is the process of separation of the mesh into the clear liquid wort and residual grain, and is done by means of mash filters or by sedimentation in the lauter tab [41]. Yeast is added to wort liquid immediately after wort lautering for fermentation; the amount of yeast added is controlled via turbidity measurement [42]. Beer filtration is performed immediately before beer barrelling, bottling or canning and then shipping. The process removes protein and tannin colloids, yeast cells and filtration auxiliary particles that are responsible for shelf life shortening and appearance (clarity and brightness) damaging of the end product [43].
- g) The detection of oil mist in all engine rooms of ocean-going tankers is another practical application of turbidity [44]. In the event of leakage in one of the high-pressure lines feeding oil to the big diesels, fine diesel oil mist is created. Because the mist is highly explosive, it is a matter of life and death to detect even tiny amounts quickly and signal their presence. This is possible with turbidimeters fitted with multiple-sampling equipment for simultaneous monitoring of all critical points.
- h) The sugar production industry is another turbidity user [45]. The thin juice extracted from the sugar-beet passes through three purification steps: addition of milk of lime for preliminary and main separation, precipitation of the calcium

carbonate with carbon dioxide (saturation), and removal of the resulting sludge by sedimentation or filtration. If a defect occurs in the filtration step, the solids suspended in the sugar juice will be deposited in the ion exchangers of the subsequent softening step or in the evaporators, thus necessitating time-consuming cleaning work. Turbidimeters are in place to detect any filtration defect early and signal filter replacement to the operator.

- i) The CO concentration and visibility are used to describe the air quality in road tunnels [46]. The former parameter increases due to the combustion process that takes place in gasoline engines, while the latter is reduced by roadway dust and by the soot spewed out by diesel engines. Maintenance of the prescribed limits is crucial in terms of driver safety: CO concentration because of the toxicity of this gas, and visibility because of accident danger. Because the values of both parameters are used to control the speed of tunnel ventilation fans, they are largely responsible for driver safety and also for keeping down the tunnel's running costs. The CO concentration was critical in the past, but the introduction of catalytic converters and the popularity of diesel engines have increased the importance of visibility. Turbidimeters placed at accessible spots outside the traffic zone and connected with air-feeding systems of powerful pumps and long pipes obtain measurements convertible to *standard visibility* values by *Koschmeider formula* [47].

1.3.1.2 Limitations

Unfortunately for the users of turbidity, there are many theoretical as well as practical drawbacks and limitations associated with the surrogate. These problems arise from the fact that scattered light, thus turbidity, depends on the concentration of impurity particles in a suspension sample (that is finally the parameter which should be measured with turbidity units in most applications) but also on the scattering angle, wavelength of light, solvent colour, properties of the scattering particles (size, shape, colour, refractive index) and practical constraints (reflection and absorption and coating losses at turbidimeter windows, ageing of light source, stray light). Therefore:

- a) Turbidimeters are calibrated using linear fits between the scattered light irradiance and the particulate matter concentration determined by a reference method. ISO 7027 compliant instruments are calibrated with standard formazine suspensions [21]. However, the adoption of formazine as reference stan-

dard has several disadvantages. First of all, formazine requires careful handling because it is toxic. Next, the preparation temperature of formazine suspensions must be kept to the limits prescribed by the ISO standard, or the particle size distribution of formazine changes noticeably (1-2% per °K) [48]. The purity of the water used for the preparation of formazine suspensions is also critical, as it contributes to light scattering (water treated in accordance with ISO 7027 has a residual scatter of about 0.02 FTU) and thus affects both instrument calibration and accuracy of very low turbidity measurements [48]. After that, formazine suspensions are stable for only one week [22], which makes them unsuitable for sporadic recalibration of portable turbidimeters. Finally, the optical properties of formazine are often quite different from those of the turbidity substances found in natural water [38] and other fluids of interest. As a matter of fact, two ISO 7027 compliant turbidimeters with different optical arrangements will give identical results for standard formazine suspension samples but slightly different turbidity outcomes for real samples. Using an instrument outside the turbidity standard might result in deviations of a factor of two or even more [49].

- b) Turbidity values obtained by a given instrument are linearly related to the concentration of impurity particles suspended in the sample provided for measurement as long as the properties of these particles do not change. This assumption does not hold in some applications, however, especially the ones with a wide span of concentration or a variety of dispersed substances (natural water quality [38], oil mist detection). In these cases the best-fit method for the calibration curve of the instrument is non-linear and has to be assessed separately for each installation, e.g. tanker A or B, not just application, e.g. tankers [50]. As a matter of fact, the manufacturing cost of the installation-oriented instrument increases considerably.
- c) Turbidity is affected by the colour of the solvent containing the impurities under consideration. Attenuance turbidimeters record increased turbidity due to the direct absorbance of the incident light beam. In contrast, nephelometric turbidimeters give reduced turbidity because the incident and scattered light is partly absorbed. Time-changing factors (window contamination, light source ageing) also affect turbidity and render distant in time measurements unreliable for comparison. Solvent colour and stochastic parameters should not be

regarded as a problem for the measurement of turbidity *per se*, but rather as parasitic factors that alter the relationship drawn between turbidity and suspended particle concentration in a manner hard to predict [38]. The ISO 7027 standard aims to eliminate these parasitic effects in nephelometric turbidimeters by suggesting the use of infrared (860 nm) light and the expression of turbidity as the ratio of 90° scattered light irradiance by transmitted light irradiance. However, research findings have shown that there is no instrument that measures turbidity independent of solvent colour, let alone anything else [51].

- d) Turbidimeters are subject to the inherent brightening effect, i.e. they give small positive turbidity values for solvents entirely free of impurities [52]. This effect is the result of three independent factors, namely *molecular scatter*, *residual particle scatter* and *stray light*. Molecular scattered light occurs because of scatter on the solvent molecules, or strictly speaking the density fluctuations created by molecular motion. The residual particle scattered light is created by particles remaining in the solvent because they cannot be filtered out. The instrument stray light is the quantity of light produced in the cell of the instrument even in the case of a theoretically pure solvent at absolute zero temperature because of reflection at the cell windows. The inherent brightening effect reduces the sensitivity and increases the error margin of turbidimeters that detect extremely low impurity particle concentrations in fluid suspensions. The monitoring of drinking water filtration, the monitoring of iron oxide content in power plant water circuits and the measurement of dust particles in exhaust gases are examples of applications where turbidimeters show their severe limitations due to this effect. A technique to overcome these limitations is the zero-point correction of the turbidity vs. concentration characteristic with “ultra-pure water”. Nevertheless, the zero point obtained by this method is uncertain (the residual particle concentration and the instrument stray light can change) and can yield negative particle concentration values for certain suspensions.
- e) The ISO 7027 standard requires that all nephelometric turbidimeters should measure scatter light irradiance at 90° with respect to the direction of transmitted light. However, it is not always possible or desirable to collect scattered light at right angle. Oceanographic turbidity sensors must be structurally rigid and small to sustain high water pressures at deep sea levels and this is the rea-

son why they are designed to measure backscatter light irradiance [36]. Overflow cell turbidimeters employed in water treatment plants require less frequent servicing when they measure scattered light irradiance at 135° [53]. Dust (flue gas, sulphuric acid mist, oil mist, road tunnel visibility) measurement turbidimeters [54] detect scattered light irradiance at 15° to reduce their sensitivity to refractive index variations of the dust particles (see [3], pages 358-360). The importance of scatter angle selection in nephelometric turbidity measurement is a matter that deserves consideration and should be addressed in future turbidity standards.

- f) Turbidity is a cumulative effect caused by all particles dispersed in a fluid suspension. As a matter of fact, the surrogate alone cannot classify the impurity particles by size or origin. Nevertheless, particle classification information is important in a number of turbidity applications. For example, the brewer needs to know whether the turbidity of filtered beer is due to the presence of protein and tannin colloids alone, or yeast cells and filtration auxiliaries alone, or both particle groups and their analogies [43]. The environmentalist should be aware of the percentage of small (under $63\ \mu\text{m}$) and large (over $63\ \mu\text{m}$) particles suspended in a turbid natural water sample so that it is possible for them to estimate the associated environmental pollution risks. The potable water treatment engineer wants to know the concentration of *cryptosporidium* (an oocyst) and *giardia* (a protozoan), microorganisms smaller than $5\ \mu\text{m}$, in filtered water in order to prevent the appearance of gastroenteral illnesses to water consumers [55]. In these and other similar cases the classification information required is obtained by metrological techniques.

The aforementioned turbidity drawbacks and limitations are reduced by the *particle characterisation* method explained later in this chapter.

1.3.2 Colour

Colour is the perceptual result of light in the visible ($\lambda=390\text{-}780\ \text{nm}$ [56]) region of the electromagnetic spectrum that is incident upon the retina of human eye. The colour of an object being illuminated by an external light source is determined by three factors: the irradiance profile of the source, the absorbance characteristic of the object and the wavelength dependence of the eye's response to visible light. The measurement of colour is the subject of *colourimetric analysis*, while the measure-

ment of absorbance as a function of visible wavelength is the subject of *absorption spectroscopy*. As far as fluid suspensions of solid particles are concerned, colour is a parameter of aesthetic perception while absorbance is an estimator of solid particle concentration (see Lambert-Beer law [24]).

The human retina has three types of colour photoreceptor *cone* cells, which respond to incident radiation with somewhat different spectral response curves. (A fourth type of photoreceptor cell, the *rod*, is also present in the retina. However, rods are effective only at extremely low light levels; they play no role in image reproduction.) As a matter of fact, three numerical components are necessary and sufficient to describe a colour, providing that the spectral response curve of each cone cell is known. In 1931, the Commission Internationale de L'Éclairage (CIE) adopted standard spectral response curves for a hypothetical *standard observer*. Also, CIE has issued a number of generic *CIE systems*, with every particular system being a mapping of the three aforementioned object colour factors (spectral characteristics) to a *triple* of numerical components. CIE systems differ in the coordinate system used to perform the mapping of the three colour characteristics to the three-dimensional colour space. The CIE systems used nowadays for colour specification include CIE XYZ, CIE xyY, CIE $L^*u^*v^*$ and CIE $L^*a^*b^*$. On the other hand, *empirical systems* are based on visual or other comparisons to define object colours; these systems have specific applicability. The American Public Health Association (APHA) and American Society for Testing and Materials (ASTM) standards describe (potable and waste) water colour in *Hazen* units, where 1 Hazen is the colour of a standard water solution with suspended platinum and cobalt chloride salts of molar ratio 2:1 and total platinum concentration of 1mg/lit (the stock solution recipe is described in [57]). The ASTM standard D156-53T defines how the colour of petroleum products can be expressed in *Saybolt* units. The *Munsell system* describes textile colour as the index number of the best-matching colour sample included in the Munsell swatch book. The *PANTONE system* defines an ink colour by specifying the proportions of standard (or secret) inks that can be mixed to make the colour. Transformations may or may not exist between CIE and empirical systems.

The CIE colour triple of a fluid suspension is usually determined by *tristimulus colourimeters* regardless of the CIE system adopted. These instruments employ a CIE standard illuminant to emit a collimated light beam of known spectral characteristic to a sample of the suspension under consideration. Three optically filtered

photodetectors whose spectral responsivities match the ones of the standard observer's cones measure the irradiance of light passing through the sample in the direction of beam propagation and return the associated CIE colour coordinates. *Visual comparison* and *single wavelength* (390 nm for water, 430 nm for beer) *absorption* methods are used to express suspension colour in empirical (Hazen, Saybolt) units. Human operators perform the visual comparison and *monochromatic colourimeters* (tristimulus colourimeters with a single photodetector that measures the irradiance of transmitted light at a single wavelength) measure colour by single wavelength absorption analysis. All colourimeters are calibrated with a blank (colourless) sample whose absorbance characteristic is standardised by the CIE.

The colour of a fluid solution with impurities is dependent on many factors. First and foremost, suspended and colloidal particles cause turbidity that contributes to the total attenuation profile of the solution [58]. The effect of turbidity on colour is so important that the latter is designated *apparent* if measured with all impurities present in the suspension or *true* when measured after filtration of the suspension through 0.45 μm membranes to remove the insoluble substances. (True colour is perceived to be turbidity-free, however reality tells otherwise. In fact, the filtration of the suspension under consideration is not sufficient to remove colloids of diameter less than 0.45 μm , i.e. particles being comparable to the wavelength of incident light and thus causing scattering and turbidity that is dependent on their size and concentration [59].) Colour is also dependent on factors that affect the solubility and stability of the dissolved and dispersed substances of the suspension such as temperature, age and pH [60]. Finally, the suspension colour is affected by the accuracy of the instrument or human operator used to obtain the measurement; human operators are obviously more susceptible to measurement errors than instruments.

The colour of particle solutions is a parameter of interest for many industrial applications. For example, the potable water treatment industry monitors the colour of filtered water to determine its aesthetic quality (colour values of 15 Hazen units or less are undetectable in a glass of water and agree with the World Health Organisation guidelines [61]). The wine production and brewery industries consider colour an important factor for the determination of product quality and the distinction of different product types. The sugar production industry detects any colour residues in the thick sugar-beet juice that follows the decolourisation step [45]. Colour measurement is of-

ten associated with turbidity monitoring in an effort to quantify or minimise the effect of turbidity on colour.

1.4 Particle characterisation

Particle characterisation is the technique of grouping a particle population with regard to one of their physical dimension or property (diameter, volume, texture etc) in accordance with the problem specifications. When a single group is defined or assigned to all particles, the outcome of a characterisation method degenerates to a statistical parameter of the population (total concentration, average size, total volume etc). Otherwise the outcome is expressed as a set of *fractions* or percentages associated with the particle groups.

A particle characterisation method gives solutions that feature:

- Low cost
- Immunity to variations of “parasitic” parameters (colour, temperature, pH etc)
- Suppression of side effects caused by theoretical and practical limitations
- Adaptability to problem-defined constraints like oval instrument shape [36]
- Calibration with materials similar to the particles considered
- Continuous on-line and in-situ monitoring of all particle properties of interest
- Functionality linked to application than installation

Particle characterisation is suitable for applications that require low-resolution particle size information in real time. Certain turbidity applications could also benefit from the replacement of turbidimeters with particle classification instruments, for example:

- a) In potable water treatment plants, particle characterisation would enable the on-line measurement of the fraction of impurities smaller than 5 μm in potable water and hence the early detection of harmful concentrations of pathogenic microorganisms. Also, particle classification instruments are more successful in the suppression of the inherent brightening effect than ordinary turbidimeters. Therefore classification instruments would be more sensitive and accurate in obtaining small impurity concentration measurements (less than 0.5 NTU in turbidity terms). The same reason can be given to recommend the employment of particle characterisation in fossil and nuclear electricity plants for the early detection of iron particles in circulating water.

b) In natural water quality monitoring applications, particle characterisation would make the continuous measurement of particles smaller than $63\ \mu\text{m}$ possible. As mentioned earlier, these tiny particles are most likely to absorb and transport pollutants. Furthermore, particle classification instruments are less sensitive to water colour and other physical parameters (temperature, pH etc) than turbidimeters. As a matter of fact, classification instruments would give more accurate measurements of total impurity concentration in river, lake and ocean waters.

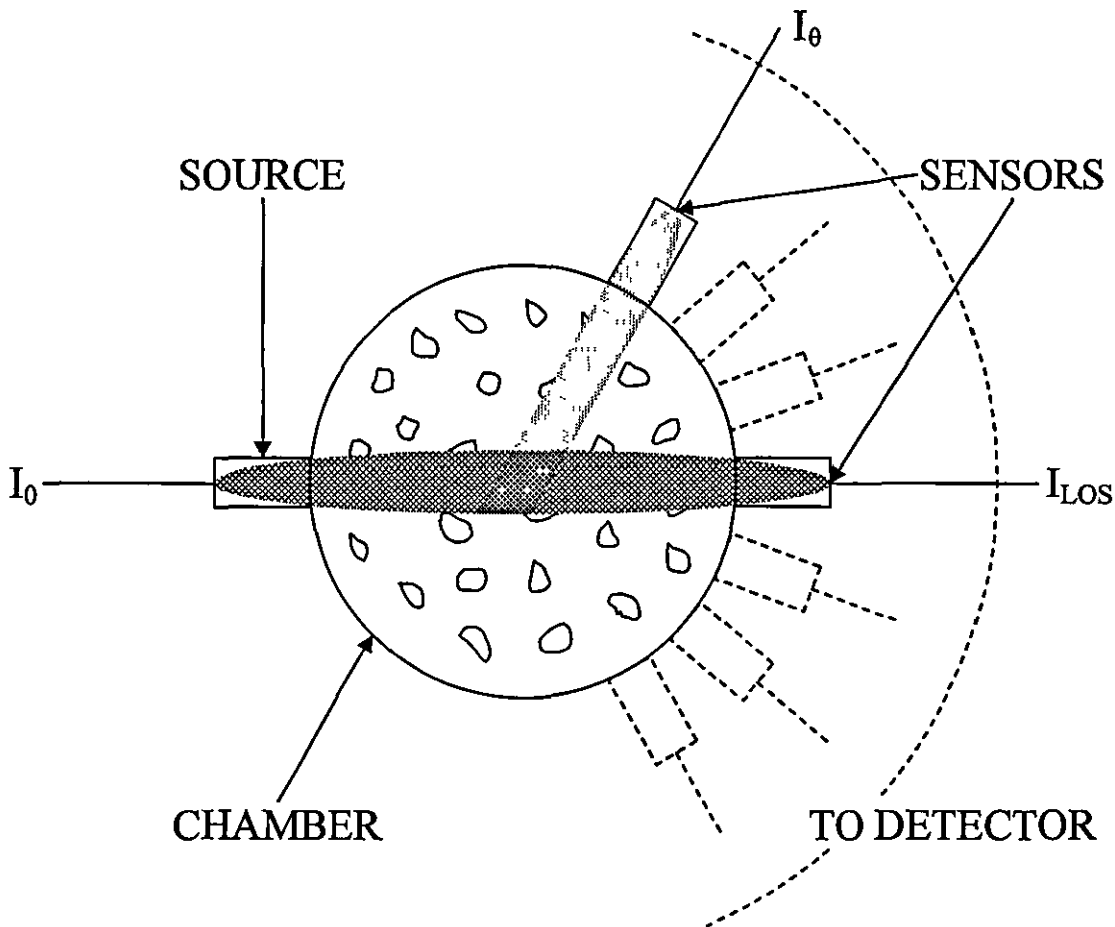


Figure 1-1 – A typical scattering nephelometer

c) In breweries, the on-line classification of impurity particles in filtered beer by texture (colloids, yeast cells, filtration auxiliaries) would become a reality if an appropriate particle characterisation method was employed. Moreover, the amount of yeast added to wort liquid could be controlled by a particle classification instrument instead of a turbidimeter to avoid inaccuracies attributed to beer colour and inherent brightening effect. The employment of particle char-

acterisation in brewing water treatment in a role similar to the one recommended to potable water treatment plants is another possibility.

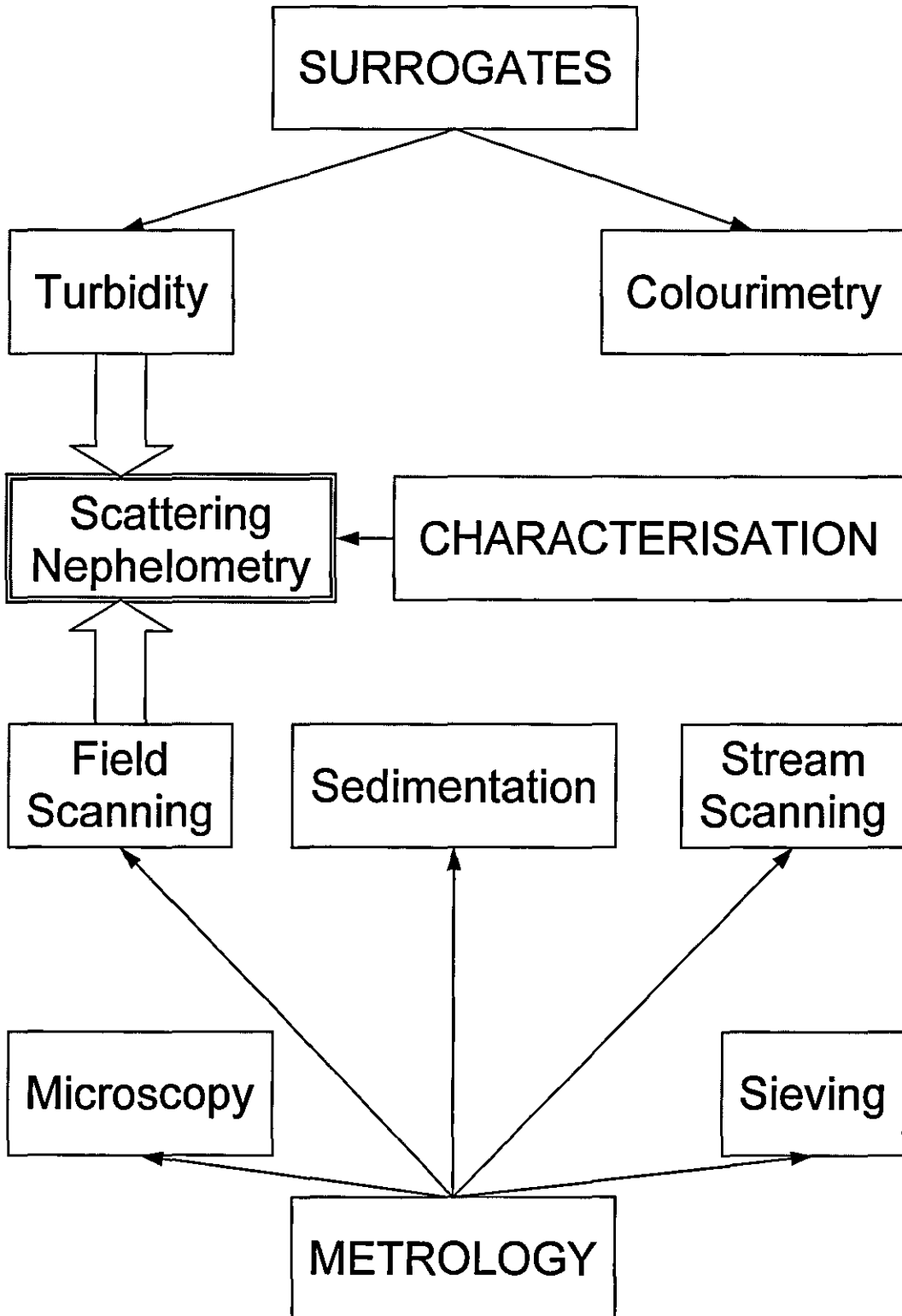


Figure 1-2 – Particle sizing, characterisation and surrogate techniques

- d) In alcoholic beverage production industries, a classification instrument would provide colour-independent concentration measurements of fatty acids remained in the final product. The same argument can also be used to justify the employment of particle characterisation in the filtration of sugar juice that takes place in sugar production industries.
- e) In oil mist detection applications, the replacement of turbidimeters by classification instruments would eliminate the need of complicated and installation-oriented calibration procedures. Particle characterisation methods are designed to provide pre-calibrated solutions that target a particular application.

Scattering nephelometry is a method applicable to characterisation problems of particles dispersed in fluids. Its principle of operation is relatively simple. A limited number of light sources with known emission spectra characteristics are used one at a time to illuminate a representative volume of the particle suspension of interest within a closed chamber (figure 1-1). The irradiances of light scattered at predetermined angles by the suspension are measured by optical sensors and reported to a data analysis system together with the irradiance and spectrum of the emitted light beam. The analysis system processes the available data and returns the answer to the classification problem specified in advance.

Scattering nephelometry is not new science. The method has been known for some time as an extension of turbidity that offers better accuracy and less sensitivity to time-varying factors [38]. The similarities between scattering nephelometry and field scanning methods such as LALLS and PCS are also apparent. Figure 1-2 illustrates the main particle sizing, characterisation and surrogate techniques mentioned so far together with their dependencies and associations.

Two major issues of scattering nephelometric model design are the modelling of the data analysis system and the selection of calibration material. As far as the data analysis system is concerned, the choice of a proper mathematical tool is of fundamental importance. The empirical data fitting technique called Multilayer Feedforward Artificial Neural Network (MFANN) is advantageous to many analytical approaches for a number of reasons: accomplishment of functional approximation and classification tasks from databases of input-output data associations without any additional knowledge, satisfaction of constraints imposed on inputs, outputs and internal complexity, distributed processing, low implementation cost and possibility of structural optimisation are only a few. MFANN optimisation is distinctively significant

because it reduces the number of data analysis system inputs to the absolute minimum. The minimisation of input dimensionality simplifies the design and reduces the cost of the scattering nephelometer (figure 1-1). Besides, a calibration material whose physical and optical properties (size, texture, refracting index etc) are similar to the ones of the particles of interest increases the precision and measurement range of the nephelometric model.

This study focuses on the optimisation aspect of the scattering nephelometric model design. The novel Second Order Sensitivity Analysis (SOSA) optimisation algorithm is theoretically developed and incorporated to an innovative MFANN optimisation scheme that gives excellent network pruning and generalisation results. The ISO 12103-1 Arizona Fine (AF) dust is considered representative of the particles found in natural suspensions and examined for its light scattering properties. Theoretical scattering profiles of water suspensions of filtered AF dust are calculated with the aid of an analytical nephelometric model derived from Mie scattering theory. Experimental scattering profiles of filtered AF dust samples dispersed in water are acquired by a commercial polar nephelometric device. Finally, two separate scattering nephelometric models are designed and optimised to classify the theoretical and experimental suspensions by AF particle volume and size respectively.

1.5 Thesis overview

Chapter 2 introduces Artificial Neural Networks as a candidate modelling tool for the realisation of non-linear mappings between scattering data and suspended particle characterisation parameters. Special attention is given to Multilayer Feedforward Artificial Neural Networks (MFANNs) due to their functional simplicity and universal applicability to problems of function approximation and classification. Data preconditioning methods and training algorithms that improve the accuracy and minimise the learning time of a MFANN model are discussed in adequate detail. A novel MFANN node optimisation algorithm called Second Order Sensitivity Analysis (SOSA) is mathematically described and compared with other algorithms in terms of pruning efficiency and computational complexity. The SOSA algorithm is included with the established Levenberg-Marquardt backpropagation (LMBP) and Optimal Brain Surgeon (OBS) algorithms in an innovative modelling scheme that systematically trains and optimises a MFANN until a fine balance between the opposing goals of acceptable accuracy and essential feature extraction is reached. The scheme is ap-

plied to two trivial nonlinear function approximation problems and the results obtained are examined for consistency and accuracy.

Chapter 3 provides the theoretical background that is necessary for the description of light and the explanation of light scattering phenomena. The Mie scattering theory that describes the interaction of light with a single homogeneous sphere placed in an otherwise simple medium is fully deployed. The irradiance and turbidity measured by an ideal scattering nephelometric device for a tenuous fluid suspension of solid spherical particle impurities are derived analytically. Nephelometric scattering and turbidity profiles of tenuous water suspensions of ISO 12103-1 [62] Arizona Fine (AF) dust are calculated theoretically. The sensitivity of these profiles to adjustable scattering parameters such as concentration, chamber radius and acceptance angle is carefully examined. The inverse scattering problem is briefly explained with special focus on the inappropriateness of conventional mathematical tools to give a unique solution.

Chapter 4 demonstrates a theoretical and a practical application of MFANNs in particle characterisation. A trained and optimised network model estimates the volume fraction per size bin of filtered AF dust dispersed in water from scattering data obtained by a simple scattering nephelometric model. The apparatuses used and procedures followed to obtain experimental nephelometric scattering measurements from a set of water suspensions of two filtered AF dust types are described in appropriate detail. The scattering nephelometric model developed in the previous chapter is applied to the same set of suspensions used in the experiment. The theoretical scattering profiles are compared to the associated experimental profiles in order to verify the appropriateness of the model and estimate the deviation factors between experiment and theory. A second trained and optimised MFANN model is built to recognise the type of AF dust dispersed in an arbitrary experimental sample from the corresponding experimental data.

Chapter 5 reiterates the main scientific conclusions drawn in this study. Hints for further research on the basis of facts and findings of this work are also included for the benefit of science.

1.6 Original contributions

- Introduction of the Second-Order Sensitivity Analysis algorithm as an alternative to the existing First-Order Sensitivity Analysis algorithm for the minimi-

sation of the number of inputs or hidden nodes in Multilayer Feedforward Artificial Neural Networks (MFANNs).

- Presentation of a complete optimisation scheme employing the novel Second-Order Sensitivity Analysis algorithm for MFANN input and node pruning and the established Optimal Brain Surgeon algorithm for MFANN weight elimination.
- Theoretical study of the infrared light scattering properties of tenuous water suspensions of ISO 12103-1 Arizona Fine sand that fill the chamber of an ISO 7027 compatible polar nephelometer.
- Design of two optimal nephelometric models for the solution of a theoretical particle volume fraction estimation problem and an experimental particle classification problem that consider tenuous water suspensions of filtered ISO 12103-1 Arizona Fine sand.

CHAPTER 2

ARTIFICIAL NEURAL NETWORK MODELLING

♪ *T.N.T. for the brain...* ♪
(*Emigma, Le Roi est mort - Vive le Roi!*, 1996)

2.1 The generic modelling procedure

The mathematical tools an experienced engineer would choose to model a particular physical problem are dictated by the nature of the facts already known about it. Those facts are usually expressed in the form of “rules” or “features” governing the underlying mechanisms of the problem and, when applied together, predicting its behaviour at a reasonable accuracy. The main difficulty, however, lies exactly there. In most cases those features cannot be detected and extracted from the physical problem “at a glance”; it usually requires laborious and persistent work to be able to identify some of them. Even if we manage to do so, it may hardly be possible to state them in a universal, unambiguous manner. Furthermore, their relative weight to the formation of responses given by the problem under certain circumstances is another question that cannot be answered until the very late stages of the model construction in the general case. So, what procedure should be followed to fulfil the modelling construction task of an arbitrary physical problem in the most appropriate manner?

An answer to our puzzle may be extracted from a closer reading of the above paragraph. At first, all facts and features associated with a given problem that can be identified and precisely expressed have to be collected before any modelling attempt is made. Existing knowledge about the mechanisms governing certain parts of the problem, or previous research on the ways the problem responds under well-known conditions are the most common resources of assistance in this matter. The next step involves the building of a preliminary, or “guiding” model that should comply with the following terms:

- a) It has to rely upon most, if not all, the features and “findings” collected by application of all existing knowledge to the problem (see above).
- b) It should be *flexible* to further modifications, supervised or unsupervised, which attempt to make its responses more accurate, i.e. be closer to the ones given by the actual problem under the same conditions.
- c) It ought to be *open* to interpretations aiming to the detection of new features driving the physical problem that could not be identified at first glance. New features help to improve the initial understanding of the problem in hand.
- d) It should be *adaptive* to any possible changes occurred in the behaviour of the real problem progressively over time, so that it can always stay within the accuracy requirements imposed to it from the beginning.

- e) It ought to allow optimisations of its structure at any modelling stage (usually the last ones) aiming to improve its efficiency in giving faster responses with fewer known conditions whilst it maintains its accuracy requirements.

When the first model has been devised successfully and shown to give responses reasonably close to the ones observed by the real problem, its evolution is a rather straightforward process. The engineer tries to include more features to the initial model that come from a variety of sources, such as thorough examination of the structure of the existing model for their identification, previous experience of the model's supervisor, or model evolution algorithms. In every case, any further progress achieved in the development of the first model is the result of a slow "trial and error" process: various modification ideas are applied to the initial model, to become permanent only if the modified model is shown by routine tests to be a better fit of the actual problem. The final steps of the model evolution process include model clarification and optimisation for efficiency. Their significance is justified by the fact that they remove all unimportant - for the generation of responses - features from the final model, as well as apply appropriate relative weights to the importance of the remaining features hidden in the details of the model structure. Of course, the final model (as well as its predecessors in the evolution chain) should still comply with all terms imposed for the very first model and mentioned above. As a matter of fact, an important by-product emerging from the final model is a better understanding of the mechanisms governing the response formation of the actual problem.

2.2 Why Artificial Neural Networks?

Having presented the general modelling procedure as above, the interest is now moved to finding the right tools for the modelling of the main problem this work deals with: *inverse optical scattering*. Chapter 3 covers adequately the optical scattering theory and shows why a complex, tedious mathematical model has to be built to explain the theory even for the simplest, marginal cases. It also demonstrates why the inverse problem does not have a unique solution in general, a fact that makes any direct modelling attempt meaningless. Is there any hope for a solution to be achieved then, to start with?

To understand the inverse scattering problem further without having to discuss its details until the next chapter, it is sufficient to compare this problem with the one of identifying an animal from its footprints. Similarly, the latter problem cannot be

given a unique solution either, *unless more information is additionally provided* (animal size, weight, class etc.). However, the footprints may uniquely identify certain animal properties without any need for further information, e.g., a small, shallow footprint corresponds to a small, light animal while a big, deep footprint belongs to a big, heavy animal. Following the parallels drawn between the two problems, it is possible to state that the inverse scattering problem can provide unique solutions under given conditions for certain properties of the medium that caused the light scattering, even if the cause cannot be unambiguously identified. Is it purposeful then to model the inverse scattering problem? Yes, *when* specific properties of the scattering medium are of importance, those properties can be found by solving the inverse scattering problem, and all conditions that assure uniqueness for the final solution are met.

If we try to apply the modelling guidelines mentioned in §2.1 to the inverse scattering problem, the first objective is to *collect and unambiguously express all given facts*. However, the inverse scattering theory can give exact solutions only in marginal and trivial cases of the problem. Nevertheless, those theoretical solutions may be used to obtain as much numerical data as we need for both the optical and physical properties of the medium that causes the light scattering. Another source of information comes from the laboratory, as it is possible to perform a limited number of experiments in order to produce more numerical data for the optical and physical properties of interest. When both pieces of information are combined together, and extra care is taken to remove any ambiguity sprung from spurious cross-links of one data set of physical properties with many data sets of optical properties, we result in the formation of two groups of data (optical and physical) linked by *one-to-one* associations. Although this initial information is too little to give a direct final solution, it can describe by itself a *functional mapping* from the optical parameter domain to the physical parameter one. This mapping is our very first, “guiding” model in the effort to solve the inverse optical scattering problem.

The next step to be taken is the most important for the success of the whole modelling project. That is, the right mathematical tool has to be selected for the realisation of the functional mapping formed by the two groups of data sets already available and their associations. That realisation has to be implemented in such a way that the “guiding” model obtained from it satisfies all necessary terms described by the generic modelling guidelines above. Furthermore, it should offer good chances for further evolution of the initial model to its final form, i.e., the one that will optimisti-

cally provide the answers for the subset of the physical properties of the scattering medium that is of high interest depending on the application circumstances. Therefore, which is the best choice to be made to this case? The author's answer is: *Artificial Neural Networks*. Why? Because, as the remaining of this chapter will demonstrate among others, these networks are designed in such a manner that allows them to realise any functional mapping regardless of complexity that is defined merely by numerical data associations, while any such realisation possesses all characteristics required by the generic modelling procedure and mentioned above.

This chapter is determined to provide the reader with all knowledge that is absolutely necessary for the comprehension and manipulation of Artificial Neural Networks (ANNs). A brief historical overview of the research carried out in the ANN field comes first to be followed by a light coverage of the functionality of the biological neuron and the associated neural network, i.e., the inspirations that led researchers to the design of the artificial counterpart. A detailed presentation of the artificial neuron and as single element and then as the sole structural unit of the most common network architectures encountered in scientific literature comes next. As soon as the method of operation of the ANN architectures of interest is adequately explained to the reader, the next step is the presentation of the learning concept as the key feature of all ANN structures and the passage from conceptuality to implementation via well-described numerical algorithms. These *training* algorithms are compared to each other against efficiency defined in terms of probability of success in achieving training of an arbitrary ANN as the result of the learning process and of the average time required for that process to emerge successful. The notion of data preconditioning is described later as a useful procedure required to increase considerably the chances of successful learning for the ANN structure of interest. After that, the concept of model generalisation is introduced as an important property of any acceptable ANN model solution, and is shown that such a property is obtained or improved by the application of ANN *optimisation* algorithms. The main optimisation algorithms are adequately discussed, and as a conclusion a novel optimisation algorithm is proposed to significantly improve optimisation accuracy at the cost of modestly increasing computational and resource requirements. Chapter 2 ends with a couple of illustrative examples that demonstrate how the ANN mathematical tool can be applied for the modelling of functional mapping problems as function approximations. These examples also demon-

strate the involvement of training and optimisation algorithms as well as data preconditioning in the modelling process.

It has to be stressed that despite the somewhat generic coverage of the ANN theory in §2.3 to §2.6, the main focus will always remain on the use of ANNs as mathematical tools for function approximation purposes due to their application to the inverse scattering problem in chapter 3 and chapter 4.

2.3 Historical overview

Two elements are required at least for the advancement of any technological field to become reality: *concept* and *implementation*. By the former we usually define any novel, revolutionary way of thinking or view of a topic that opens new scientific horizons by the clarity it offers. Instances of concepts may take the form of simple ideas or be expressed more specifically in mathematical terms. The latter element, implementation, is equally important for any concept to find practical application, though. A good example of a concept being the driving force for advancement may be encountered in the history of the heart. That human organ was thought to be the centre of the soul or a source of heat for many centuries until the 17th when medical practitioners finally began to view it as a pump and designed experiments to design its pumping operation. On the other hand, one would reasonably stress that despite all theoretical knowledge necessary for a man travel to the moon existed for long time in fragments belonging to a disparity of fields like astronomy, physics, engineering and medicine, such a travel was not performed until 1969 when all means to implement it existed and were put together. As a matter of fact, the history of any scientific field, let alone ANNs, should include records of conceptual innovations and implementation developments perplexed together.

One would expect that the history of ANNs is relatively long; another would assume that the field has progressed through steady evolution steps, i.e. in a “slow and sure” fashion; a third would think that the major contributors in the advancement of ANN technology were of limited specialty in this field alone. None of them is right however. As it emerges from the study of the following brief reference, the history of ANNs is filled with colourful, creative individuals from many different fields who have struggled since the late 19th century to develop new concepts and test them at times that may easily be grouped to form “intensive” and “recessive” advancement periods. A detailed documentation of ANN history is found in [63].

Pioneering scientists of the late 19th and early 20th centuries, such as William James [64], Hermann von Helmholtz, Ernst Mach and Ivan Pavlov, offer some of the background work for the ANN field. This early work is rather of interdisciplinary nature as it consisted primarily of elements taken from physics, physiology and neurophysiology. It emphasised general theories of learning, vision, conditioning etc., but did not include any specific mathematical models of neuron operation.

Warren McCulloch and Walter Pitts are widely acknowledged as the founders of the ANN field in its modern view due to their work dated back in the 1940s [65]. Specifically, they showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function. Later in that decade the ANN field experienced another major contribution by Donald Hebb. In his book titled "The Organization of Behavior" [66] he suggested that classical conditioning (as discovered by Pavlov) is present because of the properties of individual neurons, and proposed a mechanism for learning in biological neurons. All of those achievements were fundamental for the technological progress of the field, but they were strictly limited in theoretical findings.

Frank Rosenblatt was the first to show a practical application of ANNs in the late 1950s with his invention of the perceptron network and its associated learning rule [67]. He and his colleagues built a perceptron network and demonstrated its ability to perform pattern recognition. At about the same time, Bernard Widrow and Ted Hoff introduced a new learning algorithm and used it to train *adaptive* linear neural networks that were similar in structure and capability to Rosenblatt's perceptron [68]. Those two advancements triggered a great deal of interest in ANN research for about a decade.

The dark ages for the evolution of the ANN field started in 1969 with the famous publication of "Perceptrons" by Marvin Minsky and Seymour Papert [69]. That book demonstrated both in theory and with illustrative examples that the networks created by Rosenblatt and Widrow suffered from the same inherent limitations, i.e. they could solve only a limited class of problems known as "linearly separable". Of course, the aforementioned ANN inventors were aware of these limitations and proposed new networks that would overcome them. However, they were not able to successfully modify their learning algorithms to train the more complex networks. That failure to overcome the limitations imposed by Minsky and Papert in the performance of all ANN structures known in the late 1960s combined with the absence of powerful

digital computers on which promising researchers could experiment caused the collapse of interest in the ANN field by the scientific community for more than a decade.

Although the ANN area remained in the sidelines of scientific research performed worldwide during the 1970s, a few romantic researchers continued their efforts and managed to contribute some important work to that area. Teuvo Kohonen [70] and James Anderson [71] independently and separately developed new ANN architectures that could act as memories in 1972. Stephen Grossberg was another active researcher who investigated and designed self-organising networks. None of those inventions, however, was able to give a fresh start in the mainstream research of the ANN field. Did that attribute to the absence of new ideas or non-availability of powerful digital computers? Probably both.

The 1980s saw the dawn of the new digital computer era in the form of new powerful digital personal computers and workstations that rapidly grew in capability and became widely available. That technological progress soon provided all creative researchers worldwide with the right tools to test and implement their novel concepts for the rapid advancement of the ANN field. When the implementing means were coupled with the inspirations of talented individuals, new exciting technological achievements were expected to follow. And they came faster than expected, attributing to the rebirth of ANNs. Physicist John Hopfield described in a 1982 seminar paper [72] how statistical mechanics could be used to explain the operation of a certain class of recurrent network, which in turn could be used as an associative memory. Few years later, several different researchers discovered independently the backpropagation algorithm for training multilayer perceptron networks, giving an answer to the criticisms Minsky and Papert had made in the 1960s. The invention of that new algorithm is attributed to David Rumelhart and James McClelland due to their publication [73] being the most influential among others, another publication about the same topic is met in [74]. Those two key developments were more than sufficient to spark enthusiasm among ANN researchers worldwide and justify the rapid expansion of knowledge contributed to the field by thousands of new publications and novel practical applications seeing the light until nowadays.

How the ANN field will progress and develop in the future is something difficult to predict, especially given the field's turbulent evolution history. However, all signs of today are very encouraging, suggesting that ANNs will not only have their day but will have a permanent place, not as a solution to every problem, but as a tool

to be used in appropriate situations. Given that the field pumps new ideas from developments in the research of human brain functionality where findings are rather modest nowadays, it is expected that the most important advances in ANNs lie in the future. Possible ways of future advancement of the ANN field are presented and discussed in [75].

2.4 The biological neuron and neural network

The exact anatomy of a biological neuron cannot be given in a single manner due to significant differences existing between them due to their position in the biological neural network and thus their specialised functionality. However, a typical simplified description of such a neuron may be summarised in figure 2-1. Three principal components are identified in that schematic diagram: the *dendrites*, the *cell body* and the *axon*. The dendrites are tree-like receptive networks of nerve fibres that carry electrical signals originating from hundreds of thousands of neighbouring neurons into the cell body. The cell body effectively sums these incoming signals by suitably adjusting the *membrane potential*, i.e., the continuous internal potential of the neuron cell. When that potential exceeds a certain threshold, the neuron can propagate an all-or-none *action potential* for long distances down its single long fibre called the axon to hundreds of thousands of other neurons. However it has to be mentioned that biological neurons are *not* binary, that is, having only an on or off state as their output. Outputs are continuous valued and the membrane potential is converted into firing rate in a voltage-to-frequency-converter fashion. Anyway, the function that maps the membrane potential to action potential, no matter how, is one of the two key features of the biological neuron. The application of this feature to the design of the artificial neuron is discussed in §2.5.

The actual transmission of the electrical signal carried into a neuron's axon to another cell's dendrite terminal is performed in the points of contact of the two components known as *synapses*. Synapses come in a number of different forms, but two basic varieties are of particular note: *excitatory* synapses, which make it more likely that the receiving neuron will fire action potentials, and *inhibitory* synapses which make the receiving neuron less likely to fire action potentials. Also, synapses come in different *strengths* initially determined at birth and continuously modified during lifetime by learning knowledge and experiences. The kind and strength of synapses formed by any individual neuron with its predecessors in the complex biological neu-

ral network determine the outcome of the summation process performed in the body cell, and as such they are defined as the second of the two key features that control the operation of that neuron cell. §2.5 will give more details on how the latter feature applies in the construction of the artificial neuron.

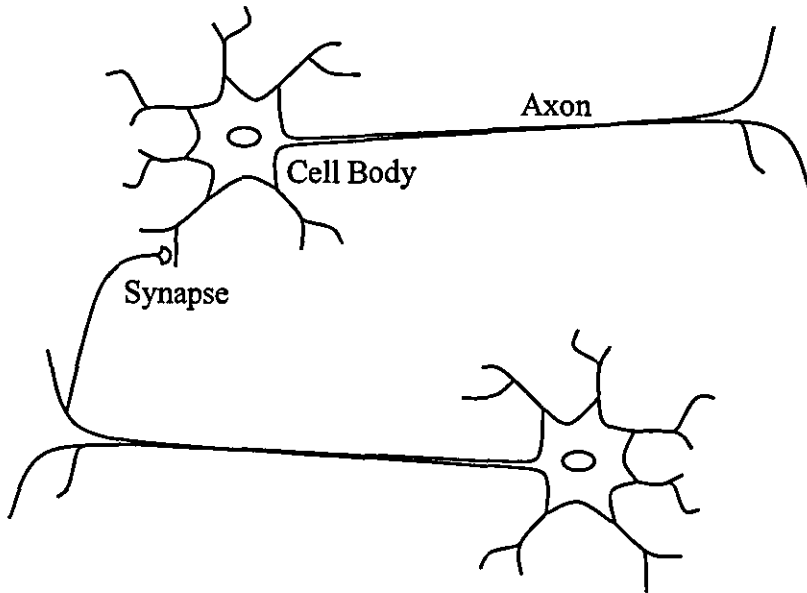


Figure 2-1 – Schematic drawing of biological neurons

The biological neural network can be described in macroscopic level as a complex system consisting of three major functional units: the *sensory system*, the *signal transmission system* and the *brain*. The sensory (visual, auditory etc) system receives information from the outside world and breaks it down into certain parameters that can be *represented* by an appropriate set of values or symbols. Furthermore, the system *pre-processes* the parameterised information locally in ways such as noise reduction, gain control, colour separation, motion detection or edge enhancement, and passes that information to the next functional block, the signal transmission system. The latter converts the incoming information into electrical signals and transmits them to the brain with the least attenuation possible. Finally, the brain processes all its input parameters via its massive structure of hundreds of billions of highly *interconnected* neurons to form a coherent perception of the cause that generated the information captured by the sensory system in the beginning.

The three key features of the biological neural network that can be extracted from its description above are *input representation*, *input pre-processing* and *brain network connectivity*. The first of them is a major research area in the ANN field by

itself and hence its detailed presentation would go far beyond the scope of this work. However, it is useful to mention the three common-sense rules that should apply in any representation case:

- a) Similar inputs usually should give rise to similar representations
- b) Inputs to be separated should be given different representations
- c) If a property or feature of an input is important, it should be embodied in the representation of that input

These rules are applied during the ANN modelling procedures taking place throughout this study and therefore their use will not be explicitly mentioned from this point on unless absolutely necessary. The second feature, input pre-processing, is a vital operation because it removes most of the input data complexity that is uncorrelated with the adaptive and learning parts of the problem modelled by the neural network and hence accelerates significantly the learning and retrieval times required for the construction and function of that network. The application of that feature in ANN design is the subject of §2.8. The last feature, brain network connectivity, is the demonstration of the power of parallelism in the design of efficient computational architectures which give answers to a wide range of problems much faster than today's conventional computers - despite the great difference in the speeds of their computing elements that strongly favours the opposite outcome in that comparison. The study of the neural network connectivity of the brain has produced incredible, though moderate still, results in the design of artificial neuron interconnection architectures capable of solving certain sets of problems quickly and accurately. A subset of those connectivity architectures are briefly presented in §2.6, and the architecture applied to function approximation problems in particular is the focus of an exhaustive analysis covering §2.7 to §2.9.

2.5 The artificial neuron model

A typical artificial neuron resulting from the direct realisation of the basic components and implementation of the major features present in a representative biological neuron is displayed in figure 2-2. The relations between the two neural models are summarised as follows:

- a) The R incoming signals to the biological neuron from nearby cells have been substituted by the column vector \mathbf{p} of scalar *inputs* p_i , $1 \leq i \leq R$, to the artificial model.

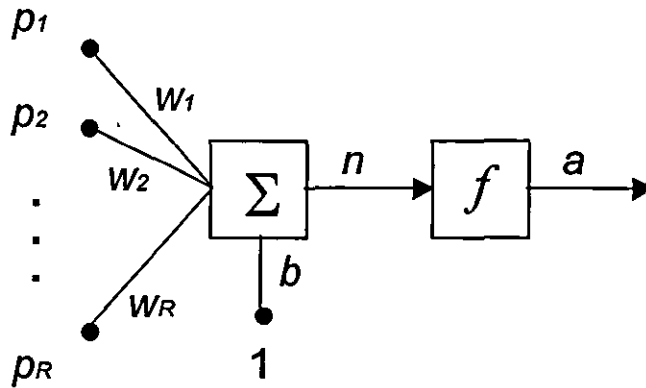


Figure 2-2 – A typical multiple-input artificial neuron

- b) The strengths of the synapses formed at the points of contact of the biological neuron with its input-feeding neighbouring cells have been replaced by the row vector \mathbf{w} of scalar *weights* w_i , $1 \leq i \leq R$, each assigned to every input of the artificial model. Similarly, the excitatory or inhibitory nature of those synapses is now represented in the *polarity* of the respective weights – positive for the former kind and negative for the latter.
- c) The threshold value that the membrane potential has to overcome in order for the biological neuron to activate its output is now represented by the scalar *bias* value, b , in the artificial counterpart. The bias is introduced in the model as an additional fixed input associated with a fixed weight. Part of ANN literature assigns values b and 1 to those two parameters, while the rest prefers to assign the same values the opposite way. Both conventions, however, are mathematically equivalent and therefore interchangeable.
- d) The membrane potential that is raised within the cell body of the biological neuron is represented by the scalar *net input* value, n , that is calculated as the outcome of the weighted algebraic sum of the artificial model's inputs including the bias, i.e. $n = \mathbf{w}\mathbf{p} + b$.
- e) The action potential that is propagated down the axon of the biological neuron as a result of the cell's activation triggered by its input signals, or the absence of such a potential when the neuron cannot be activated by its inputs, are states both determined in the associated artificial neuron model by the outcome of the *transfer function* f . That function accepts a single scalar argument, the net input value n , to produce its scalar output, a , which is considered as the artificial neuron's output, too. In mathematical terms: $a = f(n)$.

Table 2-1 – Common transfer functions of artificial neurons

Name	Input/Output Relation
Hard Limit	$f(n) = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$
Symmetrical Hard Limit	$f(n) = \begin{cases} -1 & n < 0 \\ +1 & n \geq 0 \end{cases}$
Linear	$f(n) = n$
Saturating Linear	$f(n) = \begin{cases} 0 & n < 0 \\ n & 0 \leq n \leq 1 \\ 1 & n > 1 \end{cases}$
Symmetric Saturating Linear	$f(n) = \begin{cases} -1 & n < -1 \\ n & -1 \leq n \leq 1 \\ +1 & n > 1 \end{cases}$
Log-Sigmoid	$f(n) = \frac{1}{1 + e^{-n}}$
Hyperbolic Tangent Sigmoid	$f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}$
Positive Linear	$f(n) = \begin{cases} 0 & n < 0 \\ n & n \geq 0 \end{cases}$
Competitive	$f(n) = \begin{cases} 1 & \text{neuron with max } x \\ 0 & \text{all other neurons} \end{cases}$

Following as a direct result of the above discussion, the operation performed by the artificial neuron as a whole is described by the equation:

$$a = f(\mathbf{w}\mathbf{p} + b) \quad (2-1)$$

The independent parameters of the model are therefore identified as \mathbf{p} , \mathbf{w} , b and $f(\cdot)$. The input vector is determined both in size and value by the problem the neuron,

alone or with other neurons in an arbitrary ANN structure, is a model for, or the position of that particular neuron in the whole structure, or both. The weight and bias variables are assigned random initial values and then adjusted by some *learning rule* so that the neuron input/output relationship meets some specific goal. More about learning algorithms are discussed in §2.7. Finally, the transfer function is typically chosen by the model designer to satisfy some specification of the problem that the neuron is trying to solve.

Table 2-1 summarises the transfer functions used in the vast majority of ANN structures encountered in the ANN literature. All of them apply to single neurons with the exception of the *competitive* transfer function that is meaningful when applied to a *layer* of two or more neurons. A few of those transfer functions have been introduced as candidate choices for the design of the artificial neuron because they happened to reflect the existing knowledge about the functionality of the biological neuron, a piece of knowledge that has evolved throughout the history of ANN research area with moderate success so far. Other functions were defined simply because the neurons designed upon them showed improved modelling efficiency when used as members of certain ANN architectures. Although the second approach could be branded as “un-biological”, it is perfectly acceptable for ANN model design purposes. After all, if we want to construct a useful device rather than model the brain, there is no reason whatsoever to be bounded by the way the brain works!

Three of the transfer functions listed in table 2-1 are more commonly used than the rest and therefore deserve some attention.

- a) The *hard limit* transfer function sets the output of the neuron to 0 if the function argument is less than 0, or 1 if its argument is greater than or equal to 0. It has been introduced in the 40's as a direct substitute to the binary behaviour of the biological neuron's activity. The function is used to create neurons that classify inputs into two distinct categories.
- b) The *log-sigmoid* transfer function takes the input (which may have any value between plus and minus infinity) and squashes the output into the range 0 to 1 in a continuous and constantly increasing manner. It has been introduced after the hard limit transfer function as a more accurate model of the biological neuron's activity. The log-sigmoid function is widely used in multi-layer networks that are trained using the backpropagation algorithm, in part because this func-

tion is differentiable (more about the backpropagation learning algorithm to follow).

- c) The *linear* transfer function assigns its output as equal to its input. Despite the absence of equivalent or similar function in the biological world, the linear function is very useful when the weighted sum of input signals to the associated neuron is needed as an output. Such cases frequently arise in neurons that belong to the output layer of a multi-layer ANN structure targeted to function approximation use. On the other hand, a neuron that incorporates a linear transfer function degenerates to a pure linear system. Therefore, the function cannot be used in every neuron of a feedforward ANN structure to avoid the build-up of a linear model unsuitable to give solutions for most, if not all, problems.

All kinds of transfer functions mentioned so far are characterised by their *deterministic* outputs. However, certain neurons may be designed to have *stochastic* transfer functions as in Boltzmann machines [76] and stochastic automata [77]. In the stochastic case the transfer function computes probabilities and the output of the neuron is a random value based on that probability. ANN architectures consisted of stochastic neurons are more difficult to study and train than the others, but they usually give more accurate modelling solutions as it emerges from the discussion of training algorithms in §2.7.

The choice of transfer functions for the individual neurons of a given ANN architecture together with the connectivity of those neurons are the two important aspects that determine the range of problems the ANN is able to model to a sufficient degree. Having described the functionality of the artificial neuron in sufficient detail, it is high time we moved our focus to the ANN structures and their connectivity patterns.

2.6 ANN architectures

Commonly one artificial neuron, even with many inputs, is not sufficient to model a particular problem. The early acknowledgement of this fact by the researchers of the field has forced them to design new models consisting of many neurons carefully chosen and connected together in topologies that exploit the benefits of parallel computation in the best possible manner, much alike the biological neural network. These new models are known as *Artificial Neural Networks (ANNs)* and the as-

sociated neuron selection and interconnection topology are described altogether by the term *ANN architectures*.

There are two fundamental methods by which all ANNs developed over the years may be classified. The first method divides those ANNs (and the corresponding architectures) into *feedforward*, *competitive* and *associative memory* ones, while the second method separates them into *time-independent* and *time-dependent* networks.

In feedforward networks the output is computed directly from the input in one pass, i.e., no feedback is involved. These networks are used for function approximation and classification purposes in applications spanning a variety of fields such as adaptive filtering, automatic control, and – as this work proves – applied optics. More about feedforward networks will follow after the brief presentation of the remaining network classes.

Competitive networks are solely designed as pattern recognition models. First, they compute some measure of distance between stored prototype patterns and the input pattern. Second, they perform a competition to determine which neuron represents the prototype pattern closest to the input. Moreover, competitive networks may be designed to be *adaptive*, i.e., adjust their prototype patterns as new inputs are applied to them. In effect, these adaptive networks *learn* to cluster their inputs into different categories. Examples of competitive network architectures are the Hamming network [78, 79], the counterpropagation network (CPN) [80, 81], the neocognitron [82, 83] and the ART network family [84-89].

The associative memory networks are characterised by their ability to recognise and recall temporal, as well as spatial, patterns (stored in their internal parameters) by association with input data rather by an address as in conventional memories. They may be divided further into *autoassociative* memories, which retrieve a complete data pattern given either a sub-pattern or a slight variation thereof, and *heteroassociative* memories, which store bidirectional associations between patterns of activation of their input and output units. Examples of associative memory networks are the Hopfield network [72, 90], the Li-Michel network [91, 92], the Boltzmann machine [76, 93], the Bidirectional Associative Memory (BAM) [94, 95] and the Brain-State-in-a-Box (BSB) memory model [96].

All time-independent networks share the property of output invariance with time assuming that their inputs remain constant for the same time period. On the other hand, time-dependent networks exhibit arbitrary transient but absolutely constant

steady-state *responses* when they are provided with constant *stimuli* and their time-critical internal variables are assigned to certain *initial condition* values.

ANN time-dependence is achieved either by the addition of *time-dependent functional blocks* in the network's neuron interconnections or by the inclusion of *feedback connections* that back-propagate internal signals to earlier (in processing order) functional units. Figure 2-3 presents two popular time-dependent functional blocks, the *delay block* for discrete-time ANN systems and the *integrator block* for continuous-time ANN systems. Their operation is described by the equations:

$$a(t) = u(t-1) \quad (2-2)$$

$$a(t) = \int_0^t u(\tau) d\tau + a(0) \quad (2-3)$$

for the time-delay and integrator functional blocks, respectively.

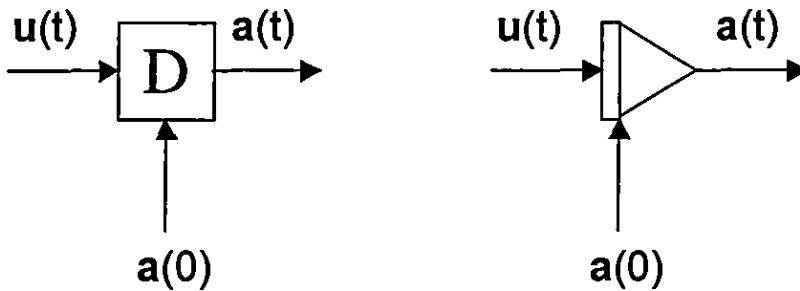


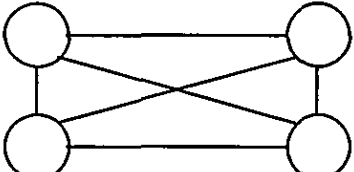
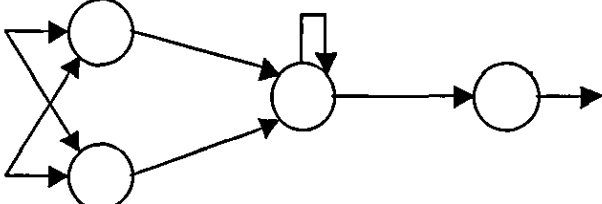
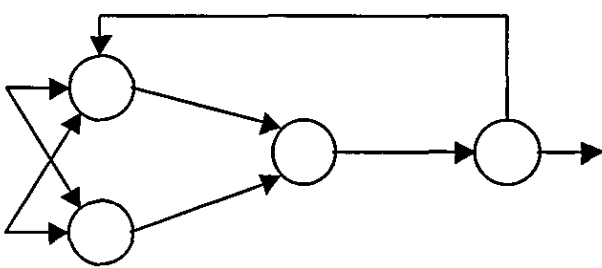
Figure 2-3 – Delay block (left) and integrator block (right)

Table 2-2 lists some typical feedback connectivity patterns used in the design of time-dependent ANN architectures together with their graphical illustrations. All functional units in the diagrams of that table are represented with circles, all unidirectional connections with directed edges, and all bidirectional connections with undirected edges.

It is apparent from the discussion above that the ANN architectural class that is suitable for function approximation purposes is the feedforward one. If the additional assumption that no time dependency exists between the inputs and outputs of the problem to be modelled is made, it becomes evident that there is absolutely no reason to include time-dependent functional blocks in the feedforward ANN model constructed for that particular problem. As the aforementioned assumption is true for all inverse scattering problems falling within the scope of this study, all time-dependent ANN architectures available in the literature such as the Time-Delay Neural Network (TDNN) [97, 98], the Finite Impulse Response (FIR) Multilayer Percep-

tron [99-101], the Pipelined Recurrent Neural Network (PPRN) [102], the Nonlinear Autoregressive Moving Average (NARMA) network [103], the Elman network [104] or the Real-Time Recurrent Network (RTRN) [105] are deemed unsuitable. After all these exclusions, the only ANN architectural class still available for the modelling of the problems falling within this work's interest is the feedforward time-independent class. To simplify the terminology used from this point on, all references to feedforward ANNs and their associative architectures will imply time independence unless explicitly mentioned otherwise.

Table 2-2 – Typical feedback connectivity patterns

Connectivity Pattern	Graphical Illustration
Bidirectional connections	
Self-connections	
Recurrent connections	

If the term *layer* is introduced to describe a subset of neurons that belong to the same ANN structure and operate in parallel, then the name *single-layer feedforward ANN* can be appointed to the network shown in figure 2-4. That network is the simplest feedforward one that can ever be designed. It accepts R inputs, includes S neurons and outputs the same number of variables. Its operation is fully described by the set of equations:

$$\mathbf{n} = \mathbf{Wp} + \mathbf{b} \quad (2-4)$$

$$\mathbf{a} = \mathbf{f}(\mathbf{n}) \quad (2-5)$$

where $1 \leq i \leq R$, $1 \leq j \leq S$ and:

- \mathbf{p} is the *input column vector* of elements $p(i) = p_i$,
- \mathbf{W} is the *weight matrix* of elements $W(j,i) = w_{j,i}$,
- \mathbf{b} is the *bias column vector* of elements $b(j) = b_j$,
- \mathbf{n} is the *net input vector* of elements $n(j) = n_j$,
- \mathbf{f} is the *transfer function column vector* of elements $f(j) = f_j(\cdot)$, and
- \mathbf{a} is the *output column vector* of elements $a(j) = a_j$.

If the notation $R-S_1-\dots-S_L$ is introduced to describe any feedforward ANN by the number of its inputs and layers' outputs in left-to-right sequence, the network of figure 2-4 is alternatively mentioned as R-S network for short.

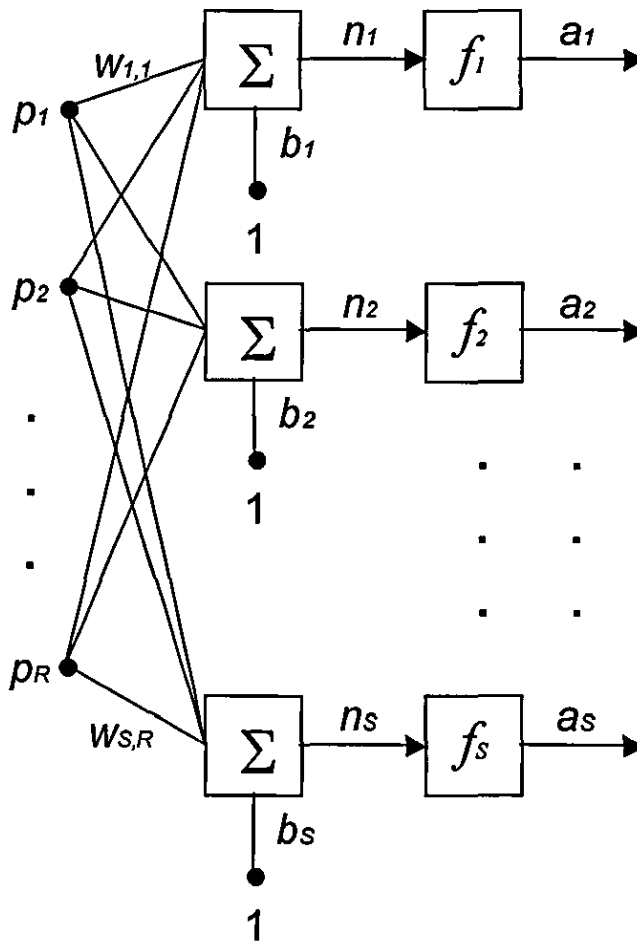


Figure 2-4 – Single-layer feedforward ANN

The selection of the transfer function vector \mathbf{f} for the R-S network may be arbitrary. However, two specific choices for that vector are considered to be significant for both historical and practical reasons. The first attempt ever made was to assign all transfer functions of the R-S network to the hard limit transfer function; the result is described by the term *perceptron* [67]. Two years later all hard limit transfer functions were replaced by linear functions and the new network was given the name ADALINE (Adaptive Linear Neuron) [68]. The two networks have been applied to similar problems: the perceptron was applied in pattern recognition and logical function simulation problems, while the ADALINE has been used in adaptive signal processing applications such as echo cancellation in long distance held phone calls. However, the ADALINE network was proved to have better performance than the perceptron due to the superiority of its learning algorithm. ADALINE's Least Mean Squared (LMS) learning algorithm tries to minimise the network's mean-squared error and therefore moves the decision boundaries as far from the training patterns as possible, while perceptron's simpler learning algorithm stops as soon as an acceptable decision boundary is found and therefore leaves the trained network potentially vulnerable to *noisy* inputs. Despite the differences in performance, both networks are capable of modelling only classification problems whose patterns are *linearly separable* [69]. In such problems there exists at least one hyperplane that divides the space of patterns into two subspaces, each including only patterns that correspond to outputs of either the same value (perceptron) or sign (ADALINE). Figure 2-5 demonstrates how a linear decision boundary is drawn and how the planar pattern space is subsequently divided in an arbitrary 2-1 network. In general, the exact position of the linear decision boundary in the pattern space is always dependent entirely upon the values of the connection weights and biases obtained by the application of a learning algorithm to the single-layer feedforward network. Such a decision boundary, however, is always orthogonal to the weight vector because (2-4) is always zero at the boundary.

A conceptually straightforward extension of the single-layer feedforward ANN involves two or more layers connected *in cascade*, i.e., the outputs of a certain layer are the inputs of the next (in predetermined sequence) layer until all available layers have been used. The new architecture is described by the term *Multilayer Feedforward Artificial Neural Network* (MFANN) and presented in figure 2-6. In general, an MFANN accepts R inputs, has L layers of S_l ($1 \leq l \leq L$) neurons each and returns S_L outputs. The set of input points to the structure are collectively described as the *input*

layer, the next $L-1$ layers of neurons are called *hidden layers*, and the final neuron layer is known by the name *output layer*. The operation of any MFANN is completely determined by the set of recursive equations:

$$\mathbf{a}_0 = \mathbf{p} \quad (2-6)$$

$$\mathbf{n}_l = \mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l \quad (2-7)$$

$$\mathbf{a}_l = \mathbf{f}_l(\mathbf{n}_l) \quad (2-8)$$

$$\mathbf{a} = \mathbf{a}_L \quad (2-9)$$

where $1 \leq i \leq S_{l-1}$, $1 \leq j \leq S_l$, $1 \leq l \leq L$ and:

- \mathbf{p} is the *input column vector* of elements $p(i) = p_i$,
- \mathbf{W}_l is the *weight matrix* of layer l with elements $W_l(j,i) = {}_l w_{j,i}$,
- \mathbf{b}_l is the *bias column vector* of layer l with elements $b_l(j) = {}_l b_j$,
- \mathbf{n}_l is the *net input vector* of layer l with elements $n_l(j) = {}_l n_j$,
- \mathbf{f}_l is the *transfer function column vector* of layer l : $f_l(j) = {}_l f_j(\cdot)$,
- \mathbf{a}_l is the *output column vector* of layer l with elements $a_l(j) = {}_l a_j$, and
- \mathbf{a} is the *output column vector* of elements $a(j) = {}_L a_j$.

Following the notational convention introduced as a short reference to the single-layer feedforward ANN, the MFANN described above will also be referred to as $R-S_1-\dots-S_L$ network for short.

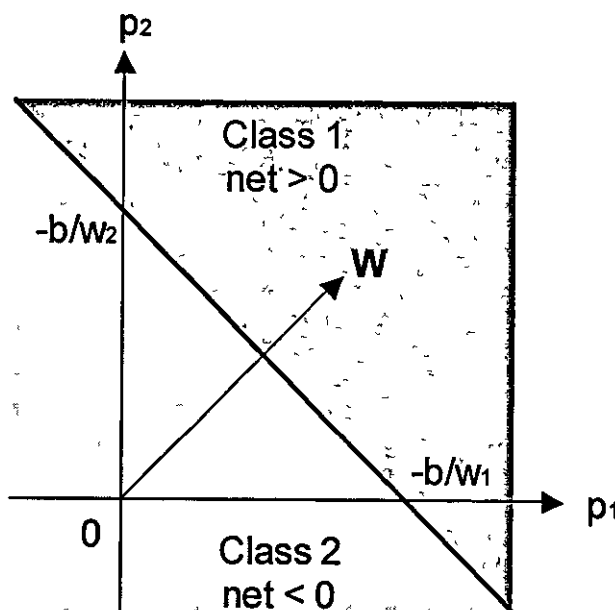


Figure 2-5 – Two-input linear decision boundary ($w_1 > 0$, $w_2 > 0$, $b < 0$)

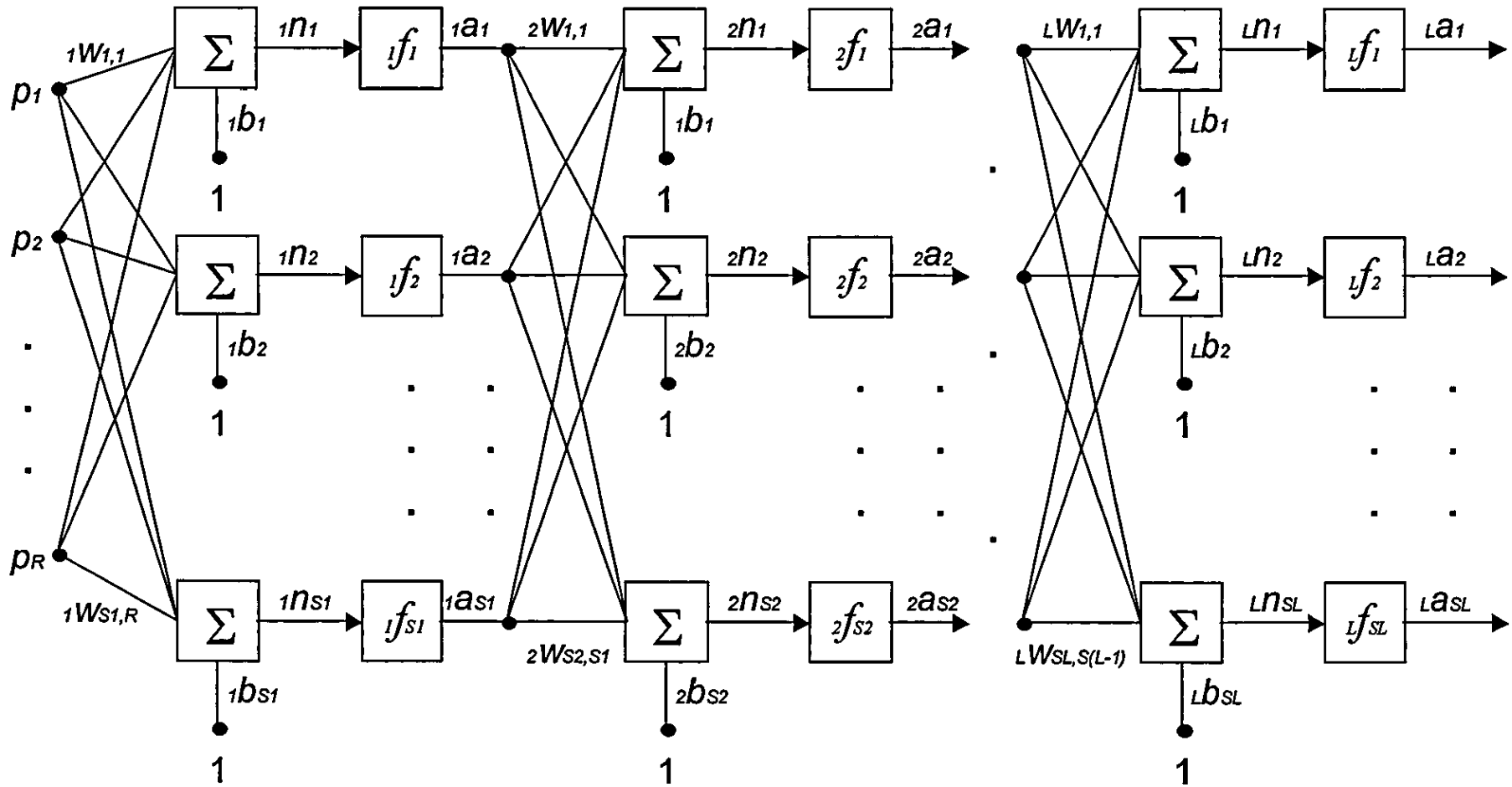


Figure 2-6 – Multilayer Feedforward Artificial Neural Network (MFANN)

Is the MNANN able to model function approximation and classification problems that do not have to satisfy strict conditions such as linear separation of patterns? Hornik *et al.* have proved that a sufficiently large MFANN with arbitrary transfer functions assigned to its neurons can approximate any Borel integrable, i.e. virtually any, function from one finite dimensional space to another finite dimensional space [106]. Moreover, Lippmann has shown that a sufficiently large three-layer MFANN can precisely define an arbitrary number of classes of arbitrary shape in the pattern space, and therefore classify all its input patterns exactly into these classes [79]. Therefore the answer to the aforementioned question is positive if the condition “sufficiently large” is met. Unfortunately, there is no precise numerical formula that ensures that for a given problem the MFANN size condition is satisfied and the network generalisation property is not sacrificed. The topic is an active area of research and as such it will be discussed in more detail in §2.9.

Even if the question of layer size is left aside for the time being, there still exist an overwhelming number of choices to be made in specifying a MFANN model suitable for a given problem. Essential guidelines that address the remaining MFANN design specification issues are the following:

- a) The number of inputs to the MFANN network and the number of outputs from it is defined by the specifications of the problem that is modelled.
- b) The number of layers necessary for the MFANN to model function classification or classification problems is known *a priori* for the vast majority of those problems. Kolmogorov’s *Mapping Neural Network Existence Theorem* proves that any *continuous* function $f : [0,1]^n \rightarrow \mathbb{R}^m$, can be modelled *exactly* by an $(2n+1)$ - m MFANN [107]. Although the theorem itself does not provide a design method for the transfer functions included in the neurons of the hidden and output layers of the “ideal” network, it offers a strong indication that a single hidden layer is sufficient for the approximation of continuous functions. However, continuity is a rather weak condition met by most functions encountered in practice. Therefore, it is reasonable to consider *single* hidden-layer MFANNs as suitable function approximation models unless all attempts to design such a model under certain circumstances lead to failure. In classification problem modelling a *two* hidden-layer MFANN is sufficient in all cases as already mentioned, while a *single* hidden-layer network is also sufficient in

problems where all class regions happen to be *convex open* or *convex closed* [79].

- c) The desired characteristics of the output signals assist the selection procedure for the transfer functions assigned to all output layer neurons. The linear transfer function is usually chosen in function approximation problems because it is simple, continuous and of infinite range. In contrast, the hard limit transfer function, or even better the log-sigmoid one, is used in classification applications because both functions saturate quickly enough to a binary set of values. The log-sigmoid function is more preferable than the hard limit one only because the first function is differentiable everywhere in its domain and therefore can be handled more easily by learning and optimisation algorithms developed for the MFANN architecture.
- d) The ideal choice of transfer functions included in the neurons of the hidden and output layers of a MFANN would be the one that matches more closely to the properties of the problem the network models. Such a choice, however, is very difficult to make in most cases. A widely followed compromise is the selection of the log-sigmoid transfer function for all neurons of concern. The preference of the log-sigmoid function is justified for biological and analytical reasons: it emerges from the study of the biological neuron operation, it is non-linear, continuous, and all its derivatives exist and are continuous. As a matter of fact, the log-sigmoid function will be preferred for the design of all MFANN models presented in this work unless there are strong indications in the nature of a problem that suggest an alternative choice to be made.
- e) All weights and biases appearing in a MFANN are assigned values as a result of the learning process governed by a suitable learning algorithm. More about the matter are discussed in §2.7.

Further evolution of the MFANN has produced more complex architectures, some of which are worth mentioning here.

The Radial Basis Function (RBF) networks are single hidden-layer MFANNs where all neurons of the hidden layer work as localised receptive fields instead of non-linear mappings of weighted sums. In other words, the neurons of the hidden layer return values determined by the distances between the network inputs and the "centre" of their basis functions: as an input moves away from a given centre, the associated neuron's output is reduced quickly to zero. Examples of variants of the RBF

network are the Cerebellar Model Articulation Controller (CMAC) network [108, 109] and the Modular Network [110, 111]. The RBF networks and their family train usually faster than the MFANNs but require many neurons for high-dimensional input spaces [112-115].

The Polynomial Networks are single-layer ANNs or MFANNs where the elements summing up weighted inputs in the networks' neurons have been substituted by other elements that calculate and return polynomial functions of those inputs. Examples of polynomial networks are the Functional Link Network [116], the Group Method of Data Handling (GMDH) network [117] and the Sigma-Pi network [118, 119]. Polynomial networks serve the same purpose as MFANNs as generic models to function approximation and classification problems at the cost of additional complexity that is not but sometimes necessary.

After the short presentation of the main ANN structures mentioned in scientific literature, it should have become apparent why the family of networks most suitable to model the problems presented in this study are the MFANNs. In titles, the MFANNs can perform function approximation and present simple architectures that promise ease and efficiency in learning and optimisation. The last two procedures will be presented in sufficient detail and mainly in conjunction with the MFANNs in §2.7 to §2.9.

2.7 ANN learning algorithms

A *learning rule* implemented by its corresponding *learning algorithm* (or *training algorithm*) is defined as a procedure for modifying all *non-predetermined, adjustable* variables of an ANN model, e.g. weights and biases, with the aim to *train* the model to perform some useful task, e.g., function approximation or data classification. Learning rules fall into three broad categories: *supervised learning*, *reinforcement (or graded) learning* and *unsupervised learning*.

A supervised learning rule is always provided with a set of examples of proper network behaviour in the form $\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_Q, t_Q\}$, where p_q is an input to the network and t_q is the corresponding output for $1 \leq q \leq Q$. This set known as the *training set* is obtained directly by the environment of the problem the ANN structure tries to model. The supervised learning rule applies a subset of the inputs p_q to the network and then adjusts a subset of the non-predetermined, adjustable parameters of the ANN in order to minimise the "distance" between the desired outputs t_q and the actual

actual outputs a_q . Examples of supervised learning rules are the ones developed for use in the perceptron, the ADALINE and MFANNs.

The reinforcement learning rules are similar to the supervised ones. The difference between the two is that the former kind of rules is provided with a *grade* (or *score*) as a measure of the ANN performance over some sequence of inputs instead of being given the correct outputs for the same input sequence as it happens in supervised learning. Reinforcement learning is more complex than either supervised or unsupervised learning because it is not clear what the correct output is for each input is, and as such is much less common than the other two learning rule categories. Examples of reinforcement learning rules are met in control system applications [77, 120].

An unsupervised learning rule modifies a subject of non-predetermined, adjustable parameters of a certain ANN in response to network inputs only. There are neither target outputs nor performance measures available. Although unsupervised learning rules might seem to be impractical at a first glance, they manage to adapt the network towards optimisation of an internal (built-in) criterion that must be specified ahead of time. Most of unsupervised learning rules perform some kind of clustering operation, i.e., categorisation of input patterns into a finite number of classes. This is especially useful in such applications such as vector quantisation [121, 122].

Most learning rules encountered in the ANN literature are *deterministic*, i.e., they always change all non-predetermined, adjustable ANN variables in the direction of improving the overall performance of the model. In contrast, *stochastic* learning rules have also been developed which allow the change of certain ANN variables in the "wrong" direction of reducing overall modelling performance, albeit with a small probability. There is a trade-off between the efficiency and ease of analysis in deterministic schemes and the possibility of achieving "globally maximum" modelling performance using stochastic schemes. Stochastic learning is usually applied as an alternative method when all deterministic learning approaches to a given ANN architecture provably fail [123].

As it has already been stated above, the MFANN is the most probable network architecture to successfully model the real problems presented in this study. Nevertheless, all common learning rules applied to MFANNs belong to the deterministic supervised class. This fact facilitates the presentation of these learning rules and their particular use in MFANN architectures in the remaining of this section.

The first parameter that has to be decided in the design of a learning rule that can be applied to MFANNs is the *performance index*, i.e., the quantitative measure of network performance which is small when the network performs well and large when the network performs poorly. A usual choice for the performance index of all learning rules of interest is the *mean-squared error* (MSE) between the actual and target output vectors associated with input vectors applied one at a time to the MFANN model. Following and extending the notation presented in the definition of supervised learning rules, the mean-squared error index is mathematically defined as:

$$F(\mathbf{e}) \equiv \frac{1}{Q} \cdot \left\| \left[\|\mathbf{e}_1\|_2^2 \quad \|\mathbf{e}_2\|_2^2 \quad \dots \quad \|\mathbf{e}_q\|_2^2 \quad \dots \quad \|\mathbf{e}_Q\|_2^2 \right]^T \right\|_1 \quad (2-10)$$

where Q is the number of MFANN training patterns, \mathbf{e} is the *error* or *distance* vector associated with the network:

$$\mathbf{e} \equiv [\mathbf{e}_1 \mid \mathbf{e}_2 \mid \dots \mid \mathbf{e}_q \mid \dots \mid \mathbf{e}_Q] \quad (2-11)$$

\mathbf{e}_q is an element of \mathbf{e} for $1 \leq q \leq Q$ defined as:

$$\mathbf{e}_q \equiv \mathbf{t}_q - \mathbf{a}_q \quad (2-12)$$

and $\|\cdot\|_1$, $\|\cdot\|_2$ are the first- and second-order norm operators defined for an arbitrary vector \mathbf{x} as follows:

$$\|\mathbf{x}\|_1 \equiv \sum_i |x_i| \quad (2-13)$$

$$\|\mathbf{x}\|_2 \equiv \sqrt{(\mathbf{x}^T \cdot \mathbf{x})} \quad (2-14)$$

The performance index defined above is dependent only on the actual output values \mathbf{a}_q obtained at every iteration of the learning algorithm when the set of examples of proper network behaviour $\{\mathbf{p}_1, \mathbf{t}_1\}$, $\{\mathbf{p}_2, \mathbf{t}_2\}$, ... $\{\mathbf{p}_Q, \mathbf{t}_Q\}$ is given. However, the actual outputs themselves are functions of all non-predetermined, adjustable variables of an MFANN as it emerges from (2-6) to (2-9). Therefore, the performance index is a complex non-linear vector function of the variables the learning rules try to find optimal values. In other words, the learning algorithms sought for MFANNs are indeed *optimisation algorithms* applicable to the performance index defined in (2-10).

The mean-squared error index has a lower bound of zero reached when all actual values coincide with the desired ones for the given set of examples of proper network behaviour. Furthermore, the same index is minimised when all respective error elements in (2-10) are minimised and hence the performance of the MFANN is maxi-

maximised. As a matter of fact, it looks straightforward to define the *objective* of any MFANN learning rule as the minimisation of the mean-squared error index. Nevertheless, the definition of a minimum in non-linear functions is *not* unique. Instead, three possible alternatives exist for such a minimum, which are precisely defined as follows:

- a) *Local (or strong) minimum.* The point \mathbf{x}^* is a local minimum of $F(\mathbf{x})$ if a scalar $\delta > 0$ exists such that $F(\mathbf{x}^*) < F(\mathbf{x}^* + \Delta\mathbf{x})$ for all $\Delta\mathbf{x}$ such that $\delta > \|\Delta\mathbf{x}\| > 0$. In other words, if there is a move away from the local minimum \mathbf{x}^* for a small distance in *any* direction, the function $F()$ will increase. However, it is possible for the same function to take values smaller than $F(\mathbf{x}^*)$ outside a small neighbourhood of \mathbf{x}^* .
- b) *Global minimum.* The point \mathbf{x}^* is a unique global minimum of $F(\mathbf{x})$ if $F(\mathbf{x}^*) < F(\mathbf{x}^* + \Delta\mathbf{x})$ for all $\Delta\mathbf{x} \neq 0$. For a global minimum \mathbf{x}^* the function $F()$ will be larger than the minimum point at every other point in its multidimensional domain.
- c) *Weak minimum.* The point \mathbf{x}^* is a weak minimum of $F(\mathbf{x})$ if it is not a local minimum, and a scalar $\delta > 0$ exists, such that $F(\mathbf{x}^*) \leq F(\mathbf{x}^* + \Delta\mathbf{x})$ for all $\Delta\mathbf{x}$ such that $\delta > \|\Delta\mathbf{x}\| > 0$. No matter which direction of movement is taken away from a weak minimum for a small distance, the function $F()$ cannot decrease, although there may be some directions in which the function does not change.

Ideally, a learning algorithm designed for the MFANN class should be able to optimise the mean-squared error function towards its global minimum if it exists. Unfortunately, such performance cannot be achieved under all circumstances. Practice has shown indeed that MFANN learning algorithms are likely to be trapped in a local minimum point of the performance index that results to an unsatisfactory performance value. A well-known workaround solution to this problem is to apply the learning algorithm many times to the same network structure, each time though with different initial conditions, and choose the solution that corresponds to the best performance obtained. Another approach is the addition of complexity to the learning algorithms that makes them capable of performing more accurate searches for optimal points at every step of theirs. Such an approach, however, comes at the cost of additional execution time due to the increase in either the number of steps required for a solution to be found, or the increase of arithmetic operations required for the execution of every

step of the algorithm, or both. The efficiency of MFANN learning algorithms is discussed further below as each candidate algorithm is presented in turn.

Three major optimisation (i.e., minimisation) algorithms have been discovered since the 17th century for the minimisation of non-linear vector functions. These are the *steepest descent algorithm*, the *Newton's algorithm* and the *conjugate gradient algorithm*. All optimisation algorithms are provided with an initial guess \mathbf{x}_0 for the global minimum point of the function $F()$ and then update the guess in stages according to an equation of the form:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + a_k \mathbf{d}_k \quad (2-15)$$

or:

$$\Delta \mathbf{x}_k \equiv \mathbf{x}_{k+1} - \mathbf{x}_k = a_k \mathbf{d}_k \quad (2-16)$$

where \mathbf{d}_k is the *search direction vector* and a_k the scalar, positive *learning rate* that determines the length of the algorithmic step. Different search direction vector choices distinctively characterise the basic algorithmic lines mentioned above, while different learning rate selection schemes simply add variety to these lines.

The steepest descent algorithm is based on the approximation of the vector function $F()$ at any point \mathbf{x}_k by the first two terms of the Taylor series expansion of $F()$ around \mathbf{x}_k to choose as search direction vector the negative of the *gradient* of $F()$ at \mathbf{x}_k :

$$\mathbf{d}_k = -\mathbf{g}_k \quad (2-17)$$

where the gradient is defined by:

$$\mathbf{g}_k \equiv \nabla F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k} \quad (2-18)$$

The learning rate for steepest descent is chosen to minimise $F()$ along the search line and is found in two steps. The first step is called *interval location* and returns some initial interval of a_k that contains a local minimum for $F(\mathbf{x}_k + a_k \mathbf{d}_k)$. After that, the *interval reduction* step reduces the size of the initial interval until the minimum a_k is located to the desired accuracy. The *function comparison method* described in [124] is suitable for interval location purposes, while the *Golden Section search* [124] can be used for interval reduction in every case.

Steepest descent has the advantage that is very simple, requiring calculation only of the gradient. It is also guaranteed to converge to a *stationary* (i.e. minimum) point if the learning rate is small enough. The main disadvantage of steepest descent is that it has only *linear termination*, i.e., it minimises a linear vector function exactly in

a finite number of steps but it does not necessarily do the same for quadratic or higher-order vector functions. As a matter of fact, steepest descent generally requires many more iterations to complete than the other two major optimisation algorithms discussed below.

Newton's optimisation algorithm is based on the approximation of the vector function $F(\mathbf{x})$ at any point \mathbf{x}_k by the second-order Taylor series expansion of $F(\mathbf{x})$ around \mathbf{x}_k to choose as search direction vector the expression:

$$\mathbf{d}_k = -\mathbf{A}_k^{-1} \mathbf{g}_k \quad (2-19)$$

where \mathbf{A}_k is the Hessian matrix evaluated at the point \mathbf{x}_k :

$$\mathbf{A}_k \equiv \nabla^2 F(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_k} \quad (2-20)$$

and assign the learning rate α_k to unity. This algorithm has the property of *quadratic termination*, i.e., it minimises a quadratic vector function in a finite number of steps (in fact, in a single step). Therefore, it is generally much faster than steepest descent. On the other hand, Newton's algorithm has two main disadvantages. Firstly, it requires calculation and storage of the Hessian matrix, as well as its inverse, at every step. Secondly, its convergence is hard to be guaranteed in any circumstance other than a quadratic vector function optimisation: the Hessian matrix may be single (i.e., non-invertible) at some stage, or the algorithm may converge to a saddle point (i.e., a point which can be regarded as either local minimum or local maximum depending on the direction by which it is approached), or the algorithm may not converge at all. A number of modification ideas have been proposed for the basic Newton's algorithm to alleviate some of its disadvantages, one of which is to be discussed in detail later in this section.

The conjugate gradient algorithm has taken its name from the fact that all search direction vectors \mathbf{d}_k assigned at every one out of n consecutive iterations of the algorithm, where n is the number of inputs to function $F(\cdot)$, are mutually conjugate with respect to a positive definite matrix \mathbf{A} . In mathematical terms:

$$\mathbf{d}_k^T \mathbf{A} \mathbf{d}_j = 0 \quad 1 \leq j \leq n, 1 \leq k \leq n, j \neq k \quad (2-21)$$

Conjugate search direction vectors are derived in practice by the recurrent formula:

$$\mathbf{d}_k = \begin{cases} -\mathbf{g}_k & k = c(n+1) \\ -\mathbf{g}_k + \beta_k \mathbf{d}_{k-1} & k = c(n+1)+1, c(n+1)+2, \dots, c(n+1)+n \end{cases} \quad (2-22)$$

where c is a nonnegative integer and β_k a scalar obtained by one of the equations:

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\Delta \mathbf{g}_{k-1}^T \mathbf{p}_{k-1}} \quad (2-23)$$

(Hestenes and Steifel method),

$$\beta_k = \frac{\mathbf{g}_k^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad (2-24)$$

(Fletcher and Reeves method), or

$$\beta_k = \frac{\Delta \mathbf{g}_{k-1}^T \mathbf{g}_k}{\mathbf{g}_{k-1}^T \mathbf{g}_{k-1}} \quad (2-25)$$

(Polak and Ribière method). In equations (2-23) to (2-25):

$$\Delta \mathbf{g}_k = \mathbf{g}_{k+1} - \mathbf{g}_k \quad (2-26)$$

The optimal learning rate α_k at every step is always found by following the same methods as the ones presented for the steepest descent algorithm case.

It can be shown that if a sequence of exact linear searches along any set of conjugate directions $\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n\}$ is made, then the exact minimum of any quadratic function $F(\cdot)$ with n inputs will be reached in at most n searches. In other words, the conjugate gradient algorithm features quadratic termination like Newton's algorithm. Despite the fact that Newton's algorithm is usually faster in convergence than the conjugate gradient in applied minimisation problems (but definitely much faster than steepest descent), the second algorithm does not require calculation and storage of the Hessian matrix nor its inverse. Therefore, conjugate gradient is well suited to minimisation problems with large numbers of parameters.

The next and fundamental step in the development of MFANN learning algorithms is the implementation of the generic optimisation algorithms presented above in the minimisation of carefully selected *variants* of the mean-squared error index (2-10). For the implementation of the steepest descent and conjugate gradient algorithms, one must calculate the gradients of the performance index with respect to every non-predetermined, adjustable MFANN parameter. For the implementation of a variant of Newton's algorithm, details of which are discussed below, a pseudo-Hessian invertible matrix has to be calculated first as a function of the gradients of all individual network output errors with respect to every non-predetermined, adjustable MFANN parameter. No matter which optimisation algorithm is actually implemented for MFANN training, the result is always defined as *backpropagation algorithm* due

to the fact that all necessary gradients are always processed from the last layer of the network to the first.

Given a set of examples of proper network behaviour $\{p_1, t_1\}, \{p_2, t_2\}, \dots \{p_Q, t_Q\}$, the first step of every MFANN learning rule presented in this work is to calculate the set of network outputs $\{a_1, a_2, \dots a_Q\}$ by repeated use of the equations (2-6) to (2-9). The next step is the update of either the *marginal-squared error* or the *sum-squared error* performance index defined in turn by the following equations:

$$\hat{F}_q(\mathbf{e}) \equiv \|\mathbf{e}_q\|_2^2 \quad (2-27)$$

$$\hat{F}(\mathbf{e}) \equiv \left\| \left[\|\mathbf{e}_1\|_2^2 \quad \|\mathbf{e}_2\|_2^2 \quad \dots \quad \|\mathbf{e}_q\|_2^2 \quad \dots \quad \|\mathbf{e}_Q\|_2^2 \right]^T \right\|_1 \quad (2-28)$$

The marginal-squared error (2-27) is an approximation of, while the sum-squared error (2-28) is directly proportional to, the mean-squared error (2-10). The marginal-squared error performance index chosen as performance index for the implementation of steepest descent and conjugate gradient in MFANN learning, while the sum-squared error performance index allows for straightforward implementation of the Levenberg-Marquardt algorithm, a variant of Newton's algorithm, in MFANN learning too.

For the calculation of all gradients of the marginal-squared error with respect to all adjustable, non-predetermined variables of a MFANN (i.e., weights and biases) it is necessary to introduce two new network variables. These variables are the *net input sensitivity* vector s_l of a network layer l ($1 \leq l \leq L$):

$$s_l \equiv \frac{\partial \hat{F}_q}{\partial \mathbf{n}_l} = \left[\frac{\partial \hat{F}_q}{\partial_1 n_1} \quad \frac{\partial \hat{F}_q}{\partial_1 n_2} \quad \dots \quad \frac{\partial \hat{F}_q}{\partial_1 n_{s_l}} \right]^T \quad (2-29)$$

and the diagonal matrix \hat{F}_l of all first-order derivatives of transfer functions assigned to neurons of layer l :

$$\hat{F}_l \equiv \text{diag} \left(\left[\frac{d_1 f_1(\cdot, n_1)}{d_1 n_1} \quad \frac{d_1 f_2(\cdot, n_2)}{d_1 n_2} \quad \dots \quad \frac{d_1 f_{s_l}(\cdot, n_{s_l})}{d_1 n_{s_l}} \right]^T \right) \quad (2-30)$$

By use of the chain rule it can be shown that the gradients of the marginal-squared error function with respect to all MFANN weights and biases are derived from the net input sensitivities as follows:

$$\mathbf{g}_k = \begin{cases} \mathbf{s}_l \mathbf{a}_{l-1}^T & \text{for weights of layer } l \\ \mathbf{s}_l & \text{for biases of layer } l \end{cases} \quad (2-31)$$

where k is the iteration index of the minimisation (or learning) algorithm.

Rumelhart *et al.* showed that all net input sensitivities of any MFANN could be calculated exactly by the following recursive formula:

$$\mathbf{s}_l = \begin{cases} -2\dot{\mathbf{F}}_l \mathbf{e}_q & l = L \\ \dot{\mathbf{F}}_l \mathbf{W}_{l+1}^T \mathbf{s}_{l+1} & l = L-1, L-2 \dots 1 \end{cases} \quad (2-32)$$

where $\mathbf{e}_q \equiv \mathbf{t}_q - \mathbf{a}_q$. This discovery was applied to the generic steepest descent minimisation algorithm in order for the first MFANN learning algorithm, the *Steepest Descent Backpropagation algorithm (SDBP)*, to be born [74]. It has to be mentioned though that the original SDBP algorithm does *not* modify its learning rate α during network training, thus featuring a small difference from the classical steepest descent algorithm. A few years later, Charalambous [125] applied the same principle to the generic conjugate gradient algorithm in order to construct the *Conjugate Gradient Backpropagation algorithm (CGBP)*. SDBP influenced the resurgence of interest in the neural network field during the 1980s, while CGBP provided significant speedup over SDBP on many MFANN learning occasions.

After the discovery of the SDBP learning algorithm a number of *heuristic techniques* were developed which tried to increase the algorithm's convergence speed. These techniques include such ideas as varying the learning rate, using momentum and rescaling variables. The main heuristic modification algorithms are:

- a) *Momentum Backpropagation (MOBP)*. The very last network parameter update is weighted by a factor γ and the resulting term known as *momentum* is then added to the current parameter update pre-multiplied with the constant $1-\gamma$. This modification allows for a larger learning rate α , i.e. greater convergence speed, to be used while the stability of the algorithm is maintained. Another feature of MOBP is that it tends to accelerate convergence even more when its trajectory drawn on the performance index surface is moving in a constant direction. More about MOBP can be found in [126].
- b) *Variable Learning Rate Backpropagation (VLBP)*. This algorithm tries to speed up SDBP convergence by suitably adjusting its learning rate α during the course of training. Specifically, if the performance index increases by more than some set percentage ζ after a network parameter update, then the update

is discarded, the learning rate is multiplied by some factor $0 < \rho < 1$, and, if a momentum term is included in the network parameter update formula of VLBP, its coefficient γ is set to zero. On the contrary, if the performance index decreases after a network parameter update, then the update is accepted and the learning rate α is multiplied by some factor $\eta > 1$ (if γ has been previously set to zero then it is reset to its original value). Finally, if the performance index increases by less than ζ , then the network performance update is accepted but the learning rate α (and the momentum coefficient γ) are left unchanged. More about VLBP are discussed in [126].

- c) *QuickProp*. This algorithm assumes that the performance index surface drawn in the network parameter space is parabolic and concave upward around the minimum point. Also, QuickProp assumes that the effect of each network parameter can be considered independently. More about the topic can be found in [127].

Other examples of successful heuristic SDBP modifications are the Delta-Bar-Delta VLBP [128], SuperSAB [129], and RPROP [130] algorithms.

Unfortunately, all heuristic modifications to SDBP suffer from two main problems. The first is that the modifications require that several (up to five or six) parameters be set (e.g., ρ , ζ , η and γ), while the only parameter required for SDBP is the learning rate α . Often the performance of the algorithm is sensitive to changes in these parameters, and even worse the choice of parameters is also problem dependent. The second drawback is that these SDBP modifications can sometimes fail to converge to problems for which the old plain SDBP can eventually find a solution. In conclusion, heuristic modifications can achieve good results for certain MFANN learning tasks but their performance on the average is rather unpredictable.

Another technique that has been applied to CGBP, SDBP and its heuristic modifications with considerable success is called *batching*. The batching technique requires all network parameter updates be calculated individually for every example of proper network behaviour $\{\mathbf{p}_q, \mathbf{t}_q\}$ available in the training set. Next, the average of all parameter updates calculated during an iteration of the learning algorithm is chosen as the eventual parameter update of that iteration. Batching produces more accurate parameter updates at the cost of additional complexity.

Perhaps the most powerful MFANN learning algorithm in terms of convergence speed and stability is the *Levenberg-Marquardt* (LM) one. LM is derived di-

rectly from another quasi-Newton optimisation algorithm known as *Gauss-Newton*, thus it is proper to present the latter algorithm first.

Gauss-Newton can be applied only to functions that are sums of squares of other non-linear functions, i.e. of the form:

$$F(\mathbf{x}) = \|\mathbf{v}(\mathbf{x})\|_2^2 \quad (2-33)$$

where $\mathbf{x} \equiv [x_1 \ x_2 \ \dots \ x_n]^T$ and $\mathbf{v}(\cdot) \equiv [v_1(\cdot) \ v_2(\cdot) \ \dots \ v_N(\cdot)]^T$. The gradient of $F(\cdot)$ in (2-33) is:

$$\nabla F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x}) \mathbf{v}(\mathbf{x}) \quad (2-34)$$

where

$$\mathbf{J}(\mathbf{x}) \equiv \begin{bmatrix} \frac{\partial v_1(\mathbf{x})}{\partial x_1} & \frac{\partial v_1(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_1(\mathbf{x})}{\partial x_n} \\ \frac{\partial v_2(\mathbf{x})}{\partial x_1} & \frac{\partial v_2(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_2(\mathbf{x})}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial v_N(\mathbf{x})}{\partial x_1} & \frac{\partial v_N(\mathbf{x})}{\partial x_2} & \dots & \frac{\partial v_N(\mathbf{x})}{\partial x_n} \end{bmatrix} \quad (2-35)$$

is the $N \times n$ *Jacobian matrix*. The Hessian of $F(\cdot)$ is given by:

$$\nabla^2 F(\mathbf{x}) = 2\mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) + 2\mathbf{R}(\mathbf{x}) \quad (2-36)$$

where

$$\mathbf{R}(\mathbf{x}) \equiv \sum_{i=1}^N v_i(\mathbf{x}) \nabla^2 v_i(\mathbf{x}) \quad (2-37)$$

is the remaining term. Gauss-Newton assumes that $\mathbf{R}(\mathbf{x})$ is a small term that can be removed from (2-36) without major consequences. The result is an approximation for the Hessian that depends only on the Jacobian, i.e., the first-order derivatives of $F(\cdot)$:

$$\nabla^2 F(\mathbf{x}) \cong 2\mathbf{J}^T(\mathbf{x}) \mathbf{J}(\mathbf{x}) \quad (2-38)$$

Finally, Gauss-Newton substitutes (2-34) and (2-38) into the search direction vector expression (2-19) of Newton's algorithm and sets its learning rate α_k to unity. Therefore (2-16) gives the following minimum guess update $\Delta \mathbf{x}_k$ for Gauss-Newton:

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k) \mathbf{J}(\mathbf{x}_k)]^{-1} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) \quad (2-39)$$

The main problem with the Gauss-Newton algorithm is that the Hessian approximation $\mathbf{H} \approx \mathbf{J}^T \mathbf{J}$ is not always invertible. This can be overcome by using the modification $\mathbf{G} \equiv \mathbf{H} + \mu_k \mathbf{I}$ to the approximate Hessian matrix. If scalar μ_k is sufficiently large,

it can be proved that matrix \mathbf{G} is positive definite and hence invertible. This leads to the following expression of the guess update $\Delta \mathbf{x}_k$ for LM:

$$\Delta \mathbf{x}_k = -[\mathbf{J}^T(\mathbf{x}_k)\mathbf{J}(\mathbf{x}_k) + \mu_k \mathbf{I}]^{-1} \mathbf{J}^T(\mathbf{x}_k) \mathbf{v}(\mathbf{x}_k) \quad (2-40)$$

The LM algorithm has the very useful feature that as its *momentum* μ_k is increased it approaches the steepest descent algorithm with small learning rate, while as μ_k is decreased to zero the algorithm becomes Gauss-Newton. Therefore LM provides a nice compromise between the speed of Newton's algorithm and the guaranteed convergence of steepest descent.

The generic LM algorithm presented above can be used for MFANN training purposes if the sum-squared error (2-28) is chosen as the performance index and the vectors \mathbf{x} and \mathbf{v} are defined as:

$$\mathbf{x} \equiv [x_1 \ \dots \ x_n]^T = [{}_1w_{1,1} \ \dots \ {}_1w_{S_1,S_0} \ {}_1b_1 \ \dots \ {}_1b_{S_1} \ {}_2w_{1,1} \ \dots \ {}_Lb_{S_L}]^T \quad (2-41)$$

$$\mathbf{v} \equiv [v_1 \ \dots \ v_N]^T = [e_{1,1} \ e_{2,1} \ \dots \ e_{S_L,1} \ e_{1,2} \ \dots \ e_{k,q} \ \dots \ e_{S_L,Q}]^T \quad (2-42)$$

where $S_0=R$, $n = \sum_{m=1}^L S_m (S_{m-1} + 1)$, $1 \leq k \leq S_L$, $1 \leq q \leq Q$, $e_{k,q}$ the k -th MFANN output error

when the vector \mathbf{p}_q is applied to the network as input, and $N=S_L Q$. Then, it can be shown that each element of the Jacobian matrix \mathbf{J} is computed by the expression:

$$J_{h,r} = \begin{cases} ({}_i\tilde{s}_{j,h}) \cdot ({}_{l-1}a_{l,q}) & \text{for weights of layer } l \\ {}_i\tilde{s}_{j,h} & \text{for biases of layer } l \end{cases} \quad (2-43)$$

where:

- i, j are neuron indices for layers S_{l-1} and S_l , i.e., $1 \leq i \leq S_{l-1}$ and $1 \leq j \leq S_l$,
- $h = (q-1) S_L + k$ is the row index of \mathbf{J} , $1 \leq h \leq N$,

$$r = \begin{cases} \sum_{m=1}^{l-1} [S_m (S_{m-1} + 1)] + (j-1) S_{l-1} + i & \text{for weights of layer } l \\ \sum_{m=1}^{l-1} [S_m (S_{m-1} + 1)] + S_l S_{l-1} + j & \text{for biases of layer } l \end{cases}$$

is the column index of \mathbf{J} , $1 \leq r \leq n$,

- ${}_{l-1}a_{l,q}$ is the output value of the i -th neuron belonging to layer $(l-1)$ when the MFANN is subject to input vector \mathbf{p}_q , and

- ${}_i\tilde{s}_{j,h} \equiv \frac{\partial v_h}{\partial {}_i n_{j,q}} = \frac{\partial e_{k,q}}{\partial {}_i n_{j,q}}$ is the *Marquardt sensitivity* of MFANN output error

$e_{k,q}$ with respect to net input ${}_i n_j$ when the network is subject to \mathbf{p}_q .

Hagan *et al.* [131] were the first to show how Marquardt sensitivities could be calculated directly from other MFANN variables of known values by application of the backpropagation principle. They considered ${}_l\tilde{s}_{j,h}$ as the (j,h)-th element of $S_{l \times N}$ Marquardt sensitivity matrix \tilde{S}_l created by augmenting the Q Marquardt sensitivity matrices $\tilde{S}_{l,q}$ of size $S_{l \times S_L}$ obtained when each input vector p_q is individually applied to the network:

$$\tilde{S}_l = [\tilde{S}_{l,1} \mid \tilde{S}_{l,2} \mid \dots \mid \tilde{S}_{l,q} \mid \dots \mid \tilde{S}_{l,Q}] \quad (2-44)$$

If the symbol $\dot{F}_{l,q}$ is used instead of \dot{F}_l to represent the matrix defined in (2-30), where the additional index q is a reminder that vector p_q has been applied to the network as input, then it is possible to summarise Hagan's findings for the calculation of Marquardt sensitivity matrices $\tilde{S}_{l,q}$ in the following recursive formula:

$$\tilde{S}_{l,q} = \begin{cases} -\dot{F}_{l,q} & l = L \\ \dot{F}_{l,q} W_{l+1}^T \tilde{S}_{l+1,q} & l = L-1, L-2 \dots 1 \end{cases} \quad (2-45)$$

The *Levenberg-Marquardt Backpropagation algorithm (LMBP)* begins with its momentum μ set to some small value. The first step in every iteration of the algorithm is to present all inputs p_q to the MFANN and compute the corresponding network outputs $a_{l,q}$ by applying (2-6) to (2-9), the network output errors $e_q = t_q - a_{L,q}$, and the sum-squared error performance index (2-28). The second step of the same iteration is the calculation of the elements of the Jacobian matrix J by (2-43) with the aid of (2-44) and (2-45). The third step requires the update of all adjustable-non-predetermined MFANN variables by substitution of (2-41), (2-42) and (2-43) into (2-40). The performance index (2-28) is recomputed in the fourth step using the updated network variable values and compared with the performance index value before the update took place. Steps three and four are repeated - with previous variable update rejected and μ multiplied by some factor $\theta > 1$ - as long as the comparison shows that the performance index has not been reduced due to the variable update. Eventually the performance index should decrease, since a smaller network variable update is taken in the direction of steepest descent by continually increasing μ . As soon as the performance index is reduced due to the last network variable update, the update is accepted, momentum μ is divided by the same factor θ so that LMBP approaches Gauss-Newton that should provide faster convergence, and the algorithm proceeds to

its next iteration. LMBP continues until the SSE performance index reaches a predetermined *error goal* SSE_{goal} , the SSE gradient becomes smaller than a limit ∇_{min} , the momentum μ becomes too larger than a constant μ_{max} , or the predefined maximum number of iterations Ep_{max} has been reached.

The key drawback of the LMBP algorithm is its storage requirement. The algorithm must store the $n \times n$ Hessian matrix G , where $n = \sum_{m=1}^L S_m (S_{m-1} + 1)$ is the number of all adjustable, non-predetermined MFANN variables subject to learning. In contrast, every other MFANN backpropagation algorithm discussed in this section requires only the storage of the gradient that is an n -dimensional vector. As a matter of fact, LMBP becomes impractical to use when n is very large and preference to the other learning algorithms is necessarily given.

A final, worth-mentioning issue that concerns all MFANN learning algorithms presented in this section (SDBP and variants, CGBP and LMBP) is the initialisation of all network parameters affected by training, i.e., weights and biases. First, as every MFANN is horizontally symmetrical by definition, zero is found to be a saddle point of the performance surface drawn in the adjustable, non-predetermined MFANN parameter space, and as such it troubles the learning algorithms. Hence the network weights and biases should *not* be initially set to zero altogether in order to avoid premature problems in MFANN learning process. Second, the aforementioned performance index surface tends to have very flat regions (where optimisation progress is painfully slow) away from its optimum points. Given that input data are usually pre-processed to small values (see §2.8), it would be unwise to initialise the network parameters affected by training to large values. In view of the two potential troubles mentioned above, initial weights and biases are usually chosen to be *small random values*. Nevertheless, it is far from certain that a single set of values selected for weight and bias initialisation would always guarantee learning algorithm convergence to a global minimum, or at least a good local minimum, of the performance index. Therefore a common practice is the choice of several different initial guesses for network parameter initialisation, separate network training with each individual guess, and comparison of results obtained by every training session.

The optical scattering problems dealt by this study are characterised mostly by intrinsic rather than input complexity. Therefore, the most suitable learning algorithm for all MFANNs models of these problems is one that guarantees fast and efficient

convergence without great concern about storage requirements, i.e., the LMBP algorithm. No matter which algorithm is chosen for a particular MFANN learning problem though, its name and parameters will be explicitly mentioned wherever necessary for clarity purposes.

2.8 ANN input pre-processing

Albeit being ignored in the discussion of many ANN modelling problems encountered in scientific literature, ANN input pre-processing is a vital process that originates from biological neural networks and assists both ANN learning and ANN optimisation procedures. The aim of input pre-processing is to reduce the range of input data to the network into intervals that couple with the intervals of weight and bias values of the network. These intervals ought to be small around zero as explained in §2.7 and §2.9.

Two main methods have been developed and followed in ANN literature for input pre-processing: *uniform normalisation* and *Gaussian normalisation*. Uniform normalisation finds the minimum $\min_q({}_q P_i)$ and maximum $\max_q({}_q P_i)$ of each element ${}_q P_i$ ($1 \leq i \leq R$) belonging to a vector \mathbf{p}_q taken from the ANN input data set $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q, \dots, \mathbf{p}_Q\}$ that is available. After that, the method forces all inputs to lie within the range $[0, 1]$ by use of the formula:

$${}_q \tilde{P}_i \equiv \frac{{}_q P_i - \min P_i}{\max P_i - \min P_i} \quad (2-46)$$

where $1 \leq i \leq R$, $1 \leq q \leq Q$, $\mathbf{p}_{\min}(i) \equiv \min P_i$, $\mathbf{p}_{\max}(i) \equiv \max P_i$,

$$\mathbf{p}_{\min} \equiv \left[\min_q({}_q P_1) \quad \min_q({}_q P_2) \quad \dots \quad \min_q({}_q P_i) \quad \dots \quad \min_q({}_q P_R) \right]^T \quad (2-47)$$

$$\mathbf{p}_{\max} \equiv \left[\max_q({}_q P_1) \quad \max_q({}_q P_2) \quad \dots \quad \max_q({}_q P_i) \quad \dots \quad \max_q({}_q P_R) \right]^T \quad (2-48)$$

and $\mathbf{p}_{\max} \neq \mathbf{p}_{\min}$. Finally, the uniformly normalised vector:

$$\tilde{\mathbf{p}}_q \equiv \left[{}_q \tilde{P}_1 \quad {}_q \tilde{P}_2 \quad \dots \quad {}_q \tilde{P}_i \quad \dots \quad {}_q \tilde{P}_R \right]^T \quad (2-49)$$

is derived from the initial input vector by this pre-processing method.

A variant of (2-46) restricts of all inputs within $[-0.5, 0.5]$ instead of $[0, 1]$:

$${}_q \tilde{P}_i \equiv \frac{{}_q P_i - 0.5(\max P_i + \min P_i)}{\max P_i - \min P_i} \quad (2-50)$$

Albeit insignificant at first glance, the variant plays a critical role in the success of the novel network optimisation algorithm explained in §2.9. Therefore all references to uniform normalisation made in this work will implicitly assume that (2-50) instead of (2-46) applies.

The Gaussian normalisation pre-processing method calculates the mean value

$$\mu_i = E({}_q p_i) \text{ and standard deviation } \sigma_i = \sqrt{E\left[({}_q p_i - \mu_i)^2\right]} \text{ of each element } {}_q p_i$$

($1 \leq i \leq R$) that belongs to a vector \mathbf{p}_q taken from the ANN input data set $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q, \dots, \mathbf{p}_Q\}$. After that, the method encloses about 95% of input data within a hyper-sphere of radius 2 by use of the formula:

$${}_q \tilde{p}_i \equiv \frac{{}_q p_i - E({}_q p_i)}{\sqrt{E\left[({}_q p_i - \mu_i)^2\right]}} \quad (2-51)$$

where $1 \leq i \leq R$, $1 \leq q \leq Q$ and $E(\mathbf{x}) \equiv \frac{1}{N} \sum_{i=1}^N x_i$, the *expected value function*. Finally, the

Gaussian normalised vector $\tilde{\mathbf{p}}_q$ is derived by substitution of (2-51) to (2-49).

Each data pre-processing method has its merits. Uniform normalisation fixes the range of values data can take in absolute limits, while Gaussian normalisation ensures that no outliers, i.e., vectors that fall significantly outside the clusters formed by groups of other vectors in the vector space, are left as a result of pre-processing. Hence uniform normalisation is usually applied in function approximation problems and Gaussian normalisation in classification problems respectively. Input data pre-processing by uniform normalisation is applied to every ANN modelling problem presented in this work unless otherwise stated.

Apart from input pre-processing, uniform normalisation is also used for *target data pre-processing* when the transfer functions assigned to neurons of the output layer in an ANN model have bounded ranges. This situation is common with network models constructed to solve classification problems, as log-sigmoid is typically selected as the transfer function for neurons belonging to their output layers. However, target data pre-processing can be mandatory for function approximation modelling as well, depending on the function's range characteristics. In brief, uniform normalisation of target vectors is conceptually and mathematically similar to the one described for input data pre-processing. The only difference between the two implementations

of the same method is that a linear transformation of (2-49) may be used instead of (2-46) so that the dynamic range of all normalised target vectors returned by the first implementation matches the range of every corresponding transfer function assigned to a neuron of network's output layer. Target data pre-processing will apply, and its use will be explicitly mentioned, whenever deemed necessary for any ANN modelling problem that is presented in this study.

2.9 ANN optimisation algorithms

2.9.1 Overview

As already mentioned in §2.1, optimisation is the procedure of transforming a generic mathematical model into another that exhibits efficiency in terms of accuracy and complexity defined in the context of computation and implementation. Specifically in the ANN field, optimisation techniques are applied during or after network training to shape or modify any aspect of network topology. As a matter of fact, optimisation may either interfere with or complement existing learning rules. Details of the major optimisation algorithms applied to feedforward ANNs and specifically MFANNs are provided in this section. The main principles of these algorithms can be applied to optimisation of other ANN classes, such as associative networks [132] or tree structures [133], but details of their application fall outside the scope of this study and therefore will not be discussed further.

The aim of MFANN optimisation is to help the network learn only the absolutely essential features of input-output training examples and generalise them in order to be able to give sufficiently accurate outputs to input data not present at any MFANN modelling phase. How this can be achieved though? If the network is too large, it has more internal parameters (known as "degrees of freedom") than the necessary ones and it uses them to learn the unimportant, possibly spurious features of the training data together with the important ones. The result of this *overfitting* behaviour is a MFANN that memorises the training examples quite well and fast but fails miserably when presented with similar but slightly different inputs. In contrast, a too small network finds it difficult to learn its training examples to an acceptable degree, as it may have less than necessary degrees of freedom or be highly sensitive to initial conditions and learning parameters. Even if learning eventually succeeds in small networks, it may well be due to the learning algorithm being trapped in a less than sat-

isfactory local minimum of the performance index. Since the brute-force method of designing and training networks of arbitrarily different size by a large number of initial conditions is way too slow and thus impractical, automated solutions had to be devised instead to tackle the MFANN optimisation problem in order to achieve maximum accuracy and minimum computational overhead.

MFANN optimisation algorithms are broadly classified into *constructional algorithms* and *pruning algorithms*. Constructional optimisation builds a network model from scratch by starting from a very primitive structure and progressively adding more complexity in a controlled and automated manner until the network has learnt the actual mapping to a sufficient degree of proximity [134-136]. On the other hand, pruning optimisation starts from a larger than necessary MFNN and then tries to simplify that network by measuring the relative importance of each of its parameters and progressively removing the unnecessary ones. Pruning continues until no more parameters can be removed without irreparable harm to model's accuracy [137]. Pruning algorithms are encountered more frequently in ANN literature for both practical and psychological reasons. Network pruning is comparatively simpler than network construction, and pruning produces poor but constantly improving modelling solutions in every step while construction provides no solution until the very end.

MFANN pruning is a process mainly achieved by either *sensitivity analysis* or *penalty-term* algorithms. Sensitivity analysis measures the contribution, or *sensitivity*, of every removable network parameter to network output or performance index *after* training has taken place, and the parameters given small sensitivity values are trimmed from the network. Alternatively, penalty-term algorithms add extra terms to the performance index that enhance and diminish the more and less important network parameters respectively *during* training. Sensitivity analysis requires more computational resources for sensitivity calculations but offers no interference with the learning algorithm in return. Therefore sensitivity analysis does not increase the risk of learning failures unlike penalty-term algorithms [137-139]. Other pruning algorithms, like interactive pruning [140], local or distributed bottlenecks [141, 142] or pruning by genetic algorithms [143], find only limited application due to their lack of automation or increased complexity.

A typical MFANN sensitivity analysis algorithm operates in three basic steps. Firstly, the algorithm assigns a *sensitivity* parameter to every MFANN parameter that can be considered for removal. The sensitivity is usually defined as the performance

index or network output change that can result from either a network parameter's perturbation around its initial value or the parameter's setting to a fixed value (usually zero). Secondly, sensitivity analysis calculates exactly or even approximately the sensitivity of each removable network parameter based upon a subset of all internal and external variables that can be defined for the MFANN. Finally, the algorithm calculates the *significance* of every removable network variable as a function of MFANN sensitivity values calculated in the previous step. The variables assigned the least significance values are removed from the network and the remaining MFANN variables, removable or not, are updated, directly or by network retraining, to minimise the effect of pruning to network's accuracy [137, 144].

MFANN sensitivity algorithms are divided into *weight elimination* and *node elimination* ones. The first class concentrates its efforts on the removal of network weights and biases, while the second class is concerned with the removal of inputs or hidden neurons from the network. However, it is impossible to separate the tasks of weight and node elimination completely. If a node is pruned, all weights and biases attached to it are automatically removed. Alternatively, the removal of all input or output weights to a node effectively eliminates the node itself. Therefore it is essential to apply sensitivity algorithms of both classes, as well as detect possible variable cross-eliminations as early as possible, in order to obtain the best network optimisation results possible.

The main weight elimination (thus weight optimisation) algorithms available for MFANNs are the *magnitude-based pruning* (MBP) [145], *optimal brain damage* (OBD) [146] and *optimal brain surgeon* (OBS) [147]. MBP detects and eliminates weights that have the smallest magnitude. However, this simple and naively plausible idea often leads to the elimination of the wrong weights, because small weights can be necessary to maintain network's accuracy. In contrast, OBD and OBS use the criterion of minimal performance index increase for weight elimination. That criterion requires the calculation of index's Hessian with respect to every weight to give accurate results, however. OBD calculates only the diagonal elements of the Hessian to reduce computational and storage requirements, while OBS uses more resources to obtain a good approximation of the inverse Hessian by recursion. Other weight elimination algorithms, such as [148], rely on rather heuristic approximations of performance index change occurred after forcing an arbitrary set of weights to zero. In any case, OBS

remains the most accurate and hence suitable weight elimination algorithm for small- and middle-sized MFANNs.

The early days of node elimination saw efforts to estimate performance index changes arisen from possible pruning of sets of inputs or hidden neurons, such as the attentional strength algorithm [138]. However, it was discovered later that network output change was a more indicative measure of node significance than performance index change because the former does not include averaging and thus it does not smooth out important node features [144]. In fact, implementations of output-change-based sensitivity algorithm in C++ can be found as early as in 1991 (see reference [149]). The establishment of network output change as node significance measure followed the invention of the *first-order sensitivity analysis (FOSA)* algorithm a few years later by Zurada *et al.* [150, 151].

Firstly, FOSA calculates all sensitivity parameters, defined as first-order derivatives of each MFANN output with respect to every node of the network, by recursion. After that, the algorithm calculates the node significance from the sensitivity parameters calculated in the first step, and sorts the nodes in descending significance order. Finally, FOSA eliminates simultaneously all nodes *clustered* at the bottom of the significance table and retrains the network to compensate for the node removals. The algorithm was initially tested for input pruning only in a variety of typical MFANN modelling applications, such as decision support systems [152], symbolic rule extraction [153] and machine learning [154], with considerable success. Later, FOSA was extended to perform hidden node elimination as well [155].

FOSA has two main advantages. Firstly, the algorithm calculates first-order sensitivities only and hence requires rather moderate storage resources. Secondly, FOSA has a single effective pass, i.e., it detects and removes many redundant MFANN nodes after its first application to the network but only few nodes, if any, after subsequent applications [151]. Therefore, the algorithm terminates relatively quickly in comparison with other node elimination algorithms. FOSA has an important disadvantage, however. Specifically, the simplistic assumption that every node contributes to network's input-to-output mapping independently from the others, an assumption upon which FOSA is based to prune any set of nodes altogether, does not hold in general. Instead, every network output is dependent on both single instances and sets of nodes. It is perfectly possible to encounter situations in which a set of nodes is considered insignificant when each node is examined separately, while the

same set is indeed necessary for the representation of basic features of the modelled problem if considered as a whole. Or, certain nodes may provide their successors in the network structure with contrary data, and hence are redundant when considered collectively, while each of these nodes returns considerable significance measurements when examined independently. Although the previous two cases may be extreme, it is still likely to encounter partial cross-correlations of node significances in any network that are impossible to be detected and dealt with successfully by first-order sensitivity measurements only [137]. Even worse, the simultaneous elimination of many network nodes dictated by FOSA involves the risk of driving a few of the remaining nodes into their non-linear or totally saturated regions in the network re-training phase of the algorithm. Should this happen, the affected nodes exhibit fatally distorted responses to the remaining inputs, and consequently the network's modelling accuracy deteriorates quickly [144].

This study introduces the novel *second-order sensitivity analysis (SOSA)* algorithm as an enhancement to FOSA. SOSA avoids the limitations and inefficiencies of FOSA mentioned above by approximating the changes in every MFANN output caused by the elimination of any network node, pair of nodes or set-of-three nodes. The accuracy of output change approximations calculated by SOSA is comparable to the one obtained by second-order Taylor approximations to analytic functions, thus quite satisfactory in most cases. Because SOSA calculates the significance of each set of one, two or three network nodes directly from MFANN output change approximations before it decides which and how many nodes to prune altogether, the new algorithm manages to detect and tackle cross-correlations of node significances much better than FOSA. Furthermore, SOSA is capable of undoing earlier node pruning decisions and reducing its elimination speed should the resulting simplified network be found unable to recover from the last node-pruning step, a very important feature that is non-existent in FOSA. Despite its obvious additional computational and storage requirements compared to FOSA, the SOSA algorithm demonstrated excellent node elimination results within reasonable time intervals when applied to small- and middle-sized MFANNs.

Both OBS and SOSA algorithms base their pruning decisions on performance index or network output change approximations that involve both the gradient and Hessian matrices. Therefore a properly combined action of these algorithms to any MFANN model considered in this study can provide excellent network optimisation

results. §2.9.2 and §2.9.3 discuss OBS and SOSA, while §2.9.4 demonstrates a novel, fully automated MFANN optimisation scheme involving both pruning algorithms.

2.9.2 Second-Order Sensitivity Analysis Algorithm

To explain the SOSA algorithm it is essential to define and calculate the first- and second-order sensitivities of network outputs with respect to each removable node. In the following discussion, the term *node* will be used to collectively describe MFANN inputs, hidden signal processing units, or outputs.

First of all, the first-order derivative ${}_i \dot{f}_i(\cdot)$ and second-order derivative ${}_i \ddot{f}_i(\cdot)$ of the transfer function ${}_i f_i(\cdot)$ assigned to the *i*-th neuron of the *l*-th layer and assumed to be twice differentiable are defined as follows:

$${}_i \dot{f}_i({}_i n_i) \equiv \frac{d({}_i a_i)}{d({}_i n_i)} \quad (2-52)$$

$${}_i \ddot{f}_i({}_i n_i) \equiv \frac{d^2({}_i a_i)}{d({}_i n_i)^2} \quad (2-53)$$

where $1 \leq l \leq L$ and $1 \leq i \leq S_l$. It is implied that ${}_i \dot{f}_i({}_i n_i)$ and ${}_i \ddot{f}_i({}_i n_i)$ are dependent on the input vector \mathbf{p}_q to the network for $1 \leq q \leq Q$, where Q is the number of MFANN training patterns. These derivatives are given by simple non-linear expressions of node outputs ${}_i a_i$, when the transfer function is linear, log-sigmoid or hyperbolic tangent sigmoid, as

table 2-3 shows. As a matter of fact, if storage space of $Q \cdot \sum_{l=1}^L S_l$ floating-point num-

bers is provided at the end of the MFANN training procedure, the derivatives are cal-

culated from node outputs at a typical cost of $O\left(Q \cdot \sum_{l=1}^L S_l\right)$ floating-point operations.

The first-order sensitivity ${}_{lm} \hat{S}_{ki}$ and second-order sensitivity ${}_{lm} \hat{H}_{ij}^k$ of the *k*-th node output belonging to layer *m*, ${}_m a_k$, with respect to *i*-th node output ${}_i a_i$ (and *j*-th node output ${}_i a_j$ for second-order sensitivity) of layer *l*, are defined as:

$${}_{lm} \hat{S}_{ki} \equiv \frac{\partial({}_m a_k)}{\partial({}_i a_i)} \quad (2-54)$$

$${}_{lm} \hat{H}_{ij}^k \equiv \frac{\partial^2({}_m a_k)}{\partial({}_i a_i) \partial({}_i a_j)} \quad (2-55)$$

Table 2-3 – Derivatives of common transfer functions

Name	First-Order Derivative	Second-Order Derivative
Linear	${}_i \dot{f}_i({}_i n_i) = 1$	${}_i \ddot{f}_i({}_i n_i) = 0$
Log-Sigmoid	${}_i \dot{f}_i({}_i n_i) = {}_i a_i \cdot (1 - {}_i a_i)$	${}_i \ddot{f}_i({}_i n_i) = {}_i a_i \cdot (1 - {}_i a_i) \cdot (1 - 2{}_i a_i)$
Hyperbolic Tangent Sigmoid	${}_i \dot{f}_i({}_i n_i) = (1 - {}_i a_i) \cdot (1 + {}_i a_i)$	${}_i \ddot{f}_i({}_i n_i) = (-2{}_i a_i) \cdot (1 - {}_i a_i) \cdot (1 + {}_i a_i)$

where $0 \leq l < m \leq L$, $1 \leq i \leq S_l$, $1 \leq j \leq S_l$, $i \neq j$ and $1 \leq k \leq S_m$. It is implied that ${}_{lm} \hat{S}_{ki}$ and ${}_{lm} \hat{H}_{ij}^k$ are dependent on the input vector \mathbf{p}_q to the network for $1 \leq q \leq Q$. If $l=0$ and $m=L$, (2-54) resembles the first-order sensitivity definition given in [150-154]. When $0 < l < L$ and $m=L$, (2-54) takes a form similar to the first-order sensitivity definition found in [155]. For $l=0$, $m=L$ and log-sigmoid transfer functions assigned to every MFANN neuron, the second-order sensitivity definition (2-55) is presented analytically in [156]. Contrary to limited definitions of first- and second-order sensitivities encountered in the seven references mentioned above, (2-54) and (2-55) are of universal value and therefore can be applied directly to any MFANN of twice-differentiable transfer functions assigned to each of its neurons.

When MFANN signal propagation rules (2-6) to (2-9) and transfer function derivative definitions (2-52) to (2-53) are applied to sensitivity definitions (2-54) and (2-55), it finally emerges that all first- and second-order sensitivities of the network are computed by the following recursive formulas:

$${}_{(u-1)m} \hat{S}_{ki} = \begin{cases} \begin{cases} 1 & i = k \\ 0 & i \neq k \end{cases} & u = m + 1 \\ \sum_{r=1}^{S_u} {}_u \dot{f}_r({}_u n_r) \cdot {}_u w_{ri} \cdot {}_{um} \hat{S}_{kr} & u = m, m-1, \dots, l+1 \end{cases} \quad (2-56)$$

$${}_{(u-1)m} \hat{H}_{ij}^k = \begin{cases} 0 & u = m + 1 \\ \sum_{r=1}^{S_u} {}_u \ddot{f}_r({}_u n_r) \cdot {}_u w_{ri} \cdot {}_u w_{rj} \cdot {}_{um} \hat{S}_{kr} + \\ \sum_{\eta_1=1}^{S_u} \sum_{\eta_2=1}^{S_u} {}_u \dot{f}_{\eta_1}({}_u n_{\eta_1}) \cdot {}_u \dot{f}_{\eta_2}({}_{(u+1)} n_{\eta_2}) \cdot {}_{um} \hat{H}_{\eta_1 \eta_2}^k & u = m, m-1, \dots, l+1 \end{cases} \quad (2-57)$$

where $S_0=R$ (if necessary). As a matter of fact, the first- and second-order sensitivities of a MFANN are calculated by backpropagation like the network's performance index sensitivities (see (2-32) and (2-45)). Both network output sensitivities are dependent on network weight and transfer function derivative values, while second-order sensitivity calculation requires previous knowledge of its first-order counterpart in addition.

Scalar sensitivity formulas (2-56) and (2-57) can be tabulated for notational and implementation ease into the following recursive matrix expressions:

$$\hat{S}_{(u-1)m} = \begin{cases} \mathbf{I} & u = m+1 \\ \hat{S}_{um} \cdot \dot{F}_u \cdot W_u & u = m, m-1, \dots, l+1 \end{cases} \quad (2-58)$$

$$\hat{H}_{(u-1)m}^k = \begin{cases} \mathbf{0} & u = m+1 \\ W_u^T \cdot \text{diag}(\hat{S}_{um}^k) \cdot \ddot{F}_u \cdot W_u + \\ (\dot{F}_u \cdot W_u)^T \cdot \hat{H}_{um}^k \cdot (\dot{F}_u \cdot W_u) & u = m, m-1, \dots, l+1 \end{cases} \quad (2-59)$$

where:

$$\hat{S}_{um}(k, l) \equiv {}_{um}\hat{S}_{kl} \quad (2-60)$$

$$\hat{S}_{um}^k(i) \equiv {}_{um}\hat{S}_{ki}^k \quad (2-61)$$

$$\hat{H}_{um}^k(i, j) \equiv {}_{um}\hat{H}_{ij}^k \quad (2-62)$$

$$F_u \equiv \text{diag}({}_u\dot{f}_1({}_u n_1) \quad {}_u\dot{f}_2({}_u n_2) \quad \dots \quad {}_u\dot{f}_{S_u}({}_u n_{S_u})) \quad (2-63)$$

$$\ddot{F}_u \equiv \text{diag}({}_u\ddot{f}_1({}_u n_1) \quad {}_u\ddot{f}_2({}_u n_2) \quad \dots \quad {}_u\ddot{f}_{S_u}({}_u n_{S_u})) \quad (2-64)$$

and \hat{S}_{um}^k is a *row* vector. Equivalent first-order sensitivity matrix formulas are also stated in [150-152, 154] (for single-hidden-layer MFANNs only) and [153, 155]. The non-trivial floating-point operations and storage space requirements for the calculation of first- and second-order sensitivities by (2-58) and (2-59) are given in table 2-4.

In the following discussion of the SOSA algorithm, layer indices l and m are fixed in value. Also, it is assumed that l points to the input layer and m to the output layer of the network, so that the algorithm's scope of action is voluntarily limited to an arbitrary section of the original network. If the input vector p_q is applied to the input layer of the original network, let $a_l \equiv ({}_l a_1 \quad {}_l a_2 \quad \dots \quad {}_l a_{S_l})^T$ denote the output vector of layer l , $a_m \equiv ({}_m a_1 \quad {}_m a_2 \quad \dots \quad {}_m a_{S_m})^T$ the output vector of layer m , and

$\tilde{\mathbf{a}}_m \equiv (\tilde{a}_1 \quad \tilde{a}_2 \quad \dots \quad \tilde{a}_{S_m})^T$ the target vector \mathbf{t}_q of the original network reduced to layer m . Vectors \mathbf{a}_l , \mathbf{a}_m and $\tilde{\mathbf{a}}_m$ are therefore the input, output and target of the MFANN section S_1 - S_{l+1} -...- S_m and correspond to the input-target vector set $\{\mathbf{p}_q, \mathbf{t}_q\}$ of the original network.

Table 2-4 – Floating-point requirements for MFANN sensitivity calculation

Sensitivity Order	Operations	Storage Space
First	$O\left(Q \cdot S_m \cdot \sum_{u=l+1}^m S_u S_{u-1}\right)$	$O\left(Q \cdot S_m \cdot \sum_{u=l}^m S_u\right)$
Second	$O\left(Q \cdot S_m \cdot \sum_{u=l+1}^m S_u S_{u-1} (S_{u-1} + S_u)\right)$	$O\left(Q \cdot S_m \cdot \sum_{u=l}^m (S_u)^2\right)$

If one, two or three nodes, indexed i , $\{i, j\}$ ($i \neq j$) and $\{i, j, h\}$ ($i \neq j, j \neq h, i \neq h$) respectively, are removed together from input layer l , then \mathbf{a}_l will change accordingly to \mathbf{a}_l^i , \mathbf{a}_l^j or \mathbf{a}_l^{jh} . Each of the three new input vectors differs from \mathbf{a}_l only in elements considered removed, or, equivalently, forced to zero. In mathematical terms:

$${}_l \mathbf{a}_r^i \equiv \begin{cases} {}_l \mathbf{a}_r & r = 1, 2, \dots, i-1, i+1, \dots, S_l \\ 0 & r = i \end{cases} \quad (2-65)$$

$${}_l \mathbf{a}_r^j \equiv \begin{cases} {}_l \mathbf{a}_r & r \in \{1, 2, \dots, S_l\} - \{i, j\} \\ 0 & r \in \{i, j\} \end{cases} \quad (2-66)$$

$${}_l \mathbf{a}_r^{jh} \equiv \begin{cases} {}_l \mathbf{a}_r & r \in \{1, 2, \dots, S_l\} - \{i, j, h\} \\ 0 & r \in \{i, j, h\} \end{cases} \quad (2-67)$$

The input vector change due to input elimination is represented by:

$$\Delta \mathbf{a}_l^i \equiv \mathbf{a}_l^i - \mathbf{a}_l \Rightarrow {}_l \Delta \mathbf{a}_r^i = \begin{cases} -{}_l \mathbf{a}_r & r = i \\ 0 & r = 1, 2, \dots, i-1, i+1, \dots, S_l \end{cases} \quad (2-68)$$

$$\Delta \mathbf{a}_l^j \equiv \mathbf{a}_l^j - \mathbf{a}_l \Rightarrow {}_l \Delta \mathbf{a}_r^j = \begin{cases} -{}_l \mathbf{a}_r & r \in \{i, j\} \\ 0 & r \in \{1, 2, \dots, S_l\} - \{i, j\} \end{cases} \quad (2-69)$$

$$\Delta \mathbf{a}_l^{jh} \equiv \mathbf{a}_l^{jh} - \mathbf{a}_l \Rightarrow {}_l \Delta \mathbf{a}_r^{jh} = \begin{cases} -{}_l \mathbf{a}_r & r \in \{i, j, h\} \\ 0 & r \in \{1, 2, \dots, S_l\} - \{i, j, h\} \end{cases} \quad (2-70)$$

An alternative way to consider the MFANN section defined between layers l and m and having twice-differentiable transfer functions assigned to its neurons is as

being a twice-differentiable non-linear mapping from \mathbb{R}^{S_l} to \mathbb{R}^{S_m} . Therefore the simultaneous elimination of a number of nodes from input layer l could be regarded as changes to zero of the corresponding elements in all input vectors of the equivalent mapping. If Taylor series approximations are used to measure the changes in all output vectors of layer m due to selective input pruning, the first- and second-order sensitivity parameters defined for the MFANN section $S_1-S_{l+1}-\dots-S_m$ are applied as coefficients to these approximations. Specifically:

$${}^l a_k^i = {}_m a_k + \hat{S}_{lm}^k \cdot \Delta a_l^i + \frac{1}{2} \cdot (\Delta a_l^i)^T \cdot \hat{H}_{lm}^k \cdot \Delta a_l^i + R(\Delta a_l^i) \quad (2-71)$$

$${}^l a_k^y = {}_m a_k + \hat{S}_{lm}^k \cdot \Delta a_l^y + \frac{1}{2} \cdot (\Delta a_l^y)^T \cdot \hat{H}_{lm}^k \cdot \Delta a_l^y + R(\Delta a_l^y) \quad (2-72)$$

$${}^l a_k^{yh} = {}_m a_k + \hat{S}_{lm}^k \cdot \Delta a_l^{yh} + \frac{1}{2} \cdot (\Delta a_l^{yh})^T \cdot \hat{H}_{lm}^k \cdot \Delta a_l^{yh} + R(\Delta a_l^{yh}) \quad (2-73)$$

where ${}^l a_m^i \equiv ({}^l a_{m1}^i \quad {}^l a_{m2}^i \quad \dots \quad {}^l a_{mS_m}^i)^T$, ${}^l a_m^y \equiv ({}^l a_{m1}^y \quad {}^l a_{m2}^y \quad \dots \quad {}^l a_{mS_m}^y)^T$ and ${}^l a_m^{yh} \equiv ({}^l a_{m1}^{yh} \quad {}^l a_{m2}^{yh} \quad \dots \quad {}^l a_{mS_m}^{yh})^T$ are the output vectors of layer m after the simultaneous elimination of one, two and three inputs from layer l, and $R()$ is the remainder term of Taylor series [157]. To claim that $R()$ is small and thus equations (2-71) to (2-73) become accurate second-order approximations without it, it is essential to ensure that input change vectors Δa_l^i , Δa_l^y and Δa_l^{yh} have as small non-zero elements as possible. Indeed, uniform normalisation of all vectors p_q , $1 \leq q \leq Q$, before they are applied as inputs to the original network is sufficient to fulfil the aforementioned requirement when $l=0$. Should $0 < l < L$, the assignment of transfer functions having narrow ranges around zero, such as log-sigmoid or hyperbolic tangent sigmoid, to all neurons of layer l serves the same purpose of fulfilment. As a matter of fact, the uniform normalisation of input vectors p_q ensures the validity of Taylor approximations arisen from equations (2-71) to (2-73) for the vast majority of MFANNs.

To determine the significance of each node belonging to MFANN section between layers l and m, it is necessary to define and use a proper function of network output change. When all inputs of layer l are present, the output prediction error in the k-th output of layer m is defined as:

$${}_m E_k \equiv {}_m a_k - {}_m \tilde{a}_k \quad (2-74)$$

If one, two or three inputs of layer l are removed altogether, the corresponding output prediction error for every such case is defined as:

$${}^i E_k^i \equiv {}^i a_k^i - {}_m \tilde{a}_k \quad (2-75)$$

$${}^i E_k^y \equiv {}^i a_k^y - {}_m \tilde{a}_k \quad (2-76)$$

$${}^i E_k^{yh} \equiv {}^i a_k^{yh} - {}_m \tilde{a}_k \quad (2-77)$$

When definitions (2-74) to (2-77) are applied to equations (2-71) to (2-73) and the outcomes are then turned into approximations by elimination of their remainder terms, the following results are finally obtained:

$${}^i E_k^i \cong {}_m E_k + \hat{S}_{im}^k \cdot \Delta a_i^i + \frac{1}{2} \cdot (\Delta a_i^i)^T \cdot \hat{H}_{im}^k \cdot \Delta a_i^i \quad (2-78)$$

$${}^i E_k^y \cong {}_m E_k + \hat{S}_{im}^k \cdot \Delta a_i^y + \frac{1}{2} \cdot (\Delta a_i^y)^T \cdot \hat{H}_{im}^k \cdot \Delta a_i^y \quad (2-79)$$

$${}^i E_k^{yh} \cong {}_m E_k + \hat{S}_{im}^k \cdot \Delta a_i^{yh} + \frac{1}{2} \cdot (\Delta a_i^{yh})^T \cdot \hat{H}_{im}^k \cdot \Delta a_i^{yh} \quad (2-80)$$

Approximations (2-78) to (2-80) demonstrate how the SOSA algorithm has managed to combine the ease of *relative* sensitivity ($\hat{S}_{im}^k, \hat{H}_{im}^k$) calculation with the accuracy of *absolute* network sensitivity i.e. the new network output error after input pruning. This fact is indeed the algorithm's beauty.

Two facts ought to be clarified before the presentation of SOSA continues. Firstly, the algorithm measures the sensitivity of node elimination to network output error, i.e., a linear function of the network output change. However, the network output error approximations (2-78) to (2-80) should *not* be confused with performance index functions like mean- or sum-squared error. The former function class acts on every network output *individually*, while the latter class treats network outputs *en masse* and thus smoothes out any existing behavioural variations between them. Secondly, simultaneous elimination of four or more inputs is *not* dealt with by SOSA. Should this be attempted, the additional Taylor approximations would require more terms, i.e. sensitivities of higher order, to remain accurate. However, the calculation of higher-order sensitivities would require too many additional computational and storage resources that could make SOSA unfeasible even for moderate-sized networks. Moreover, simultaneous elimination of many cross-correlated inputs from the network could result in hidden nodes driven into non-linear or totally saturated regions from which they are unable to represent useful input-output associations of the modelled mapping [144]. That scenario would definitely jeopardise the network's modelling efficiency if it ever came into play.

The approximations of network output error due to input node pruning (2-78) to (2-80) can be tabulated for elegance purposes. For this reason the updated output errors are grouped into vectors and matrices defined as follows:

$${}^m \mathbf{E}_1^k(i) \equiv {}^l E_k^i \tag{2-81}$$

$${}^m \mathbf{E}_2^k(i, j) \equiv {}^l E_k^{ij} \tag{2-82}$$

$${}^m \mathbf{E}_{3_h}^k(i, j) \equiv {}^l E_k^{ijh} \tag{2-83}$$

The proposed tabulation scheme can be combined with the substitution of common expressions present in approximations (2-78) to (2-80) by intermediate variables. That way, the overall computational and storage resources required for the calculation of network output errors defined in (2-75) to (2-77) are reduced considerably. The introduced variables are defined as scalar products of first- and second-order sensitivities by the outputs of their associated input-layer nodes and presented below for clarity purposes.

$$\hat{\mathbf{S}}_{lm}(k, i) \equiv {}_{lm} \hat{S}_{ki} \cdot {}_l \mathbf{a}_i \tag{2-84}$$

$$\hat{\mathbf{H}}_{lm}^k(i, j) \equiv {}_{lm} \hat{H}_{ij}^k \cdot {}_l \mathbf{a}_i \cdot {}_l \mathbf{a}_j \tag{2-85}$$

If the matrix symbol $\mathbf{1}$ is used to denote any vector or matrix of suitable dimensions having all its elements equal to one and the operator symbol \circ is introduced to denote vector or matrix element-by-element multiplication, scalar definitions (2-84) and (2-85) can easily be tabulated as follows:

$$\hat{\mathbf{S}}_{lm} \equiv \hat{S}_{lm} \circ (\mathbf{1} \cdot \mathbf{a}_l^T) \tag{2-86}$$

$$\hat{\mathbf{H}}_{lm}^k \equiv \hat{H}_{lm}^k \circ (\mathbf{a}_l \cdot \mathbf{a}_l^T) \tag{2-87}$$

The calculation of $\hat{\mathbf{S}}_{lm}$ from (2-84) requires $O(QS_m S_l)$ non-trivial floating-point operations and as much floating-point storage space, while the computation of $\hat{\mathbf{H}}_{lm}^k$ by (2-85) requires $O(QS_m S_l^2)$ additional operations and storage space.

Three more variables, the row vectors $\hat{\mathbf{S}}_{lm}^k$, ${}^l \hat{\mathbf{H}}_{lm}^k$ and column vector $\hat{\mathbf{H}}_{lm}^k \mathbf{a}_l$, are introduced for elegance only and defined as follows:

$$\hat{\mathbf{S}}_{lm}^k(i) \equiv {}_{lm} \hat{S}_{ki} \tag{2-88}$$

$${}^l \hat{\mathbf{H}}_{lm}^k(j) \equiv {}_{lm} \hat{H}_{ij}^k \tag{2-89}$$

$$\hat{\mathbf{H}}_{lm}^k \mathbf{a}_l(i) \equiv {}_{lm} \hat{H}_{ij}^k \tag{2-90}$$

By applying approximations (2-78) to (2-80) to the corresponding definitions (2-81) to (2-83) and inserting variables (2-86) to (2-90) to that first outcome by substitution, it is derived that the network output errors for all input pruning scenarios considered by SOSA are given by the following compact approximations:

$${}^m \mathbf{E}_1^k \cong {}_m E_k \cdot \mathbf{1} - (\hat{\mathbf{S}} \mathbf{a}_{lm}^k)^T + \frac{1}{2} \cdot \hat{\mathbf{H}} \mathbf{a} \mathbf{a}_{lm}^k \quad (2-91)$$

$${}^m \mathbf{E}_2^k \cong {}^m \mathbf{E}_1^k \cdot \mathbf{1} + \mathbf{1} \cdot ({}^m \mathbf{E}_1^k)^T - {}_m E_k \cdot \mathbf{1} + \hat{\mathbf{H}} \mathbf{a}_{lm}^k \quad (2-92)$$

$${}^m \mathbf{E}_{3_s}^k \cong {}^m \mathbf{E}_2^k + ({}^l E_k^h - {}_m E_k) \cdot \mathbf{1} + \mathbf{1} \cdot {}^h \hat{\mathbf{H}} \mathbf{a}_{lm}^k + ({}^h \hat{\mathbf{H}} \mathbf{a}_{lm}^k)^T \cdot \mathbf{1} \quad (2-93)$$

These final approximations have been stated in a manner that allows utilisation of already obtained error results to the calculation of other output errors. This identity is known as *encapsulation* and reduces even further the computational requirements for the expressions sharing it. Assuming that the set of initial network output errors ${}_m E_k$ defined in (2-74) and all intermediate variables appearing in (2-91) to (2-93) have already been computed and stored, the final error approximations require $O(Q S_m S_l^g)$ non-trivial floating-point operations and $O(Q S_m S_l^g)$ floating-point numbers storage space each (g equals the number of nodes pruned together from layer l).

At this point it is interesting to see why the updated network output errors ${}^m \mathbf{E}_1^k$, ${}^m \mathbf{E}_2^k$ and ${}^m \mathbf{E}_{3_s}^k$ are calculated by second-order Taylor approximations rather than directly from the network section between layers l and m . The latter method removes one to three nodes from layer l simultaneously and then applies the remaining inputs to measure the new output errors. However, the propagation of all Q input vectors \mathbf{a}_l available through the l -to- m network section costs $O\left(Q \cdot \sum_{u=l+1}^m S_{u-1} S_u\right)$ floating-point operations. For output error calculation due to three-node elimination, the input propagation has to be repeated $O(S_l^3)$ times to cover all possible three-node combinations. The result is that the brute-force method of direct error calculation requires $O\left(Q \cdot S_l^3 \cdot \sum_{u=l+1}^m S_{u-1} S_u\right)$ floating-point operations. In contrast, the error approximation method followed by SOSA requires $O\left(Q \cdot S_m \cdot \sum_{u=l+1}^m S_u S_{u-1} (S_{u-1} + S_u)\right)$ floating-point operations at worst for the calculation of all second-order sensitivity matrices $\hat{\mathbf{H}}_{um}^k$ for

$1 \leq u \leq m$ and $1 \leq k \leq S_m$. A comparison of the two results shows clearly that the error approximation method is an order of magnitude less expensive regarding the number of nodes S_u existing in a given network. This comparison outcome agrees with the intuitive feeling that the approximation of three-node elimination, a problem of three independent parameters and thus of third-order complexity, by terms of up to second order and hence of similar degree of complexity, would eventually produce computational savings of one order of magnitude. These savings come at the cost of uncertainty inherent in all approximations and extra storage space required for the variables participating in the error approximation method, but this cost is tolerable as far as small- or medium-sized MFANNs are concerned.

The next goal of SOSA algorithm is to assign a significance value to every set of input nodes that can be considered for simultaneous pruning. To achieve that goal, the algorithm begins with the network outputs ${}^m_1E_1^k$, ${}^m_1E_2^k$, ${}^m_1E_3^k$ and then removes the dependency of these variables on the input vector p_q applied to the original network as well as the m-layer output node k. That elimination task is actually performed in two steps, the first being against input vector dependency and the next tackling output node reliance.

Elimination of input vector dependency from ${}^m_1E_1^k$, ${}^m_1E_2^k$, ${}^m_1E_3^k$ is achieved by selective averaging of all error elements that correspond to the same input and output node combination but calculated for different input vectors otherwise. The associated averaged errors are denoted as ${}^m_1Eavg_1^k$, ${}^m_1Eavg_2^k$, ${}^m_1Eavg_{3_k}^k$ and defined as follows:

$${}^m_1Eavg_1^k(i) \equiv Q^{-\frac{1}{r}} \cdot \left\| \left[\begin{array}{cccc} {}^l E_k^i |_{p_1} & {}^l E_k^i |_{p_2} & \dots & {}^l E_k^i |_{p_q} & \dots & {}^l E_k^i |_{p_Q} \end{array} \right]^T \right\|_r \quad (2-94)$$

$${}^m_1Eavg_2^k(i, j) \equiv Q^{-\frac{1}{r}} \cdot \left\| \left[\begin{array}{cccc} {}^l E_k^{ij} |_{p_1} & {}^l E_k^{ij} |_{p_2} & \dots & {}^l E_k^{ij} |_{p_q} & \dots & {}^l E_k^{ij} |_{p_Q} \end{array} \right]^T \right\|_r \quad (2-95)$$

$${}^m_1Eavg_{3_k}^k(i, j) \equiv Q^{-\frac{1}{r}} \cdot \left\| \left[\begin{array}{cccc} {}^l E_k^{ijh} |_{p_1} & {}^l E_k^{ijh} |_{p_2} & \dots & {}^l E_k^{ijh} |_{p_q} & \dots & {}^l E_k^{ijh} |_{p_Q} \end{array} \right]^T \right\|_r \quad (2-96)$$

where integer r can be 1, 2 or ∞ . The names given to the averaging methods followed by (2-94) to (2-96) for each value r can take is *absolute mean* for $r=1$, *Root Mean Square (RMS)* for $r=2$ and *absolute maximum* for $r=\infty$. The norm operator $\|\cdot\|_\infty$ is defined for an arbitrary vector x as:

$$\|\mathbf{x}\|_{\infty} \equiv \max_i(|x_i|) \quad (2-97)$$

Also, definition (2-96) remains valid for $r=\infty$ as $x^{-1/\infty} \equiv 1 \quad \forall x \in \mathbb{R}^*$. No matter which averaging method being selected, definitions (2-94) to (2-96) require $O(QS_m S_l^g)$ non-trivial floating-point operations and $O(S_m S_l^g)$ floating-point numbers storage space each, where g equals the number of nodes considered for simultaneous pruning from layer l .

RMS averaging is preferred when no assumptions about the updated output error disparity can be made mainly because the formula of that method resembles the performance index functions usually chosen for MFANN training, i.e., linear functions of square output errors obtained for the whole training vector set. Alternatively, the absolute mean averaging method is used when output errors are quite disparate in nature. Finally, absolute maximum averaging is applied when it is of great interest to prevent the elimination of nodes belonging to layer l and being significant only for a small subset of input vectors applied to the original network [151]. Any future reference to SOSA algorithm in this study without explicit indication to the method selected for output error averaging will assume RMS averaging by default.

A side effect of the use of norm in definitions (2-94) to (2-96) is that their averaged outcomes are always non-negative and always measure the average magnitude of output errors without paying any attention to the signs of those errors. This effect is welcome by SOSA because it is the magnitude of error, not its direction, which should determine whether a set of nodes could be pruned from a MFANN without causing irreparable damage to its modelling accuracy and efficiency.

Removal of output node dependence from input-averaged errors ${}^m_i \mathbf{Eavg}_1^k$, ${}^m_i \mathbf{Eavg}_2^k$, ${}^m_i \mathbf{Eavg}_3^k$ is accomplished by linear averaging of these error elements that correspond to the same input node combination but calculated for different output nodes instead. The result is called the *significance* of the associated input node set and defined as follows:

$$\Phi_{im}^r \equiv \frac{1}{S_m} \cdot \sum_{k=1}^{S_m} {}^m_i \mathbf{Eavg}_r^k \quad r = 1, 2, 3_h \quad (2-98)$$

The set of significance matrices $\Phi_{im}^{3_h}$ is used next to produce the augmented matrix Φ_{im}^3 defined as follows:

$$\Phi_{lm}^3 \equiv [\Phi_{lm}^{3_1} | \Phi_{lm}^{3_2} | \dots | \Phi_{lm}^{3_h} | \dots | \Phi_{lm}^{3_{S_l}}] \quad (2-99)$$

whose elements are denoted by:

$${}_{lm}\Phi_{yh}^3 \equiv \Phi_{lm}^3(i, (h-1)S_l + j) \quad (2-100)$$

Significance variables Φ_{lm}^1 , Φ_{lm}^2 and Φ_{lm}^3 require $O(S_m S_l^g)$ non-trivial floating-point operations and $O(S_l^g)$ floating-point number storage space, where g equals the number of input nodes considered for simultaneous elimination.

The significance values of interest as far as SOSA is concerned are the *minimal significances*. Each minimal significance value represents the smallest average output error expected by the network section defined between layers l and m when a certain number of nodes are removed from section's input layer and *no retraining* has taken place yet. Three minimal significance parameters are thus defined for the purposes of SOSA algorithm, and these definitions are.

$${}_{lm}\phi_1^s \equiv \min_i ({}_{lm}\Phi_i^1) \quad (2-101)$$

$${}_{lm}\phi_2^{tv} \equiv \min_{i,j} ({}_{lm}\Phi_{ij}^2) \quad (2-102)$$

$${}_{lm}\phi_3^{xyz} \equiv \min_{i,j,h} ({}_{lm}\Phi_{yh}^3) \quad (2-103)$$

Number sets s , $\{t, v\}$ and $\{x, y, z\}$ displayed in (2-101) to (2-103) point to node(s) of layer l whose (joint) significance equals the minimal significance value associated with them.

The ultimate task of SOSA is to decide which nodes should be pruned from layer l of the original network, if any, so that the pruned network suffers the least damage and thus is most likely to recover that damage by retraining. This decision is entirely based on criteria involving only minimal significances and heuristically assigned numerical constants. The first criterion determines whether three nodes can safely be pruned altogether from layer l , and takes the form of the following ratio comparison:

$$\frac{{}_{lm}\phi_3^{xyz}}{{}_{lm}\phi_1^s} \leq c_3 \quad (2-104)$$

If ${}_{lm}\phi_1^s$ is non-zero and (2-104) succeeds, the set of nodes $\{x, y, z\}$ are pruned and the remaining network is retrained to compensate for the node loss. Otherwise, SOSA

seeks to eliminate two nodes together from layer l by checking whether ${}_{lm}\phi_1^s$ is non-zero and the following minimal significance ratio comparison holds:

$$\frac{{}_{lm}\phi_2^{tv}}{{}_{lm}\phi_1^s} \leq c_2 \quad (2-105)$$

If so, the set of nodes $\{t, v\}$ are pruned and the remaining network is retrained. Otherwise, the node-pruning algorithm does the following last check to determine whether it would be safe enough to remove a single node from the network:

$${}_{lm}\phi_1^s \leq c_1 \quad (2-106)$$

If (2-106) is true, the node s is pruned and the remaining network is retrained, or else, SOSA comes to a permanent halt.

A fair choice of constants c_3 and c_2 in (2-104) and (2-105) respectively is $c_3=3$, $c_2=2$. This choice has produced good results in a number of MFANN node optimisation tests including the ones presented in this work. Constant c_1 in (2-106) can be assigned to either a positive number when a good guess for it is somehow available to the network designer, or infinity otherwise. The latter choice for c_1 virtually disables comparison (2-106) so that a single node at least is pruned from the MFANN and the coming network retraining phase finally determines whether the last pruning decision was correct or not. As far as this study is concerned, $c_1=\infty$. A more accurate alternative to comparison constant assignments would involve the use of hypothesis testing in a manner similar to the ones encountered in [158-161]. That alternative is definitely promising but far too complicated to be considered by this work.

As already mentioned, the MFANN pruned by SOSA algorithm's decision is retrained in an effort to recover the node loss. If retraining is unsuccessful, a scheme called *backtracking* is adopted by SOSA. That is, the trained network version before the last pruning decision was taken is restored and inequalities (2-104) to (2-106) are checked again to produce a different pruning decision involving fewer nodes or none. Backtracking is repeated if necessary until either the pruned MFANN eventually passes the retraining phase successfully or no pruning decision is taken by SOSA and node optimisation stops. This way it is ensured that SOSA algorithm produces an accurate and smaller network in *every* node-pruning step.

A flowchart diagram of SOSA node-pruning algorithm is presented in figure 2-7. The diagram is grossly simplified to avoid congestion of its elements in the limited space provided by a single page, but it still includes the basic parts of the algo-

gorithm. The flowchart variables not explained in the detailed presentation of SOSA are the index of successful iterations r , the number of nodes pruned per iteration N_- , and the total number of nodes pruned from layer l , N_{all}^- . The node-pruning algorithm accepts an MFANN section bounded by layers indexed as l and m , $0 \leq l < m \leq L$, as well as all input vectors \mathbf{a}_l defined for that section and derived from the associated input vectors \mathbf{p} of the original network. The algorithm tries to eliminate up to three nodes from layer l per iteration, and stops when training fails in either the initial or most recently pruned l -to- m network section. SOSA returns the updated (pruned) set of input vectors, the total number of nodes pruned from layer l , and the total number of nodes remaining layer l after pruning.

2.9.3 Optimal Brain Surgeon Algorithm

The OBS algorithm described originally in [147] considers the performance index $F()$ of an arbitrary MFANN as being half of the mean-squared error function defined in (2-10), i.e.:

$$F(\mathbf{e}) \equiv \frac{1}{2Q} \cdot \left\| \left[\|\mathbf{e}_1\|_2^2 \quad \|\mathbf{e}_2\|_2^2 \quad \dots \quad \|\mathbf{e}_q\|_2^2 \quad \dots \quad \|\mathbf{e}_Q\|_2^2 \right]^T \right\|_1 \quad (2-107)$$

where $\mathbf{e} \equiv [\mathbf{e}_1 \mid \mathbf{e}_2 \mid \dots \mid \mathbf{e}_q \mid \dots \mid \mathbf{e}_Q]$ is the error vector of the network, $\mathbf{e}_q \equiv \mathbf{t}_q - \mathbf{a}_q$ is the error element that corresponds to the input-target training pair $\{\mathbf{p}_q, \mathbf{t}_q\}$ for $1 \leq q \leq Q$, and \mathbf{a}_q is the associated network output vector. Also, the algorithm assumes that the MFANN has already been trained to a local minimum of (2-107).

When an arbitrary weight x_r disappears from the network, it causes an overall weight change $\Delta \mathbf{x}$ given by:

$$\mathbf{v}_r^T \cdot \Delta \mathbf{x} + x_r = 0 \quad (2-108)$$

where:

$$\mathbf{x} \equiv [x_1 \quad \dots \quad x_n]^T = [{}_1w_{1,1} \quad \dots \quad {}_1w_{S_1,S_0} \quad {}_1b_1 \quad \dots \quad {}_1b_{S_1} \quad {}_2w_{1,1} \quad \dots \quad {}_Lb_{S_L}]^T \quad (2-109)$$

is the vector including all MFANN weights and biases, \mathbf{v}_r is the unit vector in weight space corresponding to x_r , $1 \leq r \leq n$, $S_0 = R$, and $n = \sum_{m=1}^L S_m (S_{m-1} + 1)$. Furthermore, the

mentioned weight elimination leads to a performance index change given by a Taylor series around x_r :

$$\Delta F = \left(\frac{\partial F}{\partial \mathbf{x}} \right)^T \cdot \Delta \mathbf{x} + \frac{1}{2} \cdot \Delta \mathbf{x}^T \cdot \mathbf{H} \cdot \Delta \mathbf{x} + O(\|\Delta \mathbf{x}\|_3^3) \quad (2-110)$$

where $\mathbf{H} \equiv \partial^2 F / (\partial \mathbf{x})^2$ is the Hessian matrix of $F()$ with respect to weight vector \mathbf{x} defined in (2-109), and $\|\cdot\|_3$ is the third-order norm function defined as:

$$\|\Delta \mathbf{x}\|_3 \equiv \sqrt[3]{\sum_i |\Delta x_i|^3} \quad (2-111)$$

For a MFANN trained to a local minimum in (2-107), the first (linear) term in (2-110) vanishes. Moreover, the third and higher order terms of Taylor approximation (2-110) are ignored because they are assumed to be quite small. Therefore OBS approximates the performance index change ΔF due to pruning of weight x_r as follows:

$$\Delta F \cong \frac{1}{2} \cdot \Delta \mathbf{x}^T \cdot \mathbf{H} \cdot \Delta \mathbf{x} \quad (2-112)$$

The optimal weight pruning decision according to OBS algorithm is the one that minimises (2-112) subject to condition (2-108) being satisfied. The optimal solution is actually found by minimisation of the Lagrangian function $L()$ given by:

$$L(\Delta \mathbf{x}) = \frac{1}{2} \cdot \Delta \mathbf{x}^T \cdot \mathbf{H} \cdot \Delta \mathbf{x} + \lambda \cdot (\mathbf{v}_r^T \cdot \Delta \mathbf{x} + x_r) \quad (2-113)$$

where λ is a Lagrange undetermined multiplier. The solution is:

$$\min_r(\Delta \mathbf{x}) = -\frac{x_r}{H_{rr}^{-1}} \cdot \mathbf{H}^{-1} \cdot \mathbf{v}_r \quad (2-114)$$

$$\min_r(L) = \frac{1}{2} \cdot \frac{x_r^2}{H_{rr}^{-1}} \quad (2-115)$$

The outcome of (2-115) is called the *saliency* of weight x_r .

The inverse Hessian matrix \mathbf{H}^{-1} and element H_{rr}^{-1} terms of (2-114) and (2-115) can be computed efficiently if the set of vectors \mathbf{X}_k^q defined as:

$$\mathbf{X}_k^q \equiv \frac{\partial_L \mathbf{a}_k}{\partial \mathbf{x}} \Big|_{p_q} = \left(\frac{\partial_L \mathbf{a}_k}{\partial x_1} \Big|_{p_q} \quad \frac{\partial_L \mathbf{a}_k}{\partial x_2} \Big|_{p_q} \quad \dots \quad \frac{\partial_L \mathbf{a}_k}{\partial x_n} \Big|_{p_q} \right)^T \quad (2-116)$$

are calculated first for $1 \leq k \leq S_L$ and $1 \leq q \leq Q$. This task, however, requires previous knowledge of all first-order sensitivities ${}_{ll} \hat{S}_{kl}$ as defined by (2-54) for $1 \leq l \leq L$. When these sensitivities are computed by (2-56) for every training input vector \mathbf{p}_q , vectors \mathbf{X}_k^q can be calculated by definitions (2-109) and (2-116) as well as the equations:

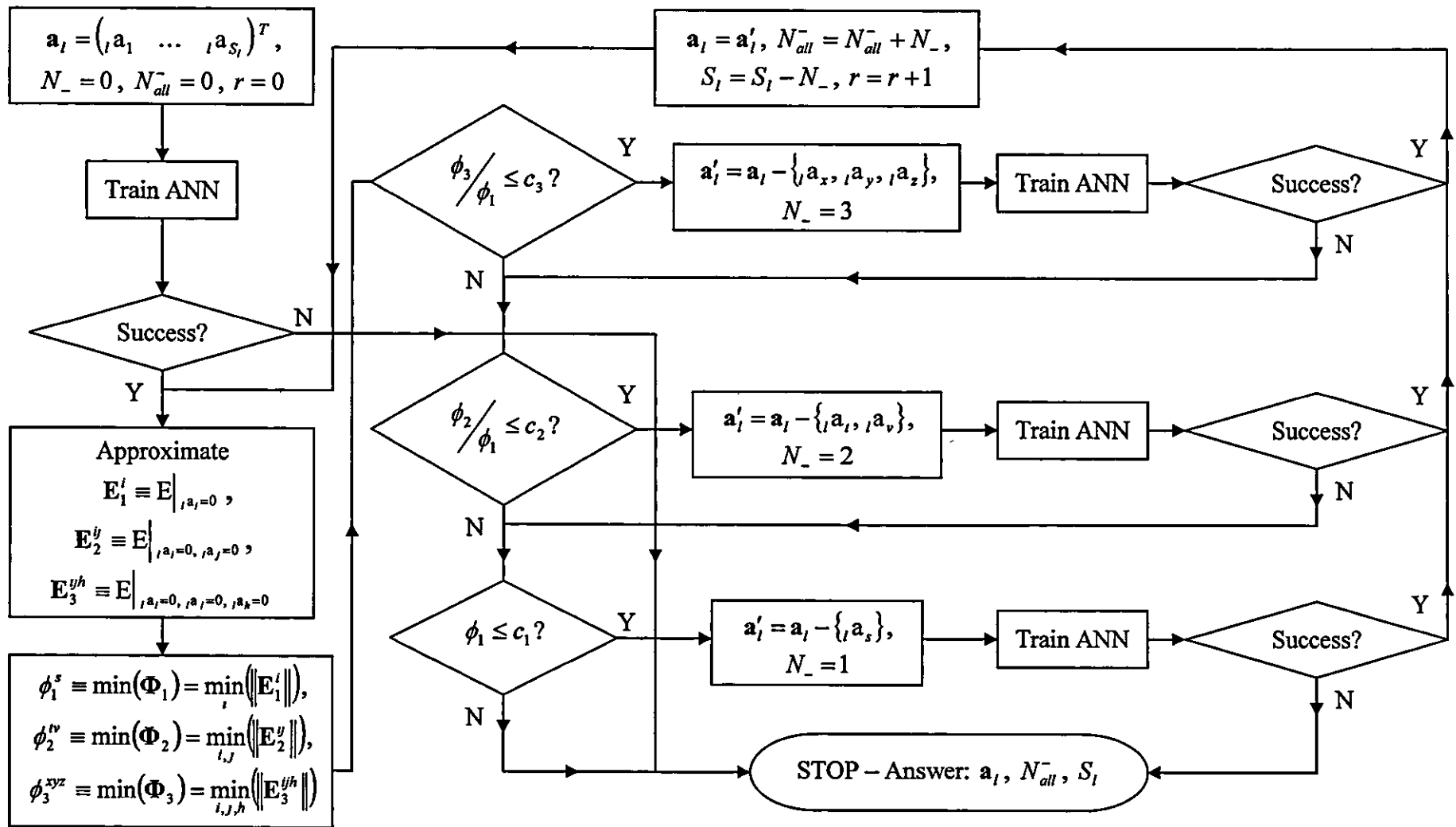


Figure 2-7 – Original Second-Order Sensitivity Analysis (SOSA) flowchart

$$\frac{\partial_L \mathbf{a}_k}{\partial_l w_{ij}} = {}_L \hat{S}_{ki} \cdot {}_l \dot{f}_i({}_l n_i) \cdot {}_{l-1} \mathbf{a}_j \quad (2-117)$$

$$\frac{\partial_L \mathbf{a}_k}{\partial_l b_i} = {}_L \hat{S}_{ki} \cdot {}_l \dot{f}_i({}_l n_i) \quad (2-118)$$

where $1 \leq i \leq S_l$ and $1 \leq j \leq S_{l-1}$. Finally, the inverse Hessian is calculated by recursion that starts from initial condition (2-119) and continues with repetitive use of equations (2-120) for $1 \leq q \leq Q$, $1 \leq k < S_L$ (and $(q, k) = (1, 0)$) and (2-121) for $1 \leq q < Q$ until final condition (2-122) is reached:

$$\mathbf{H}_{10}^{-1} = \alpha^{-1} \cdot \mathbf{I} \quad (2-119)$$

$$\mathbf{H}_{q(k+1)}^{-1} = \mathbf{H}_{qk}^{-1} - \frac{\mathbf{H}_{qk}^{-1} \cdot \mathbf{X}_{k+1}^q \cdot (\mathbf{X}_{k+1}^q)^T \cdot \mathbf{H}_{qk}^{-1}}{Q + (\mathbf{X}_{k+1}^q)^T \cdot \mathbf{H}_{qk}^{-1} \cdot \mathbf{X}_{k+1}^q} \quad (2-120)$$

$$\mathbf{H}_{(q+1)l}^{-1} = \mathbf{H}_{qS_L}^{-1} - \frac{\mathbf{H}_{qS_L}^{-1} \cdot \mathbf{X}_l^{q+1} \cdot (\mathbf{X}_l^{q+1})^T \cdot \mathbf{H}_{qS_L}^{-1}}{Q + (\mathbf{X}_l^{q+1})^T \cdot \mathbf{H}_{qS_L}^{-1} \cdot \mathbf{X}_l^{q+1}} \quad (2-121)$$

$$\mathbf{H}^{-1} = \mathbf{H}_{QS_L}^{-1} \quad (2-122)$$

The term α in (2-119) is a small constant ($10^{-8} \leq \alpha \leq 10^{-4}$, typically $\alpha = 10^{-6}$) needed to make \mathbf{H}_{10}^{-1} meaningful and to which OBS is insensitive.

The OBS algorithm considers a weight x_r as being optimal for pruning if it gives the smallest saliency value from any other weight of the trained network. When the smallest saliency is less than a predetermined tolerance parameter L_{\max} , the algorithm permanently deletes weight and updates all remaining weights by (2-114), i.e. *without* retraining of the network. Otherwise OBS does not prune any weight and waits until the network is retrained to take another pruning chance. The algorithm stops running when it fails to eliminate any weight between two successive network-retraining attempts.

2.9.4 Complete MFANN optimisation scheme

The SOSA and OBS algorithms presented above deal primarily with the node and weight optimisation of any MFANN structure. However, a network can be acknowledged as truly optimal only if it is shown to have the least complexity possible in order to model adequately the problem for which it is designed. Such a network is assembled by an optimal number of nodes *and* weights (node interconnections), and thus it has to be the outcome of a process that involves both node *and* weight optimi-

sation algorithms acting in a harmonious, precisely determined sequence. The process of multiple network optimisations is described by the term *optimisation scheme* and a novel example of that process designed to act on MFANNs only and involving both SOSA and OBS algorithms is the main subject of this subsection.

The original SOSA algorithm has been designed in such a way that it performs node optimisation on any (but the output) MFANN layer *independently* from the others. The order by which SOSA is applied on network layers is therefore important for its pruning performance as far as the complete modelling structure is concerned. The less the redundancy in between the layer considered for node optimisation and the output layer of the MFANN, the fewer the adjustable internal parameters that assist the model to compensate for the node loss by retraining, and hence the more difficult to achieve good node pruning results. For best results, the node elimination strategy followed in the case of SOSA has to be based on these guidelines:

- a) Node elimination should be performed on network layers *sequentially*, i.e., beginning with the input layer and ending with the hidden layer next to the output one. That way it is ensured that the best possible optimisation job is done regarding the input layer while hidden layer pruning results remain at satisfactory levels. Besides, it is much more preferable in economic and practical grounds to build a model with some internal redundancy still in residue rather than build another model that requires more variables of the associated problem to be known or measured in advance. The importance of this statement will be seen clearly when a model for the inverse optical scattering will be developed in chapter 3.
- b) The node-pruning algorithm should allow the controlled addition of nodes in the hidden layers lying between the layer considered for optimisation and the output layer when further node elimination in the layer currently under optimisation looks impossible. When this element of strategy is combined with the previous one that dictates the layer optimisation order, it can only result in overall improved pruning performance. The more redundancy remains in the hidden layers of a MFANN section, the more likely the further pruning of the section's input layer will be. Even if the hopes for further pruning gains regarding the section's input layer are proved to be futile, the redundancy added to the hidden layers of that section will be removed anyway, as these layers will be the next ones considered for node optimisation.

A flowchart of the modified SOSA algorithm that incorporates the redundant node addition feature suggested above is shown in figure 2-8. The additional variables of the modified algorithm are the number of nodes added between two consecutive and successful node elimination attempts, N_+ , the maximum number of nodes allowed for addition between two consecutive and successful node pruning attempts, N_{\max}^+ , and the total number of nodes added to layers $l+1$ to $m-1$, N_{all}^+ . The new algorithm adds one redundant node per hidden layer numbered from $l+1$ to $m-1$ in the original MFANN when the minimal significance criteria (2-104) to (2-106) fail to suggest a set of nodes to be pruned from layer l , and the augmented network is re-trained only to be examined again for node elimination at layer l . As soon as node addition at hidden layers $l+1$ to $m-1$ results in further node pruning at layer l , the added node counter N_+ is reset to allow further node additions at a later stage of the modified SOSA algorithm. Otherwise the new algorithm halts when node counter N_+ reaches or exceeds counter N_{\max}^+ . The latter variable has to be empirically assigned to a multiple of $m-(l+1)$ that maximises the prospects of efficient pruning of nodes belonging to layer l ; a value of 2 to 3 times $m-(l+1)$ usually suffices. As mentioned earlier, any redundant nodes remaining in layers $l+1$ to $m-1$ after the end of node optimisation at layer l will be removed when the same node optimisation procedure is applied to each of the former layer set and in ascending layer order.

The complete MFANN optimisation scheme is summarised in the flowchart of figure 2-9. The additional monitored parameters are the total number of nodes removed from the input layer of the original network structure (layer 0), N_{inp}^- , the total number of nodes pruned from all hidden layers, N_{hid}^- , and the total number of nodes added to hidden nodes as node optimisation progresses from a layer to another, N_{hid}^+ . Beginning with the MFANN input layer, the scheme optimises the number of nodes of each but the output layer in ascending layer order. Although the scheme itself does not name an algorithm for node optimisation, it is implied that the modified SOSA algorithm is employed for the task. After node optimisation phase is complete, the network is optimised for weights by the OBS (or another suitable) algorithm. If node optimisation succeeded in pruning at least one node from the input layer, the node- and weight-optimisation cycle is repeated for the remaining network structure. The same course of action is followed in case there were no input node eliminations but

the number of hidden nodes pruned from the network is bigger than the number of redundant nodes added to it during the last optimisation cycle. Otherwise the optimisation scheme comes to a halt and returns the trimmed network as its outcome.

Unless otherwise stated, the scheme of figure 2-9 is implied to be the one used for optimisation of a MFANN model developed and discussed from this point on.

2.10 ANN Modelling Examples

2.10.1 Overview

Multi-Forward Artificial Neural Networks can deliver modelling solutions to a plethora of real problems that can be stated in a form or another as functional mappings of unknown mathematical description but well-known input-to-output data coupling examples. Network training, data preconditioning and network optimisation are procedures that have already been discussed independently as key steps to the construction of an efficient MFANN model for a given problem. This section demonstrates how these procedures can be used together to build network models for a couple of illustrative problems. The first problem is the design of an optimal Fourier series generator model that can alternatively be derived by ordinary mathematical analysis. The second problem is the finding of the optimal approximation model to a function whose mathematical description is known in advance when noisy input sources are mixed up with the important ones.

2.10.2 Fourier Series Generator

A *Fourier series generator* is a mathematical tool that approximates a periodic signal by a finite set of sinusoidal terms contained in the discrete Fourier series derived by Fourier analysis of the original signal. The accuracy of that tool has a margin that is defined by the Fourier theorem and depends on the number and order of terms applied as inputs to the generator.

The modelling task is the design of an optimal MFANN that can perform as a Fourier series generator of a given periodic signal when Fourier terms of that signal mixed with irrelevant sinusoidal functions are provided as candidate input sources to the network model. The periodic signal is a square-wave pulse train defined as:

$$\tilde{y}_1(t) = \begin{cases} +0.5 & nT \leq t < nT + \tau/2 \text{ and } (n+1)T - \tau/2 \leq t < (n+1)T \\ -0.5 & nT + \tau/2 \leq t < (n+1)T - \tau/2 \end{cases} \quad (2-123)$$

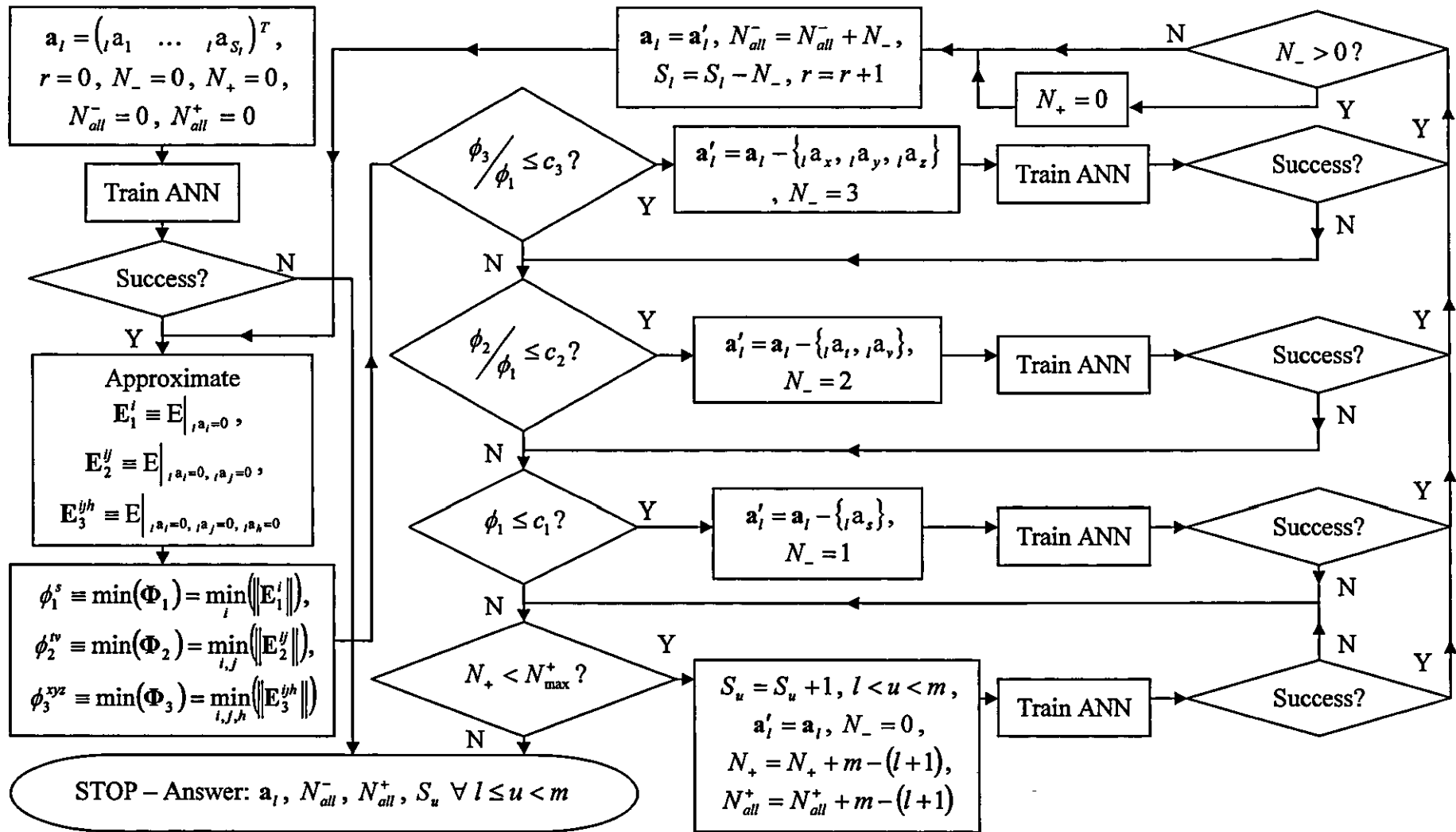


Figure 2-8 – Modified Second-Order Sensitivity Analysis (SOSA) flowchart

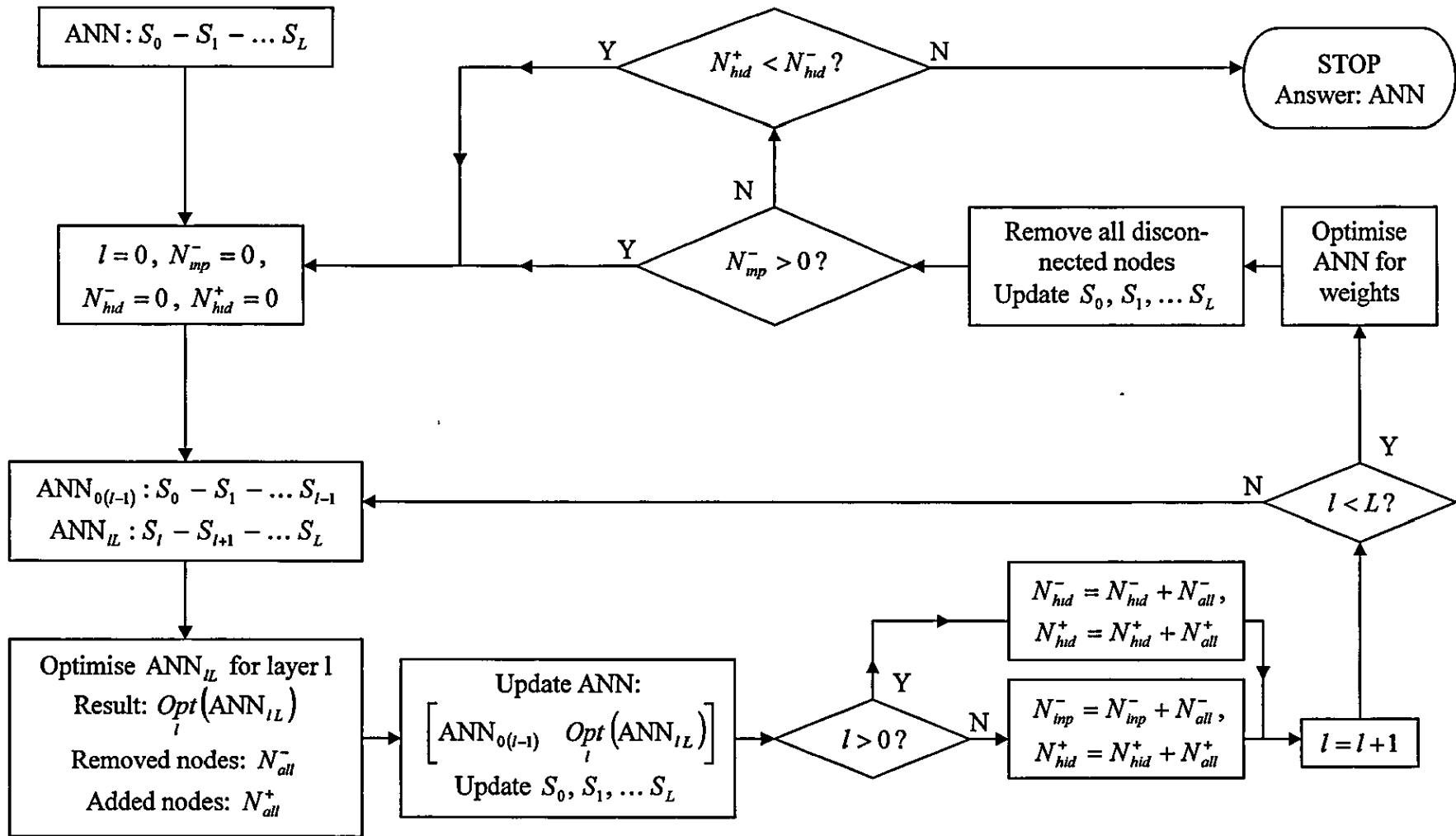


Figure 2-9 -- Flowchart of MFANN optimisation scheme

for $\tau = 0.2$ and $T = 1$. The available input sources are:

$$x_i(t) = \frac{1}{2} \cos\left(\frac{2\pi i}{T} \cdot t\right) \quad (2-124)$$

$$x_{i+4}(t) = \frac{1}{2} \cos\left(0.21 \cdot \frac{2\pi i}{T} \cdot t\right) \quad (2-125)$$

$$x_{i+8}(t) = \frac{1}{2} \cos\left(1.21 \cdot \frac{2\pi i}{T} \cdot t\right) \quad (2-126)$$

for $1 \leq i \leq 4$. The MFANN model sought has to approximate the periodic signal by using efficiently the first four inputs only and rejecting the others as redundant.

Due to the apparent simplicity of the modelling problem, the preliminary MFANN model is designed without hidden layers, i.e. as a 12-1 network, and its single output node is given a linear transfer function. Both input and output signals to the network are sampled at 100 uniformly spaced time instants that span a time interval defined by the period T of the square-wave pulse train signal. However, the samples taken are *not* pre-processed because they are guaranteed to fall within the interval $[-0.5, 0.5]$ by definitions (2-123) to (2-126). Nevertheless, these samples are combined in input-output pairs to form the training vectors of the network model. The Levenberg-Marquardt backpropagation (LMBP) algorithm is chosen for network training, and the sum-squared error (SSE) function defined in (2-28) is selected as the performance index of the training algorithm. The LMBP parameters used are: $SSE_{goal} = 2$, $E_{pmax} = 100$, $\nabla_{min} = 10^{-16}$, $\mu_0 = 10^{-6}$, $\mu_{max} = 10^6$, and $\theta = 10$. Finally, the original SOSA algorithm is applied to the trained network in order to remove all redundant input nodes and return the optimised Fourier series generator MFANN.

The optimised network obtained by the aforementioned design procedure is a 4-1 MFANN. That structure meets the first optimality criterion, i.e., it accepts input only from the sources described as sinusoidal functions of frequencies being multiples of $1/T$, the frequency of the square-wave pulse train. Moreover, the weights and biases of the network are found to match the first five coefficients of the discrete Fourier series corresponding to (2-123) and proven in theory to be:

$$a_0 = \frac{\tau}{T} - 0.5 \quad (2-127)$$

$$a_k = 2 \cdot \frac{\sin\left(k \cdot \pi \cdot \frac{\tau}{T}\right)}{k \cdot \pi}, \quad k \geq 1 \quad (2-128)$$

Table 2-5 – Fourier series coefficients vs. optimised MFANN parameters

Fourier Term	Fourier Coefficient	Optimised MFANN Weight	Relative Error %
0	-0.3000	-0.3000	0.00
1	0.3742	2*0.3741	0.03
2	0.3027	2*0.3023	0.13
3	0.2018	2*0.2012	0.30
4	0.0935	2*0.0931	0.53

Table 2-6 – Optimisation results for Fourier series generator MFANN

Optimisation Step	Training Epochs	Removed Inputs	$ \text{Error} _{\text{avg}}$	SSE	SSE % Change
1 (training only)	1	—	0.0889	1.8182	—
2	0	5, 7, 12	0.0799	1.9106	+5.08
3	0	6, 8, 11	0.0788	1.9468	+1.89
4 (1 backtrack)	1	9, 10	0.0796	1.9757	+1.48

Table 2-5 illustrates this point. Notice the gain of 2 next to MFANN weights that compensates for the factor 0.5 inherent to input source definitions (2-124) to (2-126). As a matter of fact, the final network meets the second optimality criterion, i.e., it has adequately generalised from the training vectors to universally approximate the pulse train signal by the first five terms of the discrete Fourier series associated with that signal.

The impact of node optimisation on the remarkable increase of accuracy in square-wave pulse train function's approximation obtained by the MFANN model is manifested by side-to-side comparison of figure 2-10 and figure 2-11. The first figure shows the function approximation obtained by the trained network model before optimisation against the ideal system response, while the second figure displays the same function pair for the optimised network.

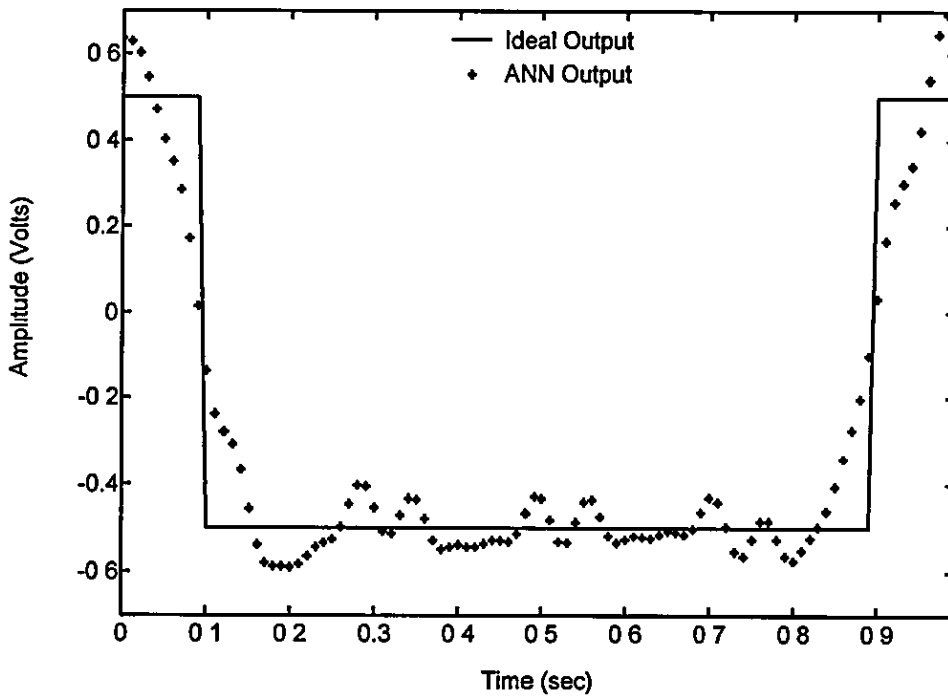


Figure 2-10 – Output of non-optimised Fourier series generator MFANN

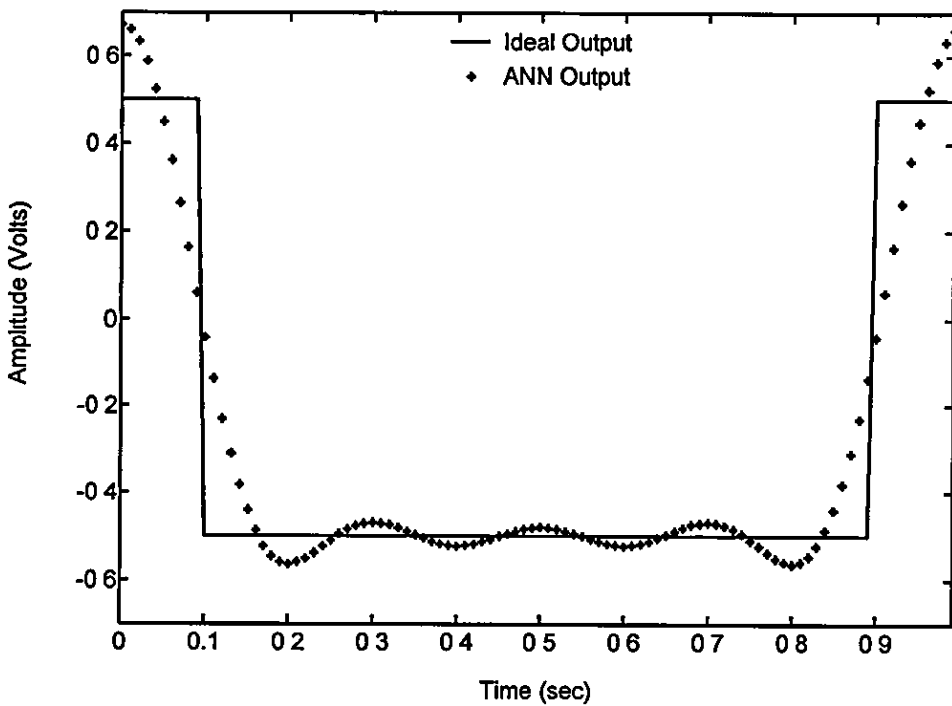


Figure 2-11 – Output of optimised Fourier series generator MFANN

Last but not least, the speed by which the original SOSA algorithm eliminates nodes from the initial trained MFANN is quickly realised by observation of table 2-6. As shown, SOSA eliminates three (the maximum allowed) redundant input nodes in all but the first and last steps, while the two remaining needless nodes are removed by the last iteration of the algorithm. Hence SOSA manages to achieve the best optimisation result for this modelling example at astonishingly little time and effort.

2.10.3 Multi-dimensional function generator

A *multi-dimensional function generator* approximates a well-defined function over a closed, compact domain from a limited number of input-output data pairs known to satisfy the relationship concerned. However, the inputs available to the generator usually come from a variety of sources whose degree of correlation with the mapping rule in question may vary arbitrarily. Therefore, a modelling procedure that makes optimal use of available inputs is essentially the only one that can deliver both optimal size and sufficient degree of accuracy for the function generator. This section demonstrates how a MFANN can be used as function generator in a typical situation and how the complete MFANN optimisation scheme discussed earlier can be applied in combination with efficient data pre-processing and network training techniques to produce excellent modelling solutions.

The function considered for approximation in this example is two-dimensional and described by the following expression:

$$\tilde{y}_1 = \frac{2}{3} \left\{ \sqrt{4|x_1 \circ x_2|} + \frac{1 + \sin[2\pi(x_1 + x_2)]}{3 + \cos[2\pi(x_1 - x_2)]} \cdot \exp\{\sin[2\pi(x_1^2 + x_2^2)] - 1\} \right\} - \frac{1}{2} \quad (2-129)$$

Each of the independent variables x_1, x_2 appearing in (2-129) is defined over the interval $[-0.5, 0.5]$. It can be shown that for that limited domain the dependent variable \tilde{y}_1 is bounded within the same interval. Therefore none of these parameters require pre-processing before they are considered eligible to participate in the associated, MFANN-implemented function generator.

The independent variables of (2-129) are not supposed to be the only input sources available for the generator. Instead, the variables are mixed with ten more input sources that involve x_1, x_2 in the following ways:

$$x_{i+2} = x_i + u \quad (2-130)$$

$$x_{i+4} = x_i \cdot u \quad (2-131)$$

$$x_{i+6} = x_i + g \quad (2-132)$$

$$x_{i+8} = x_i \cdot g \quad (2-133)$$

$$x_{i1} = u - 0.5 \quad (2-134)$$

$$x_{i2} = g \quad (2-135)$$

where $1 \leq i \leq 2$. Variables u , g represent *uniform* and *Gaussian noise sources* respectively. Specifically, source u returns uniformly distributed random numbers that fall within interval $[0,1]$, i.e. $u \equiv U(0,1)$, while source g returns normally distributed random numbers whose statistical properties indicate zero mean and unit standard deviation, i.e., $g \equiv N(0,1)$. Therefore input sources (2-130) to (2-133) can be regarded as severely corrupted duplicates of the independent variables x_1 , x_2 , while sources (2-134) and (2-135) are totally uncorrelated with the dependent variable \tilde{y}_1 . The question arisen at this point is whether the design procedure of a MFANN generating (2-129) would be able to detect the "obviousness" of input redundancy without any external intervention and pick the absolutely necessary sources from the lot. This question will be answered together with the one considering the final model's accuracy in the next few paragraphs.

An initial 12-18-1 MFANN is constructed to generate the function (2-129) from all available inputs. The log-sigmoid transfer function is assigned to all hidden neurons, while the linear transfer function is selected for the single output neuron. Both input and output data used in the network design procedure are generated by uniformly sampling the independent variables x_1 , x_2 at 20×20 points over their domains and passing the samples obtained to functions (2-129) to (2-135). The outcomes of these input generators are uniformly normalised to the interval $[-0.5 \ 0.5]$, and then combined with the samples taken from x_1 , x_2 and paired with the associated output samples to form the network training and testing patterns. Half of the patterns obtained are used for network training purposes while the other half is employed in the measurement of the *degree of generalisation* exhibited by the trained network at the end of every ANN optimisation step. The Levenberg-Marquardt backpropagation (LMBP) algorithm is selected for the MFANN training task, and the mean-squared error (MSE) function defined in (2-10) is chosen as the performance index of the training algorithm. The LMBP parameters used are: $MSE_{goal} = 0.01\%$, $Ep_{max} = 350$, $\nabla_{min} = 10^{-4}$, $\mu_0 = 10^{-2}$, $\mu_{max} = 10^{10}$, and $\theta = 1.25$. Finally, the complete MFANN opti-

misation scheme described in §2.9.4 and involving both the modified SOSA and OBS algorithms is followed throughout the model design process to ensure accuracy and efficiency for the function generator network obtained in the end.

Table 2-7 – MFANN training and optimisation results (Case 1)

Optimisation Step	Training Epochs	Removed Nodes	Training RMSE %	Testing RMSE %	Test. RMSE % Change
1 (training only)	17	—	0.97	17.83	—
2	22	7 ⁱ , 8 ⁱ , 12 ⁱ	0.99	17.40	-0.43
3	56	4 ⁱ , 10 ⁱ , 11 ⁱ	0.97	10.11	-7.29
4	243	5 ⁱ , 6 ⁱ , 9 ⁱ	0.99	2.45	-7.66
5	184	3 ⁱ	1.00	2.10	-0.35

Table 2-8 – MFANN training and optimisation results (Case 2)

Optimisation Step	Training Epochs	Removed Nodes	Training RMSE %	Testing RMSE %	Test. RMSE % Change
1 (training only)	17	—	0.97	17.83	—
2	22	7 ⁱ , 8 ⁱ , 12 ⁱ	0.99	17.40	-0.43
3	56	4 ⁱ , 10 ⁱ , 11 ⁱ	0.97	10.11	-7.29
4	243	5 ⁱ , 6 ⁱ , 9 ⁱ	0.99	2.45	-7.66
5	184	3 ⁱ	1.00	2.10	-0.35
6 (1 backtrack)	136	added 19 ^h	0.98	2.11	+0.01
7 (training only)	0	—	0.98	2.11	+0.00
8	0	5 ^h	0.98	2.12	+0.01
9 (1 backtrack)	30	4 ^h , 16 ^h	1.00	2.17	+0.05
10 (2 backtracks)	306	18 ^h	1.00	2.52	+0.35
11 (OBS)	—	w. 2 ⁱ – 19 ^h	1.00	2.52	+0.00

The first modelling effort was designed to exclude the option of adding redundant nodes to hidden layers ($N_{\max}^+ = 0$) and yielded the results summarised in table 2-7. RMSE is the square root of the MSE function and superscripts *i*, *h* stand for the layer the corresponding node/weight/bias belong or used to belong to, namely input or hidden layer. The function generator network created in this case accepts input from the undisturbed independent variable sources only and provides excellent generalisation results. However, the network design procedure failed to remove any amount of redundancy possibly existing in either the set of hidden layer nodes or the set of weights and biases.

The second modelling effort relaxed the restriction on node redundancy addition by allowing up to one node to be added to the hidden layer per input layer optimisation attempt ($N_{\max}^+ = 1$). Although this optimisation strategy obviously failed to prune any more inputs from the network than the one followed in the first optimisation case, it undoubtedly succeeded in offering hidden layer and weight/bias pruning gains. The results obtained in every optimisation step for this second function generator design case are summarised in table 2-8. RMSE stands for the root mean-squared error function and superscripts *i*, *h* are used to denote layer names as in the first modelling situation. The final MFANN accepts the undisturbed independent variables x_1 , x_2 as the only inputs, has 15 instead of 18 hidden layers, and has lost one redundant weight. The only drawback is a slight but rather insignificant degradation of network's generalisation capability from the level obtained by the first function generator modelling case.

No matter how attractive the controlled addition or redundancy during optimisation may seem, the reality is that such a strategy is controversial. Certain optimisation problems may benefit from its application, others may suffer considerable losses. Much depends on the problem's nature and the values chosen for the remaining network training and optimisation parameters. Therefore it is entirely upon the model designer to decide whether to employ the strategy and under what conditions to do so.

2.11 Summary

The purpose of this chapter was to provide the reader with essential knowledge about Artificial Neural Networks (ANNs) and their use as generic modelling tools to a multitude of problems. From the biological neuron to the complex artificial

neuron structure, the principle of operation is the formation of an output signal with certain characteristics in response to a set of inputs. Attempts to shape the output obtained by ANNs in order to meet problem specifications resulted in the generation of a range of network structures and associated training algorithms. Among them, the kind of Multilayer Feedforward Artificial Neural Networks (MFANNs) was distinguished for its inherent ability to model function approximation and classification problems. The most efficient training algorithm for small- to middle-sized MFANNs was proved to be the Levenberg-Marquardt Backpropagation (LMBP) algorithm. Data preconditioning techniques were developed to ease training with no or little loss of information stored in input-output data relationships defined by the problem concerned. The novel Second Order Sensitivity Analysis (SOSA) optimisation algorithm for MFANN input and hidden node optimisation was presented in detail and a new, complete MFANN optimisation scheme involving SOSA and Optimal Brain Surgeon (OBS) algorithms was carefully designed. Finally, the preferred ANN architecture, training algorithm, data preconditioning method, and optimisation scheme elements were exploited to form excellent modelling solutions for two carefully selected function approximation problems.

CHAPTER 3

INTRODUCTION TO MIE SCATTERING THEORY

*♪...sunshine on a rainy day
makes my soul trip away...♪*
(Emma Bunton, A girl like me, 2001)

3.1 Introduction

This chapter provides the theoretical background that is necessary for the comprehension of the *light scattering* problem and its *inverse*. The former problem deals with the description of visual phenomena, such as the blue colour of the sky at daytime or the observation of clouds, which are caused by the interaction of light with heterogeneities of its propagation medium. In fact, the term *scattering* strictly refers to the re-radiation of light incident to the heterogeneity, while the term *absorption* describes the (associated with scattering) effect of incident light energy transformation into other forms, and the term *extinction* expresses the combined process of light scattering and absorption. On the other hand, the inverse scattering problem investigates the properties of light scattered by one or many propagation medium heterogeneities (i.e., particles) in an attempt to describe the physical and optical properties of these heterogeneities. Both problems are hard to solve analytically, even in special cases, with the inverse scattering one being the hardest of them.

After a brief overview of the major research efforts and achievements made in the light scattering field throughout its history, the chapter provides the pieces of electromagnetic theory that are essential for the description of light and its properties. Next, the fundamental problem of light scattering by a single spherical particle placed in a simple non-absorbing medium, known as the Mie scattering problem, is solved analytically in its general case. The solution obtained is subsequently exploited in the analytic calculation of scattered light irradiance a typical polar nephelometric device would measure at an arbitrary scattering angle under certain conditions. The outcome of the previous calculation is used next to gather theoretical measurements of the scattered irradiance and turbidity derived from a tenuous suspension of Arizona Fine dust in water. Finally, the unfeasible conditions under which a unique inverse scattering solution could be found, the requirement for non-analytic modelling approaches and the reasonable expectation of only partial solutions to the inverse scattering problem are all explained in qualitative but meaningful terms.

3.2 Historical overview

The history of the light scattering field is an indistinguishable part of the history of optics. A detailed overview of the latter extended throughout the period of human history is available in a series of optical textbooks, e.g., [56]. However, the

two fundamental concepts the scattering theory relies upon, the description of light by waves and the explanation of light polarisation, were established in the 19th century. Fresnel explained light diffraction phenomena observed earlier by Young by combining the interference theory of the latter scientist with the secondary wave principle speculated in the 17th century by Huygens. Young gave a final explanation of light polarisation by suggesting that light propagates in a medium as a *transverse* wave. Fresnel exploited Young's idea to derive theoretically the irradiances of the polarised light components after reflection at any angle of incidence; his results matched the experimental data obtained earlier by Brewster. Later in the century, Maxwell linked together electric, magnetic and optical phenomena by his electromagnetic theory [162].

The second half of the 19th century was the era of the great mathematical physicists such as Poisson, Cauchy, Green, Kirchoff, Stokes, Rayleigh, Lamb and Lorenz. The last four scientists made personal contributions to the finding of an analytic solution to the problem of light scattering by a spherical particle in a homogeneous medium, although such a solution had arguably been submitted earlier by Clebsch [163]. Specifically, Clebsch solved the three Cartesian scalar wave equations that had been derived from the elastic wave equation (the one that scientists used to describe light as a wave before Maxwell's theory) a few years earlier by Stokes [164]. In the following decades, Rayleigh published an approximating solution to the light scattering problem under consideration by assuming small spheres [165]. Lamb solved the vector wave equation of light after accepting Maxwell's theory [166]. Lorenz simplified and refined Clebsch's solution by excluding the generality of longitudinal waves and assuming a spherical polar instead of Cartesian coordinate system [167]. Finally, Mie published in 1908 a full solution to the scattering problem concerned in a paper that explains the varied colours exhibited by small colloidal particles of gold suspended in water [25], and for that reason he was honoured by naming the scattering-by-a-sphere problem after his name.

Debye was the last scientist to study complex scattering problems before the interest in the field was apparently lost for many years. The reason was the popularity of quantum mechanics pioneered by Planck in 1900 that made many great mathematical physicists devote their time to advance the new field. However, the light scattering field regained some interest in the recent years due to advancements in associated research areas such as astronomy, chemistry and quantum mechanics. Furthermore, the

advancement of digital computers made possible for applied scientists and engineers to obtain formulas and numerical results for a variety of light scattering applications like the one being the subject of this study. Van de Hulst and Bohren are two researchers among the many who work actively in scattering field nowadays.

3.3 Electromagnetic theory elements

Light can be described as either a stream of *photons* ("particles" of energy) or an electromagnetic *wave* (self-sustaining disturbance of an electric and an induced magnetic field moving together in space) travelling at constant speed in a *homogeneous* medium or the vacuum. The former hypothesis allows light properties to be explained by the quantum mechanics theory while the latter assumption calls for the relatively older electromagnetic field theory to give answers to light phenomena. Either modelling approach leads to the same result for a given light-related problem, yet one of them is usually much easier to develop and explain than the other. This study treats light scattering as electromagnetic waves interacting with matter in order to be consistent with the majority of related publications and maintain theoretical complexity at reasonable levels.

An electromagnetic field is macroscopically described at any point by the following *Maxwell equations* (in SI units):

$$\nabla \cdot \mathbf{D} = \rho_F \quad (3-1)$$

$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = \mathbf{0} \quad (3-2)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (3-3)$$

$$\nabla \times \mathbf{H} = \mathbf{J}_F + \frac{\partial \mathbf{D}}{\partial t} \quad (3-4)$$

where \mathbf{D} is the *electric displacement*, ρ_F the *free charge density*, \mathbf{E} the *electric field*, \mathbf{B} the *magnetic induction*, \mathbf{H} the *magnetic field* and \mathbf{J}_F the *current density of free moving charges*. Moreover, variables \mathbf{D} and \mathbf{H} are defined by:

$$\mathbf{D} \equiv \epsilon_0 \mathbf{E} + \mathbf{P} \quad (3-5)$$

$$\mathbf{H} \equiv \frac{\mathbf{B}}{\mu_0} - \mathbf{M} \quad (3-6)$$

where \mathbf{P} is the *electric polarisation* (average electric dipole moment per unit volume), ϵ_0 the *permittivity of free space*, \mathbf{M} the *magnetisation* (average magnetic dipole per unit volume), and μ_0 the *permeability of free space*. An implicit assumption made for

(3-5) to be valid is that quadrupole and higher moments are negligible compared with the dipole moment. Both polarisation and magnetisation vectors vanish in free space.

Not all electromagnetic field vectors that appear in equations (3-1) to (3-6) are independent, as \mathbf{J}_F , \mathbf{M} and \mathbf{P} are actually provided by the following *constitutive relations* determined by the propagation medium:

$$\mathbf{J}_F = f_J(\mathbf{E}) \quad (3-7)$$

$$\mathbf{M} = f_M(\mathbf{H}) \quad (3-8)$$

$$\mathbf{P} = f_P(\mathbf{E}) \quad (3-9)$$

When functional relationships $f_J()$, $f_M()$ and $f_P()$ are independent of the electromagnetic field vectors (the medium is *linear*), independent of position (the medium is *homogeneous*), and independent of direction (the field is *isotropic*), the propagation medium is said to be *simple*. In that case the constitutive relations (3-7) to (3-9) can be simplified to take the form:

$$\mathbf{J}_F = \sigma \mathbf{E} \quad (3-10)$$

$$\mathbf{M} = \chi_m \mathbf{H} \quad (3-11)$$

$$\mathbf{P} = \epsilon_0 \chi_e \mathbf{E} \quad (3-12)$$

The phenomenological coefficients σ , χ_m and χ_e are the *conductivity*, the *magnetic susceptibility* and the *electric susceptibility* of the medium under consideration. All wave propagation media regarded in this study are simple unless explicitly mentioned otherwise.

When free charges are absent in the region of wave propagation ($\rho_F = 0$), the electric and magnetic fields of an electromagnetic wave are divergence-free:

$$\nabla \cdot \mathbf{E} = \nabla \cdot \mathbf{H} = 0 \quad (3-13)$$

and satisfy the following wave (Helmholtz) equations:

$$\nabla^2 \mathbf{E} = \sigma \mu_0 (1 + \chi_m) \frac{\partial \mathbf{E}}{\partial t} + \epsilon_0 \mu_0 (1 + \chi_e)(1 + \chi_m) \frac{\partial^2 \mathbf{E}}{\partial t^2} \quad (3-14)$$

$$\nabla^2 \mathbf{H} = \sigma \mu_0 (1 + \chi_m) \frac{\partial \mathbf{H}}{\partial t} + \epsilon_0 \mu_0 (1 + \chi_e)(1 + \chi_m) \frac{\partial^2 \mathbf{H}}{\partial t^2} \quad (3-15)$$

A field vector \mathbf{F} is said to be *time-harmonic* when it has the form:

$$\mathbf{F} = \mathbf{x} \cos(\omega t) + \mathbf{y} \sin(\omega t) \quad (3-16)$$

Vectors \mathbf{x} , \mathbf{y} in (3-16) are real and independent of time (*not* position), while scalar ω is the *angular frequency*. The field vector \mathbf{F} is related to the (complex) phasor vector $\tilde{\mathbf{F}}$:

$$\tilde{\mathbf{F}} = (\mathbf{x} + iy) \exp(-i\omega t) \quad (3-17)$$

in terms of:

$$\mathbf{F} = \text{Re}(\tilde{\mathbf{F}}) \quad (3-18)$$

Phasors can substitute time-harmonic fields in vector analysis if linear operators $\mathbf{L}()$ are the only ones applied on these fields. In such cases, the real part of the last phasor vector obtained is always equal to the result that would have been obtained by the same vector analysis procedure if phasors had not been considered. This happens because:

$$\mathbf{L}(\mathbf{F}) = \mathbf{L}[\text{Re}(\tilde{\mathbf{F}})] = \text{Re}[\mathbf{L}(\tilde{\mathbf{F}})] \quad (3-19)$$

Phasor vectors are more convenient to work with and therefore will be used extensively in this work for the representation of time-harmonic fields. All phasor variables will be denoted by the circumflex symbol (\sim).

A time-harmonic electromagnetic field is macroscopically described at any point of a simple medium by the following phasor Maxwell equations:

$$\nabla \cdot \tilde{\mathbf{E}} = \frac{\tilde{\rho}_F}{\varepsilon_0(1 + \chi_e)} \quad (3-20)$$

$$\nabla \times \tilde{\mathbf{E}} = i\omega\mu \tilde{\mathbf{H}} \quad (3-21)$$

$$\nabla \cdot \tilde{\mathbf{H}} = 0 \quad (3-22)$$

$$\nabla \times \tilde{\mathbf{H}} = -i\omega\varepsilon \tilde{\mathbf{E}} \quad (3-23)$$

where $\tilde{\rho}_F$ defined by:

$$\tilde{\rho}_F \equiv \rho_F \exp(-i\omega t) \quad (3-24)$$

is the *free charge density phasor*, ε given by:

$$\varepsilon \equiv \varepsilon_0(1 + \chi_e) + i\frac{\sigma}{\omega} \quad (3-25)$$

is the (*complex*) *permittivity of the medium*, and μ determined by:

$$\mu \equiv \mu_0(1 + \chi_m) \quad (3-26)$$

is the *permeability of the medium*.

When free charges are absent in the region of wave propagation ($\tilde{\rho}_F = 0$), the fields of a time-harmonic, electromagnetic wave are divergence-free:

$$\nabla \cdot \tilde{\mathbf{E}} = \nabla \cdot \tilde{\mathbf{H}} = 0 \quad (3-27)$$

and satisfy the wave equations:

$$\nabla^2 \tilde{\mathbf{E}} + \omega^2 \epsilon \mu \tilde{\mathbf{E}} = \mathbf{0} \quad (3-28)$$

$$\nabla^2 \tilde{\mathbf{H}} + \omega^2 \epsilon \mu \tilde{\mathbf{H}} = \mathbf{0} \quad (3-29)$$

A *plane electromagnetic wave* is characterised by its time-harmonic field vectors \mathbf{E} , \mathbf{H} associated with phasors $\tilde{\mathbf{E}}$, $\tilde{\mathbf{H}}$ of the form:

$$\tilde{\mathbf{E}} = \mathbf{E}_0 \exp[i(\mathbf{k} \cdot \mathbf{x} - \omega t)] \quad (3-30)$$

$$\tilde{\mathbf{H}} = \mathbf{H}_0 \exp[i(\mathbf{k} \cdot \mathbf{x} - \omega t)] \quad (3-31)$$

where \mathbf{E}_0 , \mathbf{H}_0 are *constant* vectors and \mathbf{k} is the (*complex*) *wave vector*. The latter vector may be written as:

$$\mathbf{k} \equiv \mathbf{k}' + i\mathbf{k}'' \quad (3-32)$$

with \mathbf{k}' and \mathbf{k}'' being real vectors. The amplitudes of the electric and magnetic fields are $\mathbf{E}_0 \exp(-\mathbf{k}'' \cdot \mathbf{x})$ and $\mathbf{H}_0 \exp(-\mathbf{k}'' \cdot \mathbf{x})$ respectively, and the common phase of these fields is $\mathbf{k}' \cdot \mathbf{x} - \omega t$. Vector \mathbf{k}' is perpendicular to the surfaces of constant phase, and \mathbf{k}'' is perpendicular to the surfaces of constant amplitude. The velocity of propagation of surfaces of constant phase is called *phase velocity* v and is given by:

$$v = \frac{\omega}{k'} \quad (3-33)$$

If vectors \mathbf{k}' and \mathbf{k}'' in (3-32) are parallel, or $\mathbf{k}'' = \mathbf{0}$, the surfaces of constant amplitude and phase coincide and the plane wave is said to be *homogeneous*. In that case the wave vector \mathbf{k} is given by:

$$\mathbf{k} = (k' + k'')\hat{\mathbf{e}} = k\hat{\mathbf{e}} \quad (3-34)$$

where k' and k'' are nonnegative, $\hat{\mathbf{e}}$ is a real unit vector in the direction of wave propagation and k is the nonnegative *wave number*. Planar waves propagating in vacuum are homogeneous.

The necessary and sufficient conditions for a plane wave defined by (3-30), (3-31) to be compatible with Maxwell equations (3-20) to (3-23) when it propagates through a medium without free charges ($\tilde{\rho}_F = 0$) are:

$$\mathbf{k} \cdot \mathbf{E}_0 = \mathbf{k} \cdot \mathbf{H}_0 = \mathbf{E}_0 \cdot \mathbf{H}_0 = 0 \quad (3-35)$$

$$\mathbf{k} \times \mathbf{E}_0 = \omega \mu \mathbf{H}_0 \quad (3-36)$$

$$\mathbf{k} \times \mathbf{H}_0 = -\omega \epsilon \mathbf{E}_0 \quad (3-37)$$

$$\mathbf{k} \cdot \mathbf{k} = \omega^2 \epsilon \mu \quad (3-38)$$

The wave number of a homogeneous plane wave can thus be found by substitution of equation (3-34) to (3-38), and the following result is obtained:

$$k = \omega \sqrt{\epsilon \mu} \quad (3-39)$$

If the *complex refractive index* N of the propagation medium is defined by:

$$N \equiv \sqrt{\frac{\epsilon \mu}{\epsilon_0 \mu_0}} = c \sqrt{\epsilon \mu} = n + i\kappa \quad (3-40)$$

where n and κ are the nonnegative *optical constants* and c is the *speed of light in vacuum* ($c = 3 \cdot 10^8$ m/s), equation (3-39) requires that:

$$N = \frac{kc}{\omega} = \frac{k\lambda}{2\pi} \quad (3-41)$$

holds for plane homogeneous waves. Variable λ is called the *wavelength in vacuum* and is defined by:

$$\lambda \equiv \frac{2\pi c}{\omega} \quad (3-42)$$

If $z \equiv \hat{e} \cdot \mathbf{x}$ is the projection of position vector \mathbf{x} in the direction of wave propagation, the expressions of field phasors defining a plane homogeneous wave can be written as follows:

$$\tilde{\mathbf{E}} = \mathbf{E}_0 \exp[i(kz - \omega t)] = \mathbf{E}_0 \exp\left(-\frac{2\pi\kappa z}{\lambda}\right) \exp\left[i\left(\frac{2\pi n z}{\lambda} - \omega t\right)\right] \quad (3-43)$$

$$\tilde{\mathbf{H}} = \mathbf{H}_0 \exp[i(kz - \omega t)] = \mathbf{H}_0 \exp\left(-\frac{2\pi\kappa z}{\lambda}\right) \exp\left[i\left(\frac{2\pi n z}{\lambda} - \omega t\right)\right] \quad (3-44)$$

Thus, the imaginary part of the complex refractive index N determines the attenuation of the wave as it propagates through the medium, while the real part determines the phase velocity $v = c/n$. All plane waves considered in the remaining of this study are homogeneous unless otherwise stated.

The *instantaneous Poynting vector* $\mathbf{S}(t)$ of an arbitrary electromagnetic wave is defined by:

$$\mathbf{S}(t) \equiv \mathbf{E}(t) \times \mathbf{H}(t) \quad (3-45)$$

and specifies the magnitude and direction of the rate of transfer of electromagnetic energy at all points of space and all times. As a matter of fact, the net rate at which electromagnetic energy is transferred into a volume V enclosed by a closed surface A is given by:

$$W(t) = -\oint_A \mathbf{S}(t) \cdot \hat{\mathbf{n}} dA \quad (3-46)$$

where $\hat{\mathbf{n}}$ is the unit vector normal to the element dA of the closed surface and pointing outwards the surface.

Unfortunately, the instantaneous Poynting vector is a rapidly changing function of time when light as an electromagnetic wave is considered. Most instruments are not capable of tracking the instantaneous Poynting vector, but respond to the *time-averaged Poynting vector* \mathbf{S} defined by:

$$\mathbf{S} \equiv \frac{1}{\tau} \int_t^{t+\tau} \mathbf{S}(t') dt' \quad (3-47)$$

where τ is a time interval dependent on the instrument and long compared with the time it takes for $\mathbf{S}(t)$ to change. The time-averaged Poynting vector for time-harmonic fields is given by:

$$\mathbf{S} = \frac{1}{2} \text{Re}(\tilde{\mathbf{E}} \times \tilde{\mathbf{H}}^*) \quad (3-48)$$

When plane waves are considered in particular, the time-averaged Poynting vector takes the form:

$$\mathbf{S} = \frac{1}{2} \text{Re} \left(\sqrt{\frac{\epsilon}{\mu}} |E_0|^2 \exp\left(-\frac{4\pi\kappa z}{\lambda}\right) \hat{\mathbf{e}} \right) \quad (3-49)$$

The *irradiance* (or *intensity*) I of an electromagnetic wave is a scalar quantity defined by:

$$I \equiv |\mathbf{S}| \quad (3-50)$$

As it emerges from equation (3-49), the irradiance of a plane wave is exponentially attenuated in the sense:

$$I = I_0 \exp(-\alpha z) \quad (3-51)$$

where α is the *absorption coefficient* defined by:

$$\alpha \equiv \frac{4\pi\kappa}{\lambda} \quad (3-52)$$

and I_0 is the irradiance at $z=0$. However, attenuation formula (3-51) excludes attenuation phenomena attributed to absorption and scattering due to impurities of the propagation medium.

The electric field of a plane wave in a particular plane, e.g. $z=0$ for convenience, is derived from equation (3-43) and given by:

$$\mathbf{E}(z=0) = \text{Re}[\tilde{\mathbf{E}}(z=0)] = \text{Re}(\mathbf{E}_0)\cos(\omega t) + \text{Im}(\mathbf{E}_0)\sin(\omega t) \quad (3-53)$$

A similar expression is derived for the magnetic field by equation (3-44):

$$\mathbf{H}(z=0) = \text{Re}[\tilde{\mathbf{H}}(z=0)] = \text{Re}(\mathbf{H}_0)\cos(\omega t) + \text{Im}(\mathbf{H}_0)\sin(\omega t) \quad (3-54)$$

As vectors \mathbf{E}_0 , \mathbf{H}_0 are constant in both time and space by definition, so do their real and imaginary parts. Therefore equations (3-53) and (3-54) describe two vibration ellipses. If $\text{Re}(\mathbf{E}_0) = 0$ or $\text{Im}(\mathbf{E}_0) = 0$, the vibration ellipse of the electric field is just a straight line, and the plane wave is said to be *linearly polarised*. If $|\text{Re}(\mathbf{E}_0)| = |\text{Im}(\mathbf{E}_0)|$ and $\text{Re}(\mathbf{E}_0) \perp \text{Im}(\mathbf{E}_0) = 0$, the vibration ellipse of \mathbf{E} is a circle, and the wave is called *circularly polarised*. In general, a plane wave is *elliptically polarised*.

The vibration ellipse of the electric field of an elliptically polarised plane wave is shown in figure 3-1. That wave is uniquely specified by a set of four *ellipsometric parameters*, namely *handedness* (right-handed if the ellipse is traced out clockwise as viewed by an observer who is looking toward the source of wave or left-handed otherwise), *ellipticity* b/a , *azimuth* γ ($0 \leq \gamma \leq \pi$) and irradiance I . However, these parameters (with the exception of irradiance) are difficult to be measured directly. Moreover, the ellipsometric parameters of two or more or two plane waves interfering incoherently (i.e., without fixed relationship among phases) cannot be added together to give the parameters of the superimposed plane wave. Finally the ellipsometric parameters cannot be adapted to describe partially polarised light that is formally defined later on.

An alternative way to uniquely describe a plane wave is by the four *Stokes parameters* I , Q , U , V . These parameters are related to the ellipsometric ones by the expressions:

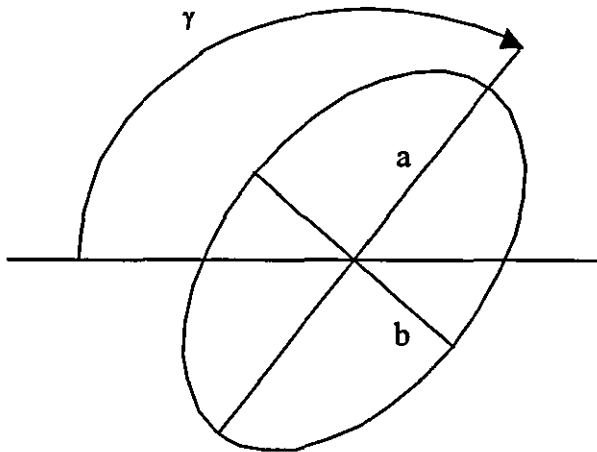


Figure 3-1 – Vibration ellipse

$$I = a^2 + b^2 \quad (3-55)$$

$$Q = (a^2 + b^2) \cos(2\eta) \cos(2\gamma) \quad (3-56)$$

$$U = (a^2 + b^2) \cos(2\eta) \sin(2\gamma) \quad (3-57)$$

$$V = (a^2 + b^2) \sin(2\eta) \quad (3-58)$$

where:

$$|\tan \eta| = \frac{b}{a} \quad (3-59)$$

for $-\pi/4 \leq \eta \leq \pi/4$ and the sign of V determines the handedness of the vibration ellipse (positive denotes right-handed and negative denotes left-handed). The Stokes parameters of a monochromatic (i.e. time-harmonic) plane wave are not all independent, as:

$$I^2 = Q^2 + U^2 + V^2 \quad (3-60)$$

An assumption made in definitions (3-30), (3-31) of a monochromatic plane wave was that vectors \mathbf{E}_0 and \mathbf{H}_0 are constant. If that assumption is relaxed and the aforementioned vectors are allowed to vary slowly over time intervals of the order of $2\pi/\omega$, the wider class of *quasi-monochromatic plane waves* is derived. The field vectors of these waves are given by the following expressions:

$$\tilde{\mathbf{E}} = \mathbf{E}_0(t) \exp[i(kz - \omega t)] \quad (3-61)$$

$$\tilde{\mathbf{H}} = \mathbf{H}_0(t) \exp[i(kz - \omega t)] \quad (3-62)$$

where:

$$\mathbf{E}_0(t) = E_{\parallel}(t) \cdot \hat{\mathbf{e}}_{\parallel}^e + E_{\perp}(t) \cdot \hat{\mathbf{e}}_{\perp}^e \quad (3-63)$$

$$\mathbf{H}_0(t) = H_{\parallel}(t) \cdot \hat{\mathbf{e}}_{\parallel}^h + H_{\perp}(t) \cdot \hat{\mathbf{e}}_{\perp}^h \quad (3-64)$$

and $(\hat{\mathbf{e}}_{\parallel}^e, \hat{\mathbf{e}}_{\perp}^e)$ and $(\hat{\mathbf{e}}_{\parallel}^h, \hat{\mathbf{e}}_{\perp}^h)$ are two arbitrarily selected pairs of unit vectors that define two orthogonal coordinate systems, each in the plane of vibration of $\tilde{\mathbf{E}}$ and $\tilde{\mathbf{H}}$ respectively.

The Stokes parameters of a quasi-monochromatic plane wave are defined by:

$$I \equiv \langle E_{\parallel} E_{\parallel}^* + E_{\perp} E_{\perp}^* \rangle \quad (3-65)$$

$$Q \equiv \langle E_{\parallel} E_{\parallel}^* - E_{\perp} E_{\perp}^* \rangle \quad (3-66)$$

$$U \equiv \langle E_{\parallel} E_{\perp}^* + E_{\perp} E_{\parallel}^* \rangle \quad (3-67)$$

$$V \equiv i \langle E_{\parallel} E_{\perp}^* - E_{\perp} E_{\parallel}^* \rangle \quad (3-68)$$

where the angular brackets indicate time averages over an interval long compared with the period $T=2\pi/\omega$ of the wave. These definitions can be applied for strict monochromatic waves (with the exception of time-averaging) and their results are in accordance with definitions (3-55) to (3-58).

The Stokes parameters of a (quasi)-monochromatic plane wave satisfy the following inequality:

$$I^2 \geq Q^2 + U^2 + V^2 \quad (3-69)$$

If the complex amplitudes $E_{\parallel}(\cdot)$ and $E_{\perp}(\cdot)$ are completely uncorrelated, the wave is said to be *unpolarised*; so-called natural light (e.g. light from the sun, incandescent and fluorescent lamps) is unpolarised. All Stokes parameters but irradiance of an unpolarised wave vanish by definition:

$$Q = U = V = 0 \quad (3-70)$$

If $E_{\parallel}(\cdot)$ and $E_{\perp}(\cdot)$ are completely correlated, the wave is called *polarised* and its Stokes parameters satisfy equality (3-60) like a strictly monochromatic wave. The last possibility is for $E_{\parallel}(\cdot)$ and $E_{\perp}(\cdot)$ to be partially correlated. In that case, the wave is *partially polarised* and its Stokes parameters satisfy inequality (3-69) in the strict sense.

The Stokes parameters can be grouped together to form the *Stokes vector* \mathbf{S} :

$$\mathbf{S} \equiv [I \quad Q \quad U \quad V]^T \quad (3-71)$$

The new vector can represent a plane wave of arbitrary polarisation, including partially polarised waves. When the state of polarisation of a plane wave is changed on interaction with impurities or changes of the propagation medium, the Stokes vector changes accordingly. A 4x4 matrix \mathbf{M} can always be found that relates the Stokes vectors before and after wave interaction, \mathbf{S}_x and \mathbf{S}_y respectively, in the following manner:

$$\mathbf{S}_y = \mathbf{M} \cdot \mathbf{S}_x \quad (3-72)$$

Matrix \mathbf{M} is called the *Muller matrix* and depends only on the properties of the matter interacting with the plane wave.

This completes the theoretical framework necessary for the comprehension of the theory of light scattering caused by either a single spherical particle (Mie scattering) or a small concentration of solid particles (single scattering) in simple matter. §3.4 and §3.5 provide thorough cover to these tough subjects.

3.4 Scattering by a single spherical particle

The problem of scattering by a spherical particle placed on a simple, non-absorbing medium without free charges is presented in figure 3-2. An arbitrarily polarised monochromatic light beam travelling at long distance from its source can be described as a plane wave ($\mathbf{E}_i, \mathbf{H}_i$). That wave is called *incident* when it reaches the surface of the particle concerned and consumes part of the energy it carries to oscillate electric charges (electrons and protons) being inside that particle. The excited particle develops an *internal* plane wave ($\mathbf{E}_1, \mathbf{H}_1$) and reradiates part of the energy it *absorbed* from the incident wave in all directions to form the *scattered* plane wave ($\mathbf{E}_s, \mathbf{H}_s$). The answer sought is the mathematical description of the internal and the scattered fields from the incident wave, particle size and electromagnetic properties of the particle and surrounding medium. The scattering problem explained above is usually called the *Mie scattering problem* and the theory that has to be developed to solve it is said to be the *Mie scattering theory*. Both names are given to honour Gustav Mie as the first scientist who published a complete solution for the scattering problem explained above [25].

Due to the spherical symmetry of the Mie scattering problem, it is much easier to express all related fields in *spherical polar coordinates* (r, θ, ϕ). The relationship between these coordinates and the ones of the Cartesian coordinate system (x, y, z) drawn in the centre of the spherical particle considered by the scattering problem is illustrated in figure 3-3. However, the conversion of a field expression from Cartesian to spherical polar coordinates is a complicated task that deserves special attention in the next few paragraphs.

The incident and scattered waves propagate through a simple, non-absorbing medium without free charges. The same assumption can be made for the internal wave when the scattering particle is viewed macroscopically. All waves therefore must satisfy the Maxwell equations (3-20) to (3-23) for $\tilde{\rho}_f = 0$ and vector wave equations (3-28), (3-29). However, the set of phasor vectors $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{N}}$, defined by:

$$\tilde{\mathbf{M}} \equiv \nabla \times (\mathbf{r}g) \quad (3-73)$$

$$\tilde{\mathbf{N}} \equiv \frac{\nabla \times \tilde{\mathbf{M}}}{k} \quad (3-74)$$

where k is the wave number of the incident wave, r the radius vector of the spherical polar coordinate system illustrated in figure 3-3, and $g()$ a scalar function that satisfies the scalar wave equation:

$$\nabla^2 g + k^2 g = 0 \quad (3-75)$$

satisfy all Maxwell and vector wave equations mentioned above. Thus, any physically realisable plane wave can be expanded in polar coordinates as an infinite series of the vector harmonics $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{N}}$ by the principle of superposition derived from the linearity of the Maxwell and vector wave equations.

Exact solutions for the vector harmonics cannot be given unless (3-75) is solved for the *generating function* $g()$ first. That scalar wave equation can be written in spherical coordinates as follows:

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial g}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial g}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 g}{\partial \phi^2} + k^2 g = 0 \quad (3-76)$$

The partial differential equation (3-76) can be solved by the variable separation method. If the generating function $g()$ depends on the single-variable functions $R()$, $\Theta()$ and $\Phi()$ in the following manner:

$$g(r, \theta, \phi) = R(r)\Theta(\theta)\Phi(\phi) \quad (3-77)$$

then (3-76) splits into three ordinary differential equations:

$$\frac{d^2 \Phi}{d\phi^2} + m^2 \Phi = 0 \quad (3-78)$$

$$\frac{1}{\sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{d\Theta}{d\theta} \right) + \left[n(n+1) - \frac{m^2}{\sin^2 \theta} \right] \Theta = 0 \quad (3-79)$$

$$\frac{d}{dr} \left(r^2 \frac{dR}{dr} \right) + [k^2 r^2 - n(n+1)] R = 0 \quad (3-80)$$

where the separation constants m and n are determined by subsidiary conditions that $g()$ must satisfy.

The linearly independent solutions of (3-78) are:

$$\Phi_{em} = \cos(m\phi) \quad (3-81)$$

$$\Phi_{om} = \sin(m\phi) \quad (3-82)$$

where subscripts e and o denote even and odd. Due to the fact that $g()$ must be a single-valued function of the azimuthal angle ϕ , i.e.:

$$\lim_{\nu \rightarrow 2\pi} g(\phi + \nu) = g(\phi) \quad (3-83)$$

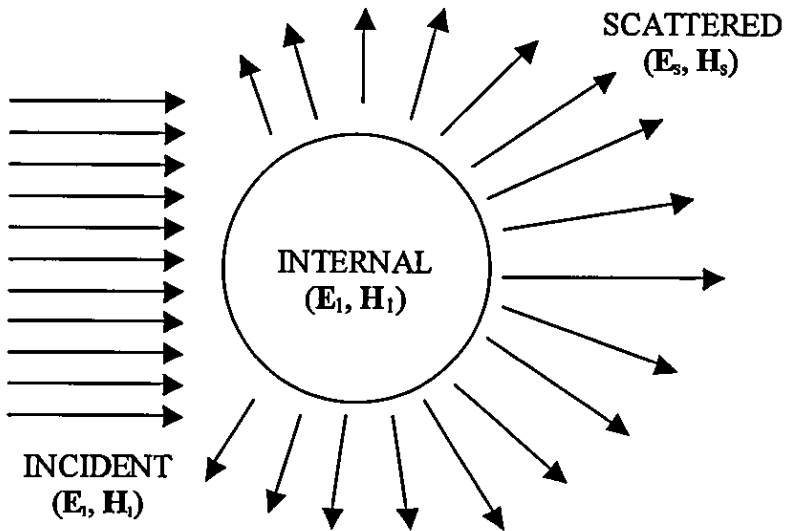


Figure 3-2 – Incident, internal and scattered fields

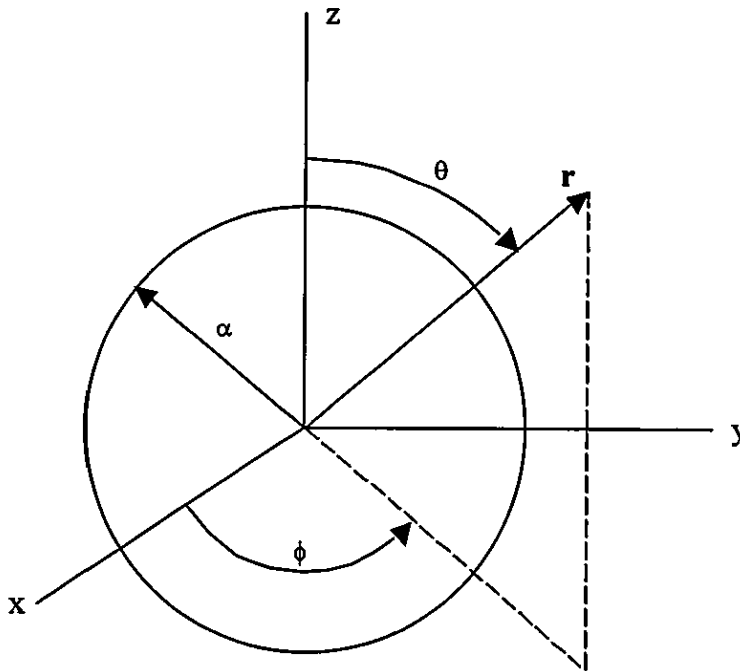


Figure 3-3 – Spherical polar coordinate system centred on a spherical particle

the separation constant m has to be an integer. Positive values of m are sufficient to generate all the linearly independent solutions to (3-78).

The solutions to (3-79) that are finite at $\theta=0$ and $\theta=\pi$ are the associated *Legendre functions* of the first kind, degree n and order m (see [168], pp. 326-327):

$$\Theta_{mn} = P_n^m(\cos\theta) \quad (3-84)$$

where $n=m, m+1, \dots$. When $m=0$ the associated Legendre functions degenerate to *Legendre polynomials* and they are denoted by $P_n(\cdot)$.

The linearly independent solutions to (3-80) are the *spherical Bessel functions* of order n and first kind, $j_n()$, second kind, $y_n()$, or third kind, $h_n()$:

$$R_n = z_n(kr) \quad (3-85)$$

where $z_n()$ stands for either of functions $j_n()$, $y_n()$ and $h_n()$. The *spherical Bessel functions of first and second kind* are defined with respect to the ordinary Bessel functions of first kind, $J_\nu()$, and second kind, $Y_\nu()$, in the following way:

$$j_n(kr) \equiv \sqrt{\frac{\pi}{2kr}} J_{n+1/2}(kr) \quad (3-86)$$

$$y_n(kr) \equiv \sqrt{\frac{\pi}{2kr}} Y_{n+1/2}(kr) \quad (3-87)$$

while $h_n()$, also known as the *spherical Hankel function*, is defined in terms of $j_n()$ and $y_n()$ by:

$$h_n(kr) \equiv j_n(kr) + iy_n(kr) \quad (3-88)$$

The first two orders of the three spherical Bessel functions are given by the equations:

$$j_0(kr) = \frac{\sin(kr)}{kr} \quad (3-89)$$

$$j_1(kr) = \frac{\sin(kr)}{(kr)^2} - \frac{\cos(kr)}{kr} \quad (3-90)$$

$$y_0(kr) = -\frac{\cos(kr)}{kr} \quad (3-91)$$

$$y_1(kr) = -\frac{\cos(kr)}{(kr)^2} - \frac{\sin(kr)}{kr} \quad (3-92)$$

$$h_0(kr) = -\frac{i}{kr} \exp(ikr) \quad (3-93)$$

$$h_1(kr) = -\frac{kr+i}{(kr)^2} \exp(ikr) \quad (3-94)$$

while the higher-order functions can be generated by use of the following recursive equation:

$$z_n(kr) = \frac{2n-1}{kr} z_{n-1}(kr) - z_{n-2}(kr) \quad (3-95)$$

The functions $y_n()$ and $h_n()$ become infinite at the origin for all orders n . Besides, $j_n()$ remains finite at the same point and order set ($j_0(0)=1$ and $j_n(0)=0$ for $n>0$).

When the solutions of differential equations (3-78) to (3-80) are combined together, they produce two sets of generating functions that satisfy the scalar wave equation (3-75). These functions are denoted by $g_{emn}()$ and $g_{omn}()$ and have the forms:

$$g_{emn}(r, \theta, \phi) = \cos(m\phi) \cdot P_n^m(\cos\theta) \cdot z_n(kr) \quad (3-96)$$

$$g_{omn}(r, \theta, \phi) = \sin(m\phi) \cdot P_n^m(\cos\theta) \cdot z_n(kr) \quad (3-97)$$

Hence the vector spherical harmonics generated by $g_{emn}()$ and $g_{omn}()$ are:

$$\tilde{\mathbf{M}}_{emn} = \nabla \times (\mathbf{r}g_{emn}) \quad (3-98)$$

$$\tilde{\mathbf{M}}_{omn} = \nabla \times (\mathbf{r}g_{omn}) \quad (3-99)$$

$$\tilde{\mathbf{N}}_{emn} = \frac{\nabla \times \tilde{\mathbf{M}}_{emn}}{k} \quad (3-100)$$

$$\tilde{\mathbf{N}}_{omn} = \frac{\nabla \times \tilde{\mathbf{M}}_{omn}}{k} \quad (3-101)$$

which, in component form, may be written as:

$$\begin{aligned} \tilde{\mathbf{M}}_{emn} = & \frac{-m}{\sin\theta} \cdot \sin(m\phi) \cdot P_n^m(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\theta \\ & - \cos(m\phi) \cdot \frac{dP_n^m(\cos\theta)}{d\theta} \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-102)$$

$$\begin{aligned} \tilde{\mathbf{M}}_{omn} = & \frac{m}{\sin\theta} \cdot \cos(m\phi) \cdot P_n^m(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\theta \\ & - \sin(m\phi) \cdot \frac{dP_n^m(\cos\theta)}{d\theta} \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-103)$$

$$\begin{aligned} \tilde{\mathbf{N}}_{emn} = & \frac{n \cdot (n+1)}{kr} \cdot \cos(m\phi) \cdot P_n^m(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_r \\ & + \frac{1}{kr} \cdot \cos(m\phi) \cdot \frac{dP_n^m(\cos\theta)}{d\theta} \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\theta \\ & - \frac{m}{kr} \cdot \sin(m\phi) \cdot \frac{P_n^m(\cos\theta)}{\sin\theta} \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-104)$$

$$\begin{aligned} \tilde{\mathbf{N}}_{omn} = & \frac{n \cdot (n+1)}{kr} \cdot \sin(m\phi) \cdot P_n^m(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_r \\ & + \frac{1}{kr} \cdot \sin(m\phi) \cdot \frac{dP_n^m(\cos\theta)}{d\theta} \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\theta \\ & + \frac{m}{kr} \cdot \cos(m\phi) \cdot \frac{P_n^m(\cos\theta)}{\sin\theta} \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-105)$$

Not only these harmonics are capable of generating any solution to the Maxwell equations (3-20) to (3-23) for $\tilde{\rho}_F = 0$ and vector wave equations (3-28), (3-29), but also they are found to be mutually orthogonal sets of functions (see [169], pp. 90-91). In

other words, an arbitrary but physically realisable plane wave propagating through the medium or being sustained inside the particle considered by the Mie scattering problem has an electric field phasor $\tilde{\mathbf{E}}$ that can be expanded as an infinite series of vector spherical harmonics:

$$\tilde{\mathbf{E}} = \sum_{m=0}^{\infty} \sum_{n=m}^{\infty} (B_{emn} \tilde{\mathbf{M}}_{emn} + B_{omn} \tilde{\mathbf{M}}_{omn} + A_{emn} \tilde{\mathbf{N}}_{emn} + A_{omn} \tilde{\mathbf{N}}_{omn}) \quad (3-106)$$

with coefficients B_{emn} , B_{omn} , A_{emn} and A_{omn} given by:

$$B_{emn} = \frac{\int_0^{2\pi} \int_0^{\pi} \tilde{\mathbf{E}} \cdot \tilde{\mathbf{M}}_{emn} \cdot \sin \theta \, d\theta \, d\phi}{\int_0^{2\pi} \int_0^{\pi} |\tilde{\mathbf{M}}_{emn}|^2 \cdot \sin \theta \, d\theta \, d\phi} \quad (3-107)$$

$$B_{omn} = \frac{\int_0^{2\pi} \int_0^{\pi} \tilde{\mathbf{E}} \cdot \tilde{\mathbf{M}}_{omn} \cdot \sin \theta \, d\theta \, d\phi}{\int_0^{2\pi} \int_0^{\pi} |\tilde{\mathbf{M}}_{omn}|^2 \cdot \sin \theta \, d\theta \, d\phi} \quad (3-108)$$

$$A_{emn} = \frac{\int_0^{2\pi} \int_0^{\pi} \tilde{\mathbf{E}} \cdot \tilde{\mathbf{N}}_{emn} \cdot \sin \theta \, d\theta \, d\phi}{\int_0^{2\pi} \int_0^{\pi} |\tilde{\mathbf{N}}_{emn}|^2 \cdot \sin \theta \, d\theta \, d\phi} \quad (3-109)$$

$$A_{omn} = \frac{\int_0^{2\pi} \int_0^{\pi} \tilde{\mathbf{E}} \cdot \tilde{\mathbf{N}}_{omn} \cdot \sin \theta \, d\theta \, d\phi}{\int_0^{2\pi} \int_0^{\pi} |\tilde{\mathbf{N}}_{omn}|^2 \cdot \sin \theta \, d\theta \, d\phi} \quad (3-110)$$

and a magnetic field phasor $\tilde{\mathbf{H}}$ obtained by (3-21).

The incident plane wave of the Mie scattering problem propagates in the direction of z-axis. Without loss of generalisation, the incident wave will be considered as x-polarised. The electric field phasor of the incident wave is written in spherical polar coordinates as:

$$\tilde{\mathbf{E}}_i = E_0 \exp[i(kr \cos \theta - \omega t)] \cdot \hat{\mathbf{e}}_x \quad (3-111)$$

where:

$$\hat{\mathbf{e}}_x = \sin \theta \cos \phi \cdot \hat{\mathbf{e}}_r + \cos \theta \cos \phi \cdot \hat{\mathbf{e}}_{\theta} - \sin \phi \cdot \hat{\mathbf{e}}_{\phi} \quad (3-112)$$

If that electric field is expanded in an infinite series of vector spherical harmonics as described by (3-106), it follows from (3-102), (3-105) and (3-112), together with the orthogonality of the sine and cosine functions, that $B_{emn}=A_{omn}=0$ for all m and n . Moreover, the remaining series coefficients vanish unless $m=1$ for the same reason. Also, the only spherical Bessel function appropriate in the generating functions $g_{emn}()$ and $g_{omn}()$ is the one of the first kind, $j_n()$, because the incident electric field is finite at the origin. Thus, the expansion of $\tilde{\mathbf{E}}_i$ takes the form:

$$\tilde{\mathbf{E}}_i = \sum_{n=1}^{\infty} (B_{o1n} \tilde{\mathbf{M}}_{o1n}^j + A_{e1n} \tilde{\mathbf{N}}_{e1n}^j) \quad (3-113)$$

where the superscript j has been appended to vector spherical harmonics as a reminder that the radial dependence of the generating functions is specified by $j_n()$.

The final expressions of the expansion coefficients in (3-113) are obtained by (3-108) and (3-109) to be:

$$B_{o1n} = i^n \frac{2n+1}{n(n+1)} E_0 \quad (3-114)$$

$$A_{e1n} = i^n \frac{2n+1}{n(n+1)} (-iE_0) \quad (3-115)$$

Hence the electric field phasor of the incident wave is finally expressed in vector spherical harmonics by:

$$\tilde{\mathbf{E}}_i = E_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} (\tilde{\mathbf{M}}_{o1n}^j - i\tilde{\mathbf{N}}_{e1n}^j) \quad (3-116)$$

and the corresponding magnetic field phasor is obtained by (3-21):

$$\tilde{\mathbf{H}}_i = -\sqrt{\frac{\epsilon}{\mu}} E_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} (\tilde{\mathbf{M}}_{e1n}^j + i\tilde{\mathbf{N}}_{o1n}^j) \quad (3-117)$$

where ϵ , μ are the (real) permittivity and permeability of the surrounding medium.

The field phasors of the internal and scattered electromagnetic waves shown in figure 3-2 can also be expanded in vector spherical harmonics by techniques similar to the one followed in the incident wave case. These expansions, however, ought to satisfy the following condition (imposed by the axiom of electromagnetic energy conservation) at the boundary between the sphere and the surrounding medium:

$$(\tilde{\mathbf{E}}_i + \tilde{\mathbf{E}}_s - \tilde{\mathbf{E}}_1) \times \hat{\mathbf{e}}_r = (\tilde{\mathbf{H}}_i + \tilde{\mathbf{H}}_s - \tilde{\mathbf{H}}_1) \times \hat{\mathbf{e}}_r = 0 \quad (3-118)$$

The boundary condition (3-118), the orthogonality of the vector harmonics, and the form of the expansion of the incident fields require that the coefficients in the

internal and scattered field expansions vanish for all $m \neq 1$. Furthermore, the only appropriate spherical Bessel function in the generating functions for the vector harmonics inside the particle is the one of first kind, $j_n()$, because the internal wave fields are finite at the origin. Thus, the expansions of $\tilde{\mathbf{E}}_1$ and $\tilde{\mathbf{H}}_1$ are:

$$\tilde{\mathbf{E}}_1 = E_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} (c_n \tilde{\mathbf{M}}_{o1n}^{j1} - i d_n \tilde{\mathbf{N}}_{e1n}^{j1}) \quad (3-119)$$

$$\tilde{\mathbf{H}}_1 = -\sqrt{\frac{\epsilon_1}{\mu_1}} E_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} (d_n \tilde{\mathbf{M}}_{e1n}^{j1} + i c_n \tilde{\mathbf{N}}_{o1n}^{j1}) \quad (3-120)$$

where c_n , d_n are coefficients whose values are derived later in this section, and ϵ_1 , μ_1 are the complex permittivity and permeability of the particle. The superscript $j1$ has been appended to vector spherical harmonics as a reminder that the radial dependence of the generating functions is specified by $j_n()$ and k_1 , the wave number of the internal wave, replaces k in equations (3-102) to (3-105).

The field expansions of the scattered wave do not require any restrictions imposed on the kind of spherical Bessel function used in $g_{emn}()$ and $g_{omn}()$, as either of functions $j_n()$, $y_n()$ and $h_n()$ is well behaved in the region outside the particle. However, the spherical Hankel function is the one selected for the vector harmonic generation task due to its excellent asymptotic behaviour (see [169], pp. 93-94). Therefore, the field expansions of the scattered wave take the form:

$$\tilde{\mathbf{E}}_s = E_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} (-b_n \tilde{\mathbf{M}}_{o1n}^h + i a_n \tilde{\mathbf{N}}_{e1n}^h) \quad (3-121)$$

$$\tilde{\mathbf{H}}_s = \sqrt{\frac{\epsilon}{\mu}} E_0 \sum_{n=1}^{\infty} i^n \frac{2n+1}{n(n+1)} (a_n \tilde{\mathbf{M}}_{e1n}^h + i b_n \tilde{\mathbf{N}}_{o1n}^h) \quad (3-122)$$

where a_n , b_n are coefficients whose values are derived later in this section, and the superscript h has been appended to vector spherical harmonics as a reminder that the radial dependence of the generating functions is specified by $h_n()$.

The vector harmonics $\tilde{\mathbf{M}}_{e1n}$, $\tilde{\mathbf{M}}_{o1n}$, $\tilde{\mathbf{N}}_{e1n}$ and $\tilde{\mathbf{N}}_{o1n}$ appearing in field expansions (3-116), (3-117), and (3-119) to (3-122) can be written in more concise form once the functions $\pi_n()$ and $\tau_n()$ defined by:

$$\pi_n[f(\theta)] \equiv \frac{P_n^1[f(\theta)]}{\sin \theta} \quad (3-123)$$

$$\tau_n[f(\theta)] \equiv \frac{dP_n^1[f(\theta)]}{d\theta} \quad (3-124)$$

are used in equations (3-102) to (3-105). The simplified expressions for the vector harmonics are:

$$\begin{aligned} \tilde{\mathbf{M}}_{e1n} = & -\sin\phi \cdot \pi_n(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\theta \\ & -\cos\phi \cdot \tau_n(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-125)$$

$$\begin{aligned} \tilde{\mathbf{M}}_{o1n} = & \cos\phi \cdot \pi_n(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\theta \\ & -\sin\phi \cdot \tau_n(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-126)$$

$$\begin{aligned} \tilde{\mathbf{N}}_{e1n} = & \frac{n(n+1)}{kr} \cdot \cos\phi \cdot \sin\theta \cdot \pi_n(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_r \\ & + \frac{1}{kr} \cdot \cos\phi \cdot \tau_n(\cos\theta) \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\theta \\ & - \frac{1}{kr} \cdot \sin\phi \cdot \pi_n(\cos\theta) \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-127)$$

$$\begin{aligned} \tilde{\mathbf{N}}_{o1n} = & \frac{n(n+1)}{kr} \cdot \sin\phi \cdot \sin\theta \cdot \pi_n(\cos\theta) \cdot z_n(kr) \cdot \hat{\mathbf{e}}_r \\ & + \frac{1}{kr} \cdot \sin\phi \cdot \tau_n(\cos\theta) \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\theta \\ & + \frac{1}{kr} \cdot \cos\phi \cdot \pi_n(\cos\theta) \cdot \frac{d}{d(kr)} [kr \cdot z_n(kr)] \cdot \hat{\mathbf{e}}_\phi \end{aligned} \quad (3-128)$$

where $z_n()$ and k are determined exactly by the superscripts appended to the individual vector harmonics of the field expansions mentioned above.

The angle-dependent functions $\pi_n()$ and $\tau_n()$ can be computed by upward recurrence from the relations:

$$\pi_n(\cos\theta) = \frac{2n-1}{n-1} \cdot \cos\theta \cdot \pi_{n-1}(\cos\theta) - \frac{n}{n-1} \cdot \pi_{n-2}(\cos\theta) \quad (3-129)$$

$$\tau_n(\cos\theta) = n \cdot \cos\theta \cdot \pi_n(\cos\theta) - (n+1) \cdot \pi_{n-1}(\cos\theta) \quad (3-130)$$

beginning with:

$$\pi_0(\cos\theta) = 0 \quad (3-131)$$

$$\pi_1(\cos\theta) = 1 \quad (3-132)$$

The functions $\pi_n()+\tau_n()$ and $\pi_n()-\tau_n()$ are mutually orthogonal, i.e.:

$$\int_0^\pi (\tau_n + \pi_n)(\tau_m + \pi_m) \sin\theta d\theta = \int_0^\pi (\tau_n - \pi_n)(\tau_m - \pi_m) \sin\theta d\theta = 0 \quad m \neq n \quad (3-133)$$

A complete solution to the Mie scattering problem can only be found if the *scattering coefficients* a_n , b_n and *internal coefficients* c_n , d_n that appear in field expansions (3-121), (3-122) and (3-119), (3-120) are derived first. The coefficient determi-

nation task is facilitated by the introduction of the *Riccati-Bessel functions* of order n and kind first ($\psi_n()$), second ($\zeta_n()$), or third ($\xi_n()$). These functions are defined by:

$$\psi_n(x) \equiv x \cdot j_n(x) \quad (3-134)$$

$$\zeta_n(x) \equiv x \cdot y_n(x) \quad (3-135)$$

$$\xi_n(x) \equiv x \cdot h_n(x) \quad (3-136)$$

The derivatives of the Riccati-Bessel functions are given by:

$$\frac{dq_n(x)}{dx} = q_{n-1}(x) - \frac{n}{x}q_n(x) \quad (3-137)$$

where $q_n()$ stands for $\psi_n()$, $\zeta_n()$, or $\xi_n()$. Equation (3-137) can also be used to compute the Riccati-Bessel derivatives in vector harmonic expressions (3-127), (3-128).

The boundary condition (3-118) can be written in field component form as a set of four linear equations:

$$E_{i\theta} + E_{s\theta} = E_{1\theta} \quad (3-138)$$

$$E_{i\phi} + E_{s\phi} = E_{1\phi} \quad (3-139)$$

$$H_{i\theta} + H_{s\theta} = H_{1\theta} \quad (3-140)$$

$$H_{i\phi} + H_{s\phi} = H_{1\phi} \quad (3-141)$$

satisfied at the spherical surface $r=\alpha$, where α is the radius of the particle (see figure 3-3). The four expansion coefficients are finally obtained from the orthogonality of sine and cosine, the identity (3-133), the boundary conditions (3-138) to (3-141) together with the expansions (3-116), (3-117), (3-119) to (3-122), and the expressions (3-125) to (3-128) for the vector harmonics. The results are:

$$a_n = \frac{\mu \cdot m \cdot \psi_n(m\chi) \cdot \frac{d\psi_n(\chi)}{d\chi} - \mu_1 \cdot \psi_n(\chi) \cdot \frac{d\psi_n(m\chi)}{d(m\chi)}}{\mu \cdot m \cdot \psi_n(m\chi) \cdot \frac{d\xi_n(\chi)}{d\chi} - \mu_1 \cdot \xi_n(\chi) \cdot \frac{d\psi_n(m\chi)}{d(m\chi)}} \quad (3-142)$$

$$b_n = \frac{\mu_1 \cdot \psi_n(m\chi) \cdot \frac{d\psi_n(\chi)}{d\chi} - \mu \cdot m \cdot \psi_n(\chi) \cdot \frac{d\psi_n(m\chi)}{d(m\chi)}}{\mu_1 \cdot \psi_n(m\chi) \cdot \frac{d\xi_n(\chi)}{d\chi} - \mu \cdot m \cdot \xi_n(\chi) \cdot \frac{d\psi_n(m\chi)}{d(m\chi)}} \quad (3-143)$$

$$c_n = \frac{\mu_1 \cdot m \cdot \psi_n(\chi) \cdot \frac{d\xi_n(\chi)}{d\chi} - \mu_1 \cdot m \cdot \xi_n(\chi) \cdot \frac{d\psi_n(\chi)}{d\chi}}{\mu_1 \cdot \psi_n(m\chi) \cdot \frac{d\xi_n(\chi)}{d\chi} - \mu \cdot m \cdot \xi_n(\chi) \cdot \frac{d\psi_n(m\chi)}{d(m\chi)}} \quad (3-144)$$

$$d_n = \frac{\mu_1 \cdot m \cdot \psi_n(\chi) \cdot \frac{d\xi_n(\chi)}{d\chi} - \mu_1 \cdot m \cdot \xi_n(\chi) \cdot \frac{d\psi_n(\chi)}{d\chi}}{\mu \cdot m \cdot \psi_n(m\chi) \cdot \frac{d\xi_n(\chi)}{d\chi} - \mu_1 \cdot \xi_n(\chi) \cdot \frac{d\psi_n(m\chi)}{d(m\chi)}} \quad (3-145)$$

where

$$\chi \equiv k\alpha = \frac{2\pi N\alpha}{\lambda} \quad (3-146)$$

is the *size parameter* and

$$m \equiv \frac{k_1}{k} = \frac{N_1}{N} \quad (3-147)$$

is the *relative refractive index*. Variables N_1 , N in equation (3-147) are the refractive indices of particle and medium respectively.

Two problems of numerical nature arise when the expansions (3-119) to (3-122) are used to obtain quantitative results for the internal or scattered fields at a point in space under Mie scattering conditions. The first difficulty is the infinite number of terms required for the exact calculation of the series expansions. Fortunately, these series are uniformly convergent [169-171], and thus can be terminated after a sufficiently large number of terms while the resulting approximation error remains arbitrarily small regardless of the point in space considered. An estimate of a series expansion termination point that provides good approximation results in most practical applications was proposed in [172] and given by:

$$n_c = \lceil \chi + 4.05 \cdot \sqrt[3]{\chi + 2} \rceil \quad (3-148)$$

This limit will be implied in all numerical field solutions to Mie scattering problems presented by this study unless otherwise stated.

The second numerical problem is the unacceptably high round-off error observed if the *upward recurrence* formula (3-95) is used to obtain all orders of interest ($0 \leq n \leq n_c$) of the spherical Bessel function of the first kind, $j_n(\cdot)$. The solution to this computational instability is the computation of all necessary $j_n(\cdot)$ orders by *downward recurrence*. Specifically, the function sequence $\hat{j}_n(\cdot)$ is calculated first for an arbitrary argument $x \neq 0$ by the initial conditions:

$$\hat{j}_{n_c}(x) = 10^{-10} \quad (3-149)$$

$$\hat{j}_{n_c+1}(x) = 0 \quad (3-150)$$

and the recurrence formula:

$$\hat{j}_n(x) = \frac{2n+3}{x} \hat{j}_{n+1}(x) - \hat{j}_{n+2}(x) \quad (3-151)$$

The index n_{st} of the $\hat{j}_n(\cdot)$ sequence is always assigned a sufficiently higher integer value than n_c , e.g.:

$$n_{st} = n_c + \sqrt{101 + n_c} \quad (3-152)$$

Provided that $\hat{j}_0(x) \neq 0$, the spherical Bessel functions $j_n(\cdot)$ are derived from the function sequence $\hat{j}_n(\cdot)$ by the relation:

$$j_n(x) = \frac{j_0(x)}{\hat{j}_0(x)} \cdot \hat{j}_n(x) \quad (3-153)$$

where $j_0(\cdot)$ is given by (3-89). Should $\hat{j}_0(x) = 0$, another small value is assigned to $\hat{j}_{n_s}(x)$ in (3-149) and the whole procedure is repeated.

If the number of terms required to approximate the series expansions (3-119) to (3-122) is $n_c > 50$, the calculation of the first n_c orders of $j_n(\cdot)$ at point x is performed in segments to avoid high round-off errors. As $j_0(x)$ is given by (3-89), the numerical algorithm described above may be used to calculate the first 50 orders of $j_n(x)$ by setting $n_c = 50$ in (3-152). As soon as $j_{50}(x)$ is computed, the next 50 orders of $j_n(x)$ are calculated by executing the algorithm as before with the only exception that $j_0(x)$ is substituted for $j_{50}(x)$ in (3-153). The last algorithmic step is repeated continuously to get 50 more orders of $j_n(x)$ per iteration (the n_{th} iteration substitutes $j_0(x)$ for $j_{50n-50}(x)$ in (3-153) and returns orders $50n-49$ to $50n$) until order n_c is finally reached.

The scattered wave ($\mathbf{E}_s, \mathbf{H}_s$) given by (3-121), (3-122) is approximately *transverse* ($\hat{\mathbf{e}}_r \cdot \tilde{\mathbf{E}}_s \cong 0, \hat{\mathbf{e}}_r \cdot \tilde{\mathbf{H}}_s \cong 0$) at sufficiently large distances from the origin, i.e. in the *far-field region* ($kr \gg n_c^2$). The transverse components of the scattered electric field phasor $\tilde{\mathbf{E}}_s$ at a point (r, θ, ϕ) in that region have the asymptotic forms:

$$\tilde{E}_{s\theta} \cong E_0 \cos \phi \cdot \frac{\exp(ikr)}{-ikr} \cdot S_2(\theta) \quad (3-154)$$

$$\tilde{E}_{s\phi} \cong -E_0 \sin \phi \cdot \frac{\exp(ikr)}{-ikr} \cdot S_1(\theta) \quad (3-155)$$

where:

$$S_1(\theta) = \sum_{n=1}^{n_c} \frac{2n+1}{n(n+1)} [a_n \pi_n(\cos \theta) + b_n \tau_n(\cos \theta)] \quad (3-156)$$

$$S_2(\theta) = \sum_{n=1}^{n_c} \frac{2n+1}{n(n+1)} [a_n \tau_n(\cos\theta) + b_n \pi_n(\cos\theta)] \quad (3-157)$$

If the incident plane wave propagates in the direction of z-axis but is no longer required being x-polarised, it has an electric field phasor $\tilde{\mathbf{E}}_i$ that lies in the (x, y) plane. It is convenient to resolve $\tilde{\mathbf{E}}_i$ into components parallel ($\tilde{\mathbf{E}}_{\parallel i}$) and perpendicular ($\tilde{\mathbf{E}}_{\perp i}$) to the *scattering plane*, i.e. the (r, z) plane:

$$\tilde{\mathbf{E}}_i = \tilde{E}_{\parallel i} \hat{\mathbf{e}}_{\parallel i} + \tilde{E}_{\perp i} \hat{\mathbf{e}}_{\perp i} = [E_0 \cos\phi \exp(ikz)] \hat{\mathbf{e}}_{\parallel i} + [E_0 \sin\phi \exp(ikz)] \hat{\mathbf{e}}_{\perp i} \quad (3-158)$$

where ($\hat{\mathbf{e}}_{\parallel i}$, $\hat{\mathbf{e}}_{\perp i}$) given by:

$$\hat{\mathbf{e}}_{\parallel i} = \sin\theta \cdot \hat{\mathbf{e}}_r + \cos\theta \cdot \hat{\mathbf{e}}_\theta \quad (3-159)$$

$$\hat{\mathbf{e}}_{\perp i} = -\hat{\mathbf{e}}_\phi \quad (3-160)$$

is a pair of orthogonal unit vectors lying in the (x, y) plane, $z = r \cos\theta$, and k is the (real) wave number of the incident wave in the medium surrounding the particle.

The scattered electric field phasor $\tilde{\mathbf{E}}_s$ in the far-field region can also be resolved into components parallel ($\tilde{\mathbf{E}}_{\parallel s}$) and perpendicular ($\tilde{\mathbf{E}}_{\perp s}$) to the scattering plane:

$$\tilde{\mathbf{E}}_s = \tilde{E}_{\parallel s} \hat{\mathbf{e}}_{\parallel s} + \tilde{E}_{\perp s} \hat{\mathbf{e}}_{\perp s} \quad (3-161)$$

where ($\hat{\mathbf{e}}_{\parallel s}$, $\hat{\mathbf{e}}_{\perp s}$) given by:

$$\hat{\mathbf{e}}_{\parallel s} = \hat{\mathbf{e}}_\theta \quad (3-162)$$

$$\hat{\mathbf{e}}_{\perp s} = -\hat{\mathbf{e}}_\phi \quad (3-163)$$

is a pair of orthogonal unit vectors lying in the (θ , ϕ) plane. The complete axis and orthonormal basis vector geometry is illustrated in figure 3-4.

If the unit vector pairs ($\hat{\mathbf{e}}_{\parallel i}$, $\hat{\mathbf{e}}_{\perp i}$) and ($\hat{\mathbf{e}}_{\parallel s}$, $\hat{\mathbf{e}}_{\perp s}$) are substituted for the triple ($\hat{\mathbf{e}}_r$, $\hat{\mathbf{e}}_\theta$, $\hat{\mathbf{e}}_\phi$) in (3-158) and (3-161) and the transverse electrical field components of the substitution outcome are compared with (3-154) and (3-155), it emerges that:

$$\begin{bmatrix} \tilde{E}_{\parallel s} \\ \tilde{E}_{\perp s} \end{bmatrix} = \frac{\exp[ik(r-z)]}{-ikr} \begin{bmatrix} S_2(\theta) & 0 \\ 0 & S_1(\theta) \end{bmatrix} \begin{bmatrix} \tilde{E}_{\parallel i} \\ \tilde{E}_{\perp i} \end{bmatrix} \quad (3-164)$$

The 2x2 diagonal matrix of (3-164) is called the *amplitude scattering matrix* and is independent of the radial component r.

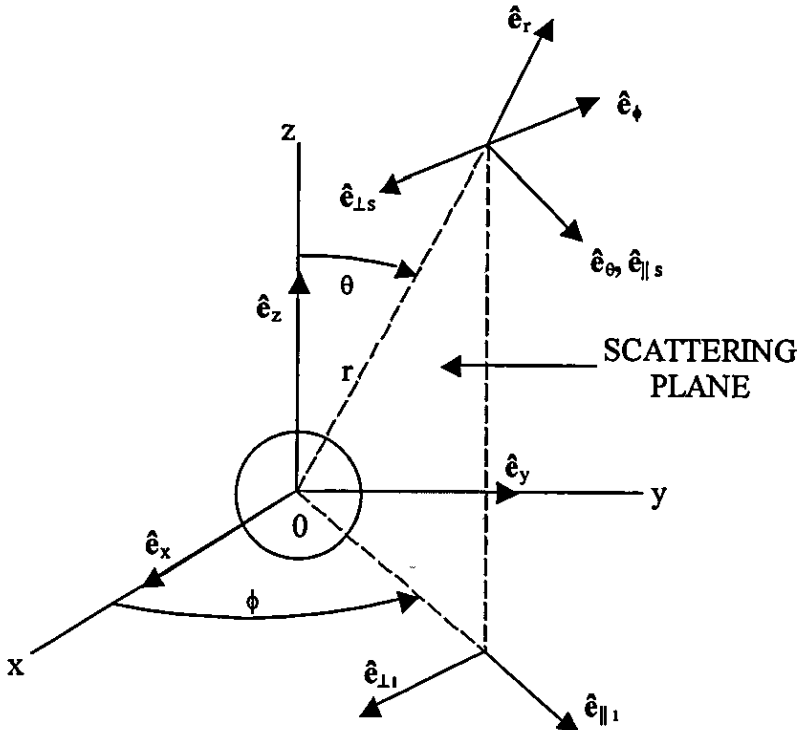


Figure 3-4 – Scattering by a spherical particle

The relation between the incident and far-field scattered Stokes parameters \mathbf{S}_i and \mathbf{S}_s follows from definitions (3-65) to (3-68) and equation (3-164):

$$\mathbf{S}_s = \mathbf{M} \cdot \mathbf{S}_i \quad (3-165)$$

where the matrix \mathbf{M} given by:

$$\mathbf{M} = \frac{1}{k^2 r^2} \begin{bmatrix} S_{11}(\theta) & S_{12}(\theta) & 0 & 0 \\ S_{12}(\theta) & S_{11}(\theta) & 0 & 0 \\ 0 & 0 & S_{33}(\theta) & S_{34}(\theta) \\ 0 & 0 & -S_{34}(\theta) & S_{33}(\theta) \end{bmatrix} \quad (3-166)$$

is the (far-field) Muller matrix of the Mie scattering problem and:

$$S_{11}(\theta) = \frac{1}{2} \cdot [|S_2(\theta)|^2 + |S_1(\theta)|^2] \quad (3-167)$$

$$S_{12}(\theta) = \frac{1}{2} \cdot [|S_2(\theta)|^2 - |S_1(\theta)|^2] \quad (3-168)$$

$$S_{33}(\theta) = \frac{1}{2} \cdot [S_2^*(\theta) S_1(\theta) + S_2(\theta) S_1^*(\theta)] \quad (3-169)$$

$$S_{34}(\theta) = \frac{i}{2} \cdot [S_2^*(\theta) S_1(\theta) - S_2(\theta) S_1^*(\theta)] \quad (3-170)$$

are the elements of \mathbf{M} expressed as functions of the truncated series (3-156) and (3-157). Only three of these four matrix elements are independent because:

$$S_{11}^2 = S_{12}^2 + S_{13}^2 + S_{14}^2 \quad (3-171)$$

The lengthy but straightforward Mie scattering theory is now complete for the purposes of this work. The next problem considered is the calculation of light irradiance scattered from illuminated tenuous suspensions of particles and measured at an arbitrary angle by a polar nephelometric device. The modelling procedure and solution of this problem are the subjects of §3.5.

3.5 Nephelometric scattering and turbidity measurements

A generic polar nephelometric device (or *polar nephelometer*) is illustrated in figure 3-5. The device consists of an arbitrarily long cylindrical chamber of radius R that contains particles suspended in a simple medium. The nephelometric arrangement also includes a light source and set of light detectors placed at fixed positions in the perimeter of a single chamber cross-section plane, the *nephelometric plane*. All light sources and detectors of the nephelometer face the centre of the circular chamber cross-section. The light source emits an infinite number of plane waves whose (visual) wavelengths λ span the interval $[\lambda_{\min}, \lambda_{\max}]$. The amount of electromagnetic irradiance I_s emitted by the light source is constant and distributed over the set of transmitted plane waves according to the *spectral intensity function* $\text{src}(\lambda)$ of that source. The emitted *polychromatic* (i.e., multi-frequency) light forms a perfectly collimated cylindrical beam of radius $w/2$ ($w \ll R$) that originates from the light source and illuminates only the part of particle suspension that falls within the beam volume. The light detector placed at a *scattering angle* θ with respect to the chamber cross-section point opposite the light source receives light scattered from the illuminated suspension volume that falls within its *detection cone*. The cone is centred at the detector considered and has an opening determined by the detector's *acceptance angle* δ . The total scattered light irradiance I_r measured by the detector is given by the weighted sum of irradiances carried by the scattered plane waves the detector intercepts. The irradiance weighing is performed over the scattered wave frequency domain with respect to the *spectral sensitivity function* $\text{rec}(\lambda)$ of the light detector.

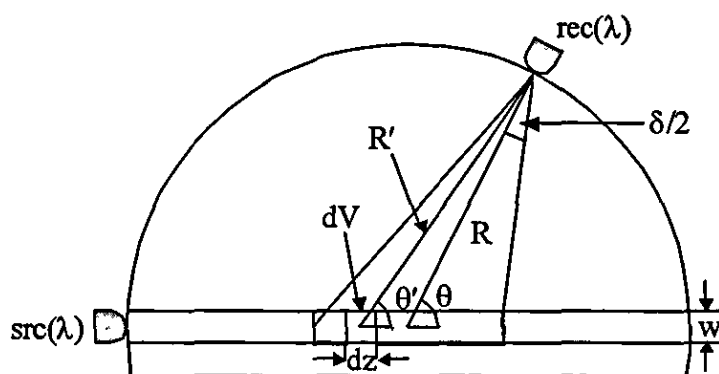


Figure 3-5 – Light scattering measurement by a polar nephelometer

The total scattered irradiance measured by the detector of the nephelometric model described above can be derived analytically under the following conditions:

- a) The scattered light has the same wavelength or wavelength range as the incident light. This condition is met by the Mie scattering problem and most practical scattering applications.
- b) The suspended particles are sufficiently far from each other so that the scattering by one particle can be studied without reference to the other ones. This is called the *independent scattering* condition and has the direct consequence that the irradiances scattered by the various particles must be added without regard to phase to obtain the total irradiance at any point in space that does not fall in the line of incident light propagation. A mutual particle distance of three times their average radius is sufficient to assume independent scattering and met by the most practical scattering problems.
- c) Each scattering particle is predominantly exposed to light originating from a distant source (or forming a collimated beam) and negligibly to light scattered by the other particles. This is the *single scattering* condition and is met by *tenuous* particle suspensions. The condition is practically demonstrated when the scattered irradiance measurements double by doubling the particle concentration in the investigated suspension.
- d) The particle suspension is macroscopically *homogeneous*, i.e., there is no statistical preference of particle shape, size and number contained within an infinitesimal volume centred at any point of the suspension.
- e) The suspended particles are spherical (i.e., LATEX spheres). If the particles do not satisfy this condition, the analytical solution of scattered irradiance obtained in this section can only be considered as first-order approximation.

The first step in the scattered irradiance derivation procedure is the consideration of an infinitesimal cylinder of radius $w/2$ and height dz . The cylinder falls entirely within the detection cone of a light detector. The centre of the cylinder sees the detector at an angle θ' with respect to the cylinder's line of symmetry; the latter is perfectly aligned to the line of symmetry of the light beam emitted by the source. As a matter of fact, the infinitesimal particle suspension volume $dV(\theta, \theta')$ enclosed by the cylinder is entirely illuminated by the nephelometric light source. Figure 3-5 illustrates a typical cylinder placement within the nephelometric chamber.

The particles contained by the suspension volume $dV(\theta, \theta')$ occupy a total volume $d\hat{V}_p(\theta, \theta')$ that is given by:

$$d\hat{V}_p(\theta, \theta') = \frac{C_m}{d_m} \cdot dV(\theta, \theta') \quad (3-172)$$

where C_m is the (bulk) mass concentration of particles suspended in the nephelometric chamber and d_m is the (bulk) mass density of these particles.

Let $d_p \in [d_{\min}, d_{\max}]$ and $V_p \in [V_{\min}, V_{\max}]$ denote the diameter and volume of an arbitrary (spherical) particle in the suspension within the nephelometric chamber. The two particle variables are random and hence described by their associated probability density functions (pdfs), namely *particle diameter pdf*, $p_{d_p}(d_p)$, and *particle volume pdf*, $p_{V_p}(V_p)$. These two functions are independent of position within the chamber due to the homogeneity property of the suspension. Therefore the total volume $\Delta\hat{V}_p(\theta, \theta', V_p, \Delta V_p)$ occupied by particles lying within the suspension volume $dV(\theta, \theta')$ and having individual volumes in the interval $[V_p, V_p + \Delta V_p]$ is given by:

$$\Delta\hat{V}_p(\theta, \theta', V_p, \Delta V_p) = d\hat{V}_p(\theta, \theta') \cdot \int_{V_p}^{V_p + \Delta V_p} p_{V_p}(x) dx \quad (3-173)$$

The integral on the right-hand side of (3-173) is the volume fraction assigned to the particle volume range $[V_p, V_p + \Delta V_p]$.

Due to the spherical shape of particles suspended in the chamber, the particle diameter d_p can be obtained from particle volume V_p by the following relation:

$$d_p(V_p) = \left(\frac{6V_p}{\pi} \right)^{\frac{1}{3}} \quad (3-174)$$

Therefore equation (3-173) may be expressed in the particle diameter domain as:

$$\Delta \hat{V}_p(\theta, \theta', d_p, \Delta d_p) = d\hat{V}_p(\theta, \theta') \cdot \int_{d_p}^{d_p + \Delta d_p} p_{d_p}(x) dx \quad (3-175)$$

If the particle diameter range Δd_p is infinitesimal, equation (3-175) degenerates to:

$$d\hat{V}_p(\theta, \theta', d_p) = d\hat{V}_p(\theta, \theta') \cdot p_{d_p}(d_p) \cdot dd_p \quad (3-176)$$

where $d\hat{V}_p(\theta, \theta', d_p)$ denotes the total volume occupied by particles lying within the suspension volume $dV(\theta, \theta')$ and having effective diameters equal to d_p .

The total number of particles of effective diameter d_p that are contained in $dV(\theta, \theta')$ is denoted by $dN_p(\theta, \theta', d_p)$ and derived by:

$$dN_p(\theta, \theta', d_p) = \frac{d\hat{V}_p(\theta, \theta', d_p)}{d\bar{V}_p(d_p)} \quad (3-177)$$

where $d\bar{V}_p(d_p)$ is the average volume of particles of effective diameter d_p given by:

$$d\bar{V}_p(d_p) = \frac{\pi d_p^3}{6} \quad (3-178)$$

due to the spherical particle shape considered. Equations (3-176) to (3-178) yield:

$$dN_p(\theta, \theta', d_p) = \frac{6 p_{d_p}(d_p)}{\pi d_p^3} \cdot d\hat{V}_p(\theta, \theta') \cdot dd_p \quad (3-179)$$

The irradiance $dI_i(\lambda)$ emitted by the light source and carried by plane waves of *effective* wavelength λ (or real wavelengths in the interval $[\lambda, \lambda + d\lambda]$ where $d\lambda$ is an infinitesimal wavelength range) is given by:

$$dI_i(\lambda) = I_i \text{src}_n(\lambda) d\lambda \quad (3-180)$$

where $\text{src}_n(\lambda)$ is the normalised spectral intensity function. The emitted irradiance is attenuated as it traverses the simple medium to reach the volume $dV(\theta, \theta')$. The attenuated irradiance is derived by (3-51) and given by:

$$dI_i(\lambda, \theta, \theta') = dI_i(\lambda) \cdot \exp[-\alpha(\lambda) \cdot D(\theta, \theta')] \quad (3-181)$$

where

$$\alpha(\lambda) = \frac{4\pi\kappa(\lambda)}{\lambda} \quad (3-182)$$

is the absorption coefficient of the medium and $\kappa(\lambda)$ the imaginary part of the complex refractive index $N(\lambda)$ of the medium. The distance between the light source and the centre of volume $dV(\theta, \theta')$ is denoted by $D(\theta, \theta')$ in (3-181) and is given by:

$$D(\theta, \theta') = \left(1 + \cos \theta - \frac{\sin \theta}{\tan \theta'} \right) \cdot R \quad (3-183)$$

The irradiance $d\hat{I}_s(\lambda, \theta, \theta', d_p)$ scattered by a single spherical particle of effective diameter d_p contained in $dV(\theta, \theta')$ is the first parameter of the scattered Stokes vector \mathbf{S}_s that is normally derived by (3-165) for the far-field region. However, the outcome of (3-165) does not take into account the attenuation of scattered irradiance in the simple medium, nor does it include the effect of irradiance modulation by the spectral sensitivity function $\text{rec}(\lambda)$ at the detector. Therefore, a more accurate formula for the irradiance intercepted by the detector is the following:

$$d\hat{I}_s(\lambda, \theta, \theta', d_p) = \frac{dI_i(\lambda, \theta, \theta')}{k(\lambda)^2 R'(\theta, \theta')^2} \cdot [S_{11}(\lambda, \theta', d_p) + S_{12}(\lambda, \theta', d_p) \cdot \frac{Q_i}{I_i}] \cdot \exp[-\alpha(\lambda) \cdot R'(\theta, \theta')] \cdot \text{rec}_n(\lambda) \quad (3-184)$$

where $\text{rec}_n(\lambda)$ is the normalised spectral sensitivity function and $k(\lambda)$ is the real wave number of the plane waves of effective wavelength λ propagating in the medium:

$$k(\lambda) = \frac{2\pi n(\lambda)}{\lambda} \quad (3-185)$$

with $n(\lambda)$ being the real part of the complex refractive index $N(\lambda)$ of the medium. Also, the distance $R'(\theta, \theta')$ between the centre of volume $dV(\theta, \theta')$ and the light detector is given by:

$$R'(\theta, \theta') = \frac{\sin \theta}{\sin \theta'} \cdot R \quad (3-186)$$

Moreover, the scattering coefficients $S_{11}()$ and $S_{12}()$ in (3-184) are given by equations (3-167) and (3-168). Finally, the overall analytic expression of $d\hat{I}_s(\lambda, \theta, \theta', d_p)$ is obtained by equations (3-180) to (3-186) and takes the form:

$$d\hat{I}_s(\lambda, \theta, \theta', d_p) = \frac{\lambda^2 \text{src}_n(\lambda) \text{rec}_n(\lambda) \cdot (\sin \theta')^2}{4\pi^2 R^2 n(\lambda)^2 (\sin \theta)^2} \cdot [S_{11}(\lambda, \theta', d_p) I_i + S_{12}(\lambda, \theta', d_p) Q_i] \cdot \exp\left[-\frac{4\pi R \kappa(\lambda)(1 + \cos \theta)}{\lambda}\right] \cdot \exp\left[-\frac{4\pi R \kappa(\lambda) \tan(\theta'/2) \sin \theta}{\lambda}\right] \cdot d\lambda \quad (3-187)$$

where the product of the normalised spectral intensity and sensitivity functions is given with respect to $\text{src}(\lambda)$ and $\text{rec}(\lambda)$ by:

$$\text{src}_n(\lambda)\text{rec}_n(\lambda) = \frac{\text{src}(\lambda)\text{rec}(\lambda)}{\int_{\lambda_{\min}}^{\lambda_{\max}} \text{src}(\lambda)\text{rec}(\lambda)d\lambda} \quad (3-188)$$

Due to the single scattering hypothesis, the total irradiance $dI_s(\lambda, \theta, \theta', d_p)$ scattered by all particles of volume $dV(\theta, \theta')$ having effective diameter d_p is given by the irradiance scattered by a single particle of size d_p , $d\hat{I}_s(\lambda, \theta, \theta', d_p)$, times the number of these particles in the infinitesimal suspension volume, i.e.:

$$dI_s(\lambda, \theta, \theta', d_p) = dN_p(\theta, \theta', d_p) \cdot d\hat{I}_s(\lambda, \theta, \theta', d_p) \quad (3-189)$$

The volume $dV(\theta, \theta')$, however, is given by the following expression:

$$dV(\theta, \theta') = \frac{\pi w^2}{4} \cdot dz(\theta, \theta') \quad (3-190)$$

while the infinitesimal height $dz(\theta, \theta')$ of that volume is given by:

$$dz(\theta, \theta') = \frac{2 \cdot \sin(\theta) \cdot \sin(d\theta')}{\cos(d\theta') - \cos(2\theta')} \cdot R \quad (3-191)$$

Due to the fact that $d\theta'$ is infinitesimal, the following two approximations apply:

$$\sin(d\theta') \cong d\theta' \quad (3-192)$$

$$\cos(d\theta') \cong 1 \quad (3-193)$$

Therefore equation (3-191) can be simplified to:

$$dz(\theta, \theta') \cong \frac{R \sin \theta}{(\sin \theta')^2} \cdot d\theta' \quad (3-194)$$

The overall analytic expression of $dN_p(\theta, \theta', d_p)$ is obtained by equations (3-172), (3-179), (3-190) and (3-194) and takes the form:

$$dN_p(\theta, \theta', d_p) = \frac{3w^2 R C_m p_{d_p}(d_p) \sin \theta}{2d_p^3 d_m (\sin \theta')^2} \cdot dd_p \cdot d\theta' \quad (3-195)$$

Finally, equations (3-187), (3-189) and (3-195) yield the infinitesimal total irradiance:

$$dI_s(\lambda, \theta, \theta', d_p) = \frac{3C_m w^2 I_i}{8\pi^2 d_m R \sin \theta} \cdot \left[S_{11}(\lambda, \theta', x) + S_{12}(\lambda, \theta', x) \frac{Q_i}{I_i} \right] \cdot \frac{\text{src}_n(\lambda)\text{rec}_n(\lambda)\lambda^2}{n(\lambda)^2} \cdot \exp\left[-\frac{4\pi R \kappa(\lambda)(1 + \cos \theta)}{\lambda}\right] \cdot \frac{p_{d_p}(x)}{x^3} \cdot \exp\left[-\frac{4\pi R \kappa(\lambda) \tan(\theta'/2) \sin \theta}{\lambda}\right] \cdot dd_p \cdot d\theta' \cdot d\lambda \quad (3-196)$$

The total irradiance $I_s(\theta)$ received by a detector placed at an angle θ other than 0 or π radians with respect to the direction of propagation of emitted light is obtained by triple integration of (3-196) over the suspended particle size range, the emitted light wavelength range and the angle θ' range as determined by the intersection of the total illuminated suspension volume with the detector's detection cone, i.e.:

$$\theta'_{\min}(\theta) = \begin{cases} \theta - \delta/2 & \theta \geq \delta \\ \theta/2 & \theta < \delta \end{cases} \quad (3-197)$$

$$\theta'_{\max}(\theta) = \begin{cases} \theta + \delta/2 & \theta \leq \pi - \delta \\ (\pi + \theta)/2 & \theta > \pi - \delta \end{cases} \quad (3-198)$$

Therefore $I_s(\theta)$ is given by the expression:

$$I_s(\theta) = \frac{3C_m w^2 I_i}{8\pi^2 d_m R \sin \theta} \cdot \int_{\lambda_{\min}}^{\lambda_{\max}} \frac{\text{src}_n(\lambda) \text{rec}_n(\lambda) \lambda^2}{n(\lambda)^2} \cdot \exp\left[-\frac{4\pi R \kappa(\lambda)(1 + \cos \theta)}{\lambda}\right] \cdot \int_{\theta'_{\min}(\theta)}^{\theta'_{\max}(\theta)} \exp\left[-\frac{4\pi R \kappa(\lambda) \tan(\theta'/2) \sin \theta}{\lambda}\right] \cdot \int_{d_{\min}}^{d_{\max}} \frac{p_{d_p}(x)}{x^3} \cdot [S_{11}(\lambda, \theta', x) + S_{12}(\lambda, \theta', x) \frac{Q_i}{I_i}] dx d\theta' d\lambda \quad (3-199)$$

The irradiance measured by a detector placed at the *line of sight* (LOS), i.e. opposite to the nephelometric light source, can be approximated by integration of (3-51) over the emitted light wavelength range $[\lambda_{\min}, \lambda_{\max}]$ for emitted irradiance $I_0=I_i$, absorption coefficient $\alpha(\lambda)$ given by (3-182) and travelled distance $z=2R$. Thus:

$$I_{LOS} \cong I_i \int_{\lambda_{\min}}^{\lambda_{\max}} \text{src}_n(\lambda) \text{rec}_n(\lambda) \exp\left[-\frac{8\pi R \kappa(\lambda)}{\lambda}\right] d\lambda \quad (3-200)$$

This approximation is accurate if the light extinction (i.e., light absorption and scattering) effects caused by particles contained in the illuminated suspension volume are small. That assumption is valid for all tenuous suspensions exhibiting the single scattering property mentioned earlier, and therefore (3-200) should apply to the nephelometric model under consideration.

The ratio between scattered light at $\pi/2$ radians and received light at the LOS defines the *turbidity* T of the suspension contained in the nephelometric chamber:

$$T \equiv \frac{I_s(\pi/2)}{I_{LOS}} \quad (3-201)$$

This definition of turbidity agrees with the non-calibrated one given by the ISO 7027 [21] for $\lambda=860\pm 30$ nm, $\delta=25\pm 5$ degrees, $R=35\pm 25$ mm and $w\leq 0.05 R$. The turbidity is often used as a quick measure of the suspension clarity and expressed in formazine nephelometric units (FNUs). The FNU is defined as the 1/400 of the turbidity of a prototype formazine ($C_6H_{12}N_4$ and $N_2H_6SO_4$ in water) solution (see [21]) measured by an arbitrary instrument (the turbidity meter).

The description and operation modelling of a polar nephelometer provided in this section are sufficient to allow the theoretical acquisition of nephelometric scattering and turbidity measurements for a well-defined particle suspension that fulfils the modelling requirements. §3.6 gives an example of how such a procedure can be applied by considering tenuous suspensions of Arizona Fine dust in water.

3.6 Scattering and turbidity of ISO Arizona Fine dust

The Arizona Fine (AF) dust is one of the four grades of test dusts specified by the ISO 12103-1 standard [62]. This kind of dust is composed of roughly textured, randomly shaped particles spanning a size range of 0-120 μm . Nevertheless, it will be assumed for the sake of simplicity that the scattering properties of this material can be sufficiently approximated by considering its particles as spherical and homogeneous. Therefore the Mie scattering theory and the associated polar nephelometric model explained in §3.4 and §3.5 can be applied to water suspensions of the dust subject to previous knowledge of chemical composition, refractive index, mass density and particle size distribution of AF. The remaining of this section deals with the aforementioned preconditions first and the application of the polar nephelometric model next.

Table 3-1 (page 138) summarises the chemical composition of AF that is provided by the ISO 12103-1 standard. The dust is composed of eight inorganic compounds, namely SiO_2 (silica), Al_2O_3 (alumina), Fe_2O_3 (rust), Na_2O (soda), CaO (lime), MgO (magnesia), TiO_2 (titania) and K_2O (potash). The actual mass fraction of any of these compounds in AF varies between dust samples as described by the ISO standard and the third column of table 3-1. However, it would be practical for the purposes of this study to lock the mass fractions of AF compounds to the average values listed in the fourth column of the table. The mass fraction values are necessary to determine the concentration of each AF compound for a given concentration C_m of bulk AF dust.

Each AF compound has its own refractive index profile over the domain of visual and near-infrared frequencies (i.e., 390-940 nm [56]). These profiles were con-

structed by interpolation of refractive index values extracted from a variety of sources. Refractive indices for SiO_2 and TiO_2 were obtained from the literature [173], namely tables X and XIV, pages 760 and 800, respectively. Alumina and magnesia have their refractive indices mentioned in book [174], tables I and II, pages 770 and 950, respectively. The refractive index values of monodispersed hematite hydrosols listed in table I on page 1625 of paper [175] were extrapolated to construct the refractive index profile of Fe_2O_3 . Extrapolation of refractive index values mentioned in references [176] and [177], pages 814 and 287, gave the profiles of soda and lime, respectively. Finally, refractive index values for K_2O could not be found anywhere in the scientific literature due to the fact that the compound concerned is unstable. Therefore it will be assumed that potash and soda have identical refractive indices throughout this study. The assumption is fair because the two compounds have similar electromagnetic properties at radio frequencies (see book [178], Chapter IV).

The refractive index profile of water at visual/near-infrared frequencies is also of importance since the material is used as dissolver in all AF suspensions under consideration. This profile is obtained from table I, page 1071 of [174].

The real and imaginary parts of the refractive indices of water and all AF compounds in visual and near-infrared frequencies are presented in figure 3-6 and figure 3-7 respectively. Rust and titania have the biggest real refractive indices, followed by the group of lime, alumina and magnesia, followed by the set of soda, potash and silica, and leaving water with the smallest real refractive index. As far as the imaginary refractive indices are concerned, rust is dominant with titania and magnesia following suit but restricted to the upper visual frequency range only (390-550 nm). Water exhibits the smallest imaginary refractive index both in relative and absolute terms. Therefore any light absorption within an illuminated AF suspension by water alone is expected to be negligible.

The bulk mass density of AF dust is given by the ISO 12103-1 standard to be 900 kg/m^3 . However, the mass densities of the eight chemical compounds the AF dust is composed of are not given by the standard. This study assumes that the bulk mass density is approximately equal to the mass densities of the compounds. Therefore the bulk mass concentration of AF divided by the bulk mass density of the dust and multiplied by the mass fraction of an AF compound should equal the *volume concentration* C_v of that compound to the water suspension under consideration.

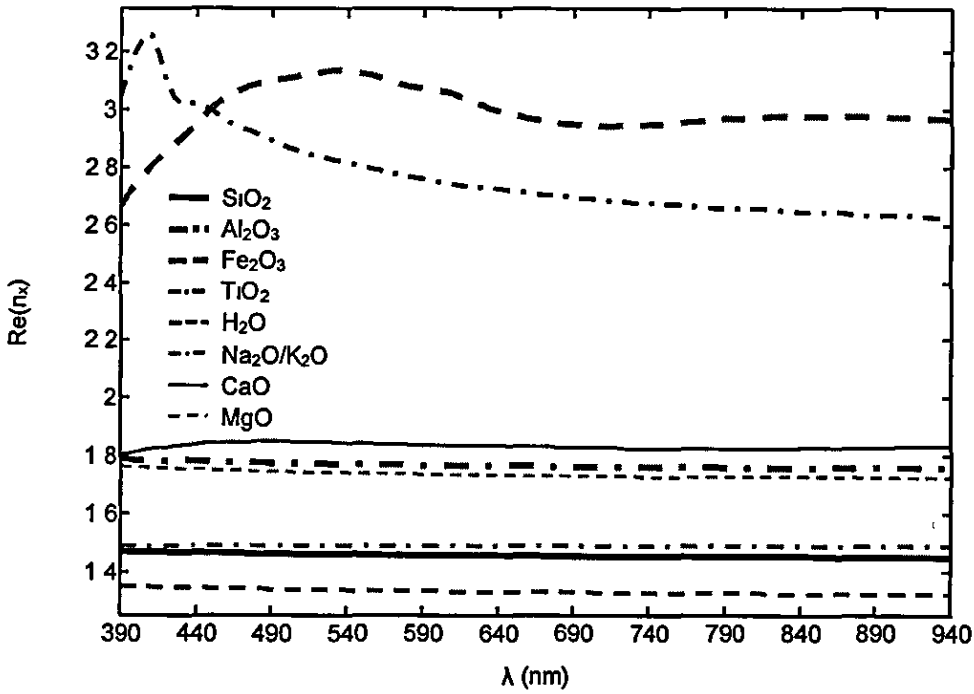


Figure 3-6 – Real refractive index profiles of AF compounds and water

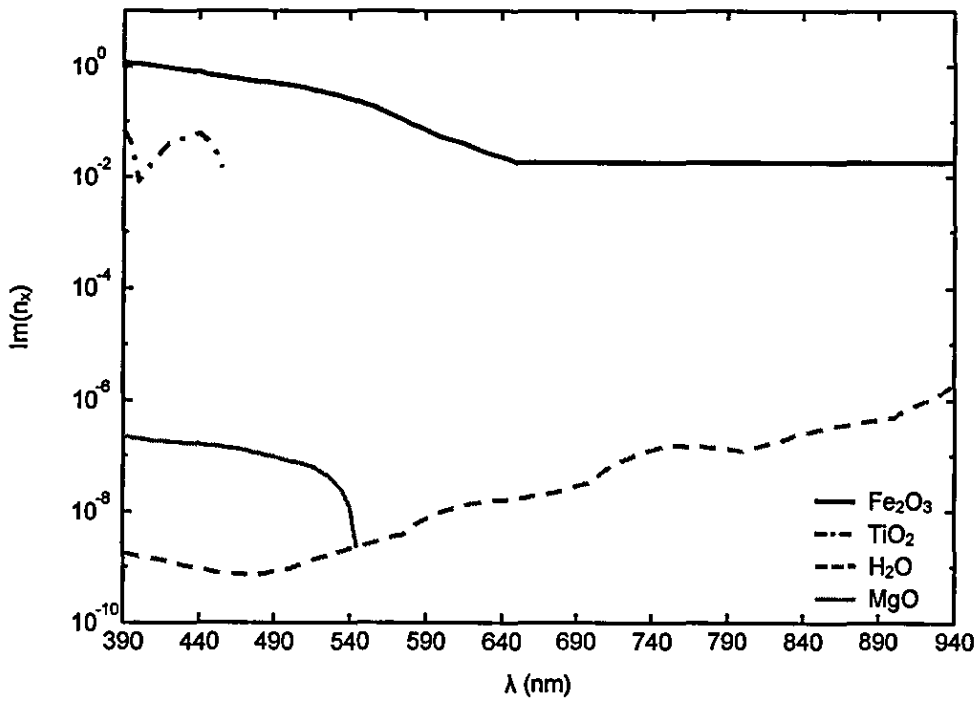


Figure 3-7 – Imaginary refractive index profiles of AF compounds and water

Table 3-1 – Chemical content of Arizona Fine dust

Chemical	Empirical Name	Mass Fraction (%)	Avg. Mass Fraction (%)
SiO ₂	Silica	68-76	72
Al ₂ O ₃	Alumina	10-15	12.5
Fe ₂ O ₃	Rust	2-5	3.5
Na ₂ O	Soda	2-4	3
CaO	Lime	2-5	3.5
MgO	Magnesia	1-2	1.5
TiO ₂	Titania	0.5-1	0.5
K ₂ O	Potash	2-5	3.5

Table 3-2 – Particle size and volume fraction details of Arizona Fine dust

Particle Size Range (µm)	Cumulative Volume Fraction (%)	Volume Fraction Distribution (%)	Average V.F. Distribution (%)
0-1	2.5-3.5	2.5-3.5	3
1-2	10.5-12.5	7-10	8.5
2-3	18.5-22	6-11.5	8.75
3-4	25.5-29.5	3.5-11	7.25
4-5	31-36	1.5-10.5	6
5-7	41-46	5-15	10
7-10	50-54	4-13	8.5
10-20	70-74	16-24	20
20-40	88-91	14-21	17.5
40-80	99.5-100	8.5-12	10.25
80-120	100	0-0.5	0.25

The particle size probability density function (pdf) of each AF compound cannot be uniquely determined by data included in the ISO 12103-1 standard. However, a single particle size pdf *can* be extracted from these data, and that pdf will apply to all AF compounds collectively as well as each of them *separately* by assumption. The pdf sought can be derived from the volume fraction distribution of the binned (by particle size) AF that is listed in the first and third columns of table 3-2. That distribution is obtained directly from the cumulative volume fraction distribution function mentioned in the ISO standard and repeated in the first two columns of table 3-2. The volume fraction distribution of table 3-2 is ambiguous however, as it allows the volume fraction of every AF particle size bin considered to vary within well-described limits. Therefore a proper mathematical model should be built that is capable of transforming the ambiguous volume fraction distribution to a set of well-defined volume fraction distribution functions. The description of such a model readily follows.

Let $\{ {}_k v f_1, {}_k v f_2, \dots, {}_k v f_{N_b} \}$ be a set of volume fractions associated with the N_b AF particle size bins and satisfying the ambiguous distribution of the first and third columns of table 3-2 for every $k \geq 1$. The elements of that set can be tabulated to form a *volume fraction distribution function* \mathbf{VF}_k defined by:

$$\mathbf{VF}_k \equiv [{}_k v f_1 \quad \dots \quad {}_k v f_i \quad \dots \quad {}_k v f_{N_b}]^T \quad (3-202)$$

where:

$$\sum_{i=1}^{N_b} {}_k v f_i = 1 \quad (3-203)$$

by definition of the notion of volume fraction.

The i_{th} element of \mathbf{VF}_k ($1 \leq i \leq N_b$) is actually a bounded variable with minimum and maximum values being the volume fraction limits that are listed in the i_{th} row (excluding the heading row) and third column of table 3-2. These limits can be tabulated to form vectors \mathbf{VF}_{min} and \mathbf{VF}_{max} as follows:

$$\mathbf{VF}_{\text{min}} \equiv \left[\min({}_k v f_1) \quad \dots \quad \min({}_k v f_i) \quad \dots \quad \min({}_k v f_{N_b}) \right]^T \quad (3-204)$$

$$\mathbf{VF}_{\text{max}} \equiv \left[\max({}_k v f_1) \quad \dots \quad \max({}_k v f_i) \quad \dots \quad \max({}_k v f_{N_b}) \right]^T \quad (3-205)$$

It can be shown that the volume fraction distribution function \mathbf{VF}_k defined by (3-202) and satisfying (3-203) can be generated by the following formula:

$$\mathbf{VF}_k = \mathbf{VF}_m + \mathbf{A}_k \circ \mathbf{VF}_s \quad (3-206)$$

where the vector \mathbf{VF}_m defined by:

$$\mathbf{VF}_m \equiv \frac{1}{2} \cdot (\mathbf{VF}_{max} + \mathbf{VF}_{min}) \quad (3-207)$$

is the *average volume fraction distribution*, and the vector \mathbf{VF}_s defined by:

$$\mathbf{VF}_s \equiv \frac{1}{2} \cdot (\mathbf{VF}_{max} - \mathbf{VF}_{min}) \quad (3-208)$$

is the *volume fraction deviation*. The average volume distribution \mathbf{VF}_m happens to satisfy (3-203) for the values of \mathbf{VF}_{min} and \mathbf{VF}_{max} assumed before, thus it is an acceptable volume fraction distribution function and will be alternatively denoted by \mathbf{VF}_0 . The values of \mathbf{VF}_m are listed in the fourth column of table 3-2.

The vector \mathbf{A}_k in (3-206) is called the *volume fraction distribution generator* and is defined by:

$$\mathbf{A}_k \equiv 2 \cdot {}^1\mathbf{U}_k - 1 \quad (3-209)$$

where the $N_b \times 1$ vector ${}^1\mathbf{U}_k$ has random numbers uniformly distributed in the range [0, 1] as elements. The volume fraction distribution generator must always satisfy the condition:

$$\mathbf{A}_k^T \cdot \mathbf{VF}_s = 0 \quad (3-210)$$

so that the associated volume fraction distribution function always satisfies (3-203). If the definition ${}^1\mathbf{U}_0 \equiv 0.5 \mathbf{1}$ is added, (3-209) produces $\mathbf{A}_0 \equiv \mathbf{0}$ and (3-206) generates \mathbf{VF}_0 .

Let the volume occupied by an AF particle that belongs to the i th particle size bin be denoted by ${}_pV_i$ for $1 \leq i \leq N_b$. That particle volume belongs to the interval $[\min({}_pV_i), \min({}_pV_i) + {}_p\Delta V_i]$, where:

$${}_p\Delta V_i \equiv \max({}_pV_i) - \min({}_pV_i) \quad (3-211)$$

The limits $\min({}_pV_i)$ and $\max({}_pV_i)$ are derived from the associated particle size limits of the i th bin $\min({}_p d_i)$ and $\max({}_p d_i)$ by the equation:

$$V_p = \frac{\pi d_p^3}{6} \quad (3-212)$$

The particle volume ranges ${}_p\Delta V_i$ defined by (3-211) can be tabulated to form the *binned particle volume range vector* $\Delta \mathbf{V}_p$:

$$\Delta \mathbf{V}_p \equiv [{}_p\Delta V_1 \quad \cdots \quad {}_p\Delta V_i \quad \cdots \quad {}_p\Delta V_{N_b}]^T \quad (3-213)$$

The probability that an AF particle volume $V_p \in [\min({}_pV_i), \min({}_pV_i) + {}_p\Delta V_i]$ and satisfies the volume fraction distribution function VF_k is equal to ${}_k v f_i$ and proportional to the particle volume range of the i th bin, i.e., ${}_p \hat{p}_i \cdot {}_p \Delta V_i$. Therefore the *binned volume probability density function* ${}_p \hat{\mathbf{P}}_k$, defined mathematically by:

$${}_p \hat{\mathbf{P}}_k \equiv [{}_p \hat{p}_1 \quad \cdots \quad {}_p \hat{p}_i \quad \cdots \quad {}_p \hat{p}_{N_b}]^T \quad (3-214)$$

is obtained by the formula:

$${}_p \hat{\mathbf{P}}_k = \text{VF}_k / \Delta V_p \quad (3-215)$$

where the symbol / denotes element-by-element vector division. The elements of ${}_p \hat{\mathbf{P}}_k$ are constant throughout the range of the particle size bin they refer to.

The *binned particle size probability density function* ${}_d \hat{\mathbf{P}}_k$, defined by:

$${}_d \hat{\mathbf{P}}_k \equiv [{}_d \hat{p}_1(\cdot) \quad \cdots \quad {}_d \hat{p}_i(\cdot) \quad \cdots \quad {}_d \hat{p}_{N_b}(\cdot)]^T \quad (3-216)$$

is obtained from ${}_p \hat{\mathbf{P}}_k$ by application of the particle diameter-volume relation:

$$d_p = \left(\frac{6V_p}{\pi} \right)^{\frac{1}{3}} \quad (3-217)$$

to the probability density function transformation equation:

$$\text{pdf}_y(y_s) = \sum_i \left[\frac{dg(x_i)}{dx_i} \right]^{-1} \cdot \text{pdf}_x(x_i) \quad (3-218)$$

where x_i denotes a root of $y=g(x)$ at $y=y_s$. Hence:

$${}_d \hat{p}_i(d_p) = \frac{\pi d_p^2}{2} \cdot {}_p \hat{p}_i \quad (3-219)$$

i.e. the elements of ${}_d \hat{\mathbf{P}}_k$ follow parabolic trajectories within the limits of the particle size bin they refer to. Besides, the binned particle size pdf satisfies the condition:

$$\sum_{i=1}^{N_b} \int_{\min(d_p)}^{\max(d_p)} {}_d \hat{p}_i(x) dx = 1 \quad (3-220)$$

for every $k \geq 0$ due to equations (3-203) and (3-219).

The binned particle size and volume pdf of the Arizona Fine dust for $0 \leq k \leq 200$ are illustrated in figure 3-8 and figure 3-9 at logarithmic coordinates for both axes.

The solid lines in these graphs represent ${}_{d_p} \hat{P}_0$ and ${}_{V_p} \hat{P}_0$ respectively, while the dashed and dash-dotted lines describe the upper and lower limits of grey-shaded areas within which the remaining 200 *dithered* versions of binned AF pdf, i.e. ${}_{d_p} \hat{P}_k$ and ${}_{V_p} \hat{P}_k$ for $1 \leq k \leq 200$, could have been drawn.

It is apparent from figure 3-8 and figure 3-9 that the binned probability density functions are very unlikely to be realisable due to their sharp discontinuities at particle size bin limits. Therefore two new sets of functions should be derived from ${}_{d_p} \hat{P}_k$ and ${}_{V_p} \hat{P}_k$ that satisfy the unit area condition of a typical pdf, are good approximations of the corresponding binned sets of functions, and are continuous, and are continuous throughout their domains. These new sets will be called the *particle size probability density function* and *volume probability density function* of AF dust, and will be denoted by the symbols ${}_{d_p}^k p(d_p)$ and ${}_{V_p}^k p(V_p)$ respectively with $k \geq 0$.

There are two favourite alternatives to the design of the AF particle size and volume pdf sets from the associated binned pdf sets. The first samples ${}_{d_p} \hat{P}_k$ at the centres of particle size bins and then produces an interpolated curve of unit area from these samples that extends to the whole AF particle size domain. After that, the constructed ${}_{d_p}^k p(d_p)$ is applied to (3-218) for $g(x)$ given by (3-212) in order to calculate ${}_{V_p}^k p(V_p)$. The second alternative follows the same interpolation procedure as the first but applies to ${}_{V_p} \hat{P}_k$ rather than ${}_{d_p} \hat{P}_k$. Next, the produced ${}_{V_p}^k p(V_p)$ is applied to (3-218) for $g(x)$ given by (3-219) only to obtain ${}_{d_p}^k p(d_p)$. Although the two alternatives give similar results, it is the first one that produces the smoothest and more likely to be realisable pdf curves. As a matter of fact, the first approach is the one adopted by this study for the AF particle size and the generation of volume pdf. Furthermore, the interpolation method chosen for the generation of volume pdf was linear for simplicity reasons. The produced particle size pdf and volume pdf sets are illustrated in figure 3-10 and figure 3-11 at logarithmic coordinates for both axes. The solid lines in these graphs represent ${}_{d_p}^0 p(d_p)$ and ${}_{V_p}^0 p(V_p)$ respectively, while the dashed and dash-dotted lines describe the upper and lower limits of grey-shaded areas within which the re-

maining 200 *dithered* versions of AF pdf, i.e. $d_p^k p(d_p)$ and $v_p^k p(v_p)$ for $1 \leq k \leq 200$, could have been drawn.

The scattering profile of a tenuous water suspension of Arizona Fine dust that is obtained by a typical polar nephelometer can be calculated theoretically by using equation (3-199) to find the scattering profile of each AF compound separately from the rest and summing the results afterwards. Four AF profiles were computed overall from four sets of nephelometric and AF parameters compatible with the ISO 7027 turbidity standard. The common parameters to all four AF scattering simulations are:

- $\theta = 5^\circ, 10^\circ, \dots 175^\circ$
- $\lambda = 830, 835, \dots 890 \text{ nm}$
- $I_i = 1 \text{ W/m}^2, Q_i = 0$ (unpolarised light source of unit emitting power)
- $w = 1 \text{ mm}$
- $d_m = 900 \text{ g/l}$
- $\text{src}_n(\lambda) = \text{rec}_n(\lambda) = 10^8/6$
- $N(\lambda) = n(\lambda) + \kappa(\lambda)$ and $N_1(\lambda) = n_1(\lambda) + \kappa_1(\lambda)$ as in figure 3-6 and figure 3-7
- $p_{d_p}(d_p) = d_p^0 p(d_p)$ as in figure 3-10
- Mass fraction of each AF compound as in table 3-2, fourth column

The additional parameters applying to the first simulation are:

- $C_m = 10, 20 \dots 60 \text{ mg/l}$
- $\delta = 25^\circ$
- $R = 35 \text{ mm}$
- $k = 0$ (no dithered versions of AF particle size pdf considered)

These parameters change for the second simulation to:

- $C_m = 35 \text{ mg/l}$
- $\delta = 20^\circ, 22^\circ, \dots 30^\circ$
- $R = 35 \text{ mm}$
- $k = 0$ (no dithered versions of AF particle size pdf considered)

The additional parameters of the third scattering simulation are:

- $C_m = 35 \text{ mg/l}$
- $\delta = 25^\circ$
- $R = 10, 20, \dots 60 \text{ mm}$
- $k = 0$ (no dithered versions of AF particle size pdf considered)

These parameters change for the fourth scattering simulation are:

- $C_m = 35 \text{ mg/l}$
- $\delta = 25^\circ$
- $R = 35 \text{ mm}$
- $0 \leq k < 20$ (19 dithered versions of AF particle size pdf considered)

The final scattering profiles were converted to dB units by use of the relation $f_{\text{dB}}(x) \equiv 10 \log_{10}(x)$.

Figure 3-12 illustrates the AF scattering profile for a set of small but linearly increasing values of the bulk mass concentration C_m . As easily predicted from equation (3-199), an increase of C_m causes a proportional increase of the AF profile level but does not alter its shape. The result is consistent with the assumption of single scattering made during the building phase of the polar nephelometric model.

Unlike C_m , changes of the acceptance angle δ of the nephelometric detector usually cause alterations of the AF scattering profile. This effect is observed in the results of the second simulation displayed in figure 3-13. The profile shape changes are more pronounced at the scattering angles of 20° , 140° and 160° , and can be explained by the following reasons:

- a) Greater acceptance angles can be interpreted as more AF particles "seeing" the nephelometric detector at an increasingly different number of angles. Therefore the scattering profile smoothens progressively with increasing δ due to the extension of the detector averaging of irradiances scattered from individual AF particles to a wider range of angles.
- b) The effective acceptance angle is physically limited at scattering angles near the line of sight according to equations (3-197) and (3-198). Therefore the scattering profiles obtained for different acceptance angles are expected to converge in the small and large scattering angle range.

The scattering profile level is also affected by δ , as greater values of acceptance angle result to more AF particles contributing their individual scattered irradiances to the detector and thus higher scattered irradiance overall. However, the effect of δ to the profile level is much weaker than the similar effect caused by changes to C_m .

The scattering profiles of AF for increasing values of the nephelometric chamber radius R are stacked in figure 3-14. As expected, the profile level drops propor-

tionally to R due to the existence of the $1/R$ factor in (3-199). No apparent modifications of the profile shape are caused by chamber radius variance.

The variation of a typical AF scattering profile due to particle size pdf dithering is the subject of the fourth scattering simulation. Figure 3-15 illustrates the AF scattering profile that is related to ${}_d^0 p(d_p)$ on top of the uncertainty area of that profile. The scattering profile is drawn as a solid line while the uncertainty area is shaded grey and bounded from above by a dashed line and from below by a dash-dotted line. All AF scattering profiles that correspond to ${}_d^k p(d_p)$ for $1 \leq k < 20$ could have been drawn within the uncertainty area. The maximum deviation of scattered irradiance I_s from its nominal value due to particle size pdf dithering is about $\pm 1\%$ in dB units.

The turbidity T of all water suspensions of Arizona Fine dust assumed by the last three scattering simulations can be estimated theoretically by application of all relative nephelometric and AF parameters to equation (3-199) and approximations (3-200) and (3-201). The only difference between the simulation parameters for scattering and turbidity is the particle size pdf that is now given by $p_{d_p}(d_p) = {}_d^k p(d_p)$ for $0 \leq k < 20$. This change allows the additional observation of the effect of AF particle size pdf dithering to turbidity measurements from a purely theoretical point of view. The pdf dithering effect will be visualised in the following AF turbidity plots by a pair of dashed and dash-dotted lines carving the upper and lower limits of grey-shaded areas that represent the uncertainty range of turbidity. Solid lines in these plots will only be drawn to signify turbidity values calculated for $p_{d_p}(d_p) = {}_d^0 p(d_p)$.

Figure 3-16 illustrates the turbidity profile for increasing values of the AF bulk mass concentration in the water suspension. The turbidity function that corresponds to $p_{d_p}(d_p) = {}_d^0 p(d_p)$ follows an ideal $10 \log_{10}(a_1 x + b_1)$ trajectory where $a_1 = 5.86 \cdot 10^{-12}$ and $b_1 = -1.29 \cdot 10^{-26}$. Therefore T is proportional to C_m in linear scale. The uncertainty margin of turbidity due to AF particle size pdf dithering is about $\pm 1\%$ in dB units.

The AF turbidity changes with respect to the acceptance angle of the nephelometric detector in a manner displayed by figure 3-17. The turbidity function for $p_{d_p}(d_p) = {}_d^0 p(d_p)$ is an approximate $10 \log_{10}(a_2 x + b_2)$ curve where $a_2 = 8.02 \cdot 10^{-12}$ and $b_2 = 4.21 \cdot 10^{-12}$. Hence T is proportional to δ in non-logarithmic units for a relative

approximation error of nearly 1%. The uncertainty margin of turbidity due to AF particle size pdf dithering is again close to $\pm 1\%$ in dB units.

Finally, the turbidity profile for increasing values of the nephelometric chamber radius is presented in figure 3-18. The turbidity function for $p_{d_p}(d_p) = d_p^0 p(d_p)$ is approximately equal to $10 \log_{10}(a_3/x+b_3)$ with $a_3 = 7.94 \cdot 10^{-9}$ and $b_3 = -2.20 \cdot 10^{-11}$. Thus T is inversely proportional to R in non-logarithmic units for a relative approximation error of almost 1%. The uncertainty margin of turbidity due to AF particle size pdf dithering is once again near $\pm 1\%$ in dB units.

The modelling procedure and results obtained so far for the physical and optical properties of Arizona Fine dust related to Mie scattering and the polar nephelometer are useful for two reasons. Firstly, they demonstrate how the Mie scattering theory can be exploited to first-degree estimations of the scattering and turbidity of physically realisable particle suspensions. Secondly, they provide the necessary background for the estimation of light scattered by tenuous water suspensions of arbitrarily *filtered* AF dust derivatives. Chapter 4 utilises that background to estimate the scattering profiles of filtered AF dust suspensions and attempt to train and optimise a number of ANNs with them.

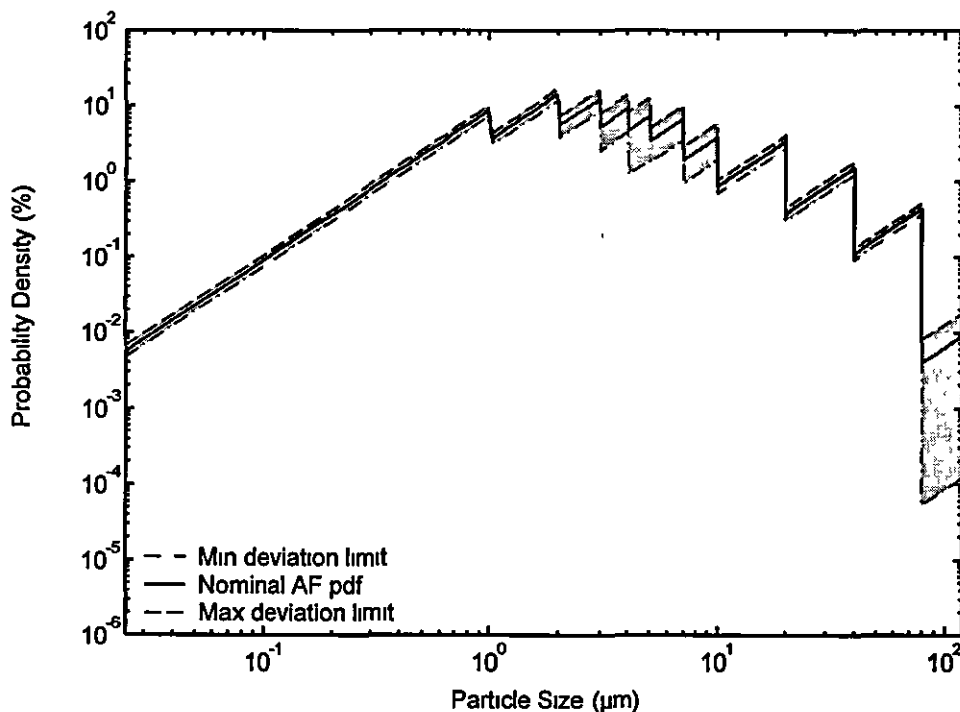


Figure 3-8 -- Binned size probability density function for Arizona Fine

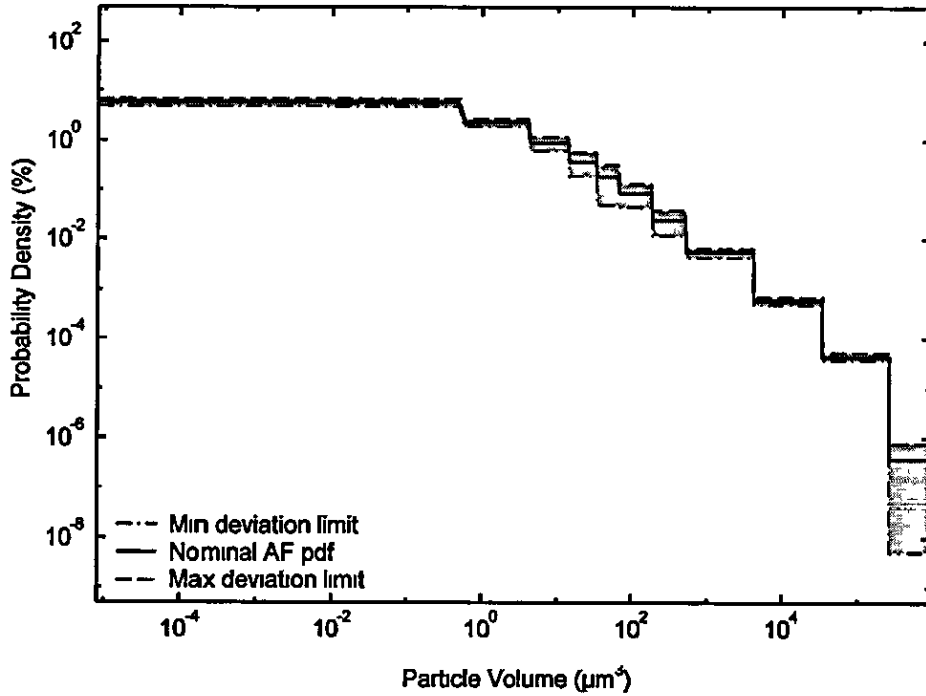


Figure 3-9 – Binned volume probability density function for Arizona Fine

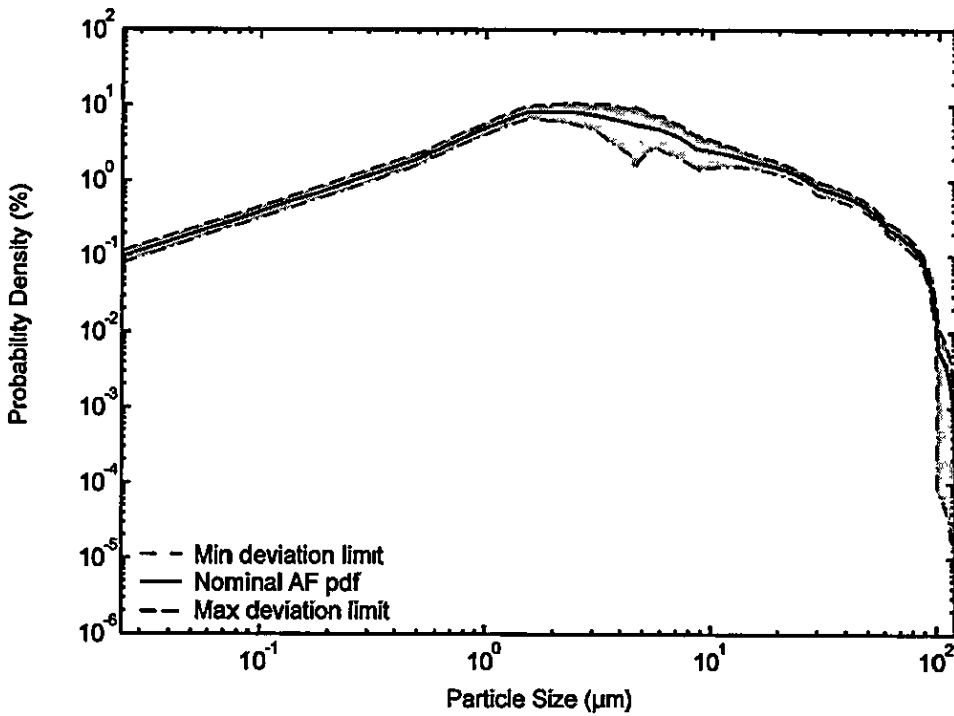


Figure 3-10 – Particle size probability density function for Arizona Fine

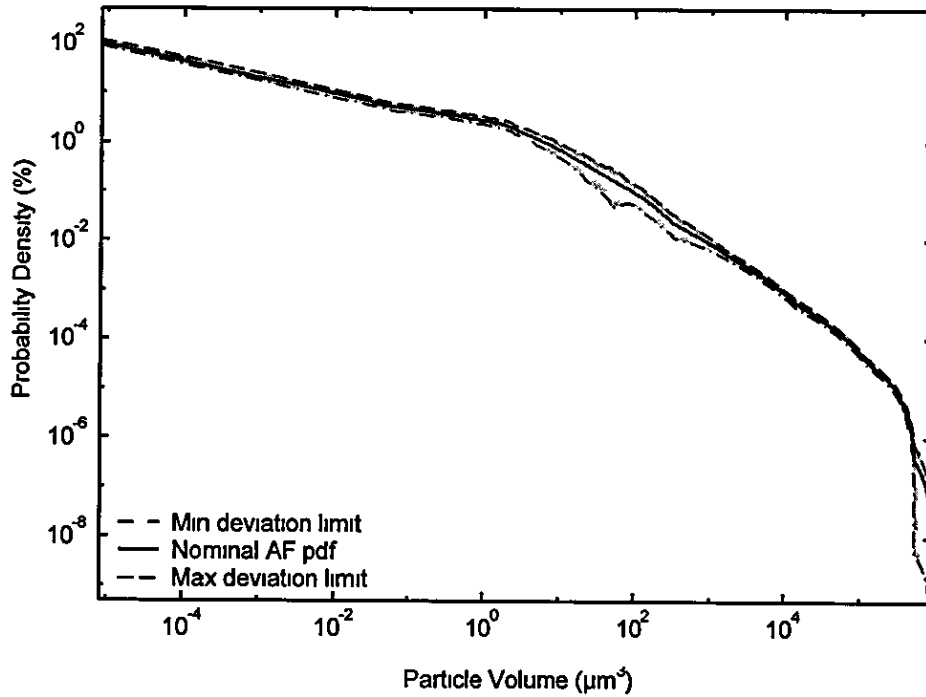


Figure 3-11 – Particle volume probability density function for Arizona Fine

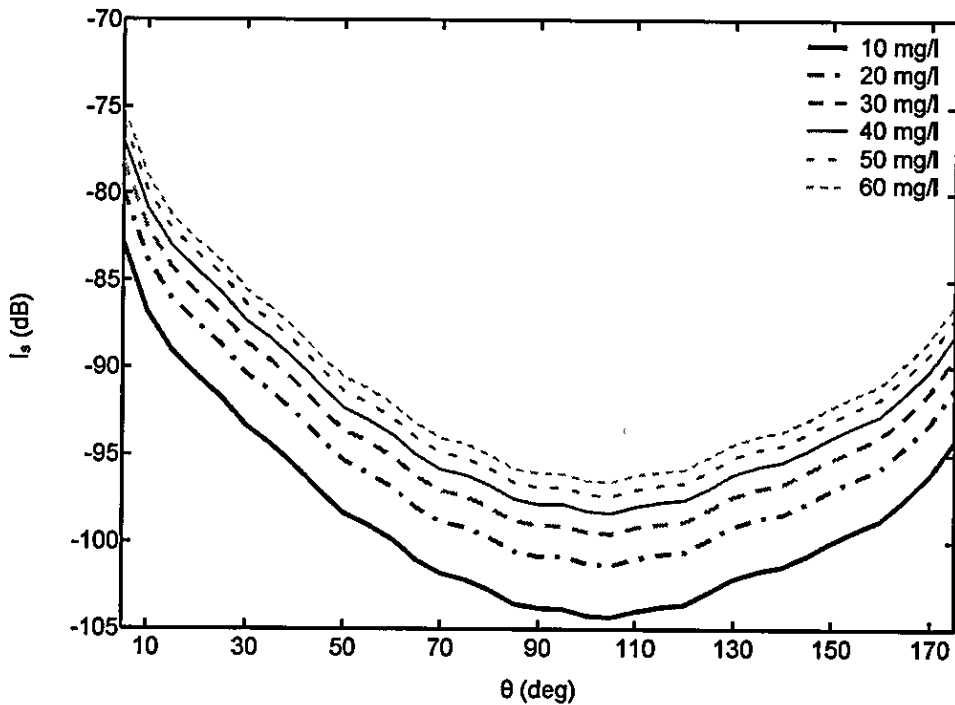


Figure 3-12 – Scattered irradiance per bulk mass concentration of Arizona Fine

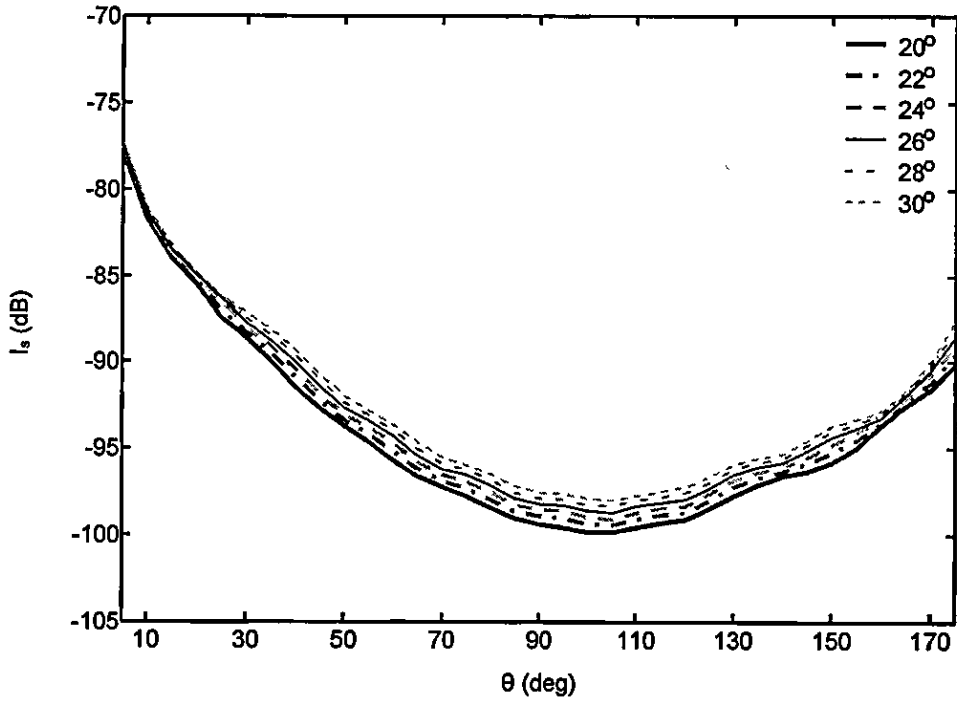


Figure 3-13 – Scattered irradiance per acceptance angle for Arizona Fine

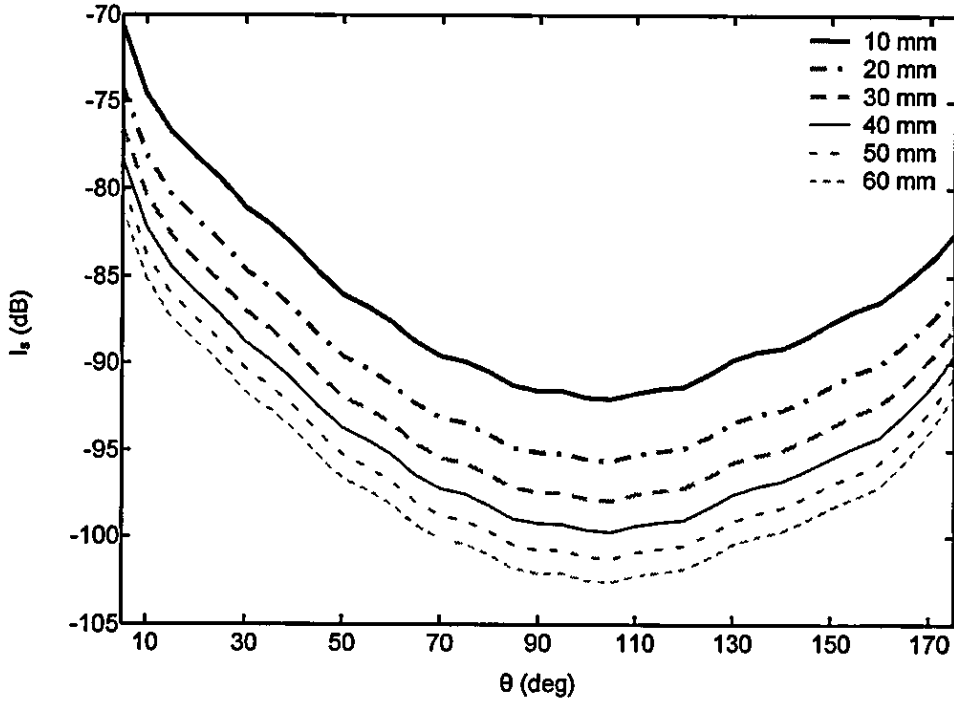


Figure 3-14 – Scattered irradiance per chamber radius for Arizona Fine

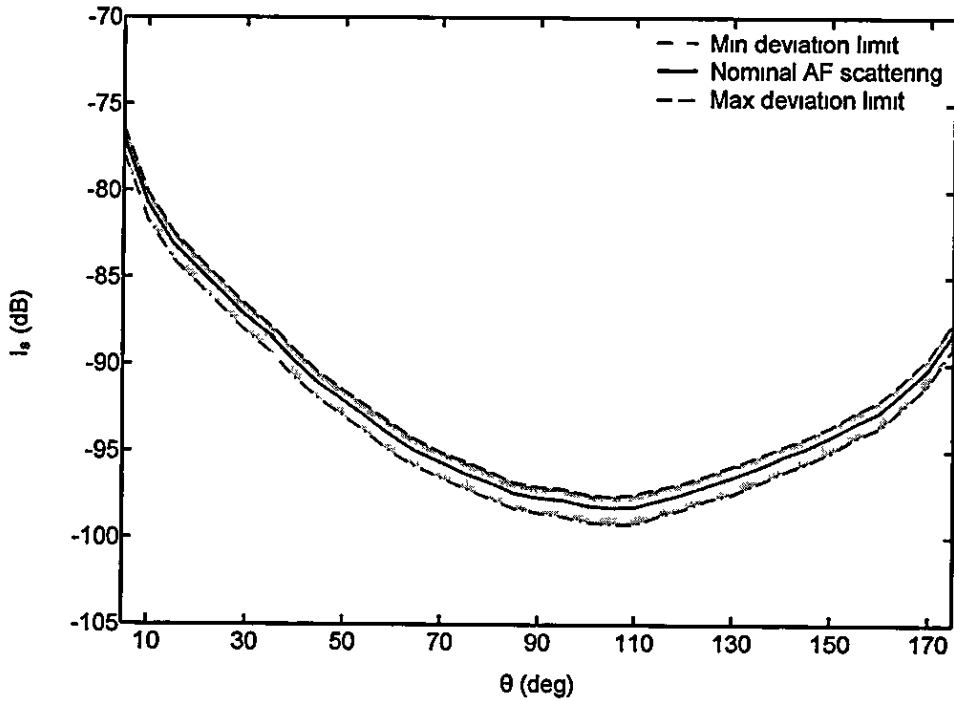


Figure 3-15 – Scattered irradiance variation due to Arizona Fine particle size probability density function dithering

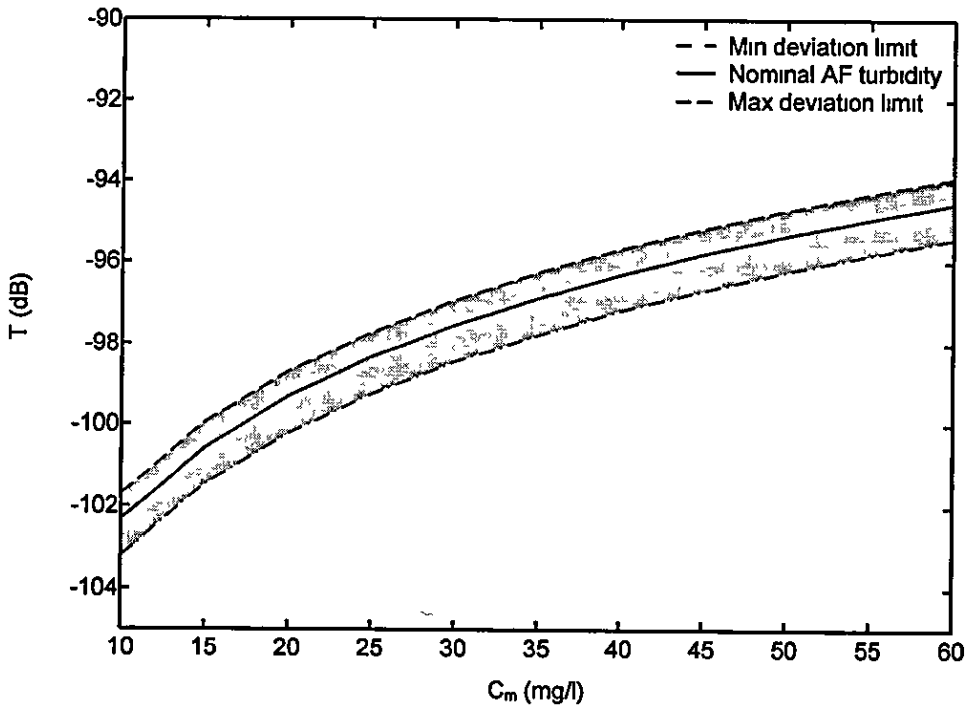


Figure 3-16 – Turbidity per bulk mass concentration of Arizona Fine

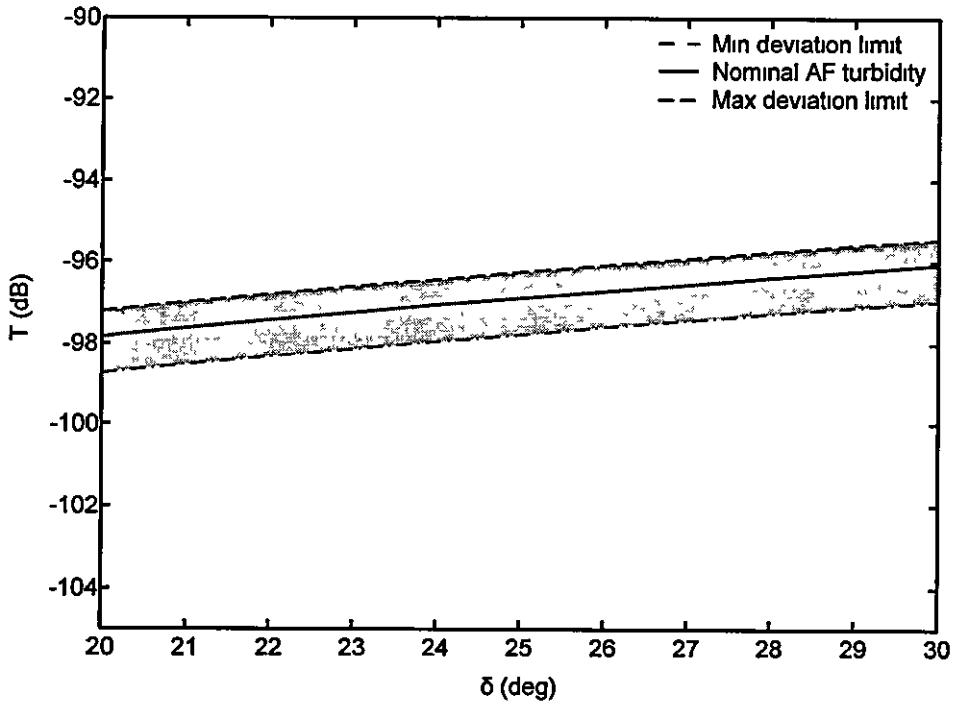


Figure 3-17 – Turbidity per acceptance angle for Arizona Fine

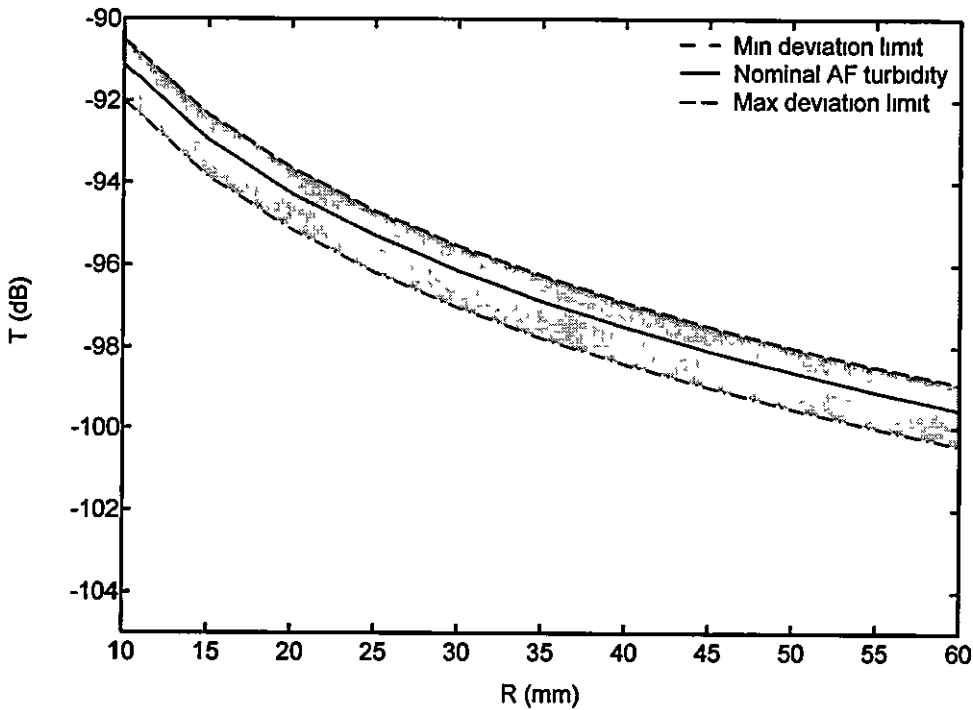


Figure 3-18 – Turbidity per chamber radius for Arizona Fine

3.7 The inverse scattering problem

The inverse light scattering problem can be stated in its most generic case as the accurate identification of a set of particles that is responsible for the creation of a scattered electromagnetic field from an incident field when partial or complete information for the scattered field is available. Examples of practical inverse scattering problems are:

- a) Description of the composition of interstellar dust from the analysis of light of various wavelengths that transverses the dust without undergoing scattering or absorption and, less frequently, light scattered in various directions by the dust
- b) Determination of the size of particles whose shape and composition is already known from light scattering techniques
- c) Exploitation of radar backscattering data to discriminate between rain and hail

The necessary information to specify a set of scattering particles uniquely is the incident field that interacts with the scattering particles the internal field that is created inside the scattering particles, and the vector amplitude and phase of the scattered field. Moreover, that information must be available at all points in space the three fields are defined [179]. However, precise knowledge of the three fields is rarely gained in practice. The incident field is the easiest to be determined but caution is required. The internal field cannot be measured directly in many practical circumstances, although under certain conditions, which are most likely to be met in the laboratory, this field can be approximated by the incident field (see [169], Chapter 6). The most difficult task though is the description of the amplitude and phase of the scattered field: although this is not impossible in principle, it is rarely achieved in practice. Therefore the inverse scattering problem is almost always an ill-posed one as less than the necessary amount information can be obtained to solve it.

Is the turbidity parameter defined in (3-201) sufficient to provide some kind of information about the scattering particles? Consider the simple case of two sets of homogeneous, spherical particles of the same material. Each set contains particles of unique size but the particle number and size differs significantly between the two sets. Specifically, the first set is made of small particles while the second one contains substantially larger particles. Small identical mass quantities are collected from the two sets and diluted in two different containers of equal volumes of the same dissolver. The two tenuous suspensions made are said to be *monodispersed* because they have

diluted particles of unique size. Turbidity measurements are obtained for the two suspensions using two identical nephelometric devices. According to (3-189), the scattered light at $\pi/2$ is proportional to the number of scattering particles and the irradiance scattered by every individual particle. The suspension of small particles owns a considerably higher amount of particles per unit volume that scatter light. On the other hand, every large particle of the alternative suspension scatters a substantially higher amount of irradiance at $\pi/2$. This effect can be adequately explained by the Mie scattering theory. A greater particle size α gives a bigger χ parameter according to (3-146) and thus an increased n_c constant due to (3-148). The higher n_c is, the more terms are required to obtain parameters S_1 and S_2 from (3-156) and (3-157) and the greater the latter set of parameters become. Finally, higher values of S_1 and S_2 yield a bigger S_{11} term by (3-167) which triggers an increase to the irradiance scattered by the particle as (3-184) suggests. According to approximation (3-200), the scattered irradiance at LOS is insensitive to the number and size of suspended particles. As a matter of fact, it is very likely that the two substantially different suspensions in terms of the number and size of suspended particles give equal or very similar turbidity values by definition (3-201). When that happens, it is virtually impossible to discriminate the two suspensions on turbidity grounds only. In conclusion, turbidity may be easy to obtain but insufficient to describe suspended particle size or related information in all but few problems [180, 181].

An interesting and still open for research question is whether partial or complete knowledge of the scattering pattern of a set of particles is sufficient to solve the inverse scattering problem. The findings up to date suggest that it is likely to gain partial information for particles of a given scattering pattern subject to previous knowledge or assumption of important particle properties by means other than light scattering techniques. More about the topic can be found in [182].

As already mentioned in chapter 1, this study is focused on the exploitation of scattering information to characterise sets of particles according to size or a derived quantity rather than describe the particles in fine detail. Chapter 4 performs the task of particle characterisation for a number of cases with the assistance of ANNs, the Mie scattering theory and the nephelometric model presented so far.

3.8 Summary

The aim of this chapter was provide the reader with the fundamentals of light scattering theory and apply the theory to a reference material described by an international standard. Aspects of the classical electromagnetic theory explaining the basic properties of light were reviewed in sufficient detail. The interaction of light with a single spherical particle placed in a simple, non-absorbing medium was rigorously examined and all fields created as a result of the interaction were analytically determined. The operation of a typical polar nephelometer was explained and a model of the device was developed to estimate the scattered irradiance and turbidity of a tenuous suspension of homogeneous particles in a simple medium. The physical and optical properties of small quantities of Arizona Fine dust diluted in water were extracted from the ISO 12103-1 standard and other sources and applied to a polar nephelometric model compatible with the ISO 7027 turbidity standard. The scattering and turbidity profiles obtained were compared to each other in order to illustrate their degree and kind of dependency on core Arizona Fine and nephelometric parameters. Finally, the inverse scattering problem was discussed, the unfeasible conditions under which the problem is guaranteed a unique solution were mentioned, and the inability of turbidity to discriminate particle sets of substantially different size was illustrated by an instructive counterexample.

CHAPTER 4

SCATTERING NEPHELOMETRY APPLICATIONS

♪...there's an ordinary world

♪ somehow I have to find...♪

(Duran Duran, Wedding album, 1993)

4.1 Introduction

The ANN modelling and light scattering theories explained in chapter 2 and chapter 3 provide the essential background for the comprehension, modelling and solution of many particle characterisation problems. This chapter utilises the aforementioned knowledge base to solve two of these problems: volume fraction estimation and suspended matter detection. Water suspensions of filtered AF sand, polychromatic light sources and polar nephelometers are assumed in both problems. The solutions obtained are structurally optimal MFANN models that require the smallest number of nephelometric detector signals possible to give the answers sought. These results are indicative of the appropriateness and usefulness of the ANN modelling approach to the field of particle characterisation as a whole.

4.2 Binned AF sand volume fraction estimation problem

This theoretical particle characterisation problem assumes that three AF sand types can be created from the separation of the ISO 12103-1 AF sand particles into three bins according to particle size (0-1 μm , 1-3 μm , 3-10 μm). A set of distinct but equiponderant mixtures of the three binned AF types is assumed to be prepared and dispersed in water. The suspensions formed are inserted in the polar nephelometric model of §3.5 and illuminated by a member of a predefined set of light sources. Finally, the theoretical scattering data calculated for every suspension are used to train, optimise and test a MFANN model that estimates the volume fraction of every binned AF type in all suspensions under consideration.

4.2.1 Theoretical sample preparation

Three virtual types of filtered AF sand are supposedly available for the preparation of sample suspensions. These types originate from the ISO 12103-1 AF sand by separating its particles in four size bins (0-1 μm , 1-3 μm , 3-10 μm and 10-120 μm) and rejecting the last bin. Hereafter the three sand types will be called *binned AF sand types* and denoted by the initials AF₁, AF₂ and AF₃.

The *particle size probability density functions* $a_p p_1(\cdot)$, $a_p p_2(\cdot)$ and $a_p p_3(\cdot)$ of the three binned AF sand types are illustrated in figure 4-1 (see page 159). These functions are derived from the particle size pdf of the ISO 12103-1 AF sand (see figure 3-10) by cutting the latter into pieces consistent with the particle size ranges of

three binned AF sand types and normalising every piece to unity with respect to integral. The unfeasibility of the derived pdfs (the shape of a feasible pdf is usually similar to a lognormal distribution curve) does not affect in any way the validity of the solution method that is explained in §4.2.3.

The bulk mass density of each of the binned AF types is assumed to be equal to the bulk mass density of the ISO 12103-1 AF sand, i.e. $d_m = 900 \text{ kg/m}^3$.

Two hundred and thirty one water suspensions of mixtures of the three binned AF sand types are hypothetically prepared for the solution of the volume fraction estimation problem. Every suspended mixture satisfies the following conditions:

- a) The contribution of every binned AF sand type to the mixture is either a multiple of 1 mg or nonexistent.
- b) The analogy of the three binned AF sand types that participate in the mixture is unique as far as the set of 231 mixtures is concerned.
- c) The total mass of the mixture is 20 mg.

The volume of water used for the dilution of the binned AF sand mixture is assumed to be $V_w = 1$ litre in every sample.

4.2.2 Nephelometric model setup

The virtual device employed to acquire the scattering profiles of the aforementioned sample suspensions is the polar nephelometric model of §3.5. The values assigned to the nephelometric parameters are the following:

- a) The chamber radius is $R = 10 \text{ mm}$.
- b) The width of the illuminating beam is $w = 0.5 \text{ mm}$.
- c) The numerical aperture of all nephelometric detectors is $NA = 0.1$. The acceptance angle δ of the detectors is related to the numerical aperture by:

$$\delta = 2 \sin^{-1} \left(\frac{NA}{n_w} \right) \quad (4-1)$$

where n_w is the real part of the complex refractive index of the sample solvent, i.e. water ($n_w \approx 4/3$). Therefore the acceptance angle is $\delta \approx 8.6^\circ$.

- d) The light source belongs to a set of three LEDs denoted by the capital letters A, B and C. The total irradiance emitted from every source is $I_{0i} = 1 \text{ W/m}^2$. The spectral intensity functions $\text{src}_i(\cdot)$ of these sources are Gaussian distribu-

tions limited to the frequency domains $[\mu_i - 5\sigma_i, \mu_i + 5\sigma_i]$ ($i = A$ for LED A, B for LED B and C for LED C), i.e.:

$$src_i(\lambda) \equiv \begin{cases} \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\lambda - \mu_i}{\sigma_i}\right)^2\right] & \mu_i - 5\sigma_i \leq \lambda \leq \mu_i + 5\sigma_i \\ 0 & \text{otherwise} \end{cases} \quad (4-2)$$

The mean value μ_i satisfies the identity:

$$src_i(\mu_i) \equiv \max_{\lambda} [src_i(\lambda)] \quad (4-3)$$

and the standard deviation σ_i is given by the formula:

$$\sigma_i = \frac{HMBW_i}{2\sqrt{2\ln 2}} \quad (4-4)$$

The variable $HMBW_i$ in equation (4-4) is the *Half Maximum Beam Width* of the i_{th} LED source and is derived from $src_i(\cdot)$ as follows:

$$HMBW_i \equiv \lambda_{2i} - \lambda_{1i} \quad (4-5)$$

$$\lambda_{1i} \leq \lambda_{2i} \quad (4-6)$$

$$src_i(\lambda_{1i}) = src_i(\lambda_{2i}) \equiv \frac{1}{2} \cdot \max_{\lambda} [src_i(\lambda)] \quad (4-7)$$

Figure 4-2 illustrates the spectral intensity functions of the three LED sources in pdf format. The mean values and Half Maximum Beam Widths of these curves are $\mu_A = 565$ nm, $\mu_B = 635$ nm, $\mu_C = 770$ nm, $HMBW_A = 25$ nm, $HMBW_B = 45$ nm and $HMBW_C = 80$ nm.

- e) All light detectors are characterised by the same spectral sensitivity function $rec(\cdot)$ that is uniform over the visible and near-infrared frequency domain (390 – 940 nm). Figure 4-3 illustrates $rec(\cdot)$ in pdf format (the total irradiance on the detector is assumed to be equal to 1 W/m^2 for graphical purposes only).

4.2.3 Theoretical results

According to the sample preparation procedure described in §4.2.1, the total mass concentration of binned AF sand is $C_m = 20$ mg/l in every sample suspension. Due to the modesty of that concentration value, the scattering conditions of §3.5 are satisfied and the analytical scattering solution derived in that section is applicable to this classification problem. As a matter of fact, the scattered irradiance $I_{sij}(\theta)$ measured by a nephelometric detector placed at angular position θ is given by:

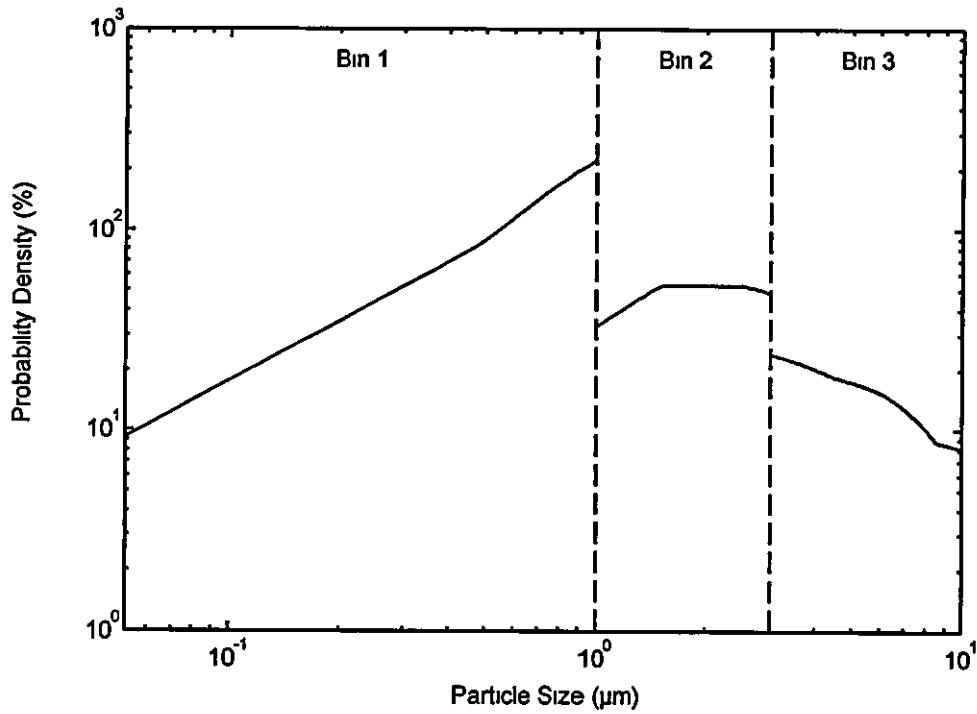


Figure 4-1 – Particle size probability density functions of $\text{AF}_{1,2,3}$

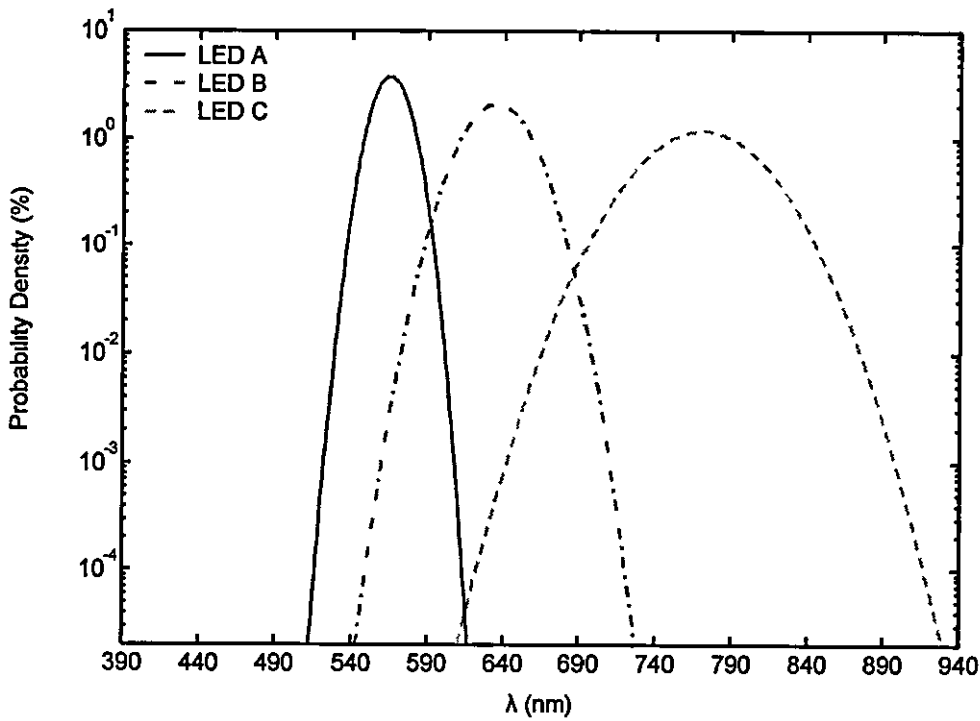


Figure 4-2 – Spectral intensity functions of the three nephelometric sources

$$\begin{aligned}
I_{sj}(\theta) = & \sum_{l=1}^3 vf_{jl} \cdot \sum_{k=1}^8 mf_k \cdot \left\{ \frac{3C_m w^2}{8\pi^2 d_m R \sin \theta} \cdot \right. \\
& \int_{\lambda_{\min}}^{\lambda_{\max}} \frac{\text{src}_{nl}(\lambda) \text{rec}_{nl}(\lambda) \lambda^2}{n_k(\lambda)^2} \cdot \exp \left[-\frac{4\pi R \kappa_k(\lambda)(1 + \cos \theta)}{\lambda} \right] \cdot \\
& \int_{\theta'_{\min}(\theta)}^{\theta'_{\max}(\theta)} \exp \left[-\frac{4\pi R \kappa_k(\lambda) \tan(\theta'/2) \sin \theta}{\lambda} \right] \cdot \\
& \left. \int_{d_{\min}}^{d_{\max}} \frac{d_p P_l(x)}{x^3} [S_{11}(\lambda, \theta', x) I_{0i} + S_{12}(\lambda, \theta', x) Q_{0i}] dx d\theta' d\lambda \right\} \quad (4-8)
\end{aligned}$$

where:

- j is the sample suspension index ($1 \leq j \leq 231$)
- The volume fraction of binned AF sand type l suspended in the j th sample, vf_{jl} , is equal to the associated mass concentration fraction cf_{jl} because the bulk mass density d_m and water volume V_w are independent of l and j . The sand mixing conditions of §4.2.1 require that $vf_{jl} = 0.05n$ where $0 \leq n \leq 20$. Also:

$$\sum_{l=1}^3 vf_{jl} \equiv 1 \quad 1 \leq j \leq 231 \quad (4-9)$$

- The set of eight (average) mass fractions mf_k that correspond to the chemical compounds of the filtered and ISO 12103-1 AF sand is stated in table 3-1.
- The complex refractive index $N_k(\lambda) = n_k(\lambda) + \kappa_k(\lambda)$ of the k -th chemical compound of the filtered AF sand is assumed to be identical to the associated index of the ISO 12103-1 AF sand. Refractive indices $n_k(\lambda)$ and $\kappa_k(\lambda)$ are displayed in figure 3-6 and figure 3-7.
- The normalised spectral functions $\text{src}_m(\lambda)$ and $\text{rec}_m(\lambda)$ are determined by figure 4-2, figure 4-3 and equation (3-188). The integration limits are determined by the second figure ($\lambda_{\min} = 390$ nm, $\lambda_{\max} = 940$ nm).
- The integration limits $\theta'_{\min}(\theta)$ and $\theta'_{\max}(\theta)$ are given by equations (3-197) and (3-198).
- The integration limits d_{\min} and d_{\max} can be extracted from figure 4-1.
- The scattering functions $S_{11}(\lambda, \theta', x)$ and $S_{12}(\lambda, \theta', x)$ are calculated exactly with the aid of Mie scattering theory (see §3.4).
- The Stokes parameter Q_{0i} is assumed to be zero because the three nephelometric sources are LEDs emitting unpolarised light.

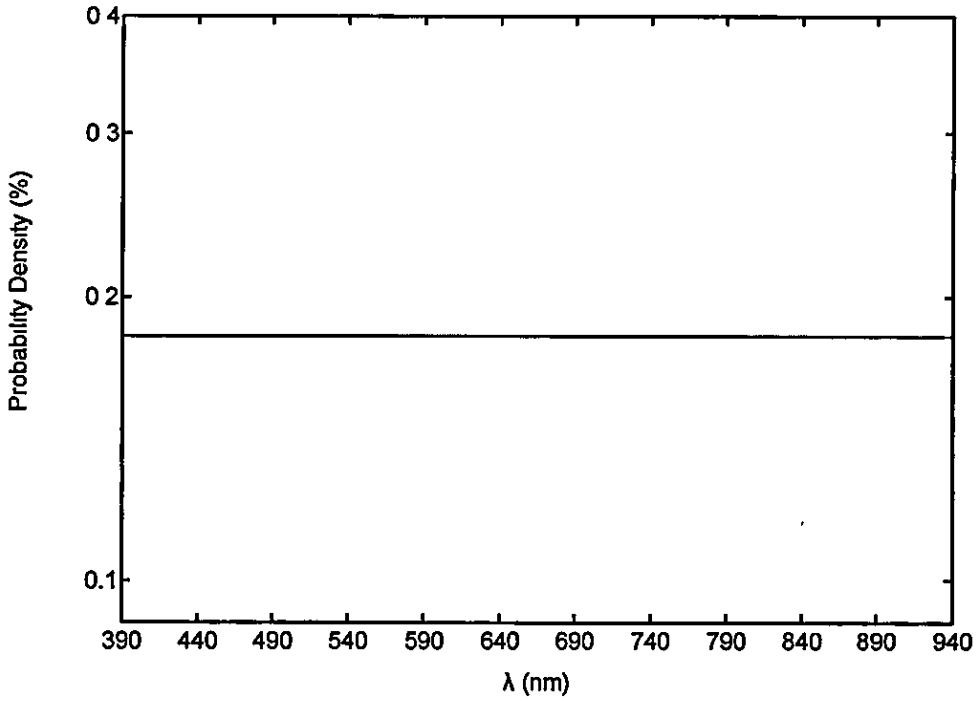


Figure 4-3 – Spectral sensitivity function of the nephelometric detectors

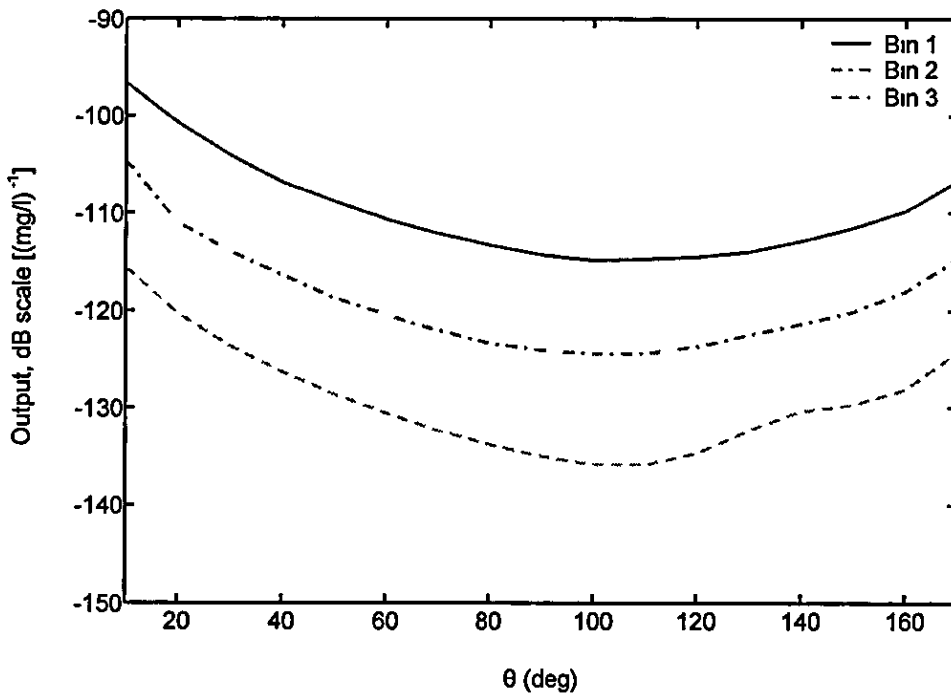


Figure 4-4 – Normalised scattering profiles of $AF_{1,2,3}$ per $AF_{1,2,3}$ mass concentration (LED A)

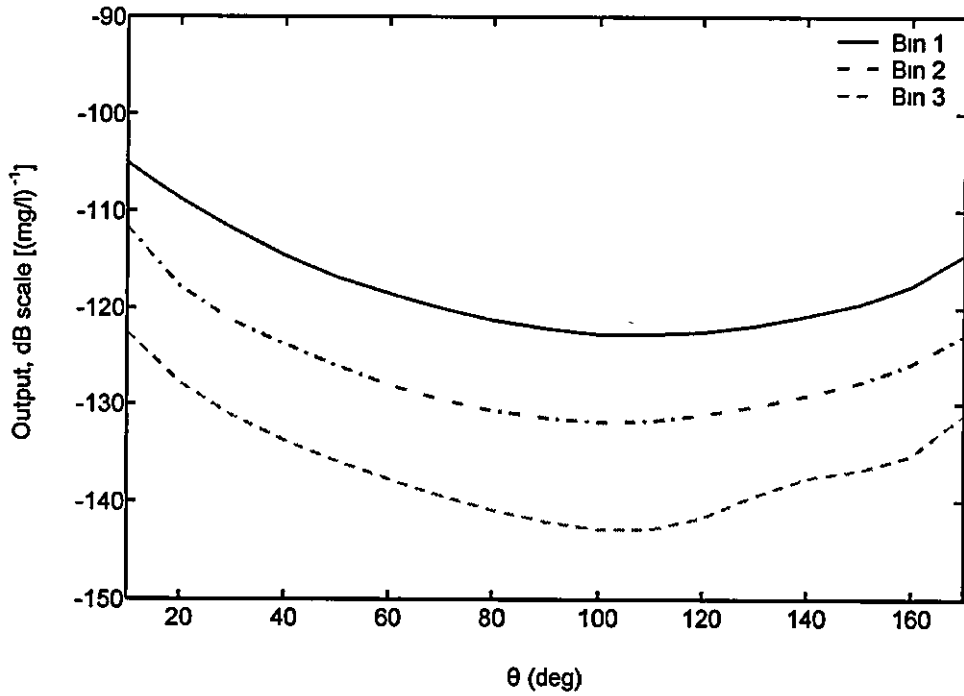


Figure 4-5 – Normalised scattering profiles of $AF_{1,2,3}$ per $AF_{1,2,3}$ mass concentration (LED B)

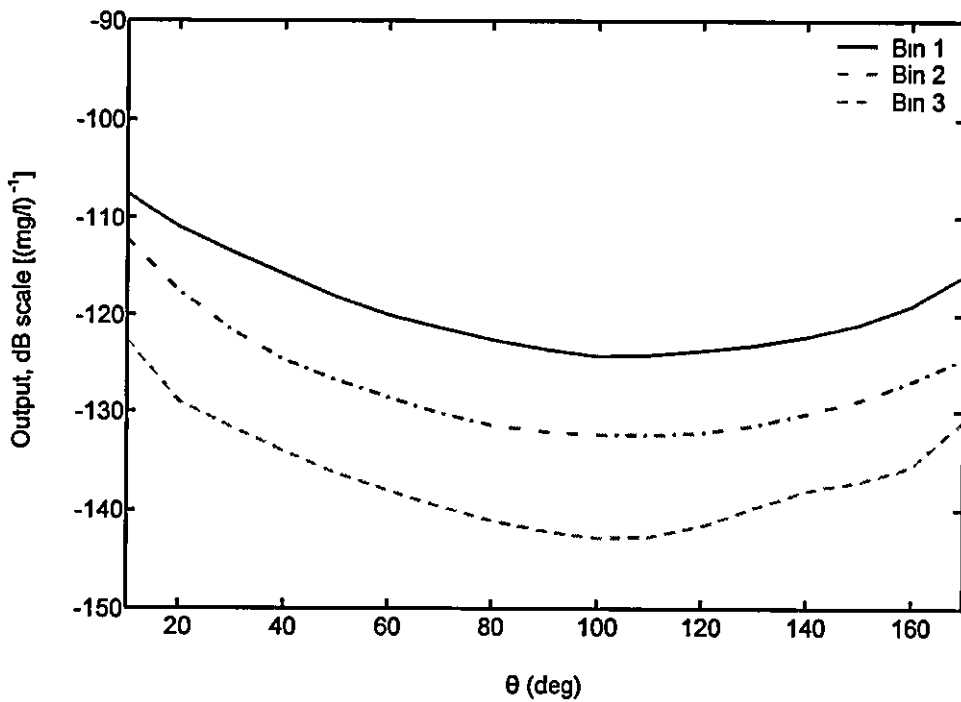


Figure 4-6 – Normalised scattering profiles of $AF_{1,2,3}$ per $AF_{1,2,3}$ mass concentration (LED C)

The numerical calculation of $I_{sj}(\theta)$ is greatly simplified by the introduction of the *normalised particle scattering per unit mass concentration* $\bar{S}_{ij}(\theta)$ defined by:

$$\bar{S}_{ij}(\theta) \equiv \frac{I_{sj}(\theta)}{C_m I_{0i}} \Big|_{\sum_{j=1}^3} \quad (4-10)$$

and expressed in inverse mass concentration units (l/mg). According to equation (4-8), the scattered irradiance function is linearly proportional to the total mass concentration of suspended AF sand and the total irradiance emitted by the nephelometric source. Therefore the tiny set of functions $\bar{S}_{ij}(\theta)$ calculated by (4-10) and (4-8) generates the full set of 3 (see §4.2.2) x 231 (see §4.2.1) scattering profiles by the linear expression:

$$I_{sj}(\theta) = C_m I_{0i} \sum_{j=1}^3 v f_{ji} \bar{S}_{ij}(\theta) \quad (4-11)$$

Figure 4-4 to figure 4-6 demonstrate the normalised particle scattering per unit mass concentration functions for the three nephelometric sources. According to these figures, $\bar{S}_{ij}(\theta)$ satisfies the following inequalities:

$$\bar{S}_{i_1}(\theta) > \bar{S}_{i_2}(\theta) \quad i_1 < i_2 \quad (4-12)$$

$$\bar{S}_{i_1}(\theta) > \bar{S}_{i_2}(\theta) \quad l_1 < l_2 \quad (4-13)$$

The scattering function $S_{11}(\lambda, \theta', x)$ is proportional to the size parameter χ defined by (3-146). However, χ is inversely proportional to wavelength λ . These remarks and the domains of spectral functions $\text{src}_i(\cdot)$ justify (4-12). Also, inequality (4-13) is satisfied because binned AF sand types of smaller particle sizes contain more scattering particles per unit volume and hence factor $1/x^3$ in (4-8) is increased accordingly. Finally, the shape of $\bar{S}_{ij}(\theta)$ becomes more irregular as the ordinal number of binned AF sand type increases. This is due to the additional terms required for the calculation of truncated series (3-156) and (3-157) according to equation (3-148).

4.2.4 Volume fraction estimation model

Two data sets, one of input and another of target vectors, are required to train, test and optimise an ANN model of particle classification problem considered. The set of 51x1 raw input vectors $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q, \dots, \mathbf{p}_{231}\}$ is constructed by sampling the scattered irradiance functions $I_{sj}(\theta)$ at all angular positions θ_n at which nephelometric

detectors can be placed according to predetermined geometrical and physical constraints. If it is assumed that θ_n is given by:

$$\theta_n = n \cdot 10^\circ \quad 1 \leq n \leq 17 \quad (4-14)$$

it emerges that three 17x1 vectors \mathbf{p}_{1q} defined for every nephelometric source by:

$$\mathbf{p}_{1q} \equiv [I_{slq}(\theta_1) \quad I_{slq}(\theta_2) \quad \dots \quad I_{slq}(\theta_n) \quad \dots \quad I_{slq}(\theta_{17})]^T \quad (4-15)$$

assemble the 51x1 raw input vector \mathbf{p}_q by augmentation:

$$\mathbf{p}_q = [\mathbf{p}_{1q}^T \mid \mathbf{p}_{2q}^T \mid \mathbf{p}_{3q}^T]^T \quad (4-16)$$

The set of 3x1 raw target vectors $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_q, \dots, \mathbf{t}_{231}\}$ is formed by grouping the mass concentrations of the three binned AF sand types dispersed in every individual sample suspension. Every target vector is linearly proportional to the vector of volume fractions that refers to the same suspension and suspended matter. In mathematical terms:

$$\mathbf{t}_q = C_m \cdot [vf_{q1} \quad vf_{q2} \quad vf_{q3}]^T \quad (4-17)$$

Once \mathbf{t}_q is estimated, an approximation of the corresponding volume fraction vector can be obtained by (4-17) and the value of C_m stated in §4.2.3 ($C_m = 20 \text{ mg/l}$).

The raw input and target vector sets are uniformly normalised to the interval [-0.5, 0.5] by applying the appropriate pre-processing algorithm of §2.8. The 462 normalised vectors $\tilde{\mathbf{p}}_q, \tilde{\mathbf{t}}_q$ are grouped in 231 pairs of the form $\{\tilde{\mathbf{p}}_q, \tilde{\mathbf{t}}_q\}$. Eleven vector pairs are randomly selected for network training use (*training pattern set*) and the remaining 220 pairs are destined to measure the degree of generalisation exhibited by the trained network at the end of every ANN optimisation step (*testing pattern set*).

An initial 51-2-3 MFANN is constructed to estimate the mass concentrations of the three binned AF sand types. The hidden and output nodes of that network are assigned log-sigmoid and linear transfer functions respectively. The network training task is performed via the LMBP algorithm, and the network optimisation goal is achieved via the modified SOSA and OBS algorithms. The performance index of the training algorithm is the MSE function defined by (4-10). The LMBP parameters are: $\text{MSE}_{\text{goal}} = 10^{-6}$, $E_{p_{\text{max}}} = 100$, $\nabla_{\text{min}} = 10^{-6}$, $\mu_0 = 10^{-3}$, $\mu_{\text{max}} = 10^{10}$, and $\theta = 10$. The optimisation parameters are: $c_1 = +\infty$, $c_2 = 2$, $c_3 = 3$, $\alpha = 10^{-6}$ and $N_{\text{max}}^+ = 0$. The training and optimisation of the initial network are performed together by applying the complete MFANN optimisation scheme described in §2.9.4. The outcome of the modelling procedure is a 2-2-3 MFANN having 9 weights and 3 biases.

Table 4-1 – MFANN training and optimisation results

Optimisation Step	Epochs	Removed Nodes	Optimisation Step	Epochs	Removed Nodes
1 (training)	21	—	12	1	90 ^A , 90 ^B , 150 ^A
2	1	20 ^B , 80 ^A , 100 ^A	13	2	20 ^A , 50 ^B , 90 ^C
3	1	30 ^B , 60 ^C , 170 ^C	14	2	40 ^B , 80 ^C , 160 ^A
4	1	40 ^C , 130 ^C , 140 ^C	15	2	130 ^A , 150 ^B , 160 ^B
5	1	50 ^A , 100 ^C , 170 ^A	16	2	60 ^B , 110 ^B , 150 ^C
6	1	50 ^C , 60 ^A , 100 ^B	17	50	10 ^A , 10 ^C , 120 ^A
7	1	20 ^C , 110 ^A , 110 ^C	18 (1 backtrack)	11	70 ^C
8	2	30 ^C , 80 ^B , 120 ^B	19 (training)	0	—
9	2	30 ^A , 120 ^C , 140 ^B	20 (OBS)	—	bias 2 ^h
10	2	40 ^A , 70 ^A , 160 ^C	21 (OBS)	—	bias 1 ^h
11	1	10 ^B , 130 ^B , 170 ^B	22 (OBS)	—	weight 2 ^h – 3 ^o

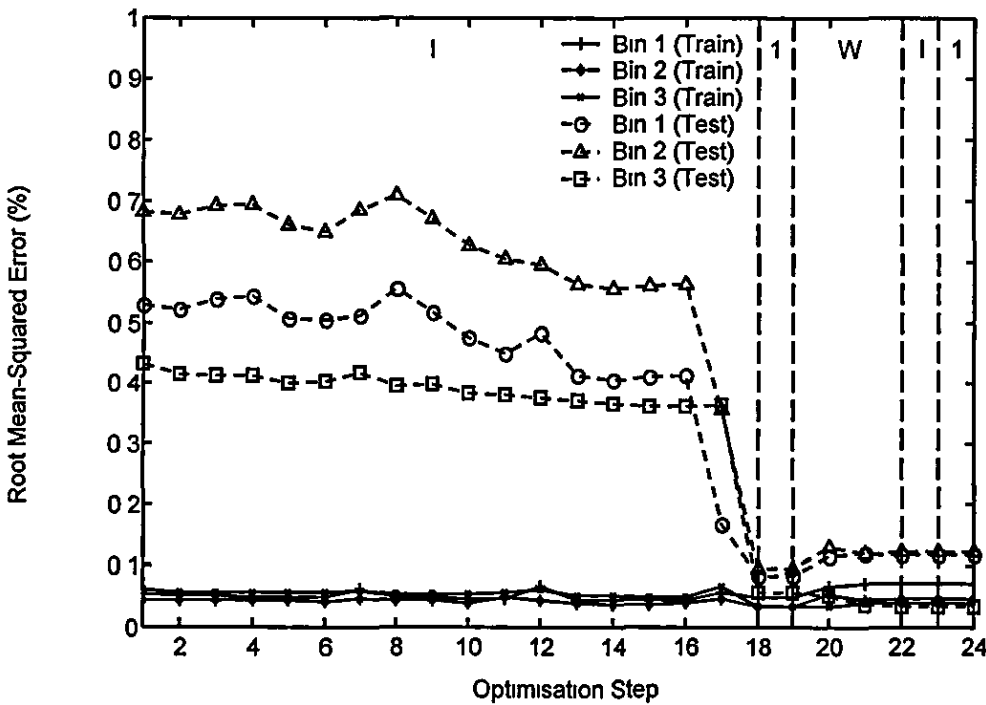


Figure 4-7 – RMS error change per MFANN optimisation step

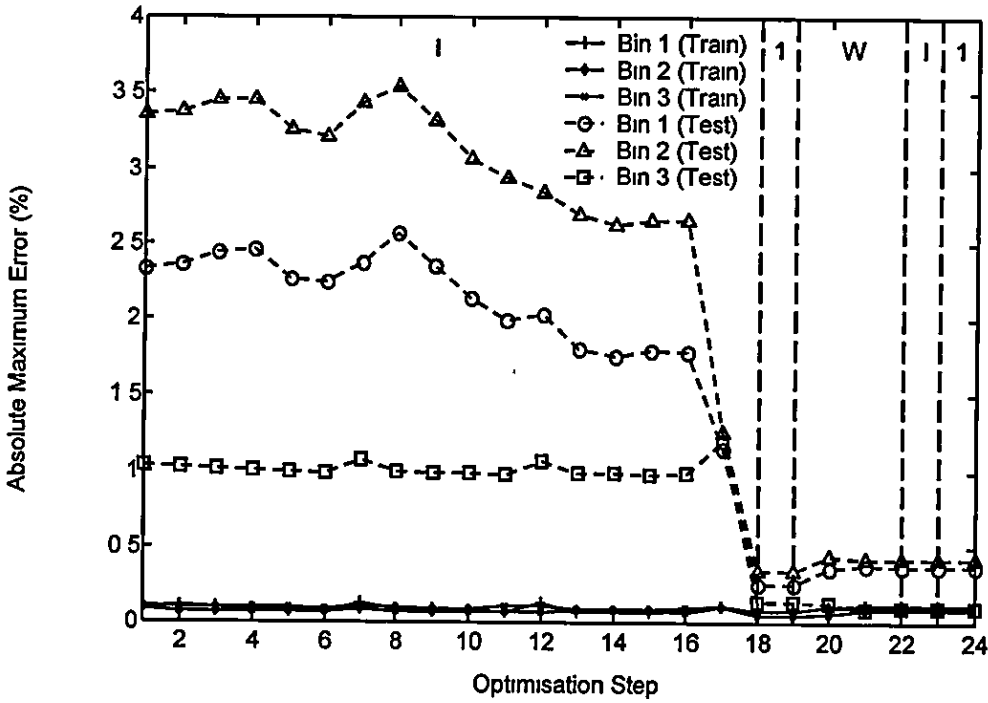


Figure 4-8 – Absolute maximum error change per MFANN optimisation step

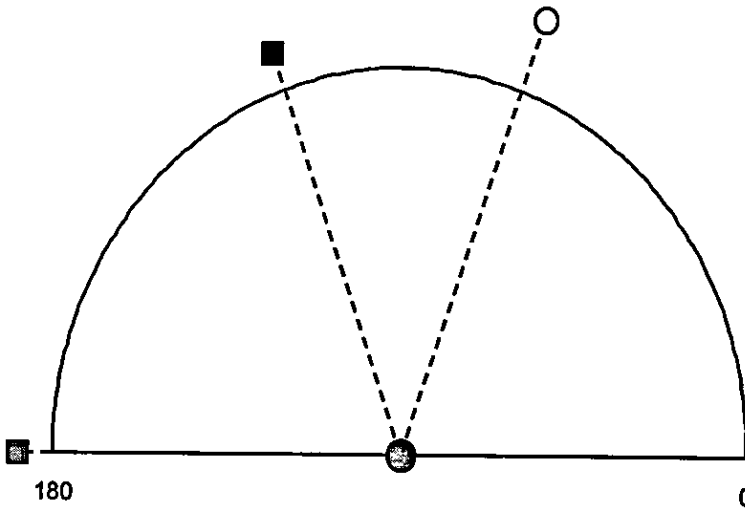


Figure 4-9 – Structurally optimal nephelometric sensor configuration

Table 4-1 lists the number of training epochs performed and the types and positions of ANN elements removed per optimisation step. Input nodes are denoted by the associated scattering angles θ_n and source indices i (A, B or C) in the form θ_n^i . Hidden and output nodes are represented by superscripts h or o placed next to their ordinal numbers (e.g. 1^h = first hidden node, 1^o = first output node). Most optimisation steps remove the maximum number of nodes (3) in fewer than 5 epochs.

Figure 4-7 and figure 4-8 demonstrate the *Root Mean Squared (RMS)* and *absolute maximum* errors of the mass concentration approximations obtained by the MFANN after each optimisation step. The generic error function:

$$F_r(\mathbf{e}_l) \equiv Q^{-\frac{1}{r}} \cdot \left\| \left[{}_l e_1 \quad {}_l e_2 \quad \dots \quad {}_l e_q \quad \dots \quad {}_l e_Q \right]^T \right\|_r \quad (4-18)$$

with Q standing for the number of network patterns and l denoting the output node index, defines the RMS error when $r = 2$ and the absolute maximum error when $r = +\infty$. Both errors drop quickly at the final input layer optimisation steps when the testing pattern set ($Q = 220$) is applied to the network, while the same errors do not change significantly between optimisation steps when they are calculated with respect to the training pattern set ($Q = 11$). The final RMS and absolute maximum errors associated with the testing pattern set are less than 0.2% and 0.5% respectively.

The angular positions of the two nephelometric sources and the single nephelometric detector displayed in figure 4-9 and supplying the final network model with scattering data are obtained from the (θ, i) associations of the final MFANN input nodes. LED A is placed at 110° , LED B at 180° and the detector at 70° with respect to the direction of propagation of light emitted by LED B. When a sample suspension is placed in the nephelometer, the two light sources start flashing alternatively and the signal returned by the detector is time-division demultiplexed. The derived signals are rectified, sampled and uniformly normalised before they are applied to the network model. Finally, the MFANN returns uniformly normalised approximations of the mass concentrations of the AF sand dispersed in the nephelometric sample.

4.3 Filtered AF sand type detection problem

This experimental particle characterisation problem considers two sets of suspensions, one of small ($<10 \mu\text{m}$) and another of small and bigger ($<35 \mu\text{m}$) AF particles dispersed in water. A polar nephelometric system coupled with an instance of a

predefined set of light sources is used to acquire the scattering profile of every sample under consideration. An analytical procedure is applied to the acquired profiles so that all contributions not associated to light scattering by the AF particles are removed. The processed experimental profiles are compared with the theoretical counterparts returned by the polar nephelometric model of §3.5 and the differences between the two are explained and proved analytically. Finally, the experimental scattered irradiance measurements are utilised to train, optimise and test a MFANN model that classifies the trial samples according to suspended AF sand type.

4.3.1 Experimental sample preparation

Two filtered AF sand types were used to prepare the suspensions for the scattering experiments. These types are commercially available by Particle Technology Ltd and have particle volume probability distribution functions $v_p P_F(\cdot)$ and $v_p P_C(\cdot)$ as shown in figure 4-10 [183]. The first (*fine*) type has particles in the 250 nm - 10 μm size range and will be denoted by AF_F, while the second (*coarse*) type has a wider particle size range (60 nm - 35 μm) and will be denoted by AF_C.

The particle volume probability distribution function is of little importance for the act of modelling the scattering experiment. What matters more is the particle size probability density function (pdf) or the equivalent particle volume pdf of the two filtered AF sand types. The requested pdfs can be derived from the distribution functions $v_p P_F(\cdot)$ and $v_p P_C(\cdot)$ of figure 4-10 in four steps. Firstly, the inverse trapezoidal integration method differentiates the two distributions and gives the *binned particle volume probability density functions* $v_p \hat{P}_F$ and $v_p \hat{P}_C$ of figure 4-11. Secondly, the binned particle volume pdfs are transformed to *binned particle size probability density functions* $d_p \hat{P}_F$ and $d_p \hat{P}_C$ of figure 4-12 via the conversion algorithm described in §3.6 and applied in a similar case (no dithered pdf versions are assumed this time). Thirdly, the binned particle size pdfs are linearly interpolated at the centres of the particle size bins and give the smooth *particle size probability density functions* $d_p p_F(\cdot)$ and $d_p p_C(\cdot)$ of figure 4-13. Finally, the pdf conversion algorithm of §3.6 is applied once again to the particle size pdfs and produces the associated *particle volume probability density functions* $v_p p_F(\cdot)$ and $v_p p_C(\cdot)$ of figure 4-14. It is implied that all

probability density (distribution) functions mentioned above are normalised to unity with respect to integral (sum) calculated over their domains.

Small quantities of the AF_F and AF_C sand were dispersed in distilled water and formed three sets of tenuous suspensions for experimental use. Every set was prepared to take part in scattering experiments involving the same light source. The mass distribution of the dispersed substance in a set of suspensions was planned to satisfy the following conditions:

- a) *Linearity.* The linear relationship between the dispersed mass concentration and the scattered irradiance measured at a fixed angle is better demonstrated if the former is distributed in linear progression (equal increments). The selection of the progression step and initial value should guarantee the clear distinction between the scattering profiles acquired by the suspension set under consideration. Mass concentrations smaller than 10 mg/l do not satisfy the profile distinction requirement.
- b) *Small-value domain.* The single and independent scattering conditions stated in §3.5 are met if the suspensions to be used in the experiment are tenuous, i.e. the dispersed mass concentration is smaller than 60 mg/l.

The procedure followed in the preparation of a set of suspensions required a millilitre-graded water tank, a milligram-graded scale, a bag of AF_F sand and a bag of AF_C sand. Firstly, the scattering profile of V_W litres of distilled water poured into the tank and the sample was obtained experimentally. Secondly, the AF_F mass of $10 V_W$ mg was dispersed in the tank water and the scattering profile of the suspension was acquired. The second step was repeated until the mass concentration of AF_F in the suspension exceeded 60 mg/l. After that, the tank was emptied and washed. The aforementioned procedure was followed once more with AF_F sand replaced by AF_C .

Table 4-2 lists the mass concentrations of filtered AF sand in the suspensions that participated in the scattering experiments. The concentration values were measured by multiplying the change in suspension volume due to filtered AF dispersion by the bulk mass density of the substance ($d_m = 2400 \text{ kg/m}^3$ for AF_F and AF_C). The deviations of mass concentration values from multiples of 10 mg/l should be mainly attributed to inaccuracies in the measurement of substance mass prior to dispersion. The LED types mentioned in table 4-2 are the light sources used in the scattering experiments (see §4.3.2 for details).

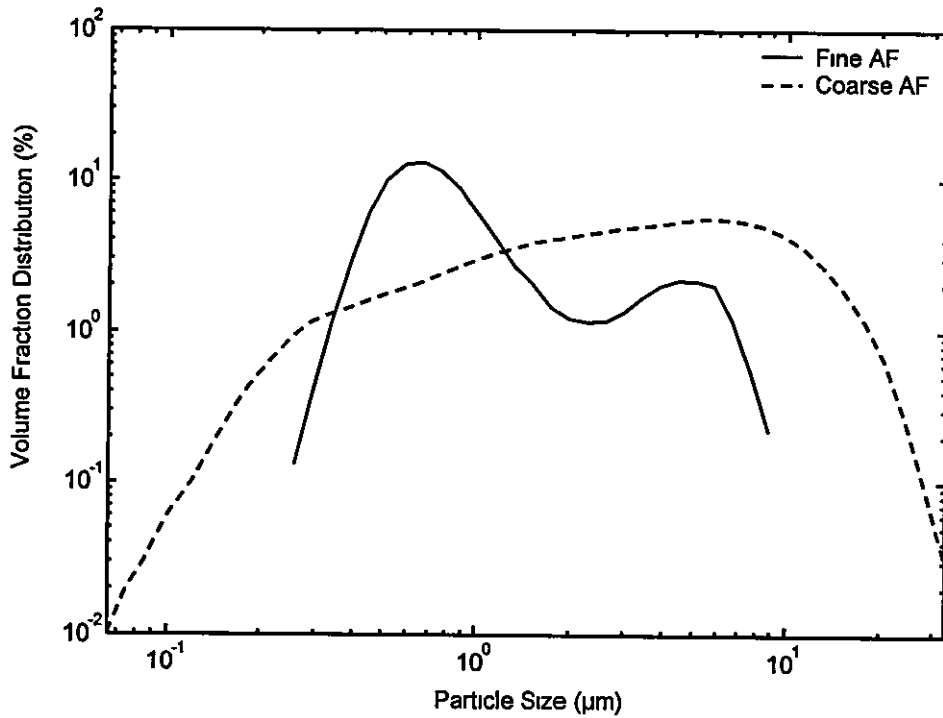


Figure 4-10 – Particle volume probability distribution functions of $AF_{F,C}$

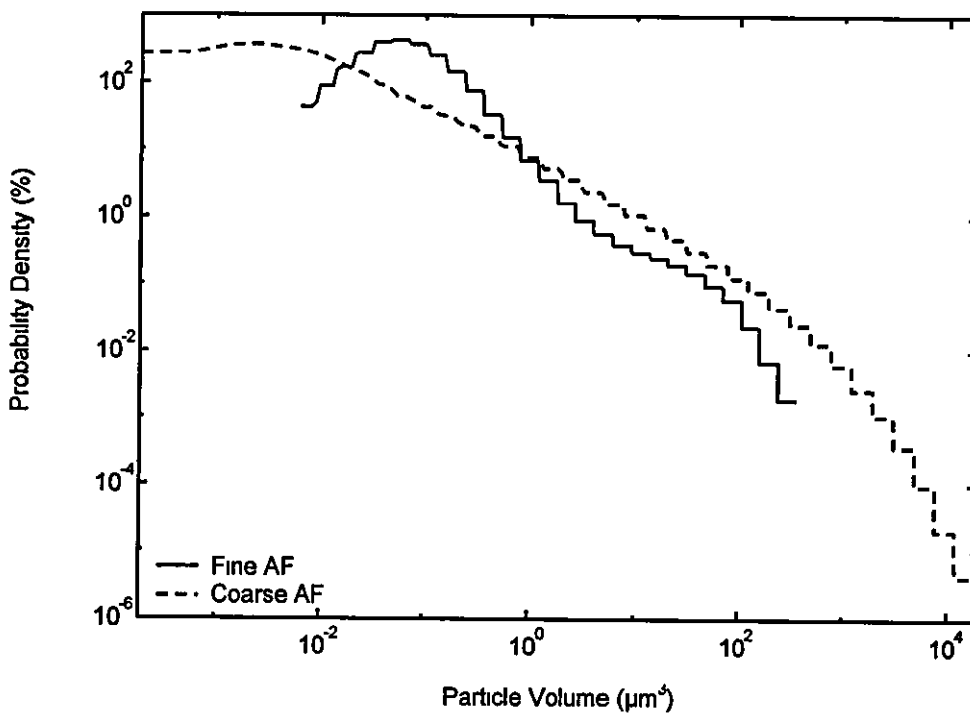


Figure 4-11 – Binned particle volume probability density functions of $AF_{F,C}$

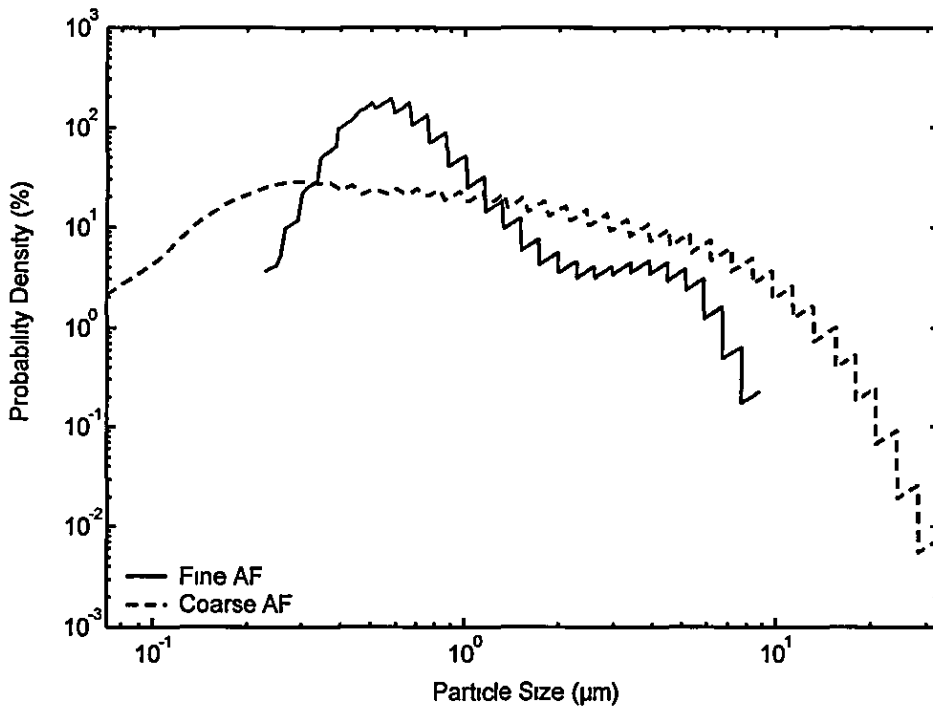


Figure 4-12 -- Binned particle size probability density functions of $AF_{F,C}$

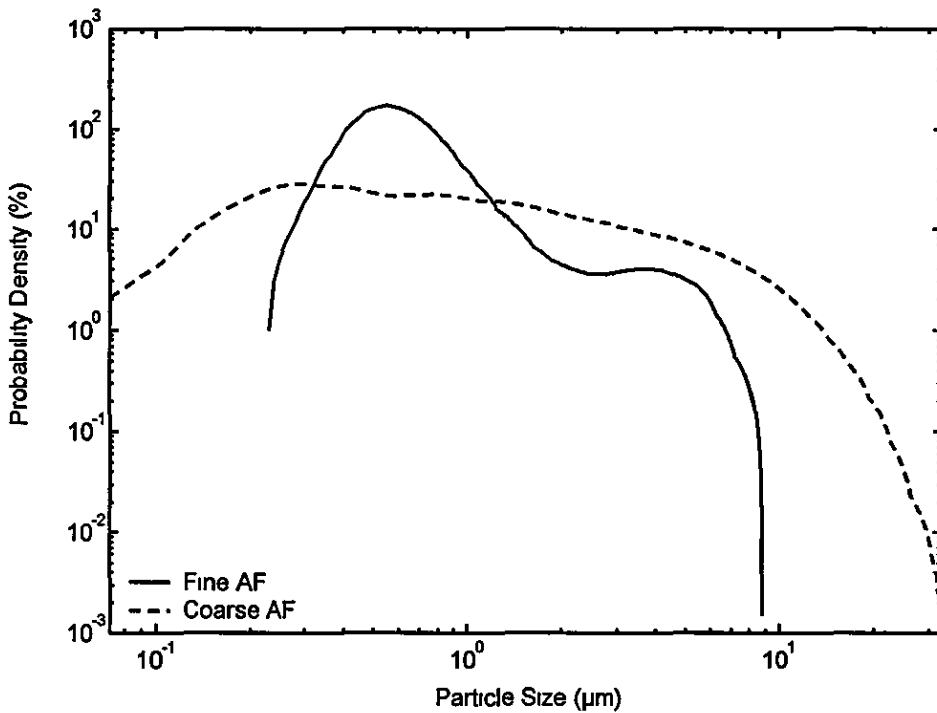


Figure 4-13 – Particle size probability density functions of $AF_{F,C}$

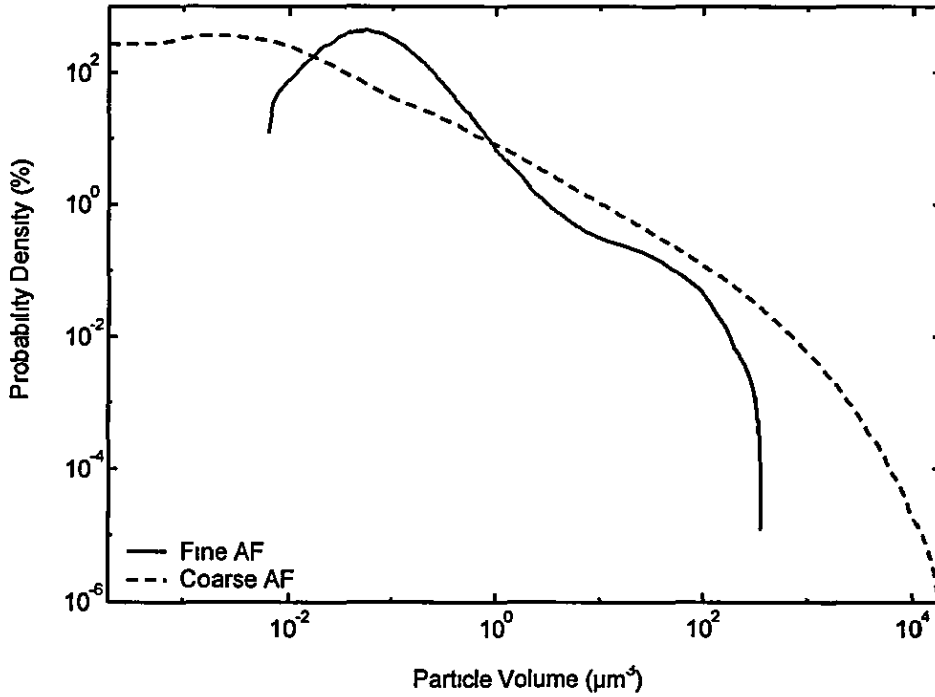


Figure 4-14 – Particle volume probability density functions of $AF_{F,C}$

Table 4-2 – Mass concentration of $AF_{F,C}$ in experimental suspensions

Filtered AF Sand Type		Fine	Coarse
LED Type	Green (RS 2281879)	10	14
		20	24
		36	34
		48	45
		58	56
	Red (RS 826701)	21	15
		34	27
		44	38
		57	55
	Infrared (RS 2678380)	12	10
		24	21
		37	32
		49	42
		59	55

4.3.2 Experimental setup

The basic apparatus used to obtain light scattering measurements from water suspensions of AF_F and AF_C was the fibre optic polar nephelometer sited at IROE headquarters in Florence, Italy [184]. The front and side views of the device are displayed in figure 4-15 and figure 4-16 respectively. The main parts of the nephelometer, most of which are distinguishable in figure 4-15, are the following:

- a) *Light illuminator.* It is composed of a piece of optical fibre FT-200-EMT [185] terminating in an SLW-1.8-0.25-0.63 GRIN lens [186]. The fibre has been manufactured by 3M and its core diameter is 200 μm . The lens has been made by Nippon Sheet Glass Co. Ltd, and is connected to the fibre via an FCM-00F-200-0.63 fibre-to-GRIN assembly also made by NSG. The lens-terminated end of the fibre is housed in a silicon-sealed receptacle of the illumination arm of the nephelometer to avoid any accidental contact between the optical circuit and the suspension sample during the data acquisition process. The front end of the receptacle is made of a glass window so that light emitted by the GRIN lens can reach the suspension. The other end of the fibre is coupled to an LED source via two components: an SMA-compatible LED receptacle and an LED-to-fibre spacer. The receptacle is commercially available by OFTI while the spacer is made by IROE [187].
- b) *Light detector.* It is a fibre-lens system identical to the light illuminator. The lens-terminated end of the fibre is placed in a silicon-sealed receptacle of the detection arm of the nephelometer for the same reason as the illuminator. The front end of the receptacle is made of a glass window so that light scattered by the suspension can be collected by the GRIN lens. The other end of the fibre is coupled to a solid-state detector (a photodiode followed by a trans-impedance amplifier of variable gain) that converts light energy to weak electrical current.
- c) *Rotating stage.* It turns the detection arm of the nephelometer around the rigid illumination arm in a circular trajectory. Both arms remain in the same plane in space at all times. The detection arm is aligned to the illumination arm at its original position with the transparent ends of the arms facing each other. This coaxial arrangement of the arms is shown in figure 4-15.
- d) *Glass container.* It is used to hold the particle suspension from which light scattering measurements are about to be obtained.

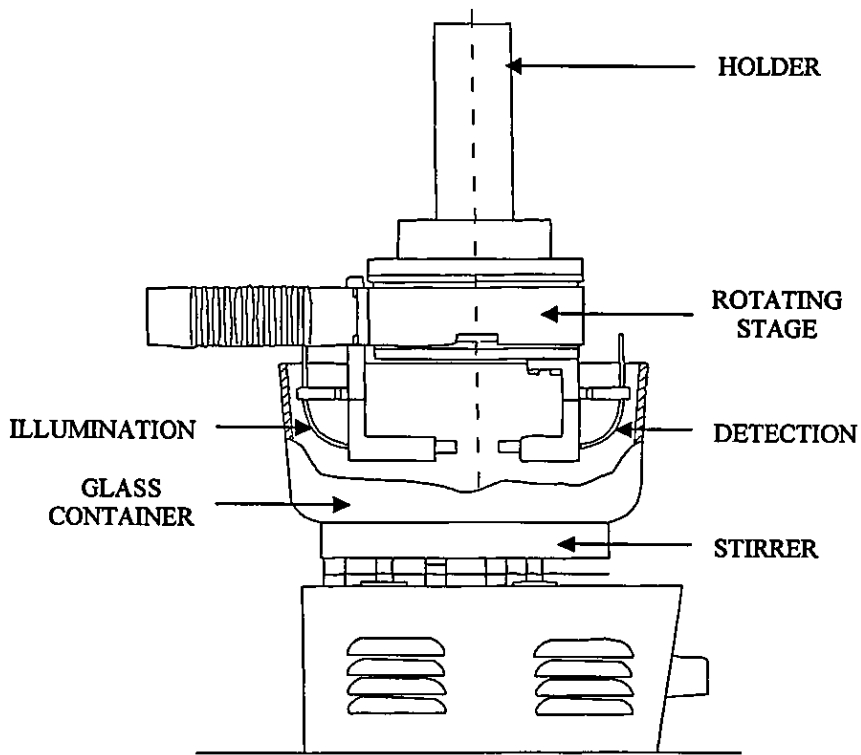


Figure 4-15 – Fibre optic polar nephelometer (front view)*

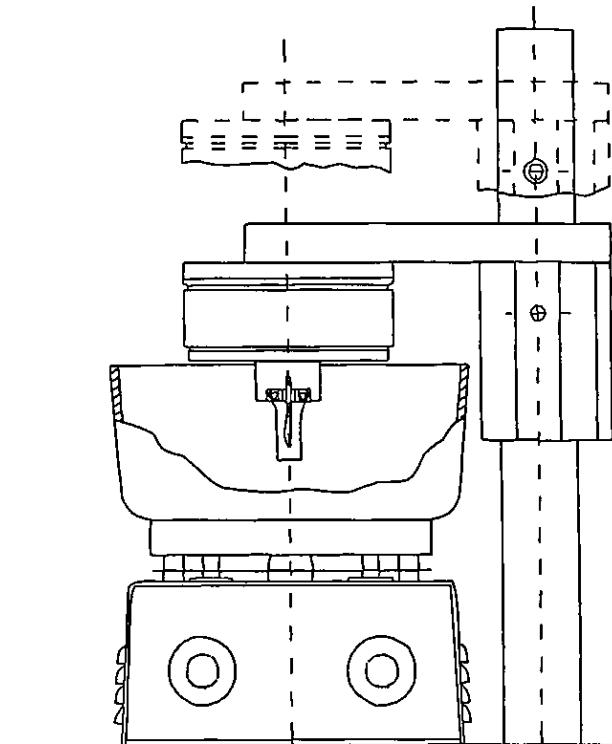


Figure 4-16 – Fibre optic polar nephelometer (side view)*

* From [184]; © 1998 by IROE. Reprinted with permission.

- e) *Stirrer*. It is the base of the glass container as shown in figure 4-15. The stirrer agitates the suspension during the scattering data acquisition process so that most dispersed particles remain in motion and hence they are kept in the suspension. The effect of particle settling in the bottom of the container due to gravity has to be avoided since it reduces the number of particles contributing to light scattering.
- f) *Holder*. It allows the movement of the rotating stage and the illumination and detection arms attached to it between two predetermined vertical positions. The stage has to move to "low" position and immerse its arms into the suspension held by the glass container before the scattering experiment can begin. The stage moves back to "high" position at the end of the experiment so that the container can be removed or replaced if necessary.

A set of auxiliary instruments had to be used with the IROE nephelometer in order to make that device work and log scattering data. These instruments are the following:

- a) *Micro Controle IP28 control unit*. It controls the stepper motor that drives the rotating stage and detection arm of the nephelometer (see [188] for details).
- b) *Micro Controle TL78 power unit*. It supplies the stepper motor that drives the rotating stage and detection arm of the nephelometer.
- c) *Princeton 5208 two-phase lock-in amplifier*. It amplifies the photocurrent returned by the light detector of the nephelometer and converts it to voltage in the mV range (see [189] for details).
- d) *Personal Computer (PC)*. It initialises and manages the control unit and the lock-in amplifier remotely via customised software written in Quick Basic 4.5. The computer is also responsible for the real-time processing and logging of the scattering data returned by the nephelometer. The communication between the PC and the controlled devices is compliant with the IEEE488-1978 GPIB protocol.

Three LED types were distinctively coupled with the light illuminator of the IROE nephelometer and formed the light sources of the scattering experiments. These LEDs are commercially available by RS with model names RS 2281879 [190], RS 826701 and RS 2678380 [191] and emit light of green, red and infrared (IR) colour respectively.

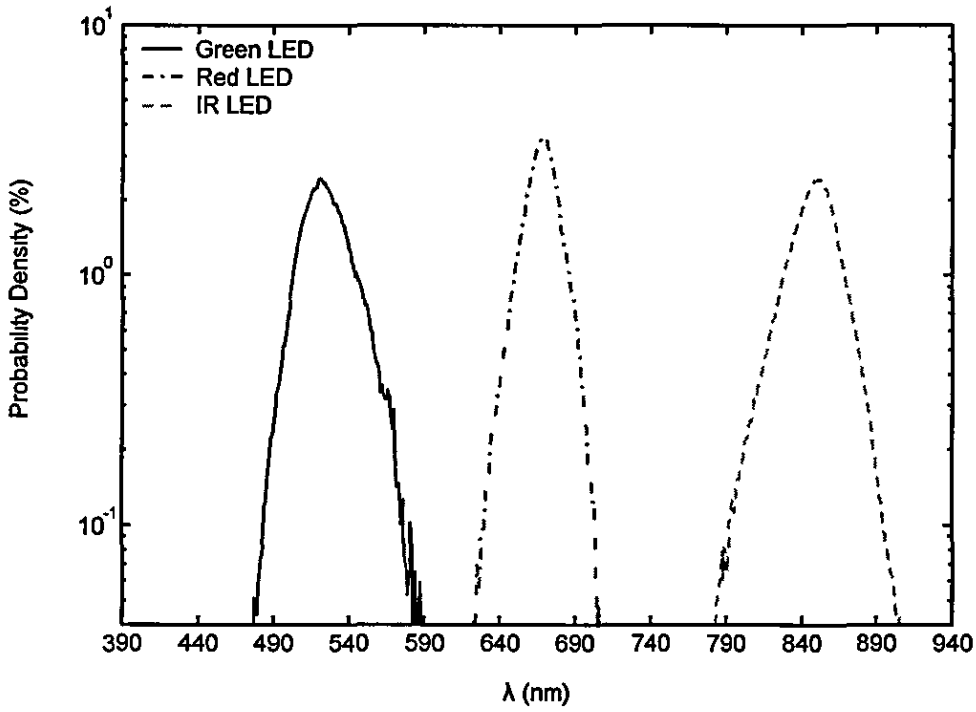


Figure 4-17 – Spectral intensities of the three nephelometric illumination systems

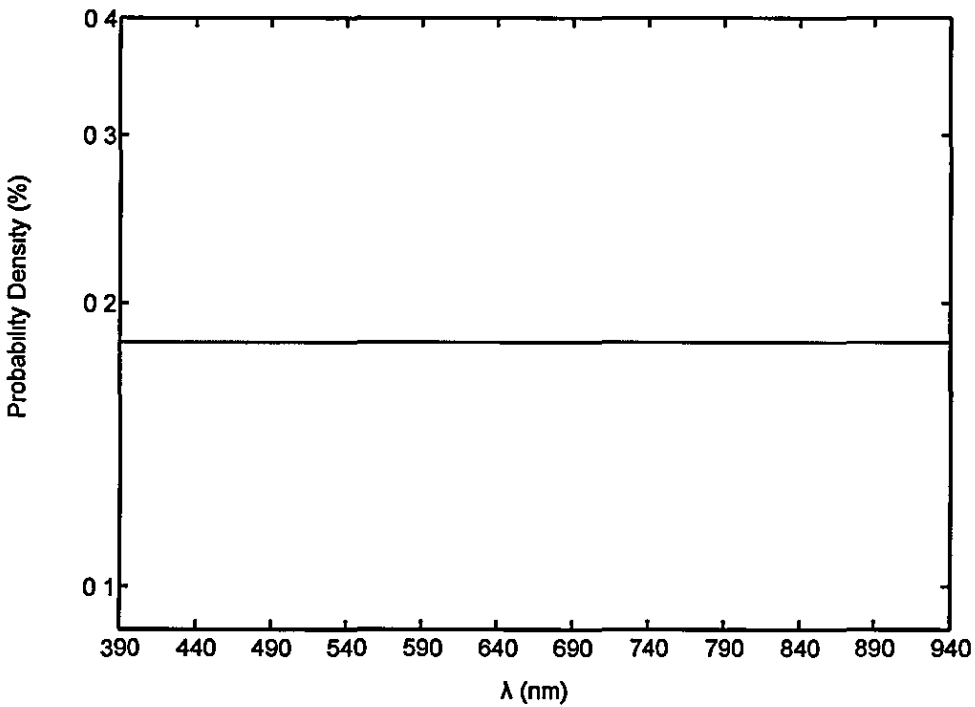


Figure 4-18 – Spectral sensitivity of the nephelometric detection system

Three different optical arrangements can be constructed by coupling the light illuminator of the IROE nephelometer with one of the LED sources selected for the experiment. Each of these arrangements is characterised by a spectral intensity function $src_i(\cdot)$ over the visible and near-infrared frequency domain ($i = G$ for green, R for red or IR for infrared light source). Figure 4-17 illustrates these functions in probability density function format (total emitted irradiance = 1 W/m^2). The functions were acquired by a photometer placed in front of the transparent side of the illumination arm while the LED sources were exposed to 20 mA [187].

The peaks of $src_i(\cdot)$ shown in figure 4-17 appear at $\lambda_{pG} = 521 \text{ nm}$, $\lambda_{pR} = 669 \text{ nm}$ and $\lambda_{pIR} = 851 \text{ nm}$. These peak values deviate only slightly from the theoretical peak values of 525 nm, 654 nm and 850 nm given by the LED source manufacturer. Therefore it is reasonable to assume that the spectral intensity functions of figure 4-17 have been measured accurately.

The light detector of the IROE nephelometer is also characterised by the spectral sensitivity function $rec(\cdot)$. The optical window placed in front of the detection arm attenuates light *uniformly* over the visible and near-infrared frequency domain. The power losses in the coupling points of lens to fibre and fibre to solid-state detector are also independent of frequency. Furthermore, the GRIN lens of the detector shows negligible preference on regions of the frequency domain under consideration and therefore can be regarded as insensitive to light frequency. The same conclusion can be drawn about the fibre optic element of the detector due to its small length (a few meters only compared to a few kilometres required for frequency-specific light attenuation to be detectable). Therefore the spectral sensitivity function of the detector can be fairly assumed to be uniform over the frequency domain of interest. Figure 4-18 illustrates $rec(\cdot)$ in probability density function format (the total irradiance on the detector is assumed to be equal to 1 W/m^2 for graphical purposes only).

4.3.3 Experimental procedure

Every scattering experiment was managed by the same software routine running on the PC described in §4.3.2. The managing routine is divided in six parts according to [184]:

- a) *Micro Controle IP28/TL78 initialisation.* Firstly, the control unit is set to transmit all future data and error messages to the PC. Secondly, the unit is in-

structed to control the stepper motor of the IROE nephelometer so that the detection arm rotates 0.001° per motor step (full step mode). Thirdly, the power unit is programmed to send 3000 pulses/sec to the stepper motor during the data acquisition phase (high speed mode). In other words, the rotation speed of the detection arm is adjusted to 3° per second. Finally, the control unit is instructed to control the spinning direction of stepper motor axis so that the detection arm rotates counter-clockwise during the acquisition phase.

- b) *Princeton 5208 initialisation.* The lock-in amplifier is set to work at remote mode, i.e. interact with the PC. That setting implies a number of internal parameter initialisations for the amplifier. Firstly, the time constant and roll-off rate of the low-pass filters built in the amplifier are adjusted to 10 ms (that makes the filter cut-off frequency equal to 16 Hz) and 6 dB per octave. Secondly, the amplifier is instructed to wait for synchronisation signals coming from the PC. Thirdly, the lock-in parameter of the device is set to polar mode. Finally, the amplifier is set to autorange mode so that the device automatically adjusts its input range to fit the scattering data.
- c) *Initialisation of software parameters.* The main variables of the managing routine are $tmin$ (first scattering angle in degrees), $tmax$ (last scattering angle in degrees), $nstep$ (number of scattering angles excluding the first angle), dth (number of motor steps required for the movement of the detection arm between two adjacent scattering angles), $nome\$$ (name of file used for averaged data storage), and $dat\$$ (data acquisition date). The first four variables are assigned the values 0, 100, 20 and 5000 respectively. The reason the maximum scattering angle is set to 100° instead of 180° is that previous experience has shown that scattering measurements obtained at angles greater than 100° are severely affected by noise and thus have little statistical significance.
- d) *IROE nephelometer initialisation.* The rotating stage is moved to holder's "low" position.
- e) *Scattering data acquisition.* The detection arm is moved to the first angular position and an integer $hall(i) \in [-2000, 2000]$ is acquired from the first channel of the lock-in amplifier. The integer is converted to voltage number by:

$$valf(i) = hall(i) \cdot \frac{fs}{2000} \quad (4-19)$$

where $\text{valf}(i)$ is the voltage in mV that corresponds to the scattered light irradiance acquired at angle $\text{th}(i)$:

$$\text{th}(i) = tmin + (i - 1) \cdot \frac{tmax - tmin}{nstep} \quad (4-20)$$

i is the scattering angle index that satisfies the inequality $1 \leq i \leq nstep + 1$ ($i=1$ for the first angle) and fs is the maximum allowable input voltage expressed in mV. The last variable is assigned a value that belongs to a predefined set of 21 numbers and comes from the lock-in amplifier (see explanation of amplifier's autorange mode given above). After the first scattering measurement the suspension is strongly stirred and the detection arm is moved to the next angular position. The integer value returned from the lock-in amplifier is converted to voltage and stored in computer memory as before. The last two steps are repeated until the detection arm takes all angular positions derived from expression (4-20). The whole scattering data acquisition process takes about 4 minutes to complete. The process is repeated another 4 times and the 5 measurement sets obtained are averaged and stored in vector **pow**.

- f) *Data storage*. A text file with name specified by variable `nome$` is created on a PC storage device, e.g. hard disk drive. The text file is then opened for writing. The number of angles ($nstep + 1$) and the date stamp (`dat$`) of the scattering experiment are written on the first line. The following lines are filled with the pairs $(\text{th}(i), \text{pow}(i))$ for $1 \leq i \leq nstep + 1$. Finally, the text file is closed and its contents are permanently saved.
- g) *Return to idle state*. The detection arm is driven back to the default position, i.e. opposite to the illumination arm. The rotating stage is moved to holder's "high" position.

4.3.4 Experimental results

The experimental scattering data recorded in the PC manager as mV values are illustrated in figure 4-19, figure 4-20 and figure 4-21 for the three experimental light sources. The following remarks can be made about these results:

- a) The measurements obtained at the angular range $0^\circ - 10^\circ$ are severely affected by light coming directly from the transparent side of the nephelometric illumination arm. These measurements do not add much information about the scat-

tering effect caused by the suspended AF particles and hence they will be disregarded in the following discussion.

- b) The measurements corresponding to angles greater than 70° (55° in the case of green LED) are undermined by noise to such a degree that the single scattering law “more scattering particles = more scattering irradiance” is no longer valid. Therefore these measurements will also be omitted in the following analysis.

The recorded experimental data are not irradiances of scattered light; they are rather voltages returned by the lock-in amplifier. However, a linear relationship between the two quantities exists and is mathematically described by the formula:

$$V_{ij}(\theta, C_{mj}) = G_{ij} I_{0i} B_i(\theta) \exp(-2R\sigma_{ij} C_{mj}) + G_{ij} I_{0i} \hat{S}_{ij}(\theta, C_{mj}) + G_{ij} K \quad (4-21)$$

(see [183]), where:

- i is the light source index ($i = G$ for green, R for red or IR for infrared).
- j is the filtered AF sand type index ($j = F$ for fine, C for coarse).
- θ is the scattering angle returned by expression (4-20) ($0^\circ \leq \theta \leq 100^\circ$).
- C_{mj} is the mass concentration of filtered AF sand type j in the suspension. The values this variable assumes are listed in table 4-3 in mg/l units.
- R is the radius of the nephelometric chamber, i.e. half distance between the GRIN lenses of the aligned nephelometric illuminator and detector. The face-to-face distance was set to 17 mm before the scattering experiment and was not changed during the experimental process.
- σ_{ij} is the *average scattering cross-section* of the j -type AF particles dispersed in the experimental solution. The term is also dependent on the spectral characteristic of the nephelometric illuminating system including light source i . The value of σ_{ij} can be analytically determined by Mie scattering theory and expressed in m^2/kg units. However, knowledge of this term is not necessary thanks to the following analysis.
- I_{0i} is the total irradiance of light emitted by the GRIN lens of the nephelometric illuminator in combination with light source i . The term can be modified to include any power losses in the elements and coupling points of the illuminating system. I_{0i} can be accurately determined and expressed in W/m^2 units. Nonetheless, the following analysis clearly shows that this task is not required.

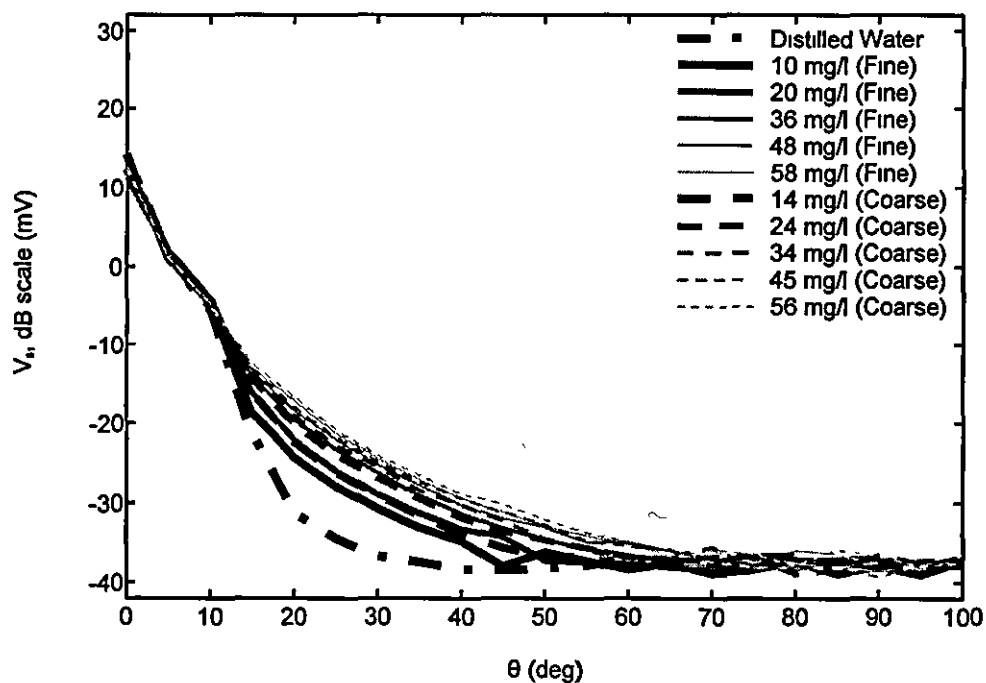


Figure 4-19 – Raw experimental scattering profiles of AF_{F,C} (green LED)

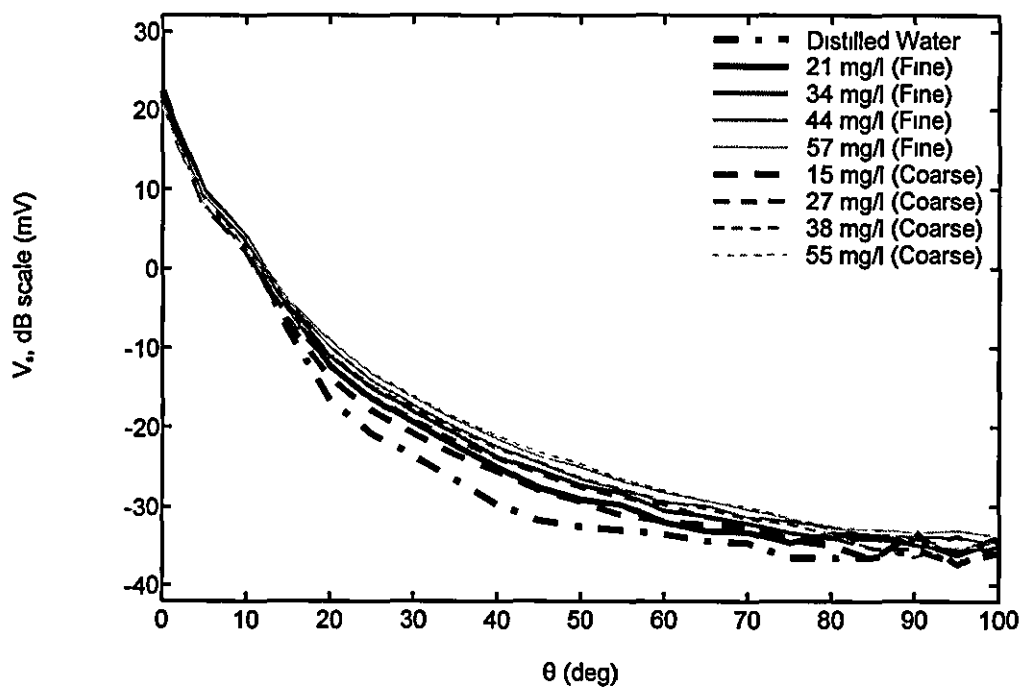


Figure 4-20 – Raw experimental scattering profiles of AF_{F,C} (red LED)

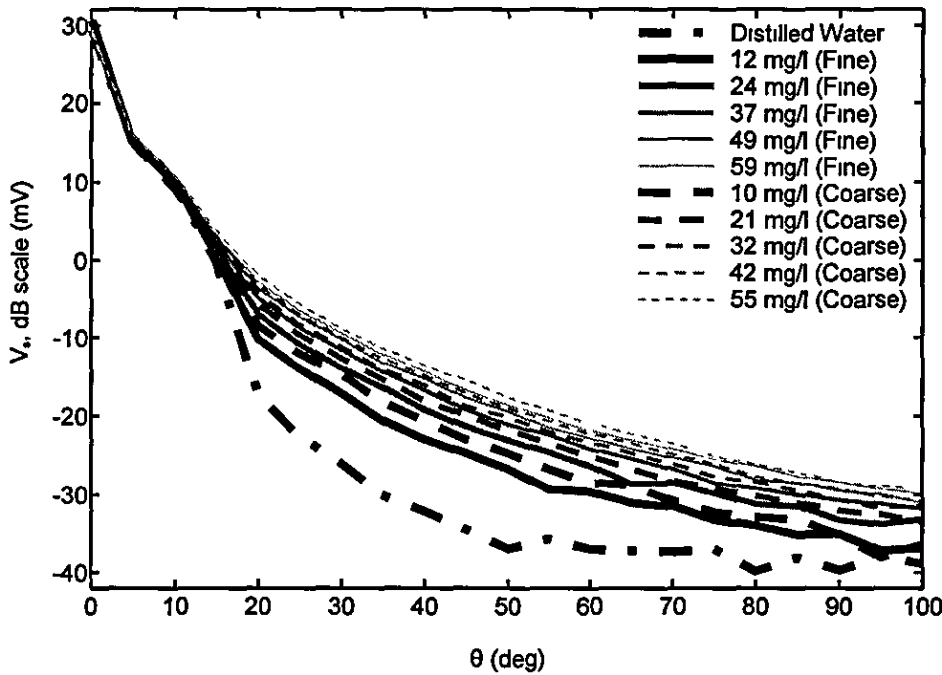


Figure 4-21 – Raw experimental scattering profiles of $AF_{F,C}$ (IR LED)

- K is the *bias* term arisen from the dark currents of the lock-in amplifier and the photodiode element of the solid-state detector. K is a random variable of W/m^2 units whose properties are not known.
- G_j is the *variable gain* of the trans-impedance amplifier multiplied by the gain of the lock-in amplifier when light source i is used to illuminate the suspension of j -type AF particles. The term can be modified to include any power losses in the windows of the nephelometric arms as well as the elements and coupling points of the nephelometric detector (adjustment of the value of bias term K is also required). G_j can be measured and expressed in m^2/A units. Nonetheless, this task is unnecessary (see analysis below).
- $B_i(\theta)$ is the scattered irradiance of distilled water at angular position θ per total emitted irradiance I_{0i} . This term represents the inherent brightening effect explained in §1.3.1.2 and is affected by the nephelometric geometry and the spectral characteristics of the light source, the nephelometric illuminator and the associated detector. $B_i(\theta)$ is a dimensionless stochastic function whose properties are unknown.

- $V_y(\theta, C_{mj})$ is the average voltage returned by the lock-in amplifier when light source i is used for the illumination of the suspension containing j -type AF particles at mass concentration C_{mj} and the nephelometric detector is at angular position θ . In other words, this function represents the recorded experimental data expressed in mV units.
- $\hat{S}_y(\theta, C_{mj})$ is the dimensionless *particle scattering* associated with the suspension containing j -type AF particles at mass concentration C_{mj} and denotes the scattered irradiance at angular position θ per total emitted irradiance I_{0i} . This term represents the scattering caused by AF particles alone and its value can be estimated by the polar nephelometric model of §3.5.

Unfortunately, $\hat{S}_y(\theta, C_{mj})$ cannot be calculated from $V_y(\theta, C_{mj})$ via (4-21) directly due to lack of reliable estimates for the stochastic variables $B_i(\theta)$ and K . However, a ratio involving $\hat{S}_y(\theta, C_{mj})$ is computable under fair assumptions for the magnitudes of the unknown parameters. Firstly, the constant term in the right-hand side of (4-21) is close to the minimum voltage recorded in distilled water (no AF particles):

$$G_{ij}K \cong \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\} \quad (4-22)$$

Secondly, the exponential factor of (4-21) may be approximated by the ratio of unbiased voltages measured at $\theta=0^\circ$ (LOS) when AF particles are present or not:

$$\frac{V_y(0, C_{mj}) - G_{ij}K}{V_y(0, 0) - G_{ij}K} = \frac{B_i(0) \exp(-2R\sigma_{ij}C_{mj}) + \hat{S}_y(0, C_{mj})}{B_i(0) + \hat{S}_y(0, 0)} \cong \exp(-2R\sigma_{ij}C_{mj}) \quad (4-23)$$

because the inherent brightening effect is stronger than particle scattering at LOS:

$$B_i(0) \gg \hat{S}_y(0, C_{mj}) \quad (4-24)$$

and by definition:

$$\hat{S}_y(\theta, 0) \equiv 0 \quad \forall \theta \in \mathbb{R} \quad (4-25)$$

Thirdly, the following relations can be derived from equations (4-21) and (4-25):

$$\frac{V_y(\theta, C_{mj}) - G_{ij}K}{V_y(\theta_{ref}, 0) - G_{ij}K} = \frac{\hat{S}_y(\theta, C_{mj})}{B_i(\theta_{ref})} + \frac{B_i(\theta)}{B_i(\theta_{ref})} \exp(-2R\sigma_{ij}C_{mj}) \quad (4-26)$$

$$\frac{V_y(\theta, 0) - G_{ij}K}{V_y(\theta_{ref}, 0) - G_{ij}K} = \frac{B_i(\theta)}{B_i(\theta_{ref})} \quad (4-27)$$

where θ_{ref} is the *reference angle* that is equal to one of the scattering angles. If the dimensionless *normalised particle scattering* $\check{S}_y(\theta, C_{mj})$ is defined by:

$$\check{S}_y(\theta, C_{mj}) \equiv \frac{\hat{S}_y(\theta, C_{mj})}{B_i(\theta_{\text{ref}})} \quad (4-28)$$

then equations (4-22) to (4-23) and (4-26) to (4-28) lead to the final result:

$$\check{S}_y(\theta, C_{mj}) \equiv \frac{V_y(\theta, C_{mj}) - \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\}}{V_y(\theta_{\text{ref}}, 0) - \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\}} \cdot \frac{V_y(0, C_{mj}) - \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\}}{V_y(0, 0) - \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\}} \cdot \frac{V_y(\theta, 0) - \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\}}{V_y(\theta_{\text{ref}}, 0) - \min_j \left\{ \min_{\theta} [V_y(\theta, 0)] \right\}} \quad (4-29)$$

Figure 4-22 to figure 4-24 display $\check{S}_y(\theta, C_{mj})$ as a function of scattering angle θ for $\theta_{\text{ref}} = 15^\circ$. The angular domain in these plots corresponds to all statistically significant measurements obtained but excludes the reference angle ($20^\circ \leq \theta \leq 55^\circ$ for the green light source and $20^\circ \leq \theta \leq 70^\circ$ for the other two sources). The difference between the normalised scattering profiles of AF_F and AF_C suspensions of similar mass concentration is small but visible. Specifically, an amount of AF_C sand dispersed in water scatters more light than a similar amount of AF_F sand under the same conditions. This observation is consistent with the corollary of Mie scattering theory that bigger particles scatter light more effectively than smaller particles of identical optical properties.

$\check{S}_y(\theta, C_{mj})$ is plotted as a function of the AF mass concentration C_{mj} in figure 4-25 to figure 4-30 for $\theta_{\text{ref}} = 15^\circ$. The angular domain in all these figures is $25^\circ \leq \theta \leq 45^\circ$ at 5° increments to avoid plot congestion. Nevertheless, the linear dependency of $\check{S}_y(\theta, C_{mj})$ to C_{mj} is apparent no matter which light source, filtered AF sand type and scattering angle is selected. Therefore all suspensions of filtered AF prepared for the scattering experiment satisfy the single scattering condition set in §4.3.1. Table 4-3 lists the average (over all statistically significant scattering angles excluding the reference angle) correlation coefficient ρ_m of the normalised particle scattering samples and the associated linear function that relates $\check{S}_y(\theta, C_{mj})$ to C_{mj} . As expected, ρ_m exceeds 98% in every possible combination of light source and filtered AF sand type.

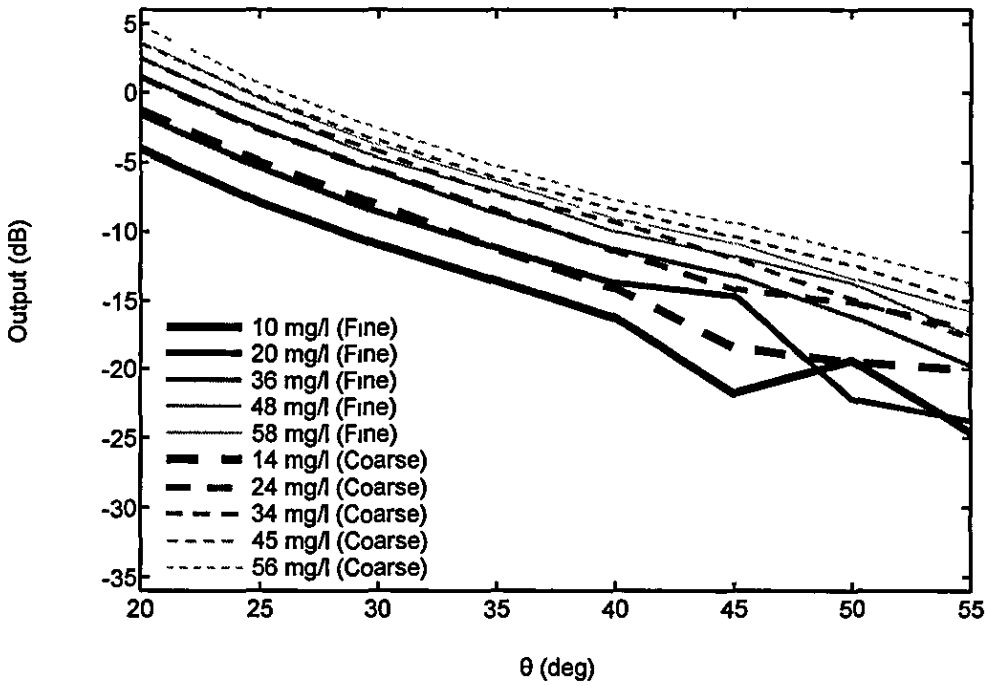


Figure 4-22 – Processed experimental scattering profiles of $AF_{F,C}$
(green LED, $\theta_{ref} = 15^\circ$)

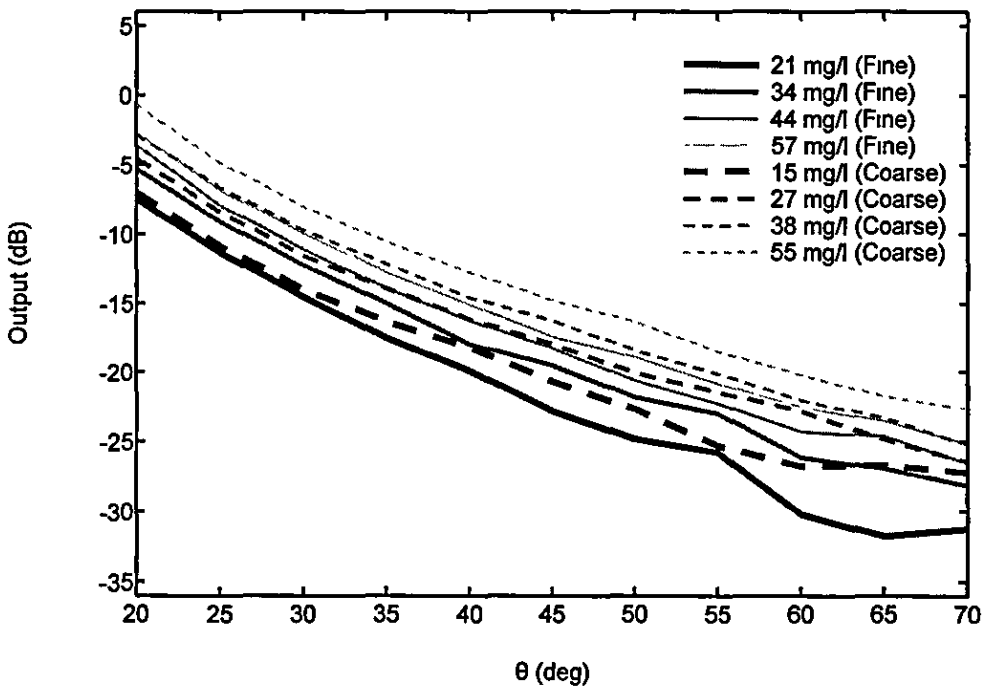


Figure 4-23 – Processed experimental scattering profiles of $AF_{F,C}$
(red LED, $\theta_{ref} = 15^\circ$)

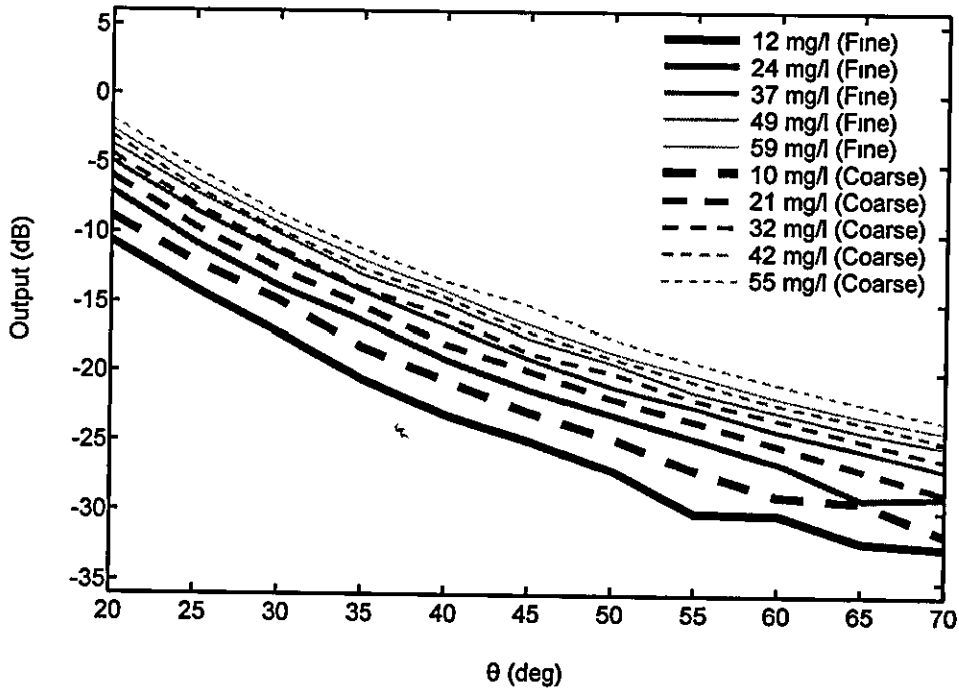


Figure 4-24 – Processed experimental scattering profiles of $AF_{F,C}$ (IR LED, $\theta_{ref} = 15^\circ$)

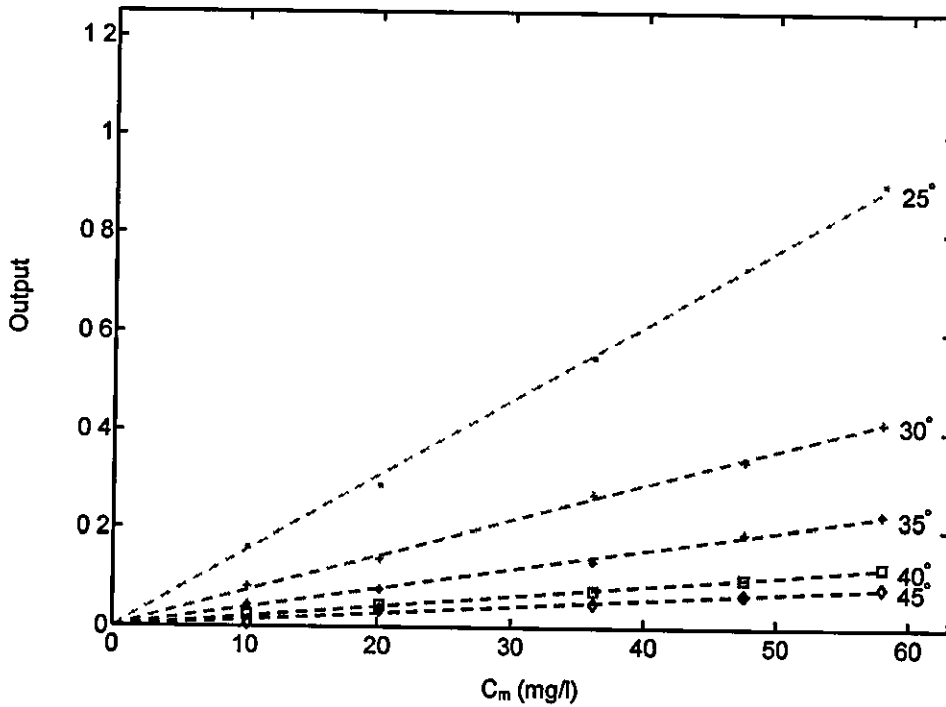


Figure 4-25 – Experimental scattering vs. AF_F mass concentration (green LED, $\theta_{ref} = 15^\circ$)

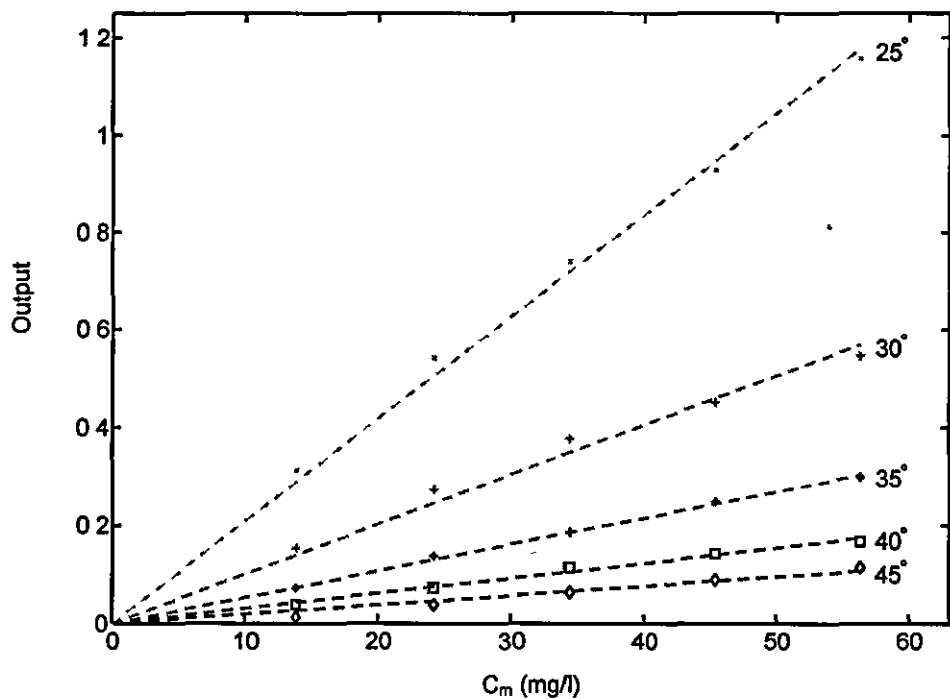


Figure 4-26 – Experimental scattering vs. AF_C mass concentration
(green LED, $\theta_{ref} = 15^\circ$)

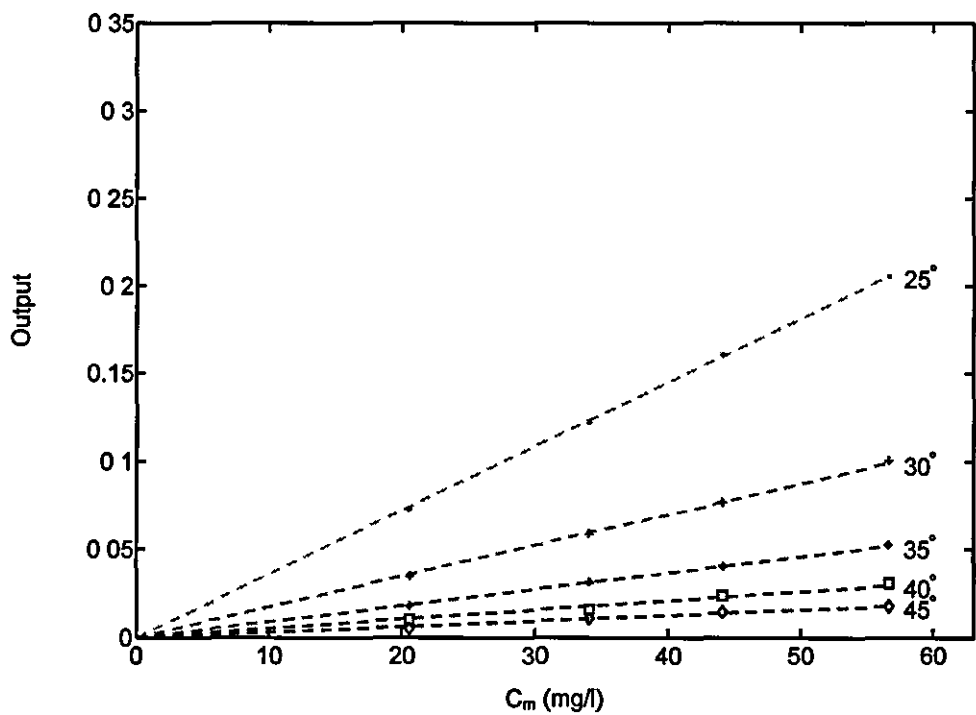


Figure 4-27 – Experimental scattering vs. AF_F mass concentration
(red LED, $\theta_{ref} = 15^\circ$)

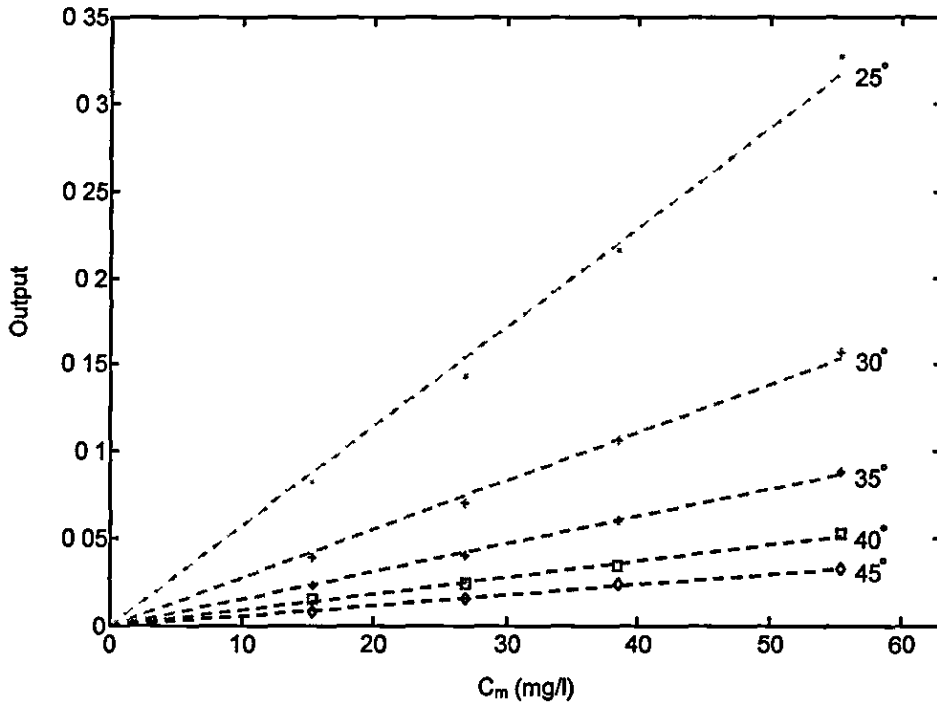


Figure 4-28 – Experimental scattering vs. AF_C mass concentration
(red LED, $\theta_{ref} = 15^\circ$)

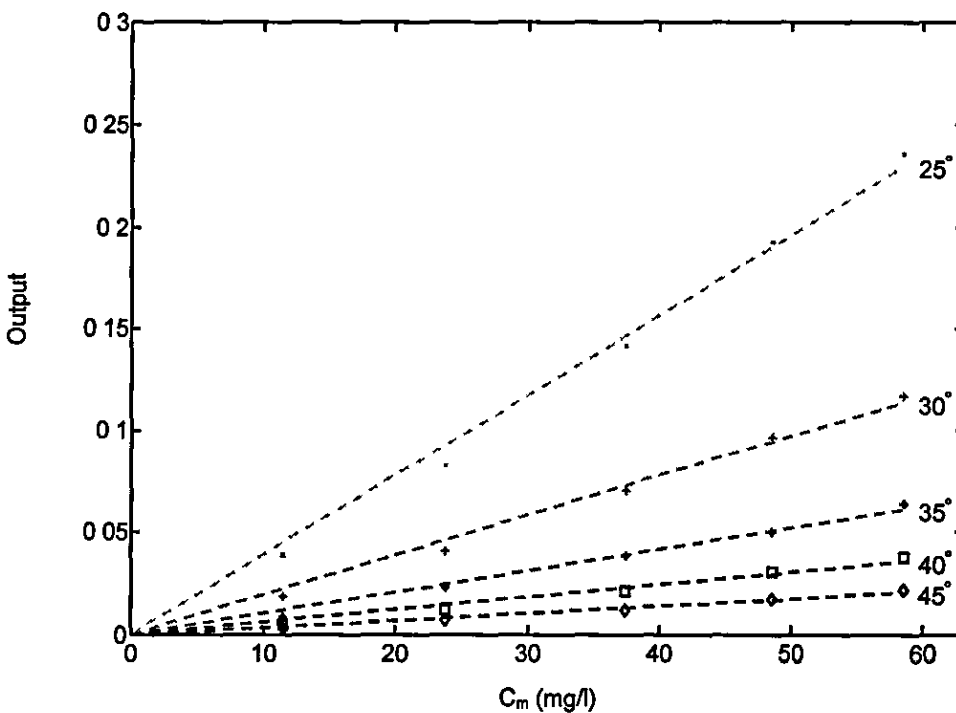


Figure 4-29 – Experimental scattering vs. AF_F mass concentration
(IR LED, $\theta_{ref} = 15^\circ$)

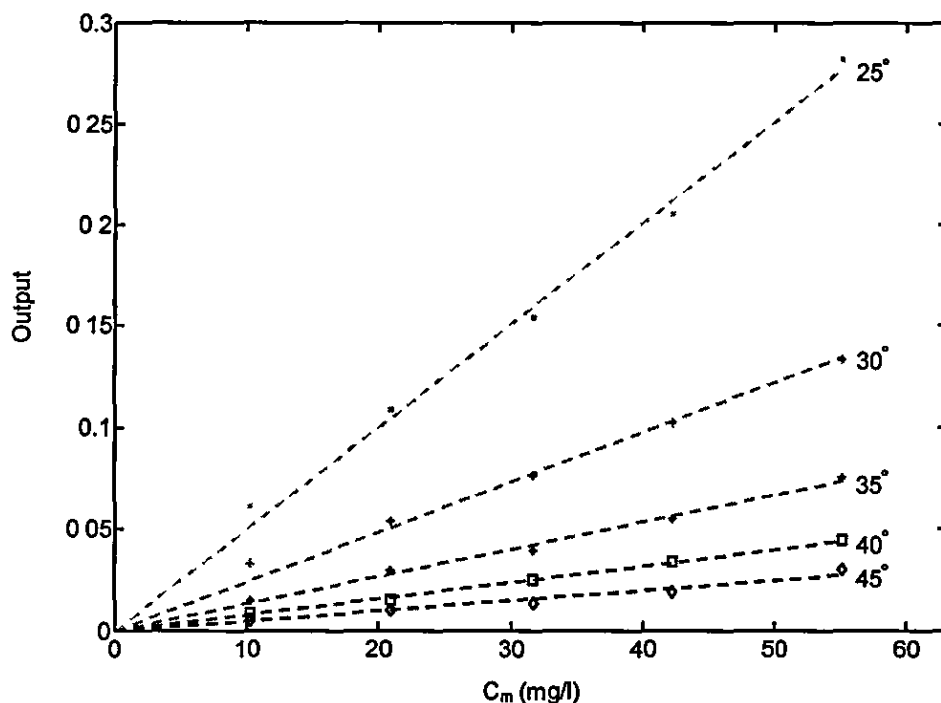


Figure 4-30 – Experimental scattering vs. AF_C mass concentration
(IR LED, $\theta_{ref} = 15^\circ$)

Table 4-3 – Correlation coefficient of scattering linearity with concentration (%)

Filtered AF Sand Type		Fine	Coarse
LED Type	Green	98.54	98.83
	Red	99.64	99.29
	Infrared	99.76	99.68

4.3.5 Modelling the experiment

The first goal a theoretical model of the scattering experiment should achieve is the determination of the scattered irradiance $I_{sj}(\theta, C_{mj})$ at the coupling point of the nephelometric detector with the solid-state detector. An equivalent goal is the calculation of the dimensionless *theoretical particle scattering* ${}_i\hat{S}_y(\theta, C_{mj})$ that is defined by:

$${}_i\hat{S}_y(\theta, C_{mj}) \equiv \frac{I_{sj}(\theta, C_{mj})}{I_{0i}} \quad (4-30)$$

where I_{0i} is the total irradiance produced by source i and emitted by the GRIN lens of the nephelometric illuminator. $I_{sj}(\theta, C_{mj})$ is the weighted sum of scattered irradiances

computed by equation (3-199) for every chemical compound of the filtered AF sand (see table 3-1). Therefore ${}_i\hat{S}_y(\theta, C_{mj})$ is given by:

$$\begin{aligned}
 {}_i\hat{S}_y(\theta, C_{mj}) = & \sum_{k=1}^8 mf_k \cdot \left\{ \frac{3C_{mj}w^2}{8\pi^2 d_m R \sin \theta} \cdot \right. \\
 & \int_{\lambda_{\min}}^{\lambda_{\max}} \frac{\text{src}_{ni}(\lambda) \text{rec}_{ni}(\lambda) \lambda^2}{n_k(\lambda)^2} \cdot \exp\left[-\frac{4\pi R \kappa_k(\lambda)(1 + \cos \theta)}{\lambda}\right] \cdot \\
 & \int_{\theta'_{\min}(\theta)}^{\theta'_{\max}(\theta)} \exp\left[-\frac{4\pi R \kappa_k(\lambda) \tan(\theta'/2) \sin \theta}{\lambda}\right] \cdot \\
 & \left. \int_{d_{\min}}^{d_{\max}} \frac{d_p p_j(x)}{x^3} \left[S_{11}(\lambda, \theta', x) + S_{12}(\lambda, \theta', x) \frac{Q_{0i}}{I_{0i}} \right] dx d\theta' d\lambda \right\} \quad (4-31)
 \end{aligned}$$

where:

- The set of eight (average) mass fractions mf_k that correspond to the chemical compounds of the filtered and ISO 12103-1 AF sand is stated in table 3-1.
- $d_m = 2400 \text{ kg/m}^3$, $R = 8.5 \text{ mm}$, $w = 1.5 \text{ mm}$ and $\delta = 4.7^\circ$ according to measurements taken before the experiment.
- The filtered AF mass concentration C_{mj} is listed in table 4-2.
- The complex refractive index $N_k(\lambda) = n_k(\lambda) + \kappa_k(\lambda)$ of the k th chemical compound of the filtered AF sand is assumed to be identical to the associated index of the ISO 12103-1 AF sand. Refractive indices $n_k(\lambda)$ and $\kappa_k(\lambda)$ are displayed in figure 3-6 and figure 3-7.
- The normalised spectral functions $\text{src}_{ni}(\lambda)$ and $\text{rec}_{ni}(\lambda)$ are determined by figure 4-17, figure 4-18 and equation (3-188). The integration limits are determined by the second figure ($\lambda_{\min} = 390 \text{ nm}$, $\lambda_{\max} = 940 \text{ nm}$).
- The integration limits $\theta'_{\min}(\theta)$ and $\theta'_{\max}(\theta)$ are given by equations (3-197) and (3-198).
- The particle size pdfs of the two filtered AF sand types are displayed in figure 4-13. The integration limits d_{\min} and d_{\max} are extracted from the same figure.
- The scattering functions $S_{11}(\lambda, \theta', x)$ and $S_{12}(\lambda, \theta', x)$ are calculated exactly with the aid of Mie scattering theory (see §3.4).
- The Stokes parameter Q_{0i} is assumed to be zero because the nephelometric sources in the scattering experiment are LEDs emitting unpolarised light.

To make the theoretical and experimental scattering outcomes comparable, ${}_i\hat{S}_y(\theta, C_{mj})$ has to be normalised with respect to $B_i(\theta_{ref})$ and the reference angle θ_{ref} should be identical to the one selected for the calculation of $\check{S}_y(\theta, C_{mj})$ ($\theta_{ref} = 15^\circ$). However, $B_i(\theta_{ref})$ is a random variable of unknown statistical properties and hence cannot be estimated analytically. An alternative approach is the use of a dimensionless *normalisation factor* β_i for the derivation of the dimensionless *theoretical normalised particle scattering* ${}_i\check{S}_y(\theta, C_{mj})$ from ${}_i\hat{S}_y(\theta, C_{mj})$. In mathematical terms:

$${}_i\check{S}_y(\theta, C_{mj}) \equiv \beta_i \cdot {}_i\hat{S}_y(\theta, C_{mj}) \quad (4-32)$$

The comparison of ${}_i\check{S}_y(\theta, C_{mj})$ with $\check{S}_y(\theta, C_{mj})$ can be facilitated if the functional dependency on the filtered AF mass concentration C_{mj} is removed. The single scattering condition set in the experimental sample preparation and verified by the plots of figure 4-25 to figure 4-30 implies that both scattering functions are linear to C_{mj} . Therefore the *normalised particle scattering per unit mass concentration* functions:

$${}_i\bar{S}_y(\theta) \equiv \frac{{}_i\check{S}_y(\theta, C_{mj})}{C_{mj}} \quad (4-33)$$

$$\bar{S}_y(\theta) \equiv E_{C_{mj}} \left[\frac{\check{S}_y(\theta, C_{mj})}{C_{mj}} \right] \quad (4-34)$$

are independent of C_{mj} and have inverse mass concentration units. The *expected value function* $E_{C_{mj}}(\cdot)$ returns the mean value of its input variable over the mass concentration domain.

The only problem that prevents function ${}_i\bar{S}_y(\theta)$ from being computed numerically by (4-31) to (4-33) is the determination of β_i in (4-32). The normalisation factor is selected to be optimal in the sense that the *Relative Mean-Squared Error* (RMSE) between ${}_i\bar{S}_y(\theta)$ and $\bar{S}_y(\theta)$ defined by:

$$RMSE_i \equiv \frac{E_{\theta} \{ [\bar{S}_{iF}(\theta) - {}_i\bar{S}_{iF}(\theta)]^2 \}}{E_{\theta} \{ [\bar{S}_{iF}(\theta)]^2 \}} + \frac{E_{\theta} \{ [\bar{S}_{iC}(\theta) - {}_i\bar{S}_{iC}(\theta)]^2 \}}{E_{\theta} \{ [\bar{S}_{iC}(\theta)]^2 \}} \quad (4-35)$$

is minimised. Therefore β_i is given by:

$$\beta_i = \frac{E_{\theta} \{ [\bar{S}_{iF}(\theta)]^2 \} E_{\theta} [{}_i\tilde{S}_{iC}(\theta) \bar{S}_{iC}(\theta)] + E_{\theta} \{ [\bar{S}_{iC}(\theta)]^2 \} E_{\theta} [{}_i\tilde{S}_{iF}(\theta) \bar{S}_{iF}(\theta)]}{E_{\theta} \{ [\bar{S}_{iF}(\theta)]^2 \} E_{\theta} \{ [{}_i\tilde{S}_{iC}(\theta)]^2 \} + E_{\theta} \{ [\bar{S}_{iC}(\theta)]^2 \} E_{\theta} \{ [{}_i\tilde{S}_{iF}(\theta)]^2 \}} \quad (4-36)$$

where the *theoretical particle scattering per unit mass concentration* ${}_i\tilde{S}_y(\theta)$ defined by:

$${}_i\tilde{S}_y(\theta) \equiv \frac{{}_i\hat{S}_y(\theta, C_{mj})}{C_{mj}} \quad (4-37)$$

is independent of C_{mj} and has inverse mass concentration units. The expected value function $E(\cdot)$ appearing in (4-35) and (4-36) returns the mean value of its input variable over the scattering angle domain.

The theoretical normalised scattering profiles derived by equations (4-31) and (4-32) are displayed in figure 4-31 to figure 4-33 for the three nephelometric illumination systems. The optimal values of the normalisation factor are $\beta_G = 2.8 \cdot 10^8$, $\beta_R = 1.8 \cdot 10^8$ and $\beta_{IR} = 3.6 \cdot 10^{10}$. The angular domains of ${}_i\tilde{S}_y(\theta, C_{mj})$ in these figures are identical to the ones of $\bar{S}_y(\theta, C_{mj})$ for comparison purposes. A side-to-side comparison of the two scattering profiles shows that ${}_i\tilde{S}_y(\theta, C_{mj})$ decays more smoothly than $\bar{S}_y(\theta, C_{mj})$ and AF_F suspensions scatter more light than AF_C suspensions of similar mass concentrations. These differences are the subject of the following analysis.

Functions ${}_i\bar{S}_y(\theta)$ and $\bar{S}_y(\theta)$ are presented together in figure 4-34 to figure 4-36 according to light source type. The intervals of θ displayed in these figures are the statistically significant parts of the scattering angle domain for each source type. The correlation coefficients ρ_{θ} and relative errors RMSE between the theoretical and experimental scattering curves are listed in the first data rows of table 4-4 and table 4-5. The correlation of these functions is strong (>93%) no matter which filtered AF and light source type is considered. However, RMSE is unacceptably large for the AF_C suspensions (>64%). Moreover, the experimental curves are always steeper than the theoretical ones. Another noticeable difference between theoretical and experimental analysis is that ${}_i\bar{S}_{iF}(\theta) > {}_i\bar{S}_{iC}(\theta)$ but $\bar{S}_{iF}(\theta) < \bar{S}_{iC}(\theta) \forall (i, \theta)$. Consequently, the theoretical model of the scattering experiment explained so far is reasonably accurate but there is still room for improvement.

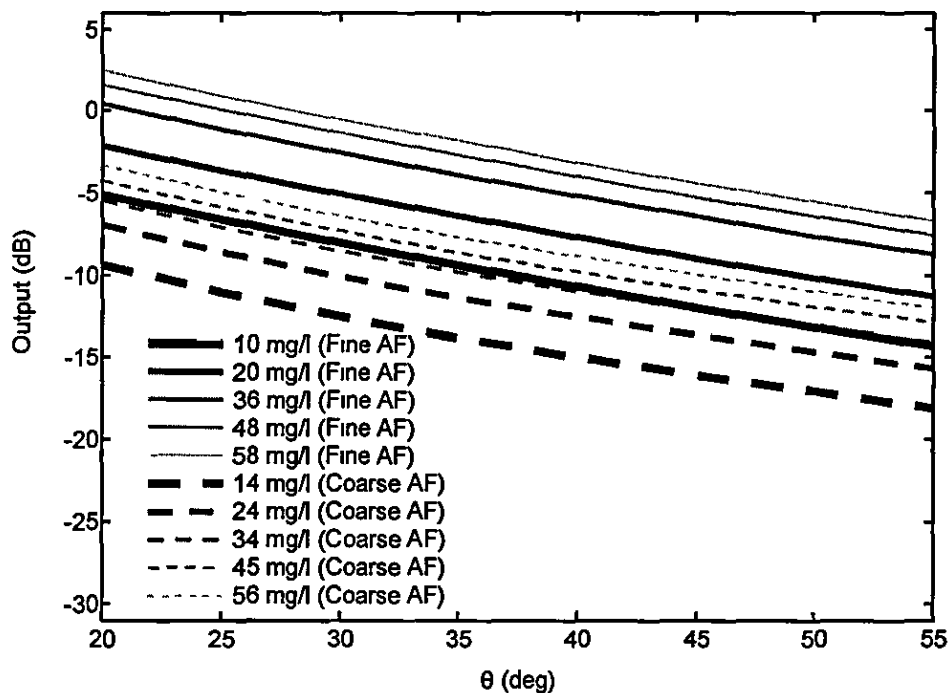


Figure 4-31 – Theoretical scattering profiles of $AF_{F,C}$ (green LED)

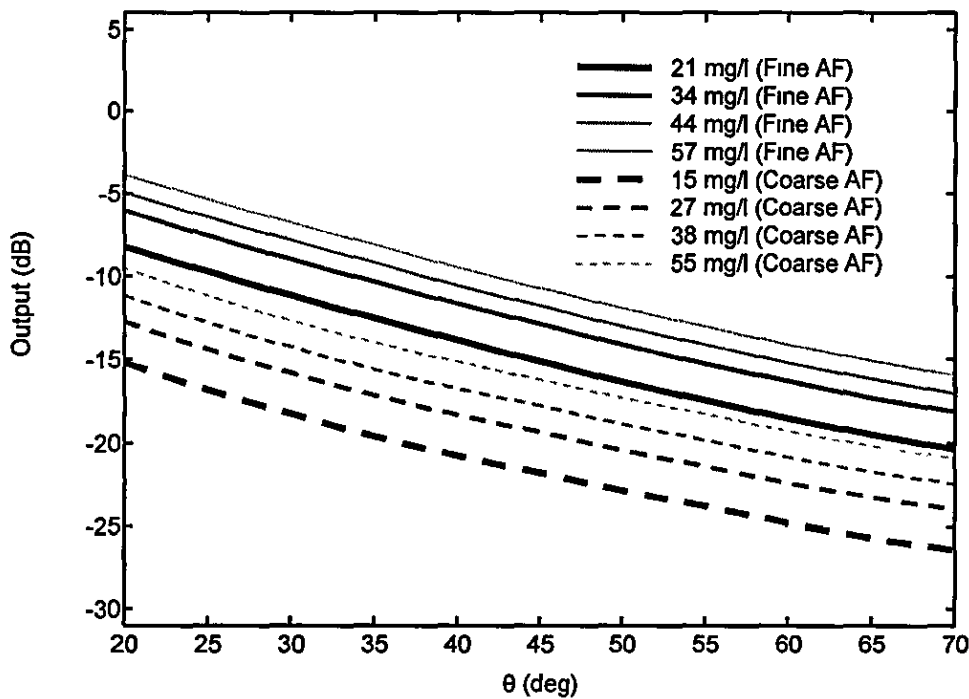


Figure 4-32 – Theoretical scattering profiles of $AF_{F,C}$ (red LED)

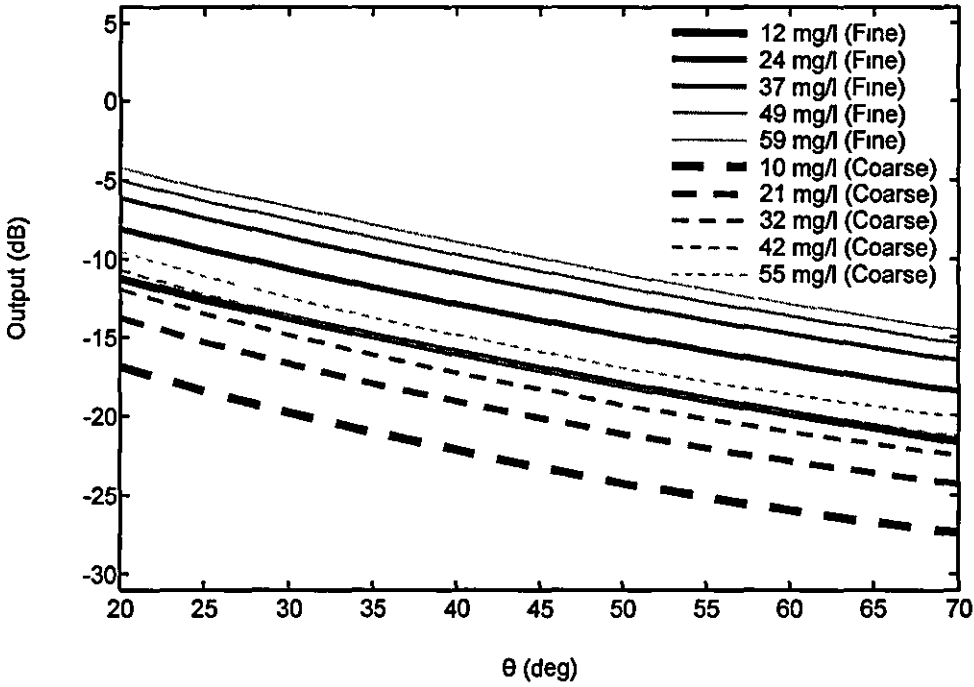


Figure 4-33 – Theoretical scattering profiles of AF_{F,C} (IR LED)

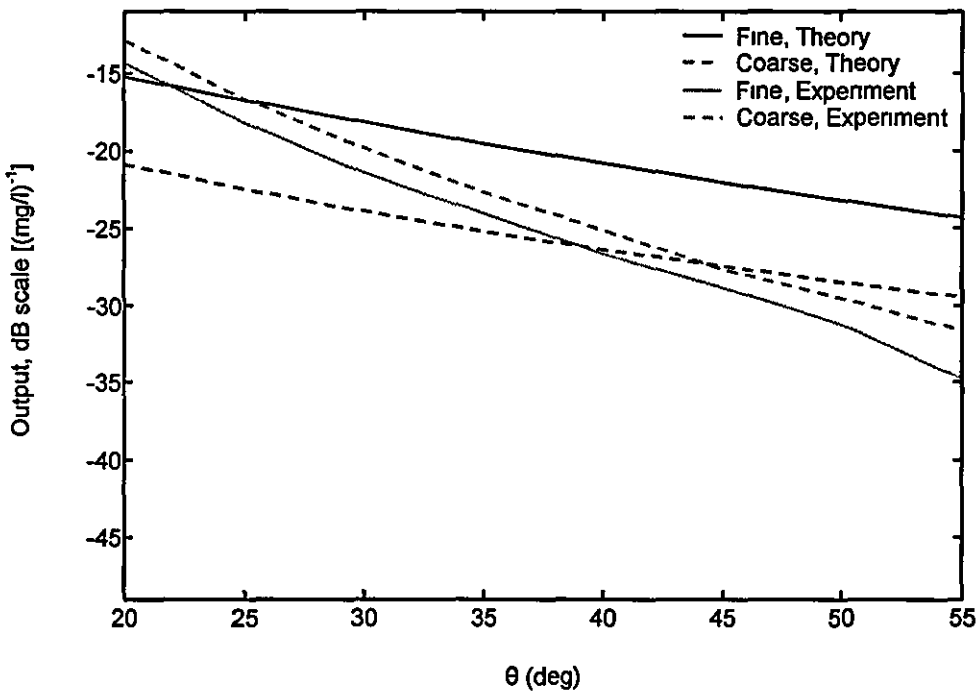


Figure 4-34 – Normalised scattering profiles of AF_{F,C} per AF_{F,C} mass concentration (green LED)

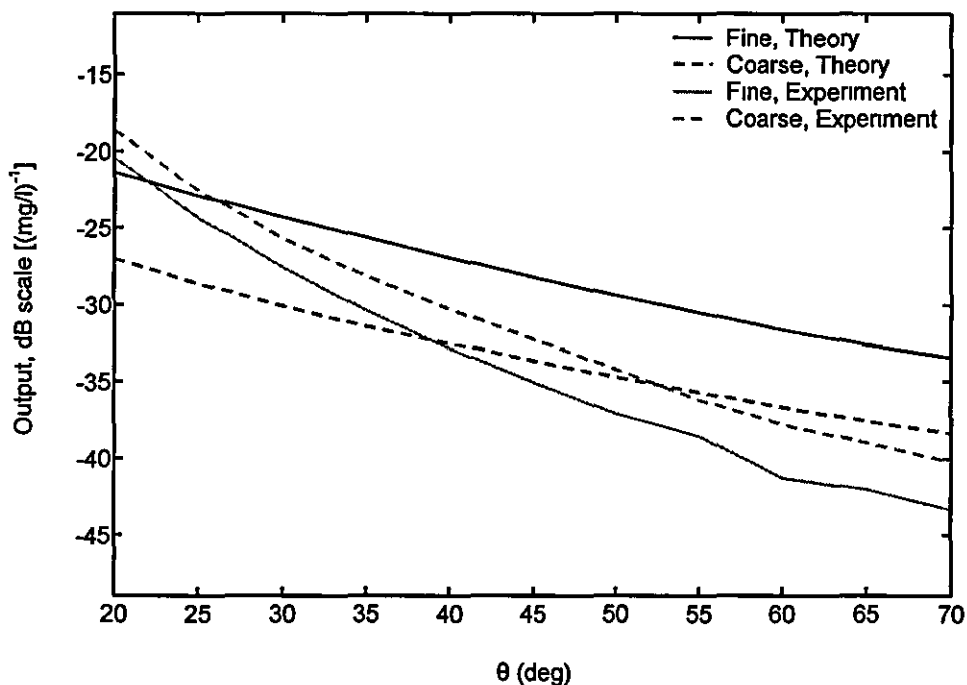


Figure 4-35 – Normalised scattering profiles of $AF_{F,C}$ per $AF_{F,C}$ mass concentration (red LED)

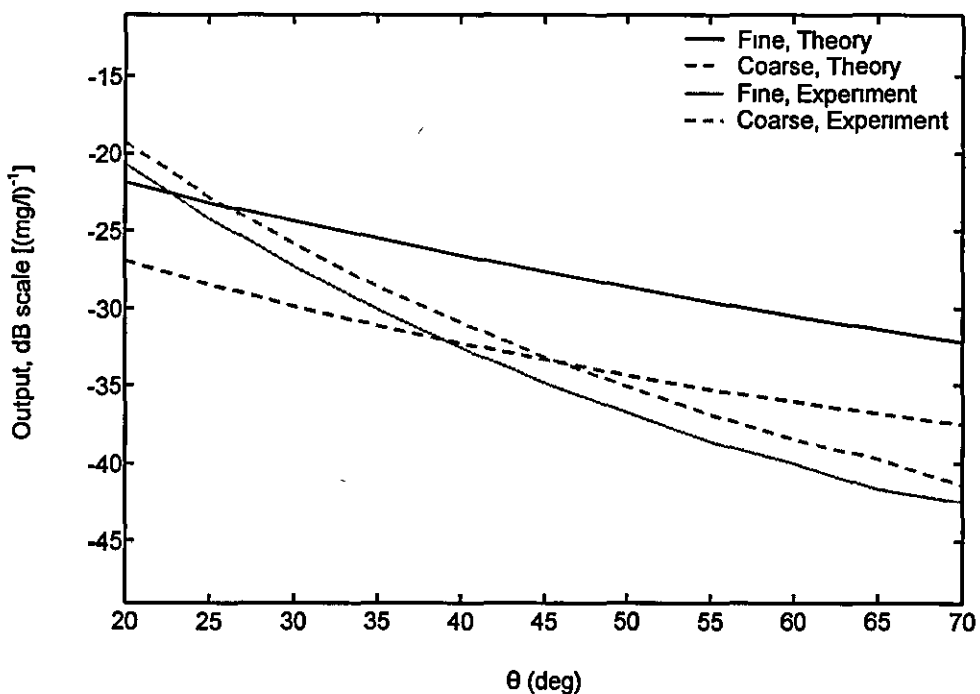


Figure 4-36 – Normalised scattering profiles of $AF_{F,C}$ per $AF_{F,C}$ mass concentration (IR LED)

Table 4-4 – Correlation coefficient of theoretical and trial scattering profiles (%)

LED Type		Green		Red		Infrared	
Filtered AF Sand Type		Fine	Coarse	Fine	Coarse	Fine	Coarse
Particle Size Range (μm)	No limits	95.2	96.2	93.6	94.8	93.4	95.3
	< 5	95.1	95.6	93.5	94.2	93.3	94.8
	> 0.5	95.7	97.3	94.7	96.5	93.8	96.6
	0.5 - 5	95.6	96.9	94.5	96.2	93.7	96.1
	> 1	99.4	96.2	98.5	94.8	98.6	97.9
	1 - 5	99.1	98.0	98.0	96.9	98.8	98.4
	> 2	96.2	95.0	94.3	93.8	94.0	94.8
	2 - 5	96.2	96.3	94.3	94.7	94.0	93.9

Table 4-5 – Relative error between theoretical and trial scattering profiles (%)

LED Type		Green		Red		Infrared	
Filtered AF Sand Type		Fine	Coarse	Fine	Coarse	Fine	Coarse
Particle Size Range (μm)	No limits	17.5	66.8	18.5	69.0	21.7	64.7
	< 5	18.0	59.6	19.1	62.1	22.3	59.4
	> 0.5	12.8	75.4	13.6	77.1	17.9	70.7
	0.5 - 5	13.7	69.5	14.4	71.6	18.7	65.7
	> 1	9.9	27.6	11.3	30.6	8.7	23.2
	1 - 5	9.1	13.0	10.9	14.4	6.3	8.7
	> 2	31.5	21.7	30.3	21.3	33.6	19.1
	2 - 5	20.8	17.8	21.0	17.8	21.7	19.0

The small difference between $\bar{S}_{IF}(\theta)$ and $\bar{S}_{IC}(\theta)$ in figure 4-34 to figure 4-36 implies that the particle size pdfs of AF_F and AF_C sands are actually similar. This observation conflicts with figure 4-13 which shows the two pdfs being quite different. The contradiction can be resolved by assuming that particle *aggregation* and particle *sedimentation* have taken place before the scattering experiment. Specifically, the aggregation of small AF particles clips the left tails of the two pdfs, while the sedimentation of large AF particles cuts off the right ends of the same functions. The problem of estimating the size distribution of particles created by aggregation of smaller particles can be avoided by assuming further that aggregated particles sediment to the bottom of the nephelometric glass container and thus do not scatter light. The combined effect of aggregation and sedimentation on the size pdf of filtered AF sand is therefore the application of a size filter to that pdf and the decrease of suspended mass concentration by the percentage of pdf area removed due to filtering.

Seven cases of AF particle aggregation and sedimentation were studied in total. A different rectangular window filter was applied to the volume probability distribution functions illustrated in figure 4-10 in every individual case. The passbands of these filters are stated in the first data columns of table 4-4 and table 4-5. Functions ${}_d p_j(\cdot)$ and ${}_i \bar{S}_y(\cdot)$ were recalculated for every filtering case by the same analytical procedures explained in §4.3.1 and this section. The correlation coefficients and relative errors between the modified ${}_i \bar{S}_y(\cdot)$ and $\bar{S}_y(\cdot)$ are listed in the remaining data columns of table 4-4 and table 4-5. The tabular data demonstrate that:

- a) The sedimentation hypothesis of large AF particles does not alter the theoretical scattering profiles significantly. In contrast, the aggregation hypothesis of small AF particles has much greater influence on the shape of these profiles.
- b) The fifth pdf filtering case (1-5 μm) gives the smallest RMSE regardless of filtered AF and light source type. The same instance features the highest ρ_0 values in four out of six filtered AF and light source type combinations.

The *limited particle volume probability distribution function* ${}_v p_j^s(\cdot)$ and the associated *limited particle size probability density function* ${}_d p_j^s(\cdot)$ of the fifth pdf filtering case are displayed in figure 4-37 and figure 4-38. As expected, the trimmed pdfs of AF_F and AF_C sand types are quite similar. However, the hypothetical sedimen-

tation of small ($<1 \mu\text{m}$) and large ($>5 \mu\text{m}$) particles before the scattering experiment leaves only 27% (44%) of the dispersed AF_F (AF_C) sand volume to suspension.

The theoretical normalised particle scattering function ${}_i\check{S}_j^5(\cdot)$ that is derived from ${}_i p_j^5(\cdot)$ by equations (4-31) and (4-32) is illustrated in figure 4-39 to figure 4-41 for the three nephelometric illumination systems and the statistically significant scattering angles. The optimal values of the normalisation factor are $\beta_G^5 = 5.6 \cdot 10^9$, $\beta_R^5 = 3.7 \cdot 10^9$ and $\beta_{IR}^5 = 4.6 \cdot 10^{11}$. Theory and experiment now agree that most AF_C suspensions scatter more light than AF_F suspensions of similar mass concentrations.

The theoretical normalised particle scattering per unit mass concentration ${}_i\bar{S}_j^5(\cdot)$ that is derived from ${}_i\hat{S}_j^5(\cdot)$ by equation (4-33) is displayed in figure 4-42 to figure 4-44 for the three nephelometric illuminator types and the statistically significant parts of the scattering angle domain. Function ${}_i\bar{S}_j^5(\cdot)$ is much closer to $\bar{S}_j(\cdot)$ than the original ${}_i\bar{S}_j(\cdot) \forall (i, j)$ as expected ($\rho_0 \geq 97\%$, $\text{RMSE} < 15\%$). The only noticeable difference between the modified theoretical and experimental scattering profiles is that $\bar{S}_j(\cdot)$ decays faster than ${}_i\bar{S}_j^5(\cdot)$ in all circumstances. That difference however is small and can be attributed to errors in measurement that increase at higher scattering angles because of lower signal-to-noise ratios. Therefore ${}_i\bar{S}_j^5(\cdot)$ is an excellent theoretical approximation of the associated experimental scattering profile.

4.3.6 Filtered AF sand type detection results

The normalised particle scattering approximation (4-29) does not require information about the type index j and mass concentration C_{mj} of the filtered AF dispersed in any experimental sample. Therefore $\check{S}_j(\cdot)$ can be computed directly from the raw experimental scattering profiles of a given sample and then employed in the training and optimisation of an ANN structure that estimates either filtered AF parameter (j or C_{mj}) in the sample. This principle is applied in §4.3.6.1 and §4.3.6.2 to solve the filtered AF sand type detection problem in two ways. The first method requires a single MFANN to perform the sample classification task directly from analytical $\check{S}_j(\cdot)$ approximations. The second approach leads to a pair of cascaded MFANNs that provide both C_{mj} estimation and filtered AF sand type information.

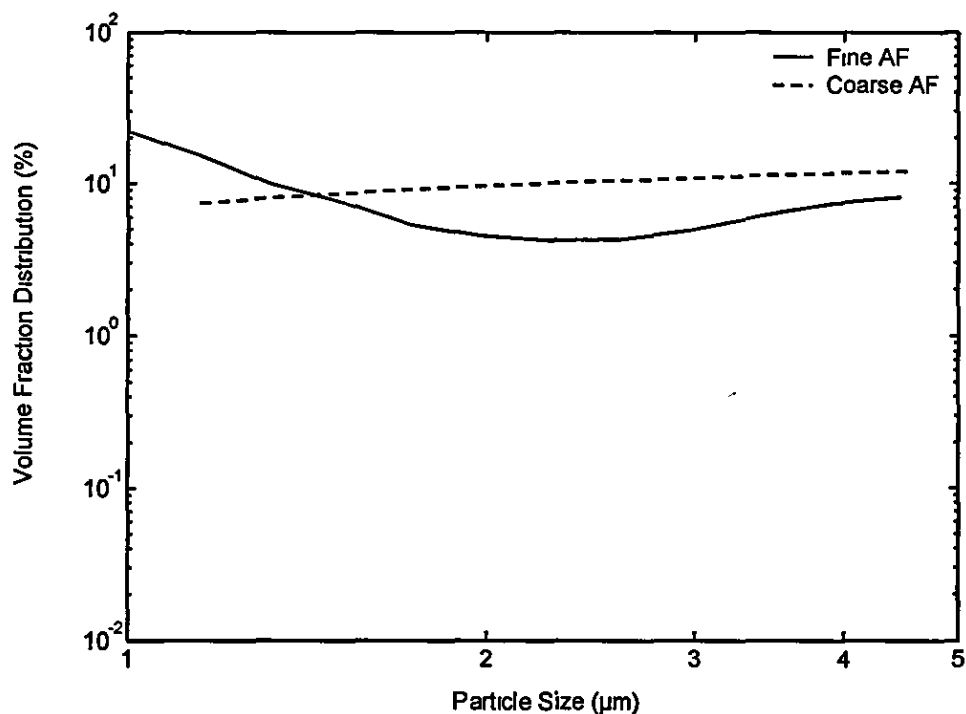


Figure 4-37 – Limited particle volume probability distribution functions of $AF_{F,C}$

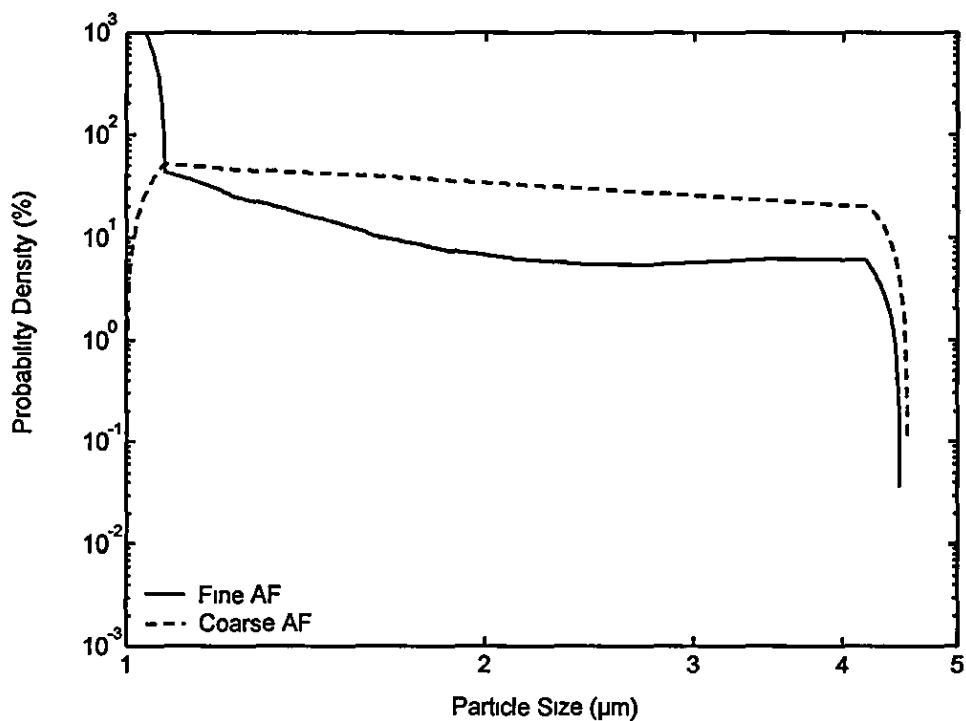


Figure 4-38 – Limited particle size probability density functions of $AF_{F,C}$

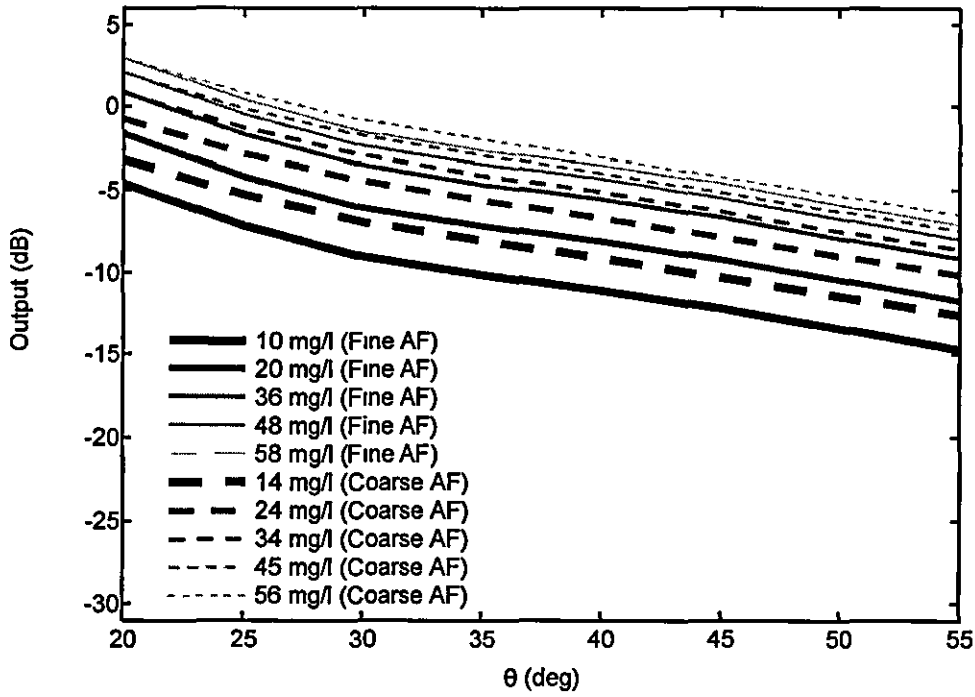


Figure 4-39 – Theoretical scattering profiles of suspended $AF_{F,C}$ (green LED)

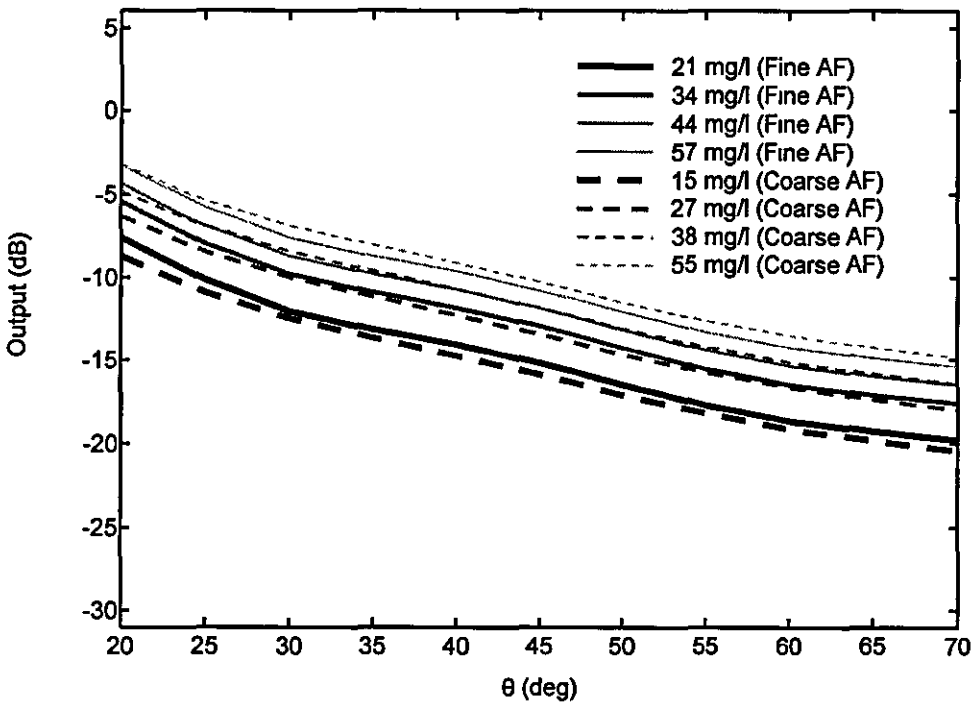


Figure 4-40 – Theoretical scattering profiles of suspended $AF_{F,C}$ (red LED)

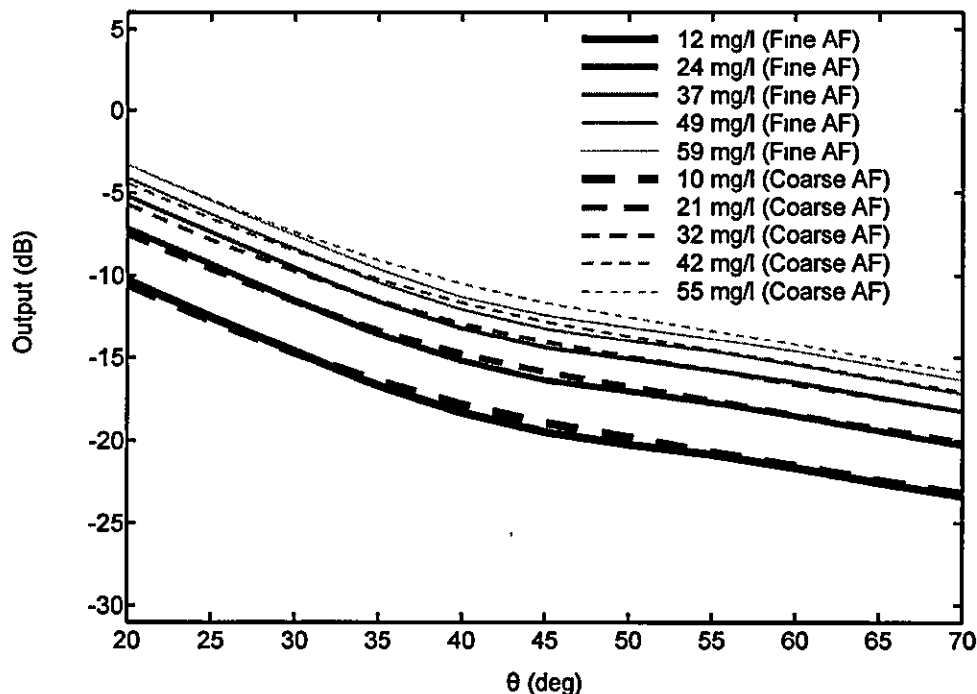


Figure 4-41 – Theoretical scattering profiles of suspended $AF_{F,C}$ (IR LED)

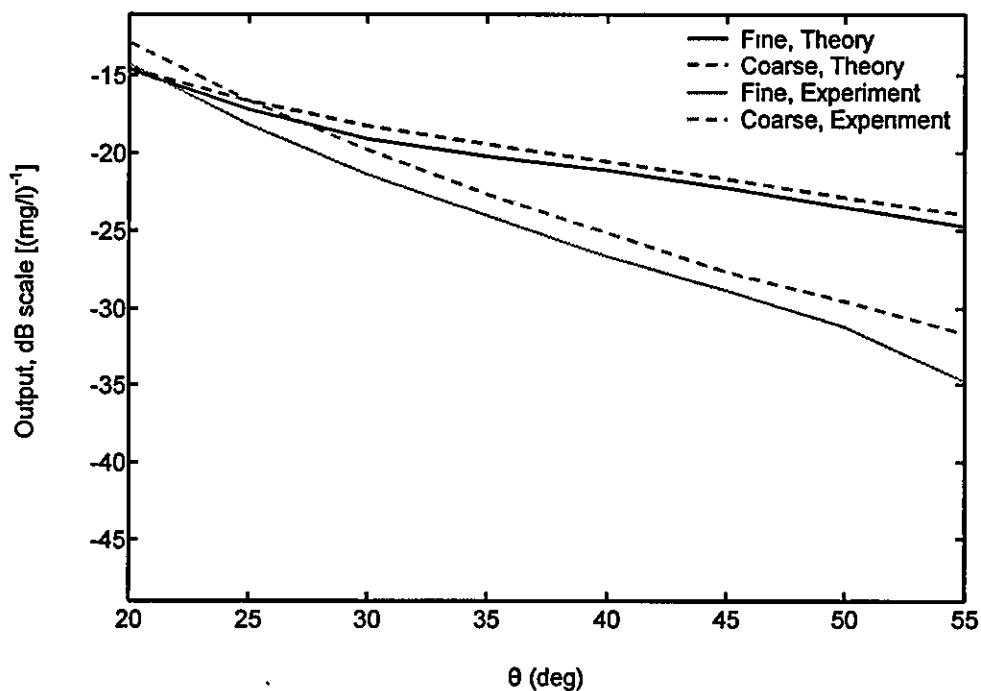


Figure 4-42 – Normalised scattering profiles of suspended $AF_{F,C}$ per $AF_{F,C}$ mass concentration (green LED)

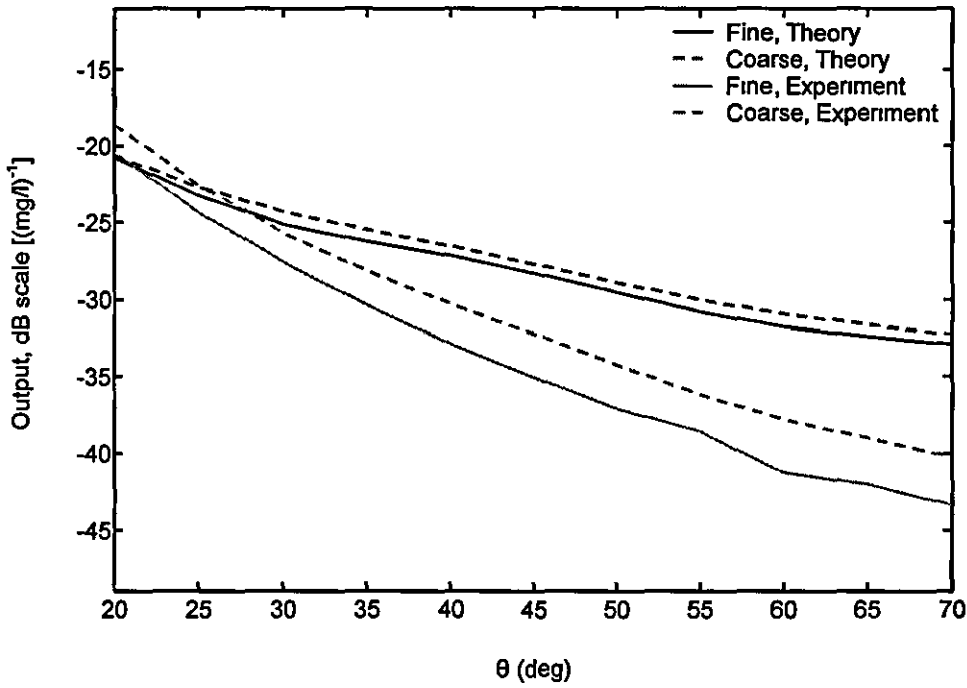


Figure 4-43 – Normalised scattering profiles of suspended $AF_{F,C}$ per $AF_{F,C}$ mass concentration (red LED)

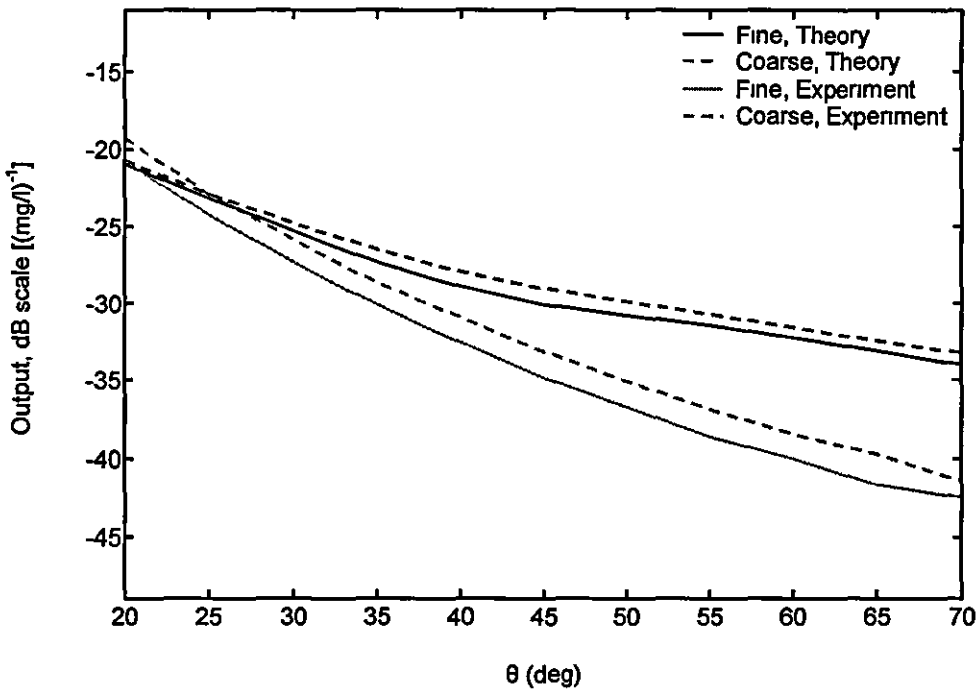


Figure 4-44 – Normalised scattering profiles of suspended $AF_{F,C}$ per $AF_{F,C}$ mass concentration (IR LED)

4.3.6.1 First solution

An initial 30-1-1 MFANN is constructed to solve the filtered AF sand type detection problem. Every MFANN input node corresponds to a pair (θ, i) where θ represents a statistically significant scattering angle and i stands for the light source index ($i = G, R, \text{ or } IR$). The hidden node is assigned the log-sigmoid transfer function while the output node is designed to have a linear transfer function. The output node returns a uniformly normalised value that indicates the type of filtered AF sand being dispersed in a given experimental sample (negative for AF_F and nonnegative for AF_C).

The input and output vector sets required to train, test and optimise the aforementioned ANN structure include 28 elements each. The set of 30×1 raw input vectors $\{p_1, p_2, \dots, p_q, \dots, p_{28}\}$ is made from the columns of 30×28 matrix P given by:

$$P = [M_F \mid M_C] \quad (4-38)$$

The 30×14 matrices M_F, M_C in (4-38) are derived by the definition:

$$M_j \equiv \begin{bmatrix} \check{S}_{1j}(\theta, C_{mj}) & 0 & 0 \\ 0 & \check{S}_{2j}(\theta, C_{mj}) & 0 \\ 0 & 0 & \check{S}_{3j}(\theta, C_{mj}) \end{bmatrix} \quad (4-39)$$

where the normalised particle scattering samples $\check{S}_{ij}(\theta, C_{mj})$ are arranged in three matrices of sizes 8×5 , 11×4 and 11×5 in light source order. The set of 1×1 raw target vectors $\{t_1, t_2, \dots, t_q, \dots, t_{28}\}$ is consisted of the elements of 1×28 row vector T given by:

$$T = [0 \mid 1] \quad (4-40)$$

with 0 and 1 being two 1×14 row vectors of elements equal to zero and one respectively. In other words, the existence of fine AF sand in the experimental sample should minimise the MFANN output ($t_q < 0.5$), while the presence of coarse AF sand in the suspension is expected to maximise the single network output ($t_q \geq 0.5$).

The raw input and target vector sets are uniformly normalised to the interval $[-0.5, 0.5]$ by applying the appropriate pre-processing algorithm of §2.8. The 56 normalised vectors \tilde{p}_q, \tilde{t}_q are grouped in 28 pairs of the form $\{\tilde{p}_q, \tilde{t}_q\}$. Fourteen vector pairs are randomly selected for network training use (*training pattern set*) and the remaining 14 pairs are destined to measure the degree of generalisation exhibited by the trained network at the end of every ANN optimisation step (*testing pattern set*).

The complete MFANN optimisation scheme described in §2.9.4 is employed in the processing of the initial network structure. The ANN training task is performed

with the aid of the LMBP algorithm and the network optimisation goal is achieved via the modified SOSA and OBS algorithms. The performance index of the training algorithm is the MSE function defined by (4-10). The LMBP parameters are: $MSE_{goal} = 10^{-6}$, $Ep_{max} = 100$, $\nabla_{min} = 10^{-6}$, $\mu_0 = 10^{-3}$, $\mu_{max} = 10^{10}$, and $\theta = 10$. The optimisation parameters are: $c_1 = +\infty$, $c_2 = 2$, $c_3 = 3$, $\alpha = 10^{-6}$ and $N_{max}^+ = 0$. The final model is a complete 6-1-1 MFANN having 7 weights and 2 biases.

Table 4-6 lists the number of training epochs performed and the positions of ANN input nodes removed per optimisation step. All removed nodes are denoted by the associated scattering angles θ and source indices i in the form θ^i . Most optimisation steps remove the maximum number of nodes allowed (3) in less than 20 epochs.

Table 4-6 – MFANN training and optimisation results

Optimisation Step	Training Epochs	Removed Nodes
1 (training only)	19	—
2	3	$20^R, 40^R, 55^R$
3	1	50^R
4	3	$35^{IR}, 50^{IR}, 70^R$
5	8	$20^G, 25^G, 30^G$
6	7	$25^{IR}, 40^{IR}, 65^{IR}$
7	3	35^R
8	11	$25^R, 60^R, 65^R$
9	10	$20^{IR}, 30^{IR}, 55^{IR}$
10 (2 backtracks)	31	45^{IR}
11	19	$35^G, 50^G, 55^G$

The relative number of patterns classified erroneously by an arbitrary ANN model is called *Relative Classification Error* and denoted by the acronym RCE. In the case of the network structure described above, a RCE corresponds to every MFANN output state. These errors are denoted by the acronyms RCE_F (for fine AF) and RCE_C (for coarse AF), and they are mathematically defined by the formulas:

$$RCE_F \equiv \frac{\sum_{q=1}^Q [(\tilde{t}_q < 0) \wedge (\tilde{o}_q \geq 0)]}{\sum_{q=1}^Q (\tilde{t}_q < 0)} \quad (4-41)$$

$$RCE_C \equiv \frac{\sum_{q=1}^Q [(\tilde{t}_q \geq 0) \wedge (\tilde{o}_q < 0)]}{\sum_{q=1}^Q (\tilde{t}_q \geq 0)} \quad (4-42)$$

where:

- \tilde{o}_q is the uniformly normalised output vector associated with the set $\{\tilde{p}_q, \tilde{t}_q\}$
- the outcome of logical operators $<, \geq$ and \wedge (AND) is 0 for false and 1 for true
- $Q = 14$ for either pattern set (training or testing)

Figure 4-45 illustrates the values of RCE_j for the MFANN model described above after each network optimisation step. The applied optimisation scheme always manages to adapt perfectly the network model to the training pattern set ($RCE_j = 0$ for that set). As far as the testing pattern set is concerned, RCE_F drops to zero after the fourth optimisation step while RCE_C converges to 28.6% at the final step. The large RCE_C value implies that more scattering measurements of coarse AF sand suspensions are required to train the MFANN further in detecting both sand types correctly.

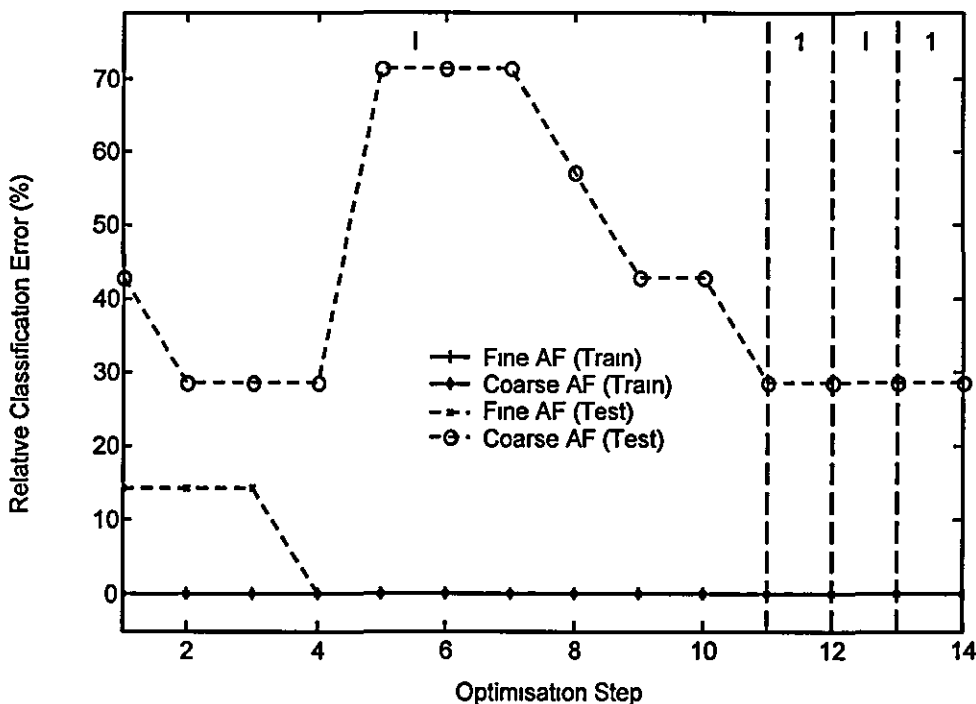


Figure 4-45 – Relative classification error change per MFANN optimisation step

The angular positions of the three nephelometric sources and four nephelometric detectors displayed in figure 4-46 and supplying the final network model with scattering data are obtained from the (θ, i) associations of the final MFANN input nodes. Green LED is placed at 170° , red LED at 180° , IR LED at 150° , and the detectors at 30° , 35° , 40° and 45° with respect to the direction of propagation of light emitted by the red light source. When a sample suspension is placed in the nephelometer the light sources start flashing alternatively. The signal returned by the detector at 30° is time-division demultiplexed, while the signals returned by the remaining detectors are accepted only at times the associated light sources are on ($35^\circ \leftrightarrow$ green LED, $40^\circ \leftrightarrow$ IR LED, $45^\circ \leftrightarrow$ red LED). The derived signals are rectified, sampled and uniformly normalised before they are passed as inputs to the network model. Three 6×1 input vectors are made from the six values obtained ($\tilde{p}_1 = 30^G$, $\tilde{p}_2 = 35^G$, $\tilde{p}_3 = 30^R$, $\tilde{p}_4 = 45^R$, $\tilde{p}_5 = 30^{IR}$, $\tilde{p}_6 = 40^{IR}$). Every input vector includes data related to a particular source ($\tilde{p}_1 = [\tilde{p}_1 \tilde{p}_2 0 0 0 0]^T$, $\tilde{p}_2 = [0 0 \tilde{p}_3 \tilde{p}_4 0 0]^T$, $\tilde{p}_3 = [0 0 0 0 \tilde{p}_5 \tilde{p}_6]^T$). The input vectors are sequentially applied to the MFANN. If most of the uniformly normalised values returned by the network are negative, the dispersed material in the experimental sample is recognised as AF_F ; otherwise the dispersed material is detected as AF_C .

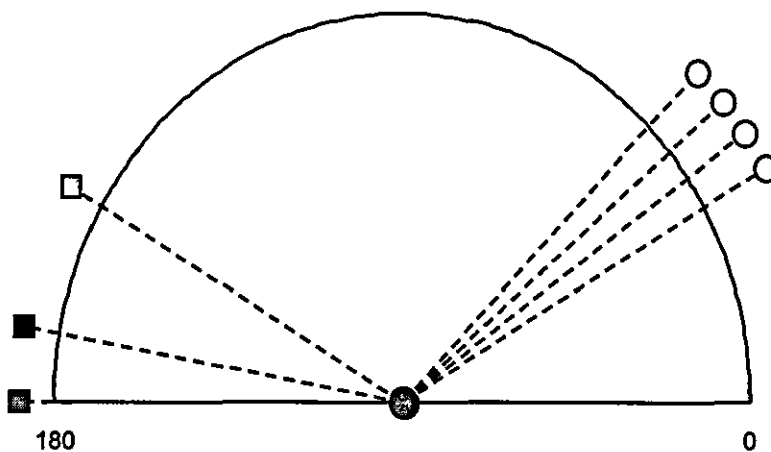


Figure 4-46 – Structurally optimal nephelometric sensor configuration

4.3.6.2 Second solution

Two MFANNs are constructed, trained and optimised independently to perform different but interrelated modelling tasks. The first network accepts samples from the normalised particle scattering profiles of the experimental samples and returns an approximation \hat{C}_{mj} of the total mass concentration C_{mj} of filtered AF sand for every sample concerned. Next, the inputs of the first network are divided by its outputs and the outcomes $\check{S}_{ij}(\theta, C_{mj})/\hat{C}_{mj}$ are used as inputs to the second network. As $\check{S}_{ij}(\theta, C_{mj})$ has a linear relationship with C_{mj} (see figure 4-25 to figure 4-30, pages 186-189), the aforementioned ratios are considered independent of C_{mj} but dependent on the filtered AF sand type. Therefore the second network that is trained to detect the type of filtered AF sand dispersed in a sample is more likely to give better results than the single MFANN model of the first solution because it does not have to consider mass concentration from the start. That judgment is demonstrated by the following analysis.

An initial 30-1-1 MFANN is constructed to become the first part of the overall solution to this particle characterisation problem. Every input node corresponds to a light source index i and a statistically significant scattering angle θ . The hidden node is assigned the log-sigmoid transfer function while the output node is designed to have a linear transfer function. The output node returns uniformly normalised approximations of the total mass concentration of filtered AF sand in every experimental sample.

The input and output vector sets required to train, test and optimise the first MFANN include 28 elements each. The set of 30x1 raw input vectors $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_q, \dots, \mathbf{p}_{28}\}$ is made from the columns of 30x28 matrix \mathbf{P} given by equation (4-38) - see also approximation (4-29) and definition (4-39). The set of 1x1 raw target vectors $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_q, \dots, \mathbf{t}_{28}\}$ is consisted of the elements of 1x28 row vector \mathbf{T} given by:

$$\mathbf{T} = \left[\mathbf{C}_{mF}^T \mid \mathbf{C}_{mC}^T \right] \quad (4-43)$$

where \mathbf{C}_{mF} and \mathbf{C}_{mC} are two 14x1 vectors having as elements the values of the second and third column of table 4-2 (see page 172).

The raw input and target vector sets are uniformly normalised to the interval [-0.5, 0.5] by applying the appropriate pre-processing algorithm of §2.8. The 56 normalised vectors $\tilde{\mathbf{p}}_q, \tilde{\mathbf{t}}_q$ are grouped in 28 pairs of the form $\{\tilde{\mathbf{p}}_q, \tilde{\mathbf{t}}_q\}$. Fourteen vector

pairs are randomly selected for network training use (*training pattern set*) and the remaining 14 pairs are destined to measure the degree of generalisation exhibited by the trained network at the end of every ANN optimisation step (*testing pattern set*).

The complete MFANN optimisation scheme described in §2.9.4 is employed in the processing of the first network structure. The ANN training task is performed with the aid of the LMBP algorithm and the network optimisation goal is achieved via the modified SOSA and OBS algorithms. The performance index of the training algorithm is the MSE function defined by (4-10). The LMBP parameters are: $MSE_{goal} = 10^{-2}$, $Ep_{max} = 100$, $\nabla_{min} = 10^{-6}$, $\mu_0 = 10^{-3}$, $\mu_{max} = 10^{10}$, and $\theta = 10$. The optimisation parameters are: $c_1 = +\infty$, $c_2 = 2$, $c_3 = 3$, $\alpha = 10^{-6}$ and $N_{max}^+ = 0$. The final network is 3-1-1 with 4 weights and 2 biases (complete MFANN).

Table 4-7 lists the number of training epochs performed and the positions of ANN input nodes removed per optimisation step. All removed nodes are denoted by the associated scattering angles θ and source indices i in the form θ^i . Most optimisation steps remove the maximum number of nodes (3) in fewer than 5 epochs.

Table 4-7 – Training and optimisation results (first MFANN)

Optimisation Step	Training Epochs	Removed Nodes
1 (training only)	4	—
2	0	$45^R, 50^{IR}, 55^G$
3	0	$35^R, 45^{IR}, 60^{IR}$
4	0	$25^G, 30^G, 65^R$
5	0	$20^{IR}, 40^{IR}, 55^{IR}$
6	0	$30^{IR}, 55^R, 65^{IR}$
7	1	$25^R, 60^R, 70^R$
8	1	$20^G, 35^G, 40^G$
9	1	$30^R, 40^R, 50^R$
10	3	$25^{IR}, 35^{IR}$
11	2	50^G

Figure 4-47 and figure 4-48 demonstrate the *Root Mean Squared (RMS)* and *absolute maximum* errors of the total mass concentration approximations returned by the first MFANN structure after each optimisation step. Both errors are defined by expression (4-18) for $r = 2$ (RMS) and $r = +\infty$ (absolute maximum), and they are calculated independently for the l_{th} output node ($l = 1$ in this case) and pattern set. The RMS errors of the training ($Q = 14$, solid line, plus marks) and testing ($Q = 14$, dashed line, star marks) pattern sets converge to 10% and 10.7% at the final optimisation steps. The corresponding absolute maximum errors are 17.9% and 22.3%. All error values are high because the number of training patterns is small. Nevertheless, error curves of both pattern sets and either error type converge to each other as network optimisation proceeds. Also, the differences between similar error figures are relatively small (0.7% for RMS and 4.4% for absolute maximum error). As a matter of fact, the final version of the first MFANN may be considered as a universal model of the functional mapping that returns C_{mj} from $\check{S}_y(\theta, C_{mj})$ samples.

An initial 30-1-1 MFANN is built to solve the second part of the particle characterisation problem concerned. Every input node corresponds to a light source index i and a statistically significant scattering angle θ . The hidden node is assigned the log-sigmoid transfer function while the output node is designed to have a linear transfer function. The output node returns a uniformly normalised value that indicates the type of filtered AF sand being dispersed in any experimental sample (negative for AF_F and nonnegative for AF_C).

The input and output vector sets required to train, test and optimise the second MFANN include 28 elements each. The set of 30x1 raw input vectors $\{p_1, p_2, \dots, p_q, \dots, p_{28}\}$ is made from the columns of 30x28 matrix P_2 given by:

$$P_2 = \frac{P}{\mathbf{1} \cdot [\hat{C}_{mF}^T \mid \hat{C}_{mC}^T]} \tag{4-44}$$

where $\mathbf{1}$ is a 30x1 vector having all its elements equal to one, P is the 30x28 matrix returned by equation (4-38), the matrix ratio denotes element-by-element matrix division and $\hat{C}_{mF}, \hat{C}_{mC}$ are two 14x1 vectors defined by:

$$\hat{C}_{mj} \equiv [{}_{14j-13}o_1 \quad {}_{14j-12}o_1 \quad \dots \quad {}_q o_1 \quad \dots \quad {}_{14j}o_1]^T \tag{4-45}$$

($F \leftrightarrow 1$ and $C \leftrightarrow 2$) with elements ${}_q o_1$ derived by:

$${}_q o_1 = \max t_1({}_q \tilde{o}_1 + 0.5) - \min t_1({}_q \tilde{o}_1 - 0.5) \tag{4-46}$$

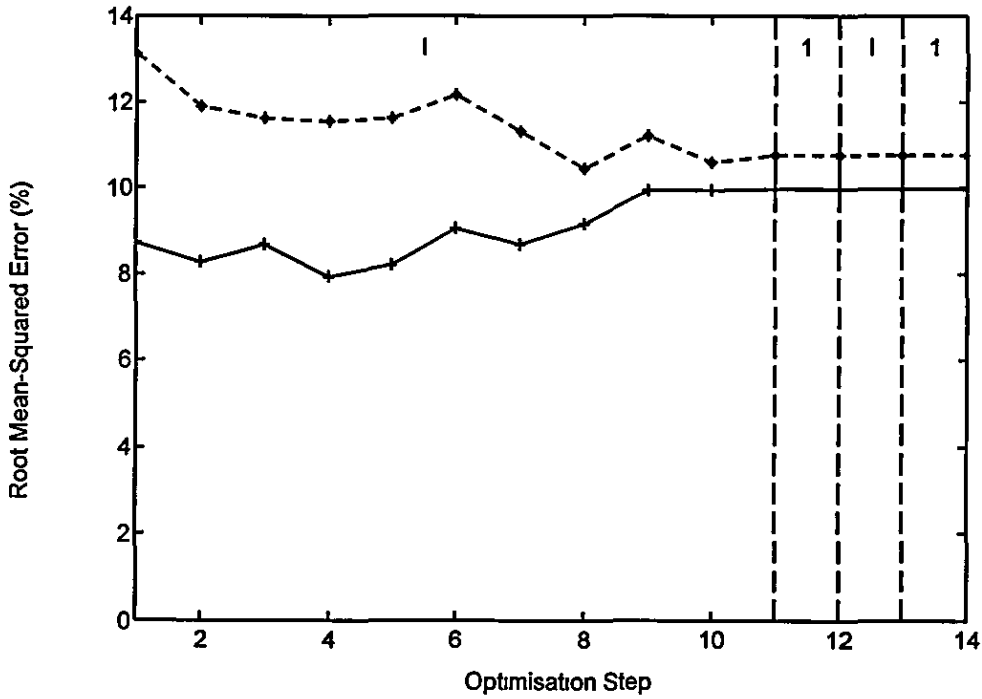


Figure 4-47 – RMS error change per optimisation step (first MFANN)

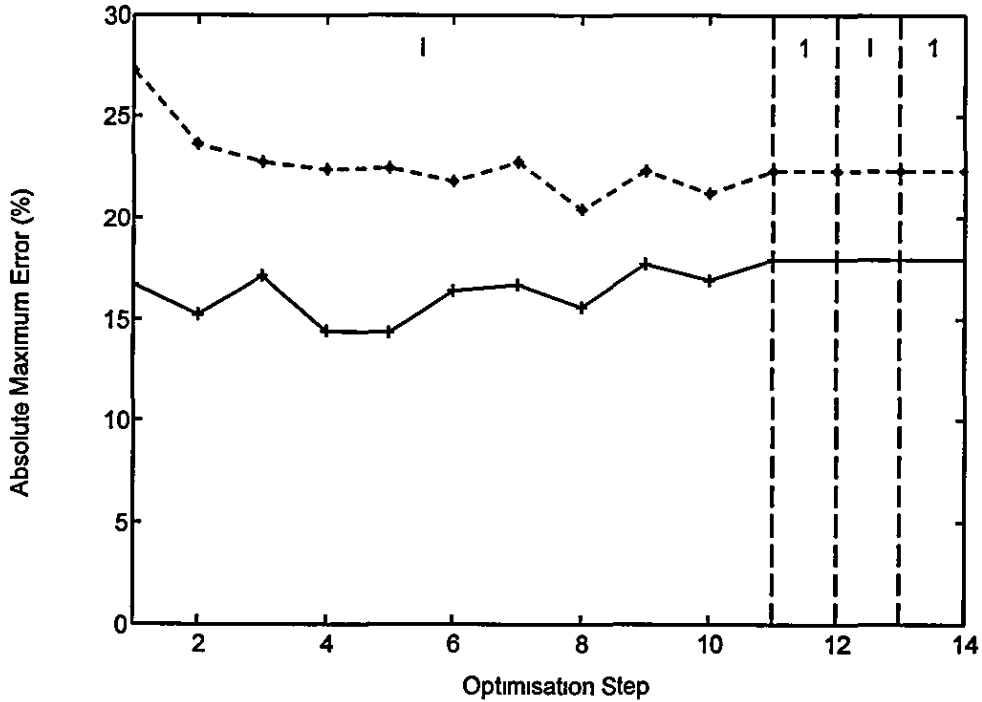


Figure 4-48 – Absolute maximum error change per optimisation step (first MFANN)

where ${}_q\tilde{o}_1$ is the q th uniformly normalised output of the first MFANN ($1 \leq q \leq 28$) and ${}_{\min} t_1, {}_{\max} t_1$ are the minimum and maximum values of row vector T set by (4-43). The set of 1×1 raw target vectors $\{t_1, t_2, \dots, t_q, \dots, t_{28}\}$ is consisted of the elements of 1×28 row vector T_2 given by:

$$T_2 = [0 \mid 1] \quad (4-47)$$

with 0 and 1 being two 1×14 row vectors of elements equal to zero and one respectively. The existence of fine AF sand in the experimental sample should minimise the output value of the second MFANN ($t_q < 0.5$), while the presence of coarse AF sand in the suspension is expected to maximise the output value concerned ($t_q \geq 0.5$).

The raw input and target vector sets are uniformly normalised to the interval $[-0.5, 0.5]$ by applying the appropriate pre-processing algorithm of §2.8. The 56 normalised vectors \tilde{p}_q, \tilde{t}_q are grouped in 28 pairs of the form $\{\tilde{p}_q, \tilde{t}_q\}$. Fourteen vector pairs are randomly selected for network training use (*training pattern set*) and the remaining 14 pairs are destined to measure the degree of generalisation exhibited by the trained network at the end of every ANN optimisation step (*testing pattern set*).

The complete MFANN optimisation scheme described in §2.9.4 is employed in the processing of the second network structure. The ANN training task is performed with the aid of the LMBP algorithm and the network optimisation goal is achieved via the modified SOSA and OBS algorithms. The performance index of the training algorithm is the MSE function defined by (4-10). The LMBP parameters are: $MSE_{goal} = 10^{-2}$, $Ep_{max} = 100$, $\nabla_{min} = 10^{-6}$, $\mu_0 = 10^{-3}$, $\mu_{max} = 10^{10}$, and $\theta = 10$. The optimisation parameters are: $c_1 = +\infty$, $c_2 = 2$, $c_3 = 3$, $\alpha = 10^{-6}$ and $N_{max}^+ = 1$. The final network is 6-2-1 with 8 weights and 3 biases.

Table 4-8 lists the number of training epochs performed and the types and positions of ANN elements removed per optimisation step. Input nodes are denoted by the associated scattering angles θ and source indices i in the form θ^i . Hidden and output nodes are represented by superscripts h or o placed next to their ordinal numbers (e.g. $1^h =$ first hidden node, $1^o =$ first output node). Most optimisation steps remove the maximum number of nodes (3) in fewer than 5 epochs.

The performance of the second MFANN is measured by the values of the relative classification errors RCE_j obtained by definitions (4-41) and (4-42). Figure 4-49 illustrates how RCE_F and RCE_C change per network optimisation step for every indi-

vidual pattern set. The applied optimisation scheme always manages to adapt perfectly the network model to the training pattern set ($RCE_j = 0$ for that set). As far as the testing pattern set is concerned, both RCE_F and RCE_C converge to 14.3% at the fourteenth optimisation step, i.e. when the first hidden layer optimisation procedure terminates. The positive values of RCE_j imply that more scattering measurements of filtered AF sand suspensions are required for further training of the second MFANN (one out of seven suspensions is misclassified). Nevertheless, the error figures obtained are moderate and perfectly balanced between the two filtered AF sand types. This is one reason why the cascaded MFANN model developed in this section is better than the single MFANN model described in §4.3.6.1.

The angular positions of the three nephelometric light sources and detectors displayed in figure 4-50 and supplying the final set of MFANNs with scattering data are obtained from the (θ, i) associations of the input nodes of the final network model. Green LED is placed at 160° , red LED at 180° , IR LED at 135° , and the detectors at 20° , 25° and 45° with respect to the direction of propagation of light emitted by the red light source. This structurally optimal nephelometric sensor configuration requires a detector less than the corresponding sensor configuration of §4.3.6.1 (see figure 4-46). This is another reason why the cascaded MFANN model is favourable.

Table 4-8 – Training and optimisation results (second MFANN)

Optimisation Step	Epochs	Removed Nodes	Optimisation Step	Epochs	Removed Nodes
1 (training)	7	—	13 (training)	0	—
2	0	$20^{IR}, 25^{IR}, 40^{IR}$	14	43	1^h
3	1	$25^G, 30^R, 55^R$	15 (training)	0	—
4	0	$35^G, 50^G, 55^G$	16 (2 backtr.)	12	added 1^h
5	1	$35^R, 50^{IR}, 60^{IR}$	17 (training)	0	—
6	1	$25^R, 65^R, 70^R$	18 (OBS)	—	weight $45^G - 2^h$
7	3	$20^G, 35^{IR}, 50^R$	19 (OBS)	—	weight $65^{IR} - 1^h$
8	2	$30^{IR}, 40^R, 55^{IR}$	20 (OBS)	—	weight $45^R - 1^h$
9	1	45^{IR}	21 (OBS)	—	weight $40^G - 2^h$
10	11	30^G	22 (OBS)	—	weight $70^{IR} - 1^h$
11	5	60^R	23 (OBS)	—	weight $20^R - 1^h$
12 (2 backtr.)	24	added 2^h	—	—	—

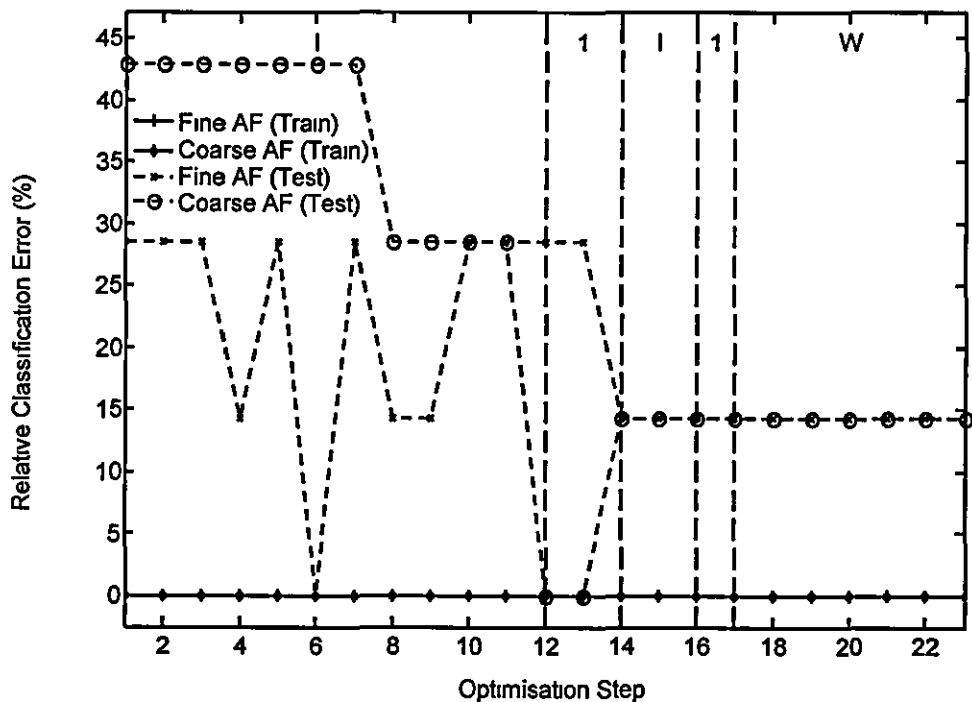


Figure 4-49 – Relative classification error change per optimisation step (second MFANN)

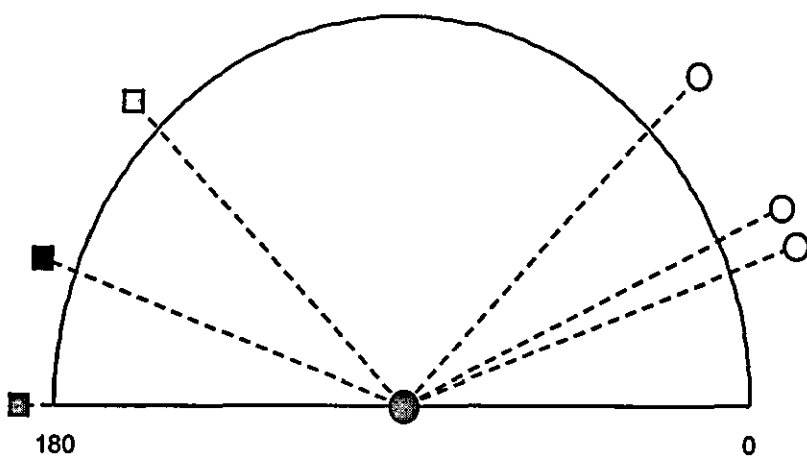


Figure 4-50 – Structurally optimal nephelometric sensor configuration

When a sample suspension is placed in the nephelometer, the three light sources start flashing alternatively. The signal returned by the detector at 20° is time-division multiplexed, while the signals returned by the remaining detectors are accepted only at times the associated light sources are on ($25^\circ \leftrightarrow$ green and IR LEDs, $45^\circ \leftrightarrow$ red LED). The derived signals are rectified and sampled in order to obtain the numerical values $x_1 = 20^G$, $x_2 = 25^G$, $x_3 = 20^R$, $x_4 = 45^R$, $x_5 = 20^{IR}$ and $x_6 = 25^{IR}$ that are necessary to create the raw input vectors of the two MFANNs. The 3×1 raw input vectors $\mathbf{p}_1 = [x_2 \ 0 \ 0]^T$, $\mathbf{p}_2 = [0 \ x_3 \ 0]^T$ and $\mathbf{p}_3 = [0 \ 0 \ x_6]^T$ are uniformly normalised and applied to the first network. The uniformly normalised outputs ${}_q\tilde{o}_1$ ($1 \leq q \leq 3$) of the first network are applied to equation (4-46) to obtain the associated raw output data ${}_qo_1$. The 6×1 raw input vectors $\mathbf{p}_1 = {}_1o_1^{-1} [x_1 \ x_2 \ 0 \ 0 \ 0 \ 0]^T$, $\mathbf{p}_2 = {}_2o_1^{-1} [0 \ 0 \ x_3 \ x_4 \ 0 \ 0]^T$ and $\mathbf{p}_3 = {}_3o_1^{-1} [0 \ 0 \ 0 \ 0 \ x_5 \ x_6]^T$ are uniformly normalised and applied to the second network. If most of the uniformly normalised output values returned by the second network are negative, the cascaded MFANN model recognises the filtered AF sand that is dispersed in the experimental sample as AF_F ; otherwise the dispersed sand is detected as AF_C .

4.4 Summary

Two typical particle characterisation problems were solved in this chapter by modelling methodologies that make full use of the ANN modelling and particle scattering theories developed in chapter 2 and chapter 3. The first problem considered 231 tenuous water suspensions of equiponderant mixtures of binned AF sand and dealt with the estimation of the distribution of total suspended volume per sand type by theoretical scattering data. The solution obtained is consisted of a plain polar nephelometer with three optical elements and a small MFANN structure. The second problem considered 28 tenuous water suspensions of AF_F and AF_C sand types and dealt with the classification of these suspensions per dispersed sand type by experimental scattering data. The instruments used and procedure followed to gather the experimental data of the problem were adequately explained. The similarities and differences between these data and the associated theoretical data obtained from the application of the polar nephelometric model of §3.5 are revealed and discussed. The solutions to the classification problem are two and require similar resources. The first solution is consisted of a polar nephelometer with seven optical elements and a small

MFANN, while the second solution includes a polar nephelometer with six optical elements and two small network structures connected in cascade. The second solution was preferred because it requires fewer optical elements and gives slightly better results.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

♪ If I could talk the mystery of tomorrow... ♪

(Sylver, Chances, 2001)

5.1 Theoretical conclusions

Particle characterisation is a technique that - unlike surrogate measurement - is able to provide low-resolution particle size information in real time. The wide applicability of the technique is the motivation for the development of methods that model and solve related problems. Scattering nephelometry is a method that promises simple, robust and inexpensive solutions under all circumstances. Two major aspects of the method, *scattering data analysis* and *selection of nephelometric calibration material*, are the subjects of research for this study.

Scattering nephelometry considers amplitude measurements of the scattered electromagnetic field at a limited number of positions around the scattering particles. However, the necessary condition for a set of particles to be uniquely identifiable is the precise knowledge of the incident, internal and scattered electromagnetic fields at all points in space. Therefore a unique analytical system that converts nephelometric data to particle descriptions does not exist. The existence of customised analytical systems for all possible particle arrangements cannot be guaranteed either. Therefore mathematical analysis is not suitable for nephelometric data conversion.

An alternative approach to the scattering data analysis problem is the employment of a data fitting method. These methods build mathematical models that approximate an unknown continuous function $f(\mathbf{x})$ to another function $F(\mathbf{a}, \mathbf{x})$ by a representative set of data pairs $(\mathbf{x}_i, f(\mathbf{x}_i))$ so that $|f(\mathbf{x}) - F(\mathbf{a}, \mathbf{x})| < \varepsilon \quad \forall \mathbf{x} \in \mathbb{R}^n$. The set of data pairs is usually obtained by experiment while the value of the parameter vector $\mathbf{a} \in \mathbb{R}^m$ is determined by the fitting method. However, the formation of the parametric function $F(\cdot)$ usually requires some knowledge about the form (polynomial, rational, exponential, logarithmic etc) of the unknown function $f(\cdot)$. That is the reason why most data fitting methods are inapplicable to nephelometric data conversion problems.

Data fitting methods employing Artificial Neural Networks (ANNs) are specialised in retuning universal approximations of completely unknown functions from representative sets of data mapping examples. Furthermore, the methods can be easily adapted to solve classification problems by sets of typical data associations. Therefore ANN fitting is a viable approach to the scattering data analysis issue. Multilayer Feed-forward Artificial Neural Network (MFANN) modelling solutions are particularly attractive in this case thanks to their architectural simplicity and operational efficiency.

The supervised learning algorithm that performs the MFANN training task in every network modelling case of this study is the Levenberg-Marquardt backpropagation (LMBP). The choice is based on the computational efficiency of the algorithm and the moderate sizes of the models. The algorithm is complemented by the mean-squared error (MSE) or sum-squared error (SSE) performance index. The MFANN input and output vector pre-processing by the modified uniform normalisation method protects network nodes from saturation whenever necessary.

The novel Second-Order Sensitivity Analysis (SOSA) and Optimal Brain Surgeon (OBS) algorithms remove all redundant MFANN nodes and weights from the trained network. The innovative SOSA algorithm removes up to three nodes per iteration subject to the average output error of the trimmed network being sufficiently small. Furthermore, SOSA cancels automatically any node pruning step that leads to a trimmed and probably retrained network of output error higher than a goal figure set a priori. OBS completes the MFANN optimisation task of SOSA by repeatedly removing weights and biases from the remaining MFANN structure until the predetermined network output error goal is no longer achieved. OBS is quite fast because it does not require network retraining between iterations.

An innovative MFANN optimisation scheme that employs LMBP, SOSA and OBS algorithms in the training and optimisation of any MFANN structure is another major contribution of this study to academic research. The scheme starts from the node optimisation of the input layer, continues with the node optimisation of hidden layers in ascending layer order, goes on with the weight and bias optimisation of the whole network and returns to the input layer for another optimisation cycle. Provision is made for the addition of extra nodes to all hidden layers that follow the layer undertaking node optimisation when better node elimination results can be achieved. The optimisation scheme halts after a complete optimisation cycle when no more input nodes are pruned and equal or more network nodes are added than removed in total.

MFANN modelling leads to excellent particle characterisation results only if the experimental scattering data used for the tasks of network training and optimisation are precise. Therefore it is essential to calibrate the nephelometric instrument with a material representative of the unknown particles that need to be characterised. The ISO 12103-1 Arizona Fine (AF) sand is better than formazine as calibration material because it has similar texture and particle size range to other substances existing in many natural and industrial suspensions. Tenuous water suspensions of AF sand in

polar nephelometers give profiles of scattered irradiances proportional to the mass concentration of the sand and inversely proportional to the nephelometric chamber radius. The same profiles are amplified and smoothed as the acceptance angle of the nephelometric detector increases. The turbidity curves extracted from these profiles show turbidity being proportional to mass concentration and acceptance angle and inversely proportional to nephelometric chamber radius. The AF particle size pdf dithering effect described in the ISO 12103-1 standard results to the similar dithering of all scattering and turbidity curves mentioned above by the relative magnitude of 1% in dB units.

The estimation of the binned particle volume distribution of a mixture of three binned AF sand types (denoted by AF_i for $i = 1, 2$ or 3) is the subject of the theoretical particle characterisation problem considered in this study. Only tenuous suspensions of the mixture are assumed because their scattering profiles can be estimated by the single scattering theory and the polar nephelometric model described in §3.5. The optimal solution obtained is a polar nephelometer of two optical elements (two light sources, one light detector) and a 2-2-3 MFANN structure of 9 weights and 3 biases. The first component of the solution is similar to a standard turbidimeter in terms of complexity.

5.2 Experimental conclusions

The classification of water suspensions of filtered AF sand in two groups with respect to the type of dispersed substance (AF_F or AF_C) is the motivation to conduct a number of scattering trials and describe them in this study. The experimental data are acquired and recorded via a commercial nephelometric system consisted of a polar nephelometer, a control unit, a power unit, a lock-in amplifier and a PC. The acquired voltages V_{ij} are converted to normalised particle scattering values \check{S}_{ij} which are linearly proportional to the mass concentration C_{mj} of suspended matter. The relation between \check{S}_{ij} and C_{mj} as well as the moderate mass concentration values in all experimental samples indicate that the scattering experiment may be simulated by an analytical model based on single scattering theory and nephelometric geometry. In fact, the correlation coefficient ρ_0 of the theoretical and experimental normalised particle scattering per unit mass concentration functions, \check{S}_{ij} and \bar{S}_{ij} is high (>93%). However, the Relative Mean-Squared Error (RMSE) of these functions is high (>64%) for samples

containing dispersed AF_C sand. RMSE and ρ_0 figures are significantly improved by assuming that particle aggregation and particle sedimentation take place at sample preparation time so that filtered AF particles smaller than 1 μm and bigger than 5 μm do not scatter light during the experiment. The comparison of the new normalised particle scattering per unit mass concentration function, \bar{S}_y^s with \bar{S}_y gives higher correlation coefficient and smaller RMSE values in all cases ($\rho_0 \geq 97\%$, RMSE < 15%) and thus it justifies the aforementioned non-scattering particle assumption. The remaining differences between the two functions can be attributed to measurement errors due to the presence of noise.

The experimental particle characterisation problem has two structurally optimal solutions. The first solution is a polar nephelometer of 7 optical elements (3 light sources, 4 light detectors) and a complete 6-1-1 MFANN structure of 7 weights and 2 biases. The model makes no mistake in detecting the presence of AF_F sand in the samples, but it mistakenly classifies 2 out of 7 test samples containing AF_C sand. The second solution is a polar nephelometer of 6 optical elements (3 light sources, 3 light detectors) and two MFANN structures connected in cascade. The first 3-1-1 network with 4 weights and 2 biases estimates the mass concentration of filtered AF sand, while the second 6-2-1 network with 8 weights and 3 biases detects the type of suspended matter. The second model is better than the first because it requires fewer optical elements and classifies correctly 6 out of 7 test sample suspensions of either AF sand type.

5.3 Recommended extensions

Plenty of ideas exist to improve or extend the novel Second-Order Sensitivity Analysis (SOSA) algorithm explained in §2.9.2 and the innovative MFANN optimisation scheme described in §2.9.4. For example:

- a) The efficiency of SOSA algorithm is measured by the average number of network nodes removed per iteration and the number of backtracked iterations. Both figures depend on the values of algorithmic constants c_1 , c_2 and c_3 . If the constants are set too high, many nodes are removed in the first iterations but many pruning attempts are cancelled as the network reaches its optimal size. If the constants are set too low, the frequency of backtracking is minimised but more iterations are required to remove all redundant nodes from the network.

A compromise would be to choose constants c_2 and c_3 being high in the beginning but decaying smoothly as the algorithm progresses. As far as constant c_1 is concerned, hypothesis testing can lead to accurate assignments that minimise the possibility of backtracking in the final SOSA iterations (see [158-161]).

- b) Node counter N_{\max}^+ determines how many nodes can be added to a network before one or more nodes are removed from the layer SOSA optimises. If the counter is set to zero or a small value, the network may fail to achieve the pre-determined error goal during the training phase because it has fewer hidden nodes than necessary. This results to either rejection of the initial network or premature halt of SOSA algorithm. On the other hand, setting N_{\max}^+ too high prolongs the optimisation phase and leads to suboptimal and overtrained networks due to the addition of too many hidden nodes. The introduction of empirical rules for the optimal value of N_{\max}^+ would reduce the MFANN modelling effort considerably.
- c) The addition of nodes by SOSA is followed by the resetting of all network weights and biases to random values. This strategy is simple to implement but prolongs the training of the trimmed network. Weight and bias initialisation rules taken from constructional optimisation algorithms (see [134-136]) and modified appropriately for use by SOSA algorithm would reduce the MFANN training time significantly.
- d) The selection of network training parameters is often critical for the optimal training of the network considered. The SOSA algorithm and the MFANN optimisation scheme do not alter these parameters between subsequent training phases. However, the alternative of modifying the training parameters as the optimisation scheme progresses is worth examining because it can lead to better optimisation results.

A significant contribution to the design of better nephelometric models that characterise solid particles dispersed in fluids at small or no error is the development of experimental procedures that minimise the scattering acquisition time and maximise the number of solid particles kept in suspension throughout the experiment. More and reliable scattering measurements give more ANN training and testing patterns which in turn can lead to nephelometric models that give more precise particle

characterisation results. Scattering data collected from field trials could also be helpful in the design of structurally optimal nephelometric models that meet the specifications of practical particle characterisation applications.

Ultrasound scattering methods that deal with the characterisation of bigger particles than the ones considered by light scattering methods could be benefited from the MFANN modelling and optimisation techniques discussed earlier in this study. Statistical optics may replace Mie scattering theory in nephelometric model design. When the ultrasound nephelometric model is complete, a MFANN structure attached to the nephelometric sensors can be trained to perform the particle characterisation task and then optimised with respect to its number of input nodes (nephelometric sensors), hidden nodes and weights.

Finally, the study of the scattering properties of AF sand in dense suspensions as a straightforward extension of the material covered in §3.6 is of particular interest. The first-order multiple scattering (FOMS) theory [192] accepts that light incident to a particle may have been scattered by another particle instead of coming directly from the light source. However, only particles that do not send scattered light to other particles are taken into account for the calculation of scattered irradiance at a certain angle. Foldy-Twersky scattering theory [193, 194] goes even further by accepting that scattered irradiance measured at an angle may be coming from particles that illuminate other particles. Both theories are applicable to dense suspensions of AF and new scattering profiles can be obtained by fitting the polar nephelometric model of §3.5 to these theories. Theoretical particle characterisation problems may be solved by assuming dense suspensions of AF sand and using the scattering data returned by the modified polar nephelometric model to train and optimise MFANN structures. Experimental scattering profiles from dense suspensions of AF sand may be acquired in order to verify the multiple scattering theories and solve practical particle characterisation problems. Those exercises would demonstrate the appropriateness of scattering nephelometry and ANN theory to deal with virtually every particle characterisation case.

APPENDIX A
REFERENCES

1. *Methods for determination of particle size distribution - Guide to microscope and image analysis methods*. 1993, The British Standards Institution.
2. Kay, D.H., *Techniques for electron microscopy*. Second ed. 1965, Oxford: Blackwell Scientific Publications.
3. Allen, T., *Powder sampling and particle size measurement*. Fifth ed. Particle size measurement. Vol. 1. 1999, Dordrecht: Kluwer Academic Publishers. 525.
4. Sarid, D., *Scanning force microscopy with applications to electric, magnetic and atomic forces*. 1992, New York: Oxford University Press.
5. *Standard specifications for perforated plate sieves for testing purposes*. 1980, The American Society for Testing Materials.
6. *Standard specification for wire-cloth sieves for testing purposes*. 1987, The American Society for Testing Materials.
7. *Standard specification for precision electroformed sieves (square aperture)*. 1987, The American Society for Testing Materials.
8. Blackford, D.B. and D.C. Grant, *Evaluating a novel liquid-borne particle counter based on interferometry (1)*. *Microcontamination*, 1993. 11(2): p. 27-32, 66-67.
9. Grant, D.C., *Evaluating a novel liquid-borne particle counter based on interferometry: experimental testing*. *Microcontamination*, 1993. 11(3): p. 37-43.
10. Dahneke, B.E., *Aerosol beam spectrometry*. *Nature (Physical Science)*, 1973. 244(134): p. 54-55.
11. Pendse, H.P. and A. Sharma, *Particle size distribution analysis of industrial colloidal slurries using ultrasonic spectroscopy*. *Particle & Particle Systems Characterisation: Measurement and Description of Particle Properties and Behaviour in Powders and other Disperse Systems*, 1993. 10(5): p. 229-233.
12. Holland-Batt, A.B., *A quantitative model of the motion of particles in the RSM/Mintek on-stream particle size analyser*. *Powder Technology*, 1975. 11(1): p. 11-25.
13. Nakajima, Y., et al., *Void fraction measurement of homogenous powder beds by frequency response method*. *Industrial & Engineering Chemistry Fundamentals*, 1970. 9(3): p. 489-495.
14. Nakajima, Y., K. Gotah, and T. Tanaka, *Design of the on-line particle size detector for fine powders*. *Industrial & Engineering Chemistry Fundamentals*, 1971. 10(2): p. 318-320.

15. Witt, W. and S. Rothele. *Laser diffraction - Unlimited?* in *Sixth European Symposium on Particle Size Characterisation*. 1995. Nuremberg: Nurnberg-Messe GmbH.
16. Boxman, A., et al., *Deconvolution of light-scattering patterns by observing intensity fluctuations*. *Applied Optics*, 1991. 30(33): p. 4818-4823.
17. Weiner, B.B. *27 years of QELS - A review of the advantages and disadvantages of particle sizing with QELS*. in *Particle Size Analysis*. 1991. Loughborough: The Royal Society of Chemistry.
18. Finsy, R., *Particle sizing by quasi-elastic light scattering*. *Colloid and Interface Science*, 1994. 52: p. 79-143.
19. Trainer, M.N., P.J. Freud, and E.L. Weiss. *Analytical Applied Spectroscopy*. in *Pittsburgh*. 1990. Symposium on Particle Size Analysis.
20. Heuer, M. and K. Leschonski, *Results obtained with a new instrument for the measurement of particle size distributions from diffraction patterns*. *Particle Characterization*, 1985. 2(1): p. 7-13.
21. *Determination of turbidity*. 1984, The British Standards Institution.
22. *Formazine recipe*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_045a.htm.
23. *Calibration (turbidity)*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_054.htm.
24. *Lambert-Beer Law*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_061.htm.
25. Mie, G., *Beitrage zur optic truber medien speziell kolloidaler metallosungen*. *Ann. Phys.*, 1908. 25: p. 377-445.
26. *Turbidity value*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_120.htm.
27. Brandolin, A. and L.H. Garcia-Rubio. *On-line particle-size distribution measurements for latex reactors*. in *Particle Size Distribution II - Assessment and characterisation*. 1991. Washington: American Chemical Society.
28. *Potable water treatment*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_117.htm.
29. *Spring water*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_087.htm.

30. *Sand filtration*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_095.htm.
31. *Minreral water*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_071.htm.
32. *Power plants, thermal*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_060.htm.
33. *Flue gas monitoring*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_088.htm.
34. *Sulfuric acid mist*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_098.htm.
35. Duntley, S.Q., *Light in the sea*. Optical Society of America, 1963. 53: p. 214-233.
36. *OBS-3 turbidity sensor*. 2000, Geneq Inc.
<http://www.geneq.com/catalog/en/obs.htm>.
37. *Seapoint turbidity meter*. 2001, Seapoint Sensors Inc.
<http://www.seapoint.com/stm.htm>.
38. Gippel, C.J., *Potential of turbidity monitoring for measuring the transport of suspended solids in streams* Hydrological Processes, 1995. 9(1): p. 83-97.
39. *Filtration (alcohol)*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_039.htm.
40. *Brewery*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_013.htm.
41. *Wort lautering*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_127.htm.
42. *Yeast addition*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_050.htm.
43. *Beer filtration*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_014.htm.
44. *Oil mist*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_078.htm.
45. *Sugar production*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_128.htm.
46. *Road tunnels*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_108.htm.

47. *Visibility determination*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_100.htm.
48. *Formazine*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_045.htm.
49. Rogner, A., *Turbidity measurement in drinking water applications - New requirements and approaches*, in *International Environmental Technology*. 1996. p. 9-10.
50. Christen, R. and A. Rogner. *Continuous monitoring of low particulate matter concentrations using scattered light*. in *Continuous Emissions Monitoring*. 1998. London.
51. Hongve, D. and G. Akesson, *Comparison of nephelometric turbidity measurements using wavelengths 400-600 and 860 nm*. *Water Research*, 1998. 32(10): p. 3143-3145.
52. *Inherent brightening effect*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_048.htm.
53. *Overflow cell*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_123.htm.
54. *Dust measurement*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_105.htm.
55. *Guidelines for drinking-water quality: Surveillance and control of community supplies*. 2 ed. Vol. 3. 1997, Geneva: World Health Organisation.
56. Hecht, E., *Optics*. Third ed. 1998, Reading: Addison-Wesley. 694.
57. *Colour standard*. 2001, Sigrist Photometer A. G.
http://www.photometer.com/en/abc/abc_035.htm.
58. Knight, A.G., *The photometric estimation of colour in turbid waters* Institute of Water Engineers, 1951. 5: p. 623-633.
59. Kirk, J.T.O., *Light and photosynthesis in aquatic ecosystems*. 1983, Cambridge: Cambridge University Press.
60. Ghassemi, M. and R.F. Christman, *Properties of the yellow organic acids of natural waters*. *Limnology and Oceanography*, 1968. 13: p. 583-597.
61. *Guidelines for drinking water quality: Recommendations*. 2 ed. Vol. 1. 1997, Geneva: World Health Organisation.
62. *ISO 12103-1: Road vehicles - Test dust for filter evaluation*. 1997, The International Organisation for Standardisation: Geneva.

63. Anderson, J.A. and E. Rosenfeld, eds. *Neurocomputing: Foundations of research*. 1989, MIT Press: Cambridge, MA.
64. James, W., *Psychology (briefer course)*. 1890, New York: Holt.
65. McCulloch, W.S. and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*. Bulletin of Mathematical Biophysics, 1943. 5: p. 115-133.
66. Hebb, D.O., *The organization of behavior*. 1949, New York: Wiley.
67. Rosenblatt, F., *The perceptron. a probabilistic model for information storage and organization in the brain*, in *Psychological Review*. 1958. p. 386-408.
68. Widrow, B. and M.E. Hoff, *Adaptive switching circuits*, in *WESCON Convention Record*. 1960, IRE: New York.
69. Minsky, M. and S. Papert, *Perceptrons*. 1969, Cambridge, MA: MIT Press.
70. Kohonen, T., *Correlation matrix memories*. IEEE Transactions on Computers, 1972. 21: p. 353-359.
71. Anderson, J.A., *A simple neural network generating an interactive memory* Mathematical Biosciences, 1972. 14: p. 197-220.
72. Hopfield, J.J., *Neural networks and physical systems with emergent collective computational properties*. Proceedings of the National Academy of Sciences, 1982. 79(4): p. 2554-2558.
73. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning internal representations by error propagation*, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D.E. Rumelhart and J.L. McClelland, Editors. 1986, MIT Press: Cambridge, MA. p. 318-362.
74. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors*. Nature, 1986. 323: p. 533-536.
75. Anderson, J.A., A. Pellionisz, and E. Rosenfeld, eds. *Neurocomputing 2: Directions for research*. 1990, MIT Press: Cambridge, MA.
76. Ackley, D.H., G.E. Hinton, and T.J. Sejnowski, *A learning algorithm for Boltzmann machines*. Cognitive Science, 1985. 9(1): p. 147-169.
77. Barto, A., R. Sutton, and C. Anderson, *Neuron-like adaptive elements can solve difficult learning control problems*. IEEE Transactions on Systems, Man and Cybernetics, 1983. 13(5): p. 834-846.
78. Lippmann, R.P., B. Gold, and M.L. Malpass, *A comparison of Hamming and Hopfield neural nets for pattern classification*. 1987, MIT Lincoln Laboratory.

79. Lippmann, R.P., *An introduction to computing with neural nets*, in *IEEE Acoustics, Speech and Signal Processing*. 1987. p. 4-22.
80. Hecht-Nielsen, R., *Counterpropagation networks*. *Applied Optics*, 1987. 26(12): p. 4979-4984.
81. Hecht-Nielsen, R., *Applications of counterpropagation networks*. *Neural Networks*, 1988. 1(2): p. 131-139.
82. Fukushima, K., S. Miyake, and T. Ito, *Neocognitron: A neural network model for a mechanism of visual pattern recognition* *IEEE Transactions on Systems, Man and Cybernetics*, 1983. 13(5): p. 826-834.
83. Fukushima, K., *Neocognitron: A hierarchical neural network capable of visual pattern recognition* *Neural Networks*, 1988. 1: p. 119-130.
84. Carpenter, G.A. and S. Grossberg, *ART2: Self-organization of stable category recognition codes for analog input patterns*. *Applied Optics*, 1987. 26(23): p. 4919-4930.
85. Carpenter, G.A. and S. Grossberg, *ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures*. *Neural Networks*, 1990. 3(23): p. 129-152.
86. Carpenter, G.A., S. Grossberg, and D.B. Rosen, *Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system* *Neural Networks*, 1991. 4: p. 759-771.
87. Carpenter, G.A., S. Grossberg, and J. Reynolds, *ARTMAP. Supervised real-time learning and classification of nonstationary data by a self-organising neural network*. *Neural Networks*, 1991. 4: p. 565-588.
88. Carpenter, G.A., *et al.*, *Fuzzy ARTMAP: A neural network architecture for incremental learning of analog multidimensional maps*. *IEEE Transactions on Neural Networks*, 1992. 3: p. 698-713.
89. Carpenter, G.A. and W.D. Ross, *ART-EMAP: A neural network architecture for object recognition by evidence accumulation*. *IEEE Transactions on Neural Networks*, 1995. 6(4): p. 805-818.
90. Hopfield, J.J., *Neurons with graded response have collective computational properties like those of two-state neurons*. *Proceedings of the National Academy of Sciences*, 1984. 81(5): p. 3088-3092.

91. Li, J., A.N. Michel, and W. Porod, *Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube*. IEEE Transactions on Circuits and Systems, 1989. 36(11): p. 1405-1422.
92. Michel, A.N. and J.A. Farrell, *Associative memories via artificial neural networks*, in *IEEE Control Systems Magazine*. 1990. p. 6-17.
93. Geman, S. and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984. 6: p. 721-741.
94. Kosko, B., *Adaptive bidirectional associative memories* Applied Optics, 1987. 26: p. 4910-4918.
95. Kosko, B., *Bidirectional associative memories*. IEEE Transactions on Systems, Man and Cybernetics, 1988. 18(1): p. 49-60.
96. Anderson, J.A., et al , *Distinctive features, categorical perception, and probability learning: some applications of a neural model*. Psychological Review, 1977. 84: p. 413-451.
97. Lang, K.J. and G.E. Hinton, *The development of the time-delay neural network architecture for speech recognition*. 1988, Carnegie-Mellon University: Pittsburgh.
98. Waibel, A., et al., *Phoneme recognition using time-delay neural networks* IEEE Transactions on Acoustics, Speech and Signal Processing, 1989. 37: p. 328-339.
99. Wan, E.A. *Temporal backpropagation for FIR neural networks*. in *IEEE International Joint Conference on Neural Networks*. 1990. San Diego.
100. Wan, E.A. *Temporal backpropagation: An efficient algorithm for finite impulse response neural networks*. in *Proceedings of the 1990 Connectionist Models Summer School*. 1990. San Mateo: Morgan Kaufmann.
101. Wan, E.A., *Time series prediction by using a connectionist network with internal delay lines*, in *Time Series Prediction: Forecasting the Future and Understanding the Past*, A.S. Weigend and N.A. Gershenfeld, Editors. 1994, Addison Wesley: Reading. p. 195-217.
102. Haykin, S. and L. Li, *Nonlinear adaptive prediction of non-stationary signals* IEEE Transactions on Signal Processing, 1995. 43(2): p. 526-535.

103. Narendra, K.S. and K. Parthasarathy, *Identification and control of dynamical systems using neural networks* IEEE Transactions on Neural Networks, 1990. 1(1): p. 4-27.
104. Elman, J.L., *Finding structure in time*. Cognitive Science, 1990. 14: p. 179-211.
105. Williams, R.J. and D. Zipser, *A learning algorithm for continually running fully recurrent neural networks*. Neural Computation, 1989. 1: p. 270-280.
106. Hornik, K.M., M. Stinchcombe, and H. White, *Multilayer feedforward networks are universal approximations*. Neural Networks, 1989. 2(5): p. 359-366.
107. Kolmogorov, A.N., *On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition*. Doklady Akademii Nauk, 1957. 144: p. 670-681.
108. Albus, J.S., *A theory of cerebellar function* Mathematical Biosciences, 1971. 10: p. 25-61.
109. Albus, J.S., *A new approach to manipulator control: The cerebellar model articulation controller (CMAC)*. Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 1975. 97: p. 220-227.
110. Jacobs, R.A. and M.I. Jordan, *A competitive modular connectionist architecture*, in *Advances in Neural Information Processing Systems*, R.P. Lippman, J.E. Moody, and D.J. Touretzky, Editors. 1991, Morgan Kaufmann: San Mateo. p. 767-773.
111. Jacobs, R.A., *et al.*, *Adaptive mixtures of local experts*. Neural Computation, 1991. 3: p. 79-87.
112. Powell, M.J.D., *Radial basis functions for multivariable interpolation: A review*, in *Algorithms for the Approximation of Functions and Data*, J.C. Mason and M.G. Cox, Editors. 1987, Clarendon Press: Oxford. p. 143-167.
113. Broomhead, D.S. and D. Lowe, *Multivariable functional interpolation and adaptive networks*. Complex Systems, 1988. 2: p. 321-355.
114. Moody, J.E. and C.J. Darken, *Fast learning in networks of locally-tuned processing units*. Neural Computation, 1989. 1: p. 281-294.
115. Poggio, T. and F. Girosi, *Networks for approximation and learning*, in *Proceedings of the IEEE*. 1990. p. 1481-1497.
116. Pao, Y.-H., *Adaptive pattern recognition and neural networks*. 1989, Reading: Addison-Wesley.

117. Ivakhnenko, A.G., *Polynomial theory of complex systems*. IEEE Transactions on Systems, Man and Cybernetics, 1971. 12: p. 364-378.
118. Heywood, M. and P. Noakes, *A framework for improved training of sigma-pi networks*. IEEE Transactions on Neural Networks, 1995. 6(4): p. 893-903.
119. Rumelhart, D.E. and J.L. McClelland, *Parallel distributed processing: Explorations in the macrostructure of cognition*. Vol. 1. 1986, Cambridge: MIT Press.
120. White, D. and D. Sofge, *Handbook of intelligent control*, ed. D. White and D. Sofge. 1992, New York: Van Nostrand Reinhold.
121. Kohonen, T., *Self-organization and associative memory*. 2 ed. 1987, Berlin: Springer-Verlag.
122. Freeman, J. and D. Skapura, *Neural networks: Algorithms, applications, and programming techniques*. 1991, Reading: Addison-Wesley.
123. Rosenblatt, F., *Chapter 13*, in *Principles of Neurodynamics*. 1962, Spartan Books: New York.
124. Scales, L.E., *Introduction to non-linear optimization*. 1985, New York: Springer-Verlag.
125. Charalambous, C., *Conjugate gradient algorithm for efficient training of artificial neural networks*. IEE Proceedings, 1992. 139(3): p. 301-310.
126. Vogl, T.P., *et al.*, *Accelerating the convergence of the backpropagation method*. Biological Cybernetics, 1988. 59(9): p. 256-264.
127. Fahlman, S.E. *Faster-learning variations on back-propagation: An empirical study*. in *Proceedings of the 1988 Connectionist Models Summer School*. 1988. San Mateo: Morgan Kaufmann.
128. Jacobs, R.A., *Increased rates of convergence through learning rate adaptation* Neural Networks, 1988. 1(4): p. 295-308.
129. Tollenaere, T., *SuperSAB: Fast adaptive back propagation with good scaling properties*. Neural Networks, 1990. 3(5): p. 561-573.
130. Riedmiller, M. and H. Braun. *A direct adaptive method for faster backpropagation learning: The RPROP algorithm*. in *Proceedings of the IEEE International Conference on Neural Networks*. 1993. San Francisco: IEEE.
131. Hagan, M.T. and M. Menhaj, *Training feedforward networks with the Marquardt algorithm*. IEEE Transactions on Neural Networks, 1994. 5(6): p. 989-993.

132. Wang, J.H., T.F. Krile, and J.F. Walkup. *Reduction of interconnection weights in higher order associative memory networks*. in *International Joint Conference on Neural Networks*. 1991. Seattle.
133. Sankar, A. and R.J. Mammone. *Optimal pruning of neural tree networks for improved generalisation*. in *International Joint Conference on Neural Networks*. 1991. Seattle.
134. Fahlman, S.E. and C. Lebiere, *The cascade-correlation learning architecture*, in *Advances in Neural Information Processing Systems*, D.S. Touretzky, Editor. 1990, Morgan Kaufmann: San Mateo, California, USA. p. 524-532.
135. Wynne-Jones, M., *Node splitting: A constructive algorithm for feed-forward neural networks*, in *Advances in Neural Information Processing Systems*, J.E. Moody, S.J. Hanson, and R.P. Lippmann, Editors. 1992, Morgan Kaufmann: San Mateo, California, USA. p. 1072-1079.
136. Zhang, J. and A.J. Morris, *A sequential learning approach for single hidden layer neural networks*. *Neural Networks*, Pergamon Press, 1998. **11**(1): p. 65-80.
137. Reed, R., *Pruning algorithms - A survey*. *IEEE Transactions on Neural Networks*, 1993. **4**(5): p. 740-747.
138. Mozer, M.C. and P. Smolensky, *Skeletonisation: A technique for trimming the fat from a network via relevance assessment*, in *Advances in Neural Information Processing Systems*, D.S. Touretzky, Editor. 1989, Morgan Kaufman: San Mateo, California, USA. p. 107-115.
139. Hanson, S.J. and L.Y. Pratt, *Comparing biases for minimal network construction with back-propagation*, in *Advances in Neural Information Processing*, D.S. Touretzky, Editor. 1989, Morgan Kaufman: San Mateo, California, USA. p. 177-185.
140. Sietsma, J. and R.J.F. Dow. *Neural net pruning - Why and how*. in *IEEE International Conference on Neural Networks*. 1988. San Diego, California, USA.
141. Kruschke, J.K. *Creating local and distributed bottlenecks in hidden layers of back-propagation networks*. in *Connectionist Models Summer School*. 1988.
142. Kruschke, J.K. *Improving generalisation in back-propagation networks with distributed bottlenecks*. in *International Joint Conference on Neural Networks*. 1989. Washington.

143. Whitley, D. and C. Bogart, *The evolution of connectivity: Pruning neural networks using genetic algorithms*. in *International Joint Conference on Neural Networks*. 1990. Washington.
144. Masters, T., *Sensitivity analysis*, in *Practical Neural Network Recipes in C++*. 1993, Academic Press. p. 195-197.
145. Hertz, J., A. Krogh, and R.G. Palmer, *Introduction to the theory of neural computation*. 1991: Addison-Wesley.
146. Le Cun, Y., J.S. Denker, and S.A. Solla, *Optimal brain damage*, in *Advances in Neural Information Processing Systems*, D.S. Touretzky, Editor. 1990, Morgan Kaufman: San Mateo, California, USA. p. 598-605.
147. Hassibi, B. and D.G. Stork, *Second order derivatives for network pruning: Optimal brain surgeon*, in *Advances in Neural Information Processing Systems*, S.J. Hanson, J.D. Cowan, and L. Giles, Editors. 1993, Morgan Kaufmann: San Mateo, California, USA. p. 164-171.
148. Karnin, E.D., *A simple procedure for pruning back-propagation trained neural networks*. IEEE Transactions on Neural Networks, 1990. 1(2): p. 239-242.
149. Klimasauskas, C.C., *Neural nets tell why: A technique for explaining a neural network's decision-making process*, in *Dr. Dobb's Journal*. 1991. p. 16-24 & 78-84.
150. Zurada, J.M., *Multilayer perceptron networks: Selected aspects of training optimisation* First National Conference on Neural Networks, 1994. 1: p. 1-22.
151. Zurada, J.M., A. Malinowski, and I. Cloete, *Sensitivity analysis for minimisation of input data dimension for feedforward neural network*. IEEE International Symposium on Circuits and Systems, 1994. 1: p. 1-10.
152. Cloete, I. and A.P. Engelbrecht, *New tools for decision support*. AMSE International Conference on Intelligent Systems: Methodologies and Applications, 1994. 1: p. 1-8.
153. Viktor, H.L., A.P. Engelbrecht, and I. Cloete, *Reduction of symbolic rules from artificial neural networks using sensitivity analysis*. IEEE International Conference on Neural Networks, 1995. 4: p. 1788-1793.
154. Engelbrecht, A.P., I. Cloete, and J.M. Zurada, *Determining the significance of input parameters using sensitivity analysis*. International Workshop on Artificial Neural Networks, 1995. 930: p. 382-388.

155. Engelbrecht, A.P. and I. Cloete, *A sensitivity analysis algorithm for pruning feedforward neural networks*. IEEE International Conference on Neural Networks, 1996. 2: p. 1274-1278.
156. Hashem, S., *Sensitivity analysis for feedforward artificial neural networks with differentiable activation functions*. IEEE International Joint Conference on Neural Networks, 1992. 1: p. 419-424.
157. Crowell, R.H. and R.E. Williamson, *Taylor expansions*, in *Calculus of Vector Functions*. 1962, Prentice Hall: New Jersey, USA. p. 338-339.
158. Cibas, T., et al., *Variable selection with neural networks*, in *Neurocomputing*, R. Battiti, M. Marinaro, and P.G. Morasso, Editors. 1996, Elsevier. p. 223-248.
159. Engelbrecht, A.P., L. Fletcher, and I. Cloete. *Variance analysis of sensitivity information for pruning multilayer feedforward neural networks*. in *International Joint Conference on Neural Networks*. 1999. Washington: IEEE.
160. Finnoff, W., F. Hergert, and H.G. Zimmermann, *Improving model selection by nonconvergent methods*. Neural Networks, 1993(6): p. 771-784.
161. Fletcher, L., et al., *Optimising the number of hidden nodes of a feed forward artificial neural network*. in *IEEE international Conference on Neural Networks*. 1998. Anchorage: IEEE.
162. Maxwell, J.C., *A dynamical theory of the electromagnetic field*. Proceedings of the Royal Society, 1864. 13: p. 531-536.
163. Clebsch, A., *Ueber die Reflexion an einer Kugelflache*. J. fur Math., 1863. 61: p. 195-262.
164. Stokes, G.G., *On the dynamical theory of diffraction* Camb. Phil. Trans., 1849. 2: p. 1-34.
165. Rayleigh, L., *On the light from the sky, its polarisation and colour*, in *Philos Mag* 1871. p. 107-120, 274-279.
166. Lamb, H., *On the oscillations of a viscous spheroid* Proceedings of the Mathematical Society, 1881. 13: p. 51-66.
167. Lorenz, L., *Sur la lumiere reflechie et refractee par une sphere transparente*. Vidensk. Selsk. Skrifter, 1890. 6: p. 1-62.
168. Courant, R. and D. Hilbert, *Methods of mathematical physics*. Vol. 1. 1953, New York: John Wiley & Sons, Inc.
169. Bohren, C.F. and D.R. Huffman, *Absorption and scattering of light by small particles*. 1983, New York: John Wiley & Sons, Inc. 530.

170. Van de Hulst, H.C., *Light scattering by small particles*. 1957, New York: John Wiley & Sons.
171. Kerker, M., *The scattering of light and other electromagnetic radiation*. 1969, New York: Academic Press.
172. Wiscombe, W.J., *Improved Mie scattering algorithms*. *Applied Optics*, 1980. **19**: p. 1505-1509.
173. Palik, E.D., *Handbook of optical constants of solids*. Vol. 1. 1984, Orlando: Academic Press.
174. Palik, E.D., *Handbook of optical constants of solids*. Vol. 2. 1991, Boston: Academic Press.
175. Hsu, W.P. and E. Matijevic, *Optical properties of monodispersed hematite hydrosols*. *Applied Optics*, 1985. **24**(11): p. 1623-1630.
176. Gmelin, L., *Natrium Ergänzungsband*. Vol. 21.2. 1965, Weinheim/Bergstrasse: Verlag Chemie.
177. Gmelin, L., *Calcium*. Vol. 28.B1. 1956, Weinheim/Bergstrasse: Verlag Chemie.
178. Samsonov, G.V., *The oxide handbook*. 1973, New York: IFI/Plenum Data Corporation.
179. Hart, R.W. and E.P. Gray, *Determination of particle structure from light scattering*. *Applied Physics*, 1964. **35**: p. 1408-1415.
180. Naimimohasses, R., *et al.*, *Sensor optimisation using neural network sensitivity measures*. *Measurement Science and Technology*, 1995. **6**: p. 1291-1300.
181. Huber, E. and M. Frost, *Light scattering by small particles*. *Water SRT - Aqua*, 1998. **47**: p. 87-94.
182. Pike, E.R. and J.B. Abbiss, *Light scattering and photon correlation spectroscopy*. *High Technology*. Vol. 40. 1997: NATO Advanced Study Institute. 488.
183. Ciaccheri, L. and A.G. Mignani, *Fast turbidity measurements using a fibre optic nephelometer*. 2001, Istituto di Ricerca sulle Onde Elettromagnetiche: Florence. p. 17.
184. Bacci, C., *et al.*, *Design and construction of the fibre optic polar nephelometer*. 1998, Istituto di Ricerca sulle Onde Elettromagnetiche: Florence. p. 12.
185. *TECS Hard-Clad Fibre*. 2001, 3M.
http://multimedia.mmm.com/mws/mediawebserver.dyn?xxxxxxfDPILxR9Yxj9YxxxGggg_XXXXv-

186. *SELFOC MicroLens*. 2001, NSG America Inc.
<http://www.nsgamerica.com/sml.shtml>.
187. Ciaccheri, L. and A.G. Mignani, *LED selection and nephelometric measurements of 0-5 μm Arizona Fine particulates*. 1999, Istituto di Ricerca sulle Onde Elettromagnetiche: Florence. p. 14.
188. *IP28 IEEE488 User's Manual*. 1986, Courcouronnes: Micro-controlle Electronique. 88.
189. *Models 5207 & 5208 Lock-in Amplifiers Instruction Manual*. 1986, Princeton: Princeton Applied Research Corp.
190. *5mm ultra-bright & wide angle, round, LEDs*. 2001, RS Components Ltd.
http://www1.rswww.com/cgi-bin/bv/search/SearchDisplay.jsp?BV_SessionID=@@@@1880031062.1009582071@@@@&BV_EngineID=cadccklmfliibemgcfkmcfdhn.0&SearchType=part&constraintOID=&viewType=text&partType=stock&SearchText=2281879.
191. *GaAs/GaAlAs emitters*. 2001, RS Components Ltd.
http://www1.rswww.com/cgi-bin/bv/search/SearchDisplay.jsp?BV_SessionID=@@@@1880031062.1009582071@@@@&BV_EngineID=cadccklmfliibemgcfkmcfdhn.0&SearchType=part&constraintOID=&viewType=text&partType=stock&SearchText=2678380.
192. Schnablegger, H. and O. Glatter, *Sizing of colloidal particles with light scattering: corrections for beginning multiple scatter* *Applied Optics*, 1995. **34**(18): p. 3489-3501.
193. Foldy, L.L., *The multiple scattering of waves*. *Physical Review*, 1945. **67**(3): p. 107-119.
194. Ishimaru, A., *Correlation functions of a wave in a random distribution of stationary and moving scatterers*. *Radio Science*, 1975. **10**(1): p. 45-52.

APPENDIX B
LIST OF FIGURES AND TABLES

B.1 List of figures

Figure 1-1 – A typical scattering nephelometer	22
Figure 1-2 – Particle sizing, characterisation and surrogate techniques.....	23
Figure 2-1 – Schematic drawing of biological neurons	37
Figure 2-2 – A typical multiple-input artificial neuron.....	39
Figure 2-3 – Delay block (left) and integrator block (right)	44
Figure 2-4 – Single-layer feedforward ANN	46
Figure 2-5 – Two-input linear decision boundary ($w_1 > 0$, $w_2 > 0$, $b < 0$).....	48
Figure 2-6 – Multilayer Feedforward Artificial Neural Network (MFANN)	49
Figure 2-7 – Original Second-Order Sensitivity Analysis (SOSA) flowchart	87
Figure 2-8 – Modified Second-Order Sensitivity Analysis (SOSA) flowchart	92
Figure 2-9 – Flowchart of MFANN optimisation scheme	93
Figure 2-10 – Output of non-optimised Fourier series generator MFANN	96
Figure 2-11 – Output of optimised Fourier series generator MFANN.....	96
Figure 3-1 – Vibration ellipse	111
Figure 3-2 – Incident, internal and scattered fields.....	116
Figure 3-3 – Spherical polar coordinate system centred on a spherical particle	116
Figure 3-4 – Scattering by a spherical particle	127
Figure 3-5 – Light scattering measurement by a polar nephelometer.....	129
Figure 3-6 – Real refractive index profiles of AF compounds and water	137
Figure 3-7 – Imaginary refractive index profiles of AF compounds and water	137
Figure 3-8 – Binned size probability density function for Arizona Fine	146
Figure 3-9 – Binned volume probability density function for Arizona Fine.....	147
Figure 3-10 – Particle size probability density function for Arizona Fine.....	147
Figure 3-11 – Particle volume probability density function for Arizona Fine...	148
Figure 3-12 – Scattered irradiance per bulk mass concentration of Arizona Fine	148
Figure 3-13 – Scattered irradiance per acceptance angle for Arizona Fine.....	149
Figure 3-14 – Scattered irradiance per chamber radius for Arizona Fine	149
Figure 3-15 – Scattered irradiance variation due to Arizona Fine particle size probability density function dithering	150
Figure 3-16 – Turbidity per bulk mass concentration of Arizona Fine.....	150
Figure 3-17 – Turbidity per acceptance angle for Arizona Fine.....	151

Figure 3-18 – Turbidity per chamber radius for Arizona Fine.....	151
Figure 4-1 – Particle size probability density functions of $AF_{1,2,3}$.....	159
Figure 4-2 – Spectral intensity functions of the three nephelometric sources ...	159
Figure 4-3 – Spectral sensitivity function of the nephelometric detectors	161
Figure 4-4 – Normalised scattering profiles of $AF_{1,2,3}$ per $AF_{1,2,3}$ mass concentration (LED A)	161
Figure 4-5 – Normalised scattering profiles of $AF_{1,2,3}$ per $AF_{1,2,3}$ mass concentration (LED B)	162
Figure 4-6 – Normalised scattering profiles of $AF_{1,2,3}$ per $AF_{1,2,3}$ mass concentration (LED C).....	162
Figure 4-7 – RMS error change per MFANN optimisation step.....	165
Figure 4-8 – Absolute maximum error change per MFANN optimisation step	166
Figure 4-9 – Structurally optimal nephelometric sensor configuration	166
Figure 4-10 – Particle volume probability distribution functions of $AF_{F,C}$.....	170
Figure 4-11 – Binned particle volume probability density functions of $AF_{F,C}$... 	170
Figure 4-12 – Binned particle size probability density functions of $AF_{F,C}$.....	171
Figure 4-13 – Particle size probability density functions of $AF_{F,C}$.....	171
Figure 4-14 – Particle volume probability density functions of $AF_{F,C}$.....	172
Figure 4-15 – Fibre optic polar nephelometer (front view)	174
Figure 4-16 – Fibre optic polar nephelometer (side view)	174
Figure 4-17 – Spectral intensities of the three nephelometric illumination systems	176
Figure 4-18 – Spectral sensitivity of the nephelometric detection system	176
Figure 4-19 – Raw experimental scattering profiles of $AF_{F,C}$ (green LED)	181
Figure 4-20 – Raw experimental scattering profiles of $AF_{F,C}$ (red LED)	181
Figure 4-21 – Raw experimental scattering profiles of $AF_{F,C}$ (IR LED)	182
Figure 4-22 – Processed experimental scattering profiles of $AF_{F,C}$ (green LED, $\theta_{ref} = 15^\circ$).....	185
Figure 4-23 – Processed experimental scattering profiles of $AF_{F,C}$ (red LED, θ_{ref} $= 15^\circ$)	185
Figure 4-24 – Processed experimental scattering profiles of $AF_{F,C}$ (IR LED, $\theta_{ref} =$ 15°)	186
Figure 4-25 – Experimental scattering vs. AF_F mass concentration (green LED, $\theta_{ref} = 15^\circ$).....	186

Figure 4-26 – Experimental scattering vs. AF_C mass concentration (green LED, $\theta_{ref} = 15^\circ$)	187
Figure 4-27 – Experimental scattering vs. AF_F mass concentration (red LED, $\theta_{ref} = 15^\circ$)	187
Figure 4-28 – Experimental scattering vs. AF_C mass concentration (red LED, $\theta_{ref} = 15^\circ$)	188
Figure 4-29 – Experimental scattering vs. AF_F mass concentration (IR LED, $\theta_{ref} = 15^\circ$)	188
Figure 4-30 – Experimental scattering vs. AF_C mass concentration (IR LED, $\theta_{ref} = 15^\circ$)	189
Figure 4-31 – Theoretical scattering profiles of AF_{F,C} (green LED)	193
Figure 4-32 – Theoretical scattering profiles of AF_{F,C} (red LED)	193
Figure 4-33 – Theoretical scattering profiles of AF_{F,C} (IR LED)	194
Figure 4-34 – Normalised scattering profiles of AF_{F,C} per AF_{F,C} mass concentration (green LED)	194
Figure 4-35 – Normalised scattering profiles of AF_{F,C} per AF_{F,C} mass concentration (red LED)	195
Figure 4-36 – Normalised scattering profiles of AF_{F,C} per AF_{F,C} mass concentration (IR LED)	195
Figure 4-37 – Limited particle volume probability distribution functions of AF_{F,C}	199
Figure 4-38 – Limited particle size probability density functions of AF_{F,C}	199
Figure 4-39 – Theoretical scattering profiles of suspended AF_{F,C} (green LED)	200
Figure 4-40 – Theoretical scattering profiles of suspended AF_{F,C} (red LED)	200
Figure 4-41 – Theoretical scattering profiles of suspended AF_{F,C} (IR LED)	201
Figure 4-42 – Normalised scattering profiles of suspended AF_{F,C} per AF_{F,C} mass concentration (green LED)	201
Figure 4-43 – Normalised scattering profiles of suspended AF_{F,C} per AF_{F,C} mass concentration (red LED)	202
Figure 4-44 – Normalised scattering profiles of suspended AF_{F,C} per AF_{F,C} mass concentration (IR LED)	202
Figure 4-45 – Relative classification error change per MFANN optimisation step	205
Figure 4-46 – Structurally optimal nephelometric sensor configuration	206

Figure 4-47 – RMS error change per optimisation step (first MFANN).....	210
Figure 4-48 – Absolute maximum error change per optimisation step (first MFANN)	210
Figure 4-49 – Relative classification error change per optimisation step (second MFANN)	213
Figure 4-50 – Structurally optimal nephelometric sensor configuration	213

B.2 List of tables

Table 1-1 – Principles of sedimentation techniques	6
Table 1-2 – Stream scanning methods	8
Table 1-3 – Comparison of formazine-based turbidity units	11
Table 2-1 – Common transfer functions of artificial neurons	40
Table 2-2 – Typical feedback connectivity patterns	45
Table 2-3 – Derivatives of common transfer functions	74
Table 2-4 – Floating-point requirements for MFANN sensitivity calculation	76
Table 2-5 – Fourier series coefficients vs. optimised MFANN parameters.....	95
Table 2-6 – Optimisation results for Fourier series generator MFANN	95
Table 2-7 – MFANN training and optimisation results (Case 1)	99
Table 2-8 – MFANN training and optimisation results (Case 2)	99
Table 3-1 – Chemical content of Arizona Fine dust	138
Table 3-2 – Particle size and volume fraction details of Arizona Fine dust.....	138
Table 4-1 – MFANN training and optimisation results	165
Table 4-2 – Mass concentration of $AF_{F,C}$ in experimental suspensions	172
Table 4-3 – Correlation coefficient of scattering linearity with concentration (%)	189
Table 4-4 – Correlation coefficient of theoretical and trial scattering profiles (%)	196
Table 4-5 – Relative error between theoretical and trial scattering profiles (%)	196
Table 4-6 – MFANN training and optimisation results	204
Table 4-7 – Training and optimisation results (first MFANN)	208
Table 4-8 – Training and optimisation results (second MFANN).....	212

APPENDIX C
NOTATION

C.1 Symbols

$\ \cdot\ _1$	First-order norm function
$\ \cdot\ _2$	Second-order norm function
∇_{\min}	Minimum value of performance index gradient
\mathbf{A}	Hessian matrix of a vector function at a given point
$a()$	Output of time-dependent functional block
\mathbf{a}	ANN output vector (<i>or</i> pattern)
\mathbf{A}_k	Particle volume fraction distribution generator (k_{th} version)
\mathbf{a}_l	MFANN output vector (layer l) <i>or</i> reduced MFANN input vector (l is the MFANN input layer)
\mathbf{a}_l^i	MFANN l_{th} layer vector (node i of layer l removed)
\mathbf{a}_l^{ij}	MFANN l_{th} layer vector (nodes i, j of layer l removed)
\mathbf{a}_l^{ijh}	MFANN l_{th} layer vector (nodes i, j, h of layer l removed)
\mathbf{a}_m	Reduced MFANN output vector (m is the MFANN output layer)
$\tilde{\mathbf{a}}_m$	Reduced MFANN target vector (m is the MFANN output layer)
${}^l \mathbf{a}_m^i$	MFANN m_{th} layer output vector (node i of layer l removed)
${}^l \mathbf{a}_m^{ij}$	MFANN m_{th} layer output vector (nodes i, j of layer l removed)
${}^l \mathbf{a}_m^{ijh}$	MFANN m_{th} layer output vector (nodes i, j, h of layer l removed)
a_n, b_n	Mie scattering coefficients
\mathbf{B}	Magnetic induction
b	Artificial neuron bias
\mathbf{b}	ANN bias vector
$B_{\text{em}}, B_{\text{om}}, A_{\text{em}}, A_{\text{om}}$	Vector spherical harmonics coefficients of electric field phasor infinite series
$B_i(\cdot)$	Scattered irradiance of distilled water per total emitted irradiance by the light source (light source i)
\mathbf{b}_l	MFANN bias vector (layer l)
c	Speed of light in vacuum ($3 \cdot 10^8$ m/s)

cf_{jl}	Mass concentration fraction of binned Arizona Fine sand type 1 suspended in the j^{th} sample
C_m	Bulk mass concentration of particles in a nephelometer
C_{mj}	Mass concentration of filtered Arizona Fine type j
\hat{C}_{mj}	Estimated mass concentration of filtered Arizona Fine type j
C_{mj}	Subset of MFANN raw target vectors (filtered Arizona Fine type j)
\hat{C}_{mj}	Second subset of MFANN raw input vectors (filtered Arizona Fine type j)
c_n, d_n	Mie internal coefficients
C_v	Volume concentration of particles in a nephelometer
$D(\cdot)$	Distance between a light source and the centre of an infinitesimal volume
D	Electric displacement
d	Search direction vector
dat\$	Data acquisition date
$dI_l(\cdot)$	Irradiance emitted by a light source at a single wavelength
$dI_s(\cdot)$	Total irradiance scattered by all particles in an infinitesimal volume
$d\hat{I}_s(\cdot)$	Irradiance scattered by a single spherical particle in an infinitesimal volume
d_m	Bulk mass density of particles in a nephelometer
$dN_p(\cdot)$	Total number of particles in an infinitesimal volume
d_p	Diameter of an arbitrary particle in a nephelometer
dth	Number of motor steps required for the movement of the detection arm between two adjacent scattering angles
$dV(\cdot)$	Volume of infinitesimal cylinder
$d\hat{V}_p(\cdot)$	Total volume of particles within an infinitesimal volume
$d\bar{V}_p(\cdot)$	Average volume of particles in an infinitesimal volume
dz	Height of infinitesimal cylinder
$E()$	Expected value function
E	Electric field
e	ANN error (<i>or</i> distance) vector
$\tilde{E}_{\parallel i}$	Incident electric field phasor component that is parallel to the scattering plane

$\hat{e}_{\parallel i}$	Orthogonal unit vector parallel to the scattering plane and lying in the (x, y) plane
$\tilde{E}_{\parallel s}$	Scattered electric field phasor component that is parallel to the scattering plane
$\hat{e}_{\parallel s}$	Orthogonal unit vector parallel to the scattering plane and lying in the (θ , φ) plane
$\tilde{E}_{\perp i}$	Incident electric field phasor component that is perpendicular to the scattering plane
$\hat{e}_{\perp i}$	Orthogonal unit vector perpendicular to the scattering plane and lying in the (x, y) plane
$\tilde{E}_{\perp s}$	Scattered electric field phasor component that is perpendicular to the scattering plane
$\hat{e}_{\perp s}$	Orthogonal unit vector perpendicular to the scattering plane and lying in the (θ , φ) plane
$(\mathbf{E}_1, \mathbf{H}_1)$	Internal plane wave
${}^m_l E_1^k$	MFANN m_{th} layer output prediction error (node i of layer l removed)
${}^m_l E_2^k$	MFANN m_{th} layer output prediction error (nodes i, j of layer l removed)
${}^m_l E_{3_h}^k$	MFANN m_{th} layer output prediction error (nodes i, j, h of layer l removed)
${}^m_l \text{Eavg}_1^k$	Average of MFANN m_{th} layer output prediction error (node i of layer l removed) over the MFANN pattern population
${}^m_l \text{Eavg}_2^k$	Average of MFANN m_{th} layer output prediction error (nodes i, j of layer l removed) over the MFANN pattern population
${}^m_l \text{Eavg}_{3_h}^k$	Average of MFANN m_{th} layer output prediction error (nodes i, j, h of layer l removed) over the MFANN pattern population
$E_{C_{mj}}(\cdot)$	Expected value function with respect to mass concentration (filtered Arizona Fine type j)
$(\mathbf{E}_i, \mathbf{H}_i)$	Incident plane wave
${}^m E_k$	MFANN k_{th} node (m_{th} layer) output prediction error
${}^i_m E_k^i$	MFANN k_{th} node (m_{th} layer) output prediction error (node i of layer l removed)
${}^i_m E_k^j$	MFANN k_{th} node (m_{th} layer) output prediction error (nodes i, j of layer l removed)

${}^l E_m^{yh}$	MFANN k_{th} node (m_{th} layer) output prediction error (nodes i, j, h of layer 1 removed)
Ep_{max}	Maximum number of ANN learning algorithm iterations (epochs)
$\hat{e}_r, \hat{e}_\theta, \hat{e}_\phi$	Orthogonal unit vectors of spherical coordinates
$(\mathbf{E}_s, \mathbf{H}_s)$	Scattered plane wave
$\hat{e}_x, \hat{e}_y, \hat{e}_z$	Orthogonal unit vectors of Cartesian coordinates
$E_\theta(\cdot)$	Expected value function with respect to scattering angle
$F()$	ANN performance index
$\tilde{\mathbf{F}}$	Phasor vector
$f()$	Artificial neuron transfer function
\mathbf{f}	ANN transfer function vector
$f_j(\cdot)$	First constitutive relation (current density, electric field)
$\hat{\mathbf{F}}_l$	Diagonal matrix of MFANN transfer function derivatives (layer 1)
$\ddot{\mathbf{F}}_l$	Diagonal matrix of the second-order derivatives of MFANN transfer functions (layer 1)
${}_l f(\cdot)$	MFANN transfer function (layer 1)
${}_l \dot{f}(\cdot)$	First-order derivative of MFANN transfer function (layer 1)
${}_l \ddot{f}(\cdot)$	Second-order derivative of MFANN transfer function (layer 1)
\mathbf{f}_l	MFANN transfer function vector (layer 1)
$f_M(\cdot)$	Second constitutive relation (magnetisation, magnetic field)
$f_P(\cdot)$	Third constitutive relation (electric polarisation, electric field)
$F_r(\cdot)$	MFANN prediction error function ($r = 2$ or $+\infty$)
f_s	Maximum input voltage that can be acquired
\mathbf{G}	Hessian matrix approximation
$g()$	Generating function of plane waves
\mathbf{g}	Gradient vector of a vector function at a given point
G_{ij}	Variable gain of the trans-impedance amplifier times the gain of the lock-in amplifier (light source i , filtered Arizona Fine type j)
\mathbf{H}	Hessian matrix of performance index after MFANN weight pruning or magnetic field

$\hat{H}aa_{lm}^k$	The main diagonal of the MFANN second-order sensitivity times the reduced MFANN inputs
hall	Value acquired from the first channel of the lock-in amplifier
$\hat{H}a_{lm}^k$	MFANN second-order sensitivity times reduced MFANN inputs
${}^i\hat{H}a_{lm}^k$	The i_{th} row of MFANN second-order sensitivity times the reduced MFANN inputs
\hat{H}_{lm}^k	MFANN second-order sensitivity matrix (layers l, m – output k)
HMBW _i	Half Maximum Beam Width of light source i
$h_n(\cdot)$	Spherical Bessel function of order n and the third kind (<i>or</i> Hankel function)
I	Irradiance (<i>or</i> intensity) – First Stokes parameter
I	Identity matrix
I_{0i}	Total irradiance emitted from every light source
I_i	Irradiance emitted by a polar nephelometric light source
I_{LOS}	Total scattered irradiance measured by a polar nephelometric light detector at line of sight
$I_s(\cdot)$	Total scattered irradiance measured by a polar nephelometric light detector at a given angle
$I_{sy}(\cdot)$	Scattered irradiance measured by a nephelometric detector
J	Jacobian matrix
J_F	Current density of free moving charges
$j_n(\cdot)$	Spherical Bessel function of order n and the first kind
$J_v(\cdot)$	Ordinary Bessel function of the first kind
K	Bias of acquired light scattering values
k	Wave number
$k(\cdot)$	Real wave number of plane waves propagating in the dissolver
\mathbf{k}	(complex) wave vector
\mathbf{k}'	Real part of wave vector
\mathbf{k}''	Imaginary part of wave vector
k_i	Wave number of the internal plane wave
L	Number of MFANN layers
$L(\cdot)$	Lagrangian function of the performance index change
$L(\cdot)$	Linear operator

M	Magnetisation
M	Muller matrix
m	Relative refractive index
$\tilde{\mathbf{M}}, \tilde{\mathbf{N}}$	Plane wave harmonics
$\tilde{\mathbf{M}}_{emn}, \tilde{\mathbf{N}}_{emn}$	Vector spherical harmonics (phasors)
$\tilde{\mathbf{M}}_{omn}, \tilde{\mathbf{N}}_{omn}$	
mf_k	Mass fraction of the k_{th} chemical compound of binned Arizona Fine sand
\mathbf{M}_j	Subset of MFANN raw input vectors (filtered Arizona Fine type j)
MSE_{goal}	Mean-squared error goal
N	(complex) refractive index
$N()$	Complex refractive index of the dissolver
N	ANN net input vector
n	Artificial neuron net input
$n()$	Real part of the complex refractive index of the dissolver
N_-	Number of nodes pruned per SOSA iteration
N_+	Number of MFANN nodes added between two consecutive and successful node elimination attempts
n, κ	Optical constants
$N(\mu, \sigma)$	Gaussian noise source of mean μ and standard deviation σ
N_1	Complex refractive index of the spherical particle
NA	Numerical aperture of light detector
N_{all}^-	Total number of nodes pruned from layer l after node pruning
N_{all}^+	Total number of MFANN nodes added to layers (l+1) – (m-1)
N_b	Number of Arizona Fine particle size bins
n_c	Mie series expansion termination point
N_{hd}^-	Total number of MFANN nodes pruned from all hidden layers
N_{hd}^+	Total number of MFANN nodes added to hidden nodes as node optimisation progresses from a layer to another
N_{inp}^-	Total number of MFANN nodes removed from the input layer
$N_k()$	Complex refractive index of the k_{th} chemical compound of binned Arizona Fine sand

$n_k(), \kappa_k()$	Optical constants of the k_{th} chemical compound of binned Arizona Fine sand
\mathbf{n}_l	MFANN net input vector (layer l)
N_{max}^+	Maximum number of MFANN nodes allowed for addition between two consecutive and successful node pruning attempts
nome\$	Name of file used for averaged data storage
n_{st}	Starting point of the downward recurrence calculation of the spherical Bessel function of the first kind
nstep	Number of scattering angles excluding the first angle
n_w	Real part of the complex refractive index of water
${}_q \tilde{o}_1$	Uniformly normalised MFANN output scalar (q_{th} pattern)
\tilde{o}_q	Uniformly normalised MFANN output vector (q_{th} pattern)
\mathbf{P}	Electric polarisation <i>or</i> set of MFANN raw input vectors
\mathbf{p}	ANN input vector (<i>or</i> pattern)
$\tilde{\mathbf{p}}$	Normalised ANN input vector (<i>or</i> pattern)
\mathbf{P}_2	Second set of MFANN raw input vectors
${}_{d_p} \hat{\mathbf{P}}_C$	Binned particle size probability density function of filtered Arizona Fine sand type "coarse"
${}_{d_p} p_C(\cdot)$	Particle size probability density function of filtered Arizona Fine sand type "coarse"
${}_{v_p} P_C(\cdot)$	Particle volume probability distribution function of filtered Arizona Fine sand type "coarse"
${}_{v_p} \hat{\mathbf{P}}_C$	Binned particle volume probability density function of filtered Arizona Fine sand type "coarse"
${}_{v_p} p_C(\cdot)$	Particle volume probability density function of filtered Arizona Fine sand type "coarse"
$p_{d_p}(\cdot)$	Particle diameter (<i>or</i> size) probability density function
${}_{d_p}^k p(\cdot)$	Particle size probability density function of Arizona Fine dust (k_{th} dithered version)
${}_{d_p} \hat{\mathbf{P}}_F$	Binned particle size probability density function of filtered Arizona Fine sand type "fine"
${}_{d_p} p_F(\cdot)$	Particle size probability density function of filtered Arizona Fine sand type "fine"

$v_p P_F(\cdot)$	Particle volume probability distribution function of filtered AF sand type "fine"
$v_p \hat{P}_F$	Binned particle volume probability density function of filtered Arizona Fine sand type "fine"
$v_p P_{FF}(\cdot)$	Particle volume probability density function of filtered Arizona Fine sand type "fine"
$d_p P_j^5(\cdot)$	Limited particle size probability density function (fifth pdf filtering case, filtered Arizona Fine type j)
$v_p P_j^5(\cdot)$	Limited particle volume probability distribution function (fifth pdf filtering case, filtered Arizona Fine type j)
$d_p \hat{P}_k$	Binned particle size probability density function (k_{th} dithered version)
$v_p \hat{P}_k$	Binned particle volume probability density function (k_{th} dithered version)
$d_p P_l(\cdot)$	Particle size probability density function of binned Arizona Fine sand type l
\mathbf{p}_{max}	Vector of maximum pattern values in every ANN input
\mathbf{p}_{min}	Vector of minimum pattern values in every ANN input
$P_n(\cdot)$	Legendre polynomials of the first kind and degree n
$P_n^m(\cdot)$	Legendre functions of the first kind, degree n and order m
pow	Average scattered light irradiance acquired in mV units
$\tilde{\mathbf{p}}_q$	Uniformly normalised input vector (q_{th} pattern)
$p_{v_p}(\cdot)$	Particle volume probability density function
$v_p^k p(\cdot)$	Particle volume probability density function of Arizona Fine dust (k_{th} dithered version)
Q	Number of ANN input vectors (patterns) <i>or</i> second Stokes parameter
Q_{0i}	Second Stokes parameter of light emitted by light sources
Q_i	Second Stokes parameter of light emitted by a polar nephelometric light source
R	Number of ANN inputs <i>or</i> nephelometric chamber radius
R	Remaining term of Hessian matrix approximation
$R'(\cdot)$	Distance between a light detector and the centre of an infinitesimal volume
RCE _j	Relative Classification Error (filtered Arizona Fine type j)

$\text{rec}()$	Spectral sensitivity function of a nephelometric light detector <i>or</i> spectral sensitivity function of light source <i>i</i>
$\text{rec}_n()$	Normalised spectral sensitivity function
$\text{rec}_m(\cdot)$	Normalised spectral sensitivity function of light source <i>i</i>
RMSE_i	Relative Mean-Squared Error (light source <i>i</i>)
\mathbf{S}	Time-averaged Poynting vector
$\mathbf{S}()$	Instantaneous Poynting vector
\mathbf{S}	Stokes vector
$\hat{\mathbf{S}}\mathbf{a}_{im}$	MFANN first-order sensitivity times reduced MFANN input
$\hat{\mathbf{S}}\mathbf{a}_{im}^k$	The k^{th} row of MFANN first-order sensitivity times the reduced MFANN input
$S_{ij}()$	Elements of the (far-field) Muller matrix of the Mie scattering problem
$\hat{\mathbf{S}}_y(\cdot)$	Particle scattering by Arizona Fine particles only (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
$\bar{\mathbf{S}}_y(\cdot)$	Normalised particle scattering by Arizona Fine particles only (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
$\bar{S}_y(\cdot)$	Normalised particle scattering per unit mass concentration (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
${}_i\hat{\mathbf{S}}_y^5(\cdot)$	Limited theoretical particle scattering (fifth pdf filtering case, light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
${}_i\bar{\mathbf{S}}_y^5(\cdot)$	Limited theoretical particle scattering per unit mass concentration (fifth pdf filtering case, light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
${}_i\hat{\mathbf{S}}_y(\cdot)$	Theoretical particle scattering by Arizona Fine particles only (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
${}_i\bar{\mathbf{S}}_y(\cdot)$	Theoretical normalised particle scattering by Arizona Fine particles only (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
${}_i\bar{S}_y(\cdot)$	Theoretical normalised particle scattering per unit mass concentration (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
${}_i\tilde{\mathbf{S}}_y(\cdot)$	Theoretical particle scattering per unit mass concentration (light source <i>i</i> , filtered Arizona Fine type <i>j</i>)
$\bar{S}_{ii}(\cdot)$	Normalised particle scattering per unit mass concentration (light source <i>i</i> , binned Arizona Fine sand type <i>l</i>)

S_1	Number of MFANN nodes (layer 1)
\tilde{S}_l	Marquardt sensitivity matrix (MFANN layer l)
s_1	MFANN net input sensitivity vector (layer 1)
\hat{S}_{lm}	MFANN first-order sensitivity matrix (layers l, m)
\hat{S}_{lm}^k	MFANN first-order sensitivity (row) vector (layers l, m - output k)
$\text{src}()$	Spectral intensity function of a nephelometric light source
$\text{src}_i(\cdot)$	Spectral intensity function of light source i
$\text{src}_n()$	Normalised spectral intensity function
$\text{src}_{ni}(\cdot)$	Normalised spectral intensity function of light source i
SSE_{goal}	Sum-squared error goal
T	Turbidity of suspension in a polar nephelometer
\mathbf{T}	Set of MFANN raw target vectors
t	Time variable
\mathbf{t}	ANN target vector (<i>or</i> pattern)
\mathbf{T}_2	Second set of MFANN raw target vectors
θ	Angular position of the detector at data acquisition time
t_{max}	Last scattering angle in degrees
t_{min}	First scattering angle in degrees
$\tilde{\mathbf{t}}_q$	Uniformly normalised target vector (q_{th} pattern)
U	Third Stokes parameter
$U(x, y)$	Uniform noise source that returns values in the interval [x, y]
$u()$	Input of time-dependent functional block
${}_0^1U_k$	Vector of numbers uniformly distributed in [0, 1] (k_{th} version)
V	Fourth Stokes parameter
valf	Scattered light irradiance acquired in mV units
vf_{jl}	Volume fraction of binned Arizona Fine sand type 1 suspended in the j_{th} sample
\mathbf{VF}_k	Particle volume fraction distribution (k_{th} dithered version)
\mathbf{VF}_m	Average particle volume fraction distribution
\mathbf{VF}_{max}	Vector of maximum particle volume fraction limits
\mathbf{VF}_{min}	Vector of minimum particle volume fraction limits

\mathbf{VF}_s	Particle volume fraction deviation
$V_y(\cdot)$	Average voltage returned by the lock-in amplifier (light source i, filtered Arizona Fine type j)
${}_pV_i$	Volume of an arbitrary particle (particle size bin i)
V_p	Volume of an arbitrary particle in a nephelometer
\mathbf{v}_r	Unit vector in MFANN weight space corresponding to the pruned weight
V_w	Volume of water (dissolver)
$W(\cdot)$	Net electromagnetic energy transfer rate into a volume
\mathbf{W}	ANN weight matrix
w	Diameter of a perfectly collimated cylindrical light beam
\mathbf{w}	ANN weight (row) vector
\mathbf{W}_l	MFANN weight matrix (layer l)
$\mathbf{x}_i(\cdot)$	Set of inputs available for the two-dimensional function approximation
\mathbf{X}_k^q	Vector of the first-order derivatives of kth MFANN output with respect to MFANN inputs (q is the MFANN pattern index)
$\tilde{y}_1(x_1, x_2)$	Two dimensional function to be approximated by a MFANN
$y_n(\cdot)$	Spherical Bessel function of order n and the second kind
$Y_v(\cdot)$	Ordinary Bessel function of the second kind
α	Artificial neuron output <i>or</i> ANN learning rate <i>or</i> absorption coefficient
$\alpha(\cdot)$	Absorption coefficient of the dissolver
β	Conjugate search direction scalar
β_i	Normalisation factor (light source i)
γ	Azimuth
δ	Acceptance angle of a light detector
$\Delta \mathbf{a}_l^i$	MFANN l_{th} layer vector change (node i of layer l removed)
$\Delta \mathbf{a}_l^{ij}$	MFANN l_{th} layer vector change (nodes i, j of layer l removed)
$\Delta \mathbf{a}_l^{ijh}$	MFANN l_{th} layer vector change (nodes i, j, h of layer l removed)
$\Delta \mathbf{V}_p$	Binned particle volume range vector
$\Delta \mathbf{x}$	MFANN weight change due to pruning of another weight
ϵ	(complex) permittivity of a medium
ϵ_0	Permittivity of free space

ε_1	Complex permittivity of the spherical particle
$\zeta_n()$	Riccati-Bessel function of order n and the second kind
θ	Momentum multiplication (<i>or</i> division) factor <i>or</i> scattering angle
θ_n	Angular position of a light detector in a polar nephelometer
θ_{ref}	Reference angle in manipulation of acquired measurements
$\kappa()$	Imaginary part of the complex refractive index of the dissolver
λ	Wavelength in vacuum <i>or</i> wavelength of emitted (scattered) light
μ	Momentum term <i>or</i> permeability of a medium
μ_0	Initial value of momentum term <i>or</i> permeability of free space
μ_1	Complex permittivity of the spherical particle
μ_i	Mean value of the normally distributed spectral intensity function of light source i
μ_{max}	Maximum value of momentum term
$\xi_n()$	Riccati-Bessel function of order n and the third kind
$\pi_n(), \tau_n()$	Functions of Legendre functions of the first kind (degree n , order 1) and scattering angle
ρ_F	Free charge density
$\tilde{\rho}_F$	Free charge density phasor
ρ_m	Correlation coefficient between the theoretical and experimental scattering functions vs. mass concentration
ρ_θ	Correlation coefficient between the theoretical and experimental scattering functions vs. scattering angles
σ	Conductivity
σ_i	Standard deviation of the normally distributed spectral intensity function of light source i
σ_{ij}	Average scattering cross-section of the j_{th} type Arizona Fine particles (light source i)
v	Phase velocity
${}_{lm}\phi_1^s$	Minimal significance of single l_{th} layer node pruning
${}_{lm}\phi_2^{lv}$	Minimal significance of double l_{th} layer node pruning
${}_{lm}\phi_3^{xyz}$	Minimal significance of treble l_{th} layer node pruning
Φ_{lm}^r	Significance matrix of the r removed MFANN l_{th} layer nodes

χ	Size parameter
χ_e	Electric susceptibility
χ_m	Magnetic susceptibility
$\psi_n()$	Riccati-Bessel function of order n and the first kind

C.2 Acronyms

ADALINE	Adaptive Linear Neuron
AF	Arizona Fine
ANN	Artificial Neural Network
APHA	American Public Health Association
ASBC	American Society of Brewing Chemists
ASTM	American Society for Testing and Materials
BAM	Bidirectional Associative Memory
BSB	Brain-State-in-a-Box
CGBP	Conjugate Gradient Backpropagation
CIE	Commission Internationale de L'Éclairage
CMAC	Cerebellar Model Articulation Controller
CPN	Counterpropagation Network
dB	Decibel
DE	Diatomaceous Earth
EBU	European Brewery Convention
FIR	Finite Impulse Response
FNU	Formazine Nephelometric Unit
FOMS	First-Order Multiple Scattering
FOSA	First-Order Sensitivity Analysis
GMDH	Group Method of Data Handling
IR	Infrared
ISO	International Organisation for Standardisation
JTU	Jackson Turbidity Unit
LALLS	Low Angle Laser Light Scattering
LED	Light Emitting Diode
LM	Levenberg-Marquardt

LMBP	Levenberg-Marquardt Backpropagation
LMS	Least Mean Squared
LOS	Line of Sight
MBP	Magnitude-Based Pruning
MFANN	Multilayer Feedforward Artificial Neural Network
MOBP	Momentum Backpropagation
MSE	Mean-Squared Error
NARMA	Nonlinear Autoregressive Moving Average
NTU	Nephelometric Turbidity Unit
OBD	Optimal Brain Damage
OBS	Optimal Brain Surgeon
PC	Personal Computer
PCS	Photon Correlation Spectroscopy
PDF	Probability Density Function
PPM	Parts Per Million
PPRN	Pipelined Recurrent Neural Network
RBF	Radial Basis Function
RCE	Relative Classification Error
RMS	Root Mean Squared
RMSE	Relative Mean-Squared Error
RTRN	Real-Time Recurrent Network
SDBP	Steepest Descent Backpropagation
SEM	Scanning Electron Microscopy
SI	Système Internationale
SOSA	Second Order Sensitivity Analysis
SSE	Sum Squared Error
STEM	Scanning Transmission Electron Microscopy
STM	Scanning Tunnelling Electron Microscopy
TDNN	Time-Delay Neural Network
TEM	Transmission Electron Microscopy
VF	Volume Fraction
VLBP	Variable Learning Rate Backpropagation

APPENDIX D
SOFTWARE OVERVIEW

D.1 Introduction

The algorithms described in this study have been implemented by proprietary software written in MATLAB (version 5.2) or ANSI C. The source code of that software is consisted of function or script files (*filename m* for MATLAB and *filename c* for ANSI C code) grouped in separate folders for every single application. The outcomes of a software application are stored in text (*filename.txt*) or MATLAB binary (*filename mat*) files for future reference. The following sections list the names of source code files and outcome files by application with brief explanations of the function performed at execution time by the contents of a source code file or the kind of data stored in an outcome file.

The source code of an application is executed by typing *main* (or *main2*, *main3* when applicable) followed by the required arguments (in brackets) on the MATLAB command window. Alternatively, the function *main()* described in §D.4, §D.5 and §D.6.1 can be executed by typing *main* and the required arguments separated by hyphens (-) on the operating system's command line window if the associated function *compile()* that compiles and links together all relevant pieces of source code to a single file is typed first on a MATLAB command window (the MATLAB compiler module is required for that option).

All pieces of software built for the purposes of this study and the outcomes of that software are included in the attached CD-ROM.

D.2 Fourier series generator

- *afterprn m*: Runs after a MFANN node pruning iteration is confirmed (see *optimise.m*).
- *aftertrn m*: Runs after a successful MFANN training step (see *optimise.m*).
- *aftrback.m*: Runs after a backtracking attempt to MFANN node pruning is made (see *optimise.m*).
- *aftrsens.m*: Runs after the MFANN input-output sensitivities have been calculated and saved (see *optimise.m*).
- *anntrnph m*: Responsible for the MFANN training (see *optimise m*).
- *antrnimp.m*: Updates certain variables of function *anntrnph()* with pre-selected constants or initialises the same constants.

- *backcand.m*: Runs after a successful backtracking attempt and deals with the new MFANN nodes selected for pruning (see *optimise.m*).
- *backrema.m*: Runs after an unsuccessful backtracking attempt and deals with the remaining MFANN nodes (see *optimise.m*).
- *befrback.m*: Runs before a backtracking attempt to MFANN node pruning is made (see *optimise.m*).
- *befrsens.m*: Runs before the next MFANN optimisation iteration starts (see *optimise.m*).
- *calcoeff.m*: Calculates the Fourier coefficients of the square-wave pulse train generated by the Fourier series generator.
- *candprun.m*: Deals with the MFANN nodes selected for pruning (see *optimise.m*).
- *cos_gen.m*: Calculates the sinusoidal input signals to the Fourier series generator.
- *err_avg.m*: Calculates the mean absolute MFANN output error.
- *est_rem.m*: Calculates the estimated mean absolute MFANN output error per pruning decision after each MFANN optimisation step.
- *ev_dbpr.m*: Calculates the second-order derivatives for all transfer functions of a MFANN layer for the entire MFANN input vector population.
- *ev_func.m*: Calculates the outputs of a MFANN layer for the entire MFANN input vector population.
- *ev_pr.m*: Calculates the first-order derivatives for all transfer functions of a MFANN layer for the entire MFANN input vector population.
- *ev_SSE.m*: Calculates the sum-squared error that is associated with an error matrix.
- *firsttrn.m*: Runs before the first MFANN training takes place (see *optimise.m*).
- *glob_var.m*: Defines and initialises all global variables.
- *init_F.m*: Makes a structure that describes a MFANN architecture from partial information.
- *init_Wb.m*: Initialises all MFANN weights and biases.
- *initfunc.m*: Initialises the MFANN training and optimisation variables.
- *main.m*: Implements the MFANN modelling and optimisation solution to the Fourier series generator example.

- *make_ANN m*: Creates and initialises all variables of a new MFANN.
- *mFSOsen2.m*: Calculates the MFANN first-order and second-order input sensitivities (multi-input version).
- *mfwd_prp m*: Calculates the MFANN outputs in forward mode (multi-input version).
- *mklststr.m*: Makes an integer list string from an integer vector.
- *norm_inp.m*: Normalises all MFANN input training vectors assuming normal distribution.
- *notraind.m*: Runs when the MFANN cannot be trained from the very beginning (no MFANN node pruning has been attempted yet) (see *optimise.m*).
- *optimise.m*: Trains a MFANN and prunes neurons from its specified layer up to the optimal minimum.
- *plot_err.m*: Plots the new (or updated) MFANN learning error during the MFANN training procedure.
- *plot_TO.m*: Plots the ideal and MFANN-approximated square-wave pulse train signals of the Fourier series generator.
- *pruning.m*: Prunes the set of nodes from a MFANN layer that is of the smallest significance.
- *pulsetrn m*: Generates the square-wave pulse train of the Fourier series generator.
- *remaprun m*: Deals with the remaining MFANN nodes after an unsuccessful MFANN optimisation iteration and end of pruning procedure (see *optimise.m*).
- *saveaprn.m*: Saves MFANN parameters after a successful training and pruning iteration (see *optimise.m*).
- *savesens.m*: Saves MFANN input-output sensitivity information (see *optimise.m*).
- *svprnimp m*: Updates certain variables of function *saveaprn()* with pre-selected constants or initialises the same constants.
- *svsenimp.m*: Updates certain variables of function *savesens()* with pre-selected constants or initialises the same constants.
- *tr_lmvp.m*: Implements the Levenberg-Marquardt backpropagation algorithm.
- *unorminp.m*: Uniformly normalises a set of MFANN input vectors to the interval [-0.5, 0.5].

- *wr_coef.m*: Stores the Fourier coefficients of the square-wave pulse train to a file or displays them to the standard output.
- *wr_del m*: Stores the mean absolute and the estimated mean absolute MFANN output errors to a file or displays them to the standard output.
- *wr_FWb m*: Stores the MFANN weights and biases to a file or displays them to the standard output.
- *wr_out.m*: Stores the MFANN output vectors to a file or displays them to the standard output.
- *wr_remin m*: Stores the candidate MFANN nodes for pruning to a file or displays them to the standard output.
- *wr_rslt m*: Stores the average values of MFANN first-order and second-order input sensitivities to a file or displays them to the standard output.
- *wr_SSE.m*: Stores the sum-squared errors of MFANN outputs to a file or displays them to the standard output.
- *coeffile.txt*: Lists the set of Fourier coefficients of the square-wave pulse train.
- *del_file.txt*: Lists the mean absolute and the estimated mean absolute MFANN output errors per pruning decision after each MFANN optimisation step.
- *FWb_file.txt*: Lists all MFANN weights and biases after each MFANN optimisation step.
- *out_file.txt*: Lists all MFANN output vectors after each MFANN optimisation step.
- *rem_file.txt*: Lists the candidate MFANN nodes for pruning after each MFANN optimisation step.
- *sen_file.txt*: Lists the average values of MFANN first-order and second-order input sensitivities after each MFANN optimisation step.
- *SSE_file.txt*: Lists the sum-squared errors of MFANN outputs after each MFANN optimisation step.

D.3 Multi-dimensional function generator

- *adjgoals.m*: Potentially increases the MFANN error goal after a successful MFANN training phase.
- *afbakimp.m*: Updates certain variables of function `aftrback()` with pre-selected constants or initialises the same constants.

- *afprnimp.m*: Updates certain variables of function `afterprn()` with pre-selected constants or initialises the same constants.
- *afsenimp.m*: Updates certain variables of function `aftrensens()` with pre-selected constants or initialises the same constants.
- *afterprn.m, aftertrn.m, aftrback.m*: See §D.2.
- *aftrnexp.m*: Stores certain variables of function `aftertrn()` for future use.
- *aftrnimp.m*: Updates certain variables of function `aftertrn()` with pre-selected constants or initialises the same constants.
- *aftrensens.m, antrtrnph.m*: See §D.2.
- *antrnexp.m*: Stores certain variables of function `antrtrnph()` for future use.
- *antrnimp.m*: See §D.2.
- *augment.m*: Updates all MFANN variables to include all MFANN nodes that have been pruned in successive MFANN node optimisation iterations.
- *backcand.m, backrema.m, befrback.m, befrsens.m*: See §D.2.
- *bfbakimp.m*: Updates certain variables of function `befrback()` with pre-selected constants or initialises the same constants.
- *bfsenimp.m*: Updates certain variables of function `befrsens()` with pre-selected constants or initialises the same constants.
- *bkcndimp.m*: Updates certain variables of function `backcand()` with pre-selected constants or initialises the same constants.
- *bkremimp.m*: Updates certain variables of function `backrema()` with pre-selected constants or initialises the same constants.
- *calc_TO.m*: Calculates the MFANN mean-squared error and (incrementally) the MFANN max-squared error.
- *candprun.m*: See §D.2.
- *caprnimp.m*: Updates certain variables of function `candprun()` with pre-selected constants or initialises the same constants.
- *est_rem.m, ev_dbpr.m, ev_func.m*: See §D.2.
- *ev_MSE.m*: Calculates the mean-squared error that is associated with an error matrix.
- *ev_pr.m, firsttrn.m*: See §D.2.
- *fnd_Xidx.m*: Finds the MFANN weight of the smallest saliency (see `OBS.m`).

- *ftrnimp.m*: Updates certain variables of function `firsttrn()` with pre-selected constants or initialises the same constants.
- *glob_var.m*, *init_F.m*, *init_Wb.m*, *initfunc.m*: See §D.2.
- *it_make.m*: Generates the input signals available for the multi-dimensional function generator.
- *lrn_test.m*: Separates the set of MFANN patterns in two groups (training and testing MFANN pattern sets).
- *main.m*: Implements the MFANN modelling and optimisation solution to the multi-dimensional function generator example.
- *make_ANN.m*, *mFSOsen2.m*, *mfwd_prp.m*, *notraind.m*: See §D.2.
- *notrnimp.m*: Updates certain variables of function `notraind()` with pre-selected constants or initialises the same constants.
- *OBS.m*: Implements the Optimal Brain Surgeon algorithm that eliminates MFANN weights.
- *obsimp.m*: Updates certain variables of function `OBS()` with pre-selected constants or initialises the same constants.
- *optimise.m*, *plot_err.m*, *pruning.m*: See §D.2.
- *randwblr.m*: Assigns small random values to MFANN weights and biases.
- *remaprun.m*: See §D.2.
- *rmprnimp.m*: Updates certain variables of function `remaprun()` with pre-selected constants or initialises the same constants.
- *saveaprn.m*, *savesens.m*: See §D.2.
- *shownode.m*: Displays MFANN-node-related messages to the standard output.
- *show_lr.m*: Displays the title of the next iteration of MFANN optimisation to the standard output.
- *show_wgt.m*: Displays the set of MFANN weights pruned by the last iteration of MFANN optimisation to the standard output.
- *shrink.m*: Updates all MFANN variables to exclude all MFANN nodes that have been pruned in successive MFANN node optimisation iterations.
- *svprnimp.m*, *svsenimp.m*, *tr_lmbp.m*: See §D.2.
- *unorm.m*: Uniformly normalises a given set of MFANN patterns to $[-0.5, 0.5]$.
- *wIHess.m*: Calculates the inverse matrix of the second-order derivatives of MFANN half-MSE with respect to MFANN weights (see `OBS.m`).

- *wr_err.m*: Stores the MFANN root mean-squared and root max-squared errors to a file or displays them to the standard output.
- *wr_FWb m*, *wr_out m*, *wr_remin.m*: See §D.2.
- *wr_RMSE.m*: Stores the MFANN root mean-squared error history to a file or displays them to the standard output.
- *wr_rslt.m*: See §D.2.
- *optimal.mat*: Keeps all variables of the multi-dimensional function generator example for future use.
- *er1_file.txt*: Lists the MFANN root mean-squared and root max-squared errors for the set of MFANN training patterns.
- *er2_file.txt*: Lists the MFANN root mean-squared and root max-squared errors for the set of MFANN testing patterns.
- *FWb_file.txt*: See §D.2.
- *MSE_file.txt*: Lists the MFANN root mean-squared error history.
- *out_file.txt*: See §D.2.
- *param.txt*: Holds the most important parameters of the multi-dimensional function generator example for reference purposes.
- *rem_file.txt*, *sen_file.txt*: See §D.2.

D.4 Scattering and turbidity of ISO Arizona Fine dust

- *alegendr.m*: Returns the sets of the scattering angle dependent functions $\pi()$ and $\tau()$.
- *compile.m*: Compiles main () and all relevant functions to make a single executable file out of them.
- *existfile.m*: Tells whether a file bearing a given name is on MATLAB's search path or not.
- *iso_pdf.m*: Returns the binned particle size and particle volume pdfs (typical and dithered) of ISO 12103-1 AF dust.
- *los_scatt.m*: Returns the irradiance measured by a nephelometric detector placed at line of sight.
- *main m*: Calculates the theoretical scattering and turbidity profiles of ISO 12103-1 AF dust.

- *main2.m*: Displays the theoretical scattering and turbidity profiles of ISO 12103-1 AF dust.
- *mrb1.m*: Returns the sets of the spherical Bessel functions $j()$ and the Riccati-Bessel functions $\psi()$ of the first kind.
- *mrb2.m*: Returns the sets of the spherical Bessel functions $y()$ and the Riccati-Bessel functions $\zeta()$ of the second kind.
- *mrb3.m*: Returns the sets of the spherical Bessel functions $h()$ and the Riccati-Bessel functions $\xi()$ of the third kind.
- *mscatc.m*: Returns the sets of Mie scattering coefficients a , b .
- *mu_lamda.m*: Returns the complex refractive indices of AF compounds and water at the visual and near-infrared frequency domains.
- *nephscat.m*: Returns the total irradiance received by a nephelometric detector placed at a positive angle.
- *pdf2pdf.m*: Returns the particle size and particle volume pdfs (typical and dithered) of AF dust.
- *plot_pdf.m*: Plots the (binned or regular) particle size and particle volume pdfs (typical and dithered) of AF dust.
- *plotscat.m*: Plots AF scattered irradiance vs. bulk mass concentration, acceptance angle or nephelometric chamber radius.
- *plotscdt.m*: Plots the dithering area of AF scattered irradiance due to the dithering of AF particle size pdf.
- *plotscer.m*: Plots the absolute and relative errors of all but the first AF scattered irradiance function with respect to the first function.
- *plotturb.m*: Plots AF turbidity vs. bulk mass concentration, acceptance angle or nephelometric chamber radius.
- *plrefidx.m*: Plots the real and imaginary refractive indices of water and all AF compounds vs. wavelength.
- *scrcnorm.m*: Returns the normalised product of the spectral sensitivity function by the spectral intensity function.
- *trimzero.m*: Returns the indices of all leading and trailing zero elements of a vector.
- *turbscrc.m*: Returns the normalised spectral intensity and spectral sensitivity functions.

- *Mexistfile c*: Tells whether a file bearing a given name is on the current working directory or not.
- *mstr2cstr c*: Copies a MATLAB string to a C++ string variable.
- *params.mat*: Keeps all variables of function `main()` for future use.

D.5 Binned AF sand volume fraction estimation problem

- *adjgoals.m, afbakimp.m, afprnimp m, afsenimp m*: See §D.3.
- *afterprn m, aftertrn.m, aftrback.m*: See §D.2.
- *aftrnexp.m, aftrnimp.m*: See §D.3.
- *aftrsens.m*: See §D.2.
- *alegendr.m*: See §D.4.
- *ang2idx.m*: Converts sets of scattering angles to equivalent sets of MFANN input node indices.
- *ang2msg m*: Displays the scattering angles being pruned or selected for pruning.
- *anntrnph m*: See §D.2.
- *antrnexp.m*: See §D.3.
- *antrnimp.m*: See §D.2.
- *augment.m*: See §D.3.
- *backcand.m, backrema.m*: See §D.2.
- *bar_TO.m*: Plots the current MFANN (training or testing) RMS and absolute maximum errors (or the current RCE).
- *befrback.m, befrsens.m*: See §D.2.
- *bfbakimp m, bfsenimp.m, bkcndimp.m, bkremimp.m*: See §D.3.
- *calc_inp.m*: Returns the set of uniformly normalised MFANN input vectors.
- *calc_tar.m*: Returns the set of uniformly normalised MFANN target vectors.
- *candprun m*: See §D.2.
- *caprnimp.m*: See §D.3.
- *compile.m*: See §D.4.
- *disp_msg.m*: Displays a message on a static text frame of a MATLAB figure.
- *drw_msen.m*: Plots the current nephelometric sensor configuration.
- *est_rem.m, ev_dbpr.m, ev_func.m*: See §D.2.
- *ev_MSE.m*: See §D.3.

- *ev_pr m*: See §D.2.
- *ev_RCE.m*: Returns the MFANN relative classification error for the MFANN pattern set considered.
- *existfile m*: See §D.4.
- *firsttrn.m*: See §D.2.
- *fnd_Xidx m, frtrnimp.m*: See §D.3.
- *glob_var m*: See §D.2.
- *gui.m*: Initialises the GUI that displays the status of the MFANN optimisation procedure.
- *idx2ang.m*: Converts a set of MFANN input node indices to the equivalent set of scattering angles.
- *init_F.m*: See §D.2.
- *init_fig.m*: Initialises the subplots of the GUI that displays the status of the MFANN optimisation procedure.
- *init_Wb.m, initfunc m*: See §D.2.
- *iso_pdf.m*: See §D.4.
- *iso2bpdf.m*: Returns the particle size pdfs of the three binned AF sand types.
- *loaddata.m*: Loads from file a subset of variables required by the MFANN optimisation procedure.
- *loadrslt.m*: Loads from file a subset of variables returned by the MFANN optimisation procedure.
- *lrn_test.m*: See §D.3.
- *main.m*: Calculates the scattering profiles of the three binned AF sand types for every nephelometric light source.
- *main2 m*: Implements the MFANN modelling and optimisation solution to the binned AF sand volume fraction estimation problem.
- *main3.m*: Displays the structurally optimal nephelometric sensor configuration and the MFANN output prediction error history.
- *make_ANN.m, mFSOsen2 m, mfwd_prp m*: See §D.2.
- *mrb1.m, mrb2.m, mrb3.m, mscatc.m, mu_lamda.m, nephscat.m*: See §D.4.
- *notraind m*: See §D.2.
- *notrnimp.m, OBS.m, obsimp.m*: See §D.3.
- *optimise.m*: See §D.2.

- *pdf2pdf.m*: See §D.4.
- *plot_bin.m*: Initialises a plot that displays MFANN output prediction errors per particle size bin.
- *plot_err.m*: See §D.2.
- *plot_pdf.m*: See §D.4.
- *plotbpdf.m*: Plots the particle size pdfs of the three binned AF sand types.
- *plotsrc.m*: Plots the current nephelometric sensor configuration.
- *plotspec.m*: Plots the normalised spectral intensity and spectral sensitivity functions.
- *pltnscat.m*: Plots the normalised scattering profiles of binned AF sand per unit mass concentration for a given nephelometric light source.
- *pltoptst.m*: Plots the MFANN RMS error, absolute maximum error or relative classification error history.
- *pruning.m*: See §D.2.
- *randwblr.m*: See §D.3.
- *read_err.m*: Reads the MFANN root mean-squared and absolute maximum errors from a file.
- *remaprun.m*: See §D.2.
- *rmprnimp.m*: See §D.3.
- *rowperm.m*: Returns all unique row permutations of a matrix.
- *saveaprn.m, savesens.m*: See §D.2.
- *scrcnorm.m*: See §D.4.
- *scrcspec.m*: Returns the normalised spectral intensity and spectral sensitivity functions.
- *sec2hms.m*: Converts a nonnegative number of seconds to HH:MM:SS format.
- *show_ang.m*: Displays the sensor angles or hidden MFANN nodes being present, selected for pruning or pruned.
- *show_lr.m, show_wgt.m, shrink.m*: See §D.3.
- *svprnimp.m, svsenimp.m, tr_lmbp.m*: See §D.2.
- *trimzero.m*: See §D.4.
- *unormmat.m*: Uniformly normalises a MFANN pattern matrix to [-0.5, 0.5].
- *wIHess.m*: See §D.3.

- *wr_del.m*: See §D.2.
- *wr_err.m*: Stores the MFANN root mean-squared and absolute maximum errors to a file or displays them to the standard output.
- *wr_FWb.m*, *wr_out.m*, *wr_remin.m*: See §D.2.
- *wr_RMSE.m*: See §D.3.
- *wr_rslt.m*: See §D.2.
- *wr_wgt.m*: Stores the MFANN weight or bias removed per OBS iteration to a file or displays them to the standard output.
- *Mexistfile.c*, *mstr2cstr.c*: See §D.4.
- *gui mat*: Keeps all variables required for the initialisation of the GUI that displays the status of the MFANN optimisation procedure.
- *optimal.mat*: Keeps all variables of function `main2()` for future use.
- *params mat*: See §D.4.
- *er1_file.txt*: Lists the MFANN root mean-squared and absolute maximum errors for the set of MFANN training patterns.
- *er2_file.txt*: Lists the MFANN root mean-squared and absolute maximum errors for the set of MFANN testing patterns.
- *FWb_file.txt*: See §D.2.
- *MSE_file.txt*: See §D.3.
- *OBS_file.txt*: Lists the MFANN weight or bias removed per OBS iteration and the associated saliency and training / testing RMSE values.
- *out_file.txt*, *rem_file.txt*, *sen_file.txt*: See §D.2.

D.6 Filtered AF sand type detection problem

D.6.1 Experimental results

- *alegendr.m*, *compile.m*, *existfile.m*: See §D.4.
- *expdata.m*: Returns the raw and processed experimental scattering profiles of the two filtered AF sand types.
- *fc_pdf.m*: Returns the binned particle size and volume pdfs and the particle volume probability distribution functions of the two filtered AF sand types.
- *fiso_pdf.m*: Returns the binned particle size and particle volume pdfs of the two filtered AF sand types.

- *get_cmtm.m*: Returns the mass concentrations of the filtered AF sand types and the experimental scattering angles.
- *iroescrc.m*: Returns the normalised spectral intensity and spectral sensitivity functions.
- *main.m*: Calculates the theoretical scattering profiles of the two filtered AF sand types.
- *main2.m*: Displays the theoretical and experimental scattering profiles of the two filtered AF sand types.
- *mrb1.m, mrb2.m, mrb3.m, mscatc.m, mu_lamda.m, nephscat.m*: See §D.4.
- *pdf_kind.m*: Returns the filtered AF sand type used in the experiment and all associated data stored in variables or files.
- *pdf2pdf.m, plot_pdf.m*: See §D.4.
- *plot_se.m*: Plots the normalised and averaged scattering per unit mass concentration profiles of the two filtered AF sand types.
- *plotalsc.m*: Plots the processed scattering profiles of the two filtered AF sand types.
- *plotconc.m*: Plots the experimental normalised particle scattering over mass concentration for every filtered AF sand type.
- *plotrWSC.m*: Plots the raw experimental scattering profiles of the two filtered AF sand types.
- *plotscat.m*: Plots the scattering profiles of the two filtered AF sand types for a given nephelometric light source.
- *plotspec.m*: See §D.5.
- *plotturb.m*: See §D.4.
- *pltscon.m*: Plots the experimental normalised scattering per unit mass concentration profiles of the two filtered AF sand types.
- *pltvfpdf.m*: Plots the particle volume probability distribution functions of the two filtered AF sand types.
- *randwblr.m*: See §D.3.
- *scatcomp.m*: Returns the normalised particle scattering functions of the two filtered AF sand types.
- *scrcnorm.m*: See §D.4.
- *simdata.m*: Returns the main parameters and outcomes of function `main()`.

- *trimzero m, Mexistfile.c, mstr2cstr.c*: See §D.4.
- *anninp mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types are not modified in any way).
- *anninp_l.mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\geq 0.5 \mu\text{m}$).
- *anninp_m mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\in [0.5, 5] \mu\text{m}$).
- *anninp_s.mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\leq 5 \mu\text{m}$).
- *anninpnm.mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\in [1, 5] \mu\text{m}$).
- *anninptm mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\in [2, 5] \mu\text{m}$).
- *anninpul.mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\geq 2 \mu\text{m}$).
- *anninpvl.mat*: Keeps all variables of **this section's** function `main2()` for future use (the filtered AF sand types have particles of diameters $\geq 1 \mu\text{m}$).
- *fcaf_pdf mat*: Keeps the sampled particle volume probability distribution functions of the two filtered AF sand types.
- *green led.mat*: Keeps the experimental mass concentration and scattering angle data associated with the green LED being used as light source.
- *ir led.mat*: Keeps the experimental mass concentration and scattering angle data associated with the infrared LED being used as light source.
- *led_spec.mat*: Keeps the sampled spectral intensity functions of the three experimental nephelometric illumination systems.
- *params.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types are not modified in any way).
- *params_l.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\geq 0.5 \mu\text{m}$).
- *params_m.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\in [0.5, 5] \mu\text{m}$).
- *params_s.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\leq 5 \mu\text{m}$).

- *paramsnm.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\in [1, 5] \mu\text{m}$).
- *paramstm.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\in [2, 5] \mu\text{m}$).
- *paramsul.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\geq 2 \mu\text{m}$).
- *paramsvl.mat*: Keeps all variables of function `main()` for future use (the filtered AF sand types have particles of diameters $\geq 1 \mu\text{m}$).
- *red led mat*: Keeps the experimental mass concentration and scattering angle data associated with the red LED being used as light source.

D.6.2 Filtered AF sand type detection results

D.6.2.1 First solution

- *adjgoals.m, afbakimp.m, afprnimp.m, afsenimp.m*: See §D.3.
- *afterprn.m, aftertrn.m, afrback.m*: See §D.2.
- *afrnexp.m, afrnimp m*: See §D.3.
- *afrsens m*: See §D.2.
- *ang2idx m, ang2msg.m*: See §D.5.
- *anntrnph m*: See §D.2.
- *antrnexp.m*: See §D.3.
- *antrnimp m*: See §D.2.
- *augment.m*: See §D.3.
- *backcand.m, backrema m*: See §D 2.
- *bar_TO m*: See §D.5.
- *befrback.m, befrsens m*: See §D.2.
- *bfbakimp.m, bfsenimp m, bkendimp m, bkremimp m*: See §D.3.
- *calc_inp m, calc_tar m*: See §D.5.
- *candprun.m*: See §D.2.
- *caprnimp m*: See §D.3.
- *disp_msg.m*: See §D.5.
- *est_rem m, ev_dbpr m, ev_func.m*: See §D.2.
- *ev_MSE.m*: See §D.5.

- *ev_pr.m*: See §D.2.
- *ev_RCE m*: See §D.5.
- *existfile.m*: See §D.4.
- *firsttrn.m*: See §D.2.
- *fnl_Xidx.m, frtrnimp.m*: See §D.3.
- *glob_var.m*: See §D.2.
- *gui m, idx2ang.m*: See §D.5.
- *init_F.m*: See §D.2.
- *init_fig.m*: See §D.5.
- *init_Wb m, initfunc m*: See §D.2.
- *ldoptdat m*: Loads from file a subset of variables required by the MFANN optimisation procedure.
- *loadrslt.m*: See §D.5.
- *lrrn_test.m*: See §D.3.
- *main.m*: Implements the single MFANN modelling and optimisation solution to the filtered AF sand detection problem.
- *main2 m*: Displays the structurally optimal nephelometric sensor configuration and the single MFANN output prediction error history.
- *make_ANN.m, mFSOsen2.m, mfw_d_prp m, notraind.m*: See §D.2.
- *notrnimp.m, OBS.m, obsimp m*: See §D.3.
- *optimise.m*: See §D.2.
- *pdf_kind.m*: See §D.6.1.
- *plot_bin.m*: See §D.5.
- *plot_err.m*: See §D.2.
- *plotsrcr.m, pltoptst.m*: See §D.5.
- *pruning.m*: See §D.2.
- *randwblr.m*: See §D.3.
- *read_err.m*: Reads the MFANN relative classification error from a file.
- *remaprun m*: See §D.2.
- *rmprnimp m*: See §D.3.
- *saveaprn m, savesens m*: See §D.2.
- *show_ang m*: See §D.5.

- *show_lr.m, show_wgt.m, shrink.m*: See §D.3.
- *svprnimp.m, svsenimp.m, tr_lmbp m*: See §D.2.
- *unormmat.m*: See §D.5.
- *wIHess.m*: See §D.3.
- *wr_del m*: See §D.2.
- *wr_err.m*: Stores the MFANN relative classification error to a file or displays it to the standard output.
- *wr_FWb.m, wr_out.m, wr_remin m*: See §D.2.
- *wr_RMSE.m*: See §D.3.
- *wr_rslt.m*: See §D.2.
- *wr_wgt.m*: See §D.5.
- *anninp.mat, anninp_l.mat, anninp_m.mat, anninp_s.mat, anninprn.mat, anninptm mat, anninpul.mat, anninpvl.mat*: See §D.6.1.
- *annoutm.mat*: Keeps all variables of function main() for future use (the filtered AF sand types have particles of diameters $\in [1, 5] \mu\text{m}$).
- *gui mat*: See §D.5.
- *er1_file.txt*: Lists the MFANN relative classification error for the set of MFANN training patterns.
- *er2_file.txt*: Lists the MFANN relative classification error for the set of MFANN testing patterns.
- *FWb_file txt*: See §D.2.
- *MSE_file.txt*: See §D.3.
- *out_file.txt, rem_file.txt, sen_file.txt*: See §D.2.

D.6.2.2 Second solution

D.6.2.2.1 First MFANN

- *adjgoals.m, afbakimp m, afprnimp.m, afsenimp m*: See §D.3.
- *afterprn.m, aftertrn.m, aftrback.m*: See §D.2.
- *aftrnexp.m, aftrnimp m*: See §D.3.
- *aftrsens.m*: See §D.2.
- *ang2idx.m, ang2msg m*: See §D.5.
- *anntrnph.m*: See §D.2.

- *antrnexp.m*: See §D.3.
- *antrnimp m*: See §D.2.
- *augment.m*: See §D.3.
- *backcand m, backrema.m*: See §D.2.
- *bar_TO m*: See §D.5.
- *befrback.m, befrsens m*: See §D.2.
- *bfbakimp m, bfsenimp m, bkcndimp m, bkremimp m*: See §D.3.
- *calc_inp m, calc_tar.m*: See §D.5.
- *candprun.m*: See §D.2.
- *caprnimp m*: See §D.3.
- *disp_msg m, drw_msen.m*: See §D.5.
- *est_rem m, ev_dbpr.m, ev_func.m*: See §D.2.
- *ev_MSE.m*: See §D.3.
- *ev_pr.m*: See §D.2.
- *ev_RCE m*: See §D.5.
- *existfile.m*: See §D.4.
- *firsttrn m*: See §D.2.
- *fnid_Xidx.m, frtrnimp.m*: See §D.3.
- *glob_var.m*: See §D.2.
- *gui m, idx2ang m*: See §D.5.
- *init_F.m*: See §D.2.
- *init_fig m*: See §D.5.
- *init_Wb.m, initfunc.m*: See §D.2.
- *ldoptdat.m*: See §D.6.2.1.
- *loadrslt.m*: See §D.5.
- *lrrn_test.m*: See §D.3.
- *main.m*: Implements the cascaded MFANN modelling and optimisation solution to the filtered AF sand detection problem (first MFANN considered).
- *main2.m*: Displays the intermediate nephelometric sensor configuration and the first MFANN output prediction error history.
- *make_ANN.m, mFSOsen2.m, mfwd_prp.m, notraind.m*: See §D.2.
- *notrnimp.m, OBS m, obsimp m*: See §D.3.

- *optimise.m*: See §D.2.
- *pdf_kind.m*: See §D.6.1.
- *plot_bin.m*: See §D.5.
- *plot_err.m*: See §D.2.
- *plotsrc m, pltoptst.m*: See §D.5.
- *pruning.m*: See §D.2.
- *randwblr.m*: See §D.3.
- *read_err.m*: See §D.5.
- *remaprun.m*: See §D.2.
- *rmprnimp m*: See §D.3.
- *saveaprn.m, savesens.m*: See §D.2.
- *show_ang m*: See §D.5.
- *show_lr m, show_wgt m, shrink.m*: See §D.3.
- *svprnimp.m, svsenimp m, tr_lmbp m*: See §D.2.
- *unormmat.m*: See §D.5.
- *wIHess m*: See §D.3.
- *wr_del.m*: See §D.2.
- *wr_err.m*: See §D.5.
- *wr_FWb.m, wr_out.m, wr_remin.m*: See §D.2.
- *wr_RMSE.m*: See §D.3.
- *wr_rslt.m*: See §D.2.
- *wr_wgt.m*: See §D.5.
- *anninp mat, anninp_l.mat, anninp_m mat, anninp_s mat, anninprnm mat, anninptm.mat, anninpul mat, anninpvl.mat*: See §D.6.1.
- *annmidnm.mat*: Keeps all variables of **this section's** function `main()` for future use (the filtered AF sand types have particles of diameters $\in [1, 5] \mu\text{m}$).
- *gui.mat, er1_file.txt, er2_file.txt*: See §D.5.
- *FWb_file.txt*: See §D.2.
- *MSE_file.txt*: See §D.3.
- *out_file.txt, rem_file.txt, sen_file.txt*: See §D.2.

D.6.2.2.2 Second MFANN

- *adjgoals m, afbakimp m, afrnimp m, afsenimp.m*: See §D.3.
- *afterprn m, aftertrn.m, afrback.m*: See §D.2.
- *afrnexp.m, afrnimp m*: See §D.3.
- *afrsens.m*: See §D.2.
- *ang2idx.m, ang2msg.m*: See §D.5.
- *ang2idx.m*: Converts sets of scattering angles (stored in cell arrays) to equivalent sets of MFANN input node indices.
- *anntrnph m*: See §D.2.
- *antrnexp.m*: See §D.3.
- *antrnimp.m*: See §D.2.
- *augment m*: See §D.3.
- *backcand.m, backrema m*: See §D.2.
- *bar_TO.m*: See §D.5.
- *befrback.m, befrsens m*: See §D.2.
- *bfbakimp.m, bfsenimp.m, bkcndimp.m, bkremimp m*: See §D.3.
- *calc_inp m, calc_tar.m*: See §D.5.
- *candprun.m*: See §D.2.
- *caprnimp.m*: See §D.3.
- *disp_msg.m, drw_msen m*: See §D.5.
- *est_rem m, ev_dbpr.m, ev_func.m*: See §D.2.
- *ev_MSE.m*: See §D.3.
- *ev_pr.m*: See §D.2.
- *ev_RCE.m*: See §D.5.
- *existfile.m*: See §D.4.
- *firsttrn.m*: See §D.2.
- *fnd_Xidx.m, frtrnimp.m*: See §D.3.
- *glob_var.m*: See §D.2.
- *gui.m, idx2ang m*: See §D.5.
- *init_F.m*: See §D.2.
- *init_fig.m*: See §D.5.
- *init_Wb.m, initfunc.m*: See §D.2.

- *ldoptdat.m*: See §D.6.2.1.
- *loadrslt m*: See §D.5.
- *lrn_test m*: See §D.3.
- *main.m*: Implements the cascaded MFANN modelling and optimisation solution to the filtered AF sand detection problem (second MFANN considered).
- *main2.m*: Displays the structurally optimal nephelometric sensor configuration and the second MFANN output prediction error history.
- *make_ANN.m, mFSOsen2.m, mfwd_prp.m, notraind.m*: See §D.2.
- *notrnimp m, OBS m, obsimp m*: See §D.3.
- *optimise m*: See §D.2.
- *pdf_kind.m*: See §D.6.1.
- *plot_bin.m*: See §D.5.
- *plot_err.m*: See §D.2.
- *plotsrc.m, pltoptst.m*: See §D.5.
- *pruning.m*: See §D.2.
- *randwblr.m*: See §D.3.
- *read_err.m*: See §D.6.2.1.
- *remaprun m*: See §D.2.
- *rmprnimp m*: See §D.3.
- *saveaprn.m, savesens.m*: See §D.2.
- *show_ang.m*: See §D.5.
- *show_lr.m, show_wgt m, shrink.m*: See §D.3.
- *svprnimp m, svsenimp m, tr_lmhp m*: See §D.2.
- *unormmat.m*: See §D.5.
- *wIHess.m*: See §D.3.
- *wr_del m*: See §D.2.
- *wr_err.m*: See §D.6.2.1.
- *wr_FWb.m, wr_out.m, wr_remin.m*: See §D.2.
- *wr_RMSE.m*: See §D.3.
- *wr_rslt.m*: See §D.2.
- *wr_wgt.m*: See §D.5.
- *annmidnm mat*: See §D.6.2.2.1.

- *annoutnm.mat*: See §D.6.2.1.
- *gui mat*: See §D.5.
- *er1_file.txt, er2_file txt*: See §D.6.2.1.
- *FWb_file txt*: See §D.2.
- *MSE_file txt*: See §D.3.
- *OBS_file.txt*: See §D.5.
- *out_file.txt, rem_file txt, sen_file txt*: See §D.2.

